

**UNIVERSIDAD INTERNACIONAL DE LAS
AMÉRICAS**

ESCUELA DE INGENIERÍA INFORMÁTICA

**SISTEMA WEB PARA LA GESTIÓN DE RECURSOS
HUMANOS EN LA EMPRESA QUESADA VARGAS
CONSTRUCCIONES, S.A., UBICADA EN GRECIA**

**TRABAJO FINAL DE GRADUACIÓN PARA OPTAR POR EL GRADO DE BACHILLERATO EN
INGENIERÍA EN SISTEMAS DE INFORMACIÓN**

ANDRÉS GERARDO GONZÁLEZ VILLEGAS

MAYO, 2025

Dedicatoria

Dedico este trabajo, primeramente, a Dios, por guiar mis pasos, darme fortaleza en cada etapa de mi vida y permitirme la oportunidad de crecer académica y profesionalmente.

A mi señora madre, Lilia María Villegas Rojas, a mi hermana Paola Villegas Rojas y a toda mi familia, por su apoyo incondicional y por creer siempre en mí. Ellos han sido parte fundamental de este proceso; sin ellos no lo habría logrado.

De manera especial, a dos grandes amigos, Wagner Chacón Brenes y Marlon Bonilla Bogantes, por su incondicional apoyo y bondad, que me dieron fuerza en los momentos más difíciles.

A cada uno de ustedes les doy las gracias por tanto apoyo, y con profunda gratitud les dedico este esfuerzo que culmina con este proyecto de graduación.

Agradecimientos

En primer lugar, le expreso mi más sincero agradecimiento a la directora Olda Bustillos Ortega, por su servicio en la dirección de la carrera, por su invaluable apoyo y por su ardua labor al servicio de todos los estudiantes. En especial, por su pronta y constante disposición a orientarme y darme guía cuando la necesité.

Les agradezco profundamente a todos los profesores que me acompañaron en este proceso, por su profesionalismo, sus consejos y el apoyo brindado durante todo este tiempo.

Extiendo también mi agradecimiento a todos los compañeros que fueron parte de esta etapa, con quienes compartí momentos de esfuerzo, desvelo y aprendizaje. De ellos aprendí el valor de la solidaridad y de la satisfacción de servir a los demás.

Y, por último, una vez más, gracias a Dios por la bendición de poder crecer, por darme salud, capacidad y la fuerza necesaria para alcanzar la meta.

CONTENIDO

Dedicatoria.....	2
Agradecimientos	3
Carta de aprobación del tutor	4
Carta del Lector	13
Declaración jurada del estudiante	14
Carta del filólogo	15
Carta de solicitud de defensa.....	16
CAPÍTULO I: INTRODUCCIÓN	27
Planteamiento del problema	27
Objetivos	29
Objetivo general	29
Objetivos específicos	29
Justificación.....	29
Viabilidad técnica	30
Viabilidad operativa	36
Viabilidad económica.....	37
Viabilidad legal.....	40
Proyecciones	44
El alcance del sistema	45
Alcance funcional	45
Alcance metodológico	47
Alcance tecnológico.....	48
CAPÍTULO II: MARCO REFERENCIAL.....	52
Gestión de recursos humanos.....	52

Funciones del departamento de recursos humanos	53
La tecnología en el departamento de recursos humanos	54
Leyes, normativas y regulaciones	55
Normativas laborales y regulaciones del MTSS	56
Metodologías de software	64
Metodologías tradicionales.....	64
Metodologías ágiles.....	69
Arquitectura de software.....	71
Estilos arquitectónicos comunes	71
Arquitectura del Sistema	75
Modelo en capas	75
Tecnologías utilizadas en el proyecto.....	77
Seguridad del software	84
Principios de la información.....	84
Seguridad aplicada al proyecto.....	85
CAPÍTULO III: MARCO METODOLÓGICO.....	89
Enfoques de la investigación	89
Enfoque cuantitativo.....	90
Enfoque cualitativo.....	90
Enfoque mixto	91
Enfoque de investigación seleccionado.....	92
Tipos de investigación.....	92
Investigación exploratoria.....	92
Investigación descriptiva.....	93
Investigación correlacional.....	94
Investigación explicativa	94

Tipo de investigación seleccionado	95
Fuentes de información	95
Fuentes primarias	96
Fuentes secundarias	97
Fuentes terciarias	98
Variables de investigación	99
Variable conceptual	99
Variable operacional	100
Variable instrumental	101
Cuadro de variables	101
Población	103
Muestra	103
Instrumentos de recolección de datos	104
Entrevistas	105
Encuestas	105
La observación	106
CAPÍTULO IV: ANÁLISIS DE RESULTADOS	107
Observación	107
Entrevista	108
Encuesta	112
Requerimientos del sistema	117
Requerimientos funcionales	117
Requerimientos no funcionales	119
CAPÍTULO V: PROPUESTA	120
Análisis del sistema	120
Análisis detallado del software	120

Análisis detallado del hardware.....	124
Análisis detallado de telecomunicaciones	125
Descripción detallada de herramientas técnicas para el desarrollo	127
Conocimiento del recurso humano para operar el sistema	128
Casos de uso	128
Diseño del sistema	209
Arquitectura del Sistema.....	209
Arquitectura del Software	210
Diseño de entradas.....	215
Diseño físico de la base de datos	219
Diseño de procesos	237
Diseño de salidas.....	245
Diagramas UML.....	249
Programación	256
Entradas	257
Salidas	259
Procesos	261
Validaciones	264
Módulos señalados en el alcance.....	265
Pruebas del Sistema	268
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES.....	278
Conclusiones	278
Recomendaciones	280
Referencias.....	282

Tablas

Tabla 1 Detalles de hardware de la laptop del estudiante	31
Tabla 2 Descripción técnica de los dispositivos periféricos disponibles	32
Tabla 3 Inventario de software del estudiante.....	32
Tabla 4 Comparación de requerimientos de hardware y de software	34
Tabla 5 Estimación de costos por etapa	37
Tabla 6 Costos estimados de hardware.	38
Tabla 7 Costos proyectados por software.....	38
Tabla 8 Módulos del sistema de gestión de recursos humanos	46
Tabla 9 Cálculo de importe de auxilio de cesantía.....	61
Tabla 10 Salarios mínimos por puestos laborales	62
Tabla 11 Cálculo de pago de vacaciones	63
Tabla 12 Comparación de arquitecturas de software	74
Tabla 13 Cuadro de categorías, variables e indicadores	102
Tabla 14 Requerimientos funcionales del sistema	118
Tabla 15 Requerimientos no funcionales del sistema	119
Tabla 16 Caso de uso para inicio de sesión en el sistema	129
Tabla 17 Caso de uso para la recuperación y el cambio de contraseña.....	130
Tabla 18 Caso de uso para actualizar datos de un usuario	133
Tabla 19 Caso de uso para crear un usuario del sistema	135
Tabla 20 Caso de uso para eliminar una cuenta de usuario.....	136
Tabla 21 Caso de uso para acceder a los usuarios activos	138
Tabla 22 Caso de uso para listar empleados activos	139
Tabla 23 Caso de uso para registrar un nuevo empleado	141
Tabla 24 Caso de uso para actualizar los datos de un empleado.....	143
Tabla 25 Caso de uso para eliminar un empleado.....	145
Tabla 26 Caso de uso para listar la asistencia del día	147
Tabla 27 Caso de uso para agregar un registro de asistencia	149
Tabla 28 Caso de uso para actualizar un registro de asistencia.....	151
Tabla 29 Caso de uso para eliminar un registro de asistencia.....	153
Tabla 30 Caso de uso para listar registros de horas extra	155
Tabla 31 Caso de uso para agregar un registro de horas extra	156
Tabla 32 Caso de uso actualizar registro de horas extra	158
Tabla 33 Caso de uso para eliminar un registro de horas extra.....	160

Tabla 34 Caso de uso para listar permisos de los empleados.....	162
Tabla 35 Caso de uso agregar registro de permisos solicitados	164
Tabla 36 Caso de uso para actualizar un registro de permisos.....	166
Tabla 37 Caso de uso para eliminar un registro de permisos	167
Tabla 38 Caso de uso para listar registro de incapacidades	169
Tabla 39 Caso de uso para agregar registro de incapacidades	171
Tabla 40 Caso de uso para actualizar registro de incapacidades.....	173
Tabla 41 Caso de uso eliminar un registro de incapacidades.....	174
Tabla 42 Caso de uso listar registros de liquidaciones.....	176
Tabla 43 Caso de uso agregar un registro de liquidación.....	178
Tabla 44 Caso de uso actualizar registro de liquidación	180
Tabla 45 Caso de uso eliminar registro de liquidación	182
Tabla 46 Caso de uso cantidad de días disponibles de vacaciones	184
Tabla 47 Caso de uso listar solicitudes de vacaciones	186
Tabla 48 Caso de uso agregar registro de vacaciones	187
Tabla 49 Caso de uso actualizar registro de solicitud de vacaciones	189
Tabla 50 Caso de uso para eliminar registro de vacaciones.....	191
Tabla 51 Caso de uso cálculo del aguinaldo	193
Tabla 52 Caso de uso cálculo de salario bruto	195
Tabla 53 Caso de uso cálculo de deducciones salariales.....	197
Tabla 54 Caso de uso registro y aplicación de rebajos.....	198
Tabla 55 Caso de uso realizar el cálculo de nómina	200
Tabla 56 Caso de uso generar reporte de empleados	201
Tabla 57 Caso de uso generar un reporte de historial laboral	203
Tabla 58 Caso de uso generar un reporte de beneficios del empleado.....	204
Tabla 59 Caso de uso generar reporte de nómina	206
Tabla 60 Caso de uso para auditoria del sistema.....	207
Tabla 61 Diccionario de datos tabla persona.....	221
Tabla 62 Diccionario de datos tabla teléfono celular	222
Tabla 63 Diccionario de datos tabla dirección	222
Tabla 64 Diccionario de datos tabla provincia.....	222
Tabla 65 Diccionario de datos tabla cantón	223
Tabla 66 Diccionario de datos tabla distrito.....	223
Tabla 67 Diccionario de datos tabla empleado	223

Tabla 68 Diccionario de datos de la tabla permiso	224
Tabla 69 Diccionario de datos tabla liquidación	224
Tabla 70 Diccionario de datos tabla horas extras.....	225
Tabla 71 Diccionario de datos tabla nómina línea	225
Tabla 72 Diccionario de datos tabla nómina periodo.....	226
Tabla 73 Diccionario de datos tabla nómina rebajo	226
Tabla 74 Diccionario de datos tabla aguinaldo	227
Tabla 75 Diccionario de datos tabla formación académica.....	228
Tabla 76 Diccionario de datos tabla jornada empleado	228
Tabla 77 Diccionario de datos tabla tipo jornada laboral.....	229
Tabla 78 Diccionario de datos de la tabla de asistencia	229
Tabla 79 Diccionario de datos tabla de la contacto de emergencia.....	230
Tabla 80 Diccionario de datos de la tabla historial laboral	230
Tabla 81 Diccionario de datos de la tabla puesto	231
Tabla 82 Diccionario de datos de la tabla incapacidad	231
Tabla 83 Diccionario de datos de la tabla vacaciones	232
Tabla 84 Diccionario de datos de la tabla saldo de vacaciones.....	232
Tabla 85 Diccionario de datos de la tabla tipo de identificación	233
Tabla 86 Diccionario de datos de la tabla <i>audit change</i>	233
Tabla 87 Diccionario de datos de la tabla <i>auth role</i>	234
Tabla 88 Diccionario de datos <i>auth role permission</i>	234
Tabla 89 Diccionario de datos de la tabla <i>auth permission</i>	234
Tabla 90 Diccionario de datos de la tabla <i>auth user</i>	234
Tabla 91 Diccionario de datos de la tabla <i>auth user role</i>	235
Tabla 92 Diccionario de datos de la tabla <i>auth refresh token</i>	235
Tabla 93 Diccionario de datos de la tabla <i>log evento</i>	236
Tabla 94 Diccionario de datos de la tabla tipo impuesto renta	236
Tabla 95 Caso de prueba para inicio de sesión exitosa	268
Tabla 96 Caso de prueba de inicio de sesión fallido	269
Tabla 97 Caso de prueba registrar una persona.....	270
Tabla 98 Caso de prueba registro de persona fallido	272
Tabla 99 Caso de prueba registro de asistencia.....	274
Tabla 100 Caso de prueba registro fallido de asistencia	276

Figuras

Figura 1 Modelo de desarrollo iterativo e incremental.....	47
Figura 2 Arquitectura monolítica.....	50
Figura 3 Arquitectura monolítica del prototipo	51
Figura 4 Fases del modelo cascada.....	66
Figura 5 Pruebas del modelo V.....	67
Figura 6 Ciclo de vida de <i>Scrum</i>	70
Figura 7 Ejemplo de arquitectura de microservicios	74
Figura 8 Ejemplo código HTML	78
Figura 9 Ejemplo de una regla en CSS	79
Figura 10 Principios fundamentales de la información	85
Figura 11 Partes de un JWT.....	86
Figura 12 Fuentes de información	96
Figura 13 Valores de K con su nivel de confianza.....	104
Figura 14 Errores de cálculo en la planilla	112
Figura 15 Gestión de vacaciones, incapacidades u horas extra	113
Figura 16 Comunicación de nómina.....	113
Figura 17 Gestión de vacaciones	114
Figura 18 Gestión de asistencia, incapacidades y permisos	114
Figura 19 Tiempo de respuesta en gestión de trámites	115
Figura 20 Sistema digital como mejora en la gestión de recursos humanos	115
Figura 21 Aspectos proyectados del sistema	116
Figura 22 Calificación de los procesos actuales	116
Figura 23 Arquitectura del sistema de recursos humanos.....	210
Figura 24 Arquitectura general del software del sistema.....	211
Figura 25 Arquitectura interna del software del sistema	214
Figura 26 Diseño de entrada inicio de sesión	216
Figura 27 Diseño de entrada pantalla principal del sistema	216
Figura 28 Diseño de pantalla formulario para ingresar un empleado.....	217
Figura 29 Diseño de entrada registro de asistencia.....	218
Figura 30 Diseño de pantalla solicitar un permiso	219
Figura 31 Modelo entidad-relación de la base de datos.....	220
Figura 32 Diagrama de flujo inicio de sesión del sistema	237

Figura 33 Diagrama de flujo registro de un empleado	238
Figura 34 <i>Diagrama de flujo registrar una asistencia</i>	239
Figura 35 Diagrama de flujo registrar horas extras	240
Figura 36 Diagrama de flujo registro de permisos	241
Figura 37 Diagrama de flujo registro de solicitud de vacaciones.....	242
Figura 38 Diagrama de flujo registro de incapacidades	243
Figura 39 Diagrama de flujo cálculo de nómina.....	244
Figura 40 Diseño de salida datos personales del empleado.....	245
Figura 41 Diseño de salida historial laboral de un empleado	246
Figura 42 Diseño de salida registro de asistencia semanal.....	246
Figura 43 Diseño de salida historial de permisos	247
Figura 44 Diseño de salida incapacidades registradas.....	248
Figura 45 Diseño de salida comprobante de pago	249
Figura 46 <i>Diagrama de clase del prototipo de recursos humanos</i>	250
Figura 47 Diagrama de secuencia crear un empleado	251
Figura 48 Diagrama de secuencia registrar asistencia	252
Figura 49 Diagrama de secuencia para registrar horas extra	253
Figura 50 Diagrama de secuencia para registrar un permiso	254
Figura 51 Diagrama de secuencia registrar solicitud de vacaciones	255
Figura 52 Diagrama de secuencia para cálculo de nómina	256
Figura 53 Agregar asistencia desde el controlador	257
Figura 54 Entrada de datos crear un periodo de nomina	258
Figura 55 Entrada para actualizar una asistencia.....	258
Figura 56 Salida de datos historial de incapacidades	259
Figura 57 Salida de datos comprobante de pago	260
Figura 58 Salida de datos horas extra de la semana	261
Figura 59 Proceso cálculo del rebajo de la CCSS	262
Figura 60 Proceso calcular salario promedio diario	262
Figura 61 Proceso construir un token JWT	263
Figura 62 Proceso cálculo de pago de incapacidades	263
Figura 63 <i>Validaciones a nivel de DTO</i>	264
Figura 64 Validación en clase controlador.....	265
Figura 65 Validación a nivel de servicios	265
Figura 66 Código registrar una asistencia.....	266

Figura 67 Código registrar horas extra	266
Figura 68 Código para registrar un permiso	267
Figura 69 Código para actualizar los datos de la persona.....	267

CAPÍTULO I: INTRODUCCIÓN

Planteamiento del problema

La empresa Quesada Vargas Construcciones, S.A. es una compañía familiar fundada en abril del 2018 en el cantón de Grecia. Su área de servicios es la construcción civil, principalmente en estructuras metálicas (hierro) y para ello presta sus servicios a lo largo y ancho de todo el país.

En los últimos años la empresa ha experimentado un gran crecimiento, por lo que ha aumentado no solo en personal sino también en la flota vehicular y en las herramientas, lo que le permite prestar un servicio de calidad, tanto en el ámbito público como en el privado del país. Hoy en día cuenta con 15 empleados, en las diferentes áreas que la conforman.

A pesar de su constante crecimiento e inversión en las herramientas que les agilizan el trabajo de campo, en lo administrativo no ha realizado una inversión significativa en tecnología que le permita agilizar sus funciones. Además del sistema de facturación electrónico y de Microsoft Office, carece de sistemas que les ayuden a optimizar sus funciones administrativas.

Como resultado de lo anterior, la empresa cuenta con algunos problemas que ha estado enfrentando, debido a la expansión de sus servicios y a la falta de sistemas que ayuden a realizar las tareas de manera más rápida y confiable. Entre las tareas que presentan problemas se encuentran:

Falta de estructura en la gestión de documentos de empleados. La gestión de los empleados actualmente es llevada a cabo por medio de documentación física, que carece de un sistema ordenado y estructurado, para almacenar la información y tener acceso a esta cuando es requerida. La gran cantidad de documentos físicos dificulta la administración y provoca ineficiencia y pérdida de información. Esto genera duplicidad de tareas y, ante la pérdida de documentación, genera molestia entre los empleados que deben volver a presentar sus documentos.

Registro ineficaz de la hora de entrada y de salida de los empleados. No se tiene un registro claro de la entrada y la salida de los colaboradores, y estos se realizan en papel, para que luego se transcriban en una hoja de Excel, y esto ocasiona constantes errores en la asistencia, a la vez que afecta el cálculo de la nómina de los servidores. Incluso, en algunas ocasiones, la transcripción errónea de los datos provoca que la información de un empleado se

mezcle con la de otro. Esto causa molestias en los empleados y obliga a tener que invertir tiempo valioso de trabajo en solucionar los problemas.

Registro y cálculo indebido de horas extra de los empleados. Ante la falta de un adecuado registro de las horas extra y eso provoca discrepancias con los empleados en cuanto a la cantidad y la aprobación de estas. En algunos casos se genera un pago indebido por la cantidad de horas por pagar, o por el monto de la hora extra incorrecto, debido al mal cálculo de esta, lo que obliga a revisar la situación y a repetir los cálculos para corregir errores o solucionar el problema, lo que representa una pérdida de recursos.

Control manual y duplicidad de tareas al realizar la nómina. El cálculo de la nómina se realiza de forma manual y también se utiliza Excel. Cada cálculo se elabora al menos dos veces por empleado para verificar que no presenta errores. Debido a esto, el tiempo que toma realizar el cálculo total de la planilla es del doble, o incluso hasta más si se detectan errores, y de ser así, se realiza una tercera vez para determinar el monto correcto. Esto no solo implica pérdida de tiempo y duplicidad de tareas, también genera desconfianza en la persona que las realiza, convirtiendo una tarea sencilla en una labor tediosa.

Errores y duplicidad en el cálculo de los aguinaldos. Debido a la falta de un sistema ordenado para la documentación de nómina y horas extras, se aumenta la posibilidad de errores en el cálculo del aguinaldo y se prolonga el proceso. Además, la repetición del cálculo genera una ineficacia en el proceso. Realizar el cálculo de todos los empleados es un proceso sumamente de cuidado y ante la falta de un control de registro adecuado el proceso se torna desgastante y eso conlleva más tiempo del previsto, por lo que se retrasan las demás labores de la empresa.

Falta de un sistema para que se gestionen los permisos solicitados por los empleados. Los permisos se gestionan de palabra o mediante mensaje de texto, que muchas veces son aprobados de la misma forma, sin una formalidad, y eso genera discrepancias entre la administración y los empleados, debido a la falta de un control adecuado. Se generan discrepancias en fechas, horas, horarios y tipo de permiso, lo que ocasiona molestia entre ambas partes.

Falta de registro y control de las incapacidades de los empleados. No se cuenta con un sistema adecuado que permita llevar el registro y control de las incapacidades, y la cantidad de días otorgados, lo que genera discrepancia entre la administración y los empleados; además de la excesiva documentación física que dificulta el acceso de forma rápida a la información cuando se requiere. También afecta de forma directa la nómina de los empleados, y también prolonga el cálculo de la planilla.

Falta de un sistema de registro y control de vacaciones. La falta de un sistema de registro y control de vacaciones dificulta el cálculo de los días disponibles y de los disfrutados por cada empleado, así como del pago correspondiente. Esto conduce a discrepancias entre la administración y los empleados, lo cual genera desconfianza entre las partes. Esto lleva a incurrir en una duplicidad de tareas que significan tiempo que se pierde.

Errores y duplicidad de cálculos debido a la falta de registros de las liquidaciones. Por la falta de registros organizados se generan cálculos erróneos y duplicidad en el proceso de las liquidaciones. Ante la afectación en los demás procesos y la falta de datos debido a la gran cantidad de documentación física que con el paso de los años se acumula, esta se vuelve una tarea complicada en cuanto a realizar un cálculo preciso para la liquidación de un empleado.

Objetivos

Objetivo general

Crear un sistema web de recursos humanos en la empresa Quesada Vargas Construcciones, S.A., mediante el uso de Visual Studio Code, con el lenguaje de programación Java con Spring Boot, Thymeleaf como motor de plantillas, Bootstrap, y como base de datos MySQL.

Objetivos específicos

Analizar los requerimientos funcionales y no funcionales del sistema web de recursos humanos. Diseñar la base de datos y la estructura del sistema de gestión de recursos humanos utilizando MySQL con la herramienta Workbench.

Programar los diferentes módulos del sistema de gestión de recursos humanos para que cumplan con los requerimientos mencionados, utilizando Java.

Realizar pruebas tanto individuales como de integración del sistema de recursos humanos en un entorno de prueba, de manera que se garantice la funcionalidad de cada módulo.

Justificación

En la actualidad la eficiencia en los procesos y las tareas de las empresas son de suma importancia, ya que representan una mejora en todas sus funciones de las que depende toda la organización. La transformación digital ha introducido nuevas herramientas y formas en las que las empresas realizan sus labores hoy en día, así como su personal utiliza esas herramientas para llevar a cabo sus tareas en la organización y con la velocidad que hoy cambia el mundo empresarial de la mano de la tecnología actual. Al respecto, Mochón y Cardona (2021) mencionan que “Internet y las nuevas tecnologías están cambiando la forma en que nos relacionamos, trabajamos y aprendemos. En este proceso, que se conoce como revolución digital, la velocidad del cambio es exponencial” (p.1).

Sin embargo, muchas empresas aún no han dado el paso a la digitalización. Así, por ejemplo, en el caso de la empresa Quesada Vargas Construcciones, S.A. solo se ha realizado una pequeña inversión en tecnologías digitales; esto debido a que, por desconocimiento de las tecnologías existentes en el mercado, por falta de capacitación o por miedo a realizar una inversión esas tecnologías no se ven reflejadas en sus funciones como se esperaba. Es así como la empresa realiza la mayoría de sus funciones y procesos con la ayuda de hojas de cálculo de Microsoft Excel y de manera manual con papel, lápiz y calculadora.

Por ello, el propósito del prototipo de gestión de recursos humanos es centralizar y automatizar algunos procesos, como el cálculo de la nómina, del aguinaldo, las vacaciones y las liquidaciones, entre otros, que le permitan a la empresa Quesada Vargas Construcciones, S.A., agilizar sus tareas y solucionar los problemas a que actualmente se enfrenta, debido a la falta de un control de registro adecuado de la información. Además, que le permita generar confianza en las nuevas tecnologías y fomentar la inversión en aquellas que el mercado hoy en día pone a disposición de las empresas.

Viabilidad técnica

En el presente caso se debe llevar a cabo un análisis técnico con los recursos técnicos disponibles para determinar si es posible el desarrollar del proyecto. Esto porque es necesario evaluar todos los factores técnicos que puedan influir en él, como hardware, software, personal humano, económico y legal, que permitan que se pueda llevar a cabo lo propuesto en el tiempo estimado y con los recursos disponibles. Al respecto, Portilla (2022) menciona:

Viable, según la RAE (2014), significa «condición del camino o vía por donde se puede transitar», es decir viable es aquello que tiene los indicios de las probabilidades de llevarse

a cabo. Se entienden por viabilidad de un proyecto de investigación aquellas características que deben tener todos los proyectos de investigación cuando estos son concebidos por un investigador. Los proyectos de investigación son viables cuando, por sus circunstancias, tienen probabilidades de poderse llevar a cabo. (p. 105).

El análisis técnico considerará los recursos disponibles actuales del estudiante, quien es responsable de desarrollar el prototipo de gestión de recursos humanos. Esto debido a que, como el estudiante no es empleado de la empresa, la creación del prototipo se realizará con sus propios recursos.

Requerimientos de hardware

Como se mencionó anteriormente, para el desarrollo del prototipo se utilizarán los recursos disponibles del estudiante. Este es un inventario de los componentes que el estudiante tiene a disposición para cumplir con los objetivos establecidos. Computadora: Se dispone de una laptop personal. En la siguiente tabla se muestran las especificaciones técnicas de ella.

Tabla 1

Detalles de hardware de la laptop del estudiante

Componente	Especificaciones	Descripción
Marca	Hacer	Empresa fabricante
Modelo	Nitro AN515-58	Laptop comercial tipo Gaming
Procesador	Core i5 – 12500H, 3.10 GHz, 12 núcleos y 16 hilos	12va generación familia de procesadores Intel
RAM	16 GB DDR4 (3200 MT/s)	15.7 GB usable
Arquitectura	64 bit	Basado en arquitectura x64
Almacenamiento	476.9 GB SSD	Tecnología SSD, alto rendimiento
Tarjeta Gráfica Integrada	8 GB, 1.3 MHz	Intel Iris Xe Graphics
Tarjeta Gráfica Dedicada	NVIDIA RTX 3050 Ti, 4GB	Tarjeta de video de alto rendimiento para videojuegos
Pantalla	15.6" IPS, 144 Hz	Resolución 1920 x 1080, soporta HDR
Conectividad WiFi	WiFi 6 AX160i 160MHz	Killer® Wireless Network Adapter (201NGW)
Conectividad Ethernet	Killer E2600 Gigabit	Controlador Ethernet de alta velocidad

Fuente: Elaboración propia, 2025.

En cuanto a periféricos, se cuenta con un monitor, teclado y mouse externos, los cuales facilitan la interacción con el equipo y mejoran la comodidad. El monitor externo permite trabajar con una pantalla más amplia, que optimiza la visualización de contenido. Por su parte,

el teclado y el *mouse* ofrecen una mayor ergonomía y precisión en la manipulación del sistema, especialmente en tareas que requieren alta productividad.

Tabla 2

Descripción técnica de los dispositivos periféricos disponibles

Periférico	Marca/Modelo	Descripción
Monitor	Samsung C27F390	27", 1920 x 1080, 60 Hz, HDMI, HDTV, HDCP
Teclado	Sharkoon SKILLER SGK30	Teclado mecánico con iluminación RGB y switches rojos
Mouse	Klipxtreme KBK-520	Mouse inalámbrico y DPI ajustable

Fuente: Elaboración propia, 2025.

El equipo descrito cuenta con un procesador multinúcleo de 12va generación, 16 GB de memoria RAM y una unidad de estado sólido (SSD), lo que permite un entorno de desarrollo ágil, ideal para trabajar con herramientas como Visual Studio Code, Spring Boot, MySQL Workbench y navegadores modernos. Por lo tanto, se considera que el equipo cumple adecuadamente con los requerimientos técnicos para el desarrollo y las pruebas del sistema propuesto.

Requerimientos de software

En cuanto al inventario de software se tomarán en cuenta los programas, las licencias y las herramientas de las que dispone el estudiante para llevar a cabo el prototipo. De igual manera, con lo que cuenta la empresa de lo mencionado en este apartado.

Por parte del estudiante se incluirá todo lo necesario para desarrollar el prototipo, como: entorno de desarrollo, *frameworks*, bases de datos, entre otros. En la tabla 3 se muestra la información respectiva:

Tabla 3

Inventario de Software del estudiante

Software	Nombre	Licencia	Versión	Descripción
Sistema Operativo	Windows 11 Pro	Licencia de uso para un dispositivo	24H2	Se pagó por el derecho de uso del sistema operativo. Microsoft conserva la propiedad intelectual del sistema operativo
Lenguaje de programación	Java	NFTC (Oracle No-Fee Terms)	SE 21	Otorga derechos de uso del JDK que permiten desarrollar, probar y ejecutar tus propias aplicaciones utilizando JDK. La licencia no afecta la propiedad del código creado.

Framework	Spring Boot	Apache 2.0	3.5.0	Permite su uso, modificación y distribución, siempre que se mantengan los avisos de copyright y los términos de la licencia
Entorno de Desarrollo Integrado	Visual Studio Code	MIT	1.100.2	Permite su uso, modificación y distribución. Software con soporte para múltiples lenguajes, extensiones y herramientas de depuración
Framework	Bootstrap	MIT	5.3.6	Permite su uso, modificación y distribución de manera libre, siempre que se mantengan el aviso de copyright y los términos de la licencia
Sistema Gestor de Base de Datos	MySQL	GPLv2	8	Software de código abierto que puede ser utilizado, modificado y distribuido libremente, siempre que se respeten los términos de la licencia GPL
Herramienta Gráfica	MySQL Workbench	GPLv2	8.0.36	Software de código abierto que puede ser utilizado, modificado y distribuido libremente, siempre que se respeten los términos de la licencia GPL
Motor de Plantillas	Thymeleaf	Apache 2.0	3.1.3	Permite su uso, modificación y distribución, siempre que se mantengan los avisos de copyright y los términos de la licencia

Fuente: Elaboración propia, 2025.

Nota: Toda la información presentada en la Tabla 3 fue obtenida de las páginas oficiales de cada producto.

El conjunto de herramientas descritas ha sido seleccionado por su compatibilidad, estabilidad y amplia documentación disponible. Java en su versión 21 garantiza rendimiento y soporte de largo plazo. Spring Boot 3.5.0, permite un desarrollo rápido y estructurado mediante la arquitectura MVC, con integración sencilla de la base de datos y la seguridad. MySQL, por su parte, es un sistema robusto de gestión de bases de datos relacional que facilita el manejo estructurado de la información.

Todas las herramientas seleccionadas presentan compatibilidad plena entre sí, lo que facilita la integración de componentes. El uso de Thymeleaf como motor de plantillas permite una comunicación efectiva entre la vista (HTML/Bootstrap) y el *backend* desarrollado con Spring Boot, para evitar conflictos y mejorar la mantenibilidad del sistema.

Evaluación de compatibilidad de recursos técnicos

A continuación se comparan los requisitos mínimos que requiere cada software que se utiliza para desarrollar el prototipo de gestión de recursos humanos (ver la Tabla 3), con los del

hardware disponible por el estudiante (ver Tabla 1). Esta comparación permitirá demostrar si se cuenta con los recursos técnicos adecuados para desarrollar el sistema.

Tabla 4

Comparación de requerimientos de hardware y de software

Software	Requerimientos mínimos	Hardware disponible	Cumplimiento Sí/No
Windows 11 Pro	Procesador 1 GHz, RAM 4GB, Almacenamiento 64 GB, Pantalla 9"	Procesador 3.10 GHz, RAM 16 GB, Almacenamiento 476.9 GB, Pantalla 15.6"	Sí
Java 21 SE	Procesador 2 GHz, RAM 4 GB, Almacenamiento 2GB, Arquitectura 64 Bits	Procesador 3.10 GHz, RAM 16 GB, Almacenamiento 476.9 GB, Arquitectura 64 Bit	Sí
Visual Studio Code	Procesador 1.6 GHz, RAM 1 GB, Windows 10	Procesador 3.10 GHz, RAM 16 GB, Windows 11 Pro	Sí
MySQL	Procesador 2 núcleos, RAM 2 GB, Almacenamiento 800 MB, Windows 10	Procesador 3.10 GHz, RAM 16 GB, Almacenamiento 476.9 GB, Windows 11 Pro	Sí
MySQL Workbench	Arquitectura 64 Bit, RAM 4GB, Resolución de Pantalla 1024 x 768	Arquitectura 64 Bit, RAM 16 GB, Resolución de Pantalla 1920 x 1080	Sí

Fuente: Elaboración propia, 2025.

Nota: Los requerimientos de Spring Boot, Thymeleaf y Bootstrap no corresponden debido a que no los requieren.

Con respecto a la comparación que se realizó (ver la Tabla 4), el hardware con el que se cuenta excede los requerimientos técnicos necesarios para que todo funcione adecuadamente. Esto garantiza un óptimo desempeño y estabilidad en el desarrollo de aplicaciones requeridas. Además, al contar con recursos superiores a los mínimos especificados se posibilita la ejecución eficiente de procesos más complejos sin comprometer el rendimiento

del sistema. Por lo tanto, se puede afirmar que la infraestructura tecnológica disponible es adecuada y suficiente para los propósitos establecidos.

Espacio físico

El espacio físico para realizar un trabajo, en cualquier lugar, juega un papel importante en el desempeño de la persona o las personas. El trabajar durante varias horas al día requiere que la persona se sienta cómoda, y a gusto, y que esto le permita no solo físicamente estar confortable sino también emocionalmente, para desempeñarse de manera óptima. Al respecto, García (2021) menciona:

El bienestar físico, de acuerdo con este estudio, incluye factores físicos del lugar de trabajo como acceso al movimiento, buena ergonomía, actividades y comportamientos saludables; por ejemplo, estaciones de trabajo ergonómicas, opciones de alimentos saludables, clases de yoga o gimnasio en las instalaciones. El bienestar emocional, por su parte, incluye factores que influyen en el bienestar emocional en el lugar de trabajo, y entre estos están el estado de ánimo, el estrés, la seguridad y la conexión con el aire libre. (p.62).

En concordancia con lo anterior, en la empresa se dispone de un espacio físico adecuado y bien equipado, que proporciona un conveniente espacio para trabajar. Además, de estar en una ubicación aislada, que ofrece un ambiente acogedor y que proporciona la concentración óptima para desenvolverse de la mejor manera.

Se cuenta con un escritorio amplio para trabajar y una silla lo suficientemente confortable para soportar las largas jornadas de planificación y ejecución del prototipo.

En cuanto a conectividad de internet, se dispone de una conexión de 150 Mbps de bajada y 25 Mbps de subida, suficiente para la conexión a Internet, de ser requerido.

Análisis de la viabilidad técnica de la empresa.

La empresa Quesada Vargas Construcciones, S.A. es una organización de pequeño tamaño que actualmente no cuenta con infraestructura tecnológica avanzada. Se dispone únicamente de una computadora de escritorio y una laptop, ambas con sistema operativo Windows, las cuales son utilizadas para las funciones administrativas generales, incluyendo las tareas relacionadas con la gestión del personal. No se cuenta con servidores dedicados ni con red interna, sistemas de respaldo ni personal técnico especializado en tecnologías de la

información. Se cuenta únicamente con sistema de facturación electrónica proporcionado por una empresa de software.

Ante esta realidad, el sistema propuesto ha sido diseñado bajo un enfoque técnico que se adapte a las capacidades actuales de la empresa, lo que minimiza los requerimientos de infraestructura y facilita su implementación. Por esta razón, se optó por una arquitectura monolítica desarrollada con Java y Spring Boot, haciendo uso del servidor embebido Tomcat, lo que permite emplear la aplicación directamente desde un archivo .jar, sin necesidad de configurar servidores externos.

El acceso al sistema se realiza mediante el uso de un navegador web moderno, lo que garantiza la compatibilidad con la computadora existente, sin requerir la instalación de software adicional. Este enfoque simplifica el despliegue y el mantenimiento del sistema.

El sistema gestor de bases de datos MySQL, al ser de código abierto y ampliamente documentado, puede instalarse localmente sin complicaciones, por lo que permite el control de la información sin depender de servicios externos o conexión a internet.

Esta solución busca un equilibrio óptimo entre funcionalidad, bajo costo, facilidad de uso y adaptabilidad al entorno actual de la empresa. Además, el sistema está diseñado para ser escalable, y esto permite su evolución en el futuro para soportar el crecimiento de la organización o el acceso desde múltiples dispositivos, sin necesidad de reescribirlo desde cero.

Viabilidad operativa

El objetivo principal es la creación del prototipo funcional de gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A. Como se mencionó anteriormente, por ser un proyecto universitario de graduación no se contemplan -de ser requeridos-, la capacitación del personal, el despliegue del software, ni el mantenimiento que este requiera. Por tanto, estos puntos quedan fuera del alcance del proyecto.

También se consideraron las capacidades del estudiante para saber si cuenta con los atestados para realizar el prototipo, además del tiempo disponible para realizar las diferentes etapas del proyecto. El estudiante cuenta con experiencia previa en las tecnologías y herramientas mencionadas anteriormente, que se utilizarán para el desarrollo de la propuesta; además de documentación oficial como apoyo y reutilización de código de proyecto previo propio. El estudiante dispone de tiempo completo para abocarse al cumplimiento del proyecto y cumplir con los objetivos.

El prototipo de gestión de recursos humanos será utilizado por el personal de recursos humanos, que en actualidad está saturado de tareas debido a la falta de estructura y control de registros. Ante el poco personal administrativo con el que cuenta la compañía, no se espera una reducción de personal provocado por el prototipo; por el contrario, será una herramienta más de trabajo que les beneficiará en sus labores y solucionará los problemas mencionados anteriormente.

En consecuencia, habrá un cambio total en cómo se realizan las funciones en el departamento de recursos humanos con la automatización y centralización de los procesos. Con el sistema se pretende cubrir las carencias que presenta el departamento en la actualidad y solventar errores repetitivos. Es un sistema nuevo con el que la empresa no cuenta actualmente. Mejorará los tiempos de cálculos como los de nómina, aguinaldos, liquidaciones, y disminuirá los errores y repeticiones de procesos con los que actualmente lidia el personal. También fortalecerá el control de registros y el acceso a estos en un tiempo mucho menor, lo que permitirá liberar espacio al personal para enfocarse en otras labores.

Viabilidad económica

Para el estudio de viabilidad económica se consideró el tiempo establecido por la Universidad para el desarrollo del prototipo, el cual abarca un aproximado de 88 días de trabajo. Además, se estimó una jornada de trabajo de 5 días a la semana que comprende de lunes a viernes, con una carga laboral de 8,5 horas diarias, sin contemplar los tiempos de descanso y alimentación, porque estos quedan a conciencia del estudiante.

Se planificaron el tiempo en días, que corresponde a las cuatro etapas que comprende el proyecto: requerimientos, diseño, implementación y despliegue, y el costo del programador por día, según lo estipulado por la legislación costarricense en el Código de Trabajo a partir del 01 de enero del 2024. Como el estudiante es un programador sin titulación, se tomó el monto correspondiente al puesto de programador de computación, con un salario mínimo de ₡15.983,96 diario.

A continuación se presenta la estimación de costos por etapas del proyecto:

Tabla 5

Estimación de costos por etapa

Etapa	Días Estimados	Costo
Requerimientos	10	₡159,839.60
Diseño	20	₡319,679.20

Implementación	45	₡719,278.20
Pruebas	13	₡207,791.48
Total	88	₡1,406,588.48

Fuente: Elaboración propia, 2025.

Cabe aclarar que el monto estimado en la Tabla 5 no representa ningún costo para la empresa Quesada Vargas Construcciones, S.A., ya que todos esos costos serán asumidos por el estudiante.

Costos relativos al hardware

También se proyectó el costo del hardware (ver Tabla 1), como parte de la viabilidad económica del proyecto. A continuación se muestran los costos correspondientes al hardware:

Tabla 6

Costos estimados de hardware.

Componente	Nombre	Costo
Laptop	Acer	₡0.00
Monitor	Samsung	₡0.00
Teclado	Sharkoon	₡0.00
Mouse	Klipxtreme	₡0.00
Total		₡0.00

Fuente: Elaboración propia, 2025.

El equipo utilizado para el desarrollo del prototipo es propiedad del estudiante, por lo que no representa costo alguno para la empresa.

Costos asociados de software

A continuación se muestran los costos de las herramientas y las aplicaciones de software utilizadas en el proyecto.

Tabla 7

Costos proyectados por software

Componente	Nombre	Costo
Sistema Operativo	Windows 11 Pro	₡0.00
Lenguaje de Programación	Java	₡0.00
Framework	Spring Boot	₡0.00
Entorno de Desarrollo Integrado	Visual Studio Code	₡0.00
Framework	Bootstrap	₡0.00

Sistema Gestor de Base de Datos	MySQL	₡0.00
Herramienta Gráfica	MySQL Workbench	₡0.00
Motor de Plantillas	Thymeleaf	₡0.00
Total		₡0.00

Fuente: Elaboración propia, 2025.

El software utilizado corresponde a herramientas de libre uso o licencias de uso adquiridas con la compra del equipo, que son aportadas por el estudiante; por lo tanto, la empresa no incurre en ningún tipo de gasto relacionado con este apartado.

Costos de inmobiliario

El proyecto no representa ningún costo económico en este apartado debido a que ya el estudiante cuenta con los recursos necesarios. La empresa no tiene gastos relativos a este apartado.

Costo de transporte y viáticos

El prototipo se desarrollará bajo la modalidad de trabajo remoto, y en las reuniones presenciales que se requieran tanto el costo de transporte como el costo de los viáticos corren por cuenta del estudiante. Para la empresa no representa ningún costo en este apartado.

Otros costos

El costo de los insumos como papel, impresiones, o algún otro por futuras licencias de software que se necesiten para desarrollar el prototipo, que no se estime actualmente; de requerirlos en su momento, será responsabilidad del estudiante cubrirlo. De requerirse, la empresa no tendrá que desembolsarlo en este apartado. Cabe recalcar que son costos mínimos, por lo que no incidirán de forma significativa en el desarrollo del proyecto, razón por la cual no representa un riesgo.

Riesgo por incumplimiento de tiempo de entrega

Es importante considerar la posibilidad de riesgo por incumplimiento de entrega del proyecto dentro del tiempo estimado. El tiempo estimado para la creación del sistema es de aproximadamente 18 semanas, según lo estipulado por la Universidad.

No obstante, de presentarse dicho riesgo no representará ningún costo económico para la empresa Quesada Vargas Construcciones, S.A., ya que el recurso humano, como también los demás recursos, es aportado en su totalidad por el estudiante.

Viabilidad legal

En este apartado se analiza la viabilidad del proyecto en varios puntos en los que se debe cumplir tanto con las leyes costarricenses como con las políticas internas de la empresa. Además del cumplimiento legal del estudiante en el desarrollo del prototipo, respetando tanto la confidencialidad de los datos a los que pueda tener acceso por parte de la empresa, como de las licencias utilizadas para desarrollar el prototipo.

Este análisis es de suma importancia para evitar consecuencias legales que puedan afectar al estudiante o a la empresa. Se debe respetar y regir bajo el cumplimiento de la legislación legal del país. Esto no solo por consecuencias propias del estudiante y del empleado sino también por terceros que se puedan ver afectados. Se deben respetar los principios de la investigación de no causar daño en el proceso de la investigación. Al respecto (Portilla, Como hacer un proyecto de investigación, 2022) menciona:

Se justifica legalmente una tesis cuando el investigador señala que hace su trabajo de tesis en cumplimiento de leyes existentes en un medio, puede ser de leyes generales como también de directivas más específicas, en tanto son emanadas de entidades que establecen normas o directivas con las cuales se precisan lineamientos de acciones. (pp. 102-103).

En conclusión, asegurar la viabilidad legal no es solo un requisito formal sino una responsabilidad que protege tanto al estudiante como a la empresa. Cumplir con las leyes del país y con las políticas internas garantiza que el proyecto se desarrolle de manera ética y segura.

Supuestos y limitaciones

Primeramente, se analiza cuáles son los supuestos y limitaciones del proyecto, según el análisis de los que se analizan a continuación:

Se respetará la propiedad intelectual. Supuesto, respetando los derechos de autor de terceros, fuentes, librerías, software, y cualquier otro recurso que se necesite en el prototipo. Limitaciones, el cambio de las licencias de software en el futuro podría afectar el uso del prototipo.

Cumplimiento de la legislación costarricense. Supuesto, la realización del proyecto está sujeta a las leyes del país, en el sentido de cumplir con las leyes que aplicables a él. Limitaciones, durante el tiempo del desarrollo del proyecto, de surgir un cambio en la legislación del país que tenga implicación en él se puede afectar la legalidad.

Protección de datos. Supuesto, de tener acceso a datos confidenciales se respetarán y no se divulgarán para respetar la integridad de los implicados. Limitaciones, de utilizarse datos sensibles sin un acuerdo de confidencialidad se corre el riesgo de incumplir la Ley de Protección de la Persona frente al Tratamiento de sus Datos Personales (Ley 8968).

Integridad de datos. Supuesto, se respetará la integridad de los datos, y se evitará una afectación del negocio. Limitaciones, si no se realizan pruebas en un entorno de prueba se pueden alterar registros reales de la empresa.

El sistema no será desplegado en la empresa. Supuesto, el despliegue no está contemplado como parte del proyecto. Limitaciones, en cualquier interés de desplegar el sistema en la empresa queda a disposición de ambas partes acordar los términos.

No se firmará un contrato legal. Supuesto, la empresa aceptó ser parte de la investigación; pero no hay un contrato de por medio entre el estudiante y la empresa; solo un acuerdo de colaboración informal. Limitaciones, se limitan las obligaciones de ambas partes, por lo cual si surgen malentendidos, conflictos, no están sujetos a condiciones legales jurídicas.

Funcionamiento actual de la empresa.

Se evalúa cómo opera la empresa internamente con el fin de identificar posibles incumplimientos legales que puedan surgir con el prototipo propuesto. Examinar si en el sistema se incumple con políticas, normas o procedimientos internos propios de la organización que puedan ser afectados. De ser así, se debe adaptar el software al cumplimiento interno de la empresa.

Evaluación de la empresa.

Actualmente la empresa Quesada Vargas Construcciones, S.A. no cuenta con políticas, normativas y procedimientos internos, además de las establecidas por la legislación costarricense.

Cuenta con un software de facturación electrónica contratado a un tercero, que no tiene relación con el prototipo que se va a desarrollar, por lo que no hay conflicto entre ambos.

Además, los registros se llevan en archivos físicos y en hojas de cálculo de Excel, de una licencia de Microsoft Office.

Requisitos del proyecto

Como se mencionó anteriormente, el proyecto está sujeto a la legislación del Estado costarricense. En Costa Rica se cuenta con un marco jurídico diseñado para resguardar los derechos de las personas contra delitos informáticos, con el objetivo de fortalecer las regulaciones de los sistemas informáticos y asumir la responsabilidad correspondiente por parte de los proveedores de servicios y desarrolladores de sistemas.

A continuación se detallan leyes que regulan los delitos informáticos, uso indebido de datos, protección intelectual de software y documentos digitales y tratamiento, y almacenamiento de información personal de las personas en sistemas digitales.

Ley N.º4573 del 2001, cuyo objetivo es sancionar los delitos informáticos. Esta es una de las primeras leyes que aborda tales delitos. Esta ley estableció las primeras definiciones técnicas de los delitos informáticos, como son: acceso no autorizado a sistemas informáticos, daño o alteraciones de datos o programas; fraude informático, espionaje informático y producción, importación, distribución o posesión de programas o equipos para cometer delitos informáticos. Sentó las bases para sancionar las nuevas formas de delitos que surgieron con el auge de la informática y el internet en Costa Rica. Los artículos que se agregaron a esta ley fueron precisamente los artículos de la Ley N.º8148, que es una reforma al Código Penal. Según lo estipulado en esa el proyecto no la incumple, contempla respetar los accesos relativos a la privacidad y la seguridad de estos; la manipulación de los datos para beneficio propio o de terceros, y no se modificará la integridad de ningún dato. Esto para evitar incurrir no solo en un delito sino también para no afectar la integridad de la empresa.

Ley N.º8148 Adición de los artículos 196 BIS, 217 BIS y 229 BIS al Código Penal. Esta ley introdujo nuevos artículos en el Código Penal de Costa Rica con el objetivo de sancionar penalmente las conductas que atenten la seguridad, la integridad y la confidencialidad de los

sistemas informáticos, los datos y la información digital. Los artículos agregados al Código Penal abordan lo siguiente: Artículo 196 BIS (Violación de comunicaciones electrónicas). Sanciona los accesos sin autorización a los sistemas informáticos, redes de comunicaciones o datos informáticos protegidos por mecanismos de seguridad; con pena de prisión de seis meses hasta dos años para personas ajenas a los sistemas, y una sanción de uno a tres años de prisión para personas encargadas de los soportes electrónicos, informáticos, magnéticos y telemáticos.

Artículo 217 bis (Fraude informático), en el caso de personas que, con la intención de obtener un beneficio patrimonial para sí mismo o para un tercero, influya en el procesamiento o el resultado de los datos de un sistema de cómputo, en cuyo caso se impondrá una pena de prisión de uno a diez años.

Artículo 229 bis (alteración de datos y sabotaje informático). Se impondrá una pena de prisión de uno a cuatro años a la persona que por cualquier medio accese, borre, suprima, modifique o inutilice sin autorización los datos que se encuentran registrados en una computadora.

De acuerdo con esta ley, el prototipo no infringe la ley porque no se pretende obtener un beneficio patrimonial propio a para terceros que dañe a la empresa o a las personas que la integran. El objetivo del software es resolver problemas puntuales en el departamento de recursos humanos. Todos los cálculos y procesos se harán con respeto de las respectivas leyes.

Ley de Derechos de Autor, N.º 6683 de 1982. Esta ley regula los derechos de autor de las obras literarias y artísticas, todas las producciones en los campos literario, científico y artístico, como libros, folletos, cartas y otros escritos. Además, los programas de cómputo, dentro de los cuales se incluyen sus versiones sucesivas y los programas derivados, en todo el país.

El proyecto no viola los derechos de autor intelectual de las fuentes que se utilizan ni viola las licencias de software. Se respetan los términos de uso. Tampoco se está distribuyendo un producto con derechos de autor. El prototipo es creado por el estudiante con aporte de su propio intelecto y experiencia.

Ley N.º 8968 sobre la protección de la persona frente al tratamiento de sus datos personales. El objetivo de esta ley es garantizar a cualquier persona, independientemente de su nacionalidad o condición, el respeto a sus derechos fundamentales de la autodeterminación informativa. Esto significa que las personas tienen derecho a controlar cómo se recopilan, almacenan, utilizan y divulgan sus datos personales. La ley también introduce conceptos importantes como bases de datos, datos personales, datos personales de acceso irrestricto, datos

personales de acceso restringido, datos sensibles, deber de confidencialidad, interesado, responsable de la base de datos y tratamiento de datos personales, entre otras definiciones más.

Todos estos conceptos están contenidos en el artículo 3, que corresponde a las definiciones.

En el prototipo se indicarán las medidas de seguridad necesarias para que los datos recopilados mantengan su confidencialidad. Esto además de lo referente a la integridad para que estos persistan y no se comprometa el negocio.

Proyecciones

En este apartado se tiene como objetivo presentar las proyecciones estimadas para el desarrollo del prototipo de gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A. El objetivo de las proyecciones es validar la factibilidad del proyecto y su alcance.

Con el desarrollo del prototipo se pretende resolver problemas que la empresa enfrenta actualmente. Con el sistema se pretende automatizar los procesos, disminuir los errores y reducir los tiempos en los que se realizan las tareas. Además, se implementará un control de registros que permitirá el almacenamiento adecuado de los datos y un rápido acceso a ellos, para solucionar el problema de falta de control de estos, y que, como resultado, están provocando problemas en el departamento de recursos humanos de la compañía.

Entre los beneficios proyectados están: reducción de cargas administrativas, que los procesos se realizan de manera repetitiva, como los cálculos de nómina, los aguinaldos y las vacaciones, lo que genera que los demás procesos se recarguen y provoquen atrasos en las labores. Disminución de errores. Ante la falta de un control de registros y los constantes errores que esto genera en los cálculos de la nómina, aguinaldos y liquidaciones, datos extraviados o pérdida de documentos. En este caso el sistema reducirá los errores por medio de la automatización de los procesos. Eficiencia operativa, el sistema disminuirá los tiempos en los procesos y le permitirá al personal dedicar más tiempo en las demás funciones y le permitirá el cumplimiento de los objetivos al departamento de recursos humanos. Control de registros, creará un control adecuado en función del negocio que permitirá almacenar la información organizada, segura y accesible; además de asegurar la confidencialidad e integridad de los datos. Centralización de la información, se centralizará; permitirá un adecuado tratamiento de esta y que esté al alcance del departamento en el momento en que se requiera. En cuanto a

automatización de procesos se automatizarán procesos como cálculos de nómina, aguinaldos, vacaciones y liquidaciones, lo que disminuirá los tiempos y los errores, con el objetivo de mejorar la eficiencia operativa del departamento.

El alcance del sistema

El prototipo está proyectado para el uso exclusivo en el departamento de recursos humanos de la empresa Quesada Vargas Construcciones, S.A., por lo que su alcance se limita exclusivamente a las funciones y necesidades de dicha área. No contempla la integración o vinculación con otros departamentos como contabilidad, finanzas, compras, proyectos u otros, ya que el sistema ha sido diseñado para operar de forma autónoma.

El sistema funcionará en un entorno controlado y dedicado, lo cual permite mantener la estabilidad, seguridad y simplicidad de uso, considerando la infraestructura tecnológica limitada con la que actualmente cuenta la empresa. Dado que la organización no dispone de una red empresarial robusta ni servidores dedicados, el prototipo está planteado como una solución monolítica que incluye su propio servidor embebido (Tomcat), utilizado localmente desde una computadora de escritorio.

Por otro lado, el sistema busca solucionar los problemas previamente mencionados, por lo que no abarcará otro tipo de funciones administrativas fuera de la gestión de recursos humanos.

Alcance funcional

El sistema de gestión de recursos humanos propuesto busca automatizar, centralizar y optimizar los procesos del departamento de recursos humanos de la empresa Quesada Vargas Construcciones, S.A. El diseño responde a las necesidades actuales de la organización, y ante la falta de un control adecuado de registro y tratamiento de los datos. Con el sistema se busca la reducción de errores de procesos manuales, el control adecuado de los datos, ahorro de tiempo en ejecución de tareas y la eficiencia en el cumplimiento de los objetivos del departamento de recursos humanos.

Tabla 8*Módulos del sistema de gestión de recursos humanos*

Nombre del módulo	Descripción del módulo
Gestión de Empleados	Este módulo se encargará de gestionar la información personal de todos los empleados, además de su posición laboral, incluyendo currículum, títulos o certificaciones profesionales, responsabilidades e historial de promociones. Proporcionado una estructura eficiente para el desarrollo de la gestión de los empleados.
Gestión de Asistencia	Este módulo se encargará de registrar las horas de entrada y salida de cada uno de los empleados.
Gestionar Horas Extras	Este módulo se encargará de gestionar el registro adecuado de las horas extra diarias laboradas por cada empleado. El sistema se podrá llevar el registro de las horas extras aprobadas por mutuo acuerdo de las partes o por solicitud del empleador ante la necesidad del momento.
Gestión de Nómina	Este módulo se encargará de gestionar el cálculo de nómina de cada uno de los colaboradores, e integra todas las deducciones de ley y otras establecidas por mutuo acuerdo de las partes, como ahorro voluntario, préstamos, entre otros.
Calcular aguinaldos	Este módulo se encargará de calcular los aguinaldos de los empleados de la empresa, en cumplimiento de la legislación costarricense.
Gestionar permisos	Este módulo se encargará de gestionar los permisos de los empleados, desde su solicitud hasta su aprobación y o rechazo, además de las condiciones del permiso solicitado.
Gestionar incapacidades	Este módulo se encargará de registrar de manera adecuada las incapacidades de los empleados y la distinción de estas, ya sean incapacidades por la CCSS o incapacidades por el INS, además de la cantidad de días de incapacidad otorgados al empleado.
Gestionar vacaciones	Este módulo se encargará de gestionar las vacaciones de los empleados, así como la cantidad de días disponibles del empleado, lo mismo que la cantidad de días aprobados para su disfrute. Además, se realizará el cálculo del pago correspondiente a cada día de vacaciones, para asegurar el cumplimiento de la legislación costarricense.
Gestionar Liquidaciones	Este módulo se encargará de generar el cálculo de las liquidaciones de cada colaborador al finalizar su relación laboral, para asegurar la precisión y eliminar la duplicidad y asegurar el cumplimiento de la legislación nacional.

Fuente: Elaboración propia, 2025.

Con la implementación de estos módulos la empresa podrá centralizar la información, automatizar procesos y mejorar la productividad del departamento de recursos humanos. Se promueve así el desarrollo organizacional, el bienestar de los colaboradores y el fortalecimiento de la empresa mediante la digitalización de los procesos clave del departamento de recursos humanos.

Alcance metodológico

Para el prototipo funcional del sistema de recursos humanos se utilizará la metodología de desarrollo modelo iterativo e incremental. Este modelo es una metodología cascada, pero mejorada. La diferencia de la metodología cascada es que el modelo iterativo e incremental permite a lo largo del proyecto iterar e incrementar parte de un conjunto de requerimientos y no todos a la vez, lo que permite obtener prototipos funcionales de cada conjunto de estos requerimientos. Una vez finalizados son presentados al cliente y, de dar el visto bueno, se continúa con otro conjunto de requerimientos para empezar con otro prototipo funcional. Y se repite el mismo proceso. No se obtiene todo el prototipo completo a la vez, pero sí parte de este. Al respecto, Solano y Porras (2020) mencionan:

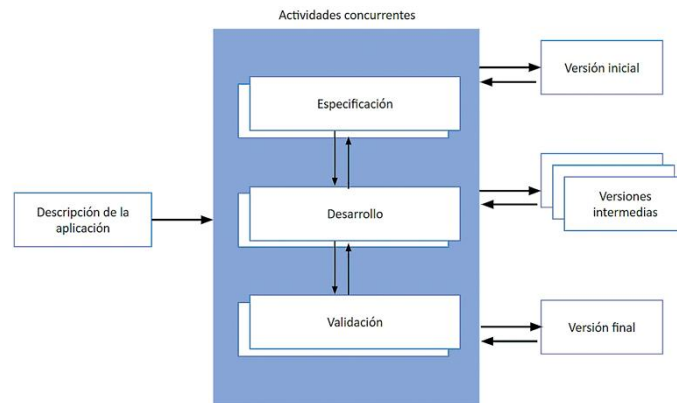
La idea detrás de la mejora iterativa sobre el modelo de cascada es tomar como punto de inicio un conjunto de requerimientos que se pueden considerar básicos y bien definidos, y, a partir de una primera versión del sistema de software, identificar mejoras en esta e incorporar otros requerimientos que se han terminado de definir con mayor claridad, o aquellos que pueden ir surgiendo durante la ejecución del proyecto. (p. 6).

Un incremento permite obtener parte de todas las funciones que se desea desarrollar, no todas. Por ejemplo, se inicia con el incremento 1. Este incremento se pasa por todas las fases del ciclo de vida de forma secuencial y al finalizar se obtiene un incremento o un prototipo. Este prototipo no incluye todos los requerimientos del proyecto sino una parte de ellos. Una vez aceptado por el cliente se toma el incremento 2 y se continúa con el mismo proceso, hasta finalizar con el incremento n. Esto agiliza el desarrollo del producto y permite integrar al cliente de forma más rápida.

Las iteraciones se aplican dentro de las mismas fases del desarrollo. Si durante la fase de diseño de un incremento se detectan necesidades de ajuste, se realizan iteraciones específicas sobre dicha fase antes de avanzar. Esto permite una mayor flexibilidad y la mejora continua dentro de cada ciclo de trabajo.

Figura 1

Modelo de desarrollo iterativo e incremental.



Fuente: Solano (2020).

Las etapas del modelo de cascada incremental combinan la estructura secuencial del modelo en cascada con la entrega progresiva. El ciclo de vida sigue las siguientes fases. En la fase de requisitos se recopilan y documentan los requerimientos iniciales del sistema. Luego, en la etapa de diseño, se crea la arquitectura del software considerando los módulos que puedan desarrollarse de forma incremental. Posteriormente, en la fase de implementación, el software se desarrolla en incrementos funcionales, cada uno pasando por las fases de diseño, codificaciones y pruebas. A continuación, en la fase de pruebas, cada incremento se evalúa individualmente y luego se integra con los anteriores para garantizar la funcionalidad del sistema completo. Durante el despliegue, el software se entrega en versiones incrementales, lo que permite mejoras y ajustes según la retroalimentación del usuario. Finalmente, en la fase de mantenimiento se realizan las correcciones y mejoras continuas en los incrementos ya entregados.

El modelo iterativo e incremental permite ofrecer versiones funcionales tempranas que pueden ser validadas directamente por el encargado de recursos humanos, lo cual facilita la comprensión del avance del sistema y asegura que las funcionalidades desarrolladas respondan a sus necesidades reales.

Alcance tecnológico

El sistema está diseñado para ser accesible mediante tecnologías web como interfaz gráfica, aunque su funcionamiento no se basa en una aplicación web desplegada en la nube, sino que opera localmente en un servidor embebido dentro del mismo sistema, lo cual significa que se ejecuta en un entorno cerrado y controlado en el equipo de la empresa. Para visualizar

y utilizar el sistema se requiere un navegador web moderno y actualizado, lo que facilita el acceso sin necesidad de instalar software adicional.

A continuación se describen las tecnologías que se van a utilizar.

Java.

Es un lenguaje de programación que permite el desarrollo de aplicaciones empresariales confiables, escalables y que se puedan ejecutar en cualquier sistema operativo hoy en día. Se utiliza en múltiples áreas de la industria moderna, desde componentes electrónicos, comercios, banca, entre muchas áreas más. Es uno de los lenguajes de programación más utilizados en el desarrollo de software en la actualidad. Java es la tecnología que se utilizará en el *backend* de la aplicación.

Spring Boot 3.5.0.

