

UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS

ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADUACIÓN PARA OPTAR POR EL GRADO DE

BACHILLERATO EN INGENIERÍA DE SOFTWARE

Título de la investigación:

**PROPUESTA DE UN SISTEMA WEB PARA LA GESTIÓN DE RECURSOS HUMANOS
EN LA EMPRESA RESTAURANTE TUMI'S HOUSE, UBICADA EN TIBÁS, SAN JOSÉ.**

Nombre del estudiante:

AYRAN DAVID TOLENTINO QUESADA

Tutor:

LIC. FABIÁN RODRÍGUEZ SIBAJA

Sede Central

Abril, 2026

Dedicatoria

Con profundo agradecimiento y humildad, dedico el logro de este trabajo a quienes han sido pilares fundamentales a lo largo de mi camino académico y personal:

A mis padres, pilares fundamentales en mi vida, les dedico este logro con todo mi amor y admiración. Gracias por su esfuerzo incansable, por sus consejos llenos de sabiduría y por la educación que me han brindado con tanto amor y dedicación. Gracias a sus valores, enseñanzas y ejemplo, hoy soy la persona que soy. Cada paso que doy es reflejo del amor, la paciencia y la confianza que han depositado en mí. Sin su guía y su ejemplo, este camino habría sido mucho más difícil.

A mi familia, que ha sido mi sostén en cada etapa de este proceso. Gracias por su comprensión en los momentos de ausencia, por celebrar cada pequeño avance conmigo y por darme siempre un motivo para seguir adelante. Su presencia, cercana o a la distancia, ha sido un recordatorio constante de que no estoy solo, y que todo esfuerzo tiene un sentido más grande cuando se comparte con quienes uno ama.

A mis seres queridos, amigos y personas especiales que, de una u otra forma, formaron parte de este trayecto. Gracias por sus palabras de aliento, por acompañarme en los días de cansancio y por creer en mí incluso cuando yo dudaba. Cada conversación, cada gesto de apoyo, cada sonrisa, dejó una huella que me ayudó a mantenerme firme y con esperanza.

Y también me lo dedico a mí mismo. Por no rendirme, por seguir adelante incluso en los momentos más difíciles. Por las noches de desvelo, por los sacrificios hechos en silencio, por la disciplina y el compromiso con mis metas. Este logro es el resultado de mucho más que esfuerzo académico: es la prueba de que fui capaz de superar los desafíos y seguir creyendo en mí.

Este trabajo no solo representa una meta cumplida, sino también el fruto de todo el amor, el apoyo y la confianza que he recibido de ustedes. A cada uno, les dedico este logro con profunda gratitud y todo mi corazón.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que hicieron posible la realización de este proyecto:

A la señora directora de carrera, Olda Bustillos Ortega, por su liderazgo, su constante apoyo y, especialmente, por su comprensión durante el desarrollo de este trabajo. Su disposición para escuchar, orientar y brindar soluciones en los momentos necesarios fue fundamental para que este proceso se llevara a cabo de forma exitosa.

A mi profesor tutor de proyecto, Fabián Rodríguez Sibaja, por su paciencia, por brindarme siempre su tiempo y por confiar en mis capacidades incluso en los momentos más desafiantes. Su orientación cercana, sus observaciones oportunas y su apoyo constante fueron fundamentales para culminar con éxito este esfuerzo.

Al señor José Figueroa Huamán, por brindarme la oportunidad de desarrollar este prototipo dentro de su empresa, por su disposición, confianza y apertura a colaborar con este proceso académico. Su apoyo permitió que esta experiencia fuera no solo formativa, sino también enriquecedora en lo profesional y personal.

Finalmente a Juan Páez y su familia, por darme la invaluable oportunidad de poder acceder a estudios universitarios y por el apoyo incondicional que han brindado tanto a mí como a mi familia. Su generosidad y compromiso han marcado profundamente mi vida, y han sido fundamentales para hacer realidad este proyecto y muchos sueños más. Gracias por creer en mí con tanta confianza.

CONTENIDO

CAPÍTULO I: INTRODUCCIÓN	29
Planteamiento del Problema	29
Objetivos	30
Objetivo General	30
Objetivos Específicos.....	30
Justificación del proyecto	31
Viabilidad Técnica	31
Desarrollo del software.....	31
Requerimientos Para La Implementación del Software.....	32
Requerimientos Para Espacio Físico.....	33
Viabilidad Operativa	33
Viabilidad Económica.....	34
Viabilidad Legal.....	35
Proyecciones	35
Alcance Funcional	36
Alcance Metodológico	38
Alcance Tecnológico.....	39
CAPÍTULO II: MARCO REFERENCIAL	40
Tecnología Aplicada en el Desarrollo de un Prototipo Funcional	40

Importancia de un Prototipo Funcional en el Desarrollo de Software.....	40
Características de un prototipo funcional.	41
Enfoque Metodológico del Desarrollo.....	42
Enfoque Tecnológico para el Desarrollo de un Sistema	43
Stack tecnológico.....	43
Lenguajes de programación	44
Frameworks.....	44
Bases de datos.	45
Experiencia de Usuario	46
Principios de Usabilidad.	46
Ámbito Tecnológico del Sistema Propuesto	48
Selección de Tecnologías	49
Sistema Web.....	49
Arquitectura del sistema.	49
Gestión de datos y stack tecnológico.....	51
Pruebas de Software.....	53
Sistemas para la Gestión de Recursos Humanos	55
Funciones de los Sistemas de Recursos Humanos.....	56
Impacto de la Digitalización en la Gestión de Recursos Humanos en las PYMES.....	57
Aspectos Legales en la Gestión de Recursos Humanos.....	58

	17
Jornada laboral.....	59
Cálculo de horas extras.....	59
Cálculo de vacaciones.....	60
Cálculo de Aguinaldo y liquidación.....	60
Deducciones de ley.....	61
Incapacidades.....	62
CAPÍTULO III: MARCO METODOLÓGICO.....	63
Enfoques de Investigación.....	63
Enfoque Cuantitativo.....	63
Enfoque Cualitativo.....	64
Enfoque Mixto.....	65
Enfoque de Investigación Seleccionado.....	66
Tipos de Investigación.....	66
Investigación Exploratoria.....	67
Investigación Descriptiva.....	67
Investigación Correlacional.....	68
Investigación Explicativa.....	69
Tipo de Investigación Seleccionada.....	69
Fuentes de Información.....	70
Fuentes Primarias.....	70

	18
Fuentes Secundarias.....	71
Fuentes Terciarias	71
VARIABLES.....	72
Variable Conceptual.....	72
Variable Operacional.....	73
Variable Instrumental.....	74
Población.....	76
Muestra	77
Instrumentos para la Recolección de Datos	78
Proceso para la Recolección y Análisis de datos	79
CAPÍTULO IV: ANÁLISIS DE DATOS.....	80
Cuestionario	80
Pregunta 1: Tiempo trabajando en la empresa	80
Pregunta 2: ¿Cómo calificaría el proceso actual de control de asistencia y horarios en el restaurante?	81
Pregunta 3: ¿Ha tenido problemas con el cálculo de horas extras trabajadas?.....	82
Pregunta 4: ¿Considera que el proceso de pago de nómina es claro y transparente?	83
Pregunta 5: ¿Cómo recibe actualmente la información sobre su salario y horas trabajadas?...	84
Pregunta 6: ¿Con qué frecuencia recibe retroalimentación sobre su desempeño laboral?	85
Pregunta 7: En general, ¿qué tan satisfecho está con el sistema actual de gestión de recursos humanos?	86

Pregunta 8: ¿Cree que el sistema actual impacta en su desempeño laboral?.....	87
Pregunta 9: ¿Cree que la implementación de un sistema digital ayudaría a mejorar la gestión de RRHH en la empresa?.....	88
Pregunta 10: ¿Estaría dispuesto a utilizar un sistema digital para gestionar su asistencia, nómina y horarios?.....	89
Observación	90
Aspecto por observar 1: Manejo de documentación y archivos	90
Aspecto por observar 2: Existencia de criterios claros para la evaluación del desempeño	90
Aspecto por observar 3: Observar el registro de asistencia de empleados	91
Aspecto por observar 4: Observar cómo se calculan y asignan las vacaciones.....	91
Aspecto por observar 5: Tiempo de respuesta y aprobación en solicitudes de permisos	92
Aspecto por observar 6: Eficiencia en el registro y seguimiento de incapacidades y permisos.....	92
Aspecto por observar 7: Detección oportuna de ausencias, retardos y faltas injustificadas.....	93
CAPÍTULO V: PROPUESTA.....	94
Análisis	94
Análisis detallado del Software	94
Módulo controlar asistencia y tiempos de marcas.	95
Módulo gestionar horarios.	95
Módulo gestionar evaluaciones de empleados.....	95
Módulo gestionar permisos.....	95
Módulo calcular liquidación.	96

	20
Módulo calcular vacaciones.....	96
Módulo gestionar incapacidades.....	96
Módulo gestionar planilla y nóminas.....	96
Módulo mantenimientos.	97
Módulo consultas.....	97
Módulo reportes.....	97
Módulo seguridad.	97
Análisis Detallado del Hardware	98
Desarrollo del Prototipo.....	98
Implementación del prototipo.....	98
Casos de Uso.....	100
Diseño	125
Arquitectura del sistema	125
Arquitectura del Software.....	127
Diseño de entradas y salidas	128
Diseño de la Base de Datos.....	132
Diccionario de Datos.....	134
Diseño de Procesos	153
Diagramas UML	159
Programación	167

Entradas y Salidas de datos.....	167
Procesos	170
Módulos Señalados.....	174
Validaciones	183
Pruebas.....	185
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES.....	194
Conclusiones.....	194
Recomendaciones	195
REFERENCIAS.....	198
APÉNDICES.....	201
Cuestionario	201
Guía de observación.....	203

FIGURAS

Figura 1 Diagrama del modelo de cascada en el desarrollo de software.	38
Figura 2 Diagrama de la arquitectura MVC en el desarrollo de software.	50
Figura 3 Estructura de Aplicación Basada en el MERN stack con MySQL.....	52
Figura 4 Distribución del personal según el tiempo laborando en la empresa	81
Figura 5 Percepción del personal sobre el proceso de control de asistencia y horarios	82
Figura 6 Frecuencia de problemas en el cálculo de horas extras trabajadas.....	83
Figura 7 Percepción del personal sobre la claridad y transparencia del proceso de pago de nómina	84
Figura 8 Medios utilizados para la comunicación de la información salarial y horas trabajadas	85
Figura 9 Frecuencia de retroalimentación recibida sobre el desempeño laboral	86
Figura 10 Nivel de satisfacción del personal con el sistema actual de gestión de recursos humanos	87
Figura 11 Percepción del impacto del sistema de gestión de recursos humanos en el desempeño laboral	87
Figura 12 Percepción del personal sobre la implementación de un sistema digital para la gestión de recursos humanos	88
Figura 13 Disposición del personal para utilizar un sistema digital en la gestión de asistencia, nómina y horarios	89
Figura 14 Diagrama UML de Casos de Uso	100
Figura 15 Diagrama de arquitectura del sistema.....	126

Figura 16 Diagrama de arquitectura MVC	127
Figura 17 Diseño de entrada para la página de iniciar sesión.....	128
Figura 18 Diseño de salida para la pantalla de gestión de empleados	129
Figura 19 Diseño de entrada para la pantalla de agregar empleado	130
Figura 20 Diseño de entrada para agregar un turno	131
Figura 21 Diseño de salida para la boleta de pago de nómina.....	132
Figura 22 Diseño de la base de datos.....	133
Figura 23 Diagrama de flujo para registrar marca	153
Figura 24 Diagrama de flujo para solicitar permiso	154
Figura 25 Diagrama de flujo para aprobar o rechazar solicitudes	155
Figura 26 Diagrama de flujo para el cierre de nomina	156
Figura 27 Diagrama de flujo para cierre de liquidación	157
Figura 28 Diagrama de flujo para cierre de aguinaldos.....	158
Figura 29 Diagrama de clases del sistema	160
Figura 30 Diagrama de secuencia de registrar marca	161
Figura 31 Diagrama de secuencia para solicitar permiso	162
Figura 32 Diagrama de secuencia para aprobar o rechazar solicitudes	163
Figura 33 Diagrama de secuencia para el cierre de nomina	164
Figura 34 Diagrama de secuencia para cierre de aguinaldo	165
Figura 35 Diagrama de secuencia para cierre de liquidación	166

Figura 36	Entrada de datos para insertar empleado	167
Figura 37	Entrada de datos para insertar turno	168
Figura 38	Salida de datos para obtener nóminas	169
Figura 39	Salida de datos para obtener turnos asignados	170
Figura 40	Proceso de evaluar incidencias.....	171
Figura 41	Proceso de calcular vacaciones pendientes al liquidar	172
Figura 42	Proceso de calcular horas extra al generar nómina	173
Figura 43	Función del módulo controlar asistencia y tiempos de marca.....	174
Figura 44	Función del módulo gestionar horarios	175
Figura 45	Función del módulo gestionar evaluaciones de empleado	176
Figura 46	Función del módulo gestionar permisos.....	177
Figura 47	Función del módulo gestionar liquidación	178
Figura 48	Función del módulo calcular horas extras	179
Figura 49	Función del módulo calcular vacaciones.....	180
Figura 50	Función del módulo gestionar incapacidades.....	181
Figura 51	Función del módulo gestionar planilla y nóminas.....	182
Figura 52	Función validadora para insertar solicitud de vacaciones	183
Figura 53	Función validadora para insertar empleado.....	184

TABLAS

Tabla 1 Costo del prototipo por día según el salario mínimo en Costa Rica 2026.	34
Tabla 2 Nombre y descripción de los módulos requeridos para el sistema.	36
Tabla 3 Tabla de variables de análisis.	75
Tabla 4 Caso de uso para controlar asistencia y tiempos de marcas.....	101
Tabla 5 Caso de uso para gestionar horarios.....	103
Tabla 6 Caso de uso para calcular liquidación.....	105
Tabla 7 Caso de uso para calcular horas extras.....	107
Tabla 8 Caso de uso para gestionar permisos	109
Tabla 9 Caso de uso para gestionar incapacidades	111
Tabla 10 Caso de uso para gestionar planilla y nóminas	114
Tabla 11 Caso de uso para gestionar evaluaciones de empleados	116
Tabla 12 Caso de uso para calcular vacaciones	118
Tabla 13 Caso de uso para mantenimientos	120
Tabla 14 Caso de uso para consultas.....	121
Tabla 15 Caso de uso para reportes.....	122
Tabla 16 Caso de uso para seguridad.....	124
Tabla 17 Diccionario de datos para tabla Empleado.....	134
Tabla 18 Diccionario de datos para tabla Empleado_Detalle	134

Tabla 19	Diccionario de datos para tabla Género	135
Tabla 20	Diccionario de datos para tabla Estado_Civil	135
Tabla 21	Diccionario de datos para tabla Puesto	135
Tabla 22	Diccionario de datos para tabla TelefonoEmpleado.....	136
Tabla 23	Diccionario de datos para tabla TipoTelefono	136
Tabla 24	Diccionario de datos para tabla CorreoEmpleado.....	137
Tabla 25	Diccionario de datos para tabla TipoCorreo.....	137
Tabla 26	Diccionario de datos para tabla Dirección	137
Tabla 27	Diccionario de datos para tabla Distrito.....	138
Tabla 28	Diccionario de datos para tabla Cantón	138
Tabla 29	Diccionario de datos para tabla Provincia.....	138
Tabla 30	Diccionario de datos para tabla Usuario	139
Tabla 31	Diccionario de datos para tabla Rol_Usuario.....	139
Tabla 32	Diccionario de datos para tabla Tipo_Cese.....	139
Tabla 33	Diccionario de datos para tabla Liquidación.....	140
Tabla 34	Diccionario de datos para tabla LiquidacionDetalle	140
Tabla 35	Diccionario de datos para tabla PeriodoNomina.....	141
Tabla 36	Diccionario de datos para tabla NominaDetalle	141
Tabla 37	Diccionario de datos para tabla Nómina	141
Tabla 38	Diccionario de datos para tabla DeduccionLey	142

Tabla 39	Diccionario de datos para tabla DeduccionEmpleado	142
Tabla 40	Diccionario de datos para tabla CreditoISR.....	143
Tabla 41	Diccionario de datos para tabla PeriodoAguinaldo.....	143
Tabla 42	Diccionario de datos para tabla Aguinaldo	144
Tabla 43	Diccionario de datos para tabla Incapacidad.....	144
Tabla 44	Diccionario de datos para tabla TipoIncapacidad	144
Tabla 45	Diccionario de datos para tabla HorasExtras	145
Tabla 46	Diccionario de datos para tabla TurnoAsignado	145
Tabla 47	Diccionario de datos para tabla TipoTurno	146
Tabla 48	Diccionario de datos para tabla MarcaAsistencia	146
Tabla 49	Diccionario de datos para tabla TipoMarca	146
Tabla 50	Diccionario de datos para tabla IncidenciaAsistencia.....	147
Tabla 51	Diccionario de datos para tabla TipoIncidencia	147
Tabla 52	Diccionario de datos para tabla PermisoLaboral	148
Tabla 53	Diccionario de datos para tabla TipoPermisoLaboral	149
Tabla 54	Diccionario de datos para tabla Vacaciones	149
Tabla 55	Diccionario de datos para tabla VacionesSaldo	149
Tabla 56	Diccionario de datos para tabla AsistenciaDia.....	150
Tabla 57	Diccionario de datos para tabla SolicitudHorasExtra	150
Tabla 58	Diccionario de datos para tabla DiaFeriado	151

Tabla 59	Diccionario de datos para tabla CategoriaEvaluacion	151
Tabla 60	Diccionario de datos para tabla PreguntaEvaluacion.....	151
Tabla 61	Diccionario de datos para tabla EvaluacionCliente	152
Tabla 62	Diccionario de datos para tabla RespuestaEvaluacion.....	152
Tabla 63	Diccionario de datos para tabla Notificacion.....	152
Tabla 64	Caso de prueba Iniciar Sesión.....	185
Tabla 65	Caso de prueba Registrar marcas de asistencia.....	187
Tabla 66	Caso de prueba Generación de nómina.....	188
Tabla 67	Caso de prueba Registro y validación de permisos laborales	190
Tabla 68	Caso de prueba Generación de liquidación.....	191

CAPÍTULO I: INTRODUCCIÓN

La empresa Restaurante Tumi's House es un establecimiento familiar fundado en 2005 ubicado en Tibás, San José. Con más de 19 años en el mercado, el restaurante se especializa en ofrecer lo mejor de la cocina peruana, utilizando insumos, condimentos y hasta artículos de ambientación importados directamente desde el Perú para garantizar una experiencia que refleje la riqueza de la cultura y la tradición del país.

Actualmente, el restaurante cuenta con un equipo conformado por su dueño y administrador, cuatro cocineros, dos saloneros, un bartender y un guarda de seguridad. La empresa busca mejorar la gestión de su talento humano para mejorar sus procesos internos. La finalidad de este proyecto es desarrollar un prototipo funcional de un sistema de recursos humanos que permita una administración más eficiente y así contribuir con el crecimiento y estabilidad del restaurante.

Planteamiento del Problema

- El primer problema identificado es la frecuencia de ausencias y retrasos debido al registro manual de las marcas de entrada y salida en hojas de cálculo. Este método genera imprecisiones al depender de la memoria de los empleados para reportar sus horarios, lo que resulta en errores frecuentes.
- La gestión manual de horarios rotativos provoca superposiciones de turnos y asignaciones incorrectas. Esto lleva a que algunos empleados trabajen horas adicionales, afectando su planificación personal, satisfacción laboral y motivación, además de aumentar el riesgo de rotación en el personal.
- La falta de evaluaciones de desempeño formal y periódica ha generado inconsistencias en el rendimiento de los empleados. Esto dificulta identificar áreas de mejora y limita la capacidad de reconocer a los colaboradores destacados.
- Por otro lado, la ausencia de un registro preciso de permisos provoca desorganización en la asignación de tareas, cobertura de turnos y coordinación de actividades, lo que afecta la eficiencia del equipo y genera tensiones entre sus miembros.

- El cálculo manual de liquidaciones al finalizar la relación laboral de los empleados genera errores y retrasos que provocan descontento entre los trabajadores al no recibir su remuneración adecuada y a tiempo.
- La falta de un sistema para registrar y calcular las horas extras ocasiona errores y malentendidos en los pagos, afectando la motivación de los trabajadores y complicando la gestión de la nómina.
- Un sistema organizado para gestionar las vacaciones de los empleados resulta esencial, ya que su ausencia genera confusión tanto para ellos, al no tener claridad sobre sus días disponibles, como para la administración al planificar los descansos.
- Gestionar manualmente las incapacidades provoca errores en los registros y afecta la remuneración adecuada de los empleados durante sus ausencias por incapacidad.
- Finalmente, los cálculos manuales de nómina y planilla generan inconsistencias que afectan la moral y el rendimiento de los empleados.

Objetivos

Objetivo General

- Desarrollar un prototipo funcional de un sistema web para la gestión de recursos humanos de la empresa Restaurante Tumi's House.

Objetivos Específicos

- Analizar los requerimientos del sistema para la administración y empleados del restaurante.
- Diseñar la estructura del sistema, incluyendo la base de datos, la estructura del backend y la organización de los componentes frontend.
- Desarrollar el prototipo funcional del sistema utilizando el MERN Stack.
- Realizar pruebas funcionales, de usabilidad y de seguridad mediante datos de prueba.

Justificación del proyecto

Como se infiere según lo mencionado anteriormente, este proyecto se lleva a cabo con el objetivo de optimizar la administración y operatividad de Tumi's House a través de un sistema automatizado que permita mejorar el registro de asistencia, la organización de turnos, el manejo de la nómina y la evaluación del rendimiento del equipo de trabajo. Actualmente, la dependencia de procesos manuales genera inconsistencias en los registros, lo que se traduce en repercusiones negativas tanto en el ambiente de trabajo como en el servicio brindado a los clientes.

Viabilidad Técnica

Desarrollo del software.

El desarrollo del sistema para el restaurante se realiza utilizando tecnologías modernas y de software libre, reduciendo costos de licenciamiento y permitiendo flexibilidad en la personalización del sistema. El prototipo es desarrollado en el equipo de cómputo propio del estudiante y está basado en el MERN stack, (con la variante de MySQL en lugar de MongoDB), que incluye ExpressJS, ReactJS y Node.js.

Según MDN Web Docs (2024), React es una biblioteca de JavaScript diseñada para la construcción de interfaces de usuario interactivas. Su enfoque principal es reducir errores en el desarrollo mediante el uso de componentes, los cuales son bloques de código independientes que representan distintas partes de la interfaz. Estos componentes pueden combinarse para formar una UI completa, mientras que React se encarga del proceso de renderizado, permitiendo a los desarrolladores centrarse en la experiencia del usuario y la funcionalidad del sistema.

Para el backend o motor de la aplicación, se utilizará Node.js junto con Express.js. Node.js es un entorno de ejecución basado en JavaScript, diseñado para desarrollar aplicaciones escalables y eficientes del lado del servidor (MDN Web Docs, 2024). Express, por su parte, es un framework minimalista que se ejecuta sobre Node.js y facilita la creación de APIs y servidores web mediante un enfoque modular y flexible (Express.js, 2024). Gracias a esta combinación, se podrá gestionar de manera eficiente las solicitudes de los clientes, el enrutamiento y la manipulación de datos, etc.

Se utilizará MySQL como sistema de gestión de bases de datos. MySQL es una base de datos relacional de código abierto ampliamente utilizada para almacenar y gestionar datos estructurados. Según (2025), MySQL es conocido por su alto rendimiento, confiabilidad y escalabilidad, lo que lo convierte en una excelente opción para aplicaciones web. Además, es compatible con diversas plataformas y ofrece soporte para transacciones, replicación de datos y múltiples motores de almacenamiento.

El prototipo se desarrollará en una laptop Dell Inspiron 5406, equipada con un procesador Intel Core i7-1165G7 de 11ª generación, con una frecuencia base de 2.80 GHz. Cuenta con 8 GB de memoria RAM y un SSD con espacio de 500gb. Además, el equipo opera bajo el sistema operativo Windows 10, siendo este sistema compatible con las herramientas de desarrollo mencionadas anteriormente. Sumado a esto, el desarrollo del sistema también se apoyará en otras herramientas como Figma, librerías externas, GitHub, entre otras, que facilitarán el diseño, implementación y pruebas del proyecto.

Requerimientos Para La Implementación del Software.

Tumi's House es un restaurante de tamaño reducido que opera como una pequeña y mediana empresa (PYME), por lo que sus necesidades tecnológicas no requieren una infraestructura compleja. A nivel de hardware, el sistema podrá ejecutarse sin inconvenientes en una computadora de gama media, preferiblemente mínimo 8gb de RAM, 256GB de SSD, un procesador i5/i7 o Ryzen 5/7 y sería suficiente para gestionar las operaciones administrativas y el almacenamiento de datos.

Además, para brindar mayor flexibilidad y facilitar el acceso a las funcionalidades del sistema, se recomienda contar con una tableta o un teléfono celular, lo que permitirá a los empleados gestionar solicitudes de manera ágil. Asimismo, el uso de estos dispositivos será clave para la implementación del Módulo de Gestión de Evaluaciones de Empleados, permitiendo que los clientes puedan brindar retroalimentación directa sobre el servicio recibido mediante encuestas, mejorando así el control del desempeño del personal.

Requerimientos Para Espacio Físico.

Para los requerimientos de infraestructura y espacio físico, el restaurante ya cuenta con conexión a internet estable y dispone del espacio suficiente para la instalación del equipo necesario. La misma ya cuenta con cámaras de seguridad y un par de computadoras adicionales utilizadas para la gestión del sistema de facturación, caja y monitoreo de seguridad, por lo que se demuestra que la implementación del nuevo sistema pueda integrarse a la distribución actual del restaurante sin requerir modificaciones significativas en su infraestructura.

Viabilidad Operativa

Para cumplir con la viabilidad operativa, el desarrollo del sistema para la empresa buscará priorizar la practicidad y accesibilidad, con la intención de lograr que la implementación no genere complicaciones innecesarias en la operatividad del restaurante. El personal del restaurante no necesita conocimientos técnicos avanzados para utilizar el sistema, ya que se diseñará con una interfaz intuitiva y fácil de usar.

El sistema lo usará tanto la administración, como los empleados y los clientes a la hora de realizar las encuestas. La administración tendrá un control más profundo, ya que contará con accesos y permisos especiales que le permitirán supervisar y gestionar la información clave del restaurante y los empleados. Mientras que los empleados solo podrán interactuar con las funciones relacionadas con su asistencia, horarios y permisos. Por otro lado, no se busca reemplazar al personal existente, sino optimizar y agilizar los procesos administrativos.

No obstante, sí transformará la manera en que se llevan a cabo ciertas tareas, sustituyendo métodos manuales por una solución digital. Por ejemplo, anteriormente los empleados reportaban su asistencia y ausencias a través de mensajes de texto, lo que podía generar confusión y errores en el control de horarios. Con el sistema propuestos, estos procesos serán más precisos y transparentes, permitiendo una mejor gestión más confiable y ética.

Cabe mencionar que, para este trabajo, no se toman en cuenta posibles capacitaciones adicionales para el personal. Se asume que los usuarios podrán adaptarse al uso del sistema sin necesidad de formación especializada, dado que su diseño priorizará que sea intuitivo y de fácil

uso. Cualquier ajuste en términos de capacitación o entrenamiento quedará sujeto a evaluaciones posteriores según las necesidades específicas del restaurante y su equipo de trabajo.

Viabilidad Económica

En este apartado se analizan los costos asociados al desarrollo e implementación del sistema, con el objetivo de determinar si su ejecución es financieramente sostenible. Debido a las circunstancias de este proyecto, el prototipo no representaría una inversión significativa, ya que se basa en el uso de tecnologías de software libre y en la infraestructura existente del restaurante. Sin embargo, el costo del desarrollo sí debe considerarse en términos del tiempo y las fases involucradas en su implementación.

Según el Ministerio de Trabajo (2026), a partir del 01 de enero del año 2026, el salario mínimo de un programador en computación sin título por día es de ₡16.244,50. Por este motivo, si se estimara el costo del desarrollo del sistema con base en este salario mínimo, y considerando las cuatro fases del desarrollo descritas en esta investigación (análisis, diseño, desarrollo y pruebas), el costo total dependería del tiempo invertido en cada una de ellas. Esto se puede observar en la tabla No. 1:

Tabla 1

Costo del prototipo por día según el salario mínimo en Costa Rica 2026.

Etapa de desarrollo	Costo al día	Cantidad de días	Total
Análisis	₡16.244,50	7	₡113.711,50
Diseño	₡16.244,50	14	₡227.423,00
Desarrollo	₡16.244,50	55	₡893.447,50
Pruebas	₡16.244,50	20	₡324.890,00
Total Final	₡16.244,50	116	₡1.559.427,00

Fuente: Tolentino, 2026.

Viabilidad Legal

Para la viabilidad legal del sistema, será fundamental asegurarse de que se cumplan la Ley No. 8968, Ley de Protección de la Persona frente al Tratamiento de sus Datos Personales y el Código de Trabajo de Costa Rica. De esta manera, el restaurante podrá mantener un entorno legalmente seguro para la administración,

Sin embargo, existen ciertas limitaciones que deben considerarse. Una de las principales es la necesidad de contar con el consentimiento informado de los empleados. La Asamblea Legislativa de Costa Rica exige que la recolección y uso de datos personales estén respaldados y autorizados por los individuos, lo cual implica la obligación del restaurante de obtener dicho consentimiento.

Otra limitación está relacionada con las restricciones en el monitoreo del personal. Aunque el sistema contará con herramientas como, por ejemplo, registrar la asistencia y evaluar el desempeño, el Código de Trabajo establece que no se deben implementar mecanismos que vulneren la privacidad de los empleados. Esto requiere que el sistema opere de manera transparente y ética, con el conocimiento pleno de los trabajadores sobre las funcionalidades que los involucran.

Finalmente, la administración tendrá acceso a datos sensibles de empleados lo que implica la obligación de garantizar la confidencialidad de la información. Solo el personal autorizado podrá consultar y gestionar estos datos, y será indispensable implementar en el prototipo funcional, distintas medidas de seguridad para prevenir los accesos no autorizados o fugas de información sensible.

Proyecciones

El desarrollo del prototipo para Tumi's House tiene como objetivo principal optimizar la gestión de los procesos de recursos humanos, mediante la automatización y digitalización de procesos clave, como el control de asistencia, la gestión de nómina, etc. Se pretende proporcionar una herramienta eficiente y fácil de usar que permita a la administración y al personal trabajar de manera más organizada y con menor margen de error. Básicamente se busca mejorar la eficiencia, la toma de decisiones, la satisfacción laboral y reducir los errores humanos.

Alcance Funcional

En esta sección, se incluyen todos los módulos y su descripción, que el sistema requiere para cumplir exitosamente con su funcionalidad:

Tabla 2

Nombre y descripción de los módulos requeridos para el sistema.

Nombre del módulo	Descripción del módulo
Controlar asistencia y tiempos de marcas	El módulo se encargará de registrar los horarios mediante una funcionalidad de marcas, obteniendo datos precisos y en tiempo real, reduciendo el margen de error en el control de ausencias y retrasos.
Gestionar Horarios	Este módulo ofrecerá la flexibilidad de automatizar la asignación de horarios, así como la opción de ajustar manualmente los turnos cuando sea necesario. Además, el sistema tomará en cuenta solicitudes de permisos previamente aprobadas, garantizando que se respeten las ausencias programadas y evitando asignaciones de turnos a empleados que no estarán disponibles en esas fechas.
Gestionar evaluaciones de empleados	El módulo permitirá tener un seguimiento periódico del rendimiento de cada empleado a través de la retroalimentación del cliente. Este módulo incluirá encuestas que los clientes podrán completar para calificar la atención y el servicio recibidos de cada colaborador. De esta manera, la empresa podrá recopilar información valiosa sobre el desempeño individual y de equipo, identificar áreas de mejora, así como reconocer a los empleados destacados.
Gestionar Permisos	El módulo de Gestionar Permisos facilitará a los trabajadores solicitar permisos y ausencias de manera formal, mientras que el administrador del restaurante tiene la posibilidad de aceptar o negar tales peticiones con sencillez. Este módulo ofrecerá un registro en tiempo real de todas las faltas previstas y mantener un control preciso.
Calcular Liquidación	Este módulo permitirá automatizar el cálculo de liquidaciones conforme al código de trabajo y las políticas de la empresa. Generará cálculos precisos para evitar errores y retrasos, permitiendo gestionar las liquidaciones de manera eficiente y transparente.

Calcular Horas Extras	Calcular Horas Extras permitirá el registro y cálculo automático de las horas extras empleadas por los trabajadores, garantizando que los pagos se efectúen conforme a las horas registradas y la tarifa correspondiente según el código de trabajo.
Calcular Vacaciones	Este módulo permitirá registrar y controlar los días de vacaciones de cada empleado. Facilitando la programación de descansos y permite tanto a los empleados como a la administración revisar la disponibilidad y el historial de días tomados, asegurando una gestión ordenada y sin conflictos.
Gestionar Incapacidades	Este módulo facilitará el registro y administración de los días de incapacidad de los trabajadores, garantizando el cálculo preciso de los pagos correspondientes conforme a lo estipulado en el código de trabajo. Además, también facilita la consulta del historial de incapacidades, incrementando la claridad y eficacia en la gestión de estas ausencias y respetando las normas legales para salvaguardar los derechos de los trabajadores.
Gestionar Plantilla y Nóminas	Este módulo facilitará la administración del cálculo de pago de la plantilla, garantizando que todos los conceptos, tales como sueldos, deducciones y horas extras, se implementen de manera adecuada de acuerdo con el código de trabajo. También permitirá la revisión de la información de cada trabajador, y la simplificación de emisión de informes financieros. Además, se conservará un registro de pagos, ofreciendo claridad y exactitud en la gestión de las compensaciones y minimizando posibles fallos administrativos.
Mantenimientos	Este módulo se encargará de realizar el borrado, inserción, modificación, actualización de datos.
Consultas	Este módulo se encargará de generar información proporcionada de las diferentes tablas.
Reportes	Este módulo se encargará de generar información proporcionada de las diferentes tablas y procesos, pero con un formato específico, según lo solicite el usuario. Podrá ser impreso o, por pantalla.
Seguridad	Este módulo se encargará de realizar la autenticación de contraseñas y definición de perfiles.

Fuente: Tolentino, 2026.

Alcance Metodológico

El desarrollo del sistema seguirá el enfoque del Modelo en Cascada (Waterfall), una metodología estructurada y lineal que se utiliza ampliamente en proyectos donde los requerimientos están claramente definidos desde el inicio. Según ProjectManager (2025), este modelo divide el desarrollo en fases secuenciales, como análisis, diseño, implementación, pruebas y mantenimiento, asegurando que cada etapa esté completamente terminada antes de pasar a la siguiente. Esta metodología proporciona una visión clara del progreso del proyecto, facilita la documentación y minimiza la incertidumbre, siendo ideal para proyectos como el sistema del trabajo presente, donde los objetivos están bien establecidos desde el principio.

Se comienza con la fase de Análisis de Requisitos, donde se documenta para asegurar una comprensión clara de lo que se desarrollará. En la fase de Diseño del Sistema, se crean los diagramas de base de datos, arquitecturas y prototipos de interfaz. La fase de Implementación consiste en desarrollar el sistema utilizando las tecnologías de React.js para el frontend, Express.js para el backend y MySQL para la base de datos. Posteriormente, en la fase de Pruebas, se valida el funcionamiento del sistema mediante un proceso exhaustivo para garantizar que cumpla con los requisitos establecidos.

Figura 1

Diagrama del modelo de cascada en el desarrollo de software.



Fuente: <https://www.ionos.es>

Como se menciona anteriormente, el modelo de desarrollo en cascada tiene un enfoque secuencial donde cada fase depende de la finalización de la anterior. En la Figura 1 se ilustran las principales etapas de este modelo, destacando su estructura lineal y los puntos de comprobación que garantizan la calidad del proceso, así como las flechas de retroceso que indican la necesidad de volver a fases anteriores si se encuentran errores.

El que sea un desarrollo estructurado, por un lado, permite una adecuada administración de los recursos, ya que cada fase tiene objetivos y productos claramente definidos que permiten tener un control sobre el avance del proyecto; especialmente en entornos donde se requiere cumplir con tiempo de entrega y con la calidad del producto final. Por otro lado, ya que cada etapa se debe acabar para pasar a la siguiente, se reduce la probabilidad de la existencia de errores acumulativos; lo que ayuda a la estabilidad y confianza del sistema desarrollado.

Alcance Tecnológico

El sistema para Tumi's House se desarrollará utilizando el stack MERN, pero en lugar de MongoDB, se adaptará con MySQL, lo que permitirá gestionar los datos de manera estructurada y eficiente. Este stack incluye React.js para el desarrollo del frontend, lo que proporcionará una interfaz de usuario dinámica, Express.js y Node.js para el backend, que manejarán la lógica del sistema y las operaciones del servidor y MySQL como base de datos relacional.

Se seguirá el patrón de diseño MVC (Modelo-Vista-Controlador), lo que garantizará una separación clara entre la lógica de negocio, la presentación y el manejo de datos. En este modelo, el Modelo se encargará de la estructura y la lógica de los datos utilizando MySQL; la Vista, desarrollada con React.js, ofrecerá a los usuarios una experiencia intuitiva; y el Controlador, implementado con Express.js, conectará el modelo con la vista, gestionando las solicitudes y respuestas del sistema.

Es importante mencionar también que, será un desarrollo web modular, integrando tanto el frontend como el backend en una única aplicación. Esta arquitectura simplificada centralizará todas las funcionalidades, facilitando su desarrollo, despliegue y mantenimiento. Además, se implementará de manera local, aprovechando los recursos tecnológicos disponibles en el restaurante, eliminando la necesidad de servicios externos en la nube y garantizando un mayor control sobre los datos.

CAPÍTULO II: MARCO REFERENCIAL

El objetivo del marco de referencial es proporcionar una base teórica para orientar, en este caso, el desarrollo del sistema, asegurando que se fundamente en principios sólidos y prácticas establecidas en el ámbito tecnológico. En esta sección, se pretende explicar conceptos clave que sustentan el desarrollo del sistema, como qué es un prototipo funcional, que es el desarrollo web, cómo opera la arquitectura MVC, sistemas de software en recursos humanos y cuáles son las ventajas de la digitalización de procesos administrativos en PYMES.

Tecnología Aplicada en el Desarrollo de un Prototipo Funcional

Desarrollar un sistema requiere considerar aspectos técnicos y metodológicos que fundamenten su estructura y funcionamiento. Un sistema bien estructurado no solo facilita la validación del software antes de su implementación final, sino que también permite identificar mejoras y optimizar su rendimiento. Para lograr esto, es fundamental definir con precisión la arquitectura del sistema, seleccionar las tecnologías adecuadas, utilizar una metodología de desarrollo apropiada y establecer criterios, los cuales deben estar respaldados por conceptos teóricos y antecedentes que sustenten su implementación.

Importancia de un Prototipo Funcional en el Desarrollo de Software

El desarrollo de software es un proceso complejo que exige validaciones continuas para así asegurar que el producto final cumpla con cada uno de los requisitos definidos y la satisfacción de las expectativas del usuario. En tal sentido, la construcción de un prototipo funcional ocupa lugar relevante dentro de todo este entorno, debido a que puede habilitar la evaluación, corrección y optimización de la amplia diversidad de aspectos que puede presentar el sistema. Un prototipo funcional no solo habilita la detección de errores en su fase inicial sino que permite el aseguramiento de una mejor experiencia de usuario, la optimización de la eficacia del software y la disminución de los costos y tiempos del desarrollo.

