

**UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS
ESCUELA DE INGENIERÍA INFORMÁTICA**

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

**PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN
LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ**

LIBER GABRIEL SALAS BARBOZA

**CARLOS HUMBERTO AGUILAR MORA
TUTOR**

Sede Central

Julio, 2024

DEDICATORIA

Culminar una etapa importante de mi vida, que comenzó con un sueño desde el colegio, porque sentir que uno puede crear algo desde cero y verlo funcionar fue un descubrimiento impresionante a mi edad, que me creó una sensación indescriptible y muy impactante.

Les agradezco a los profesores de la carrera de Ingeniería de Software y a la directora todo el esfuerzo y tiempo que han dispuesto, cómo nos han guiado desde el principio dándonos consejos importantes y siempre estando a disposición de nosotros.

AGRADECIMIENTOS

Les agradezco profundamente a todos mis compañeros durante estos años, porque me han dado una serie de experiencias buenas y malas, a través de las cuales he forjado un camino el cual no hubiera logrado, sin su ayuda.

Expreso profundamente agradecimientos al profesor Carlos Aguilar Mora, que me ha brindado experiencias y enseñanzas para completar de la mejor forma el desarrollo del proyecto y a la directora quien ha estado para nosotros en todo momento para guiarnos en el camino.

Contenido

DEDICATORIA	2
AGRADECIMIENTOS	3
Tablas	8
Ilustraciones	10
Resumen Ejecutivo	12
CAPÍTULO I: INTRODUCCIÓN	13
OBJETIVOS	13
Objetivo General	13
Objetivos Específicos	14
JUSTIFICACIÓN DEL PROYECTO	14
Viabilidad Técnica	14
Viabilidad Operativa	15
Viabilidad Económica	16
Viabilidad Legal	18
Proyecciones	18
Alcance Funcional	18
Alcance Metodológico	20
Alcance Tecnológico	20
CAPÍTULO II: MARCO REFERENCIAL	22
Ámbito de Recursos Humanos	22
Evaluación y Retroalimentación	23
Sistema de Remuneración	23
Control de Asistencia	24
Automatización de Procesos	25
Leyes Informáticas	26
Patentes de Software	27
Normativas ISO	28
Licencias de Software	28
Requerimientos Técnicos	29
Ciclo del Software	30
Las Bases de Datos	33

Modelado de Datos Relacionales	34
Modelado de Datos no Relacionales	34
Copias de Seguridad.....	35
Protección de Datos.....	36
Encriptación	37
Lenguajes de Programación.....	39
JavaScript	39
El Lenguaje C#.....	40
CAPITULO III: MARCO METODOLÓGICO.....	41
Enfoques de Investigación.....	41
Enfoque Cualitativo	41
Enfoque de Investigación Seleccionado	42
Tipos de Investigación.....	43
Investigación Descriptiva.....	44
Tipo de Investigación Seleccionado	45
Fuentes de Información.....	45
Variables	46
Variable Conceptual	46
Variable Operacional.....	46
Variable Instrumental	47
Instrumentos de Recolección de Datos.....	50
Observación	50
Entrevista	50
Proceso de Recolección y Análisis de Datos	51
CAPITULO IV: ANÁLISIS DE RESULTADOS.....	53
Aplicación de los Instrumentos de Recolección de Datos.....	53
Instrumento de Observación	53
Instrumento de Entrevista	54
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	56
Conclusiones.....	56
Recomendaciones	57
CAPÍTULO VI: PROPUESTA.....	59

Análisis	59
Análisis Detallado del Software por Desarrollar	59
Calcular Planilla	59
Calcular Vacaciones	59
Calcular Incapacidades.....	60
Calcular Horas Extras.....	60
Evaluar Empleados.....	60
Calcular Liquidación	60
Calcular Aguinaldo.....	61
Control Asistencia	61
Análisis Detallado del Hardware Requerido.....	61
Análisis Detallado de los Elementos Relacionados con las Telecomunicaciones	62
Descripción Detallada de las Herramientas Técnicas Utilizadas Para el Desarrollo	62
Descripción Detallada del Conocimiento Básico	64
Casos de Uso	64
Caso de Uso 1	65
Caso de Uso 2.....	68
Caso de Uso 3.....	70
Caso de Uso 4.....	72
Caso de Uso 5.....	73
Caso de Uso 6.....	75
Caso de Uso 7.....	76
Caso de Uso 8.....	78
Caso de uso 9.....	79
Caso de uso 10.....	81
Diseño.....	83
Arquitectura del Sistema.....	83
Arquitectura del Software	84
Diseño de Entradas	86
Registrar Asistencia.....	86
Crear Incapacidad	87
Registro de Usuario.....	88

Trámite de Permisos.....	89
Inicio de Sesión.....	90
Diseño Físico de la Base de Datos.....	90
Diccionario de Datos.....	91
Diseño de procesos.....	105
Diseño de salidas.....	113
Diagramas UML.....	116
Programación Entradas y Salidas.....	122
Procesos y Validaciones.....	127
Módulos alcanzados.....	132
Aguinaldo.....	132
Evaluación.....	132
Horas Extras.....	133
Incapacidad.....	133
Liquidación.....	134
Planilla.....	134
Control Asistencia.....	135
Vacaciones.....	136
Trámite de permisos.....	137
Pruebas.....	137
REFERENCIAS.....	142
APÉNDICES.....	145
Apéndice A: Guía de observación.....	145
Apéndice B: GUÍA DE ENTREVISTA.....	147

Tablas

Tabla 1.	17
Tabla 2.	48
Tabla 3.	61
Tabla 4.	63
Tabla 5.	63
Tabla 6.	67
Tabla 7.	68
Tabla 8.	71
Tabla 9.	72
Tabla 10.	74
Tabla 11.	75
Tabla 12.	77
Tabla 13.	78
Tabla 14.	80
Tabla 15.	81
Tabla 16.	92
Tabla 17.	92
Tabla 18.	93
Tabla 19.	93
Tabla 20.	94
Tabla 21.	95
Tabla 22.	95
Tabla 23.	96
Tabla 24.	96
Tabla 25.	97
Tabla 26.	97
Tabla 27.	98
Tabla 28.	98
Tabla 29.	98
Tabla 30.	99
Tabla 31.	99
Tabla 32.	99
Tabla 33.	99
Tabla 34.	100
Tabla 35.	101
Tabla 36.	101
Tabla 37.	102
Tabla 38.	102
Tabla 39.	102
Tabla 40.	103
Tabla 41.	103
Tabla 42.	104

Tabla 43.	104
Tabla 44.	137
Tabla 45.	138
Tabla 46.	139
Tabla 47.	140
Tabla 48.	140

Ilustraciones

Ilustración 1	31
Ilustración 2	65
Ilustración 3	84
Ilustración 4	85
Ilustración 5	86
Ilustración 6	87
Ilustración 7	88
Ilustración 8	89
Ilustración 9	90
Ilustración 10	91
Ilustración 11	105
Ilustración 12	106
Ilustración 13	107
Ilustración 14	108
Ilustración 15	109
Ilustración 16	110
Ilustración 17	111
Ilustración 18	112
Ilustración 19	113
Ilustración 20	114
Ilustración 21	114
Ilustración 22	115
Ilustración 23	115
Ilustración 24	116
Ilustración 25	117
Ilustración 26	117
Ilustración 27	118
Ilustración 28	118
Ilustración 29	119
Ilustración 30	119
Ilustración 31	120
Ilustración 32	120
Ilustración 33	121
Ilustración 34	121
Ilustración 35	122
Ilustración 36	122
Ilustración 37	123
Ilustración 38	123
Ilustración 39	124
Ilustración 40	125
Ilustración 41	126
Ilustración 42	127

Ilustración 43	127
Ilustración 44	128
Ilustración 45	129
Ilustración 46	129
Ilustración 47	129
Ilustración 48	130
Ilustración 49	130
Ilustración 50	131
Ilustración 51	132
Ilustración 52	132
Ilustración 53	133
Ilustración 54	133
Ilustración 55	134
Ilustración 56	134
Ilustración 57	135
Ilustración 58	136
Ilustración 59	137

Resumen Ejecutivo

En el entorno actual de la empresa la gestión de recursos humanos es parte fundamental para un ambiente laboral sano y transparente. El restaurante la Victoria ha enfrentado a varios problemas interno cuando aplican los procesos de la gestión de recursos humanos, que ha afectado anímicamente a los colaboradores, con respecto a los temas salariales y las vacaciones.

Para solucionar la problemática que se enfrenta la organización, se decidió realizar un prototipo funcional de los recursos humanos que cumpla con la ejecución correcta de todos los procesos que debe realizar recursos humanos y logren controlar la asistencia de cada uno de los colaboradores donde ellos efectúen consultas de los salarios con la información detallada para mantener una transparencia entre recursos humanos y empleados.

En conclusión, el informe detalla los procesos que el prototipo funcional va a ejecutar en función a las necesidades que se plantearon en la problemática. Cumpliendo cada una de ellas con bastante satisfacción permitiendo que la empresa logre ver resultados y beneficios una vez que implementen el sistema en el restaurante ya que, el *software* cuenta con funcionalidades completas y probadas para el uso adecuado de los colaboradores.

CAPÍTULO I: INTRODUCCIÓN

El restaurante “La Victoria” ha operado por más de cincuenta años, durante los cuales ha cambiado de administración. La última de ellas logró obtener el permiso para operar como restaurante, pues con anterioridad era solamente un bar. El local se ubica en el cantón de San José, en el nuevo barrio chino. Actualmente cuenta con siete colaboradores quienes se ocupan de preparar platillos típicos de cantina y servir diferentes tipos de bebidas con alcohol. Ellos se están enfrentando a muchos desafíos significativos en el momento de realizar los procesos de gestión en recursos humanos.

Entre los problemas que se identificaron durante la investigación se encuentran los siguientes:

- La cantidad de horas laboradas por cada empleado se registra en un cuaderno, este método genera que en el momento del pago no se contemplen todas las horas laboradas, o se le añadan más horas de las laboradas; de ahí que se generen, problemas en los pagos exactos de cada uno de los colaboradores.
- El cálculo de vacaciones, de igual forma, se registra en un cuaderno, por lo cual hay inexactitud respecto de si el trabajador ha cumplido con los días por ley para recibir la cantidad de días de vacaciones que se merece.
- Se desconocen los permisos de ausencias de los empleados, ya que estos solo se solicitan de palabra y no se mantiene ningún registro que corrobore la ausencia de uno o varios días del colaborador.

OBJETIVOS

Objetivo General

- Desarrollar un prototipo funcional de un sistema de gestión de recursos que optimice y automatice los procesos de administración de personal donde mejore la toma decisiones y promueva la transparencia.

Objetivos Específicos

- Analizar los requerimientos que solucionen la problemática planteada.
- Diseñar la arquitectura del prototipo, definiendo una estructura clara y robusta relacionada a la gestión de recursos humanos.
- Programar el sistema requerido con el fin de que resuelva la problemática planteada.
- Probar el funcionamiento correcto del prototipo funcional.

JUSTIFICACIÓN DEL PROYECTO

En la investigación previa, se analizó el caso de la organización, observando las dificultades al manejar pagos, incapacidades y vacaciones de los colaboradores.

Con esto la propuesta se enfoca en mejorar las debilidades del restaurante y brindarle beneficios inmediatos en el momento de la contratación del personal. Estos beneficios son los siguientes:

- Agilizar los pagos de cada uno de los colaboradores.
- Mejorar la asistencia de los colaboradores, ya que el propio sistema permitirá registrar la entrada y salida del trabajo de los empleados. A la vez, registrará las horas extras trabajadas por el colaborador, sin tener que anotarlo en ninguna otra parte.
- Gestionar permisos sin demora y con el tiempo suficiente para su aprobación.
- Tramitar incapacidades válidas y efectivas para los colaboradores.

Viabilidad Técnica

Para comprobar la viabilidad técnica se analizó la infraestructura del restaurante. Este cuenta con dispositivos que se pueden conectar a la misma red donde se podrán acceder al sistema, que se encontrará en un *software* interno en la computadora principal; cada colaborador podrá contar con acceso a ella en cualquier momento. También, se podrá acceder desde dispositivos móviles como celulares o tabletas que se encuentren dentro de la red interna del local.

En cuanto a licencias se usarán la de SQL Server Express 2022, que es gratuita y el *software* se publicará internamente en el IIS del computador principal, ya que con esto se minimizará los costos de alojamiento en la nube del sistema.

El *hardware* que tiene la organización es bastante pequeño, pues al tener pocos colaboradores y como su enfoque principal no es el uso de la tecnología, carece de equipos que ayuden al mejoramiento. Para que el prototipo funcional sea viable se necesita al menos la compra de dos computadores de gama media baja para que se vean los resultados esperados a la hora de la implementación, porque al contar con solo un computador, los empleados tendrán que turnarse para el uso del *software*. La organización debe comprar la tecnología para que el proyecto sea técnicamente viable, en caso contrario, se podría aplicar el plan B, consistente en dar acceso a los colaboradores desde sus celulares para que accedan de la forma más cómoda, mientras se adquieren los nuevos equipos.

En conclusión, si la organización se permite la adquisición de nuevos equipos tecnológicos, la implementación es cien por ciento viable, ya que cada colaborador podrá tener acceso al sistema en cualquier momento sin tener que esperar. En cambio, si se opta por no adquirir los equipos, se podrá realizar un plan B para que los usuarios se sientan cómodos con el nuevo *software*.

Viabilidad Operativa

Para el uso del prototipo funcional se deben tener en cuenta ciertos aspectos importantes para el desarrollo tales como:

- Los conocimientos de los empleados en la utilización de software.
- Qué tipo de personas usarán el sistema.
- Si el desarrollo tendrá algún impacto negativo en el personal.

Validando los aspectos anteriormente mencionados y con conocimiento de las habilidades requeridas por parte de a los usuarios que harán uso del prototipo funcional, se tiene claro que la implementación no provocará un impacto negativo en la reducción de personal. Aun así, se tiene claro que para la comprensión y uso correcto de la tecnología se necesitará una capacitación de personal para que ellos logren comprender correctamente las funcionalidades

que tendrá a su disposición. Cabe recalcar, que la capacitación para los colaboradores no está incluida dentro del proceso para la creación del proyecto de graduación.

Por otra parte, se desarrollará una guía de ayuda en cada pantalla para que el usuario pueda consultar cómo funciona el módulo. Adicionalmente, esto será de ayuda para futuros ingresos de personal, que no hayan tenido la capacitación correcta del sistema.

En cuanto a las tareas, se analizó a profundidad el impacto que habrá una vez el prototipo esté disponible. Se llega a la conclusión de que no habrá impacto negativo y los procesos de recursos humanos se agilizarán, por lo cual el beneficio es para ambas partes, pues el desarrollo evitará los posibles errores humanos que se presentaban en la empresa por tener los registros en manuales.

Viabilidad Económica

Se realizó una investigación exhaustiva para conocer el precio del mercado para el desarrollo de *software*, que se lleva por cantidad de horas requeridas para la elaboración de un módulo. Se toma como base la tabla del Ministerio de Trabajo y Seguridad Social, ente regulador de leyes para la justicia social, en este consta que la hora trabajada para un programador de cómputo es de ₡1916,625 colones (MTSS, 2024). Con este monto es posible calcular el costo por módulo requerido en el sistema. A continuación, se mencionan los módulos que contempla el sistema con el total de horas y su coste total:

Tabla 1.*Cotización de módulos*

Módulo	Cantidad de horas	Coste por hora	Coste total
Calcular planilla	40h	€1916,625	€76665
Calcular vacaciones	38h	€1916,625	€72831,75
Calcular incapacidades	25h	€1916,625	€47915,625
Calcular horas extras	25h	€1916,625	€47915,625
Evaluar empleados	18h	€1916,625	€34499,25
Calcular liquidación	42h	€1916,625	€80918,25
Tramitar permisos	28h	€1916,625	€53665,5
Calcular aguinaldos	50h	€1916,625	€95831,25
Mantenimientos	45h	€1916,625	€86248,125
Consultas	40h	€1916,625	€76665
Reportes	40h	€1916,625	€76665
Seguridad	40h	€1916,625	€76665
Control de asistencia	25h	€1916,625	€47915,625
Total	456h		€849901,75

Fuente: Elaboración Propia.

Cabe resaltar que al realizarse para un proyecto de graduación el coste de los €849901,75 colones por el desarrollo del proyecto van a resultar en un coste de 0 colones para la empresa.

Viabilidad Legal

Según el artículo 10 de la Ley n.º 8968 de protección de persona, dice que el sistema debe garantizar los datos personales evitando su alteración, destrucción accidental o pérdida de la persona. Con esta ley se establece la pauta que los datos registrados en la base de datos serán de carácter privado y no se usará la información de los colaboradores para ningún fin ilícito o venta de esta, prevaleciendo la integridad y privacidad de los datos personales. Por la Ley n.º 6683 de Derechos de Autor y Derechos Conexos, el total del desarrollo está ligado a la organización "La Victoria" y el uso de este será exclusivo para el restaurante, por lo cual la organización tendrá la potestad y exclusividad del sistema, con respecto a cualquier distribución o imitación del *software*.

Mencionado las leyes anteriores, se velará por la integridad de la información del personal y la protección de datos.

Proyecciones

Con los objetivos claros se espera que el sistema cumpla con todas las funcionalidades que requiere la empresa para el manejo correcto de planillas, vacaciones y demás. Estas funcionalidades deben de ir de acuerdo con una serie de leyes que resguarden la información de los colaboradores y respondan a los códigos de trabajo para que estén a la mano de la ley. Con esto la empresa se beneficia agilizando los procesos manuales que tienen, reduciendo los tiempos a la hora de crear la plantilla evitando cualquier error humano que pueda ocurrir en el proceso. Además, se aseguran cumplir la ley como es debido evitando cualquier denuncia por parte de los trabajos que yacen en la empresa.

Alcance Funcional

A continuación, se mencionan todas las funcionalidades que requiere el desarrollo del sistema para el restaurante:

- Calcular planilla: se encargará de hacer el cálculo correspondiente para el pago de los salarios de los colaboradores, tomando en cuenta deducciones impuestas por la ley, horas extras y liquidaciones. De esta manera, se generará salario bruto y neto.
- Calcular vacaciones: este módulo se encargará de validar la cantidad de días disponibles. Se acumula un total de 1 día de vacaciones por cada mes laboral, el sistema permitirá solicitar vacaciones solamente una vez el colaborador tenga un mínimo de un año laborado. El colaborador realizará la solicitud, vía sistema, a la jefatura inmediata, si esta la aprueba, entonces les llegará una notificación a recursos humanos para que también vía sistema apruebe y notifique al segundo colaborador. Si la jefatura no lo aprueba, esta le notifica, vía sistema al colaborador
- Calcular incapacidades: este módulo se encargará de manejar las incapacidades de los empleados, incluyendo fechas de inicio y final de la incapacidad, el tipo y su historial.
- Calcular horas extras: se calculará las horas extras automáticamente para el empleado que sobrepase las horas del horario establecido, lo cual se verá reflejado en el pago mensual.
- Evaluar empleados: se encargará de aplicar un instrumento de evaluación a cada empleado de manera que al final del mes se obtenga un promedio o una nota la cual se considerará a través del tiempo para obtener un ponderado anual.
- Calcular liquidación: este módulo se encarga de hacer el cálculo correspondiente para el pago de las liquidaciones de los exempleados de la empresa, esto con el tal de evitar molestias con los mismo.
- Tramitar permisos: permitirá al usuario generar una solicitud de permiso que será enviada a la jefatura inmediata, para su aprobación.
- Calcular aguinaldos: este módulo se encarga de hacer el cálculo correspondiente para el pago de aguinaldos de los empleados, tomando en cuenta las horas trabajadas a lo largo del año. Esto para realizar los cálculos automáticos y evitar errores.
- Control de asistencia: registrará la hora de entrada y salida de cada empleado de manera manual, calculando automáticamente las horas trabajadas como las horas extras.

Alcance Metodológico

Para el desarrollo del sistema se utilizará una de las metodologías más usadas y conocidas a la hora de llevar a cabo un proyecto, esta metodología es Scrum. La organización dice que es un proceso de gestión que reduce la complejidad en el desarrollo de productos para satisfacer las necesidades de los clientes. Interpretando lo mencionado por Francia (2017) “La gerencia y los equipos de Scrum trabajan juntos alrededor de requisitos y tecnologías para entregar productos funcionando de manera incremental usando el empirismo” (párr.1), por lo tanto, al utilizar Scrum su mejora continua dará mejores resultados.

Con este marco de trabajo se puede definir cada tarea y entregable que se hará a lo largo del producto final, con el propósito de evitar atrasos o errores en la implementación de los módulos en el *software* que se vaya a desarrollar. En este caso como la elaboración es para un proyecto de graduación, se debe realizar solo y no aplicaría la designación de roles para agilizar los procesos.

Alcance Tecnológico

La propuesta que se realizó al restaurante fue un desarrollo web el cual se almacenará en el computador principal de ellos; con esto ahorran un poco del coste del servicio en la nube, DNS y otros servicios. Para llevar a cabo la creación e implementación del *software*, se necesitan las siguientes soluciones de software:

- Se creará con base en los lenguajes de web como: HTML, CSS, C# y JS.
- Para el caso de la base de datos se usará SQL Server Express 2022.
- Para la herramienta de desarrollo, se utilizará Microsoft Visual Studio como el entorno de desarrollo integrado (IDE).

Para la base de datos se utilizará el servicio brindado por Microsoft SQL Server, en la versión 2022 Express; la cual se usará para el guardado y protección de datos. Este servicio es muy robusto a la hora de manejar datos y dar informes al sistema. Además, cumple con la compatibilidad de C# que será la base del prototipo funcional, con la posibilidad de conexión

segura, consultas e instancias; lo cual hace que el desarrollo sea factible y procura la seguridad de los datos sean más adecuada para los usuarios.

El entorno de desarrollo que se será de utilidad es el de Microsoft Visual Studio 2022, que permite el uso fácil de código y el manejo de la arquitectura de una forma sencilla, adaptándose a las necesidades que se requieren. Este entorno da acceso a las librerías y paquetes de manera instantánea proporcionando un desarrollo ágil y seguro con la actualización de paquetes. Cuenta con licencia gratuita y es compatible con sistemas operativos de Windows 10 y 11.

CAPÍTULO II: MARCO REFERENCIAL

La elaboración de un prototipo funcional es parte fundamental a la hora de realizar un desarrollo de *software* programado para establecer los puntos necesarios que eviten cometer errores y donde no haya lugar o tiempo a la corrección sin una detección temprana. Como menciona Chaves (2005):

Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiéndonos conseguir una importante retroalimentación en cuanto a si el sistema diseñado con base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva(párr.1).

De acuerdo con Chaves (2005) la importancia de los requerimientos son la base esencial del prototipo para lograr obtener el producto final con la mayor eficiencia y eficacia esperada. El sistema debe contemplarse como una solución al problema presente y que para el desarrollo hay que investigar a fondo los puntos críticos que el cliente mencione en las especificaciones del prototipo.

Ámbito de Recursos Humanos

Los recursos humanos son parte importante para las empresas, es el departamento de gestión al personal. Lucena (2023) menciona que “La gestión de recursos humanos (RRHH) es la práctica de reclutar, contratar, colocar y gestionar a los empleados de una organización. La gestión de recursos humanos suele denominarse simplemente recursos humanos.” (párr.1), por lo tanto, una empresa que no gestione el departamento de RRHH efectivamente va a tener complicaciones con el personal que impacta de forma negativa el rendimiento de la empresa.

Para el uso correcto de los recursos humanos, las leyes desempeñan un papel importante en el ámbito de los recursos humanos, ya que proporcionan un marco legal que regula las relaciones laborales. Según las disposiciones del Ministerio de Trabajo, la jornada de trabajo efectivo no puede exceder las ocho horas diarias, seis horas nocturnas y cuarenta y ocho horas semanales. Además, se establece que la jornada extraordinaria, sumada a la ordinaria, no puede exceder las doce horas, a menos que ocurra un siniestro o exista un riesgo

inminente que ponga en peligro las personas. Las pautas mencionadas son uno de los cientos de artículos que el Ministerio de Trabajo pauta y se deben tomar en cuenta a la hora de la elaboración del *software*.

Evaluación y Retroalimentación

En el departamento de Recursos Humanos es de suma importancia el evaluar el rendimiento de cada colaborador en la empresa, para la toma de decisiones en los equipos de trabajo. Es posible realizar métricas de distintas maneras y en diferentes situaciones, porque el desempeño del trabajador no se limita solo al tener un buen día de trabajo o manejar cargas pequeñas de trabajo, sino al estar constante bajo presión con situaciones complejas o dar buena imagen al cliente que deba atender. Obteniendo la evaluación debería gestionar una retroalimentación a cada colaborador como redacta Farias (2019):

En todos los casos cuando demos una retroalimentación debemos hacerlo uno a uno con nuestros colaboradores. Debemos considerar que cada caso es individual y mantener un ambiente de respeto, tanto en retroalimentaciones positivas como en negativas(párr.7).

Siguiendo la idea de Farias (2019) por respeto al colaborador la retroalimentación debe realizarse por aparte sin ningún otro empleado en medio para garantizar la integridad del trabajador, ya sea positivamente para él o en caso contrario dar recomendaciones para mejorar el funcionamiento de las labores dentro de la empresa. El objetivo de gestionar el personal con evaluaciones permite comprobar las mejorías que pueden alcanzar cada uno y realizar algún tipo de compensación para manejar una motivación constante en cada colaborador.

Sistema de Remuneración

Comprender un sistema de remuneración y aplicarlo de manera adecuada puede garantizar un mejor rendimiento del personal a cargo, manteniendo una competitividad sana en un ambiente laboral. Las remuneraciones como el salario y las extras son fundamentales

en la organización, sin embargo, son la base principal con lo que debe remunerar a un trabajador, como explica Ersep (2020):

Implementar un sistema de compensación justo y equitativa mejora la marca empleadora, contribuye a atraer talento y disminuye la rotación del personal. Además, cuando los empleados perciben que la empresa valora y recompensa su trabajo, aumenta su motivación laboral y el compromiso con la organización, lo cual incrementa la productividad y redundante en una mayor satisfacción de los clientes. (párr.4)

Como menciona Ersep (2020) implementar el sistema trae beneficios cuantificables, porque evita que el personal tenga grandes cambios que genere nuevas capacitaciones o retrasos en la labor. Manteniendo un entorno justo y bien cuidado la productividad debería incrementar significativamente obteniendo mayores beneficios en la empresa. Algunas de las remuneraciones más son:

- **Medición de rendimiento:** El colaborador en el tiempo va mejorando sus funciones en la organización y comete menos errores en las labores, lo cual agiliza sus procesos de trabajo y los de sus compañeros.
- **Objetivos y metas alcanzadas:** Establecer metas en caso de ventas o captación de nuevos clientes potenciales para obtener mayor beneficio a la empresa.

