

**UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS**

**ESCUELA DE INGENIERÍA INFORMÁTICA**

**PROTOTIPO PARA APLICACIÓN DE GESTIÓN DE RECURSOS  
HUMANOS PARA RASTREADORES MUNDIALES**

**RANDALL FERNÁNDEZ MEZA**

**SAN JOSÉ, NOVIEMBRE 2024**

## **Agradecimiento**

Quiero expresar mi profundo agradecimiento a mis padres, quienes han sido mi mayor apoyo a lo largo de mi carrera universitaria y de todo mi proceso de educación. Sus palabras de aliento han sido mi mayor motivación. Gracias por creer en mí y por enseñarme a siempre seguir adelante. También quiero agradecer a mi hermana Glenda por comprarme mi primera laptop, con esa computadora empecé en el mundo de la ingeniería de software, fue un regalo que ha sido fundamental para mi educación. Tanto mi hermana Glenda, como mi hermano Jose y mi otra hermana Miriam son personas que admiro mucho por su disciplina, compromiso y profesionalismo, han sido un ejemplo para mí. A Jenni, mi novia, le agradezco por impulsarme a retomar mis estudios. Su compromiso con el estudio, su disciplina para seguir aprendiendo han sido una fuente de inspiración. Y finalmente, agradezco a la directora de carrera, doña Olda, por su guía en este camino. Ha sido una líder ejemplar en mi educación.

## **Dedicatoria**

A mis padres, hermanos y sobrina, quienes agradezco tanto su amor incondicional, han estado a mi lado en cada paso de este viaje. Su amor, apoyo y su ejemplo han sido mi mayor fortaleza. Este logro es también suyo. A mi novia, Jenni, quien ha sido mi compañera de sueños y desafíos. Gracias por creer en mí y motivarme en retomar este camino.

## Tabla de Contenidos

Agradecimiento .....	2
Dedicatoria .....	3
APROBACIÓN DEL TRIBUNAL EXAMINADOR.....	6
CÓDIGO DE ÉTICA .....	9
CARTA DE REVISIÓN FILOLÓGICA .....	<b>Error! Bookmark not defined.</b>
<b>DECLARACIÓN JURADA</b> .....	2
CAPÍTULO I: INTRODUCCIÓN .....	7
Descripción de la Empresa .....	7
Problema.....	7
Objetivos .....	8
Objetivo General .....	8
Objetivos Específicos .....	8
Justificación.....	8
Viabilidad .....	9
Proyecciones.....	12
Alcance Funcional .....	12
Alcance Metodológico .....	14
Alcance Tecnológico .....	15
CAPÍTULO II. MARCO REFERENCIAL.....	16
La Gestión de Recursos Humanos.....	16
Salario y Sus Componentes .....	16
Deducciones al Salario .....	17
Horas Extra.....	19
Días Feriados .....	19

	4
Comisiones .....	20
Incapacidades .....	20
Cálculo de Aguinaldo .....	21
Gestión de Vacaciones .....	22
Liquidaciones .....	23
Preaviso .....	23
Cesantía .....	24
Aspectos Tecnológicos.....	25
Definición de Software y Tipos de Software .....	25
Base de Datos .....	30
Infraestructura .....	30
Metodología de Desarrollo de Software.....	32
CAPÍTULO III. MARCO METODOLÓGICO .....	34
Enfoques de Investigación.....	34
Enfoque cuantitativo.....	34
Enfoque cualitativo.....	35
Enfoque mixto .....	35
Enfoque de investigación seleccionado.....	36
Tipos de Investigación.....	36
Exploratorio.....	36
Descriptivo .....	36
Correlacional .....	37
Explicativo.....	37
Tipo de investigación seleccionado.....	38
Fuentes de información .....	38

Fuentes de información primarias .....	39
Fuentes de información secundarias.....	39
Fuentes de información terciarias.....	39
Variables.....	40
Variable conceptual.....	40
Variable operacional.....	41
Variable instrumental .....	41
Población.....	43
Muestra.....	44
Instrumentos de Recolección de Datos .....	45
La observación .....	46
Encuesta.....	46
Entrevistas .....	46
Proceso para la Recolección y Análisis de Datos.....	47
<b>CAPÍTULO IV. ANÁLISIS DE RESULTADOS .....</b>	<b>50</b>
<b>CAPÍTULO V. PROPUESTA .....</b>	<b>54</b>
Análisis.....	54
Análisis Detallado del Prototipo Desarrollado.....	54
Análisis Detallado del Hardware Requerido.....	58
Análisis Detallado de los Elementos de Telecomunicaciones .....	59
Descripción Detallada de la Base de Datos.....	60
Descripción de los Conocimientos de Recursos Humanos para el Uso del Prototipo .....	60
Casos de Uso .....	61
Diseño.....	81
Arquitectura del Sistema .....	81

Arquitectura de Software.....	81
Diccionario de Datos .....	83
Diseño de Bases de Datos .....	90
Diseño de Entradas .....	91
Diseño de Procesos.....	99
Diseño de salidas .....	102
Programación.....	113
Entradas .....	115
Salidas .....	120
Procesos.....	121
Validaciones .....	130
Módulos.....	131
Pruebas .....	146
CAPÍTULO VI, CONCLUSIONES Y RECOMENDACIONES .....	161
Conclusiones .....	161
Recomendaciones .....	162
Referencias .....	163
Apéndices .....	167
Apéndice 1.....	167
Apéndice 2.....	169

## **CAPÍTULO I: INTRODUCCIÓN**

### **Descripción de la Empresa**

Rastreadores Mundiales es una empresa dedicada a la venta, instalación, mantenimiento y monitoreo de sistema de localización para vehículos. Sus sistemas de localización se realizan por medio de GPS y/o radiofrecuencia. Estos sistemas pueden ser monitoreados tanto por Rastreadores Mundiales como por el dueño del vehículo o flotilla, que tiene a su disposición una aplicación para celular que muestra la localización actual del vehículo. Rastreadores Mundiales cuenta con 35 colaboradores entre puestos administrativos y operativos. Se ubica en Desamparados Centro, 100 m este del costado noreste de la Iglesia de Desamparados.

### **Problema**

Para este trabajo se debe entender primero cuál es el problema que se aborda. Según Sampieri (2017): “El problema debe estar formulado mediante objetivos y preguntas claras, concretas y sin ambigüedades” (p. 41). Rastreadores Mundiales no cuenta con un sistema informático que le permita realizar las tareas relacionadas a la planilla, de momento usan Excel o inclusive hay cálculos que se realizan manualmente por los encargados de Recursos Humanos. Entre los problemas principales de Rastreadores Mundiales se encuentra que debido a que el cálculo de salario se realiza de manera manual la persona encargada invierte mucho tiempo realizando estos cálculos y luego verificando que se hayan realizado de manera correcta. Debido a que no existe un sistema que lleve el total de horas extras se debe revisar y calcular manualmente el impacto de las horas extras en el salario. Al no haber un sistema que calcule el monto de una liquidación, se debe llevar a cabo de manera manual, lo cual implica posibles errores de cálculo y pérdida del tiempo consumido en estas labores. No existe un control digital del histórico de incapacidades, y el cálculo del impacto de las incapacidades en el salario se hace manualmente.

Las dificultades actuales de la empresa respecto a la información y cálculo de la nómina no solo existen en cuanto al cálculo de salario, sino que además hay otras situaciones que la empresa necesita solventar. Por ejemplo, para el rubro de las vacaciones cada año hay que añadir manualmente los nuevos días adquiridos de vacaciones. Además, la deducción por goce de vacaciones también se realiza de manera manual. El cálculo del aguinaldo se realiza mediante la

aplicación de Microsoft Excel, pero para ello se debe ingresar los datos mensualmente, En términos generales, los problemas con respecto al manejo actual de la nómina se reflejan en el tiempo consumido para realizar las tareas descritas manualmente y también la posibilidad del error humano en los cálculos descritos.

## **Objetivos**

### **Objetivo General**

Desarrollar un prototipo funcional para la gestión de nómina para la compañía de Rastreadores Mundiales.

### **Objetivos Específicos**

Analizar los requerimientos de la empresa para el manejo de nómina mediante la evaluación de la situación actual y la situación esperada.

Diseñar la estructura del prototipo para el sistema de manejo de nómina según el análisis y los requerimientos de la empresa.

Desarrollar la programación del prototipo para el cumplimiento de los requerimientos de la empresa guiándose en el diseño de la estructura del prototipo.

Realizar las pruebas funcionales para que se garantice el cumplimiento de los requerimientos previamente establecidos.

## **Justificación**

La realización de este prototipo se justifica para entregar un prototipo que le ayude a Rastreadores Mundiales a poder ahorrar tiempo en el cálculo de la nómina, así como para disminuir los fallos humanos durante el cálculo de salarios, vacaciones, liquidaciones, entre otros. Para Rastreadores Mundiales un sistema funcional para manejo de nómina les permitiría tener un control más eficiente desde una plataforma centralizado para la gestión de la nómina y las diferentes operaciones que se explicarán más adelante.

## **Viabilidad**

Para que este prototipo sea viable se necesita que mejore el estado actual de la empresa con respecto al manejo de planillas. Sobre la viabilidad, Sampieri (2017) la define como: "La viabilidad de la investigación significa que es posible realizarla en un tiempo determinado y con los recursos disponibles." (p. 45) En este caso la investigación y el producto final deben cumplir con diversos criterios que se detallan adelante.

### **Viabilidad técnica.**

Este proyecto es viable técnicamente debido a que para la empresa la mejor manera de disponer de un sistema de gestión es permitiendo a sus colaboradores usar sus propios dispositivos móviles para que se conecten a la web y desde ahí poder realizar sus gestiones. Los colaboradores del Departamento de Recursos Humanos también podrán usar las computadoras con las que actualmente trabajan.

Rastreadores Mundiales actualmente cuenta con un servidor, sin embargo, la empresa entiende que por temas de seguridad y fiabilidad es mejor adquirir un servidor de base de datos y un servidor de aplicación en Azure. Este servidor se alojará en la nube y debe contar con un núcleo, 1,5GB de memoria RAM, 10GB de espacio en Disco si se instala un sistema operativo Windows Server 2019 o Windows Server 2022.

En cuanto al *software* se desarrollará en Visual Studio Code, un servidor web IIS o Apache, y un explorador web que será Microsoft Edge. La base de datos será SQL Server 2022 Express, en caso de que sea servidor físico, o Azure SQL Server si fuera en la nube, y se contará con Notepad++ en el sistema para ayudar en la lectura de Logs. Los lenguajes de programación serán JavaScript para la parte funcional y CSS con HTML para la interfaz gráfica.

### **Viabilidad operativa.**

En cuanto a la viabilidad operativa se determina que la empresa utilizaría la aplicación para la gestión de la nómina con el fin de migrar sus procesos actuales de documentación de nómina al

prototipo. En cuanto a los usuarios finales, ellos utilizarían sus dispositivos móviles o un computador que ya existe en la compañía para solicitud de vacaciones, de reporte de horas extras y demás tareas relacionadas con gestión de nómina.

Respecto al conocimiento requerido por el personal, no es necesario un conocimiento profundo en sistemas, solo que estén familiarizados con el uso del celular en el caso de los usuarios finales. Para los usuarios administrador o administrador de nómina sí se requiere alguna experiencia en el uso de páginas web. En este sentido, es el diseño del prototipo el que debe ajustarse para que sea una aplicación amigable con el usuario y que sea intuitiva. De igual manera se explicará a los usuarios administrador el uso de la aplicación y serán ellos los que decidan si se debe realizar un entrenamiento para aprender a usar el prototipo.