Características de un prototipo funcional.

Un prototipo funcional es una versión preliminar de un sistema de software que simula el comportamiento real una aplicación. Carbo Velez, et al., (2021, citando a Máximo, 2016 y Díaz & Florez, 2020) indica que, "por definición, un prototipo es un primer modelo funcional, aunque incompleto y vago de una aplicación". Además, "un prototipo de software se refiere a una versión no terminada o completa que implementa algunas características de un producto, y que sirve como evaluación y retroalimentación para corregir o añadir nuevas funciones a la aplicación, con el objetivo de perfeccionar detalles a de fin de lograr un mejor resultado final".

A diferencia de un prototipo de baja fidelidad, los cuales se centran únicamente en la apariencia y en la estructura de un sistema, un prototipo funcional incorpora elementos interactivos y operativos. Sus principales características incluyen interactividad, permitiendo a los usuarios probar funcionalidades básicas; validación temprana, facilitando la detección de errores y ajustes antes de la implementación final; escalabilidad, ya que puede evolucionar progresivamente hasta convertirse en el producto final; y alineación con los requerimientos del usuario, sirviendo como referencia para que el software cumpla con las expectativas del cliente.

Hay varios ejemplos de aplicaciones reales de un prototipo funcional en el desarrollo de software. Se pueden aplicar para validar requisitos o funcionalidades, ya que permite comprobar si el conjunto de requerimientos definido es factible y si puede responder a las expectativas del usuario antes de empezar a desarrollarlo por completo. También se pueden aplicar durante las pruebas de integración con otras tecnologías, por ejemplo, puede utilizarse para verificar la compatibilidad del sistema con bases de datos, APIs, servicios en la nube, etc.; en definitiva, para conseguir una implementación sin inconvenientes.

Además, los prototipos funcionales son útiles para capacitación y entrenamiento, ya que permiten que los futuros usuarios se familiaricen con la plataforma antes de su implementación definitiva, reduciendo la curva de aprendizaje. Por otro lado, también suelen utilizarse en presentaciones a inversores o en demostraciones a directivos; en este caso, la funcionalidad del prototipo del sistema sirve como una herramienta visual y práctica para mostrar el valor y el potencial del producto, para dar confianza en su factibilidad y para captar repercusión mediática o inversores potenciales antes de que se produzca su lanzamiento oficial.

Como se infiere según lo explicado en los párrafos anteriores, un prototipo es un modelo inicial de un producto que, aunque no está completamente terminado, sirve como una representación funcional inicial. Su objetivo principal es permitir la evaluación, la experimentación y obtener comentarios para identificar mejoras, corregir errores y agregar nuevas características. Los prototipos son herramientas clave en los procesos de diseño y desarrollo, ya que facilitan la visualización anticipada del producto final y ayudan a optimizar su calidad antes de su implementación final.

Enfoque Metodológico del Desarrollo

La elaboración de un prototipo funcional necesita de una buena metodología que ayude a dar forma al proceso de una forma eficaz e intentar asegurar que los resultados sean los previstos en el objetivo del proyecto. Una buena metodología se convierte en la finalidad de la estructura general del proceso mediante planificación, elaboración y validación de la especificación funcional del prototipo, logrando así, el objetivo de satisfacción del usuario y de las características técnicas.

De acuerdo con Armijos Ortega et al. (2024), las metodologías de desarrollo de software comprenden un conjunto de técnicas, prácticas y herramientas que permiten planificar, diseñar, construir y mantener sistemas de software de manera eficiente. Estas metodologías proporcionan una estructura para todo el ciclo de vida del software, definiendo procesos, roles y actividades necesarias para la correcta gestión del proyecto.

Es por esto por lo que, la selección de un modelo de desarrollo es un aspecto primordial ya que determina cómo se estructuran las etapas del proyecto y que permite representar cómo se procederá con la continuidad del software. Existen diversas metodologías de desarrollo adaptadas según la naturaleza de estas o del proyecto en cuestión, unas y otras con sus ventajas pero también inconvenientes.

Como se ha mencionado previamente en el documento; en este trabajo, la decisión de emplear el modelo de desarrollo en cascada se debe a la necesidad de contar con un proceso estructurado y predecible en el que, antes de la ejecución de cada una de las fases, se deben tener bien definidas las entregas de la fase que haya finalizado. Como las necesidades del sistema están claramente definidas desde un primer momento, esta metodología en cascada permite ejecutar un

plan detallado y previo con el que se logra reducir el riesgo a desviaciones en el desarrollo del sistema.

Este es el caso donde la naturaleza del propio proyecto no requiere contar con iteraciones constantes con el cliente ni cambios en los requerimientos de éste durante el desarrollo o al menos cambios que no requieran una reestructuración significativa que impacte a gran medida el diseño o propósito del sistema. De esta manera, se logra mejorar la eficiencia de la gestión de tiempo y de los recursos asignados, facilitando que cada una de las etapas se encuentre completa con la consistencia y precisión necesaria para lograr que el sistema obtenga el cumplimiento de los objetivos.

Enfoque Tecnológico para el Desarrollo de un Sistema

El proceso de desarrollo del sistema conlleva la elección de un conjunto de tecnologías y patrones que permitan garantizar su correcto funcionamiento, su escalabilidad y su mantenibilidad. Un enfoque tecnológico válido contribuye a una eficiente implementación de la funcionalidad, adecuado rendimiento del sistema y fluidez en la integración de los diferentes componentes tanto software como hardware.

Stack tecnológico.

La elección de tecnologías en el desarrollo de software es un factor determinante para cualquier sistema. MongoDB (s/f) indica que, un stack tecnológico es un conjunto de tecnologías que se combinan para crear cualquier aplicación. Son esenciales para crear aplicaciones web que sean fáciles de mantener y escalables. Los stacks tecnológicos determinan el tipo de aplicaciones que se pueden crear, el nivel de personalización que se puede realizar y los recursos que se necesitan para desarrollar la aplicación.

Los stacks tecnológicos incluyen lenguajes de programación, frameworks, bibliotecas, sistemas de bases de datos y herramientas de infraestructura que interactúan entre sí para el desarrollo, ejecución y mantenimiento de un software. Los componentes que forman parte de un stack se organizan en capas, donde cada capa está encargada de una función determinada, como es

el caso de las capas encargadas de gestionar la interfaz gráfica, procesar la lógica de negocio o almacenar la información.

La elección de un determinado stack tecnológico depende de diferentes cuestiones, entre las que se encuentran los requisitos del proyecto, la escalabilidad deseada, la compatibilidad con otras tecnologías y la facilidad para el mantenimiento. Un stack bien estructurado permite optimizar el rendimiento del sistema, aumentar la seguridad y garantizar que todos los componentes se integren correctamente entre ellos.

Lenguajes de programación.

Un lenguaje de programación es un conjunto estructurado de reglas, símbolos y normas sintácticas que permiten a los programadores escribir instrucciones comprensibles tanto para los humanos como para las máquinas. Su función principal es servir como un medio de comunicación entre la persona y el hardware. Como señalan Galarza y Carriel (2022, citando a Olarte, 2018), los lenguajes de programación actúan como un puente entre la lógica humana y la capacidad operativa de los dispositivos, permitiendo traducir ideas y procesos en comandos que las computadoras pueden interpretar y ejecutar de manera precisa.

A lo largo del tiempo, los lenguajes de programación han evolucionado significativamente, adaptándose a las necesidades tecnológicas y mejorando su accesibilidad y eficiencia. Existen distintos niveles de lenguajes, desde los de bajo nivel, como el ensamblador, que están más cercanos al lenguaje de máquina, hasta los de alto nivel, como Python o Java, que utilizan estructuras más intuitivas para los desarrolladores. Según Galarza y Carriel (2022), esta evolución ha permitido optimizar la forma en que los programadores diseñan y estructuran sus códigos, facilitando la creación de software más complejo y funcional.

Frameworks.

Un framework es un conjunto de herramientas, bibliotecas y reglas que proporcionan una estructura predefinida para facilitar el desarrollo de software. Ferreira (2025), menciona que los frameworks funcionan como una especie de "esqueleto" o plataforma base, sobre la cual los programadores y desarrolladores pueden construir y personalizar sus aplicaciones. Ofrece un

conjunto de prácticas estandarizadas y componentes reutilizables que aceleran el proceso de desarrollo.

En la programación, existen diversos frameworks diseñados para distintos lenguajes y propósitos. Por ejemplo, en el desarrollo web, frameworks como Angular, React y Vue.js son ampliamente utilizados para la creación de interfaces de usuario dinámicas y reactivas. Por otro lado, en el desarrollo backend, Django (Python), Express (Node.js) y Ruby on Rails (Ruby) proporcionan un conjunto robusto de herramientas para la gestión de bases de datos, autenticación de usuarios y manejo de solicitudes HTTP.

Bases de datos.

Un tema muy importante en el ámbito tecnológico es, sin duda, la gestión y el almacenamiento de datos. En la actualidad, la información ha pasado a ser un recurso básico para la toma de decisiones y para el correcto funcionamiento de los sistemas informáticos. Las bases de datos son sistemas que permiten organizar y estructurar grandes volúmenes de información de una manera eficiente, garantizando además su accesibilidad, su integridad y su seguridad.

Según Oracle (2020), una base de datos es una recopilación organizada de información o datos estructurados que se almacenan electrónicamente dentro de un sistema informático. Usualmente, estas bases de datos son gestionadas por un Sistema de Gestión de Bases de Datos (DBMS), el cual permite el acceso, control y administración eficiente de la información. En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas, forman un sistema de bases de datos, facilitando su manipulación y mantenimiento.

Además, Oracle (2020), señala que los modelos más comunes de bases de datos utilizan estructuras de filas y columnas, organizadas en tablas, lo que optimiza la eficiencia en el procesamiento y consulta de datos. Mediante estas estructuras, es posible acceder, modificar y actualizar la información con facilidad. Para ello, la mayoría de las bases de datos emplean un lenguaje de consulta estructurado (SQL), el cual permite gestionar los datos de manera efectiva y estructurada.

Experiencia de Usuario

La experiencia de usuario (UX) en sistemas de software de RRHH ocupa un sitio importante en la eficiencia y productividad de las organizaciones. Un buen diseño permite que los empleados puedan interactuar con el sistema de la manera más fácil y beneficiosa, de la misma forma que se optimizan todos los procesos administrativos, además de disminuir el esfuerzo relacionado con la administración del personal.

Principios de Usabilidad.

La usabilidad es un factor clave para que los sistemas sean eficientes, accesibles y brinden una experiencia satisfactoria a los usuarios. Es la facilidad con la que los usuarios pueden interactuar con un sistema. Según Mógica (2024), Los 10 principios de usabilidad propuestos por Jakob Nielsen son fundamentales para el diseño de interfaces de usuario eficientes y efectivas. Estos principios, también conocidos como heurísticas de Nielsen, sirven como directrices para mejorar la experiencia del usuario en sitios web y aplicaciones. A continuación, se destacan algunos de estos principios:

- **Visibilidad del estado del sistema:** Es esencial que el sistema mantenga informado al usuario sobre lo que está ocurriendo en todo momento, proporcionando retroalimentación adecuada en tiempo razonable. Por ejemplo, al cargar una página o procesar una solicitud, es recomendable mostrar indicadores de progreso que informen al usuario sobre el estado actual del proceso.
- **Relación entre el sistema y el mundo real:** El sistema debe hablar el lenguaje del usuario, utilizando palabras, frases y conceptos familiares, en lugar de terminología técnica. Además, debe seguir las convenciones del mundo real, presentando la información de manera lógica y natural. Por ejemplo, el uso de íconos reconocibles, como una lupa para la función de búsqueda, facilita la comprensión y navegación del usuario.
- **Control y libertad del usuario:** Los usuarios suelen cometer errores al interactuar con el sistema; por ello, es fundamental ofrecer opciones claras para deshacer o rehacer acciones, permitiendo una navegación flexible y sin restricciones. Por ejemplo, proporcionar una

opción de "deshacer" después de eliminar un elemento evita la frustración del usuario y mejora su experiencia.

- **Consistencia y estándares:** Es importante mantener la coherencia en el diseño del sistema, siguiendo convenciones y estándares reconocidos para que los usuarios no tengan que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo.
- **Prevención de errores:** Más que proporcionar buenos mensajes de error, es preferible diseñar el sistema de manera que prevenga la ocurrencia de problemas desde el principio.
- **Reconocer en lugar de recordar:** El diseño del sistema debe reducir la carga cognitiva del usuario al hacer visibles los objetos, acciones y opciones importantes. En lugar de obligar al usuario a recordar información entre diferentes partes del sistema, es recomendable mostrar opciones de manera clara. Un ejemplo de esto es el uso de menús desplegables y autocompletado en formularios, lo que permite a los usuarios seleccionar en lugar de escribir desde cero.
- **Flexibilidad y eficiencia de uso:** Un sistema debe ser útil tanto para usuarios principiantes como para usuarios avanzados. Se pueden incluir funciones como accesos directos o atajos de teclado que permitan a los usuarios con más experiencia realizar tareas de manera más eficiente. Esta flexibilidad mejora la productividad y permite personalizar la experiencia de usuario según sus necesidades.
- **Diseño estético y minimalista:** La interfaz de usuario debe centrarse en lo esencial, evitando elementos innecesarios que puedan distraer o dificultar la navegación. Un diseño limpio y bien estructurado mejora la comprensión de la información y la usabilidad general del sistema. Un ejemplo de esto es el uso de espacios en blanco y una jerarquía visual clara en los sitios web, lo que facilita la lectura y navegación.
- **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores:** Cuando un usuario comete un error, el sistema debe proporcionar mensajes de error claros y comprensibles, explicando el problema y sugiriendo una solución. En lugar de mostrar códigos técnicos de error, los mensajes deben utilizar un lenguaje sencillo y relevante, guiando al usuario sobre cómo corregir la situación.
- **Ayuda y documentación:** Aunque un sistema bien diseñado debería ser fácil de usar sin necesidad de instrucciones extensas, en ocasiones es útil proporcionar documentación o asistencia. Esto puede incluir tutoriales interactivos, preguntas frecuentes (FAQ) o una

función de búsqueda para resolver dudas. Por ejemplo, en plataformas de software empresarial, contar con una sección de ayuda detallada puede mejorar la experiencia del usuario y reducir la necesidad de soporte técnico.

La incorporación de estos principios en los sistemas de recursos humanos resulta fundamental para asegurar que los empleados puedan utilizar la aplicación sin problemas. Un sistema cuya interfaz sea intuitiva y accesible resulta propicio para optimizar los procesos administrativos, por ejemplo, lo que respecta a el control de la asistencia. La visibilidad del estado del sistema es clave a la hora de proporcionar continuamente el feedback al trabajador, lo cual es determinante en los sistemas de RRHH, donde el trabajador necesita la información en tiempo real correspondiente a trámites como por ejemplo, las solicitudes de vacaciones, etc.

Otro aspecto también importante en este tipo de sistemas es la consistencia y la evitación de errores. Dado que en un sistema de RRHH suelen ser utilizados por empleados con distintos niveles de experiencia tecnológica, es importante que la navegación y acciones dentro del sistema, se mantenga lo más claro y repetitivos posible. Por ejemplo, tener siempre la misma forma en las pantallas de solicitud o en los reportes, simplifica ampliamente la adopción del sistema y el uso real del mismo, así como la reducción de la necesidad de formación.

Finalmente, la flexibilidad y la eficiencia marcan la diferencia en este tipo de sistemas, dado que los diferentes usuarios tienen diferentes necesidades. Mientras que los encargados necesitan acceso a informes detallados, por su parte los empleados solo necesitan un acceso simplificado para consultar su información personal o actualizar, si procede, información básica. Incorporar accesos directos, menús intuitivos y asistentes virtuales dentro del sistema facilita la interacción y reduce la curva de aprendizaje. Además, una adecuada documentación permite, en muchos casos, que los usuarios resuelvan dudas de manera autónoma.

Ámbito Tecnológico del Sistema Propuesto

El ámbito tecnológico en el que se desarrolla un sistema es un factor determinante para su eficacia, escalabilidad y sostenibilidad a largo plazo. Comprender este contexto es fundamental para conocer las herramientas, los lenguajes, las plataformas y las arquitecturas más adecuadas para satisfacer las exigencias del proyecto. Además, facilita la alineación entre los objetivos

funcionales del sistema y las capacidades tecnológicas disponibles, permitiendo así un desarrollo más coherente, eficiente y orientado al usuario final.

Selección de Tecnologías

Sistema Web.

En el presente proyecto, el sistema propuesto se puede clasificar como un sistema orientado a la gestión de recursos humanos, diseñado para automatizar y optimizar procesos administrativos. Se trata de un sistema web basado en tecnologías modernas y su propósito es mejorar la ejecución de procesos operativos, así como la toma de decisiones gracias a la digitalización y a la centralización de la información.

Según Coppola (2023), se conoce como desarrollo web al proceso de crear y mantener un sitio web que sea funcional en internet, a través de diferentes lenguajes de programación, según el modelo y la parte de la página que corresponda. Cada sitio tiene una URL única que lo distingue de los demás en la red informática mundial.

El desarrollo web se divide en dos grandes áreas: el frontend y el backend. El frontend se encarga de la interfaz con la que interactúan los usuarios, utilizando tecnologías como HTML, CSS y JavaScript. Por otro lado, el backend gestiona la lógica del negocio, las bases de datos y la seguridad, empleando lenguajes como Node.js, Python, PHP o Java.

Dado lo anterior, el sistema propuesto en este proyecto será una aplicación web, lo que permitirá su accesibilidad desde distintos dispositivos sin necesidad de instalación local. La implementación de una plataforma basada en la web facilita su mantenimiento, escalabilidad y mejora la experiencia del usuario al proporcionar acceso remoto en cualquier momento y lugar.

Arquitectura del sistema.

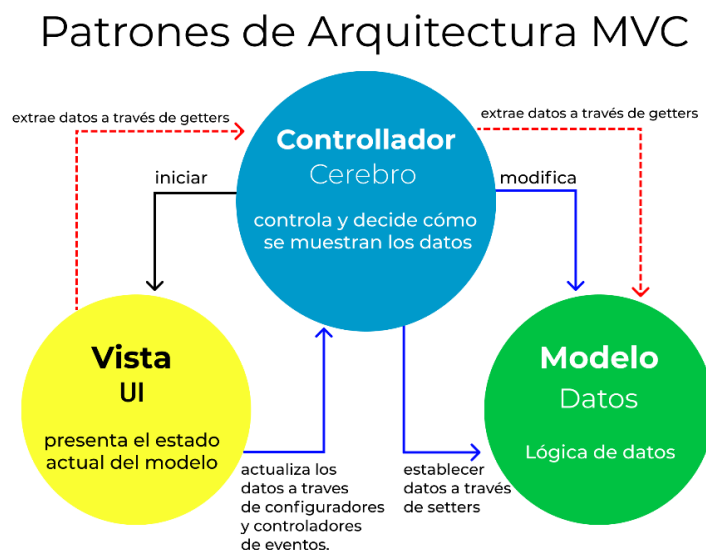
Para lograr un software escalable, mantenible y fácil de desarrollar, es fundamental emplear arquitecturas bien definidas que permitan dividir la aplicación en módulos independientes pero interconectados. En este contexto, el enfoque modular facilita la separación de responsabilidades dentro del sistema, una de las arquitecturas más utilizadas en el desarrollo web es Modelo-Vista-

Controlador (MVC), un patrón que organiza la aplicación en tres capas principales, asegurando una clara diferenciación entre la gestión de datos, la presentación visual y la interacción con el usuario.

Según Hernandez, R. D (2021), el patrón Modelo-Vista-Controlador (MVC) es una arquitectura de software que permite organizar una aplicación en tres componentes principales: Modelo, Vista y Controlador, facilitando la separación de responsabilidades y mejorando la escalabilidad y mantenibilidad del sistema. El Modelo gestiona la lógica de los datos, incluyendo su recuperación, almacenamiento y actualización; la Vista es la interfaz de usuario que muestra la información y permite la interacción con el sistema; mientras que el Controlador actúa como intermediario, procesando las entradas del usuario y coordinando la comunicación entre la Vista y el Modelo. Este enfoque promueve la separación de preocupaciones, permitiendo que los desarrolladores trabajen simultáneamente en diferentes partes del sistema sin interferencias, lo que facilita la gestión y modificación del código.

Figura 2

Diagrama de la arquitectura MVC en el desarrollo de software.



Fuente: <https://www.freecodecamp.org>

Adicionalmente, API REST es un grupo de reglas que sirve para que diversos sistemas se puedan comunicar a través de Internet. Funciona intercambiando datos en forma de JSON o XML y utilizando los métodos estándar de HTTP, que son GET, POST, PUT y DELETE. Este tipo de API se utiliza mucho en el desarrollo web porque consume pocos recursos, es flexible y habilita la integración de tecnologías diferentes.

Cuando se combina con el patrón Modelo-Vista-Controlador (MVC), una API REST permite estructurar las peticiones y respuestas. Así, el Modelo se ocupa de los datos y de las operaciones, el Controlador interpreta la petición y hará de intermediario, y la Vista son la respuesta que se da al cliente en formato JSON. Este patrón permite una separación de las distintas partes del sistema que permiten que cada una de estas partes tenga su razón de ser facilitando el mantenimiento y escalabilidad del proyecto.

Dentro del marco general del sistema que se propone, se va a incorporar a la arquitectura MVC para lograr que el código esté bien organizado. Del mismo modo, la creación de una API REST permitirá definir cómo se deben estructurar las peticiones y respuestas del sistema con el objetivo de facilitar la comunicación entre el frontend y el backend. La API procesará la información del sistema a través del Modelo, mientras que el Controlador interpretará las peticiones realizadas por los usuarios, la cual dirige la información que se envía a la Vista (en este caso, las respuestas en formato JSON). La combinación de ambos implica que se obtendrá una buena aplicación (en términos de organización, escalabilidad y mantenimiento), integrándose con las necesidades del restaurante.

Gestión de datos y stack tecnológico.

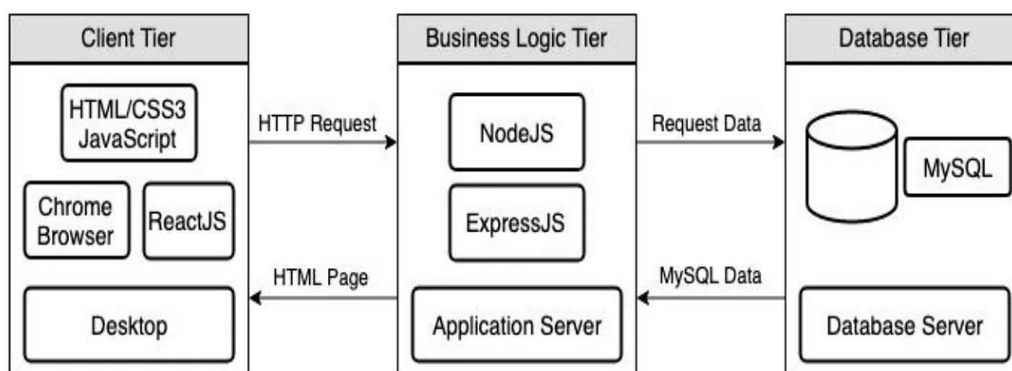
El desarrollo del sistema propuesto se basa en un stack moderno que asegura la escalabilidad, la eficiencia y la facilidad de mantenimiento. Para cumplir con tal propósito, se ha decidido aplicar un conjunto de herramientas y tecnologías que permiten mejorar la gestión de datos, interacción de las personas con el sistema y el procesamiento de la lógica del negocio. La arquitectura del sistema se corresponde con un modelo modular y también garantiza la separación de responsabilidades a la vez que facilita la reutilización de componentes.

Uno de los aspectos clave del desarrollo lo constituye el stack MERN, que está formado clásicamente por MongoDB, Express.js, React.js y Node.js; pero para la realización de este

proyecto se ha considerado sustituir MongoDB por MySQL, que, como sea indicado previamente, es un sistema ampliamente usado e indiscutiblemente valorado de gestión de bases de datos relacional. Esta elección, por contra, permite cubrir la integridad de los datos por medio de diferentes esquemas estructurados, relaciones entre tablas y consultas optimizadas con la implementación del lenguaje SQL.

Figura 3

Estructura de Aplicación Basada en el MERN stack con MySQL.



Fuente: <https://levelup.gitconnected.com>

Según SkillVertex (s.f.), el uso de MySQL en el stack MERN ofrece diversas ventajas, entre ellas:

- Estructura de datos organizada: MySQL permite gestionar información en tablas con relaciones claras, facilitando la administración y búsqueda de datos.
- Familiaridad para desarrolladores: Al ser una base de datos relacional ampliamente utilizada, muchos programadores ya conocen SQL, lo que simplifica su implementación en el stack MERN.
- Mejora en el rendimiento: En algunos escenarios, MySQL puede superar a MongoDB en velocidad, especialmente en procesos que requieren la lectura de grandes volúmenes de datos.
- Soporte para transacciones: MySQL garantiza la consistencia de los datos al permitir transacciones, lo que resulta fundamental cuando se ejecutan múltiples operaciones al mismo tiempo. (párr. 2)

Pruebas de Software.

En cualquier desarrollo de un sistema, las pruebas de software constituyen un paso clave para permitir saber si su funcionamiento y usabilidad es el adecuado. Las pruebas de software tienen como algunas de las funciones comprobar si existen errores, verificar la seguridad sobre los datos y mejorar la experiencia del usuario asegurando así el cumplimiento de los objetivos esperados desde su construcción. A pesar de que en el tipo de proyecto presente no se contempla una implementación real, la validación mediante pruebas funcionales, de usabilidad y de seguridad resulta esencial para demostrar la viabilidad del sistema.

Para tratar de asegurar la calidad y el correcto funcionamiento de un sistema, es fundamental llevar a cabo pruebas de software que permitan detectar errores antes de una implementación. Según IBM (2025), "las pruebas de software son el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe hacer. Entre los beneficios de unas buenas pruebas se incluyen la prevención de errores y la mejora del rendimiento". Esto resulta especialmente relevante en el contexto del desarrollo de sistemas empresariales, donde la fiabilidad y el rendimiento impactan directamente en la operatividad de una organización

Además, la realización temprana de pruebas en el desarrollo del software permite abordar problemas de arquitectura, decisiones de diseño incorrectas y vulnerabilidades de seguridad antes de que el sistema se implemente. Como señala IBM (2025), cuando el desarrollo deja un amplio margen para las pruebas, mejora la fiabilidad del software y se entregan aplicaciones de alta calidad con pocos errores. En el caso de los sistemas de RRHH, esta fiabilidad se traduce en plataformas más eficientes, seguras y fáciles de usar, que optimizan la gestión del talento y mejoran la productividad de la empresa al reducir la carga de trabajo en procesos administrativos y evitar fallos que puedan afectar la operatividad y economía del negocio.

Pruebas de funcionalidad.

Es necesario efectuar pruebas que validen el correcto funcionamiento del sistema antes de la implementación. Las pruebas funcionales permiten encontrar fallos en el sistema, de forma que el producto final sea funcional de acuerdo con todos los requisitos enunciados y que satisfaga al

máximo la experiencia de los usuarios. Según Tamushi (2024), Estas pruebas verifican que cada función de la aplicación opere según los requisitos establecidos. Se centran en la interacción del usuario con el sistema, asegurando que las entradas proporcionadas generen las salidas esperadas.

Un ejemplo común es la validación de inicio de sesión y autenticación, asegurando que los usuarios puedan acceder con credenciales válidas y que las medidas de seguridad, como el bloqueo tras intentos fallidos, funcionen correctamente. De igual forma, las pruebas sobre formularios de que garantizan que las entradas de datos de los formularios cumplen con los formatos adecuados, la evaluación de roles o permisos de un determinado usuario, que garantiza que cada perfil de usuario tendrá acceso a las funciones que le corresponden a su perfil, etc.

Además, incluso se recomienda aplicar estas pruebas en diferentes escenarios y múltiples combinaciones de los datos de entrada para detectar fallos que, en caso contrario, quedarían ocultos en un entorno controlado. Por último, los resultados obtenidos como consecuencia de la realización de estas pruebas deberán ser registrados y analizados con el fin de hacer mejoras en una futura versión del sistema, pasando a un modo de evolución y optimización continua.

Pruebas de usabilidad

Las pruebas de usabilidad son imprescindibles para evaluar la facilidad de uso, la accesibilidad y la experiencia del usuario de un sistema. El objetivo de estas pruebas de usabilidad es confirmar que el usuario puede interactuar con la plataforma de una forma intuitiva y sin esfuerzos o dificultades innecesarias. Tamushi (2024) indica que, “evalúan qué tan fácil es para los usuarios interactuar con el software y si pueden cumplir con sus tareas de manera eficiente”. Estas pruebas permiten identificar obstáculos en la navegación y mejorar la interfaz para optimizar la experiencia del usuario.

Las pruebas de usabilidad incluyen variadas técnicas para evaluar la experiencia del usuario del sistema o la facilidad de uso. Un ejemplo es la prueba de navegación, que consiste en ver si los usuarios encuentran la información que necesitan y completan tareas, evitando la confusión. De igual manera a la anterior, los resultados de estas pruebas deben ser documentados y utilizados para mejorar continuamente el diseño y la experiencia del usuario para que el programa cumpla con los estándares de usabilidad requeridos.

Pruebas de seguridad

Llevar a cabo pruebas de seguridad hace que el sistema sea resistente a vulnerabilidades y ataques que puedan comprometer la integridad, confidencialidad y disponibilidad de la información. Estas pruebas permiten detectar posibles fallos en la protección de datos y en los mecanismos de acceso del sistema. De nuevo, Tamushi (2024), indica que las pruebas de seguridad evalúan la capacidad de un software para protegerse contra amenazas y accesos no autorizados, asegurando que los controles de seguridad implementados funcionen correctamente.

Un ejemplo habitual es la conocida prueba de penetración (pentesting), que consiste en realizar ciberataques con el fin de encontrar posibles vulnerabilidades de seguridad. Del mismo modo, en las pruebas de autenticación se intenta verificar que únicamente los usuarios autorizados puedan acceder a ciertas funciones del sistema utilizando credenciales sólidas y mecanismos de autenticación, como la autenticación multifactor. También se realizan pruebas de cifrado para verificar que la información sensible almacenada y/o transmitida esté protegida de accesos no autorizados.

Sistemas para la Gestión de Recursos Humanos

La gestión de los recursos humanos es una función fundamental en el interior de las organizaciones en la medida que integra una serie de procesos fundamentales como son la gestión del talento, el control de la asistencia, la planificación de los turnos de los trabajadores, la evaluación del desempeño o la gestión de nóminas. La llegada de la tecnología ha dado lugar a la aparición de los sistemas de gestión de recursos humanos, que permiten automatizar y optimizar estos procesos, mejorando así la organización de las compañías y dándoles la posibilidad de fundamentar sus decisiones estratégicas sobre datos.

Según SAP (s/f), Un sistema de para recursos humanos o HRIS es la sigla en inglés para "sistema de información de recursos humanos", una solución de software que ayuda a las empresas a gestionar y automatizar los procesos de RR. HH. centrales y a dar soporte a nómina, administración de beneficios, horas trabajadas y asistencia, y otros flujos de trabajo, así como al almacenamiento de datos del empleado tales como información personal, demográfica y de remuneración. (p. 4).

Otro aspecto clave de los sistemas de gestión de recursos humanos es su capacidad de integración con otras herramientas empresariales. Actualmente, muchas soluciones de RRHH están diseñadas para sincronizarse con software de contabilidad, plataformas de comunicación interna y sistemas de planificación empresarial (ERP), lo que permite una administración más eficiente y centralizada.

En la situación actual, en donde la transformación digital está redefiniendo el modo en el que las empresas desarrollan su actividades, un sistema de gestión de RRHH eficaz no es simplemente una ventaja competitiva sino que se convierte en una necesidad; ya que la digitalización de estos procesos abarata costes, reduce errores y permite que el departamento de recursos humanos se oriente hacia nuevas tareas que cogen el protagonismo, como pueden ser la retención del talento o el desarrollo organizacional.

En los siguientes apartados, se explorarán las principales funcionalidades y su impacto en la administración del talento. Se analizarán los precedentes y ejemplos de sistemas de RRHH, las herramientas tecnológicas que los componen y las funciones esenciales que desempeñan en la optimización de procesos como la contratación, la gestión de nómina, el control de asistencia y la evaluación del personal.

Funciones de los Sistemas de Recursos Humanos

Los sistemas de gestión de recursos humanos han evolucionado para integrar una serie de funciones esenciales que optimizan la administración del capital humano en las organizaciones. Según Manatal (2025), una de las principales funciones es la gestión de datos de empleados, la cual centraliza y organiza información detallada sobre cada trabajador, incluyendo datos personales, historial laboral y registros de formación. Esto facilita la toma de decisiones basada en datos actualizados y accesibles.

La gestión de tiempo y asistencia es otro componente esencial, ya que permite supervisar horarios, ausencias y permisos, proporcionando herramientas para el control de asistencia y la planificación de turnos. Este módulo ayuda a reducir inconsistencias y facilita la administración eficiente del tiempo laboral (SAP, s.f.). Esta función les permite a las empresas a tener un mayor control sobre la productividad y la eficiencia operativa, asegurando que los empleados cumplan con sus horarios de manera precisa y transparente.

Además, los sistemas incluyen módulos de formación y desarrollo, permitiendo gestionar programas de capacitación, seguimiento de competencias y planes de carrera para promover el crecimiento profesional dentro de la empresa. (Manatal, 2025). Finalmente, Fortia (2025) menciona que los sistemas modernos ofrecen un portal de autoservicio para empleados, donde los trabajadores pueden acceder a su información personal, actualizar datos, solicitar permisos y consultar nóminas sin necesidad de intermediarios, mejorando la transparencia y autonomía en la gestión de sus propios datos.

Aunque estas funciones constituyen solo a algunos de los componentes más relevantes de los sistemas de gestión de recursos humanos, su alcance podría ser todavía mayor, siempre y cuando se presenten como parte de aquellos sistemas ERP que, directamente o a través de la interconexión de diferentes sistemas, articulen aquellos y otros procesos organizacionales, esto es, la gestión financiera, la logística y la gestión de los Recursos Humanos; de esta forma, la organización obtiene una mayor visión del mismo. La automatización y digitalización de estos procesos permitir cada vez más no sólo la eficiencia, sino también la capacidad de adaptación a entornos empresariales con graves modificaciones y evoluciones, asegurando de esta forma el cumplimiento de la gestión.

Impacto de la Digitalización en la Gestión de Recursos Humanos en las PYMES

La expansión y adopción de los sistemas digitales, HRIS o ERP en el contexto empresarial resultan en un elevado potencial de impacto por su capacidad de optimizar la gestión de recursos humanos, propiciando a las empresas la mejora de la administración del talento en la empresa misma, la planificación de turnos, la administración de nóminas, la valoración del rendimiento de los trabajadores y la automatización de procesos clave en este sentido, favoreciendo la mayor eficiencia y reducciones de errores administrativos.

En complemento, Chinchilla Bravo (2022) señala que, en un estudio realizado en conjunto por la Escuela de Administración de Empresas, el TEC y la UNED, sobre la digitalización en las PYMES costarricenses, se ha identificado un avance significativo en la adopción de herramientas digitales. Tecnologías como el uso de redes sociales con fines comerciales (85.7%) y la banca digital (74.8%) han sido ampliamente adoptadas. Sin embargo, el estudio destaca que los sistemas

ERP han sido una de las soluciones más implementadas, con una valoración de 3.5 sobre 5, evidenciando la creciente necesidad de digitalización en la administración empresarial.

Este ascenso relacionado con la digitalización empresarial está vinculado a la mejora de la gestión de los recursos humanos, puesto que estos sistemas conllevan una mejor administración del capital humano al cambiar la carga de trabajo, permitiendo de mejor forma autogestionar la asignación de recursos. Además, las empresas enfrentan desafíos como los costos de inversión en tecnología, lo que hace crucial el desarrollo de este tipo de soluciones que se puedan adaptar a necesidades específicas de las organizaciones.

El 91,1% de las empresas encuestadas indicó que realizó cambios o mejoras en sus productos o servicios, mientras que el 87,1% lanzó nuevos productos o servicios, lo que representa una leve mejora en comparación a años anteriores. Además, el 88,8% reportó haber implementado cambios en sus procesos productivos, y el 85,6% adquirió nuevos equipos, también mostrando un ligero avance respecto a años anteriores. Asimismo, el 88,8% mencionó que realizó mejoras en la dirección y gestión de sus empresas, el 88,1% en sus procesos de compras y aprovisionamiento, y el 90,9% en comercialización y ventas, manteniendo la tendencia de incremento. (Instituto Tecnológico de Costa Rica, 2022).

En definitiva, la digitalización de los procesos de gestión de los recursos humanos de las PYMES indica transformación en la forma en que las empresas gestionan su talento humano y optimizan sus propios procesos internos. La introducción de los sistemas ERP y HRIS provee incluso de una mayor eficacia operacional, disminuyendo de este modo los errores administrativos y facilitando así una mejor toma de decisiones basadas en datos exactos. A pesar de los desafíos asociados a la inversión en tecnología, la automatización y digitalización de estos procesos seguirá creciendo, debido a la necesidad de una mayor competitividad y de adaptación a un entorno empresarial más ágil.

Aspectos Legales en la Gestión de Recursos Humanos

La administración del recurso humano, en Costa Rica, está regida por una serie de disposiciones laborales establecidas fundamentalmente en el Código de Trabajo, así como también en los reglamentos o directrices de la Caja Costarricense de Seguro Social (CCSS) y el Ministerio de Trabajo y Seguridad Social, las cuales entienden aspectos como el cálculo de vacaciones, el

pago del aguinaldo, el cumplimiento de los aportes patronales o laborales a la seguridad social y demás.

Si bien el sistema que se propone no reemplaza la responsabilidad legal del empleador, busca ofrecer herramientas que faciliten la administración de estos procesos conforme a lo establecido por la ley. Esta sección desarrolla los principales elementos legales que deben tomarse en cuenta para asegurar que el sistema contribuya al cumplimiento normativo.

Jornada laboral.

El Código de Trabajo de Costa Rica, en su Capítulo Segundo, establece las bases legales sobre las jornadas laborales, diferenciando entre jornada diurna, nocturna y mixta. Cada tipo de jornada tiene una duración máxima legal. La jornada diurna no debe superar las 8 horas diarias ni las 48 semanales, la nocturna está limitada a 6 horas diarias y 36 semanales, y la mixta a 7 horas diarias y 42 semanales. Estas limitaciones buscan proteger la salud del trabajador y garantizar condiciones laborales dignas, evitando la sobrecarga de trabajo.

Existen excepciones a estas reglas para ciertos cargos específicos, no obstante, incluso en estos casos, la ley establece que su jornada no puede exceder las 12 horas diarias y debe incluir al menos una pausa para descanso. Además, el Código exige que los empleadores mantengan un registro preciso de las jornadas laborales, indicando claramente el inicio y finalización de cada turno. Este control es clave para verificar el cumplimiento de las disposiciones legales y para la correcta liquidación de horas ordinarias y extraordinarias.

Cálculo de horas extras.