Esta herramienta permite crear aplicaciones Java que proporciona una gran cantidad de funcionalidades. Spring Boot proporciona un servidor embebido Tomcat 10.1, que permite correr nuestra aplicación sin tener que instalar un servidor por separado.

Tecnologías web frontend.

Para las interfaces gráficas de la aplicación se utilizarán tecnologías web, como HTML5, CSS, Bootstrap y Thymeleaf, que es un motor de plantillas web que se integra fácilmente con Spring Boot. Esto con la finalidad de realizar interfaces de usuario atractivas, elegantes y adaptables a los navegadores web de hoy en día.

Bases de Datos

Como sistema de gestión de bases de datos relacional, se utilizará MySQL server, con la herramienta gráfica llamada Workbench. Con este sistema de gestión y su herramienta gráfica se pueden crear la base de datos y el modelado de las tablas de nuestro proyecto.

Arquitectura de software

La arquitectura de software por la que se optó para el proyecto es monolítica, en que todas las funcionalidades se integran en un solo paquete base, para favorecer la simplicidad en la implementación. Al respecto, IBM (2023) se menciona:

Una aplicación monolítica implementa la aplicación en una sola unidad. Internamente, la lógica puede ser modular, pero externamente la aplicación está totalmente disponible o no está disponible en lo absoluto. Los monolíticos funcionan bien en comparación con los microservicios y son menos complejos a la hora de gestionar la seguridad y el contexto de transacción. El escalado de monolitos implica añadir instancias de toda la aplicación, por lo que no se pueden escalar partes individuales. (p. 3).

La arquitectura se divide en tres principales capas, las cuales se detallan a continuación.

Capa de presentación es la capa que se encarga de presentar las vistas al usuario del sistema y la interacción con él. Aquí se encuentran las vista, controladores y elementos gráficos.

Capa de lógica de negocio. Esta capa contiene la lógica de la aplicación, que se encarga de procesar las reglas de negocio y coordinar la ejecución de los servicios.

Capa acceso de datos. Se encarga de la persistencia y recuperación de los datos, e interactúa con la base de datos y otros sistemas de almacenamiento.

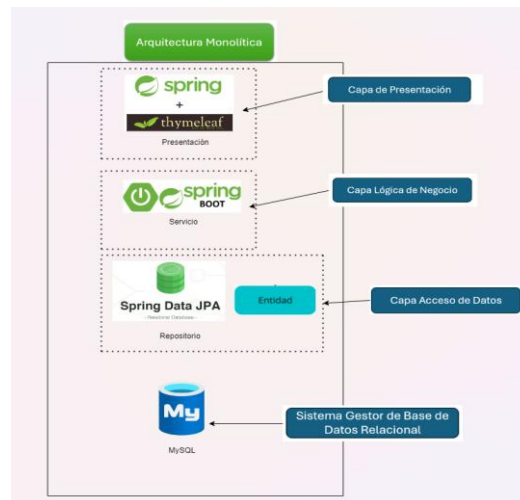
Figura 2

Arquitectura monolítica



Fuente: IBM (2023).

A continuación, en la Figura 3 se muestra la arquitectura del software que se pretende construir con las tecnologías que se van a utilizar. La arquitectura es monolítica, lo que significa que todos los componentes del sistema están integrados en una única unidad funcional. Este enfoque facilita la implementación y el despliegue, aunque puede presentar limitaciones en escalabilidad y mantenimiento.

Figura 3*Arquitectura monolítica del prototipo*

Fuente: Elaboración propia, 2025.

La arquitectura monolítica elegida para el proyecto permite una implementación más sencilla y rápida, ya que todos los componentes están integrados en un único código base. Esto facilita la creación del proyecto, su gestión y que la configuración sea más sencilla.

CAPÍTULO II: MARCO REFERENCIAL

En el presente marco referencial se establecen los fundamentos teóricos y metodológicos que sustentan esta investigación. Se abordan temas clave relacionados con la gestión del talento humano, leyes, normativas y regulaciones, es decir, las metodologías de desarrollo de software, la arquitectura tecnológica y las herramientas necesarias para la implementación del prototipo.

Gestión de recursos humanos

El departamento de recursos humanos es el encargado de gestionar o administrar todo lo relacionado con el capital humano de una empresa. Para ello asegura que los empleados tengan un ambiente laboral óptimo para cumplir con las responsabilidades asignadas por la empresa, utilizando para ello todo su potencial. Al respecto, Martín (2025) menciona que “el departamento de Recursos Humanos, también conocido o mencionado como RR. HH; mientras que *People* o gestión del capital humano es el área que administra todo lo relacionado con los empleados de una empresa” (párr. 60).

La gestión del recurso humano es un componente esencial en cualquier empresa en la actualidad, ya que busca maximizar el rendimiento y el bienestar de los colaboradores, elementos claves para alcanzar los objetivos establecidos por la organización.

La tecnología digital, utilizada como herramienta en el departamento de recursos humanos, busca alcanzar los objetivos establecidos en beneficio de la empresa y de los asociados. Al respecto, Moran *et al* (2025) mencionan:

El entorno empresarial actual se caracteriza por un ritmo acelerado de cambios tendiente a mantenerse en constantes tendencias y necesidades del personal que se involucra en ellas. A medida que las organizaciones buscan mantenerse competitivas se ven obligadas a adaptarse en un nivel operacional y estratégico ante las evoluciones, por lo que la tecnología no solo afecta la manera en que se gestiona internamente la organización, sino que tiene un impacto profundo que se evidencia desde los medios de comunicación interna, como la toma de decisiones. En este sentido, la tecnología en la era actual se vuelve esencial para comprender su aplicación e influencia en la eficacia y la eficiencia de la gestión del capital humano. (p. 2).

En conclusión, la digitalización es un pilar estratégico que potencia la capacidad y competitividad de las compañías. Con ello, la gestión del recurso humano se transforma en una ventaja eficiente que impulsa el desarrollo tanto de los colaboradores como de la organización.

Funciones del departamento de recursos humanos

El departamento de recursos humanos es muy importante para el cumplimiento de los objetivos que la organización se propone alcanzar. Las empresas, al dedicarse a diferentes servicios, exigen que el departamento se adapte al tipo de sector empresarial al que pertenecen. La empresa Quesada Vargas Construcciones, S.A. se dedica a la obra civil, principalmente de estructuras metálicas; por lo tanto, las funciones del departamento de recursos humanos están enfocadas en el área de la construcción.

Las funciones del departamento de recursos humanos abarcan diversas áreas clave para la gestión del talento dentro de la organización. La administración del personal incluye la gestión de contratos, permisos, incapacidades, vacaciones, nóminas y control de horarios, lo que asegura el cumplimiento normativo y organizacional. En cuanto al reclutamiento y la selección, este departamento se encarga de definir los requisitos del puesto y seleccionar al personal que los cumpla. Además, la gestión de recompensas y beneficios implica la administración de salarios, incentivos y prestaciones laborales, para promover la satisfacción y la retención de los empleados.

Otros aspectos fundamentales son la formación y el desarrollo, en que se diseñan programas de capacitación, tanto para nuevos empleados como para el fortalecimiento de habilidades del equipo existente. La evolución del desempeño es otro componente clave, ya que permite la creación de programas de medición y planes de mejora para optimizar el rendimiento del personal. Por último, las relaciones laborales juegan un papel esencial en la construcción de un ambiente laboral sano, basado en valores y principios de la empresa, además de brindar herramientas para la resolución de conflictos dentro del equipo de trabajo. Al respecto, Martín (2025) menciona lo que sigue:

La parte que, sin duda, más “papeleo” genera dentro de la empresa. Al mencionar la gestión y la administración del personal la referencia es a todas aquellas tareas relacionadas con nóminas, importaciones a nóminas, datos de los empleados, ausencias, vacaciones, fichajes para el control de horario, documentos y firmas digitales, gestión de turnos, partes de trabajo, y otros. (párr. 7).

El talento humano es el activo más importante de una organización, porque las personas son las que hacen funcionar todo, la tecnología, las máquinas, automotores, todo lo que la empresa necesita para un óptimo funcionamiento. Al respecto, Laradola (2023) menciona:

¿Sabías que el activo más valioso de cualquier organización eres tú? Así es, el capital humano, es decir, las habilidades, conocimientos, actitudes y experiencia de los empleados, es el factor que más influye en el éxito de una empresa. Las tendencias actuales en gestión de recursos humanos apuntan a desarrollar y potenciar el talento de las personas. (párr. 1).

En definitiva, el departamento de recursos humanos juega un papel fundamental dentro de una organización, pues garantiza el desarrollo del talento humano. Al adaptar sus funciones a las necesidades específicas de cada empresa, como en el sector de la construcción, esto se convierte en un pilar estratégico para el crecimiento y el éxito de la organización.

La tecnología en el departamento de recursos humanos

La reciente pandemia del COVID-19, que impactó a todo el mundo en cada área de la vida cotidiana de las personas, la forma de relacionarnos con nuestros familiares, amigos, colegas de trabajo, compañeros de estudio y las formas de trabajar, entre otras áreas, sumándose a la cuarentena en casi todos los países del mundo, forzó al mundo a buscar soluciones para poder seguir realizando nuestras tareas diarias.

Fue así como la implementación de la tecnología digital en las empresas se mostró como una de las soluciones más equánimes a la situación del momento, lo cual les permitió a las empresas realizar sus funciones para cumplir con sus objetivos y, a su vez, cuidar la salud de sus colaboradores. Se incorporaron soluciones del mercado digital a todos los mercados laborales para facilitarles a las empresas cumplir con sus objetivos. Al respecto, Pazmiño *et al* (2022) mencionan:

La pandemia de salud por el COVID-19 impactó tremendamente en los mercados laborales de todo el mundo, y provocó un cambio repentino en el estilo de trabajo que se desarrollaba, y por eso se debió realizar la mayoría de las labores en forma remota desde casa, de manera virtual. (p. 180).

Actualmente existen varias soluciones tecnológicas que digitalizan las funciones que realiza el departamento de recursos humanos. En el mercado se encuentran varias opciones de software especializadas en esta área. Por ejemplo, Endalia es una herramienta de gestión de recursos humanos y nómina, adaptada a las necesidades de cada empresa, que permite

automatizar los procesos. Por otro lado, Workday está diseñado para grandes empresas, ya que ofrece integración con otros sistemas internos de la empresa y proporciona herramientas para la gestión del personal y la toma de decisiones estratégicas. Finalmente, Woffu se enfoca en la gestión del tiempo y el control de horarios, lo que optimiza la planificación y el seguimiento de la jornada laboral.

Para las empresas en la actualidad es casi una “obligación” adoptar herramientas tecnológicas para mejorar la eficiencia y los resultados de sus objetivos. Al respecto, Fernández (2025) menciona:

Y como parte de la innovación deben procurar adaptarse al cambio que exige el entorno internacional y nacional. La tecnología exige estar más alertas y en muchas ocasiones, sobrepasar la capacidad de respuesta de las Pymes. Sin embargo, es imperante que las empresas procuren esta adaptación de acuerdo con sus necesidades de crecimiento, y aprovechen estas herramientas para la mejora de sus procesos internos, las necesidades cambiantes de sus clientes, y con esto aprovechar las nuevas oportunidades de negocios. (párr. 3).

Con la aparición de la inteligencia artificial muchas empresas ya están realizando una inversión importante en implementarla para beneficiarse, para impulsar sus procesos y mejorar la atención al cliente, mantener la competitividad económica, entre otras ventajas. Al respecto, SUMMA (2025) menciona lo que sigue:

De las Pymes en Costa Rica, 50% ya utilizan algún tipo de IA en sus labores diarias, lo que refleja una adopción positiva, en lo que se destacan que 48% de ellas lo usan con el objetivo de mejorar el servicio al cliente y la satisfacción de él, así como 34% lo emplean para asegurar la continuidad del negocio. (párr. 4).

La tecnología en recursos humanos ya no es solo una herramienta de soporte sino un factor importante en la estrategia empresarial. La digitalización es un factor clave en la adaptación de las empresas a los acelerados cambios globales. La evolución tecnológica continúa, y es de suma importancia que las empresas estén preparadas para afrontar los cambios que esta implica en el mercado.

Leyes, normativas y regulaciones

La gestión del talento humano y la incorporación de tecnología en el departamento de Recursos Humanos de la empresa se debe implementar bajo un marco normativo que garantice

el cumplimiento de los derechos laborales y estándares éticos. Por ello se analizarán las principales leyes, normativas y regulaciones que impactan en la digitalización de los procesos en recursos humanos.

El marco regulatorio no solo establece obligaciones para las empresas sino que también protege los derechos de los trabajadores, lo que asegura las condiciones mínimas establecidas por la legislación del estado costarricense; esto como garantías sociales, salud, información, entre otros.

Normativas laborales y regulaciones del MTSS

El Ministerio de Trabajo y Seguridad Social (MTSS) en Costa Rica es la institución pública encargada de la dirección, estudio y despacho de todos los asuntos relacionados con el trabajo y la seguridad social. En esencia, el MTSS se asegura de que se cumpla con las leyes, decretos y acuerdos relacionados con el trabajo y la seguridad social; esto al promover el trabajo digno y el desarrollo y la protección social. Al respecto, el Ministerio de Trabajo y Seguridad Social (2024) menciona:

“Somos la institución líder en el marco socio-laboral dirigida a la ciudadanía, vigilante del trabajo decente, la equidad, la justicia social, el resguardo de los derechos fundamentales del trabajo y que impulsa el cierre de brechas para la ocupación de puestos de trabajo.” (párr. 1).

Gracias a su labor, el MTSS continúa impulsando políticas y programas que favorecen la estabilidad laboral y el bienestar social en Costa Rica. Su compromiso con la equidad y la justicia social asegura que los trabajadores cuenten con condiciones dignas y oportunidades de desarrollo en el país.

Derechos de los trabajadores en Costa Rica

En Costa Rica, la legislación laboral protege los derechos fundamentales de los trabajadores al garantizar condiciones justas y equitativas en el entorno laboral. Estas normativas, establecidas en el Código de Trabajo y reguladas por el Ministerio de Trabajo y Seguridad Social, abarcan aspectos clave, como la remuneración, el tiempo de descanso y la seguridad laboral.

Entre los derechos más importantes se encuentran el aguinaldo, las vacaciones pagadas, la jornada laboral regulada, el pago de horas extra, el seguro social y la protección contra despidos injustificados, entre otros derechos.

De seguido se analizan las principales regulaciones que respaldan los derechos de los trabajadores costarricenses.

Cálculo y derechos del aguinaldo

El aguinaldo es un derecho de todos los costarricenses que consiste en una remuneración económica que reciben los empleados al final de cada año o al terminar una relación laboral. Es un derecho, sin importar el tipo de trabajo, jornada laboral y que tenga como mínimo un mes laborado para un mismo empleador de forma continua. Al respecto, el MTSS (2025) menciona que “es un “salario adicional” que debe pagar toda persona empleadora a las personas trabajadoras, cualquiera que sea su actividad, dentro de los primeros veinte días de diciembre de cada año” (párr. 10).

Las normativas correspondientes al derecho al aguinaldo corresponden a:

- La Ley N.º 2412 y sus reformas, Sueldo adicional o Ley del Aguinaldo en la Empresa Privada
- Decreto Ejecutivo N.º 20236-TSS, Reglamento a la Ley N.º 2412.

Algunos de los aspectos que la Ley y el decreto le cubren a los trabajadores son:

- El cálculo se realiza con base en todos los salarios ordinarios y extraordinarios durante los 12 meses comprendidos entre del 1º de diciembre del año anterior hasta el 30 de noviembre del año en curso.
- Para el cálculo se deben tomar en cuenta las horas ordinarias y extraordinarias, así como cualquier otro pago que el trabajador reciba durante ese tiempo.
- El tiempo en que un colaborador esté incapacitado en sus funciones no se toma en cuenta para el cálculo del aguinaldo, esto debido a que la remuneración que recibe durante ese tiempo no es un salario sino un subsidio.
- Al aguinaldo solo se le puede deducir el monto por pensión alimentaria; no se le puede aplicar ninguna otra deducción, ni por cargas sociales.
- La persona no pierde el derecho al aguinaldo porque este está vigente mientras el contrato esté vigente.

- De terminar antes del mes de diciembre, de igual manera el colaborador recibe un aguinaldo; pero para el cálculo se toman los salarios recibidos hasta la fecha de finalización del contrato, divididos entre 12.

El aguinaldo es un beneficio esencial para todos los trabajadores en Costa Rica, como un reconocimiento al esfuerzo del trabajador durante todo el año. En definitiva, el aguinaldo es más que una compensación anual, pues es un mecanismo que refuerza la equidad laboral y promueve la justicia laboral.

Regulación y pago de los días feriados

Los días feriados son días estipulados por el Estado costarricense para que los trabajadores puedan disfrutar de las actividades festivas especiales, ya sean cívicas, religiosas, sociales o históricas. Al respecto, el MTSS (2025) menciona:

Existen dos tipos de días feriados:

1. Son días feriados de pago obligatorio los siguientes:

1° de enero (Circuncisión-Año Nuevo), 11 de abril (Día de Juan Santamaría), Jueves y Viernes Santos, 1° de mayo (Día Internacional del Trabajo), 25 de julio (Anexión del Partido de Nicoya a Costa Rica). 15 de agosto (Día de la Madre y Asunción de la Virgen María), 15 de septiembre (Fiesta Nacional de Conmemoración de la Independencia de Costa Rica) y el 25 de diciembre (Natividad de Jesucristo).

2. Son días feriados de pago no obligatorio:

El 2 de agosto (día de la Virgen de los Ángeles), el 31 de agosto (Día de la Persona Negra y la Cultura Afrocostarricense) y el 01 de diciembre (Día Abolición del Ejército). (párr. 2).

Las personas que no profesan la religión católica también tienen derecho a que les sean respetados sus días de culto. Para disfrutar de dichos días se deben solicitar a la persona empleadora. No pueden ser más de cuatro días libres al año, pues esos días deben estar registrados en el Ministerio de Relaciones Exteriores y Culto, y dichos días no son de pago obligatorio por parte del patrono.

Si la persona trabaja en un día feriado, si el día es de pago obligatorio, recibirá el pago doble de ese día. Si es un día feriado no de pago obligatorio se le hará un pago sencillo del día.

Si el trabajador realiza en el día feriado de pago obligatorio una jornada extraordinaria, la jornada ordinaria se paga doble, pero la jornada extraordinaria se paga a tiempo y medio (1.5), pero calculada con la hora doble.

Los días feriados en Costa Rica representan momentos de celebración cívica, religiosa e histórica. Son un derecho fundamental para los trabajadores, que garantizando el disfrute de los días festivos. Esto muestra un equilibrio entre la productividad y el bienestar social, para promover un entorno laboral digno para los trabajadores costarricenses.

Tipos de jornadas laborales y distribución horaria

Una jornada laboral es el máximo de horas ordinarias permitidas por la ley, en la que los trabajadores están sujetos a las órdenes de sus patronos. En Costa Rica existen varias jornadas laborales ordinarias. Al respecto, el MTSS (2025) menciona lo que sigue:

Hay varios tipos de jornadas ordinarias de trabajo. Según el número de horas que se labore por día, unas pueden llamarse jornadas ordinarias normales y son la generalidad; otras, que son las menos, se pueden llamar jornadas ordinarias especiales o de excepción. Además, dependiendo si se trabaja en el día o en la noche, las jornadas ordinarias se denominan diurnas, nocturnas o mixtas. (párr. 2).

A continuación se detallarán las diferentes jornadas labores permitidas por el Estado costarricense. La jornada ordinaria diurna es el periodo laboral comprendido entre las cinco de la mañana y las siete de la noche. El máximo de horas laboradas en esta jornada es de ocho horas laborales por día y son de un total de cuarenta y ocho horas semanales. En contraste, la jornada ordinaria nocturna es la jornada comprendida entre las siete de la noche y las cinco de la mañana del siguiente día. También establece que las horas diarias laborales son de seis horas y un máximo de treinta y seis horas semanales.

Por otro lado, la jornada ordinaria mixta combina trabajo de la jornada diurna (entre las cinco de la mañana y las siete de la noche) y la otra parte en la jornada nocturna (entre las siete de la noche y las cinco de la mañana). Se establece un máximo de siete horas laborales diarias y un máximo de cuarenta y dos horas semanales. De trabajar después de las diez y treinta de la noche o más, la jornada mixta se convierte en una jornada nocturna y en ese caso se aplica lo estipulado para dicha jornada.

Jornada ordinaria acumulativa es aquella en la que se labora el sexto día de la semana después de los cinco días anteriores, que por lo general corresponde al sábado. Para efectos legales el sexto día sigue siendo un día hábil y se toma en cuenta para otorgar vacaciones.

Además, existen jornadas ordinarias especiales. Existen las siguientes: a) la jornada de las personas trabajadoras a domicilio; b) la de personas en puestos de gerencia, administradores, apoderados y quienes trabajan sin fiscalización superior; c) personas en puestos de confianza; d) agente comisionistas y empleados similares que no cumplen con sus labores en el local del establecimiento o empresa; e) realizan funciones discontinuas o que solo requieren su presencia; f) las que por su naturaleza no están sometidas a jornadas laborales ordinarias normales. Los trabajadores comprendidos en los puntos anteriores no están obligados a permanecer en el centro de trabajo por más de doce horas diarias y tienen derecho de una hora y media de descanso. Las personas adolescentes comprendidas entre los quince años y menores de dieciocho años que laboren se consideran de una jornada ordinaria especial, y no deben laborar más de seis horas diarias y treinta y seis horas semanales.

Finalmente, la jornada extraordinaria se refiere al tiempo laborado por la persona más allá de las horas comprendidas en las jornadas ordinarias. También se establece que las personas no pueden trabajar más de doce horas diarias.

Las jornadas laborales buscan garantizar condiciones de trabajo justas y saludables para los empleados. Las diferentes jornadas laborales permiten la flexibilidad laboral tanto para los empleados como para el bienestar de la empresa. Cada parte adopta la jornada que más le convenga para su propio beneficio.

Regulación del auxilio de cesantía de los trabajadores.

Es un derecho que tienen los empleados en caso de finalización del contrato laboral con responsabilidad laboral. El objetivo es que la persona despedida cuente con un pago mínimo para mantenerse mientras encuentra otra opción laboral.

La persona empleadora deberá cumplir con las regulaciones establecidas para el pago de auxilio de cesantía. En el caso de un periodo laboral comprendido entre tres y seis meses el importe debe ser igual al de siete días de salario. Si el trabajo realizado abarca entre seis meses y menor al de un año, ese importe será igual a catorce días de salario. Para los periodos laborales a un año el importe de días de salario es el indicado en la siguiente tabla.

Tabla 9*Cálculo del importe de auxilio de cesantía*

Años	Días de importe
1	19.5 días por año laborado
2	20 días por año laborado o fracción superior a seis meses
3	20.5 días por año laborado o fracción superior a seis meses
4	21 días por año laborado o fracción superior a seis meses
5	21.24 días por año laborado o fracción superior a seis meses
6	21.5 días por año laborado o fracción superior a seis meses
7	22 días por año laborado o fracción superior a seis meses
8	22 días por año laborado o fracción superior a seis meses
9	22 días por año laborado o fracción superior a seis meses
10	21.5 días por año laborado o fracción superior a seis meses
11	21 días por año laborado o fracción superior a seis meses
12	20.5 días por año laborado o fracción superior a seis meses
13	y siguientes: 20 días por año laborado o fracción superior a seis meses

Fuente: Elaboración propia, 2025.

El pago de la cesantía no podrá ser mayor al de los últimos ocho años de relación laboral. Se deberá realizar el pago del auxilio de cesantía aunque el trabajador pase a servirle inmediatamente a otro patrono.

Para el cálculo del auxilio de cesantía se toma en cuenta el promedio de todos los salarios que el colaborador recibió en los últimos seis meses, ordinarios y extraordinarios. Para las personas cuyo contrato es de pago mensual se divide el monto en 30 días y para los de pago semanal se divide entre 26 días, para obtener la porción por día. El tiempo en que la persona no laboró por incapacidad no es tomando en cuenta. En este caso se realiza el cálculo con los seis meses antes de que la persona estuviera incapacitada. En el caso de incapacidad por maternidad, este tiempo sí es tomado en cuenta para el cálculo de la cesantía.

El auxilio de cesantía es una garantía de todos los trabajadores que se vean afectados por la ruptura de un contrato laboral con responsabilidad laboral. Proporciona un apoyo económico para el trabajador, mientras encuentra otra oportunidad laboral, por lo que disminuyen las cargas económicas durante ese tiempo.

Regulaciones del salario mínimo de los trabajadores

El salario mínimo corresponde al salario que toda persona debe recibir según su ocupación. Ningún trabajador debe recibir menos de lo estipulado en el salario mínimo. Para

determinar el salario mínimo por puesto el MTSS utiliza un instrumento para clasificar los puestos de trabajo. Al respecto, el MTSS (2025) menciona:

¿Qué son los perfiles ocupacionales? Es el instrumento que se utiliza para clasificar un puesto de trabajo. Para clasificar un puesto de trabajo se deben conocer y describir las diversas funciones, labores o tareas que realiza la persona trabajadora. Definidas estas, se ubican o clasifican en los perfiles ocupacionales aprobados por el Consejo Nacional de Salarios y de acuerdo con esos perfiles se relacionan con los renglones o nomenclaturas del Decreto de Salarios Mínimos. (párr.2).

A continuación se muestra un ejemplo de los puestos y sus salarios mínimos que rigen a partir del 01 de enero del 2025:

Tabla 10

Salarios mínimos por puestos laborales

Puesto Laboral	Salario mínimo
Salonero	₡12,236.95
Bachiller universitario	₡653,427.21
Zapatero	₡13,767.45
Futbolista Primera División	₡15,983.96

Fuente: Elaboración propia, 2025.

El salario mínimo es fundamental para garantizar condiciones laborales dignas y equitativas de los trabajadores. Debido a la clasificación de puestos y regulación de los montos mínimos de pago, se asegura que todos los trabajadores reciban una remuneración mínima justa, evitando la explotación y promoviendo la estabilidad económica del trabajador.

Regulaciones sobre vacaciones laborales

Las vacaciones son el derecho de que gozan los trabajadores costarricenses que representa un descanso remunerado, y que tiene como propósito un descanso físico y mental en pro de la salud del empleado. Al respecto, el MTSS (2025) menciona: “Las vacaciones son un derecho y una necesidad biológica de toda persona trabajadora. Consisten en un descanso anual remunerado que tiene como propósito permitirle a la persona trabajadora reponer el desgaste de energías realizado durante el año de labores” (párr. 1).

La ley estipula en este apartado que los trabajadores tienen derecho a dos semanas de vacaciones, independientemente del horario de trabajo y después de cincuenta semanas continuas de trabajo con el mismo patrono.

Cálculo de las vacaciones.

El cálculo de las vacaciones de un trabajador tiene diferentes variables y depende del tipo de horario laboral, el sector en el que se desempeña y la forma de pago que la empresa utiliza.

De acuerdo con el MTSS (2025), estos son elementos que se deben considerar para realizar el cálculo del pago de vacaciones:

Se calcula con base en el promedio de los salarios ordinarios y extraordinarios recibidos durante la última semana, si se trata de explotación agrícola y ganadera; o durante las últimas cincuenta semanas en el resto de las actividades de los centros de trabajo, contadas a partir del momento en que la persona trabajadora adquiera su derecho al descanso. El pago dependerá, asimismo, de la forma de pago que aplica el centro de trabajo. (párr. 3).

Con esta información se pueden identificar las variables necesarias para calcular correctamente el pago de vacaciones.

A continuación, en la Tabla 11 se muestran datos correspondientes al pago de las vacaciones y de acuerdo con la forma de pago de la empresa.

Tabla 11

Cálculo del pago de vacaciones

Forma de Pago	Días Para Pagar	Fórmula
Semanal	12	Salario semanal / 6 * 12
Mensual	14	Salario mensual / 30 * 14
Actividades comerciales	14	Salario semanal / 7 * 14

Fuente: Elaboración propia, 2025.

Las vacaciones no solo representan un período de descanso para los trabajadores sino que también son un derecho fundamental para garantizar el bienestar de los trabajadores y la productividad laboral. Además, reciben una compensación económica justa y acorde con el tiempo trabajado que les solventa la parte económica durante el disfrute de sus vacaciones.

Metodologías de software

Las metodologías de Software tienen como objetivo proporcionar una guía estructurada para los desarrolladores, a fin de optimizar sus esfuerzos y facilitar el cumplimiento de los objetivos en cada etapa del ciclo de vida del desarrollo del software. Mediante procedimientos definidos, estas metodologías aseguran un proceso eficiente y organizado. Al respecto, Maida y Pacienza (2015) menciona que: “una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información” (como se citó en Morales *et al*, 2021).

Existen diferentes tipos de metodologías de desarrollo de software, alguna tradicionales de hace muchos años que han evolucionado en otras metodologías, como también metodologías recientes adaptadas a la actualidad. En general, estas metodologías se pueden clasificar en tradicionales, caracterizadas por enfoques estructurados y secuenciales y ágiles, que priorizan la flexibilidad, la colaboración y la entrega rápida de productos funcionales.

A lo largo de los años, las metodologías de desarrollo de software han evolucionado para adaptarse a las necesidades del mercado y la tecnología. En los siguientes apartados se exploran en detalle cada una de estas categorías.

Metodologías tradicionales

Estas metodologías tradicionales de desarrollo de software son enfoques estructurados y secuenciales que han sido utilizados durante décadas para la creación de sistemas. Estas metodologías se caracterizan por seguir un proceso lineal y bien definido, en el que cada etapa debe completarse antes de pasar a la siguiente.

Entre sus principales características destacan el seguimiento de un orden lógico que se inicia en la planificación y culmina en la implementación. Además, requieren una planificación detallada antes de comenzar el desarrollo para garantizar que cada etapa esté bien definida. En cuanto a la flexibilidad, los cambios en el proyecto suelen ser limitados, lo que puede hacer que su implementación sea costosa y compleja. Por esta razón, estas metodologías resultan óptimas para proyectos con requisitos bien definidos y poco cambiantes.

Los equipos de trabajo deben estar seguros de elegir estas metodologías, ya que en el desarrollo de software es muy normal que los requerimientos del software cambien

constantemente, por lo que podría resultar en un proyecto más costoso de lo estipulado inicialmente, al ser menos flexible a los cambios.

Algunas de las metodologías tradicionales incluyen el modelo Cascada, caracterizado por su enfoque secuencial en el desarrollo de software. También está el Modelo V, que sigue una estructura similar a la del modelo Cascada, pero enfatiza la validación y verificación en cada fase. Por último, el modelo Incremental e Iterativo permite desarrollar el software en etapas, y mejora progresivamente su funcionalidad a medida que se avanza en el proceso.

Estos modelos con el paso del tiempo han evolucionado y dado lugar a variantes más flexibles y menos costosas que buscan mejorar el proceso de desarrollo.

Modelo Cascada

El modelo Cascada es una metodología tradicional de desarrollo de software que sigue un enfoque secuencial y estructurado. Se caracteriza porque debe completarse la fase para poder avanzar a la siguiente. De acuerdo con Quiroz y Goodwin (2024), el modelo Cascada se define como un proceso ALM (Application Lifecycle Management, por sus siglas en inglés y en español gestión del ciclo de vida) secuencial, con una serie de pasos establecidos:

El modelo Cascada es un proceso ALM secuencial, con una serie de pasos claramente definidos. Con este modelo los proyectos no avanzan a la siguiente fase sin la aprobación final de partes interesadas específicas. Los modelos de cascada funcionan bien con proyectos que tienen pasos más predecibles y menos variables. (p. 5).

El modelo Cascada ha sido una metodología clave en el desarrollo de software, ya que proporciona una estructura clara y ordenada para proyectos con requisitos bien establecidos y poco cambiantes. A pesar de su rigidez sigue siendo una opción válida hoy en día.

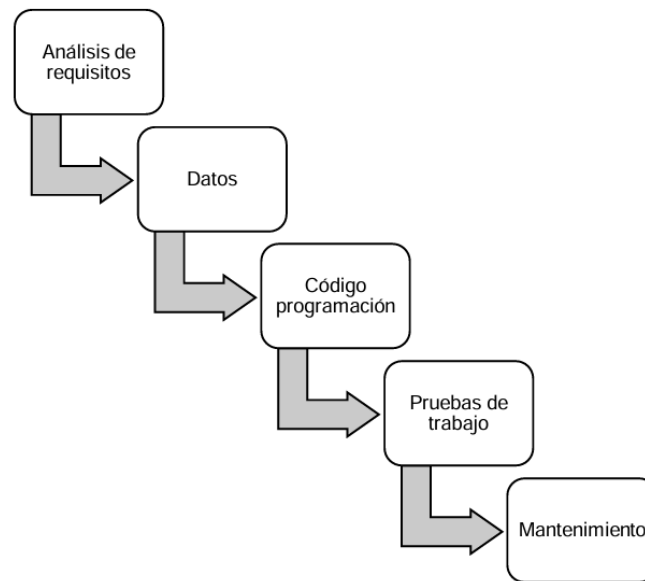
Fases del modelo Cascada

El modelo Cascada se basa en un enfoque estructurado y secuencial, que avanza fase por fase hasta completar el ciclo de vida del desarrollo. Primero, en la fase de análisis de requisitos se recopilan y documentan todos los datos sobre las necesidades del proyecto para definir los objetivos del software. A continuación, en la etapa de diseño del diseño se establecen aspectos clave como la arquitectura del software, las bases de datos y la estructura general del sistema.

Posteriormente, en la fase de codificación, se desarrolla el código del software siguiendo el diseño previamente establecido. Una vez completado este proceso se procede a la etapa de pruebas, en que se realizan evaluaciones para validar y verificar que el software funcione correctamente. Finalmente, en la fase de mantenimiento se aplican correcciones y mejoras con el objetivo de solucionar problemas que puedan surgir una vez que el software esté en uso.

Figura 4

Fases del modelo Cascada



Fuente: Betancur (2022).

Este modelo sigue una referencia para el desarrollo de software, especialmente en proyectos con requisitos bien definidos y poca variabilidad. Sus fases abarcan cada etapa necesaria para completar un proyecto, lo que garantiza un proyecto garantizado y organizado.

Modelo V

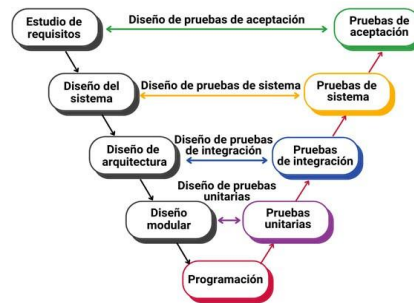
El modelo en V es una metodología de desarrollo de software derivada del modelo Cascada, que busca mejorar el proceso de desarrollo al incorporar pruebas en cada etapa, por lo que forma una estructura en forma de “V”. De acuerdo con Betancur (2022), el Modelo V es:

También conocido como modelo tipo “V”, el modelo de variación de cascada procura que el ciclo de trabajo objeto de estudio sea más efectiva y productiva, es decir, que disponga de ciertas cualidades características en metodologías de gestión de proyectos ágiles. (p. 23).

Esta variación del modelo Cascada permite una mayor detección temprana de fallas, debido a que las pruebas se realizan en cada etapa, con lo que se logran una disminución de costos y tiempos de corrección. Es ideal para proyectos con requisitos bien definidos.

Figura 5

Pruebas del Modelo V



Fuente: (Servicios de Información Profesional, S.A., 2023).

El modelo V establece un enfoque estructurado en el desarrollo de software, en que cada fase cuenta con una etapa de prueba y validación, lo que permite una mejor detección de errores y una mayor optimización de los recursos. La figura 5 muestra cómo las pruebas se integran de manera progresiva, lo cual asegura que cada componente sea verificado antes de avanzar a la siguiente fase.

Modelo incremental e iterativo

Este modelo es una metodología cascada, pero mejorada. La diferencia con la metodología cascada es que el modelo iterativo e incremental permite, a lo largo del proyecto, iterar e incrementar parte de un conjunto de requerimientos y no todos a la vez, lo que permite obtener prototipos funcionales de cada conjunto de estos requerimientos. Una vez finalizados son presentados al cliente y, de darse el visto bueno, este continúa con otro conjunto de requerimientos para iniciar con otro prototipo funcional. Y se repite el mismo proceso. No se obtiene todo el prototipo completo a la vez, pero sí parte de este. Al respecto, Solano y Porras (2020) mencionan:

La idea detrás de la mejora iterativa sobre el modelo de cascada es tomar como punto de inicio un conjunto de requerimientos que se pueden considerar básicos y bien definidos, y a partir de una primera versión del sistema de software, identificar mejoras en esta e

incorporar otros requerimientos que se han terminado de definir con mayor claridad o aquellos que pueden ir surgiendo durante la ejecución del proyecto. (p. 6).

Un incremento permite obtener parte de todas las funciones que se desea desarrollar, no todas. Por ejemplo, se inicia con el incremento 1, luego este incremento se pasa por todas las fases del ciclo de vida de forma secuencial y al finalizar se obtiene un incremento o un prototipo. Este prototipo no incluye todos los requerimientos del proyecto, sino una parte de ellos. Una vez aceptado por el cliente se toma el incremento 2 y se continua con el mismo proceso, hasta finalizar con el incremento n. Esto agiliza el desarrollo del producto y permite integrar al cliente de forma más rápida.

Las iteraciones se realizan sobre la actividad estructural en la que nos encontramos. Si estamos en un incremento para desarrollar X función o proceso, etc., y estamos en la fase de desarrollo como modelado e iteramos sobre esa misma fase de modelado.

Las etapas del modelo de cascada incremental combinan la estructura secuencial del modelo en cascada con la entrega progresiva que permite mejoras constantes. En primer lugar, en la fase de requisitos, se recopilan y documentan los requerimientos iniciales del sistema para definir los objetivos y alcances del desarrollo. Luego, en la etapa de diseño se establece la arquitectura del software, considerando los módulos que puedan desarrollarse de manera incremental.

Posteriormente, en la fase de implementación, el software se desarrolla en incrementos funcionales, cada uno pasando por las etapas de diseño, codificación y pruebas. A continuación, en la fase de pruebas, cada incremento es evaluado individualmente y luego se integra con los anteriores para garantizar la funcionalidad completa del sistema.

Una vez validado, en la fase de despliegue, el software es entregado en versiones incrementales, lo que permite realizar mejoras y ajustes según la retroalimentación del usuario. Finalmente, en la fase de mantenimiento se llevan a cabo las correcciones y mejoras continuas en los incrementos ya entregados, lo cual asegura la estabilidad y la eficiencia del sistema en el tiempo.

Los modelos Incremental e Iterativo representan una evolución del enfoque tradicional en cascada, y esto permite mayor flexibilidad en el desarrollo de software al implementar los requerimientos de manera progresiva. Las iteraciones y entregas incrementales facilitan la retroalimentación continua con el cliente, lo cual asegura que el producto se ajuste a sus necesidades.

Metodologías ágiles

Las ágiles son un conjunto de metodologías de desarrollo de software que promueven la colaboración, la adaptabilidad y la entrega incremental. Su objetivo es impulsar la adaptabilidad, en la que los requisitos cambian constantemente y se centran en el trabajo en equipo y el liderazgo de estos. Las metodologías ágiles se basan en un conjunto de valores que a su vez acompañan 12 principios que guían el desarrollo del proyecto. Según Mora (2020), los principios del manifiesto Ágil de Desarrollo de Software son:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente dentro de dos semanas y dos meses, con preferencia del periodo más corto posible.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo es comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua de la excelencia técnica y el buen diseño mejoran la agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia. (p. 33).

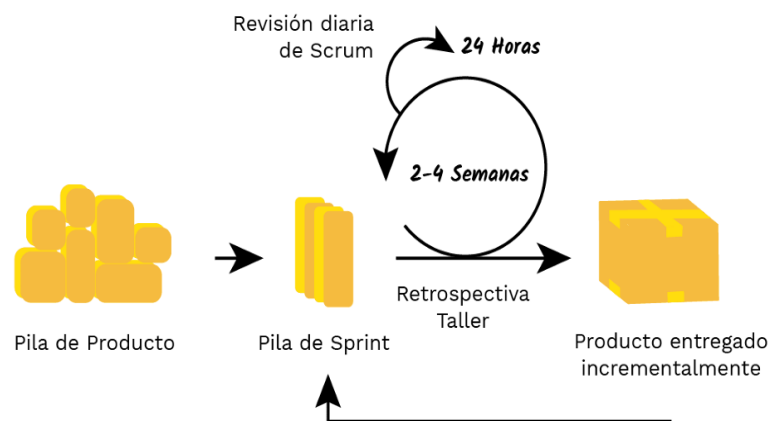
Las metodologías ágiles han revolucionado la forma en que se desarrollan los proyectos de software, en el que se priorizan la adaptabilidad, la colaboración y la entrega incremental. Gracias a sus principios, permiten responder a cambios de requerimientos y optimizar la producción de software. Dentro de estas metodologías destacan Scrum y Kanban, los cuales ofrecen marcos de trabajo flexibles para el desarrollo del software.

Scrum

Scrum es un marco de trabajo utilizado hoy en día en el desarrollo de software que gestiona proyectos de forma incremental. Se basa en ciclos cortos de tiempo llamados “Sprints”, normalmente comprendidos en un tiempo de dos semanas y al final de cada sprints el cliente recibe parte funcional del software. Las metas de cada sprints están relacionadas con los resultados de los anteriores, en los que se toma en cuenta al cliente para que haga sus observaciones y dé su aprobación.

Figura 6

Ciclo de vida de Scrum



Fuente: Mora (2020).

En la Figura 6 se puede apreciar el ciclo de vida de Scrum, en la que se aprecia una “Pila de producto” que representa las funcionalidades del producto. La pila de sprint contiene las funcionalidades en las que se debe trabajar y el producto entregado representa el producto funcional de la pila de sprint.

El marco de trabajo Scrum ofrece un enfoque ágil e incremental para la gestión de proyectos de software. Permite una entrega cada cierto tiempo de software funcional, gracias a los sprint. Este enfoque mejora la colaboración entre los equipos, pues optimiza el tiempo de entrega y aumenta la calidad del software. Scrum se ha consolidado como una de las metodologías ágiles más utilizadas.

Arquitectura de software

Una arquitectura de software es el conjunto de decisiones que definen la estructura y el comportamiento de un sistema de software. Especifican componentes principales del sistema, sus responsabilidades, cómo se comunican entre sí, y cómo se satisfacen tanto los requerimientos funcionales como los no funcionales (como rendimiento, seguridad, escalabilidad y mantenibilidad).

Es un aspecto importante en la ingeniería de software que influye directamente en la calidad del sistema, la escalabilidad, el mantenimiento y la eficiencia del desarrollo. Una arquitectura bien definida también facilita la colaboración del equipo y la evolución del sistema a lo largo del tiempo. Al respecto, Cambarieri *et al* (2020) mencionan lo que sigue: “La arquitectura de software es la representación de alto nivel de la estructura de un sistema, pues describe las partes que la integran, las interacciones entre ellas, los patrones que supervisan la composición, y las restricciones de aplicar esos patrones” (p. 195).

Existen diferentes arquitecturas de software y dentro las más conocidas se encuentran la arquitectura monolítica, la arquitectura orientada al servicio (SOA) y la arquitectura orientada a microservicios, cada una tiene con sus ventajas y desafíos.

Estilos arquitectónicos comunes

Las arquitecturas de software han evolucionado para adaptarse a los desafíos tecnológicos actuales, lo cual ha permitido mayor flexibilidad y escalabilidad. A lo largo del tiempo han surgido varios estilos arquitectónicos. Entre los más comunes se encuentran la arquitectura de monolítica, la arquitectura orientada a servicios (SOA) y la arquitectura de microservicios.

En este apartado se exploran sus características, ventajas y desafíos, enfocados particularmente en la arquitectura monolítica, que ha sido elegida para este proyecto por sus características específicas. Al respecto, Cambarieri *et al* (2020) mencionan algunas de las arquitecturas que han surgido en las últimas décadas:

En las últimas décadas surgieron diferentes ideas de las arquitecturas de los sistemas, como la arquitectura hexagonal (AH, también conocida como Puertos y Adaptadores); desarrollada por Alistair Cockburn y Datos, Contexto e Interacción (DCI, por sus siglas en

inglés Data, Context and Interaction) de James Coplien y Trygve Reenskaug, entre otras. (p. 195).

Independientemente del enfoque adoptado, el objetivo primordial de toda arquitectura es desarrollar software eficiente, escalable y bien estructurado, para minimizar costos tanto económicos como operativos.

Arquitectura monolítica

Las arquitecturas monolíticas tienen como principal característica que empaquetan toda la funcionalidad de una aplicación en un único código base, lo que facilita el desarrollo inicial y las pruebas. Las tres partes de la aplicación, la interfaz de usuario, la lógica de negocio y la gestión de datos funcionan como un todo en un mismo ordenador o servidor. Todos los componentes están interconectados entre sí y son dependientes unos de los otros, para formar un único programa unificado. Al respecto, Hernández *et al* (2021) mencionan:

Dicha arquitectura tiene un diseño para ser considerado autónomo. Sus componentes ligados en el código son desplegados dentro de un solo compilador en el lenguaje de programación en el que se realiza la aplicación, obligado a estar interconectado y ser dependiente de sí mismo para su funcionalidad. (párr. 7).

La arquitectura monolítica fue en su momento una solución económica y efectiva, ya que permite un desarrollo ágil y un despliegue sencillo. Por esta razón, siguen siendo útil para aplicaciones con requisitos bien definidos y poco cambiantes, como sistemas internos o herramientas empresariales con baja demanda de escalabilidad. Sin embargo, con el avance tecnológico y la creciente necesidad de adaptabilidad las arquitecturas modernas han evolucionado hacia enfoques más flexibles, como los microservicios, que permiten mayor escalabilidad y mantenimiento eficiente.

La arquitectura monolítica fue seleccionada para este proyecto debido a la naturaleza del software, que está diseñado exclusivamente para el área de recursos humanos que no requiere integración con ningún otro sistema de la empresa. Al no ser una aplicación web sino un sistema local ejecutado en un ordenador o servidor. La arquitectura monolítica resulta la opción más adecuada, ya que permite un desarrollo más sencillo, un despliegue directo y un mantenimiento centralizado. Además, esta elección favorece la estabilidad y coherencia del sistema, al concentrar todas sus funcionalidades en un único entorno sin depender de componentes externos.

Arquitectura orientada a servicios (SOA)

La arquitectura orientada a servicios (por sus siglas en inglés SOA, Service-Oriented Architecture) es un enfoque de diseño de software que organiza las funcionalidades del sistema en servicios independientes y reutilizables. Estos servicios pueden interactuar entre sí a través de protocolos estándar, como HTTP y XML, que facilitan la interoperabilidad entre distintos sistemas. Al respecto, Hernández *et al* (2021) mencionan:

Para lograr la flexibilidad, interacción e integración, se requiere la existencia de una arquitectura que preste estos servicios de interacción en distintas capas, para lo cual existe la arquitectura orientada a servicios (SOA Service Oriented Architecture), que hace uso de los Servicios Web o Web Services, para lograr el objeto de la integración y flexibilidad de las aplicaciones. (párr. 21).

Algunas de las características de esta arquitectura es que es una arquitectura modular, lo que significa que los servicios están desacoplados, y esto permite que sean usados en diferentes aplicaciones. Otra característica es la interoperabilidad, que permite la integración entre aplicaciones desarrolladas en distintos lenguajes de programación y plataformas. Permite agregar nuevos servicios sin afectar el funcionamiento del sistema existente. Y concede la modificación de los servicios sin necesidad de alterar toda la arquitectura.

Arquitectura orientada a microservicios

La arquitectura de microservicios es un enfoque moderno del software, el cual consiste en crear pequeñas aplicaciones de software, manejables e independientes de las otras, con el objetivo de resolver una tarea en concreto del negocio. Ahora, el hecho de crear pequeñas aplicaciones funcionales que resuelven una necesidad no significa que el tamaño sea lo más importante, o que por defecto deben ser pequeñas. Se basa en crear servicios que por lo general resuelven una necesidad. Al respecto, Colombero *et al* (2024) mencionan:

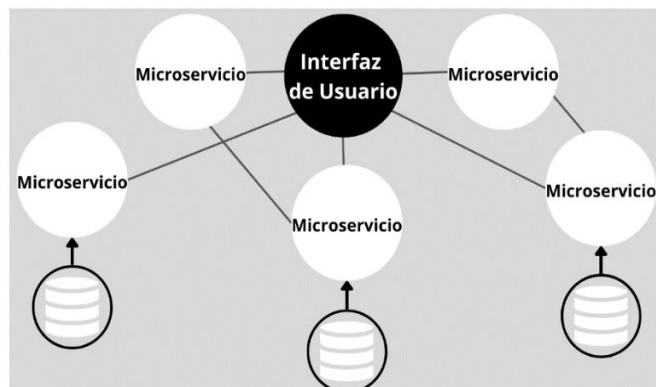
La arquitectura de microservicios emerge como un enfoque innovador y proporciona una alternativa prácticamente opuesta a la arquitectura monolítica tradicional. Este paradigma arquitectónico se basa en la idea de descomponer una aplicación en una serie de servicios independientes, cada uno enfocado en realizar una función específica. (p. 46).

Esta arquitectura permite que cada equipo trabaje en su servicio completamente independiente de los otros, incluso en diferentes lenguajes de programación, diferente ciclo

de vida, y que se integran por medio de APIs, que al final para el usuario es como una sola aplicación; pero en realidad con múltiples aplicaciones pequeñas que trabajan en conjunto.

Figura 7

Ejemplo de arquitectura de microservicios



Fuente: Colombero et al (2024)

Comparación de arquitecturas de software

En el siguiente apartado se presenta un cuadro comparativo en el que se destacan las ventajas y desventajas de las tres arquitecturas analizadas, lo que permite obtener una visión general de sus características.

Tabla 12

Comparación de arquitecturas de software

Arquitectura	Ventajas	Desventajas
Monolítica	Simplicidad en el desarrollo y despliegue	Difícil de escalar y mantener
	Mayor eficiencia en comunicación interna	Cambios afectan a toda la aplicación
	Facilidad de despliegue	Menos flexible para nuevas tecnologías
	Menos sobre carga en integración	Equipos más centrados en correcciones
	Menor costo	No se aprovecha la computación en la nube
SOA	Modularidad y reutilización de servicios	Mayor complejidad en gestión
	Interoperabilidad de sistemas	Latencia en comunicación entre servicios
	Permite integración con diversas plataformas	Puede requerir mayor infraestructura
Microservicios	Independencia de servicios	Complejidad en integración y comunicación
	Facilidad de escalabilidad	Mayor consumo de recursos en la red
	Equipos pueden trabajar en forma autónoma	Requiere gestión avanzada para despliegue

Fuente: Elaboración propia, 2025.

En la comparación de arquitecturas de software es evidente que cada enfoque tiene sus propias fortalezas y desafíos. La elección entre una arquitectura monolítica, SOA o Microservicios dependerá de las necesidades específicas del sistema, el contexto del negocio y

los objetivos de escalabilidad y mantenimiento. Al respecto, Colombero *et al* (2024) mencionan:

La implementación y el desarrollo práctico de las arquitecturas monolíticas y de microservicios presentan diferencias significativas que influyen en la elección de la estructura arquitectónica más adecuada para un proyecto específico. Si bien ambos enfoques tienen sus ventajas y desventajas, es crucial considerar diversos factores, como el tamaño de la complejidad del proyecto, las capacidades del equipo de desarrollo y los requisitos de escalabilidad y mantenimiento de largo plazo, condiciones que más allá de las bases teóricas de la arquitectura impactan en el desarrollo en aspectos prácticos y en escenarios reales. (p. 47).

Las arquitecturas monolíticas ofrecen simplicidad y eficiencia en la comunicación interna, pero pueden volverse rígidas frente a cambios y actualizaciones. SOA ofrece modularidad y reutilización, lo cual favorece la interoperabilidad, aunque a costa de una escalabilidad y la autonomía de los equipos, pero requieren una infraestructura más sofisticada.

Arquitectura del Sistema

En este apartado se presenta la arquitectura del sistema propuesto, abordada su estructura mediante un modelo en capas, el uso de diversas tecnologías y el enfoque de desarrollo adoptado. Se analizarán los principales principios de diseño empleados, destacando cómo la segmentación en capas facilita la separación de responsabilidades, la escalabilidad y el mantenimiento del sistema.

Se detalla la implementación del modelo Vista-Controlador (MVC) dentro de la arquitectura en capas, en lo que se resalta su papel en la organización lógica del sistema. Asimismo, se exploran tecnologías utilizadas como Spring Boot, Thymeleaf, Bootstrap, Java y Mysql, por su contribución al rendimiento y la eficiencia del desarrollo.

Modelo en capas

El modelo en capas es una arquitectura de software ampliamente utilizada para organizar sistemas de manera estructurada, dado que facilita la separación de responsabilidades y mejora la escalabilidad y el mantenimiento del código. Su principal ventaja radica en la

división de la aplicación en distintas capas funcionales, cada una con un propósito específico, ya que esto permite el desarrollo más modular y flexible.

Este modelo se compone generalmente de tres capas fundamentales, que permiten separar el código de acuerdo con sus funcionalidades. Una de las tres capas es la capa de presentación, que se encarga de la interfaz de usuario y de la interacción visual de la aplicación, en la que se definen el diseño, la presentación y los demás aspectos relacionados con la parte visual de la aplicación. La capa llamada lógica de negocio se encarga de gestionar todos los procesos y reglas del sistema, como cálculos, validaciones, seguridad, entre otros aspectos requeridos por el sistema. Y, por último, se tiene la capa de datos, que es responsable del almacenamiento y acceso a los datos. Al respecto, Rojas y Rodríguez (2020) mencionan:

La programación por capas es un modelo de desarrollo que separa las partes de los componentes del software (Vargas, R. 2010). Estas representan las funciones en varios niveles, con el fin de que si se presenta algún cambio durante el mismo desarrollo solo se afecte el nivel o capa donde se encuentra esta funcionalidad. Esto facilita el trabajo de una forma organizada, y de esta manera se obtendrá una reducción de costos y tiempo en el desarrollo. (p. 50).

En este apartado se explotará la aplicación del modelo en capas en el sistema propuesto, y se analiza su integración con tecnologías como Spring Boot, Thymeleaf, Bootstrap, Java y MySQL, así como su relación con el patrón Modelo Vista Controlador (MVC).

Modelo Vista Controlador (MVC).

El Modelo Vista Controlador es un patrón de diseño orientado a objetos de los más utilizados. Consiste en separar el código en tres capas, como su nombre lo describe, con el fin de crear software escalable y modular. Al respecto, Sánchez (2020) menciona:

MVC o Modelo-Vista-Controlado es un patrón de diseño de software que se utiliza y a la vez mantiene separados los tres componentes descritos en esta metodología. Se trata de una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos, como en sistemas empresariales. Además, es muy utilizado para la arquitectura en la mayoría de los frameworks modernos y uno de los más usados en la industria para poder crear proyectos escalables y modulares. (p. 2).

Entre las ventajas de utilizar este patrón de diseño se encuentra la separación de responsabilidades entre las capas, que ayuda a evitar el acoplamiento, y permite que cada

componente pueda ser modificado o mejorado sin afectar a las demás. Facilita el trabajo en equipo debido a que cada equipo puede trabajar en su capa correspondiente si no interfiere en las demás. Otra de las ventajas es su escalabilidad, que permite el aumento de trabajo sin comprometer el sistema. También facilita el mantenimiento. Este es un concepto clave en las aplicaciones modernas, ya que permite que un sistema crezca de manera eficiente. Otra de las ventajas es la reutilización de código, pues cada componente puede ser reutilizado en diferentes partes de la aplicación, y esto permite ahorrar tiempo de trabajo y utilizar código funcional de calidad y probado.

Las capas, como se mencionó anteriormente, tienen sus propias responsabilidades. La capa de modelo representa los datos y la lógica de negocio, gestiona el acceso y la manipulación de la información y se comunica con la base de datos y otras fuentes de datos. La capa llamada controlador actúa como un intermediario entre la capa de modelo y la capa de vista, procesa solicitudes de usuario y determina qué datos mostrar, y gestiona la interacción y el flujo de la aplicación. La vista se encarga de presentar la información al usuario, utiliza tecnologías como HTML, CSS y framework de diseño, y no contiene lógica de negocio; solo muestra los datos proporcionados por el modelo.

Tecnologías utilizadas en el proyecto

En el presente apartado se explican las tecnologías utilizadas en cada capa, con detalle de cómo contribuyen al funcionamiento del sistema y las ventajas que ofrecen en el desarrollo y mantenimiento de la aplicación. Además, se analizan la compatibilidad entre estas tecnologías y su impacto en la eficiencia del sistema. Se destacan las características y el papel que desempeñan de cada framework, librerías, entre otros más.

Capa de presentación

La capa de presentación, como se mencionó anteriormente, es la capa que se encarga de interactuar con el usuario del sistema. Se utilizarán tecnologías como HTML, CSS, Bootstrap y Thymeleaf para crear las interfaces gráficas de usuario y comunicación con las demás capas.

HTML (Lenguaje de marcado de hipertexto).

El lenguaje de marcado de hipertexto, según sus siglas en inglés (HyperText Markup Language), es un lenguaje compuesto por un conjunto de etiquetas definidas con un nombre rodeado de paréntesis angulares <>, que permiten crear elementos como tablas, botones y otros componentes visuales. Su función principal es facilitar el diseño de interfaces para la interacción entre el usuario y el sistema. Los archivos HTML tienen una extensión .html. Al respecto, Buscaglia (2020) menciona:

Como primer acercamiento se puede decir que HTML, en sus siglas Hypertext Markup Language (lenguaje de marcas de hiper texto), es el lenguaje por defecto en los navegadores a la hora de definir la estructura y el código que representa el contenido en nuestra aplicación web. (párr. 21).

Fue desarrollado por Tim Berners-Lee en 1991 como un lenguaje de marcado para estructurar documentos en la web. Desde entonces ha evolucionado mediante varias versiones, y es HTML5 la versión más reciente. HTML es la base de todas las páginas web y se utiliza para estructurar contenido como textos, imágenes, formularios y enlaces. Además, se complementa con CSS para el diseño y con JavaScript para la interactividad, lo cual permite la creación de aplicaciones dinámicas y funcionales.