La utilización de este prototipo no provocará alguna reducción de personal, pues el fin último es que sea de apoyo para el personal existente que gestiona las nóminas. Los mayores cambios que percibirán los usuarios administradores de nómina es que al utilizar el prototipo los cálculos con respecto a salarios, vacaciones, entre otros, serán realizados automáticamente por la aplicación para así evitar errores humanos en los cálculos mencionados.

### **Viabilidad económica.**

En cuanto al costo del prototipo, de momento se plantea el uso de la nube para así no tener que adquirir equipo físico nuevo. Se sugiere utilizar Azure SQL Database, localizada en la región Este de EE. UU., con base de datos única, el modelo de compra de núcleo virtual, con nivel de servicio de uso general, nivel de proceso es aprovisionado, tipo de Hardware estándar Gen 5. Adicionalmente, se necesita un App Service y se recomienda el nivel básico con 1 núcleo, 1.5GB de RAM y 10GB de almacenamiento, con sistema operativo Windows.

**Tabla 1***Costos de operación en Azure*

Azure SQL Database en Serie Estándar Gen 5 con pago por uso	\$222.4 mensual
Licencia de SQL pago por uso	\$145.95 mensual
Almacenamiento de 10GB	\$1.5 mensual
Retención de copia de 1 mes	\$0.25 mensual
Servicio de aplicación en Azure	\$54.75 mensual
Costo total	\$427.97 mensual

Fuente. Elaboración Propia (2024)

En cuanto a la mano de obra no habrá un costo al ser un proyecto universitario, aun así, se calcula el costo de salario tomando el costo por día para programador de computación sin título en ₡15 613.91 según Ministerio de Trabajo de Costa Rica (2024). El cálculo de horas de trabajo para culminar el proyecto aún está pendiente de realizarse. Con respecto al precio se detalla lo siguiente tomando en cuenta la calculadora de precios en línea en el sitio de Microsoft Azure.

**Viabilidad legal.**

Se debe considerar en un proyecto de este tipo que sea viable legalmente, para garantizar que se cumple con las leyes nacionales de informática y así mitigar cualquier posible riesgo de conflicto legal, que puede llevar a demandas o sanciones. Se mencionan algunos de las leyes y partes de su contenido que se deben tomar en cuenta tanto en la creación del prototipo como en la capacitación del personal sobre el correcto uso de la información.

Al ser una aplicación de gestión de planillas, en esta se guarda información que no debe es compartida sin consentimiento, por ejemplo, el salario que devenga un colaborador. Sobre compartir esta información la Ley Número 4573 del Código Penal (1973) detalla en el Artículo 196 bis sobre la Violación de comunicaciones electrónicas:

“Será reprimida con pena de prisión de seis meses a dos años, la persona que, para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento, se apodere, accese[sic], modifique, altere, suprima, intercepte, interfiera, utilice, difunda o desvíe de su destino, mensajes, datos e imágenes contenidas en soportes: electrónicos, informáticos, magnéticos y telemáticos”

Entre las acciones que pueden causar problemas legales se debe considerar también el procesamiento de datos ya que el prototipo calcula los salarios con diferentes rubros como por ejemplo las horas extra, si estas son alteradas entonces eso causaría que los montos por pagar no sean correctos.

Respecto de lo anterior, el Código Penal en su ley 4573 (1973) establece en el artículo 217 bis: "Se impondrá pena de prisión de uno a diez años a la persona que, con la intención de procurar u obtener un beneficio patrimonial para sí o para un tercero, influya en el procesamiento o el resultado de los datos de un sistema de cómputo". Estos componentes como las horas extra se explican más adelante en el marco referencial con mayor detalle.

Los datos no solo pueden ser modificados durante el procesamiento de la información, también se pueden modificar en la base de datos, esto también podrá ser causal de denuncia penal. Sobre esto el Código Penal en la ley 4573 Artículo 229 bis indica que: "Se impondrá pena de prisión de uno a cuatro años a la persona que por cualquier medio accese[sic], borre, suprima, modifique o inutilice sin autorización los datos registrados en una computadora.". La base de datos en este prototipo no podrá ser accedida por ningún colaborador para su modificación.

### **Proyecciones**

Con este prototipo se pretende optimizar el proceso de gestión de nómina en cuanto a que los cálculos mencionados se realizaran de manera automática, con esto se espera ahorrar tiempo en estas tareas que en la actualidad se realizan en forma manual, además de evitar los errores que ocasionalmente ocurren al realizar estas gestiones de dicha manera, tanto los cálculos de montos como en la inconsistencia de donde se guarda la información. Para la empresa el beneficio está en realizar las tareas de nómina de manera más eficiente y disminuyendo el tiempo que toman estas.

### **Alcance Funcional**

Este prototipo es sobre una aplicación para la gestión de nóminas. En la actualidad las operaciones de gestión de nómina como el cálculo de salarios, de vacaciones, de liquidaciones, entre otros que se detallan a continuación, se llevan a cabo utilizando hojas de cálculo de Microsoft Excel, así como por medio de comunicaciones informales, por ejemplo, la petición de vacaciones

y la aprobación de estas se realiza de manera oral y luego se envía un correo de confirmación. En vista del funcionamiento actual se considera que este proyecto es una alternativa que ayuda a realizar las tareas mencionadas de una manera más eficiente. La mayor oportunidad que presenta es la disminución de los errores por cálculo humano, como la posibilidad de perder información que actualmente se maneja de manera oral o por correo al tener en una aplicación toda la información pertinente a la gestión de nómina.

## Tabla 2

### *Descripción de módulos funcionales*

Nombre del Módulo	Descripción del Módulo
Gestionar Nómina	Este módulo realiza el cálculo de los salarios de cada empleado tomando en cuenta los impuestos y deducciones de ley, las ausencias justificadas e injustificadas y el pago de horas extra y comisiones.
Gestionar Vacaciones	Este módulo se encargará de calcular la cantidad de vacaciones que tiene un empleado utilizando la información de su tiempo de pertenecer a la empresa. Para solicitar vacaciones el colaborador enviará la solicitud vía sistema a su jefatura inmediata, si esta aprueba, se deberá notificar vía sistema a RRHH, y este se encargará de notificar vía sistema al colaborador. En caso de rechazar la solicitud, se notifica vía sistema al colaborador y descontará los días de vacaciones del acumulado de cada colaborador.
Gestionar Horas Extra	En este módulo el colaborador enviará la solicitud de pago extra vía sistema a su jefatura inmediata, si esta aprueba, se deberá notificar vía sistema a RRHH, y este se encargará de notificar vía sistema al colaborador. En caso de rechazar la solicitud, se notifica vía sistema al colaborador. Si se aprueban, el sistema ingresará las horas extras de cada colaborador y se utilizará para el cálculo sobre el salario.
Gestionar Liquidaciones	En este módulo se podrán calcular los montos de liquidación con base en el tiempo laborado, los salarios y los días de vacaciones pendientes. Este módulo guardará un histórico de liquidaciones.
Gestionar Incapacidades	En este módulo se procesarán las incapacidades de los colaboradores, se calcularán los días de incapacidad, y con base en el tipo de incapacidad, se calculará el impacto que tendrá en el salario, también de los permisos sin goce salarial. Este módulo guardará un histórico de incapacidades.
Gestionar Aguinaldo	Este módulo calcula el aguinaldo de cada colaborador apoyándose en el histórico del cálculo salarial.
Mantenimientos	Este módulo se encargará de realizar el borrado, inserción, modificación y actualización de datos.
Consultas	Este módulo se encargará de generar información proporcionada de las diferentes tablas.
Reportes	Este módulo se encargará de generar información proporcionada de las diferentes tablas y procesos, pero con un formato específico, según lo solicite el usuario. Podrá ser impreso o por pantalla
Seguridad	Este módulo se encargará de realizar la autenticación de contraseñas y definición de perfiles.

Fuente: Elaboración Propia (2024).

## Alcance Metodológico

La metodología que se utilizará es la metodología en cascada, esta consta de varias etapas, de acuerdo con Córdoba (2023) “Las etapas del modelo de desarrollo cascada suelen ser: Recopilación de requisitos, Análisis de requisitos, Diseño, Implementación, Pruebas y Mantenimiento.”

A continuación, se describen cada una de estas etapas:

Recopilación de requisitos: se toman en cuenta los requisitos de Rastreadores Mundiales, además, se realizarán preguntas para confirmar el entendimiento de los requisitos.

Análisis de requisitos: Se realiza un análisis de los requisitos planteados para determinar su factibilidad; de ser factible, se trata el cómo se va a abordar su cumplimiento. Aquí se plantean las soluciones a los requisitos.

Diseño: en esta etapa se crean los diferentes diagramas como los de casos de uso o modelado de base de datos.

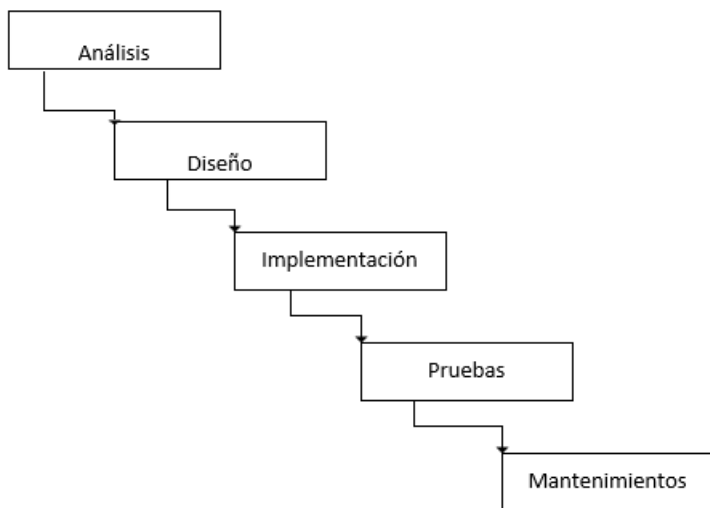
Desarrollo: con el apoyo de un diseño robusto en el paso anterior, ahora se desarrollará el código para cumplir con los requerimientos.

Validación: en esta etapa se probará que el *software* funcione y no haya errores.

Mantenimiento: se realiza el mantenimiento y se corrigen errores detectados.

### Figura 1.

#### *Metodología Cascada*



Fuente: Elaboración Propia (2024)

### **Alcance Tecnológico**

Se empleará una aplicación web que estará en un servidor IIS en Azure, o lo que se llama en Azure como App Service con las siguientes características: 1 núcleo con 1.75GB de RAM y de 10GB de almacenamiento. Los lenguajes por utilizar son Javascript para la parte funcional de los módulos, con HTML y CSS para el diseño gráfico, se desarrollará utilizando Visual Studio Cod como IDE. En cuanto al motor de base de datos se recomienda utilizar Azure SQL Server de la serie Estándar de Gen 5, con 2vCore.

Se utilizarán los dispositivos móviles de los colaboradores para que ellos puedan conectarse a la aplicación mediante su navegador web preferido y también se dispondrá de un computador en la empresa que pueda utilizarse en caso de que se prefiera este medio antes que el teléfono celular.