El trabajo realizado más allá de las jornadas establecidas se considera hora extraordinaria. Según BGA (2024), una hora extra es el tiempo que un trabajador dedica a sus funciones más allá de su jornada laboral establecida. Por ejemplo, si la jornada laboral es de 8 horas y el trabajador permanece una hora adicional, esta hora se considera extra. Es importante señalar que las horas extras no se deben a la mera decisión del trabajador de quedarse más tiempo; deben contar con la previa autorización del patrono para ser válidas y pagadas.

Por otro lado, el Ministerio de Trabajo y Seguridad Social indica en el Artículo 139 que las horas extras deben ser remuneradas con un 50% adicional al salario habitual. Por ejemplo, si el salario ordinario es de 2,000 colones por hora, el pago por cada hora extra será de 3,000 colones. Cabe recalcar que también se aplica la normativa respecto a los límites máximos de jornada establecidos, así como la obligación de respetar los tiempos mínimos de descanso y la prevención del agotamiento físico y mental del trabajador.

Cálculo de vacaciones

En el Artículo 153 de El Código de Trabajo establece que todo trabajador tiene derecho a un período mínimo de quince días hábiles de vacaciones al cumplir cincuenta semanas de trabajo continuo con el mismo patrono. Estas vacaciones deben ser remuneradas con el salario ordinario que el trabajador perciba al momento de tomarlas. Es importante destacar que las vacaciones deben concederse de forma continua, salvo que exista un acuerdo entre las partes para dividir las en dos períodos.

En cuanto a la liquidación de vacaciones no disfrutadas, si el contrato de trabajo finaliza antes de que el trabajador haya gozado de sus días de descanso, el trabajador tendrá derecho, como mínimo, a un día de vacaciones por cada mes trabajado y el patrono está en la obligación de pagar las vacaciones acumuladas, según el cálculo correspondiente; en el momento del retiro de la ocupación.

Cálculo de Aguinaldo y liquidación.

El aguinaldo es un derecho laboral establecido por la Ley No. 2412, el cual consiste en un pago adicional anual que todo patrono debe realizar a sus trabajadores. El aguinaldo debe ser entregado a más tardar el 20 de diciembre de cada año y corresponde al promedio de los salarios ordinarios y extraordinarios devengados por el trabajador entre el 1.º de diciembre del año anterior y el 30 de noviembre del año en curso. En caso de que el contrato de trabajo finalice antes del mes de diciembre, el trabajador tiene derecho a recibir el aguinaldo de forma proporcional al tiempo laborado dentro del período correspondiente. Este pago debe hacerse junto con la liquidación final.

El cálculo del aguinaldo se realiza sumando todos los salarios ordinarios y extraordinarios percibidos durante ese período de 12 meses, y dividiendo el resultado entre doce. Es decir: $\text{Aguinaldo} = (\text{Suma de salarios de diciembre a noviembre}) \div 12$. Deben incluirse en el cálculo todas las remuneraciones fijas y variables que constituyan ingreso para el trabajador, como horas extra, comisiones, incentivos o bonificaciones. No se incluyen viáticos no sujetos a salario ni beneficios que no representen ingresos directos.

Por otro lado, la liquidación laboral es un conjunto de pagos que el empleador debe realizar a un trabajador cuando finaliza la relación laboral, ya sea por despido sin responsabilidad patronal, renuncia voluntaria o finalización de contrato. El cálculo de la liquidación varía según el tipo de salida, pero debe contemplar los derechos adquiridos por el trabajador durante el tiempo de servicio.

Los principales rubros que se deben incluir en una liquidación son: salario pendiente, vacaciones acumuladas no disfrutadas, aguinaldo proporcional y, en caso de despido sin responsabilidad del trabajador, también se debe pagar la preaviso o su sustitución en dinero y auxilio de cesantía (si corresponde). Cada uno de estos componentes tiene su base legal en el Código de Trabajo y otras normativas complementarias.

Deducciones de ley.

En Costa Rica las deducciones de ley son rebajos obligatorios que afectan el salario bruto de todos los trabajadores, ya que tienen como finalidad el cumplimiento de obligaciones establecidas por la jurisdicción nacional. Las deducciones obligatorias están reguladas, esencialmente, por la Caja Costarricense de Seguro Social (CCSS). Entre estas deducciones se puede mencionar el Impuesto sobre la Renta, aplicado a trabajadores cuyos ingresos superan el mínimo exento establecido anualmente por la Dirección General de Tributación. Este impuesto se calcula de manera escalonada según tramos de salario, y su recaudación busca financiar el funcionamiento del Estado costarricense.

Otra deducción obligatoria es el aporte del trabajador a la seguridad social, que equivale a un 10.5% de su salario bruto. Dicho porcentaje, se distribuye entre el seguro de salud, el invalidez, vejez y muerte (IVM) y a otros fondos complementarios. Las contribuciones las gestiona la CCSS, y les permiten a los trabajadores acceder a los servicios médicos, pensiones y a otros beneficios

sociales. El empleador debe realizar, por su parte, aportes patronales adicionales que no se descuentan al trabajador pero que se deben gestionar de forma paralela.

Incapacidades

Las incapacidades laborales están reguladas por la Caja Costarricense de Seguro Social (CCSS) y tienen como objetivo proteger el ingreso de los trabajadores cuando, por motivos de salud, no pueden desempeñar temporalmente sus funciones. Para que una incapacidad sea reconocida, debe ser emitida por un profesional médico autorizado por la CCSS, y reportada dentro del plazo establecido.

Cuando un trabajador es declarado incapacitado, la CCSS cubre un porcentaje del salario, el cual varía dependiendo del tipo de incapacidad. En el caso de enfermedad común, el subsidio cubre el 60% del salario promedio reportado en la planilla. Este beneficio se otorga a partir del cuarto día de incapacidad; los primeros tres días suelen estar a cargo del patrono según políticas internas o convenios colectivos.

El cálculo de la incapacidad se basa en el promedio de salarios reportados a la CCSS en los tres meses previos al inicio de la condición. Es importante que el empleador mantenga actualizada su planilla, ya que cualquier inconsistencia puede afectar el subsidio recibido por el trabajador.

CAPÍTULO III: MARCO METODOLÓGICO

El objetivo del presente capítulo de esta investigación es describir la metodología utilizada, detallando el enfoque adoptado, el tipo de investigación seleccionado y los criterios que respaldan estas elecciones. También se especifican los métodos y técnicas empleadas para la recolección y análisis de datos. Esto es muy importante ya que proporciona una estructura clara que permite mejorar la planificación, ejecución y validación del sistema, garantizando la viabilidad técnica y operativa mediante la realización de métodos y procedimientos adecuados.

Enfoques de Investigación

En el proceso de una investigación científica, el enfoque metodológico toma un papel crucial, ya que define el camino que va a tomar la investigación en cuanto a la recolección de datos, su análisis e interpretación. Elegir el enfoque correcto no es un proceso injustificado, sino determinado por la forma de definirse el problema a investigar y los objetivos que se plantean, en términos generales se pueden enumerar tres grandes enfoques de investigación: cuantitativo, cualitativo, y mixto; cada uno de ellos con sus propias características, métodos e intenciones.

Escoger un enfoque de investigación no solo va a influir en cómo se diseñan los instrumentos para la recolección de datos, sino el cómo se va a interpretar los resultados y las explicaciones que se van a dar en las conclusiones. Además, determinará la validez y fiabilidad de las conclusiones, permitiendo que la investigación se ajuste a los propósitos iniciales. Un enfoque bien definido permite estructurar la investigación en coherencia con lo planteado, facilitando así la obtención de datos relevantes e incluso de su análisis.

Enfoque Cuantitativo

El enfoque cuantitativo en la investigación se basa en la recopilación y análisis de datos numéricos con el propósito de identificar patrones, establecer relaciones entre variables y comprobar hipótesis de forma objetiva. De acuerdo con Creswell y Creswell (2023), este enfoque se centra en la comprobación de teorías mediante el análisis de la relación entre variables o la comparación entre grupos. Para ello, las variables son medidas a través de instrumentos que

permiten obtener datos cuantificables, los cuales posteriormente son analizados mediante procedimientos estadísticos. Además, se caracteriza por plantear problemas de investigación claramente definidos y preguntas orientadas en aspectos específicos y medibles.

Su aplicación resulta especialmente útil en estudios donde se busca medir variables y establecer relaciones entre ellas. Al basarse en procedimientos sistemáticos y estructurados, permite obtener resultados comparables y generalizables a distintos contextos. Por esta razón, es ampliamente utilizado en investigaciones que requieren sustentar la toma de decisiones en evidencia empírica obtenida mediante métodos estadísticos y matemáticos.

Este enfoque sigue un proceso secuencial y organizado, en el que cada etapa se desarrolla de manera lógica, lo que facilita la replicabilidad del estudio y la validación de los resultados en diferentes escenarios. Asimismo, el uso de instrumentos como encuestas, experimentos y pruebas estandarizadas permite comprobar o refutar hipótesis con base en evidencia objetiva. También se incorporan mecanismos para reducir sesgos y controlar posibles explicaciones alternativas, con el fin de garantizar la validez de los hallazgos y su posible generalización a una población más amplia.

Enfoque Cualitativo

El enfoque cualitativo en la investigación se centra en la comprensión de los significados que los individuos o grupos atribuyen a un problema social o humano. Así lo señalan Creswell y Creswell (2023), quienes indican que este enfoque se orienta a explorar e interpretar las experiencias, percepciones y contextos en los que se desarrollan los fenómenos estudiados. Se caracteriza por la formulación de preguntas abiertas y por la recolección de datos en el entorno natural de los participantes, lo que permite captar la realidad desde su propia perspectiva.

Su utilización está especialmente indicada en aquellos estudios en los que se busca comprender la complejidad de las situaciones sociales y humanas, más allá de la medición numérica. Además, al centrarse en los significados individuales y en la interpretación de los datos, el enfoque cualitativo permite obtener una visión profunda y detallada de los fenómenos, considerando el contexto en el que ocurren.

En este sentido, Creswell y Creswell (2023) destacan que el enfoque cualitativo sigue un proceso inductivo, en el cual el análisis parte de datos particulares para construir temas o patrones generales. Asimismo, se caracteriza por su flexibilidad, ya que la estructura del estudio puede adaptarse conforme avanza la investigación. A través de técnicas como entrevistas, observaciones y análisis de contenido, se busca interpretar los datos y comprender la realidad desde la perspectiva de los participantes, enfatizando la complejidad de las situaciones analizadas.

En conclusión, la investigación cualitativa se caracteriza por su flexibilidad y su enfoque interpretativo, y constituye un marco idóneo para el análisis de los fenómenos sociales. Su uso resulta fundamental en los estudios en los que se requiere la comprensión de la complejidad de las experiencias humanas, desvelando los significados, las interacciones y las dinámicas sociales presentes en la realidad en la que se indaga.

Enfoque Mixto

El enfoque de investigación mixto es una estrategia que surge de la necesidad de abordar problemas complejos mediante la combinación de métodos cuantitativos y cualitativos, con el fin de obtener una comprensión más completa de los fenómenos estudiados. De acuerdo con Creswell y Creswell (2023), este enfoque implica la recolección, el análisis y la integración de datos numéricos y descriptivos dentro de un mismo diseño de investigación, lo que permite generar conclusiones más amplias a partir de la información obtenida.

Su aplicación resulta especialmente útil en aquellos estudios donde un solo enfoque no es suficiente para explicar la realidad, ya que permite complementar los resultados cuantitativos con interpretaciones cualitativas. De esta forma, se logra un análisis más profundo, considerando tanto la medición objetiva de variables como la comprensión de los significados y contextos en los que se desarrollan los fenómenos.

El enfoque mixto no sustituye a los enfoques tradicionales, sino que integra sus principales fortalezas, permitiendo superar sus limitaciones individuales. Además, se caracteriza por seguir un diseño estructurado en el que se define cómo y en qué momento se combinan los datos, favoreciendo una interpretación más completa y coherente de los resultados. Esto contribuye a la generación de conocimiento más sólido, facilitando la toma de decisiones fundamentadas y su aplicación en distintos contextos de estudio.

Enfoque de Investigación Seleccionado

Dado a la naturaleza del presente proyecto, se llevará a cabo el enfoque de investigación de tipo mixto para evaluar la viabilidad y funcionalidad del prototipo del sistema desde una perspectiva tanto metodológica como teórica; es decir, se realizará una comparación entre el análisis cuantitativo de la parte técnica del sistema y la evaluación cualitativa de la aplicabilidad de este en la gestión de recursos humanos.

Desde una perspectiva cuantitativa, es posible evaluar la estructura del sistema desde el punto de vista del rendimiento y de la mejora de la gestión de datos o el tratamiento de la información revisando, por ejemplo, tiempos de respuesta, el número de registros o capacidad de almacenamiento, el tratamiento de la información, etc. Asimismo, se revisará que las soluciones propuestas sean coherentes con los problemas que se han detectado y con las necesidades de la digitalización en gestión de recursos humanos.

Desde un enfoque cualitativo, se analizará la aplicabilidad del sistema dentro del contexto del proyecto, utilizando métodos como entrevistas, encuestas o revisión documental para evaluar la percepción sobre la digitalización de la gestión de recursos humanos. A través de estas técnicas, se podrá indagar en las necesidades, expectativas y posibles desafíos que enfrentan las empresas al implementar sistemas similares, permitiendo fundamentar el diseño del prototipo en base a información relevante y experiencias previas.

Tipos de Investigación

La selección del tipo de estudio es un elemento crucial, puesto que fija la forma en la que se asume el problema de investigación y cómo se procedería para obtener y analizar los datos. No se puede escoger aleatoriamente cualquiera de los tipos de investigación, sino que los estudios dependen para su elección de los niveles de aspiración y de lo que se quiere investigar. En general, se entienden los tipos de investigación como las diferentes maneras de clasificarlas, existiendo varias maneras, destacando entre todas ellas la investigación exploratoria, descriptiva, explicativa, experimental y correlacional, entre otras.

Investigación Exploratoria

Es una metodología utilizada para examinar problemas poco estudiados o desconocidos en profundidad. Su objetivo es generar una comprensión inicial del fenómeno de estudio, permitiendo la identificación de ideas clave, variables y conceptos relevantes que pueden guiar investigaciones más estructuradas en el futuro. Según Vásquez *et al.*, (2023), este tipo de investigación "busca descubrir ideas, conceptos y variables relevantes para investigaciones posteriores más detalladas y rigurosas" (p. 25). Para lograr este objetivo, se basa en técnicas como entrevistas informales, encuestas, grupos focales y análisis de documentos, lo que facilita la recopilación de información preliminar sin necesidad de hipótesis rígidas.

Pese a que la investigación exploratoria permite obtener una serie de ventajas, también tiene una serie de limitaciones a tener en consideración. La limitación más importante es que con esta investigación no se alcanza a ofrecer por sí solo los argumentos suficientes para la fundamentación de decisiones, pero sí el de ofrecer una visión parcial y, así como información preliminar sobre el problema en cuestión. Es necesario ser complementado con otras investigaciones más estrictas si se piensa utilizar para la fundamentación de decisiones.

Investigación Descriptiva

Se enfoca en detallar y caracterizar un fenómeno o situación con precisión, permitiendo analizar y comprender sus propiedades y tendencias. Su finalidad principal es la obtención y examinación de datos, de manera que pueda dar respuesta a la totalidad del objeto de estudio. Vásquez *et al.*, (2023) indica que,

La investigación descriptiva es importante en la investigación científica porque proporciona una descripción precisa y completa del fenómeno o situación estudiados. La investigación descriptiva puede utilizarse para describir características y propiedades del fenómeno, identificar patrones y tendencias, y explorar relaciones entre variables. Además, la investigación descriptiva es útil para desarrollar teorías y modelos que explican el fenómeno o situación. (pp. 27-28)

Otro aspecto importante según Vásquez *et al.*, (2023) es que, “proporciona una base sólida para investigaciones posteriores más rigurosas. La investigación descriptiva puede utilizarse para identificar preguntas de investigación importantes y desarrollar hipótesis que se pueden probar en investigaciones posteriores más rigurosas” (p. 28).

Para conseguir esto, se utilizan diversos métodos para lograr el objetivo, entre los que se destacan las encuestas, la observación y el análisis de datos. Las encuestas, por ejemplo, permiten obtener información de una muestra representativa de la población en estudio, y así permitir obtener datos susceptibles de análisis cuantitativo para dar cuenta de tendencias y correlaciones significativas.

Investigación Correlacional

La investigación correlacional permite la identificación de asociaciones con diferentes factores, simplificando el progreso hacia la construcción de hipótesis y la elaboración de estudios posteriores. Se utiliza ampliamente en áreas como la psicología, la sociología, la educación y la gestión de empresas en las que resulta esencial construir conexiones entre variable sin realizar una manipulación entre las mismas.

Vásquez *et al.*, (2023) explica que, es un método de estudio que permite examinar la relación existente entre dos o más variables sin modificarlas de manera intencional. Este enfoque se centra en medir el grado de asociación entre los elementos analizados, sin que ello implique una relación de causa y efecto. Facilita la identificación de conexiones entre variables, permitiendo la formulación de hipótesis para estudios más profundos. Su utilidad radica en la posibilidad de detectar posibles asociaciones significativas, lo que puede servir como base para investigaciones futuras con mayor nivel de rigor metodológico.

La investigación correlacional es una herramienta importante para el análisis de datos en diversas áreas, ya que permite detectar patrones y relaciones entre variables. Esto facilita la orientación de futuras investigaciones sin necesidad de aplicar una intervención directa, lo que permite trabajar a partir de los datos obtenidos. Además, su aplicación es clave en estudios que buscan identificar tendencias y correlaciones sin alterar el entorno de estudio.

Investigación Explicativa

La investigación explicativa busca entender el porqué de los fenómenos, es decir, identificar qué causa un determinado evento o comportamiento y cómo se relacionan diferentes variables entre sí. Según Vásquez et al., (2023), este enfoque se basa en la manipulación de una variable con el propósito de observar su impacto en otra, permitiendo así establecer conexiones causales entre ellas. Su importancia en el ámbito científico radica en su capacidad para proporcionar explicaciones detalladas sobre distintos fenómenos.

Es útil para comprender fenómenos complejos y tomar decisiones informadas. Sin embargo, su implementación puede presentar desafíos, como la dificultad para controlar todas las variables que influyen en un estudio y los costos asociados a los experimentos. A pesar de estas limitaciones, este enfoque sigue siendo una herramienta clave en la generación de conocimiento y en la validación empírica de hipótesis científicas.

Tipo de Investigación Seleccionada

Para el desarrollo del presente proyecto, el tipo de investigación más adecuado es la investigación descriptiva. La investigación descriptiva resulta adecuada porque posibilita el estudio y la información de las prácticas y los procedimientos que son utilizados en la actualidad en la gestión de recursos humanos de la empresa y que manifiestan una serie de necesidades concretas y de problemas que se requiere atender con el sistema, tales como el control de asistencia, la planificación de horarios, la gestión de la nómina y la evaluación del rendimiento.

Según Vásquez et al., (2023), este tipo de investigación se enfoca en caracterizar fenómenos sin intervenir directamente en ellos. Esto es ideal para describir el impacto que podría tener la digitalización en la administración de personal dentro de la empresa ya que el proyecto no contempla la implementación real del sistema, sino el diseño y validación de un prototipo funcional.

En este contexto, la metodología seleccionada permitirá recopilar información a través de la revisión documental, el análisis de casos y la consulta de estudios sobre la implementación de software de RRHH en PYMES, lo que facilita la identificación de las mejores prácticas en cuanto a la digitalización de los procesos llevados a cabo por la administración, así como recoger las

funcionalidades más relevantes relacionadas con el sistema para el restaurante. A su vez, el uso de datos previos y la existencia de estudios sobre el sistema permitirán comprobar la viabilidad del prototipo.

Fuentes de Información

Las fuentes de información permiten recopilar, analizar y validar datos para el estudio de un fenómeno. Se pueden presentarse en diversos formatos, como libros, artículos científicos, documentos digitales, etc. Existen fuentes primarias, que contienen información original y de primera mano; fuentes secundarias, que analizan o interpretan datos previamente recopilados; y fuentes terciarias, que organizan y sintetizan información proveniente de múltiples referencias.

Fuentes Primarias

Las fuentes primarias suponen las informaciones puras que no llevan ningún tipo de interpretación o análisis previo. Constituyen el punto de partida de cualquier investigación ya queda acceso a una información original sobre un acontecimiento o un fenómeno dado. De acuerdo con Rodríguez (2025), las fuentes primarias incluyen materiales como publicaciones seriadas, documentos gubernamentales, artículos de noticias y registros audiovisuales históricos. Estas fuentes tienen la particularidad de que reflejan la información de primera mano, lo que les otorga un alto grado de confiabilidad para investigaciones académicas y científicas.

Si bien las fuentes primarias pueden aportar información muy importante, su acceso puede resultar difícil dependiendo de la disponibilidad y conservación de los diferentes documentos. En la mayoría de los casos estas fuentes proceden de archivos históricos, bibliotecas o incluso bases de datos restrictivas. Por este motivo el investigador debe de complementar su trabajo a partir de alguna metodología o herramienta que facilite adquirir y analizar esta información; para que así, la información utilizada sea pertinente y confiable en el desarrollo del estudio que se esté desarrollando.

Fuentes Secundarias

Las fuentes secundarias representan información que ha sido analizada, interpretada o reorganizada a partir de fuentes primarias. Estas fuentes permiten acceder a conocimientos previamente estudiados y estructurados, facilitando el proceso de investigación. Rodríguez (2025), menciona que las fuentes secundarias incluyen libros de texto, resúmenes, reseñas, artículos de revisión y documentos que interpretan datos primarios. Su importancia radica en que brindan un contexto más amplio sobre un tema, permitiendo a los investigadores fundamentar sus estudios en análisis previos y teorías establecidas.

En contra parte a las primarias, una de las grandes ventajas de las fuentes secundarias es su accesibilidad y facilidad de consulta, ya que suelen encontrarse bien organizadas en bibliotecas, bases de datos, repositorios académicos, etc. Gracias a estas fuentes es posible economizar tiempo en la recolección de información, ya que son datos que han sido previamente elaborados. Es importante que los investigadores evalúen la credibilidad y la actualidad de éstas para no caer en sesgos o interpretaciones erróneas de sus estudios.

También es importante que el investigador complemente el análisis con los datos primarios siempre que le sea posible. Así se lograría un balance entre información original y su correspondiente interpretación, fortaleciendo así la validez del estudio. En este sentido, el uso combinado de primarias y secundarias permite ahondar en el fenómeno investigado, logrando así un nivel de calidad en el trabajo académico o científico.

Fuentes Terciarias

Las fuentes terciarias cumplen la función de organizar, compilar y estructurar la información proveniente de fuentes primarias y secundarias. Estas fuentes no presentan información original ni análisis propio, sino que sirven como herramientas de referencia para acceder a datos previamente estudiados. Según Rodríguez (2025), dentro de las fuentes terciarias se encuentran los índices, compendios, bases de datos bibliográficas y fuentes de referencia como diccionarios y enciclopedias. Su propósito es facilitar la localización de información relevante para el investigador, permitiéndole acceder a diversas fuentes de manera más eficiente.

Esas fuentes ayudan a los investigadores a localizar documentos importantes sin tener que examinar ampliamente una gran cantidad de material. Además, en entornos académicos y científicos, las bases de datos bibliográficas y los catálogos de publicaciones juegan un papel fundamental en la gestión y recuperación de información especializada, facilitando la estructuración del marco teórico de cualquier investigación.

Sin embargo, el uso de fuentes terciarias debe ser con criterio, ya que los datos ofrecidos por este tipo de material tienden a ser referencial y no necesariamente se ocupan en realizar análisis exhaustivos. Es conveniente utilizarlas en conjunto con fuentes primarias y secundarias. En este sentido, su función es servir como un punto de partida para la investigación, proporcionando acceso organizado a recursos confiables y facilitando la exploración de un campo de estudio específico.

Variables

Las variables son fundamentales en el contexto de la investigación científica puesto que permiten identificar y medir las características de los objetos de estudio; entenderlas e identificarlas correctamente, permite alcanzar un nivel adecuado de análisis. Según Rodríguez, *et al.*, (2021) las variables “son entidades abstractas que toman diversos valores o modalidades. Son atributos de los objetos de estudio abstractos o concretos. Su naturaleza variante se determina por las condiciones de contorno espaciotemporales que caracterizan a las unidades de análisis” (p. 33).

En términos generales, una variable se puede definir como una determinada característica o propiedad que puede obtener diferentes valores y cuya variación puede llegar a ser observada y medida. Por lo tanto, la altura, la edad o la temperatura son unos ejemplos de variables a las cuales se les puede observar cierto tipo de variabilidad entre los individuos o elementos observados. Estas diferencias permiten establecer comparaciones y relaciones significativas entre los datos recopilados. Además, la correcta definición de las variables fortalece la validez y confiabilidad de los resultados obtenidos en una investigación.

Variable Conceptual

Las variables conceptuales definen de manera abstracta fenómenos, atributos o eventos que se desean investigar. Es imprescindible conocer y delimitar correctamente esas variables para llevar a cabo un análisis coherente con el objetivo del estudio. Rodríguez, et al., (2021), indica que un concepto, “es una representación mental de fenómenos, atributos, objetos, sucesos o cualidades a la que se asocia una definición, la cual desarrolla el significado del término. La definición involucra un problema semántico cuyo objetivo principal es dilucidar los significados que suministran las bases conceptuales de la comunicación entre pares” (p. 43).

En este sentido, una variable conceptual hace referencia a una idea o una representación teórica que describe fenómenos no observables de una forma directa pero dada la cual se puede llegar a valorarse mediante indicadores. Ejemplos de variables conceptuales pueden ser la inteligencia, la motivación o la satisfacción; variables que requieren ser definidas teóricamente antes que ser operacionalizadas en el estudio. La correcta conceptualización es positiva para generar una mayor coherencia en el desarrollo de la investigación y propiciar una mayor facilidad en la interpretación de los resultados que se obtienen.

Variable Operacional

Las variables operativas posibilitan el paso de conceptos abstractos a elementos observables y medibles. Consecuentemente facilitan el análisis de datos y que los resultados sean válidos y fiables. Según Arroyo (2022), la dimensión operativa de una variable implica establecer “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” (pp. 5-6). Esto significa que las variables operacionales permiten transformar fenómenos abstractos en datos concretos que pueden ser analizados empíricamente.

Por lo tanto, cualquier variable operacional, en este sentido, se deriva de los hechos observables y objetivos que permiten medir el cambio o la transformación de una determinada propiedad. Estas variables deben caracterizarse por establecer normas para medirla, por definir los datos que deben usarse para determinar exactamente el cambio o transformación que se produzca en un determinado fenómeno y por fijar patrones de análisis que faciliten la interpretación de los resultados obtenidos. De esta forma, el fenómeno que se mide esté siendo medido de tal manera que pueda ser reproducido en distintas investigaciones.

Variable Instrumental

Las variables instrumentales sirven como herramientas para registrar de forma detallada y precisa los cambios o modificaciones que experimentan las características de los objetos de estudio. Arroyo (2022), comenta que la dimensión instrumental “tiene por función facilitar el registro completo y detallado de la mutación que experimenta la entidad de la cosa, en ocasiones no sensible a ser captada de forma sensorial; esto facilita su cuantificación o cualificación” (p. 7). Lo que quiere decir es que la función principal es proporcionar instrumentos que permitan recolectar datos de manera objetiva.

Se apoya en un conjunto de enunciados o preguntas que, organizados de manera lógica, conforman un instrumento de medición; permitiendo identificar la presencia, frecuencia o magnitud de la variable, haciendo que la medición sea consistente y precisa. Además, para ser válido, debe cumplir con dos requisitos: garantizar la repetibilidad de los resultados al medir la misma variable varias veces y asegurar la congruencia con la definición conceptual de la variable. Por ejemplo, si la variable conceptual es la inteligencia, el instrumento utilizado debe estar diseñado específicamente para evaluar esa capacidad mental, y no otro atributo como la memoria o la creatividad.

Tabla 3*Tabla de variables de análisis.*

Objetivo	Variable	Variable Conceptual	Variable Operacional	Variable instrumental
Analizar los requerimientos del sistema para la administración y empleados del restaurante.	Requerimientos del sistema	Según Visure Solutions (2022), los requisitos funcionales describen lo que debe hacer un sistema, mientras que los requisitos no funcionales imponen restricciones sobre cómo funciona el sistema.	Observación Encuestas	Cuestionario Guía de observación
Diseñar la estructura del sistema, incluyendo la base de datos, la estructura del backend y la organización de los componentes frontend.	Diseño del sistema	Según GeeksforGeeks (2025), el diseño de sistemas es el proceso de definir la arquitectura, componentes, módulos, interfaces y datos de un sistema con el fin de satisfacer requisitos específicos. Su objetivo es crear una estructura bien organizada y eficiente que cumpla con el propósito previsto, considerando factores como escalabilidad, mantenibilidad y rendimiento.	Casos de uso Diagramas Diseño de interfaces	Figma Draw.io
Desarrollar el prototipo funcional del sistema utilizando el MERN Stack.	Desarrollo del prototipo	Un prototipo es una versión preliminar de un sistema que simula el comportamiento real una aplicación. Carbo Velez, et al. (2021, citando a Máximo, 2016 y Díaz & Florez, 2020) indica que: " un prototipo es	Creación de las funcionalidades principales del sistema	Visual Studio Code MySQL Workbench

Objetivo	Variable	Variable Conceptual	Variable Operacional	Variable instrumental
		un primer modelo funcional, aunque incompleto y vago de una aplicación		
Realizar pruebas funcionales, de usabilidad y de seguridad mediante datos de prueba.	Evaluación del Sistema	Según Certus (2021), las pruebas de software son un proceso fundamental en el desarrollo de sistemas, ya que permiten identificar y corregir errores antes de su implementación final. Estas pruebas garantizan que el software funcione correctamente y cumpla con los requisitos establecidos, asegurando así la entrega de un producto de calidad al usuario o cliente.	Casos de prueba	Guía de casos de prueba

Fuente: Tolentino, 2026.

Población

Al realizar una investigación es necesario definir la población objeto de estudio, dado que ésta es la totalidad de los elementos sobre los cuales se desea obtener información. En el caso del desarrollo de un sistema de gestión de recursos humanos, la población podría estar constituida por todos los empleados de la empresa, quienes deberán utilizar el software que se va a desarrollar y cuyos requerimientos deben ser tenidos en consideración en su diseño.

Zambrano Monserrate et al. (2023), define que "La población, en estadística, se refiere a la recolección completa de todas las observaciones de interés para el investigador." (p. 15). En este sentido, cualquier análisis o implementación de software debe basarse en el entendimiento de dicha población. De este modo, en el caso del sistema propuesto, la población de estudio la conformarían los empleados del restaurante, en tanto que van a ser las personas que más se beneficien de la

digitalización de los procesos de recursos humanos. La inclusión de esta población permitirá asegurar que el sistema diseñado va a estar condicionado a las necesidades que dicho colectivo tiene en términos funcionales y operativos.

Dado que muchas veces no resulta posible estudiar a toda la población debido a restricciones de tiempo y/o recursos, se tiende a recurrir a técnicas que permitan caracterizarla sin la necesidad de incluir la totalidad de los individuos en el estudio. Para ello, se emplea el análisis de una muestra representativa de la población, de modo que las decisiones en las que puede tomarse en el desarrollo del sistema se encuentren sustentadas por datos reales y aplicables a la totalidad de los individuos.

Muestra

La limitación de estudiar todas las personas que componen la población objeto de estudio se resuelve a través de la elección de una muestra. Zambrano Monserrate et al. (2023), explica que "En estadística la muestra es una porción o parte representativa de la población que es escogida para ser estudiada debido a que la población es demasiado grande, lo que dificulta estudiarla en su totalidad." (p. 15), es decir, un conjunto de la población objeto de estudio que permite obtener información interesante sin tener que estudiar todas las personas participantes. En el caso de un sistema de gestión de recursos humanos, la muestra puede ser generada por el conjunto de empleados de distintas áreas de la empresa para procurar que las opiniones, así como las necesidades de todos los perfiles, sean tenidas en cuenta a la hora de preparar el diseño del software.

En el contexto del presente proyecto, la población de estudio es finita y está conformada por un total de 9 personas, quienes representan a los empleados y administrador del restaurante Tumi's House. La población se encuentra claramente delimitada, ya que solo se tomarán en cuenta los individuos que interactuarán directamente con el prototipo del sistema. Para calcular la muestra se utiliza la siguiente fórmula:

$$n = \frac{NZ^2pq}{e^2(N-1) + Z^2pq}$$

Donde según (Alcoba Zenteno, 2023):

- n = tamaño de la muestra
- N = tamaño de la población
- Z = nivel de confianza.
- p = variabilidad positiva
- q = variabilidad negativa
- e = precisión absoluta (margen de error)

Para un nivel de confianza del 85%, el valor de Z es aproximadamente 1.44. La variable p representa la probabilidad de éxito o proporción esperada de la población que presenta el fenómeno en estudio; cuando no se dispone de información previa, se utiliza el valor conservador de 0.5 (50%) para maximizar la variabilidad. Por su parte, q corresponde a la probabilidad de fracaso, calculada como $q = 1 - p$, por lo que si $p = 0.5$, entonces $q = 0.5$. Finalmente, d representa la precisión o margen de error máximo admisible en términos de proporción, el cual se ha establecido en 5% (0.05). Sustituyendo los valores:

$$n = \frac{(1.44)^2(9)(0.5)(0.5)}{(0.05)^2(9 - 1) + (1.44)^2(0.5)(0.5)}$$

En este caso, el cálculo del tamaño de la muestra utilizando la fórmula indicada da como resultado 8.67, lo que significa que se debería considerar una muestra de 9 personas (redondeando al entero más cercano). Dado que la población es pequeña ($N=9$), en este caso la muestra incluiría prácticamente a todo el personal del restaurante, siendo así, un análisis representativo para el estudio del sistema.

Instrumentos para la Recolección de Datos

Son herramientas utilizadas en la investigación para obtener información relevante sobre un fenómeno de estudio, permitiendo recopilar, registrar y analizar datos de manera estructurada, facilitando la obtención de evidencia que respalde los objetivos de la investigación. En el presente, se emplearán dos instrumentos: encuestas y observaciones. La encuesta es una herramienta ampliamente utilizada en investigaciones que permite obtener datos estructurados de manera

rápida y eficiente. Por otro lado, la observación permitirá evaluar el comportamiento de los empleados y practicas dentro de la empresa.

Proceso para la Recolección y Análisis de datos

Como se menciona anteriormente, se aplicarán en este caso dos métodos, que son la encuesta y la observación. En el caso de la encuesta, la información será tomada de los empleados y del administrador del restaurante para poder saber la percepción sobre la gestión de recursos humanos, su satisfacción con los procesos actuales y las dificultades a las que se están viendo abocados en cuanto a la administración; en el caso de la observación, los datos obtenidos darán lugar a una secuenciación objetiva sobre cómo se están llevando a cabo los procesos en la práctica con la finalidad de obtener posibles carencias, errores y oportunidades de mejora. En la sección de Apéndices se presenta el detalle sobre los instrumentos que se han utilizado para cada una de estas metodologías, incluyendo el cuestionario de la encuesta y la guía de observación.

CAPÍTULO IV: ANÁLISIS DE DATOS

En este apartado se presenta la fundamentación del diseño y el objetivo de las herramientas utilizadas para el estudio de la gestión de recursos humanos en el restaurante Tumi's House. Se apeló a dos técnicas: el cuestionario (encaminado a recoger la percepción del personal respecto a la asistencia, la nómina, la retroalimentación y el uso de la tecnología) y la guía de observación (enfocada en identificar conductas, rutinas, o fallos en los procedimientos del día a día de gestión del personal).

Cada instrumento se diseñó para recoger información válida de modo que se construyese un sistema adecuándose a las necesidades reales del entorno en el que se trabaja. A continuación se especifican los objetivos a los que se dirigen cada instrumento y a la interpretación de la datos recolectados

Cuestionario

El cuestionario fue diseñado como una herramienta clave para explorar la percepción del personal del restaurante respecto a todos los procesos que componen la gestión de este. Su objetivo principal es identificar áreas de mejoras por lo que cada pregunta fue elaborada con base en problemáticas detectadas previamente en el entorno laboral, lo que permite orientar el desarrollo de la solución a las necesidades reales del equipo. A continuación, se detallan los objetivos específicos que justifican la inclusión de cada uno de los ítems del cuestionario.

Pregunta 1: Tiempo trabajando en la empresa

Esta pregunta tiene como finalidad poder determinar el grado de experiencia y de conocimiento que tiene la persona colaboradora sobre los procesos internos del restaurante, ya que el tiempo de permanencia puede estar relacionado directamente con la percepción de eficiencia o de deficiencia sobre la actuación de las gestiones. Además permite segmentar las respuestas en base a la antigüedad y poder determinar si la percepción de los problemas es distinta entre la gente que es nueva y la gente que hace tiempo que trabaja en la empresa.

Figura 4

Distribución del personal según el tiempo laborando en la empresa

1. Tiempo trabajando en la empresa:



Fuente: Tolentino, 2026.

Los resultados muestran que la mayoría del personal encuestado cuenta con una antigüedad significativa en la empresa, ya que un 38 % indicó tener más de tres años laborando en el restaurante, mientras que un 25 % corresponde a colaboradores con menos de seis meses. Esto evidencia la coexistencia de personal con amplia experiencia y personal de reciente incorporación. Esta diversidad de tiempo permite obtener percepciones variadas según el tiempo en la empresa sobre los procesos en la gestión de recursos humanos, lo cual resulta relevante para el análisis y la propuesta de mejora del sistema actual.

Pregunta 2: ¿Cómo calificaría el proceso actual de control de asistencia y horarios en el restaurante?

Dicha pregunta se orienta a poder saber cuál es la percepción que tiene el personal respecto a uno de los elementos más importantes, el control de asistencia y control de la jornada laboral. Su justificación se encuentra en que un sistema ineficaz puede ser un foco de desorganización, de falta de control sobre el cumplimiento de los horarios y de conflicto laboral, lo que implica una necesidad de evaluar su situación en la actualidad.

Figura 5

Percepción del personal sobre el proceso de control de asistencia y horarios

2. ¿Cómo calificaría el proceso actual de control de asistencia y horarios en el restaurante?



Fuente: Tolentino, 2026.

Los resultados evidencian que la mayoría del personal percibe el proceso actual de control de asistencia y horarios como regular, ya que un 63 % de las personas encuestadas seleccionó esta opción. Además, un 25 % considera que el sistema es deficiente, mientras que solo un 13 % lo califica como eficiente. No se registraron valoraciones en las categorías de muy eficiente ni muy deficiente.

Esta distribución refleja una percepción predominantemente neutral a negativa sobre el control de asistencia y la gestión de horarios, lo que sugiere la existencia de debilidades en los procedimientos actuales. Dichos resultados indican que se debe mejorar este proceso, dado su impacto directo en la organización del trabajo y el cumplimiento de la jornada laboral, reforzando la necesidad de una solución tecnológica orientada a optimizar su gestión.

Pregunta 3: ¿Ha tenido problemas con el cálculo de horas extras trabajadas?

La intención de esta pregunta es identificar posibles incoherencias que pueden haberse producido en el cálculo de las horas extras, una de las áreas más sensibles para el trabajador. Su justificación está en la necesidad de asegurar un trato correcto y legal de la retribución del trabajo adicional y que también va a incidir el tema de la motivación y satisfacción que el trabajador pueda tener.

Figura 6

Frecuencia de problemas en el cálculo de horas extras trabajadas

3. ¿Ha tenido problemas con el cálculo de horas extras trabajadas?



Fuente: Tolentino, 2026.