Figura 8

Ejemplo de código HTML

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Carrito</title>
  </head>
  <body>
    <header id="header" class="header">
      <div class="container">
        <div class="row">
          <div class="col-12">
            <h1>Carrito de compras</h1>
          </div>
        </div>
      </div>
    </header>
  </body>
</html>
```

Fuente: Elaboración propia, 2025.

La Figura 8 muestra un ejemplo de código HTML, en el que se destaca la estructura básica del documento. Esa estructura está conformada por tres elementos fundamentales, la cabeza del documento, representada con la etiqueta <head> en donde se incluyen metadatos y enlaces de archivos externos; y el cuerpo del documento definido por la etiqueta <body>, que

contiene las etiquetas con el contenido visible de la página. Y las etiquetas <html>, que delimitan el código HTML del documento.

CSS (Hojas de estilo en cascada)

Las hojas de estilo en cascada, por sus siglas en inglés CCS (Cascading Style Sheets), es un lenguaje utilizado para definir el diseño y la presentación de páginas web, que permiten controlar aspectos como colores, fuentes, tamaños y disposición de elementos. Se utiliza para definir los estilos de los elementos HTML, como color, fondo, tamaño, entre otros más. Para declarar estilos personalizados, CSS utiliza propiedades y valores, en los cuales las propiedades son características que se pueden modificar, como el color y los valores y son los ajustes específicos que se le aplican a la propiedad, como, por ejemplo, al color se le puede asignar un valor azul. Además, las propiedades y valores se definen mediante reglas de estilo que determinan como se verá y funcionará un elemento en una página web. Al respecto, Buscaglia (2020) menciona:

Para poder hacer que todo lo que programemos en Javascript se vea lo mejor posible se utiliza CSS, en sus siglas Cascading Style Sheets (Hojas de estilo en cascada), es un lenguaje con el cual se dará el *look and feel* que se quiere para nuestra aplicación desde colores hasta animaciones. (párr. 20)

Figura 9

Ejemplo de una regla en CSS