Control de Asistencia

Llevar un control de entrada y salida de los empleados al principio puede parecer una medida de control absoluto para ellos, pero su enfoque no se basa en solo registrar correctamente si el colaborador cumple el horario establecido por el contrato, sino que maneja bastante información de cada uno de ellos porque al llevar el control puede ser de conocimiento los días que ha faltado por justificación válida o permiso de ausencia remunerada o no. Además de manejar las horas extras de cada uno, esto da facilidades a la jefatura como menciona Márquez (2023):

Entre las facilidades que proporciona un sistema tecnológica para esta tarea, está la aplicación de filtros que permitan hacer un análisis profundo. Así, se pueden definir ejes por departamento, proyectos, cliente, tarea, entre otras. De igual manera, se puede

especificar el tiempo en minutos, horas, días; así como elegir la periodicidad de seguimiento de manera semanal, mensual, anual, etc.(párr.16)

Como dice Márquez (2023) un sistema tecnológico proporciona muchas facilidades, tanto que un buen sistema permite filtrar la información de manera exacta y casi instantánea que se requiera para su análisis debido. El alcance que tiene un control de asistencia se ve reflejado al pasar del tiempo donde si el sistema es tecnológico podría agilizar el proceso de los pagos de los colaboradores en cuestión de segundos. Algunos de los sistemas más usados son los siguientes:

- Registros escritos: los registros son de lo más comunes en las organizaciones porque se tiene la idea de ser de menor costo para implementar, aunque los procesos sean manuales.
- Hojas de Excel: las hojas de Excel, en lugar de los registros escritos proporcionan mayor facilidad en los datos, sin embargo, todavía requiere procesos humanos para llevarlo correctamente.
- Huella digital: un sistema fácil de uso que solo requiere de marcar entradas y salidas para registrarlas en el sistema.
- Usuario y contraseña: un sistema que puede llevar un poco más de tiempo al registrar a los empleados, pero permite dar más funcionalidades dentro del sistema y aprovechar los recursos tecnológicos.

Automatización de Procesos

El proceso de automatizar labores en la empresa, de entrada, puede resultar bastante chocante a los colaboradores por la incertidumbre que genera no tener conocimiento de lo nuevo o de la tecnología; aparte de creer que serán reemplazados por los programas tecnológicos. Sin embargo, es todo lo contrario, ya que al automatizar los procesos estos se les facilita a los empleados haciendo que sus funciones se beneficien y comentan menos errores al ejecutar los procesos asignados, como cita Motessi (2022):

Al automatizar las tareas, no tendrás empleados ociosos, sino que éstos serán mejores y más eficientes, con tiempo suficiente para centrarse en actividades estratégicas en las que los humanos marcan la diferencia. Así, en lugar de redactar correos

electrónicos para el proceso de onboarding, podrán activar un flujo de trabajo y notificar automáticamente las acciones necesarias a las partes implicadas. (párr.2)

Prosiguiendo lo dicho por Motessi (2022) el beneficio que obtiene la empresa al ser más automatizados no solo en materia económica, sino que mejora el rendimiento de los colaboradores, lo que permite que el tiempo pueda aprovecharse en otras funciones o toma de decisiones que requieran el pensamiento humano. Además, de garantizar que en la automatización no haya contratiempos ni interferencia humana en la gestión los procesos.

Leyes Informáticas

Sin embargo, el sistema requiere cumplir con las normas establecidas por el Código Penal debido al mal uso de los recursos informáticos. Por ello la Ley n.º8148 adiciona los artículos de estafa o fraude informático (217 bis), la alteración de datos y sabotaje informático (229 bis) cuyo fin es proteger la integridad e información ya que, todos los datos personales no deben ser manejados de manera conveniente o fraudulenta. El prototipo debe cumplir las pautas establecidas para que los usuarios tengan la certeza que el sistema que les prestará servicios está legalmente creado y no infrinja las normas del código informático.

Parte importante a la hora de desarrollar un prototipo funcional es conocer la privacidad y seguridad de los datos, Al respecto cabe recalcar que no son términos diferentes ya que, la protección de datos es un proceso donde se previene el acceso no autorizado a los datos. Y como clarifica McAfee (2020):

Cualquiera que tenga acceso a su información puede apropiarse de su identidad. Por lo tanto, mantener la privacidad de los datos privados y las categorías especiales de datos es fundamental para proteger la identidad. El robo de identidad es el uso deliberado de la identidad de otra persona. Suele utilizarse para lograr beneficios financieros como sacar créditos y otros beneficios(párr.4).

Siguiendo con lo dicho por McAfee (2020) mantener la privacidad de los datos es fundamental para proteger a los usuarios que utilicen el sistema para que la información brindada al software no se utilice de manera fraudulenta y perjudique tanto de manera personal como financiera; ya que, los sistemas suelen registrar tarjetas de crédito o débito y

si no se tratan con la seguridad de datos, puede llegar a filtrar información sensible que dañe la reputación de la empresa.

Como informático, es imperativo cumplir con la legislación vigente en materia de protección de datos. El deber no solo se fundamenta en la ética como profesional, también en la responsabilidad de preservar la información de los usuarios y clientes. El incumplimiento de alguna de las leyes de protección de datos recae en sanciones legales, multas significativas y consecuencias legales que agravan la profesionalidad de los informáticos. Además, de que al ser profesionales la actuación debe ser a favor de los usuarios y brindar confianza, seguridad respeto a los datos personales de cada uno.

Patentes de Software

Las patentes de *software* se ejecutan para darle una protección al programa desarrollado, lo que genera tener control sobre las funciones totales del sistema, impidiendo que haya réplicas idénticas en el mercado y teniendo la potestad de solo distribuirlo con la patente del *software*, aunque Jedrusik (2017) recalca:

El criterio definitivo para la protección por patente debería ser la calidad de la invención y no su modo de aplicación. La decisión de poner en servicio una invención utilizando un soporte físico o un programa informático suele ser una cuestión de diseño que debería dejarse en manos de los expertos técnicos en lugar de limitarse a lo dispuesto en la legislación de patentes(párr.21).

Lo dicho por Jedrusik (2017) abre un debate sobre si cualquier sistema debe ser patentando o si, en cambio, los que correspondan a una calidad significativa deben ser protegidos de distribuciones sin autorización. Es una idea interesante, ya que en el mundo tecnológico que existe la innovación y replicación van evolucionando sobremanera al pasar de los años, lo cual genera impactos positivos en la sociedad actual.

Normativas ISO

Las normas ISO (Organización Internacional de Normalización) son reglas o parámetros que establecen y garantizan la gestión de productos o prestación de servicios en la industria. Su función se da para lograr coordinar, orientar y simplificar criterios para todo tipo de empresa, reduciendo costes y elevando la eficiencia y eficacia del producto. Como menciona Equipo Orca (2021):

El cumplimiento de las normas ISO es un paso hacia la certificación de tu empresa, lo que puede ser una herramienta de gran utilidad para agregar credibilidad y buena reputación a tus productos y servicios al demostrar que se obedece a las expectativas de calidad de tus clientes. (párr.1)

Como menciona Equipo Orca (2021), el uso de las certificaciones ISO en una empresa mejora la credibilidad del producto, demostrando que posee grandes cualidades aptas para los clientes. A la vez, genera que los clientes tengan la seguridad de estar obteniendo productos que tienen un costo debido a la calidad y seguridad que proporciona las ISO. Es importante reconocer los tipos de ISO existentes en la normativa y comprender mejor el funcionamiento de cada una de ellas y cómo generan un valor diferente.

Licencias de Software

Las licencias de *software* son permisos que tienen uno o varios autores o pertenecen a una entidad, la que otorga el uso del sistema a terceros. Las licencias van desde uso libre para uso personal, código abierto que cualquier persona puede modificar a su gusto o las propietarias que se distribuyen a un precio de mercado para adquiridas. Como explican Brocca y Casamiquela (2005):

En la gran mayoría de los casos, el software se entrega con una licencia de uso. Dicha licencia es un contrato entre el productor y el usuario que establece cuales son los derechos y obligaciones de cada una de las partes. (párr.1)

Siguiendo lo dicho por Brocca y Casamiquela (2005), cada autor define sus derechos de autor y establece un contrato con el que puede usar para vender la licencia a terceros o bien vender la licencia por completo haciendo que tenga un nuevo poseedor. Las licencias

tienen un valor inicial que a medida que crece el producto van adquiriendo mayor valor de mercado, esto hace que los sistemas sean más completos y robustos y se logra que puedan tener un gran valor a futuro.

Requerimientos Técnicos

Para la creación de unos requerimientos robustos, se analizan a detalle todas las necesidades del cliente donde se traduzca en el nivel de desarrollo los problemas presentados que cubrirá el sistema. Como declara Sommerville (2011):

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimientos (IR) (p.83).

Siguiendo la idea de Sommerville (2011), los requerimientos tienen como propósito cubrir las necesidades del cliente. En el proceso de definir los requerimientos hay que documentar la información proporcionada por los clientes para que sea analizada y lograr crear unos requisitos de *software* que cubran las necesidades del cliente y cree un prototipo funcional correcto. Todo el proceso debe documentarse y registrarse para que en caso de modificaciones o desacuerdos en un punto específico haya un respaldo de la información que fue proporcionada.

Los requerimientos se dividen en dos partes básicas: funcionales, que especifican lo que debe realizar un sistema como acciones, y servicios que espera el cliente que proporcione el *software* desde características importantes como las menos obvias. Por su contra parte, los no funcionales son las restricciones o calidades que se imponen al sistema y que por lo general se denomina con adjetivos como Wong (2017) relata:

Describen el comportamiento y los datos que el sistema administrará. También, las capacidades que el sistema podrá realizar en términos de comportamientos o acciones o respuestas de aplicación de tecnología de la información específicas de las operaciones (p.52).

Los requerimientos funcionales, como aclara Wong (2017), administrarán los datos y el comportamiento del sistema, por su contra parte los no funcionales son las restricciones o atributos de calidad que no están asociadas directamente con las funciones específicas del sistema. Al respecto, Wong (2017) especifica:

Este tipo de requisitos están relacionados con la operación del software y hacen referencia al comportamiento de la solución, describen condiciones ambientales bajo las cuales la solución debe permanecer activa por determinados períodos de tiempo o cualidades que los sistemas deben tener a nivel operacional (p.52).

Una vez se especifican todos los requerimientos necesarios para crear un prototipo funcional en la gestión de recursos humanos, se realiza una documentación detallada con las funcionalidades del sistema que será desarrollado, planteando los objetivos claros y realistas que debe tener el software; también se hace una descripción completa de los requisitos funcionales y no funcionales. El proceso de documentación tiene que pasar por una revisión exhaustiva por los involucrados en el proyecto para que sean aceptados y se aprueben para comenzar la designación de tareas.

Ciclo del Software

Comprendiendo los conceptos para un buen sistema de gestión de recursos humanos, la prioridad se enfoca una calidad de *software* ideal, que para ello se debe entender cómo un sistema puede tener el mejor ciclo de vida para que sea un programa de alta calidad. Para minimizar los riesgos del prototipo, para que el proyecto sea excelente una planificación anticipada que cumpla las expectativas del cliente y, a su vez, se compagine con una metodología ágil como menciona Meneses (2023):

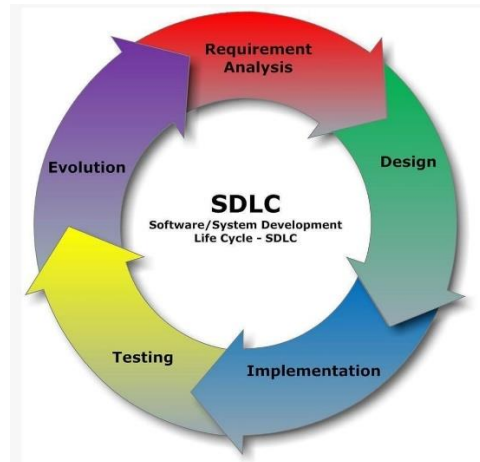
SDLC le da a cada proyecto un framework individualizado para trabajar dentro. Mantiene a todo el equipo de desarrollo en la misma página para que el proceso permanezca ordenado y eficiente y resulte en productos de software de alta calidad y bajo costo(párr.5).

Recalcando lo antes mencionado por Meneses (2023), el SDLC toma a todo el equipo de desarrollo para que elaboren el proyecto de forma simultánea y eficiente, que permite tener un mejor control de los procesos en desarrollo. Para la ejecución correcta del ciclo de

vida se atiende a las fases comunes del desarrollo de *software* como se muestran en la ilustración 1 relativa al ciclo vida del sistema.

Ilustración 1.

Ciclo del software



Fuente: Insight, 2024

Las fases son parte importante a la hora de llevar a cabo una implementación de *software* independientemente del tipo de sistema que requiera el cliente, debido a que la planificación adecuada logra que a futuro el *software* vaya evolucionando y adaptándose a las nuevas tecnologías no quedando obsoleto en el tiempo, así también con un código ordenado se ahorra mucho tiempo a los desarrolladores que implementen nuevos cambios.

Las fases del SDLC son:

- **Planificación:** Contiene tareas como análisis de costos y beneficios, programación, la estimación de recursos y su asignación. Recopila toda la información y requisitos de las partes interesadas para la creación de documentación precisa de los requerimientos.
- **Diseño:** Los ingenieros de *software* analizan los requerimientos del documento previamente creado, para plantear la integración de módulos e identificar las herramientas de desarrollo.

- **Implementación:** En la fase de implementación, el equipo de desarrollo descodifica el producto. Una vez analizado los requerimientos, proceden a identificar las tareas y codificación más pequeña para realizar diariamente y obtener el resultado final.
- **Pruebas:** El equipo de desarrollo combina las pruebas automáticas y manuales para comprobar si el software cuenta con errores.
- **Despliegue:** Cuando los equipos desarrollan software, lo codifican y prueban en una copia diferente que no es a la que acceden los usuarios. A esto es lo que se le conoce como producción, mientras en otras copias se realizan pruebas de entorno para garantizar el funcionamiento.
- **Mantenimiento:** En la fase de mantenimiento el equipo corrige los errores, resuelve los problemas del cliente y administra cambios en el software. Además, se supervisa el rendimiento general del sistema, la seguridad y la experiencia de usuario.

Gracias a las fases antes mencionadas, la gestión del *software* obtiene mayor valor de vida y costo, porque permite documentar y repartir los procesos de forma eficiente, dando paso a futuras mejoras en el sistema. Los SDLC aparte de contemplar las fases se basan en adaptar los modelos de desarrollo de software a requerimientos específicos para el cliente, como menciona Hernández (2022):

Le llamamos modelos de desarrollo de software a los diversos procesos o metodologías que se seleccionan para desarrollar un proyecto en función de sus objetivos. Existen muchos modelos, y estos mismos especifican las distintas etapas del proceso y el orden en que se llevan a cabo(párr.2).

Como especifica (Hernández, 2022) hay diversos procesos y metodologías que van en función de los objetivos planteados para el proyecto. Los modelos comprenden todo tipo de proyecto y la elección debe ser siempre analizada por los integrantes para comprobar la eficiencia y rendimiento que cada modelo le puede brindar al desarrollo. Los modelos del ciclo de vida son:

- **Modelo Ágil:** funciona muy bien para proyectos que tienen requisitos cambiantes. Implementa las fases SDLC en una estructura circular que permite flexibilidad. Los beneficios que aporta el modelo ágil ayuda al cambio constante del proyecto, gracias a la flexibilidad que permite al ser tan volátil, también al ser de los más eficientes para trabajar a plazos rápidos y recopilar comentarios de los clientes.

- **Modelo Espiral:** utiliza una estructura circular para moverse por los pasos del SDLC. Utiliza iteraciones, o repeticiones de procedimientos, y cada iteración aborda los requisitos específicos del proyecto. Es un modelo basado en el riesgo que funciona bien para proyectos de alto riesgo con requisitos complejos. El beneficio del modelo espiral es su pronta identificación de riesgos como su transparencia ya que, en cada fase se obtienen una retroalimentación, al ser tan eficiente en la detección de errores hace que las estimaciones de costos y tiempo sean muy precisas.
- **Modelo Cascada:** su utilización es bastante conocida, ya que es de los modelos más probados. Avanza a través de las fases de forma lineal, lo que lo hace ideal para proyectos pequeños con requisitos claros e invariables. La implementación del modelo cascada les da ventaja los programadores inexpertos, debido a que su uso es simple, logrando crear puntos de entradas y salidas comprensibles en cada fase. A su vez, conduce a que el producto final sea de muy alta calidad.
- **Modelo iterativo:** es un modelo cíclico que repite una serie de pasos, con cada iteración acercándose al desarrollo del producto final. Comienza con una versión simple del *software* y se vuelve más complejo con cada iteración. Este modelo funciona mejor cuando los requisitos del proyecto se entienden claramente. Al ser un modelo muy flexible, permite detectar fallos muy pronto a la hora que se ejecutan los procesos, rentabilizando los cambios que se deban ejecutar en los procesos.

Los modelos tienen estructuras distintas, pero a su vez se basan en las seis fases del ciclo de vida, donde el cambio que tienen es la forma como ejecutan las fases ya que, no hay un camino correcto, sino varios caminos que cumplen con los requisitos para llegar a la meta. Aunque el modelo cascada es el más utilizado en el día a día de los desarrolladores por su avance lineal para proyectos de una escala menor, lo que permite distribuir las tareas en actividades pequeñas.

Las Bases de Datos

Las bases de datos son una forma de almacenar información de un sistema de forma física en servidores o computadores, así como el almacenamiento en la nube. Su importancia se basa en el alcance y rendimiento que dan a las empresas, porque al tener guardados tantos

datos su filtración ocasiona una toma de decisiones que beneficien a las organizaciones como redacta Quintana (2017):

Las Bases de Datos juegan un papel importante en la mayoría de las áreas donde se utilizan computadores, permitiendo almacenar grandes volúmenes de datos acerca de la empresa, los cuales son percibidos a través de los usuarios, de la misma manera la información obtenida de los datos almacenados debe estar en una forma que sirva para administrar, planear, controlar y tomar decisiones dentro de una organización. (párr.1)

Prosiguiendo con lo mencionado por Quintana (2017) la información almacenada se usa para administrar y tomar decisiones en el accionar de las empresas, sin importar qué funciones lleve a cabo la organización. Tener información valiosa hace que haya una cercanía con los usuarios que consuman productos, que accedan a las funciones del *software* o las reseñas en las páginas webs.

Modelado de Datos Relacionales

Una base de datos relacional sirve para dar una organización de los datos que vayan a ser registrados en el sistema que conecta de manera lógica las tablas una con las otras para manejar los datos de forma consistente. Las tablas están conformadas por columnas que tienen como función realizar búsquedas de datos de forma ágil, fácil y de forma ordenada.

Modelado de Datos no Relacionales

Las bases de datos conocidas como no relacionales son un sistema que se caracteriza por no usar consultas del lenguaje SQL. Como dice su nombre “no relacionales” es porque no mantienen una estructura definida, los datos no se almacenan en tablas como el caso de las relaciones. El uso para la base de datos es almacenar documentación y datos no organizados para soportar grandes volúmenes de datos.

Copias de Seguridad

El almacenamiento en las bases de datos no provee una garantía total de la información, debido a que existen riesgos que comprometen los datos, los riesgos pueden ir desde ataques malintencionados desde fuera de la empresa como daños en el equipo físico que generan pérdidas económicas y de tiempo; por eso las copias de seguridad juegan un papel clave como relata Santiago (2023):

La realización de copias de seguridad para sitios web y bases de datos es fundamental, ya que es una medida crítica para proteger los datos y la información digital de pérdidas imprevistas o daños tanto a nivel personal como empresarial. Sobre todo, estas medidas representan una clave para asegurar la continuidad del negocio, pues la preservación de la información valiosa. Protege intereses tanto físicos como financieros(párr.3).

Como menciona Santiago (2023), las copias de seguridad dan continuidad de negocio porque preservan toda la información valiosa de la empresa, haciendo que al presentar algún tipo de inconveniente la organización siga en funcionamiento, mientras se intenta recuperar la información perdida o dañada. No existe un manual concreto por seguir para realizar las copias de seguridad, sin embargo, sí hay tipos de copias para las necesidades que requiera la organización, tales como las mencionadas por Dolbuck Ciberseguridad (2023):

- Externas: lo correcto es guardar una copia en un dispositivo diferente al sistema informático en el que se encuentra el original.
- Automáticas: porque las copias de seguridad manuales son menos seguras y dependen de que el usuario las realice cada día, lo cual puede dar lugar a descuidos y errores humanos.
- Programadas: porque es esencial que las copias de seguridad se realicen con una cierta periodicidad, reservando el espacio necesario para su mantenimiento.
- Encriptadas: porque de este modo, en caso de que las copias de seguridad cayeran en manos de un ciberdelincuente, no podría averiguar muy fácilmente los datos que se encuentran en ellas porque estarían encriptados.
- Escalables: porque sabemos que la información que genera el negocio va en aumento y se necesita utilizar un mismo sistema, sin tener que llevar a cabo una migración

pasados unos años porque ya no hay suficiente espacio en los servidores del proveedor, o el propio servidor.

Las formas en las que puede guardar las copias de seguridad van acorde al tipo de necesidad de la organización, por eso debe analizar qué modo de copias de seguridad es la más eficiente y si el rendimiento cumple a la perfección para evitar situaciones adversas. Hay maneras de buenas prácticas de la seguridad de los datos, cada empresa debe gestionar y crear políticas de copias de seguridad que solucionen la problemática que puede ocurrir en los daños a la información sensible. Algunas de las políticas por implementar son:

- Generar copias de seguridad automáticas que se ejecuten una vez por día, por semana o que la empresa decida el tiempo entre cada una de las copias.
- Programar un horario fijo donde las copias de seguridad se realicen con éxito, que podría ejecutarse después de la hora de salida o en horas de la madrugada.
- Establecer métricas donde el proceso de las pruebas sea efectivo a la hora de recuperar los datos.
- Revisar periódicamente las copias de seguridad para comprobar la validez de las copias.

Implementando las políticas anteriores y analizando qué otras se requieren para la empresa, la aseguración de los datos tiene mayor robustez a la hora de trabajar con la información. Sin embargo, hay más factores que deben tomarse en cuenta para salvaguardar los datos correctamente sin manipulación externa, por eso es importante la seguridad de los datos, porque no solo se pueden ver afectados dentro de la empresa, sino también fuera de la misma.

Protección de Datos

La protección de los datos cumple una función necesaria a la hora que haya manejo información delicada en la base de datos como aclara IBM (s.f):

La seguridad de datos es la práctica que consiste en proteger la información digital contra el acceso no autorizado, la corrupción o el robo durante todo su ciclo de vida. Es un concepto que comprende todos los aspectos de la seguridad de la información, desde la seguridad física del hardware y los dispositivos de almacenamiento hasta los

controles administrativos y de acceso, así como la seguridad lógica de las aplicaciones de software (párr.1).

Siguiendo a IBM (s.f), proteger la información no solo es en la parte tecnológica, también entra en juego la seguridad física de los *hardware*, por lo que se requiere restringir el acceso a los almacenamientos de datos a personas no autorizadas tanto física como digitalmente. En ocasiones los accesos a los departamentos informáticos aplican parámetros para la entrada y salida del personal, algunas de ellas son:

- Entrada con identificador único de los informáticos.
- Ingresar sin ningún objeto que puede sustraer información del hardware donde se almacena la información.
- Cámaras de seguridad en la entrada del departamento u oficina donde se hayan los servidores.

Los puntos anteriores son algunos de tantas restricciones que implementa una empresa para asegurarse que se protejan los datos. Cabe recalcar, que aplicar las políticas físicas no significa que la información está segura al cien por ciento, debido a que existen muchas vulnerabilidades digitales que atentan con la seguridad.

Encriptación

La encriptación es una forma de cifrar la información en patrones complejos para mantener los datos seguros, permite que los terceros que intenten acceder a la información no logren descifrar lo que hay en la encriptación; así se evita el riesgo de robo o secuestro de los datos, tal como recalca Cabrera (2023):

La encriptación de datos es esencial para garantizar la privacidad y la seguridad de la información en la nube. Cuando los datos se almacenan o se transfieren sin encriptar, están en riesgo de ser interceptados o robados por terceros malintencionados. La encriptación de datos en tránsito (*in transit*) y en reposo (*at rest*) ayuda a proteger los datos de estos riesgos, asegurando que solo las personas autorizadas puedan acceder a ellos(párr.1).

Como especifica Cabrera (2023) a diario las empresas confían en la encriptación de la información porque al realizar el proceso de protección de datos se aseguran de que no

sean de fácil acceso para los atacantes, lo que provoca que, si se trata de vulnerar la información, no puedan tomar los datos confidenciales tan fácilmente. La encriptación va de la mano en la programación, porque un *software* robusto debe tener al menos un módulo que restrinja el acceso y asegure la información en la base de datos. Como menciona Alegsa (2023):

En programación, un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo(párr.1).

Prosiguiendo lo dicho por Alegsa (2023), se ahorra tiempo en el desarrollo al reutilizar los módulos a lo largo del sistema, cumpliendo con eficiencia en el código. Una vez todos los módulos tengan las funcionalidades completas, se debe instalar en un servidor para que los usuarios tengan accesos a los recursos disponibles del servicio en caso de que sea una aplicación web como recalca Godaddy (2023):

Los servidores son responsables de almacenar y distribuir diversos tipos de recursos, como archivos, páginas web, correos electrónicos, bases de datos o aplicaciones. Además de almacenar y entregar datos, los servidores también pueden realizar tareas de procesamiento y cálculo, lo que les permite ejecutar aplicaciones complejas de manera eficiente(párr.2)

En consonancia con Godaddy (2023), el servidor almacena todos los recursos disponibles en un sistema, como la base de datos, aplicaciones o páginas web. Además, el servidor da acceso a que los usuarios ingresen a las páginas webs sin necesidad de estar localmente conectados a la misma red con los recursos web. Las páginas web normalmente funcionan solo al tener acceso a la red y responde únicamente a las URL como clarifica 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(párr.4)

Como menciona Coppola (2023) los sitios web tienen un identificador llamado URL (Uniform Resource Locator) cuyas direcciones son traducidas por un servidor DNS

encargado de darle una dirección en palabras porque normalmente las páginas webs se identifican con una serie de números son difíciles de aprender para acceder a ellos.