Los resultados indican que una parte significativa del personal ha experimentado problemas con el cálculo de las horas extras trabajadas, ya que un 50 % de las personas encuestadas manifestó haber tenido inconvenientes algunas veces, mientras que un 13 % indicó que estos problemas se presentan frecuentemente. Por otro lado, un 13 % señaló que estas situaciones ocurren rara vez, y un 25 % afirmó no haber tenido problemas.

Estos datos evidencian que, aunque no se trata de una problemática constante para todo el personal, sí existe una recurrencia relevante de inconsistencias en el cálculo de las horas extras. Esta situación puede afectar la confianza del trabajador hacia la gestión administrativa, por lo que se identifica como un aspecto crítico a considerar en la mejora de los procesos y en la justificación del desarrollo de un sistema que permita un control más preciso y transparente.

Pregunta 4: ¿Considera que el proceso de pago de nómina es claro y transparente?

La finalidad es comprobar el nivel de confianza que los colaboradores tienen en la información y notificación de pago. Una nómina poco clara o entendible puede fomentar la desconfianza, las inconformidades y, en general, un clima que no sea positivo dentro de la organización. Justificando así la necesidad de procesos más claros, documentados y, preferiblemente, automatizables.

Percepción del personal sobre la claridad y transparencia del proceso de pago de nómina

4. ¿Considera que el proceso de pago de nómina es claro y transparente?



Fuente: Tolentino, 2026.

Los resultados reflejan una percepción mayoritariamente neutral a negativa respecto a la claridad y transparencia del proceso de pago de nómina. Un 50 % manifestó una postura neutral, un 38 % indicó estar en desacuerdo con que el proceso sea claro y transparente y solo un 13 % expresó estar totalmente de acuerdo. No se registraron respuestas en las categorías de acuerdo ni totalmente en desacuerdo. Esta distribución sugiere que una parte importante del personal no cuenta con una comprensión clara del proceso de pago de nómina o no percibe suficiente transparencia en su ejecución. Se evidencia la necesidad de fortalecer los mecanismos de información, documentación y automatización del proceso, reforzando así la justificación para la implementación de un sistema de gestión más estructurado.

Pregunta 5: ¿Cómo recibe actualmente la información sobre su salario y horas trabajadas?

Este ítem permite conocer los canales de comunicación explícitos en la empresa de transmisión de información financiera al trabajador. La justificación que encuentra en esta pregunta es la evaluación del nivel de formalidad y accesibilidad del proceso: por ejemplo, un envío verbal puede ser dado lugar a errores o malentendidos, mientras que un documento digital facilita la trazabilidad.

Figura 8

Medios utilizados para la comunicación de la información salarial y horas trabajadas

5. ¿Cómo recibe actualmente la información sobre su salario y horas trabajadas?



Fuente: Tolentino, 2026.

Los resultados evidencian que la principal forma en que el personal recibe la información sobre su salario y horas trabajadas es mediante comunicación verbal con el encargado, opción seleccionada por un 63 % de las personas encuestadas, mientras que un 38 % indicó recibir dicha información a través de otros medios, sin registrarse el uso de documentos físicos o digitales como mecanismos formales de comunicación. Esta situación pone de manifiesto un bajo nivel de formalización y trazabilidad en la transmisión de información relacionada con la nómina y las horas laboradas, reforzando así la necesidad de implementar procesos más estructurados y digitales que permitan un acceso claro, verificable y transparente a esta información.

Pregunta 6: ¿Con qué frecuencia recibe retroalimentación sobre su desempeño laboral?

La retroalimentación es fundamental para el progreso profesional. Esta pregunta intenta explorar las condiciones que pudieran existir en torno a la fomentación de una cultura en la que se pueda mantener el diálogo y la evaluación de manera constructiva. Se justifica añadiendo esta pregunta por la necesidad de promover un entorno de trabajo más eficiente con respecto a la colaboración y la motivación.

Figura 9

Frecuencia de retroalimentación recibida sobre el desempeño laboral

6. ¿Con qué frecuencia recibe retroalimentación sobre su desempeño laboral?



Fuente: Tolentino, 2026.

Los resultados muestran una baja frecuencia de retroalimentación sobre el desempeño laboral, ya que un 50 % de las personas encuestadas indicó nunca recibir retroalimentación, mientras que un 38 % señaló que esta se presenta rara vez y únicamente un 13 % manifestó recibirla algunas veces; no se registraron respuestas en la categoría de frecuentemente. Esta situación evidencia una debilidad en los procesos de comunicación y evaluación del desempeño, lo que puede afectar la motivación, el desarrollo profesional y la mejora continua del personal, reforzando la necesidad de implementar mecanismos formales que promuevan una cultura de retroalimentación constante y constructiva dentro de la organización.

Pregunta 7: En general, ¿qué tan satisfecho está con el sistema actual de gestión de recursos humanos?

Esta pregunta permite obtener una visión general del nivel de satisfacción del personal con los procesos administrativos actuales. Su objetivo es servir como indicador global de desempeño del área de RRHH, y su justificación se encuentra en la importancia de alinear la gestión del talento humano con el bienestar y percepción del trabajador.

Figura 10

Nivel de satisfacción del personal con el sistema actual de gestión de recursos humanos

7. En general, ¿qué tan satisfecho está con el sistema actual de gestión de recursos humanos?



Fuente: Tolentino, 2026.

Los resultados reflejan un nivel de satisfacción predominantemente bajo respecto al sistema actual de gestión de recursos humanos, ya que un 50 % de las personas encuestadas indicó estar poco satisfecho, mientras que un 13 % manifestó no estar nada satisfecho. Por otra parte, un 38 % señaló estar satisfecho, y no se registraron respuestas en la categoría de muy satisfecho. Esta distribución evidencia que una parte significativa del personal percibe deficiencias en los procesos actuales de gestión de recursos humanos, lo que resalta la necesidad de implementar mejoras orientadas a fortalecer la eficiencia administrativa y el bienestar del trabajador.

Pregunta 8: ¿Cree que el sistema actual impacta en su desempeño laboral?

El propósito es explorar si los procesos administrativos actuales inciden de manera directa en la productividad o motivación del colaborador. Su inclusión permite establecer una relación entre la eficiencia del sistema de RRHH y el rendimiento del personal, lo cual es fundamental para justificar propuestas de mejora.

Figura 11

Percepción del impacto del sistema de gestión de recursos humanos en el desempeño laboral

8. ¿Cree que el sistema actual impacta en su desempeño laboral?



Fuente: Tolentino, 2026.

Los resultados evidencian que una mayoría significativa del personal percibe que el sistema actual de gestión de recursos humanos sí impacta en su desempeño laboral, ya que un 50 % de las personas encuestadas manifestó estar de acuerdo y un 38 % indicó estar totalmente de acuerdo con esta afirmación, mientras que únicamente un 13 % adoptó una postura neutral y no se registraron respuestas en desacuerdo. Esta percepción generalizada sugiere que los procesos administrativos actuales influyen directamente en la productividad y motivación del personal, lo que refuerza la necesidad de implementar mejoras en el sistema de gestión de recursos humanos con el fin de optimizar el desempeño laboral y el funcionamiento organizacional.

Pregunta 9: ¿Cree que la implementación de un sistema digital ayudaría a mejorar la gestión de RRHH en la empresa?

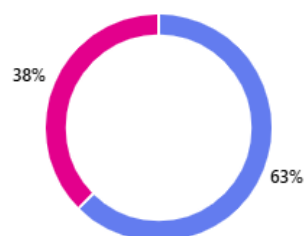
Esta pregunta busca obtener una perspectiva sobre el nivel de aceptación que tendría una transformación digital dentro del área de recursos humanos. Evaluar la disposición del personal ante la posibilidad de incorporar un sistema automatizado permite detectar barreras como la resistencia al cambio o, por el contrario, el interés por herramientas que optimicen la experiencia laboral. Su inclusión se justifica porque toda implementación tecnológica requiere no solo de condiciones técnicas y funcionales, sino también de un entorno humano que la respalde. Conocer esta percepción permite enfocar el diseño del sistema de manera funcional y adaptada al contexto.

Figura 12

Percepción del personal sobre la implementación de un sistema digital para la gestión de recursos humanos

9. ¿Cree que la implementación de un sistema digital ayudaría a mejorar la gestión de RRHH en la empresa?

● Totalmente de acuerdo	5
● De acuerdo	3
● Neutral	0
● En desacuerdo	0
● Totalmente en desacuerdo	0



Fuente: Tolentino, 2026.

Pregunta 10: ¿Estaría dispuesto a utilizar un sistema digital para gestionar su asistencia, nómina y horarios?

Este ítem tiene como objetivo confirmar la disposición concreta del colaborador a incorporar herramientas digitales como parte de su rutina laboral. A diferencia de la pregunta anterior, que evalúa una percepción general, esta busca medir la actitud personal frente al uso cotidiano de una plataforma tecnológica. Además, permite anticipar si es necesario prever estrategias de capacitación o adaptación gradual. En resumen, esta pregunta cierra el bloque de análisis con una valoración crucial sobre la viabilidad real de implementar una solución digital en el entorno laboral del restaurante.

Figura 13

Disposición del personal para utilizar un sistema digital en la gestión de asistencia, nómina y horarios

10. ¿Estaría dispuesto a utilizar un sistema digital para gestionar su asistencia, nómina y horarios?



Fuente: Tolentino, 2026.

Los resultados evidencian una alta disposición del personal para utilizar un sistema digital en la gestión de asistencia, nómina y horarios, ya que un 50 % de las personas encuestadas manifestó estar totalmente de acuerdo y un 38 % indicó estar de acuerdo con su uso, mientras que solo un 13 % adoptó una postura neutral y no se registraron respuestas en desacuerdo. Esta tendencia confirma una actitud favorable hacia la adopción de herramientas digitales en la rutina laboral, lo que sugiere que la implementación de un sistema automatizado no solo es percibida como necesaria, sino también viable desde el punto de vista de aceptación del personal, reduciendo posibles barreras de resistencia al cambio y reforzando la factibilidad del prototipo propuesto.

Observación

La guía de observación fue elaborada como un instrumento complementario para identificar, de manera directa, aspectos operativos y administrativos relevantes dentro del restaurante. Su objetivo es evaluar el cumplimiento de ciertos procesos clave, así como detectar fallas, rutinas ineficientes u oportunidades de mejora que puedan no ser percibidas a través de la opinión del personal. Este instrumento se centra en la observación de evidencias físicas, registros y prácticas cotidianas que permiten validar o contrastar las necesidades detectadas previamente. A continuación, se detallan los aspectos incluidos en la guía, junto con la justificación de su selección dentro del contexto del estudio.

Aspecto por observar 1: Manejo de documentación y archivos

Este aspecto se incluyó para observar el nivel de organización y resguardo de la información relacionada con los empleados, como contratos, planillas, permisos y otros documentos formales. Una gestión documental deficiente puede generar pérdidas de información, incumplimientos legales o desorden administrativo, por lo que su revisión es clave para valorar la necesidad de un sistema digital que centralice y respalde esta información.

Como resultado, se observó que la información del personal se gestiona principalmente de manera manual y dispersa, sin un sistema centralizado que facilite su organización, resguardo y consulta. Lo anterior refuerza la necesidad de implementar un sistema digital que permita almacenar, organizar y respaldar la documentación del personal de forma segura y eficiente.

Aspecto por observar 2: Existencia de criterios claros para la evaluación del desempeño

El objetivo de este punto es verificar si el restaurante cuenta con lineamientos establecidos para valorar el rendimiento de sus colaboradores. La falta de criterios definidos puede provocar decisiones subjetivas, inconformidades y desmotivación. Este aspecto es importante para implementar sistemas de retroalimentación justos y basados en evidencia, lo cual fortalece la transparencia y el desarrollo del personal.

La observación permitió identificar que no se cuenta con lineamientos formales ni instrumentos definidos para valorar de manera sistemática el rendimiento de los colaboradores. Las evaluaciones, cuando se realizan, dependen principalmente de apreciaciones informales por parte del encargado, sin criterios documentados ni indicadores objetivos que respalden dichas valoraciones.

Aspecto por observar 3: Observar el registro de asistencia de empleados

Se pretende identificar el mecanismo utilizado para registrar las entradas y salidas del personal, así como su efectividad. La observación directa permite determinar si este control es manual, automatizado o informal, y si existe riesgo de errores o manipulaciones. Crucial para sustentar la necesidad de un sistema confiable de control horario.

En relación con el registro de asistencia de los empleados, la observación permitió identificar que no existe un mecanismo formal para el control de entradas y salidas del personal. En la práctica, la asistencia se gestiona de manera informal, ya que los colaboradores únicamente avisan su llegada o salida mediante mensajes, y en algunos casos ni siquiera se realiza dicha notificación. No se cuenta con registros físicos ni digitales que respalden de forma sistemática la información sobre los horarios laborados.

Aspecto por observar 4: Observar cómo se calculan y asignan las vacaciones

Este ítem busca conocer el proceso aplicado para el cálculo de vacaciones. La observación permite determinar si los periodos vacacionales se otorgan de manera equitativa, si se llevan registros adecuados del tiempo laborado y si se respetan los plazos correspondientes. También se analiza si existen solicitudes por escrito, autorizaciones formales y un control que permita evitar confusiones o duplicidades. Un manejo inadecuado de las vacaciones puede provocar inconformidades, desorganización operativa y posibles sanciones legales. Por tanto, este aspecto justifica la necesidad de una solución automatizada que garantice transparencia, trazabilidad y cumplimiento normativo en la gestión de este derecho laboral.

Se identifica que, si bien la empresa cuenta con una política interna que establece el otorgamiento de un día de vacaciones por mes laborado, no existe un registro formal ni un sistema

que permita llevar el control adecuado de dicho beneficio. La gestión de las vacaciones se realiza de manera informal, principalmente mediante acuerdos verbales, sin evidenciarse solicitudes por escrito ni autorizaciones documentadas. La ausencia de un registro sistematizado ha dado lugar a confusiones respecto al saldo de vacaciones disponible para cada colaborador, así como a la dificultad para determinar cuántos días han sido acumulados o disfrutados.

Aspecto por observar 5: Tiempo de respuesta y aprobación en solicitudes de permisos

Este aspecto se enfoca en evaluar la agilidad y formalidad con que se atienden las solicitudes de permisos por parte del personal. A través de la observación, se busca identificar si existe un canal claro para presentar las solicitudes y si hay criterios definidos para aprobar o denegar los permisos. Una administración lenta o desorganizada en este sentido puede generar molestias, afectar la planificación de turnos y reducir la percepción de justicia. Además, una respuesta tardía puede impactar en la productividad del equipo y evidenciar una necesidad urgente de digitalizar o estandarizar este proceso. Por ello, este ítem es clave para detectar áreas que podrían beneficiarse significativamente de una herramienta digital que permita realizar solicitudes y aprobaciones en tiempo real.

En relación con el tiempo de respuesta y aprobación en las solicitudes de permisos, la observación permitió identificar que no existe un canal formal ni un procedimiento estandarizado para la gestión de este tipo de solicitudes. Los permisos suelen solicitarse de manera verbal o mediante mensajes informales, sin un registro que permita dar seguimiento al estado de cada solicitud ni evidenciar los criterios utilizados para su aprobación o denegación.

Aspecto por observar 6: Eficiencia en el registro y seguimiento de incapacidades y permisos

Este ítem tiene como objetivo analizar si se lleva un control de las incapacidades médicas y permisos otorgados a los empleados, considerando el respaldo de la documentación. La observación permite determinar si estos casos se reportan de forma oficial y si se reflejan adecuadamente en los sistemas internos, incluyendo planillas y reportes de asistencia. Una gestión ineficiente en este ámbito puede provocar errores en el cálculo de salarios y desajustes en la

asignación de tareas. Por tanto, se justifica la necesidad de herramientas que integren este tipo de información, para mejorar la precisión y la trazabilidad del historial de ausencias justificadas.

La observación permitió identificar que no existe un control sistematizado que integre de forma clara las incapacidades médicas y los permisos otorgados al personal. Si bien en algunos casos se presenta documentación de respaldo, esta no se gestiona de manera centralizada ni se refleja de forma consistente en los registros de asistencia o en el cálculo de planillas. Como se menciona, la falta de un seguimiento formal puede dar lugar a errores en el cálculo salarial, inconsistencias en los reportes de asistencia y desajustes en la asignación de tareas dentro del equipo de trabajo

Aspecto por observar 7: Detección oportuna de ausencias, retardos y faltas injustificadas

Este aspecto se orienta a observar si el sistema actual de control permite identificar, en tiempo real o con frecuencia adecuada, a los empleados que presentan inasistencias, llegadas tardías o ausencias sin justificación válida. La detección tardía de estas situaciones puede repercutir negativamente en la operación diaria del restaurante. Si se identifican deficiencias, se refuerza la necesidad de implementar un sistema automatizado que facilite el monitoreo y la respuesta rápida ante este tipo de eventos.

Se observa que el sistema actual no cuenta con mecanismos que permitan reconocer estas situaciones de manera inmediata ni con una frecuencia adecuada. Debido a la ausencia de un control formal de asistencia, la identificación de inasistencias o llegadas tardías suele realizarse de forma tardía o depender de la comunicación informal entre el personal y el encargado.

CAPÍTULO V: PROPUESTA

En este capítulo se presenta una propuesta basada en los hallazgos obtenidos a lo largo de la investigación. El objetivo principal es ofrecer una solución concreta y viable que atienda las debilidades identificadas en la gestión de recursos humanos dentro del restaurante Tumi's House. La propuesta busca optimizar los procesos administrativo y mejorar el control interno haciendo uso de herramientas tecnológicas adaptadas a las necesidades de la empresa.

La formulación de esta propuesta responde directamente a las problemáticas observadas en los capítulos anteriores, las cuales incluyen la falta de automatización, deficiencias en el registro de asistencias, control de vacaciones, cálculo de planillas, entre otros aspectos clave del área de recursos humanos. La estructura del capítulo abarca los distintos componentes que conforman el sistema, comenzando por el análisis y diseño de este, seguido por la arquitectura general, el modelado de datos, la descripción de procesos, los diagramas UML y finalmente los aspectos relacionados con la programación y validaciones.

Análisis

En esta sección se presenta una descripción técnica del sistema, incluyendo los distintos módulos que lo conforman, su funcionamiento interno y la estructura general que lo sustenta. Se abordan los distintos módulos que integran el sistema, junto con los requerimientos técnicos relacionados con el software, el equipo físico y la infraestructura de red. También se presentan los casos de uso como una guía práctica de la interacción entre los actores del sistema y sus funcionalidades. Esta información es esencial para identificar el alcance real de la solución desarrollada, así como las condiciones necesarias para su correcta implementación y uso.

Análisis detallado del Software

En esta sección se detalla el funcionamiento interno del sistema propuesto, desglosando cada uno de los módulos que lo componen y explicando su propósito dentro del proceso de gestión de recursos humanos. Se describen las funcionalidades específicas de cada módulo y su interacción con otros componentes del sistema.

Módulo controlar asistencia y tiempos de marcas.

Este módulo se encarga de registrar las marcas de entrada y salida de los colaboradores, generando un control en tiempo real de la asistencia. Permite obtener datos precisos y automáticos que reduzcan el margen de error en el control de ausencias, retrasos y cumplimiento de la jornada laboral.

Módulo gestionar horarios.

Este módulo permite automatizar la planificación y asignación de horarios para cada empleado, considerando disponibilidad y permisos aprobados. Ofrece también la posibilidad de editar turnos de manera manual, manteniendo la flexibilidad operativa. El módulo garantiza que no se asignen turnos a colaboradores con ausencias programadas, y permitirá exportar los horarios asignados en formato digital para su distribución.

Módulo gestionar evaluaciones de empleados.

El módulo permite evaluar el rendimiento de los colaboradores mediante herramientas de retroalimentación como encuestas. En el caso de los saloneros, las evaluaciones se almacenarán en un historial individual por empleado, permitiendo el análisis continuo de su desempeño, la identificación de debilidades y el reconocimiento de logros. Para los demás servicios, la retroalimentación se registrará de forma grupal o por área, con fines de análisis del desempeño del equipo.

Módulo gestionar permisos.

Este módulo permite a los colaboradores enviar solicitudes de permisos de forma digital, indicando tipo de permiso, motivo y fechas requeridas. El administrador podrá aprobar o rechazar las solicitudes desde el panel de control, y el sistema notificará automáticamente la decisión al empleado. Todas las solicitudes quedarán registradas en un historial accesible para auditorías y revisiones futuras, facilitando una gestión organizada de las ausencias.

Módulo calcular liquidación.

Este módulo automatiza el cálculo de liquidaciones laborales al momento de finalizar una relación contractual. Se tomarán en cuenta componentes como salario base, días laborados en el mes, vacaciones no disfrutadas, aguinaldo proporcional, preaviso y cesantía, según lo estipulado en el Código de Trabajo. Además, el sistema generará un desglose detallado de cada concepto incluido, permitiendo al empleado y a la administración validar la exactitud de los cálculos.

Módulo calcular vacaciones.

Este módulo permite llevar un control actualizado de los días de vacaciones generados y utilizados por cada empleado. El sistema calculará el saldo acumulado mes a mes, tomando como base la antigüedad del colaborador. Permitirá la visualización del historial de descansos y facilitará la planificación de vacaciones sin afectar la operación.

Módulo gestionar incapacidades.

Este módulo registra los periodos de incapacidad médica presentados por los colaboradores, incluyendo fechas, diagnóstico, tipo de incapacidad y duración. Permite calcular los montos que la empresa debe cubrir, según lo establecido por la legislación laboral y la política interna. Además, brinda acceso al historial completo de incapacidades por colaborador, permitiendo la consulta rápida de registros en caso de auditorías o revisiones internas.

Módulo gestionar planilla y nóminas.

Este módulo gestiona la elaboración completa de la planilla de sueldos, considerando datos de asistencia, horas extra, deducciones, impuestos, bonificaciones, y más. El sistema permite validar los montos antes de generar la nómina final, emitir comprobantes de pago, y exportar la información. También incluye herramientas para generar informes periódicos.

Módulo mantenimientos.

Este módulo permitirá gestionar de manera integral la información contenida en las diferentes tablas del sistema. Estará habilitado para realizar operaciones como inserción, modificación, eliminación y actualización de datos. Su función será asegurar que la base de datos se mantenga actualizada, limpia y coherente con la realidad operativa de la empresa. Estará restringido a usuarios con privilegios administrativos para garantizar la integridad de la información.

Módulo consultas.

Este módulo estará orientado a facilitar el acceso a información almacenada en el sistema por medio de consultas directas a las tablas. Permite a los usuarios visualizar datos relevantes sin posibilidad de alterarlos, lo que asegura un entorno de solo lectura. Las consultas podrán filtrarse por fechas, empleados, tipos de registros u otros criterios definidos, lo que facilitará la revisión de información para auditorías, control interno o toma de decisiones.

Módulo reportes.

El módulo de reportes generará información consolidada en formatos estructurados a partir de los datos registrados en el sistema. Permitirá producir documentos exportables que incluyan resúmenes de asistencia, nómina, vacaciones, horas extra, entre otros. Los reportes podrán visualizarse en pantalla o imprimirse según la necesidad del usuario.

Módulo seguridad.

Este módulo gestionará el acceso al sistema mediante la autenticación de usuarios a través de credenciales únicas. Permitirá definir distintos niveles de acceso según perfiles de usuario (administrador, supervisor, empleado, etc.), asegurando que cada usuario tenga permisos únicamente sobre las funciones correspondientes a su rol. También incluirá mecanismos de

encriptación de contraseñas y control de sesiones, reforzando la protección de la información almacenada.

Análisis Detallado del Hardware

Para asegurar el adecuado funcionamiento del sistema que se propone, será necesario poseer una infraestructura de hardware que se encargue de: realizar el tratamiento de los datos e interactuar con los usuarios de forma continuada. En el presente análisis se hacen evidentes aquellos componentes físicos, imprescindibles para garantizar la correcta puesta en marcha del sistema.

Desarrollo del Prototipo.

El desarrollo del prototipo se llevó a cabo utilizando herramientas accesibles y acordes con las necesidades del proyecto. Para ello, se utilizó un equipo de cómputo personal como estación principal de trabajo, el cual cumplía con las especificaciones técnicas necesarias para realizar tareas de programación, modelado, pruebas y documentación del sistema. El equipo que se utiliza cuenta con un procesador Intel Core i7 de 11ª generación, 8 GB de memoria RAM y unidad de almacenamiento tipo SSD de 500 GB, sistema operativo Windows y gestores de bases de datos y otras herramientas complementarias.

Implementación del prototipo.

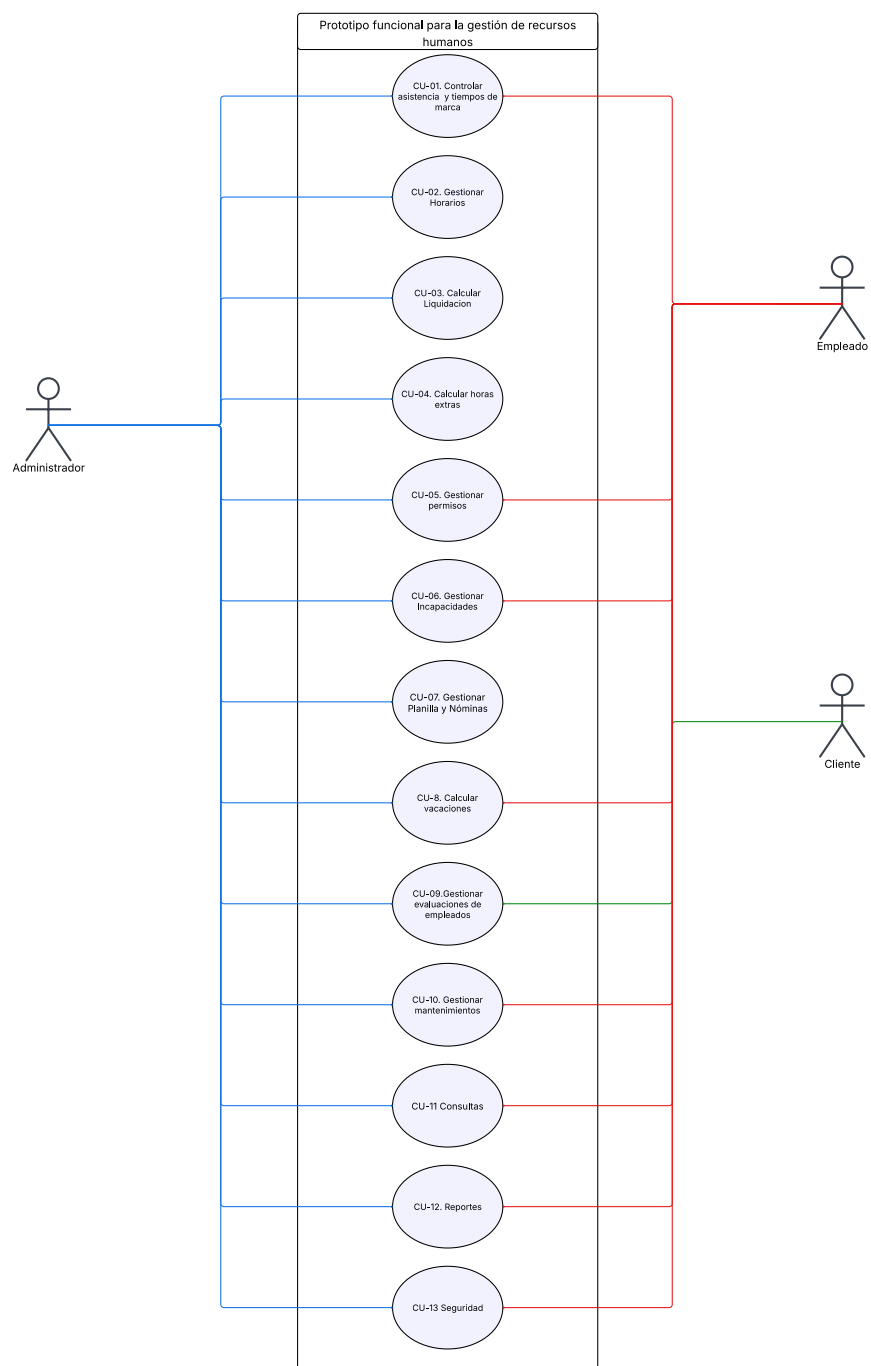
Para la correcta operación del sistema se requiere una infraestructura de hardware básica pero funcional, que permita ejecutar el sistema de forma continua y segura. En este caso, el restaurante Tumi's House ya cuenta con el equipo necesario para implementar el sistema sin requerir inversiones adicionales significativas en infraestructura tecnológica, lo que facilita su puesta en marcha inmediata. Las características del equipo de cómputo que fungirá como servidor para alojar los servicios son los siguientes:

- Equipo principal: Una computadora de escritorio con procesador Intel Core i5, 8 GB de memoria RAM, unidad SSD de 256 GB, y sistema operativo Windows 10. Este equipo puede funcionar como servidor local, alojando el backend del sistema, la base de datos MySQL y sirviendo la interfaz web para los usuarios internos.
- Red local y conectividad: El restaurante cuenta con una red local estable, respaldada por un router y puntos de acceso Wi-Fi, que permite conectar los dispositivos del sistema entre sí sin necesidad de cableado adicional.
- Dispositivo móvil para evaluaciones: El restaurante dispone de una tableta Samsung con sistema operativo Android y conectividad Wi-Fi, la cual puede utilizarse como dispositivo de apoyo para realizar evaluaciones de desempeño de los empleados mediante el sistema. Este dispositivo permite que los encargados o supervisores puedan acceder al sistema desde diferentes áreas del restaurante y registrar las evaluaciones del personal de manera rápida y práctica.

Casos de Uso

Figura 14

Diagrama UML de Casos de Uso



Fuente: Tolentino, 2026.

La Figura 14 muestra el diagrama de casos de uso que representa el prototipo funcional del sistema, mostrando las principales funcionalidades y su interacción con los distintos actores. El administrador es responsable de procesos como la gestión de horarios, permisos, incapacidades, nómina y vacaciones, mientras que el empleado interactúa con módulos como el registro de marcas, solicitudes de permisos e incapacidades, y consulta de información personal. Por su parte, el cliente accede a funciones específicas relacionadas con la evaluación del personal.

Tabla 4

Caso de uso para controlar asistencia y tiempos de marcas

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-01	Nombre del caso de uso: Controlar asistencia y tiempos de marcas
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite a los colaboradores registrar su entrada o salida mediante un PIN personal, y gestiona en tiempo real el control de asistencia, ausencias y retrasos.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Empleados
Precondiciones:	El colaborador debe tener un PIN asignado en el sistema.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. En la pantalla principal del sistema, el colaborador selecciona la opción “Registrar marca”. 2. El sistema solicita el ingreso del PIN personal. 3. El colaborador digita su PIN. (FA01-Validar PIN, FA02-Validar Turno Asignado) 4. El sistema registra automáticamente la marca según corresponda. 5. Se calcula automáticamente el tiempo registrado en relación con la jornada establecida para ese día. 6. El sistema determina si hay marcajes incompletos, retardos o ausencias. 7. Se muestra un mensaje de confirmación visual. 8. El sistema retorna a la pantalla principal para permitir que otro colaborador marque. 	

Subflujos	
N/A	N/A
Flujos alternos	
FA01- Validar PIN	<ol style="list-style-type: none"> 1. Si el PIN ingresado es incorrecto o no existe, el sistema muestra un mensaje de error y permite reintentar.
FA02-Validar Turno Asignado	<ol style="list-style-type: none"> 1. Si el colaborador no tiene un turno asignado para la fecha correspondiente, el sistema impide el registro de la marca y muestra un mensaje indicando que la situación debe ser gestionada por la administración del sistema.
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe registrar fecha y hora de la marca. • Debe evaluar automáticamente si el colaborador presenta ausencias, retrasos o tiempos incompletos según la jornada asignada. • La interfaz debe ser rápida, accesible y sin necesidad de login completo. • El sistema debe generar reportes de asistencia acumulada por día, semana y mes, accesibles desde otros módulos. 	
Postcondiciones	
<ul style="list-style-type: none"> • La marca queda registrada en la base de datos y vinculada al historial del colaborador. • El sistema actualiza los indicadores de asistencia y los módulos de planilla y reportes. 	

Fuente: Tolentino, 2026.

Tabla 5

Caso de uso para gestionar horarios

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-02	Nombre del caso de uso: Gestionar Horarios
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite al administrador asignar, modificar o consultar los horarios de trabajo de los colaboradores, según las necesidades operativas del restaurante.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Administrador
Precondiciones:	El usuario debe haber iniciado sesión con un rol de administrador autorizado para gestionar horarios.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El administrador inicia sesión en el sistema. 2. Accede al módulo de “Gestionar horarios” desde el menú lateral. 3. El sistema muestra la lista de colaboradores activos. 4. El sistema despliega un calendario semanal editable. 5. El administrador selecciona la casilla correspondiente al día y colaborador en el calendario semanal para asignar un turno. (SF01-Asignar Horario Semanal) 6. El administrador selecciona el turno de trabajo (apertura o cierre) en el calendario y define, cuando corresponde, los tiempos de almuerzo y café asociados a la jornada. (FA01- Validar Asignación De Turno) 7. El sistema permite guardar los cambios realizados. (FA-02 Validar Conflictos De Horario) 8. El sistema confirma que los horarios han sido actualizados exitosamente. 9. El sistema da la opción de exportar horario. (SF02-Exportar) 	
Subflujos	
SF01-Asignar Horario Semanal	<ol style="list-style-type: none"> 1. El administrador escoge la opción “Asignar semana”. 2. El administrador selecciona un colaborador desde la lista

	<ol style="list-style-type: none"> 3. El sistema genera automáticamente el horario semanal del colaborador según un patrón escogido (turnos de apertura o cierre). 4. El administrador puede revisar y, si es necesario, ajustar manualmente los turnos.
SF02-Exportar	<ol style="list-style-type: none"> 1. El administrador puede exportar el horario semanal en formato PDF o Excel para uso interno.
Flujos alternos	
FA01- Validar Asignación De Turno	<ol style="list-style-type: none"> 1. Si la información del turno es incompleta o inválida (horas faltantes, datos incorrectos etc.), el sistema muestra un mensaje de error y no permite guardar los cambios.
FA02-Validar Conflictos De Horario	<ol style="list-style-type: none"> 1. Si el colaborador se encuentra con vacaciones, permiso o incapacidad registrada en el período o día seleccionado, el sistema impide la asignación del horario y notifica la condición.
Requerimientos especiales	
<ul style="list-style-type: none"> • La interfaz del calendario debe ser visual, clara y editable. • Debe integrarse con el módulo de marcas para calcular correctamente los tiempos laborados y generar reportes precisos. 	
Postcondiciones	
<ul style="list-style-type: none"> • Los horarios asignados quedan almacenados y vinculados al historial de cada colaborador. • El sistema utiliza estos datos para el control de asistencia, detección de ausencias o retardos, y cálculo de planilla. 	

Fuente: Tolentino, 2026.

Tabla 6*Caso de uso para calcular liquidación*

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-03	Nombre del caso de uso: Calcular liquidación
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite al administrador calcular automáticamente el monto de liquidación de un colaborador que finaliza su relación laboral. El sistema considera antigüedad, salario promedio, vacaciones no disfrutadas, aguinaldo proporcional y otros rubros legales aplicables.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador
Precondiciones:	El administrador debe haber iniciado sesión.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El administrador inicia sesión en el sistema. 2. Accede al módulo “Liquidaciones” desde el menú lateral. 3. Selecciona al colaborador que ha finalizado su contrato. 4. El sistema solicita al administrador indicar el tipo de finalización del contrato (SF01- Seleccionar tipo de finalización laboral) 5. El sistema recopila automáticamente los siguientes datos: <ul style="list-style-type: none"> • Fecha de ingreso y salida • Salario promedio de los últimos meses • Días de vacaciones pendientes • Aguinaldo proporcional del año en curso • Información relacionada con preaviso y cesantía (según corresponda) 6. Calcula el monto total de liquidación con desglose por rubro. 7. Muestra una vista previa del cálculo al administrador. 8. El administrador revisa y confirma el resultado. 9. El sistema genera un documento en PDF con el resumen de liquidación. 10. La liquidación queda registrada en el historial del colaborador. 	

Subflujos	
SF01- Seleccionar tipo de finalización laboral	<ol style="list-style-type: none"> 1. El administrador selecciona el tipo de finalización del contrato, seleccionando una de las opciones disponibles: <ul style="list-style-type: none"> • Con responsabilidad patronal • Sin responsabilidad patronal • Renuncia • Vencimiento de contrato • Despido con causa justificada 2. El sistema ajusta automáticamente los rubros y fórmulas de cálculo aplicables a la liquidación.
Flujos alternos	
N/A	N/A
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe calcular automáticamente montos basados en la legislación laboral costarricense (vacaciones, aguinaldo, preaviso, cesantía). • La interfaz debe mostrar un desglose claro. • El PDF generado debe incluir: nombre del colaborador, fechas, fórmula utilizada y totales. • La liquidación debe quedar registrada como parte del expediente del colaborador. 	
Postcondiciones	
<ul style="list-style-type: none"> • La liquidación queda almacenada en el sistema, vinculada al historial del colaborador. • El documento puede ser descargado. • El estado del colaborador queda como “liquidado” en el sistema. 	

Fuente: Tolentino, 2026.

Tabla 7

Caso de uso para calcular horas extras

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-04	Nombre del caso de uso: Calcular horas extras
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite al sistema identificar y calcular automáticamente las horas extra laboradas por los colaboradores, comparando las marcas de entrada y salida con los horarios asignados. También permite que el empleado solicite realizar horas extras.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador
Precondiciones:	El sistema debe tener configurados los horarios asignados a cada colaborador y acceso a los registros de marcas.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El sistema identifica las marcas registradas por el colaborador (entrada y salida del día). 2. Consulta el horario asignado correspondiente para ese día. 3. Compara la hora de salida registrada con la hora de salida programada. 4. Si existe tiempo trabajado posterior al horario ordinario, el sistema calcula la diferencia en minutos. 5. Valida que el exceso supere el umbral mínimo para considerarse como hora extra. 6. Calcula el monto correspondiente con base en la legislación laboral vigente. 7. Las horas extra quedan registradas en estado "Pendiente de aprobación". 8. El administrador accede al módulo "Horas extras". 9. Revisa el detalle por colaborador: fecha, horas, tipo, monto estimado. 10. Selecciona "Aprobar" o "Rechazar" para cada registro. 11. El empleado también puede solicitar realizar horas extras. (SF01 – Solicitar Horas Extra) 12. El sistema actualiza el estado de cada hora extra según la decisión. 	

13. Solo las horas aprobadas se integran al módulo de planilla.	
Subflujos	
SF01 – Solicitar Horas Extra	<ol style="list-style-type: none"> 1. El colaborador accede al módulo “Solicitar Horas extras” desde su panel. 2. El sistema muestra el formulario de solicitud de horas extra. 3. El colaborador completa la solicitud indicando: <ul style="list-style-type: none"> • Fecha en la que realizará las horas extra • Motivo o justificación • Cantidad horas <p style="text-align: center;">(FA01 – Validar Solitud De Hora Extra)</p> <ol style="list-style-type: none"> 4. El colaborador envía la solicitud. 5. El sistema registra la solicitud con estado “Pendiente de aprobación” y notifica al administrador para su revisión.
Flujos alternos	
FA01 – Validar Solitud De Hora Extra	<ol style="list-style-type: none"> 1. Si la información de la solicitud es incompleta o inválida (campos faltantes, datos incorrectos etc.), el sistema muestra un mensaje de error y no permite realizar la solicitud.
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe identificar excesos de jornada automáticamente, según horario y marca registrada. • Debe clasificar las horas extra por tipo (diurna, nocturna) y aplicar las tarifas legales. • Solo las horas aprobadas por el administrador se consideran para el pago. • El módulo debe ofrecer filtros por colaborador, fecha y estado (pendiente, aprobado, rechazado). 	
Postcondiciones	

- Las horas extra aprobadas quedan registradas en la base de datos y se integran a la planilla correspondiente.
- Las rechazadas quedan registradas pero excluidas de cálculo salarial.
- El sistema actualiza el historial del colaborador y permite generar reportes por periodo.