## **CAPÍTULO II. MARCO REFERENCIAL**

### **La Gestión de Recursos Humanos**

En el contexto empresarial, cuando el concepto de recursos humanos refiere al conjunto de colaboradores que laboran para una empresa. Al iniciar una empresa es común empezar definiendo la visión, la misión y los objetivos. En la inmensa mayoría de los casos y dependiendo de cuán ambicioso sea el proyecto de la empresa, es imposible que solo con el fundador se puedan abarcar todas las tareas, tanto administrativas como operativas. De ahí que plantearse una empresa implica considerar cuántas personas se necesitan para cumplir los objetivos, además de definir el perfil de dichos colaboradores.

La elección del recurso humano es de suma importancia en una empresa, sobre esta importancia Agudelo, *et al.* (2016) concluyen que: “El personal es el factor clave en la capacidad de adaptación de la organización y en la consolidación de una ventaja competitiva, de ahí que el futuro de aquella dependa de lograr un capital humano idóneo y comprometido.” (p. 1). Por ello, tener una buena gestión del talento humano es imprescindible para alcanzar el éxito empresarial.

Varios factores inciden en una gestión del recurso humano adecuada, a saber: la captación de talento, la retención del talento, las posibilidades de desarrollo y crecimiento, la comunicación clara y sincera desde los puestos de liderazgo. Así mismo, se puede hablar de un adecuado manejo de los sistemas informáticos y utilización de la tecnología para la gestión del recurso humano. Respecto de esto último Carvajal (2020) indica:

El acelerado avance de la tecnología y su incorporación en los procesos laborales ha modificado no solo la forma en que se realiza el trabajo y sus resultados, sino que también ha producido un cambio en el acceso a la información (párr. 20).

De lo anterior se concluye que, uso de la tecnología para la gestión del recurso humano es sumamente importante y ahí radica la esencia de este proyecto.

### **Salario y Sus Componentes**

En las relaciones obrero-patronales hay diferentes compromisos que contraen las partes, en cuanto al obrero o colaborador, este se compromete a realizar las labores asignadas por el patrono. A cambio de ello el patrono debe devengar un salario para pagarle por los servicios prestados. Los

salarios pueden representar parte alta de los gastos de una empresa, por lo cual realizar un cálculo adecuado de este rubro sin cometer errores es de suma importancia.

Sobre el cálculo de los salarios, cabe destacar que en las diferentes legislaciones o códigos laborales es común que se contemplen diferentes aspectos para calcular un salario. Con respecto al cálculo de los salarios, Morales (2008) recuerda que:

Los salarios, como elementos primordiales de la prestación de un servicio personal subordinado, son objeto de un régimen contributivo bastante complejo. Entender el tratamiento fiscal de éstos y de las prestaciones derivadas de una relación de trabajo no es una tarea fácil. (p. 26).

Los elementos que componen un salario se explicarán en los siguientes párrafos, e incluyen componentes que aumentan el salario como lo son las horas extras, días de pago doble y comisiones, pero también hay otros rubros que reducen el salario como lo son los permisos de ausencia sin goce salarial y las incapacidades. Todos estos elementos se deben tomar en cuenta para cada cálculo de salario que se realice.

### **Deducciones al Salario**

Anteriormente, se comentó sobre el cálculo de salarios. Dentro de los componentes más importantes del salario se encuentran las deducciones, esto se refiere a los montos que se restan del salario inicial del colaborador. Según Alfaro (2024) “Como mínimo, los trabajadores y los patronos se deben enfrentar a 11 rebajos, en conjunto, cada vez que inician una relación de trabajo debidamente acreditada.” (párr. 1). Al monto del salario inicial previo a cálculos que contemplen deducciones se le llama salario bruto, ese es el monto total que el patrón aporta. Al monto que le llega al obrero luego de las deducciones se le llama salario neto.

En Costa Rica se aplican varias deducciones laborales. Algunas de ellas deben ser pagadas por medio del patrono, las otras son asumidas por el colaborador. En términos generales, estas de deducciones se clasifican en tres, a saber: Caja Costarricense de Seguro Social, otras instituciones y Ley de Protección al Trabajador. En la tabla 3 se detallan los aspectos mencionados con anterioridad.

**Tabla 3***Deducciones salariales en Costa Rica*

Caja Costarricense de Seguro Social		
Concepto	Patrono	Trabajador
Seguro de Enfermedad y Maternidad (SEM)	9.25%	5.50%
Invalidez, Vejez y Muerte (IVM)	5.42%	4.17%
Total CCSS	14.67%	9.67%
Recaudación Otras Instituciones		
Institución	Patrono	Trabajador
Cuota Patronal Banco Popular	0.25%	
Asignaciones Familiares	5.00%	
Instituto Mixto de Ayuda Social	0.5%	
Instituto Nacional de Aprendizaje	1.5%	
Total Otras Instituciones	7.25%	
Ley de Protección al Trabajador (LPT)		
Concepto	Patrono	Trabajador
Aporte Patrono Banco Popular	0.25%	
Fondo de Capitalización Laboral	1.50%	
Fondo de Pensiones Complementarias	2.00%	
Aporte Trabajador Banco Popular		1.00%
Instituto Nacional Seguros	1.00%	
Total Ley Protección al Trabajador	4.75%	1.00%
Total		
	Patrono	Trabajador
Porcentajes Totales-	26.67%	10.67%

*Fuente:* Caja Costarricense de Seguro Social

## **Horas Extra**

Se ha descrito el pago de elementos que disminuyen el monto total del salario, además de estos rubros existen otros relacionados con el aumento del monto del salario. El primero de estos elementos son las horas extra, conocidas formalmente como jornada extraordinaria. En la normativa costarricense y de acuerdo con el Ministerio de Trabajo y Seguridad Social (2024) se establece sobre el pago de jornada extraordinaria lo siguiente:

“Es el tiempo que se labora más allá de la jornada ordinaria correspondiente, o de la jornada inferior que hubieren convenido las partes. Debe pagarse a razón de una hora ordinaria más un cincuenta por ciento (tiempo y medio) por la cantidad de horas extras laboradas. Tratándose de días feriados, cada hora extra deberá pagarse doble.”

En el contexto específico de la empresa Rastreadores Mundiales es frecuente que se deba trabajar jornadas extraordinarias debido a que las instalaciones de los rastreadores en los vehículos pueden tomar más tiempo del esperado; por lo tanto, es importante llevar un registro de cuántas horas de más se trabajan por cada instalador en cualquier otro puesto con estas características de jornada extraordinaria.

## **Días Feriados**

Otro componente relacionado con el aumento del monto de salario es el pago de días feriados. Con días feriados se hace referencia a aquellos días que por ley se deben permitir a los colaboradores el descanso laboral, ya sea por celebraciones religiosas, históricas, entre otros. Sobre los días feriados el Código de Trabajo de Costa Rica (1947) determina lo siguiente:

ARTÍCULO 148.- Se considerarán días feriados y, por lo tanto, de pago obligatorio los siguientes:

el 1 de enero, el 11 de abril, el Jueves y Viernes Santos, el 1° de mayo, el 25 de julio, el 15 de agosto, el 15 de setiembre y el 25 de diciembre. Los días 2 y 31 de agosto y el 1° de diciembre también se considerarán días feriados, pero su pago no será obligatorio

En los días feriados también se podría dar una situación de jornada extraordinaria, para estos casos específicos el Ministerio de Trabajo y Seguridad Social (2024) establece que “Si se

trata de pago mensual, la jornada ordinaria se pagaría doble y cada hora extraordinaria, a 1.5 (tiempo y medio), pero calculada sobre la hora doble, correspondiente al día feriado laborado”. Es decir, que si toma el monto de pago doble y se divide entre 8 horas, sobre este monto se multiplica por 1.5 para obtener el valor total de las horas extra para los días feriados.

## **Comisiones**

Un tercer componente que puede modificar el salario al alza son las comisiones. Estas son diferentes del salario base en cuanto no se calculan de acuerdo con la cantidad de horas laboradas, sino que se refiere a una remuneración que puede ganar un colaborador si cumple con ciertos objetivos, ya sean estos relacionados con el rendimiento en su productividad, a metas específicas, a montos de ventas, entre otros. Estas comisiones no son iguales para todas las empresas y se tomarán en cuenta únicamente las especificadas por Rastreadores Mundiales.

## **Incapacidades**

En una empresa habrá momentos donde un trabajador no pueda presentarse a laborar. Una razón para estas ausencias es una incapacidad laboral, esto se refiere a un periodo en días donde debido a un deterioro de salud el colaborador no puede cumplir con sus funciones. El Reglamento de Seguro de Salud tipifica lo siguiente:

“Período de reposo ordenado por los médicos u odontólogos de la Caja o médicos autorizados por ésta, al asegurado directo activo que no esté en posibilidad de trabajar por pérdida temporal de las facultades o aptitudes para el desempeño de las labores habituales u otras compatibles con ésta.” (p.5).

Las razones para la incapacidad laboral son tres: por enfermedad, por licencia de maternidad y por riesgo de trabajo. En la mayoría de los casos de enfermedad la empresa solicitará al colaborador presentar un comprobante de salud, para esto el colaborador debe presentarse a un hospital de la Caja Costarricense de Seguro Social (CCSS) o a un Equipo Básico de Atención Integral en Salud (EBAIS). También es posible que la incapacidad sea dictaminada por un hospital o clínica privada, pero de igual manera esa incapacidad debe ser aprobada por la CCSS.

Otra posibilidad es que la incapacidad sea dictaminada por el Instituto Nacional de Seguros (INS) en el caso de que se trate de alguna enfermedad o lesión ocurrida durante el tiempo laboral. Sobre el riesgo de trabajo el Código de Trabajo en el artículo 195 indica que:

los accidentes y las enfermedades que ocurran a los trabajadores, con ocasión o por consecuencia del trabajo que desempeñen en forma subordinada y remunerada, así como la agravación o reagravación que resulte como consecuencia directa, inmediata e indudable de esos accidentes y enfermedades. (p. 58).

Para la empresa es importante conocer los estatutos con respecto a las incapacidades, con el fin de apegarse a los pagos correspondientes. En la incapacidad por enfermedad el patrono tiene la obligación de cancelar al menos el 50% del salario del empleado durante los primeros tres días. A partir del cuarto día la CCSS paga el 60% del salario del colaborador y el empleador no tiene obligación de pagar durante este periodo.

### **Cálculo de Aguinaldo**

Las empresas, además de pagar un salario mensual, quincenal o semanal, también se encuentran en la obligación de cubrir el aguinaldo, componente descrito como un salario extra, este pago debe realizarse entre los primeros 20 días del mes de diciembre. Para el cálculo del aguinaldo el MTSS en su *brochure* sobre El Aguinaldo en la Empresa Privada informa:

“El aguinaldo es un doceavo de todos los salarios ordinarios y extraordinarios, devengados por la persona trabajadora durante los doce meses que van del 1° diciembre del año anterior al 30 de noviembre del año que se trate, por lo que se obtiene de la suma de dichos salarios y se divide entre doce. (p. 1).”

Es importante recalcar que se mencionan los salarios extraordinarios, que incluyen jornadas extraordinarias y comisiones. En el cálculo deben tomarse en cuenta las incapacidades que se explicará más de este concepto en los siguientes párrafos, pero de momento es importante recalcar que el MTSS en el mismo *brochure* de El Aguinaldo en la Empresa Privada informa que: “No se toma en cuenta el período que la persona trabajadora estuvo incapacitada por enfermedad o riesgo de trabajo, por cuanto no recibió salario sino un “subsidio. No sucede así cuando se trata de una trabajadora que disfrutó licencia por maternidad” (p. 1).