Lenguajes de Programación

Los lenguajes de programación son la base principal del desarrollo de *software* y tienen aplicaciones distintas que hacen darle vida a un sistema, ya sea visualmente o en sus funciones como aclara TalentCross (2023): “Para cualquier profesional IT, conocer cuáles son los lenguajes de programación más usados es clave en su desarrollo, pues le permite identificar hacia dónde dirigir sus estudios y capacitaciones para ganar valor en el mercado.” (párr.1). Es importante conocer los lenguajes más usados y tener claro que van actualizando su tecnología a través del tiempo para no quedar obsoletos.

JavaScript

El lenguaje JavaScript es el lenguaje por excelencia en la parte de desarrollo web, pues cuenta con múltiples funcionalidades que le permiten al usuario interactuar con el sistema de forma dinámica y al instante. Al ser un lenguaje tipado dinámico las variables no deben declararse algún tipo de dato como menciona Henry (2023):

JavaScript es un lenguaje de tipado dinámico, lo que significa que no necesitamos declarar y establecer un tipo –como número o cadena de caracteres– específico e inflexible para las variables, lo cual da mucha libertad y sencillez, en especial al aprender el lenguaje(párr.6).

Como recalca Henry (2023), el lenguaje JavaScript da mucha libertad cuando se va a aprender la sintaxis, porque tiene diferentes formas de aplicarlo en las páginas webs creando una sintonía con peticiones en tiempo real sin recargar el sitio para visualizar algún cambio hecho por el usuario, también les da vida a las páginas con el lenguaje CSS (Cascading Style Sheets) que es el encargado de darle un diseño agradable a la vista del usuario con diferentes modos o diseños que permite CSS.

El Lenguaje C#

El lenguaje C# es un lenguaje que se utiliza tanto para el desarrollo *BackEnd* (es la arquitectura interna del sistema) como en el desarrollo del *FrontEnd* (parte visible del *software*). Su uso va a depender de las necesidades que requiera el proyecto. El lenguaje tiene una orientación a crear componente o estructura en la parte interna como recalca Microsoft (2024):

C# es un lenguaje multiplataforma de uso general que hace que los desarrolladores sean productivos al escribir código de alto rendimiento. Con millones de desarrolladores, C# es el lenguaje .NET más popular. C# tiene una amplia compatibilidad con el ecosistema y todas las cargas de trabajo de .NET. Basado en principios orientados a objetos, incorpora muchas características de otros paradigmas, en particular de la programación funcional. (párr.2).

Como menciona Microsoft (2024), muy en el fondo el uso principal del lenguaje es orientar el uso de los objetos para consolidar una arquitectura de *software* robusta, que permite ser reutilizado en cualquier sitio del programa con acceso a los objetos declarados en el sistema.

CAPITULO III: MARCO METODOLÓGICO

La importancia del marco metodológico es plantear una serie de estrategias para la recopilación de datos que independientemente del enfoque por vaya a utilizar, se procede a obtener la información mediante la investigación que sea válida para el estudio por ejecutar. Para abordar el marco debe tener claridad en elegir el enfoque correcto para garantizar el éxito de la investigación.

Enfoques de Investigación

Para una buena investigación, se debe determinar el enfoque más indicado y obtener las pruebas correctas, eficientes que ayuden a solucionar la problemática. Según como se quiera expresar la información, ya sea de manera cuantificable o cualitativa, así se van a plantear los enfoques, porque varía según el tamaño por investigar, como menciona Mata (2019):

Quando hablamos de enfoque de investigación, nos referimos a la naturaleza del estudio, la cual se clasifica como cuantitativa, cualitativa o mixta; y abarca el proceso investigativo en todas sus etapas: desde la definición del tema y el planteamiento del problema de investigación, hasta el desarrollo de la perspectiva teórica, la definición de la estrategia metodológica, y la recolección, análisis e interpretación de los datos. (párr.1).

Siguiendo lo mencionado por Mata (2019), la investigación se basa en una serie de etapas con que se debe plantear el marco metodológico, con el fin de recolectar información para el análisis de datos. Los datos se interpretan para corroborar si la investigación resultante es correcta o debe usar otro enfoque para obtener los resultados esperados, siendo importante ejecutar un análisis de la situación como de los datos obtenidos.

Enfoque Cualitativo

Para abordar la problemática que presenta la organización se opta por un enfoque cualitativo, debido a los métodos que ofrece para recopilar datos de manera no numérica, siendo el método más factible y con mejor resultado debido a que la empresa cuenta con

pocos colaboradores. La toma de información para el enfoque resulta ser eficiente y exacta para el problema por resolver, como menciona Santander (2021):

Este tipo de investigación se basa en el juicio de los investigadores, por lo que se debe reflexionar cuidadosamente sobre sus elecciones y suposiciones. Se trata de una técnica que se emplea habitualmente en áreas como la antropología, la sociología, la educación o la historia, entre otras, ya que esta les ayuda a obtener una mejor comprensión de conceptos complejos, interacciones sociales o fenómenos culturales(párr.1).

De acuerdo con Santander (2021), usar el enfoque cualitativo dará una visión más amplia para comprender las interacciones sociales o conceptos complejos. Además de reflexionar sobre las elecciones que tome un investigador cuando realice la investigación, los resultados que el investigador obtenga se expresan en palabras. Para realizar el enfoque cualitativo se requiere aplicar los siguientes métodos de recolección de datos:

- Observaciones: se observa cómo los empleados realizan los procesos de asistencia, pagos efectuados, así como marcar las horas extras y las ausencias. Todo lo ejecutan en bitácoras escritas.
- Historias: se pide a los colaboradores contar las experiencias en los temas de recursos humanos, para recopilar información importante que brinden para la investigación.

Existen otros métodos para utilizar en el enfoque cualitativo, sin embargo, se optó por estos dos, debido a que con la observación se aprecia cómo los colaboradores ejercen los procesos de manera manual para comprobar la ineficiencia de manejar en bitácoras los registros de cada empleado. También, obtener las historias de cada colaborador permite contrarrestar la información tomada mediante las observaciones que fueron previamente anotadas, con el fin de analizar los procesos y los errores que cometen diariamente en la jornada laboral.

Enfoque de Investigación Seleccionado

La elección del enfoque seleccionado permite obtener información valiosa de la problemática planteada por la empleada, debido a que el proceso que ejecutan los colaboradores genera errores y molestias tanto a jefatura como a los empleados. Dado que la

información recopilada no genera resultados numéricos, entonces el análisis de los datos comprende observaciones e historias que dan los colaboradores para contemplar el impacto negativo de los procesos manuales. Dicho lo anterior, las razones de la elección de los métodos mencionados son las siguientes:

- Al observar los procesos que realizan los colaboradores se anota detallado como ejecutan cada proceso en las horas laborales.
- No tiene impacto negativo en la jornada laboral, al no interferir directamente a los colaboradores, esto no produce retraso a los empleados.
- Las anotaciones recopiladas sirven para tomar decisiones después de un análisis correcto de la información.
- Las historias de los colaboradores son de gran aporte debido a que sus experiencias son valiosas a la hora de elaborar el prototipo ya que, son los encargados de utilizar el sistema.
- Cada experiencia de los empleados es importante porque dan puntos de vistas diferentes que enriquecen la investigación y permite ahondar otras perspectivas por tener en cuenta durante el proceso.

Las razones anteriores brindan una oportunidad de vivir de cerca los procesos de la problemática, así como distintos enfoques del problema que se presenta, esto conlleva a que la investigación requiera un análisis exhaustivo de las anotaciones hechas por observación y todas las experiencias de los colaboradores, hace plantear la investigación más completa abordando todos los diferentes puntos de vista observados en la documentación recopilada.

Tipos de Investigación

La investigación es parte fundamental en cualquier trabajo que requiera obtener información precisa y respaldar la documentación con una mayor veracidad con la que fundamente los conceptos en el documento, pasando a ser importante cómo se va a enfocar la investigación. Para solucionar los problemas presentes en la investigación, se aplica el método científico como aclara Suárez (2023):

En el ámbito del conocimiento científico, existen múltiples maneras de abordar los problemas existentes, en vista de que todas no se pueden estudiar de la forma más

efectiva a partir del uso de algunas metodologías, lo que a su vez ha propiciado que sea necesaria la implementación de nuevas maneras de abordar estas realidades presentes en el mundo(párr.6).

De acuerdo con Suárez (2023) hay nuevas maneras de afrontar la investigación con diferentes métodos para su implementación, haciendo que haya más opciones para el tipo de investigación que necesite, adaptando el uso a las situaciones que requieran en el momento. Existen varios tipos de búsquedas de información que se adecuen a la profundidad de conocimientos que requieran y los tipos son:

- Investigación descriptiva: se realiza un informe detallado sobre el tema por estudiar, no importa su causa ni las consecuencias, se basa en tener una visión clara del problema.
- Investigación explicativa: la explicativa se relaciona en muchas ocasiones con la existencia de la causa y su consecuencia. Se llega a conocer el porqué de la situación.
- Investigación exploratoria: el objetivo de la investigación exploratoria es marcar la información analizada e investigada con el fin de conocer de cerca el problema y más adelante detallar la investigación.

Los tipos que existen se realizan de formas distintas con el fin de proveer una mejor investigación de la manera que necesite abordarla, lo que permite que haya caminos donde el resultado esperado sea satisfactorio para los investigadores. También, es importante analizar cuál de los tres tipos mencionados son más eficientes en el problema por solventar y analizar el tipo con el enfoque que se haya seleccionado para la investigación.

Investigación Descriptiva.

La investigación dará una solución al problema de manera eficiente ya que, acepta utilizar el método de observación que permite visualizar el comportamiento que tienen los trabajadores cuando elaboran los procesos del problema. Además, no interrumpe las labores de los colaboradores, debido a que la observación puede ser participante o simplemente un observador que anota en una bitácora la información valiosa.

La ventaja directa de la recopilación de datos de los colaboradores hace que la problemática sea sencilla de entender y que al tomar la información comparando con las

anotaciones realizadas durante la observación permite aclarar dudas que presente la problemática que no se hayan resuelto antes. Sin embargo, el análisis debe comprender toda la información desde la problemática, revisando las observaciones y concluyendo con las experiencias de los colaboradores sin excepción, para lograr los resultados esperados y dar una solución completa al problema.

Tipo de Investigación Seleccionado

El tipo de investigación seleccionado da una visión clara del problema por investigar, ya que se llegan a analizar los diferentes puntos críticos de la investigación y se aporta información de gran valor para el tema por resolver. La investigación aclara de forma detallada del tema sin importar las causas ni consecuencias, lo que permite explorar a fondo la información recopilada en la investigación.

Fuentes de Información

Las fuentes de investigación son documentos, revistas, libros o blogs donde se toman como referencia de origen la información recolectada por otras personas. Dependiendo de las fuentes que consulte, los datos del autor pueden ser más o menos fidedignos, esto genera que se deba realizar una búsqueda profunda del tema investigado para obtener varios autores y contrarrestar la información encontrada Suárez (2024), lo describe así:

Es importante tener en cuenta que no todas las fuentes de información son igualmente confiables o relevantes para cada tema o disciplina. Por eso, es fundamental saber cómo evaluar la calidad y pertinencia de las fuentes de información para poder seleccionar las mejores para nuestro trabajo (párr.5).

Siguiendo con lo mencionado por Suárez (2024), es importante el saber cómo evaluar la calidad de las fuentes de información, dado que, al vivir en un mundo con fácil acceso al internet, mucha de la información existente corresponde a son traducciones de páginas inglesas, por lo que no se comprueba los datos que extraen; también hay blogs que copian la información de otras fuentes sin ver la veracidad de los datos. Existen tres tipos de información, a saber:

- Fuentes Primarias: son la base principal de la investigación, son las más cercanas al evento posible, es información con datos originales sin interpretación con el tema de estudio.
- Fuentes Secundarias: su objetivo principal no es ofrecer información original, sino de recolectar datos de libros o revistas donde haya interpretación de los temas investigados.
- Fuentes Terciarias: la información contiene análisis de datos de fuentes primarias que dirigen al autor original y se interpreta de la fuente original.

Variables

La ventaja directa de la recopilación de datos con los colaboradores hace que el análisis de la problemática sea más sencillo de comprender para el investigador, pues al tener de primera mano los datos de los involucrados, se puede entender la situación para elaborar una estrategia de implementación. Las variables son de diferentes tipos, por lo que se obtiene más información de distintas maneras para la investigación.

Variable Conceptual

Es la que define y explica lo que se va a estudiar en la investigación.

Variable Operacional

La variable operacional es la que mide la variable conceptual con el fin de cuantificar el estudio realizado, son las que pasan lo abstractos a términos medibles.

Variable Instrumental

Es la variable que se va a utilizar como instrumento que sirve para el estudio de la investigación con el fin de medir o evaluar.

Tabla 2.*Cuadro de variables*

Objetivo Específico	Variable	Variable Conceptual	Variable Operacional	Variable Instrumental
Analizar los requerimientos que solucionen la problemática planteada	Requerimientos	Según Velázquez. (2022) “un requisito de <i>software</i> es cualquier término que defina las características y funciones de una aplicación o sistema”. (párr.6)	Entrevista Observación	Guía de Entrevista Guía de Observación
Diseñar la arquitectura del prototipo, definiendo una estructura clara y robusta relacionada a la gestión de recursos humanos	Arquitectura	Según Ojeda (2023) “La arquitectura de <i>software</i> es el diseño de alto nivel de un sistema de <i>software</i> que abarca la estructura de sus componentes, sus relaciones, y los principios	Casos de uso Diagrama de entidades	Draw.io MySQL Workbench

Objetivo Específico	Variable	Variable Conceptual	Variable Operacional	Variable Instrumental
		que guían su organización” (párr.3)		
Programar el sistema requerido con el fin de que resuelva la problemática planteada	Programar	Como menciona Swanros (2024) “Programar es la tarea mecánica de escribir código y decirle a una máquina qué hacer” (párr.9)	Creación de base de datos para desarrollar el sistema.	Visual Studio 2022 Visual Studio Code MS SQL Server Express Edition
Probar el funcionamiento correcto del prototipo funcional.	Puntos débiles	Según Deusto (2015) “Aunque en su empresa todos los procesos sean "redondos" y aunque haya ajustado los procedimientos a ciertos requisitos específicos, es posible que todavía existan	Pruebas del <i>software</i> para el funcionamiento correcto.	Guía de pruebas

Objetivo Específico	Variable	Variable Conceptual	Variable Operacional	Variable Instrumental
		puntos débiles y carencias en algunos lugares” (párr.1).		

Fuente: Elaboración Propio

Instrumentos de Recolección de Datos

Analizar los instrumentos de recolección de datos es parte fundamental para tomar las mejores decisiones para la investigación, ya que hay varios tipos de instrumentos que funcionan de diferente manera y dan datos cualitativos o cuantitativos. Dependiendo de qué se busque de los instrumentos, se van a modificar para obtener la información acertada que ayude a comprender la situación o problemática de la investigación.

Observación

La recolección de datos para la observación puede ser de forma participativa o no participativa, es decir, al recolectar información para la investigación se observa los procesos definidos con anterioridad, observando cada proceso desde un lugar que no interrumpa las actividades de los involucrados o, por el contrario, siendo parte del proceso y vivir desde dentro como funcionan los procesos por analizar.

Entrevista

Es de suma importancia que la recolección de datos para la entrevista sean solo dos personas, el entrevistador y el entrevistado para que no haya terceros en la escena que influyan en las respuestas de los entrevistados, para lograr obtener la información limpia y

sincera de los colaboradores. Las preguntas no deben ser planificadas sino seguir una línea congruente con el tema por investigar para obtener diferentes puntos de vista.

Proceso de Recolección y Análisis de Datos

En la toma de información para la investigación se utilizarán dos instrumentos de recolección de datos fundamentales en las decisiones de la elaboración del prototipo funcional. Antes de recolectar la información se plantea una estrategia para comprobar de qué manera sería más efectiva la utilización de los instrumentos, para obtener los datos fidedignos que se busque para no recibir información que no se aplique en la investigación.

Para el instrumento de observación se procederá a ir a las instalaciones de la organización en diferentes fechas para apreciar desde un lugar donde no haya interrupción a las actividades de los empleados, viendo cada proceso como lo ejecuta cada colaborador y comprendiendo el impacto negativo que sucede al realizarlo de la manera planteada por la empresa. Los procesos que se observan en la empresa son los siguientes:

- registro de asistencia,
- cálculo de horas extras,
- pagos de salario,
- trámites de permisos y
- trámites de incapacidad.

El segundo instrumento para utilizar en la investigación será la entrevista, que contiene un total de diez interrogantes entre preguntas abiertas y cerradas. Al colaborador le tomará entre 15 a 20 minutos contestarlo. La entrevista se hará en secuencias de las preguntas establecidas de acuerdo con una guía lineal para no perder el hilo conductor de la investigación. Las personas entrevistadas son todos los empleados de la empresa, al ser muy pocos colaboradores la entrevista será factible con poca duración de tiempo. Algunas de las preguntas se destacan a continuación:

- ¿Cómo es el proceso de cálculo de salarios?
- ¿Cómo registran la asistencia?
- ¿Cree que puede mejorar la forma de registrar las entradas y salidas de los empleados?
- ¿Cómo es el proceso de solicitud de permisos de trabajo?

- ¿Alguna vez le han pagado de más por las mismas horas trabajadas o menos por las horas extras?

CAPITULO IV: ANÁLISIS DE RESULTADOS

Se procedió a visitar la organización para aplicar los instrumentos de recolección donde se observaron los diferentes procesos que plantea la problemática y se consultó directamente a los colaboradores sobre los procesos. También, se aplicó el instrumento de entrevista para cada uno de los colaboradores; a partir de este se obtuvo información valiosa que da respuesta a las mejoras que podrían tener los procesos para el beneficio mutuo.

Aplicación de los Instrumentos de Recolección de Datos

Instrumento de Observación

El 29 de abril del 2024 se observó desde una zona de trabajo los procesos que ejecutan los trabajadores diariamente para no influir en los comportamientos que podrían tener al ver una persona externa captando los procesos en cada momento. Después de apreciar cómo se llevaban los procesos se procedió a preguntar a cada colaborador por qué realizan los procesos así y si no resultaba problemático no controlar correctamente los procesos para que no haya errores humanos en él y cómo se sentían al respecto.

En el proceso de registro de asistencia no hay una persona encargada que compruebe la asistencia correcta de los colaboradores, ellos llegan y con su firma escriben la hora de entrada establecida y al finalizar la jornada registran al lado la hora de salida. Al implementar un control de asistencia en el sistema, la jefatura y los empleados se benefician de la exactitud de los cálculos de salario sin que los registros se alteren o se marquen incorrectamente.

Respecto al pago de salarios, se toma la bitácora y el gerente empieza a calcular las horas totales del mes de cada empleado, para ejecutar el pago correspondiente de los colaboradores. El tiempo que toma calcular cada salario es bastante y demora los pagos para el final del día. Con el proceso automatizado, la jefatura ahorra un día completo, debido al que el sistema generará los cálculos presionando un botón, para mostrar en pocos segundos todos los salarios correctamente calculados.

Para el trámite de permisos se procede de una manera peculiar, porque el colaborador consulta vía WhatsApp si puede ausentarse un día o medio día laboral con su motivo correspondiente; el gerente debe analizar la situación del restaurante para poder conceder el

permiso al empleado o negarlo, en caso de que no se llegue a un acuerdo. Las ventajas de mantener los permisos en el sistema es que se registrarán en la base de datos para ser consultados por la jefatura, como también tendrá la opción de negar o aceptar los permisos de cada permiso con goce de salario o no.

Como en el proceso anterior, las incapacidades también se manejan vía WhatsApp, el colaborador solo envía foto del documento de la incapacidad y no lo lleva para su almacenamiento. El gerente solo confirma la incapacidad y no registra en ninguna bitácora los días de ausencia del colaborador por incapacidad. Al implementar las incapacidades, la jefatura se beneficia en el cálculo del salario sin tener que buscar las incapacidades de los trabajadores en los registros, sino que se automatiza el proceso y se optimiza el tiempo.

Instrumento de Entrevista

Las entrevistas se llevaron a cabo dentro del restaurante el día 29 de abril (para evitar ausencia de algún colaborador), después de ver cómo realizaban los procesos. En su momento de descanso se procedió a ejecutar la entrevista a cada uno por aparte sin ninguna interrupción, para que los datos fueran confiables y no se vieran influidos por respuestas externas o por presión de algún otro colaborador.

Mediante las preguntas se puede comprender que todos los colaboradores conocen cómo es el proceso de asistencia y lo aplican de manera normal, exceptuando que las horas registradas no son exactas en su totalidad ni para la entrada ni para la salida. La bitácora está disponible en cualquier momento y no lo comprueban en el instante que están firmando, por lo que se desconocen las horas totales al día. Esta situación puede mejorarse con el sistema, al evitar situaciones confusas y garantizar que los empleados no se vean afectados por los malos cálculos que ocasionaba la bitácora, -al dar los salarios correspondientes.

La mayoría de los colaboradores desconocían o no tenían claro cómo funcionaba los cálculos correspondientes a los días de vacaciones. El sistema les facilita tanto el cálculo correcto como el poder consultar las veces que sean necesarias los días disponibles para las vacaciones y solicitar las fechas para tomarlos. La jefatura podrá observar y aceptar las fechas seleccionadas por el colaborador o llegar a un acuerdo para que las fechas sean convenientes para ambos.

Todos los empleados creen que una gestión de perfil e historial traería buenos beneficios en la empresa para tener disponibles los perfiles de usuario. Automatizar las consultas genera una ventaja importante al poder realizar consultas de los salarios anteriores como de las vacaciones o permisos solicitados a lo largo de la estadía laboral. Cabe recalcar que, al no tener un proceso simple, implementar los perfiles provoca una mejor experiencia a los colaboradores.

Todos los colaboradores creen que un sistema puede llegar a mejorar los procesos que se ejecutan tanto a la hora de calcular las horas trabajadas como para el pago correspondiente; también opinan que traería bastante beneficios para los colaboradores y la jefatura. Por otra parte, mencionaron que no han pensado implementar ningún sistema por temas de costos, pero si contemplan adquirir uno en el futuro, se podría cotizar para el uso de la empresa.

Con los datos recopilados de los instrumentos, el implementar el prototipo funcional proporcionará cambios significativos en la organización tanto al control del personal como a los procesos administrativos. Lo anterior por cuanto al ejecutar los procesos manuales, estos llegaban a ser tediosos o confusos. Una vez el prototipo funcione correctamente, la jefatura podrá dedicar el tiempo que empleaba en mejorar la estrategia de negocio y, a su vez, tomar decisiones que favorezcan al crecimiento laboral de la persona disponible. La empresa se beneficia al contar con un control más adecuado de la planilla sin generar pérdidas económicas o molestias en el personal, al mismo tiempo que agiliza cada proceso sin errores humanos.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Los objetivos establecidos para el desarrollo del prototipo se plantearon de acuerdo con las necesidades de la empresa; por la cual al terminar el proyecto se presentan las siguientes conclusiones:

Se plantearon los requerimientos de la organización para solventar la problemática encontrada. Al ejecutar correctamente los requerimientos descritos, la carga de trabajo de los colaboradores se verá aliviada, lo cual logrará evitar errores puntuales en la aplicación correcta de los procesos y permitirá que se administren mejor los tiempos en las labores cotidianas.

Además, gracias a un planteamiento claro y conciso, la arquitectura del prototipo es muy robusta y segura, lo cual es un punto fuerte para que la administración confíe en el proyecto, y así brindar las necesidades requeridas en los objetivos. El sistema cumple de manera eficaz y eficiente.

En el desarrollo del sistema, cada función implementada para la aplicación correcta de los recursos humanos es bastante sólida, ya que recopila la información guardada en la base de datos; esto permite que los usuarios puedan acceder a los registros deseados en cualquier momento y logren generar reportes en PDF para visualizar mejor la información. De esta manera, se puede analizar los reportes y mejorar la toma de decisiones.

Las pruebas realizadas en el prototipo cumplen cada una de las necesidades detalladas en la problemática presentada en la organización, con lo cual se logra el objetivo del proyecto. El sistema maneja correctamente las validaciones dentro del sistema, esto permite un control y acceso para los usuarios que podrá facilitar el uso de las funciones y la navegación del *software*.

En síntesis, tomando en cuenta lo anteriormente mencionado, se da por acabado de una manera grata el desarrollo del prototipo. Con esto se logró cumplir a cabalidad cada uno de los objetivos, para así poder brindarle un sistema robusto de la gestión de recursos humanos. Mediante la organización y la información recopilada, se logró desarrollar el sistema con gran calidad y un adecuado funcionamiento del prototipo.

Recomendaciones

Para la elaboración de un proyecto de tal alcance se recomienda una planificación exacta de las tareas por desarrollar, dividiéndolas en partes pequeñas para un manejo fácil de cada una de ellas. La mejor opción es utilizar una metodología ágil que permita implementar un cronograma de las tareas y los tiempos requeridos para que la organización sea efectiva y admita aplicar cambios de una manera fácil sin tener que atrasar las tareas del cronograma.

Se recomienda al dueño del restaurante la implementación del prototipo en un mes posterior a la entrega de esta propuesta, con una duración de tres meses, para que maneje las entradas y salidas de los colaboradores, ya que este fue uno de los puntos críticos de la problemática; de esta manera lograrán ver resultados en el uso del sistema. Una vez implementado, este dará garantías inmediatas del control de asistencia, con lo que se le facilitarán los procesos de planilla.

Aunque el prototipo tiene una interfaz sencilla e intuitiva y maneja pequeñas explicaciones de cada funcionalidad en las pantallas del sistema, se recomienda impartir pequeñas charlas de quince minutos por dos semanas para que los colaboradores comprendan de una mejor manera el funcionamiento de software y se adapten rápidamente y, así, ambas partes logren ver resultados a partir de los tres meses de la implementación del sistema.