```
h6, .h6, h5, .h5, h4, .h4, h3, .h3, h2, .h2, h1, .h1 {
  margin-top: 0;
  margin-bottom: 0.5rem;
  font-weight: 500;
  line-height: 1.2;
}
```

Fuente: Elaboración propia, 2025.

La figura 9 muestra una regla de CSS en la que se le aplican las propiedades y los valores definidos a todos los elementos h6 al h1. También se están aplicando los estilos a clases .h6 al .h1, valores como margin-top: 0 que elimina el margen superior. Esto permite estandarizar los estilos de los títulos y clases relacionadas, para asegurar uniformidad en el diseño del sitio.

Bootstrap como framework

Es un framework de código abierto diseñado para facilitar el desarrollo de sitios web responsivos y modernos. Creado por Twitter en el 2011, se ha convertido en una de las herramientas más populares para diseñadores y desarrolladores web, debido a su facilidad de uso y compatibilidad con múltiples dispositivos. El framework tiene entre sus características diseño responsivo que permite que las páginas web se adapten automáticamente al tamaño de la pantalla, componentes predefinidos que incluyen botones, formularios, menús de navegación, entre muchos más, listos para usar; además de que es compatible con CSS y el Javascript. Este permite crear interfaces interactivas y funcionales con menos esfuerzo. Al respecto, Bootstrap (s.f) menciona que “Bootstrap es un conjunto de herramientas de interfaz potente y repleto de funciones. Construye cualquier cosa, desde el prototipo hasta la producción, en minutos” (párr. 1).

EL framework emplea estilos y configuraciones globales, buscando la normalización de estilos entre navegadores. Bootstrap se ha convertido en un estándar en el desarrollo frontend, lo que les permite a diseñadores y desarrolladores crear sitios web modernos con menos esfuerzo y mayor eficiencia.

Thymeleaf como plantilla java

Es un motor de plantillas Java que permite generar vistas dinámicas en aplicaciones web. Su principal objetivo es facilitar la creación de páginas HTML que pueden ser interpretados tanto por navegadores como por el propio servidor. Es capaz de procesar HTML, XML, JavaScript, CSS e incluso texto plano. Algunas de las ventajas que ofrece Thymeleaf es que son plantillas naturales que permiten que los archivos HTML sean legibles y funcionales sin necesidad de un servidor, lo que facilita el diseño y la colaboración entre desarrolladores. Además, puede procesar documentos en distintos formatos para manipular datos dentro de las vistas. Se integra muy bien con Spring MVC, en reemplazo de tecnologías como JSP. Al respecto Calderón (2023) menciona:

Thymeleaf es un motor de plantillas HTML que trabaja en sintonía con Spring Boot. Genera contenido dinámico a partir de la vista que obtiene los valores del modelo. Su objetivo es simplificar y “naturalizar” las plantillas HTML, manteniendo su esencia y legibilidad y también haciendo más sencilla la comunicación entre la interfaz del usuario y el servidor,

puesto que son plantillas directamente generadas por el backend que, además, no solo pueden funcionar por sí mismas sino que también pueden ser integradas en otros lugares, como por ejemplos frameworks de JavaScript (Thymeleaf, s.f). (p. 24).

Gracias a sus características y ventajas, Thymeleaf se ha consolidado como una herramienta esencial en el desarrollo de aplicaciones Java. Su capacidad para generar vistas dinámicas sin perder la estructura y legibilidad del código HTML permite mejorar la colaboración entre desarrolladores y diseñadores. Además, su integración con Spring Boot y compatibilidad con múltiples formatos lo convierten en una alternativa eficiente ante otros motores de plantillas como JSP.

Capa lógica

La capa lógica o capa de negocio es la que se encarga de todo lo que tiene que ver con las reglas y procesos que definen cómo debe comportarse el sistema según las necesidades del negocio o dominio. Esta capa actúa de intermediaria entre la capa de presentación y la capa de acceso a datos. Para la capa lógica se utilizará Java como lenguaje de programación y Spring Boot.

Java como lenguaje de programación

Java es un lenguaje de programación orientado a objetos desarrollados por Sun Microsystems en 1995. Su diseño permite que los programas escritos en Java puedan ejecutarse en cualquier sistema operativo gracias a la Java Virtual Machine (JVM). Java es uno de los lenguajes más utilizados del mundo. Al respecto, Altadill y Pérez (2020) mencionan:

Siguiendo en esa línea visionaria en Sun trataban de crear un lenguaje de programación que pudiera funcionar en cualquier tipo de plataforma: “Write once, run everywhere” era el primer lema de ese nuevo lenguaje. Un par de ideas estaban claras: debía ser orientado a objetos, pero sin ser tan complicado como C++. (p. 60).

Entre las principales características del lenguaje destaca la multiplataforma, que puede ejecutarse en cualquier sistema operativo sin tener que realizar cambios en el sistema operativo. Es orientado a objetos, lo que facilita la reutilización de código. Es también un lenguaje seguro y robusto y es utilizado en múltiples aplicaciones, web, móviles, empresarial, entre muchas más.

Java sigue siendo una de las tecnologías más influyentes en el desarrollo de software. Gracias a su versatilidad, seguridad y compatibilidad con múltiples plataformas, se ha convertido en la base de muchas aplicaciones modernas, desde sistemas empresariales hasta desarrollo móvil con Android. Su constante evolución, respaldada por una gran comunidad y por actualizaciones periódicas, garantiza su relevancia en la industria. Java, a hoy en día, continúa siendo una de las mejores opciones para el desarrollo de software robusto y escalable.

Spring boot como framework java

Spring Boot es un framework de desarrollo de aplicaciones Java que simplifica la creación de aplicaciones basadas en Spring Framework. Su objetivo principal es reducir la configuración manual y acelerar el desarrollo de aplicaciones empresariales y microservicios.

Al respecto, IBM, (s.f) menciona:

Java Spring Boot (Spring Boot) es una herramienta que hace que el desarrollo de aplicaciones web y microservicios con Spring Framework sea más rápido y fácil mediante las tres funcionalidades principales:

1. Configuración automática
2. Un enfoque obstinado de la configuración
3. La capacidad de crear aplicaciones independientes

Estas características funcionan juntas para ofrecer una herramienta que le permite configurar una aplicación basada en Spring, con una configuración y preparación mínimas. (párr. 2-3).

Entre las principales características que ofrece Spring Boot a los desarrolladores se encuentran configuraciones automáticas de los componentes necesarios, para obviar la necesidad de archivos XML extensos. También permite ejecutar aplicaciones sin la necesidad de servidores externos, gracias a servidores embebidos con Tomcat, Jetty o Undertow. Además, ofrece preconfiguraciones para facilitar la integración con bases de datos, seguridad, APIs REST, entre otras tecnologías.

En el contexto de este proyecto la elección de Spring Boot responde a la necesidad de implementar una arquitectura robusta que permita el desarrollo ágil de la aplicación, para asegurar una mejor gestión de los recursos y una integración fluida con bases de datos, seguridad y tecnologías web.

Capa de datos

La capa de datos es un componente fundamental en la arquitectura de software, responsable de la gestión, almacenamiento y recuperación de información dentro de un sistema. Su objetivo principal es proporcionar un acceso eficiente y seguro a los datos, que asegure su integridad y disponibilidad para otras capas, como la lógica de negocio y la presentación.

MySQL como gestor de bases de datos

MySQL es un sistema de gestión de bases de datos relacional de código abierto, desarrollado en 1995 por Michael Widenius, David Axmark y Allan Larsson, en Suecia, con el objetivo de crear un sistema de bases de datos rápido y confiable. En el 2008 fue adquirido por Sun Microsystems y posteriormente, en el 2010, pasó a formar parte de Oracle Corporation. Utiliza SQL (Structured Query Language) para gestionar los datos y es ampliamente utilizado en aplicaciones web, sistemas empresariales y plataformas de comercio electrónico. Al respecto, ORACLE (2024) menciona:

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto que se utiliza para almacenar y gestionar datos. Su fiabilidad, rendimiento, escalabilidad y facilidad de uso hacen de MySQL una opción popular para los desarrolladores. De hecho, lo encontrarás en la base de aplicaciones exigentes y de alto tráfico como Facebook, Netflix, Uber, Airbnb, Shopify y Booking.com. (párr. 1).

MySQL presenta entre sus principales características el que se destacan su alta velocidad y rendimiento, optimizado para mejorar grandes volúmenes de datos, compatibilidad con múltiples plataformas como Windows, Linux, MacOS, entre otras plataformas. Ofrece seguridad avanzada con autenticación basada en contraseñas y verificación por host y la escalabilidad, que permite manejar desde pequeñas aplicaciones hasta grandes sistemas empresariales.

La implementación de MySQL en el proyecto de Recursos Humanos de la empresa garantiza una gestión eficiente y estructurada de la información, y asegura integridad, disponibilidad y seguridad de los datos. Además, su escalabilidad permite que el sistema pueda adaptarse al crecimiento de la empresa, pues ofrece soluciones dinámicas y optimizadas para la consulta y manipulación de datos. Gracias a su robustez y confiabilidad, MySQL se posiciona como una base de datos óptima para este proyecto

Seguridad del software

La seguridad del software es un aspecto fundamental en el desarrollo de aplicaciones, para garantizar la protección de datos, la privacidad de los usuarios y la integridad del sistema frente a amenazas cibernéticas. Al respecto, Mena y Ordóñez (2021) mencionan:

La seguridad es una condición que permite tener libertad ante el peligro, su objetivo es la protección contra los adversarios, por ejemplo, en la seguridad de un estado se debe proteger su soberanía, sus activos y su gente para que estos no sean atacados. (p. 1).

Los sistemas de información deben garantizar la seguridad de los datos que gestionan, ya que solo así pueden generar confianza en los usuarios. Si un sistema no ofrece seguridad, por más eficiente o funcional que sea resultará inútil para cualquier organización. Al respecto, Mena y Ordóñez (2021) mencionan que hay varias formas de vulnerar la seguridad de un sistema de software:

En el software se pueden encontrar formas de vulnerarlo, por ejemplo a partir de errores de implementación, defectos presentes en la fase de diseño, por medio de un desbordamiento de *buffer*, falta de seguridad en el código, mal manejo de errores, entre otros. (p. 3).

Por ello, como responsables del desarrollo de software se deben implementar todas las medidas de seguridad que estén al alcance para garantizar la protección e integridad de los datos, para dar así la confianza que tanto los usuarios como las empresas necesitan para utilizar el sistema. Es importante tener presente que los datos representan uno de los activos más valiosos de una organización, por lo que su resguardo constituye un asunto crítico que debe ser abordado con el mayor rigor posible.

Principios de la información

Para fortalecer la seguridad de un sistema informático se aplican diversos principios fundamentales orientados a garantizar la protección y confidencialidad de la información. Estos principios son la confidencialidad, y para cumplir con este principio se pueden implementar el cifrado y el acceso restringido. Otro principio es la integridad, que consiste en que los datos no deben ser cambiados por usuarios no autorizados que puedan comprometer el negocio, y, por último, la disponibilidad, que es garantizar que la información esté accesible a los usuarios legítimos. En relación con los pilares fundamentales de la seguridad de la información, Mena y Ordóñez (2021) mencionan lo siguiente:

Los pilares de la información se fundamentan en la necesidad que los datos sean confiables, íntegros y estén disponibles para obtener el máximo rendimiento con un mínimo de riesgo. Si la información que es vital para la toma de decisiones estuviera en manos equivocadas, perdería su valor con lo que se perderá la capacidad de maniobra, la reputación, además de sufrir daños por la cantidad de información a la que se puede acceder. (p. 5).

Figura 10

Principios fundamentales de la información



Fuente: Mena y Ordóñez (2021).

La protección de la información es un pilar fundamental en cualquier sistema informático, ya que garantiza su confidencialidad, integridad y disponibilidad. Aplicar estos principios permite minimizar riesgos y fortalecer la confianza en la gestión de datos, a fin de evitar accesos no autorizados, alteraciones maliciosas y pérdidas de información sensible.

Seguridad aplicada al proyecto

En el desarrollo de sistemas informáticos la seguridad es un componente esencial para proteger la integridad, la confidencialidad y la disponibilidad de la información. En el presente proyecto se implantarán diversas medidas de seguridad orientadas a garantizar el acceso controlado y seguro al sistema. Entre ellas se destacan la autenticación de usuarios mediante JSON Web Tokens (JWT), así como la gestión de roles y la restricción de permisos, según el nivel de acceso correspondiente. Estas estrategias permitirán prevenir accesos no autorizados y asegurar que cada usuario interactúe únicamente con los recursos que le han sido asignados.

JSON Web Tokens (JWT).

JSON Web Token (JWT) es un estándar abierto que permite la transmisión segura de información entre partes de un formato JSON. Se utiliza principalmente en autenticación y autorización, lo que asegura que los datos enviados sean confiables y verificables. Al respecto González (2020) menciona:

JSON Web Token es un estándar abierto definido en el RFC 7519, que permite crear tokens de acceso para aplicaciones y poder transmitir la información de una forma segura entre el cliente y el servidor. El servidor se encarga de firmar el token con una clave predefinida en que el cliente y el servidor podrán validar la autenticidad del token enviado. (p. 12).

Cuando un usuario inicia sesión en una aplicación el servidor genera un token JWT y lo envía al cliente. Este token contiene información codificada sobre el usuario y sus permisos. En cada solicitud posterior el cliente envía el JWT en el encabezado de autorización, lo que permite que el servidor valide la identidad del usuario sin necesidad de almacenar sesiones. Al respecto González (2020) menciona: “Los tokens presentan un formato del tipo JSON para el intercambio de datos” (p. 12).

Este mecanismo no solo optimiza la autenticación sino que también mejora la escalabilidad de las aplicaciones al reducir la dependencia de sesiones almacenadas en el servidor. Gracias a este enfoque se facilita la integración con múltiples servicios, lo cual garantiza un acceso seguro y eficiente para los usuarios.

Estructura de un JWT.

El cuerpo de un JWT está compuesto por tres partes separadas por punto (.). Estas partes contienen información esencial para el proceso de autenticación. La primera es la cabecera (header), que especifica el tipo de token y el algoritmo de firma utilizado. La segunda es la carga útil (payload), en que se incluyen los datos del usuario y otros atributos relevantes. Finalmente, se encuentra la firma (signature), generada mediante una clave secreta, cuya función es garantizar la integridad y autenticidad del token frente a posibles manipulaciones.

Figura 11

Partes de un JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ	Header	<pre>{ "alg": "HS256", "typ": "JWT" }</pre>
Jpc3Mi01J5b3VyLWFwaS1rZXkiLCJqdGkiOiIwLjQ3MzYyOTQ0NjIzNDU1NDA1IiwiaWF0IjoxNDQ3MjczMDk2LCJleHAiOjE0NDcyNzIxNTZ9.fQG	Payload	<pre>{ "sub": "1234567890", "name": "Eduardo Salas", "iat": 1516239022 }</pre>
MjczMDk2LCJleHAiOjE0NDcyNzIxNTZ9.fQG <i>Figura 2 - Token JWT separados por punto (elaboración propia, baso en la web de jwt.io [16])</i>	Signature	<pre>HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), TFM-UOC-2020) secret base64 encoded</pre> <p><i>Figura 3 - JWT formato JSON. (elaboración propia, baso en la web de jwt.io)</i></p>

Fuente: González (2020).

La Figura 11 ilustra la composición interna de JSON Web Token. Al lado izquierdo de la tabla se puede observar la cadena cifrada de cada parte correspondiente al cuerpo del JWT, en que se muestra cómo la información se codifica para garantizar la seguridad. A la derecha se pueden ver los datos originales, tal como fueron digitados, lo cual evidencia la transformación de la información antes de ser transmitida. Esta representación permite visualizar el proceso de codificación y autenticación, y asegura la protección de los datos durante la comunicación entre el cliente y el servidor.

Gestión de roles y permisos

En los sistemas informáticos la gestión de roles y permisos es fundamental para definir el acceso y las acciones que los usuarios pueden realizar dentro de una aplicación. Los roles representan distintos niveles de acceso, al agrupar usuarios bajo categorías según sus responsabilidades. Por otro lado, los permisos especifican cuáles acciones están permitidas para cada rol, y se garantizan así el control y la protección de la información. Al respecto, IBM (2021) menciona:

La utilización de roles le permite definir fácilmente permisos para todo un grupo de usuarios. Solo los gestores de acceso pueden crear nuevos roles y asignar usuarios a ese rol. Como parte de la creación de roles los gestores de acceso pueden personalizar el menú de navegación y los permisos para ese rol. (párr. 25).

Este mecanismo no solo mejora la seguridad sino que también optimiza la administración de recursos, con lo que se evitan accesos no autorizados, al tiempo que asegura

que los usuarios interactúen únicamente con las funciones que les corresponden. Además, la asignación de roles facilita la gestión en gran escala, para reducir la necesidad de configurar permisos individuales para cada usuario.

CAPÍTULO III: MARCO METODOLÓGICO

En el presente capítulo se presenta el enfoque metodológico utilizado para desarrollar un sistema web de gestión de Recursos Humanos en la empresa Quesada Vargas Construcciones, S.A. Se explican los enfoques de investigación considerados, así como el tipo de investigación por el que se orienta el estudio. Además, se exponen los métodos de recolección de información aplicados para analizar la situación actual del manejo descriptivo del personal, con el fin de fundamentar el diseño y el desarrollo del sistema propuesto. Esta sección es clave para garantizar la coherencia entre los objetivos del proyecto y las estrategias utilizadas para alcanzarlos.

Enfoques de la investigación

Los enfoques de investigación representan las distintas formas de abordar un fenómeno o problema, y orientan la manera en que se recopilan, analizan e interpretan los datos. Tradicionalmente se distinguen tres enfoques principales: el enfoque cuantitativo, que se basa en la recolección de datos numéricos y en el análisis estadístico; el enfoque cualitativo, que se centra en la interpretación de información no numérica para comprender fenómenos en profundidad; y el enfoque mixto, que integra elementos de ambos para lograr una visión más completa del objeto de estudio. La elección del enfoque adecuado depende de la naturaleza del problema y los objetivos del proyecto. Al respecto, Portilla (2022) menciona:

Al elaborar su diseño metodológico, el autor de un proyecto de investigación enunciará el enfoque del estudio, es decir, la precisión de la manera de ver sus variables del estudio que propone y el método de investigación. También indicará el nivel de investigación, teniendo en cuenta la función que cumplen en el estudio las precisiones sobre población y muestras, y luego técnicas e instrumentos de recolección de datos y las técnicas de procesamiento de información. (pp. 167-168).

En resumen, los enfoques de investigación son herramientas fundamentales para poder estudiar y entender mejor los fenómenos que de interés. Cada uno ofrece una manera distinta de observar el problema, ya sea desde los números y estadísticas, desde las experiencias y significados, o desde una combinación de ambos. Es importante reconocer que no hay un enfoque “mejor” que otro por sí solo, dado que todo depende de lo que se quiere investigar y cómo se espera hacerlo.

Enfoque cuantitativo

El enfoque cuantitativo se basa en la medición numérica de variables, empleando herramientas estadísticas para analizar datos objetivos. Busca establecer relaciones de causa y efecto y se utiliza en estudios que requieren precisión y generalización de resultados, como encuestas, experimentos y análisis de grandes conjuntos de datos. Al respecto, Molina y Garza (2021) mencionan: “Dicha metodología usualmente parte de cuerpos teóricos aceptados por la comunidad científica, con base en los cuales formula hipótesis sobre relaciones esperadas entre las variables que forman parte del problema que se estudia” (p. 152).

Para garantizar el rigor y la objetividad que lo caracterizan, el desarrollo de una investigación cuantitativa se articula mediante un proceso sistemático y secuencial. Al respecto, Molina y Garza (2021) mencionan:

La estructura para la investigación cuantitativa se integra por las siguientes fases: teoría, hipótesis, observaciones, recolección de datos, análisis de datos y resultados. Entre la teoría y la hipótesis se da la deducción; entre la hipótesis y la recolección de datos se presenta la operacionalización; entre la recolección y el análisis de datos se suscita el procesamiento de datos; entre el análisis de datos y los resultados se origina la inducción. (p. 153).

En resumen, el enfoque cuantitativo permite estructurar la investigación de manera lógica y controlada, que ofrece resultados concretos que facilitan la toma de decisiones. Su aplicación es valiosa cuando se requiere medir fenómenos, comparar situaciones o detectar patrones generales por medio de datos confiables y replicables.

Enfoque cualitativo

El enfoque cualitativo se enfoca en la interpretación de significados, experiencias y percepciones. Se basa en métodos como entrevistas, observaciones y análisis de contenido para comprender fenómenos desde una perspectiva subjetiva y contextual. Es común en investigaciones sociales y estudios de comportamiento. Al respecto, Molina y Garza (2021) mencionan:

Dicha metodología se interesa por captar la realidad social a través de la mirada de la gente que está siendo estudiada, es decir, a partir de la percepción que tiene el sujeto de su propio contexto. Plantea que la realidad no es exterior a quien la examina, dado que existe una relación estrecha entre el sujeto y el objeto de conocimiento. (p. 154).

Este enfoque no se rige por una estructura rígida sino que sigue un proceso flexible y adaptivo que permite al investigador adentrarse en la realidad estudiada. De esta manera se valora la voz de los participantes y se reconoce que el conocimiento se construye a partir de sus vivencias, interpretaciones y contextos particulares.

En resumen, el enfoque cualitativo ayuda a comprender mejor lo que las personas piensan, sienten o experimentan en una situación específica. Más que enfocarse en números busca conocer las opiniones y vivencias de quienes forman parte del problema, lo que permite tener una visión más humana y cercana del tema que se estudia.

Enfoque mixto

El enfoque mixto combina elementos del enfoque cuantitativo y el cualitativo para proporcionar una visión más completa del fenómeno estudiado. Permite integrar datos numéricos con interpretaciones subjetivas, con lo que enriquece el análisis y facilita la validación de resultados desde distintas perspectivas. Al respecto, Luza *et al* (2023) mencionan:

Los métodos mixtos de investigación son enfoques que combinan tanto elementos cualitativos como cuantitativos en un estudio o investigación. Estos métodos buscan aprovechar las fortalezas de ambos enfoques para obtener una comprensión más completa y profunda de un fenómeno o problema de investigación. (p. 77).

A partir de esta combinación de enfoques, surgen múltiples beneficios que hacen del enfoque mixto una herramienta valiosa para el análisis investigativo. Al integrar distintos tipos de datos, se mejora la calidad de la información recopilada y se logran resultados más completos y equilibrados, lo cual fortalece la toma de decisiones en el desarrollo de la investigación. Al respecto, Luza, et al. (2023) mencionan:

Las ventajas de los métodos mixtos es que permiten una comprensión más profunda y rica del fenómeno estudiado al combinar la amplitud y generalización de los datos cuantitativos con la riqueza contextual y la comprensión en profundidad proporcionada por los datos cualitativos. Además, el uso de métodos mixtos puede ayudar a validar y enriquecer los hallazgos, al permitir la triangulación de datos y el análisis complementario. (p. 77).

En conclusión, el enfoque mixto resulta muy útil porque permite ver el problema desde diferentes puntos de vista. Como estudiantes, es una forma que ayuda a entender mejor la situación al combinar lo que se puede medir con lo que se puede observar o escuchar de las personas. Esto hace que los resultados sean más claros y completos, lo cual es muy útil para tomar decisiones en el proyecto.

Enfoque de investigación seleccionado

El enfoque seleccionado para la investigación es el mixto, debido a que se van a combinar tanto datos cualitativos (como entrevistas y observación de procesos), como datos cuantitativos (como encuestas, tiempos en procesos y errores repetidos). Esta combinación de métodos cualitativos y cuantitativos permite obtener una visión más completa del problema, ya que no solo se recopilan números o estadísticas, sino también opiniones y experiencias de las personas involucradas en el proceso. De esta forma se logra entender tanto lo que está ocurriendo como por qué ocurre. Además, este enfoque permite comparar la información desde diferentes puntos de vista, lo cual ayuda a tomar mejores decisiones para el desarrollo del sistema de gestión de recursos humanos que se propone.

Además, este enfoque adapta muy bien el objetivo del proyecto, que busca desarrollar un sistema web para mejorar la gestión de recursos humanos en la empresa. Gracias al enfoque mixto se pueden analizar desde diferentes perspectivas las necesidades del sistema, lo que ayuda a proponer soluciones más acertadas. Por último, esta integración de métodos también permite validar mejor los resultados, ya que se comparan datos de diferentes fuentes para llegar a conclusiones más confiables.

Tipos de investigación

Cuando se realiza una investigación es importante saber qué tipo se va a utilizar, ya que cada uno tiene una forma distinta de abordar el problema o tema que se quiere estudiar. Los tipos de investigación ayudan a organizar el trabajo y a definir qué se va a buscar, cómo se va a hacer y qué tanto se pretende conocer del fenómeno. Existen varias formas de clasificar la investigación, pero entre las más comunes están: investigación exploratoria, descriptiva, correlacional y explicativa. Cada una tiene sus propias características y objetivos y se eligen dependiendo de qué tanto se conoce el tema y lo que se espera descubrir. A continuación se explica cada una de ellas con más detalle.

Investigación exploratoria

La investigación exploratoria es un tipo de estudio que se utiliza cuando todavía no se sabe mucho sobre el tema que se quiere investigar. Sirve para tener una idea general, conocer mejor el problema y empezar a reunir información que luego puede ayudar a hacer estudios

más profundos. Es como el primer paso cuando se quiere investigar algo nuevo o poco estudiado. No busca dar respuestas definitivas, sino más bien abrir el camino para futuras investigaciones, haciendo preguntas, observando y recolectando datos que orienten mejor el trabajo. Al respecto, Albornoz *et al.* (2023) mencionan:

Se plantean cuando no existe un cuerpo teórico abundante que ilumine el estudio de un fenómeno observado; y los resultados que se obtengan sean un aporte al reconocimiento e identificación de los problemas. No hay preguntas que conduzcan a problemas precisos, se exploran áreas problemáticas. Se trata de una investigación cualitativa; por lo que no se requiere manejo estadístico. (p. 91).

En resumen, la investigación exploratoria es una forma de empezar a conocer un tema que todavía no está muy claro o no ha sido muy estudiado. Es útil para tener un panorama general y saber por dónde seguir investigando. Aunque no da respuestas exactas, sí ayuda a entender mejor la situación y a formular preguntas más precisas para futuras investigaciones. Es como el primer paso en el camino del conocimiento, especialmente útil cuando todo es nuevo o poco conocido.

Investigación descriptiva

La investigación descriptiva se usa cuando ya se tiene un conocimiento básico del tema y se quiere describir cómo es o cómo se comporta una situación, grupo de personas, proceso u objeto. No busca explicar por qué suceden las cosas, sino más bien decir cómo son. Se enfoca en observar, registrar y analizar datos que permitan tener una imagen clara de la realidad que se está estudiando. Es muy útil cuando se necesita conocer características, comportamientos o condiciones de algo en particular, como por ejemplo saber cuántas personas usan una aplicación o cuáles son las funciones que más utilizan. Al respecto, Albornoz *et al.* (2023) mencionan:

Describe fenómenos sociales o clínicos en una circunstancia temporal y geográfica determinada. Desde el punto de vista cognoscitivo su finalidad es describir y desde el punto de vista estadístico su propósito es estimar parámetros. La estadística consiste en estimar frecuencias y/o promedios y otras medidas univariadas. Ej. Los estudios de frecuencia de la enfermedad: incidencia y prevalencia. (p. 91).

En resumen, la investigación descriptiva es una herramienta muy útil cuando se quiere entender mejor cómo son las cosas en un momento y lugar específico. No intenta buscar causas o relaciones profundas sino que se enfoca en observar y describir lo que ocurre. Es una forma

clara y ordenada de reunir datos que ayudan a tener una visión general del tema, lo cual puede ser un buen paso para estudios más complejos más adelante.

Investigación correlacional

La investigación correlacional se usa cuando se quiere saber si existe una relación entre dos o más variables. No busca explicar por qué pasa algo, pero sí ayuda a ver si hay una conexión o si una cosa cambia cuando cambia otra. Por ejemplo, si se quiere saber si las horas de estudio tienen algo que ver con las calificaciones de los estudiantes, este tipo de investigación puede ayudar a ver si hay una relación entre esas dos variables. Es importante entender que encontrar una relación no significa que una cosa cause la otra, solo que podrían estar relacionadas de alguna forma. Este tipo de estudio es útil cuando ya se tiene información básica del tema y se quiere profundizar un poco más. Al respecto, Albornoz *et al* (2023) mencionan:

Alcance correlacional: es aquella que sirve para determinar la relación positiva o negativa entre dos o más conceptos. Esta se realizará con base en un mismo patrón para el mismo grupo de estudio. La investigación correlacional no solo describe sino también analiza y relaciona. (p. 128).

En resumen, la investigación correlacional es muy útil cuando se quiere saber si dos cosas están conectadas de alguna manera. No dice si una causa la otra, pero sí permite ver si cambian juntas. Es una forma de avanzar en el estudio de un tema, ya que va más allá de solo describir, buscando patrones o relaciones que pueden ser importantes para futuras investigaciones más profundas.

Investigación explicativa

La investigación explicativa se utiliza cuando se quiere entender por qué ocurren las cosas. No solo se enfoca en describir un hecho o ver si dos cosas están relacionadas, sino que busca encontrar las causas o razones detrás de un fenómeno. Por ejemplo, si ya se sabe que las personas con más horas de estudio tienen mejores calificaciones (relación), la investigación explicativa trataría de entender por qué pasa eso. Este tipo de investigación se usa cuando ya hay bastante información sobre el tema y se quiere profundizar más para dar respuestas claras y detalladas sobre cómo y por qué suceden las cosas. Al respecto, Portilla (2022) menciona:

Entre las investigaciones explicativas destacan los estudios sobre búsquedas de causas, los cuales indagan acerca de causas en vistas de que los afectos llaman la atención de los investigadores y hace que se preocupen por las causas que producen esos efectos observados. En esta clase de estudios el diseño ideal correspondiente exige hacer un experimento. (p. 172).

En conclusión, la investigación explicativa es ideal cuando ya se tiene bastante información sobre un tema y se quiere ir más allá, buscando entender las causas de lo que ocurre. Es como dar el siguiente paso después de haber observado y encontrado relaciones, ya que trata de responder el “por qué” de los hechos. Aunque es más profunda, sigue siendo parte del proceso de aprender y descubrir con más claridad cómo funciona todo lo que se está investigando.

Tipo de investigación seleccionado

El tipo de investigación elegido es el descriptivo, ya que permite detallar de manera clara y ordenada las características del problema actual en la gestión de recursos humanos dentro de la empresa. Por medio de este enfoque se pueden identificar con precisión aspectos como los procesos que presentan fallas, la frecuencia de errores, el uso actual de la tecnología y las necesidades de los usuarios, lo cual es fundamental para proponer un sistema web que realmente responda a la realidad observada.

Además, este tipo de investigación permite recolectar información concreta sobre la situación actual, lo que facilita establecer una base sólida para tomar decisiones informadas durante el desarrollo del sistema. También se complementa de forma natural con el enfoque mixto seleccionado, ya que los datos descriptivos obtenidos por medio de encuestas, entrevistas y observaciones fortalecen la comprensión general del contexto y contribuyen a una propuesta de solución más completa y ajustada a las necesidades reales.

Fuentes de información

Las fuentes de información son todos aquellos recursos que un investigador consulta para poder responder las preguntas de su trabajo. Estas fuentes le permiten conocer lo que otras personas ya han estudiado o descubierto sobre un tema, y sirven como base para desarrollar su propia investigación. Según su origen y cómo se presenta la información, las fuentes se pueden clasificar en tres tipos: fuentes primarias, que ofrecen información directa u original; fuentes

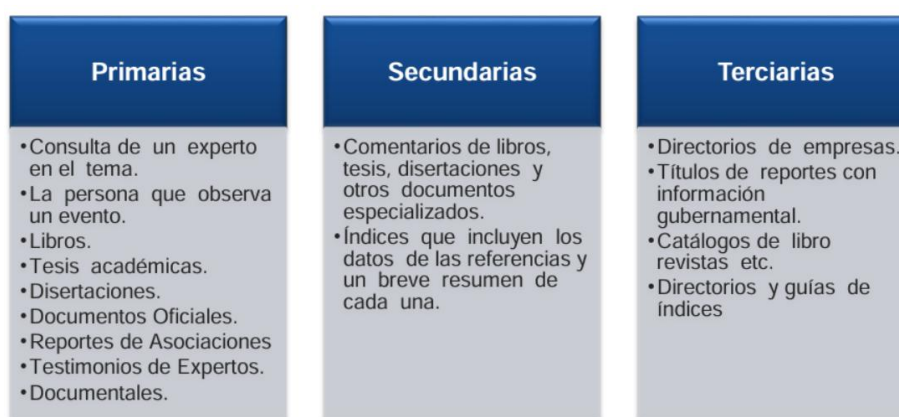
secundarias, que analizan o interpretan información ya existente; y fuentes terciarias, que ayudan a organizar y localizar otras fuentes. Cada una cumple un papel importante en el proceso de investigación. Al respecto, Ramírez (2021) menciona:

Son todos aquellos medios donde se obtiene la información, que satisfacen las necesidades de conocimiento de una situación o problema presentado y que posteriormente será utilizado para lograr los objetivos esperados. Las fuentes de información se dividen en primarias, secundarias y terciarias. (p. 6).

La Figura 12 muestra los tres tipos de fuentes de información y algunos de los medios que se consideran como parte de cada fuente:

Figura 12

Fuentes de información



Fuente: Ramírez (2021).

Como se muestra en la Figura 12, cada tipo de fuente tiene ejemplos claros que ayudan a entender mejor su función en una investigación. Por ejemplo, las entrevistas y encuestas son fuentes primarias porque provienen directamente de las personas, mientras que los libros y artículos científicos suelen ser secundarios porque explican o analizan la información ya conocida. Las fuentes terciarias, como enciclopedias o los catálogos, son útiles para iniciar la búsqueda y encontrar otros materiales confiables.

En conclusión, conocer los diferentes tipos de fuentes de información es fundamental para realizar una buena investigación. Saber dónde buscar, cómo utilizar cada fuente y para qué sirve, permite que el trabajo tenga una base sólida y más completa.

Fuentes primarias

Las fuentes de información primarias son aquellas que provienen directamente de las personas que vivieron, experimentaron o generaron la información. Se consideran originales

porque no han sido modificadas ni interpretadas por otros. Son importantes porque ofrecen una visión directa de los hechos, lo que permite analizarlos de manera más objetiva. Estas fuentes sirven como base para construir ideas, planear hipótesis y llegar a conclusiones en una investigación. Algunos de los ejemplos de fuentes primarias pueden ser fotografías, cartas, vídeos, diarios personales o cualquier otro documento creado por la persona involucrada. También se consideran aquellas fuentes que se obtienen mediante la interacción directa con las personas, como entrevistas, encuestas y testimonios. Al respecto, Dupuis (2020) menciona: “Son documentos originales [es decir, no se trata de otro documento o cuenta] y reflejan el punto de vista individual de un participante u observador” (párr. 4).

La definición ofrecida por Dupuis (2020) respalda y complementa la idea central de que las fuentes primarias son valiosas por su autenticidad y cercanía a los hechos. Al tratarlas como originales que reflejan directamente la perspectiva de los involucrados se reafirma su importancia en el proceso investigativo.

En conclusión, las fuentes primarias representan un recurso esencial para cualquier investigación seria, ya que permiten el acceso a la información en su forma más pura. Su análisis contribuye a una comprensión más profunda y directa de los temas estudiados, que sirven como punto de partida para interpretar la realidad y formular nuevos conocimientos.

Fuentes secundarias

Las fuentes de información secundarias son aquellas que no provienen directamente de la persona que vivió o generó los hechos, sino que han sido elaborados por alguien que interpreta, analiza o resume la información original. Su función principal es ayudar a entender mejor el contexto, el significado o las consecuencias de los eventos descritos en las fuentes primarias. Por ejemplo, un libro de historia en que se estudia y comentan documentos oficiales de un gobierno involucrado en la Segunda Guerra Mundial sirve para explicar el contenido de esos documentos y ponerlos en perspectiva. Al respecto, Life Pacific University (2023) menciona: “Una fuente secundaria es aquella que fue creada más tarde por alguien que no tuvo experiencia de primera mano o participó en los eventos” (párr. 1).

Esta definición refuerza la idea de que las fuentes secundarias surgen a partir de un análisis posterior y desde una cierta distancia de los hechos. Esto permite que quienes no vivieron el evento puedan aportar interpretaciones que enriquecen la comprensión del tema.

Algunos ejemplos comunes de fuentes secundarias incluyen artículos académicos que analizan otros estudios, bibliografías y reseñas, libros de texto, documentales. Esas fuentes no

presentan los hechos tal como ocurrieron, sino que explican o interpretan con base en otras evidencias. Por ejemplo, un documental analiza la vida de una figura histórica a partir de documentación de la persona, como cartas, entrevistas previas, vídeos, entre otros.

En resumen, las fuentes secundarias de información son fundamentales para complementar la investigación, ya que ofrecen una mirada más amplia y crítica de la información original. Aunque no tienen el mismo nivel de cercanía que las fuentes primarias, aportan contexto, explicaciones y distintas perspectivas que facilitan el análisis profundo de un tema. Saber usarlas correctamente permite desarrollar investigaciones más sólidas y bien fundamentadas.

Fuentes terciarias

Las fuentes de información terciarias son aquellas que se encargan de recopilar, organizar y resumir datos provenientes tanto de fuentes primarias como secundarias, con la particularidad de que no analizan ni interpretan. Su función principal es facilitar el acceso a información ya publicada, actuando como una herramienta de consulta útil para los investigadores que desean ampliar o verificar sus fuentes. Algunos de los ejemplos de fuentes terciarias son las enciclopedias, diccionarios, guías, almanaques, bibliografías e índices temáticos, entre otros. Al respecto, Dupuis (2020) menciona:

Una fuente terciaria consolida y organiza las fuentes primarias y secundarias juntas en una sola fuente para facilitar el acceso rápido a la información. Las fuentes terciarias son buenos puntos de partida para proyectos de investigación porque a menudo extraen el significado esencial o los aspectos más importantes de grandes cantidades de información conveniente. (párr. 19).

Tal como señala Dupuis (2020), este tipo de fuentes resultan útiles en las etapas iniciales de una investigación, ya que permiten ubicar rápidamente información clave y orientan al investigador hacia materiales más profundos. Funcionan como una guía general que facilita la búsqueda y organización de contenidos relevantes, y es un apoyo importante para comenzar con una base sólida.

En resumen, las fuentes terciarias no aportan contenido original ni análisis, pero su valor es la manera en que agrupan y presentan información ya existente. Son especialmente útiles cuando se requiere una visión general o cuando se busca estructurar de forma ordenada el camino hacia otras fuentes más detalladas. Por eso, conocer su función y saber utilizarlas correctamente permite avanzar con mayor seguridad en el proceso de investigación.

Variables de investigación

Una variable de investigación es un atributo, cualidad, característica que se puede observar, medir o analizar dentro de un estudio, y que guarda relación o influencia con el tema que se está investigando. En otras palabras, es aquello que se quiere entender, examinar, y que puede variar según la situación o el enfoque del proyecto. En todo estudio de investigación se deben considerar al menos dos variables para poder establecer relaciones o comparaciones entre ellas. Al respecto, Brage (2022) mencionan:

Una variable es una propiedad que puede variar y cuya variación es susceptible de medirse. Ejemplo de variable: método de enseñanza, aprovechamiento escolar, edad, género, inteligencia, motivación, ingreso económico, violencia, etc.

Las variables adquieren valor para la investigación científica cuando se pueden relacionar unas con otras, porque pasan a formar parte de hipótesis. (p. 69).

Como menciona Brage *et al* (2022), se entiende que una variable adquiere verdadero valor en una investigación cuando puede medirse, manipularse o relacionarse con otras, ya que esto permite construir hipótesis. De lo contrario, si una variable no puede vincularse con otros elementos del estudio o no es posible analizar su comportamiento, no aporta información relevante ni contribuye al objetivo del trabajo investigativo.

En conclusión, conocer y definir adecuadamente las variables es un paso clave para el desarrollo de una investigación, ya que a partir de ellas se construyen las hipótesis, se determinan los métodos de análisis y se da sentido a los resultados obtenidos.

Variable conceptual

La variable conceptual o también llamada constitutiva es la forma en que se define teóricamente una variable dentro de una investigación, se describe la definición teórica o abstracta de una variable, es decir, cómo se entiende o conceptualiza la variable. Al respecto, Valenciano (2022) menciona:

La dimensión constitutiva tiene por finalidad definir la variable desde la relación *difiniendum-definiens*, es decir, desde la correspondencia entre lo que se define y el conjunto de términos a los que recurre la persona investigadora que son necesarios para describir semánticamente a la variable en su esencia. Una interpretación de los aportes señalados por Copi y Cohen (2013) y Gianella (2010) permite establecer que para que la

definición de una variable, desde la dimensión constitutiva deben cumplirse tres condiciones básicas:

- Exponer el significado de la variable sin usarla como término de su propia definición.
- Enunciar el significado con base en las características y cualidades que mejor representan la mutación que experimenta la identidad de la cosa.
- Determinar que el significado dado a la variable sea preciso, esto es, que no exceda o carezca en su enunciado de las características o cualidades esenciales de esta. (p. 4).

Es decir, se refiere a lo que significa esa variable desde un punto de vista general o académico, sin entrar aún en cómo se va a medir. Esta definición sirve como base para que el lector entienda de qué se está hablando y cuál es el enfoque del estudio respecto a ese concepto.

En resumen, la variable conceptual permite establecer con claridad qué se entiende por cada término dentro de la investigación, lo cual es fundamental para evitar ambigüedades o confusiones. Definirla correctamente desde el inicio facilita la comprensión del estudio, tanto del investigador como para quienes lo lean, ya que brinda una base teórica clara sobre la cual se desarrollarán las demás etapas del proceso investigativo.

Variable operacional

La variable operacional es la forma en que se lleva a la práctica la variable conceptual, es decir, cómo se convierte en algo que se puede observar y medir. Esta variable responde a la pregunta: ¿cómo se mide?, ya que su propósito es definir con claridad los indicadores o criterios que se usarán para obtener datos concretos en la investigación. Al respecto, Valenciano (2022) menciona:

Concretada la definición constitutiva de la variable según corresponda a una categoría y subcategoría, la persona investigadora debe proceder a definirla desde la dimensión operativa. En decir, desde el conjunto de operaciones empíricas, indicadores o situaciones mediante las cuales se podrá observar, identificar, medir y valorar la mutación que experimenta la característica o cualidad de la cosa. (p. 6).

Puede decirse que la identificación operacional de una variable es un paso clave en el diseño de toda investigación, ya que permite transformar ideas abstractas en elementos medibles y concretos. Esta etapa asegura que lo que se pretende estudiar pueda ser observado de forma sistemática, coherente y objetiva. Al establecer con claridad cómo se medirá cada

variable se garantiza mayor precisión en la recolección de datos, y, por ende, mayor validez en los resultados obtenidos.

Variable instrumental

La variable instrumental es el instrumento o medio que se utiliza para obtener la información, y sirve como puente entre la teoría y la medición. El instrumento por utilizar debe estar previamente validado por otros autores. Al respecto, Benites (2024) menciona:

Una variable instrumental es una tercera variable introducida en el análisis de regresión que está correlacionada con la variable predictora, pero no correlacionada con la variable de respuesta. Al usar esta variable, es posible estimar el verdadero efecto causal que alguna variable predictora tiene sobre una variable de respuesta. (párr. 6).

Lo que plantea Benites (2024) ayuda a comprender que la variable instrumental no solo cumple una función práctica como herramienta de medición, sino que también puede tener un rol más analítico en algunos enfoques de investigación. En la mayoría de los estudios aplicados, como los de carácter social, educativo o de sistemas, se entiende como el recurso concreto que permite obtener los datos, como encuestas, cuestionarios, pruebas u otros medios similares.

Cuadro de variables

En toda investigación es fundamental dejar claro qué se va a estudiar y cómo se va a hacer. Por eso, antes de recolectar información y analizar datos se requiere organizar las variables que estarán presentes en el estudio. Esto se logra por medio del cuadro de variables, que permite presentar de forma clara y ordenada los elementos más importantes que se medirán. Al respecto, Carvajal (2023) menciona:

La operacionalización de variables se presenta en una tabla o cuadro que recoge de forma ordenada, los siguientes aspectos: definición conceptual de las variables, la definición operacional, las dimensiones (en caso de variable compleja), los indicadores y la escala de medición. (p. 6).

En esta investigación se ha construido un cuadro de variables con cinco columnas: objetivos, variable, definición conceptual, definición operacional e instrumento de medición. Esta estructura permite mantener el orden lógico de los objetivos y la forma en que cada variable será tratada en el estudio.

A continuación, la Tabla 13 muestra la información relacionada con la operacionalización de las variables del estudio. En ella se detallan los objetivos específicos, las variables correspondientes a cada objetivo, su definición conceptual y operacional, así como el instrumento que será utilizado para recolectar los datos, con el fin de garantizar una adecuada medición y coherencia en la investigación.

Tabla 13

Cuadro de categorías, variables e indicadores

Objetivos	Variable	Variable Conceptual	Variable Operacional	Variable Instrumental
Analizar los requerimientos funcionales y no funcionales del sistema web de recursos humanos.	Requerimientos funcionales Requerimientos no funcionales	Según Reyes (2020) los requerimientos funcionales se definen como: "son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas" (p. 4). Según Serna (2021), los requerimientos no funcionales se definen como: "La complejidad de los sistemas software está determinada por la funcionalidad y por los aspectos de calidad: rendimiento, fiabilidad, exactitud, seguridad y usabilidad. Estos aspectos se conocen como requisitos no-funcionales del software" (p.280).	Entrevistas Encuestas Observación	Guía de entrevistas Cuestionarios Guía de Observación
Diseñar la base de datos y la estructura del sistema de gestión de recursos humanos utilizando MySQL con la herramienta Workbench.	Diseño de Base de datos Estructura del sistema	Según Casas (2020) el diseño de bases de datos es: "Por lo tanto, el diseño de una base de datos es el proceso en que se define la estructura de los datos que debe tener la base de datos de un sistema de información determinado" (p. 7). Según Vargas (2024) refiriéndose a la estructura del sistema: "La estructura del software es como el esqueleto de una página web, es la base sobre la cual se construye y organiza todo el contenido. Es como un mapa detallado que guía al programador y al diseñador en el proceso de desarrollo" (párr. 1).	Diseño Base de Datos Diagrama de Casos de Uso Diagrama de Clases Diagrama de Arquitectura	Workbench 8.0.36 Draw.io Lucidchart Microsoft Visio
Programar los diferentes módulos del sistema de gestión recursos humanos que cumplan con los requerimientos definidos, utilizando Java.	Módulos programados	Según Vargas (2024) menciona lo siguiente de los módulos: "Los módulos son las unidades básicas de un software, cada uno con una funcionalidad específica. Estos módulos se dividen en diferentes partes del programa para facilitar su mantenimiento y reutilización. Cada módulo tiene un conjunto de tareas y procesos que se encargan de realizar.	Creación Base de Datos Desarrollo del prototipo	Workbench 8.0.36 Visual Studio Code

Realizar pruebas tanto individuales como de integración del sistema de recursos humanos en un entorno de prueba, garantizando la funcionalidad de cada módulo.	Pruebas funcionales Pruebas de integración	Según Melgar (2020) citando a SSTQB (2018), se refiere a las pruebas funcionales como: "Se ve en el nivel de pruebas del sistema, son pruebas que evalúan lo que el sistema debe hacer. Según Tamushi (2022) menciona al respecto sobre las pruebas de integración: "Las pruebas de integración de software son la herramienta que conjunta cada uno de los módulos de un sistema para comprobar su funcionamiento entre sí" (párr. 7).	Ejecución de pruebas funcionales Ejecución de pruebas de integración de módulos	de de de	Casos de prueba de Guía de verificación Postman JUnit
----------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------	----------------	-------------------------------------------------------------

Fuente: Elaboración propia, 2025.

Población

La población se refiere al grupo de personas, elementos o casos que tienen características comunes y que son de interés para el estudio de investigación. Es decir, es el conjunto del cual se quiere obtener información para responder a los objetivos de la investigación. Al respecto, Meza *et al* (2021) mencionan: "Población es un conjunto definido, limitado y accesible del universo que forma el referente para la elección de la muestra. Es el grupo al que se intenta generalizar los resultados" (p. 84).

En este proyecto, la población de investigación está conformada por todos los colaboradores de la empresa Quesada Vargas Construcciones, S.A., incluido tanto el personal del área de Recursos Humanos como los demás colaboradores de la empresa. Debido a que la empresa cuenta con un total de 15 trabajadores, se decidió aplicar la encuesta a la totalidad de ellos, ya que es un grupo manejable y accesible para obtener la información necesaria de forma directa.

Muestra

La muestra es una parte de la población que se elige para participar en una investigación. En lugar de trabajar con todo el grupo se seleccionan algunas personas que representan a los demás, con el fin de obtener información útil para el estudio. Al respecto, Pérez y García (2023) mencionan:

Una buena muestra es aquella que es representativa de la población objeto de estudio (del universo); no obstante, los criterios de representatividad no son unívocos. Generalmente las metodologías cuantitativas se basan en criterios sustentados en la diversidad del universo, es decir, una buena muestra es aquella que puede dar cuenta de toda la diversidad

existente; por lo tanto, cuanto más diverso sea el universo, mayor (tanto en diversidad como en tamaño) será la muestra. (p. 127).

Para calcular el número de personas que deben formar parte de la investigación se usará una fórmula diseñada para poblaciones pequeñas o finitas. A continuación se detalla la fórmula por utilizar:

$$n = \frac{K^2 N q p}{e^2 (N - 1) + K^2 q p}$$

Fuente: Universidad de las Ciencias y el Arte de Costa Rica, (s.f).

En que n representa el tamaño de la muestra, N el tamaño de la población, K representa el nivel de confianza, p es la proporción esperada, q representa la probabilidad de fracaso y la e representa la precisión (margen de error).

El valor K es la cantidad de desviaciones estándar en que una proporción determinada se aleja de la media. A continuación, en la siguiente figura se muestran los valores correspondientes a K, con el nivel de confianza correspondiente a cada valor.

Figura 13

Valores de K con su nivel de confianza

K	1,15	1,28	1,44	1,65	1,96	2	2,58
Nivel de confianza	75%	80%	85%	90%	95%	95,5%	99

Fuente: Universidad de las Ciencias y el Arte de Costa Rica, (s.f).

Dado que la población objeto de estudio está conformada por únicamente 15 personas, no se consideró necesario aplicar la fórmula de muestreo, y se optó por un abordaje censal.

Instrumentos de recolección de datos

Los instrumentos de recolección de datos son herramientas que se utilizan para obtener la información necesaria de una investigación. Sirven para recopilar datos de manera ordenada y confiable, ya sea mediante encuestas, entrevistas, cuestionarios, observaciones, entre otros más. La principal función es ayudar al investigador a conocer mejor el problema que se está estudiando, para facilitar la obtención de información directa de las personas o situaciones relacionadas con el tema. Al respecto, Meza *et al* (2021) mencionan: “Un instrumento será apropiado cuando mida el concepto que se quiere medir, esto significa, que el instrumento es

capaz de dar valor estadístico a los datos que recoja por medio de cuestionarios, entrevistas o de cualquier otra técnica que establezca para la investigación” (p. 140).

Para recolectar información en este proyecto se utilizarán entrevistas, encuestas y la observación, ya que son métodos comunes que permiten obtener datos importantes. Estos instrumentos ayudarán a comprender mejor la situación que se investiga y a reunir la información necesaria para alcanzar los objetivos propuestos.

Entrevistas

La entrevista es una forma de obtener información conversando directamente con una persona. Por medio de preguntas previamente definidas el investigador puede conocer opiniones, experiencias o conocimientos relacionados con el tema que se está estudiando. Es una herramienta útil porque permite obtener respuestas más profundas y detalladas, ya que se da un intercambio directo entre quien pregunta y quien responde. Al respecto, Medina *et al* (2023), quien cita a Kerlinger (1975) y Kerlinger y Lee (2001) menciona:

La entrevista es una técnica de investigación que involucra la interacción directa entre el entrevistador y el entrevistado con el objetivo de obtener información y opiniones detalladas sobre un tema específico. La entrevista se utiliza ampliamente en diferentes campos, como la psicología, la sociología, la antropología y la investigación de mercado, para investigar actitudes, comportamientos, motivaciones y experiencias personales. (p. 26).

La entrevista es una herramienta muy valiosa para obtener información directa y detallada de personas que tienen conocimientos o experiencias relacionadas con el tema de estudio. Gracias a esta interacción personal es posible profundizar en aspectos que otras técnicas no permiten. Por ello, en este proyecto de investigación se empleará la entrevista como uno de los instrumentos de recolección de datos, ya que permitirá obtener información relevante que ayudará a cumplir con los objetivos propuestos.

Encuestas

Las encuestas son una forma de recopilar información haciendo preguntas a un grupo de personas. Se utilizan para conocer opiniones, actitudes o experiencias sobre un tema específico. Generalmente, las personas responden de forma anónima y las respuestas se analizan para sacar conclusiones que ayuden en una investigación o estudio. Al respecto, Abad,

et al (2021) mencionan: “La encuesta es propia de la investigación con enfoque cuantitativo y, se considera, de manera general, como un cuestionario estructurado que se aplica a la muestra de una población, y está diseñada para obtener información específica de los participantes” (p. 48).

Se puede decir que las encuestas por medio de cuestionarios resultan ser una herramienta útil y práctica para obtener información clara y directa de las personas involucradas. Por eso, en este proyecto se optará por utilizar encuestas con cuestionarios, ya que permiten recopilar datos relevantes de forma ordenada y esto facilitará el análisis necesario para cumplir con los objetivos de la investigación.

La observación

La observación es un instrumento que permite recolectar información al ver y registrar lo que ocurre en un lugar o situación determinada. Consiste en prestar atención de manera directa a comportamientos, acciones o hechos, sin intervenir, con el fin de entender mejor lo que se está estudiando. Es una forma útil de obtener datos reales tal como suceden en la práctica, lo que ayuda a tener una visión más completa del problema de investigación. Al respecto, *Abad et al* (2021) mencionan:

La observación como técnica de investigación científica es un proceso riguroso que permite conocer, de forma directa, el objeto de estudio para luego describir y analizar situaciones sobre la realidad estudiada. Para garantizar el cumplimiento de los requisitos de la observación científica es preciso elaborar, antes de ir a recoger los datos, un plan y una guía de observación. (p. 38).

Se puede afirmar que la observación es una herramienta valiosa para obtener información directa de la realidad que se investiga. Gracias a esta técnica es posible registrar hechos tal como suceden, sin alterar el entorno. Por esta razón, en el presente proyecto se utilizará la observación como uno de los instrumentos de recolección de datos, ya que permitirá analizar de manera más precisa el contexto y las acciones relacionadas con el tema de estudio.

CAPÍTULO IV: ANÁLISIS DE RESULTADOS

En este capítulo se presentan los hallazgos obtenidos a partir de la aplicación de los instrumentos de recolección de información, específicamente la guía de entrevista, la guía de observación y el cuestionario, realizados en la empresa Quesada Vargas Construcciones S.A. El propósito de este análisis es identificar la situación actual de los procesos administrativos relacionados con la gestión del personal, así como las principales limitaciones que enfrentan en su operatividad diaria.

Los resultados permiten evidenciar las necesidades, carencias y oportunidades de mejora en la gestión de recursos humanos dentro de la organización. A partir de ellos, se establecen los elementos fundamentales que debe contemplar el desarrollo del sistema propuesto, con el fin de ofrecer soluciones concretas a los problemas detectados y facilitar una administración más ágil, confiable y eficiente de la información del personal.

Observación

La observación realizada en la empresa Quesada Vargas Construcciones, S.A. el día 5 de setiembre del 2025 tuvo como propósito identificar la forma en que actualmente se desarrollan las labores en el área de Recursos Humanos. Por medio de este proceso fue posible reconocer prácticas, herramientas utilizadas y dificultades que enfrenta la encargada en sus tareas y procesos. Los resultados de la observación se detallan a continuación, como un aporte fundamental para el desarrollo del sistema propuesto.

Entre los principales hallazgos se identificó el uso excesivo de hojas de cálculo de Excel para la gestión del personal. Estas no cuentan con validaciones que garanticen la integridad de los datos ni con mecanismos de integración que reduzcan los errores por transcripción. Esta dependencia de múltiples archivos genera duplicidad de tareas y procesos, provocando desorden y retrasos importantes en el trabajo del departamento de Recursos Humanos.

Como consecuencia, los procesos de consulta de información y elaboración de reportes suelen tardar horas, e incluso días, ya que requieren revisar hojas de cálculo, transcribir datos y verificarlos en más de una ocasión para minimizar errores. Además, parte de la información se almacena únicamente en documentos físicos, los cuales se guardan en carpetas y cajas de años anteriores. Esta práctica no solo dificulta el acceso a los datos, sino que también ha reducido el espacio de trabajo disponible, obligando a trasladar los archivos a un lugar más

alejado. Por ello, cuando se necesita información de años anteriores, es necesario desplazarse hasta esa otra área y revisar manualmente las cajas para ubicar los documentos requeridos.

La observación realizada permitió constatar que el área de Recursos Humanos depende en gran medida de procesos manuales y del uso de hojas de cálculo, lo cual ocasiona duplicidad de tareas, dificultades para el acceso a la información y retrasos en la generación de reportes. Asimismo, se evidenció la necesidad de contar con un sistema que centralice y organice los datos, reduzca el uso de archivos físicos y ofrezca herramientas que faciliten el control de planilla, vacaciones, ausencias y demás gestiones del personal. Estos hallazgos son de gran valor para orientar el diseño y desarrollo del prototipo del sistema de gestión de recursos humanos propuesto.

Para revisar en detalle la guía de observación empleada en el estudio puede consultarse el Apéndice C: Guía de observación.

Entrevista

Con el propósito de obtener información cualitativa relevante para el desarrollo del sistema de gestión de Recursos Humanos se llevó a cabo una entrevista dirigida a la encargada del área de recursos humanos de la empresa Quesada Vargas Construcciones, S.A., el día 05 de setiembre de 2025 y para el detalle completo de las preguntas referirse al apéndice A. Este instrumento permitió recopilar percepciones, experiencias y necesidades específicas relacionadas con la administración del personal y los procesos internos que actualmente se gestionan de manera manual.

El análisis de la entrevista se presenta a continuación, organizado de manera que facilite la identificación de los principales hallazgos, tendencias y puntos críticos señalados por la participante.

En relación con la primera pregunta, referida a las principales tareas que realiza actualmente en el área de recursos humanos y las herramientas que utiliza, la entrevistada señaló que entre sus funciones más frecuentes se encuentra la elaboración de la planilla, el cálculo del aguinaldo, el control de vacaciones, así como el registro de asistencias para verificar si los colaboradores se presentaron o no a trabajar. Además, indicó que también debe llevar un control de licencias, incapacidades y permisos. Con respecto a las herramientas mencionó que se apoya principalmente en Google Drive para la organización y almacenamiento de la información.

Respecto a la segunda pregunta, orientada a identificar los procesos más importantes o críticos en la gestión del personal dentro de la empresa, la entrevista manifestó que el pago de la planilla es la tarea principal o que considera más importante. Señaló que este proceso debe realizarse de manera correcta y puntual, ya que constituye la base del cumplimiento de las obligaciones con los colaboradores. Recalcó que la planilla se paga semanalmente.

En cuanto a la tercera pregunta, relacionada con un ejemplo práctico de un proceso que le genera mayores dificultades, la entrevista explicó que uno de los momentos más complicados se presenta a final de año, cuando se debe elaborar el reporte del aguinaldo para cada trabajador. Indicó que este proceso requiere recopilar los salarios de cada semana y unificarlos en un solo archivo de Excel, lo cual demanda tiempo y aumenta la posibilidad de cometer errores.

Con respecto a la cuarta pregunta, enfocada en los datos esenciales que deberían registrarse de cada empleado en el sistema, la entrevista indicó que la información mínima necesaria incluye el nombre completo, el número de cédula, el número de teléfono, la fecha de nacimiento, el estado civil y la dirección de correo electrónico.

En cuanto a la quinta pregunta, sobre las funciones que debería incorporar un sistema de gestión de recursos humanos, la entrevista señaló que lo más importante es que incluya todo lo relacionado con la planilla y el expediente del trabajador. Este expediente debería contemplar el nombre, los datos personales completos y un número de contacto para casos de emergencia, así como la fecha de ingreso a la empresa. Además, consideró relevante que el sistema permita registrar la información sobre los contratos, por ejemplo, cuando un colaborador inicia con un contrato de tres meses y este posteriormente se extiende a seis.

Con respecto a la sexta pregunta, dirigida a conocer el tipo de consultas que resultarían más útiles en el desarrollo diario de sus funciones, la entrevista mencionó que sería fundamental poder tener acceso con facilidad a toda la información relacionada con la planilla. Entre estos datos señaló el salario por hora normal, horas extra, horas dobles, así como los montos correspondientes a aguinaldo, vacaciones e incapacidades. También destacó la importancia de contar con un historial que permita visualizar cuántas incapacidades, permisos o ausencias (ya sean justificadas o injustificadas) acumuladas por cada empleado. Finalmente, subrayó que lo más valioso sería disponer de un sistema que brinde esta información de manera rápida y eficiente.

Continuando con la séptima pregunta, sobre los reportes que serían de mayor utilidad dentro del sistema, en la entrevista se destacó, en primer lugar, la necesidad de contar con un reporte actualizado del aguinaldo, que permita estimar el monto acumulado hasta la fecha y calcular cuánto faltaría por cubrir según los meses pendientes del año. Además, se señaló la

importancia de disponer de un reporte de vacaciones por empleado, en el que se detalla cuántos días ha disfrutado, cuántos le restan y cuál sería el equivalente económico de esos días según su salario. Asimismo, se consideró fundamental contar con un reporte del historial de ausencias de cada trabajador, clasificadas en incapacidades, permisos o faltas injustificadas. Dicho reporte debería indicar, de todos los días laborales del año, cuántos no fueron laborados y desglosarlos por motivo (incapacidad, permisos, con o sin comprobante). Según se explicó, esta información sería de gran valor para la toma de decisiones relacionadas con incrementos salariales o reconocimientos, ya que permitiría identificar a los colaboradores con mayor compromiso y asistencia.

En la octava pregunta se abordó la gestión de aspectos clave como vacaciones, incapacidades y horas extra. La entrevista confirmó que es esencial que el sistema permita controlar todos los elementos, Además de simplemente registrarlos, el sistema debe ofrecer la capacidad de generar reportes y consultas detalladas.

En la pregunta nueve, sobre la importancia de que un sistema sea fácil de usar, la respuesta no se centró tanto en la funcionalidad sino en la experiencia visual y estética. La entrevistada enfatizó que la limpieza visual y la armonía de los colores son aspectos fundamentales para ella. Mencionó que prefiere que el diseño sea sobrio, con una paleta de colores limitada (por ejemplo, blanco, gris y verde) y sin elementos distractores, como logos repetidos o una saturación información. Para ella la facilidad de usos está directamente ligada a la tranquilidad visual. La diferencia en el color de los botones (por ejemplo, botón azul claro para “cancelar” y uno azul para “guardar”), o el uso de negritas para destacar la información son las herramientas visuales que prefiere. Explicó que los colores “de paz” como el azul y el verde son sus favoritos, y evitar los rojos, que le parecen demasiado llamativos. En resumen, la usabilidad para ella se traduce en un diseño intuitivo y limpio en el que la información se presente de forma sencilla y agradable a la vista.

En la décima pregunta se abordó el caso de la migración de datos que se requiera a la hora de implementar el sistema y la respuesta dejó claro que, aunque la mayoría de su información se encuentra en Google Drive, la migración de algunos datos clave es fundamental. Específicamente, se mencionó la necesidad de trasladar los datos del aguinaldo. La razón principal para esta solicitud es verificar la exactitud del nuevo sistema. La entrevistada mostró una ligera desconfianza de si el sistema calculará los montos del aguinaldo correctamente, por lo que migrar esta información le daría la tranquilidad de poder comparar los resultados y asegurarse de que los cálculos son precisos y confiables.

La onceava pregunta aborda la seguridad del sistema. La respuesta se centró en un aspecto clave, el control del acceso. La persona entrevistada considera fundamental que el sistema permita crear diferentes niveles de usuario, para que la persona tenga permisos específicos. Recalcó la importancia de poder definir el alcance de cada usuario, lo que permite que algunos puedan ver y agregar información, pero restringiendo acciones más críticas, como borrar o modificar datos ya existentes. Además, señaló la necesidad básica de un inicio de sesión seguro para evitar accesos no autorizados.

Con respecto a la pregunta número doce, que se enfoca en el diseño visual del sistema, la respuesta se enfocó en el tipo de letra que le gustaría. A pesar de que la pregunta era más general sobre colores y navegación, pero como fue abordada en una pregunta anterior, la entrevistada recalcó la importancia de usar una tipografía limpia y agradable. Se señaló que su tipo favorito de letra es Cambria, la cual se utiliza en programas como Excel y Word. La razón por la que respalda este tipo de letra es que la considera “limpia y bonita”.

En la pregunta número 13 se abordaron funciones adicionales que le gustaría ver en el sistema. En la entrevistada se hizo énfasis en el módulo de liquidación de empleados. Este módulo, según su descripción, debería registrar la fecha de salida de un empleado y el monto final que se pagó. La capacidad de tener un registro claro y accesible a estas liquidaciones es crucial para ella.

Con respecto a la décimo cuarta pregunta sobre las mejoras que el sistema podría aportar al trabajo diario en Recursos Humanos, la respuesta se centró en la eficiencia y la reducción de errores. La entrevistada destacó que su principal expectativa es obtener los reportes de forma más rápida, sin tener que crearlos manualmente, lo que actualmente le consume mucho tiempo. Además, busca que el sistema minimice los errores que suelen ocurrir con el manejo manual de datos, lo que le permitiría trabajar con mayor confianza. En esencia, el objetivo es reducir las tareas repetitivas y tediosas, para permitirle centrarse en otras labores.

Para finalizar la entrevista se hizo una pregunta abierta a fin de explorar cualquier tema que no se hubiera abordado. La respuesta de la entrevistada fue concisa, e indicó que de momento no había ningún otro punto importante que añadir.

El análisis de la entrevista con la encargada de Recursos Humanos de Quesada Vargas Construcciones, S.A. ha sido fundamental para entender sus necesidades y expectativas respecto a un nuevo sistema. A lo largo de las preguntas se identificaron varios puntos clave que serán tomados en cuentas como una guía para plantear la solución.

Encuesta

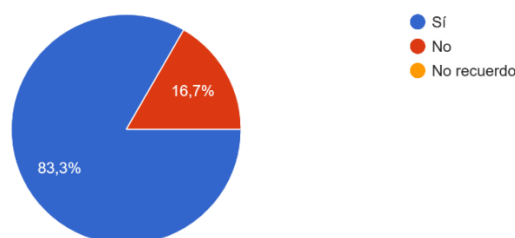
Con el objetivo de recopilar información valiosa y cuantitativa directamente de los colaboradores, se diseñó y distribuyó un cuestionario digital por medio de la plataforma Google Forms. Al inicio del proyecto la empresa contaba con 15 colaboradores; sin embargo, debido a ajustes internos, cinco de ellos fueron despedidos antes de la aplicación del instrumento, por lo que la población se redujo a 10 empleados. De estos, se aplicó la encuesta a 9 y se obtuvo respuesta efectiva de 6 colaboradores. Este instrumento permitió obtener datos de manera eficiente y llegar a un mayor número de empleados para entender sus necesidades y percepciones. El cuestionario se encuentra disponible en el Apéndice B. Los resultados de este análisis se presentan a continuación, y se ofrece un análisis de los hallazgos más relevantes.

Al analizar los resultados de la encuesta aplicada a los empleados se observa que, en la primera pregunta, relacionada con errores de cálculo de salarios, pagos extras o deducciones, la mayoría de los colaboradores respondieron que sí han tenido inconvenientes de este tipo.

Figura 14

Errores de cálculo en la planilla

Pregunta 1. ¿Ha tenido alguna vez errores en el cálculo de su salario, pagos extra o deducciones?
6 respuestas



Fuente: Elaboración propia, 2025.

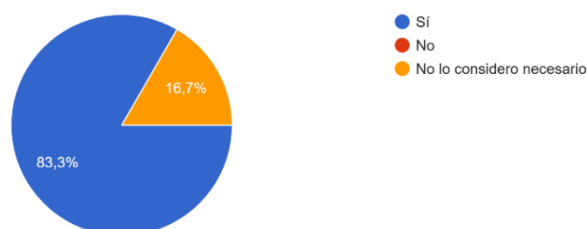
Como se muestra en la Figura 15, 83% de los colaboradores han experimentado algún error en el cálculo de la planilla, pagos extra o deducciones. Esto pone en evidencia que existe una debilidad importante en el área de planillas, ya que los errores en los pagos generan inconformidad y desconfianza en el personal. Solo una persona indicó no haber tenido problemas, lo que confirma que, aunque el error no se da en todos los casos, es un aspecto que debe mejorarse.

En cuanto a la segunda pregunta, que se refería a la rapidez con que se gestionan trámites de vacaciones, incapacidades u horas extras, la mayoría de los encuestados respondieron de manera positiva.

Figura 15

Gestión de vacaciones, incapacidades u horas extras

Pregunta 2. ¿Siente que los trámites de vacaciones, incapacidades u horas extras se gestionan con rapidez?
6 respuestas



Fuente: Elaboración propia, 2025.

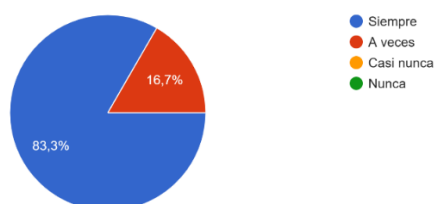
La percepción general es que estos procesos se manejan con agilidad y cumplen con las expectativas de los empleados, como se puede apreciar en la Figura 16, en que 83% respondieron que sí hay fluidez en los trámites. Sin embargo, hubo una persona que señaló que no considera necesario este tipo de trámite, lo que demuestra que no todos perciben su relevancia de la misma forma.

Respecto a la comunicación sobre detalles importantes como fechas de pago, rebajos o feriados, la mayoría de los trabajadores, en 83% de los colaboradores, según la Figura 17, aseguraron que siempre reciben la información a tiempo.

Figura 16

Comunicación de nómina

Pregunta 3. ¿Le informan a tiempo sobre detalles importantes como fechas de pago, rebajos o días feriados?
6 respuestas



Fuente: Elaboración propia, 2025.

Aun así, algunos mencionaron que a veces no es tan clara o constantes, lo que indica que, aunque la comunicación es en su mayoría efectiva, todavía existe un pequeño margen de mejora para garantizar que toda la información llegue de manera efectiva a todos los colaboradores.

Sobre las dificultades para conocer los días de vacaciones acumulados o utilizados las respuestas estuvieron divididas. De ellos 33% ha presentado alguna dificultad para conocer cuántos días de vacaciones tiene disponibles y cuántos ha disfrutado, mientras que 66% no han presentado dificultades.

Figura 17

Gestión de vacaciones



Fuente: Elaboración propia, 2025.

Eso refleja que no todos cuentan con la misma facilidad para consultar sus vacaciones, lo cual puede generar incertidumbre y dificultades en la planificación personal.

En lo que respecta a la percepción de justicia y claridad en la gestión de asistencia, incapacidades y permisos, la mayoría de los empleados, según la Figura 18, consideran que estos procesos se manejan de manera adecuada, con 83% de los colaboradores que respondieron.

Figura 18

Gestión de asistencia, incapacidades y permisos



Fuente: Elaboración propia, 2025.

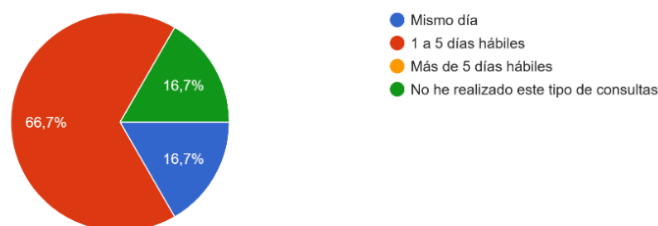
A pesar de eso persona indicó que a veces no lo percibe tan claro, por lo cual señala que, aunque en general la gestión es vista como justa, se pueden reforzar algunos aspectos para que la experiencia sea positiva para todos.

Sobre el tiempo de respuesta para consultas relacionadas con vacaciones u otros trámites laborales, la mayoría de los encuestados indicaron que suele tardar entre uno y cinco días hábiles, según la Figura 20, y esto representó 66% de los colaboradores que respondieron.

Figura 19

Tiempo de respuesta en gestión de trámites

Pregunta 6. ¿Cuál es el promedio de respuesta para consultas relacionadas con vacaciones, aguinaldo u otros trámites laborales?
6 respuestas



Fuente: Elaboración propia, 2025.

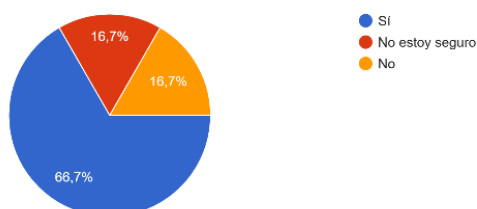
También se registró un caso en el que la respuesta es recibida el mismo día, mientras que otro empleado comentó que no ha realizado este tipo de consultas. Estos resultados muestran que, aunque el tiempo de respuesta no es excesivo, sí existe la oportunidad de mejorar la inmediatez y la eficiencia en la atención a los colaboradores.

En relación con la pregunta acerca de si un sistema digital podría mejorar la gestión de Recursos Humanos, la mayoría de los empleados coincidieron en que sí representaría un beneficio para la empresa, y según la Figura 21, 66% de los colaboradores que respondieron.

Figura 20

Sistema digital como mejora en la gestión de recursos humanos

Pregunta 7. ¿Cree que un sistema digital podría mejorar la gestión de Recursos Humanos en la empresa?
6 respuestas



Fuente: Elaboración propia, 2025.

No obstante, hubo respuestas que reflejan cierta duda o resistencia al cambio, ya que un colaborador indicó que no lo cree necesario y otro manifestó que no está seguro. Esto

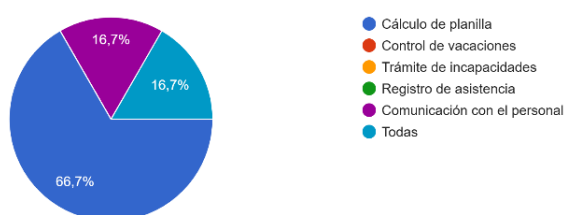
demuestra que, aunque la digitalización es vista de manera positiva en general, aún queda trabajo por hacer en términos de aceptación y adaptación al cambio.

Como se puede apreciar en la Figura 22, cuando se consultó sobre los aspectos que podrían mejorarse con un sistema digital la mayoría señalaron el cálculo de planilla como el área principal por reforzar, dado que es en donde más errores se han identificado.

Figura 21

Aspectos proyectados del sistema

Pregunta 8. ¿Cuáles de estos aspectos considera que podrían mejorarse con un sistema digital?
(puede marcar más de una opción)
6 respuestas



Fuente: Elaboración propia, 2025.

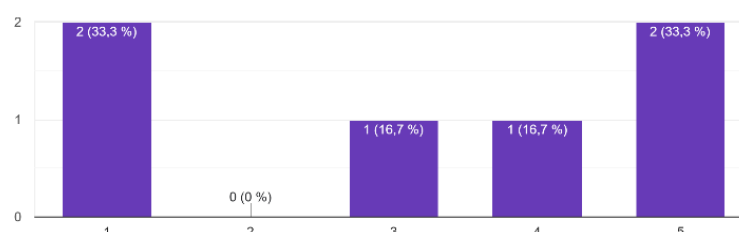
También se mencionó la comunicación con el personal y, en un caso, se sugirió que todas las áreas se verían beneficiadas con un sistema de este tipo. De esta manera los resultados evidencian que los colaboradores perciben en la digitalización una oportunidad clara de mejorar la exactitud, la rapidez y la eficiencia en la gestión.

En la pregunta relacionada con la calificación general de los procesos de Recursos Humanos las respuestas fueron muy variadas, con calificaciones que oscilaron entre 1 y 5. Según la Figura 23 esto refleja que no existe una percepción uniforme: algunos consideran que los procesos están bien gestionados mientras que otros tienen una opinión más crítica.

Figura 22

Calificación de los procesos actuales

Pregunta 9. En una escala del 1 al 5, ¿cómo calificaría actualmente la forma en que se manejan los procesos de Recursos Humanos?
6 respuestas



Fuente: Elaboración propia, 2025.

El promedio general se ubica en alrededor de 3, lo que indica un nivel de satisfacción intermedio que da cuenta de que la gestión cumple en lo básico, pero aún no genera plena confianza entre todos los colaboradores.

Finalmente, en el espacio abierto para comentarios o sugerencias algunos empleados aprovecharon para recomendar la implementación de un sistema digital que asegure exactitud y rapidez, aunque también hicieron énfasis en la importancia de mantener la integración personal de la empresa y el trabajador, ya que esto fortalece la relación laboral. Otros colaboradores no hicieron observaciones o manifestaron que todo funciona bien, lo cual puede interpretarse como conformidad o falta de interés en profundizar. Un trabajador sugirió un aumento salarial, lo que no está directamente relacionado con los procesos de gestión de Recursos Humanos, pero sí refleja una necesidad y expectativa común en los colaboradores.

En general, los resultados de la encuesta muestran que existen avances importantes en la gestión de Recursos Humanos, especialmente en lo relacionado con la rapidez en trámites y la comunicación, pero también evidencian problemas claros en el cálculo de planilla y en el acceso a información sobre vacaciones. Asimismo, queda claro que la mayoría de los empleados ve en un sistema digital una solución que podría mejorar los procesos y aumentar la satisfacción laboral.

Requerimientos del sistema

A partir del análisis mediante la observación, entrevista y encuesta aplicada en la empresa Quesada Vargas Construcciones, S.A., se identificaron los requerimientos que debe atender el sistema web de gestión de Recursos Humanos para cubrir las necesidades del departamento. Estos requerimientos se clasifican en funcionales y no funcionales, y constituyen la base para el diseño y desarrollo del prototipo funcional.

Requerimientos funcionales

En los requerimientos funcionales se indican las acciones o servicios específicos que el sistema debe atender para satisfacer las necesidades del área de Recursos Humanos de la empresa Quesada Vargas Construcciones, S.A. Estos requerimientos fueron definidos a partir del análisis de los procesos actuales y de la información recolectada mediante los instrumentos aplicados (entrevista, encuestas y observación).

Tabla 14*Requerimientos funcionales del sistema*

Módulo	Código	Descripción del requerimiento funcional
Gestión de Empleados	REQ-EMP-01	Registrar, consultar, modificar y eliminar de los registros de empleados.
	REQ-EMP-02	Asociar los datos personales con su dirección, provincia, cantón y distrito.
	REQ-EMP-03	Asignar el puesto, tipo de jornada y salario base del empleado.
Gestión de Asistencia	REQ-ASI-01	Registrar la hora de entrada y salida de cada empleado en una fecha.
	REQ-ASI-02	Validar que no existan asistencias duplicadas
	REQ-ASI-03	Mostrar un historial de asistencia de cada empleado.
Gestionar Permisos	REQ-PER-01	Registrar permisos indicando el motivo, fechas y estado de cada empleado.
	REQ-PER-02	Mostrar un historial de permisos entre un rango de fechas por cada empleado.
Gestión de Incapacidades	REQ-INC-01	Registrar incapacidades otorgadas por el INS y la CCSS, registrando las fechas en que está inhabilitado el empleado.
	REQ-INC-02	Mostrar un historial de incapacidades de cada empleado filtrado por INS, CSS o todas.
Gestión de Vacaciones	REQ-VAC-01	Registrar las solicitudes de vacaciones y su estado (pendiente, aprobado o rechazado).
	REQ-VAC-02	Registrar la cantidad de días disponibles que cada empleado tiene a disposición.
Gestión de Horas Extra	REQ-HEX-01	Registrar la fecha, hora de inicio, hora de finalización y observaciones para las jornadas excepcionales de cada empleado.
	REQ-HEX-02	Mostrar un registro de horas extras entre un rango de fechas para cada colaborador.
Calcular Nómina	REQ-NOM-01	Generar la planilla semanal con salario bruto, deducciones y salario neto a pagar.
	REQ-NOM-02	Registrar rebajos legales como el de la CCSS y rebajos de cualquier otra índole.
Cálculo Aguinaldo	REQ-AGUI-01	Calcular automáticamente el aguinaldo con base en el promedio de salarios del periodo.
	REQ-AGUI-02	Generar el comprobante de pago correspondiente.
Liquidaciones	REQ-LIQ-01	Calcular la liquidación según el tipo de finalización del contrato.
	REQ-LIQ-02	Cumplir con la legislación costarricense, completando el pago de cesantía, aguinaldo, vacaciones y preaviso.
Seguridad	REQ-SEG-01	Controlar el acceso mediante usuario y contraseña.
	REQ-SEG-02	Asignar roles y permisos diferentes según tipo de usuario.
	REQ-SEG-03	Registrar las acciones de los usuarios en el sistema para tener trazabilidad.

Reportes	REQ-REP-01	Generar reportes de empleado, asistencia, nómina, vacaciones, permisos y aguinaldo.
----------	------------	-------------------------------------------------------------------------------------

Fuente: Elaboración propia, 2025.

Requerimientos no funcionales

Los requerimientos no funcionales muestran las características de calidad del sistema de gestión de Recursos Humanos para asegurar su rendimiento, seguridad, confiabilidad y facilidad de uso. Estos aspectos definen las condiciones bajo las cuales debe operar el sistema para garantizar su correcto desempeño.

Tabla 15

Requerimientos no funcionales del sistema

Categoría	Código	Descripción del requerimiento no funcional
Rendimiento	RNF-REN-01	El sistema debe permitir realizar operaciones comunes (registros, consulta, actualización y eliminación de datos) y cálculos de forma fluida, sin generar retrasos en condiciones normales de uso.
Seguridad	RNF-SEG-01	El acceso al sistema debe estar protegido mediante autenticación de usuario y contraseña, con encriptación de credenciales y control de roles.
Disponibilidad	RNF-DIS-01	El sistema debe estar disponible durante toda la jornada laboral siempre que el equipo servidor esté en funcionamiento.
Usabilidad	RNF-USA-01	La interfaz debe ser clara, intuitiva y con colores neutros que faciliten la lectura y reduzcan la fatiga visual.
Mantenibilidad	RNF-MANT-01	El sistema debe permitir la actualización de los módulos sin afectar la estabilidad del resto del sistema.
Confiabilidad	RNF-CON-01	El sistema debe garantizar la integridad de los datos en todas las operaciones y cálculos que realice el sistema y restricciones en base de datos.

Fuente: Elaboración propia, 2025.

Estos lineamientos permiten orientar el diseño, desarrollo y validación del prototipo funcional, asegurando que la solución propuesta responda a las necesidades reales de la empresa y opere de manera eficiente, segura y confiable dentro de su entorno actual.

CAPÍTULO V: PROPUESTA

En este capítulo se presenta la propuesta de solución del problema identificado en la investigación. En él se detalla el prototipo funcional del sistema, con el cual se busca responder a las necesidades detectadas en la empresa y optimizar el manejo de la información relacionada con los recursos humanos.

Además, se incluyen elementos que respaldan el diseño y el desarrollo del sistema para la solución, como los casos de uso, el diagrama entidad-relación, el diccionario de datos, el diseño de vistas principales del sistema, la infraestructura necesaria, el hardware y el software, entre otros aspectos más que son empleados para obtener el prototipo funcional.

El objetivo es mostrar, de forma ordenada y clara, cómo se materializa la propuesta, desde su concepción en el análisis hasta su representación visual y técnica.

Análisis del sistema

En este apartado se elabora un estudio integral de los componentes técnicos y operativos necesarios para obtener la propuesta del prototipo funcional del departamento de Recursos Humanos. Se describen en detalle los módulos que integran el sistema, como su estructura y funcionalidades. Asimismo, se realiza el análisis del hardware requerido, tanto para el entorno de desarrollo del prototipo como para su posterior implementación en un entorno productivo.

De igual forma, se analizan recursos de telecomunicaciones, como herramientas tecnológicas utilizadas para el desarrollo del prototipo. También se indican los conocimientos mínimos que deberá poseer el personal responsable que operará el sistema y se elaboran los casos de uso conforme a los lineamientos establecidos.

Análisis detallado del software

En este apartado se presentan los módulos que conforman la propuesta de solución desarrollada. La descripción de cada módulo se basa en la información obtenida mediante los instrumentos de recolección de datos aplicados a los colaboradores de la empresa Quesada Vargas Construcciones, S.A. Dichos resultados permitieron identificar las principales necesidades del área de Recursos Humanos y orientar el diseño del sistema para responder de manera efectiva a esos requerimientos.

Módulo de gestión de empleados

Este módulo se encarga de administrar la información personal de todos los colaboradores, incluidos datos como número de identificación, números de teléfono, correo electrónico, dirección y contactos de emergencia. Además, permite registrar los atestados académicos y profesionales de cada empleado, como títulos y certificaciones obtenidos. Asimismo, gestiona aspectos relacionados con la jornada laboral, el puesto asignado, el horario de trabajo y el historial de promociones del colaborador durante su permanencia en la empresa.

Módulo Gestión de asistencia

Este módulo se encarga de registrar las horas de entrada y salida de cada empleado, con el objetivo de mantener un control preciso del cumplimiento de la jornada laboral. Además, permite consultar el historial de asistencia de cada colaborador durante la semana de trabajo, lo que facilita el seguimiento y la supervisión del registro de asistencia.

Módulo gestión de horas extra

Este módulo se encarga de registrar las horas extra diarias laboradas por cada empleado. Permite llevar un seguimiento detallado de las jornadas que exceden el horario regular, para facilitar la consulta del historial durante la semana laboral correspondiente. De esta manera la empresa puede administrar de forma eficiente el pago de horas extra.

Módulo de nómina

Este módulo se encarga de gestionar el cálculo de nómina de cada uno de los colaboradores, e integra todas las deducciones establecidas por la legislación laboral vigente, así como aquellas acordadas por mutuo acuerdo entre las partes, tales como ahorros voluntarios, préstamos u otros descuentos aplicables. Su funcionamiento permite automatizar el proceso de pago, lo que garantiza exactitud y transparencia.

Módulo calcular aguinaldos

Este módulo tiene como función principal calcular de manera automática el aguinaldo correspondiente a cada colaborador, conforme a la legislación laboral vigente en Costa Rica. Para ello, toma en cuenta los salarios devengados durante el periodo establecido por la ley, garantizando exactitud en los resultados y cumplimientos con las obligaciones patronales. Además, el módulo permite consultar el historial de aguinaldos para cada empleado, facilitando la trazabilidad de los cálculos realizados.

Módulo gestión de permisos

Este módulo se encarga de administrar las solicitudes de permisos realizadas por los colaboradores, que abarca todo el proceso, desde la presentación de la solicitud hasta su aprobación o rechazo. Además, permite registrar las condiciones específicas de cada permiso, tales como el motivo y las fechas establecidas. Asimismo, el módulo ofrece la posibilidad de consultar el historial completo de permisos de cada empleado, con filtros por estado, lo que facilita un control más eficiente.

Módulo gestionar incapacidades

Este módulo se encarga de gestionar de manera adecuada las incapacidades de los colaboradores, lo que permite distinguir aquellas emitidas por la Caja Costarricense de Seguro Social (CCSS) y las otorgadas por el Instituto Nacional de Seguros (INS). Además, registra la cantidad de días de incapacidad asignados a cada empleado, y esto facilita el control y el seguimiento de los periodos. El módulo también da acceso a un historial detallado de incapacidades por empleado, con la posibilidad de filtrar la información según el tipo de incapacidad (CCSS o INS).

Módulo de gestión de vacaciones

Este módulo se encarga de gestionar las vacaciones de los empleados, para registrar y controlar la cantidad de días disponibles, así como los días aprobados para su disfrute. Además, automatiza el cálculo del pago correspondiente por los días de vacaciones utilizados, para

garantizar el cumplimiento de la legislación laboral vigente de Costa Rica. Asimismo, el módulo facilita la consulta del historial de vacaciones de cada empleado, permitiendo una gestión adecuada y transparente del tiempo de descanso disfrutado por el personal.

Módulo de liquidaciones

Este módulo se encarga de generar el cálculo de las liquidaciones de cada colaborador al finalizar su relación laboral, lo que asegura la precisión y el cumplimiento de la legislación laboral vigente en Costa Rica.

Módulo de consultas

Este módulo le permite al usuario el acceso de forma rápida y sencilla a la información que necesita. Su función principal es mostrar datos específicos del sistema, como historial de movimientos y los datos de los empleados, entre otros. Cada módulo del sistema incluye su propia sección de consultas si este lo requiere, adaptado a la información que maneja para facilitar la búsqueda.

Módulo de reportes

Este módulo permite generar distintos tipos de reportes relacionados con la información de los empleados activos en el sistema. Entre los reportes disponibles se incluyen datos personales, puesto y jornada laboral, registros de asistencia, horas extras, permisos e incapacidades; beneficios como vacaciones y aguinaldos; y cálculos de nómina como salario bruto, neto, deducciones. El objetivo es ofrecer al usuario una forma ordenada y completa de visualizar la información más importante.

Módulo de mantenimientos