## Gestión de Vacaciones

Otro motivo de ausencia de un colaborador es por el goce de vacaciones, para esta investigación es relevante entender cómo se calculan los días de vacaciones. Sobre esto el Código de Trabajo en su artículo 153 estipula que: “Todo trabajador tiene derecho a vacaciones anuales remuneradas, cuyo mínimo se fija en dos semanas por cada cincuenta semanas de labores continuas, al servicio de un mismo patrono.” (p. 51). Para las empresas es importante tener documentado en qué momento pasan esas cincuenta semanas, para así poder organizar con el colaborador el periodo de vacaciones, tanto para otorgar el descanso al empleado como para planificar su ausencia sin que esto cause problemas a la operativa de la empresa.

Con respecto a las vacaciones también es necesario comprender cómo se remuneran, para ello se toma en cuenta la normativa del Ministerio de Trabajo y Seguridad Social:

**Pago semanal:** se le deben pagar doce (12) días, porque en esta modalidad no se paga el día de descanso. El cálculo se realiza tomando el salario semanal dividiéndolo entre 6 y multiplicando el resultado por 12.

**Pago mensual o quincenal:** le corresponden catorce (14) días, porque incluye en el pago los dos días de descanso semanal. El cálculo se realiza tomando el salario mensual dividiéndolo entre 30 y multiplicando el resultado por 14.

En el caso de actividades comerciales, aunque la forma de pago sea semanal, para efectos de vacaciones se aplica la siguiente fórmula: se toma el salario semanal dividiéndolo entre 7 y multiplicando el resultado por 14.

Las vacaciones son un derecho del trabajador, en caso de que se termine la relación obrero-patronal el Código de Trabajo establece lo siguiente: “Cuando el trabajador cese en su trabajo por cualquier causa, tendrá derecho a recibir en dinero el importe correspondiente por las vacaciones no disfrutadas.” (p. 51). Por lo tanto, se debe tener en cuenta el cálculo de salario mencionado en el párrafo anterior, si así lo ameritara para el pago de vacaciones.

## Liquidaciones

La liquidación se refiere a los montos que deben pagarse al colaborador cuando se termine su relación laboral con la empresa. Anteriormente se mencionó que cuando se finaliza una relación laboral se debe pagar al colaborador los días de vacaciones pendientes, en caso de que los hubiere; esto es parte fundamental de las liquidaciones. El aguinaldo es, también, un componente del monto de las liquidaciones que se debe tomar en cuenta para el pago total al colaborador. Además de las vacaciones y el aguinaldo existen otros dos componentes por tomar en cuenta: el preaviso y cesantía

### Preaviso

Cuando una relación laboral finaliza, ya sea por renuncia del colaborador o porque el colaborador es despedido con responsabilidad patronal por el empleador se debe dar un preaviso, básicamente es un aviso de que la relación se está terminando. Si es por renuncia, entonces el colaborador debe de cumplir con un periodo de días de trabajo para no afectar la operativa de la compañía; en caso de ser por despido entonces el preaviso es un periodo en que el colaborador seguirá trabajando y tendrá derecho de obtener el salario. El cálculo del preaviso se realiza tal y como se muestra en la tabla 4:

**Tabla 4**

*Pago de preaviso*

Periodo laborado	Periodo de preaviso
Antes de los tres meses	No aplica
Más de tres meses – menos de seis meses	1 semana de preaviso
Más de seis meses – menos de un año	15 días de preaviso
Más de un año	1 mes de preaviso

*Fuente: MTSS brochure Preaviso y Cesantía*

## Cesantía

La cesantía se refiere al pago que debe de hacer el empleador al colaborador en caso de despido con responsabilidad laboral, el objetivo es el realizar un pago al colaborador para ayudarlo a mantenerse durante el periodo que se encuentre sin trabajo. Sobre la cesantía el MTSS en su *brochure* Preaviso y Cesantía indica que: “Es un derecho que tienen las personas trabajadoras a ser indemnizados en caso de terminación de la relación laboral con responsabilidad patronal.” (p. 1). El cálculo del auxilio de cesantía se realiza como se detalla en la tabla 5.

**Tabla 5**

*Pago de auxilio de cesantía.*

Periodo de tiempo laborado.	Pago de cesantía
Más de tres meses – menos de seis meses	Siete días de salario
Más de seis meses – menos de un año	Catorce días de salario
Un año	19.5 días por año o fracción mayor a seis meses
Dos años	20 días por año o fracción mayor a seis meses
Tres años	20.5 días por año o fracción mayor a seis meses
Cuatro años	21 días por año o fracción mayor a seis meses
Cinco años	21.24 días por año o fracción mayor a seis meses
Seis años	21,5 días por año o fracción mayor a seis meses
Siete años	22 días por año o fracción mayor a seis meses
Ocho años	22 días por año o fracción mayor a seis meses
Nueve años	22 días por año o fracción mayor a seis meses
Diez años	21.5 días por año o fracción mayor a seis meses
Once años	21 días por año o fracción mayor a seis meses
Doce años	20.5 días por año o fracción mayor a seis meses
Trece años o más	20 días por año o fracción mayor a seis meses

*Fuente: MTSS brochure Preaviso y Cesantía*

Hasta el momento se han explicado los rubros referentes a la nómina de una empresa, a continuación, se detallarán los conceptos relacionados con la parte tecnológica, ya que el propósito del proyecto es lograr una mejor gestión de la planilla por medio de la tecnología.

## Aspectos Tecnológicos

### Definición de Software y Tipos de Software

Para empezar, por *software* se entiende, de acuerdo con McIver (2011, citado por Rivas *et al* 2015): “El conjunto de programas, procedimientos y documentación relacionada que asocia un sistema computacional, específicamente a parte lógica de la computadora.”. Hay diversos tipos de *software*, como el *software* de sistema que se refiere a un sistema operativo y software de aplicación, como lo es una aplicación móvil. Para el prototipo se desarrollará un *software* de aplicación, en este caso una aplicación web, pero este debe existir en un ordenador con un sistema operativo.

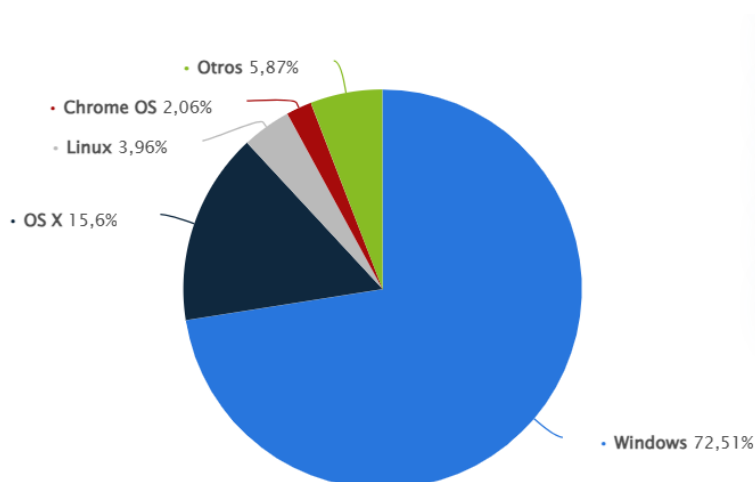
#### **Sistema operativo.**

La sugerencia para este proyecto es utilizar el sistema operativo Windows, que es el sistema operativo propio de Microsoft. La principal razón de seleccionar este sistema operativo es que el prototipo existirá en Azure, que es la nube de Microsoft (los conceptos de nube se explicarán más adelante), para aprovechar el sistema operativo nativo de la empresa. Sobre la definición de sistema operativo Bergero, *et al.* (2014) indican que “es el principal programa que se ejecuta en toda computadora de propósito general” (p. 17). Es en el sistema operativo donde se ejecutan todos los demás programas de un computador, es la base estructural lógica donde estarán los componentes que conforman el prototipo como lo serán la aplicación web y la base de datos.

El sistema operativo no solo es importante para el resultado final (el prototipo terminado), sino que también es donde se desarrolla la presente investigación y donde se instalarán las aplicaciones en las cuales se desarrollará el código del prototipo, por tanto, es fundamental en todo este proyecto. Para el desarrollo también se utilizará Windows, en este caso Windows 11, 23H2. Se eligió este sistema ya que es el de mayor uso para computadores de escritorio y por la experiencia del desarrollador del prototipo en este sistema.

**Figura 2.**

*Cuota de mercado de los sistemas operativos para ordenador a nivel mundial en el primer trimestre del 2024*



*Fuente:* Statista.com 2023

**Aplicación web.**

El principal tipo de *software* de este proyecto es la aplicación de gestión de nóminas. Existen diversos tipos de aplicación como aplicaciones de escritorio, aplicaciones web, aplicaciones móviles nativa. Para este proyecto se desarrollará el prototipo como una aplicación web. El término aplicación web es explicado por Lerma-Blasco, *et al.* (2013) de la siguiente manera: “La arquitectura de las aplicaciones web consta de máquinas conectadas a una red, por lo general, Internet o una Intranet corporativa que sigue el esquema cliente-servidor en nuestro caso de servidores web.” (p. 11). En los siguientes párrafos se aclara qué significa cliente-servidor.

La razón de optar por este tipo de aplicación es debido a que en el entorno actual la mayoría de los colaboradores cuentan con un dispositivo móvil propio. En el caso de Rastreadores Mundiales todos cuentan con un celular en el cual puede instalarse un buscador web como lo puede ser Google Chrome. Safari, entre otros. Esto permite que los colaboradores accedan desde su dispositivo a la aplicación sin tener que utilizar un computador.

El buscador web es la aplicación por la que se accede al prototipo, en este caso sería el cliente, pero, además, una aplicación web debe conectarse a un servidor web, en este caso se utilizará IIS (Internet Information Services). En esta aplicación el protocolo será el HTTPS que es una versión moderna y segura del protocolo HTTP que utiliza encriptación para la transferencia de datos. Sobre esta relación de cliente-servidor Luján (2002) explica que:

Una aplicación web es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones. (p. 48).

Con respecto al servidor IIS, se seleccionó este como la opción recomendada, debido al respaldo de los productos de Microsoft y que cuenta con amplia documentación sobre sus características, configuraciones y uso. Respecto de IIS, Microsoft (2023) menciona que: “Las opciones de configuración desatendida, línea de comandos y GUI disponibles en IIS facilitan la administración de la superficie de seguridad y mantenimiento de un servidor web de IIS.” (párr. 3). Otra razón para utilizar IIS es su integración con otras tecnologías de Microsoft como ASP.Net, además, existen servidores IIS en la nube de Microsoft Azure (más adelante se explicará el tema de la nube) que no requieren la instalación de un servidor desde cero, configurando un sistema operativo y luego el servidor web.