Fuente: Elaboración Propia, 2026.

Tabla 8

Caso de uso para gestionar permisos

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-05	Nombre del caso de uso: Gestionar permisos
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite a los colaboradores registrar solicitudes de permisos laborales y al jefe inmediato revisarlas y aprobarlas o rechazarlas según criterios internos del restaurante.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Administrador y empleado
Precondiciones:	El colaborador o administrador debe haber iniciado sesión en el sistema con su usuario y contraseña.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El colaborador inicia sesión en el sistema. 2. Accede al módulo “Permisos” desde el menú lateral. 3. El sistema despliega el historial de solicitudes anteriores y un botón para crear una nueva. 4. El colaborador presiona “Nueva solicitud” y completa el formulario con: <ul style="list-style-type: none"> • Fecha de inicio y fin del permiso • Motivo o justificación 	

<ul style="list-style-type: none"> • Adjuntar un documento justificativo (como constancia médica). • Comentarios adicionales (opcional). <p>(FA01 – Campos Obligatorios Incompletos)</p> <p>5. El sistema valida que las fechas estén en formato correcto y que el rango solicitado sea coherente. (FA02 – Rango De Fechas Incoherente y FA03 – Superposición Con Otras Solicitudes)</p> <p>6. El colaborador envía la solicitud.</p> <p>7. El sistema registra el estado inicial como “Pendiente” y notifica al jefe inmediato.</p> <p>8. El jefe inmediato ingresa al sistema, revisa las solicitudes pendientes y aprueba o rechaza cada una. El jefe inmediato puede dejar un comentario junto a su decisión de aprobación o rechazo.</p> <p>9. El sistema actualiza el estado a “Aprobado” o “Rechazado” según corresponda.</p> <p>10. El colaborador es notificado sobre la decisión.</p>	
Subflujos	
N/A	N/A
Flujos alternos	
FA01 – Campos Obligatorios Incompletos	<ol style="list-style-type: none"> 1. El colaborador intenta continuar o enviar la solicitud sin completar los campos obligatorios. 2. El sistema muestra un mensaje indicando los campos obligatorios pendientes. 3. El sistema no permite registrar la solicitud hasta completar la información requerida.
FA02 – Rango De Fechas Incoherente	<ol style="list-style-type: none"> 1. El colaborador ingresa una fecha fin anterior a la fecha inicio (o el mismo día cuando no aplica). 2. El sistema muestra un mensaje indicando que el rango de fechas no es válido. 3. El sistema no permite continuar hasta corregir el rango.
FA03 – Superposición Con Otras Solicitudes	<ol style="list-style-type: none"> 1. El colaborador solicita un permiso en fechas que se superponen con otro permiso, vacaciones o incapacidad ya registrada.

	<ol style="list-style-type: none"> 2. El sistema muestra un mensaje indicando el conflicto y las fechas afectadas. 3. El sistema no permite enviar la solicitud hasta que se modifique el rango.
Requerimientos especiales	
<ul style="list-style-type: none"> • Las fechas ingresadas deben validarse para evitar superposición con otras solicitudes. • Se debe garantizar que solo el administrador correspondientes puedan aprobar solicitudes. • El historial debe incluir fecha de solicitud, estado, decisión del jefe y observaciones. 	
Postcondiciones	
<ul style="list-style-type: none"> • La solicitud queda registrada en la base de datos con su estado correspondiente. • El colaborador puede consultar el resultado desde su panel personal. • El sistema actualiza automáticamente los registros de ausencias del colaborador, para su consideración en asistencia y planilla. 	

Fuente: Tolentino, 2026.

Tabla 9

Caso de uso para gestionar incapacidades

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-06	Nombre del caso de uso: Gestionar Incapacidades
Fecha elaboración:	13/05/2025
Descripción caso de uso:	Permite registrar y administrar las incapacidades laborales de los colaboradores y mantener el historial accesible para fines de control y cumplimiento legal.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Administrador

Precondiciones:	El administrador debe haber iniciado sesión en el sistema con su usuario y contraseña.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El administrador inicia sesión en el sistema. 2. Accede al módulo “Gestionar incapacidades” desde el menú lateral. 3. El sistema muestra el historial de incapacidades registradas. 4. El sistema permite filtrar el historial por colaborador, fechas o tipo de incapacidad 5. El administrador puede registrar una nueva incapacidad (SF01-Ingresar Incapacidad) 6. El administrador puede anular una incapacidad (SF02-Anular Incapacidad) 7. Si el administrador desea modificar una incapacidad ir a SF03-Modificar Incapacidad 	
Subflujos	
SF01 - Ingresar Incapacidad	<ol style="list-style-type: none"> 1. El administrador presiona el botón “Registrar nueva incapacidad”. 2. Completa el formulario con los siguientes datos: <ul style="list-style-type: none"> • Colaborador afectado • Fecha de inicio y fin de la incapacidad • Motivo o tipo de incapacidad • Entidad emisora (ej. CCSS, INS, etc.) • Documento de respaldo (adjunto) (FA01 – Validar Registro De Incapacidad) 3. El sistema valida que las fechas sean coherentes y que no se superpongan con otras ausencias. (FA01 – Superposición de Incapacidades) 4. El administrador confirma el registro. 5. El sistema guarda la información en la base de datos y actualiza el estado del colaborador como “Incapacidad activa” durante el rango indicado. 6. El sistema muestra un mensaje de confirmación.
SF02 – Anular incapacidad	<ol style="list-style-type: none"> 1. El Administrador selecciona la incapacidad a anular.

	2. El sistema le cambia el estado a la incapacidad y le da la opción de Guardar
SF03-Modificar Incapacidad	<ol style="list-style-type: none"> 1. El cliente selecciona la incapacidad a modificar. 2. El usuario ajusta los valores permitidos y la da Guardar. (FA01 – Validar Registro De Incapacidad)
Flujos alternos	
FA01 – Validar Registro De Incapacidad	<ol style="list-style-type: none"> 1. Si la información de la incapacidad es incompleta o inválida (campos faltantes, datos incorrectos etc.), el sistema muestra un mensaje de error y no permite guardar la incapacidad o los cambios.
FA02 – Superposición De Incapacidades	<ol style="list-style-type: none"> 1. El administrador registra una incapacidad que se cruza con otra ausencia existente. 2. El sistema muestra una advertencia indicando el conflicto. 3. El sistema bloquea el registro hasta ajustar el rango.
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe validar que las fechas ingresadas no se crucen con otras ausencias existentes. • Se debe permitir subir y almacenar documentos justificativos con seguridad y respaldo. • El historial debe incluir: colaborador, fechas, motivo, estado y observaciones. • El sistema debe integrarse con los módulos de asistencia y planilla para suspender marcas y ajustar el cálculo salarial. 	
Postcondiciones	
<ul style="list-style-type: none"> • La incapacidad queda registrada en la base de datos y vinculada al historial del colaborador. • El sistema actualiza los módulos de asistencia y planilla para reflejar adecuadamente la ausencia. • El administrador puede consultar y exportar el historial de incapacidades para gestión interna o fiscalización externa. 	

Fuente: Tolentino, 2025.

Tabla 10

Caso de uso para gestionar planilla y nóminas

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-07	Nombre del caso de uso: Gestionar Planilla y Nóminas
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite al administrador generar, revisar y registrar la planilla salarial de los colaboradores, integrando automáticamente salarios base, horas extras aprobadas, deducciones legales, permisos, vacaciones e incapacidades, conforme a la legislación laboral vigente.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador
Precondiciones:	El administrador debe haber iniciado sesión.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El administrador inicia sesión en el sistema. 2. Accede al módulo “Planilla y Nóminas” desde el menú lateral. 3. El sistema muestra el historial de planillas generadas y la opción “Generar nueva planilla”. 4. El administrador presiona “Generar nueva planilla”. 5. El sistema muestra el formulario para seleccionar el período de pago. (FA01– Período Inválido y FA02 – Planilla Ya Registrada Para El Período). 6. El administrador selecciona el período y presiona “Calcular planilla”. 7. El sistema recopila automáticamente la información necesaria por colaborador (salario base, asistencia, horas extra aprobadas, vacaciones, permisos, incapacidades, deducciones, etc.). 8. El sistema calcula los ingresos, aplica deducciones y determina el salario neto por colaborador. 9. El sistema muestra una vista previa de la planilla con el resumen por colaborador y las acciones disponibles. (SF01– Detalle De Cálculo Por Colaborador) 	

<p>10. El administrador revisa la información y presiona “Confirmar planilla”.</p> <p>11. El sistema solicita confirmación final y, al aprobarse, registra la planilla en la base de datos.</p> <p>12. El sistema muestra un mensaje de confirmación de planilla generada correctamente.</p> <p>13. El sistema actualiza el historial salarial del colaborador y habilita la exportación de reportes. (SF02– Exportar Planilla)</p>	
Subflujos	
SF01– Detalle de Cálculo por Colaborador	<ol style="list-style-type: none"> 1. En la vista previa de la planilla, el administrador selecciona “Ver detalle” en un colaborador. 2. El sistema muestra el desglose del cálculo: <ul style="list-style-type: none"> • Ingresos (salario ordinario, extras aprobadas, vacaciones/permisos con goce) • Deducciones (CCSS, renta, permisos sin goce, deducciones personalizadas) • Resumen final (total ingresos, total deducciones, neto) 3. El administrador cierra el detalle y retorna a la vista previa.
SF02– Exportar Planilla	<ol style="list-style-type: none"> 1. El administrador selecciona la opción “Exportar” en la planilla confirmada. 2. El sistema genera el reporte en formato PDF o Excel. 3. El administrador descarga el archivo generado.
Flujos alternos	
FA01– Período Inválido	<ol style="list-style-type: none"> 1. El administrador selecciona un período con fechas incoherentes. 2. El sistema muestra un mensaje de error indicando el problema. 3. El sistema no permite continuar hasta corregir el período.

FA02 – Planilla ya Registrada para el Período	<ol style="list-style-type: none"> 1. El administrador intenta calcular una planilla para un período que ya fue generado anteriormente. 2. El sistema bloquea la acción y muestra una advertencia. 3. El administrador debe seleccionar un período diferente.
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe calcular automáticamente la planilla salarial conforme a la legislación laboral costarricense, considerando salario ordinario, horas extra aprobadas, vacaciones, permisos, incapacidades y deducciones legales aplicables. • La interfaz debe mostrar un desglose claro y detallado de ingresos, deducciones y salario neto por colaborador, permitiendo su revisión previa a la confirmación. • La planilla confirmada no debe permitir modificaciones posteriores, con el fin de preservar la integridad de la información salarial. 	
Postcondiciones	
<ul style="list-style-type: none"> • La planilla queda registrada y almacenada en el sistema, asociada al período de pago correspondiente. • El historial salarial de cada colaborador se actualiza con la información de la planilla generada. • Las horas extra aprobadas, deducciones aplicadas y montos netos quedan reflejados correctamente en los registros del sistema. 	

Fuente: Tolentino, 2026.

Tabla 11

Caso de uso para gestionar evaluaciones de empleados

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-03	Nombre del caso de uso: Gestionar evaluaciones de empleados
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite a los clientes del restaurante completar formularios de evaluación sobre el desempeño de

	los colaboradores, sin necesidad de autenticación, y al administrador visualizar los resultados para fines de control, retroalimentación, etc.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Cliente, Administrador del sistema
Precondiciones:	El cliente se encuentra en el restaurante y acepta realizar la evaluación.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El salonerero abre el formulario de evaluación en la tableta del restaurante. 2. El salonerero entrega la tableta al cliente para que pueda realizar la evaluación. 3. El sistema muestra el formulario interactivo de evaluación. 4. El cliente responde las preguntas relacionadas con la atención, el servicio y la experiencia recibida. 5. El cliente completa el formulario y lo envía. 6. El sistema valida que todos los campos obligatorios estén completos. (<i>FA01 – Validar Envío de Evaluación</i>). 7. El sistema registra las respuestas y guarda la evaluación en la base de datos. 8. El sistema muestra un mensaje de confirmación agradeciendo la participación del cliente. 9. El salonerero recupera la tableta para continuar con el servicio. 10. El administrador puede ingresar posteriormente al módulo de evaluaciones para consultar los resultados. 	
Subflujos	
N/A	N/A
Flujos alternos	
FA01 – Validar Envío de Evaluación	<ol style="list-style-type: none"> 1. El cliente intenta enviar el formulario sin completar uno o más campos obligatorios. 2. El sistema muestra un mensaje indicando que existen preguntas pendientes por responder y no permite el envío de la evaluación hasta que la información requerida sea completada.

Requerimientos especiales
<ul style="list-style-type: none"> • El sistema debe registrar: fecha, hora, respuestas y comentarios. • El acceso a los resultados de las evaluaciones debe estar restringido al administrador, a través del módulo de evaluaciones.
Postcondiciones
<ul style="list-style-type: none"> • La evaluación queda registrada en la base de datos y vinculada al colaborador correspondiente.

Fuente: Tolentino, 2025.

Tabla 12

Caso de uso para calcular vacaciones

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-09	Nombre del caso de uso: Calcular vacaciones
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite al sistema calcular automáticamente la cantidad de días de vacaciones acumulados por cada colaborador, según la antigüedad laboral, los días efectivamente trabajados. El resultado se refleja en los registros del empleado del empleado.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador, colaborador
Precondiciones:	El colaborador debe estar activo en el sistema. Debe contar con registros de asistencia válidos.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El sistema realiza un proceso programado para evaluar la acumulación de vacaciones. 2. Revisa el historial de asistencia y antigüedad de cada colaborador. 3. Calcula proporcionalmente los días de vacaciones acumulados. 	

<ol style="list-style-type: none"> 4. Actualiza el total disponible en el perfil del colaborador. 5. El colaborador puede solicitar vacaciones seleccionando la opción “Solicitar vacaciones” y completando el formulario. (SF01 – Solicitar vacaciones) 6. El sistema registra esta información en el historial de vacaciones del usuario. 7. El administrador puede consultar la acumulación detallada desde su panel. 8. El colaborador puede visualizar el total acumulado y el historial desde su perfil. 	
Subflujos	
SF01 – Solicitar vacaciones	<ol style="list-style-type: none"> 1. El colaborador selecciona la opción “Solicitar vacaciones” desde el módulo de Vacaciones. 2. El sistema muestra el formulario de solicitud. 3. El colaborador completa la solicitud indicando: <ul style="list-style-type: none"> • Fecha de inicio y fin de las vacaciones • Comentarios adicionales (opcional) 4. El colaborador envía la solicitud. (FA01 – Días Insuficientes o Rango Inválido) 5. El sistema registra la solicitud con estado “Pendiente” y la asocia al historial de vacaciones del colaborador. 6. El sistema notifica al administrador para su revisión.
Flujos alternos	
FA01 – Días Insuficientes o Rango Inválido	<ol style="list-style-type: none"> 1. El colaborador intenta enviar una solicitud de vacaciones. 2. El sistema valida el rango de fechas y la cantidad de días solicitados. 3. Si el rango de fechas es incoherente o el colaborador no cuenta con suficientes días de vacaciones disponibles: <ul style="list-style-type: none"> • El sistema muestra un mensaje indicando la causa del rechazo.

	<ul style="list-style-type: none"> El sistema no permite registrar la solicitud hasta que se corrija la información.
Requerimientos especiales	
<ul style="list-style-type: none"> Debe actualizar el saldo de forma periódica. Debe permitir la visualización tanto para el colaborador como para el administrador. 	
Postcondiciones	
<ul style="list-style-type: none"> El colaborador cuenta con un saldo actualizado de días disponibles. El sistema refleja los días acumulados en la vista del empleado y en los reportes del administrador. 	

Fuente: Tolentino, 2026.

Tabla 13

Caso de uso para mantenimientos

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-10	Nombre del caso de uso: Gestionar mantenimientos
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite a los usuarios realizar acciones de mantenimiento en el sistema, tales como insertar, modificar, eliminar o actualizar información en las diferentes tablas internas, según necesidad administrativa o corrección de datos.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador, empleado
Precondiciones:	El usuario debe haber iniciado sesión con un perfil válido. Solo podrá realizar acciones dentro de los límites de su rol.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> El usuario inicia sesión. Ingresa a un módulo del sistema (ej. permisos, incapacidades, vacaciones). 	

3. El sistema muestra las acciones disponibles según su rol: <ul style="list-style-type: none"> • Colaborador: puede insertar solicitudes o marcar asistencia. • Administrador: puede aprobar, rechazar crear, modificar y eliminar registros 	
4. El usuario realiza la acción correspondiente.	
5. El sistema valida la entrada, ejecuta la acción y guarda los cambios.	
6. Se muestra un mensaje de confirmación.	
Subflujos	
N/A	N/A
Flujos alternos	
N/A	N/A
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe controlar permisos por rol. • Las acciones deben estar disponibles desde cada módulo, no desde un único menú. 	
Postcondiciones	
<ul style="list-style-type: none"> • El registro queda actualizado en la base de datos. 	

Fuente: Elaboración Propia, 2026.

Tabla 14

Caso de uso para consultas

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-11	Nombre del caso de uso: Consultas
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite a los usuarios consultar información almacenada en los distintos módulos del sistema. Los colaboradores solo acceden a su propia información y los administradores acceden a todos los datos
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador, empleado

Precondiciones:	El usuario debe iniciar sesión. El sistema debe tener información disponible para consulta.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario inicia sesión. 2. Ingresa al módulo que desea consultar (ej. asistencia, vacaciones, incapacidades). 3. El sistema filtra automáticamente la información visible según su rol. 4. El usuario aplica filtros (fecha, tipo, estado). 5. El sistema muestra los resultados en pantalla. 6. El usuario puede navegar entre páginas o cambiar los filtros. 	
Subflujos	
N/A	N/A
Flujos alternos	
N/A	N/A
Requerimientos especiales	
<ul style="list-style-type: none"> • Las consultas deben aplicarse por rol. • El sistema debe ofrecer filtros por rango de fechas, estado, colaborador, etc. • Las consultas deben ser rápidas y en tiempo real. 	
Postcondiciones	
<ul style="list-style-type: none"> • La información consultada se visualiza sin modificarse. 	

Fuente: Tolentino, 2026.

Tabla 15

Caso de uso para reportes

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-12	Nombre del caso de uso: Reportes
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite a los usuarios generar reportes formateados con base en la información

	almacenada en el sistema. Los colaboradores generan reportes personales (asistencia, vacaciones, deducciones) y el administrador accede a reportes generales.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador, empleado
Precondiciones:	El usuario debe iniciar sesión y tener permisos de acceso al módulo correspondiente.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario inicia sesión. 2. Accede al módulo que permite reportes (ej. asistencia, nómina, horas extra). 3. Selecciona el tipo de reporte. 4. Aplica filtros (colaborador, fecha, tipo, estado). 5. El sistema genera una vista previa del reporte. 6. El usuario selecciona si desea exportarlo (PDF o Excel) o imprimirlo. 7. El sistema genera el documento solicitado. 	
Subflujos	
N/A	N/A
Flujos alternos	
N/A	N/A
Requerimientos especiales	
<ul style="list-style-type: none"> • Los reportes deben estar disponibles en formatos PDF y Excel. • Deben tener filtros flexibles según rol del usuario. • El formato debe ser profesional, imprimible y organizado. 	
Postcondiciones	
<ul style="list-style-type: none"> • El reporte queda disponible para descarga inmediata. 	

Fuente: Tolentino, 2026.

Tabla 16

Caso de uso para seguridad

Propuesta de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi´s House, ubicada en Tibás, San José.	
Número caso de uso: CU-13	Nombre del caso de uso: Seguridad
Fecha elaboración:	31/01/2026
Descripción caso de uso:	Permite proteger el acceso al sistema mediante mecanismos de autenticación, validación de sesiones y control de permisos según el rol del usuario. Se asegura que cada actor solo acceda a las funciones y datos autorizados, y que las credenciales estén protegidas mediante cifrado y restricciones.
Autor caso de uso:	Ayran Tolentino Quesada
Actores relacionados:	Sistema, administrador, empleado
Precondiciones:	El usuario debe estar registrado en el sistema con una cuenta activa y rol asignado.
Flujo básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario accede al sistema. 2. El sistema muestra el formulario de inicio de sesión. 3. El usuario digita su nombre de usuario y contraseña. 4. El sistema valida las credenciales ingresadas. 5. Si son correctas, permite el acceso y redirige al panel correspondiente según el rol. 6. Se activa la sesión de usuario. 7. Durante el uso del sistema, se controlan las funciones permitidas para su perfil. 8. El sistema puede cerrar sesión automáticamente tras un período de inactividad. 9. El usuario puede cerrar sesión manualmente desde el panel. 	
Subflujos	
N/A	N/A
Flujos alternos	
N/A	N/A

Requerimientos especiales
<ul style="list-style-type: none"> • Las contraseñas deben almacenarse de forma cifrada. • Los usuarios deben tener roles definidos con permisos limitados. • Debe permitir revocar o desactivar cuentas.
Postcondiciones
N/A

Fuente: Tolentino, 2026.

Diseño

La sección de diseño del presente proyecto detalla los elementos que conforman la estructura interna y el funcionamiento lógico del sistema. En esta parte se abordan aspectos como el modelo arquitectónico adoptado, diagramas, diseño de la base de datos, diseño de interfaz de usuario, etc. Su propósito es ofrecer una visión clara de cómo está construido el sistema, no solo desde una perspectiva visual u operativa, sino también desde su lógica interna.

Asimismo, se explican los principios de diseño que permiten que el sistema sea escalable, seguro, mantenible y fácil de utilizar tanto para colaboradores como para el personal administrativo. Todo esto permite comprender cómo se logra una integración eficiente entre la solución tecnológica propuesta y las necesidades específicas del restaurante.

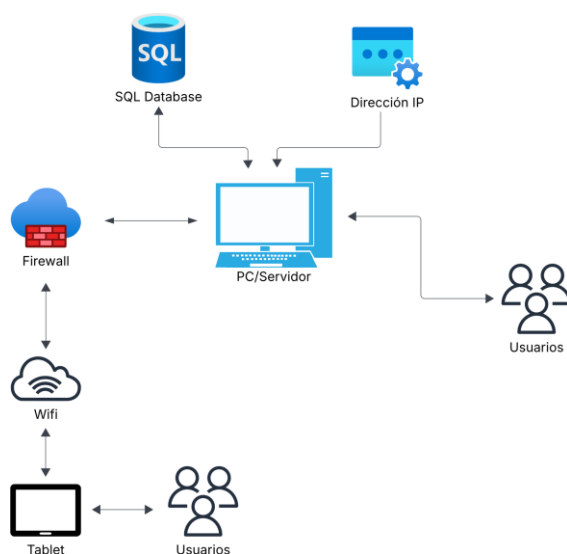
Arquitectura del sistema

La arquitectura del sistema propuesta está basada en un modelo centralizado, donde una computadora actúa como servidor principal (PC/Servidor). En esta máquina se encuentra alojada la aplicación del sistema de gestión, así como la conexión directa con la base de datos SQL que almacena toda la información relacionada con los usuarios, asistencias, vacaciones, planilla, entre otros módulos. Este servidor estará configurado con una dirección IP fija que permitirá que otros dispositivos dentro de la red puedan acceder a la aplicación de manera estable y sin interrupciones.

Los usuarios pueden interactuar con el sistema de dos formas. Por un lado, directamente desde la misma PC/Servidor, lo que permite un acceso inmediato sin depender de conexiones externas. Por otro lado, otros usuarios pueden ingresar al sistema mediante dispositivos móviles como tabletas, los cuales se conectan a través de una red WiFi interna. Para asegurar esta comunicación, el tráfico de datos pasa por un firewall que controla y protege el acceso, previniendo accesos no autorizados.

Figura 15

Diagrama de arquitectura del sistema



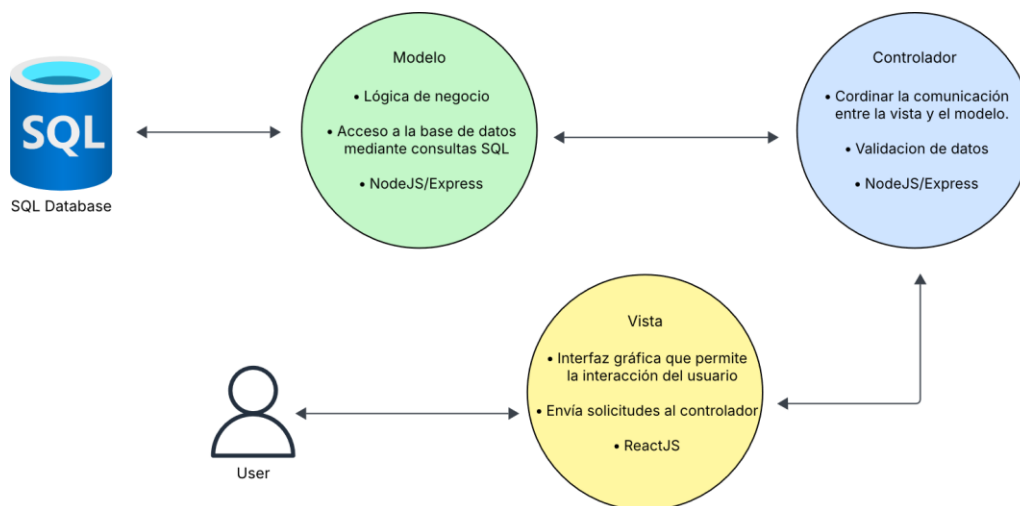
Fuente: Tolentino, 2026.

El hecho de que el servidor y la aplicación estén en la misma computadora reduce costos y facilita el mantenimiento, mientras que la posibilidad de conectarse desde otros dispositivos brinda comodidad al personal gerencial. Esta flexibilidad permite que el sistema pueda ser utilizado tanto de forma local como remota dentro de la misma red, adaptándose a las necesidades operativas del negocio sin requerir una infraestructura tecnológica compleja.

Arquitectura del Software

Figura 16

Diagrama de arquitectura MVC



Fuente: Tolentino, 2026.

Como se explica anteriormente en este proyecto, la arquitectura del software está estructurado bajo el patrón MVC (Modelo-Vista-Controlador), el cual separa la lógica de negocio, la interfaz gráfica y el flujo de control. En la figura 16 se puede observar cómo el usuario interactúa directamente con la Vista, que representa la interfaz del sistema, enviando solicitudes que son procesadas por el Controlador. Este último se encarga de interpretar las acciones del usuario y comunicarse con el Modelo para acceder o modificar los datos necesarios.

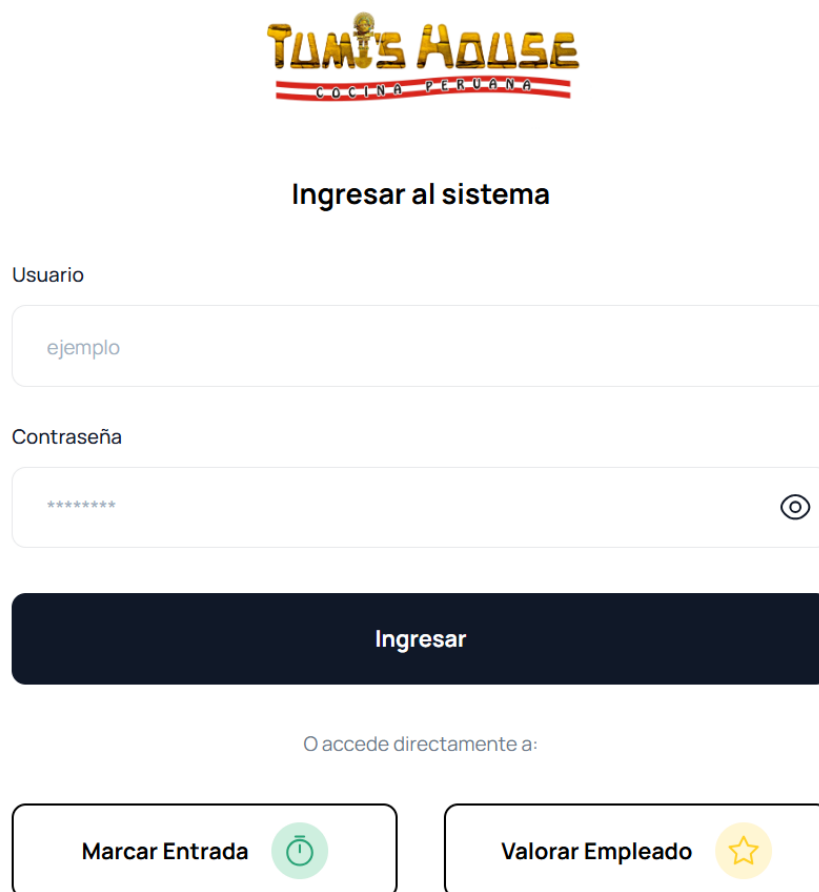
El Modelo representa la lógica de negocio y se conecta directamente con la base de datos SQL, de donde obtiene o actualiza la información según lo indicado por el controlador. Una vez procesados los datos, el modelo los retorna al controlador, que a su vez actualiza la vista para mostrar al usuario la información correspondiente. Este flujo asegura una organización clara del código y facilita el mantenimiento y escalabilidad del sistema.

Diseño de entradas y salidas

En este apartado se presentan las interfaces de usuario del sistema, las cuales muestran cómo se visualizan algunas pantallas y funcionalidades dentro de la aplicación. Estas interfaces permiten observar la estructura visual, la organización de los elementos y la forma en que los usuarios interactúan con el sistema para realizar distintas tareas. El objetivo es ilustrar el diseño de la aplicación y la experiencia que tendrán los usuarios al utilizar el sistema.

Figura 17

Diseño de entrada para la página de iniciar sesión





The image shows a login interface for 'TUMI'S HOUSE COCINA PERUANA'. At the top is the logo, which features the text 'TUMI'S HOUSE' in a stylized, bold font with a small house icon above the 'I', and 'COCINA PERUANA' in a smaller font below it, all set against a red and white striped background. Below the logo is the heading 'Ingresar al sistema'. There are two input fields: 'Usuario' with the placeholder text 'ejemplo' and 'Contraseña' with placeholder text '*****'. A dark blue button labeled 'Ingresar' is positioned below the password field. Underneath the button is the text 'O accede directamente a:'. At the bottom, there are two buttons: 'Marcar Entrada' with a clock icon and 'Valorar Empleado' with a star icon.

Fuente: Tolentino, 2026.

La interfaz de la figura 17 corresponde a la pantalla de inicio de sesión del sistema. En esta página, el usuario debe ingresar su nombre de usuario y contraseña para acceder a las funcionalidades del sistema de gestión de recursos humanos. Además, se incluyen accesos rápidos que permiten dirigirse directamente a funciones específicas, como marcar la entrada o valorar a un empleado, facilitando el uso del sistema para los diferentes tipos de usuarios. La interfaz fue diseñada de forma simple y clara para permitir un acceso rápido y seguro al sistema.

Figura 18

Diseño de salida para la pantalla de gestión de empleados

Empleado	Cédula	Usuario	Puesto	Estado
 Pablo Valverde pablo.valverde@...	407495103	Pvalverde	Cocinero	Activo
 David Torres davidtorres@g...	117990135	Dtorres	Cocinero	Activo

Fuente: Tolentino, 2026.

La figura 18 corresponde a la sección de gestión de empleados del sistema. En esta pantalla se muestra un listado de los empleados registrados, incluyendo información relevante como nombre, cédula, usuario, puesto y estado. Además, la interfaz permite buscar, filtrar y administrar los registros, así como agregar nuevos empleados o descargar la información. Este diseño facilita la consulta y administración eficiente de los datos del personal dentro del sistema.

Figura 19

Diseño de entrada para la pantalla de agregar empleado

The image shows a user registration interface. On the left, there is a progress indicator with five green dots, the text 'PASO 5 DE 5', and the main heading 'Agrega un nuevo empleado'. Below the heading, it says 'Completa los siguientes campos para registrar un empleado en el sistema.' At the bottom left are two buttons: 'Atrás' and 'Finalizar'. On the right, a form titled 'Usuario' contains the following fields: 'Nombre Usuario' (with the value 'Knavas'), 'PIN*' (with the placeholder 'Ingrese PIN'), 'Rol de usuario*' (a dropdown menu with 'Seleccione una opción'), and 'Nueva contraseña*' (with the placeholder 'Enter new password'). Below the password field, there are four error messages: 'Min. 8 caracteres', 'Al menos una letra', 'Al menos un número', and 'Al menos un símbolo(-, ., #, etc)'. There are also eye icons for toggling password visibility.

Fuente: Tolentino, 2026.

La interfaz de la figura 19 corresponde a un formulario de registro de empleados dividido en varios pasos. Este tipo de diseño permite ingresar la información de un nuevo empleado de manera organizada y progresiva, separando los datos en diferentes etapas para facilitar el proceso. De esta forma, el usuario puede completar la información requerida paso a paso, reduciendo errores y mejorando la experiencia de uso al momento de registrar un nuevo empleado en el sistema.

Figura 20*Diseño de entrada para agregar un turno*

Horarios

Asignar semana ← Semana atrás

lun 2	mar 3	mié 4	jue 5	vie 6	sáb 7
APERTURA 9:00 A.M - 18:00 P.M Almuerzo: 12:00:00 Café: 15:15:00 8h	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno
+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno
+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno
+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno
+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno
NCAPACIDAD	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno	+ Agregar turno

Agregar turno

Fecha: domingo, 8 de marzo de 2026
 Empleado: Ayram Tolentino Quesada

Tipo Turno *

Turno Apertura: 09:00 A.M - 18:00 P.M

Hora Almuerzo *

12:00 P.M

Hora Almuerzo *

03:14 P.M

Cancelar Guardar

Fuente: Tolentino, 2026.

En la figura 20 se muestra el módulo de gestión de horarios del sistema. En esta sección se visualiza la planificación semanal de turnos de los empleados, permitiendo organizar y administrar los horarios de trabajo de manera estructurada. A través de esta interfaz, el administrador puede asignar nuevos turnos, modificar los existentes y gestionar la distribución de jornadas laborales durante la semana, facilitando la planificación del personal dentro de la empresa.

Figura 21

Diseño de salida para la boleta de pago de nómina

Tumi's House

BOLETA DE PAGO

PLANILLA 13	CEDULA NO. 788546238	NOMBRE DEL EMPLEADO Diego Solano Rojas	FECHA INICIO 28 ene 2026	FECHA FINAL 11 feb 2026	MONEDA CRC
NOMINA 153	FECHA INGRESO 10 jun 2020	PUESTO Seguridad	SALARIO REF. € 428,900.00		

Ingresos	Días/Horas	Total	Egresos	Días/Horas	Total
SALARIO ORDINARIO	104 horas	€ 185,856.67	CCSS		€ 22,881.82
VACACIONES	2 días	€ 28,593.33	EXCESO ALMUERZO	20 min	€ 595.69
Total ingresos		€ 214,450.00	Total egresos		€ 23,477.51

Total Ganado Neto:	€ 190,972.49
---------------------------	---------------------

Fuente: Tolentino, 2026.

La figura 21 corresponde a la boleta de pago generada por el sistema de nómina. En esta pantalla se presenta un resumen detallado de los ingresos y egresos asociados a un periodo de pago específico de un empleado. Además, se muestran datos generales del trabajador y del periodo de nómina, permitiendo visualizar de forma clara el cálculo del salario y el monto neto a recibir. Esta interfaz facilita la consulta y verificación de la información relacionada con el pago del empleado.

Diseño de la Base de Datos

La base de datos del sistema está estructurada en entidades principales que permiten la gestión de recursos humanos. La estructura central gira alrededor del empleado, representado por las tablas Empleado y EmpleadoDetalle, donde se almacena la información personal, laboral y de contacto, complementada por catálogos como puesto, género, estado civil y ubicación geográfica. De forma complementaria, el sistema incorpora la entidad Usuario, la cual se encuentra asociada al empleado y permite la gestión de credenciales de acceso, roles y permisos, garantizando un control adecuado de la autenticación y autorización dentro del sistema.

Diccionario de Datos

Tabla 17

Diccionario de datos para tabla Empleado

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id empleado	INT	X		No	
cedula	VARCHAR(100)			No	
nombre	VARCHAR(100)			No	
primer_apellido	VARCHAR(100)			No	
segundo_apellido	VARCHAR(100)			No	

Fuente: Tolentino, 2026.

Tabla 18

Diccionario de datos para tabla Empleado_Detalle

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id empleado	INT	X	X	No	
fecha_nacimiento	DATE			No	
cantidad_hijos	TINYINT			No	
estado_laboral	BIT			No	0 = Inactivo 1 = Activo
salario_base	DECIMAL(10,2)			No	
fecha_entrada	DATE			No	
foto_url	VARCHAR(255)			Sí	
id_genero	TINYINT		X	No	
id_estado_civil	TINYINT		X	No	
id_puesto	TINYINT		X	No	

Fuente: Tolentino, 2026.

Tabla 19*Diccionario de datos para tabla Género*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_genero	TINYINT	X		No	1. Masculino 2. Femenino
genero_descripcion	VARCHAR(50)			No	

*Fuente: Tolentino, 2026.***Tabla 20***Diccionario de datos para tabla Estado_Civil*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_estado_civil	TINYINT	X		No	1. Soltero(a) 2. Casado(a) 3. Divorciado 4. Viudo(a) 5. Unión libre
estado_civil_descripcion	VARCHAR(50)			No	

*Fuente: Tolentino, 2026.***Tabla 21***Diccionario de datos para tabla Puesto*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_puesto	TINYINT	X		No	1. Administrador 2. Salonero 3. Bartender 4. Seguridad 5. Cocinero 6. Cajero

					7. Conserje 8. Ayudante de cocina 9. Encargado de bodega 10. Repartidor 11. Anfitrión o Recepcionista 12. Lavaplatos
nombre_puesto	VARCHAR(100)			No	

Fuente: Tolentino, 2026.

Tabla 22

Diccionario de datos para tabla TelefonoEmpleado

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id telefono	INT	X		No	
id_ empleado	INT		X	No	
id_tipo_telefono	TINYINT		X	No	
titulo_telefono	VARCHAR(50)				
telefono	VARCHAR(100)			No	
principal	BIT(1)			No	

Fuente: Tolentino, 2026.

Tabla 23

Diccionario de datos para tabla TipoTelefono

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_tipo_telefono	TINYINT	X		No	1. Personal 2. Laboral 3. Emergencia
tipo_telefono_descripcion	VARCHAR(50)			No	

Fuente: Tolentino, 2026.

Tabla 24*Diccionario de datos para tabla CorreoEmpleado*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id correo	INT	X		No	
id_ empleado	INT		X	No	
id_tipo_correo	TINYINT		X	No	
titulo_correo	VARCHAR(50)				
correo	VARCHAR(100)			No	
principal	BIT(1)			No	

*Fuente: Tolentino, 2026***Tabla 25***Diccionario de datos para tabla TipoCorreo*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_tipo_correo	TINYINT	X		No	1. Personal 2. Laboral 3. Emergencia
tipo_correo_descripcion	VARCHAR(50)			No	

*Fuente: Tolentino, 2026***Tabla 26***Diccionario de datos para tabla Dirección*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id direccion	INT	X		No	
id_ empleado	INT		X	No	
id_distrito	SMALLINT		X	No	
direccion_exacta	VARCHAR(200)			No	

Fuente: Tolentino, 2026.