Este módulo le permite al usuario agregar, modificar o eliminar la información que forma parte del sistema, como los datos de empleados, puestos, jornadas, entre otros. Su principal función es mantener actualizada la base de datos para que el sistema funcione correctamente y refleje los cambios que se den en la organización o cuanto a la legislación costarricense.

Módulo de seguridad

Este módulo se encarga de proteger el acceso al sistema y la información que contiene. Permite definir usuarios, asignar roles y establecer permisos según las funciones que cada persona debe realizar. Su objetivo principal es garantizar que solo personas autorizadas puedan ingresar al sistema y tener acceso a los datos, manteniendo la confidencialidad, el control de la información y el cumplimiento de la legislación vigente.

Análisis detallado del hardware

En esta sección se detallan los equipos necesarios para desarrollar y poner en funcionamiento el sistema propuesto. Se incluyen especificaciones técnicas, costos estimados y requerimientos tanto para el prototipo como para la versión final en producción, lo cual asegura que el sistema opere de forma estable y eficiente.

Hardware para el desarrollo del prototipo

En esta sección se presenta el hardware del equipo utilizado para construir el prototipo funcional del sistema. Se detallan características técnicas disponibles, considerando su capacidad para programar, probar y validar las funciones principales. Es importante para conocer la capacidad del equipo ante los requerimientos de los sistemas involucrados para desarrollar el prototipo.

El equipo que se va a utilizar para realizar el prototipo funcional es propiedad del estudiante. Es una laptop marca Acer con las siguientes especificaciones técnicas; cuenta con un procesador Intel Core i5-12500H, con una capacidad de 16.0 GB de memoria RAM, un almacenamiento SSD con una capacidad de 476.9 GB, además de una tarjeta gráfica dedicada NVIDIA GeForce RTX 3050 Ti, una pantalla interna de 15 pulgadas con 1920 x 1080 con 144 Hz y una tarjeta de conectividad Wifi modelo Killer Wi-Fi 6 AX1650i 160MHz Wireless Network Adapter (201NGW), y una tarjeta de conectividad Ethernet modelo Killer E2600 Gigabit Ethernet Controller. También se cuenta con una pantalla externa modelo Samsung C27F390 con 1080p, 1920 x 1080 (nativo) con 60 Hz, con un tamaño de 27 pulgadas.

Hardware para la implementación del prototipo

Para la implementación del prototipo funcional se usarán los equipos que ya posee la empresa Quesada Vargas Construcciones, S.A., por lo que no se deberá incurrir en gastos en compra de equipos. La empresa cuenta actualmente con dos equipos, una computadora de escritorio y una laptop. A continuación se detallan las características técnicas de cada equipo.

La computadora de escritorio cuenta con las siguientes especificaciones técnicas: un procesador Intel Core i7-3770 a 3.40 GHz, acompañado de una memoria RAM de 12 GB. Dispone de un sistema operativo de 64 bits sobre una arquitectura x64, y una tarjeta gráfica dedicada NVIDIA GeForce 210 con 973 MB de memoria. El almacenamiento total disponible es de 1.13 TB.

La laptop cuenta con las siguientes especificaciones técnicas: un procesador Intel Core i7-1165G7 de 11ª generación a 2.80 GHz, acompañado de 12.0 GB de memoria RAM. Cuenta con una tarjeta gráfica integrada Intel Iris Xe Graphics con 128 MB de memoria dedicada, y un almacenamiento interno de 477 GB.

Dadas las especificaciones técnicas de ambos equipos, la laptop cuenta con especificaciones técnicas óptimas como servidor para la implementación del sistema, y cuenta de forma general con un mejor rendimiento.

Análisis detallado de telecomunicaciones

En esta sección se describen los elementos de conectividad necesarios para el funcionamiento del sistema. Se analizan el tipo de red requerido, el uso de internet, los puertos involucrados, y los equipos especializados que podrían ser necesarios, así como sus posibles costos. El objetivo es asegurar que el entorno de telecomunicaciones sea adecuado para garantizar el acceso, la estabilidad y el rendimiento del sistema.

Infraestructura de red local y conectividad

Debido a que la empresa no cuenta con infraestructura tecnológica avanzada, el sistema se implementa de manera local y autónoma en un único equipo portátil que actúa simultáneamente como servidor de la aplicación y servidor de bases de datos.

El sistema opera bajo un entorno cliente-servidor interno, en que el acceso se realiza mediante un navegador web conectado al mismo equipo (localhost) y por el puerto 8080. La comunicación entre el usuario y la aplicación se realiza por protocolo HTTP (Hypertext Transfer Protocol), sin requerirse conexión a Internet, ya que todo el entorno se ejecuta de forma interna en la empresa.

Esta configuración permite que el sistema funcione de manera estable con mínimos recursos de telecomunicaciones, para aprovechar el hardware existente y sin necesidad de equipos de red adicionales.

Dispositivo para el sistema web embebido

La aplicación se ejecuta en una laptop designada como servidor, en que se aloja tanto la aplicación web desarrollada en Spring Boot con servidor Tomcat 10.1 embebido, como la base de datos MySQL. Este equipo central procesa las solicitudes del usuario y responde directamente a través del navegador web.

Firewall

El firewall del sistema operativo cumple con una función esencial en la protección del entorno. Debe configurarse para permitir únicamente el tráfico entrante al puerto 8080, en el que se ejecuta la aplicación web, y el puerto 3306, utilizado internamente por el servicio de base de datos MySQL. De esta forma se restringe el acceso desde equipos no autorizados, para proteger la integridad de la información del sistema.

Respaldo de la información

Para prevenir la pérdida de datos ante fallas del equipo o errores humanos se recomienda establecer un plan de respaldo periódico que incluya respaldo de la base de datos MySQL y copia de archivo ejecutable de la aplicación, y los archivos de configuración. Estos respaldos pueden almacenarse en un dispositivo externo (como una memoria USB o disco duro portátil) o en servicios de almacenamiento en la nube (OneDrive o Google Drive).

Disponibilidad y escalabilidad

Dado que todo el sistema depende de un único equipo, su disponibilidad estará sujeta al correcto funcionamiento de la laptop. En caso de apagarse o presentar fallas el sistema no estará disponible. Por esta razón se recomienda que en el futuro la empresa valore la posibilidad de adquirir un servidor dedicado o un equipo con mayor capacidad, que permita mantener el sistema en funcionamiento continuo, incluso de manera remota, en caso de ampliarse su uso a más empleados.

Descripción detallada de herramientas técnicas para el desarrollo

En esta sección se detallan las herramientas empleadas para el desarrollo del sistema, incluido el entorno de programación, el motor de la base de datos y otros recursos técnicos relevantes. También se consideran aspectos como licenciamiento, costos asociados y capacidades necesarias para asegurar un funcionamiento eficiente y sostenible del sistema.

El lenguaje de programación principal es Java, aprovechando su estabilidad y compatibilidad con múltiples plataformas. Se utiliza el *framework Spring Boot*, que facilita la creación de aplicaciones web al integrar componentes como controladores, servicios y accesos a bases de datos de forma organizada. Además, se utiliza Spring Security para implementar medidas básicas de autenticación y control de acceso dentro del sistema.

Como motor de base de datos se eligió MySQL Community Server versión 8.2.0, por ser gratuito, confiable y ampliamente utilizado con buen soporte y documentación accesible. Para gestionar la base de datos de forma visual y práctica se utiliza la herramienta MySQL Workbench, que permite realizar consultas, generar modelos E-R, revisar estructuras y administrar la base de datos. Para escribir y organizar el código se utilizó el editor Visual Studio Code versión 1.104.3, que ofrece extensiones útiles para trabajar con Java, Spring Boot, HTML y demás tecnologías implicadas en el desarrollo del prototipo.

En la parte visual del sistema se aplican tecnologías como HTML5 y CSS para estructurar y dar estilo a las páginas, junto con Thymeleaf como motor de plantillas, que permite integrar datos dinámicos desde el backend. También se utiliza Bootstrap para mejorar el diseño y la presentación de los formularios, botones y tablas, para lograr una apariencia más profesional, sin necesidad de crear estilos desde cero.

Para el control de versiones y respaldo del proyecto se utiliza GitHub, lo que permitirá mantener un historial de cambios, facilitar el trabajo ordenado y asegurar la disponibilidad del

código en caso de fallas locales. Todo el desarrollo se realiza sobre el sistema operativo Windows 11 Pro, instalado en el equipo del estudiante, el cual ofrece compatibilidad con las herramientas utilizadas y un entorno estable para la programación y ejecución del sistema.

Todas las herramientas utilizadas son de código abierto o gratuitas, lo que permite desarrollar el sistema sin generar costos adicionales. Además, su documentación es accesible y existe una gran comunidad en línea que facilita el aprendizaje y la resolución de dudas durante el proceso de desarrollo, además de recursos como libros, cursos y demás, que el estudiante adquirió bajo su propio costo para extender su conocimiento. Estas tecnologías son compatibles entre sí y se ajustan a los objetivos del proyecto y a los recursos disponibles del estudiante.

Conocimiento del recurso humano para operar el sistema

El sistema está diseñado para ser utilizado por personal administrativo o encargado de recursos humanos dentro de la empresa, sin necesidad de conocimientos técnicos avanzados. Para operar el sistema correctamente el recurso humano debe tener habilidades básicas en el uso de computadoras y navegación en sistemas web. Es importante que comprendan cómo ingresar, consultar y modificar información esencial del negocio, siguiendo los flujos establecidos por el sistema.

Además, deben estar familiarizados con conceptos generales de la administración y gestión del departamento de Recursos Humanos como asistencia, nómina, vacaciones, liquidaciones y demás aspectos que el sistema integra. También se requiere que el usuario pueda interpretar los mensajes de validación de pantalla, utilizar los filtros para consultas y generar reportes.

No se requiere experiencia en programación no conocimientos técnicos sobre bases de datos, ya que el sistema oculta la complejidad técnica y presenta una interfaz amigable. Con una capacitación básica sobre el funcionamiento de cada módulo, el personal podrá utilizar el sistema de forma eficiente y segura, y cumplir con los objetivos administrativos para los que fue diseñado.

Casos de uso

En esta sección se presentan los casos de uso en que se describen las principales interacciones entre los usuarios y el sistema. En cada caso de uso se detalla una funcionalidad

específica, incluidos los pasos que sigue el usuario, las condiciones necesarias y las respuestas esperadas del sistema.

Tabla 16

Caso de uso para inicio de sesión en el sistema

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número caso de Uso: CU-MS-01	Nombre del caso de uso: Inicio de sesión
Fecha elaboración:	10/09/2025
Descripción Caso de Uso:	Le permite al usuario autenticarse para ingresar al sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuarios del sistema Sistema de gestión de Recursos Humanos
Precondiciones:	El usuario debe estar registrado en el sistema. El usuario debe tener un rol y los permisos asignados. El sistema debe estar disponible.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para que un usuario pueda autenticarse en el sistema al ingresar sus credenciales (usuario y contraseña), para obtener accesos a las funcionalidades de acuerdo con su rol y permisos.	
<ol style="list-style-type: none"> 1. El usuario ingresa a la pantalla de inicio de sesión. 2. El sistema muestra el formulario de login. 3. El usuario introduce en el formulario su nombre de usuario y la contraseña. 4. El sistema valida los campos. 5. El sistema valida las credenciales en la base de datos. 6. El sistema autentica al usuario y genera un token de sesión. 7. El sistema registra el inicio de sesión en la tabla de “audit_change” y “log_evento”. 8. El sistema redirige al usuario a la página principal del Dashboard según su rol. 9. El caso de uso finaliza exitosamente. 	
Flujos alternos	
Flujo alternativo validación de campos vacíos	<ul style="list-style-type: none"> • El usuario deja campos en blanco. • Se muestra un mensaje de error y no se continua con la autenticación. • El flujo continúa en el punto 2 del flujo principal.

Flujo alternativo validación de reglas de contraseña	<ul style="list-style-type: none"> • La contraseña no cumple con los parámetros establecidos para la contraseña. • Se muestra un mensaje de error y no se continúa con la autenticación. • El flujo continúa en el punto 2 del flujo principal.
Flujo alternativo validación de credenciales	<ul style="list-style-type: none"> • Las credenciales no son válidas y rechaza el inicio de sesión. • El sistema muestra un mensaje de error. • El flujo continúa en el punto 2 del flujo principal.
Flujo alternativo Máximo de intentos fallidos	<ul style="list-style-type: none"> • El usuario supera el número máximo de intentos fallidos. • El sistema bloquea temporalmente la cuenta y se registra en el “audit_change”. • El flujo finaliza sin acceso al sistema.
Flujo alternativo error de sistema	<p>El sistema no puede registrar o validar debido a un error técnico.</p> <p>Se muestra un mensaje de error.</p> <p>El flujo finaliza sin acceso al sistema.</p>
Requerimientos especiales	
<p>Cumplimiento de autenticación, bloqueo por intentos fallidos y trazabilidad de accesos.</p> <p>Todas las acciones deben registrarse en “audit_change” y “log_evento”.</p> <p>Las contraseñas deben viajar cifradas.</p>	
Postcondiciones	
<p>El usuario queda autenticado y con una sesión activa.</p> <p>El sistema asegura que se respeten los permisos según el rol asignado.</p>	

Fuente: Elaboración propia, 2025.

Tabla 17

Caso de uso para la recuperación y cambio de contraseña

Prototipo Funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A., ubicada en Grecia	
Número caso de uso: CU-MS-02	Nombre del caso de uso: Recuperación y cambio de contraseña
Fecha elaboración:	10/09/2025

Descripción del caso de uso:	Permite al usuario recuperar y cambiar su contraseña en caso de olvidarla
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario de RRHH Usuario Administrador de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario Administrador de RRHH debe estar registrado en el sistema. El usuario Administrador de RRHH debe haber asignado una contraseña temporal y marcado el campo <code>cambiar_contrasenna = true</code> . El sistema debe estar disponible.
Flujo básico del caso de uso	
Este caso de uso describe el proceso completo que permite al usuario recuperar su acceso cuando ha olvidado su contraseña. El proceso se inicia con la asignación de una contraseña temporal por parte del administrador de RRHH, quien facilita una contraseña nueva provisional. Cuando el usuario inicia sesión con esta contraseña provisional el sistema redirige automáticamente a la vista de cambio de contraseña, caso en el que deberá establecer una nueva contraseña válida para continuar utilizando el sistema.	
<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de inicio de sesión. 2. El sistema muestra el formulario de login. 3. El usuario ingresa en el formulario su nombre de usuario y contraseña temporal. 4. El sistema valida los campos. 5. El sistema valida las credenciales contra la base de datos 6. El sistema detecta el campo <code>cambiar_contrasenna = true</code>. 7. El sistema redirige automáticamente al usuario a la vista de cambio de contraseña. 8. El usuario ingresa la nueva contraseña. 9. El sistema valida la nueva contraseña. 10. El sistema actualiza la nueva contraseña en la base de datos. 11. El sistema registra la acción en <code>audit_change</code> y <code>log_evento</code>. 12. El sistema redirige al usuario nuevamente al login. 13. El usuario ingresa las credenciales nuevamente con su contraseña nueva. 14. El sistema valida los campos. 15. El usuario es redirigido a la vista principal del dashboard. 16. Se registra la acción en <code>“audit_change”</code>. 17. El caso de uso finaliza exitosamente. 	

Flujos alternos	
Flujo alternativo validación de campos vacíos	<ul style="list-style-type: none"> • El usuario deja campos en blanco. • Se muestra un mensaje de error. • El flujo regresa al punto 3 del flujo principal.
Flujo alternativo validación de reglas de contraseña	<ul style="list-style-type: none"> • La contraseña no cumple con los parámetros establecidos para la contraseña. • Se muestra un mensaje de error. • El flujo regresa al punto 3 del flujo principal.
Flujo alternativo validación de credenciales	<ul style="list-style-type: none"> • Las credenciales no son válidas y rechaza el inicio de sesión. • El sistema muestra un mensaje de error. • El flujo regresa al punto 3 del flujo principal.
Flujo alternativo validación de campo cambio de contraseña	<ul style="list-style-type: none"> • El usuario ingresa el usuario y la contraseña. • El sistema valida el campo de cambio_contraseña = true. • Si es true: <ul style="list-style-type: none"> ○ El flujo se redirige a la vista de cambio de contraseña • El usuario ingresa la nueva contraseña. • El sistema valida la nueva contraseña. • El usuario selecciona el botón de actualizar. • El flujo regresa al punto 2 del flujo principal.
Flujo alternativo máximo de intentos permitidos	<ul style="list-style-type: none"> • El usuario supera el número máximo de intentos fallidos. • El sistema bloquea temporalmente la cuenta y se registra en el audit_change. • El flujo finaliza sin acceso al sistema.
Flujo alternativo error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza sin acceso al sistema.
Requerimientos especiales	
Cumplimiento de autenticación, bloqueo por intentos fallidos y trazabilidad de accesos.	
Todas las acciones deben registrarse en audit_change y log_evento.	

Las contraseñas deben viajar encriptadas.
Postcondiciones
El usuario queda autenticado y con una sesión activa. Si el cambio fue exitoso, el campo cambiar_contrasenna = false. El usuario puede iniciar sesión con una nueva contraseña. Se registran en “audit_change” los cambios realizados.

Fuente: Elaboración propia, 2025.

Tabla 18

Caso de uso para actualizar datos de un usuario

Prototipo funcional para la gestión de Recursos Humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MS-03	Nombre del caso de uso: Actualizar datos de un usuario
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Permite al usuario administrador de RRHH actualizar los datos y la contraseña de un usuario para recuperar su cuenta
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrador de RRHH Sistema de gestión de Recursos Humanos
Precondiciones:	El usuario Administrador de RRHH debe estar registrado en el sistema. El usuario Administrador de RRHH debe tener los permisos correspondientes para actualizar la contraseña de un usuario. El sistema debe estar disponible.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo que le permite al usuario Administrador de RRHH actualizar los datos de la cuenta de un usuario y la contraseña para crear una contraseña temporal que le permita a un usuario del sistema recuperar su cuenta.	
<ol style="list-style-type: none"> 1. El usuario Administrador de RRHH accede al módulo de seguridad. 2. Selecciona la opción Usuarios del Sistema. 3. El sistema valida si el usuario tiene los permisos correspondientes para ingresar. 4. El sistema despliega la vista con todos los empleados disponibles. 5. El usuario selecciona el botón Actualizar del empleado que necesita recuperar su cuenta. 6. El sistema despliega el formulario Actualizar Cuenta. 	

<p>7. El usuario ingresa los datos que se le solicitan en el formulario.</p> <p>8. El sistema valida que los campos cumplan con los requerimientos.</p> <p>9. El usuario da clic en el botón de actualizar.</p> <p>10. El sistema regresa a la vista con todos los empleados.</p> <p>11. El caso de uso finaliza exitosamente.</p>	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • El usuario deja campos en blanco. • Se muestra un mensaje de error. • El flujo regresa al punto 7 del flujo principal.
Flujo Alternativo Validación de reglas de contraseña	<ul style="list-style-type: none"> • La contraseña no cumple con los parámetros establecidos para la contraseña. • Se muestra un mensaje de error. • El flujo regresa al punto 7 del flujo principal.
Flujo Alternativo Validación de coincidencia de contraseñas	<ul style="list-style-type: none"> • El usuario digita los campos de nueva contraseña y confirmar contraseña. • El sistema muestra un mensaje de error si ambos campos no coinciden. • El flujo regresa al punto 7 del flujo principal.
Flujo Alternativo Actualización de campo cambio_contrasenna	<ul style="list-style-type: none"> • El usuario da clic en el botón Actualizar. • El sistema actualiza el campo de cambio_contrasenna = true.
Flujo Alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Cumplimiento de autenticación.</p> <p>Todas las acciones deben registrarse en “audit_change” y “log_evento”.</p> <p>Solo los usuarios con rol Administrador de RRHH o Administrador del Sistema pueden modificar cuentas de usuario.</p> <p>El sistema debe mantener control de trazabilidad sobre cada modificación de datos sensibles.</p>	
Postcondiciones	
<p>El usuario afectado recupera la cuenta.</p> <p>Si el cambio fue exitoso, el campo cambiar_contrasenna = true.</p>	

El usuario puede iniciar sesión con una nueva contraseña temporal.
Se registra en “audit_change” los cambios realizados.

Fuente: Elaboración propia, 2025.

Tabla 19

Caso de uso para crear un usuario del sistema

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S. A, ubicada en Grecia	
Número de caso de uso: CU-MS-04	Nombre del caso de uso: Crear usuario
Fecha elaboración:	10/09/2025
Descripción Caso de Uso:	Permite al usuario Administrador de RRHH crear un nuevo usuario en el sistema
Autor de caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrador de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario Administrador de RRHH debe estar registrado en el sistema. El usuario Administrador de RRHH debe tener los permisos correspondientes para crear un usuario. Se debe tener acceso a los roles disponibles. El sistema debe estar disponible.
Flujo básico del caso de uso	
Este caso de uso describe el proceso completo que permite al usuario Administrador de RRHH crear un usuario del sistema, con su rol y permisos correspondientes.	
<ol style="list-style-type: none"> 1. El usuario Administrador de RRHH accede al módulo de seguridad. 2. Selecciona la opción Usuarios del Sistema. 3. El sistema valida si el usuario tiene los permisos correspondientes para ingresar. 4. El sistema despliega la vista con todos los empleados disponibles. 5. El usuario da clic en el botón Crear Usuario. 6. El sistema despliega el formulario Nuevo Usuario. 7. El usuario ingresa los datos que se le solicitan en el formulario. 8. El sistema valida que los campos cumplan con los requerimientos. 9. El usuario da clic en el botón Crear. 10. El sistema regresa a la vista con todos los usuarios. 11. El caso de uso finaliza exitosamente. 	
Flujos alternos	

Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • El usuario deja campos en blanco. • Se muestra un mensaje de error. • El flujo regresa al punto 6 del flujo principal.
Flujo alternativo Validación de reglas de contraseña	<ul style="list-style-type: none"> • La contraseña no cumple con los parámetros establecidos para la contraseña. • Se muestra un mensaje de error. • El flujo regresa al punto 6 del flujo principal.
Flujo alternativo Validación de coincidencia de contraseñas	<ul style="list-style-type: none"> • El usuario digita los campos de nueva contraseña y confirmar contraseña. • El sistema muestra un mensaje de error si ambos campos no coinciden. • El flujo regresa al punto 6 del flujo principal.
Flujo Alternativo Actualización de campo cambio_contrasenna	<ul style="list-style-type: none"> • El usuario da clic en el botón Actualizar. • El sistema actualiza el campo de cambio de contrasenna = true.
Flujo Alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Cumplimiento de autenticación.</p> <p>Todas las acciones deben registrarse en “audit_change” y “log_evento”.</p> <p>Las contraseñas deben viajar encriptadas.</p>	
Postcondiciones	
<p>Se crea un nuevo usuario en el sistema.</p> <p>Si la creación del usuario fue exitosa, el campo cambiar_contrasenna = true.</p> <p>El usuario puede iniciar sesión con una nueva contraseña temporal.</p> <p>Se registran en “audit_change” los cambios realizados.</p>	

Fuente: Elaboración propia, 2025.

Tabla 20

Caso de uso para eliminar una cuenta de usuario

<p>Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia</p>

Número de caso de uso: CU-MS-05	Nombre del caso de uso: Eliminar una cuenta de usuario
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al usuario administrador de RRHH eliminar un usuario en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrador de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario Administrador de RRHH debe estar registrado en el sistema. El usuario Administrador de RRHH debe tener los permisos correspondientes para eliminar un usuario. El sistema debe estar disponible.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo que le permite al usuario Administrador de RRHH eliminar un usuario del sistema.	
<ol style="list-style-type: none"> 1. El usuario Administrador de RRHH accede al módulo de seguridad. 2. Selecciona la opción Usuarios del sistema. 3. El sistema valida si el usuario tiene los permisos correspondientes para ingresar. 4. El sistema despliega la vista con todos los usuarios disponibles. 5. El usuario da clic en el botón Eliminar correspondiente al usuario que se debe eliminar. 6. El sistema despliega una ventana de confirmación y pregunta si desea eliminar este usuario. 7. El usuario da clic en el botón Sí del mensaje de confirmación. 8. El sistema cierra la ventana de confirmación. 9. El sistema regresa a la vista con todos los usuarios. 10. El caso de uso finaliza exitosamente. 	
Flujos alternos	
Flujo alternativo cancelar eliminar usuario	<ul style="list-style-type: none"> • El usuario da clic en el botón No de la ventana de confirmación. • El sistema cancela la operación. • El flujo regresa al punto 4.
Flujo alternativo	<ul style="list-style-type: none"> • Si el usuario da clic en el botón Sí de la ventana de confirmación. • El campo activo se cambia a false en la base de datos.

Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede eliminar el usuario debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Cumplimiento de autenticación. Todas las acciones deben registrarse en “audit_change” y “log_evento”.	
Postcondiciones	
Se elimina un usuario en el sistema. El usuario queda inhabilitado para iniciar sesión en el sistema. Se registran en audit_change los cambios realizados.	

Fuente: Elaboración propia, 2025.

Tabla 21

Caso de uso para acceder a los usuarios activos

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MS-06	Nombre del caso de uso: Acceder a todos los usuarios activos en el sistema
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al usuario administrador de RRHH ver todos los usuarios activos en el sistema para poder realizar acciones sobre ellos.
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario administrador de RRHH Sistema de gestión de Recursos Humanos
Precondiciones:	El usuario administrador de RRHH debe estar registrado en el sistema. El usuario administrador de RRHH debe tener los permisos correspondientes para crear un usuario. El sistema debe estar disponible.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo que le permite al usuario administrador de RRHH el acceso a la lista de todos los usuarios activos del sistema, y poder realizar acciones sobre estos, como actualizar datos y eliminarlos, además de crear nuevos usuarios.	

<ol style="list-style-type: none"> 1. El usuario administrador de RRHH accede al módulo de seguridad. 2. Selecciona la opción Usuarios del sistema. 3. El sistema valida si el usuario tiene los permisos correspondientes para ingresar. 4. El sistema despliega la vista con todos los empleados disponibles. 5. El caso de uso finaliza exitosamente. 	
Flujos alternos	
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Cumplimiento de autenticación.	
Todas las acciones deben registrarse en “audit_change” y “log_evento”.	
Postcondiciones	
Se registran en “audit_change” los cambios realizados.	

Fuente: Elaboración propia, 2025.

Tabla 22

Caso de uso para listar empleados activos

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-ME-01	Nombre del caso de uso: Listar todos los empleados activos en el sistema
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Les permite al encargado de RRHH y al usuario de RRHH observar los datos activos en el sistema
Autor de caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Usuario de RRHH Sistema de gestión de recursos humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. Acceso a la base de datos.
Flujo básico del caso de uso	

<p>En este caso de uso se describe el proceso completo para listar a todos los empleados activos en el sistema con sus datos personales del empleado; además de contar con los botones necesarios para realizar las acciones de editar, eliminar y agregar a la persona.</p>	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Personas del módulo de empleados. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra los datos personales de cada empleado, como nombre, apellidos, número de identificación, teléfonos, correo, dirección, estado civil, fecha de nacimiento. 4. La vista cuenta con los botones Editar, Eliminar y Crear persona. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.
Flujo alternativo botón Editar	<ul style="list-style-type: none"> • El usuario selecciona el botón Editar. • El sistema redirecciona a la vista Actualizar datos.
Flujo alternativo del botón Eliminar	<ul style="list-style-type: none"> • El usuario selecciona el botón Eliminar • El sistema muestra una ventana de confirmación.
Flujo alternativo del botón Agregar persona	<ul style="list-style-type: none"> • El usuario selecciona el botón Agregar persona. • El sistema redirecciona la vista Crear persona.
Flujo alternativo botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar • El sistema redirecciona la vista principal del sistema
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Validación de empleado activo. Auditoría extendida con trazabilidad. Asociación transaccional entre persona, dirección y teléfonos.</p>	
Postcondiciones	
<p>Acceso a los datos personales de los empleados.</p>	

Fuente: Elaboración propia, 2025.

Tabla 23*Caso de uso para registrar un nuevo empleado*

Prototipo funcional para la gestión de Recursos Humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-ME-02	Nombre del caso de uso: Registrar un nuevo empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Les permite al encargado de RRHH y al usuario registrar un nuevo empleado en el sistema con toda su información personal y sus validaciones correspondientes.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Usuario de RRHH Sistema de gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. Los catálogos de provincia, cantón, distrito y tipo de identificación deben estar disponibles.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para registrar a un nuevo empleado en el sistema, incluidos la carga de catálogos, el ingreso de datos personales, las validaciones de consistencia y la confirmación del registro exitoso.	
<ol style="list-style-type: none"> 6. El usuario accede a la opción Empleados del módulo de empleados. 7. El sistema muestra la vista con todos los empleados activos en el sistema. 8. El usuario selecciona la opción “Registrar persona”. 9. El sistema despliega el formulario de registro. 10. El sistema carga los catálogos: tipo de identificación, provincia, cantón, distrito. 11. La encargada ingresa los datos obligatorios del empleado. 12. El usuario selecciona el tipo de identificación y lo ingresa. 13. El usuario agrega uno o más teléfonos celulares y selecciona uno como el principal. 14. Usuario selecciona provincia, cantón y distrito e ingresa la dirección exacta. 15. El sistema valida todos los campos. 16. La encargada de RRHH selecciona la opción Guardar. 17. El sistema almacena la información en la base de datos. 18. El sistema confirma el registro exitoso. 19. El sistema sale del formulario y regresa a la vista principal del módulo de empleados. 	

20. Finaliza el caso de uso exitosamente.	
Flujos alternos	
Flujo alternativo Validación de identidad duplicada	<ul style="list-style-type: none"> • El sistema consulta si el número de identificación ya existe. • Si existe se muestra un mensaje de advertencia. • Y se bloquea el envío.
Flujo alternativo Validación de formato según el tipo de identificación	<ul style="list-style-type: none"> • Se verifica que el formato del número coincida con el tipo seleccionado (ej. cédula o permiso de trabajo de Migración). • Si no coincide se muestra una advertencia. • Y se bloquea el envío.
Flujo alternativo Validación de dirección incompleta	<ul style="list-style-type: none"> • Se verifica qué provincia, cantón y distrito estén seleccionados. • Si falta alguno se bloquea el envío.
Flujo alternativo validación de teléfono duplicado	<ul style="list-style-type: none"> • Se verifica que no se repita el número de teléfono. • Si hay duplicados se muestra una advertencia. y se bloquea el envío.
Flujo alternativo Validación de formato de teléfono fijo y celular	<ul style="list-style-type: none"> • Se valida que el número cumpla con el patrón nacional. • Si no cumple se rechaza el envío.
Flujo alternativo Asociación fallida entre persona y entidades relacionadas (tipo de identificación, teléfono celular y dirección)	<ul style="list-style-type: none"> • Si ocurre error en la asociación se revierte la transacción. • Se registra el evento en el log de trazabilidad.
Flujo alternativo cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación entre tipo de identificación y el formato.	

Auditoría extendida con trazabilidad de eventos, PI y usuario. Uso de validadores personalizados para duplicidad y formato de teléfono fijo y teléfono celular. Asociación transaccional entre persona, dirección y números de teléfono.
Postcondiciones
La persona queda registrada con su tipo de identificación, números de teléfono y dirección. El teléfono celular principal queda marcado correctamente. El evento queda registrado en el sistema de trazabilidad.

Fuente: Elaboración propia, 2025.

Tabla 24

Caso de uso para actualizar los datos de un empleado

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-ME-03	Nombre del caso de uso: Actualizar los datos del empleado
Fecha de elaboración:	10/09/2025
Descripción de caso de uso:	Le permite al encargado de RRHH y al usuario de RRHH actualizar datos del empleado en el sistema con toda su información personal y sus validaciones.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo Usuario de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe tener los permisos correspondientes para realizar la acción. Los catálogos de provincia, cantón, distrito y tipo de identificación deben estar disponibles. El empleado tiene que estar registrado previamente en el sistema. Acceso a los datos del empleado en la base de datos.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para actualizar los datos de un empleado registrado en el sistema, incluidos la carga de catálogos, el ingreso de datos personales, las validaciones de consistencia y la confirmación del registro exitoso.	
<ol style="list-style-type: none"> 1. El usuario selecciona a la persona del módulo de empleados. 2. El sistema muestra la pantalla de personas con todos los empleados activos del sistema. 	

<ol style="list-style-type: none"> 3. El usuario selecciona el botón Editar dentro de la información del empleado que desea actualizar sus datos. 4. El sistema despliega el formulario de registro, pero cargado con los datos ya registrados en el sistema del empleado. 5. El sistema carga los catálogos: tipo de identificación, provincia, cantón, distrito. 6. El usuario modifica los datos que desea actualizar de la persona. 7. El sistema valida todos los campos. 8. El usuario selecciona la opción “Guardar”. 9. El sistema almacena la información en la base de datos. 10. El sistema confirma el registro exitoso. 11. El sistema sale del formulario y regresa a la vista principal del módulo de empleados. 12. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo validación de identidad existente	<ul style="list-style-type: none"> • El sistema consulta si el número de identificación ya existe. • Si no existe se muestra un mensaje de advertencia y se bloquea el envío.
Flujo alternativo Validación de formato según tipo de identificación	<ul style="list-style-type: none"> • Se verifica que el formato del número coincida con el tipo seleccionado (ej. cédula o permiso de trabajo de Migración). • Si no coincide se muestra una advertencia.
Flujo alternativo Validación de dirección incompleta	<ul style="list-style-type: none"> • Se verifica que provincia, cantón y distrito estén seleccionados. • Si falta alguno se bloquea el envío.
Flujo alternativo validación de teléfono duplicado	<ul style="list-style-type: none"> • Se verifica que no se repita el número de teléfono. • Si hay duplicados se muestra una advertencia.
Flujo alternativo validación de formato de teléfono y de celular	<ul style="list-style-type: none"> • Se valida que el número cumpla con el patrón nacional. • Si no cumple se rechaza el envío.
Flujo alternativo Asociación fallida entre persona y entidades relacionadas (tipo de identificación, teléfono celular y dirección)	<ul style="list-style-type: none"> • Si ocurre error en la asociación se revierte la transacción. • Se registra el evento en el log de trazabilidad.
Flujo alternativo Cancelar el proceso	<ul style="list-style-type: none"> • El usuario selecciona el botón Cancelar. • El sistema no realiza ningún campo.

	<ul style="list-style-type: none"> • Regresa a la pantalla principal de Personas.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación entre tipo de identificación y formato.	
Auditoría extendida con trazabilidad de eventos, PI y usuario.	
Uso de validadores personalizados para duplicidad y formato de teléfono fijo y teléfono celular.	
Asociación transaccional entre persona, dirección y teléfonos.	
Postcondiciones	
Se actualizan los datos de la persona con su tipo de identificación, teléfonos asociados con el caso y dirección.	
El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 25

Caso de uso para eliminar a un empleado

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-ME-04	Nombre del caso de uso: Eliminar un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH eliminar a un empleado en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe contar con permisos de eliminación. El empleado tiene que estar registrado previamente en el sistema y estar activo.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para eliminar de forma lógica a un empleado registrado en el sistema, por medio de un usuario del sistema con los permisos correspondientes para realizar la acción, lo que asegura que los datos históricos se conserven para fines de auditoría pero manteniendo la trazabilidad.	

<ol style="list-style-type: none"> 1. El usuario accede al módulo de gestión de empleados. 2. El sistema despliega la lista de empleados activos. 3. El usuario selecciona la opción de eliminar en el empleado correspondiente. 4. El sistema valida que el usuario tenga permisos para realizar la acción. 5. El sistema solicita confirmación al usuario mediante un cuadro de diálogo. 6. El usuario confirma la eliminación. 7. El sistema actualiza el estado del empleado en la base de datos, y lo actualiza como inactivo (activo = false). 8. El sistema registra la acción en las tablas de “audit_change” y “log_evento”. 9. El sistema regresa a la vista de personas. 10. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de permisos	<ul style="list-style-type: none"> • El sistema valida si el usuario tiene los permisos correspondientes para realizar la acción. • Si el usuario no tiene los permisos correspondientes el botón de eliminar se encuentra desactivado.
Flujo alternativo Validación de persona existente	<ul style="list-style-type: none"> • El sistema consulta por id si la persona existe en el sistema. • Si no existe se muestra un mensaje de advertencia y se bloquea el envío.
Flujo alternativo validación de confirmación de eliminación	<ul style="list-style-type: none"> • El sistema muestra un mensaje de conformación para eliminar al empleado. • Si se confirma la eliminación se realiza el acto de eliminación.
Flujo alternativo Cancelar la operación	<ul style="list-style-type: none"> • El usuario desea cancelar la acción. • En la ventana de conformación selecciona la opción No. • El sistema cierra el cuadro de dialogo y no realiza ninguna acción relacionada con el empleado. • Retorna a la vista de empleados.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error.

	<ul style="list-style-type: none"> El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Registro en la tabla de Log Evento por usuario.</p> <p>Auditoría extendida con trazabilidad de eventos, PI y usuario.</p> <p>El sistema debe garantizar que se realice un borrado lógico y no físico para preservar la trazabilidad histórica.</p>	
Postcondiciones	
<p>Se elimina a la persona y queda marcado como inactivo en el sistema.</p> <p>Se conserva toda la información histórica del empleado.</p> <p>El registro de la acción queda almacenado en “auth_change” y “log_evento”.</p> <p>El sistema refleja los cambios en la lista de empleados activos.</p>	

Fuente: Elaboración propia, 2025.

Tabla 26

Caso de uso para listar la asistencia del día

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número caso de uso: CU-MA-01	Nombre del caso de uso: Listar los registros de asistencia del día actual
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Les permite al encargado de RRHH y al usuario de RRHH observar los datos activos en el sistema
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Usuario de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. Tener los permisos correspondientes para realizar la acción. Acceso a la base de datos.
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para listar a todos los empleados activos en el sistema con sus datos de la asistencia del día actual, como nombre del empleado, fecha del registro, hora de entrada, hora de salida, estado (presente, ausente o tardía), descripción y asistencia</p>	

completada. Además de contar con los botones necesarios para realizar las acciones de editar, eliminar, agregar, y el botón de regresar a la pantalla principal del sistema y salir del módulo.	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Asistencia del módulo de asistencia. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha, hora de entrada, hora de salida, estado; si esta completada o no, descripción. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo y validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.
Flujo alternativo del botón Editar	<ul style="list-style-type: none"> • El usuario selecciona el botón Editar. • El sistema redirecciona a la vista de Actualizar datos.
Flujo alternativo botón Eliminar	<ul style="list-style-type: none"> • El usuario selecciona el botón Eliminar • El sistema muestra una ventana de confirmación.
Flujo alternativo botón Agregar Asistencia	<ul style="list-style-type: none"> • El usuario selecciona el botón Agregar. • El sistema redirecciona a la vista Crear un Registro de Asistencia.
Flujo alternativo botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar • El sistema redirecciona a la vista principal del sistema
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de empleado activo. Auditoría extendida con trazabilidad.	
Postcondiciones	
Se observa el registro de los datos de asistencia de los empleados en el día actual de trabajo.	

Fuente: Elaboración propia, 2025.

Tabla 27*Caso de uso para agregar un registro de asistencia*

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MA-02	Nombre del caso de uso: Agregar un registro de asistencia de un empleado
Fecha elaboración:	10/09/2025
Descripción caso de uso:	Les permite al encargado de RRHH y al usuario de RRHH agregar un nuevo registro de asistencia de un empleado.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Usuario de RRHH Sistema de gestión de recursos humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Datos del empleado previamente registrados.
Flujo básico del caso de uso	
Este caso de uso describe el proceso completo para agregar un nuevo registro de asistencia de un empleado en el sistema, la fecha, la hora de entrada, la hora de salida, el estado, observaciones y asistencia completada.	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Asistencia del módulo de asistencia. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra datos del día actual como nombre del empleado, fecha, hora de entrada, hora de salida, estado, si esta completada o no, descripción. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario da clic al botón Agregar en la persona correspondiente. 6. El sistema redirecciona a la vista de Agregar asistencia y despliega el formulario. 7. El usuario completa los datos requeridos del empleado, como fecha, hora de entrada, hora de salida, estado, descripción y completado o incompleto. 8. El sistema valida todos los campos. 9. El usuario selecciona la opción “Guardar”. 10. El sistema almacena la información en la base de datos. 11. El sistema confirma el registro exitoso. 	

<p>12. El sistema sale del formulario y regresa a la vista principal del módulo de asistencia.</p> <p>13. Finaliza el caso de uso exitosamente.</p>	
Flujos alternos	
Flujo alternativo validación de fecha no futura	<ul style="list-style-type: none"> • Se verifica que la fecha no sea futura al día presente. • El sistema bloquea los días futuros.
Flujo alternativo validación de hora	<ul style="list-style-type: none"> • El sistema verifica que la hora esté dentro del rango de horas trabajadas. • Si no cumple se muestra un mensaje de advertencia. • Y se bloquea el envío.
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo alternativo Validación máximo de horas	<ul style="list-style-type: none"> • Se valida que las horas de entrada y de salida no sean mayores a 8 horas. • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de asistencia.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Validación de fecha futura, máximo de horas diarias, de horario y de campos requeridos.</p> <p>Auditoría extendida con trazabilidad de eventos.</p>	
Postcondiciones	
<p>La condición de la asistencia del empleado queda registrada.</p> <p>El evento queda registrado en el sistema de trazabilidad.</p>	

Fuente: Elaboración propia, 2025.

Tabla 28*Caso de uso para actualizar un registro de asistencia*

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MA-03	Nombre del caso de uso: Actualizar los datos del registro de asistencia
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH y al usuario de RRHH actualizar datos del registro de asistencia del empleado en el sistema con toda su información correspondiente y sus validaciones.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario administrativo Usuario de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe tener los permisos necesarios para realizar la acción. El empleado tiene que estar registrado previamente en el sistema. El registro debe estar creado previamente. Acceso a los datos del empleado en la base de datos.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para actualizar los datos de un registro de asistencia de un empleado, incluidas las validaciones de consistencia y botones Guardar y Cancelar.	
<ol style="list-style-type: none"> 1. El usuario selecciona asistencia del módulo Asistencia. 2. El sistema muestra la pantalla de asistencia con todos los empleados activos del sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha, hora de entrada, hora de salida, estado, si esta completada o no, descripción. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón Editar. 6. El sistema redirecciona a la vista Actualizar asistencia y muestra el formulario con los datos previamente registrados cargados. 7. El usuario actualiza los registros correspondientes. 8. El sistema valida todos los campos. 9. El usuario selecciona la opción Guardar. 	

<p>10. El sistema almacena la información en la base de datos.</p> <p>11. El sistema confirma el registro exitoso.</p> <p>12. El sistema sale del formulario y regresa a la vista principal del módulo de asistencia.</p> <p>13. Finaliza el caso de uso exitosamente.</p>	
Flujos alternos	
Flujo alternativo Validación de fecha no futura	<ul style="list-style-type: none"> • Se verifica que la fecha no sea futura al día presente. • El sistema bloquea los días futuros.
Flujo Alternativo Validación de hora	<ul style="list-style-type: none"> • El sistema verifica que la hora esté dentro del rango de horas trabajadas. • Si no cumple se muestra un mensaje de advertencia. • Y se bloquea el envío.
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo alternativo validación máximo de horas	<ul style="list-style-type: none"> • Se valida que entre la hora de entrada y la de salida no sea mayor a 8 horas. • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo Alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de asistencia.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Validación de fecha futura, máximo de horas diarias, de horario y de campos requeridos.</p> <p>Auditoría extendida con trazabilidad de eventos.</p>	
Postcondiciones	

Se actualizan los datos de la asistencia del empleado seleccionado.
El evento queda registrado en el sistema de trazabilidad.

Fuente: Elaboración propia, 2025.

Tabla 29

Caso de uso para eliminar un registro de asistencia

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MA-04	Nombre del caso de uso: Eliminar un registro de asistencia
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH eliminar un registro de asistencia de un empleado en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe contar con permisos de eliminación. El empleado tiene que estar registrado previamente en el sistema y estar activo. Se tiene que haber creado previamente el registro de asistencia.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para eliminar de forma lógica un registro de asistencia de un empleado, lo que asegura que los datos históricos se conserven para fines de auditoría y mantienen la trazabilidad.	
<ol style="list-style-type: none"> 1. El usuario selecciona Asistencia del módulo de empleados. 2. El sistema muestra la pantalla de asistencia con todos los empleados activos del sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha, hora de entrada, hora de salida, estado, si esta completada o no, descripción. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón Eliminar del empleado que requiere eliminar el registro. 6. El usuario confirma la eliminación. 7. El sistema actualiza el estado del empleado en la base de datos, actualizada como Inactivo (activo = false). 8. El sistema registra la acción en las tablas de “audit_change” y “log_evento”. 9. El sistema regresa a la vista de principal de Asistencia. 	

10. Finaliza el caso de uso exitosamente.	
Flujos alternos	
Flujo Alternativo Validación de permisos	<ul style="list-style-type: none"> • El sistema valida si el usuario tiene los permisos correspondientes para realizar la acción. • Si el usuario no tiene los permisos correspondientes el botón de eliminar se encuentra desactivado.
Flujo alternativo Validación de confirmación de eliminación	<ul style="list-style-type: none"> • El sistema muestra un mensaje de conformación para eliminar al empleado. • Si se confirma la eliminación se realiza la acción de eliminación. • El sistema regresa a la vista principal de asistencia.
Flujo alternativo Cancelar la operación	<ul style="list-style-type: none"> • El usuario desea cancelar la acción. • En la ventana de conformación selecciona la opción No. • El sistema cierra el cuadro de dialogo y no realiza ninguna acción relativa al empleado. • Retorna a la vista de asistencia.
Flujo alternativo de error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Registro en la tabla de Log Evento por usuario.</p> <p>Auditoria extendida con trazabilidad de eventos.</p> <p>El sistema debe garantizar que se realice un borrado lógico y no físico para preservar la trazabilidad histórica.</p>	
Postcondiciones	
<p>Se elimina el registro de asistencia de la persona y queda marcado como inactivo en el sistema.</p> <p>Se conserva toda la información histórica de la asistencia del empleado.</p> <p>El registro de la acción queda almacenado en “auth_change” y ”log_evento”.</p> <p>El sistema refleja los cambios en la lista de asistencia actual.</p>	

Fuente: Elaboración propia, 2025.

Tabla 30*Caso de uso para listar registros de horas extra*

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MHE-01	Nombre del caso de uso: Listar los registros de horas extra de los empleados del día actual
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH observar los datos activos en el sistema
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. Tener los permisos correspondientes para realizar la acción. Acceso a la base de datos.
Flujo Básico del caso de uso	
En este caso de uso se describe el proceso completo para listar a todos los empleados activos en el sistema con sus datos de horas extra registradas del día actual, como nombre del empleado, fecha del registro, hora de entrada, hora de salida, estado (aprobado o rechazado), y motivo. Además de contar con los botones necesarios para realizar las acciones: Editar, Eliminar, Agregar y el botón de regresar a la pantalla principal del sistema para que el usuario pueda realizar acciones.	
<ol style="list-style-type: none"> 1. El usuario accesa la opción Horas extra del módulo de asistencia. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha, hora de entrada, hora de salida, estado, y el motivo. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.
Flujo Alternativo Botón Editar	<ul style="list-style-type: none"> • El usuario selecciona el botón Editar. • El sistema redirecciona a la vista de actualizar datos.

Flujo alternativo botón Eliminar	<ul style="list-style-type: none"> • El usuario selecciona el botón Eliminar. • El sistema muestra una ventana de confirmación.
Flujo alternativo del botón Agregar asistencia	<ul style="list-style-type: none"> • El usuario selecciona el botón Agregar. • El sistema redirecciona a la vista Crear un Registro de horas extra.
Flujo alternativo del botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar. • El sistema redirecciona a la vista principal del sistema.
Flujo alternativo del botón Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de empleado activo. Auditoría extendida con trazabilidad.	
Postcondiciones	
Se accede a la información de las horas extras de los empleados activos en el sistema.	

Fuente: Elaboración propia, 2025.

Tabla 31

Caso de uso para agregar un registro de horas extra

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MHE-02	Nombre del caso de uso: Agregar un registro de horas extra de un empleado
Fecha de elaboración:	10/09/2025
Descripción de caso de uso:	Le permite al encargado de RRHH agregar un nuevo registro de asistencia de un empleado.
Autor de un caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado.

	<p>El usuario debe tener los permisos correspondientes para realizar la acción.</p> <p>Datos del empleado previamente registrados.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para agregar un nuevo registro de horas extra de un empleado en el sistema, la fecha, la hora de entrada, la hora de salida, el estado y el motivo.</p>	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Horas extra del módulo de horas extra. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha, hora de entrada, hora de salida, estado, y el motivo. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario da clic al botón Agregar en la persona correspondiente. 6. El sistema redirecciona a la vista Agregar horas extra y despliega el formulario. 7. El usuario completa los datos requeridos del empleado, como fecha, hora de entrada, hora de salida, estado y motivo. 8. El sistema valida todos los campos. 9. El usuario selecciona la opción Guardar. 10. El sistema almacena la información en la base de datos. 11. El sistema confirma el registro exitoso. 12. El sistema sale del formulario y regresa a la vista principal del módulo de asistencia. 13. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de fecha no futura	<ul style="list-style-type: none"> • Se verifica que la fecha no sea futura al día presente. • El sistema bloquea los días futuros.
Flujo alternativo Validación de hora	<ul style="list-style-type: none"> • El sistema verifica que la hora esté dentro del rango de horas trabajadas. • Si no cumple se muestra un mensaje de advertencia. • Y se bloquea el envío.
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.

Flujo alternativo Validación máximo de horas	<ul style="list-style-type: none"> • Se valida que entre la hora de entrada y la de salida no transcurran más de 12 horas diarias al sumar las horas base. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de horas extra.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha futura, máximo de horas diarias, de horario y de campos requeridos. Auditoría extendida con trazabilidad de eventos.	
Postcondiciones	
La condición de las horas extra del empleado queda registrada. El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 32

Caso de uso Actualizar registro de horas extra

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MHE-03	Nombre del caso de uso: Actualizar los datos del registro de horas extra
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH actualizar datos del registro de horas extra del empleado en el sistema, con toda su información correspondiente y sus validaciones.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión.

	<p>El usuario debe tener los permisos necesarios para realizar la acción.</p> <p>El empleado tiene que estar registrado previamente en el sistema.</p> <p>El registro debe estar creado previamente.</p> <p>Acceso a los datos del empleado en la base de datos.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para actualizar los datos de un registro de asistencia de un empleado, incluidas las validaciones de consistencia y los botones Guardar y Cancelar.</p>	
<ol style="list-style-type: none"> 1. El usuario selecciona horas extra del módulo de horas extra. 2. El sistema muestra la pantalla de horas extra con todos los empleados activos del sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha, hora de entrada, hora de salida, estado y motivo. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón Editar. 6. El sistema redirecciona a la vista Actualizar horas extra y muestra el formulario con los datos previamente registrados cargados. 7. El usuario actualiza los registros correspondientes. 8. El sistema valida todos los campos. 9. El usuario selecciona la opción “Guardar”. 10. El sistema almacena la información en la base de datos. 11. El sistema confirma el registro exitoso. 12. El sistema sale del formulario y regresa a la vista principal del módulo de horas extra. 13. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de fecha no futura	<ul style="list-style-type: none"> • Se verifica que la fecha no sea futura al día presente. • El sistema bloquea los días futuros.
Flujo alternativo Validación de hora	<ul style="list-style-type: none"> • El sistema verifica que la hora esté dentro del rango de horas trabajadas. • Si no cumple se muestra un mensaje de advertencia. • Y se bloquea el envío.
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios.

	<ul style="list-style-type: none"> • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo alternativo Validación máximo de horas	<ul style="list-style-type: none"> • Se valida que entre la hora de entrada y la de salida no transcurran más de 8 horas. • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de horas extra.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha futura, máximo de horas diarias, de horario y de campos requeridos. Auditoría extendida con trazabilidad de eventos.	
Postcondiciones	
El registro de horas extra del empleado es actualizado correctamente. El evento queda registrado en el sistema de auditoría y trazabilidad. El sistema garantiza la integridad de la información.	

Fuente: Elaboración propia, 2025.

Tabla 33

Caso de uso para eliminar un registro de horas extra

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MHE-04	Nombre del caso de uso: Eliminar un registro de horas extra de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH eliminar un registro de horas extra de un empleado en el sistema
Autor del caso de uso:	Andrés Gerardo González Villegas

Actores relacionados:	<p>Usuario administrativo RRHH</p> <p>Sistema de Gestión de Recursos Humanos</p>
Precondiciones:	<p>El usuario debe haber iniciado sesión.</p> <p>El usuario debe contar con permisos de eliminación.</p> <p>El empleado tiene que estar registrado previamente en el sistema y estar activo.</p> <p>Debe haberse creado previamente el registro de asistencia.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para eliminar de forma lógica un registro de horas extra de un empleado, para asegurar que los datos históricos se conserven para fines de auditoría y para que mantenga la trazabilidad.</p>	
<ol style="list-style-type: none"> 1. El usuario selecciona Horas extra del módulo de horas extra. 2. El sistema muestra la pantalla de horas extra con todos los empleados activos del sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha, hora de entrada, hora de salida, estado y motivo. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón Eliminar del empleado que requiere eliminar el registro. 6. El usuario confirma la eliminación. 7. El sistema actualiza el estado del empleado en la base de datos, y lo actualiza como inactivo (activo = false). 8. El sistema registra la acción en las tablas de “audit_change” y “log_evento”. 9. El sistema regresa a la vista principal de Horas extra. 10. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de permisos	<ul style="list-style-type: none"> • El sistema valida si el usuario tiene los permisos correspondientes para realizar la acción. • Si el usuario no tiene los permisos correspondientes el botón eliminar se encuentra desactivado.
Flujo alternativo Validación de confirmación de eliminación	<ul style="list-style-type: none"> • El sistema muestra un mensaje de conformación para eliminar al empleado. • Si se confirma la eliminación se realiza la acción de eliminación. • El sistema regresa a la vista principal de horas extras.

Flujo alternativo Cancelar la operación	<ul style="list-style-type: none"> • El usuario desea cancelar la acción. • En la ventana de conformación selecciona la opción No. • El sistema cierra el cuadro de dialogo y no realiza ninguna acción para el empleado. • Retorna a la vista de horas extra.
Flujo alternativo error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Registro en la tabla de LogEvento por usuario.</p> <p>Auditoría extendida con trazabilidad de eventos.</p> <p>El sistema debe garantizar que se realice un borrado lógico y no físico para preservar la trazabilidad histórica.</p>	
Postcondiciones	
<p>Se elimina el registro de horas extra de la persona y queda marcado como inactivo en el sistema.</p> <p>Se conserva toda la información histórica de la asistencia del empleado.</p> <p>El registro de la acción queda almacenado en “auth_change” y “log_evento”.</p> <p>El sistema refleja los cambios en la lista de asistencia actual.</p>	

Fuente: Elaboración propia, 2025.

Tabla 34

Caso de uso para listar permisos de los empleados

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MP-01	Nombre del caso de uso: Listar los registros de los permisos de los empleados
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH observar los datos activos en el sistema
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario administrativo de RRHH Sistema de Gestión de Recursos Humanos

Precondiciones:	<p>El usuario debe haber iniciado la sesión con un rol autorizado.</p> <p>Tener los permisos correspondientes para realizar la acción.</p> <p>Acceso a la base de datos.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para listar todos los empleados activos en el sistema con los registros de permisos pendientes, aprobados y rechazados de un empleado, fecha de inicio del permiso, fecha de finalización del permiso y motivo. Además de contar con los botones necesarios para realizar las acciones editar, eliminar, agregar y el botón Regresar a la pantalla principal del sistema, para que el usuario pueda realizar acciones.</p>	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Permisos del módulo de permisos. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra datos correspondientes a los permisos como nombre del empleado, fecha de inicio del permiso, fecha final del permiso, el estado y el motivo. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.
Flujo alternativo botón Editar	<ul style="list-style-type: none"> • El usuario selecciona el botón Editar. • El sistema redirecciona a la vista Actualizar Datos.
Flujo alternativo botón Eliminar	<ul style="list-style-type: none"> • El usuario selecciona el botón Eliminar. • El sistema muestra una ventana de confirmación.
Flujo alternativo botón Agregar Asistencia	<ul style="list-style-type: none"> • El usuario selecciona el botón Agregar. • El sistema redirecciona a la vista Crear un Registro de Permisos.
Flujo alternativo botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar. • El sistema redirecciona a la vista principal del sistema.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error.

	<ul style="list-style-type: none"> El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de empleado activo. Auditoría extendida con trazabilidad.	
Postcondiciones	
Acceso a los registros de permisos correspondiente a cada empleado.	

Fuente: Elaboración propia, 2025.

Tabla 35

Caso de uso agregar registro de permisos solicitados

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas, S.A, ubicada en Grecia	
Número de caso de Uso: CU-MP-02	Nombre del caso de uso: Agregar un registro de permiso de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH agregar un nuevo registro de permiso de un empleado.
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Datos del empleado previamente registrados.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para agregar un nuevo registro de permiso de un empleado en el sistema, con los datos como tipo de permiso, la fecha de inicio del permiso, la fecha de finalización del permiso, el estado y el motivo.	
<ol style="list-style-type: none"> El usuario accede a la opción permisos del módulo de permisos. El sistema muestra la vista con todos los empleados activos en el sistema. El sistema muestra datos del día actual como nombre del empleado, fecha de inicio del permiso, fecha de finalización del permiso, tipo de permiso, estado y motivo. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. El usuario da clic al botón Agregar en la persona correspondiente. 	

<p>6. El sistema redirecciona a la vista de agregar permiso y despliega el formulario.</p> <p>7. El usuario completa los datos requeridos del empleado, como fecha de inicio, fecha de finalización del permiso, tipo de permiso, estado y motivo.</p> <p>8. El sistema valida todos los campos.</p> <p>9. El usuario selecciona la opción Guardar.</p> <p>10. El sistema almacena la información en la base de datos.</p> <p>11. El sistema confirma el registro exitoso.</p> <p>12. El sistema sale del formulario y regresa a la vista principal del módulo de permisos.</p> <p>13. Finaliza el caso de uso exitosamente.</p>	
Flujos alternos	
Flujo alternativo Validación de fecha no pasada	<ul style="list-style-type: none"> • Se verifica que la fecha no sea pasada al día presente. • El sistema bloquea los días pasados.
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha futura, máximo de horas diarias, de horario y de campos requeridos.	
Auditoría extendida con trazabilidad de eventos.	
Postcondiciones	
La condición de las horas extra del empleado queda registrada.	
El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 36*Caso de uso para actualizar un registro de permisos*

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MHE-03	Nombre del caso de uso: Actualizar los datos de un permiso
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH actualizar datos del registro de un permiso del empleado en el sistema, con toda su información correspondiente y sus validaciones.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo RRHH Sistema de gestión de recursos humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe tener los permisos necesarios para realizar la acción. El empleado tiene que estar registrado previamente en el sistema. El registro debe estar creado previamente. Acceso a los datos del empleado en la base de datos.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para actualizar los datos de un registro de permiso laboral de un empleado, incluidos las validaciones de consistencia y los botones Guardar y Cancelar.	
<ol style="list-style-type: none"> 1. El usuario selecciona permisos del módulo de permisos. 2. El sistema muestra la pantalla de permisos con todos los empleados activos del sistema. 3. El sistema muestra datos como nombre del empleado, fecha de inicio de los permisos, fecha de finalización de los permisos, tipo de permiso, estado y motivo. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón Editar. 6. El sistema redirecciona a la vista Actualizar Permiso y muestra el formulario con los datos previamente registrados cargados. 7. El usuario actualiza los registros correspondientes. 8. El sistema valida todos los campos. 9. El usuario selecciona la opción Guardar. 10. El sistema almacena la información en la base de datos. 	

<p>11. El sistema confirma el registro exitoso.</p> <p>12. El sistema sale del formulario y regresa a la vista principal del módulo de horas extra.</p> <p>13. Finaliza el caso de uso exitosamente.</p>	
Flujos alternos	
Flujo alternativo Validación de fecha no pasada	<ul style="list-style-type: none"> • Se verifica que la fecha no sea pasada al día presente. • El sistema bloquea los días pasados.
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha pasada y de campos requeridos.	
Auditoría extendida con trazabilidad de eventos.	
Postcondiciones	
Se actualizan los datos del permiso de un empleado seleccionado.	
El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 37

Caso de uso para eliminar un registro de permisos

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MP-04	Nombre del caso de uso: Eliminar un registro de permiso de un empleado
Fecha de elaboración:	10/09/2025

Descripción del caso de uso:	Le permite al encargado de RRHH eliminar el registro del permiso de un empleado en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe contar con permisos de eliminación. El empleado tiene que estar registrado previamente en el sistema y estar activo. Se tiene que haber creado previamente el registro de asistencia.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para eliminar de forma lógica el registro de un permiso correspondiente a un empleado, para asegurar que los datos históricos se conserven para fines de auditoría, y mantengan la trazabilidad.	
<ol style="list-style-type: none"> 1. El usuario selecciona permisos del módulo de permisos. 2. El sistema muestra la pantalla de todos los empleados activos del sistema. 3. El sistema muestra datos del día actual, como nombre del empleado, fecha de inicio, fecha de finalización, tipo de permiso, estado y motivo. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón Eliminar del empleado del que requiere eliminar el registro. 6. El usuario confirma la eliminación. 7. El sistema actualiza el estado del empleado en la base de datos, actualizado como inactivo (activo = false). 8. El sistema registra la acción en las tablas de “audit_change” y “log_evento”. 9. El sistema regresa a la vista principal de Permisos. 10. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de permisos	<ul style="list-style-type: none"> • El sistema valida si el usuario tiene los permisos correspondientes para realizar la acción. • Si el usuario no tiene los permisos correspondientes el botón eliminar se encuentra desactivado.
Flujo alternativo Validación de confirmación de eliminación	<ul style="list-style-type: none"> • El sistema muestra un mensaje de confirmación para eliminar al empleado. • Si se confirma la eliminación se realiza esta.

	<ul style="list-style-type: none"> • El sistema regresa a la vista principal de permisos.
Flujo alternativo Cancelar la operación	<ul style="list-style-type: none"> • El usuario desea cancelar la acción. • En la ventana de conformación selecciona la opción No. • El sistema cierra el cuadro de diálogo y no realiza ninguna acción para el empleado. • Retorna a la vista de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Registro en la tabla de “LogEvento” por usuario. Auditoría extendida con trazabilidad de eventos. El sistema debe garantizar que se realice un borrado lógico y no físico para preservar la trazabilidad histórica.</p>	
Postcondiciones	
<p>Se elimina el registro de horas de permiso de la persona y queda marcado como inactivo en el sistema. Se conserva toda la información histórica de la asistencia del empleado. El registro de la acción queda almacenado en “auth_change” y “log_evento”. El sistema refleja los cambios en la lista de permisos actual.</p>	

Fuente: Elaboración propia, 2025.

Tabla 38

Caso de uso para listar el registro de incapacidades

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MI-01	Nombre del caso de uso: Listar los registros de incapacidades de los empleados
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH observar todas las incapacidades activas por empleado en el sistema
Autor del caso de uso:	Andrés Gerardo González Villegas

Actores relacionados:	Usuario administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. Tener los permisos correspondientes para realizar la acción. Acceso a la base de datos.
Flujo básico del caso de uso	
Este caso de uso describe el proceso completo para listar todos los registros activos de incapacidades del empleado, con datos como nombre del empleado, fecha de inicio de la incapacidad, fecha de finalización de la incapacidad, tipo de motivo y cantidad de días de incapacidad; además de contar con los botones necesarios para realizar las acciones de editar, eliminar, agregar, y el botón de regresar a la pantalla principal del sistema para que el usuario pueda realizar acciones.	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Incapacidades del módulo de incapacidades. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra datos correspondientes a los permisos, como nombre del empleado, fecha de inicio de la incapacidad, fecha final de la incapacidad, el tipo y la cantidad de días de incapacidad. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.
Flujo alternativo botón Editar	<ul style="list-style-type: none"> • El usuario selecciona el botón Editar. • El sistema redirecciona a la vista de Actualizar Datos.
Flujo alternativo botón Eliminar	<ul style="list-style-type: none"> • El usuario selecciona el botón Eliminar. • El sistema muestra una ventana de confirmación.
Flujo alternativo botón Agregar Asistencia	<ul style="list-style-type: none"> • El usuario selecciona el botón Agregar. • El sistema redirecciona a la vista Crear un Registro de Incapacidad.
Flujo alternativo botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar. • El sistema redirecciona a la vista principal del sistema.

Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de empleado activo. Auditoría extendida con trazabilidad.	
Postcondiciones	
El usuario tiene acceso a la información correspondiente de cada empleado con respecto a las incapacidades.	

Fuente: Elaboración propia, 2025.

Tabla 39

Caso de uso para agregar registro de incapacidades

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MI-02	Nombre del caso de uso: Agregar un registro de incapacidad de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH agregar un nuevo registro del permiso de un empleado.
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Datos del empleado previamente registrados.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para agregar un nuevo registro de la incapacidad de un empleado en el sistema, con los datos como tipo de incapacidad, fecha de inicio de la incapacidad, fecha de finalización y cantidad de días de incapacidad.	

<ol style="list-style-type: none"> 1. El usuario accede a la opción Incapacidades del módulo de incapacidades. 2. El sistema muestra la vista con todos los empleados activos en el sistema. 3. El sistema muestra datos como nombre del empleado, fecha de inicio de la incapacidad, fecha de finalización de la incapacidad, tipo de incapacidad, y la cantidad de días de incapacidad. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario da clic al botón Agregar en la persona correspondiente. 6. El sistema redirecciona a la vista de Agregar incapacidad y despliega el formulario. 7. El usuario completa los datos requeridos del empleado, como fecha de inicio, fecha de finalización, tipo de incapacidad, estado y motivo. 8. El sistema valida todos los campos. 9. El usuario selecciona la opción Guardar. 10. El sistema almacena la información en la base de datos. 11. El sistema confirma el registro exitoso. 12. El sistema sale del formulario y regresa a la vista principal del módulo de incapacidades. 13. Finaliza el caso de uso exitosamente. 	
Flujos Alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha futura, máximo de horas diarias, de horario y de campos requeridos. Auditoría extendida con trazabilidad de eventos.	
Postcondiciones	
La incapacidad del empleado queda registrada. El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 40

Caso de uso para actualizar el registro de incapacidades

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MI-03	Nombre del caso de uso: Actualizar los datos de una incapacidad
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH actualizar datos del registro de incapacidad del empleado en el sistema, con toda su información correspondiente y sus validaciones.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe tener los permisos necesarios para realizar la acción. El empleado tiene que estar registrado previamente en el sistema. El registro debe estar creado previamente. Acceso a los datos del empleado en la base de datos.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para actualizar los datos de un registro de incapacidad de un empleado, incluidas las validaciones de consistencia y los botones Guardar y Cancelar.	
<ol style="list-style-type: none"> 1. El usuario selecciona incapacidades del módulo de incapacidad. 2. El sistema muestra la pantalla de incapacidades con todos los empleados activos del sistema. 3. El sistema muestra datos como nombre del empleado, fecha de inicio de incapacidad, fecha de finalización de la incapacidad, tipo de incapacidad y cantidad de días de incapacidad. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón Editar. 6. El sistema redirecciona a la vista Actualizar permiso y muestra el formulario con los datos previamente registrados cargados. 7. El usuario actualiza los registros correspondientes. 8. El sistema valida todos los campos. 	

<p>9. El usuario selecciona la opción Guardar.</p> <p>10. El sistema almacena la información con la base de datos.</p> <p>11. El sistema confirma el registro exitoso.</p> <p>12. El sistema sale del formulario y regresa a la vista principal del módulo incapacidades.</p> <p>13. Finaliza el caso de uso exitosamente.</p>	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Si no cumple se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha pasada y de campos requeridos.	
Auditoría extendida con trazabilidad de eventos.	
Postcondiciones	
Se actualizan los datos de la incapacidad de un empleado seleccionado.	
El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 41

Caso de uso *Eliminar un registro de incapacidades*

Prototipo funcional para la gestión de Recursos Humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MI-04	Nombre del caso de uso: Eliminar un registro de incapacidad de un empleado
Fecha elaboración:	10/09/2025
-----Descripción Caso de Uso:	Permite al encargado de RRHH eliminar un registro de incapacidad de un empleado en el sistema

Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe contar con permisos de eliminación. El empleado tiene que estar registrado previamente en el sistema y estar activo. Se tiene que haber creado previamente el registro de asistencia.
Flujo básico del caso de uso	
Este caso de uso describe el proceso completo para eliminar de forma lógica un registro de incapacidad correspondiente a un empleado, para asegurar que los datos históricos se conserven para fines de auditoría manteniendo la trazabilidad.	
<ol style="list-style-type: none"> 1. El usuario selecciona Incapacidad del módulo de incapacidades. 2. El sistema muestra la pantalla de Incapacidad con todos los empleados activos del sistema. 3. El sistema muestra datos como nombre del empleado, fecha de inicio, fecha de finalización, tipo de incapacidad, tipo y cantidad de días. 4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar. 5. El usuario selecciona el botón de Eliminar del empleado que requiere eliminar el registro. 6. El usuario confirma la eliminación. 7. El sistema actualiza el estado del empleado en la base de datos, actualizándolo como inactivo (activo = false). 8. El sistema registra la acción en las tablas de “audit_change” y “log_evento”. 9. El sistema regresa a la vista principal de incapacidades. 10. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de permisos	<ul style="list-style-type: none"> • El sistema valida si el usuario tiene los permisos correspondientes para realizar la acción. • Si el usuario no tiene los permisos correspondientes, el botón de Eliminar se encuentra desactivado.
Flujo alternativo Validación de confirmación de eliminación	<ul style="list-style-type: none"> • El sistema muestra de un mensaje de confirmación para eliminar al empleado. • Si se confirma la eliminación, se realiza la acción de eliminación.

	<ul style="list-style-type: none"> • El sistema regresa a la vista principal de permisos.
Flujo Alternativo Cancelar la operación	<ul style="list-style-type: none"> • El usuario desea cancelar la acción. • En la ventana de conformación selecciona la opción No. • El sistema cierra el cuadro de dialogo y no realiza ninguna acción para el empleado. • Retorna a la vista de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Registro en la tabla de LogEvento por usuario.</p> <p>Auditoría extendida con trazabilidad de eventos.</p> <p>El sistema debe garantizar que se realice un borrado lógico y no físico para preservar la trazabilidad histórica.</p>	
Postcondiciones	
<p>Se elimina el registro de incapacidad de la persona y queda marcado como inactivo en el sistema.</p> <p>Se conserva toda la información histórica de incapacidades del empleado.</p> <p>El registro de la acción queda almacenado en “auth_change” y “log_evento”.</p> <p>El sistema refleja los cambios en la lista de incapacidades.</p>	

Fuente: Elaboración propia, 2025.

Tabla 42

Caso de uso listar registros de liquidaciones

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones S.A, ubicada en Grecia	
Número de caso de Uso: CU-ML-01	Nombre del caso de uso: Listar los registros de liquidaciones de los empleados
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Permite al encargado de RRHH observar las liquidaciones de empleados en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas

Actores relacionados:	<p>Usuario administrativo de RRHH</p> <p>Sistema de gestión de Recursos Humanos</p>
Precondiciones:	<p>El usuario debe haber iniciado sesión con un rol autorizado.</p> <p>Tener los permisos correspondientes para realizar la acción.</p> <p>Acceso a la base de datos.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para listar todos los empleados activos en el sistema con los registros de liquidaciones de un empleado, con datos como nombre del empleado, fecha de ingreso a la empresa, fecha de salida, fecha en la que se realizó la liquidación, tipo, motivo, monto por pagar de la liquidación y sus detalles. Además de contar con los botones necesarios para realizar las acciones de editar, eliminar, agregar, y el botón de regresar a la pantalla principal del sistema para que el usuario pueda realizar acciones.</p>	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Liquidaciones del módulo de liquidaciones. 2. El sistema muestra la vista con todos los empleados con liquidaciones en el sistema. 3. El sistema muestra datos correspondientes a las liquidaciones realizadas, como nombre del empleado, fecha de ingreso, fecha de finalización laboral; el tipo, el motivo y los detalles. 4. El usuario puede elegir acciones disponibles de editar, eliminar, agregar y regresar. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.
Flujo alternativo Botón Editar	<ul style="list-style-type: none"> • El usuario selecciona el botón Editar. • El sistema redirecciona a la vista de Actualizar datos.
Flujo alternativo Botón Eliminar	<ul style="list-style-type: none"> • El usuario selecciona el botón Eliminar. • El sistema muestra una ventana de confirmación.
Flujo alternativo Botón Agregar Asistencia	<ul style="list-style-type: none"> • El usuario selecciona el botón Agregar. • El sistema redirecciona a la vista Crear un registro de liquidación.
Flujo alternativo botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar. • El sistema redirecciona a la vista principal del sistema.

Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Registro de auditoria con trazabilidad.	
Postcondiciones	
El usuario visualiza la información histórica y puede acceder a otras acciones relacionadas con ella.	

Fuente: Elaboración propia, 2025.

Tabla 43

Caso de uso agregar un registro de liquidación

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-ML-02	Nombre del caso de uso: Agregar un registro de liquidación de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH agregar un nuevo registro de liquidación de un empleado por finalización de contrato.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Datos del empleado previamente registrados. Registros de base de datos de nómina, empleado, vacaciones, aguinaldo u horas extras.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para agregar un nuevo registro de liquidación de un empleado en el sistema, con los datos como nombre del empleado, fecha de ingreso a la empresa, fecha de finalización, fecha en la que se realiza la liquidación, tipo de liquidación, motivo, monto total que se paga y detalles.	
1. El usuario accede al módulo de liquidaciones.	

<ol style="list-style-type: none"> 2. Selecciona la opción Agregar liquidación. 3. El sistema despliega el formulario de nueva liquidación. 4. El usuario completa los datos requeridos del empleado, como fecha de inicio, fecha de finalización, tipo de liquidación, estado y motivo. 5. El sistema valida todos los campos obligatorios y la coherencia de fechas. 6. El sistema calcula automáticamente los montos correspondientes (aguinaldo, vacaciones, cesantía, salarios promediados). 7. El usuario selecciona la opción Guardar. 8. El sistema almacena la información en la base de datos. 9. El sistema confirma el registro exitoso. 10. El sistema regresa a la vista principal del módulo de liquidaciones. 11. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fechas futuras	<ul style="list-style-type: none"> • Se valida que la fecha de salida y la fecha en la que se realiza la liquidación no sea futura. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fecha de ingreso	<ul style="list-style-type: none"> • Se valida que la fecha de finalización sea posterior a la fecha de ingreso. • Si no cumple se muestra un mensaje de error. • Y se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.

Requerimientos especiales
Validación de fecha futura y máximo de horas diarias, de horario y de campos requeridos. Auditoria extendida con trazabilidad.
Postcondiciones
El registro de liquidación queda guardado en el sistema. Queda almacenado el detalle de cálculos realizados (aguinaldo, vacaciones, cesantía, etc). El evento queda registrado en el sistema de trazabilidad.

Fuente: Elaboración propia, 2025.

Tabla 44

Caso de uso Actualizar el registro de liquidación

Prototipo funcional para la Gestión de Recursos Humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-ML-03	Nombre del caso de uso: Actualizar un registro de liquidación de un empleado
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH actualizar un registro existente de liquidación de un empleado por finalización de contrato.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Datos del empleado previamente registrados. El registro debe estar creado previamente. Registros de base de datos de nómina, empleado, horas extras, vacaciones.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para actualizar un registro de liquidación de un empleado en el sistema, esto con los datos como nombre del empleado, la fecha de ingreso a la empresa, la fecha de finalización, la fecha en la que se realiza la liquidación; tipo de liquidación, motivo, monto total que se paga y detalles.	
<ol style="list-style-type: none"> 1. El usuario accede al módulo de liquidaciones. 2. El usuario selecciona un registro de liquidación y pulsa la opción Editar. 	

<ol style="list-style-type: none"> 3. El sistema despliega el formulario con los datos de la liquidación existente. 4. El usuario modifica los campos necesarios como fecha de inicio, fecha de finalización, tipo de liquidación, estado y motivo. 5. El sistema valida todos los campos requeridos. 6. El sistema recalcula automáticamente los montos (aguinaldo, vacaciones, cesantía, salario promedio). 7. El usuario selecciona la opción Guardar. 8. El sistema almacena la información en la base de datos. 9. El sistema confirma el registro exitoso. 10. El sistema retorna a la vista principal del módulo de liquidaciones. 11. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fechas futuras	<ul style="list-style-type: none"> • Se valida que la fecha de salida y la fecha en la que se realiza la liquidación no sea futura. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fecha de ingreso	<ul style="list-style-type: none"> • Se valida que la fecha de finalización sea posterior a la fecha de ingreso. • Si no cumple se muestra un mensaje de error. • Y se rechaza el envío.
Flujo alternativo Error de cálculos automáticos	<ul style="list-style-type: none"> • Si al realizar un cálculo automático (cálculo de salarios promedio por mes y día de los últimos 6 meses, cálculo de vacaciones y cálculo de aguinaldos) falla. • El sistema muestra un mensaje de error. • Y no se modifica ningún dato en la base de datos.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información.

	<ul style="list-style-type: none"> • Y regresa a la pantalla principal de permisos.
Flujo alternativo error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha futura, máximo de horas diarias, de horario y de campos requeridos. Los cálculos de aguinaldo, vacaciones y cesantía se ajustan a la nueva información. El evento de actualización queda trazado en el sistema.	
Postcondiciones	
El registro de liquidación queda actualizado. Los cálculos de aguinaldo, vacaciones y cesantía se ajustan a la nueva información. El evento de actualización queda trazado en el sistema.	

Fuente: Elaboración propia, 2025.

Tabla 45

Caso de uso Eliminar registro de liquidación

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones S.A, ubicada en Grecia	
Número de caso de uso: CU-ML-04	Nombre del caso de uso: Eliminar el registro de liquidaciones de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH eliminar el registro de liquidaciones de un empleado en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe contar con permisos de eliminación. El empleado tiene que estar registrado previamente en el sistema y estar activo. Se tiene que haber creado previamente el registro de asistencia.
Flujo básico del caso de uso	

En este caso de uso se describe el proceso completo para eliminar de forma lógica un registro de liquidación correspondiente a un empleado, para asegurar que los datos históricos se conserven para fines de auditoría y mantenga la trazabilidad.

1. El usuario selecciona Liquidaciones del módulo de liquidaciones.
2. El sistema muestra la pantalla de liquidaciones con todos los empleados activos del sistema.
3. El sistema muestra datos como nombre del empleado, fecha de ingreso, fecha de salida de la empresa, fecha en la que se realiza la liquidación, tipo de liquidación, motivo, monto total de la liquidación y detalles.
4. La vista cuenta con los botones Editar, Eliminar, Agregar y Regresar.
5. El usuario selecciona el botón Eliminar del empleado que requiere eliminar el registro.
6. El usuario confirma la eliminación.
7. El sistema actualiza el estado del registro de liquidación en la base de datos, marcado como inactivo (activo = false).
8. El sistema registra la acción en las tablas de “audit_change” y “log_evento”.
9. El sistema regresa a la vista de principal de Liquidaciones.
10. Finaliza el caso de uso exitosamente.

Flujos alternos

Flujo Alternativo Validación de permisos	<ul style="list-style-type: none"> • El sistema valida si el usuario tiene los permisos correspondientes para realizar la acción. • Si el usuario no tiene los permisos correspondientes el botón de eliminar se encuentra desactivado.
Flujo alternativo Validación de confirmación de eliminación	<ul style="list-style-type: none"> • El sistema muestra un mensaje de conformación para eliminar al empleado. • Si se confirma la eliminación se realiza la acción de eliminación. • El sistema regresa a la vista principal de Liquidaciones.
Flujo alternativo Cancelar la operación	<ul style="list-style-type: none"> • El usuario desea cancelar la acción. • En la ventana de conformación se selecciona la opción No. • El sistema cierra el cuadro de dialogo y no realiza ninguna acción sobre el empleado. • Retorna a la vista de permisos.

Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Registro en la tabla de LogEvento por usuario.</p> <p>Auditoria extendida con trazabilidad de eventos.</p> <p>El sistema debe garantizar que se realice un borrado lógico y no físico para preservar la trazabilidad histórica.</p>	