### **Marco de trabajo y patrón de diseño.**

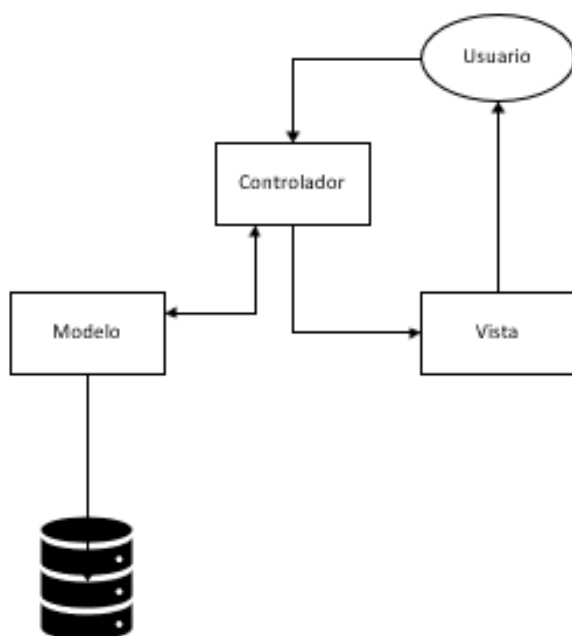
Se opta por utilizar ASP.NET MVC debido a su integración con .NET, también por la existencia de documentación sobre configuración y errores comunes. Su fabricante, Microsoft, explica ASP.NET como sigue: "ASP.NET es un marco web de código abierto, creado por Microsoft, para crear servicios y aplicaciones web modernos con .NET.". Al poder utilizarse con .NET se facilita el uso de sus librerías.

Su marco de trabajo para desarrollar aplicaciones con el patrón de diseño MVC es otra razón de uso. Montes (2015) indica que: “La separación de conceptos busca precisamente que cada “bloque” de la aplicación realice solo el trabajo que le corresponde” (p. 11). Con “conceptos” se refiere a tres elementos: modelo, vista y controlador. El modelo es donde se gestionan los datos y

está la lógica de negocio. La vista es la parte de la interfaz gráfica. Sobre el controlador Montes (2015) menciona: "Un controlador se conforma de una clase con funciones que responden solicitudes HTTP. La clase equivale al nombre del controlador y la función equivale al nombre de la acción." (p. 23). El controlador también se encarga de comunicar el modelo y la vista.

### Figura 3.

*Modelo Vista Controlador*



*Fuente:* Elaboración Propia (2024)

### Lenguajes de programación para aplicaciones web.

Es necesario explicar los lenguajes que se utilizarán para el desarrollo de este prototipo. En una aplicación web se utiliza tanto un lenguaje de programación como lenguaje de modelado. Con respecto al último también se le conoce como el diseño web, para ello se utilizará HTML que utiliza etiquetas para dar un formato a la página web. Sobre HTML Gómez, *et al.* (2018) explican que: "Cabe recalcar que no se trata de un lenguaje de programación puesto que no contiene funciones aritméticas, estructuras de control, entre otras características de un lenguaje de programación." (p. 36). HTML es al prototipo lo que el sistema óseo es para el cuerpo; es el que mantiene una estructura firme de la aplicación. En este proyecto se utilizará HTML versión 5.

Si del HTML se mencionó que es como el esqueleto, entonces sobre CSS se puede hacer la analogía con la piel. Por medio de CSS se puede controlar los colores, fuentes, márgenes y otros elementos visuales del diseño. De este lenguaje Álvarez, *et al.* (s.f.) lo describen como: "es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma" (p. 2). Usando HTML con CSS es como se crea el diseño visual de la aplicación.

Anteriormente se detallaron los lenguajes para la parte visual de la aplicación, además de ellos se ocupa un lenguaje que sirva para la lógica de la aplicación, que pueda procesar los datos y realizar los cálculos de los que se ha hablado sobre la gestión de la nómina. Para ello se utilizará Javascript, según Flanagan (2007, citado por Gómez, *et al.*, 2018): "JavaScript es el lenguaje interpretado más utilizado, principalmente en la construcción de páginas Web, con una sintaxis muy semejante a Java y a C" (p. 35). Javascript tiene la particularidad de ser un lenguaje interpretado y no necesita ser compilado previo a su ejecución.

### **Entorno de desarrollo integrado.**

El Entorno de Desarrollo Integrado (IDE por sus siglas en inglés), es otro *software* fundamental para el prototipo. En este caso su importancia es durante el desarrollo del código, por lo tanto, solamente el desarrollador es el que lo utilizará, los usuarios finales no tendrán contacto con este *software*. Una definición clara dada por González (2015) es que: "Un IDE es un programa compuesto por un conjunto de herramientas que proveen facilidades a los programadores para agilizar el proceso de desarrollo de software." (p. 13). Es en el IDE donde el programador depura el código y realiza las pruebas hasta obtener los resultados deseados en la aplicación.

Para la elección del IDE existen diferentes *softwares*, muchos de ellos tienen la capacidad de codificar en los lenguajes que se mencionaron anteriormente, se tomaron en cuenta varias opciones como Dreamweaver o JSFiddle; no obstante, se eligió Visual Studio Code 2022 de Microsoft, esto debido a la experiencia que se cuenta en la plataforma, la amplia documentación que existe de este IDE, y que tiene soporte para múltiples lenguajes. Otra razón es que al ser otro producto Microsoft se integra con otros componentes de los que se ha hablado como los son el sistema operativo Windows, el servidor web IIS en Azure y SQL como la base de datos.

## Base de Datos

Las operaciones que se realicen en la aplicación deben de tener un lugar donde guarde la información generada y donde se consuma la información que debe utilizarse para los cálculos. Para esto se utiliza una base de datos, según Gutiérrez, *et al*, (2017) se define de la siguiente manera: “Una base de datos es un conjunto ordenado y estructurado de datos que representan una realidad objetiva y que están organizados independientemente de las aplicaciones.” (p. 11). Las bases de datos son donde se guarda, se gestiona y se consulta la información.

Existen las bases de datos relacionales (SQL) y no relaciones (NoSQL), en este proyecto se utilizará una base de datos relacional, en esta la información se almacena en tablas, que representan principalmente lo que en Lenguaje Orientados a Objetos se conoce como objetos. Suárez (2008) lo explica así: "Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre: 'Modelo Relacional'"(p.1). Las tablas se conforman de columnas y filas, las columnas representan un atributo del objeto de la tabla y la fila el valor de ese atributo para cada objeto.

Para el proyecto se utilizará Azure SQL Database, esta se fundamenta en las bases de datos SQL de Microsoft, el principal motivo es la familiaridad del desarrollador con la Base de Datos de Microsoft, además que en Azure ofrece ventajas como la escalabilidad, que es la capacidad de crecer si fuera necesario, y la alta disponibilidad que se puede configurar en Azure a un bajo costo. También por la capacidad de redundancia, por ejemplo, para una base de datos de 10GB el costo de restauración de 10GB es solo de \$2 mensuales.

## Infraestructura

Seleccionar el equipo adecuado donde se instalará una aplicación es una tarea primordial. En el servidor es donde estarán tanto la base de datos como el IIS donde esté la aplicación web. Con el pasar de los años se han incrementado las posibilidades de la infraestructura por elegir para una empresa. Primero con la adquisición de máquinas físicas (*on-premise*), luego la virtualización y más recientemente la nube. En la nube existen también dos modelos de servicio que se pueden utilizar para este prototipo, IaaS (Infraestructura como servicio) y PaaS (Plataforma como servicio).

En el caso de IaaS se trata de alquilar un servidor y manejar desde su sistema operativo. Con PaaS el mantenimiento del sistema operativo y de la máquina virtual es responsabilidad de Azure, en este modelo solo se paga por el servidor web, base de datos y otros elementos que se necesiten como certificados SSL.

**Tabla 6.**

*Características de IaaS, PaaS, On-premise*

Característica	PaaS (Azure SQL Database/Managed Instance)	IaaS (Azure VM)	On-Premise
Ubicación del servidor	En la nube	En la nube	En las instalaciones propias
Mantenimiento infraestructura	Automatizado por Azure	Responsabilidad del usuario	Responsabilidad del usuario
Escalabilidad	Automática (serverless, escalado vertical)	Manual (puede ser escalado)	Manual (requiere inversión)
Disponibilidad	Alta disponibilidad y replicación geográfica	Configurable por el usuario	Configurable por el usuario
Respaldo	Automático y gestionado por Azure	Configurable por el usuario	Configurable por el usuario. Mayor labor que en IaaS
Inversión inicial	Menor inversión inicial	Costo de VM y configuración	Mayor inversión inicial
Costo mensual	Variable según uso y rendimiento	Variable según recursos	Electricidad, Aire Acondicionado.
Actualizaciones y cambios	Automático y gestionado por Azure	Configurable por el usuario	Configurable por el usuario

*Fuente:* Elaboración Propia (2024)

Entre las características más importantes y tangibles para una empresa se encuentran los costos. La estructura de costos de Azure IaaS, PaaS y *On-Premise* varía debido a que para cada posible solución se deben adquirir ciertos componentes de la nube, licencias o *hardware*. Microsoft Azure ofrece una calculadora de precios que permite revisar los precios para ambos modelos de servicio. En el caso de *On-Premise*, el costo se concentra principalmente en una inversión inicial y en gastos como electricidad y mantenimiento de instalaciones físicas. En la tabla 1 se mostró el costo de PaaS en Azure, otros precios importantes se detallan en la tabla 7.

**Tabla 7***Costos de operación en Azure IaaS*

<b>IaaS</b>	<b>VM D2 v3</b>	<b>Disco S4</b>	<b>Licencia SQL Server</b>
Costo Mensual	\$137.24	\$2.04	
Costo Anual			\$209
CPUs	2	N/A	
RAM	8 GB	N/A	
Almacenamiento	50 GB temporal	32 GB permanente	

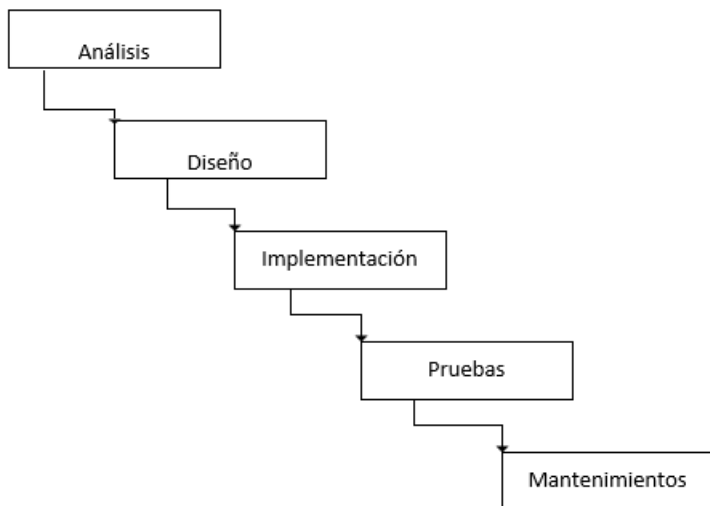
*Fuente:* Elaboración Propia (2024)

## **Metodología de Desarrollo de Software**

Para desarrollar un *software* adecuadamente, es necesario elegir la metodología con que se desarrollará. No se puede simplemente empezar a escribir líneas de código en IDE sin tener una planificación correcta. Las metodologías de desarrollo de *software* facilitan esa planificación para crear el proyecto de la mejor manera.

Existen dos corrientes principales: las metodologías ágiles y la metodología en cascada. Sobre la metodología ágil Hadida y Troilo (2020) mencionan que: "La metodología ágil fomenta la entrega de productos mínimamente viables (MVP, sigla en inglés) y la retroalimentación constante con el cliente" (p. 3). Entre sus características se destacan las iteraciones cortas para codificar y probar el producto, es más flexible, Scrum y Kanban son dos ejemplos.