Se recomienda a la administrador mejorar el control de asistencia con un lector de huella que se instale en el computador principal de la empresa para agilizar el proceso a más tardar una vez implementado el sistema, con una duración de dos meses.

Finalmente se recomienda al dueño migrar el sistema a la nube para manejar un respaldo de la información segura y que los usuarios puedan acceder en cualquier momento sin estar ubicados en el restaurante. Con fecha de inicio del 10 de enero del 2025 y una duración de mes y quince días. La organización deberá contratar un programador para la implementación y migración que duraría tres meses en gestionar las mejoras.

**UNIVERSIDAD INTERNACIONAL DE LAS
AMÉRICAS**

ESCUELA DE INGENIERÍA INFORMÁTICA

**PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE
RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE
LA VICTORIA, UBICADA EN SAN JOSÉ**

PROPUESTA

NOMBRE DEL ESTUDIANTE

Liber Gabriel Salas Barboza

ENERO, 2024

CAPÍTULO VI: PROPUESTA

Análisis

Análisis Detallado del Software por Desarrollar

A continuación, se detallará cada módulo por desarrollar en el prototipo, explicando el funcionamiento paso a paso de los procesos que se automatizarán para mejorar la eficiencia y rendimiento de la organización, y así lograr los objetivos planteados en el prototipo. También, se contemplarán los temas legales por cumplir en el momento de remunerar a los colaboradores por sus horas trabajadas.

Calcular Planilla

El calcular planilla es el módulo principal del prototipo, encargado de reunir toda la información que sale de los otros módulos, como el de control de asistencia, tramitar permisos, incapacidades o vacaciones que se utilizan para controlar los salarios de los empleados, evitando errores en los pagos de cada uno de los colaboradores. El módulo de planilla calcula las horas totales de la tabla control asistencia analizando las faltas de marca en el mes, revisando si la ausencia es justificada y pagada; después envía los datos al módulo calcular horas extras que le retorna el monto total de las horas trabajadas.

Calcular Vacaciones

Para el módulo de vacaciones se calcula el total de asistencias en la tabla de control de asistencia, pues la ley dicta que solo tendrá vacaciones el empleado que cumpla las 50 semanas seguidas laboradas; en caso de recibir pago mensual o quincenal, las vacaciones correspondientes serán de dos semanas y dos días de descanso, para un total de catorce días, independientemente el tiempo que labore. En caso contrario, si el contrato del empleado dice que obtendrá un día de vacación por mes laborado, el módulo lo calculará mensual, en caso de no completar las 50 semanas.

Calcular Incapacidades

Se requirió un módulo de incapacidades para precisar los días que la empresa pagará por ley y evitar que haya errores en el salario final del colaborador. El módulo tendrá el rango de fechas en el cual el empleado se ausentará de sus labores, con su debido documento emitido con la entidad de seguro. Se almacenará toda la información para que el administrador pueda realizar las consultas cuando sea necesario. Se calculará el pago los primeros tres días de incapacidad, es decir, que durante la incapacidad solo se le pagará el 50% del salario por esos tres días, ya que la Caja Costarricense del Seguro Social (CCSS) asumirá la remuneración del empleado desde el cuarto día de incapacidad.

Calcular Horas Extras

El cálculo horas extras solo se aplicará en el caso que la jefatura acepte que el colaborador labore más horas en el día para la finalización de su trabajo. El cálculo se realizará a partir del monto que se le paga al colaborador por hora y se multiplicará por 1.5 o se le pagará cada hora de más al 150% que se estableció previamente. En caso de no contar con el permiso de la jefatura, el pago de las horas extras no se verá reflejadas en el sistema.

Evaluar Empleados

La evaluación empleados es el módulo que va a medir el rendimiento de los colaboradores con un formulario de preguntas calificadas del 1 al 5; solamente los administradores y gerentes tienen acceso a aplicar y consultar las evaluaciones para la toma de decisiones para el bien del personal. Las evaluaciones se aplicarán mensualmente, para contar con un promedio por año del total de calificaciones.

Calcular Liquidación

Para calcular la liquidación de un empleado se debe tomar el dato del salario pendiente del mes de renuncia o despido, teniendo en cuenta que si hubo preaviso, se le otorgará un salario completo en caso de que haya trabajado más de un año; si no, se calculará

el salario correspondiente al período de preaviso, sumando el monto de las vacaciones no tomadas y calculando la cesantía, la cual sería por años completados o por meses trabajados.

Calcular Aguinaldo

Para el cálculo de aguinaldo se toma el periodo del primero de diciembre al treinta de noviembre; se debe tener en cuenta el pago de la jornada laboral, las horas extras y cualquier remuneración económica para el empleado. Se suman los salarios de cada mes laborado, así si se suman 6 salarios, el cálculo debería tomar el monto total de los salarios y dividir entre los meses de los salarios multiplicado 2; es decir, que para 6 meses de salarios deberá dividir el monto total entre los meses laborados para obtener el aguinaldo completo.

Control Asistencia

Respecto al manejo de control asistencia, el usuario podrá marcar solo una vez al día la entrada o la salida en el sistema. Si omite una marca, se guardará que la marca fue omitida con el fin de medir el rendimiento del colaborador y su compromiso con la empresa; no obstante, no va a recibir ninguna sanción económica, ya que la jefatura mencionó que no se va a sancionar económicamente a los trabajadores, si no marcan la entrada o salida en el sistema; pero sí tendrán un reporte de las marcas omitidas en el mes.

Análisis Detallado del Hardware Requerido

Tabla 3.

Hardware Requerido

Computador	Procesador	Memoria Ram	Tarjeta Gráfica	Sistema Operativo
Para el desarrollo	Ryzen 5 3600	16GB	AMD Radeon 5600xt	Windows 10
Para la implementación	I5 de octava generación	8GB	No tiene	Windows 10

Fuente: Elaboración Propia.

Se requirió una computadora con un procesador AMD Ryzen 5 3600, dos memorias RAM de 16GB, una gráfica AMD Radeon 5600xt con un sistema operativo de Windows 10. Si bien las especificaciones del computador son altas, el desarrollo del prototipo no requería de muchos recursos para su elaboración, ya que con un procesador de gama media baja como un I3 de octava generación con 8GB de RAM, corriendo en un entorno de Windows 10 sería suficiente para el desarrollo del sistema.

Con respecto a la implementación del prototipo, se realizará en un computador de gama media baja con un procesador I5 de quinta generación con gráficos integrados, 8GB de memoria RAM en un entorno de Windows 10. El motivo de hacer la implementación localmente es debido a que la organización no puede costear un almacenamiento del sistema y la base de datos en la nube. Por otro lado, tampoco disponen espacio para implementar un servidor local en las instalaciones debido a que es una organización pequeña.

Análisis Detallado de los Elementos Relacionados con las Telecomunicaciones

El tipo de red que se va a utilizar es de tipo LAN (Local Area Network), debido a que la implementación del prototipo será localmente y su función principal de registrar asistencia debe ser dentro de la organización. Y se requiere al menos 5MB de internet para el sistema, ya que se consumirá el api del Tribunal Supremo de Elecciones para obtener los datos de las personas. La empresa cuenta con una banda ancha de 20MB de subida y 20MB de bajada, esto es, usa internet simétrico, lo cual es un coste 0 porque ya contaba con el acceso a internet. Para la implementación del prototipo se utilizarán 3 puertos principales para el funcionamiento correcto del sistema. El puerto 4433 se define para la base de datos que se instalará en el computador principal, seguidamente del puerto 4048 que se dispondrá para el consumo del api local que se utiliza para el *BackEnd* y, por último, el puerto 8080 para la ejecución del *FrontEnd* que se alojará en el IIS del computador principal.

Descripción Detallada de las Herramientas Técnicas Utilizadas Para el Desarrollo

Las herramientas empleadas para el desarrollo del prototipo funcional fueron de uso gratuito, debido a que la organización no podía costear un motor de base de datos ni un

alojamiento en la nube para el sistema; por ende, se optó por utilizar SQL Server 2022 que proporciona un manejo de datos seguro, y para el alojamiento se propuso instalar el IIS en el computador principal de la empresa para acceder al sistema dentro de la empresa.

Tabla 4.

Software requerido

Software	Versión	Coste
SqlServer	2022	€0
Visual Studio Code	2022	€0
MySQL WorkBench	8.0	€0
IIS	Windows Server 2019	€0

Fuente: Elaboración Propia

Para el desarrollo del prototipo se requirió utilizar lenguajes en la parte web con distintos *frameworks* que proporcionan los lenguajes que mejoran gráficamente el sistema como para el tipo de letra, funciones que sincronicen la página para que no recargue al presionar dentro de la ventana y también para el tema responsivo con el cual se logra adaptar el prototipo a diferentes pantallas y dispositivos.

Tabla 5.

Lenguajes utilizados

Lenguaje	Framework	Versión
C#	.Net Core	
HTML	No	5
Java Script	Jquery	3.7.1
CSS	Bootstrap	5.0.2
CSS	Font Awesome	6.5.2

Fuente: Elaboración Propia

Descripción Detallada del Conocimiento Básico

Los colaboradores deberán tener conocimientos básicos para el inicio de sesión, ya que el sistema solo maneja a los usuarios por sesiones. Además de conocer las peticiones que pueden realizar dentro del prototipo con el fin de manejar correctamente la planilla. La organización comentó que los empleados trabajan con un *software* de facturación el cual emplean de manera efectiva; entonces, la implementación de la tecnología no retrasará los procesos que el prototipo plantea solucionar.

Casos de Uso

En la siguiente ilustración se muestra cómo los diferentes actores van a interactuar con el sistema. El autor llamado administrador podrá tener acceso a todas las funciones que muestra la imagen, con el fin de crear, eliminar o editar registros que necesite; también puede realizar reportes para la toma de decisiones. Y, por otro lado, el autor llamado colaborador solo tendrá acceso a la asistencia, los permisos, la seguridad, como el inicio de sesión y las consultas de salarios o permisos creados.

Ilustración 2.*Diagrama de Casos de Uso*

Fuente: Elaboración Propia

Caso de Uso 1

Con el caso de uso se plantea identificar y manejar el flujo de actividades y errores para el módulo calcular planilla que consultará en la base de datos la asistencia de los

colaboradores para calcular el salario exacto de cada uno, con el fin de automatizar el proceso de la empresa.

Tabla 6.*Caso de Uso 1 Calcular Planilla*

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 1	Nombre del Caso de Uso: Calcular Planilla
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	Se ejecuta el caso de uso para tomar la cantidad de horas en la tabla asistencia y calcula los salarios de los empleados con las deducciones
Autor caso de uso:	Liber Salas
Actores relacionados:	Gerente
Precondiciones:	Debe haber registros en la tabla asistencia
Flujo Básico del caso de uso	
El gerente presiona el botón “generar planilla” lo que genera que se haga el cálculo de los salarios de cada empleado con las deducciones correspondiente, devolviendo una respuesta de “planilla creada” con la fecha correspondiente.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El autor presiona el botón “generar planilla”. 2. El sistema consulta en la tabla de asistencia las horas de cada empleado. 3. El sistema calcula las horas laboradas de cada empleado con sus horas extras. 4. El sistema calcula el salario de por las horas trabajadas y rebaja las deducciones, la CCSS y la renta. 5. El sistema crea en la tabla planilla cada registro por empleado. 6. El sistema devuelve una respuesta positiva al autor. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	El sistema genera una alerta de “la planilla ya fue creada” en caso de haber creado previamente la planilla del mes

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
	correspondiente.
Flujo Alternativo No. 12	El sistema genera una alerta de “no hay registros en asistencia” en caso de consultar a la tabla asistencia y no haya ningún registro del mes actual, no se genera la planilla.
Requerimientos especiales	
No hay.	
Post-Condiciones	
El gerente podrá consultar la planilla del mes actual o anteriores.	

Fuente: Elaboración Propia

Caso de Uso 2

El caso de uso 2 busca solucionar la problemática con las horas extras, debido a una inexactitud de los registros en la bitácora, consultando directamente desde la base de datos las horas de los trabajadores en todo mes, calculando las horas mínimas quedando el cálculo de las horas extras para el salario.

Tabla 7.

Caso de Uso 2 Calcular horas extras

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 2	Nombre del Caso de Uso: Calcular horas extras
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	Toma los registros de la tabla asistencia y realiza el cálculo de las horas laboradas y resta las horas de la jornada normal,

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
	obteniendo las horas extras.
Autor caso de uso:	Liber Salas
Actores relacionados:	Gerente
Precondiciones:	Se debe activar el caso de uso 1 para ejecutar el caso de uso 2.
Flujo Básico del caso de uso	
Cuando se activa el caso de uso 1, se procederá a calcular las horas trabajadas en la tabla asistencia, se restará las horas de la jornada normal, lo cual da como resultado las horas extras de los colaboradores, que se hará el cálculo para el salario correspondiente, y se devolverá el monto extra en el salario de la planilla.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El autor activa el caso de uso 1. 2. Se procede a consultar los registros en la tabla asistencia. 3. Se calcula las horas de los colaboradores. 4. Se resta las horas de la jornada normal y devuelve el cálculo de las horas extras. 5. Retorna el cálculo de las horas extras a la planilla. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	1. De no haber horas extras, no se hará registro de las horas extras.
Requerimientos especiales	
No hay.	
Post-Condicion	
El gerente podrá ver las horas extras de los colaboradores.	

Fuente: Elaboración Propia

Caso de Uso 3

Para evitar el registro manual de los empleados, se plantea manejar un sistema de registro en la base de datos con el fin de mejorar y controlar las entradas y salidas de los colaboradores, facilitando el cálculo de salarios para la planilla.

Tabla 8.*Caso de Uso 3 Control de asistencia*

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 3	Nombre del Caso de Uso: Control de asistencia
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El colaborador entrará en el sistema y presionará el botón de entrada o el botón de salida para hacer el registro de asistencia.
Autor caso de uso:	Liber Salas
Actores relacionados:	Empleado
Precondiciones:	Haber iniciado sesión en el sistema
Flujo Básico del caso de uso	
El autor presiona el botón entrada o el botón salida para registrarlo, de no haber registro previo, el sistema ingresa el registro a la tabla asistencia y devuelve respuesta de ingreso con éxito o registro existente.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 7. El autor inicia sesión. 8. Presiona el botón de entrada o salida. 9. El sistema, válida si hay registro del día actual. 10. El sistema registra la entrada o salida en la tabla Asistencia. 11. Alerta que se hizo correctamente el ingreso o por contraparte alerta que ya se hizo registro previo. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	2. El sistema alertará que ya se registró previamente la entrada en el sistema.
Flujo Alternativo No. 2	3. El sistema alertará que ya se registró

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
	previamente la salida en el sistema.
Requerimientos especiales	
No hay.	
Post-Condiciones	
El usuario podrá iniciar sesión.	

Fuente: Elaboración Propia.

Caso de Uso 4

El caso 4 permite poder evaluar a cada empleado mensualmente, midiendo el rendimiento optimo en las labores empeñadas como también su responsabilidad en el cumplimiento de horario; se le asigna una calificación para toma de decisiones de la jefatura.

Tabla 9.

Caso de Uso 4 Evaluar Empleado

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 4	Nombre del Caso de Uso: Evaluar Empleado
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El gerente abrirá el formulario de evaluación, donde asignará al colaborador correspondiente, completando los campos requeridos.
Autor caso de uso:	Liber Salas
Actores relacionados:	Gerente
Precondiciones:	Tener registros de empleados en el sistema.

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Flujo Básico del caso de uso	
El gerente presionará el botón “crear evaluación”, donde se abrirá un formulario para evaluar a los empleados, el gerente selecciona el empleado y completa los campos del formulario.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El autor crea la evaluación. 2. Selecciona el colaborador para la evaluación. 3. Completa los campos necesarios de la evaluación. 4. Registra el formulario en la tabla evaluación. 5. Alerta de registro exitoso. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	El sistema alertará que no hay empleados para la evaluación.
Requerimientos especiales	
No hay.	
Post-Condiciones	
El gerente podrá observar las calificaciones de los empleados.	

Fuente: Elaboración Propia.

Caso de Uso 5

El caso de uso calcular vacaciones tomará los registros del control de asistencia, para lo cual calculará los días registrados para obtener los días de vacaciones disponibles, cálculo realizado de acuerdo con lo establecido por la ley de 1.5 días por mes laborado.

Tabla 10.*Caso de Uso 5 Calcular vacaciones*

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 5	Nombre del Caso de Uso: Calcular vacaciones
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El usuario presiona el botón “calcular vacaciones”, el sistema cuenta los meses trabajados y calcula los días que posee el usuario disponible.
Autor caso de uso:	Liber Salas
Actores relacionados:	Usuario
Precondiciones:	Haber iniciado sesión y tener al menos un mes de registro en planilla.
Flujo Básico del caso de uso	
El usuario selecciona el botón “calcular vacaciones”, de ser usuario tipo gerente, el sistema calculará y mostrará los días disponibles de todos los empleados registrados en el sistema, en caso de ser tipo empleado, el sistema solo mostrará su cálculo correspondiente. De no tener un mes registrado, alertará que no tiene días disponibles.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El autor presiona el botón de “calcular vacaciones”. 2. El sistema calcula los meses trabajados por 1.5. 3. El sistema muestra en pantalla los días disponibles del empleado. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	De no tener registro de al menos un mes, el sistema alertará que no tiene días disponibles.
Flujo Alternativo No. 2	La solicitud la hará un usuario tipo gerente, mostrará los días disponibles de todos los colaboradores.

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Flujo Alternativo No. 3	La solicitud la hará un usuario tipo usuario, mostrará sus días disponible.
Requerimientos especiales	
El sistema debe calcular las vacaciones de 1.5 días por mes trabajado.	
Post-Condiciones	
El usuario podrá consultar de nuevo las vacaciones disponibles.	

Fuente: Elaboración Propia

Caso de Uso 6

El caso de uso calcular aguinaldo consulta los registros en la planilla para realizar el cálculo de cada uno de los empleados que tengan registros en la tabla planilla, después del cálculo se registra cada monto en la tabla aguinaldo con los datos correspondientes.

Tabla 11.

Caso de Uso 6 Calcular aguinaldo

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 6	Nombre del Caso de Uso: Calcular aguinaldo
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El sistema calculará el aguinaldo mediante la tabla planilla con los últimos seis meses para calcular el promedio de cada empleado.
Autor caso de uso:	Liber Salas
Actores relacionados:	Gerente
Precondiciones:	Tener registros en planilla

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Flujo Básico del caso de uso	
El autor selecciona “calcular aguinaldo” y el sistema consulta la tabla planilla para generar el cálculo de cada uno de los colaboradores e insertarlo en la tabla aguinaldo, devuelve una alerta de éxito y muestra los aguinaldos de los colaboradores.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El autor selecciona el botón “calcular aguinaldo”. 2. El sistema consulta los registros de planilla. 3. El sistema calcula el aguinaldo de cada empleado y los registra. 4. El sistema dará una alerta de éxito en los registros o de duplicidad de datos. 5. El sistema muestra en pantalla el cálculo de los colaboradores. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	Alerta de duplicidad en el cálculo de aguinaldo
Requerimientos especiales	
No hay.	
Post-Condiciones	
El usuario podrá observar la tabla de aguinaldos del año correspondiente.	

Fuente: Elaboración Propia

Caso de Uso 7

El caso uso 7 solicita los datos necesarios para ejecutar el cálculo de la liquidación que le corresponde al empleado, ya sea por caso de despido o renuncia.

Tabla 12.

Caso de Uso 7 Calcular liquidación

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 7	Nombre del Caso de Uso: Calcular liquidación
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El gerente presiona selecciona calcular liquidación y rellena los campos solicitados, el sistema hace el cálculo correspondiente y lo muestra en pantalla.
Autor caso de uso:	Liber Salas
Actores relacionados:	Gerente
Precondiciones:	Haber registros en la tabla planilla
Flujo Básico del caso de uso	
El autor presiona el botón “calcular liquidación”, completa el formulario con los campos para generar la liquidación, el sistema calcula liquidación y la ingresa en la tabla liquidación, mostrando en pantalla el monto final de la liquidación.	
Detallar el paso a paso del Flujo Básico	
12. El autor selecciona el calcular liquidación. 13. El autor marca el colaborador, los días de preaviso y motivo de la renuncia o despido. 14. El sistema calcula el monto de liquidación. 15. Ingresa el registro en la tabla liquidación. 16. Lo muestra en pantalla.	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	4.
Requerimientos especiales	

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ
No hay.
Post-Condiciones
El gerente podrá observar el detalle de la liquidación.

Fuente: Elaboración Propia

Caso de Uso 8

El caso de uso para el inicio de sesión, valida las credenciales de los usuarios permitiendo el acceso al inicio del sistema, en caso de validar correctamente en el sistema; de no ser el caso, el sistema alertará de credenciales inválidas o, por el contrario, bloqueará al usuario.

Tabla 13.

Caso de Uso 8 Iniciar sesión

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 8	Nombre del Caso de Uso: Iniciar sesión
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El usuario ingresará las credenciales para el inicio de sesión; de ser correctas el sistema ingresará a la pantalla de inicio.
Autor caso de uso:	Liber Salas
Actores relacionados:	Usuario
Precondiciones:	Acceder al sistema
Flujo Básico del caso de uso	

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
El autor ingresa las credenciales y presiona el botón “iniciar sesión”, el sistema validará el usuario y contraseña del usuario; de ser incorrectos mostrará una alerta de “credenciales incorrectas”, de fallar 3 veces el sistema bloqueará el usuario por intentos fallidos, de validar correctamente las credenciales, el sistema le mostrará al usuario la pantalla de asistencia.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El autor ingresa las credenciales. 2. El sistema valida las credenciales, de ser incorrectas alertará de credenciales incorrectas; de no ser así, ingresará al inicio. 3. El sistema muestra la pantalla de asistencia. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	El sistema alerta de credenciales incorrectas.
Flujo Alternativo No. 2	El sistema bloquea al usuario por intentos fallidos.
Flujo Alternativo No. 3	El sistema alerta al usuario de usuario bloqueado.
Requerimientos especiales	
No hay.	
Post-Condiciones	
El usuario accederá a las funciones del sistema.	

Fuente: Elaboración Propia

Caso de uso 9

El caso de uso 9, el usuario podrá realizar una solicitud de permiso para un día en específico, describiendo el tipo de permiso. Una vez creado el sistema notificará a jefatura

para la aprobación del permiso. Si la solicitud es aceptada, el sistema notificará recursos humanos para su debida confirmación, de caso contrario el permiso será eliminado de no ser aceptado por alguno de los 2 departamentos.

Tabla 14.

Caso de uso trámite de permiso

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 9	Nombre del Caso de Uso: Trámite de permiso
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El usuario ingresará los datos en el formulario de permiso y podrá crear la solicitud para ser aprobada por jefatura y RRHH.
Autor caso de uso:	Liber Salas
Actores relacionados:	Usuario, RRHH, JEFATURA
Precondiciones:	Acceder al sistema
Flujo Básico del caso de uso	
El usuario realiza la petición del permiso a la espera de ser aceptada por jefatura y RRHH. El sistema notifica que hay una solicitud pendiente. Si jefatura rechaza el permiso el sistema elimina la solicitud, en caso contrario, le notifica a RRHH para ser acepta o denegada.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El usuario ingresa el formulario con los campos necesarios. 2. El sistema crea la solicitud y notifica a jefatura. 3. Jefatura acepta el permiso y el sistema notifica a RRHH, sino el permiso se elimina de ser denegado. 4. RRHH recibe la solicitud y la aprueba o lo deniega. 	
Sub-Flujos	

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Flujos Alternos	
Flujo Alternativo No. 1	El sistema alerta de campos incorrectos.
Flujo Alternativo No. 2	El sistema notifica a jefatura.
Flujo Alternativo No. 3	El sistema notifica a RRHH.
Requerimientos especiales	
No hay.	
Post-Condiciones	
El usuario podrá observar las peticiones aceptadas.	

Fuente: Elaboración Propia

Caso de uso 10

Solo los usuarios con el tipo de recursos humanos podrán generar las incapacidades de los colaboradores que hayan llevado el comprobante médico para la incapacitación inmediata. Una vez generada el usuario verá la incapacidad desde el usuario propio.

Tabla 15.

Caso de uso incapacidad

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Número Caso de Uso: 10	Nombre del Caso de Uso: Incapacidad
Fecha elaboración:	06/05/2024
Descripción Caso de Uso:	El usuario de RRHH podrá crear las incapacidades con los documentos oficiales del usuario.

Prototipo: PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE RECURSOS HUMANOS EN LA EMPRESA RESTAURANTE LA VICTORIA, UBICADA EN SAN JOSÉ	
Autor caso de uso:	Liber Salas
Actores relacionados:	RRHH
Precondiciones:	Acceder al sistema
Flujo Básico del caso de uso	
El colaborador de RRHH podrá llenar el formulario de la incapacidad subiendo al sistema el comprobante del usuario.	
Detallar el paso a paso del Flujo Básico	
<ol style="list-style-type: none"> 1. El usuario ingresa el formulario con los campos necesarios. 2. El sistema notifica la creación de la incapacidad. 3. El usuario de la incapacidad podrá ver sus incapacidades creadas por RRHH. 	
Sub-Flujos	
Flujos Alternos	
Flujo Alternativo No. 1	El sistema inserta la incapacidad.
Flujo Alternativo No. 2	El sistema notifica que la incapacidad se guardó con éxito.
Flujo Alternativo No. 3	Las incapacidades se muestran al usuario.
Requerimientos especiales	
No hay.	
Post-Condiciones	
El usuario podrá observar las incapacidades creadas.	