Tabla 27*Diccionario de datos para tabla Distrito*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_distrito	SMALLINT	X		No	
id_canton	TINYINT		X	No	
Nombre	VARCHAR(50)			No	

*Fuente: Tolentino, 2026.***Tabla 28***Diccionario de datos para tabla Cantón*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_canton	TINYINT	X		No	
id_provincia	TINYINT		X	No	
nombre	VARCHAR(50)			No	

*Fuente: Tolentino, 2026.***Tabla 29***Diccionario de datos para tabla Provincia*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_provincia	TINYINT	X		No	
nombre	VARCHAR(50)			No	

Fuente: Tolentino, 2026.

Tabla 30*Diccionario de datos para tabla Usuario*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_usuario	INT	X		No	
id_empleado	INT		X	No	
id_rol_usuario	INT		X	No	
usuario	VARCHAR(50)			No	
contrasena	VARCHAR(250)			No	
pin	VARCHAR(4)			No	

*Fuente: Tolentino, 2026.***Tabla 31***Diccionario de datos para tabla Rol_Usuario*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_rol_usuario	INT	X		No	1. Administrador 2. Empleado
rol_descripcion	VARCHAR(50)			No	

*Fuente: Tolentino, 2026.***Tabla 32***Diccionario de datos para tabla Tipo_Cese*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id tipo cese	INT	X		No	
descripcion	VARCHAR(50)			No	1. Despido sin responsabilidad del trabajador 2. Despido con responsabilidad del trabajador 3. Renuncia voluntaria

paga_preaviso	BIT(1)			No	
paga_cesantia	BIT(1)			No	

Fuente: Tolentino, 2026.

Tabla 33

Diccionario de datos para tabla Liquidación

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id liquidacion	INT	X		No	
id_employado	INT		X	No	
id_tipo_cese	INT		X	No	
fecha_ingreso	DATE			No	
fecha_salida	DATE			No	
estado	CHAR(10)			No	
credado_en	DATETIME			No	

Fuente: Tolentino, 2026.

Tabla 34

Diccionario de datos para tabla LiquidaciónDetalle

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id detalle liquidacion	INT	X		No	
id_liquidacion	INT		X	No	
tipo_detalle	CHAR(10)			No	
categoria_detalle	CHAR (45)			No	
cantidad	DECIMAL(12,2)			No	
unidad	CHAR(10)			No	
monto	DECIMAL(12,2)			No	

Fuente: Tolentino, 2026.

Tabla 35*Diccionario de datos para tabla PeríodoNomina*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id periodo	INT	X		No	
fecha_inicio	DATE		X	No	
fecha_fin	DATE			No	
tipo_periodo	VARCHAR(100)			No	
estado	DECIMAL(10,2)			No	
creado_en	DATETIME			No	

Fuente: Tolentino, 2026.**Tabla 36***Diccionario de datos para tabla NóminaDetalle*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_detalle	INT	X		No	
id_nomina	INT		X	No	
id_deducción_ley	INT		X	No	
tipo_detalle	CHAR(10)			No	
cantidad	DECIMAL(12,2)			No	
unidad	CHAR(10)			No	
monto	DECIMAL(12,2)			No	

Fuente: Tolentino, 2026.**Tabla 37***Diccionario de datos para tabla Nómina*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_nomina	INT	X		No	
id_periodo	INT		X	No	
id_empleado	INT		X	No	

estado_nomina	CHAR(10)			No	
generado_en	DATETIME			No	

Fuente: Tolentino, 2026.

Tabla 38

Diccionario de datos para tabla DeducciónLey

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id deduccion	INT	X		No	
tipo_deduccion	VARCHAR(20)			No	
codigo_deduccion	VARCHAR(20)			No	
nombre_deduccion	VARCHAR(60)			No	
descripcion	VARCHAR(500)			No	
porcentaje	DECIMAL(6,4)			No	
limite_inferior	DECIMAL(12,2)			Sí	
limite_superior	DECIMAL(12,2)			Sí	
tasa	DECIMAL(6,4)			Sí	
fecha_inicio	DATE			No	
fecha_fin	DATE			Sí	
estado_deduccion_ley	BIT(1)			No	
creado_en	DATETIME			No	

Fuente: Tolentino, 2026.

Tabla 39

Diccionario de datos para tabla DeducciónEmpleado

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id deduccion empleado	INT	X		No	
id_empleado	INT		X	No	
tipo_deduccion empleado	VARCHAR(20)			No	
descripcion_deduccion	VARCHAR(60)			No	

modalida	CHAR(10)			No	
valor	DECIMAL(12,2)			No	
fecha_inicio	DATE			No	
fecha_fin	DATE			Sí	
activa	BIT(1)			No	
creado_en	DATETIME			No	

Fuente: Tolentino, 2026.

Tabla 40

Diccionario de datos para tabla CréditoISR

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_credito	INT	X		No	
id_empleado	INT		X	No	
codigo	VARCHAR(20)			No	
monto_mensual	DECIMAL(12,2)			No	
fecha_inicio	DATE			No	
fecha_fin	DATE			Sí	
activo	DATE			No	

Fuente: Tolentino, 2026.

Tabla 41

Diccionario de datos para tabla PeríodoAguinaldo

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_periodo	INT	X		No	
fecha_inicio	DATE		X	No	
fecha_fin	DATE			No	
estado	DECIMAL(10,2)			No	
creado_en	DATETIME			No	

Fuente: Tolentino, 2026.

Tabla 42*Diccionario de datos para tabla Aguinaldo*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id aguinaldo	INT	X		No	
id_periodo	INT		X	No	
id_empleado	INT		X	No	
estado_aguinaldo	DECIMAL(12,2)			No	
creado_en	DATETIME			No	

*Fuente: Tolentino, 2026.***Tabla 43***Diccionario de datos para tabla Incapacidad*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id incapacidad	INT	X		No	
id_empleado	INT		X	No	
fecha_inicio	DATE			No	
fecha_fin	DATE			No	
id_tipo_incapacidad	TINYINT		X	No	
incapacidad_detalle	VARCHAR(250)			No	
fecha_registro	TIMESTAMP			No	
estado_incapacidad	CHAR(10)			No	
dias_incapacidad	SMALLINT			No	
archivo_url	VARCHAR(255)			Sí	
motivo_estado	VARCHAR(100)			No	

*Fuente: Tolentino, 2026.***Tabla 44***Diccionario de datos para tabla TipoIncapacidad*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_tipo_incapacidad	TINYINT	X		No	

tipo_incapacidad_nombre	VARCHAR(100)		X	No	
tipo_incapacidad_descripcion	VARCHAR(100)			No	
porcentaje_patrono	DECIMAL(5,2)			No	
porcentaje_entidad_externa	DECIMAL(5,2)			No	
entidad_responsable	VARCHAR(100)			No	

Fuente: Tolentino, 2026.

Tabla 45

Diccionario de datos para tabla HorasExtras

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id horas extras	INT	X		No	
id_empleador	INT		X	No	
id_solicitud_horas_extra	INT		X	No	
fecha	DATE			No	
hora_inicio	TIME			No	
hora_fin	TIME			No	
horas_total	DECIMAL(4,2)			No	
hora_extra_estado	CHAR(10)			No	
observacion	TEXT			Sí	

Fuente: Tolentino, 2026.

Tabla 46

Diccionario de datos para tabla TurnoAsignado

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id turno asignado	INT	X		No	
id_empleador	INT		X	No	
id_tipo_turno	TINYINT		X	No	
hora_almuerzo	TIME			No	
hora_cafe	TIME			No	
Fecha	DATE			No	

Fuente: Tolentino, 2026.

Tabla 47

Diccionario de datos para tabla TipoTurno

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id tipo turno	TINYINT	X		No	
tipo_turno_descripcion	VARCHAR(100)			No	
hora_inicio	TIME			No	
hora_fin	TIME			No	
total_horas	DECIMAL(4,2)			No	

Fuente: Tolentino, 2026.

Tabla 48

Diccionario de datos para tabla MarcaAsistencia

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_marca	INT	X		No	
id_empleado	INT		X	No	
id_tipo_marca	TINYINT		X	No	
fecha	DATE			No	
hora	TIME			No	

Fuente: Tolentino, 2026.

Tabla 49

Diccionario de datos para tabla TipoMarca

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id tipo_marca	TINYINT	X		No	

tipo_marca_descripcion	VARCHAR(50)		X	No	1. Ingreso 2. Almuerzo Inicio 3. Almuerzo Fin 4. Cafe Inicio 5. Cafe fin 6. Salida
------------------------	-------------	--	---	----	---

Fuente: Tolentino, 2026.

Tabla 50

Diccionario de datos para tabla IncidenciaAsistencia

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_incidencia	INT	X		No	
id_empleado	INT		X	No	
id_tipo_incidencia	TINYINT		X	No	
fecha	DATE			No	
minutos_exceso	INT			No	
estado	CHAR(10)			No	
observacion	TEXT			No	

Fuente: Tolentino, 2026.

Tabla 51

Diccionario de datos para tabla TipoIncidencia

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_tipo_incidencia	TINYINT	X		No	

tipo_incidencia_descripcion	VARCHAR(50)			No	1. Atraso 2. Exceso_almuerzo 3. Salida_anticipada 4. Exceso_descanso 5. Falta_marca_salida 6. Ausente
-----------------------------	-------------	--	--	----	--

Fuente: Tolentino, 2026.

Tabla 52

Diccionario de datos para tabla PermisoLaboral

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_permiso_laboral	INT	X		No	
id_empleado	INT		X	No	
id_tipo_permiso	TINYINT		X	No	
fecha_solicitud	DATE			No	
fecha_inicio	DATE			No	
fecha_fin	DATE			No	
unidad_permiso	CHAR(10)			No	
días_solicitados	SMALLINT			No	
hora_inicio	TIME			No	
hora_fin	TIME			No	
horas_solicitadas	DECIMAL(5,2)			No	
permiso_detalle	VARCHAR(250)			No	
permiso_estado	CHAR(10)			No	
motivo_estado	VARCHAR(250)			No	

Fuente: Tolentino, 2026.

Tabla 53*Diccionario de datos para tabla TipoPermisoLaboral*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id tipo permiso	TINYINT	X		No	
tipo_descripcion	VARCHAR(100)			No	
goce_salarial	BOOLEAN			No	

*Fuente: Tolentino, 2026.***Tabla 54***Diccionario de datos para tabla Vacaciones*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_solicitud	INT	X		No	
id_empleado	INT		X	No	
fecha_solicitud	DATE			No	
fecha_fin	DATE			No	
fecha_inicio	DATE			No	
dias_solicitados	INT			No	
estado_solicitud	CHAR(10)			No	
solicitud_detalle	VARCHAR(100)			No	
motivo_estado	VARCHAR(100)			Sí	

*Fuente: Tolentino, 2026.***Tabla 55***Diccionario de datos para tabla VacacionesSaldo*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_empleado	INT	X	X	No	
dias_acumulados	TINYINT			No	
dias_usados	SMALLINT			No	

Fuente: Tolentino, 2026.

Tabla 56*Diccionario de datos para tabla AsistenciaDia*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id asistencia dia	INT	X		No	
id_ empleado	INT		X	No	
id_dia_feriado	INT		X	Sí	
fecha	DATE			No	
hora_ingreso	TIME			Sí	
hora_salida	TIME			Sí	
estado_dia	CHAR(20)			No	
tipo_dia	CHAR(15)			No	
horas_ordinarias	INT			No	
calculado_en	VARCHAR(250)			No	
es_feriado	BIT(1)			No	

*Fuente: Tolentino, 2026.***Tabla 57***Diccionario de datos para tabla SolicitudHorasExtra*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id solicitud horas extra	INT	X		No	
id_ empleado	INT		X	No	
fecha	DATE			No	
tipo_solicitud	CHAR(15)			No	
hora_inicio_solicitada	TIME			Sí	
hora_fin_solicitada	TIME			Sí	
horas_solicitadas	DECIMAL(4,2)			No	
estado_solicitud	CHAR(15)			No	
motivo_solicitud	INT			No	
motivo_respuesta	VARCHAR(250)			No	
fecha_solicitud	BIT(1)			No	
estado_ejecucion	CHAR(15)			No	

Fuente: Tolentino, 2026.

Tabla 58*Diccionario de datos para tabla DíaFeriado*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id día feriado	INT	X		No	
fecha	DATE			No	
dia_feriado_descripcion	VARCHAR(100)			No	
pago_obligatorio	BIT(1)			No	

*Fuente: Tolentino, 2026.***Tabla 59***Diccionario de datos para tabla CategoríaEvaluación*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id categoria	INT	X		No	
nombre	VARCHAR(50)			No	
descripcion	VARCHAR(100)			No	

*Fuente: Tolentino, 2026.***Tabla 60***Diccionario de datos para tabla PreguntaEvaluación*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id pregunta	INT	X		No	
id_categoria	INT		X	No	
texto_pregunta	VARCHAR(255)			No	
orden_visual	INT			No	
activa	BIT(1)			No	

Fuente: Tolentino, 2026.

Tabla 61*Diccionario de datos para tabla EvaluaciónCliente*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id evaluacion	INT	X		No	
id_empleado	INT		X	No	
fecha	DATE			No	
comentario	TEXT			No	

*Fuente: Tolentino, 2026.***Tabla 62***Diccionario de datos para tabla RespuestaEvaluación*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id respuesta	INT	X		No	
id_evaluacion	INT		X	No	
id_pregunta	INT			No	
Puntaje	TINYINT			No	

*Fuente: Tolentino, 2026.***Tabla 63***Diccionario de datos para tabla Notificación*

Columna	Tipo de Dato	PK	FK	Permite Nulos	Descripción
id_notificacion	INT	X		No	
id_usuario	INT		X	No	
titulo	VARCHAR(150)			No	
mensaje	TEXT			No	
tipo	VARCHAR(50)			No	
leida	BIT(1)			No	
fecha_creacion	DATETIME			No	

url_destino	VARCHAR(255)			No	
-------------	--------------	--	--	----	--

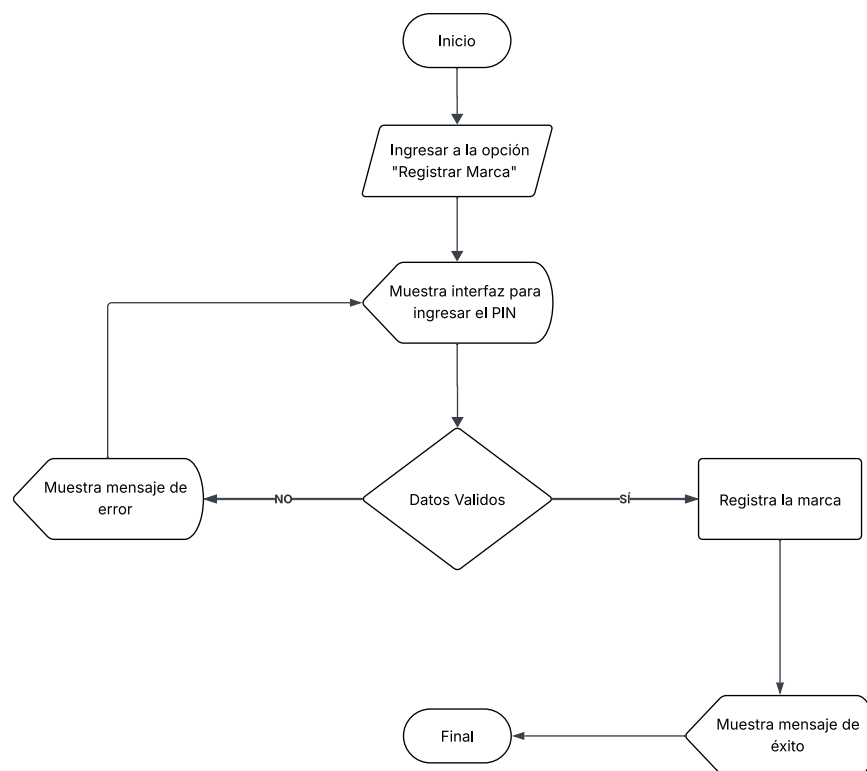
Fuente: Tolentino, 2026.

Diseño de Procesos

En este apartado se presentan los diagramas de procesos que describen el funcionamiento de las operaciones principales del prototipo. Estos diagramas permiten visualizar de manera clara la secuencia de actividades y los puntos de decisión, facilitando la comprensión del flujo de trabajo general. Asimismo, sirven como apoyo para el análisis y la documentación de los procesos, permitiendo detectar oportunidades de mejora y asegurar la correcta alineación con los objetivos del sistema.

Figura 23

Diagrama de flujo para registrar marca

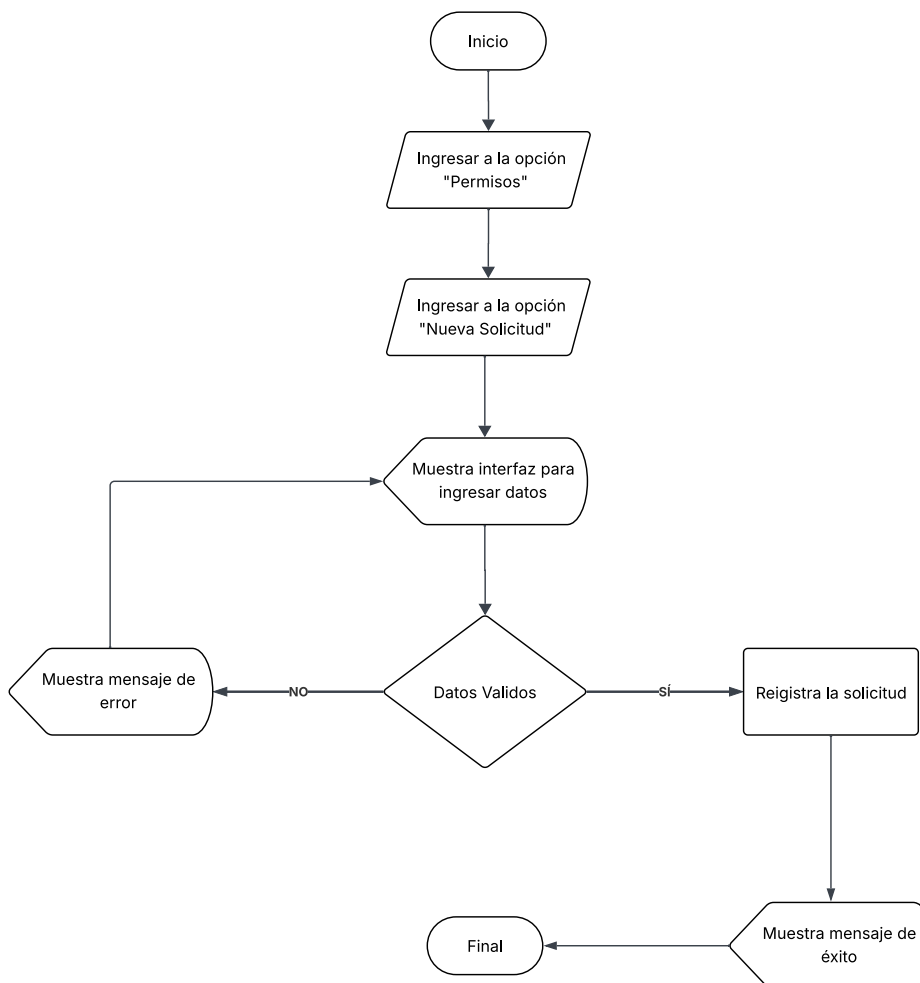


Fuente: Tolentino, 2026.

La Figura 23 presenta el diagrama de flujo correspondiente al proceso de registro de marcas de asistencia dentro del sistema, iniciando cuando el colaborador accede a la opción “Registrar Marca” e ingresa su PIN en la interfaz. Posteriormente, el sistema valida los datos proporcionados para verificar su autenticidad y coherencia. En caso de que la información sea incorrecta, se muestra un mensaje de error que solicita una nueva entrada de datos. Si la validación es exitosa, se procede a registrar la marca en la base de datos del sistema. Finalmente, se muestra un mensaje de confirmación indicando que el registro fue realizado correctamente, concluyendo así el proceso.

Figura 24

Diagrama de flujo para solicitar permiso

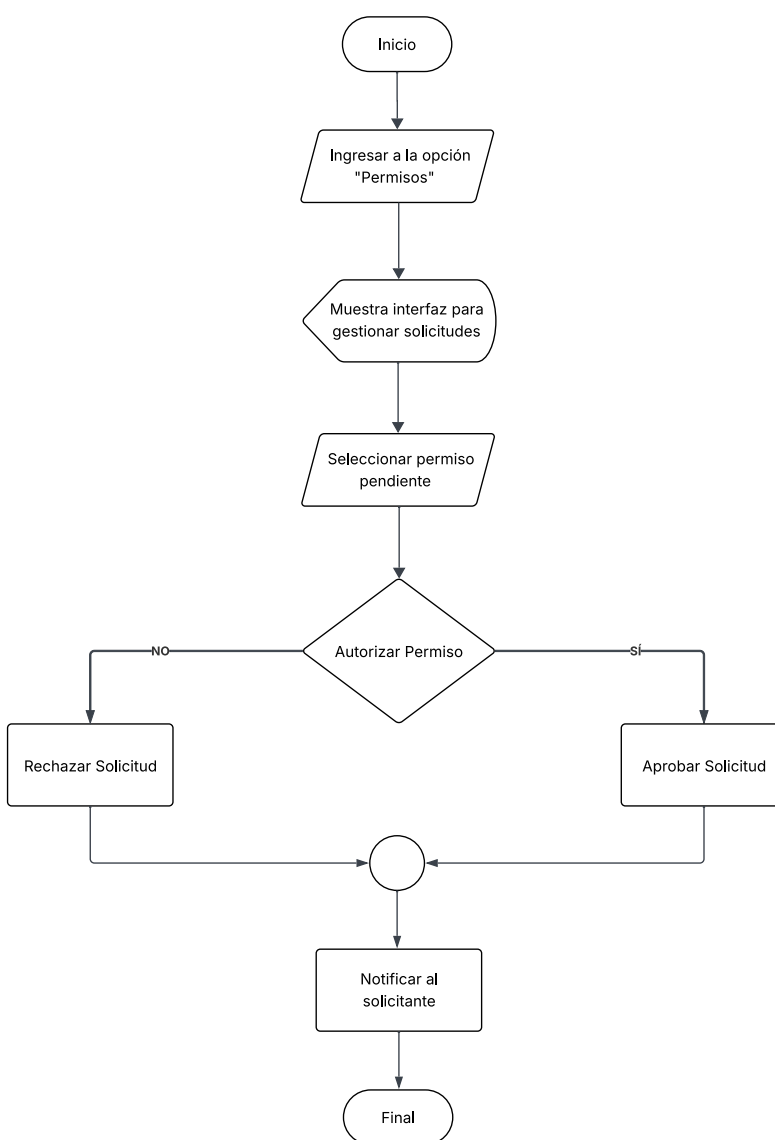


Fuente: Tolentino, 2026.

La Figura 24 presenta el diagrama de flujo para la solicitud de permisos, el cual sigue una estructura similar al proceso de registro de marcas de asistencia. El colaborador accede al módulo de permisos, ingresa los datos de la solicitud y el sistema valida la información proporcionada. Si existen errores, se muestra un mensaje de corrección; en caso contrario, se registra la solicitud correctamente. Finalmente, el sistema confirma la acción mediante un mensaje de éxito, concluyendo el proceso.

Figura 25

Diagrama de flujo para aprobar o rechazar solicitudes

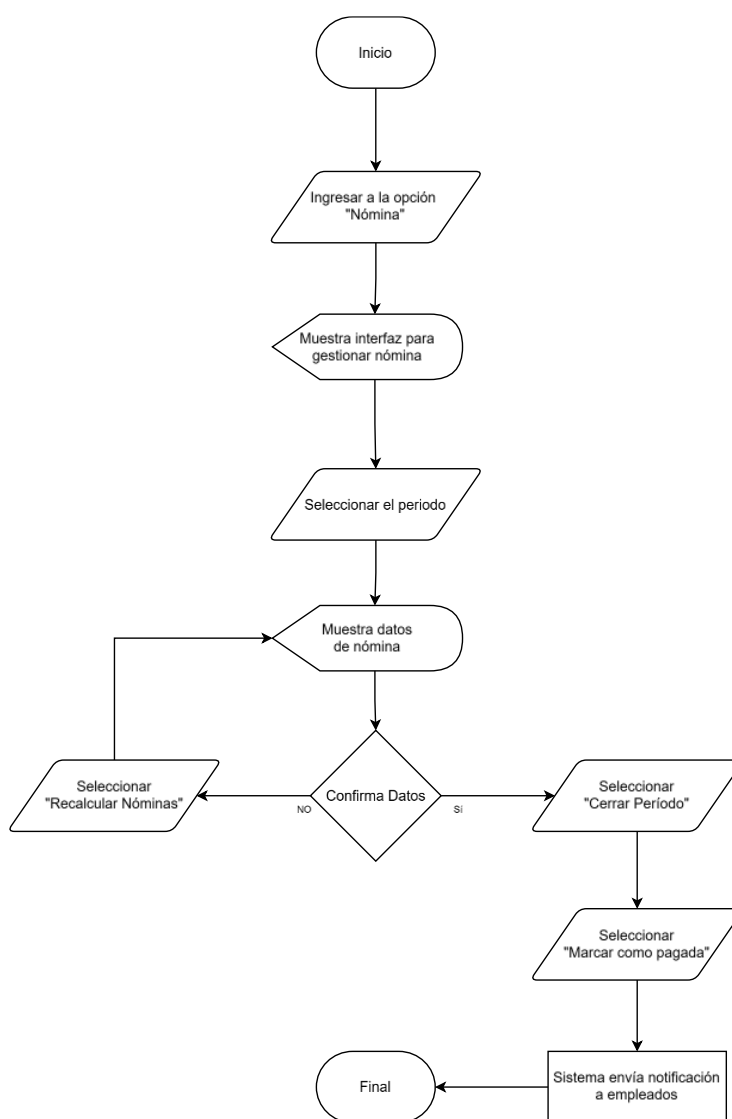


Fuente: Tolentino, 2026.

La Figura 25 ilustra el diagrama de flujo para la aprobación o rechazo de solicitudes de permisos por parte del administrador. El proceso inicia al ingresar al módulo de permisos, donde se muestra la interfaz para gestionar las solicitudes pendientes. Seguidamente, se selecciona una solicitud y se toma la decisión de autorizar o rechazar el permiso según corresponda. Una vez definida la acción, el sistema actualiza el estado de la solicitud y notifica al colaborador sobre la resolución. Finalmente, el proceso concluye tras enviar la notificación correspondiente.

Figura 26

Diagrama de flujo para el cierre de nómina

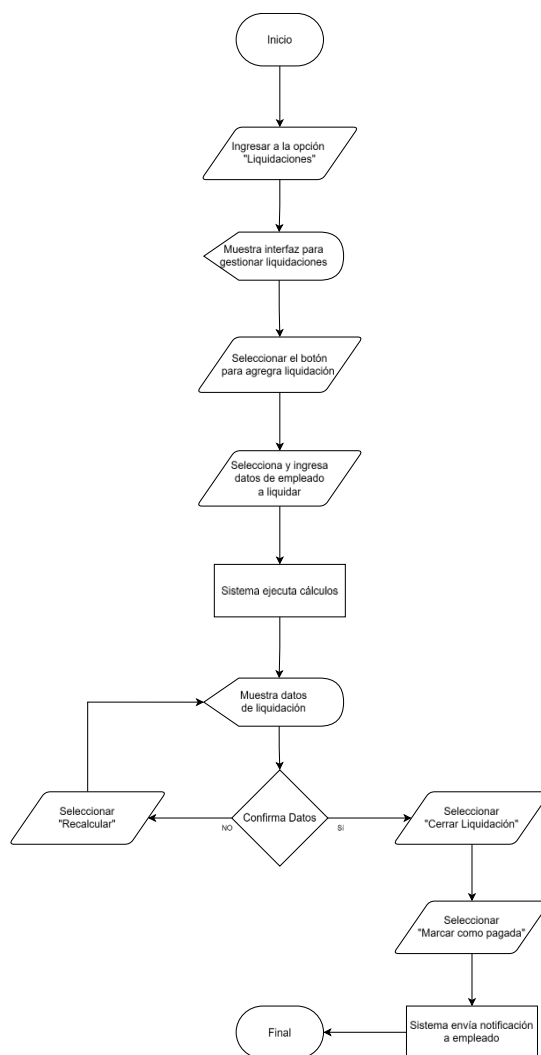


Fuente: Tolentino, 2026.

La Figura 26 muestra el diagrama de flujo correspondiente al proceso de cierre de nómina en el sistema. El procedimiento inicia cuando el administrador accede al módulo de “Nómina” y selecciona la opción de cerrar el período correspondiente. Posteriormente, el sistema ejecuta los cálculos automáticos y presenta los datos de la nómina para su revisión. Si la información es confirmada, se procede a notificar a los empleados sobre el cierre realizado. En caso contrario, el flujo retorna a la gestión de nómina para realizar los ajustes necesarios antes de finalizar el proceso.

Figura 27

Diagrama de flujo para cierre de liquidación

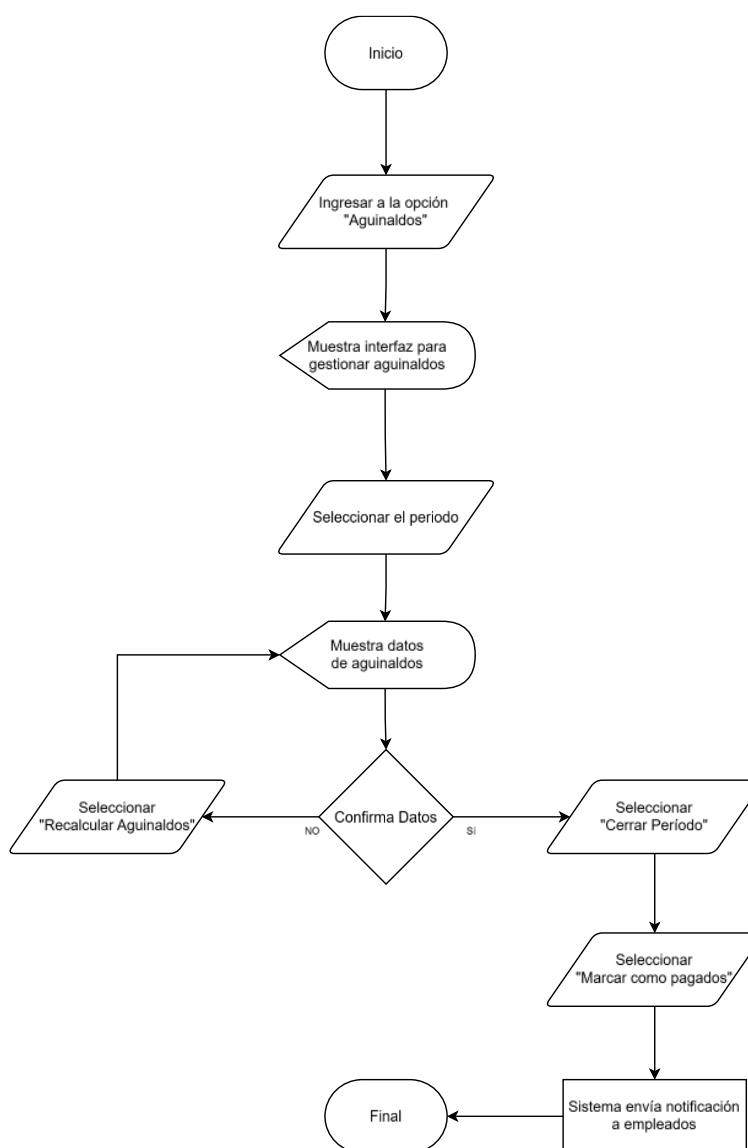


Fuente: Tolentino, 2026

La Figura 27 muestra el diagrama de flujo del proceso de liquidación de empleados, el cual inicia cuando el administrador ingresa al módulo de “Liquidación” del sistema. Posteriormente, se presenta la interfaz para gestionar las liquidaciones y se selecciona al colaborador a liquidar. El sistema ejecuta los cálculos automáticos y muestra el detalle correspondiente para su validación. Si la información es confirmada, se envía una notificación al empleado y se finaliza el proceso; en caso contrario, se retorna a la gestión para realizar los ajustes necesarios.

Figura 28

Diagrama de flujo para cierre de aguinaldos



Fuente: Tolentino, 2026

La Figura 28 presenta el diagrama de flujo del proceso para la gestión y cierre de aguinaldos, el cual es muy similar al procedimiento utilizado en el cierre de nómina y en la liquidación de empleados. El flujo inicia al ingresar al módulo de “Aguinaldo”, donde se muestra la interfaz de administración correspondiente. Posteriormente, el sistema ejecuta los cálculos automáticos y presenta los datos para su revisión. Una vez confirmada la información, se envía una notificación a los colaboradores y se finaliza el proceso. En caso de inconsistencias, el flujo retorna a la gestión para realizar los ajustes necesarios.

Diagramas UML

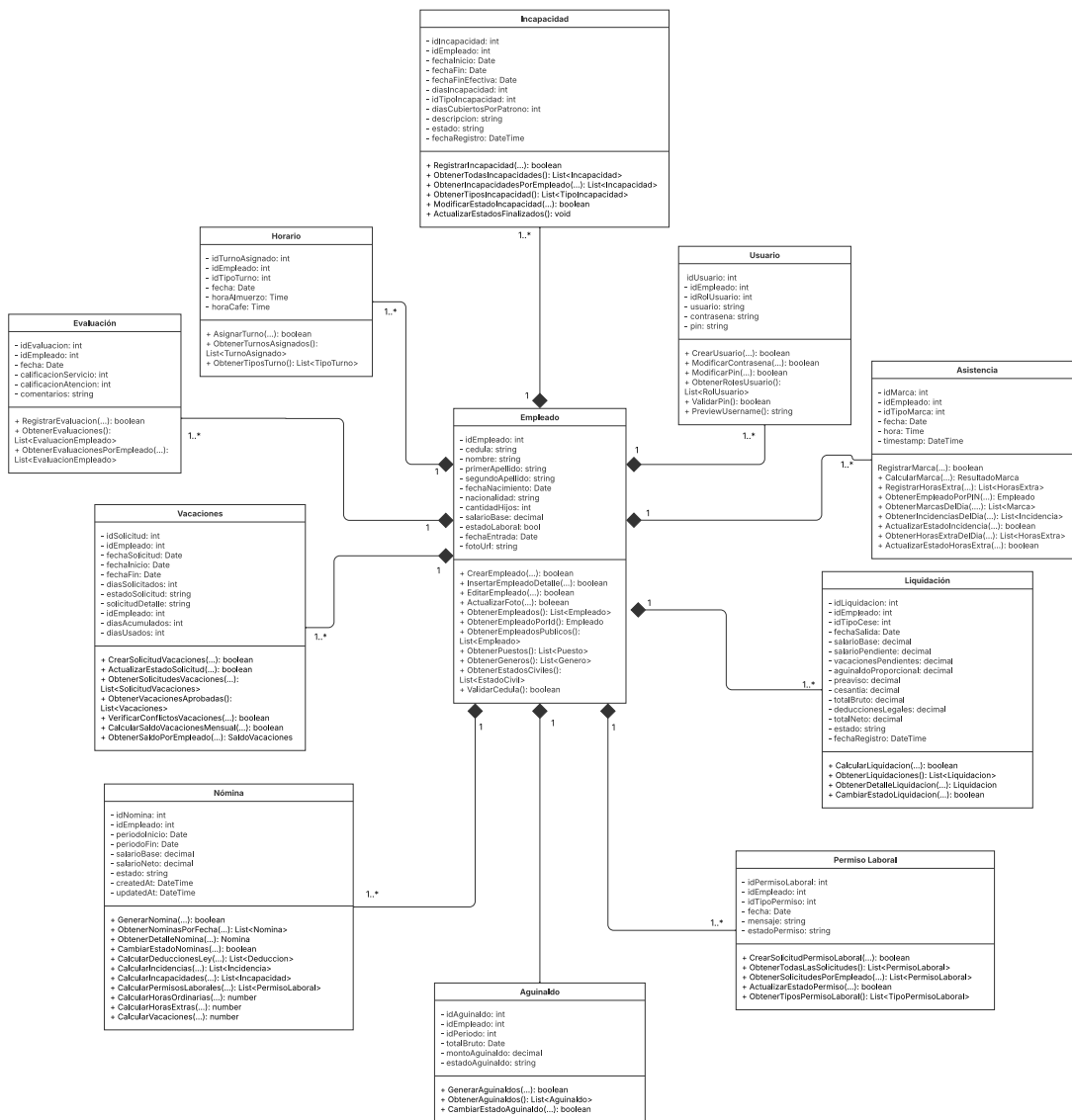
Los diagramas UML son utilizados para modelar la estructura y el comportamiento del sistema de gestión desarrollado. Estos diagramas permiten representar de forma gráfica cómo interactúan los distintos módulos, usuarios y procesos, ofreciendo una visión clara del diseño general de la aplicación. Su uso contribuye a organizar la información técnica, mejorar la comprensión del funcionamiento del sistema y apoyar tanto el desarrollo como la validación de los requerimientos planteados.

A continuación, se presentan diversos diagramas de diversos procesos. Cada diagrama permite visualizar de manera detallada el flujo de actividades y la interacción entre los actores involucrados, facilitando la comprensión de cómo se ejecutan las funciones clave dentro del sistema y evidenciando la integración entre la interfaz, el backend y la base de datos.

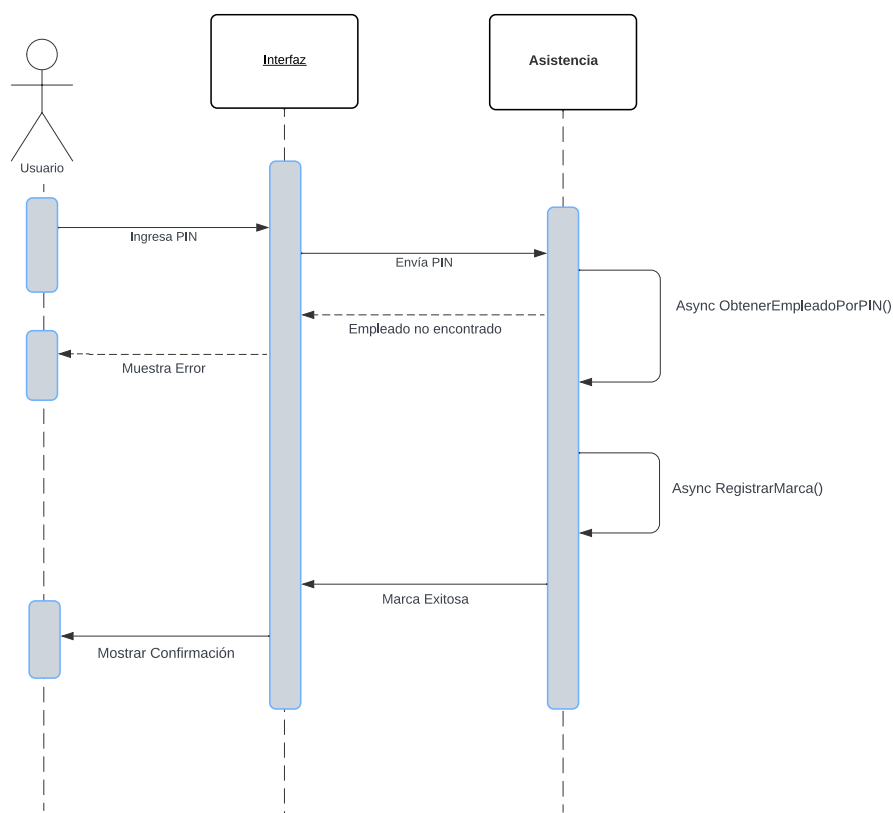
El siguiente diagrama de clases (ver Figura 29), representa la estructura lógica del sistema, donde la entidad central es Empleado, a partir de la cual se relacionan los principales módulos funcionales como asistencia, horarios, vacaciones, incapacidades, permisos laborales, nómina, liquidación y evaluaciones. Cada clase contiene sus atributos principales así como métodos que permiten registrar, consultar, actualizar y calcular información. En conjunto, el diagrama permite visualizar de forma clara la organización del sistema, la interacción entre módulos y la lógica de negocio que respalda la automatización de los procesos administrativos.