Postcondiciones	
<p>El registro de liquidación queda marcado como inactivo en el sistema.</p> <p>Se conserva toda la información histórica de liquidaciones de un empleado.</p> <p>El registro de la acción queda almacenado en “audit_change” y “log_evento”.</p> <p>Se conserva toda la información histórica de liquidaciones del empleado.</p> <p>El sistema refleja los cambios en la lista de liquidaciones.</p>	

Fuente: Elaboración propia, 2025.

Tabla 46

Caso de uso cantidad de días disponibles de vacaciones

Prototipo Funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MV-01	Nombre del caso de uso: Listar los registros de los días disponibles de vacaciones de los empleados
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH observar la cantidad de días disponibles de vacaciones que tienen los empleados en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	<p>El usuario debe haber iniciado sesión con un rol autorizado.</p> <p>Tener los permisos correspondientes para realizar la acción.</p> <p>Acceso a la base de datos.</p>
Flujo básico del caso de uso	

<p>En este caso de uso se describe el proceso completo para listar los días disponibles de vacaciones que los empleados tienen a favor para disfrutar, con datos como nombre del empleado, fecha de solicitud de los días, fecha de inicio de las vacaciones, fecha en la que finalizan las vacaciones, cantidad de días solicitados, el estado de la solicitud (pendiente, aprobada o rechazada) y las observaciones, de ser necesarias. Además de contar con los botones necesarios para realizar las acciones de editar, eliminar, agregar y el botón de regresar a la pantalla principal del sistema.</p>	
<ol style="list-style-type: none"> 1. El usuario entra a la opción Vacaciones del módulo de vacaciones. 2. El sistema muestra la vista con todos los empleados del sistema y con sus días de vacaciones disponibles. 3. El sistema muestra el nombre de los empleados con su cantidad de días disponibles de disfrute de vacaciones. 4. El usuario puede elegir acciones disponibles de solicitudes de vacaciones y regresar. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.
Flujo alternativo botón Solicitud de vacaciones	<ul style="list-style-type: none"> • El usuario selecciona el botón Solicitud de vacaciones en el registro de la persona que solicita los días. • El sistema redirecciona a la vista con un formulario para ingresar los datos necesarios para la solicitud.
Flujo alternativo botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar. • El sistema redirecciona a la vista principal del sistema.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Registro en la tabla de LogEvento por usuario. Auditoría extendida con trazabilidad de eventos.</p>	
Postcondiciones	

El usuario visualiza la información actualizada de los días de vacaciones disponibles y puede acceder a otras acciones relacionadas con ella.

Fuente: Elaboración propia, 2025.

Tabla 47

Caso de uso listar solicitudes de vacaciones

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MV-02	Nombre del caso de uso: Listar los registros de solicitudes de vacaciones de los empleados
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH observar las solicitudes de vacaciones de empleados en el sistema
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. Tener los permisos correspondientes para realizar la acción. Acceso a la base de datos.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para listar las solicitudes de vacaciones de un empleado en el sistema, con registros como nombre del empleado, fecha de solicitud del día de vacaciones, fecha de inicio de las vacaciones, fecha de finalización de las vacaciones, cantidad de días, estado de la solicitud (pendiente, aprobada y rechazada) y observaciones. Además de contar con los botones necesarios para realizar las acciones de editar, eliminar, agregar y el botón de regresar a la pantalla principal del módulo de vacaciones.	
<ol style="list-style-type: none"> 1. El usuario accede a la opción Solicitud de vacaciones. 2. El sistema muestra la vista del empleado seleccionado. 3. El sistema muestra una lista de los registros de solicitudes de vacaciones y su estado. 4. El usuario puede elegir acciones disponibles de editar, eliminar, agregar y regresar. 5. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de empleado activo	<ul style="list-style-type: none"> • El sistema consulta si el empleado está activo en el sistema. • Si existe muestra la información del empleado.

Flujo alternativo del botón Editar	<ul style="list-style-type: none"> • El usuario selecciona el botón Editar. • El sistema redirecciona a la vista de Actualizar datos.
Flujo alternativo del botón Eliminar	<ul style="list-style-type: none"> • El usuario selecciona el botón Eliminar. • El sistema muestra una ventana de confirmación.
Flujo alternativo del botón Agregar Solicitud de vacaciones	<ul style="list-style-type: none"> • El usuario selecciona el botón Agregar. • El sistema redirecciona a la vista Crear solicitud de vacaciones.
Flujo alternativo Botón Regresar	<ul style="list-style-type: none"> • El usuario selecciona el botón Regresar. • El sistema redirecciona a la vista principal del sistema.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Registro en la tabla de LogEvento por usuario. Auditoría extendida con trazabilidad de eventos.	
Postcondiciones	
El usuario visualiza la información del historial de solicitudes de vacaciones y puede acceder a otras acciones relacionadas con ella.	

Fuente: Elaboración propia, 2025.

Tabla 48

Caso de uso Agregar registro de vacaciones

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MV-03	Nombre del caso de uso: Agregar un registro de solicitud de vacaciones de un empleado
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH agregar un nuevo registro de solicitud de vacaciones de un empleado.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos

Precondiciones:	<p>El usuario debe haber iniciado sesión con un rol autorizado.</p> <p>El usuario debe tener los permisos correspondientes para realizar la acción.</p> <p>Datos del empleado previamente registrados.</p> <p>El empleado tener días disponibles por solicitar.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para agregar un nuevo registro de solicitud de vacaciones de un empleado en el sistema, con los datos como nombre del empleado, la fecha de solicitud de las vacaciones, la fecha de inicio de vacaciones, la fecha en que finalizan las vacaciones, cantidad de días solicitados, estado (aprobado, pendiente o rechazado) y observaciones.</p>	
<ol style="list-style-type: none"> 1. Selecciona la opción Agregar solicitud. 2. El sistema despliega el formulario de agregar solicitud de vacaciones. 3. El usuario completa los datos requeridos del empleado, la fecha de inicio de vacaciones, la fecha en que finalizan las vacaciones, la cantidad de días solicitados, el estado (aprobado, pendiente o rechazado) y las observaciones. 4. El sistema valida todos los campos obligatorios y la coherencia de fechas. 5. El usuario selecciona la opción Guardar. 6. El sistema almacena la información en la base de datos. 7. El sistema confirma el registro exitoso. 8. El sistema regresa a la vista principal del historial de solicitudes de vacaciones del empleado. 9. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fecha de inicio	<ul style="list-style-type: none"> • Se valida que la fecha de finalización sea posterior a la fecha de inicio de vacaciones. • Si no cumple se muestra un mensaje de error. • Y se rechaza el envío.
Flujo alternativo Validación de días disponibles	<ul style="list-style-type: none"> • El sistema valida que la solicitud de días solicitados no sea mayor a la cantidad de días disponibles.

	<ul style="list-style-type: none"> • Si no cumple el sistema muestra una advertencia. • El sistema bloque el envío.
Flujo alternativo cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha futura, máximo de días disponibles y de campos requeridos. Auditoría extendida con trazabilidad.	
Postcondiciones	
El registro de solicitud de vacaciones queda guardado en el sistema. El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 49

Caso de uso Actualizar registro de solicitud de vacaciones

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MV-04	Nombre del caso de uso: Actualizar un registro de solicitud de vacaciones de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH actualizar un registro existente de una solicitud de vacaciones de un empleado.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Datos del empleado previamente registrados.

	<p>El registro debe estar creado previamente.</p> <p>Registros de base de datos de cantidad de días disponibles.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para actualizar un registro de solicitud de vacaciones de un empleado en el sistema, la fecha de inicio de vacaciones, la fecha de finalización, la cantidad de días solicitados, el estado de la solicitud y las observaciones.</p>	
<ol style="list-style-type: none"> 1. Selecciona la opción Editar solicitud. 2. El sistema despliega el formulario de editar la solicitud de vacaciones. 3. El usuario completa los datos requeridos del empleado, la fecha de inicio de vacaciones, la fecha en la que finaliza las vacaciones, la cantidad de días solicitados, el estado (aprobado, pendiente o rechazado) y las observaciones. 4. El sistema valida todos los campos obligatorios y la coherencia de fechas. 5. El sistema valida todos los campos requeridos. 6. El usuario selecciona la opción Actualizar. 7. El sistema almacena la información en la base de datos. 8. El sistema confirma el registro exitoso. 9. El sistema retorna a la vista principal del historial de solicitudes de vacaciones del empleado. 10. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fecha de inicio	<ul style="list-style-type: none"> • Se valida que la fecha de finalización sea posterior a la fecha de inicio de vacaciones. • Si no cumple se muestra un mensaje de error. • Y se rechaza el envío.
Flujo alternativo Validación de días disponibles	<ul style="list-style-type: none"> • El sistema valida que la solicitud de días solicitados no sea mayor a la cantidad de días disponibles. • Si no cumple el sistema muestra una advertencia. • El sistema bloque el envío.

Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Validación de fecha futura, máximo de días disponibles y de campos requeridos. Auditoría extendida con trazabilidad.	
Postcondiciones	
El registro de solicitud de vacaciones queda actualizado en el sistema. El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 50

Caso de uso para eliminar registro de vacaciones

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-MV-05	Nombre del caso de uso: Eliminar el registro de una solicitud de vacaciones de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH eliminar un registro de solicitud de vacaciones de un empleado en el sistema
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario administrativo RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión. El usuario debe contar con permisos de eliminación. El empleado tiene que estar registrado previamente en el sistema y estar activo. Se tiene que haber creado previamente el registro de asistencia.
Flujo básico del caso de uso	

<p>En este caso de uso se describe el proceso completo para eliminar de forma lógica un registro de solicitud de vacaciones correspondiente a un empleado, para asegurar que los datos históricos se conserven para fines de auditoría pero manteniendo la trazabilidad.</p>	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón Eliminar del empleado que requiere eliminar del registro. 2. El usuario confirma la eliminación. 3. El sistema actualiza el estado del registro de liquidación en la base de datos y lo marca como inactivo (activo = false). 4. El sistema registra la acción en las tablas de “audit_change” y “log_evento”. 5. El sistema regresa a la vista de principal de historial de solicitudes de vacaciones del empleado. 6. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de permisos	<ul style="list-style-type: none"> • El sistema valida si el usuario tiene los permisos correspondientes para realizar la acción. • Si el usuario no tiene los permisos correspondientes el botón de eliminar se encuentra desactivado.
Flujo alternativo Validación de confirmación de eliminación	<ul style="list-style-type: none"> • El sistema muestra un mensaje de conformación para eliminar al empleado. • Si se confirma la eliminación se realiza la acción de eliminación. • El sistema regresa a la vista principal de liquidaciones.
Flujo alternativo Cancelar la operación	<ul style="list-style-type: none"> • El usuario desea cancelar la acción. • En la ventana de conformación selecciona la opción No. • El sistema cierra el cuadro de dialogo y no realiza ninguna acción sobre el empleado. • Retorna a la vista de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Registro en la tabla de LogEvento por usuario.	

Auditoría extendida con trazabilidad de eventos. El sistema debe garantizar que se realice un borrado lógico y no físico para preservar la trazabilidad histórica.
Postcondiciones
El registro de solicitud de vacaciones queda marcado como inactivo en el sistema. Se conserva toda la información histórica de solicitud de vacaciones de un empleado. El registro de la acción queda almacenado en “audit_change” y “log_evento”. Se conserva toda la información histórica de vacaciones del empleado. El sistema refleja los cambios en la lista de liquidaciones.

Fuente: Elaboración propia, 2025.

Tabla 51

Caso de uso del cálculo del aguinaldo

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número Caso de Uso: CU-CAG-01	Nombre del caso de uso: Realizar el cálculo del aguinaldo de los empleados
Fecha elaboración:	10/09/2025
Descripción caso de uso:	Le permite al encargado de RRHH realizar el cálculo del aguinaldo de cada uno de los empleados.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Registros de base de datos de nómina, empleado, horas extras.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar el cálculo de aguinaldos de los empleados, con los datos como nombre del empleado, año, la fecha de inicio del periodo, la fecha de finalización del periodo, el monto calculado por pagar o promedio salarial entre ese rango de periodos, fecha de creación y observaciones.	
<ol style="list-style-type: none"> 1. El usuario accede al módulo de aguinaldos. 2. El usuario selecciona la opción Calcular aguinaldo. 3. El sistema despliega la vista con todos los empleados activos. 	

<ol style="list-style-type: none"> 4. El usuario selecciona al empleado para realizar el cálculo. 5. El sistema despliega la vista con los campos necesarios para realizar el cálculo. 6. El usuario completa los datos requeridos del empleado, como fecha de inicio del periodo que va a calcular y la fecha de finalización del periodo por calcular. 7. El sistema valida todos los campos obligatorios y la coherencia de fechas. 8. El usuario selecciona la opción Calcular. 9. El sistema calcula automáticamente el monto del aguinaldo usando la fórmula: $\text{aguinaldo} = \text{suma de salarios_brutos} / 12$. 10. Se muestran el desglose de los salarios y el monto calculado. 11. El usuario selecciona la opción Guardar. 12. El sistema almacena la información en la base de datos. 13. El sistema confirma el registro exitoso. 14. El sistema visualiza los datos del registro con el nombre y el monto por pagar. 15. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fechas futuras	<ul style="list-style-type: none"> • Se valida que la fecha en la que se realiza el cálculo del aguinaldo no sea futura. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de fecha de ingreso	<ul style="list-style-type: none"> • Se valida que la fecha de finalización sea posterior a la fecha de ingreso. • Si no cumple se muestra un mensaje de error. • Y se rechaza el envío.
Flujo Alternativo Fecha de inicio del periodo	<ul style="list-style-type: none"> • El sistema valida que la fecha de inicio del periodo de cálculo no sea anterior al 01 de diciembre del año anterior del año por pagar. • Si no se cumple el sistema muestra un mensaje de advertencia. • El sistema bloque el cálculo del aguinaldo.

Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Debe existir al menos una línea de nómina.</p> <p>El cálculo debe considerar únicamente los salarios brutos devengados.</p> <p>Auditoría extendida con trazabilidad.</p>	
Postcondiciones	
<p>El registro del cálculo del aguinaldo queda guardado en el sistema.</p> <p>El registro del monto del aguinaldo queda visible con el nombre del empleado.</p> <p>El evento queda registrado en el sistema de trazabilidad.</p>	

Fuente: Elaboración propia, 2025.

Tabla 52

Caso de uso del cálculo de salario bruto

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-CN-01	Nombre del caso de uso: Realizar el cálculo del salario bruto de un empleado
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al sistema calcular automáticamente el salario bruto de cada empleado
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	<p>El usuario debe haber iniciado sesión con un rol autorizado.</p> <p>El usuario debe tener los permisos correspondientes para realizar la acción.</p> <p>Debe existir un registro de horas trabajadas base y horas extra del empleado en el periodo.</p>

	El empleado debe tener registrado el valor de su salario por hora previamente.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar el cálculo del salario bruto de cada empleado, esto con base en las horas base y en las horas extra trabajadas registradas, de acuerdo con su salario por hora.	
<ol style="list-style-type: none"> 1. El sistema accede a la información del empleado en el periodo activo. 2. El sistema obtiene la cantidad de horas base trabajadas. 3. El sistema obtiene la cantidad de horas extra registradas. 4. El sistema calcula el salario bruto según la fórmula: $\text{salario_bruto} = (\text{cantidad_horas_base} * \text{salario_hora}) + (\text{cantidad_horas_extras} * (\text{salario_hora} * (\text{si es extra a tiempo y medio y se se multiplica por 1.5, y si es doble se multiplica por 2}))$. 5. El sistema calcula deducciones de forma automática: <ul style="list-style-type: none"> • $\text{CCSS} = \text{salario_bruto} * \text{aporte vigente}$. • Impuesto de la renta = según la tabla progresiva vigente. 6. El sistema guarda el salario bruto calculado en la tabla nómina_linea. 7. El sistema genera un registro de trazabilidad 8. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo no existen horas registradas en el periodo	<ul style="list-style-type: none"> • El sistema valida si existen horas registradas en el sistema. • Si no cumple se muestra un mensaje de advertencia. • Y el salario bruto queda en 0.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Auditoría extendida con trazabilidad.	
Postcondiciones	
<p>El salario bruto del empleado queda calculado y registrado en la nómina_Línea.</p> <p>Los cálculos quedan disponibles para la consolidación automática de la nómina.</p> <p>El evento queda registrado en el sistema de trazabilidad.</p>	

Fuente: Elaboración propia, 2025.

Tabla 53*Caso de uso Cálculo de deducciones salariales*

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número Caso de Uso: CU-CN-02	Nombre del caso de uso: Realizar el cálculo de deducciones salariales (CCSS e impuestos sobre la renta)
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH calcular las deducciones aplicables a la nómina de los empleados.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Existir un registro de nómina activa en el sistema. Existir salario bruto registrado.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar el cálculo de las deducciones que se le aplican a la nómina de los empleados, como rebajos de la Caja Costarricense del Seguro Social e impuestos sobre la renta.	
<ol style="list-style-type: none"> 9. El usuario accede al módulo de Nómina. 10. El usuario selecciona la opción Calcular deducciones. 11. El sistema despliega la lista de empleados activos con salario bruto en la nómina seleccionada. 12. El usuario selecciona al empleado para realizar el cálculo. 13. El sistema calcula deducciones de forma automática: <ul style="list-style-type: none"> • $CCSS = \text{salario_bruto} \times \text{aporte vigente}$. • $\text{Impuesto de la renta} = \text{según la tabla progresiva vigente}$. 14. El sistema despliega el desglose de deducciones. 15. El usuario selecciona la opción Guardar. 16. El sistema almacena la información en la base de datos. 17. El sistema confirma el registro exitoso. 18. El sistema visualiza los datos del registro con el nombre y el monto por pagar. 19. Finaliza el caso de uso exitosamente. 	

Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Uso de tablas de porcentajes y rangos tributarios. Auditoría extendida con trazabilidad.	
Postcondiciones	
Deducciones quedan registradas en la base de datos. El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 54

Caso de uso registro y aplicación de rebajos

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-CN-03	Nombre del caso de uso: Registro y aplicación de rebajos
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH registrar los cálculos y los rebajos de préstamos otorgados a empleados y cobros judiciales (embargos, pensiones alimentarias).
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos

Precondiciones:	<p>El usuario debe haber iniciado sesión con un rol autorizado.</p> <p>El usuario debe tener los permisos correspondientes para realizar la acción.</p> <p>Existir un registro de nómina activa en el sistema.</p>
Flujo básico del caso de uso	
<p>En este caso de uso se describe el proceso completo para registrar el rebajo de cobros que se le aplican al empleado por concepto de préstamos, cobros judiciales, entre otros.</p>	
<ol style="list-style-type: none"> 1. El usuario accede al módulo de Nómina. 2. El usuario selecciona la opción Rebajos. 3. El sistema despliega una vista con el formulario para ingresar los rebajos. 4. El usuario selecciona Agregar rebajo. 5. El sistema despliega un formulario con los campos requeridos, como tipo de rebajo, monto por rebajar, porcentaje aplicado (si se aplica) y descripción. 6. El usuario selecciona la opción Guardar. 7. El sistema valida la información. 8. El sistema almacena la información en la base de datos. 9. El sistema confirma el registro exitoso. 10. El rebajo queda referido a un empleado y marcado como activo para aplicarse en la siguiente nómina. 11. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Duplicidad de rebajo	<ul style="list-style-type: none"> • Si el sistema detecta dos rebajos del mismo tipo en el mismo periodo. • El sistema muestra un mensaje de error. • El sistema bloquea el envío. • El flujo vuelve al punto 5 del flujo básico.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error.

	<ul style="list-style-type: none"> El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Auditoría extendida con trazabilidad.	
Postcondiciones	
Las deducciones quedan registradas en la base de datos.	
El rebajo activo se aplicará automáticamente en el cálculo de la siguiente nómina.	
El evento queda registrado en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 55

Caso de uso Realizar el cálculo de nómina

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-CN-04	Nombre del caso de uso: Realizar el cálculo completo de la nómina
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH registrar los cálculos de la nómina completa de un periodo.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Debe existir un registro de nómina activa en el sistema.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar el cálculo de la nómina de un periodo, considerando salario bruto, deducciones, préstamos, cobros judiciales y salario neto final.	
<ol style="list-style-type: none"> El usuario accede al módulo Nómina. El usuario selecciona la opción Calcular nómina. El sistema despliega la lista de todas las nóminas en línea activas del periodo. El usuario selecciona el empleado al que le va a realizar el cálculo. El sistema ejecuta cálculos automáticos como salario bruto, deducciones, rebajos y salario neto. El usuario revisa los cálculos por empleado. 	

<p>7. El usuario selecciona la opción “Guardar”.</p> <p>8. El sistema almacena la información en la base de datos.</p> <p>9. El sistema confirma el registro exitoso.</p> <p>10. Finaliza el caso de uso exitosamente.</p>	
Flujos alternos	
Flujo alternativo Validación de campos vacíos	<ul style="list-style-type: none"> • Se valida que se ingrese la información en los campos requeridos obligatorios. • Si no cumple se muestra un mensaje de advertencia. • Y se rechaza el envío.
Flujo alternativo Validación de nómina cerrada	<ul style="list-style-type: none"> • El sistema valida que la nómina creada al periodo correspondiente esté activa. • Si no, muestra un mensaje de error. • No se realizan los cálculos. • No se modifican los datos en la base de datos.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de guardar, el sistema no registra la información. • Y regresa a la pantalla principal de permisos.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>Automatización de cálculo según fórmulas predefinidas.</p> <p>Auditoría extendida con trazabilidad.</p>	
Postcondiciones	
<p>Nómina queda calculada en el sistema.</p> <p>Registros de nómina_Línea quedan actualizados con los salarios netos.</p> <p>El evento queda registrado en el sistema de trazabilidad.</p>	

Fuente: Elaboración propia, 2025.

Tabla 56

Caso de uso Generar reporte de empleados

<p>Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas</p> <p>Construcciones, S.A, ubicada en Grecia</p>

Número Caso de Uso: CU-RPT-01	Nombre del caso de uso: Generar reporte de empleados
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH generar un reporte de los empleados registrados en el sistema, incluidos sus datos personales, puesto actual, jornada laboral.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Existir un registro del empleado en el sistema.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar un reporte consolidado de los empleados registrados que se encuentran activos en el sistema, incluidos sus datos personales, puesto actual y jornada laboral.	
<ol style="list-style-type: none"> 1. El usuario accede al módulo Reportes. 2. El usuario selecciona la opción Reporte de empleados. 3. El sistema despliega los filtros disponibles. 4. El usuario selecciona los filtros deseados o deja la selección “Todos”. 5. El sistema consulta en la base de datos los registros de empleados activos. 6. El sistema genera el reporte con la siguiente información: datos personales, puesto actual y jornada laboral actual. 7. El reporte se muestra en la pantalla de forma tabular. 8. El usuario puede optar por exportar el reporte en PDF o Excel. 9. El sistema confirma el registro exitoso. 10. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo No existen empleados registrados	<ul style="list-style-type: none"> • Se valida si existen empleados en el sistema. • Si no cumple se muestra un mensaje de advertencia. • El reporte no se genera.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de generar el reporte se cancela la generación. • Y el flujo regresa al flujo básico número 3.

Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Auditoría extendida con trazabilidad. Debe permitir exportación de reportes a PDF y Excel.	
Postcondiciones	
El reporte queda disponible en pantalla para el usuario. Si se exporta el archivo queda generado en el formato seleccionado. La acción queda registrada en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 57

Caso de uso generar un reporte de historial laboral

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-RPT-02	Nombre del caso de uso: Generar reporte de historial laboral
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH generar un reporte de la asistencia, horas extra, permisos e incapacidades de los empleados.
Autor del caso de uso:	Andrés Gerardo González Villegas
Actor relaciones:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Existir un registro del empleado en el sistema.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar un reporte consolidado de los empleados registrados que se encuentran activos en el sistema de la asistencia, horas extra, permisos e incapacidades de los empleados en un rango de tiempo específico.	
1. El usuario accede al módulo de reportes.	

<ol style="list-style-type: none"> 2. El usuario selecciona la opción Reporte de tiempo laboral. 3. El sistema despliega los filtros disponibles. 4. El usuario selecciona los filtros deseados o deja la selección “Todos”. 5. El sistema consulta en la base de datos los registros de empleados activos. 6. El sistema genera el reporte con la siguiente información: asistencia por rango de fecha, horas extra por empleado y totales por mes, e incapacidades por tipo y por empleado. 7. El reporte se muestra en la pantalla de forma tabular. 8. El usuario puede optar por exportar el reporte en PDF o Excel. 9. El sistema confirma el registro exitoso. 10. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo no existen empleados registrados	<ul style="list-style-type: none"> • Se valida si existen empleados en el sistema. • Si no cumple se muestra un mensaje de advertencia. • El reporte no se genera.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de generar el reporte se cancela la generación. • Y el flujo regresa al flujo básico número 3.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Auditoría extendida con trazabilidad.	
Debe permitir exportación de reportes a PDF y Excel.	
Postcondiciones	
El reporte queda disponible en la pantalla para el usuario.	
Si se exporta, el archivo queda generado en el formato seleccionado.	
La acción queda registrada en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 58

Caso de uso Generar un reporte de beneficios del empleado

<p>Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia</p>

-Número de caso de uso: CU-RPT-03	Nombre del caso de uso: Generar reporte de beneficios
Fecha elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH generar un reporte de beneficios laborales de los empleados.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Deben existir registros de vacaciones y aguinaldos en el sistema. Debe existir un registro del empleado en el sistema.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar un reporte consolidado de los empleados registrados que se encuentran activos en el sistema de los beneficios laborales, tales como vacaciones, aguinaldos y su estado de pago.	
<ol style="list-style-type: none"> 1. El usuario accede al módulo Reportes. 2. El usuario selecciona la opción Reporte de beneficios. 3. El sistema carga los datos de las tablas, como vacaciones tomadas, y disponibles por empleados, aguinaldos calculados y pagados. 4. El usuario selecciona la opción Generar. 5. El reporte se muestra en pantalla de forma tabular. 6. El usuario puede optar por exportar el reporte en PDF o Excel. 7. El sistema confirma el registro exitoso. 8. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo, no existen empleados registrados	<ul style="list-style-type: none"> • Se valida si existen empleados en el sistema. • Si no cumple se muestra un mensaje de advertencia. • El reporte no se genera.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de generar el reporte se cancela la generación. • Y el flujo regresa al flujo básico número 3.

Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Auditoría extendida con trazabilidad. Debe permitir exportación de reportes a PDF y Excel.	
Postcondiciones	
El reporte queda disponible en la pantalla para el usuario y para exportación. Si se exporta, el archivo queda generado en el formato seleccionado. La acción queda registrada en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 59

Caso de uso Generar reporte de nómina

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones S.A, ubicada en Grecia	
Número de caso de uso: CU-RPT-04	Nombre del caso de uso: Generar reporte de nómina
Fecha de elaboración:	10/09/2025
Descripción del caso de uso:	Le permite al encargado de RRHH generar un reporte de los cálculos de nómina.
Autor caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Deben existir registros de nómina en el sistema. Debe existir un registro del empleado en el sistema.
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar un reporte consolidado de los empleados registrados que se encuentran activos en el sistema de los cálculos de nómina, incluidos salario bruto y neto, deducciones y liquidaciones por salida de empleados.	
<ol style="list-style-type: none"> 1. El usuario accede al módulo Reportes. 2. El usuario selecciona la opción Reporte de Nómina. 	

<ol style="list-style-type: none"> 3. El sistema carga los datos de las tablas como salario bruto y neto, deducciones aplicadas, rebajos y liquidaciones realizadas por salida de empleados. 4. El usuario selecciona la opción Generar. 5. El reporte se muestra en la pantalla de forma tabular. 6. El usuario puede optar por exportar el reporte en PDF o Excel. 7. El sistema confirma el registro exitoso. 8. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo No existen empleados registrados	<ul style="list-style-type: none"> • Se valida si existen empleados en el sistema. • Si no cumple se muestra un mensaje de advertencia. • El reporte no se genera.
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de generar el reporte se cancela la generación. • Y el flujo regresa al flujo básico número 3.
Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
Auditoría extendida con trazabilidad. Debe permitir exportación de reportes a PDF y Excel.	
Postcondiciones	
El reporte queda disponible en pantalla para el usuario y para exportación. Si se exporta el archivo queda generado en el formato seleccionado. La acción queda registrada en el sistema de trazabilidad.	

Fuente: Elaboración propia, 2025.

Tabla 60

Caso de uso para auditoría del sistema

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
Número de caso de uso: CU-AUD-01	Nombre del caso de uso: Realizar auditoría del sistema
Fecha elaboración:	10/09/2025

Descripción del caso de uso:	Les permite al administrador del sistema y al administrador de RRHH consultar los registros de auditoría y trazabilidad
Autor del caso de uso:	Andrés Gerardo González Villegas
Actores relacionados:	Usuario del Administrador del Sistema Usuario Administrativo de RRHH Sistema de Gestión de Recursos Humanos
Precondiciones:	El usuario debe haber iniciado sesión con un rol autorizado. El usuario debe tener los permisos correspondientes para realizar la acción. Deben existir registros en las tablas de auditoría (“log_evento” y “audit_change”)
Flujo básico del caso de uso	
En este caso de uso se describe el proceso completo para realizar una consulta de los registros de auditoría y trazabilidad almacenados en las tablas “log_evento” y “audit_change”, con el fin de verificar las acciones realizadas en el sistema, garantizar la transparencia y cumplir con requerimientos de control interno.	
<ol style="list-style-type: none"> 1. El usuario accede al módulo Auditoría. 2. El usuario selecciona la opción Consultar auditoría. 3. El sistema despliega un filtro con opciones como rango de fechas, usuario que ejecutó la acción, tipo de acción, tabla afectada. 4. El usuario aplica los filtros deseados. 5. El sistema consulta las tablas “log_evento” y “audit_change”. 6. El sistema muestra los datos de la información filtrada. 7. El usuario puede visualizar en la pantalla o exportar en PDF o Excel. 8. El sistema registra la acción en la trazabilidad. 9. Finaliza el caso de uso exitosamente. 	
Flujos alternos	
Flujo alternativo, filtros vacíos	<ul style="list-style-type: none"> • Si no se da ningún filtro. el sistema muestra todos los datos.
Flujo alternativo Sin filtros	<ul style="list-style-type: none"> • Si el sistema no encuentra los datos filtrados el sistema muestra un mensaje de advertencia
Flujo alternativo Cancelación de proceso	<ul style="list-style-type: none"> • Si el usuario decide cancelar la operación antes de generar el reporte se cancela la generación. • Y el flujo regresa al flujo básico número 3.

Flujo alternativo Error de sistema	<ul style="list-style-type: none"> • El sistema no puede registrar o validar debido a un error técnico. • Se muestra un mensaje de error. • El flujo finaliza y el sistema no ejecuta los cambios.
Requerimientos especiales	
<p>El sistema debe garantizar solo lectura de los registros (no se pueden modificar ni eliminar)</p> <p>Debe permitir exportación de reportes a PDF y Excel.</p> <p>La visualización debe estar paginada para eficiencia.</p> <p>Auditoría extendida con trazabilidad.</p>	

Fuente: Elaboración propia, 2025.

Diseño del sistema

En esta sección se presenta el diseño integral del sistema, que abarca tanto su estructura externa como la interna del prototipo de gestión de recursos humanos para la empresa Quesada Vargas Construcciones, S.A. Se incluyen la arquitectura del sistema, la arquitectura del software, los diseños de entrada y salida mediante pantallas representativas, y el diseño físico de la base de datos construido con MySQL Workbench, y el diccionario de datos correspondiente. Además, se presentan los diagramas de flujo que ilustran los procesos clave del sistema y los diagramas UML más relevantes. Todo este conjunto de elementos permite comprender cómo está estructurado el sistema y cómo se espera que este funcione.

Arquitectura del Sistema

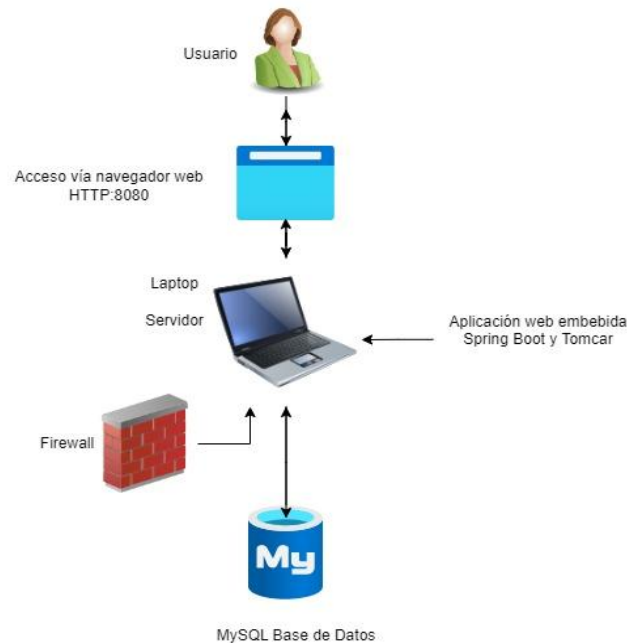
La arquitectura del sistema se basa en un modelo cliente-servidor local, diseñado para operar dentro de las condiciones tecnológicas actuales de la empresa Quesada Vargas Construcciones, S.A. Debido a que la empresa no cuenta con infraestructura tecnológica avanzada ni con un servidor dedicado, se utiliza una computadora portátil como servidor central, encargada de ejecutar la aplicación web embebida desarrollada en Spring Boot con servidor Tomcat 10.1 y de gestionar la base de datos MySQL instalada en el mismo equipo.

El sistema ejecuta de forma local y es accesible mediante un navegador web desde el mismo dispositivo (por medio de <http://localhost:8080>). Esta configuración permite que el sistema funcione sin conexión a Internet, aprovechando los recursos disponibles de la empresa.

La arquitectura propuesta es simple, funcional y viable, ya que se adapta a las condiciones actuales de la empresa sin requerir inversión adicional en infraestructura de red o servidores externos. Además, permite validar el prototipo funcional del sistema de gestión de Recursos Humanos en un entorno real que garantiza su operatividad bajo las condiciones reales del negocio.

Figura 23

Arquitectura del sistema de recursos humanos



Fuente: Elaboración propia, 2025.

Arquitectura del Software

El sistema web de gestión de recursos humanos está construido sobre un conjunto de tecnologías integradas que conforman la arquitectura de software. En el frontend se emplean Thymeleaf, HTML5, CSS3 y Bootstrap5, los cuales permiten construir una interfaz moderna, adaptable y visualmente organizada. Estas tecnologías se comunican con el *backend*, desarrollado en Java bajo el framework Spring Boot 3.x, que incorpora módulos de Spring MVC, Spring Security y Spring AOP para la gestión de la lógica de negocio, la seguridad de acceso y la trazabilidad de operaciones.

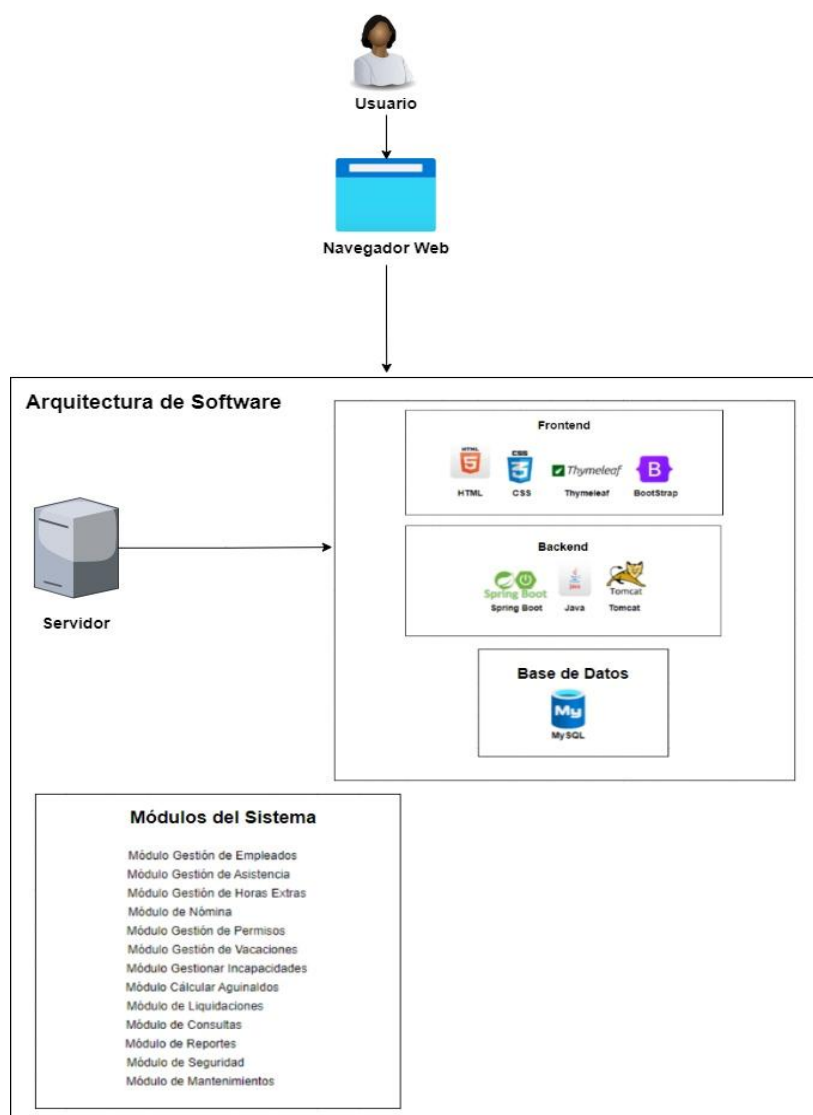
El sistema se ejecuta sobre un servidor embebido Tomcat 10.1 que almacena la información en una base de datos MySQL, elegida por su estabilidad y compatibilidad con Spring Data JPA.

El conjunto conforma una arquitectura monolítica de tipo cliente-servidor, en que el usuario accede al sistema a través de un navegador web por el puerto HTTP 8080.

La siguiente figura muestra una vista general de las tecnologías que intervienen en el desarrollo y la ejecución del sistema.

Figura 24

Arquitectura general del software del sistema



Fuente: Elaboración propia, 2025.

La figura anterior presenta la arquitectura tecnológica del sistema, en que se ilustran las tecnologías empleadas en cada capa y los principales módulos que conforman la aplicación.

La arquitectura del software del sistema se fundamenta en una estructura por capas que organiza los componentes internos de forma lógica, mantenible y escalable. Este enfoque permite una separación clara de responsabilidades, mejora el flujo entre capas y facilita la modificación o ampliación del sistema sin comprender su funcionamiento general. Se

desarrolló utilizando el framework de Spring Boot, en lenguaje Java, el cual simplifica la configuración y el despliegue de aplicaciones web. Se integraron, además, tecnologías como Spring Security para el control del acceso y la gestión de roles, Spring AOP (Programación Orientada a Aspectos) para la trazabilidad y auditoría, y MySQL como motor de base de datos. En la capa de presentación se emplearon Thymeleaf, HTML, CSS y Bootstrap, con el propósito de construir una interfaz moderna, clara y adaptable a dispositivos, que garantiza una experiencia de usuario fluida.

El sistema adopta el patrón Modelo-Vista-Controlador (MVC), que facilita la separación entre la lógica de presentación, la lógica de negocio y el acceso a datos. En la capa de vista se ubican las páginas HTML renderizadas con Thymeleaf y con estilos de Bootstrap y CSS, que construyen la interfaz directa con el usuario. A través de esta capa se muestran los datos procesados en el sistema y se reciben las solicitudes que serán enviadas en la capa de control. La capa de controlador se conforma por las clases anotadas con `@Controller`, actúa como intermediaria entre la interfaz y la lógica de negocio, y gestiona las peticiones del usuario, aplicando las reglas de seguridad mediante Spring Security y retornando las vistas correspondientes.

La lógica de negocio se encuentra concentrada en las capas `Service` y `ServiceImpl`, en las que se definen e implementan las operaciones esenciales del sistema. Este diseño se aplica al patrón `Service Layer`, el cual centraliza las reglas de negocio y facilita su modificación sin afectar otras partes del software, y promueve la escalabilidad y la reutilización de código. Complementando esta capa se implementó un componente de trazabilidad utilizando Spring AOP, que registra automáticamente las operaciones ejecutadas por los usuarios, tales como la creación, modificación o eliminación de registros. Gracias a este enfoque es posible mantener un historial detallado de acciones, lo que refuerza la transparencia y trazabilidad del sistema interno.

Para el intercambio de información entre las distintas capas se emplean los `Data Transfer Object (DTO)`, estructuras que permiten transportar datos de manera segura y controlada sin exponer directamente las entidades del modelo. La conversión entre los DTO y las entidades se realiza mediante los `Mapper`, los cuales aplican el patrón `Data Mapper` para transformar los datos sin afectar la lógica de negocio. Este diseño mejora la organización y mantiene la comunicación limpia entre los distintos niveles de la aplicación.

El modelo de datos está compuesto por entidades que representan las tablas de la base de datos, que refleja finalmente el dominio del sistema, que en este caso corresponde a la gestión de recursos humanos. Dichas entidades establecen las relaciones necesarias entre los

distintos módulos, tales como empleados, permisos, incapacidades, vacaciones, liquidaciones, asistencia, horas extra, permisos y nómina, lo que garantiza la integridad y coherencia de la información. Para acceder y manipular los datos se implementó la capa de repositorio, basada en Spring Data JPA, que realiza operaciones CRUD (crear, leer, actualizar y eliminar) de forma eficiente. Además, permite escribir consultas manuales que se requieran para extender la funcionalidad del sistema y adaptarse a los requerimientos del sistema. Finalmente, la información se almacena en una base de datos MySQL, seleccionada por su estabilidad, compatibilidad y facilidad de integración con Spring Boot.

La arquitectura implementada no solo organiza el flujo de trabajo desde la vista hasta la base de datos, sino que también sigue principios de diseño orientado a objetos, especialmente principios SOLID (Single Responsibility, Open/Close, Liskov Substitution, Interface Segregation y Dependency Inversion), propuestos por Robert C. Martin. Al respecto, Sánchez (2023) acerca de los principios SOLID menciona lo siguiente:

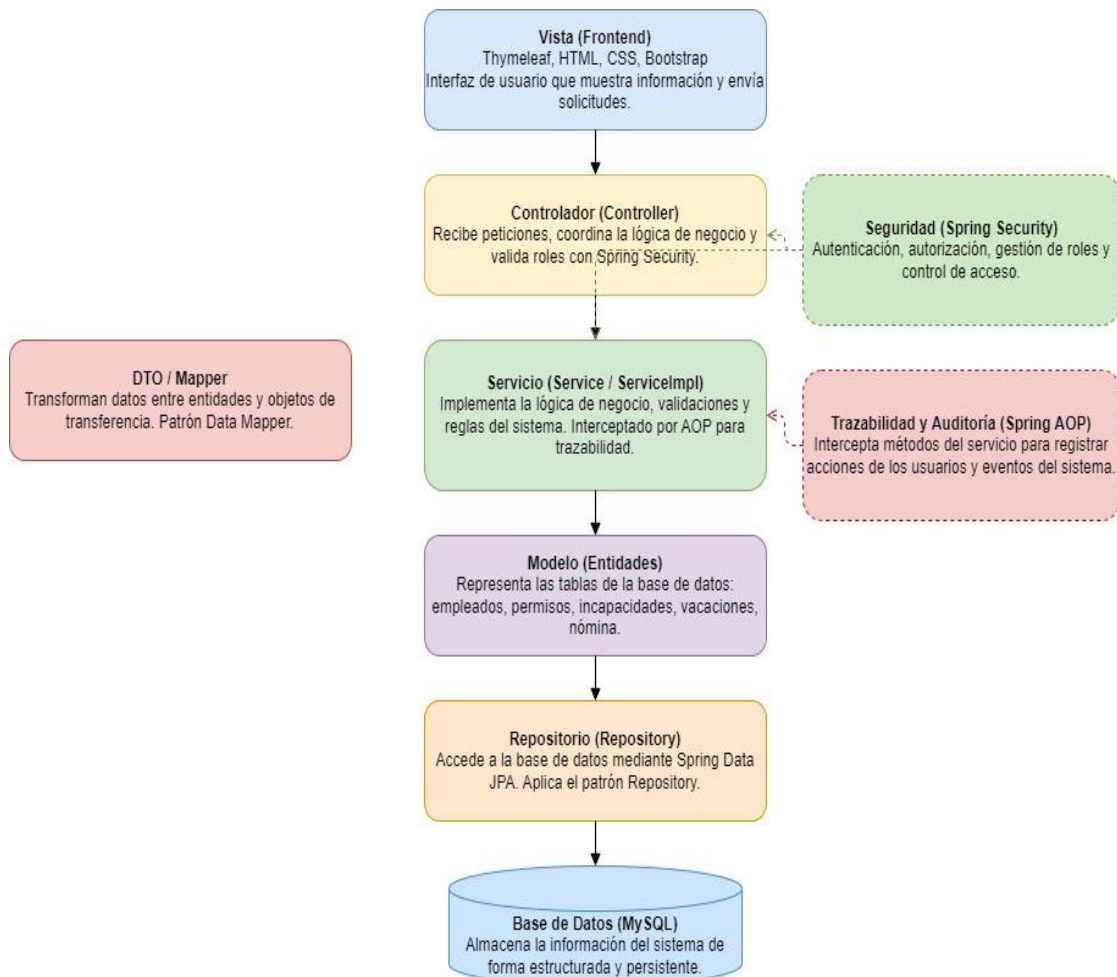
SOLID es un acrónimo que encapsula cinco principios del diseño de software orientado a objetos (Single Responsibility, Open/Close, Liskov Substitution, Interface Segregation y Dependency Inversion), promulgado por Robert C. Martin, cuyo enfoque es crear sistemas más comprensibles, flexibles y mantenibles.

Estos principios son cinco buenas prácticas de programación orientada a objetos para construir software. Single Responsibility, en español se traduce como Principio de Responsabilidad Única (representa la S de la palabra SOLID), y dice que cada clase debe encargarse de una sola tarea o responsabilidad dentro del sistema. El principio de Open/Close (representa la O de la palabra SOLID) en español se conoce como principio de Abierto/Cerrado, y establece que el código debe estar abierto para ser extendido y cerrado al ser modificado, y esto significa que se pueden agregar nuevas funcionalidades sin tener que alterar el código existente. El principio de sustitución de Liskov (Liskov Substitution Principle) se refiere a que las clases hijas deben poder sustituir a sus clases padre sin alterar el comportamiento del sistema. El principio de segregación de interfaces (Interface Segregation Principle) principio dice que es mejor tener varias o muchas interfaces pequeñas que una sola interfaz grande y general, lo que permite que las clases que realmente necesiten de la interfaz solo implementen lo que realmente necesiten, a fin de evitar dependencias innecesarias. Por último, el principio de inversión de dependencias (Dependency Inversion Principle), enseña que las clases deben depender de abstracciones (interfaces) y no de implementaciones concretas, y esto permite realizar cambios internos cuando se requieran sin afectar al resto del sistema.

En conjunto, la arquitectura del sistema integra los patrones MVC, Service Layer, Data Mapper, Repository y programación orientada a aspectos AOP, lo que proporciona una estructura sólida, coherente y alineada con las buenas prácticas de la ingeniería de software. Gracias a esta organización el sistema es fácil de mantener, auditar y extender, para asegurar su funcionamiento estable y preparado para futuras mejoras.

Figura 25

Arquitectura interna del software del sistema



Fuente: Elaboración propia, 2025.

La figura muestra la arquitectura interna del sistema desarrollado en Java utilizando Spring Boot. En ella se observa la organización por capas que estructuran el flujo de información desde la interfaz de usuario hasta la base de datos. En la parte superior se ubica la Vista, compuesta por páginas creadas con Thymeleaf, HTML, CSS Y Bootstrap, las cuales representan el punto de interacción directa entre el usuario y el sistema. Desde esta capa, las solicitudes son dirigidas al Controlador, responsable de coordinar la comunicación con la lógica de negocio, validar peticiones y gestionar la seguridad de acceso mediante Spring Security.

Posteriormente, la información es procesada en la capa de Servicio, en la que se concentra la lógica de negocio del sistema. En este nivel se aplican las reglas específicas de los procesos de gestión de recursos humanos, como el cálculo de nómina, control de permisos, vacaciones, incapacidades, horas extra, liquidaciones, aguinaldos o gestiones del empleado. Esta capa es interceptada por Spring AOP, que implementa la trazabilidad y auditoría del sistema, que registra automáticamente las acciones ejecutadas por los usuarios, lo que fortalece la transparencia y el control interno.

El flujo de datos continúa hacia las capas de Modelo y Repositorio, en que las entidades del sistema representan la estructura de las tablas de la base de datos y permite realizar las operaciones de persistencia. La comunicación entre la lógica de negocio y la base de datos se realiza mediante Spring Data JPA, para realizar las operaciones CRUD y facilitar el mantenimiento. Los resultados obtenidos se transforman mediante los DTO y Mapper, para trasladar la información de manera segura y controlada entre las capas, para evitar exponer directamente las entidades del dominio. Finalmente, los datos procesados se devuelven a la Vista, y son presentados de forma clara y estructurada al usuario.

De esta manera el sistema propuesto logra integrar una arquitectura robusta y escalable, sustentada en buenas prácticas de ingeniería de software, para garantizar la seguridad, la trazabilidad y la correcta separación de responsabilidades entre sus diferentes capas.

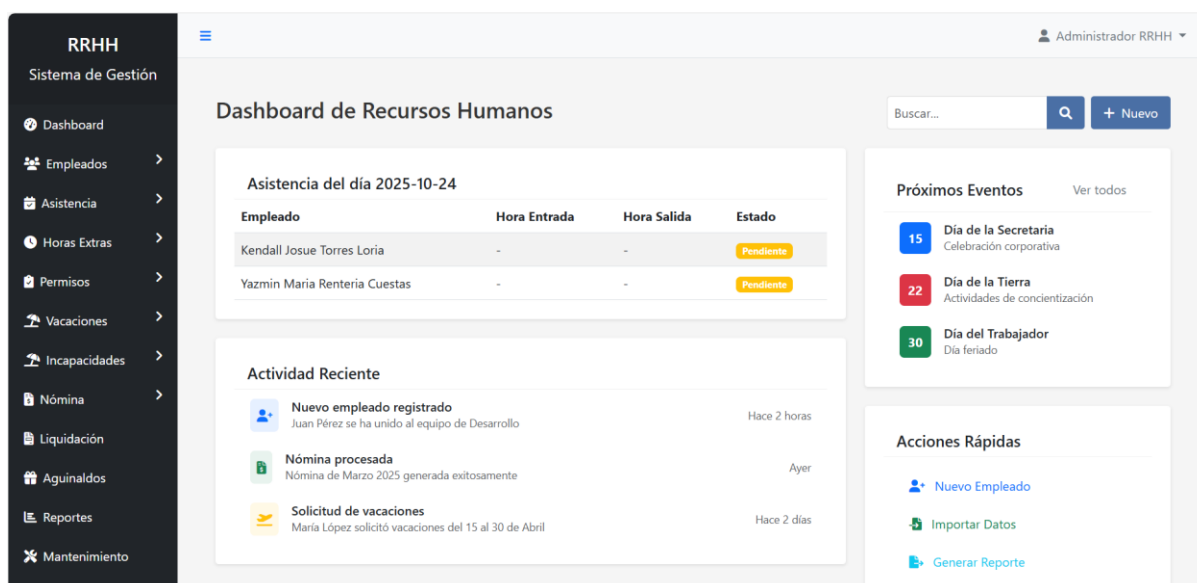
Diseño de entradas

En este apartado se representan las principales pantallas que conforman el prototipo del sistema de gestión de recursos humanos. Cada interfaz fue diseñada siguiendo criterios de claridad y facilidad de uso, con el objetivo de garantizar una interacción intuitiva entre el usuario y el sistema. Las siguientes figuras muestran las pantallas más representativas del desarrollo, y evidencian la estructura de navegación, los formularios de ingreso de datos y las funcionalidades esenciales implementadas en el sistema.

Figura 26*Diseño de entrada inicio de sesión*

Fuente: Elaboración propia, 2025.

La pantalla de inicio de sesión permite al usuario autenticarse en el sistema mediante la introducción de su nombre de usuario y contraseña. Esta pantalla permite el acceso al sistema únicamente a los usuarios autorizados. Además, incorpora un botón de ingreso que valida las credenciales y redirige al usuario a la interfaz principal del sistema.

Figura 27*Diseño de entrada pantalla principal del sistema*

Fuente: Elaboración propia, 2025.

La pantalla principal del sistema funciona como el punto de inicio desde el cual el usuario puede acceder a los diferentes módulos del sistema. Su diseño permite una

navegación ágil y organizada, y facilita la interacción con las principales funcionalidades disponibles, como gestión de empleados, asistencia, horas extra, permisos, vacaciones, incapacidades, nómina, liquidación y aguinaldos, entre otras funciones. Desde esta pantalla el usuario tiene una visión general del sistema y puede acceder fácilmente a las secciones que requiere utilizar.

Figura 28

Diseño de pantalla formulario para ingresar a un empleado

The image shows a web form titled 'Sistema de Recursos Humanos' and 'Registro de Nueva Persona'. The form contains the following fields and sections:

- Personal Information:**
 - Primer Nombre (First Name)
 - Segundo Nombre (Second Name)
 - Primer Apellido (First Surname)
 - Segundo Apellido (Second Surname)
 - Tipo de identificación (Type of identification) - dropdown menu
 - Número de identificación (Identification number)
 - Correo Electrónico (Email)
 - Fecha de Nacimiento (Date of birth) - dd/mm/aaaa format with a calendar icon
 - Estado Civil (Marital status) - dropdown menu
 - Teléfono Fijo (Fixed phone)
 - Género (Gender) - dropdown menu
- Telefonos Celulares (Cellular Phones):**
 - + Añadir Teléfono (Add phone)
- Dirección (Address):**
 - Provincia (Province) - dropdown menu
 - Cantón (Canton) - dropdown menu
 - Distrito (District) - dropdown menu
 - Dirección Exacta (Exact address) - text area
- Buttons:**
 - ← Cancelar (Cancel)
 - Guardar Persona (Save person)

Fuente: Elaboración propia, 2025.

La pantalla del formulario de empleados tiene como finalidad registrar la información del personal de cada colaborador. A través de esta interfaz se recopilan datos esenciales, como nombre completo, número de identificación, correo electrónico y otros detalles relevantes que la empresa requiere para sus labores administrativas dentro del departamento de Recursos Humanos. Su diseño facilita la captura ordenada y precisa de la información, para garantizar que los registros se mantengan actualizados y completos.

Figura 29*Diseño de entrada registro de asistencia*

Gestión de Asistencia

Registrar Asistencia

Empleado
Kendall Josue Torres Loria

Fecha
24/10/2025

Tipo de hora extra
Normal

Hora de Entrada
--:--

Hora de Salida
--:--

Observaciones

← Cancelar **Guardar**

Fuente: Elaboración propia, 2025.

La pantalla del formulario de asistencia permite registrar la presencia de los empleados en fechas determinadas, al asignar la hora de entrada y la hora de salida correspondientes. Este diseño facilita llevar un control ordenado y preciso de la asistencia personal, a fin de contribuir con la gestión administrativa y el seguimiento puntual del cumplimiento de la jornada laboral.

Figura 30

Diseño de pantalla Solicitar un permiso

Gestión de Permisos

Registrar Permiso

Empleado
Kendall Josue Torres Loria

Tipo de Permiso

Fecha de Inicio
dd/mm/aaaa

Fecha de Fin
dd/mm/aaaa

Estado
PENDIENTE

Motivo

← Cancelar Guardar

Fuente: Elaboración propia, 2025.

La pantalla del formulario de permisos permite registrar las solicitudes de ausencias presentadas por los colaboradores de la empresa. A través de esta interfaz se ingresan datos como la fecha de inicio y la finalización del permiso, el estado asignado y el motivo correspondiente. Su diseño facilita la gestión y el seguimiento de las solicitudes, a fin de asegurar un control adecuado de las ausencias dentro del departamento de Recursos Humanos.

Diseño físico de la base de datos

En esta sección se presenta el diseño físico de la base de datos del sistema de gestión de recursos humanos. Para su elaboración se utilizó la herramienta MySQL Workbench, la cual permite visualizar y organizar las tablas, las relaciones y la estructura necesaria para almacenar la información de forma ordenada y segura. Además, se incluye el diccionario de datos, en el que se detallan los campos, tipos de datos y descripciones de cada elemento, con el fin de facilitar la comprensión y el mantenimiento del sistema.

Tabla 61*Diccionario de datos, tabla persona*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_persona	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la persona
primer_nombre	varchar(50)	No				Primer nombre de la persona
segundo_nombre	varchar(50)	Sí				Segundo nombre de la persona
primer_apellido	varchar(50)	No				Primer apellido de la persona
segundo_apellido	varchar(50)	No				Segundo apellido de la persona
correo	varchar(150)	Sí				Correo electrónico de la persona
telefono_fijo	varchar(15)	Sí				Teléfono fijo de la persona
tipo_identificacion_id	bigint	No		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tipo de identificación de la persona, cédula, permiso de trabajo para extranjeros
numero_identificacion	varchar(50)	No			<input checked="" type="checkbox"/>	Número identificación de la persona
direccion_id	bigint	No		<input checked="" type="checkbox"/>		Clave foránea que referencia la tabla dirección. Identifica la dirección asociada al registro.
fecha_de_nacimiento	date	Sí				Fecha de nacimiento de la persona
genero	varchar(20)	Sí				Género de la persona
estado_civil	varchar(20)	Sí				Estado civil de la persona
is_activo	tinyint(1)	Sí				Booleano que define si la persona esta activa (1) o no (0) en el sistema.
fecha_creacion	timestamp	Sí				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 62*Diccionario de datos tabla-teléfono celular*

Columna	Tipo de dato	NN	PK	FK	UK	Descripción
id_telefono_celular	bigint	No	<input checked="" type="checkbox"/>			Identificador único para el teléfono celular
persona_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea para la tabla persona
numero	varchar(30)	No			<input checked="" type="checkbox"/>	Número de celular
es_principal	tinyint(1)	Sí				Campo para determinar si es el número principal, de tener más números
fecha_creacion	timestamp	Sí				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.**Tabla 63***Diccionario de datos tabla dirección*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_direccion	bigint	No	<input checked="" type="checkbox"/>			Identificador único para la dirección
provincia_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea para la tabla provincia
canton_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea para la tabla cantón
distrito_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea para la tabla distrito
direccion_exacta	varchar(500)	No				Dirección exacta de la persona
fecha_creacion	timestamp	Sí				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.**Tabla 64***Diccionario de datos tabla provincia*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_provincia	bigint	No	<input checked="" type="checkbox"/>			Identificador único para la provincia
nombre	varchar(100)	No				Nombre de la provincia
activo	tinyint(1)	Sí				Booleano para determinar si está activo o no. Para

control lógico sin
eliminar datos

Fuente: Elaboración propia, 2025.

Tabla 65

Diccionario de datos tabla cantón

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_canton	bigint	No	<input checked="" type="checkbox"/>			Identificador único para el cantón
provincia_id	bigint	No		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Llave foránea de la tabla provincia
nombre	varchar(100)	No			<input checked="" type="checkbox"/>	Nombre del cantón
codigo	varchar(255)	No				Código del cantón

Fuente: Elaboración propia, 2025.

Tabla 66

Diccionario de datos tabla distrito

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_distrito	bigint	No	<input checked="" type="checkbox"/>			Identificador único del distrito
canton_id	bigint	No		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Llave foránea de la tabla cantón
nombre	varchar(100)	No			<input checked="" type="checkbox"/>	Nombre del distrito
codigo	varchar(255)	No				Código del distrito

Fuente: Elaboración propia, 2025.

Tabla 67

Diccionario de datos tabla empleado

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_empleado	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la tabla empleado
persona_id	bigint	No		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Llave foránea de la tabla persona
fecha_ingreso	date	No				Fecha de ingreso del empleado
activo	tinyint(1)	No				Booleano para determinar si está activo o no. Para control lógico sin eliminar datos
fecha_creacion	datetime(6)	Sí				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 68*Diccionario de datos tabla permiso*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_permiso	bigint	No	<input checked="" type="checkbox"/>			Identificador único del permiso
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
tipo_permiso	varchar(100)	No				Tipo de permiso que solicita
fecha_inicio	date	No				Fecha de inicio del permiso
fecha_fin	date	No				Fecha final del permiso
estado	enum	No				Estado del permiso, alguna de las tres opciones disponibles (PENDIENTE, APROBADO y RECHAZADO)
motivo	text	Sí				Motivo por la cual se solicita el permiso
fecha_creacion	datetime(6)	No				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.**Tabla 69***Diccionario de datos tabla liquidación*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_liquidacion	int	No	<input checked="" type="checkbox"/>			Identificador único de la liquidación
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
fecha_ingreso	date	Sí				Fecha de ingreso del empleado
fecha_salida	date	No				Fecha de salida del empleado
fecha_liquidacion	date	No				Fecha del cálculo de la liquidación
tipo_liquidacion	enum	No				Tipo de liquidación de acuerdo al origen de salida del empleado
motivo	text	Sí				Motivo de la liquidación
monto_total	decimal(12,2)	No				Monto total del cálculo de la liquidación

detalles	json	Sí				Detalles de la liquidación
fecha_de creacion	datetime(6)	No				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 70

Diccionario de datos de la tabla horas extra

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_hora_extra	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la hora extra
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
fecha	date	No				Fecha del día en que se labora la hora extra
hora_fin	time	No				Hora de finalización de las horas extras
hora_inicio	time	No				Hora de inicio de las horas extra
aprobada	bit(1)	No				Booleano para determinar si fue o no aprobada la hora extra
motivo	varchar(500)	Sí				Motivo de la aprobación o rechazo de la hora extra
tipo_feriado	enum	Sí				Tipo que se realizan las horas extra.
fecha_creacion	datetime(6)	Sí				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 71

Diccionario de datos tabla nómina línea

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_nomina_linea	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la nómina línea
periodo_id_nomina	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla nomina periodo
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
salario_bruto	decimal(12,2)	No				Salario bruto de la nómina del empleado

total_descuentos	decimal(12,2)	No	Total de rebajos a los que se les aplica al salario bruto
salario_netos	decimal(12,2)	No	Monto total para pagar al empleado
estado	enum	No	Estado de la línea de nómina
detalles	json	Sí	Información adicional que se requiera agregar
fecha_creacion	datetime(6)	NO	Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 72

Diccionario de datos tabla nómina periodo

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_periodo_nomina	int	No	<input checked="" type="checkbox"/>			Identificador único de la nómina periodo
fecha_inicio	date	No				Fecha del inicio del periodo de la nómina
fecha_fin	date	No				Fecha final del periodo de la nómina
descripcion	varchar(255)	Sí				Descripción del periodo de la nómina que se requiera agregar
activo	bit(1)	No				Indica si el período está activo
estado	enum	No				Estado actual del período de nómina
fecha_creacion	datetime(6)	Sí				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 73

Diccionario de datos tabla nómina rebajo

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_nomina_rebajo	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la nómina rebajo

linea_id	bigint	No	<input checked="" type="checkbox"/>	Llave foránea de la tabla nómina línea
id_tipo_rebajo	enum	No	<input checked="" type="checkbox"/>	Llave foránea de la tabla tipo rebajo
porcentaje	decimal(5,2)	Sí		Porcentaje correspondiente de acuerdo con el tipo de rebajo
monto	decimal(12,2)	No		Monto por rebajar correspondiente al porcentaje aplicado
descripcion	text	Sí		Descripción de la nómina de rebajo
referencia_externa	varchar(255)	Sí		Referencia externa que respalde el tipo de rebajo y su porcentaje
fecha_creacion	datetime(6)	No		Fecha y hora de creación del registro de rebajo

Fuente: Elaboración propia, 2025.

Tabla 74

Diccionario de datos tabla aguinaldo

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_aguinaldo	int	No	<input checked="" type="checkbox"/>			Identificador único del aguinaldo
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
anio	int	No				Año correspondiente del pago del aguinaldo
periodo_inicio	date	No				Periodo de inicio del cálculo del aguinaldo
periodo_fin	date	No				Periodo de finalización del cálculo del aguinaldo
monto	decimal(12,2)	No				Monto correspondiente para pagar
promedio_salario	decimal(12,2)	Sí				Promedio del salario del empleado entre las fechas de cálculo
observaciones	text	Sí				Observaciones que se requiera agregar
fecha_creacion	datetime	No				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 75

Diccionario de datos tabla formación académica

Columna	Tipo de dato	NN	PK	FK	UK	Descripción
id_formacion_academica	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la formación académica
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
nivel	varchar(100)	No				Nivel de la formación académica
titulo	varchar(100)	No				Nombre del título obtenido
institucion	varchar(100)	No				Institución en la que se llevó a cabo la formación académica
fecha_obtencion	date	No				Fecha de la obtención
activo	bit(1)	No				Indica si el registro está activo
fecha_creacion	date	No				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propias, 2025.

Tabla 76

Diccionario de datos tabla jornada empleado

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_jornada_empleado	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la jornada del empleado
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
tipo_jornada_laboral_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla tipo de jornada laboral
fecha_inicio	date	No				Fecha de inicio de la jornada
fecha_fin	date	Sí				Fecha final de la jornada
activo	bit(1)	No				Indica si esta jornada es la vigente del empleado
fecha_creacion	datetime(6)	No				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 77*Diccionario de datos tabla jornada laboral*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_tipo_jornada_laboral	bigint	No	<input checked="" type="checkbox"/>			Identificador único del tipo de jornada laboral
nombre	varchar(100)	No			<input checked="" type="checkbox"/>	Nombre del tipo de jornada laboral
hora_inicio	time	No				Hora de inicio del tipo de jornada laboral
hora_fin	time	No				Hora en que finaliza la jornada laboral
horas_laborales_diarias	decimal(38,2)	No				Cantidad de horas de la jornada laboral
dias_laborales_semanales	int	No				Días de trabajo de la jornada laboral
total_horas_semanales	int	No				Total de horas semanales de la jornada laboral
porcentaje_recargo	decimal(5,2)	Sí				Porcentaje de recargo que aplica a la jornada
es_especial	bit(1)	No				Booleano para identificar si la jornada laboral es especial
activo	bit(1)	No				Booleano para determinar si está activo o no. Para control lógico sin eliminar datos

Fuente: Elaboración propia, 2025.**Tabla 78***Diccionario de datos tabla asistencia*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_asistencia	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la asistencia
empleado_id	bigint	No		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Llave foránea de la tabla empleado
fecha	date	No			<input checked="" type="checkbox"/>	Fecha del día de la asistencia
hora_entrada	time	No				Hora que ingreso
hora_salida	time	Sí				Hora que salió
completado	bit(1)	No				Indica si el registro de asistencia está completado (entrada y salida)

observaciones	Text	Sí	Observaciones de ser requeridas para dar más detalle
tipo_feriado	enum	Sí	Indica si el día fue normal o un tipo de feriado
fecha_creacion	datetime(6)	No	Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 79

Diccionario de datos tabla contacto de emergencia

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_contacto_emergencia	bigint	No	<input checked="" type="checkbox"/>			Identificador único del contacto de emergencia
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
nombre	varchar(50)	No				Nombre de la persona del contacto de emergencia
apellido	varchar(50)	No				Apellido de la persona del contacto de emergencia
telefono	varchar(15)	No				Número de celular del contacto de emergencia
parentesco	varchar(50)	No				Parentesco con el empleado
es_principal	tinyint(1)	No				Booleado que permite definir identificar cuál es el contacto principal
activo	bit(1)	No				Indica si el contacto de emergencia está activo
fecha_creacion	timestamp	Sí				Fecha y hora de creación. Se asigna automáticamente al insertar

Fuente: Elaboración propia, 2025.

Tabla 80

Diccionario de datos tabla historial laboral

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_historial_laboral	int	No	<input checked="" type="checkbox"/>			Identificador único del historial laboral
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado

puesto_id	int	No	<input checked="" type="checkbox"/>	Llave foránea de la tabla puesto
salario_hora	decimal(12,2)	No		Monto del salario por hora
fecha_inicio	date	No		Fecha de inicio en que empezó a laborar
fecha_fin	date	Sí		Fecha de finalización de labores
tipo_jornada_laboral_id	bigint	No	<input checked="" type="checkbox"/>	Tipo de la jornada laboral
fecha_creacion	datetime(6)	No		Fecha y hora de creación de registro

Fuente: Elaboración propia, 2025.

Tabla 81

Diccionario de datos tabla puesto

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_puesto	int	No	<input checked="" type="checkbox"/>			Identificador único del puesto
nombre	varchar(100)	No			<input checked="" type="checkbox"/>	Nombre del puesto
descripcion	text	Sí				Descripción del puesto
salario_base	decimal(12,2)	No				Salario base del puesto
activo	bit(1)	No				Booleano para determinar si está activo o no. Para control lógico sin eliminar datos

Fuente: Elaboración propia, 2025.

Tabla 82

Diccionario de datos tabla incapacidad

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_incapacidad	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la incapacidad
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
fecha_inicio	date	No				Fecha de inicio de la incapacidad
fecha_fin	date	No				Fecha de finalización de la incapacidad
tipo	enum	No				Tipo de incapacidad
dias	int	Sí				Cantidad de días de incapacidad
fecha_creacion	datetime(6)	Sí				Fecha y hora de creación del registro

Fuente: Elaboración propia, 2025.

Tabla 83*Diccionario de datos tabla vacaciones*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_vacaciones	bigint	No	<input checked="" type="checkbox"/>			Identificador único de la vacación
empleado_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla empleado
fecha_solicitud	datetime(6)	No				Fecha de solicitud de la vacación
fecha_inicio	date	No				Fecha de inícia de la vacación
fecha_fin	date	No				Fecha final de la vacación
dias_solicitados	int	No				Cantidad de días solicitados
estado	enum	No				Estado de la vacación (PENDIENTE, APROBADA, RECHAZADA)
monto_pago	decimal(38,2)	No				Monto que se debe pagar por las vacaciones
valor_promedio_dia	decimal(38,2)	Sí				Valor promedio del salario por día para el cálculo
observaciones	varchar(255)	Sí				Observaciones de la vacación
fecha_creacion	datetime(6)	Sí				Fecha y hora de creación del registro

Fuente: Elaboración propia, 2025.**Tabla 84***Diccionario de datos tabla saldo de vacaciones*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_saldo_vacaciones	bigint	No	<input checked="" type="checkbox"/>			Identificador único del saldo de vacación
empleado_id	bigint	No		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Llave foránea de la tabla empleado
dias_disponibles	int	No				Días disponibles de vacaciones del empleado
ultima_actualizacion	date	No				Fecha de la última actualización

Fuente: Elaboración propia, 2025.

Tabla 85*Diccionario de datos tabla tipo de identificación*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_tipo_identificacion	bigint	No	<input checked="" type="checkbox"/>			Identificador único del tipo de identificación
nombre	varchar(50)	No			<input checked="" type="checkbox"/>	Nombre del tipo de identificación
descripcion	varchar(255)	Sí				Descripción del tipo de identificación
longitud_minima	int	NO				Longitud mínima de dígitos del número de identificación
longitud_maxima	int	No				Longitud máxima de dígitos del número de identificación
activo	tinyint(1)	Sí				Booleano para determinar si está activo o no. Para control lógico sin eliminar datos

Fuente: Elaboración propia, 2025.**Tabla 86***Diccionario de datos tabla audit change*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_change	bigint	No	<input checked="" type="checkbox"/>			Identificador único del cambio registrado
tabla	varchar(100)	No				Nombre de la tabla donde ocurrió el cambio
id_registro	bigint	No				Identificador del registro afectado dentro de la tabla indicada
columna	varchar(100)	No				Nombre de la columna que fue modificada
valor_anterior	text	Sí				Valor anterior antes del cambio
valor_nuevo	text	Sí				Valor nuevo después del cambio
usuario	varchar(100)	No				Usuario que realizó el cambio
fecha	timestamp	No				Fecha en que se registró el cambio

Fuente: Elaboración propia, 2025.

Tabla 87*Diccionario de datos tabla auth role*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_role	bigint	No	<input checked="" type="checkbox"/>			Identificador único del rol
nombre	varchar(50)	No			<input checked="" type="checkbox"/>	Nombre del rol
descripcion	varchar(255)	Sí				Descripción del role

Fuente: Elaboración propia, 2025.**Tabla 88***Diccionario de datos auth role permission*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
role_id	bigint	No	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Llave foránea de la tabla role
permission_id	bigint	No	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Llave foránea de la tabla auth permission

Fuente: Elaboración propia, 2025.**Tabla 89***Diccionario de datos tabla auth permission*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_permission	bigint	No	<input checked="" type="checkbox"/>			Identificador único del permiso
nombre	varchar(100)	No			<input checked="" type="checkbox"/>	Nombre del permiso y debe ser único
descripcion	varchar(255)	Sí				Descripción del permiso

Fuente: Elaboración propia, 2025.**Tabla 90***Diccionario de datos tabla auth user*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_user	bigint	No	<input checked="" type="checkbox"/>			Identificador único del usuario
username	varchar(100)	No			<input checked="" type="checkbox"/>	Nombre del usuario y debe ser único
password_hash	varchar(255)	No				Contraseña del usuario
password_last_changed	datetime(6)	Sí				Fecha del último cambio de la contraseña
password_reset_required	bit(1)	No				Indica si se requiere un cambio de la contraseña
nombre_completo	varchar(200)	No				Nombre completo de la persona
email	varchar(150)	Sí				Correo electrónico del usuario

activo	bit(1)	No		Booleano para determinar si está activo o no. Para control lógico sin eliminar datos
id_empleado	bigint		<input checked="" type="checkbox"/>	Identificador del empleado relativo al usuario del sistema. Puede ser nula si el usuario no está vinculado a un empleado.
fecha_creacion	datetime(6)	Sí		Fecha y hora de la creación del registro

Fuente: Elaboración propia, 2025.

Tabla 91

Diccionario de datos tabla auth user role

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
user_id	bigint	No	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Llave foránea de la tabla auth user
role_id	bigint	No	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Llave foránea de la tabla auth role

Fuente: Elaboración propia, 2025.

Tabla 92

Diccionario de datos tabla auth refresh token

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_refresh	bigint	No	<input checked="" type="checkbox"/>			Identificador único del token de refresh
user_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla auth user
token	varchar(512)	No				Cadena codificada del token de refresh
issued_at	datetime	No				Fecha y hora en que se emitió el token
expires_at	datetime(6)	No				Fecha y hora de expiración del token
revoked_at	datetime(6)	Sí				Fecha y hora en que fue revocado el token
revoked_by_ip	varchar(45)	Sí				Dirección IP desde la cual se revocó el token
replaced_by_token	varchar(512)	Sí				Token que reemplazo al actual

Fuente: Elaboración propia, 2025.

Tabla 93*Diccionario de datos tabla log evento*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_log	bigint	No	<input checked="" type="checkbox"/>			Identificador único del evento de auditoría
fecha_hora	timestamp	No				Fecha y hora en que ocurrió la acción
usuario	varchar(100)	Sí				Nombre del usuario que realizó la acción
accion	enum	No				Tipo de acción realizada (CREAR, ACTUALIZAR, ELIMINAR, etc.)
entidad	varchar(100)	Sí				Nombre de la entidad que se afectó con la acción
id_entidad	varchar(100)	Sí				Identificador de la entidad afectada
descripcion	text	Sí				Descripción general del evento
ip_origen	varchar(45)	Sí				Dirección IP desde la cual se ejecutó la acción
navegador	text	Sí				Identificación del dispositivo o navegador usado
detalles	text	Sí				Datos adicionales en formato estructurado
fecha_creacion	timestamp	Sí				Fecha de creación del registro en la base de datos
usuario_id	bigint	No		<input checked="" type="checkbox"/>		Llave foránea de la tabla auth user

Fuente: Elaboración propia, 2025.**Tabla 94***Diccionario de datos tabla tipo impuesto renta*

Columna	Tipo de Dato	NN	PK	FK	UK	Descripción
id_tipo_impuesto_renta	bigint	No	<input checked="" type="checkbox"/>			Llave primaria del tipo de impuesto.
porcentaje	decimal(5,2)	No				Porcentaje aplicado al impuesto.
salario_maximo	decimal(12,2)	Sí				Límite superior del rango salarial.
salario_mínimo	decimal(12,2)	No				Límite inferior del rango salarial.

Fuente: Elaboración propia, 2025.

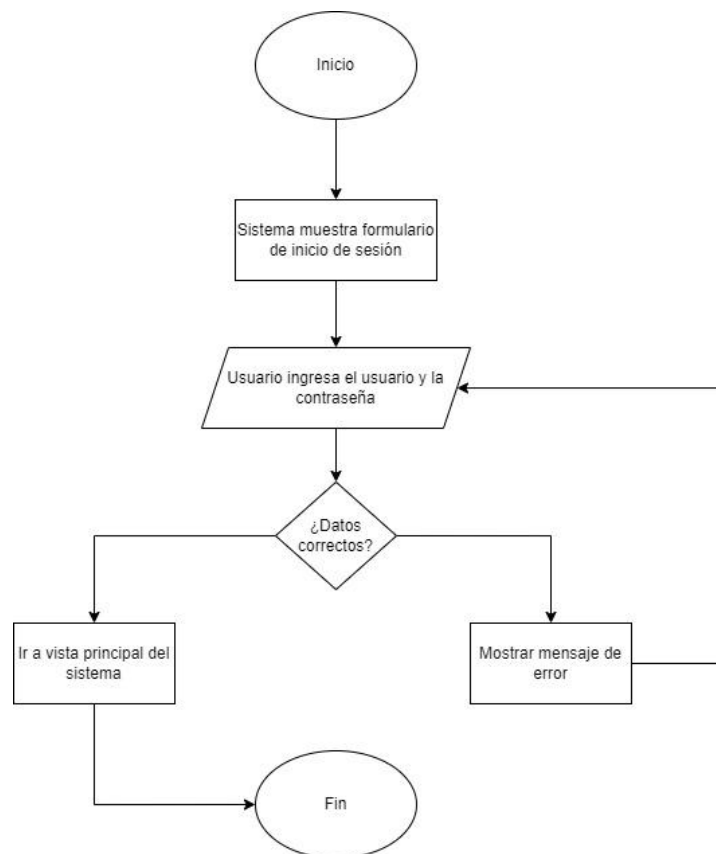
Diseño de procesos

En esta sección se presentan los diagramas de flujo en los que se describen los principales procesos del prototipo funcional de recursos humanos. Los diagramas permiten visualizar de forma clara y ordenada la secuencia de pasos que sigue el usuario dentro de cada módulo, así como las decisiones que el sistema toma durante los diferentes procesos.

Con estos diagramas se pretende facilitar la comprensión del comportamiento de los procesos del sistema, mostrando cómo se relacionan las acciones del usuario con las respuestas del programa de forma secuencial y ordenada. Estos procesos sirven como apoyo para el análisis y la documentación del desarrollo, y ayudan a identificar las entrada, procesos y salidas.

Figura 32

Diagrama de flujo inicio de sesión del sistema



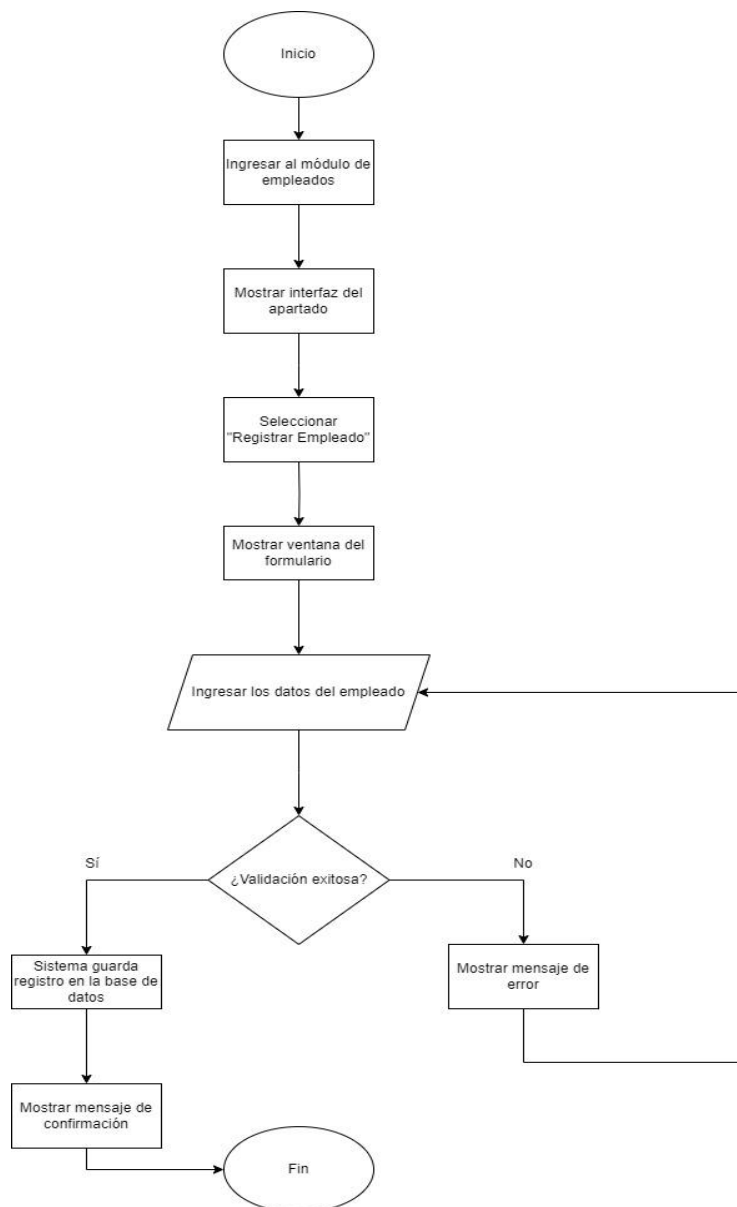
Fuente: Elaboración propia, 2025.

El diagrama de la Figura # muestra el proceso que le permite al usuario el acceso al sistema. El flujo se inicia con la visualización del formulario de inicio de sesión, al que el usuario ingresa su nombre de usuario y contraseña. Si los datos ingresados son correctos el

sistema permite el acceso a la vista principal, y en caso contrario se muestra un mensaje de error para indicar que las credenciales no son válidas.

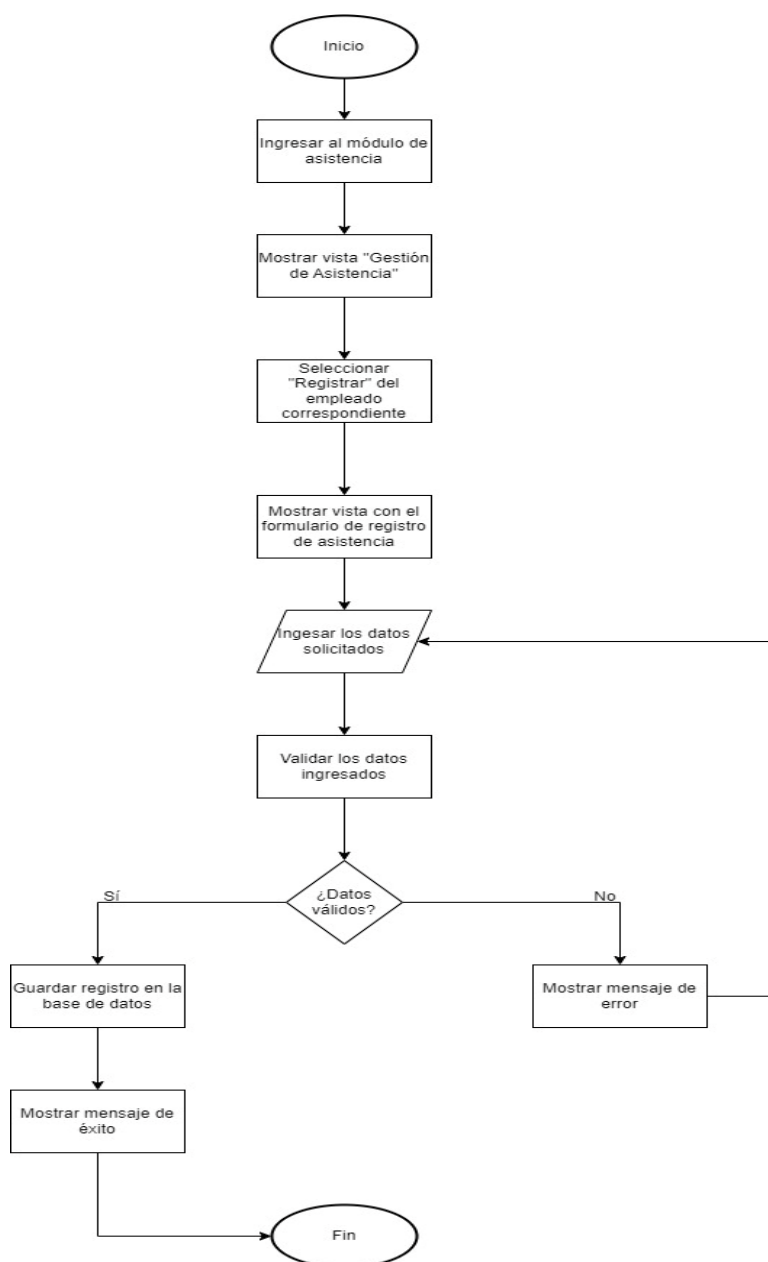
Figura 33

Diagrama de flujo registro de un empleado



Fuente: Elaboración propia, 2025.

En el diagrama se representa el proceso que sigue el sistema para registrar un nuevo empleado. El flujo se inicia cuando el usuario accede al módulo de empleados y selecciona la opción “Registrar empleado”. A continuación se muestra el formulario correspondiente para ingresar los datos requeridos. Una vez completada la información el sistema valida los datos ingresados por el usuario. Si la validación es exitosa el registro se guarda en la base de datos y se muestra un mensaje de confirmación. En caso contrario el sistema genera un mensaje de error y solicita corregir la información antes de finalizar el proceso.

Figura 34*Diagrama de flujo registrar una asistencia*

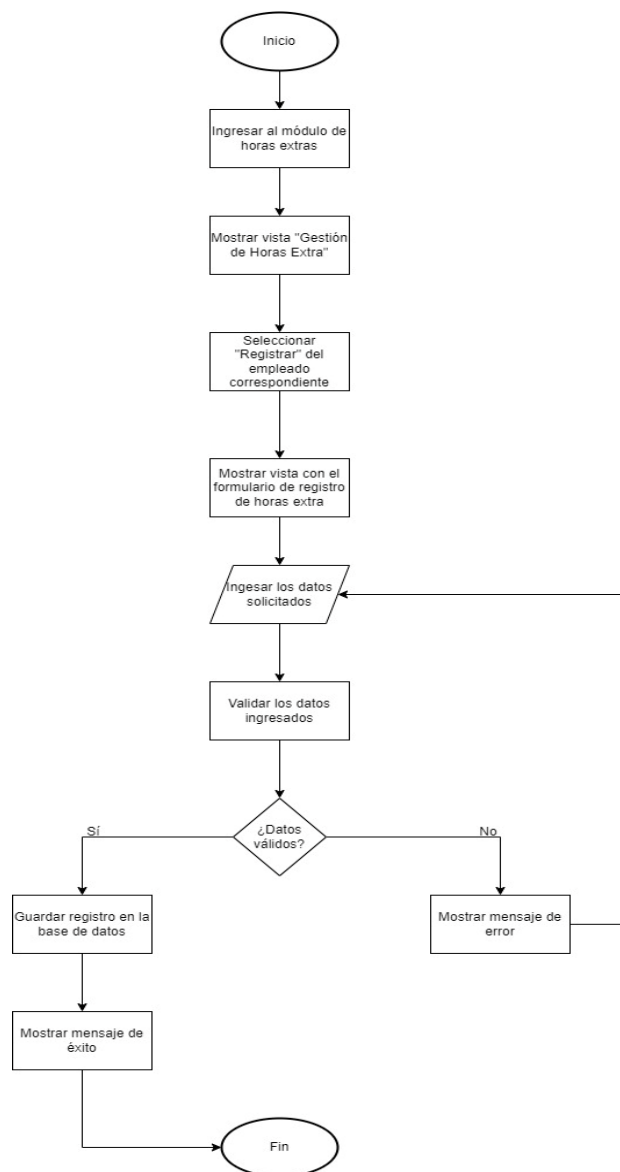
Fuente: Elaboración propia, 2025.

En el diagrama se describe el proceso mediante el cual el usuario de recursos humanos registra la asistencia diaria de los empleados. El flujo se inicia con el ingreso al módulo de asistencia, en donde se muestra la vista principal del módulo. Posteriormente, se selecciona “Registrar” del empleado correspondiente y se despliega el formulario para ingresar la información requerida. Una vez que se ingresan todos los datos requeridos por el formulario el sistema realiza la validación. Si la información es válida se guarda el registro en la base de

datos y se muestra un mensaje de éxito. En caso contrario se notifica el usuario mediante un mensaje de error, y se solicita la corrección de los datos antes de finalizar el proceso.

Figura 35

Diagrama de flujo registrar horas extra



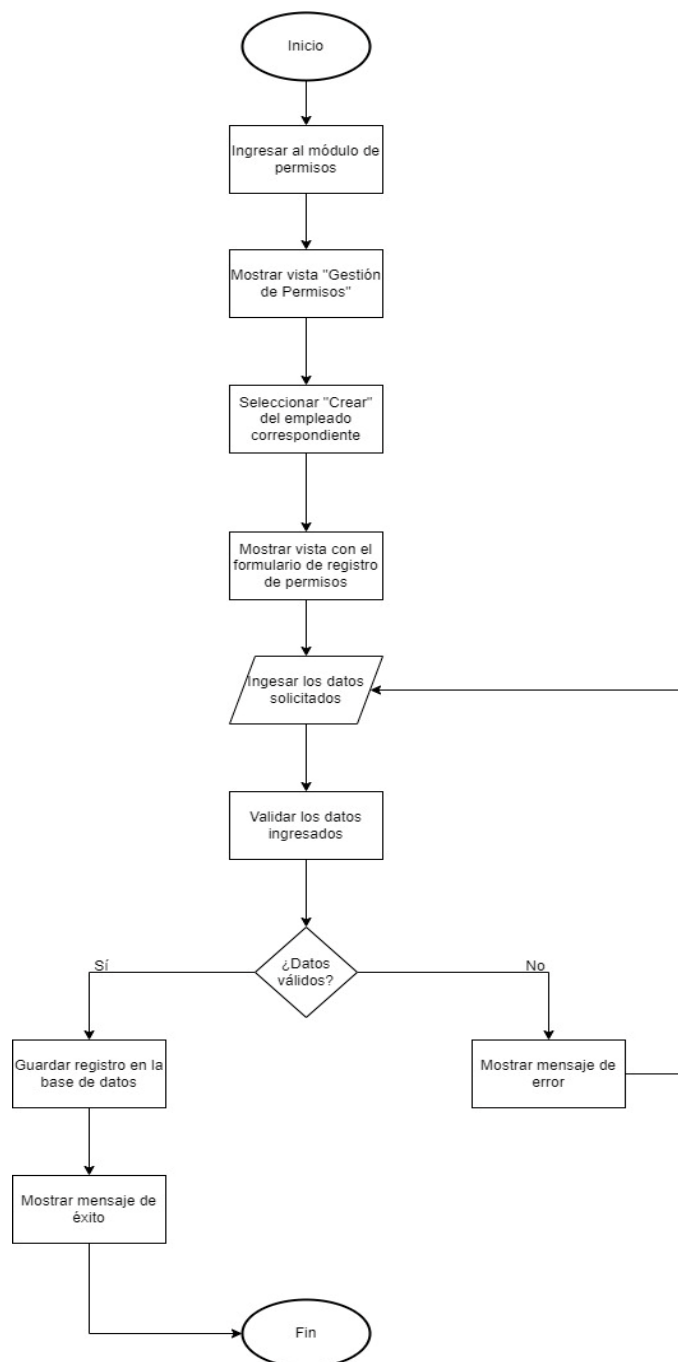
Fuente: Elaboración propia, 2025.

En el diagrama se ilustra el proceso que permite registrar horas extra trabajadas por un empleado. El flujo se inicia con el ingreso al módulo de horas extra, en que se muestra la vista principal del módulo. Luego, el usuario selecciona “Registrar” correspondiente al empleado que se requiere ingresar al registro, y se despliega el formulario para ingresar los datos solicitados. Una vez completados los campos que el formulario solicita el sistema valida la información ingresada. Si los datos son válidos se guarda el registro en la base de datos y se

muestra un mensaje de éxito. En caso contrario el sistema despliega un mensaje de error, lo que le permite al usuario corregir la información.

Figura 36

Diagrama de flujo Registro de permisos



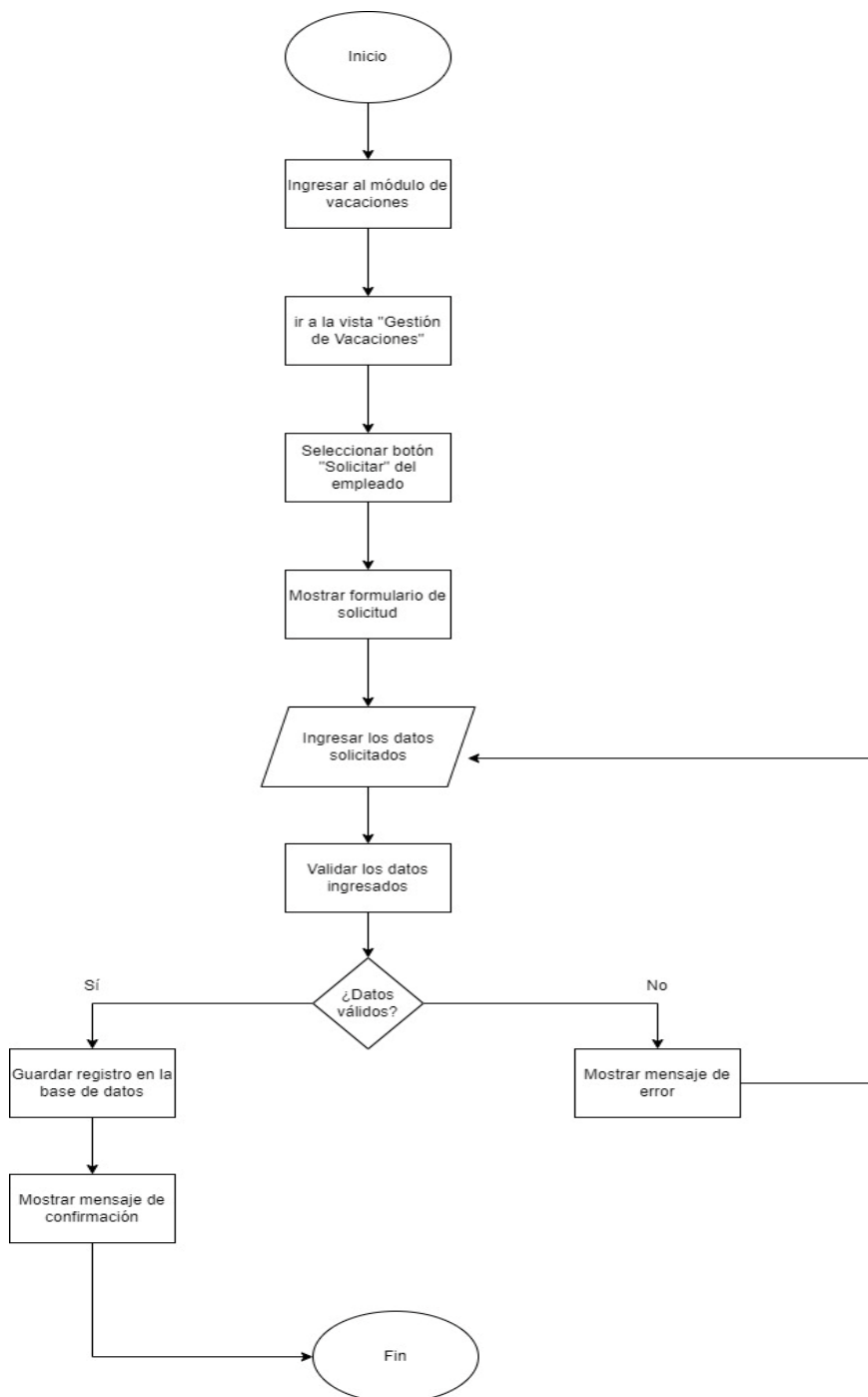
Fuente: Elaboración propia, 2025.

El diagrama muestra el proceso que le sigue al sistema para registrar el permiso de un empleado. El flujo se inicia al ingresar al módulo de permisos, en que selecciona la opción “Crear” para el empleado correspondiente. Luego el usuario completa el formulario con los datos requeridos y el sistema valida la información ingresada. Si los datos son correctos el

registro se guarda en la base de datos y se muestra un mensaje de éxito. De lo contrario, se presenta un mensaje de error.

Figura 37

Diagrama de flujo Registro de solicitud de vacaciones



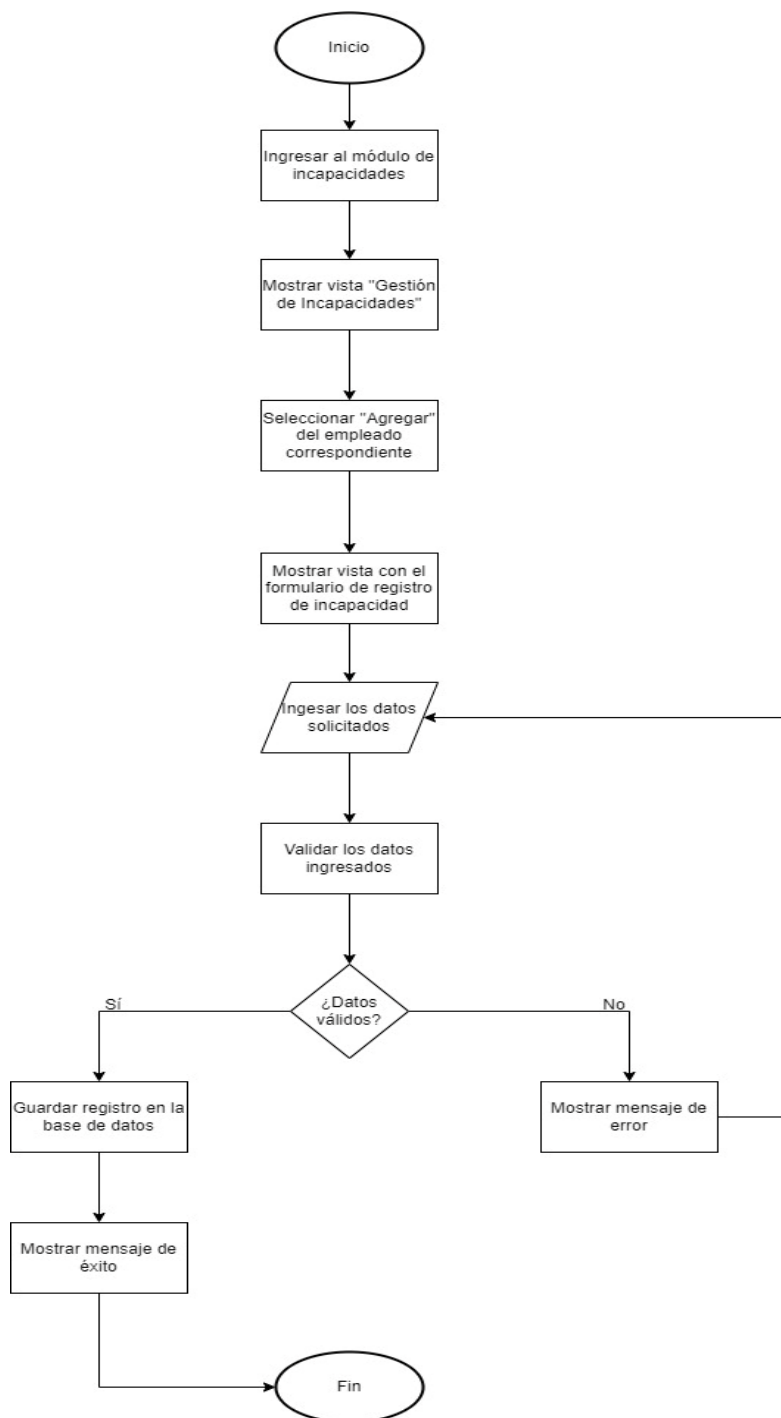
Fuente: Elaboración propia, 2025.

En el diagrama se muestra el proceso que realiza el usuario para registrar la solicitud de vacaciones. El flujo se inicia con el ingreso del módulo de vacaciones y con la selección de “Solicitar” del empleado. Luego, el sistema despliega el formulario correspondiente para

ingresar los datos requeridos. Si la información es válida el sistema guarda el registro en la base de datos y muestra un mensaje de éxito; de lo contrario se presenta un mensaje de error.

Figura 38

Diagrama de flujo Registro de incapacidades



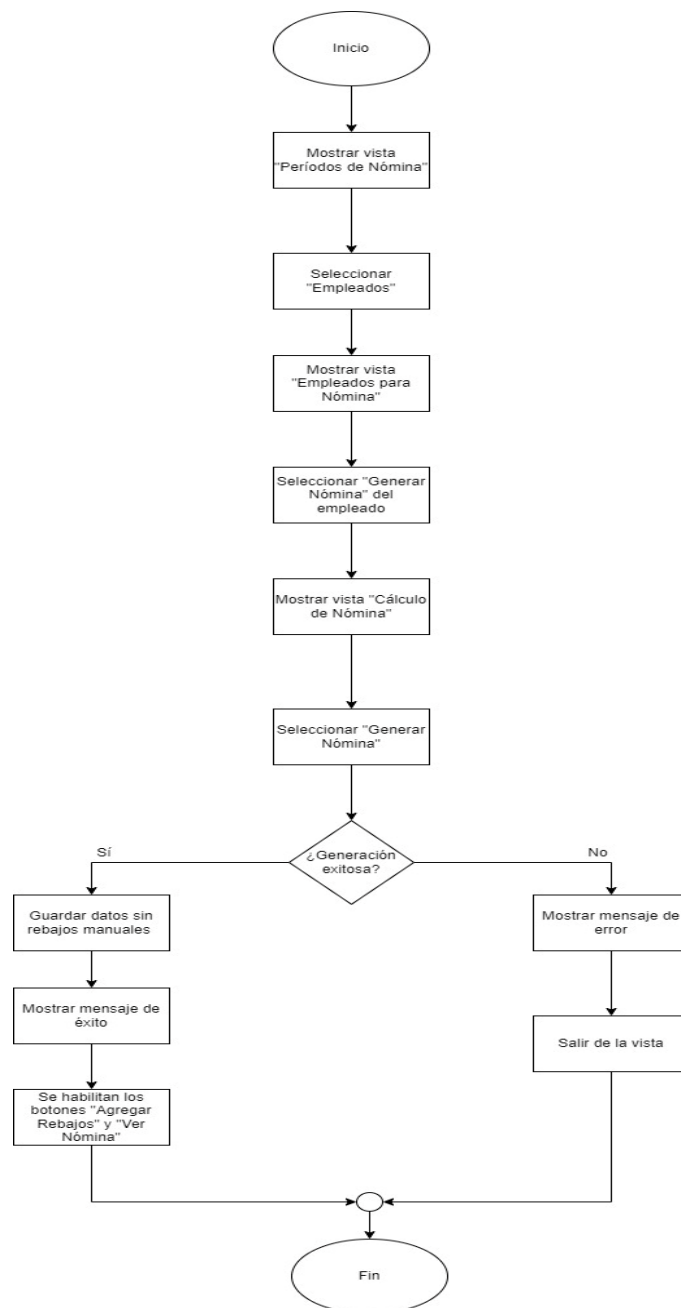
Fuente: Elaboración propia, 2025.

El diagrama representa el proceso para registrar una incapacidad de un empleado en el sistema. El flujo se inicia con el ingreso al módulo de incapacidades, en que el usuario selecciona la opción “Agregar” del empleado correspondiente. Luego muestra el formulario

para ingresar la información y, si es correcta, guarda el registro en la base de datos y muestra un mensaje de éxito; de lo contrario se presenta un mensaje de error.

Figura 39

Diagrama de flujo cálculo de nómina



Fuente: Elaboración propia, 2025.

El diagrama ilustra el proceso mediante el cual el sistema genera la nómina de un empleado. El flujo se inicia con la visualización de los periodos de nómina y la selección de “Empleados”. Luego se selecciona “Generar nómina” del empleado al que debe calcular la nómina. Luego, el usuario ingresa a la vista de cálculo y selecciona “Generar nómina”. Si la generación es exitosa, los datos persisten en la base de datos y se muestra un mensaje de éxito,

y se habilitan las opciones de agregar rebajos o visualizar la nómina generada. En caso de error se muestra un mensaje y el proceso finaliza.

Diseño de salidas

En este apartado se presentan los principales formatos de salida del sistema, los cuales muestran la información procesada y los resultados generados a partir de los datos almacenados. Estas salidas le permiten al usuario visualizar de manera clara y ordenada los registros y reportes del sistema, lo que facilita la toma de decisiones dentro del área de recursos humanos. Cada formato fue diseñado y adaptado de acuerdo con las necesidades de la empresa, lo cual garantiza que la información mostrada sea precisa y útil para la gestión administrativa.

Figura 40

Diseño de salida datos personales del empleado

The screenshot shows a web application interface for HR management. On the left is a dark sidebar with the logo 'RRHH' and 'Sistema de Gestión'. Below it are menu items: 'Dashboard', 'Empleados', 'Personas', 'Contacto Emergencia', 'Formación Académica', 'Condición Contractual', 'Asistencia', and 'Horas Extras'. The main content area is titled 'Datos Personales' and shows the profile of 'Kendall Josue Torres Loria' with an 'Activo' status. Below the name is a table with the following data:

Identificación	Correo	Fecha Nacimiento	Estado Civil	Género	Teléfono Fijo	Celulares	Dirección
Cédula: 402500777	kjosut85@gmail.com	14/06/1993	Casado	Masculino		88414433 71524872 Principal	Heredia, San Pablo, Rincón de Sabanilla De la plaza de deportes, 100 metros este

At the bottom right of the table, there are two action buttons: 'Editar' (with a pencil icon) and 'Eliminar' (with a trash icon).

Fuente: Elaboración propia, 2025.

Esta vista muestra de manera clara y organizada la información personal del empleado seleccionado dentro del sistema de gestión de recursos humanos. Esto permite visualizar datos como identificación, correo electrónico, fecha de nacimiento, estado civil, género, teléfonos y dirección. Además, opciones de edición o eliminación, lo cual facilita la administración ágil y segura de los registros personales dentro del sistema.

Figura 41

Diseño de salida historial laboral de un empleado

Historial Laboral

Registros laborales del empleado

Empleado: Kendall Josue Torres Loria

Puesto	Tipo Jornada	Hora Inicio	Hora Fin	Horas Diarias	Días por Semana	Total Semanal	Recargo	Salario por Hora	Fecha Inicio	Fecha Fin	Acciones
Maestro de obra	Jornada Sam a 5pm	05:00	17:00	12.00	5	48	N/A	€2,500.00	06/10/2025	Activo	Cerrar historial
Soldador	Jornada Sam a 5pm	05:00	17:00	12.00	5	48	N/A	€2,000.00	24/09/2025	25/09/2025	
Soldador calificado	Jornada Sam a 5pm	05:00	17:00	12.00	5	48	N/A	€2,150.00	25/09/2024	04/10/2025	

[← Regresar](#)

Fuente: Elaboración propia, 2025.

Esta vista presenta el historial laboral del empleado y muestra de forma estructurada los diferentes puestos que ha ocupado dentro de la empresa. Incluye información relevante como tipo de jornada, horarios, días laborados por semana, salario por hora, fechas de inicio y finalización, así como el estado actual del registro. Además, incorpora opciones como el cierre del historial. Esta vista facilita la consulta rápida y el seguimiento de los cambios en las condiciones laborales de cada colaborador.

Figura 42

Diseño de salida Registro de asistencia semanal

Asistencia Semanal

Fecha de referencia: 06/10/2025 [Buscar semana](#)

Empleado: Kendall Josue Torres Loria
Semana: 2025-10-06 a 2025-10-12

[+ Agregar asistencia](#)

Fecha	Hora Entrada	Hora Salida	Estado	Observaciones	Acciones
2025-10-06	06:00	14:00	Completada		Editar
2025-10-07	06:00	14:00	Completada		Editar
2025-10-08	06:00	14:00	Completada		Editar
2025-10-09	06:00	13:30	Completada	Salio mas temprano por cita medica	Editar
2025-10-10	06:00	14:00	Completada		Editar

[← Regresar](#)

Fuente: Elaboración propia, 2025.

Esta vista corresponde al registro de asistencia semanal del empleado, en la que se detalla la información diaria de las entradas y salidas durante la semana específica. Permite consultar de manera clara el estado de cada jornada, agregar nuevas asistencias o editar las existentes, así como registrar observaciones relevantes sobre la jornada laboral. Su diseño facilita el seguimiento y control del registro de asistencia de los empleados, y ofrece al usuario una herramienta práctica para la gestión del control de asistencia del sistema.

Figura 43

Diseño de salida Historial de permisos

Historial de Permisos

Permisos registrados del empleado

Todos

Tipo	Inicio	Fin	Estado	Motivo	Creado
Permiso sin goce salarial	2025-10-10	2025-10-10	PENDIENTE	Personales	30/09/2025 12:52
Sin goce salarial	2025-10-10	2025-10-10	RECHAZADO	Personales	30/09/2025 12:56
Personal	2025-10-17	2025-10-17	RECHAZADO		04/10/2025 08:42
Personal	2025-10-28	2025-10-28	PENDIENTE	Cita medica	24/10/2025 21:15

Fuente: Elaboración propia, 2025.

Esta vista muestra el historial de permisos registrados de cada empleado, lo cual permite visualizar de forma ordenada los diferentes tipos de permisos solicitados, junto con sus fechas de inicio y fin, estado actual, motivo y fecha de creación. Permite llevar un control claro y actualizado de las solicitudes, aprobaciones y rechazos de permisos, con lo cual se garantiza una gestión eficiente.

Figura 44*Diseño de salida Incapacidades registradas*

RRHH

Sistema de Gestión

Dashboard

Empleados

Asistencia

Horas Extras

Permisos

Vacaciones

Incapacidades

Nómina

Liquidación

Aguinaldos

Reportes

Mantenimiento

Administrador RRHH

Historial de Incapacidades

Incapacidades registradas del empleado

Todos dd/mm/aaaa dd/mm/aaaa Filtrar

Tipo	Inicio	Fin	Días	Creado	Acciones
CCSS	2025-10-08	2025-10-10	3	09/10/2025 10:19	Editar

Editar incapacidad más reciente

Regresar

Fuente: Elaboración propia, 2025.

Esta vista corresponde al historial de incapacidades del empleado en la que se presenta un listado detallado de las incapacidades otorgadas, incluidos el tipo, la fecha de inicio y finalización, la cantidad de días y la fecha de registro. Su diseño le permite al administrador mantener un control preciso de las incapacidades gestionadas, lo que asegura una trazabilidad clara y correcta de la información médica de los empleados.

Figura 45

Diseño de salida Comprobante de pago



Fuente: Elaboración propia, 2025.

Esta vista representa el comprobante de pago del empleado, en el cual se detalla la información correspondiente al periodo laboral seleccionado. Incluye los datos generales del colaborador, su puesto, tipo de jornada y fecha de ingreso, así como el desglose de ingresos, deducciones y cálculo del salario neto.

Diagramas UML

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) constituye una herramienta esencial dentro del desarrollo orientado a objetos, y permite representar de forma visual la estructura y el comportamiento del sistema. Su utilización facilita la comprensión del diseño, la comunicación entre los desarrolladores, entre otros aspectos que resultan útiles a la hora de diseñar software de calidad.

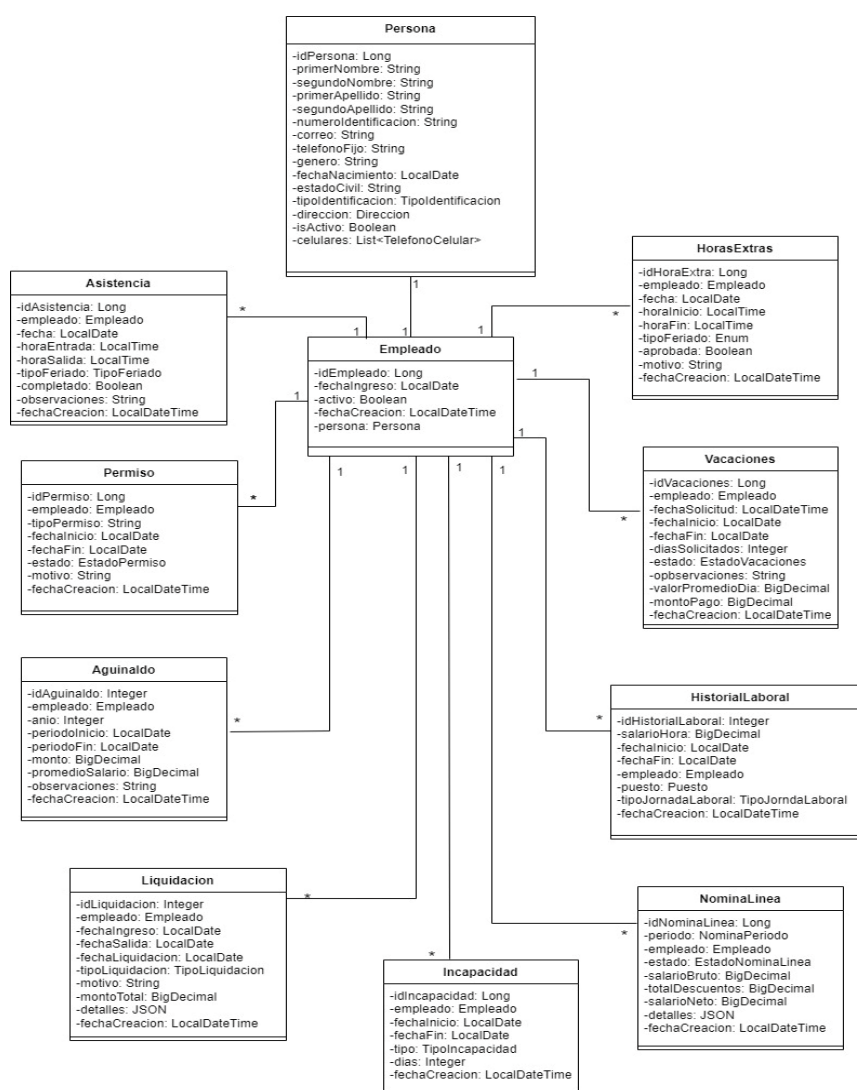
Para el sistema de gestión de recursos humanos se emplean los diagramas UML más relevantes, para reflejar los componentes principales y la interacción entre ellos. En particular,

se presentan el diagrama de clases, en los que se describen la estructura estática del sistema y las relaciones entre las entidades que lo conforman, y el diagrama de secuencia, que representa el flujo dinámico de mensajes entre los objetos durante la ejecución de determinados procesos clave.

Estos diagramas permiten visualizar de manera clara la arquitectura interna del sistema, el comportamiento de sus módulos y la lógica de interacción entre sus componentes, para asegurar la coherencia entre el diseño y el desarrollo final del prototipo.

Figura 46

Diagrama de clase del prototipo de recursos humanos



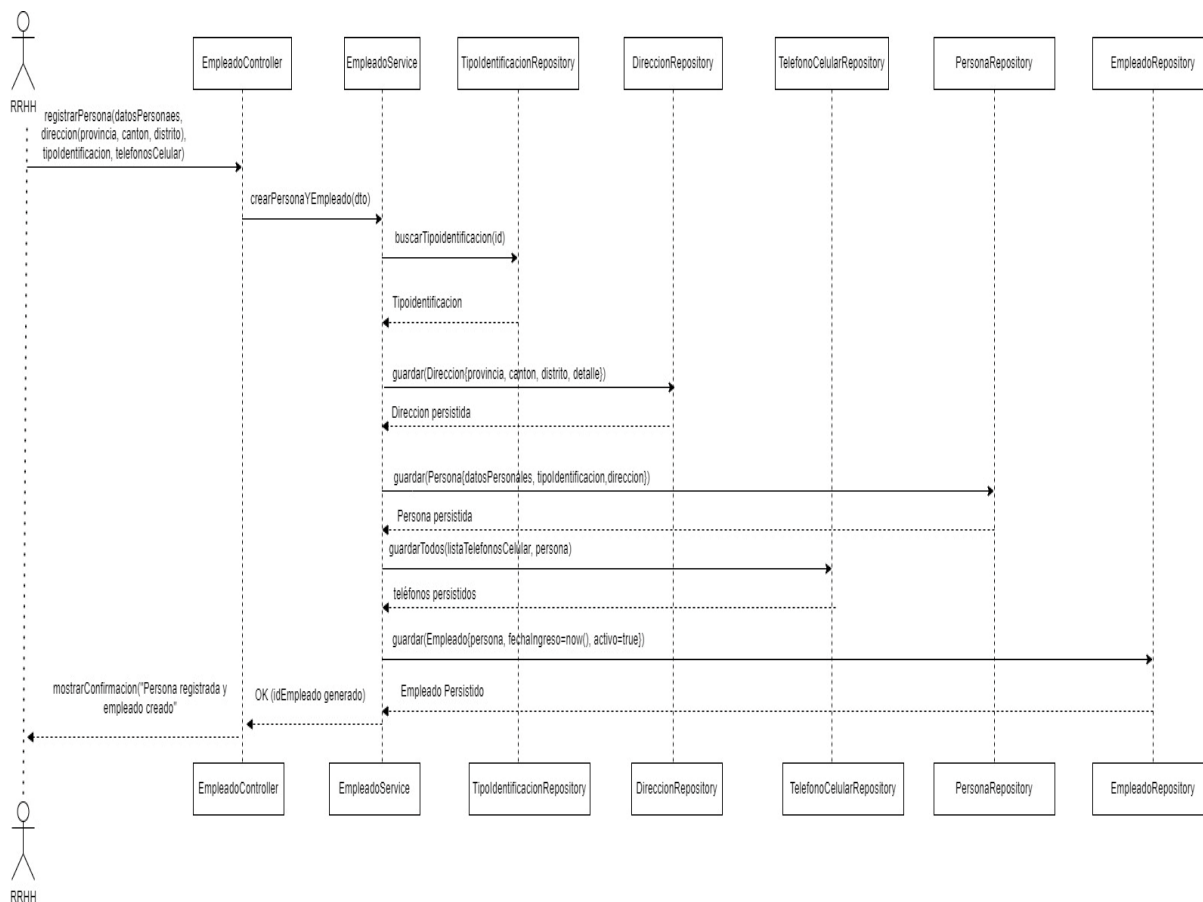
Fuente: Elaboración propia, 2025.

El diagrama de clases presentado refleja de manera detallada la estructura interna del sistema, muestra todas a las entidades que conforman el modelo de datos, sus atributos y relaciones existentes entre ellas. Esta representación permite comprender la organización

lógica del sistema y la forma en la cual las distintas clases interactúan para cumplir con las funciones de gestión de información dentro del módulo de Recursos humanos.

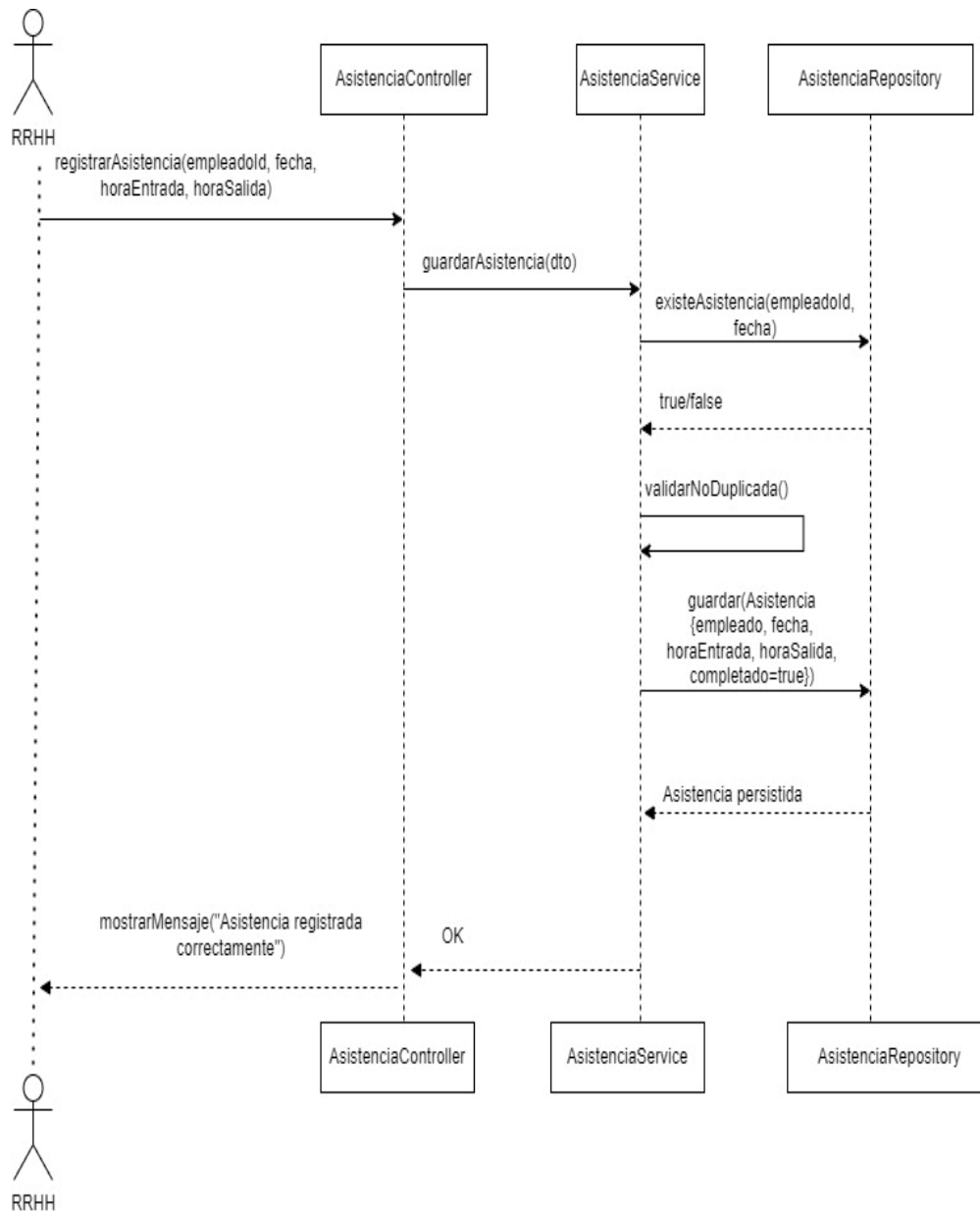
Figura 47

Diagrama de secuencia Crear un empleado



Fuente: Elaboración propia, 2025.

El diagrama de secuencia ilustra el proceso completo de registro de los datos personales de un empleado dentro del sistema. En este flujo el área de Recursos Humanos ingresa la información general, selecciona los catálogos de provincia, cantón y distrito, y el tipo de identificación y el sistema guarda los datos en sus respectivas tablas. Finalmente, se genera automáticamente el registro del empleado, el cual garantiza la integridad y la relación entre las distintas entidades del modelo.

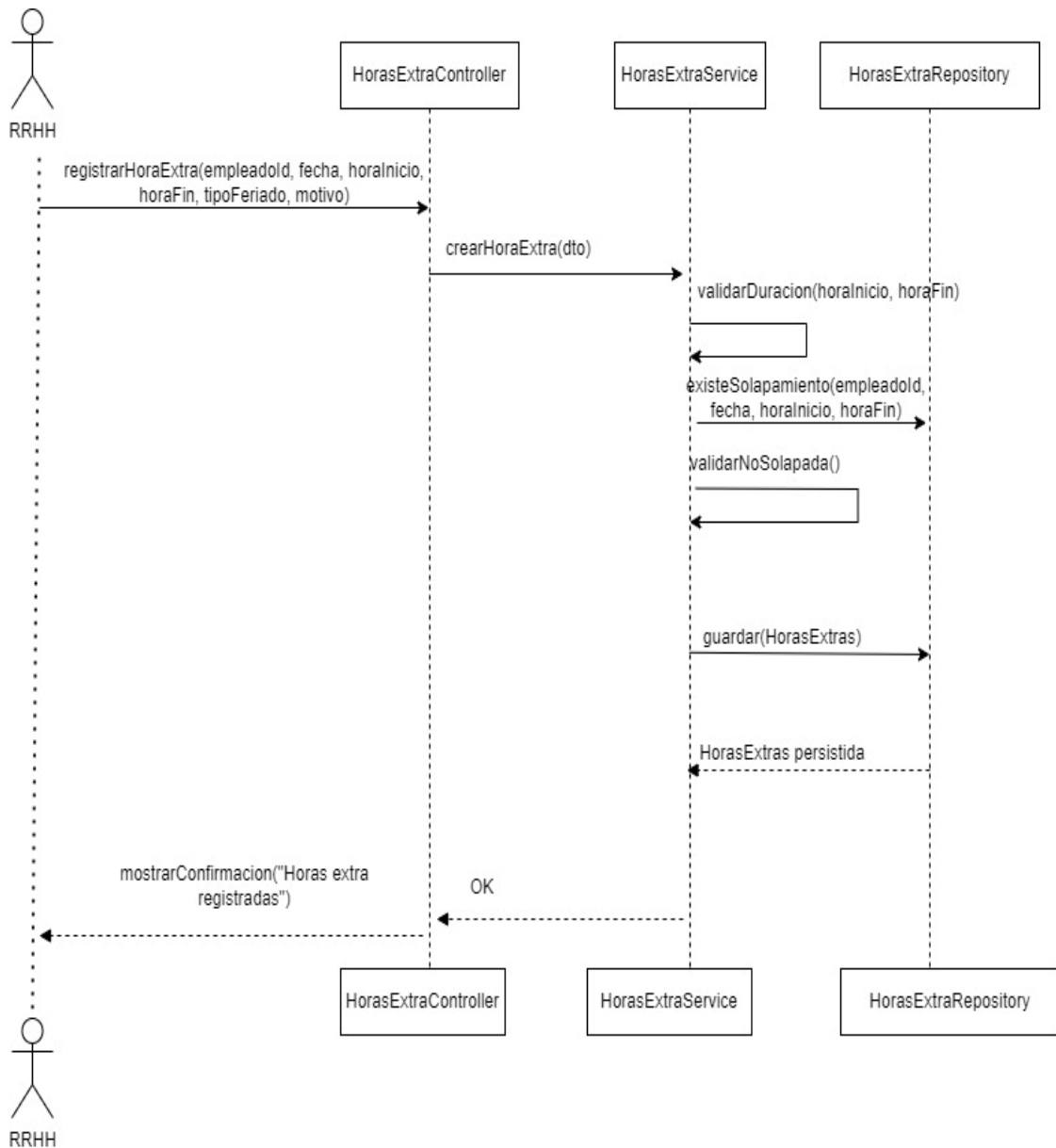
Figura 48*Diagrama de secuencia Registrar asistencia*

Fuente: Elaboración propia, 2025.

El diagrama de secuencia muestra el proceso mediante el cual el área de Recursos Humanos registra la asistencia de un empleado en una fecha específica. El sistema verifica que no exista un registro previo para el mismo día, a fin de evitar duplicidades, y posteriormente almacena la información correspondiente a la entrada y salida laboral.

Figura 49

Diagrama de secuencia para registrar horas extra

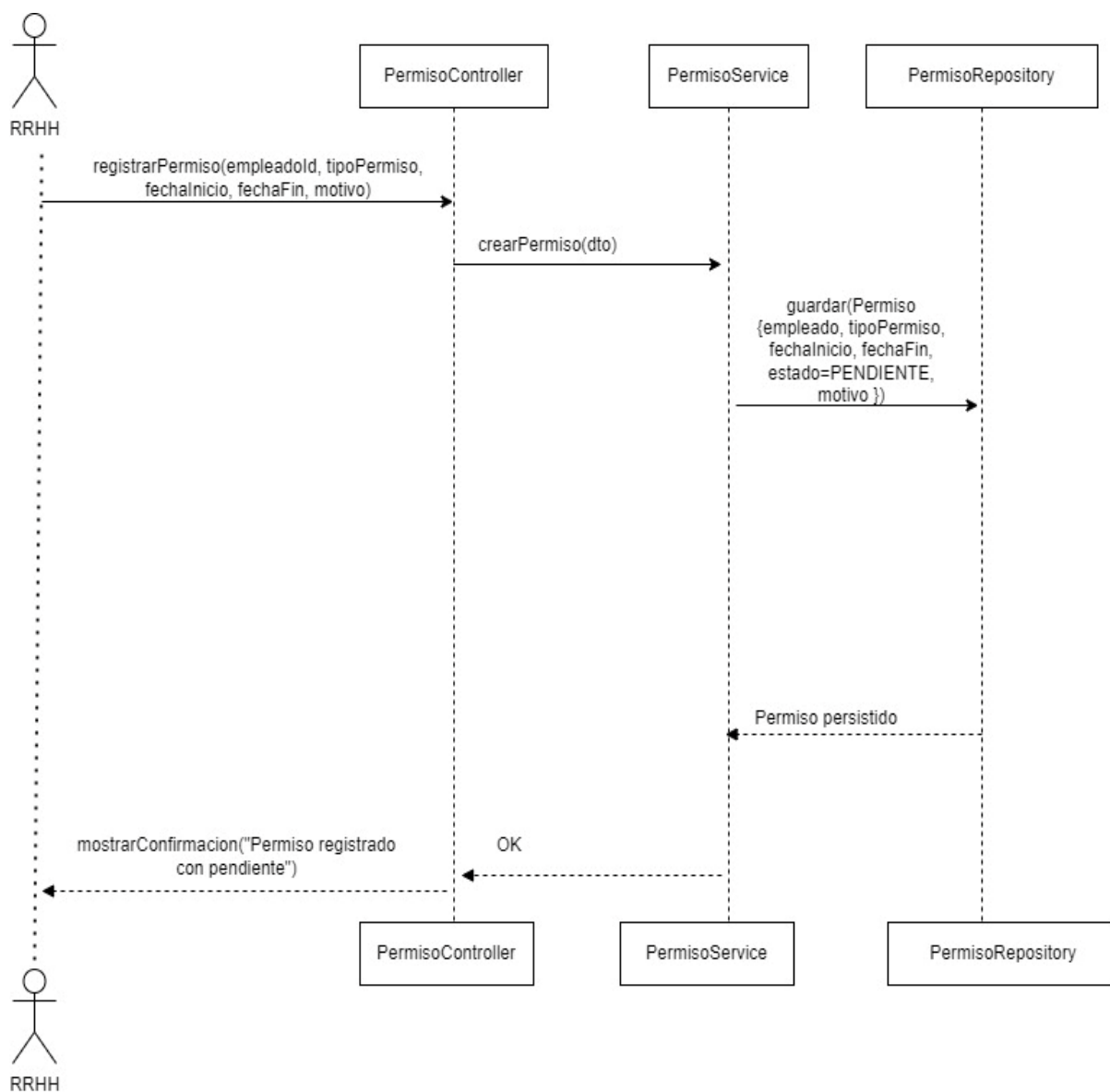


Fuente: Elaboración propia, 2025.

El diagrama de secuencia representa el proceso mediante el cual el área de Recursos Humanos registra las horas extra trabajadas por un empleado. El sistema valida que el rango de horario sea correcto, comprueba que no exista solapamiento con otros registros de horas extra en la misma fecha y, una vez verificados los datos, guarda la información correspondiente.

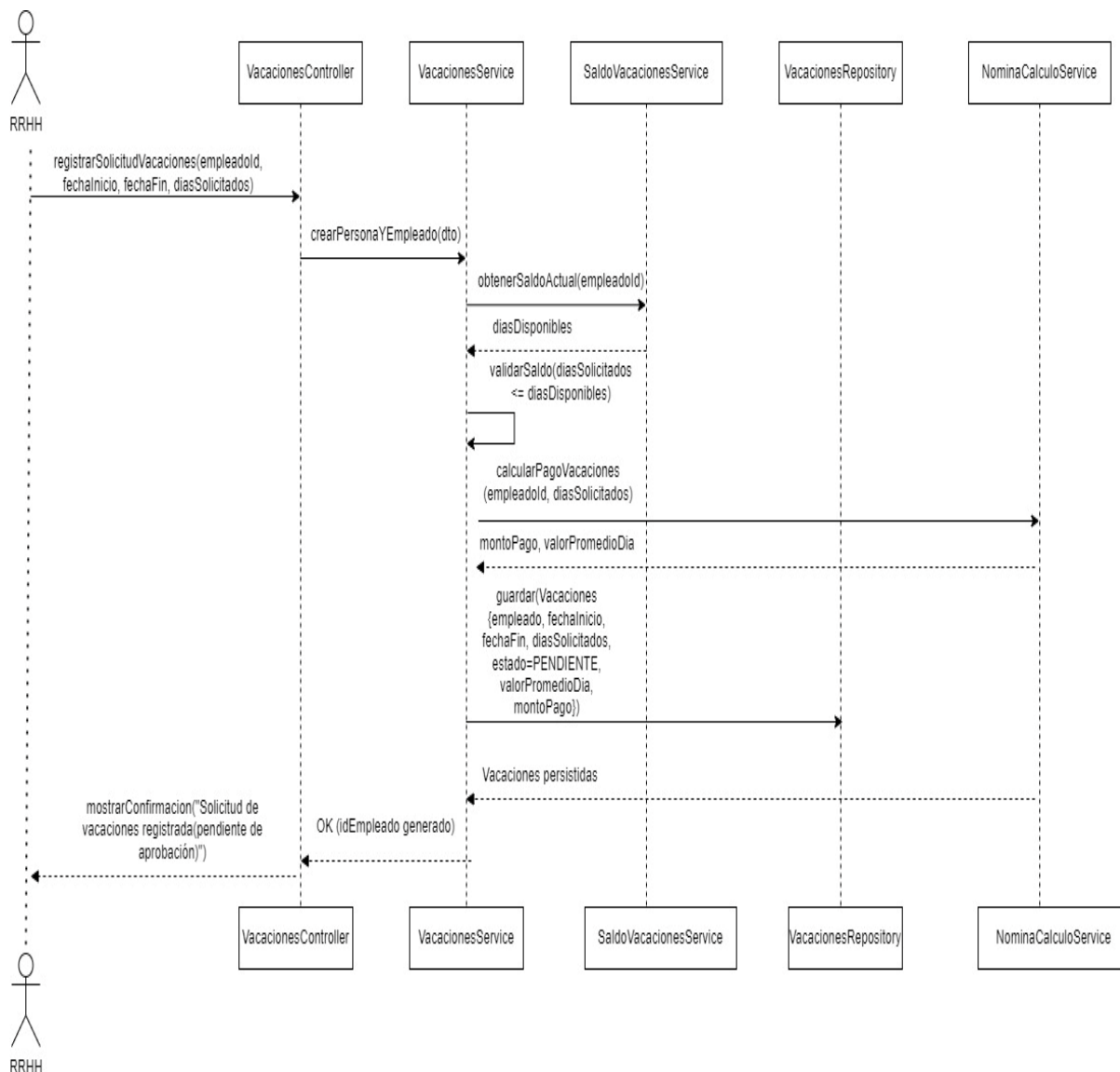
Figura 50

Diagrama de secuencia para registrar un permiso



Fuente: Elaboración propia, 2025.

El diagrama de secuencia ilustra el proceso mediante el cual el área de Recursos Humanos registra el permiso para un empleado, con especificación del tipo, las fechas y el motivo correspondiente. El sistema crea el registro con un estado inicial de “Pendiente”, lo que permite su posterior aprobación o rechazo según la decisión tomada por el encargado del Departamento de Recursos Humanos.

Figura 51*Diagrama de secuencia registrar solicitud de vacaciones*

Fuente: Elaboración propia, 2025.

El diagrama de secuencia representa el proceso para registrar una solicitud de vacaciones de un empleado. El sistema consulta el saldo disponible de días, valida que la cantidad solicitada sea correcta y calcula el monto por pagar según el salario promedio del trabajador. Finalmente, la solicitud se guarda como pendiente de forma automática o de acuerdo con el estado que el encargado seleccione dentro de las opciones.

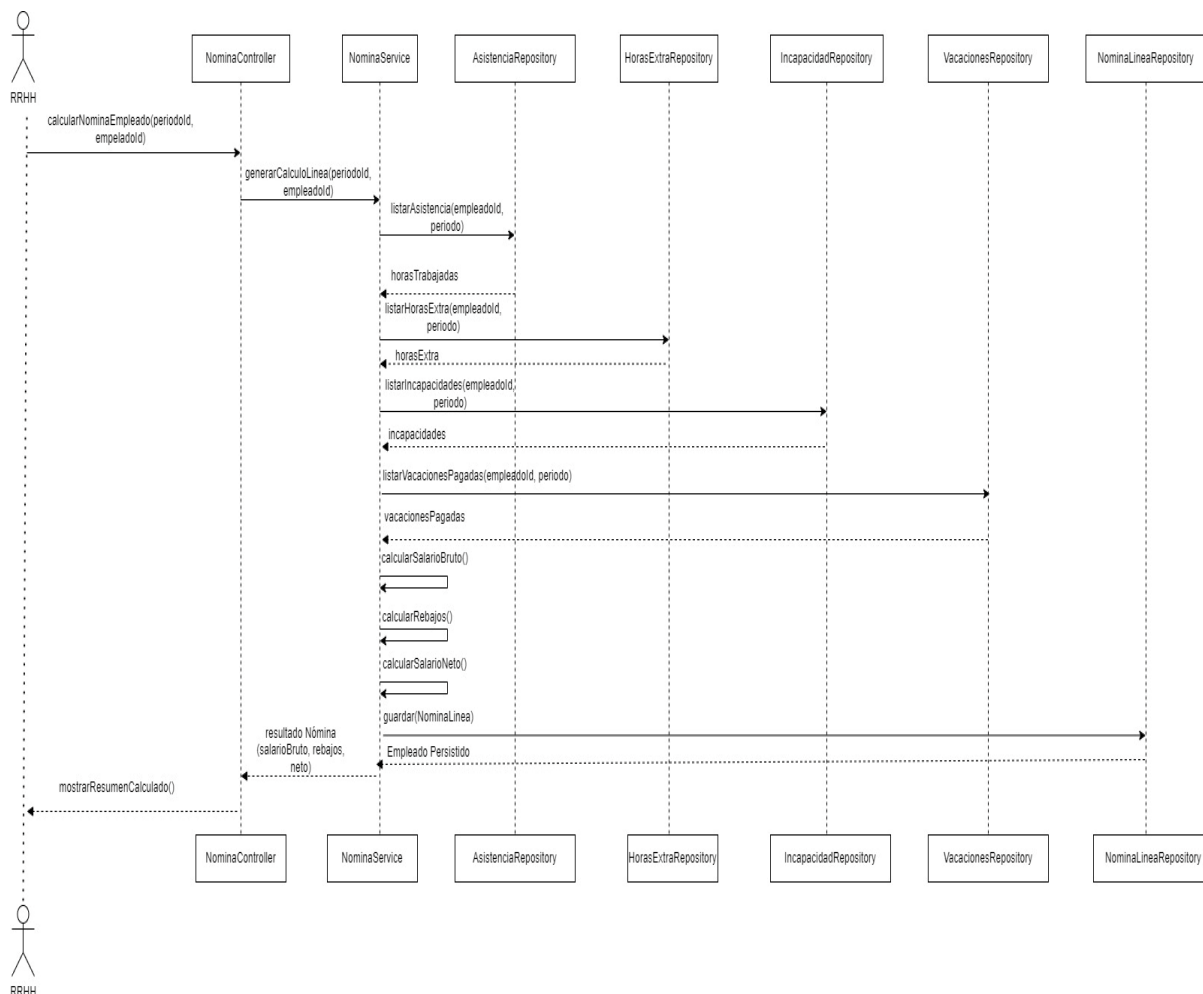
Figura 52*Diagrama de secuencia para cálculo de nómina*

Figura: Elaboración propia, 2025.

El diagrama de secuencia muestra el proceso completo de cálculo de nómina para un empleado dentro de un período determinado. El sistema recopila la información proveniente de los registros de asistencia, horas extra, incapacidades y vacaciones pagadas, con el fin de calcular el salario bruto, los rebajos correspondientes y el salario neto. Finalmente, la información se almacena en la entidad de línea de nómina.

Programación

En esta sección se presentan fragmentos representativos del código fuente desarrollado para el sistema de gestión de recursos humanos. Estos fragmentos de código corresponden a diferentes componentes del desarrollo, tales como la captura de entradas, la generación de

salidas, el manejo de procesos internos, la aplicación de validaciones y la estructura de los módulos definidos en el alcance del proyecto.

De esa manera se evidencia la correcta aplicación de las buenas prácticas de programación, la utilización del framework Spring Boot y las tecnologías complementarias empleadas para garantizar un sistema funcional seguro y mantenible.

Entradas

En este apartado se presentan ejemplos de código relacionados con la captura de datos dentro del sistema, como los formularios utilizados para registrar información en los distintos módulos. Estos fragmentos ilustran cómo se gestionan las entradas mediante formularios diseñados con Thymeleaf y válidos en el backend con Spring boot.

Figura 53

Agregar Asistencia desde el controlador