Fuente: Elaboración Propia

Diseño

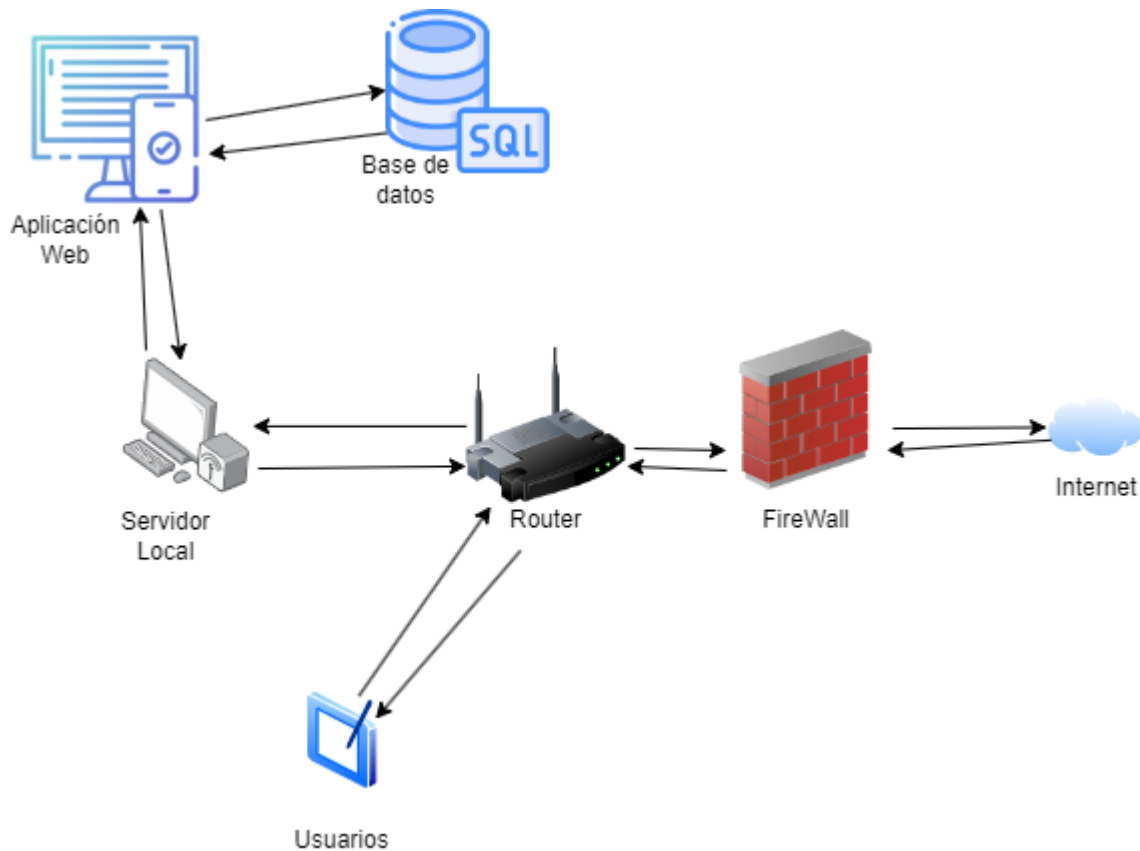
Arquitectura del Sistema

El diagrama de la siguiente imagen muestra cómo se ve la infraestructura donde se implementará el prototipo funcional. El acceso a la red local del *router* será el conector para que los usuarios se puedan comunicar con el sistema, ya que como la base de datos y el desarrollo web están en el computador principal, con el acceso a internet se podrá comunicar el sistema con el Gmail y el consumo de un api que se utiliza en la creación de personas.

Los usuarios solo tendrán acceso al prototipo cuando se encuentren dentro de las instalaciones del restaurante, debido a que la implementación del sistema en un servidor local solo permitirá el ingreso de dispositivos en la misma red y los puertos de acceso en el servidor, están cerrados para que no haya ningún acceso al sistema fuera de la red propia de la empresa.

Ilustración 3.

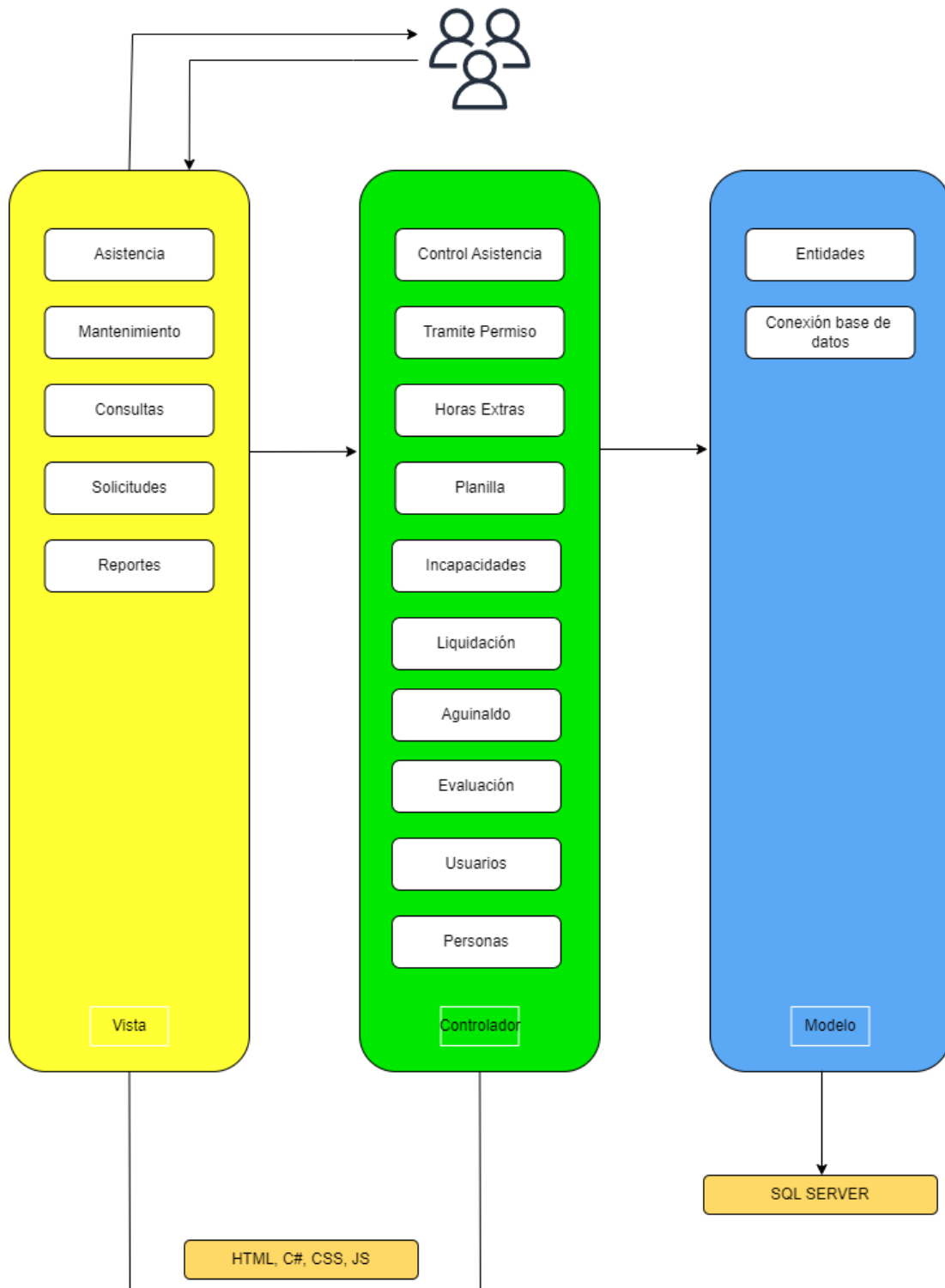
Arquitectura del sistema



Fuente: Elaboración Propia

Arquitectura del Software

En relación con la arquitectura del sistema, como muestra la siguiente imagen, se observa que el prototipo está basado en la arquitectura MVC (Modelo, Vista, Controlador), que son los encargados de conectar al usuario con la base de datos para realizar todas las funciones de todo el sistema. Para la vista se utilizó .Net Framework que se basa en Razor que está conformado por los lenguajes HTML, CSS y JavaScript que se conectará con C# al servidor SQL Server 2022.

Ilustración 4.*Arquitectura del sistema*

Fuente: Elaboración Propia

Diseño de Entradas

Registrar Asistencia

Ilustración 5.

Registrar Asistencia

Registrar Asistencia



Fuente: Elaboración Propia

Crear Incapacidad

Ilustración 6.

Crear incapacidad

The image shows a web form titled "Crear Incapacidad" (Create Disability). The form is set against a light beige background. It contains the following elements:

- Colaborador:** A text input field containing "Guillermo 1 1".
- Fecha_Inicio:** A date input field with the placeholder "dd/mm/aaaa --:--" and a calendar icon on the right.
- Fecha_Final:** A date input field with the placeholder "dd/mm/aaaa --:--" and a calendar icon on the right.
- ComprobanteFile:** A file selection area with a button labeled "Elegir archivo" and the text "No se ha seleccionado ningún archivo".
- Create:** A light green button at the bottom left of the form.
- Volver:** An orange button at the bottom center of the form.

Fuente: Elaboración Propia

Registro de Usuario

Ilustración 7.

Crear Usuario

Crear Usuario

 **Personales**  **Direccion**  **Contacto**  **Familia**  **Final**

Datos Personales

Cédula

Nombre

Apellido1

Apellido2

Fecha de entrada

15/07/2024 10:21 

Tipo de horario

08:00:00-16:00:00

Tipo de cargo

Salonero

Siguiente

Fuente: Elaboración Propia

Trámite de Permisos

Ilustración 8.

Crear Permiso



The image shows a web form titled "Crear Permiso" (Create Permission) on a light beige background. The form is contained within a white rounded rectangle. It has the following fields and elements:

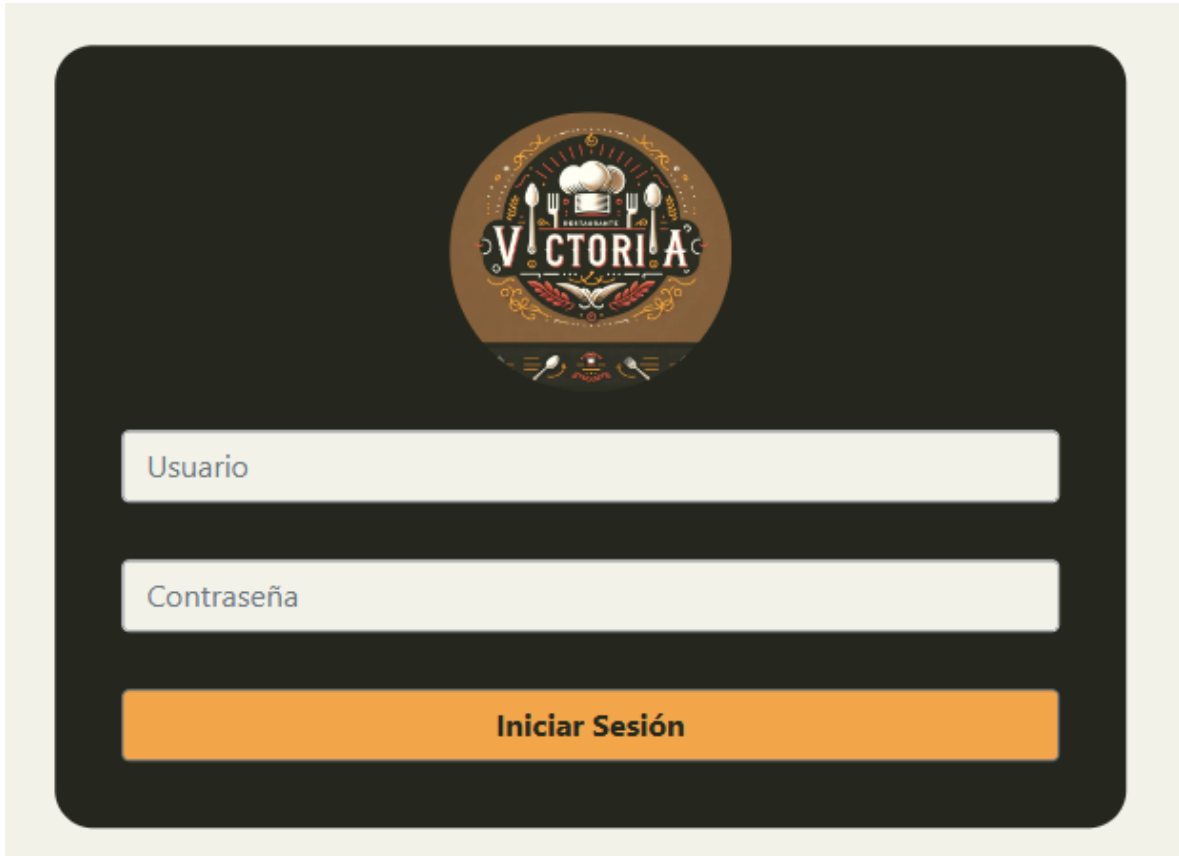
- Título:** "Crear Permiso" in large black font at the top center.
- Tipo de permiso:** A text input field containing "Asunto Persona".
- Descripción:** A text input field that is currently empty.
- Fecha:** A date and time input field with the placeholder "dd/mm/aaaa --:--" and a calendar icon on the right.
- Botones:** A light green "Crear" button at the bottom left of the form, and an orange "Volver" button centered below the form.

Fuente: Elaboración Propia

Inicio de Sesión

Ilustración 9.

Iniciar Sesión

The image shows a login interface for a restaurant named 'Victoria'. At the top center is a circular logo with a chef's hat, a fork, and a knife, with the word 'VICTORIA' in a stylized font. Below the logo are three input fields: the first is labeled 'Usuario', the second is labeled 'Contraseña', and the third is a large orange button labeled 'Iniciar Sesión'. The entire form is set against a dark background with rounded corners.

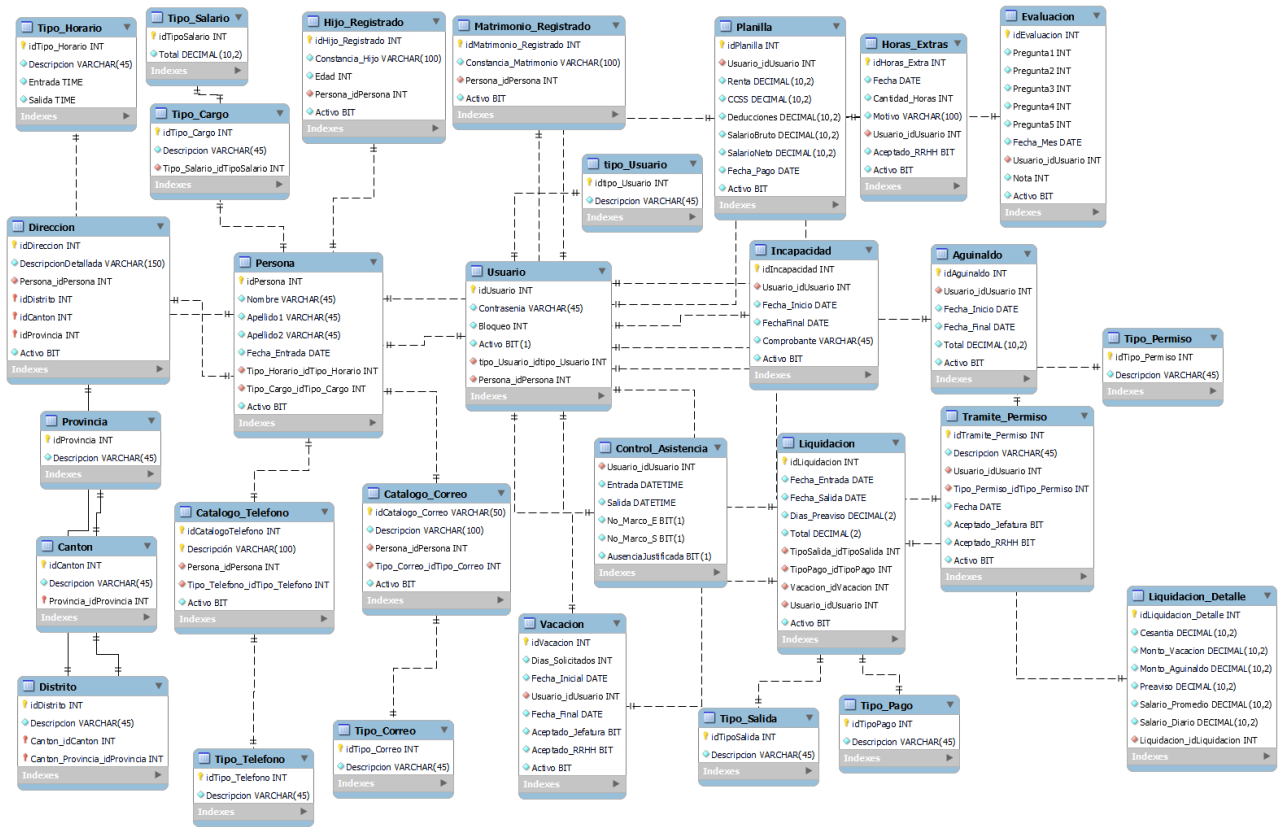
Fuente: Elaboración Propia

Diseño Físico de la Base de Datos

El diagrama de base de datos muestra las tablas necesarias para lograr de la mejor manera el manejo de la información de los colaboradores y los procesos de recursos humanos. En el diseño se hallan como tablas principales las siguientes: Dirección, Personas, Usuarios, Control Asistencia, Planilla, Evaluación, Liquidación, Trámite Permisos, Aguinaldo e Incapacidades. Y como tablas secundarias: Provincia, Cantón, Distrito y los catálogos de Usuario, Permiso, Salida, Pago, Correo, Teléfono, Cargo, Horario y Salario.

Ilustración 10.

Diagrama de base de datos



Fuente: Elaboración Propia

Diccionario de Datos

A continuación, se muestran todas las tablas con sus respectivas celdas y sus tipos de campo. En un total de 26 tablas para la elaboración del diagrama para la base de datos.

Tabla 16.*Diccionario de datos Persona*

Persona						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idPersona	INT	Cedula de la persona	-	NN	Sí	Primary Key
Nombre	VARCHAR	Nombre de la persona	45	NN	Sí	
Apellido1	VARCHAR	Primer apellido de la persona	45	NN	Sí	
Apellido2	VARCHAR	Segundo apellido de la persona	45	NN	Sí	
Fecha_Entrada	DATE	Día de la entrada de la persona	-	NN	Sí	
Tipo_Horario_idTipo_Horario	INT	El horario de la persona	-	NN	Sí	Foreing Key
Tipo_Cargo_idTipo_Cargo	INT	El puesto de la persona	-	NN	Sí	Foreing Key
Activo	BIT	Estado de la persona	-	NN	Sí	

*Fuente: Elaboración Propia***Tabla 17.***Diccionario de datos usuario*

Usuario						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idUsuario	INT	Identificación del usuario	-	NN		Primary Key
Contrasenia	VARCHAR	Contraseña del usuario	45	NN		
Bloqueo	INT	Intentos de entrada	-	NN		
Activo	BIT	Estado del usuario	-	NN	Sí	
tipo_Usuario_idtipo_Usuario	INT	El tipo de perfil del usuario	-	NN	Sí	Foreing Key
Persona_idPersona	INT	Cedula de la persona	-	NN	Sí	Foreing Key

Fuente: Elaboración Propia

Tabla 18.*Diccionario de datos de Vacación*

Vacacion						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idVacacion	INT	Identificador de vacación	-	NN		Primary Key
Dias_Solicitados	INT	Días solicitados	-	NN		
Fecha_Inicial	DATE	Día inicial de la vacación	-	NN		
Fecha_Final	DATE	Día final de la vacación	-	NN		
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreing Key
Aceptado_Jefatura	BIT	Estado de la vacación por Jefatura	-	NN		
Aceptado_RRHH	BIT	Estado de la vacación por RRHH	-	NN		
Activo	BIT	Estado de la vacación	-	NN	Sí	

*Fuente: Elaboración Propia***Tabla 19.***Diccionario de datos control de asistencia*

Control_Asistencia						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreing Key
Entrada	DATETIME	Hora de entrada	-	NN		
Salida	DATETIME	Hora de salida	-	NN		
No_Marco_E	BIT	Entrada marcada	-	NN		
No_Marco_S	BIT	Salida marcada	-	NN		
Ausencia_Justificada	BIT	Ausencia se justifico	-	NN	Sí	

Fuente: Elaboración Propia

Tabla 20.*Diccionario de datos liquidación*

Liquidacion						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idLiquidacion	INT	Identificador de liquidación	-	NN		Primary Key
Fecha_Entrada	DATE	Dia de entrada	-	NN	Sí	
Fecha_Salida	DATE	Fecha de salida	-	NN	Sí	
Dias_Preaviso	DECIMAL	Días de preaviso	2	NN		
Total	DECIMAL	Total de la liquidación	10,2	NN		Foreing Key
TipoSalida_idTipoSalida	INT	Identificador del tipo de salida	-	NN		Foreing Key
TipoPago_idTipoPago	INT	Identificador del tipo de pago	-	NN		Foreing Key
Vacacion_idVacacion	INT	Identificador de las vacaciones	-	NN		Foreing Key
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreing Key
Activo	BIT	Estado de la liquidación	-	NN	Sí	

Fuente: Elaboración Propia

Tabla 21.*Diccionario de datos evaluación*

Evaluacion						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idEvaluacion	INT	Identificación de la evaluación	-	NN		Primary Key
Pregunta1	INT	Primer pregunta de la evaluación	-	NN		
Pregunta2	INT	Segunda pregunta de la evaluación	-	NN		
Pregunta3	INT	Tercera pregunta de la evaluación	-	NN		
Pregunta4	INT	Cuarta pregunta de la evaluación	-	NN		
Pregunta5	INT	Quinta pregunta de la evaluación	-	NN		
Fecha_Mes	DATE	Fecha de la evaluación	-	NN	Sí	
Nota	INT	Nota de la evaluación	-	NN		
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreing Key
Activo	BIT	Estado de la evaluación	-	NN	Sí	

*Fuente: Elaboración Propia***Tabla 22.***Diccionario de datos planilla*

Planilla						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idPlanilla	INT	Identificador de la planilla	-	NN		Primary Key
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreing Key
CCSS	DECIMAL	Porcentaje de la caja	10,2	NN		
Renta	DECIMAL	Porcentaje de la renta	10,2	NN		
Deducciones	DECIMAL	Deducciones del salario	10,2	NN		
Salario_Bruto	DECIMAL	Total de salario bruto	10,2	NN		
Salario_Neto	DECIMAL	Total del salario neto	10,2	NN		
Fecha_Pago	DATE	Día de pago	-	NN		
Activo	BIT	Estado de la planilla	-	NN	Sí	

Fuente: Elaboración Propia

Tabla 23.*Diccionario de datos aguinaldos*

Aguinaldo						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idAguinaldo	INT	Identificador de Aguinaldo	-	NN		Primary Key
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreing Key
Fecha_Inicio	DATE	Inicio de pago	-	NN	Sí	
Fecha_Final	DATE	Finalización de pago	-	NN		
Total	DECIMAL	Total de aguinaldo	10,2	NN		
Activo	BIT	Estado del aguinaldo	-	NN	Sí	

*Fuente: Elaboración Propia***Tabla 24.***Diccionario de datos incapacidad*

Incapacidad						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idIncapacidad	INT	Identificador de la incapacidad	-	NN		Primary Key
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreing Key
Fecha_Inicio	DATE	Inicio de la incapacidad	-	NN		
Fecha_Final	DATE	Finalización de la incapacidad	-	NN		
Comprobante	VARCHAR	Nombre del comprobante de la incapacidad	45	NN		
Activo	BIT	Estado de la incapacidad	-	NN	Sí	

Fuente: Elaboración Propia

Tabla 25.*Diccionario de datos trámite permiso*

Tramite_Permission						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idTramite_Permission	INT	Identificador del permiso	-	NN		Primary Key
Descripcion	VARCHAR	Descripción del permiso	45	NN		
Usuario_idUsuario	INT	Identificación del usuario	-	NN	Sí	Foreign Key
Tipo_Permission_idTipo_Permission	INT	Identificador del tipo de permiso	-	NN		
Fecha	DATE	Fecha del permiso	-	NN	Sí	
Aceptado_Jefatura	BIT	Estado del permiso por Jefatura	-	NN		
Aceptado_RRHH	BIT	Estado del permiso por RRHH	-	NN		
Activo	BIT	Estado del permiso	-	NN	Sí	

*Fuente: Elaboración Propia***Tabla 26.***Diccionario de datos tipo horario*

Tipo_Horario						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idTipo_Horario	INT	Identificador del horario	-	NN		Primary Key
Descripcion	VARCHAR	Descripción del horario	45	NN		
Entrada	TIME	Entrada del horario	-	NN		
Salida	TIME	Salida del horario	-	NN		

Fuente: Elaboración Propia

Tabla 27.*Diccionario de datos tipo cargo*

Tipo_Cargo						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idTipo_Cargo	INT	Identificador del cargo	-	NN		Primary Key
Descripcion	VARCHAR	Descripción del cargo	45	NN		
Tipo_Salario_idTipoSalario	INT	Identificador del salario		NN		Foreing Key

*Fuente: Elaboración Propia***Tabla 28.***Diccionario de datos tipo salario*

Tipo_Salario						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idTipo_Salario	INT	Identificador del salario	-	NN		Primary Key
Total	DECIMAL	Total del salario	10,2	NN		

*Fuente: Elaboración Propia***Tabla 29.***Diccionario de datos tipo usuario*

Tipo_Usuario						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idtipo_Usuario	INT	Identificador del usuario	-	NN		Primary Key
Descripcion	VARCHAR	Descripción del usuario	45	NN		

Fuente: Elaboración Propia

Tabla 30.*Diccionario de datos tipo salida*

Tipo_Salida						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idtipo_Salida	INT	Identificador de la salida	-	NN		Primary Key
Descripcion	VARCHAR	Descripción de la salida	45	NN		

*Fuente: Elaboración Propia***Tabla 31.***Diccionario de datos tipo pago*

Tipo_Pago						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idtipo_Pago	INT	Identificador del pago	-	NN		Primary Key
Descripcion	VARCHAR	Descripción del pago	45	NN		

*Fuente: Elaboración Propia***Tabla 32.***Diccionario de datos tipo permiso*

Tipo_Permission						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idtipo_Permission	INT	Identificador del permiso	-	NN		Primary Key
Descripcion	VARCHAR	Descripción del permiso	45	NN		

*Fuente: Elaboración Propia***Tabla 33.***Diccionario de datos dirección*

Direccion						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idDireccion	INT	Identificador de dirección	-	NN		Primary Key

Direccion						
Descripcion_Detallada	VARCHAR	Descripción de la dirección	150	NN		
Persona_idPersona	INT	Cedula de la persona	-	NN	Sí	Foreing Key
Provincia_idProvincia	INT	Identificador de provincia	-	NN		Foreing Key
idCanton	INT	Identificador de cantón	-	NN		Primary Key
idDistrito	INT	Identificador de Distrito	-	NN		Primary Key
Activo	BIT	Estado de la dirección	-	NN	Sí	

Fuente: Elaboración Propia

Tabla 34.