Con respecto a la metodología en Cascada, se caracteriza porque se realiza un plan muy detallado antes de comenzar el código, Hadida y Troilo (2020) explican que: "El modelo en cascada es una metodología de trabajo controlada y estricta, donde se suele atribuir mayor importancia a las etapas iniciales y a la planificación del proyecto" (p. 4). Para este proyecto se decidió utilizar la metodología en Cascada porque se prioriza efectuar un análisis inicial sólido de las tareas que deben realizarse por un solo desarrollador para concluir el prototipo. Para cumplir este objetivo se enfatiza en apoyarse en casos de uso y diagramas relacionales. Esta metodología consta de las etapas que se detallan en la figura 4.

**Figura 4.***Metodología Cascada.*

*Fuente:* Elaboración Propia (2024).

En la figura. 4 se representan las cinco fases de la metodología en cascada. Lo primero es realizar un análisis exhaustivo de los requerimientos de la organización y revisar su viabilidad; en la etapa de diseño se quiere aprovechar la diagramación, como por ejemplo los casos de uso para poder planificar un orden concreto para la posterior implementación o desarrollo del código; después se procede a realizar pruebas de los diferentes módulos que se desarrollaron y, posteriormente, se realizan los mantenimientos que necesite el prototipo.

## CAPÍTULO III. MARCO METODOLÓGICO

Cuando se define el alcance de una investigación y se tiene claridad sobre un problema, luego se debe buscar la posible solución planteando hipótesis. Según Arias (2012 citado por Azuero) el marco metodológico es el “conjunto de pasos, técnicas y procedimientos que se emplean para formular y resolver problemas”. (p. 112).

En este capítulo se aborda la metodología que se aplicará en este trabajo y se repasarán algunos conceptos relacionados con la investigación, a saber: los tipos de enfoque, los tipos de investigación y cuál será el tipo de investigación predominante para este trabajo, las fuentes de información utilizadas, las variables que nos ayudarán a entender el escenario de la aplicación y los instrumentos de recolección de datos para poder definir la utilidad del proyecto.

Partiendo de las necesidades de la empresa Rastreadores Mundiales se plantea la hipótesis de una posible solución, pero es solo por medio de la investigación que se prueba si las hipótesis formuladas son eficientes para la resolución de un problema. Para empezar a desarrollar este marco se procederá a explicar los tres tipos de enfoque existentes en una investigación.

### Enfoques de Investigación

Cuando se realiza una investigación se debe elegir el tipo de enfoque que se utilizará. Es la visión o perspectiva desde la que se elige realizar el estudio del objeto. Esta selección del tipo de enfoque se plantea de acuerdo con el alcance del tema, de los objetivos y de los resultados que se espera obtener tras realizar el estudio. Existen tres tipos de enfoque que se pueden elegir, ya sea cuantitativo, cualitativo o un enfoque híbrido.

#### Enfoque cuantitativo

En este enfoque se recopilan datos numéricos y luego se utilizan estadísticas para llegar a resultados. Ávila *et al.* (2014) explican sobre este enfoque: “Confía en la medición numérica, el conteo y frecuentemente en el uso de la estadística para establecer con exactitud patrones de comportamiento en una población” (p. 525). En este proyecto se pueden realizar encuestas donde se definan *a priori* las posibles respuestas y una vez se tenga el resultado de las preguntas, se pueden agrupar y numerar y de aquí sacar conclusiones. La idea es que una vez se determinen las variables del objeto de estudio se debe elegir un sistema de medición, como por ejemplo, medir la

satisfacción de realizar cálculos manuales para obtener los salarios del 1 al 5, donde 1 es muy malo y 5 muy bueno.

### **Enfoque cualitativo**

El otro enfoque se basa en utilizar métodos como entrevistas y la observación para profundizar el conocimiento de un objeto, Ávila *et al.* (2014) explican que: “Se basa en métodos de recolección de datos sin medición numérica como la descripción y la observación del fenómeno. El proceso es flexible y se mueve entre los eventos y su interpretación” (p. 526). El enfoque cualitativo se basa en respuestas más detalladas y no numéricas, un ejemplo es el realizar entrevistas donde se permita a la persona consultada profundizar en sus respuestas.

Aquí es importante comprender que lo que para un desarrollador de *software* puede determinar la calidad de un prototipo, puede no coincidir con lo que es calidad para el cliente al que se le brindará la aplicación, Por ello, es necesario permitir a los representantes de la compañía que se expliquen en sus necesidades y entender el enfoque de negocio, para poder darles un producto que la compañía considere de calidad, ya sea por lo beneficioso, eficiencia o capacidad de solucionar problemas actuales.

### **Enfoque mixto**

El tercer enfoque es el híbrido que es una mezcla de ambos, para obtener resultados tanto de índole cualitativos como cuantitativos. También se le conoce como enfoque mixto y enfoque multimétodo. Sobre ello Ávila *et al.*- (2014) sugieren que: “Las ventajas de la investigación multimétodo son la complementariedad, ampliación de la comprensión teórica, incrementalidad de la validez y ampliación de las fronteras del conocimiento” (p. 527). Es posible que el enfoque que se elige híbrido prioriza alguno de los enfoques descritos anteriormente y que tenga en mayor medida variables cuantitativas o cualitativas.

### **Enfoque de investigación seleccionado.**

Para comprender las perspectivas de los colaboradores de Rastreadores Mundiales sobre el manejo de planilla, se propone un modelo mixto donde se entrevista a las personas encargadas de recursos humanos para obtener detalles de cómo se maneja el proceso actualmente y cómo puede mejorar, así como al gerente de la empresa. Con respecto a los colaboradores que no son parte del Departamento de Recursos Humanos pero que también realizan gestiones como recibir salario, reportar horas extras, pedir vacaciones, entre otros; a un grupo de ellos se les enviará una encuesta sobre su perspectiva de cómo se administran estas operaciones en la empresa.

### **Tipos de Investigación**

Existen cuatro tipos de investigación, cada uno de ellos presenta una capa más profunda del objeto de estudio, desde las consideraciones iniciales hasta la explicación de por qué un fenómeno ocurre. Los tipos de investigación no son excluyentes, es decir, una investigación correlacional puede tener elementos de investigación exploratorias y descriptivas. Lo más común es que en el estudio sí haya un tipo de investigación predominante.

#### **Exploratorio**

Es el tipo de investigación para aquellas hipótesis donde aún se tiene poco conocimiento como para poder describirse en detalle el fenómeno de estudio. Según Sampieri (2017) “Los estudios exploratorios sirven para preparar el terreno y, por lo común, anteceden a investigaciones con alcances descriptivos, correlaciones o explicativos” (p. 72). Es entre los diferentes tipos de investigación el que se emplea para hipótesis iniciales de un tema.

#### **Descriptivo**

Este es un enfoque en donde se pasa a describir los detalles que se han confirmado de las diferentes variables que puede tener el objeto de estudio. Citando a Sampieri (2017) “Con los estudios descriptivos se busca especificar las propiedades, características y perfiles de personas,

grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis.” (p. 76). Normalmente se realiza luego de que el objeto de estudio ha pasado por una etapa exploratoria previamente para tener certeza de las cualidades que se describen

## Correlacional

Para este tipo de investigación el enfoque es de demostrar una causa y su efecto de dos o más variables. Según Arias, Covinos y Cáceres (2020): "el fin de los estudios con alcance correlacional es asociar o relacionar las variables del estudio"(p. 243). El propósito es entender la relación de ambos objetos, existe la posibilidad de que el resultado de estos estudios sea que no haya una relación directa entre las variables de estudio.

## Explicativo

El objetivo es responder al por qué ocurre un fenómeno o cómo ocurre. Se centra en detallar en profundidad el comportamiento del objeto de estudio. Según Sampieri (2017): “están diseñados para determinar las causas de los eventos y fenómenos físicos o sociales” (p. 78). De los diferentes tipos de investigación es el que se utiliza cuando para fenómenos de los que ya se tiene más conocimiento por medio de resultados iniciales en etapas anteriores.

**Tabla 8**

### *Tipos de Estudio*

Estudio	Hipótesis	Posibles Diseños
Exploratorio	No se establecen, lo que se puede formular son conjeturas iniciales.	Transeccional exploratorio. Preexperimental
Descriptivo	Descriptiva	Preexperimental Transeccional descriptivo
Correlacional	Diferencia de grupos sin atribuir causalidad.	Cuasiexperimental Transeccional correlacional Longitudinal (no experimental)
	Correlacional	Cuasiexperimental Transeccional correlacional Longitudinal (no experimental)
Explicativo	Diferencia de grupos atribuyendo causalidad.	Experimental "puro". Cuasiexperimental, longitudinal y transeccional causal (cuando hay bases para inferir causalidad, un mínimo de control y análisis estadísticos apropiados para relaciones causales).

	Causales.	Experimental puro. Cuasiexperimentat: longitudinal y transaccional causal (cuando hay bases para inferir causalidad, un mínimo de control y análisis estadísticos apropiados para relaciones causales).
--	-----------	--

Fuente: Sampieri (2017).

### **Tipo de investigación seleccionado**

Para el proyecto se prioriza una investigación correlacional donde la hipótesis principal es el uso de la tecnología en una aplicación de gestión de nómina puede disminuir el número de errores en los cálculos de salario, vacaciones y otras operaciones relativas a planilla, debido a que omite el error humano en el procesamiento de los resultados. También, la misma causa, a saber, la utilización de una aplicación de gestión de nómina puede reducir el tiempo que conllevan estas tareas al personal de recursos humanos. Esto último con el objetivo de liberarlos para realizar otras labores. Como se mencionó, un tipo de investigación no necesariamente es excluyente, en este caso, también se plantean hipótesis de estudio exploratorio para tener unos resultados sobre el tiempo de realizar las operaciones descritas anteriormente.

### **Fuentes de información**

Para realizar una investigación se debe consultar en diferentes fuentes para poder contrastar la información ya existente de un determinado tema de estudio. Según Tamayo (1980): "Se denomina fuente o bibliografía los libros que hemos empleado y consultado para la realización de un trabajo" (p. 52). Como se verá más adelante no todas las fuentes son libros, existen revistas, videos, artículos, leyes, códigos, entre otros, que también pueden utilizarse en una investigación.

Consultar varias fuentes de diversos actores es ideal para poder tener una perspectiva más amplia sobre el objeto de estudio, esto es vital dado que, si se está investigando sobre un tema del que ya se tiene alguna base de conocimiento, es posible que se presenten sesgos o al menos una visión más estrecha sobre el tema de estudio, por lo que la revisión de material de diferentes actores puede afirmar, debatir o corregir el conocimiento que se tiene previo a la consulta de fuentes.

En el caso de un prototipo de aplicación se debe realizar un análisis de los diferentes tipos de tecnología que se podrían utilizar y así poder elegir lo que mejor se ajusta al proyecto de acuerdo con su alcance y sus objetivos. En un sistema como este donde existen diferentes componentes

como infraestructura, base de datos, lenguajes de programación, conforme ha avanzado la informática se ha profundizado más en el conocimiento de estos temas, y se han desarrollado más opciones para poder utilizarse, por ello, es importante buscar literatura de expertos de diferentes áreas.