```

@PostMapping("/guardar")
public String guardarAsistencia(@ModelAttribute("asistencia") @Valid AsistenciaDTO dto, BindingResult bindingResult,
                                RedirectAttributes redirect, Model model) {
    if (bindingResult.hasErrors()) {
        Empleado empleado = empleadoRepository.findById(dto.getEmpleadoId()).orElse(other: null);

        if (empleado != null) {
            PersonaResponseDTO empleadoDTO = personaMapper.toResponseDTO(empleado.getPersona());
            model.addAttribute("empleado", empleadoDTO);
        }

        model.addAttribute("title", "Registrar Asistencia");
        model.addAttribute("content", "asistencia/formularioAsistencia");
        model.addAttribute("tiposFeriado", TipoFeriado.values());
        return "fragments/layout";
    }

    try {
        asistenciaService.registrarAsistencia(dto);
        redirect.addFlashAttribute("success", "Asistencia registrada correctamente");
    } catch (BussinesException | IllegalArgumentException ex) {
        redirect.addFlashAttribute("error", ex.getMessage());
        return "redirect:/rh/asistencia/registrar?empleadoId=" + dto.getEmpleadoId();
    }

    return "redirect:/rh/asistencia/listar-empleados/asistencia";
}

```

Fuente: Elaboración propia, 2025.

Este método pertenece al controlador encargado de registrar las asistencias de los empleados. En él se procesan los datos enviados desde el formulario de registro, aplicando validaciones sobre la información recibida y gestionando los posibles errores antes de guardar la asistencia en la base de datos.

Figura 54*Entrada de datos crear un periodo de nomina*

```

@Transactional
@Override
public NominaPeriodoDTO crear(NominaPeriodoRequestDTO requestDTO) {
    if(requestDTO.getFechaFin() == null){
        requestDTO.setFechaFin(calcularFechaFinDesdeInicio(requestDTO.getFechaInicio()));
    }

    if(requestDTO.getFechaInicio().isAfter(requestDTO.getFechaFin())){
        throw new IllegalArgumentException(s: "La fecha de inicio no puede ser posterior a la fecha fin");
    }

    if(validarSolapamiento(requestDTO.getFechaInicio(), requestDTO.getFechaFin())){
        throw new IllegalArgumentException(s: "El periodo de nómina se solapa con uno existente.");
    }

    NominaPeriodo nuevoPeriodo = NominaPeriodoMapper.toEntity(requestDTO);
    nuevoPeriodo.setFechaCreacion(LocalDateTime.now());

    NominaPeriodo guardado = nominaPeriodoRepository.save(nuevoPeriodo);

    return NominaPeriodoMapper.toDTO(guardado);
}

```

Fuente: Elaboración propia, 2025.

Este método se encarga de crear un nuevo período de nómina a partir de los datos ingresados por el usuario. Ante de almacenar la información valida que las fechas sean coherentes y que no existan solapamientos con otros periodos registrados. Finalmente, guarda el registro en la base de datos dentro de una transacción para garantizar la integridad de la operación.

Figura 55*Entrada para actualizar una asistencia*