Diccionario de datos provincia

Provincia						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idProvincia	INT	Identificador de provincia	-	NN	Sí	Primary Key
Descripcion	VARCHAR	Descripción de la provincia	45	NN		

Fuente: Elaboración Propia

Tabla 35.

Diccionario de datos cantón

Canton						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idCanton	INT	Identificador de cantón	-	NN	Sí	Primary Key
Descripcion	VARCHAR	Descripción del cantón	45	NN		
Provincia_idProvincia	INT	Identificador de distrito	-	NN		Foreing Key

Fuente: Elaboración Propia

Tabla 36.*Diccionario de datos distrito*

Distrito						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idDistrito	INT	Identificador de Distrito	-	NN	Sí	Primary Key
Descripcion	VARCHAR	Descripción del Distrito	45	NN		
Canton_idCanton	INT	Identificador del cantón	-	NN		
Canton_Provincia_idProvincia	INT	Identificador de la provincia	-	NN		

*Fuente: Elaboración Propia***Tabla 37.***Diccionario de datos catálogo teléfono*

Catalogo_Telefono						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idCatalogo_Telefono	INT	Identificación del teléfono	-	NN	Sí	Primary Key
Descripcion	VARCHAR	Descripción del teléfono	45	NN		
Tipo_Telefono_idTipo_Telefono	INT	Identificador del tipo de teléfono				
Persona_idPersona	INT	Cedula de la persona	-	NN		Foreing Key
Activo	BIT	Estado del teléfono	-	NN	Sí	

Fuente: Elaboración Propia

Tabla 38.*Diccionario de datos catálogo correo*

Catalogo_Correo						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idCatalogo_Correo	INT	Identificación de correo	-	NN	Sí	Primary Key
Descripcion	VARCHAR	Descripción del correo	45	NN		
Tipo_Correo_idTipo_Correo	INT	Identificador del tipo de correo	-	NN		
Persona_idPersona	INT	Cedula de la persona	-	NN	Sí	Foreing Key
Activo	BIT	Estado del correo	-	NN	Sí	

*Fuente: Elaboración Propia***Tabla 39.***Diccionario de datos tipo teléfono*

Tipo_Telefono						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idTipo_Telefono	INT	Identificación de correo	-	NN	Sí	Primary Key
Descripcion	VARCHAR	Descripción del correo	45	NN		

*Fuente: Elaboración Propia***Tabla 40.***Diccionario de datos tipo correo*

Tipo_Correo						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idTipo_Correo	INT	Identificación de correo	-	NN	Sí	Primary Key
Descripcion	VARCHAR	Descripción del correo	45	NN		

Fuente: Elaboración Propia

Tabla 41.*Diccionario de datos matrimonio registrado*

Matrimonio_Registrado						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idMatrimonio_Registrado	INT	Id del matrimonio registrado	-	NN	Sí	Primary Key
Constancia_Matrimonio	VARCHAR	Nombre de la constancia	100	NN		
Persona_idPersona	INT	Cedula de la persona	-	NN		Foreing Key
Activo	BIT	Estado del matrimonio registrado	-	NN	Sí	

*Fuente: Elaboración Propia***Tabla 42.***Diccionario de datos hijo registrado*

Hijo_Registrado						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idHijo_Registrado	INT	Id del hijo registrado	-	NN	Sí	Primary Key
Constancia_Hijo	VARCHAR	Nombre de la constancia de nacimiento	100	NN		
Edad	INT	Edad del hijo	-	NN		
Persona_idPersona	INT	Cedula de la persona	-	NN		Foreing Key
Activo	BIT	Estado del hijo registrado	-	NN	Sí	

Fuente: Elaboración Propia

Tabla 43.*Diccionario de datos liquidación detalle*

Liquidacion_Detalle						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idLiquidacion_Detalle	INT	Identificador del detalle de la liquidación	-	NN	Sí	Primary Key
Cesantia	DECIMAL	Monto de la cesantía	10,2	NN		
Monto_Vacacion	DECIMAL	Monto de las vacaciones	10,2	NN		
Monto_Aguinaldo	DECIMAL	Monto del aguinaldo	10,2	NN		
Preaviso	DECIMAL	Monto del preaviso	10,2	NN		
Salario_Promedio	DECIMAL	Monto del salario promedio	10,2	NN		
Salario_Diario	DECIMAL	Monto del salario diario	10,2	NN		
Liquidacion_idLiquidacion	INT	Identificador de la liquidación	-	NN		

*Fuente: Elaboración Propia***Tabla 44.***Diccionario de datos horas extras*

Horas_Extras						
Columna	Tipo Dato	Descripción	Tamaño	Nulos	Índice	Detalle
idHoras_Extras	INT	Identificador de horas extras	-	NN	Sí	Primary Key
Fecha	DATE	Fecha de las horas extras	-	NN		
Cantidad_Horas	INT	Cantidad de horas	-	NN		
Motivo	VARCHAR	Motivo de las horas extras	100	NN		
Usuario_idUsuario	INT	Identificador del usuario	10,2	NN		
Aceptado_RRHH	BIT	Aceptado por RRHH	-	NN		
Activo	BIT	Estado de las horas extras	-	NN		

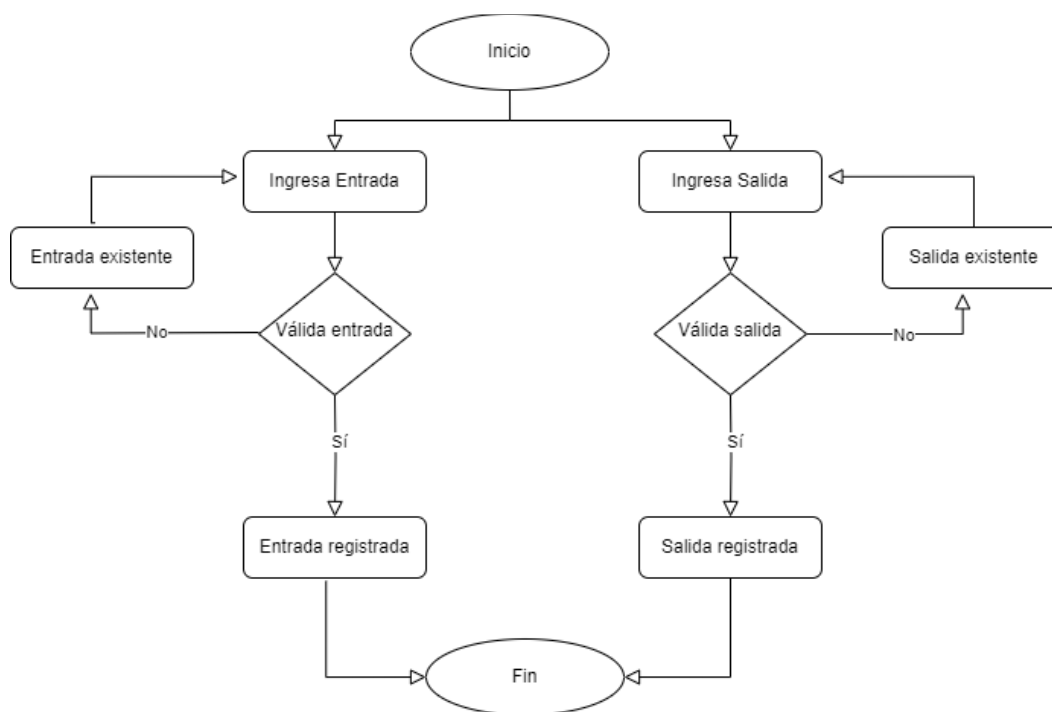
Fuente: Elaboración Propia

Diseño de procesos

A continuación, se muestran cada uno de los diagramas de flujo para los módulos planteados en el alcance funcional.

Ilustración 11.

Diagrama para control de asistencia



Fuente: Elaboración Propia

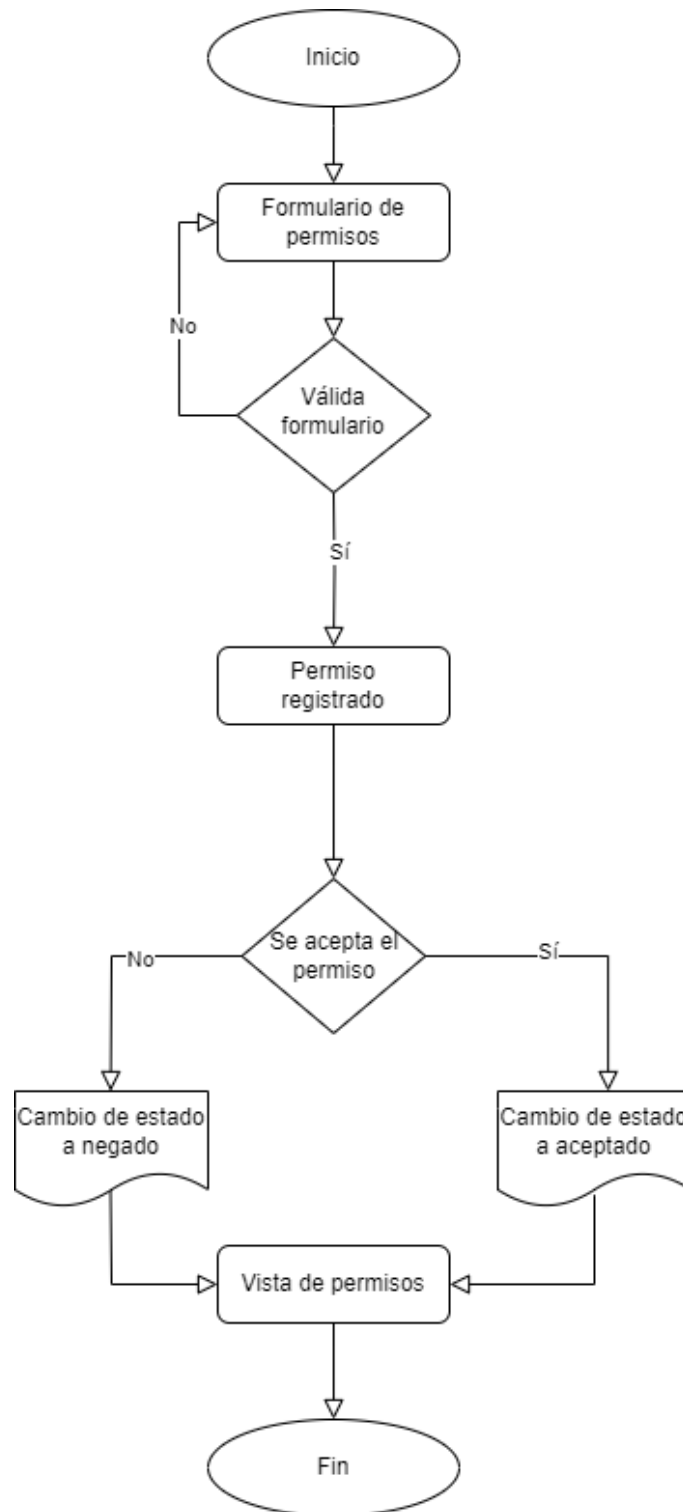
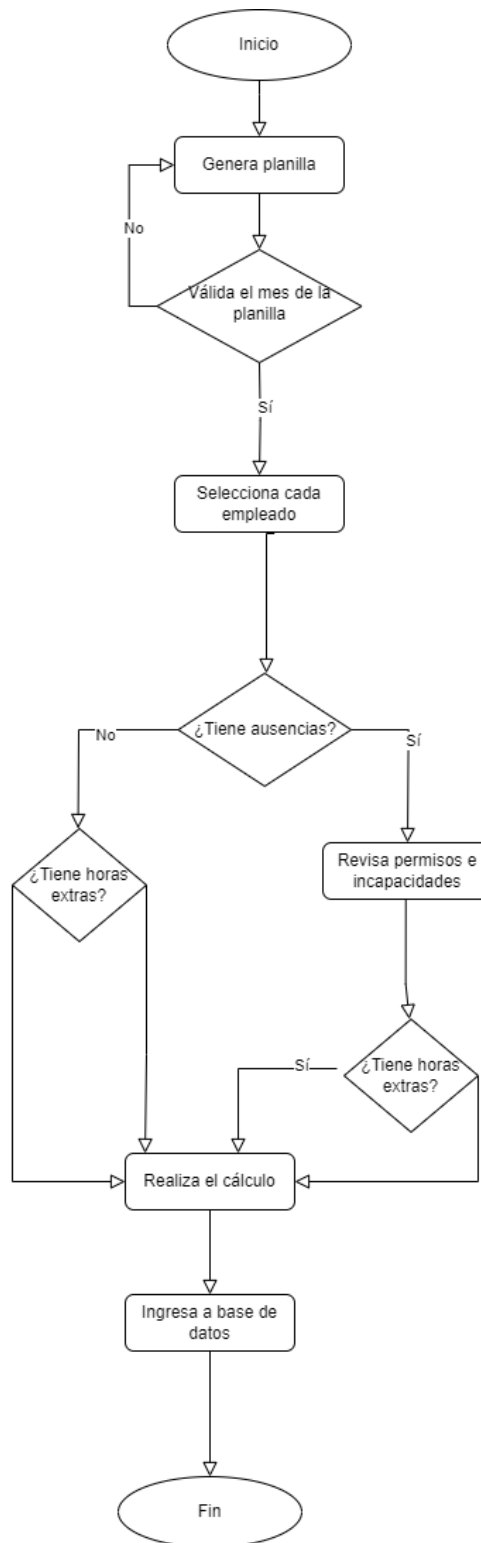
Ilustración 12.*Diagrama de trámite de permisos**Fuente: Elaboración Propia*

Ilustración 13.

Diagrama de evaluación de empleado



Fuente: Elaboración Propia

Ilustración 14.*Diagrama de planilla*

Fuente: Elaboración Propia

Ilustración 15.*Diagrama de aguinaldo**Fuente: Elaboración Propia*

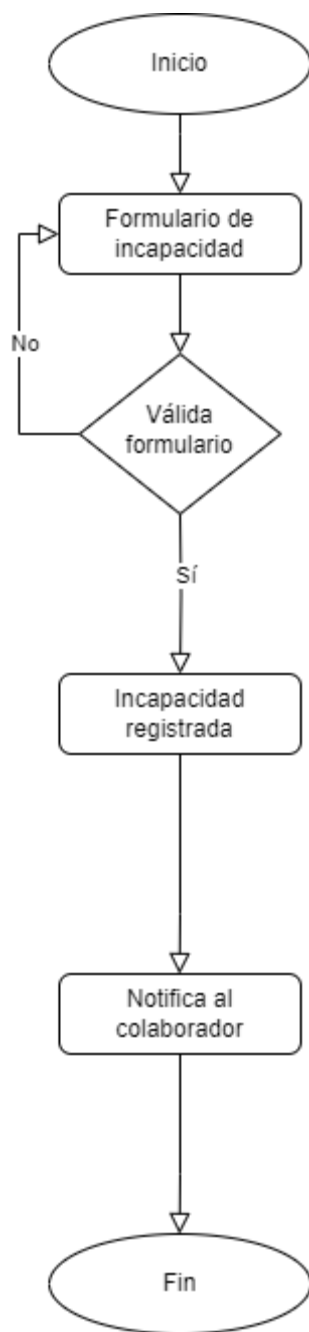
Ilustración 16.*Diagrama de incapacidad**Fuente: Elaboración Propia*

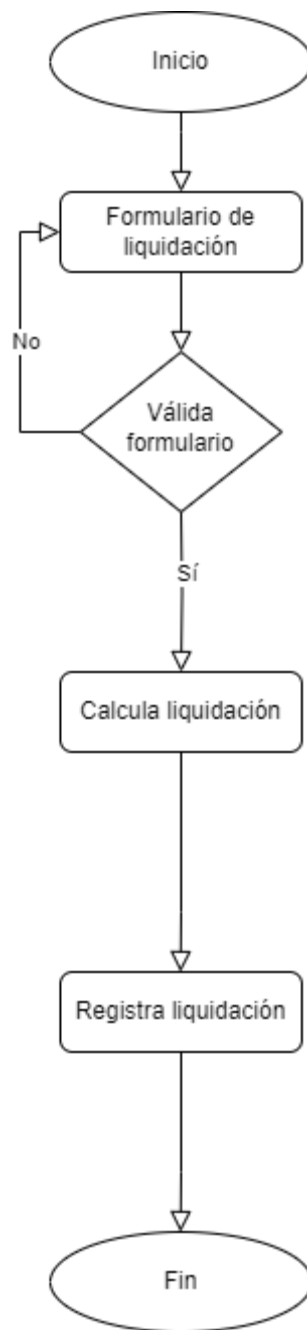
Ilustración 17.*Diagrama de liquidación**Fuente: Elaboración Propia*

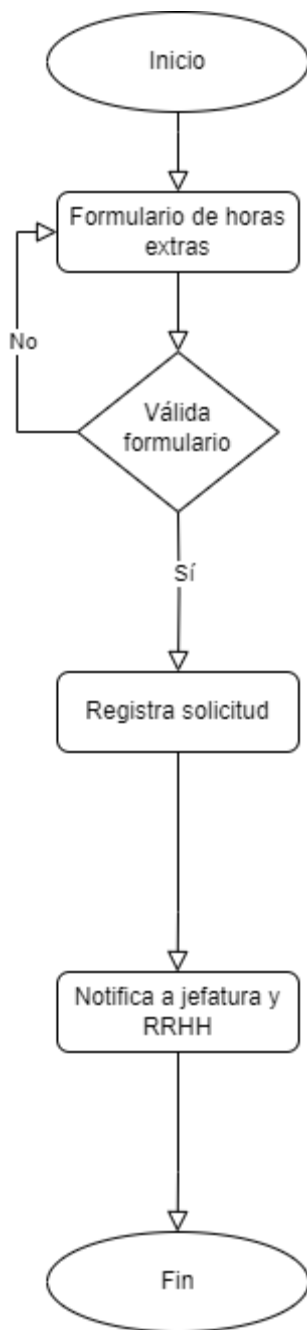
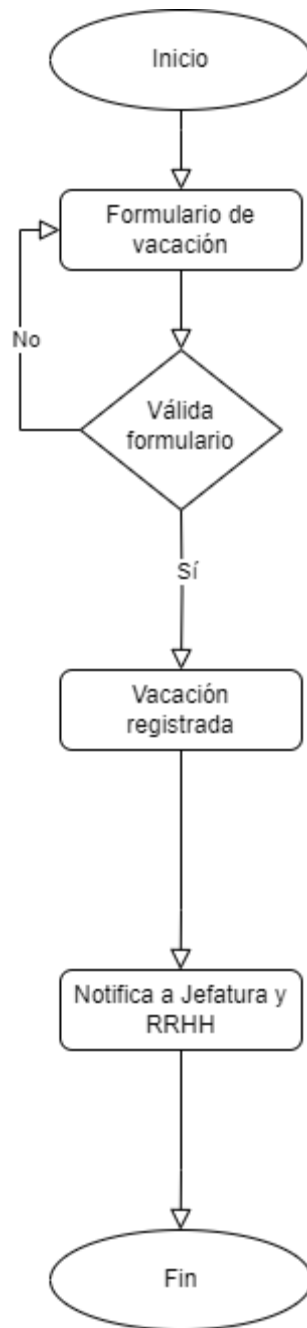
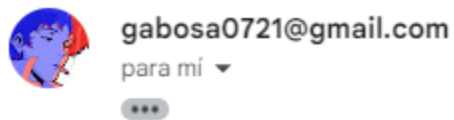
Ilustración 18.*Diagrama de horas extras**Fuente: Elaboración Propia*

Ilustración 19.*Diagrama de vacaciones*

Fuente: Elaboración Propia

Diseño de salidas

A continuación, se muestran algunas pantallas de la salida de datos que proporciona el software en los distintos módulos.

Ilustración 20.*Salida de correo***Resultado de la evaluación del mes: julio**

Su nota es:56

Su mes es: Debe Mejorar

*Fuente: Elaboración Propia***Ilustración 21.***Salida de horas extras*

Solicitud de Horas Extras

[Pedir Horas Extras](#)

PDF Buscar:

Fecha	Horas solicitadas	Colaborador	Motivo	Salario x Hora	Total de extras	Acciones
17/07/2024 0:00:00	2	Azul Castro Castro	Salón de mesas lleno	1520	4560	✓ ❌ i

Mostrando 1 a 1 de 1 entradas Primero Anterior **1** Siguiente Último

Fuente: Elaboración Propia

Ilustración 22.*Salida de permisos*

Permisos pendientes

[Crear Permiso](#)

PDF Buscar:

Colaborador	Tipo de permiso	Descripción	Fecha	Acciones
Guillermo 1 1	Reunión Escolar	1	06/07/2024 0:00:00	✓
LIBER GABRIEL SALAS BARBOZA	Reunión Escolar	1	14/07/2024 0:00:00	✓

Mostrando 1 a 2 de 2 entradas Primero Anterior 1 Siguiente Último

*Fuente: Elaboración Propia***Ilustración 23.***Salida de evaluaciones*

Evaluaciones

[Crear nueva evaluación](#)

PDF Buscar:

Fecha_Mes	Colaborador	Nota	Acciones
06/07/2024 0:00:00	Guillermo 1 1	56	
11/07/2024 0:00:00	Guillermo 1 1	64	

Mostrando 1 a 2 de 2 entradas Primero Anterior 1 Siguiente Último

Fuente: Elaboración Propia

Ilustración 24.*Salida de PDF de liquidación*

Index

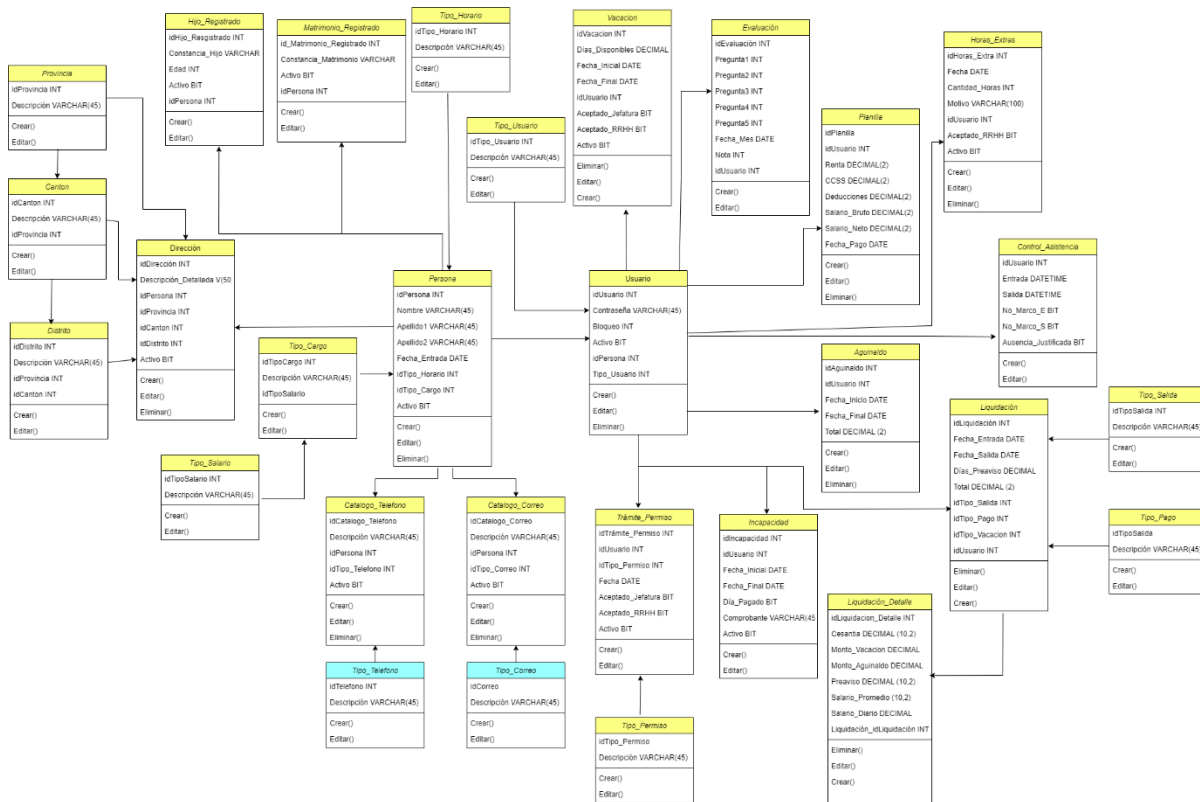
Fecha Entrada	Fecha Salida	Preaviso	Total	Salida	Tipo de pago	Colaborador
01/01/2023 0:00:00	31/05/2024 0:00:00	1	2700000,00	Despido con responsabilidad patronal	Quincenal/ mensual	Guillermo 1 1
01/01/2023 0:00:00	31/05/2024 0:00:00	1	2633333,33	Despido con responsabilidad patronal	Quincenal/ mensual	Guillermo 1 1
01/01/2023 0:00:00	31/05/2024 0:00:00	1	1416663,33	Despido sin responsabilidad patronal	Quincenal/ mensual	Guillermo 1 1

*Fuente: Elaboración Propia***Diagramas UML**

A continuación, se muestra el diagrama UML con todos los objetos creados para el funcionamiento correcto del prototipo.

Ilustración 25.