Figura 29

Diagrama de clases del sistema



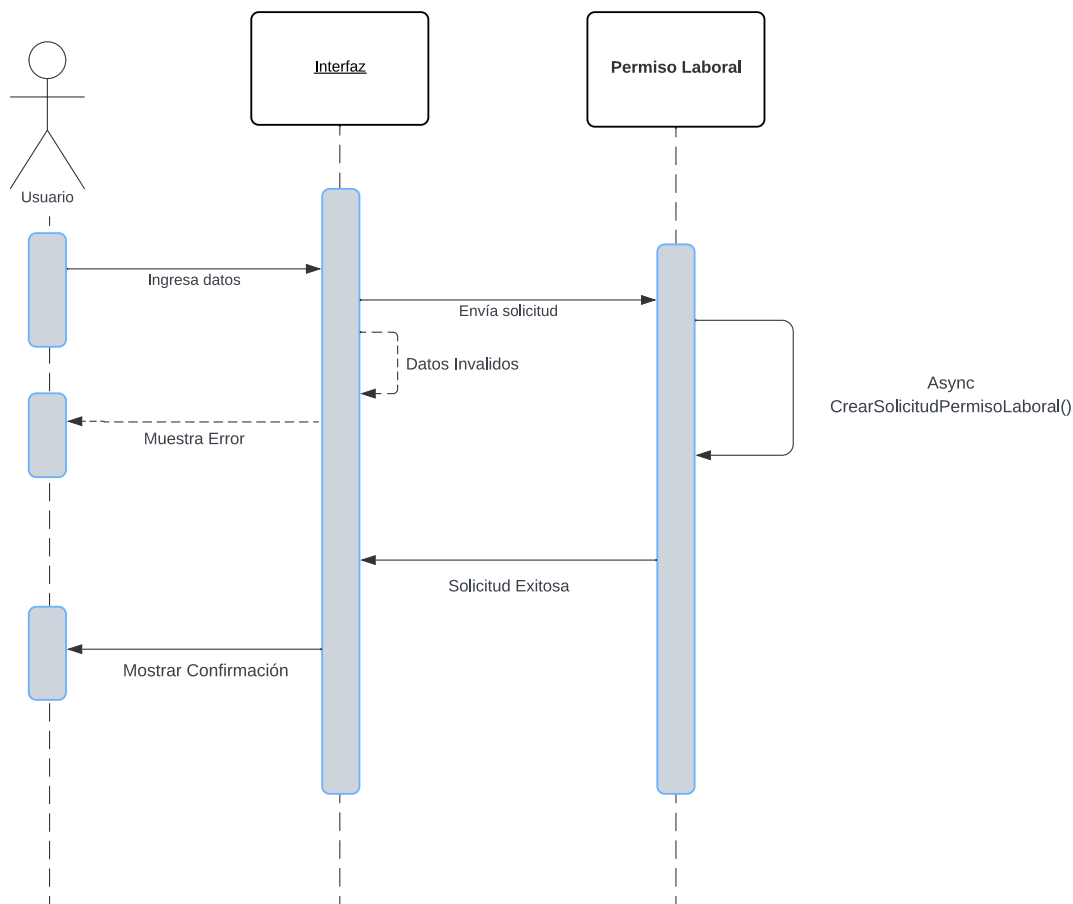
Fuente: Tolentino, 2026

Figura 30*Diagrama de secuencia de registrar marca**Fuente:* Tolentino, 2026

El diagrama de secuencia de la Figura 30 muestra el proceso de registro de una marca de asistencia, donde el usuario ingresa su PIN en la interfaz, la cual lo envía al sistema para validar la existencia del empleado en la base de datos. Si el empleado no es encontrado, el sistema retorna un error y se notifica al usuario; en caso contrario, se procede a insertar la marca de asistencia y se confirma exitosamente la operación. Este flujo garantiza la validación previa del empleado antes de registrar cualquier marca en el sistema.

Figura 31

Diagrama de secuencia para solicitar permiso

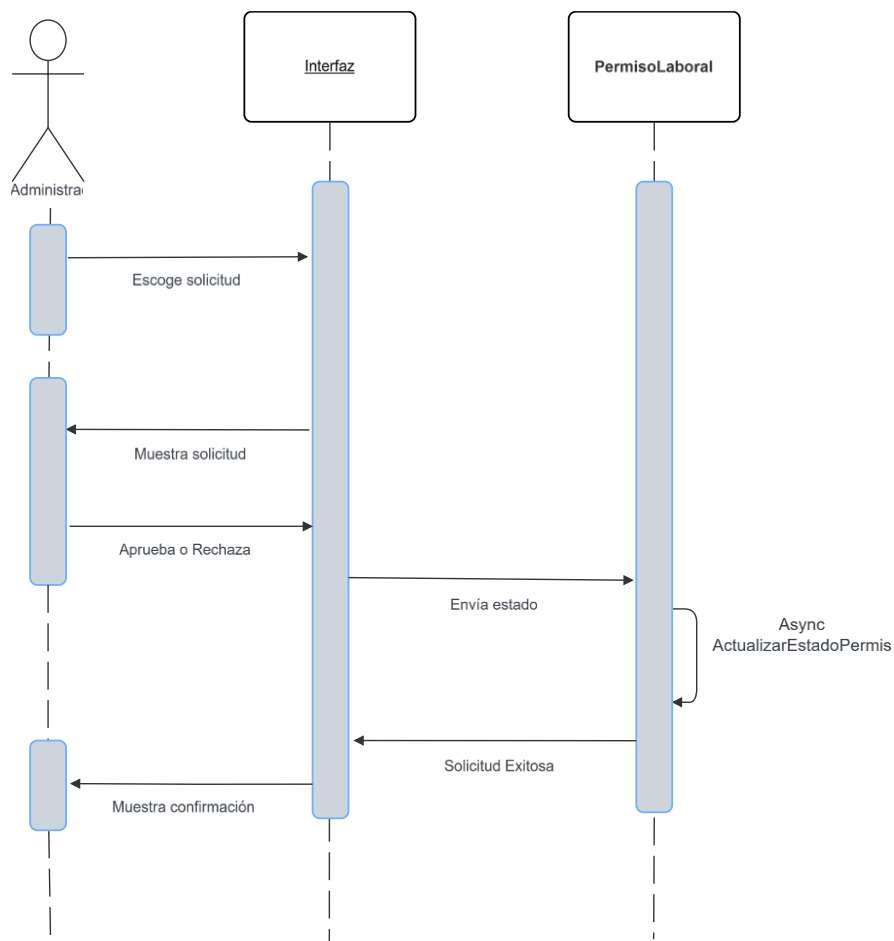


Fuente: Tolentino, 2026.

El diagrama de la Figura 31 muestra el proceso de registro de una solicitud de permiso laboral dentro del sistema. El flujo inicia cuando el usuario ingresa los datos de la solicitud en la interfaz. Posteriormente, la interfaz envía la información al módulo de permisos laborales para procesarla y crear la solicitud. Si los datos ingresados son inválidos, el sistema devuelve un error que se muestra al usuario. En caso contrario, la solicitud se registra correctamente y el sistema devuelve una confirmación indicando que el permiso fue creado exitosamente. Este diagrama permite visualizar cómo interactúan el usuario, la interfaz y el módulo de permisos durante este proceso.

Figura 32

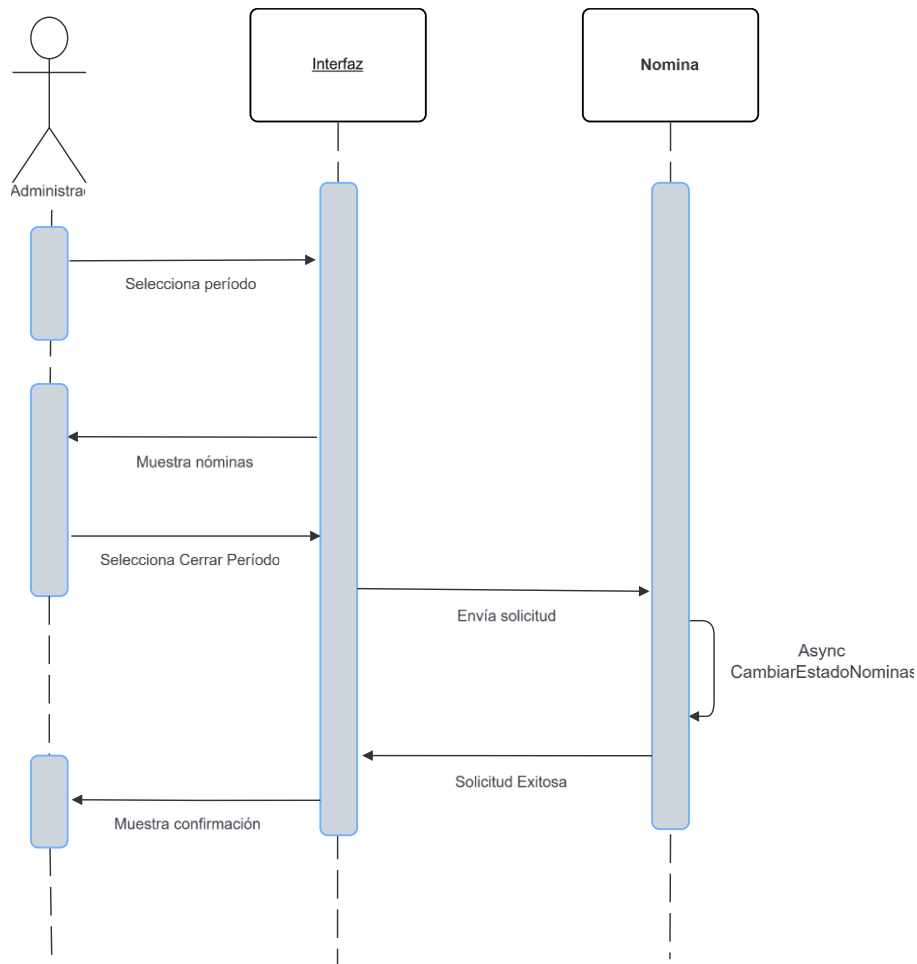
Diagrama de secuencia para aprobar o rechazar solicitudes



Fuente: Tolentino, 2026.

La Figura 32 muestra el proceso de gestión de solicitudes de permisos laborales dentro del sistema. El flujo inicia cuando el empleado selecciona una solicitud desde la interfaz del sistema, la cual es mostrada para su revisión. Posteriormente, el usuario puede decidir aprobar o rechazar la solicitud, enviando el nuevo estado al módulo de permisos laborales. El sistema procesa la actualización de manera asíncrona y, una vez completada, devuelve una confirmación indicando que la solicitud fue actualizada correctamente. Finalmente, la interfaz muestra al usuario la confirmación de la operación realizada.

Diagrama de secuencia para el cierre de nómina

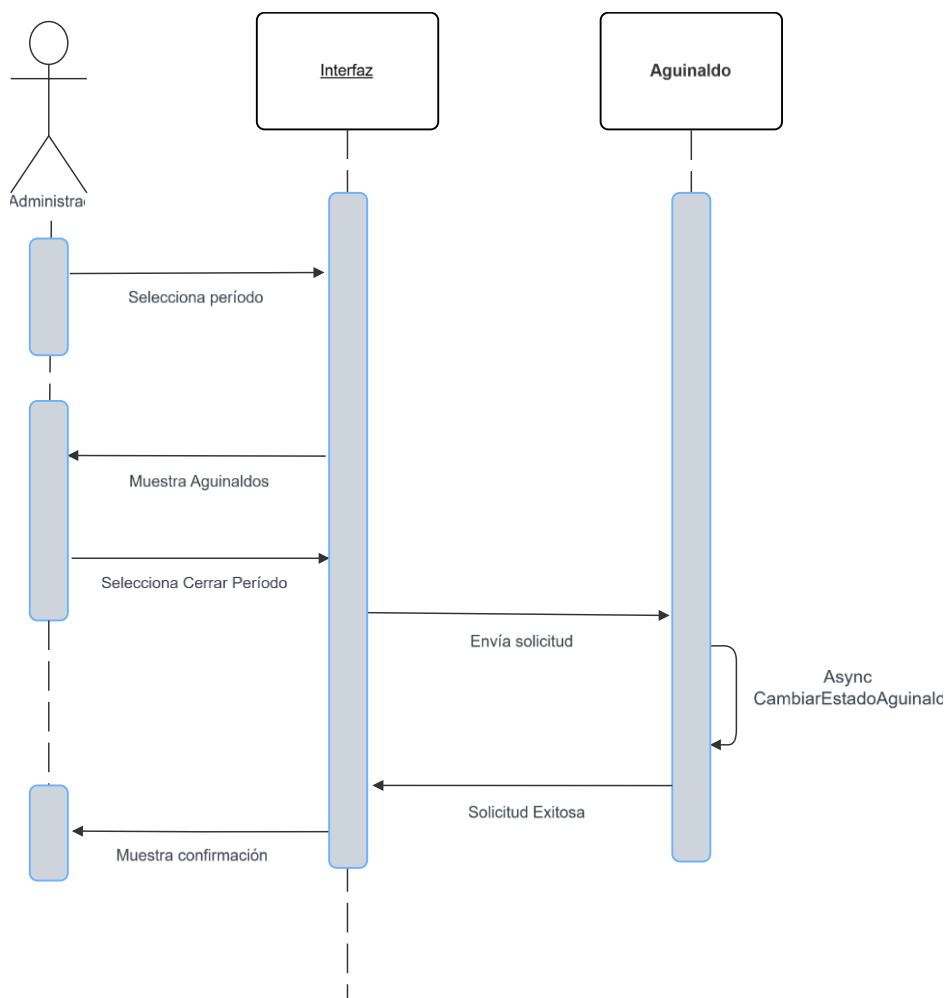


Fuente: Tolentino, 2026.

En la figura 33 se muestra el proceso de cierre de un período de nómina dentro del sistema. El flujo inicia cuando el usuario selecciona un período desde la interfaz, lo que permite visualizar las nóminas asociadas a dicho período. Posteriormente, el usuario selecciona la opción de cerrar el período, enviando la solicitud al módulo de nómina. El sistema procesa la solicitud de forma asincrónica para actualizar el estado de las nóminas y, una vez completado el proceso, devuelve una confirmación que es mostrada al usuario en la interfaz.

Figura 34

Diagrama de secuencia para cierre de aguinaldo

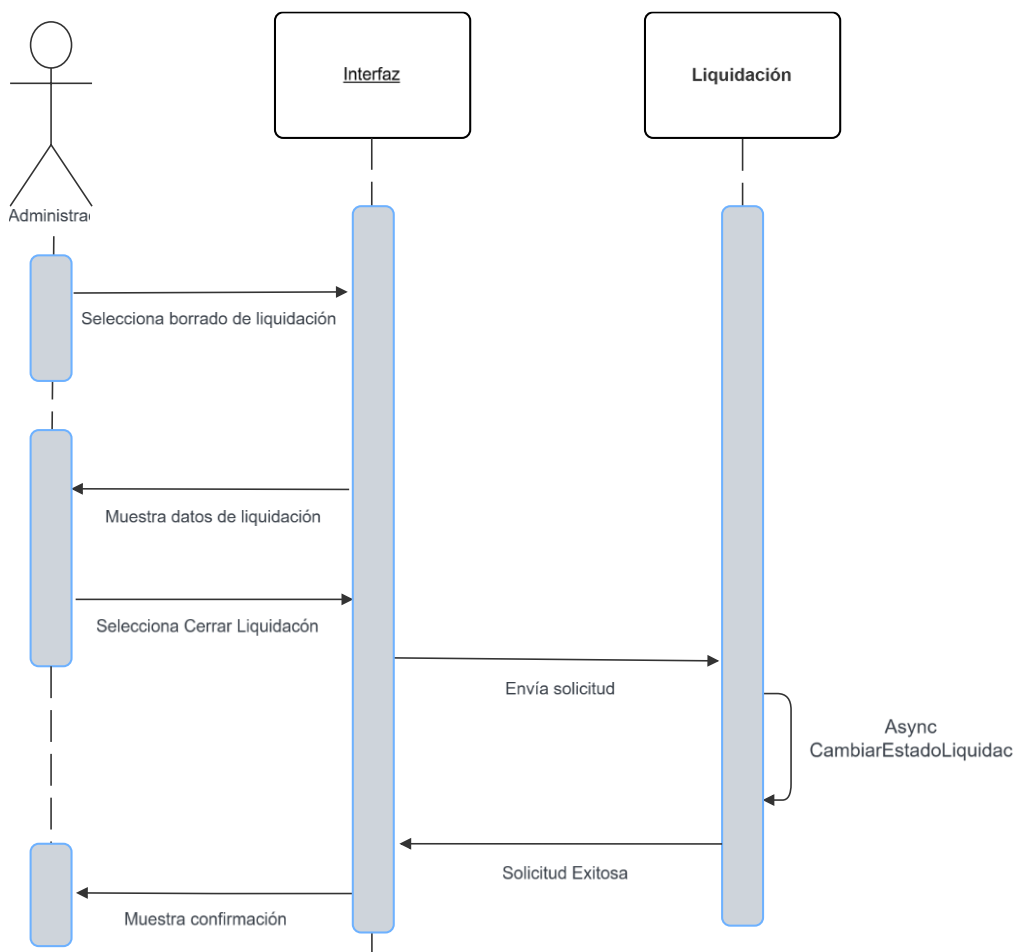


Fuente: Tolentino, 2026.

El diagrama de secuencia de la figura 34 muestra el proceso de gestión y cierre de un período de aguinaldo dentro del sistema. El flujo inicia cuando el usuario selecciona un período desde la interfaz, lo que permite visualizar los aguinaldos generados para ese período. Posteriormente, el usuario puede seleccionar la opción de cerrar el período, enviando la solicitud al módulo de aguinaldo. El sistema procesa la actualización de forma asincrónica para cambiar el estado del período y, una vez completado el proceso, devuelve una confirmación que es mostrada al usuario en la interfaz.

Figura 35

Diagrama de secuencia para cierre de liquidación



Fuente: Tolentino, 2026.

La Figura 35 muestra el proceso de cierre de una liquidación de un empleado dentro del sistema. El flujo inicia cuando el administrador selecciona una liquidación desde la interfaz, lo que permite visualizar los datos correspondientes. Posteriormente, el administrador puede seleccionar la opción de cerrar la liquidación, enviando la solicitud al módulo de liquidaciones. El sistema procesa la actualización de forma asincrónica para cambiar el estado de la liquidación y, una vez finalizado el proceso, devuelve una confirmación que es mostrada al usuario en la interfaz.

Programación

En esta sección se presentan fragmentos representativos del código fuente utilizados en el desarrollo del prototipo del sistema. Estos ejemplos permiten ilustrar la forma en que se han implementado algunas de las funcionalidades principales descritas en los distintos módulos del sistema. Además, muestran la estructura general de programación utilizada para organizar la lógica y el manejo de la información dentro de la aplicación.

Entradas y Salidas de datos

En este apartado se presentan algunas figuras correspondientes a fragmentos del código fuente que permiten observar de forma general el manejo de la entrada y salida de datos dentro del sistema. Estas estructuras muestran cómo la aplicación recibe información desde la interfaz de usuario, la procesa mediante la lógica del sistema y posteriormente devuelve los resultados correspondientes.

Figura 36

Entrada de datos para insertar empleado

```
static async CrearEmpleado (req, res) {
  const validated = validateEmpleado(req.body);
  if (!validated.success) { ...
  }
  const { cedula, nombre, apellido1, apellido2, fechaNacimiento, nacionalidad } = validated.data;
  try {
    const result = await EmpleadoModel.CrearEmpleado({ ...
    });
    return !result.success
      ? res.status(400).json({ status: 'error', message: result.message })
      : res.status(200).json({
        status: 'success',
        message: result.message || 'Empleado creado correctamente',
        idEmpleado: result.idEmpleado,
        empleado: result.empleado,
        detalle: result.detalle,
        direccion: result.direccion,
        telefono: result.telefono,
        correo: result.correo,
        usuario: result.usuario
      });
  } catch (error) {
    console.log(error.message);
    return res.status(500).json({
      status: 'error',
      message: 'Error interno del servidor',
      error: error.message
    });
  }
}
```

Fuente: Tolentino, 2026.

El fragmento de código de la figura 36 muestra el método CrearEmpleado del controlador, encargado de gestionar el registro de un nuevo empleado en el sistema. El controlador recibe la información enviada desde la solicitud y la envía al modelo para realizar el proceso de creación en la base de datos y posteriormente devuelve una respuesta en formato JSON indicando si la operación fue exitosa o si ocurrió algún error durante el proceso.

Figura 37

Entrada de datos para insertar turno

```
static async asignarTurno (req, res) {  
  const { idEmpleado, idTipoTurno, horaAlmuerzo, horaCafe, fecha } = req.body;  
  
  try {  
    const result = await HorarioModel.AsignarTurno({  
      idEmpleado, idTipoTurno, horaAlmuerzo, horaCafe, fecha  
    });  
  
    return !result.success  
      ? res.status(400).json({ status: 'error', message: result.message })  
      : res.status(200).json({ status: 'success', message: result.message, turno: result.turno });  
  } catch (error) {  
    return res.status(500).json({ success: false, message: error.message || 'Error al asignar el turno' });  
  }  
}
```

Fuente: Tolentino, 2026.

La figura 37 muestra el método asignarTurno del controlador, encargado de gestionar la asignación de turnos a los empleados. El método recibe desde la solicitud los datos necesarios, como el identificador del empleado, el tipo de turno, las horas de almuerzo y café, y la fecha correspondiente. Posteriormente, esta información se envía al modelo, que se encarga de registrar la asignación del turno en la base de datos. Finalmente, el sistema devuelve una respuesta indicando si la operación se realizó correctamente o si ocurrió algún error durante el proceso.

Salida de datos para obtener nóminas

```
static async ObtenerNominas (req, res) {
  try {
    const { idEmpleado, idPeriodo } = req.query;

    if (!idPeriodo) {
      return res.status(400).json({
        status: 'error',
        message: 'idPeriodo es obligatorio'
      });
    }

    const nominas = await NominaModel.ObtenerNominas({
      idEmpleado: idEmpleado ? Number(idEmpleado) : null,
      idPeriodo: Number(idPeriodo)
    });

    return res.status(200).json({
      status: 'success',
      message: 'Nóminas obtenidas correctamente',
      nominas
    });
  } catch (error) {
    console.error('Error en ObtenerNominas:', error);

    return res.status(500).json({
      status: 'error',
      message: error.message
    });
  }
}
```

Fuente: Tolentino, 2026.

El fragmento de la figura 38 muestra el método `ObtenerNóminas`, encargado de consultar las nóminas registradas en el sistema. El método recibe desde la solicitud los parámetros `idEmpleado` y `idPeriodo`, los cuales permiten filtrar las nóminas que se desean obtener. Posteriormente, se realiza una llamada al modelo para recuperar la información correspondiente desde la base de datos. Finalmente, el sistema devuelve una respuesta en formato JSON con la lista de nóminas obtenidas o un mensaje de error en caso de que ocurra algún problema durante el proceso.

Figura 39

Salida de datos para obtener turnos asignados

```
static async obtenerTurnosAsignados (req, res) {
  try {
    const result = await HorarioModel.ObtenerTurnosAsignados();

    return res.status(200).json({
      status: 'success',
      message: result.length
        ? 'Turnos obtenidos correctamente'
        : 'No hay turnos asignados',
      turnosAsignados: result
    });
  } catch (error) {
    return res.status(500).json({ success: false, message: error.message || 'Error al obtener los turnos asignados' });
  }
}
```

Fuente: Tolentino, 2026.

El código de la figura 39 muestra el método `obtenerTurnosAsignados` del controlador, encargado de consultar los turnos que han sido asignados a los empleados en el sistema. El método realiza una llamada al modelo para obtener la información almacenada en la base de datos. Posteriormente, devuelve una respuesta en formato JSON indicando si se encontraron turnos registrados o si no existen turnos asignados. En caso de presentarse algún error durante el proceso, el sistema retorna una respuesta indicando que ocurrió un error al obtener la información.

Procesos

En este apartado se describen los procesos principales que realiza el sistema, es decir, las operaciones o acciones que permiten gestionar la información dentro de la aplicación. Estos procesos representan las actividades que ejecuta el sistema para cumplir con las funcionalidades definidas. Para ilustrar su funcionamiento, se presentan fragmentos representativos del código fuente que muestran la lógica utilizada para procesar los datos y ejecutar estas operaciones.

Figura 40

Proceso de evaluar incidencias

```
static async EvaluarIncidencia ({ idEmpleado, idTipoMarca, hora, turno, marcasDelDia }) {
  const ATRASO = 1;
  const EXCESO_ALMUERZO = 2;
  const SALIDA_ANTICIPADA = 3;
  const EXCESO_DESCANSO = 4;

  const upsert = async ({ idTipoIncidencia, minutos, observacion }) => {...
  };

  if (idTipoMarca === 1) {
    const minutos = toMin(hora) - toMin(turno.hora_inicio);
    await upsert({
      idTipoIncidencia: ATRASO,
      minutos,
      observacion: `Marcó entrada ${minutos} min tarde`
    });
    return;
  }

  if (idTipoMarca === 3) {
    const ini = marcasDelDia.iniAlmuerzo;
    const fin = marcasDelDia.finAlmuerzo || hora;
    if (ini && fin) {
      const dur = toMin(fin) - toMin(ini);
      const exceso = dur - 60;
      await upsert({
        idTipoIncidencia: EXCESO_ALMUERZO,
        minutos: exceso,
        observacion: `Excedió almuerzo por ${exceso} min`
      });
    }
    return;
  }

  if (idTipoMarca === 5) {
```

Fuente: Tolentino, 2026.

La figura 40 corresponde a un proceso que se ejecuta automáticamente cuando se registra una marca de asistencia en el sistema. La función `EvaluarIncidencia` analiza la marca registrada por el empleado y la compara con los horarios establecidos en su turno de trabajo para determinar si existe alguna incidencia, como atraso en la entrada, exceso de tiempo en el almuerzo, salida anticipada o exceso en el descanso. A partir de esta evaluación, el sistema calcula los minutos correspondientes a la incidencia detectada y registra la información en la base de datos junto con una observación descriptiva. De esta manera, el sistema permite llevar un control automatizado de las irregularidades en la jornada laboral de los empleados.

Figura 41

Proceso de calcular vacaciones pendientes al liquidar

```

static async CalcularVacacionesPendientes ({
  const [diasRows] = await executor.query(
    \
    SELECT (DATEDIFF(?, ?) + 1) AS dias_trabajados;
    \
    [fechaSalida, fechaIngreso]
  );

  const diasTrabajados = Number(diasRows?.[0]?.dias_trabajados ?? 0);
  const diasAcumulados = Number((diasTrabajados / 30).toFixed(6));
  const [usadosRows] = await executor.query(
    \
    SELECT COALESCE(SUM(v.dias_aplicados), 0) AS dias_usados
    FROM Vacaciones v
    WHERE v.id_empleado = ?
      AND v.estado_solicitud IN ('APROBADA', 'FINALIZADA')
      AND v.fecha_inicio ≤ ?;
    \
    [idEmpleado, fechaSalida]
  );

  const diasUsados = Number(usadosRows?.[0]?.dias_usados ?? 0);

  const diasPendientes = Number(
    Math.max(0, diasAcumulados - diasUsados).toFixed(6)
  );

  const montoVacaciones = Number(
    (diasPendientes * salarioPromedioDiario).toFixed(2)
  );

  return { ...
  };
}

```

Fuente: Tolentino, 2026.

El código de la figura 41 corresponde a un proceso que se ejecuta durante el cálculo de la liquidación de un empleado. La función `CalcularVacacionesPendientes` determina la cantidad de días de vacaciones que el empleado tiene acumulados y que aún no ha utilizado al momento de finalizar la relación laboral. Para ello, el sistema calcula primero los días trabajados entre la fecha de ingreso y la fecha de salida y consulta en la base de datos los días de vacaciones que ya han sido utilizados por el empleado. Finalmente, el sistema calcula los días de vacaciones pendientes y el monto económico correspondiente que debe incluirse dentro de la liquidación.

Figura 42

Proceso de calcular horas extra al generar nómina

```

static async calcularHorasExtra ({
  const [rows] = await executor.query(
    `
    SELECT he.horas_total
    FROM HorasExtras he
    WHERE he.id_empleado = ?
      AND DATE(he.fecha) BETWEEN DATE(?) AND DATE(?)
      AND he.hora_extra_estado = 'APROBADA';
    `
    , [idEmpleado, fechaInicio, fechaFin]
  );

  const totalHorasExtra = rows.reduce(
    (acc, r) => acc + Number(r.horas ?? 0),
    0
  );

  if (totalHorasExtra <= 0) {
    return { tipoDetalle: 'INGRESO', monto: 0 };
  }

  const tarifaHora = salarioBaseMensual / 30 / 8;
  const factorHE = 1.5;
  const montoHorasExtra = Number((totalHorasExtra * tarifaHora * factorHE).toFixed(2));

  await NominaModel.InsertarNominaDetalle({
    idNomina,
    tipoDetalle: 'INGRESO',
    categoriaDetalle: 'HORAS EXTRA',
    cantidad: totalHorasExtra,
    unidad: 'HORA',
    monto: montoHorasExtra,
    conn
  });

  return { tipoDetalle: 'INGRESO', monto: montoHorasExtra, totalHorasExtra };
}

```

Fuente: Tolentino, 2026.

En la figura 42 se muestra un proceso que se ejecuta durante el cálculo de la nómina de un empleado. La función `CalcularHorasExtra` consulta en la base de datos las horas extra aprobadas que el empleado ha trabajado dentro del periodo de nómina seleccionado. Posteriormente, el sistema suma el total de horas registradas y calcula el monto correspondiente utilizando la tarifa por hora derivada del salario base mensual, aplicando el factor establecido para el pago de horas extra. Finalmente, el sistema registra este monto como un ingreso dentro del detalle de la nómina, permitiendo que las horas extra formen parte del cálculo final del salario del empleado.

Módulos Señalados

Aquí se presentan las principales funciones de los módulos que conforman el sistema, los cuales agrupan funcionalidades relacionadas dentro de una misma área de gestión. Cada módulo está diseñado para administrar un conjunto específico de procesos dentro del sistema de recursos humanos, permitiendo organizar las operaciones de forma estructurada.

Figura 43

Función del módulo controlar asistencia y tiempos de marca

```
static async InsertarMarca (req, res) {
  const { PIN, hora } = req.body;

  try {
    const idEmpleado = await AsistenciaController.ObtenerEmpleadoPorPIN({ PIN });

    const turno = await AsistenciaController.ObtenerTurnoDelDia({ idEmpleado });

    const marcasAntes = await AsistenciaModel.ObtenerMarcasDelDia({ idEmpleado });

    if (marcasAntes.ultimaHora) { ...
    }

    const mapaMarcas = { ...
    };

    const idTipoMarca = mapaMarcas[marcasAntes.total];

    if (!idTipoMarca) { ...
    }

    const result = await AsistenciaModel.InsertarMarca({ idEmpleado, idTipoMarca, hora }); ...
    const marcasDespues = await AsistenciaModel.ObtenerMarcasDelDia({ idEmpleado });

    await AsistenciaController.EvaluarHoraExtra({ ...
    });

    await AsistenciaController.EvaluarIncidencia({ ...
    });

    return res.status(201).json({ ...
    });
  } catch (error) { ...
  }
}
```

Fuente: Tolentino, 2026.

El fragmento de código de la figura 43 pertenece al módulo de controlar asistencia y tiempos de marca. El método `InsertarMarca` identifica al empleado correspondiente y obtiene el turno asignado para ese día. Posteriormente, el sistema consulta las marcas ya registradas para determinar el tipo de marca que corresponde (por ejemplo, entrada, inicio de almuerzo o salida).

Una vez registrada la nueva marca en la base de datos, el sistema ejecuta procesos adicionales para evaluar posibles horas extra o incidencias en la jornada laboral, manteniendo así un control automatizado de la asistencia del empleado.

Figura 44

Función del módulo gestionar horarios

```
static async AsignarTurno ({ idEmpleado, idTipoTurno, horaAlmuerzo, horaCafe, fecha }) {
  try {
    const almuerzo = (horaAlmuerzo === '' || horaAlmuerzo === undefined) ? null : horaAlmuerzo;
    const cafe = (horaCafe === '' || horaCafe === undefined) ? null : horaCafe;

    await pool.query(
      `
      INSERT INTO TurnoAsignado (id_empleado, id_tipo_turno, hora_almuerzo, hora_cafe, fecha)
      VALUES (?, ?, ?, ?, ?)
      ON DUPLICATE KEY UPDATE
        id_tipo_turno = VALUES(id_tipo_turno),
        hora_almuerzo = VALUES(hora_almuerzo),
        hora_cafe = VALUES(hora_cafe);
      `
      , [idEmpleado, idTipoTurno, almuerzo, cafe, fecha]
    );

    const [turnoAsignado] = await pool.query(
      `
      SELECT ta.id_turno_asignado, ta.fecha, ta.hora_almuerzo, ta.hora_cafe,
             t.id_tipo_turno, t.tipo_turno_descripcion, t.hora_inicio, t.hora_fin, t.total_horas
      FROM TurnoAsignado ta
      LEFT JOIN TipoTurno t ON ta.id_tipo_turno = t.id_tipo_turno
      WHERE ta.id_empleado = ? AND ta.fecha = ?
      LIMIT 1;
      `
      , [idEmpleado, fecha]
    );

    return turnoAsignado.length === 0
      ? { success: false, message: 'Error al asignar/modificar el turno' }
      : { success: true, message: 'Turno guardado correctamente', turno: turnoAsignado[0] };
  } catch (error) {
    throw new Error(`Error al asignar/modificar el turno: ${error.message}`);
  }
}
```

Fuente: Tolentino, 2026.

Este fragmento de código pertenece al módulo de gestión de horarios, encargado de administrar la asignación de turnos a los empleados. La función `AsignarTurno` recibe la información del empleado, el tipo de turno, los horarios de descanso y la fecha correspondiente. Posteriormente, el sistema registra o actualiza la asignación del turno en la base de datos y finalmente, el sistema consulta el turno asignado y devuelve la información correspondiente, confirmando que el turno fue guardado correctamente.

Figura 45

Función del módulo gestionar evaluaciones de empleado

```
static async RegistrarEvaluacion ({ idEmpleado, calificacionServicio, calificacionAtencion, comentarios }) {
  const [result] = await pool.query(
    `
    INSERT INTO Evaluacion_Empleado
      (id_empleado, calificacion_servicio, calificacion_atencion, comentarios)
    VALUES (?, ?, ?, ?)
    `
    ,
    [idEmpleado, calificacionServicio, calificacionAtencion, comentarios]
  );

  return { success: true, id_evaluacion: result.insertId };
}

static async ObtenerEvaluaciones () {
  const [rows] = await pool.query(`
  SELECT
    ee.id_evaluacion,
    ee.fecha,
    CONCAT(e.nombre, ' ', e.primer_apellido, ' ', e.segundo_apellido) AS empleado,
    ee.calificacion_servicio,
    ee.calificacion_atencion,
    ee.comentarios
  FROM Evaluacion_Empleado ee
  JOIN Empleado e ON e.id_empleado = ee.id_empleado
  ORDER BY ee.fecha DESC;
  `);
  return rows;
}
```

Fuente: Tolentino, 2026.

Este fragmento de código pertenece al módulo de gestión de evaluaciones de empleados, encargado de registrar y consultar las evaluaciones realizadas por los clientes sobre el desempeño del personal. La función RegistrarEvaluacion inserta en la base de datos la calificación otorgada al empleado en aspectos como servicio y atención, junto con los comentarios del cliente. Por otra parte, la función ObtenerEvaluaciones permite recuperar las evaluaciones registradas, mostrando información como la fecha, el nombre del empleado evaluado, las calificaciones obtenidas y los comentarios asociados. Esto permite al administrador analizar el desempeño del personal y mejorar la calidad del servicio.

Figura 46*Función del módulo gestionar permisos*

```

static async InsertarSolicitudPermisos ({
  fechaInicio,
  fechaFin,
  diasSolicitados,
  permisoDetalle,
  estadoSolicitud = 'PENDIENTE'
}) {
  try {
    const [result] = await pool.query(
      `
      INSERT INTO PermisoLaboral
      (id_empleado,
      id_tipo_permiso,
      fecha_solicitud,
      fecha_inicio,
      fecha_fin,
      dias_solicitados,
      dias_aplicados,
      permiso_detalle,
      estado_permiso)
      VALUES (?, ?, CURDATE(), ?, ?, ?, 0, ?, ?);
      `
      ,
      [
        idEmpleado,
        idTipoPermiso,
        fechaInicio,
        fechaFin,
        diasSolicitados,
        permisoDetalle,
        estadoSolicitud
      ]
    );

    return result.affectedRows;
  } catch (error) {
    throw new Error(`Error al crear la solicitud de permiso (BD): ${error.message}`);
  }
}

```

Fuente: Tolentino, 2026.

El código que se visualiza en la figura 46 pertenece al módulo de gestión de permisos laborales, encargado de registrar las solicitudes de permisos realizadas por los empleados. La función `InsertarSolicitudPermisos` almacena en la base de datos la información relacionada con la solicitud, como el empleado que realiza el permiso, el tipo de permiso solicitado, las fechas de inicio y fin, la cantidad de días solicitados y un detalle o justificación del permiso. Además, el sistema registra automáticamente la fecha de la solicitud y asigna un estado inicial de pendiente, permitiendo que posteriormente el administrador pueda revisar y actualizar el estado de la solicitud.

Figura 47

Función del módulo gestionar liquidación

```
static async GenerarLiquidacion ({
  const salarioPend = await LiquidacionModel.CalcularSalarioPendiente({...
  });
  await LiquidacionModel.InsertarDetalleLiquidacion({...
  });
  const vac = await LiquidacionModel.CalcularVacacionesPendientes({...
  });
  await LiquidacionModel.InsertarDetalleLiquidacion({...
  });
  const ag = await LiquidacionModel.CalcularAguinaldoProporcional({
    idEmpleado,
    fechaSalida,
    conn
  });
  await LiquidacionModel.InsertarDetalleLiquidacion({...
  });
  if (tipoCese.pagaCesantia) {...
  }
  if (tipoCese.pagaPreaviso) {...
  }
  const totales = await LiquidacionModel.ActualizarTotalLiquidacion({...
  });
  await conn.commit();
  return {
    idLiquidacion: idLiqFinal,
    idEmpleado,
    idTipoCese,
```

Fuente: Tolentino, 2026.