```

@Transactional
@Override
public Asistencia actualizarAsistencia(Long id, AsistenciaDTO dto) {
    Asistencia existente = asistenciaRepository.findById(id)
        .orElseThrow(() -> new BussinesException(message: "Asistencia no encontrada"));

    if (dto.getEmpleadoId() != null) {
        Empleado empleado = empleadoRepository.findById(dto.getEmpleadoId())
            .orElseThrow(() -> new BussinesException(message: "Empleado no encontrado"));
        existente.setEmpleado(empleado);
    }

    existente.setFecha(dto.getFecha());
    existente.setHoraEntrada(dto.getHoraEntrada());
    existente.setHoraSalida(dto.getHoraSalida());
    existente.setObservaciones(dto.getObservaciones());
    existente.setCompletado(dto.getHoraEntrada() != null && dto.getHoraSalida() != null);
    existente.setTipoFeriado(dto.getTipoFeriado());

    validarDatos(dto);
    validarDuracion(dto);
    validarLimiteDiario(dto);
    validarLimiteSemanal(dto);

    return asistenciaRepository.save(existente);
}

```

Fuente: Elaboración propia, 2025.

Este método permite actualizar los datos de una asistencia previamente registrada. Valida la existencia del registro y del empleado asociado, aplica verificaciones sobre la información ingresada y actualiza los campos correspondientes. Su ejecución se realiza dentro de una transacción para garantizar la integridad y la consistencia de los datos almacenados.

Salidas

En este apartado se presentan los ejemplos del código relacionados con la generación y visualización de información dentro del sistema. Estos fragmentos muestran cómo se construyen las salidas de datos a partir del procesamiento interno, mediante vistas dinámicas y reportes que permiten al usuario consultar resultados, listados y comprobantes generados por el sistema.

Figura 56

Salida de datos historial de incapacidades

```

@GetMapping("/historial")
public String verHistorial(@RequestParam("empleadoId") Long personaId,
                          @RequestParam(value = "tipo", required = false) TipoIncapacidad tipo,
                          Model model) {
    Empleado empleado = empleadoService.findByPersonaId(personaId)
        .orElseThrow(() -> new IllegalArgumentException("Empleado no encontrado para la persona ID: " + personaId));

    List<Incapacidad> historial = tipo != null ?
        incapacidadService.filtrarPorEmpleadoYTipo(empleado.getIdEmpleado(), tipo) :
        incapacidadService.obtenerHistorialPorEmpleado(empleado.getIdEmpleado());

    Incapacidad masReciente = historial.isEmpty() ? null : historial.get(index: 0);

    model.addAttribute("empleado", empleado);
    model.addAttribute("historial", historial);
    model.addAttribute("masReciente", masReciente);
    model.addAttribute("tipoSeleccionado", tipo != null ? tipo.name() : "TODOS");

    model.addAttribute("title", "Historial de incapacidades");
    model.addAttribute("content", "incapacidades/historialIncapacidades");

    return "fragments/layout";
}

```

Fuente: Elaboración propia, 2025.

Este método muestra el historial de incapacidades de un empleado, lo que permite filtrar la información según el tipo de incapacidad. Obtiene los registros desde el servicio correspondiente, identifica la incapacidad más reciente y envía los datos al modelo para que sean visualizados en la interfaz del sistema.

Figura 57*Salida de datos comprobante de pago*

```

@Transactional(readonly = true)
@Override
public ComprobanteNominaDTO obtenerComprobanteNomina(Long idNominaLinea) {
    NominaLinea linea = nominaLineaRepository.findById(idNominaLinea)
        .orElseThrow(() -> new IllegalArgumentException(s: "No se encontró la línea de nómina"));

    Empleado empleado = linea.getEmpleado();
    NominaPeriodo periodo = linea.getPeriodo();

    HistorialLaboral historialLaboral = null;
    List<HistorialLaboral> historiales = historialLaboralRepository.findHistorialVigenteOPorReciente(empleado.getIdEmpleado());
    if(historiales != null && !historiales.isEmpty()){
        historialLaboral = historiales.get(index: 0);
    }

    ComprobanteNominaDTO dto = new ComprobanteNominaDTO();
    dto.setNombreCompleto(EmpleadoMapper.getNombreCompleto(empleado));
    dto.setNumeroIdentificacion(empleado.getPersona().getNumeroIdentificacion());
    dto.setPuesto(historialLaboral != null ? historialLaboral.getPuesto().getNombre() : "No asignado");
    dto.setTipoJornada(historialLaboral != null ? historialLaboral.getTipoJornadaLaboral().getNombre() : "No definido");
    dto.setFechaIngreso(empleado.getFechaIngreso());
    dto.setSalarioBase(historialLaboral != null ?
        historialLaboral.getSalarioHora().setScale(newScale: 2, RoundingMode.HALF_UP)
        : BigDecimal.ZERO.setScale(newScale: 2, RoundingMode.HALF_UP));

    dto.setDescripcionPeriodo(periodo.getDescripcion());
    dto.setFechaInicioPeriodo(periodo.getFechaInicio());
    dto.setFechaFinPeriodo(periodo.getFechaFin());

    dto.setSalarioBruto(linea.getSalarioBruto());
    dto.setTotalRebajos(linea.getTotalDescuentos());
    dto.setSalarioNeto(linea.getSalarioNeto());

    DetalleNominaDTO detalle = null;
    try {
        detalle = objectMapper.readValue(linea.getDetalles(), valueType: DetalleNominaDTO.class);
        dto.setDetalle(detalle);
    } catch (Exception e) {
        throw new RuntimeException("Error al leer detalles de nómina: " + e.getMessage());
    }

    dto.setIdEmpleado(empleado.getIdEmpleado());
    dto.setIdPeriodo(periodo != null ? periodo.getIdPeriodoNomina() : null);

    return dto;
}

```

Fuente: Elaboración propia, 2025.

Este método genera el comprobante de nómina correspondiente a una línea de cálculo específica. Recupera la información del empleado, su historial laboral y el período respectivo, con lo que integra todos los datos en un objeto de transferencia que representa el comprobante final.

Figura 58

Salida de datos horas extra de la semana

```

@GetMapping("/semana")
public String verHoraExtraSemanal(@RequestParam("empleadoId") Long empleadoId,
    @RequestParam(required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate referencia,
    Model model, RedirectAttributes redirect) {

    try {
        Empleado empleado = empleadoRepository.findByPersona_IdPersona(empleadoId)
            .orElseThrow(() -> new BusinessException(message: "Empleado no encontrado"));

        Persona persona = empleado.getPersona();
        PersonaResponseDTO empleadoDTO = personaMapper.toResponseDTO(persona);

        LocalDate fecha = (referencia != null) ? referencia : LocalDate.now();
        LocalDate inicioSemana = fecha.with(DayOfWeek.MONDAY);
        LocalDate finSemana = inicioSemana.plusDays(daysToAdd: 6);

        List<HorasExtraDTO> registros = horasExtraService.listarPorEmpleadoYPeriodo(
            empleado.getIdEmpleado(), inicioSemana, finSemana);

        model.addAttribute("empleado", empleadoDTO);
        model.addAttribute("empleadoId", empleadoDTO.getIdPersona());
        model.addAttribute("referencia", fecha);
        model.addAttribute("inicioSemana", inicioSemana);
        model.addAttribute("finSemana", finSemana);
        model.addAttribute("registros", registros);

        model.addAttribute("title", "Horas Extras Semanal de " + empleadoDTO.getNombreCompleto().trim());
        model.addAttribute("content", "asistencia/asistenciaHorasExtraSemana");

        return "fragments/layout";
    } catch (BusinessException e) {
        redirect.addFlashAttribute("error", e.getMessage());
        return "redirect:/rh/horas-extra/listar-empleados";
    }
}

```

Fuente: Elaboración propia, 2025.

Este método permite visualizar el registro semanal de horas extra de un empleado. Calcula automáticamente el período correspondiente a la semana y obtiene los registros almacenados en la base de datos. Los resultados se envían al modelo para su presentación en la interfaz del sistema.

Procesos

En este apartado se presentan ejemplos de código que representan la lógica interna del sistema, en que se ejecutan los cálculos, validaciones y operaciones principales que transforman los datos ingresados por el usuario. Estos fragmentos muestran cómo se implementan los procesos que permiten el funcionamiento de los distintos módulos y la correcta gestión de la información dentro del sistema.

Figura 59

Proceso cálculo del rebajo de la CCSS

```

@Override
public RebajoAutomaticoDTO calcularRebajoCCSS(BigDecimal salarioBruto) {
    if(salarioBruto == null || salarioBruto.compareTo(BigDecimal.ZERO) <= 0){
        throw new IllegalArgumentException(s: "El salario bruto debe ser mayor que cero para calcular el rebajo CCSS.");
    }

    BigDecimal monto = salarioBruto.multiply(PORCENTAJE_CCSS).divide(BigDecimal.valueOf(val: 100), scale: 2, RoundingMode.HALF_UP);

    RebajoAutomaticoDTO dto = new RebajoAutomaticoDTO();
    dto.setTipoRebajo(TipoRebajo.CCSS);
    dto.setPorcentaje(PORCENTAJE_CCSS);
    dto.setSalarioBruto(salarioBruto.setScale(newScale: 2, RoundingMode.HALF_UP));
    dto.setMontoCalculado(monto);

    return dto;
}

```

Fuente: Elaboración propia, 2025.

Este método forma parte del proceso de cálculo del rebajo correspondiente a la CCSS. Valida que el salario bruto sea válido y aplica la fórmula establecida para determinar el monto por deducir. Finalmente, retorna un objeto con la información del rebajo calculado, para garantizar precisión.

Figura 60

Proceso calcular salario promedio diario

```

@Override
public BigDecimal calcularPromedioDiario(Long empleadoId, LocalDate hasta) {
    if(empleadoId == null || hasta == null){
        throw new IllegalArgumentException(s: "El ID del empleado y la fecha no pueden ser nulos");
    }

    LocalDate desde = hasta.minusWeeks(weeksToSubtract: 50);
    LocalDateTime inicio = desde.atStartOfDay();
    LocalDateTime fin = hasta.atTime(LocalTime.MAX);

    List<Nominalinea> lineas = nominalineaRepository.findByEmpleadoAndFechas(empleadoId, inicio, fin);

    if(lineas.isEmpty() || lineas == null){
        return BigDecimal.ZERO;
    }

    BigDecimal totalBruto = lineas.stream()
        .map(Nominalinea::getSalarioBruto)
        .filter(Objects::nonNull)
        .reduce(BigDecimal.ZERO, BigDecimal::add);

    BigDecimal promedioSemanal = totalBruto.divide(BigDecimal.valueOf(val: 50), scale: 2, RoundingMode.HALF_UP);
    BigDecimal promedioDiario = promedioSemanal.divide(BigDecimal.valueOf(val: 6), scale: 2, RoundingMode.HALF_UP);

    return promedioDiario;
}

```

Fuente: Elaboración propia, 2025.

Por este método se realiza el cálculo del salario promedio diario, conforme a la legislación laboral costarricense y las directrices del Ministerio de Trabajo y Seguridad Social. Para ello, se suman los salarios brutos percibidos durante las últimas cincuenta semanas y se

obtiene el promedio semanal. Posteriormente dicho promedio se divide entre seis días laborales, considerando que la empresa aplica la modalidad de pago semanal sin remuneración del día de descanso. De esta forma se garantiza un cálculo justo para ambas partes y se cumple con el Código de Trabajo de Costa Rica.

Figura 61

Proceso construir un token JWT

```
public String generateToken(UserDetails userDetails) {
    return Jwts.builder()
        .setSubject(userDetails.getUsername())
        .claim(name: "authorities", userDetails.getAuthorities())
        .setIssuedAt(new Date())
        .setExpiration(new Date(System.currentTimeMillis() + 100 + 60 * 60 * 10))
        .signWith(key, SignatureAlgorithm.HS256)
        .compact();
}
```

Fuente: Elaboración propia, 2025.

Este método forma parte del proceso de autenticación del sistema, en que se genera un token JWT, que contiene la información del usuario autenticado y sus roles. El token se firma digitalmente utilizando el algoritmo HMAC SHA-256 para garantizar la integridad y seguridad de las sesiones dentro de la aplicación.

Figura 62

Proceso de cálculo de pago de incapacidades

```
@Override
public BigDecimal calcularMontoIncapacidadEnNomina(Long empleadoId, LocalDate inicioNomina, LocalDate finNomina) {
    if(empleadoId == null || inicioNomina == null || finNomina == null){
        throw new IllegalArgumentException(s: "Parámetros nulos en el cálculo de incapacidad.");
    }

    BigDecimal salarioDiario = salarioPromedioService.calcularPromedioDiario(empleadoId, finNomina);
    if(salarioDiario == null || salarioDiario.compareTo(BigDecimal.ZERO) <= 0){
        return BigDecimal.ZERO;
    }

    List<Incapacidad> incapacidades = incapacidadRepository.findIncapacidadEnRangos(empleadoId, inicioNomina, finNomina);
    if(incapacidades == null || incapacidades.isEmpty()){
        return BigDecimal.ZERO;
    }

    BigDecimal total = BigDecimal.ZERO;

    for(Incapacidad i : incapacidades){
        if(i.getTipo() != TipoIncapacidad.CCSS){
            continue;
        }

        LocalDate desde = i.getFechaInicio().isBefore(inicioNomina) ? inicioNomina : i.getFechaInicio();
        LocalDate hasta = i.getFechaFin().isAfter(finNomina) ? finNomina : i.getFechaFin();

        long diasEnNomina = desde.datesUntil(hasta.plusDays(daysToAdd: 1)).count();
        long diasPagados = Math.min(diasEnNomina, b: 3);

        BigDecimal montoPorDia = salarioDiario.multiply(BigDecimal.valueOf(val: 0.5));
        total = total.add(montoPorDia.multiply(BigDecimal.valueOf(diasPagados)));
    }

    return total.setScale(newScale: 2, RoundingMode.HALF_UP);
}
```

Fuente: Elaboración propia, 2025.

Por este método se calcula el monto correspondiente a las incapacidades reportadas por enfermedad o accidente común, conforme a la legislación laboral costarricense. En caso de incapacidades cubiertas por el Instituto Nacional de Seguros (INS), el pago asumido en su totalidad por dicha institución, por lo que no genera costo para la empresa. De esta manera el sistema garantiza un cálculo exacto y legal.

Validaciones

En este apartado se presentan fragmentos de códigos destinados a garantizar la integridad de datos y el correcto funcionamiento del sistema. Estas validaciones permiten detectar y prevenir errores en la información ingresada o procesada, lo que asegurando que los cálculos, registros y operaciones cumplan con las reglas de negocio.

Figura 63

Validaciones en el nivel de DTO

```
public class HistorialLaboralDTO {
    @NotNull(message = "El ID del empleado es obligatorio")
    @Positive(message = "El ID del empleado debe ser positivo")
    private Long idEmpleado;

    @NotNull(message = "El ID del puesto es obligatorio")
    @Positive(message = "El ID del puesto debe ser positivo")
    private Integer idPuesto;

    @NotNull(message = "El salario por hora es obligatorio")
    @DecimalMin(value = "0.01", message = "El salario por hora debe ser mayor a cero")
    @Digits(integer = 12, fraction = 4, message = "Formato inválido para salario por hora.")
    private BigDecimal salarioHora;

    @NotNull(message = "La fecha de inicio es obligatoria")
    private LocalDate fechaInicio;

    private LocalDate fechaFin;

    @NotNull(message = "El ID del tipo de jornada laboral es obligatorio.")
    @Positive(message = "El ID del tipo de jornada laboral debe ser positivo")
    private Long idTipoJornadaLaboral;
}
```

Fuente: Elaboración propia, 2025.

Este código define validaciones aplicadas a los datos del historial de un empleado, con lo que se garantiza que los valores ingresados cumplan con las reglas establecidas antes de ser procesadas o almacenadas en la base de datos.

Figura 64

Validación en clase controlador

```

@PostMapping("/contacto-emergencia/guardar")
public String crear(@Valid ContactoEmergenciaDTO dto, BindingResult result, Model model) {
    if(result.hasErrors()){
        model.addAttribute("contacto", dto);

        //Codigo del layout
        model.addAttribute("title", "Crear Contacto de Emergencia");
        model.addAttribute("content", "persona/crearContactoEmergencia");

        return "fragments/layout";
    }
    Persona persona = personaService.buscarPorId(dto.getPersonaId());
    var contacto = contactoEmergenciaMapper.toEntity(dto, persona);
    contactoEmergenciaService.save(contacto);

    return "redirect:/rh/persona/contacto-emergencia";
}

```

Fuente: Elaboración propia, 2025.

Este fragmento muestra la validación en el nivel de controlador, en que se verifican los datos ingresados por el usuario antes de registrar un contacto de emergencia. Mediante el uso de `@Valid` y `BindingResult`, el sistema detecta y gestiona errores en tiempo de ejecución, para garantizar que únicamente la información completa y correcta sea procesada.

Figura 65

Validación en el nivel de servicios

```

private BigDecimal validarSalarioBruto(SalarioBrutoDTO salarioBrutoDTO) {
    BigDecimal salarioBruto = salarioBrutoDTO.getTotalGeneral();
    if (salarioBruto == null || salarioBruto.compareTo(BigDecimal.ZERO) <= 0) {
        throw new IllegalArgumentException(s: "El salario bruto calculado no es válido.");
    }
    return salarioBruto;
}

```

Fuente: Elaboración propia, 2025.

Este fragmento representa una validación en el nivel de servicio, en el que verifica la validez del salario bruto calculado antes de continuar con los procesos de nómina. La verificación garantiza que el valor sea mayor a cero, para evitar errores lógicos o inconsistentes en los cálculos salariales y reforzar la integridad de los datos dentro del sistema.

Módulos señalados en el alcance

En este apartado se presentan los principales módulos desarrollados dentro del sistema, los cuales fueron definidos en el alcance del proyecto. Cada módulo cumple con una función específica dentro de la gestión de recursos humanos, lo que contribuye al alcanzar los objetivos planteados.

Figura 66*Código registrar una asistencia*

```

@Override
@Transactional
public void registrarAsistencia(AsistenciaDTO dto) {
    validarDatos(dto);
    validarDuplicado(dto);
    validarDuracion(dto);
    validarLimiteDiario(dto);
    validarLimiteSemanal(dto);

    Empleado empleado = empleadoRepository.findById(dto.getEmpleadoId())
        .orElseThrow(() -> new BussinesException(message: "No se encontró el empleado asociado a la persona seleccionada"));

    Asistencia asistencia = new Asistencia();
    asistencia.setEmpleado(empleado);
    asistencia.setFecha(dto.getFecha());
    asistencia.setHoraEntrada(dto.getHoraEntrada());
    asistencia.setHoraSalida(dto.getHoraSalida());
    asistencia.setCompletado(dto.getHoraEntrada() != null && dto.getHoraSalida() != null);
    asistencia.setObservaciones(dto.getObservaciones());

    asistenciaRepository.save(asistencia);
}

```

Fuente: Elaboración propia, 2025.

Este método pertenece al módulo de gestión de asistencias y se encarga de registrar la información diaria del empleado en el sistema. Una vez validada la información se asocia con la asistencia al empleado correspondiente y se almacena en la base de datos.

Figura 67*Código Registrar horas extra*

```

@Override
public void registrarHorasExtra(HorasExtraDTO dto) {
    validarDatos(dto);
    validarDuracion(dto);
    validarLimiteDiario(dto);
    validarSolapamiento(dto);

    if (dto.getTipoFeriado() == TipoFeriado.FERIADO_NO_TRABAJADO) {
        throw new BussinesException(message: "No se pueden registrar horas extras en un día feriado no trabajado");
    }

    Empleado empleado = empleadoRepository.findById(dto.getEmpleadoId())
        .orElseThrow(() -> new BussinesException(message: "No se encontró el empleado seleccionado"));

    HorasExtras entity = horaExtraMapper.toEntity(dto);

    entity.setEmpleado(empleado);

    horaExtraRepository.save(entity);
}

```

Fuente: Elaboración propia, 2025.

En método forma parte del módulo de gestión de horas extra y permite las horas trabajadas por un empleado fuera de su jornada ordinaria. Una vez validados los datos el sistema asocia las horas extra con el empleado correspondiente y las almacena en la base de datos.

Figura 68*Código para registrar un permiso*

```

@Override
public PermisoDTO registrarPermiso(PermisoDTO dto) {
    Empleado empleado = empleadoService.buscarPorId(dto.getIdEmpleado());
    Permiso permiso = permisoMapper.toEntity(dto, empleado);
    Permiso guardado = permisoRepository.save(permiso);
    return permisoMapper.toDTO(guardado);
}

```

Fuente: Elaboración propia, 2025.

Este método corresponde al módulo de gestión de permisos y permite registrar las solicitudes de permisos realizadas por los empleados.

Figura 69*Código para actualizar los datos de la persona*

```

@Transactional
public Persona actualizarPersona(PersonaRequestDTO dto){
    Persona persona = personaRepository.findById(dto.getIdPersona()).orElseThrow();
    persona.setPrimerNombre(dto.getPrimerNombre());
    persona.setSegundoNombre(dto.getSegundoNombre());
    persona.setPrimerApellido(dto.getPrimerApellido());
    persona.setEstadoCivil(dto.getEstadoCivil());
    persona.setCorreo(dto.getCorreo());
    persona.setTelefonoFijo(dto.getTelefonoFijo());
    persona.setNumeroIdentificacion(dto.getNumeroIdentificacion());
    persona.setFechaNacimiento(dto.getFechaNacimiento());
    persona.setGenero(dto.getGenero());
    persona.setIsActive(isActivo: true);

    Set<ConstraintViolation<Persona>> violaciones = validator.validate(persona);
    if(!violaciones.isEmpty()){
        throw new ConstraintViolationException(message: "Errores de validación en Persona", violaciones);
    }
    TipoIdentificacion tipo = tipoIdentificacionService.buscarPorId(dto.getTipoIdentificacionId());
    persona.setTipoIdentificacion(tipo);

    if(dto.getCelulares() != null){
        List<TelefonoCelular> celulares = dto.getCelulares().stream()
            .filter(cel -> cel.getNumero() != null && cel.getNumero().trim().matches(regex: "\\d{8}"))
            .map(celDto -> celularMapper.toEntity(celDto, persona))
            .collect(Collectors.toList());
        persona.getCelulares().clear();
        persona.getCelulares().addAll(celulares);
    }

    if(dto.getDireccion() != null){
        persona.setDireccion(direccionMapper.toEntity(dto.getDireccion()));
    }
    Persona actualizada = personaRepository.save(persona);
    return actualizada;
}

```

Fuente: Elaboración propia, 2025.

Este método forma parte del módulo de gestión de empleados y se encarga de actualizar la información personal de un colaborador en el sistema. Durante el proceso se actualizan los datos generales, el tipo de identificación, los números de teléfono y la dirección respectiva.

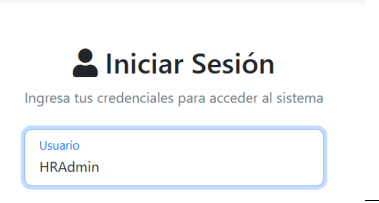
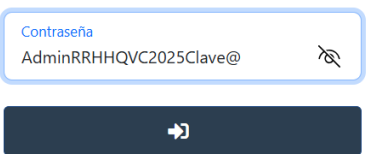
Pruebas del Sistema


En esta sección se presentan los casos de prueba aplicados al sistema de gestión de recursos humanos, con el propósito de verificar su correcto funcionamiento, confiabilidad y cumplimiento de los requerimientos establecidos durante el desarrollo.

Cada caso de prueba incluye la descripción del objetivo, los datos de entrada, los pasos ejecutados, el resultado esperado y el resultado obtenido, con el fin de comparar ambos y determinar si su el comportamiento del sistema es el adecuado. Además, se adjuntan capturas de pantalla que evidencian la ejecución de las pruebas realizadas en los distintos módulos implementados.

Tabla 95

Caso de prueba para inicio de sesión exitoso

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia			
ID del Caso de Prueba	CT-001		
Nombre del caso de Prueba	Autenticación con credenciales válidas		
Módulo/ Funcionalidad	Seguridad / Login		
Tipo de Prueba	Sistema / Aceptación		
Prioridad	Alta		
Precondiciones	El usuario administrador de Recursos Humanos está registrado en el sistema con un rol válido y está activo. Se conocen el usuario y la contraseña correctos.		
Datos de Entrada	Usuario: HRAdmin Contraseña: AdminRRHHQVC2025Clave@		
Detalles de pruebas			
Pasos Por Seguir	Resultado esperado	Resultado obtenido	Estado
1. Ingresar el usuario en el campo "Usuario"	Permite escribir en el campo texto "Usuario"		Aprobado
2. Ingresar la contraseña en el campo "Contraseña"	Permitir ingresar texto en la caja de texto "Contraseña"		Aprobado

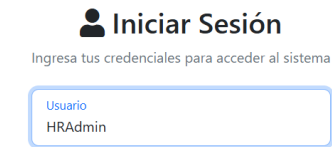

3. Presionar el botón “Iniciar sesión”	Se validan las credenciales y los roles		Aprobado
----------------------------------------	-----------------------------------------	------------------------------------------------------------------------------------	----------


Fuente: Elaboración propia, 2025.

La prueba se ejecutó correctamente lo que permite el acceso al sistema con las credenciales del usuario autorizado. El comportamiento observado coincide con el resultado esperado, ya que el sistema validó las credenciales y las redirigió al panel principal.

Tabla 96

Caso de prueba de inicio de sesión fallido

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia			
ID del Caso de prueba	CT-002		
Nombre del Caso de prueba	Autenticación con credenciales incorrectas		
Módulo/ funcionalidad	Seguridad / Login		
Tipo de prueba	Sistema / Aceptación negativa		
Prioridad	Alta		
Precondiciones	El usuario administrador de recursos humanos está registrado en el sistema con un rol válido y está activo. El usuario intenta ingresar con credenciales inválidas		
Datos de Entrada	Usuario: HRAdmin Contraseña: ContraseñaInvalida123		
Detalles de Pruebas			
Pasos por seguir	Resultado esperado	Resultado obtenido	Estado
1. Ingresar al usuario en el campo “Usuario”	Permitir ingresar texto en la caja de texto “Usuario”		Aprobado
2. Ingresar la contraseña en el campo “Contraseña”	Permitir ingresar texto en la caja de texto “Contraseña”		Aprobado

3. Presionar el botón Iniciar Sesión	El sistema rechaza el inicio de sesión, muestra un mensaje de error (“Credenciales incorrectas”) y no permite avanzar al dashboard.		Aprobado
--------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------	----------



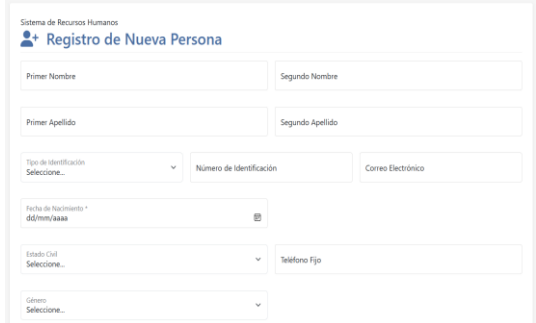
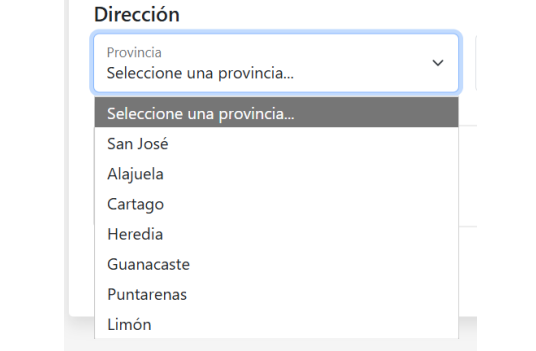
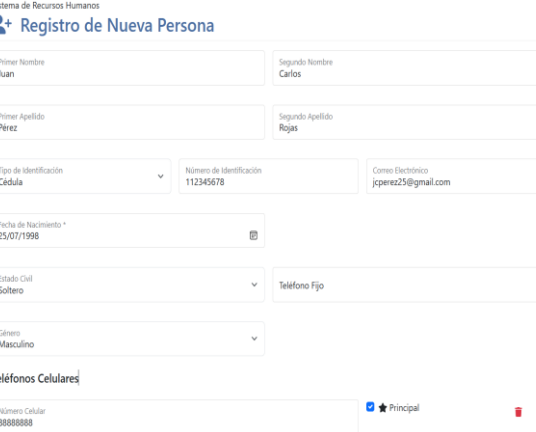
Fuente: Elaboración propia, 2025.

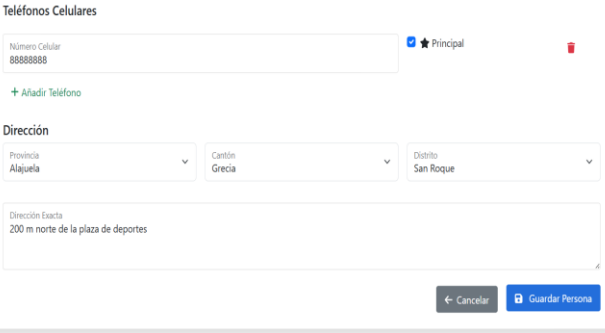

El sistema respondió correctamente ante credenciales incorrectas, muestra el mensaje de error de autenticación y evita el acceso al sistema. Esto confirma el correcto funcionamiento del mecanismo de validación y protección del módulo de inicio de sesión.

Tabla 97

Caso de prueba Registrar una persona

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia			
ID del Caso de prueba	CT-003		
Nombre del caso de Prueba	Registro de datos de un empleado nuevo		
Módulo/ Funcionalidad	Gestión de empleado / persona		
Tipo de Prueba	Integración / Sistema		
Prioridad	Alta		
Precondiciones	El usuario de RRHH tiene sesión iniciada. Catálogos necesarios existen en la base de datos: tipo de identificación, provincia, cantón y distrito. No existe previamente una persona con el mismo número de cédula.		
Datos de Entrada	Tipo de identificación: Cédula Número de Identificación: 112345678 Nombre: Juan Carlos Pérez Rojas Fecha de nacimiento: 25/07/1998 Correo Electrónico: jcperez25@gmail.com Estado civil: Soltero Género: Masculino Teléfono celular: 8888-8888 Provincia: Alajuela Cantón: Grecia Distrito: San Roque Dirección exacta: 200 m norte de la plaza de deportes		
Detalles de pruebas			
Pasos por seguir	Resultado esperado	Resultado obtenido	Estado

1. Ingresar al módulo de empleados	Permitir ingresar a la vista principal de empleados en el sistema		Aprobado
2. Seleccionar el botón "+Registrar persona"	Botón visible para ingresar al formulario de registro		Aprobado
3. Presionar el botón "+Registrar Persona"	El sistema redirige al formulario de "Registro de Nueva Persona"		Aprobado
4. Formulario carga los catálogos previamente definidos	El sistema carga los catálogos para que sean seleccionados		Aprobado
5. Se completa el formulario con los datos solicitados	Debe permitir ingresar los datos en sus respectivos campos		Aprobado

6. Seleccionar el botón “Guardar persona”	Se validan los datos y de ser correctos se crea la persona y se almacena en la base de datos		Aprobado
7. Persona registrada correctamente. Aparece en la lista de empleados	Si la persona fue agregada correctamente se debe mostrar en la lista de empleados		Aprobado

Fuente: Elaboración propia, 2025.

La prueba demostró el funcionamiento exitoso del registro de personas ya que validó los campos requeridos y correctamente la información en la base de datos. El nuevo empleado se visualizó en la lista de empleados activos, lo que evidencia la persistencia de los datos y el cumplimiento del flujo definido.

Tabla 98

Caso de prueba Registro de persona fallido

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
ID del Caso de prueba	CT-004
Nombre del caso de prueba	Registro fallido por validaciones de datos
Módulo/ Funcionalidad	Gestión de Empleado / Persona
Tipo de prueba	Sistema / Validación negativa
Prioridad	Alta
Precondiciones	El usuario de RRHH tiene sesión iniciada. Catálogos necesarios existen en la base de datos: Tipo de identificación, provincia, cantón y distrito. Se intenta registrar a una persona con datos incompletos o inválidos.
Datos de Entrada	Tipo de identificación: Cédula Número de identificación: vacío Nombre: Ana María Rojas: vacío Fecha de nacimiento: 01/11/1990 Correo Electrónico: anamaria77@gmail.com Estado civil: Soltera Género: Femenino Teléfono celular: 71584433

	Provincia: San José Cantón: vacío Distrito: Jaris Dirección exacta: 500 m al este del salón comunal		
Detalles de pruebas			
Pasos por seguir	Resultado esperado	Resultado obtenido	Estado
1. Ingresar al módulo de empleados	Permitir ingresar a la vista principal de empleados en el sistema		Aprobado
2. Seleccionar el botón "+Registrar persona"	Botón visible para ingresar al formulario de registro		Aprobado
3. Presionar el botón "+Registrar Persona"	El sistema redirige al formulario de "Registro de nueva persona"		Aprobado
4. El Formulario carga los catálogos previamente definidos	El sistema carga los catálogos para que sean seleccionados		Aprobado
5. Se completa el formulario con los datos incorrectos	Campo de primer apellido vacío y falla guardar a la persona		Aprobado

6. Se completa el formulario con los datos incorrectos	El campo correspondiente al número de identificación vacío y debe fallar el guardar y mostrar un mensaje de advertencia		Aprobado
7. Se completa el formulario con los datos incorrectos	El campo correspondiente al cantón vacío. Debe fallar y mostrar un mensaje de advertencia		Aprobado

Fuente: Elaboración propia, 2025.

Durante la ejecución de la prueba el sistema reaccionó correctamente ante los campos incompletos e inválidos, y mostró mensajes de advertencia y de bloqueó el guardado del registro. El resultado obtenido confirma que las validaciones en el nivel de formulario y servicio funcionan de forma adecuada, lo que garantiza la integridad y consistencia de los datos almacenados.

Tabla 99

Caso de prueba Registro de asistencia

Prototipo funcional para la gestión de recursos humanos en la empresa Quesada Vargas Construcciones, S.A, ubicada en Grecia	
ID del caso de prueba	CT-005
Nombre del Caso de prueba	Registrar asistencia de un empleado
Módulo/ Funcionalidad	Gestión de Asistencia
Tipo de prueba	Sistema
Prioridad	Media
Precondiciones	El usuario de RRHH tiene sesión iniciada. Se cuenta con la fecha y horas de la jornada laboral No existe otra asistencia registrada para ese mismo día
Datos de Entrada	Empleado: Kendall Josué Torres Loría Fecha: 28/10/2025 Hora de inicio: 6:00 Tipo de Día laboral: Normal Hora de fin: vacío Observaciones: Prueba de registro de asistencia
Detalles de pruebas	

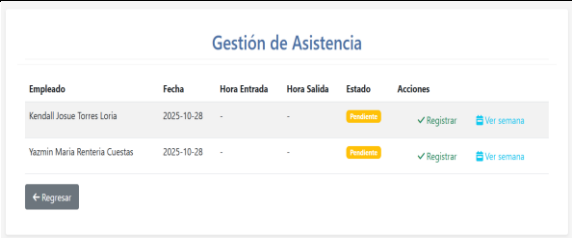

Pasos por seguir	Resultado esperado	Resultado obtenido	Estado
1. Ingresar al módulo de asistencia	Permitir ingresar a la vista principal de asistencia en el sistema	 <p>The screenshot shows a dashboard titled "Gestión de Asistencia". It features a table with columns: Empleado, Fecha, Hora Entrada, Hora Salida, Estado, and Acciones. Two rows are visible: Kendall Josue Torres Loria and Yazmin Maria Renteria Cuestas, both with a "Pendientes" status and "Registrar" and "Ver semana" actions.</p>	Aprobado
2. Seleccionar el botón "Registrar" del empleado correspondiente y redirigir al formulario de Registrar asistencia	Seleccionar el botón "Registrar" del empleado correspondiente y redirigir al formulario de Registrar asistencia	 <p>The screenshot shows the "Registrar Asistencia" form. It includes fields for Empleado (Kendall Josue Torres Loria), Fecha (28/10/2025), Tipo de Día Laboral (Normal), Hora de Entrada, and Hora de Salida. There is an Observaciones field and buttons for "Cancelar" and "Guardar".</p>	Aprobado
3. Completar los datos que el formulario solicita	El formulario debe permitir completar los campos solicitados	 <p>This screenshot is identical to the previous one, but the Observaciones field now contains the text "Prueba de registro de asistencia".</p>	Aprobado
4. Seleccionar el botón "Guardar"	El sistema valida los campos y de ser válidos permite guardar el registro en la base de datos y se puede visualizar en la pantalla de Asistencia Semanal el registro guardado	 <p>The screenshot shows the "Asistencia Semanal" view. It displays the "Fecha de referencia" (28/10/2025) and a "Buscar semana" button. Below, it shows the employee name and the week range. A table lists the attendance record for 2025-10-28 at 06:00 with a "Pendientes" status and "Prueba de registro de asistencia" observation. A green "+ Agregar asistencia" button and a blue "Editar" button are also visible.</p>	Aprobado

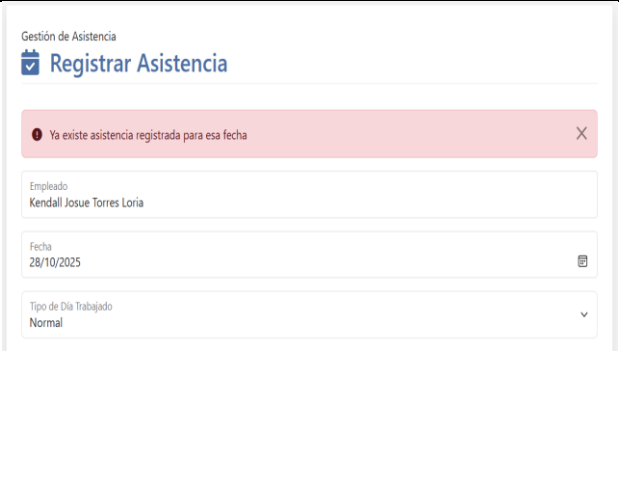
Fuente: Elaboración propia, 2025.

La prueba se ejecutó satisfactoriamente, permitiendo registrar la asistencia del empleado y visualizar en la lista semanal. Este comportamiento confirma el correcto funcionamiento del proceso de asistencia y su persistencia en la base de datos.

Tabla 100

Caso de prueba registro fallido de asistencia

Prototipo Funcional para la Gestión de Recursos Humanos en la Empresa Quesada Vargas Construcciones S.A, ubicada en Grecia			
ID del Caso de Prueba	CT-006		
Nombre del Caso de Prueba	Registro fallido de asistencia duplicada		
Módulo/ Funcionalidad	Gestión de Asistencia		
Tipo de Prueba	Sistema / Validación negativa		
Prioridad	Alta		
Precondiciones	El usuario de RRHH tiene sesión iniciada. Existe un registro previo de asistencia para el mismo empleado y la misma fecha El sistema valida solapamientos de fechas		
Datos de Entrada	Empleado: Kendall Josué Torres Loría Fecha: 28/10/2025 Hora de inicio: 6:00 Tipo de Día Laboral: Normal Hora de fin: vacío Observaciones: Prueba de registro de fallido		
Detalles de Pruebas			
Pasos Por Seguir	Resultado Esperado	Resultado obtenido	Estado
1. Ingresar al módulo de Asistencia	Permitir ingresar a la vista principal de asistencia en el sistema		Aprobado
2. Seleccionar el botón "Registrar"	Seleccionar el botón "Registrar" del empleado correspondiente y redirige al formulario de registrar asistencia		Aprobado

3. Completar los datos que el formulario solicita	Completar el formulario con los datos de la fecha de un registro ya existente para ese empleado y se espera que falle la creación del registro y muestra un mensaje de error	 <p>The screenshot shows a web interface for 'Gestión de Asistencia' with a 'Registrar Asistencia' button. Below the button is a red error message box that says 'Ya existe asistencia registrada para esa fecha'. The form fields are filled with: Empleado: Kendall Josue Torres Loria, Fecha: 28/10/2025, and Tipo de Día Trabajado: Normal.</p>	Aprobado
---------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------

Fuente: Elaboración propia, 2025.

El sistema reaccionó ante un intento de registro duplicado. Mostró un mensaje de advertencia y evita la inserción del nuevo registro. Esto confirma que la validación de solapamiento por empleado y fecha funciona adecuadamente, pues mantiene la integridad y evita duplicidades en el módulo de asistencia.

La ejecución de los casos de prueba permitió verificar el correcto funcionamiento de los diferentes módulos del sistema de gestión de recursos humanos, tanto en escenarios exitosos como en situaciones de error controlado.

Los errores obtenidos demuestran que el sistema cumple con los requerimientos funcionales establecidos, valida adecuadamente los datos ingresados y garantiza la integridad de la información almacenada. De esta manera se confirma la estabilidad, confianza y consistencia del prototipo funcional desarrollado para la empresa Quesada Vargas Construcciones, S.A.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones y recomendaciones relativas al desarrollo del prototipo funcional del sistema de gestión de recursos humanos para la empresa Quesada Vargas Construcciones, S.A. En las conclusiones se exponen los principales logros obtenidos durante el proyecto, mientras que las recomendaciones proponen acciones de mejora y buenas prácticas para garantizar un uso adecuado y un aprovechamiento óptimo del sistema desarrollado.

Conclusiones

En este apartado se resumen los hallazgos más importantes, el cumplimiento de los objetivos planteados y el impacto del prototipo en la empresa Quesada Vargas Construcciones, S.A. Además, se destaca la relevancia del proyecto como un aporte académico y práctico en el área de gestión de recursos humanos.

El desarrollo del sistema web de gestión de Recursos Humanos para la empresa Quesada Vargas Construcciones, S.A. permitió cumplir con los objetivos planteados del proyecto. Se analizaron los requerimientos funcionales y no funcionales con el propósito de diseñar una solución acorde con las necesidades de la empresa y para entender el problema identificado. Se construyó una arquitectura ajustada a su infraestructura y se estableció una base de datos robusta en MySQL. Además, se implementó un prototipo funcional con Java y Spring Boot que automatiza los principales procesos del área de recursos humanos. Finalmente, se realizaron pruebas para verificar la integridad, seguridad y el cumplimiento de la legislación costarricense.

El prototipo demostró que la automatización de tareas, como el cálculo aguinaldo, vacaciones y las liquidaciones reduce significativamente los errores humanos, la duplicidad de tareas y el tiempo invertido en cálculos manuales. De esta forma se mejora la eficiencia operativa del departamento, se minimizan los errores frecuentes en los procesos actuales y se incrementa la confiabilidad de la información.

La elección de una arquitectura monolítica con servidor embebido Tomcat y tecnologías de código abierto, como Spring Boot, Thymeleaf, Bootstrap, MySQL garantizó la compatibilidad con la infraestructura limitada que la empresa, a fin de reducir costos y facilitar su implementación y el mantenimiento futuro. Esta decisión permitió que el proyecto fuera viable tanto técnica como económicamente.

El sistema representa una herramienta moderna que impulsa la transformación digital de la empresa, fortalece la gestión administrativa y promueve una cultura organizacional más eficiente y orientada a la tecnología.

Finalmente, el proyecto permitió aplicar conocimientos adquiridos en la carrera de Ingeniería en Sistemas de la Información, a fin de fortalecer las competencias del estudiante y de tener la oportunidad de adquirir nuevas habilidades mediante la experiencia práctica, para consolidar tanto los conocimientos teóricos como los aplicados en un entorno real.

Recomendaciones

En este apartado se presentan las recomendaciones acerca del proyecto, que orientan futuras acciones para favorecer la correcta implementación y aprovechamiento del prototipo funcional desarrollado. Estas sugerencias se basan en los resultados obtenidos y en las observaciones realizadas durante el proceso de desarrollo, con el fin de fortalecer el sistema y de garantizar su uso eficiente dentro de la empresa Quesada Vargas Construcciones, S.A.

Se recomienda implementar el sistema en un servidor local dedicado dentro de la empresa Quesada Vargas Construcciones, S.A., con el fin de garantizar la estabilidad y la continuidad del sistema de gestión de recursos humanos. Para ello se deben generar copias de respaldo automáticas de la base de datos, el resguardo del código fuente y configuraciones que protejan la información ante fallas. Es fundamental aplicar respaldos diarios y conservar una copia adicional fuera del sitio físico, a fin de asegurar los principios de integridad, disponibilidad y confidencialidad de la información. Es recomendable aplicar esta idea antes de la puesta en producción del sistema, es decir, una vez que el prototipo sea adoptado oficialmente por la empresa. De esta forma se garantizará que, ante cualquier incidente o falla técnica, la empresa pueda recuperar la información y continuar sus operaciones.

Se recomienda implementar un plan piloto de prueba en un entorno simulado de producción, con el objetivo de verificar el correcto funcionamiento del prototipo, detectar posibles errores y realizar los ajustes necesarios antes de su uso oficial en la empresa Quesada Vargas Construcciones, S.A. El plan piloto debe contemplar un escenario que reproduzca las condiciones reales de la empresa, utilizando datos de ejemplo y usuarios de prueba que simulen las tareas cotidianas del departamento de recursos humanos. Durante esa etapa se evaluarán la estabilidad, seguridad y usabilidad del sistema, y se documentarán los hallazgos para aplicar las correcciones correspondientes. Esta recomendación debe ejecutarse previo a la puesta en producción del sistema, una vez que el prototipo haya sido finalizado y revisado por el desarrollador.

Se recomienda implementar un plan de capacitación para el personal de recursos humanos, con el objetivo de garantizar el uso adecuado del sistema y de gestión y fomentar la aceptación de este dentro de la empresa. La capacitación puede desarrollarse mediante sesiones prácticas en las que el personal utilice el sistema en un entorno controlado, mediante la resolución de ejemplos reales dentro de la empresa. Además de instruir sobre las funciones técnicas, es importante promover una actitud positiva hacia el cambio, explicando los beneficios del sistema. Este plan debe llevarse a cabo antes de la adopción oficial del sistema,

de modo que el personal ya esté familiarizado con su funcionamiento cuando entre en operación. Además, se recomienda una sesión de acompañamiento durante las primeras semanas de uso real, para atender dudas y reforzar el aprendizaje.

Se recomienda que la implementación del sistema esté acompañada por el desarrollo de una cultura organizacional que promueva la confianza, la apertura al cambio y el uso responsable de la tecnología de la empresa Quesada Vargas Construcciones, S.A. El objetivo es lograr que el personal no perciba la digitalización como una amenaza, sino como una herramienta que facilita y mejora su trabajo diario. Para ello, se sugiere fomentar espacios de diálogo, capacitación y demostraciones prácticas en las que se evidencien los beneficios concretos del sistema. De esta manera, se genera confianza y sentido de pertenencia hacia la herramienta tecnológica y para que los empleados lo integren como parte natural de sus procesos laborales. Esta recomendación debe aplicarse antes de la implementación del sistema o en forma paralela a la fase de capacitación.

Referencias

- Altadill, P., & Pérez, E. (2020). *Java Limpio*.
- Benites, L. (21 de 07 de 2024). *Variables instrumentales: definición y ejemplos*. Obtenido de statologos: https://statologos.com/variables-instrumentales/#Que_es_una_variable_instrumental
- Betancur, J. V. (2022). *Comparativo metodológico y financiero entre las metodologías tradicionales y ágiles estipuladas para el diseño, formulación y gestión de proyectos de software*.
- Bootstrap. (s.f.). Obtenido de <https://getbootstrap.esdocu.com/docs/5.3/getting-started/introduction/>
- Brage, L. B., Cristobál, A. N., Fernández, J. A., & Serra, V. Q. (2022). *Métodos y técnicas de investigación educativa*. Palma: Universitat de les Illes Balears.
- Buscaglia, A. (2020). *Cómo ser front-end sin fallar en el intento*.
- Cambarieri, M. G., Difabio, F., & Martínez, N. G. (2020). Implementación de una Arquitectura de Software guiada por el Dominio. *Universidad Nacional de Río Negro Viedma*.
- Carvajal, C. C. (2023). Las variables y su operacionalización. *Instituto de investigación académica*.
- Castaños, S. P., & Santamaría, S. G. (2023). *La investigación cuantitativa*.
- Castillo, M. J. (9 de 10 de 2023). *Modelos de Capacitación para el Desarrollo Profesional*. Obtenido de Buk: <https://www.buk.mx/blog/modelos-de-capacitacion-para-el-desarrollo-profesional#:~:tex>
- Colombero, V. T., Estelles, J. P., Gallegos, L., & López, P. (2024). Monolitos vs. Microservicios en Arquitectura de Software: Perpectivas para un Desarrollo Eficiente. *Facultas de Tecnología Informática, Universidad Abierta Iternacional (UAI)*.
- Dupuis, A. (23 de 03 de 2020). *Fuentes de información primarias, secundarias y terciarias*. Obtenido de Técnicas de Investigación: https://tecnicasdeinvestigacion.com/fuentes-de-informacion-primaria-y-secundaria-y-terciaria/?utm_source=chatgpt.com
- Elsa Josefina Albornoz Zamora, M. d. (2023). *Metodología de la Investigación aplicada a las ciencias de la salud y la educación*. Quito, Ecuador: Mawil Publicaciones de Ecuador.
- Ericka Solano Fernández, D. P. (2020). El modelo iterativo e incremental para el desarrollo de la aplicación de realidad aumentada Amón_RA. *Tecnología en marcha*, 14.

- Estévez, J. R. (2020). La ingeniería de requisitos en el desarrollo de aplicaciones informáticas. *Revista Cuba de Informática Médica*.
- Fernández, Y. (21 de enero de 2025). *La importancia de la innovación en las PyMes*. Obtenido de La República: <https://www.larepublica.net/noticia/la-importancia-de-la-innovacion-en-las-pymes>
- García, M. E. (2021). *Entorno físico del espacio de trabajo y su impacto en el clima laboral de las organizaciones en México*. Puebla: Repositorio Universidad Autónoma de México.
- González, E. M. (2020). Aplicando seguridad a una API REST con JSON Web Tokens. *Universidad Oberta de Catalunya*.
- Hernández, L. M., Romero, V. A., González, S. A., & Rodríguez, J. A. (2021). Arquitectura REST para el desarrollo de aplicaciones web empresariales. *Revista Electrónica sobre Ciencia, Tecnología y Sociedad*.
- IBM. (s.f.). *¿Qué es Java Spring Boot?*
- IBM. (12 de 10 de 2021). *Gestión de roles y permisos*. Obtenido de <https://www.ibm.com/docs/es/gdp/11.4.0?topic=guardium-managing-roles-permissions>
- IBM. (2023). *Desarrollo de microservicios con MicroProfile*. IBM.
- Laradola, E. (5 de junio de 2023). *sesame*. Obtenido de Capital humano: el activo más valioso de las empresas: <https://www.sesamehr.co/blog/gestion-de-equipos/capital-humano-activo-valioso-empresas/>
- Luza, T. C., Córdova, O. J., Gonzáles, J. A., Limo, F. A., & Flores, C. A. (2023). *Métodos mixtos de investigación para principiantes*. Perú: Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú S.A.C.
- M, E. S. (2021). *Métodos Formales, Ingeniería de Requisitos y Pruebas de Software*. Instituto Antioqueño de Investigación.
- Martín, D. (01 de mayo de 2025). *Recursos Humanos: definición, concepto y funciones*. Obtenido de openhr: <https://www.openhr.cloud/blog/recursos-humanos-funciones-dentro-de-la-empresa>
- Matínez, B. P., Díez, L. G., & Cuadrado, P. P. (2022). Propuesta metodológica para el análisis de imágenes informativas impresas y en línea. *RSCH*.
- Mena, E. A., & Ordóñez, J. A. (2021). *Fundamentos de seguridad informática*. Guayaquil-Ecuador: Grupo Compás.

- Meza, A. L., Araujo, J. G., Vázquez, M. C., Cisneros, L. E., Manqueros, J. M., Macias, A. B., . . . Piedra, J. A. (2021). *Manual de temas nodales de la Investigación Cuantitativa. Un abordaje didáctico*. México: Universidad Pedagógica de Durango.
- Ministerio de Trabajo y Seguridad Social. (14 de 06 de 2024). *El Ministerio*. Obtenido de Ministerio de Trabajo y Seguridad Social: <https://www.mtss.go.cr/elministerio/index.html>
- Ministerio de Trabajo y Seguridad Social. (25 de 03 de 2025). *Temas Laborales*. Obtenido de MTSS: <https://www.mtss.go.cr/temas-laborales/index.html>
- Mochón, F., & Cardona, P. (2021). *Transformación Digital de la Empresa*. Argentina: Alfaomega.
- Molina, A. A., & Garza, A. M. (2021). Enfoques metodológicos en la investigación histórica: cuantitativa, cualitativa y comparativa.
- Mora, R. C. (2020). *Conceptos ágiles aplicados a distintas áreas de una empresa*.
- Morales-Carrillo, J., Cedeño-Valarezco, L., Bravo, J. S., & Calderón, J. G. (2021). Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática. *Iberian Journal of Information System and Technologies*, 18.
- Morales-Carrillo, J., Cedeño-Valarezco, L., Bravo, J. S., & Calderón, J. G. (2021). Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática. *Iberian Journal of Information System and Technologies*, 18.
- Moran, N. V., Beltrán, C. F., Palma, C. M., Macias, S. C., & Miranda, D. I. (2025). La influencia de la tecnología en la gestión de los recursos humanos. *South Florida Journal of Development*, 11.
- Ocampo, D. S. (04 de 8 de 2020). *investigalia*. Obtenido de Fuentes primarias y secundarias de información cuantitativa: <https://investigaliacr.com/investigacion/fuentes-de-informacion-primarias-y-secundarias-en-la-investigacion-cuantitativa/>
- ORACLE. (29 de agosto de 2024). *MySQL: qué es y cómo se usa*. Obtenido de <https://www.oracle.com/cr/mysql/what-is-mysql/>
- Pazmiño, R. S., Andrade, A. S., & Montalvo, F. F. (2022). Disrupción tecnológica en la gestión del Talento Humano. *Journal Of Science And Research*, 20.
- Peña, G. A., Rodríguez, K. L., Menoscal, S. E., León, J. C., & Jordán, E. N. (2021). *La investigación educativa: teoría y práctica*. Tenocientífica Americana.
- Portilla, R. T. (2022). *Como hacer un proyecto de investigación*. Bogotá, Colombia: Alpha Editorial.

- Portilla, R. T., & Sotomayor, M. I. (2022). *Cómo hacer un proyecto de investigación*. Bogotá: Alphaeditorial.
- Quiroz-Vásquez, C., & Goodwin, M. (10 de abril de 2024). *¿Qué es la gestión del ciclo de vida de las aplicaciones (GCV)?* Obtenido de IBM: <https://www.ibm.com/es-es/topics/application-lifecycle-management>
- Rampirez, N. (2021). Marco Metodológico.
- Rica, U. d. (s.f.). *Tamaño de la muestra*. Obtenido de Studocu: <https://www.studocu.com/latam/document/universidad-de-las-ciencias-y-el-arte-de-costa-rica/contabilidad-5/tamano-de-la-muestra/84341247>
- Rojas, M. J., & Rodríguez, L. A. (2020). Diseño e implementación de software basado en la arquitectura por capas para la gestión de la cadena de suministro de la empresa de interventoría Cal y Mayor. *Universidad Distrital Francisco José de Caldas*.
- Roma, J. C. (2020). Introducción al diseño de bases de datos. *Fundació per a la Universitat Oberta de Catalunya (FUOC)*.
- Romero, M. M., León, R. R., Hoces, W. B., Carrasco, R. L., Carranza, C. M., & Acobo, R. C. (2023). *Metodología de la investigación*. Puno, Perú: Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú S.A.C.
- Romero, M. M., León, R. R., Hoces, W. B., Carrasco, R. L., Carranza, C. M., & Acono, R. C. (2023). *Metodología de la investigación: técnicas e instrumentos de investigación*. Perú: Instituto Universitario de Innovación Ciencia y Tecnología.
- Sánchez, P. Y. (2020). Implementación del patrón arquitectónico MVC en aplicaciones web para la arquitectura del software del sistema de capellanía de la UM. *Universidad de Montemorelos*.
- Servicios de Información Profesional, S.A. (29 de noviembre de 2023). *¿Qué es V-Model?* Obtenido de Servicios de Información Profesional, S.A.: <https://www.linkedin.com/pulse/qu%C3%A9-es-v-model-servicios-de-informatica-profesion-qotgf/>
- SUMMA. (12 de mayo de 2025). *50% de las pymes en Costa Rica utiliza algún tipo de IA*. Obtenido de Revista SUMMA: <https://revistasumma.com/50-de-las-pymes-en-costa-rica-utiliza-algun-tipo-de-ia/>
- Tamushi. (13 de 07 de 2022). *Pruebas de integración de software: qué son, niveles y tipos*. Obtenido de Testing iT: <https://www.testingit.com.mx/blog/pruebas-de-integracion-de-software>

- University, L. P. (15 de 09 de 2023). *Estrategias de investigación: Tipos de fuentes*. Obtenido de <https://lifepacific.libguides.com/c.php?g=924149&p=6696820>
- Valenciano, J. A. (2022). Las variables como elemento sustancial en el método científico. *Revista Educación Universidad de Costa Rica*.
- Vargas, B. (14 de 02 de 2024). *La estructura del software: una mirada profunda a su división y clasificación*. Obtenido de Byron Vargar: <https://www.byronvargas.com/web/como-se-divide-el-software/>
- Velásquez, R. M. (2020). *Implementación de un modelo de gestión de pruebas de software según ISTQB para mejorar el proceso del área de certificación en tecnologías web de una entidad financiera*. Perú: Universidad Tecnológica del Perú.

Apéndice A: Guía de Entrevista

GUÍA DE ENTREVISTA

Organización:

Nombre del entrevistado:

Cargo:

Preguntas:

Pregunta 1: ¿Cuáles son las principales tareas que realiza actualmente en el área de Recursos Humanos y qué herramientas usa hoy?

Pregunta 2: ¿Qué procesos considera más importante o crítico en la gestión del personal dentro de la empresa?

Pregunta 3: ¿Podría darme un ejemplo práctico de un proceso que le genera más trabajo o problemas actualmente?

Pregunta 4: ¿Qué datos esenciales de cada empleado deben registrarse en el sistema?

Pregunta 5: ¿Qué funciones considera necesarias que tenga un sistema de gestión de Recursos Humanos?

Pregunta 6: ¿Qué tipo de consultas le serían útiles en el diario de sus funciones?

Pregunta 7: ¿Qué tipo de reportes (ausencias, costos de nómina) le serían útiles que el sistema pueda generar?

Pregunta 8: ¿Desea que el sistema permita controlar aspectos como vacaciones, incapacidades u horas extras?

Pregunta 9: ¿Qué tan importantes es para usted que el sistema sea fácil de usar?

Pregunta 10: ¿Requiere migrar datos existentes al nuevo sistema?

Pregunta 11: ¿Qué nivel de seguridad espera que tenga el sistema respecto a la información del personal?

Pregunta 12: ¿Hay algún diseño técnico o visual que considere importante (por ejemplo, colores, diseño o forma de navegar)?

Pregunta 13: ¿Qué otras funciones o características le gustaría que se incluyan en el sistema?

Pregunta 14: ¿Qué espera que el sistema mejore respecto a su trabajo diario en Recursos Humanos?

Pregunta 15: ¿Hay algo más que considera importante y que no hayamos mencionado?

Apéndice B: Cuestionario

CUESTIONARIO

En el marco de una investigación sobre Sistema web para la gestión de Recursos Humanos en la empresa Quesada Vargas Construcciones S.A., ubicada en Grecia, le invitamos a completar este cuestionario. Su participación es de gran importancia para comprender cómo el tema en estudio influye en la actividad de la organización.

Este cuestionario es confidencial. Sus respuestas solo se utilizarán con fines de investigación y no serán compartidas con ninguna otra persona o institución. Completar el cuestionario tomará aproximadamente 10 minutos

1. Pregunta: ¿Ha tenido alguna vez errores en el cálculo de su salario, pagos extra o deducciones?
 Sí
 No
 No recuerdo
2. Pregunta: ¿Siente que los trámites de vacaciones, incapacidades u horas extras se gestionan con rapidez?
 Sí
 No
3. Pregunta: ¿Le informan a tiempo sobre detalles importantes como fechas de pago, rebajos o días feriados?
 Siempre
 A veces
 Casi nunca
 Nunca
4. Pregunta: ¿Ha tenido dificultades para saber cuántos días de vacaciones ha acumulado o utilizado?
 Sí
 No
 No recuerdo
5. Pregunta: ¿Considera que la información relacionada con su asistencia, incapacidades o permisos es gestionada de forma justa y clara?
 Sí, siempre
 Casi siempre

- A veces
- Casi nunca
- No, nunca
6. Pregunta: ¿Cuál es el tiempo de respuesta para consultas relacionadas con vacaciones, aguinaldo u otros trámites laborales?
- Mismo día
- 1 a 5 días hábiles
- Más de 5 días hábiles
- No he realizado este tipo de consultas
7. Pregunta: ¿Cree que un sistema digital podría mejorar la gestión de Recursos Humanos en la empresa?
- Sí
- Tal vez
- No
8. Pregunta: ¿Cuáles de estos aspectos considera que podrían mejorarse con un sistema digital? (puede marcar más de una opción).
- Cálculo de planilla
- Control de vacaciones
- Trámite de incapacidades
- Registro de asistencia
- Comunicación con el personal
- Otros: _____
9. Pregunta: En una escala del 1 al 5, donde 1 es la calificación mínima y 5 la calificación máxima, ¿cómo calificaría actualmente la forma en que se manejan los procesos de Recursos Humanos?
- 1
- 2
- 3
- 4
- 5
10. ¿Tienes algún comentario o sugerencia para mejorar los procesos de Recursos Humanos?
-
-

Apéndice C: Guía de Observación

GUÍA DE OBSERVACIÓN

Nombre de la Empresa: Quesada Vargas Construcciones S.A.

Actividad de la Empresa: Construcción de obras civiles

OBJETIVO: Observar y evaluar las actividades relacionadas con el proyecto de Sistema web para la gestión de Recursos Humanos en la empresa Quesada Vargas Construcciones S.A, ubicada en Grecia.

No	Aspectos por observar	Cumple	No Cumple	Oportunidad de mejora	Detalle de Observación
1	Registro y almacenamiento de la información del personal				
2	Control de vacaciones e incapacidades				
3	Cálculo de planilla y otros pagos				
4	Acceso rápido a la información de los empleados				
5	Uso de herramientas tecnológicas (Excel, Word, correo, etc.)				

No	Aspectos por observar	Cumple	No Cumple	Oportunidad de mejora	Detalle de Observación
6	Organización y orden en el archivo de documentos				
7	Tiempo dedicado a tareas repetitivas				
8	Flujo de comunicación con los demás empleados				
9	Claridad en los procesos administrativos				
10	Condiciones del espacio de trabajo para su labor				