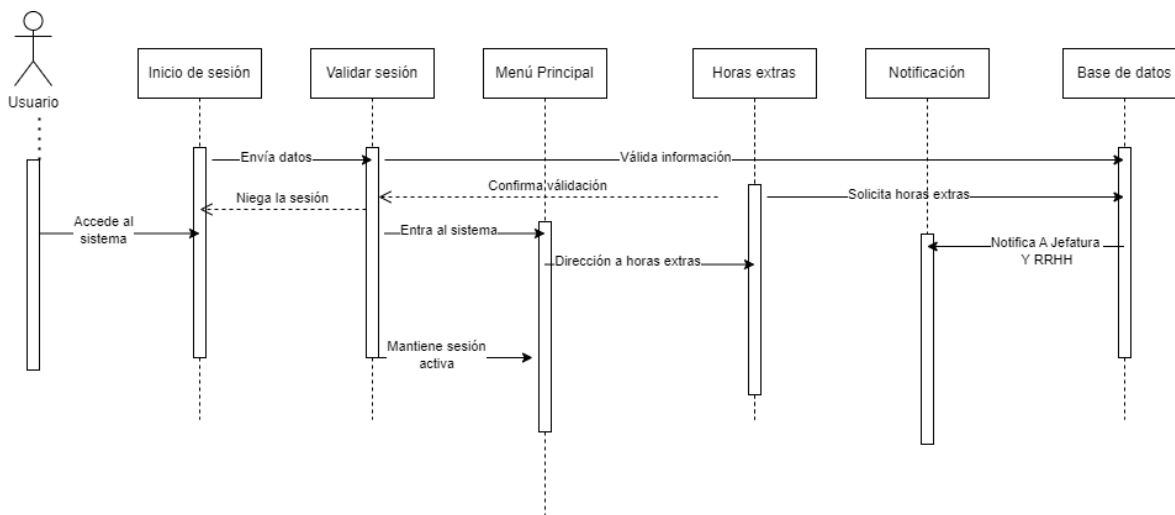
Diagrama UML



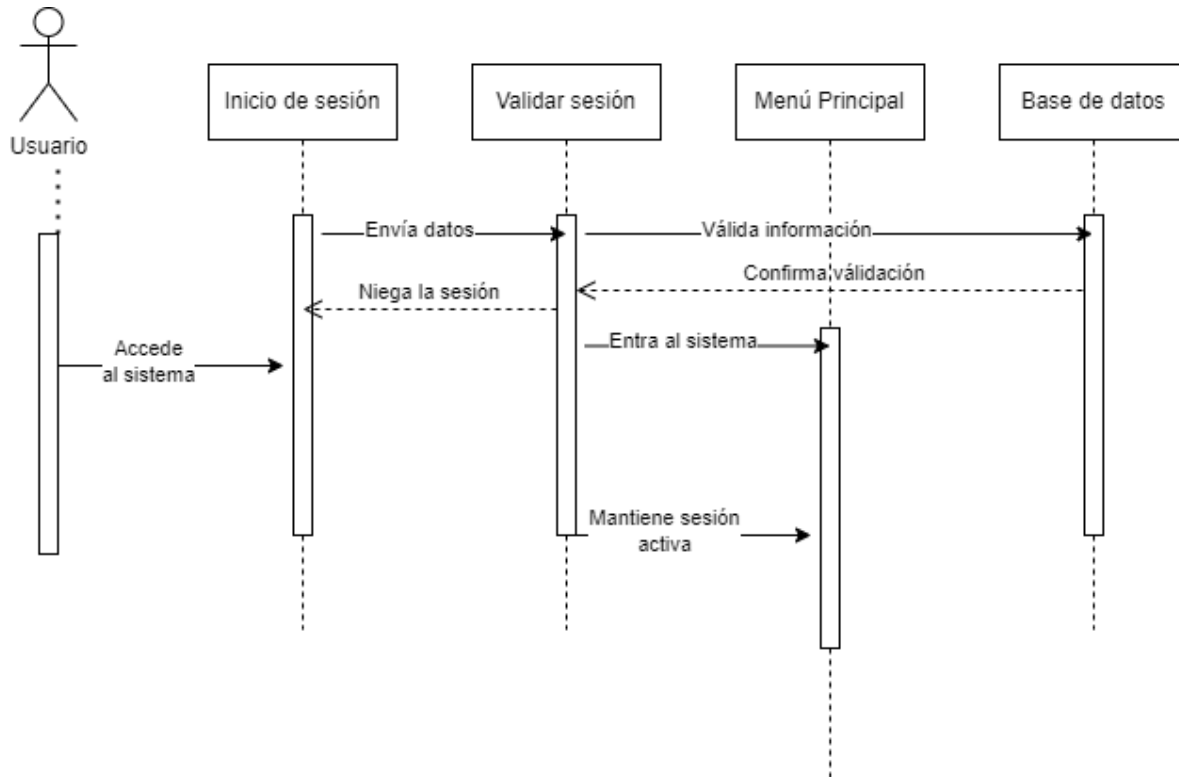
Fuente: Elaboración Propia

Ilustración 26.

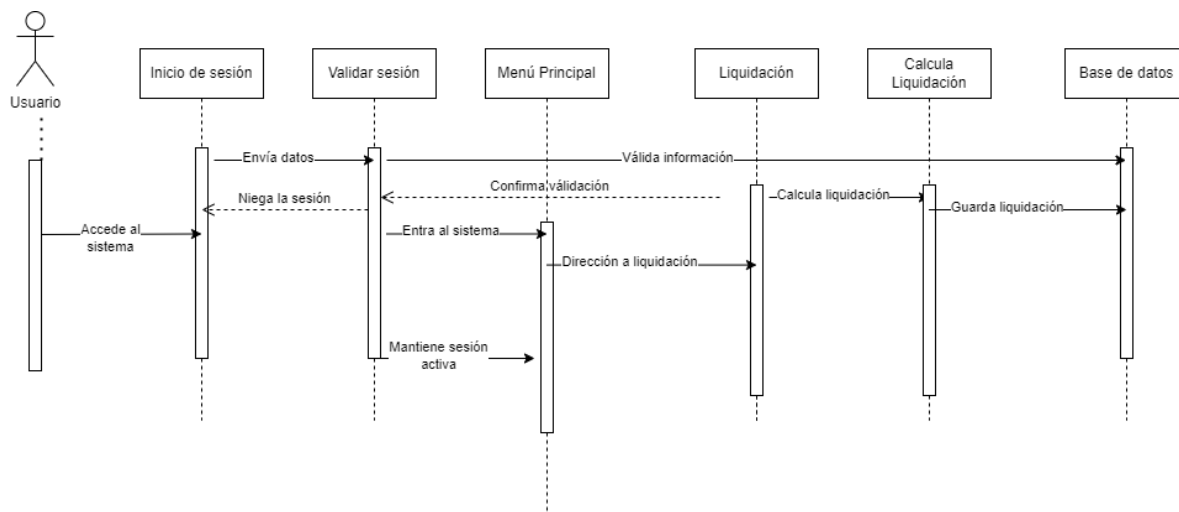
Diagrama de secuencia horas extra



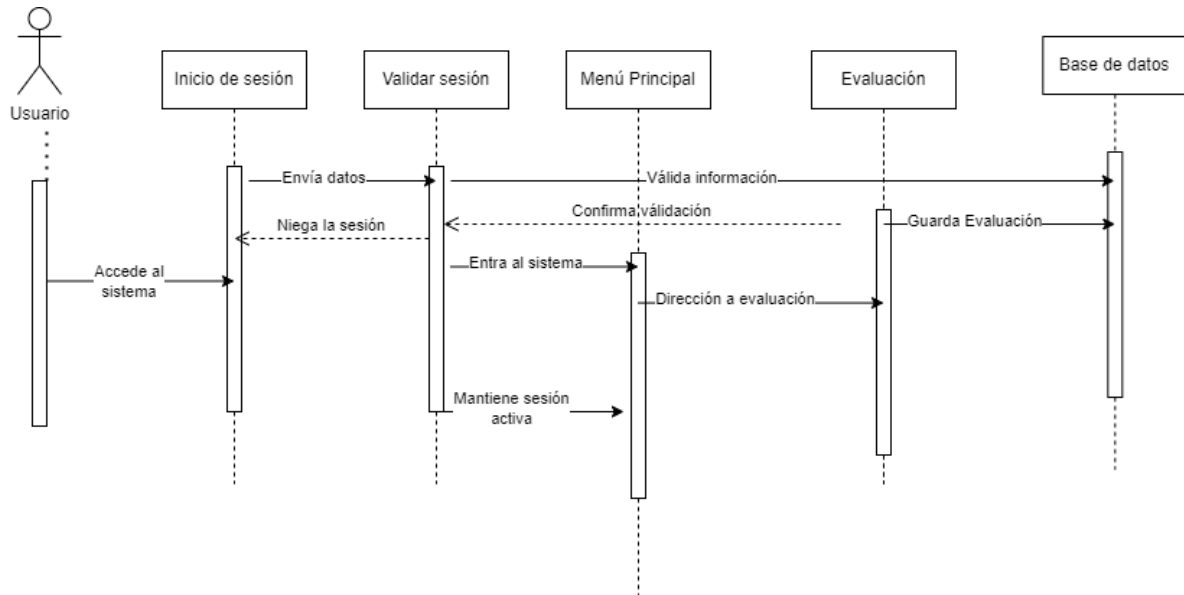
Fuente: Elaboración Propia

Ilustración 27.*Diagrama de secuencia control de asistencia*

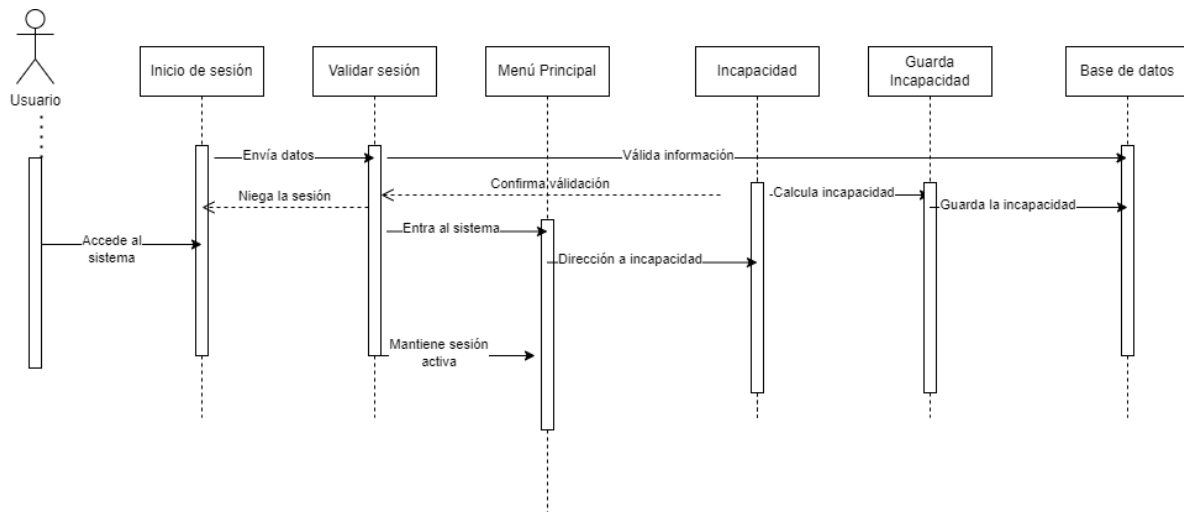
Fuente: Elaboración Propia

Ilustración 28.*Diagrama de secuencia liquidación*

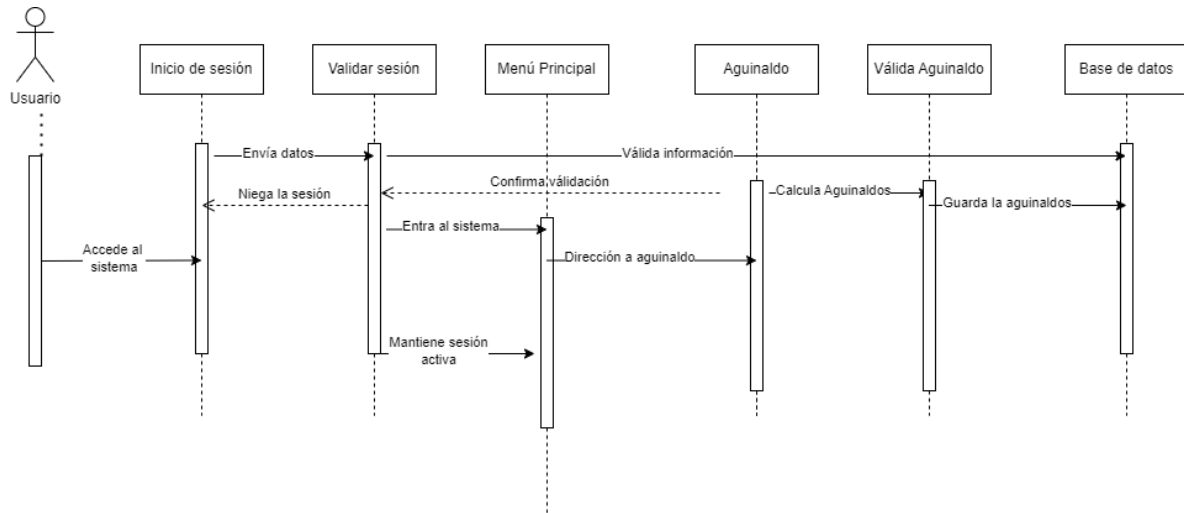
Fuente: Elaboración Propia

Ilustración 29.*Diagrama de secuencia evaluación*

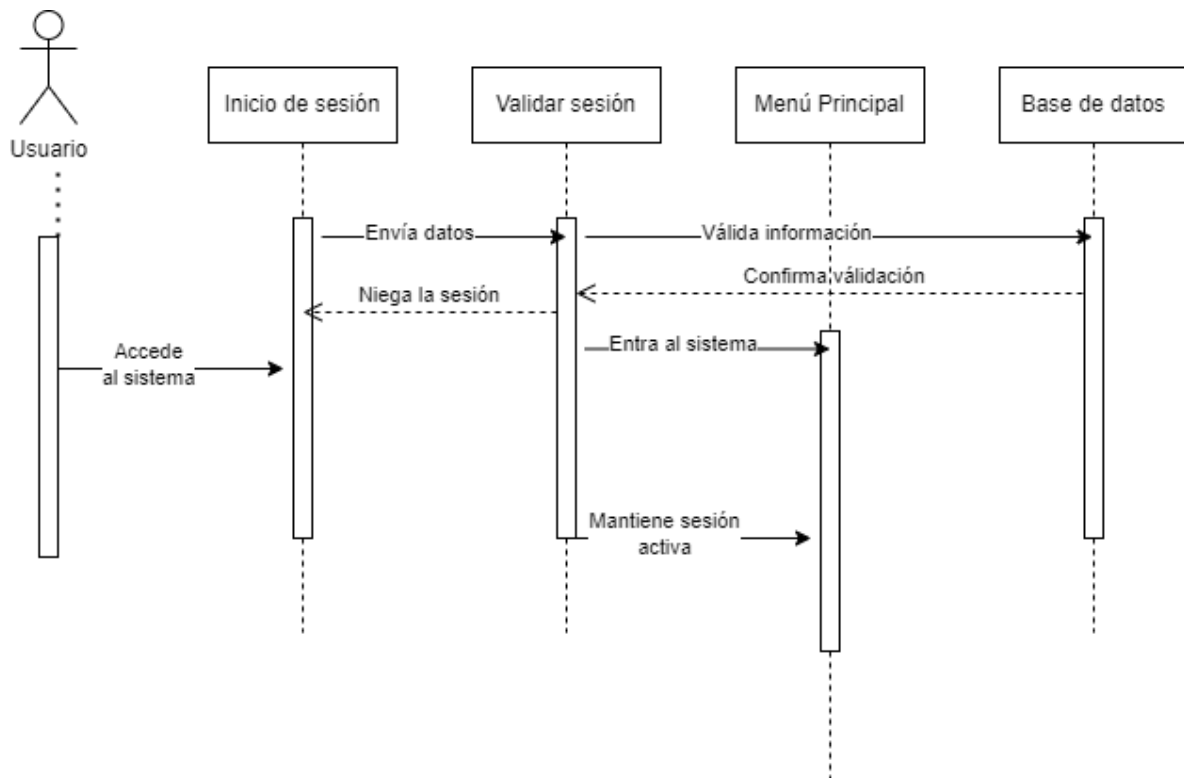
Fuente: Elaboración Propia

Ilustración 30.*Diagrama de secuencia incapacidad*

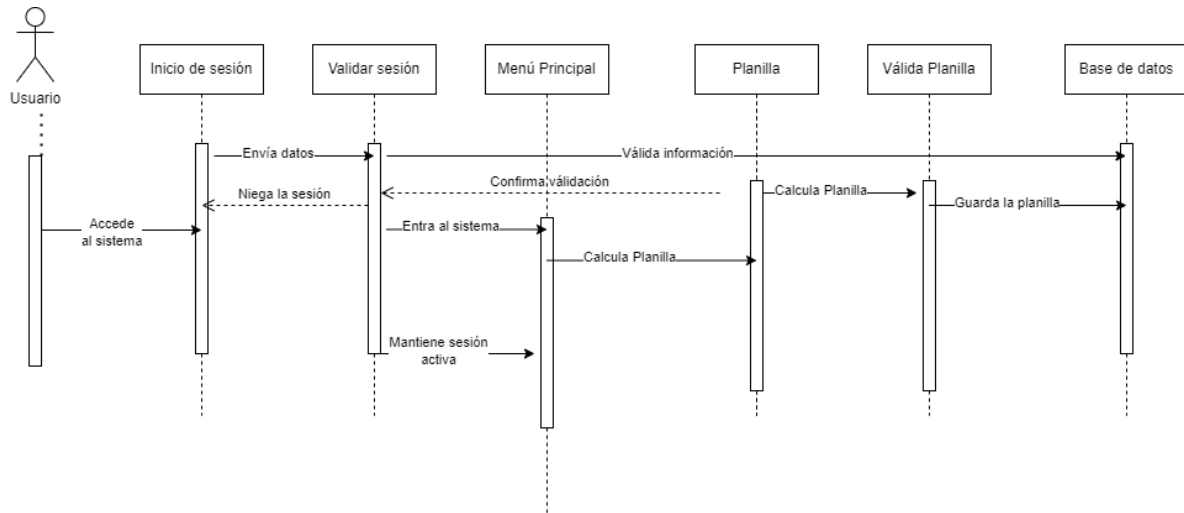
Fuente: Elaboración Propia

Ilustración 31.*Diagrama de secuencia aguinaldo*

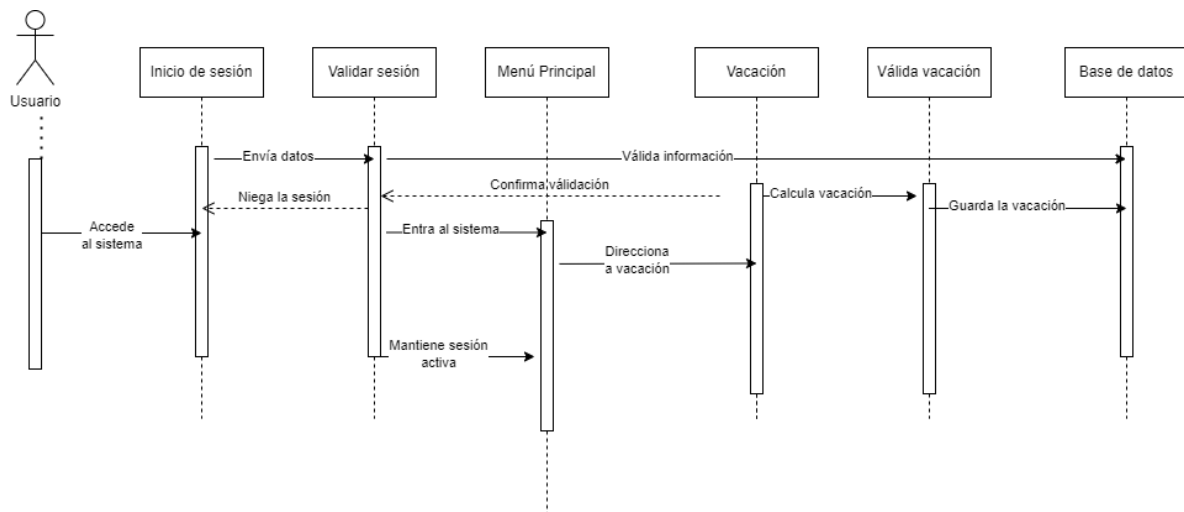
Fuente: Elaboración Propia

Ilustración 32.*Diagrama de secuencia de iniciar sesión*

Fuente: Elaboración Propia

Ilustración 33.*Diagrama de secuencia Planilla*

Fuente: Elaboración Propia

Ilustración 34.*Diagrama de secuencia Vacación*

Fuente: Elaboración Propia

El método crear evaluación solicita el objeto evaluación, que trae los datos previos del formulario y los ingresa a la base de datos; una vez ingresado a la base de datos, el sistema envía la nota del colaborador mediante un correo.

Ilustración 37.

Método entrada crear evaluación

```
public async Task<IActionResult> Create(Evaluacion evaluacion)
{
    if (ModelState.IsValid)
    {
        int ultimoId = 0;

        if (await _context.Catalogo_Telefono.AnyAsync())
        {
            ultimoId = (int)await _context.Evaluacion.MaxAsync(u => u.idEvaluacion);
        }

        var fechaActual = DateTime.Today;
        int puntos = evaluacion.Pregunta1 + evaluacion.Pregunta2 + evaluacion.Pregunta3 + evaluacion.Pregunta4 + evaluacion.Pregunta5;
        evaluacion.idEvaluacion = ultimoId+1;
        evaluacion.Nota = puntos * 4;
        evaluacion.Fecha_Mes = fechaActual;

        _context.Add(evaluacion);
        await _context.SaveChangesAsync();
        string notable = Estadosd(evaluacion.Nota);
        EnviarCorreo(evaluacion, notable);
        return RedirectToAction(nameof(Index));
    }
    return View(evaluacion);
}
```

Fuente: Elaboración Propia

El método index de permisos trae los permisos activos y no aceptados o denegados por la jefatura.

Ilustración 38.

Método salida mostrar trámites

```
public async Task<IActionResult> Index()
{
    var permisosActivos = await (from permiso in _context.Tramite_Permission
                                join usuario in _context.Usuario on permiso.Usuario_idUsuario equals usuario.idUsuario
                                join persona in _context.Persona on usuario.Persona_idPersona equals persona.idPersona
                                join tipo_permiso in _context.Tipo_Permission on permiso.Tipo_Permission_idTipo_Permission equals tipo_permiso.idTipo_Permission
                                where permiso.Activo == true
                                select new Tramite_Permission
                                {
                                    idTramite_Permission = permiso.idTramite_Permission,
                                    Fecha = permiso.Fecha,
                                    NombrePersona = persona.Nombre,
                                    DescripcionTipoPermiso = tipo_permiso.Descripcion
                                }
                                ).ToListAsync();

    var permisosActivos1 = _context.Tramite_Permission.Where(e => e.Activo == true).ToListAsync();

    return View(permisosActivos);
}
```

Fuente: Elaboración Propia

El método trámite permiso trae el objeto que se llenó en el formulario y lo almacena en la base de datos.

Ilustración 39.

Método entrada crear trámites

```
public async Task<IActionResult> Create(Tramite_Permission tramite_Permission)
{
    if (ModelState.IsValid)
    {
        int ultimoId = 0;

        if (await _context.Tramite_Permission.AnyAsync())
        {
            ultimoId = (int)await _context.Tramite_Permission.MaxAsync(u => u.idTramite_Permission);
        }

        tramite_Permission.idTramite_Permission = ultimoId + 1;
        tramite_Permission.Aceptado = false;
        tramite_Permission.Activo = true;
        _context.Add(tramite_Permission);
        await _context.SaveChangesAsync();
    }

    return RedirectToAction(nameof(Index));
}
```

Fuente: Elaboración Propia

El método control de asistencia inicia la variable del día actual, se verifica si el usuario previamente había realizado la asistencia; de no ser así se guardan los datos del usuario con la fecha actual para la creación de asistencia. Si ya existía un registro previo, se alerta al usuario que ya había ingresado la asistencia.

Ilustración 40.

Método entrada crear asistencia

```

public async Task<IActionResult> Ingresar_Entrada(int idUsuario)
{
    var fecha_actual = DateTime.Now;
    var estado = 0;
    var msg = "";
    var asistencia = await _context.Control_Asistencia.Where(a => a.Usuario_idUsuario == idUsuario && a.Entrada.Date == fecha_actual.Date).FirstOrDefaultAsync();
    if (asistencia == null)
    {
        var control_Asistencia = new Control_Asistencia
        {
            Usuario_idUsuario = idUsuario,
            Entrada = fecha_actual,
            Salida = fecha_actual,
            No_Marco_E = false,
            No_Marco_S = true,
            Ausencia_Justificada = false
        };
        _context.Add(control_Asistencia);
        await _context.SaveChangesAsync();
        estado = 1;
        msg = "Se insertó correctamente la entrada";
    }
    else
    {
        estado = 2;
        msg = "Ya se registró la entrada para el día de hoy";
    }
    return Json(new { estado = estado, msg = msg });
}

```

Fuente: Elaboración Propia

El método crear aguinaldo toma el año actual y elige la fecha de diciembre del mes anterior para ejecutar la búsqueda de los salarios de los empleados en el sistema y realizar el cálculo, para guardar en la base de datos.

Ilustración 41.*Método entrada crear aguinaldo*

```

public async Task<IActionResult> Crear()
{
    try
    {
        int año_Anterior = DateTime.Now.Year - 1;
        DateTime actual = new DateTime(DateTime.Now.Year, DateTime.Now.Month, 30);
        DateTime fecha_Inicial = new DateTime(año_Anterior, 12, 1);
        DateTime fecha_Final = DateTime.Now;

        var usuario_Planilla = await (_context.Planilla
            .Where(p => p.Fecha_Pago >= fecha_Inicial && p.Fecha_Pago <= actual)
            .GroupBy(p => p.Usuario_idUsuario)
            .Select(g => new
                {
                    Usuario_idUsuario = g.Key,
                    Total = g.Sum(p => p.Salario_Bruto)
                }).ToListAsync());

        var temp = "";
        int ultimoId = 0;

        if (await _context.Aguinaldo.AnyAsync())
        {
            ultimoId = (int)await _context.Aguinaldo.MaxAsync(u => u.idAguinaldo);
        }

        foreach (var item in usuario_Planilla)
        {
            ultimoId = ultimoId + 1;
            var Aguinaldo = new Aguinaldo();
            Aguinaldo.idAguinaldo = ultimoId;
            Aguinaldo.Usuario_idUsuario = Convert.ToInt32(item.Usuario_idUsuario);
            Aguinaldo.FechaInicio = fecha_Inicial;
            Aguinaldo.FechaFinal = fecha_Final;
            Aguinaldo.Total = Convert.ToDecimal(item.Total / 12);

            temp = Aguinaldo.Total.ToString();
            await _context.AddAsync(Aguinaldo);
            await _context.SaveChangesAsync();
        }

        return Content(temp.ToString());
    } catch (Exception ex)
    {
        return StatusCode(500, ex.Message);
    }
}

```

Fuente: Elaboración Propia

Procesos y Validaciones

Ilustración 42.