### **Fuentes de información primarias**

Con esto se refiere a que es información que viene directamente de la fuente. Son las explicaciones de un actor sin haber sido interpretadas por alguien más. En esta investigación se utilizan mayoritariamente fuentes primarias como libros en internet. Sampieri (2017) indica sobre las fuentes primarias:

son libros, antologías, artículos de publicaciones periódicas, monografías, tesis y disertaciones, documentos oficiales, reportes de asociaciones, trabajos presentados en conferencias o seminarios, artículos periodísticos, testimonios de expertos, documentales, videograbaciones en diferentes formatos, foros, páginas y sitios en internet, etc. (p. 59).

### **Fuentes de información secundarias**

A diferencia de las fuentes primarias, aquí la información no viene directamente del actor, sino que han sido interpretadas por alguien más. Según Torres *et al.* (s.f.): “las fuentes secundarias son aquellas que parten de datos pre-elaborados, como pueden ser datos obtenidos de anuarios estadísticos, de Internet, de medios de comunicación.” (p. 3). Este tipo de fuente difiere de la primaria en que se basa el análisis y perspectiva de un actor sobre la información proveída de otro actor.

### **Fuentes de información terciarias**

Este tipo de fuentes recopilan, resumen y simplifican la información proveniente de otras fuentes; comparten con las fuentes secundarias que no contienen datos originales. Sobre estas fuentes Henríquez y Zepeda (2003) indican: “Se refieren a documentos que compendian nombres

y títulos de revistas u otras publicaciones periódicas, así como los boletines, conferencias, catálogos de libros que contienen referencias y datos bibliográficos”. Las fuentes terciarias pueden presentar la información en formatos distintos de como vengan de fuentes primarias y secundarias. Su objetivo principal es el de recoger datos sobre un objeto de estudio.

Para este trabajo mayoritariamente se usan fuentes primarias, principalmente libros. Otra fuente de información primaria que se utiliza es la perspectiva que ofrecen los colaboradores de Rastreadores Mundiales sobre la situación actual de la gestión de planilla, sobre el cómo se recabó la información de estas fuentes se explicará más adelante cuando se informe sobre los instrumentos. Además de fuentes primarias también se encontraron algunos libros, citas sobre otros actores que son relevantes para este proyecto.

## **Variables**

Cuando se realiza una investigación y se define el objeto de estudio, luego se debe de indicar de qué manera se va a examinar dicho objeto, para esto se debe seleccionar qué variables se analizarán. Es por medio de las variables que se pueden comprender el comportamiento del objeto o fenómeno de estudio. Sobre las variables Bilbao y Escobar (2020) explican que: "Se refiere a una cualidad, propiedad o característica de personas o cosas en estudio y varía de un sujeto a otro o en un mismo sujeto en diferentes momentos." (p. 32). La capacidad de medir estas variables es lo que ayudará a entender la conducta del fenómeno que se investiga. Existen tres tipos de variables que se deben explicar.

### **Variable conceptual**

Este tipo de variable se refiere a lo que se ha considerado como una característica del objeto de estudio. Se trata de esos conceptos que guiarán la investigación. Sobre este tipo de variable Bilbao y Escobar (2020) indican que: “los términos variables con los que esta enunciado el problema deben ser claros, con un contenido conceptual bien definido y sobre todo debe ser comprobable.” (p. 22). En este contexto del proyecto la “mejora en el manejo de la nómina” es un

concepto que se debe analizar para comprender si este prototipo es útil para la organización para optimizar su operación.

### **Variable operacional**

La variable operacional es la manera como se puede tomar una variable conceptual y medirla, para ello se necesitan indicadores para poder realizar esa métrica. Carballo y Guelmes (2016) explican sobre las variables operativas: “En el proceso de operacionalización de las variables es necesario determinar los parámetros de medición a partir de los cuales se establece la relación de variables enunciadas por la hipótesis, interrogante o idea a defender.” (p. 145). En esta investigación se necesita medir la variable conceptual de “mejora en el manejo de la nómina”, para ello se utilizan variables operativas como: tiempo que toma el cálculo de salarios, cantidad de errores que suceden mensualmente durante el cálculo de horas extras, entre otros.

### **Variable instrumental**

Este tipo de variable es aquella que influye en los resultados del análisis de un objeto de estudio, pero no son la variable principal de estudio. Sobre estas variables Carballo y Guelmes (2016) las describen como: "aquellas que pueden influir directamente sobre la variable dependiente y en los estudios experimentales" (p. 144). En este trabajo se presentan variables instrumentales como lo son la velocidad de la conexión de internet para poder utilizar la aplicación; la cantidad de horas extra, incapacidades, goces sin permiso de salario, puede ser diferente de un mes a otro, lo que causa que las métricas de tiempo que toma el cálculo de salarios difieran en el tiempo.

**Tabla 9***Variables para el prototipo funcional*

<b>Objetivo específico</b>	<b>Variable</b>	<b>Variable Conceptual</b>	<b>Variable Instrumental</b>	<b>Variable Operacional</b>
Analizar los requerimientos de la empresa para el manejo de nómina mediante la evaluación de la situación actual y la situación esperada.	Problemas Planteados	Según Sampieri <i>et al.</i> (2017): “El problema debe estar formulado mediante objetivos y preguntas claras, concretas y sin ambigüedades” (p. 41).	Guía de Entrevista  Guía de Encuestas  Guía de Observación	Entrevista  Encuestas  Observación
Diseñar la estructura del prototipo para el sistema de manejo de nómina según el análisis y los requerimientos de la empresa.	Arquitectura de <i>Software</i>	Según McIver (2011) define software como: “El conjunto de programas, procedimientos y documentación relacionada que asocia un sistema computacional, específicamente a parte lógica de la computadora.”	Visual Studio Code	Documentos de Análisis

Desarrollar la programación del prototipo para el cumplimiento de los requerimientos de la empresa guiándose en el diseño de la estructura del prototipo.	Aplicación Web	Según Luján (2002): “Una aplicación web es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones” (p. 48).	Visual Studio Code  Azure PaaS	Casos de Uso  Diagramas Entidad Relación
Realizar las pruebas funcionales para que se garantice el cumplimiento de los requerimientos previamente establecidos.			Guía de casos de prueba	Casos de prueba

*Fuente:* Elaboración Propia (2024).

### **Población**

La población se refiere al conjunto de objetos que comparten características y que serán estudiados en la investigación. Es importante definir el concepto con claridad. Saézn y Tames (2014) señalan: "Un primer paso es identificar y describir la población, mencionar el tamaño de esa población y establecer los medios de identificación de los individuos que la conforman." (p. 438). En este proyecto, la población la constituyen los colaboradores de Rastreadores Mundiales.

## Muestra

El tamaño de la población es variable y puede ser inmensa, por lo que en ocasiones se prefiere realizar el estudio sobre una muestra de la investigación y no en toda la población. Según Sampieri (2017): "Una muestra es un subgrupo de la población o universo que nos interesa, sobre el cual se recolectarán los datos pertinentes y deberá ser representativo de dicha población" (p. 128). Para esta investigación la muestra elegida es cinco de los treinta y cinco colaboradores que laboran en Rastreadores Mundiales, este valor se toma aplicando la siguiente fórmula que se explica en detalle en la Tabla 10.

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

**Tabla 10**

*Variables del Cálculo de Muestra*

Variabl	Descripción
e	
n	tamaño de la muestra
N	tamaño de la población
K	nivel de confianza
p	proporción esperada
q	probabilidad de fracaso
e	precisión (margen de error)

*Fuente:* Aguilar (2024).

Para despejar la fórmula y obtener el tamaño de la muestra es necesario tener el total de la población. Luego la variable p, de probabilidad esperada suele ser uno, y lo mismo sucede que q, la probabilidad de fracaso. Para "e" se tomó también uno como el valor por utilizar, ya que es un valor bajo. Para obtener el valor de K es necesario referenciar la tabla 11.

**Tabla 11***Cálculo de nivel confianza*

<b>Nivel de Confianza Deseado</b>	<b>Valor de K</b>
80%	1.28
85%	1.44
90%	1.65
95%	1.96
99%	2.58

*Fuente:* Aguilar (2024).

Luego de definir el valor de confianza deseado para saber el valor de K, se procede a aplicar la fórmula descrita anteriormente. En la tabla 12 se muestra el valor de cada variable para realizar el cálculo correspondiente, que resulta en un valor de 3.55, por ellos se procede a definir que la muestra de colaboradores de Recursos Humanos será de 4 personas. Ya definida la muestra se debe continuar con la creación y aplicación de un instrumento de investigación que se verá más adelante.

**Tabla 12***Cálculo de Muestra*

<b>Variable</b>	<b>Valor</b>
N	?
N	35
K	1.96
Q	1
P	1
E	1

*Fuente:* Elaboración Propia (2024)

### **Instrumentos de Recolección de Datos**

Una vez seleccionada la muestra para la investigación, se debe proceder a definir de qué manera se recopilarán los datos que se estudiarán de dicha muestra. Con esta información se pretende obtener las métricas de las variables operativas. Según Bustamante *et al.* (2023): "Un instrumento de investigación es una herramienta específica utilizada para recopilar y analizar

información en el proceso de investigación. Estos instrumentos pueden incluir fichas de cotejo, cuestionarios, escalas de medición, fichas de entrevistas estructuradas, pruebas estandarizadas, entre otros." (p. 12). Existen diferentes tipos de instrumentos que se explicarán a continuación:

### **La observación**

Una manera de estudiar un objeto es mediante la observación de este, de sus características y sus comportamientos. Bustamante *et al.* (2023) indican que: "La técnica de observación es un método de investigación en el que se registra y analiza el comportamiento y las acciones de individuos, grupos o fenómenos en su entorno natural" (p. 20). En este proyecto se pretende observar y contabilizar el tiempo que toma el colaborador de recursos humanos en realizar un cálculo de salarios de nómina.

### **Encuesta**

Para recabar información desde la perspectiva de una muestra sobre una variable específica se puede acudir a una encuesta, en ella se delimitan las respuestas a que sean concisas. Saénz y Tames (2014) describen la utilidad de las encuestas: "Uno de los diseños ampliamente utilizados en ciencias sociales es el realizado a través de encuestas, las cuales proporcionan una descripción cuantitativa o numérica de tendencias, actitudes u opiniones de una población." (p. 438). En este caso es importante para Rastreadores Mundiales comprender qué piensan los trabajadores de su empresa sobre la manera en la que actualmente se realizan trámites como la solicitud de vacaciones o si han tenido alguna vez problemas con el pago del salario debido a que el monto no era el correcto.

### **Entrevistas**

En las entrevistas hay ocasiones en las que se prefiere que el individuo consultado realice una explicación más amplia de su perspectiva sobre algún objeto o fenómeno, para esto son útiles las entrevistas. Paz y Torres (s.f.) explican sobre la entrevista: "Es una conversación generalmente entre 2 personas, (uno el entrevistador y otro el entrevistado). Las preguntas pueden ir registradas

en una boleta que se llama cuestionario o bien se puede auxiliar de una grabadora para registrar los datos obtenidos." (p. 6). Para entender el sentir de los colaboradores de recursos humanos de Rastreadores Mundiales se realizará una encuesta para determinar qué les parece desafiante del proceso actual de gestión de nómina. Se pretende realizar varias consultas a través de las cuales el entrevistado desarrolle su pensar con respecto a este tema para comprender la situación actual.