La figura 47 muestra código del módulo de gestión de liquidaciones, encargado de calcular el monto correspondiente cuando finaliza la relación laboral de un empleado. La función `GenerarLiquidacion` ejecuta diferentes cálculos necesarios para determinar el pago final, incluyendo el salario pendiente, las vacaciones no disfrutadas y el aguinaldo proporcional. Además, dependiendo del tipo de cese, el sistema puede incluir conceptos adicionales como cesantía o preaviso.

Figura 48

Función del módulo calcular horas extras

```
static async InsertarHorasExtra ({ idEmpleado, horaInicio, horaFin, horasTotal, observacion = null, estado = null }) {
  try {
    const [result] = await pool.query(
      `INSERT INTO horasextras
      (id_empleado, fecha, hora_inicio, hora_fin, horas_total, hora_extra_estado, observacion)
      VALUES (?, CURRENT_DATE, ?, ?, ?, ?, ?);`,
      [idEmpleado, horaInicio, horaFin, horasTotal, estado, observacion]
    );

    if (result.affectedRows === 0) {
      return { success: false, message: 'No se pudo registrar la hora extra' };
    }

    return {
      success: true,
      message: 'Horas extra registradas correctamente',
      idHorasExtra: result.insertId
    };
  } catch (error) {
    throw new Error('Error al insertar horas extra: ' + error.message);
  }
}
```

Fuente: Tolentino, 2026.

El código de la figura 48 forma parte del módulo de gestión de horas extra, encargado de registrar las horas adicionales trabajadas por los empleados. La función `InsertarHorasExtra` almacena en la base de datos la información relacionada con las horas extra realizadas, incluyendo el empleado correspondiente, la hora de inicio y fin, la cantidad total de horas trabajadas y una observación opcional. Además, el sistema registra automáticamente la fecha en que se realizó la hora extra y su estado dentro del proceso de aprobación, permitiendo posteriormente que estas horas puedan ser consideradas durante el cálculo de la nómina.

Figura 49

Función del módulo calcular vacaciones

```

static async InsertarSolicitudVacaciones ({
  idEmpleado,
  fechaInicio,
  fechaFin,
  diasSolicitados,
  estadoSolicitud = 'PENDIENTE',
  solicitudDetalle = null
}) {
  try {
    const [result] = await pool.query(
      `
      INSERT INTO Vacaciones
      (id_empleado, fecha_solicitud, fecha_inicio, fecha_fin, dias_aplicados,
      dias_solicitados, estado_solicitud, solicitud_detalle, motivo_estado)
      VALUES (?, CURDATE(), ?, ?, 0, ?, ?, ?, NULL);
      `
      , [idEmpleado, fechaInicio, fechaFin, diasSolicitados, estadoSolicitud, solicitudDetalle]
    );

    return result.affectedRows;
  } catch (error) {
    throw new Error(`Error al crear la solicitud de vacaciones (BD): ${error.message}`);
  }
}

```

Fuente: Tolentino, 2026.

La figura 49 muestra código que pertenece al módulo de calcular vacaciones, encargado de registrar las solicitudes de vacaciones realizadas por los empleados. La función `InsertarSolicitudVacaciones` almacena en la base de datos la información de la solicitud, incluyendo el empleado que realiza la petición, las fechas de inicio y fin, la cantidad de días solicitados y un detalle opcional de la solicitud. Además, el sistema registra automáticamente la fecha en que se realiza la solicitud y establece su estado inicial como pendiente, permitiendo que posteriormente el administrador pueda revisar y aprobar o rechazar la solicitud.

Figura 50

Función del módulo gestionar incapacidades

```

static async InsertarIncapacidad ({
  idEmpleado, idTipoIncapacidad, fechaInicio, fechaFin, diasIncapacidad, incapacidadDetalle = null, estadoInc
}) {
  try {
    const [result] = await pool.query(
      `
      INSERT INTO Incapacidad
      (id_empleado,
      fecha_inicio,
      fecha_fin,
      dias_incapacidad,
      dias_aplicados,
      id_tipo_incapacidad,
      dias_cubiertos_por_patrono,
      incapacidad_detalle,
      estado_incapacidad,
      archivo_url)
      VALUES (?, ?, ?, ?, ?, 0, ?, NULL, ?, ?, ?);
      `
      ,
      [
        idEmpleado,
        fechaInicio,
        fechaFin,
        diasIncapacidad,
        idTipoIncapacidad,
        incapacidadDetalle,
        estadoIncapacidad,
        archivoURL
      ]
    );
    return result.affectedRows;
  } catch (error) {
    throw new Error(`Error al insertar incapacidad (BD): ${error.message}`);
  }
}

```

Fuente: Tolentino, 2026.

En la figura 50 se muestra un fragmento de código que pertenece al módulo de gestión de incapacidades, encargado de registrar las incapacidades médicas de los empleados dentro del sistema. La función `InsertarIncapacidad` almacena en la base de datos la información relacionada con la incapacidad, incluyendo el empleado afectado, el tipo de incapacidad, las fechas de inicio y fin, la cantidad de días correspondientes y un detalle descriptivo. Además, permite asociar un archivo o documento de respaldo y establece el estado inicial de la incapacidad, lo que facilita posteriormente su control y seguimiento dentro del sistema.

Figura 51

Función del módulo gestionar planilla y nóminas

```
static async GenerarNomina ({
  try {
    await conn.beginTransaction();

    const salarioBaseMensual = await EmpleadoModel.ObtenerSalarioBaseMensual({...});

    const { idNomina } = await NominaModel.InsertarNomina({...});

    const totales = { totalIngresos: 0, totalEgresos: 0 };

    // Salario ordinario
    const rOrdinario = await NominaModel.calcularSalarioOrdinario({...});
    NominaModel.AcumularTotales(totales, rOrdinario);

    // Horas extra
    const rHorasExtra = await NominaModel.calcularHorasExtra({...});
    NominaModel.AcumularTotales(totales, rHorasExtra);

    // Vacaciones
    const rVacaciones = await NominaModel.calcularVacaciones({...});
    NominaModel.AcumularTotales(totales, rVacaciones);

    // Permisos
    const rPermisos = await NominaModel.calcularPermisos({...});
    totales.totalIngresos += Number(rPermisos.montoIngreso ?? 0);
    totales.totalEgresos += Number(rPermisos.montoEgreso ?? 0);

    // Incapacidades
    const rIncap = await NominaModel.calcularIncapacidades({...});
    totales.totalIngresos += Number(rIncap.montoIngreso ?? 0);
    totales.totalEgresos += Number(rIncap.montoEgreso ?? 0);
  }
}
```

Fuente: Tolentino, 2026.

El código que se muestra en la figura 51 pertenece al módulo de gestión de planillas y nóminas, encargado de generar la nómina correspondiente a un empleado dentro de un periodo determinado. La función `GenerarNomina` inicia una transacción en la base de datos y obtiene el salario base del empleado para posteriormente calcular los distintos componentes que conforman la nómina, como el salario ordinario, las horas extra, las vacaciones, los permisos laborales y las incapacidades. A medida que se calculan estos elementos, el sistema va acumulando los ingresos y egresos correspondientes para determinar el total final de la nómina del empleado. De esta forma, el sistema automatiza el proceso de cálculo salarial y asegura que todos los conceptos relacionados con la remuneración sean considerados.

Validaciones

En esta sección se describen las validaciones implementadas en el prototipo con el fin de garantizar la integridad de la información y prevenir errores durante el uso del sistema. Las validaciones permiten controlar que los datos ingresados cumplan con el formato y las reglas definidas, evitando registros incompletos o inconsistentes en la base de datos.

Figura 52

Función validadora para insertar solicitud de vacaciones

```
export const insertarVacacionesSchema = (saldoDisponible) =>
  fecha_fin: z
    .regex(DATE_YYYY_MM_DD, { message: 'Formato inválido. Use YYYY-MM-DD' }),

  dias_solicitados: z.coerce
    .number()
    .int()
    .min(1, { message: 'Debe ser al menos 1 día' }),

  solicitud_detalle: z
    .string()
    .trim()
    .max(250, { message: 'La descripción no puede exceder 250 caracteres' })
    .optional()

}).superRefine((data, ctx) => {
  const inicio = new Date(data.fecha_inicio);
  const fin = new Date(data.fecha_fin);

  if (fin < inicio) {
    ctx.addIssue({
      path: ['fecha_fin'],
      message: 'La fecha de fin debe ser posterior o igual a la fecha de inicio',
      code: z.ZodIssueCode.custom
    });
  }

  if (
    typeof saldoDisponible === 'number' &&
    data.dias_solicitados > saldoDisponible
  ) {
    ctx.addIssue({
      path: ['dias_solicitados'],
      message: `No puede solicitar más de ${saldoDisponible} días disponibles`,
      code: z.ZodIssueCode.custom
    });
  }
});
```

Fuente: Tolentino, 2026.

Este fragmento de código de la figura 52 corresponde a una validación del formulario de solicitud de vacaciones. El esquema verifica que el formato de las fechas sea correcto, que los días solicitados sean válidos y que la descripción no exceda el límite permitido. Además, valida que la fecha de fin no sea anterior a la fecha de inicio y que los días solicitados no superen el saldo de vacaciones disponible del empleado.

Figura 53

Función validadora para insertar empleado

```
export const empleadoSchema = z.object({
  segundo_apellido: z
  },

  cedula: z
    .string()
    .nonempty({ message: 'La cédula es obligatorio' })
    .min(9, { message: 'Deben ser 9 dígitos' })
    .max(9, { message: 'Deben ser 9 dígitos' });
    .refine((val) => /^[0-9,]+$/.test(val), {
      message: 'Solo se permiten números'
    }),

  fecha_nacimiento: z
    .string()
    .nonempty({ message: 'La fecha de nacimiento es obligatoria' })
    .regex(
      DATE_YYYY_MM_DD,
      { message: 'Formato inválido. Use YYYY-MM-DD' }
    )
    .refine(esMayorDeEdad, { message: 'El empleado debe ser mayor de 18 años' }),

  nacionalidad: z
    .string()
    .nonempty({ message: 'La nacionalidad es obligatoria' })
    .max(40, { message: 'La nacionalidad no puede exceder 40 caracteres' }),

  id_genero: z.coerce
    .number()
    .int()
    .positive({ message: 'Debe seleccionar un género' }),

  id_estado_civil: z.coerce
    .number()
    .int()
    .positive({ message: 'Debe seleccionar un estado civil' }),

  cantidad_hijos: z.preprocess(
    (v) => (v === '' || v === null ? undefined : v),
    z.union([
      z
      .coerce
    ]
  )
);
```

Fuente: Tolentino, 2026.

Finalmente, la figura 53 corresponde a la validación del formulario de registro de empleados. El esquema verifica que los datos ingresados, como cédula, fecha de nacimiento, nacionalidad, género y estado civil, cumplan con los formatos y reglas establecidas antes de ser registrados en el sistema.

Pruebas

En esta sección se describen las pruebas realizadas al sistema con el objetivo de verificar su correcto funcionamiento y asegurar que los diferentes módulos desarrollados cumplen con los requisitos definidos durante el análisis y diseño. A través de estas pruebas se evaluó el comportamiento del sistema ante distintos escenarios de uso, permitiendo identificar posibles errores, validar la lógica implementada y comprobar que las funcionalidades responden adecuadamente a las acciones del usuario. De esta manera, se garantiza que el sistema sea confiable, estable y adecuado para su implementación en el entorno de trabajo.

El caso de prueba “Iniciar Sesión” tiene como objetivo verificar que el proceso de inicio de sesión del sistema funcione correctamente. En la prueba se evalúan diferentes situaciones que pueden ocurrir cuando un usuario intenta ingresar al sistema. Primero, se comprueba que el sistema permita el acceso cuando se ingresan credenciales válidas, redirigiendo al usuario al panel principal según su rol. Además, se valida que el sistema detecte y muestre mensajes de error adecuados cuando se ingresa una contraseña incorrecta, cuando los campos se dejan vacíos o cuando se intenta iniciar sesión con una cuenta desactivada. De esta manera, el caso de prueba permite confirmar que el sistema valida correctamente las credenciales, controla los errores y protege el acceso al sistema.

Tabla 64

Caso de prueba Iniciar Sesión

Propuesta de un sistema web para la gestión de recursos humanos en la empresa restaurante Tumi´s House, ubicada en Tibás, San José.			
Código	CP-01	Nombre del caso de prueba:	Iniciar sesión
Fecha de elaboración:	10/07/2025	Autor:	Ayran Tolentino
Precondiciones:	El usuario debe estar previamente registrado en el sistema y tener su cuenta activa.	Prioridad:	Alta

Descripción:	Este caso de prueba verifica la funcionalidad del módulo de autenticación de usuarios para asegurar que el inicio de sesión se realice correctamente con credenciales válidas y que el sistema responda adecuadamente en caso de errores.		
Detalles de Resultados			
Datos de entrada	Resultado esperado	Resultado Actual	Resultado actual
1. Ingresar nombre de usuario válido y contraseña correcta.	El sistema valida las credenciales y redirige al panel principal según el rol.	Resultado esperado obtenida.	Prueba exitosa
2. Ingresar nombre de usuario válido y contraseña incorrecta.	El sistema muestra un mensaje de error indicando "Contraseña o nombre de usuario incorrecto".	Resultado esperado obtenida.	Prueba exitosa
3. Dejar los campos de usuario y contraseña en blanco y hacer clic en "Ingresar".	El sistema muestra un mensaje de advertencia indicando que ambos campos son obligatorios.	Resultado esperado obtenida.	Prueba exitosa
4. Ingresar nombre de usuario y contraseña de una cuenta de usuario desactivada (estado = inactivo).	El sistema muestra un mensaje de error indicando "Contraseña o nombre de usuario incorrecto".	Resultado esperado obtenida.	Prueba exitosa

Fuente: Tolentino, 2026.

El caso de prueba “Registrar marcas de asistencia” verifica que el registro de marcas de asistencia funcione correctamente. Se comprueba que el sistema registre las marcas dentro del horario asignado, detecte cuando existen horas extra, y evite registros duplicados en un corto periodo de tiempo. Además, valida que las diferencias de horario se clasifiquen como incidencias para su posterior revisión.

Tabla 65

Caso de prueba Registrar marcas de asistencia

Propuesta de un sistema web para la gestión de recursos humanos en la empresa restaurante Tumi´s House, ubicada en Tibás, San José.			
Código	CP-02	Nombre del caso de prueba:	Registrar marcas de asistencia
Fecha de elaboración:	10/07/2025	Autor:	Ayran Tolentino
Precondiciones:	El empleado debe estar activo en el sistema y debe tener un turno asignado para ese día.	Prioridad:	Alta
Descripción:	Este caso de prueba valida el correcto funcionamiento del módulo de registro de marcas de asistencia, asegurando que se almacenen correctamente. El sistema detecta si hay horas extra, calcula incidencias automáticamente, y clasifica cualquier marca fuera del rango del turno o incidencia como "Pendiente" para posteriormente hacer su revisión.		
Detalles de Resultados			
Datos de entrada	Resultado esperado	Resultado Actual	Resultado actual
1. Empleado realiza las marcas dentro del horario asignado y cumpliendo reglas del negocio.	El sistema registra todas las marcas y no detecta incidencias ni horas extra.	Resultado esperado obtenida.	Prueba exitosa
2. Empleado marca salida 45 minutos después de su hora de salida.	El sistema registra la diferencia como hora extra y la deja en estado "pendiente".	Resultado esperado obtenida.	Prueba exitosa
3. Empleado marca salida 45 minutos después de su hora de salida.	El sistema registra la diferencia como hora extra y la deja en estado "Pendiente"	Resultado esperado obtenida.	Prueba exitosa
4. El empleado marca entrada a las 09:00 y vuelve a intentarlo a las 09:05.	El sistema bloquea el intento (excepción con la marca de ingreso del café) y muestra mensaje: "Ya registró"	Resultado esperado obtenida.	Prueba exitosa

	una marca hace x minutos. Debe esperar al menos 10 minutos para registrar una nueva marca."		
--	---	--	--

Fuente: Tolentino, 2026.

El siguiente caso de prueba “Generación de nómina” verifica que el proceso de generación de nómina se realice correctamente. Se valida que el sistema calcule el salario quincenal considerando el salario base, horas extra, vacaciones, incapacidades y permisos, además de aplicar correctamente las deducciones legales y rebajos por incidencias como ausencias.

Tabla 66

Caso de prueba Generación de nómina

Propuesta de un sistema web para la gestión de recursos humanos en la empresa restaurante Tumi´s House, ubicada en Tibás, San José.			
Código	CP-03	Nombre del caso de prueba:	Generación de nómina
Fecha de elaboración:	10/07/2025	Autor:	Ayran Tolentino
Precondiciones:	El empleado debe tener marcas, vacaciones o permisos registrados durante la quincena, así como salario base asignado.	Prioridad:	Alta
Descripción:	Este caso de prueba valida que el sistema calcule correctamente la nómina de un empleado tomando en cuenta salario base, horas trabajadas, vacaciones, incapacidades, permisos, horas extra, y aplique deducciones legales (CCSS, renta) y personalizadas.		

Detalles de Resultados			
Datos de entrada	Resultado esperado	Resultado Actual	Resultado actual
1. Empleado con salario base ₡600,000 mensuales, con 8 horas extra aprobadas en la quincena	El sistema calcula el salario ordinario quincenal (₡300,000) + monto por horas extra. Deducciones legales aplicadas correctamente.	Resultado esperado obtenida.	Prueba exitosa
2. Empleado con 2 días de vacaciones disfrutadas.	El sistema suma monto proporcional por vacaciones al total de ingresos.	Resultado esperado obtenida.	Prueba exitosa
3. Empleado con 4 día de incapacidad registrada (INCAPACIDAD CCSS).	El sistema suma el monto de incapacidad por parte del patrono (3 días al 50% por enfermedad común) al total de ingresos.	Resultado esperado obtenida.	Prueba exitosa
4. Empleado con permiso sin goce.	El sistema resta el monto equivalente del total devengado, como deducción adicional.	Resultado esperado obtenida.	Prueba exitosa
5. Empleado sin marcas en un día laboral. Incidencia de Ausencia registrada.	El sistema descuenta las horas no trabajadas.	Resultado esperado obtenida.	Prueba exitosa

Fuente: Tolentino, 2026.

El siguiente caso de prueba verifica que el registro y la validación de permisos laborales funcionen correctamente. Se comprueba que el sistema permita solicitar permisos, gestione su aprobación desde el panel administrativo y aplique validaciones para evitar fechas pasadas, duplicidad de permisos o cruces con vacaciones ya registradas.

Tabla 67

Caso de prueba Registro y validación de permisos laborales

Propuesta de un sistema web para la gestión de recursos humanos en la empresa restaurante Tumi´s House, ubicada en Tibás, San José.			
Código	CP-04	Nombre del caso de prueba:	Registro y validación de permisos laborales
Fecha de elaboración:	10/07/2025	Autor:	Ayran Tolentino
Precondiciones:	El empleado debe estar activo en el sistema y tener acceso al formulario de solicitud de permisos. Debe existir al menos un tipo de permiso registrado.	Prioridad:	Alta
Descripción:	Este caso de prueba valida que el sistema permita registrar solicitudes de permisos laborales, diferencie entre permisos con y sin goce salarial, y que no permita duplicidad de fechas ni cruce con incapacidades o vacaciones.		
Detalles de Resultados			
Datos de entrada	Resultado esperado	Resultado Actual	Resultado actual
1. Solicitud de permiso sin goce para un día laboral.	El sistema registra la solicitud con estado "PENDIENTE" y lo refleja en el panel de revisión del administrador.	Resultado esperado obtenida.	Prueba exitosa
2. Aprobar un permiso desde el panel administrativo.	El sistema actualiza el estado a "APROBADO" y se refleja en la vista del empleado.	Resultado esperado obtenida.	Prueba exitosa

3. Solicitud de permiso para un día ya pasado.	El sistema bloquea las fechas pasadas desde la fecha actual para no permitir el ingreso de una fecha antigua.	Resultado esperado obtenida.	Prueba exitosa
4. Solicitar un permiso que se cruza con un periodo de vacaciones ya registrado.	El sistema bloquea la acción y muestra: “Ya existe un periodo de vacaciones registradas en la fecha escogida.”	Resultado esperado obtenida.	Prueba exitosa
5. Registrar dos permisos distintos en el mismo día. Y uno de ellos ya fue aprobado.	El sistema bloquea la segunda solicitud con mensaje: “Ya existe un permiso aprobado para esta fecha.”	Resultado esperado obtenida.	Prueba exitosa

Fuente: Tolentino, 2026.

Finalmente el caso de prueba siguiente verifica que el proceso de generación de liquidación laboral se realice correctamente cuando finaliza la relación laboral de un empleado. Se valida que el sistema calcule los montos correspondientes, como aguinaldo proporcional, vacaciones pendientes y otros pagos, además de aplicar o no preaviso y cesantía según el tipo de cese registrado.

Tabla 68

Caso de prueba Generación de liquidación

Propuesta de un sistema web para la gestión de recursos humanos en la empresa restaurante Tumi´s House, ubicada en Tibás, San José.			
Código	CP-05	Nombre del caso de prueba:	Generación de liquidación
Fecha de elaboración:	10/07/2025	Autor:	Ayran Tolentino
Precondiciones:		Prioridad:	Alta

	El empleado debe tener fecha de ingreso registrada, salario base, cantidad de días pendientes y saldo vacaciones pendientes		
Descripción:	Este caso de prueba verifica que el sistema realice correctamente el cálculo de la liquidación laboral al finalizar la relación de trabajo, considerando el tipo de cese, el aguinaldo proporcional, vacaciones no disfrutadas, y la aplicación o no de preaviso y cesantía según el caso.		
Detalles de Resultados			
Datos de entrada	Resultado esperado	Resultado Actual	Resultado actual
1. Empleado despedido sin responsabilidad patronal (sin cesantía ni preaviso), con salario ¢500,000, 60 días laborados del último periodo y 5 días de vacaciones pendientes.	El sistema calcula y muestra: aguinaldo proporcional, monto por vacaciones, y genera resumen con total bruto de liquidación.	Resultado esperado obtenida.	Prueba exitosa
2. Empleado renuncia de forma voluntaria.	El sistema no incluye preaviso ni cesantía, solo vacaciones y aguinaldo proporcional.	Resultado esperado obtenida.	Prueba exitosa
3. Se genera y guarda la liquidación de un empleado.	El sistema cambia el estado del empleado a inactivo y lo excluye de futuras nóminas, horarios y marcas.	Resultado esperado obtenida.	Prueba exitosa
4. Solicitar un permiso que se cruza con un periodo de vacaciones ya registrado.	El sistema bloquea la acción y muestra: “Ya existe un periodo de vacaciones registradas en la fecha escogida.”	Resultado esperado obtenida.	Prueba exitosa

5. Registrar dos permisos distintos en el mismo día. Y uno de ellos ya fue aprobado.	El sistema bloquea la segunda solicitud con mensaje: "Ya existe un permiso aprobado para esta fecha."	Resultado esperado obtenida.	Prueba exitosa
--	---	------------------------------	----------------

Fuente: Tolentino, 2026.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Luego de la realización del presente trabajo, se obtuvieron resultados que permitieron cumplir con los objetivos planteados en el proyecto. En relación con el primer objetivo específico, orientado al análisis de los requerimientos del sistema, se logró identificar las principales necesidades. Durante esta etapa se analizaron los procesos existentes dentro de la empresa, lo que permitió detectar diversas dificultades asociadas a la gestión manual de la información. Entre estas se identificaron problemas en el control de asistencias, el registro de jornadas laborales, cálculos de la nómina, etc. Este análisis evidenció que la gestión tradicional de estos procesos puede generar errores en los registros, duplicidad de tareas y un uso considerable de tiempo por parte del personal administrativo.

Con respecto al segundo objetivo, relacionado con el diseño de la estructura del sistema, se desarrolló una arquitectura que integra la base de datos, la estructura del backend y la organización de los componentes del frontend. Este proceso permitió establecer una base tecnológica sólida para el funcionamiento del sistema, definiendo la forma en que se gestionan los datos, las relaciones entre los distintos módulos y la interacción entre los usuarios y la aplicación. De esta manera se puede gestionar la información de forma organizada, segura y eficiente.

En cuanto al tercer objetivo específico, correspondiente al desarrollo del prototipo funcional del sistema, se logró implementar una aplicación web que integra diferentes funcionalidades orientadas a la gestión del recurso humano. Entre estas se incluyen el control de asistencias, la asignación de jornadas laborales, el registro de solicitudes de vacaciones y permisos, así como el cálculo automatizado de la nómina. La implementación de estas funcionalidades permite optimizar diversos procesos administrativos dentro de la empresa, reduciendo la carga de trabajo manual y mejorando la precisión en el manejo de la información.

Asimismo, en relación con el cuarto objetivo específico, se realizaron pruebas funcionales, de usabilidad y de seguridad utilizando datos de prueba. Estas pruebas permitieron verificar el correcto funcionamiento de los diferentes módulos del sistema y evaluar la interacción de los usuarios con la aplicación. Los resultados obtenidos permitieron confirmar que el sistema cumple con los requerimientos establecidos y que es capaz de responder adecuadamente a las necesidades.

Finalmente, a partir de los resultados obtenidos durante el desarrollo del proyecto, se concluye que se logró cumplir con el objetivo general de desarrollar un prototipo funcional de un sistema web para la gestión de recursos humanos en la empresa Restaurante Tumi's House. El sistema propuesto contribuye a mejorar la eficiencia administrativa, facilitar el manejo de la información y apoyar el cumplimiento de las obligaciones laborales dentro del contexto empresarial costarricense.

Asimismo, este proyecto reafirma la importancia del uso de la informática en las organizaciones, demostrando que la tecnología, cuando se aplica de manera adecuada, puede optimizar los procesos internos y generar un impacto positivo en la gestión del personal. En particular, este tipo de soluciones representa un aporte significativo para pequeñas y medianas empresas (PYMES), ya que les permite profesionalizar su gestión, optimizar recursos y fortalecer el cumplimiento normativo sin incurrir en altos costos operativos.

Recomendaciones

Se recomienda realizar la implementación del sistema dentro del restaurante, con el fin de centralizar la información relacionada con los empleados, la asistencia, los horarios, la nómina y otros procesos administrativos del área de recursos humanos. Debido a que la empresa no cuenta con un departamento de tecnologías de información, la instalación, configuración inicial y puesta en funcionamiento del sistema podrían realizarse con el apoyo del desarrollador del sistema o un programador en computación con conocimientos en desarrollo web, quien sería el perfil idóneo para llevar a cabo la implementación técnica del sistema y asegurar su correcto funcionamiento dentro de la infraestructura tecnológica existente.

El proceso de implementación y estabilización del sistema podría desarrollarse en un periodo aproximado de una semana, considerando la instalación del sistema, la configuración inicial y la capacitación básica del personal encargado de su uso. Según el Ministerio de Trabajo (2026), el salario diario mínimo de un programador en computación sin título es de aproximadamente ₡16 244,50 colones, por lo que el costo estimado de implementación podría situarse aproximadamente en ₡81 222 colones.

Otra recomendación para futuras mejoras del sistema es la integración con dispositivos de marcaje de asistencia, tales como relojes biométricos. Esta integración permitiría automatizar el registro de entradas y salidas de los empleados, reduciendo errores humanos y mejorando la precisión de los datos utilizados para el control de asistencia y el cálculo de la nómina.

La implementación de esta mejora podría ser gestionada por la administración del restaurante en conjunto con un desarrollador, quien sería responsable de realizar la configuración del dispositivo y la integración con la base de datos del sistema. El tiempo estimado para la instalación, configuración e integración del dispositivo podría ser de aproximadamente dos semanas, considerando las pruebas necesarias para garantizar el correcto funcionamiento del sistema.

En cuanto al costo de implementación, un ejemplo de dispositivo compatible es el ZKTeco K20 de la empresa ZKTeco, el cual permite administrar los registros de asistencia a través de su interfaz de red TCP/IP, facilitando la comunicación con sistemas web. El dispositivo biométrico tiene un precio aproximado en el mercado entre ₡40 000 y ₡70 000 colones, dependiendo del proveedor. (ZKTeco, s.f.). Como se mencionó anteriormente, el salario diario mínimo de un programador en computación sin título es de aproximadamente ₡16 244,50 colones, por lo que el costo asociado a la integración técnica podría estimarse dentro del rango de esas dos semanas.

Además, se recomienda realizar la publicación del sistema en un entorno de computación en la nube, con el objetivo de que la plataforma pueda ser accesible desde cualquier ubicación con conexión a internet y no únicamente desde una red local. De esta manera, se facilitaría el acceso para el personal administrativo y otros usuarios autorizados, permitiendo gestionar la información del sistema de recursos humanos de forma más flexible y eficiente.

La implementación de esta mejora podría ser realizada por el desarrollador del sistema o por un programador en computación con conocimientos en desarrollo web, quien sería el perfil idóneo para configurar el servidor, la base de datos y las medidas básicas de seguridad necesarias para la publicación del sistema.

Debido a que se trata de una pequeña empresa, es posible utilizar proveedores de alojamiento con planes gratuitos, los cuales permiten desplegar aplicaciones web sin incurrir en costos iniciales. Entre las opciones más utilizadas se encuentran Render, Railway y Vercel, los cuales ofrecen planes gratuitos que permiten alojar aplicaciones web, bases de datos o servicios backend con ciertas limitaciones, suficientes para proyectos de pequeña escala.

El proceso de despliegue y configuración del sistema en la nube podría realizarse en un periodo aproximado de dos a cinco días, considerando la configuración del servidor, la base de datos y las pruebas básicas de funcionamiento en línea. En cuanto al costo de implementación, este estaría asociado principalmente al trabajo del profesional encargado del despliegue. Nuevamente, el salario diario de referencia de un programador sin título es de ₡16 244,50 colones, por lo que el costo estimado del proceso podría oscilar entre ₡32 489 y ₡81 222 colones, dependiendo del tiempo requerido para completar la publicación del sistema.

Finalmente, la recolección continua de retroalimentación por parte de los usuarios será clave para futuras mejoras del sistema. Para ello, se sugiere que la administración del restaurante promueva la recopilación de comentarios por parte del personal que utilice la plataforma, mediante encuestas breves o espacios de retroalimentación. Esta información permitiría identificar posibles mejoras en la interfaz, funcionalidades adicionales o ajustes necesarios para optimizar el uso del sistema. La retroalimentación no solo debería realizarse una vez implementado el sistema, sino mantenerse como una práctica continua que contribuya al proceso de mejora y evolución del sistema a lo largo del tiempo.

REFERENCIAS

- Alcoba Zenteno, R. A. (2023). Tamaño de la muestra: alternativas de selección. Universidad Autónoma Juan Misael Saracho, Departamento de Investigación, Ciencia y Tecnología. <https://dicyt.uajms.edu.bo/revistas/index.php/investigacion-y-desarrollo/article/view/1636/1600>
- Armijos Ortega, L. M., Vélez Macas, C. A., & Lojan Cueva, E. L. (2024). Estudio de la adopción de metodologías ágiles en proyectos de desarrollo de software en la región 7 del Ecuador. *Revista Espacios*, 45(4), Art. 6. <https://revistaespacios.com/a24v45n04/a24v45n04p06.pdf>
- Arroyo, J. (2022). Las variables como elemento sustancial en el método científico. *Revista Educación*, 46(1), 1–25. <https://doi.org/10.15517/revedu.v46i1.45609>
- Asamblea Legislativa de Costa Rica. (2011). Ley N°8968: Ley de Protección de la Persona frente al Tratamiento de sus Datos Personales. <https://www.pgrweb.go.cr/scij>
- BGA. (2024). Pago de Horas Extras (en Costa Rica). BG&A Abogados Corporativos. <https://bgacorp.com/pago-horas-extras-costa-rica/>
- Carbo-Velez, C. R., Carbo-Velez, R. A., Yunga-Solorzano, D. J., & Villamar-Monserrate, C. A. (2021). Diseño de prototipos de software. Fundamentos epistémico-metodológicos para su elaboración. *Dominios de las Ciencias*, 7(6), 1520-1531. <https://doi.org/10.23857/dc.v7i6.2407>
- Certus. (2021). Descubre qué son las pruebas de software y su importancia. Certus. <https://www.certus.edu.pe/blog/pruebas-de-software-importancia/#:~:text=Una%20prueba%20de%20software%20es,cliente%20un%20producto%20de%20calidad.>
- Creswell, J. W. & Creswell, J. D. (2023). *Research Design. Qualitative, Quantitative, and Mixed Methods Approaches*. <https://repositorio.ciem.ucr.ac.cr/items/cec131b5-1934-4fc7-81cf-bc413895502e>

- Evaluación de la calidad de software: Pruebas y marcos de referencia. (s/f). Com.mx. Recuperado el 6 de marzo de 2025, de <https://www.testingit.com.mx/blog/evaluacion-de-la-calidad-de-software>
- Expressjs.com. (2025). Express - Infraestructura de aplicaciones web Node.js. Expressjs.com. <https://expressjs.com/es>
- Ferreira, A. C. (2025). Qué es un framework, para qué sirve y ejemplos. InboundCycle. <https://www.inboundcycle.com/diccionario-marketing-online/framework>
- Galarza Alay, J. I., & Carriel Sevillano, R. P. (2022). Evolución de los sistemas de lenguaje de programación a lo largo de la historia. E-IDEA Journal of Engineering Sciences, 4(10), 15-26. <https://doi.org/10.53734/esci.vol4.id237>
- Hernandez, R. D. (2021). El modelo de arquitectura Modelo-Vista-Controlador (MVC). FreeCodeCamp. <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>
- Manatal (2025). ¿Qué es el sistema de gestión de recursos humanos? <https://www.manatal.com/es/glossary/human-resources-management-system-hrms>
- MDN Web Docs (2024). Getting started with React. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started
- Ministerio de Trabajo y Seguridad Social (2024). Código de Trabajo de Costa Rica. https://www.mtss.go.cr/elministerio/marco-legal/documentos/Codigo_Trabajo_RPL.pdf
- Ministerio de Trabajo y Seguridad Social de Costa Rica. (2026). Lista de salarios https://www.mtss.go.cr/temas-laborales/salarios/lista_salarios_minimos_2026.pdf
- Mógica. (2024). Los 10 principios de usabilidad de Nielsen en webs y aplicaciones. Uup.com <https://www.uup.es/blog/post/10-principios-de-usabilidad/>

- MongoDB. (s. f.). What is a tech stack and how do they work? MongoDB.
<https://www.mongodb.com/resources/basics/technology-stack>
- Oracle. (2025). ¿Qué es MySQL? Oracle.com. <https://www.oracle.com/ar/mysql/what-is-mysql/>
- Oracle. (2025). ¿Qué es una base de datos? Oracle.com. <https://www.oracle.com/cr/database/what-is-database/>
- Rodríguez, C. (2021) Tipos de Fuentes de información. Pontificia Universidad Católica de Puerto Rico. <https://pucpr.libguides.com/bibliotecamayaguez/fuentesinformacion>
- Rodríguez, C., Breña, J. L., & Esenarro, D. (2021). Las variables en la metodología de la investigación científica. Editorial Área de Innovación y Desarrollo, S.L. <https://doi.org/10.17993/IngyTec.2021.784o>
- Suárez, E. (2026). Fuentes de Información: qué son, tipos y ejemplos. Experto Universitario. <https://expertouniversitario.es/blog/fuentes-de-informacion/>
- Skill Vertex. (2024). MERN stack with MySQL. Skill Vertex. <https://skillvertex.com/blog/mern-stack-with-mysql/>
- Tamoshi Evaluación de la calidad de software: Pruebas y marcos de referencia. (s/f). Com.mx. <https://www.testingit.com.mx/blog/evaluacion-de-la-calidad-de-software>
- Vásquez Ramírez, A. A., Guanuchi Orellana, L. M., Cahuana Tapia, R. D., Vera Treves, R., & Holgado Tisoc, J. (2023). Métodos de investigación científica. Instituto Universitario de Innovación, Ciencia y Tecnología (INUDI PERÚ). <https://doi.org/10.35622/inudi.b.094>
- Visure Solutions. (2022). Requisitos funcionales y no funcionales (con ejemplos). Visure Solutions. <https://visuresolutions.com/es/requirements-management-traceability-guide/functional-vs-non-functional-requirements/>
- Zambrano Monserrate, M., Berrús Zhumi, A., & Goncalves Guillén, G. (2023). Principios de estadística. Universidad Espíritu Santo (UEES). <https://uees.edu.ec/descargas/libros/2023/principios-de-estadistica.pdf>
- ZKTeco. (s/f). K20. Zktecolatinoamerica.com. <https://zktecolatinoamerica.com/producto/k20/>

APÉNDICES

Cuestionario

En el marco de una investigación sobre la gestión de Recursos Humanos en el restaurante Tumi's House, le invitamos a completar este cuestionario. Su participación es fundamental para comprender cómo los procesos actuales de administración de personal afectan la operatividad y eficiencia en la empresa.

Este cuestionario es confidencial. Sus respuestas serán utilizadas únicamente con fines académicos y no serán compartidas con terceros. Completar el cuestionario tomará aproximadamente 10 minutos. Agradecemos su colaboración.

1. Tiempo trabajando en la empresa:
 - Menos de 6 meses
 - 6 meses - 1 año
 - 1 - 3 años
 - Más de 3 años
2. ¿Cómo calificaría el proceso actual de control de asistencia y horarios en el restaurante?
 - Muy eficiente
 - Eficiente
 - Regular
 - Deficiente
 - Muy deficiente
3. ¿Ha tenido problemas con el cálculo de horas extras trabajadas?
 - Frecuentemente
 - Algunas veces
 - Rara vez
 - Nunca
4. ¿Considera que el proceso de pago de nómina es claro y transparente?

- Totalmente de acuerdo
 - De acuerdo
 - Neutral
 - En desacuerdo
 - Totalmente en desacuerdo
5. ¿Cómo recibe actualmente la información sobre su salario y horas trabajadas?
- En papel (hoja de nómina física)
 - Comunicación verbal con el encargado
 - Mediante un documento digital
 - Otro
6. ¿Con qué frecuencia recibe retroalimentación sobre su desempeño laboral?
- Frecuentemente
 - Algunas veces
 - Rara vez
 - Nunca
7. En general, ¿qué tan satisfecho está con el sistema actual de gestión de recursos humanos?
- Muy satisfecho
 - Satisfecho
 - Poco satisfecho
 - Nada satisfecho
8. ¿Cree que el sistema actual impacta en su desempeño laboral?
- Totalmente de acuerdo
 - De acuerdo
 - Neutral
 - En desacuerdo
 - Totalmente en desacuerdo
9. ¿Cree que la implementación de un sistema digital ayudaría a mejorar la gestión de RRHH en la empresa?
- Totalmente de acuerdo

- De acuerdo
- Neutral
- En desacuerdo
- Totalmente en desacuerdo

10. ¿Estaría dispuesto a utilizar un sistema digital para gestionar su asistencia, nómina y horarios?

- Totalmente de acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Totalmente en desacuerdo

Guía de observación

No.	Aspectos por observar	Cumple	No cumple	Oportunidad de mejora	Detalle de observación
1	Manejo de documentación y archivos				
2	Existencia de criterios claros para la evaluación del desempeño				
3	Observar el Registro de asistencia de empleados				
4	Observar cómo se calculan y asignan las vacaciones				
5	Tiempo de respuesta y aprobación en solicitudes de permisos.				

6	Eficiencia en el registro y seguimiento de incapacidades y permisos.				
7	Detección oportuna de ausencias, retardos y faltas injustificadas				