Proceso de registro de asistencia

```

public async Task<ActionResult> Ingresar_Entrada(int idUsuario)
{
    var fecha_actual = DateTime.Now;
    var estado = 0;
    var msg = "";
    var asistencia = await _context.Control_Asistencia.Where(a => a.Usuario_idUsuario == idUsuario && a.Entrada.Date == fecha_actual.Date).FirstOrDefaultAsync();
    if (asistencia == null)
    {
        var control_Asistencia = new Control_Asistencia
        {
            Usuario_idUsuario = idUsuario,
            Entrada = fecha_actual,
            Salida = fecha_actual,
            No_Marco_E = false,
            No_Marco_S = true,
            Ausencia_Justificada = false
        };
        _context.Add(control_Asistencia);
        await _context.SaveChangesAsync();
        estado = 1;
        msg = "Se insertó correctamente la entrada";
    }
    else
    {
        estado = 2;
        msg = "Ya se registró la entrada para el día de hoy";
    }
    return Json(new { estado = estado, msg = msg });
}

```

Fuente: Elaboración Propia

Ilustración 43.

Proceso de validar usuario

```

public bool ValidarUsuario(Login model)
{
    try
    {
        Usuario userTemp = UsuarioExiste(model.Usuario, model.contrasenia);

        if (userTemp != null)
        {
            HttpContext.Session.SetInt32("idUsuario", userTemp.idUsuario);
            HttpContext.Session.SetInt32("Tipo_Usuario", userTemp.tipo_Usuario_idtipo_Usuario);

            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception ex)
    {
        return false;
    }
}

```

Fuente: Elaboración Propia

Ilustración 44.*Proceso de cálculo de CCSS*

```
public decimal CalcularCCSS(decimal salario_Neto)
{
    decimal renta = 0;
    var catalogo_renta = (from c_renta in _context.Catalogo_Renta
                          select new Catalogo_Renta
                          {
                              Porcentaje = c_renta.Porcentaje,
                              Rango_Inferior = c_renta.Rango_Inferior,
                              Rango_Superior = c_renta.Rango_Superior,
                              Monto_Maximo = c_renta.Monto_Maximo,
                          }
                          ).ToList();
    foreach (var c in catalogo_renta)
    {
        if (salario_Neto > c.Rango_Superior)
        {
            renta = c.Monto_Maximo;
        }
        else
        {
            renta = renta + (salario_Neto - c.Rango_Inferior) * (c.Porcentaje / 100);
        }
    }

    return renta;
}
```

Fuente: Elaboración Propia

Ilustración 45.*Validar usuario*

```

public Respuesta ValidarUsuario(Login model)
{
    try
    {
        bool centinela = usuarioExiste(model.Usuario);
        if (centinela)
        {
            bool bloqueado = usuarioBloqueado(model.Usuario);
            if (bloqueado) {
                Usuario userTemp = validarUsuario(model.Usuario, model.contrasenia);
                if (userTemp != null)
                {
                    Bloquear(model.Usuario, 0);
                    HttpContext.Session.SetInt32("idUsuario", userTemp.idUsuario);
                    HttpContext.Session.SetInt32("Tipo_Usuario", userTemp.tipo_Usuario_idtipo_Usuario);
                    return respuesta;
                }
            }
            else
            {
                Bloquear(model.Usuario, 1);
                respuesta.codigo = -1;
                respuesta.mensaje = "La contraseña es incorrecta";
                return respuesta;
            }
        }
    } else
    {

```

Fuente: Elaboración Propia

Ilustración 46.*Validar bloque de usuario*

```

// Referencia
private bool usuarioBloqueado(int id)
{
    return _context.Usuario.Any(u => u.Persona_idPersona == id && u.Bloqueo != 0);
}

```

Fuente: Elaboración Propia

Ilustración 47.*Validar existencia de usuario*

```

private bool usuarioExiste(int id)
{
    return _context.Usuario.Any(e => e.Persona_idPersona == id && e.Activo == true);
}

```

Fuente: Elaboración Propia

Ilustración 48.*Validar usuario con contraseña*

```
private Usuario validarUsuario(int inputId, string inputPassword)
{
    try
    {
        var user = _context.Usuario.SingleOrDefault(u => u.Persona_idPersona == inputId &&
            u.Contrasenia == inputPassword);
        return user;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error al buscar usuario: {ex.Message}");
        return null;
    }
}
```

*Fuente: Elaboración Propia***Ilustración 49.***Validar tipo de usuario en denegar vacaciones*

```
public async Task<IActionResult> NegarVacacion(int? id)
{
    if (HttpContext.Session.GetInt32("Tipo_Usuario") == 1)
    {
        var vacacion = await _context.Vacacion.FindAsync(id);
        vacacion.Activo = false;
        vacacion.Aceptado_Jefatura = false;

        _context.Update(vacacion);
        await _context.SaveChangesAsync();
        respuesta.codigo = 1;
        respuesta.mensaje = "Las vacaciones se han denegado por Jefatura";
    }
    else if (HttpContext.Session.GetInt32("Tipo_Usuario") == 2)
    {
        var vacacion = await _context.Vacacion.FindAsync(id);
        vacacion.Activo = false;
        vacacion.Aceptado_RRHH = false;

        _context.Update(vacacion);
        await _context.SaveChangesAsync();
        respuesta.codigo = 1;
        respuesta.mensaje = "Las vacaciones se han denegado por RRHH";
    }
}
```

Fuente: Elaboración Propia

Ilustración 50.*Validar formulario de persona*

```
public class Persona
{
    [Key]
    [Required(ErrorMessage = "Cedula Requerido")]
    [RegularExpression("[0-9]*$", ErrorMessage = "El campo solo debe contener números")]
    44 referencias
    public int idPersona { get; set; }
    [Required(ErrorMessage = "Nombre Requerido")]
    34 referencias
    public string Nombre { get; set;}
    [Required(ErrorMessage = "Apellido 1 Requerido")]
    26 referencias
    public string Apellido1 { get; set; }
    [Required(ErrorMessage = "Apellido 2 Requerido")]
    26 referencias
    public string Apellido2 { get; set; }
    15 referencias
    public DateTime Fecha_Entrada { get; set; }
    [Required(ErrorMessage = "Horario Requerido")]
    13 referencias
    public int Tipo_Horario_idTipo_Horario { get; set; }
    [Required(ErrorMessage = "Cargo Requerido")]
    15 referencias
    public int Tipo_Cargo_idTipo_Cargo { get; set; }
    14 referencias
    public Boolean Activo { get; set; }
}
```

Fuente: Elaboración Propia

Módulos alcanzados

Aguinaldo

Ilustración 51.

Módulo de aguinaldo

```
int año_Anterior = DateTime.Now.Year - 1;
DateTime actual = new DateTime(DateTime.Now.Year, DateTime.Now.Month, 30);
DateTime fecha_Inicial = new DateTime(año_Anterior, 12, 1);
DateTime fecha_Final = DateTime.Now;

var usuario_Planilla = await (_context.Planilla
    .Where(p => p.Fecha_Pago >= fecha_Inicial && p.Fecha_Pago <= actual)
    .GroupBy(p => p.Usuario_idUsuario)
    .Select(g => new
    {
        Usuario_idUsuario = g.Key,
        Total = g.Sum(p => p.Salario_Bruto)
    }).ToListAsync());
```

Fuente: Elaboración Propia

Evaluación

Ilustración 52.

Módulo de evaluación

```
var fechaActual = DateTime.Today;
int puntos = evaluacion.Pregunta1 + evaluacion.Pregunta2 + evaluacion.Pregunta3
    + evaluacion.Pregunta4 + evaluacion.Pregunta5;
evaluacion.idEvaluacion = ultimoId+1;
evaluacion.Nota = puntos * 4;
evaluacion.Fecha_Mes = fechaActual;
_context.Add(evaluacion);

await _context.SaveChangesAsync();
string rendimiento = Rendimiento(evaluacion.Nota);

EnviarCorreo(evaluacion, rendimiento);
respuesta.codigo = 1;
respuesta.mensaje = "La evaluación se ha enviado al colaborador";
}
return Json(new { estado = respuesta.codigo, msg = respuesta.mensaje });
```

Fuente: Elaboración Propia

Horas Extras

Ilustración 53.

Módulo de horas extras

```

    if (await _context.Horas_Extras.AnyAsync())
    {
        ultimoId = (int)await _context.Horas_Extras.MaxAsync(u => u.idHoras_Extras);
    }
    horas_Extras.idHoras_Extras = ultimoId+1;

    _context.Add(horas_Extras);
    await _context.SaveChangesAsync();
    respuesta.codigo = 1;
    respuesta.mensaje = "La solicitud de horas extras se ha creado";
}
return Json(new { estado = respuesta.codigo, msg = respuesta.mensaje });

```

Fuente: Elaboración Propia

Incapacidad

Ilustración 54.

Módulo incapacidad

```

    if (incapacidad.ComprobanteFile!=null && incapacidad.ComprobanteFile.Length > 0)
    {
        var carpeta = Path.Combine(_hostingEnvironment.WebRootPath, "assets/pdf");
        var nombre = DateTime.Now.ToString("dd-MM-yyyy") + incapacidad.ComprobanteFile.FileName;
        var path = Path.Combine(carpeta, nombre);
        using (FileStream fs = System.IO.File.Create(path))
        {
            incapacidad.ComprobanteFile.CopyTo(fs);
        }
        incapacidad.Comprobante = nombre;
    }
    int ultimoId = 0;
    if (await _context.Incapacidad.AnyAsync())
    {
        ultimoId = (int)await _context.Incapacidad.MaxAsync(u => u.idIncapacidad);
    }
    incapacidad.idIncapacidad = ultimoId +1;
    _context.Add(incapacidad);
    await _context.SaveChangesAsync();
    respuesta.codigo = 1;
    respuesta.mensaje = "La incapacidad se creo correctamente";
}
return Json(new { estado = respuesta.codigo, msg = respuesta.mensaje });

```

Fuente: Elaboración Propia

Liquidación

Ilustración 55.

Módulo liquidación

```

var usuario_Planilla = await (_context.Planilla
    .Where(p => p.Fecha_Pago >= fecha_Inicial && p.Fecha_Pago <= actual)
    .Where(p => p.Usuario_idUsuario == liquidacion.Usuario_idUsuario)
    .Select(g => new
    {
        Total = g.Salario_Bruto
    }).ToListAsync());

decimal aguinaldo = 0;
decimal total = 0;
foreach (var i in usuario_Planilla)
{
    total = i.Total;
    aguinaldo = aguinaldo + i.Total;
}

decimal preaviso = 0;

```

Fuente: Elaboración Propia

Planilla

Ilustración 56.

Módulo planilla

```

// buscar salario
var colaborador = await (from p in _context.Persona
    join c in _context.Tipo_Cargo on p.Tipo_Cargo_idTipo_Cargo equals c.idTipo_Cargo
    join s in _context.Tipo_Salario on c.Tipo_Salario_idTipoSalario equals s.idTipo_Salario
    where p.idPersona == idPersona
    select s.Total).FirstOrDefaultAsync();

double salario_hora = (colaborador / 30) / 8;
double salario_Bruto = (double)(item.TotalHoras * salario_hora);
decimal renta = CalcularCCSS(Convert.ToDecimal(salario_Bruto));
double renta_c = Convert.ToDouble(renta);

double porcentaje = 0.1067;
double ccss = salario_Bruto * porcentaje;
double salario_Neto = salario_Bruto - ccss - renta_c;

```

Fuente: Elaboración Propia

Control Asistencia

Ilustración 57.

Módulo control asistencia

```
var fecha_actual = DateTime.Now;
var asistencia = await _context.Control_Asistencia.Where
    (a => a.Usuario_idUsuario == idUsuario && a.Entrada.Date == fecha_actual.Date)
    .FirstOrDefaultAsync();

if (asistencia == null)
{
    var control_Asistencia = new Control_Asistencia
    {
        Usuario_idUsuario = idUsuario,
        Entrada = fecha_actual,
        Salida = fecha_actual,
        No_Marco_E = false,
        No_Marco_S = true,
        Ausencia_Justificada = false
    }
}
```

Fuente: Elaboración Propia

Vacaciones

Ilustración 58.

Módulo vacaciones

```
if (HttpContext.Session.GetInt32("Tipo_Usuario") == 1)
{
    var vacacion = await _context.Vacacion.FindAsync(id);
    vacacion.Aceptado_Jefatura = true;

    _context.Update(vacacion);
    await _context.SaveChangesAsync();
    respuesta.codigo = 1;
    respuesta.mensaje = "Las vacaciones se han aceptado por Jefatura";
}
else if (HttpContext.Session.GetInt32("Tipo_Usuario") == 2)
{
    var vacacion = await _context.Vacacion.FindAsync(id);
    vacacion.Activo = false;
    vacacion.Aceptado_RRHH = true;

    _context.Update(vacacion);
    await _context.SaveChangesAsync();
}
```

Fuente: Elaboración Propia

Trámite de permisos

Ilustración 59.

Módulo trámite de permisos

```

if (ModelState.IsValid)
{
    int ultimoId = 0;

    if (await _context.Tramite_Permission.AnyAsync())
    {
        ultimoId = (int)await _context.Tramite_Permission.MaxAsync(u => u.idTramite_Permission);
    }

    tramite_Permission.idTramite_Permission = ultimoId + 1;
    _context.Add(tramite_Permission);
    await _context.SaveChangesAsync();
    respuesta.codigo = 1;
    respuesta.mensaje = "El permiso se ha creado correctamente";
}

return Json(new { estado = respuesta.codigo, msg = respuesta.mensaje });

```

Fuente: Elaboración Propia

Pruebas

Tabla 45.

Prueba del módulo de seguridad

Prueba 01			
Nombre:	Prueba Login		
Módulo de la prueba:	Módulo de seguridad		
Caso para probar:	Resultado deseado:	Resultado obtenido:	Estado de la prueba:

Prueba 01			
Verifica si el usuario ingresado existe, no está bloqueado y la contraseña es la correcta.	Ingresar al sistema con las credenciales correctas y que localice en la página de inicio del <i>software</i> .	Se ingresaron las credenciales correctas y se accedió a la página principal del sistema.	Satisfactoria.
Realizado por: Liber Gabriel Salas Barboza			
Fecha de la prueba: 05/07/2024			

Fuente: Elaboración Propia

Tabla 46.

Prueba del módulo de planilla

Prueba 02			
Nombre:	Prueba de planilla		
Módulo de la prueba:	Módulo de planilla		
Caso para probar:	Resultado deseado:	Resultado obtenido:	Estado de la prueba:
El usuario presiona el botón de “Generar planilla” en el mes actual.	El usuario presionó el botón y creó los registro de planilla en el mes actual con los cálculos exactos.	Se crearon correctamente los registros de los colaboradores correctamente.	Satisfactoria.
Realizado por: Liber Gabriel Salas Barboza			
Fecha de la prueba: 05/07/2024			

Fuente: Elaboración Propia

Tabla 47.*Prueba del módulo de vacaciones*

Prueba 03			
Nombre:	Prueba de vacaciones		
Módulo de la prueba:	Módulo de vacaciones		
Caso para probar:	Resultado deseado:	Resultado obtenido:	Estado de la prueba:
El usuario ingresa a crear una vacación y realiza la solicitud.	Poder crear una solicitud de vacaciones, si el usuario posee días disponibles.	Se creó correctamente la solicitud, debido a que contaba con días disponibles.	Satisfactoria.
Realizado por: Liber Gabriel Salas Barboza			
Fecha de la prueba: 05/07/2024			

Fuente: Elaboración Propia

Tabla 48.*Prueba módulo de liquidación*

Prueba 04			
Nombre:	Prueba de liquidación		
Módulo de la prueba:	Módulo de liquidación		
Caso para probar:	Resultado deseado:	Resultado obtenido:	Estado de la prueba:
El usuario ingresa los datos para la liquidación y realiza el cálculo.	El cálculo de la liquidación se realiza de forma correcta y muestra los datos de la liquidación.	Se aplicó el cálculo de la liquidación mostrando la información correcta de la liquidación.	Satisfactoria.
Realizado por: Liber Gabriel Salas Barboza			
Fecha de la prueba: 05/07/2024			

Fuente: Elaboración Propia

Tabla 49.*Prueba módulo de aguinaldo*

Prueba 05			
Nombre:	Prueba de aguinaldo		
Módulo de la prueba:	Módulo de aguinaldo		
Caso para probar:	Resultado deseado:	Resultado obtenido:	Estado de la prueba:

Prueba 05			
El usuario presiona el botón de “Generar el aguinaldo” en el año actual.	El usuario presionó el botón y creó los registros de aguinaldo en el año actual con los cálculos exactos.	Se crearon correctamente los registros de los aguinaldos de los colaboradores correctamente.	Satisfactoria.
Realizado por: Liber Gabriel Salas Barboza			
Fecha de la prueba: 05/07/2024			

Fuente Elaboración Propia

REFERENCIAS

- Alegsa, L. (06 de diciembre de 2023). *Definición de Módulo* (programación). Alegsa. <https://www.alegsa.com.ar/Dic/modulo.php#gsc.tab=0>
- Arias Chaves, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, VI (10), 1-13.
- Brocca J, & Casamiquela R. (2005). Las licencias de software desde la perspectiva del usuario final. *Revista Pilquen*, (7) Recuperado en 18 de julio de 2024, de http://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S1851-31232005000100012&lng=es&tlng=es
- Cabrera, M. (07 de marzo de 2023). *Encriptación de datos*. LinkedIn. <https://www.linkedin.com/pulse/encriptación-de-datos-martin-cabrera/>
- Coppola, M. (02 de mayo de 2023). *Desarrollo web: qué es, etapas y principales lenguajes*. HubSpot. Recuperado el 14 de febrero del 2024 de <https://blog.hubspot.es/website/que-es-desarrollo-web>
- Deusto, H. (octubre, 2015). *Análisis de puntos débiles*. Harvard Deusto. <https://www.harvard-deusto.com/analisis-de-puntos-debiles>
- Dolbuck Ciberseguridad (14 de julio de 2023). *La importancia de las copias de seguridad para tu empresa*. LinkedIn. <https://www.linkedin.com/pulse/la-importancia-de-las-copias-seguridad-para-tu-empresa-dolbuck/?originalSubdomain=es>
- Equipo Orca (18 de noviembre de 2021). Cumplimiento de normas ISO, evita multas y sanciones. <https://blog.orcagrc.com/cumplimiento-normas-iso>
- Ersep (8 de diciembre de 2020). SISTEMA DE REMUNERACIÓN Y COMPESACIÓN. <https://es.eserp.com/articulos/sistemas-de-remuneracion/>
- Farias, A (18 de septiembre). Evaluación de desempeño: Cómo dar retroalimentación efectiva a tu equipo de trabajo. <https://es.linkedin.com/pulse/evaluación-de-desempeño-cómo-dar-retroalimentación-efectiva-farias>
- Francia, J. (25 de septiembre de 2017). *¿Qué es Scrum?* Scrum. <https://www.scrum.org/resources/blog/que-es-scrum>
- GoDaddy. (31 de julio de 2023). *Qué es un servidor y para qué sirve*. Godaddy. <https://www.godaddy.com/resources/latam/stories/que-es-un-servidor-para-que-sirve>

- Henry. (3 de febrero de 2023). *¿Por qué JavaScript es tan importante?* SoyHenry. <https://blog.soyhenry.com/por-que-javascript-es-tan-importante/>
- Hernández, D. (01 de julio de 2022). *Modelo del ciclo de vida del software*. LinkedIn. <https://www.linkedin.com/pulse/modelos-del-ciclo-de-vida-software-daniel-hernandez/?originalSubdomain=es>
- IBM (s.f.) *¿Por qué es importante la seguridad de datos?* IBM. <https://www.ibm.com/es-es/topics/data-security>
- Insight Consultants (s.f.) *The Software Development Life Cycle*. Recuperado el 28 de enero del 2024 de <https://insightconsultants.co/the-software-development-life-cycle-sdlc/>
- Jedrusik, A (febrero del 2017). La protección por patentes de las invenciones implementadas por ordenador. https://www.wipo.int/wipo_magazine/es/2017/01/article_0002.html
- Lucena, P. (18 de mayo de 2023). *¿Qué es la gestión de recursos humanos?* Universidad Cesuma. <https://www.cesuma.mx/blog/que-es-la-gestion-de-recursos-humanos.html>
- Márquez, L (5 de enero de 2023). Control de asistencia: ¿Qué es y para qué sirve? <https://www.geovictoria.com/es-mx/blog/recursos-humanos/control-de-asistencia-que-es/>
- Mata, L. (7 de mayo de 2019). *El enfoque de investigación; la naturaleza del estudio*. InvestigaliaCr. <https://investigaliacr.com/investigacion/el-enfoque-de-investigacion-la-naturaleza-del-estudio/>
- McAfee (01 de abril de 2020). *¿Qué es la privacidad de datos y cómo se puede proteger?* McAfee. <https://www.mcafee.com/blogs/es-mx/privacy-identity-protection/que-es-la-privacidad-de-datos-y-como-se-puede-proteger/>
- Meneses, N. (16 de junio de 2023). *Guía de Ciclo de Vida del Desarrollo de Software*. Coding Dojo. <https://www.codingdojo.la/2023/06/16/guia-del-ciclo-de-vida-del-desarrollo-de-software/>
- Microsoft (14 de mayo de 2024). Paseo por el lenguaje C#. Microsoft. <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>
- Ministerio de Trabajo y Seguridad Social. (enero, 2024). Lista de salarios 2024. Recuperado el 25 de enero del 2024 de https://www.mtss.go.cr/temas-laborales/salarios/Documentos-Salarios/lista_salarios_2024.pdf
- Motessi, C (7 de diciembre de 2022). Automatización de Recursos Humanos: beneficios, desafíos y ejemplos. <https://blog.invgate.com/es/automatizacion-de-recursos-humanos>

- Ojeda, J. (13 de septiembre de 2023). *Arquitectura de Software: Fundamentos, Características, Usos y Ejemplos*. LinkedIn. <https://www.linkedin.com/pulse/arquitectura-de-software-fundamentos-características-y-ojeda-montoya/>
- Quintana, R. (3 de marzo de 2017). *Bases de datos y su importancia dentro de una organización*. Gestipolis. <https://www.gestipolis.com/bases-datos-importancia-dentro-una-organizacion/>
- Santander Universidades (10 de diciembre de 2021). *Investigación cualitativa y cuantitativa: características, ventajas y limitaciones*. <https://www.santanderopenacademy.com/es/blog/cualitativa-y-cuantitativa.html>
- Santiago, A. (30 septiembre de 2023). *Copias de seguridad y su importancia para tus apps*. Bambu Mobile. <https://bambu-mobile.com/importancia-de-las-copias-de-seguridad-de-tus-apps/>
- Sommerville, I. (2011). *Ingeniería de Software*. Editorial Pearson
- Suárez, E. (10 de abril de 2023). *Tipos de investigación y su clasificación*. Experto Universitario. Recuperado el 14 de febrero de <https://expertouniversitario.es/blog/tipos-de-investigacion/>
- Suárez, E. (2 de febrero de 2024). *Guía práctica sobre los distintos tipos de fuentes de información*. Experto Universitario. Recuperado el 20 de febrero de <https://expertouniversitario.es/blog/fuentes-de-informacion/#:~:text=Las%20fuentes%20de%20información%20son,datos%2C%20entrevistas%2C%20entre%20otros.>
- Swanros, O. (enero, 2024). *La diferencia entre programar y ser desarrollador de software*. Recuperado el 29 de febrero del 2024 de <https://www.linkedin.com/pulse/la-diferencia-entre-programar-y-ser-desarrollador-de-software-oscar-zob4c/>
- TalentCross. (20 de abril de 2023). *12 lenguajes de programación más usados*. LinkedIn. <https://www.linkedin.com/pulse/12-lenguajes-de-programación-más-usados-talentcross/?originalSubdomain=es>
- Velázquez, E. (12 de julio de 2022). *Ingeniería de Requerimientos de Desarrollo de Software*. LinkedIn. <https://www.linkedin.com/pulse/ingeniería-de-requerimientos-desarrollo-software-emma-velázquez/>
- Wong, S. (2017). *Análisis y requerimientos de software*. Editorial Universidad Continental

APÉNDICES

Apéndice A: Guía de observación

Entidad:	Restaurante La Victoria
Dirección física de la entidad:	San José, barrio chino
Fecha de la actividad de observación:	
Nombre del estudiante:	Liber Gabriel Salas Barboza

Tabla de control de aspectos observados:

No	Aspectos por observar	Cumple	No Cumple	Oportunidad de mejora	Detalle de Observación
1	Registro de asistencia				
2	Pago de salarios				

No	Aspectos por observar	Cumple	No Cumple	Oportunidad de mejora	Detalle de Observación
3	Cálculo de horas extras.				
4	Trámites de permisos.				
5	Trámites de incapacidad.				

Apéndice B: GUÍA DE ENTREVISTA

Entidad:	Restaurante La Victoria
Nombre del entrevistado:	
Puesto del entrevistado:	
Nombre del estudiante:	Liber Gabriel Salas Barboza
Fecha de la entrevista:	
Lugar o medio de la entrevista:	Zona de descanso

Preguntas:

- Pregunta 1- ¿Dónde almacenan los registros de asistencia?
- Pregunta 2- ¿Los registros de pagos pueden consultarse en cualquier momento?
- Pregunta 3- ¿Cómo es el proceso de cálculo de salarios?
- Pregunta 4- ¿Cómo registran la asistencia?
- Pregunta 5- ¿Creé que puede mejorar la forma de registrar las entradas y salidas de los empleados?
- Pregunta 6- ¿Cómo es el proceso de solicitud de permisos de trabajo?
- Pregunta 7- ¿Cómo calculan los días de vacaciones que tiene cada empleado?
- Pregunta 8- ¿Alguna vez le han pagado de más por las mismas horas trabajadas o menos por las horas extras?
- Pregunta 9- ¿Qué piensa de un sistema que le permita gestionar su perfil de empleado y revisar el historial de la empresa?
- Pregunta 10- ¿Cree usted que automatizar los procesos de la gestión de personal va a tener algún beneficio tanto para la empresa como para los empleados?