### **Proceso para la Recolección y Análisis de Datos**

Para este proyecto los primeros datos relevantes se obtuvieron por medio de una entrevista inicial no estructurada donde se consultó al Gerente General si había algún tipo de operativa que se pudiera ver beneficiada al utilizar una aplicación, se sugirió específicamente que se contemplara la posibilidad en que una aplicación web podría ayudar con el manejo de nómina. Con respecto a esto las impresiones iniciales es que al utilizar un método manual para calcular salarios se consumía más tiempo que si existiera un método automático de realizar los cálculos.

Luego de esta entrevista inicial se recurrió a observar la operativa por medio de una sesión remota. Durante este proceso de observación se constató que los cálculos de salarios se realizan en Microsoft Excel, pero hay datos que se ingresan manualmente cada vez, como el porcentaje de deducciones que se ingresa en la celda de cálculo (no se está utilizando una fórmula de Excel donde existan celdas que guarden constantes).

Durante la observación que se realizó por un tiempo aproximado de una hora, se alcanzó a calcular solo siete salarios en alrededor de cincuenta y cinco minutos, para un promedio de casi ocho minutos para calcular el salario de un colaborador, Se determinó que una razón para ralentizar este proceso es que hay componentes salariales que no están debidamente documentados en el mismo documento de Excel, un ejemplo de esto son las horas extras, puesto que muchas veces solamente quedan documentadas en conversaciones vía mensajería por WhatsApp y esto implica que cada pago se debe revisar una por una las diferentes conversaciones para confirmar las horas extras aprobadas.

En una siguiente conversación con la gerencia de la empresa se le consultó sobre que oportunidades de mejora han notado con la gestión de nómina y la respuesta fue que regularmente ha habido errores con el cálculo del salario que posterior al pago deben corregirse, lo que implica un doble trabajo.

Cuando se consultó sobre el origen de los errores, la afirmación que brindó la gerencia fue que sucede en cálculos con respecto a incapacidades, comisiones y horas extras. Se preguntó basándose en el ejemplo de horas extras, que se mantienen documentadas en conversaciones vía Whatsapp de teléfono si eso puede ocasionar problemas y la respuesta fue que sí, que es un problema mayor porque ellos consideran que la documentación no está unificada en un solo lugar, y por esa razón el proceso se vuelve lento y con mayor probabilidad de errores.

En esa segunda entrevista también se consultó sobre cuánto tiempo estiman o tienen documentado que se toma el cálculo de salarios, la respuesta fue que creen que toma casi una jornada laboral entera de la persona encargada. Cuando se mencionó el beneficio de reducir ese tiempo de labor del personal de recursos humanos para otras labores importantes, la respuesta fue afirmativa, puesto que este colaborador también tiene otras tareas que realizar y no solo con respecto a la nómina.

Algunas otras consultas abarcaron el tema del cálculo de liquidaciones e informaron que se hace utilizando Excel y la calculadora de Windows. En general, la gerencia espera disminuir el tiempo de cálculo de operaciones de gestión de nómina, reducir los errores cometidos durante los cálculos y centralizar la información que manejan en diferentes hojas de cálculo y archivos de texto, en un solo lugar. Por último, se presentó como propuesta un cuadro de costos y la gerencia estuvo de acuerdo con el precio, sin embargo, se comunicó que se seguirá trabajando en la investigación y con esto determinar si hay alguna otra manera menos costosa utilizando Azure PaaS o si en todo caso usando otro proveedor de la nube, pero con IaaS.

En los siguientes días se va a realizar una encuesta tomando una muestra de cuatro colaboradores, donde se va a consultar sobre los siguientes temas en específico: la frecuencia de errores en el cálculo de salario en el último año; problemas con la documentación de las vacaciones; problema de inconsistencia de horas extras reportadas; deficiencias con el cálculo de comisiones; errores en el cálculo de días feriados trabajados. Además, se consultará sobre el nivel de satisfacción de 1 a 5 con respecto a la operativa actual del pago de salarios, de la manera como se manejan los pedidos y confirmación de vacaciones, además de las horas extras.

Cuando se comparta el prototipo, se espera una nueva entrevista para entender el nivel de satisfacción del uso de la aplicación con respecto a la mitigación de los problemas actuales, a saber: disminuir tiempo de cálculos, reducir errores, y de tener la información centralizada. Se consultará y pedirá retroalimentación sobre el rendimiento de la aplicación, sus opiniones sobre la interfaz

gráfica (diseño), cuán amigable ha sido para los usuarios y si el entrenamiento y explicaciones de uso solucionan sus dudas y es robusto o no.

## CAPÍTULO IV. ANÁLISIS DE RESULTADOS

Para este proyecto los primeros datos relevantes se obtuvieron por medio de una entrevista inicial no estructurada donde se consultó si había algún tipo de operativa que se pudiera ver beneficiada al utilizar una aplicación, se sugirió específicamente que se contemplara la posibilidad de que una aplicación web podría ayudar con el manejo de nómina. Con respecto a esto, las impresiones iniciales es que al utilizar un método manual para calcular salarios se consumía más tiempo que si existiera un método automático de realizar los cálculos.

Luego de esta entrevista inicial se recurrió a observar la operativa por medio de una sesión remota. Durante este proceso de observación se constató que los cálculos de salarios se realizan con la herramienta Microsoft Excel, pero hay datos que se ingresan manualmente cada vez, como el porcentaje de deducciones que se ingresa en la celda de cálculo (no se está utilizando una fórmula de Excel donde existan celdas que guarden constantes).

Durante la observación realizada por una hora se pudieron calcular apenas siete salarios en alrededor de cincuenta y cinco minutos, para un promedio de casi ocho minutos para calcular el salario de un colaborador, Se determinó que una causa de la ralentización de este proceso es que hay componentes salariales que no están debidamente documentados en el mismo archivo de Excel; un ejemplo de esto son las horas extras, puesto que muchas veces solamente quedan documentadas en conversaciones de mensajería por WhatsApp y esto implica que cada pago hay que revisar las diferentes conversaciones para confirmar las horas extras aprobadas.

En una siguiente entrevista con la gerencia de la empresa se consultó lo siguiente

**Pregunta 1: ¿utiliza una sola herramienta para la gestión de nómina como cálculos de salarios, aguinaldos, liquidaciones?**

Es necesario entender la operativa actual para así poder entender a fondo la problemática y proponer soluciones. Para la gestión de nómina se utilizan diferentes herramientas, principalmente las hojas de cálculo de Microsoft Excel, sin embargo, no tienen centralizadas todas las operaciones de nómina en un solo Excel, sino que existen uno o varios documentos para el cálculo de salarios (varios porque usan un documento para cada mes en el caso de los salarios), gestión de vacaciones,

gestión de permisos. Además, utilizan WhatsApp como herramienta para solicitar vacaciones y permisos, y es por este mismo medio que se autorizan. Dicha situación es un indicador de que centralizar todas las operaciones de gestión de nómina en una sola herramienta le ayudaría a Rastreadores Mundiales a mantener un mayor orden en la administración del recurso humano.

### **Pregunta 2 ¿se comenten errores frecuentes en el cálculo de salario?**

En el entendido de que sí existen oportunidades de mejoras en las operaciones de gestión de nómina desde la observación realizada, se consultó en la entrevista cuáles son los mayores problemas que se han notado y la respuesta fue que regularmente ha habido errores con el cálculo del salario que luego deben corregirse, lo que implica un doble trabajo.

Cuando se consultó sobre el origen de los errores, la afirmación de la gerencia fue que suceden por cálculos con incapacidades, comisiones y horas extras. Se preguntó si tomando específicamente como ejemplo las horas extras documentadas en conversaciones de teléfono, si eso puede ocasionar problemas y la respuesta fue afirmativa, pues es un problema mayor porque ellos sienten que la documentación no está en un solo lugar, y que eso hace el proceso más lento y que puedan ocasionar errores.

### **Pregunta 3 ¿Se han presentado errores en el cálculo de horas extras?**

Sabiendo que hay errores en el cálculo de salarios y que se utiliza WhatsApp como herramienta para solicitar y aprobar horas extras, se consulta si estos cálculos han presentado errores. La respuesta es que es frecuente que haya errores en documentación de las horas extras y que debido a esto se presentan errores en el cálculo total del salario. Muchas veces se aprueba la hora extra en WhatsApp; pero luego, en el momento de realizar los cálculos de salario se olvidan de ingresar estas horas extras.

### **Pregunta 4. ¿Cuánto tiempo cree que se toma el cálculo de salarios?**

Para entender si el prototipo trae beneficios a la empresa se toma en cuenta que disminuir el tiempo de cálculos de salario, vacaciones, liquidaciones y otras operaciones podría resultar en un beneficio tangible para la compañía. Se consultó sobre cuánto tiempo creen que se toma el cálculo de salarios y la respuesta es que creen que toma casi una jornada laboral entera de la persona encargada. Cuando se mencionó que si disminuir ese tiempo podría liberar al encargado de recursos

humanos para otras labores importantes la respuesta fue afirmativa puesto que este colaborador también tiene otras tareas por realizar y no solo con respecto a la nómina.

**Pregunta 5 ¿Tienen sistemas informáticos, de qué tipo de sistemas son y dónde los tienen?**

La razón de esta pregunta es entender si ya Rastreadores Mundiales cuenta con experiencia en el uso de sistema informáticos y si tienen servidores que se puedan utilizar para este prototipo. Rastreadores Mundiales sí cuenta con un servidor donde tienen su principal aplicación de geolocalización, que la tienen en la nube con un proveedor alemán. No cuentan con servidores físicos o en la nube que se puedan utilizar en este momento para poder tener el prototipo.

**Pregunta 6. ¿Tiene alguna preferencia sobre la nube o infraestructura física?**

Esta pregunta se planteó antes de saber que la empresa tenía un servidor en la nube. En un principio, se consultaría para saber si estaban anuentes a utilizar la nube, en el entendido de que esta tecnología lleva menos años en el mercado que los servidores físicos y puede que haya empresas que aún no se sientan cómodos con ella. De la pregunta anterior se desprendió que sí tenían experiencia en la nube; de hecho se comentó que les ha sido beneficioso contar con esta infraestructura, por lo que se explicó que se revisarían los precios físicos y los de la nube.

**Pregunta 7. ¿Es viable utilizar un sistema PaaS en donde el mantenimiento del sistema operativo lo hace el proveedor de la nube, pero por un costo mayor?**

En el caso de los sistemas de la nube PaaS no se necesita tener un servidor sino que la aplicación corre en un contenedor, pero estos precios pueden ser más altos que en un servidor físico o un servidor en la nube, sin embargo esta infraestructura omite los costos de tiempo de mantenimiento de un servidor. Se informó que se están realizando los cálculos de costos para presentar la propuesta y que la razón de la pregunta es si estarían dispuestos a aumentar el costo pero sabiendo que no deben invertir en el mantenimiento del servidor.

La respuesta fue que depende del monto, pero que, si se consigue un buen precio, sería factible y preferible este tipo de infraestructura para no tener que estar planificando y realizando mantenimiento al servidor, ya que son tareas complejas y no se cuenta con mucho personal para este tipo de labores.

**Pregunta 8 ¿Estaría de acuerdo con que los colaboradores usen su propio teléfono para gestionar sus operaciones como pedir vacaciones?**

El motivo de esta pregunta es entender si se percibe algún problema operativo en que los colaboradores usen su propio celular, y no una computadora física de la empresa, para realizar las gestiones de nómina, principalmente al usuario final para solicitar vacaciones, permisos y otros. Rastreadores Mundiales prefiere que sí se realicen las gestiones por medio de los dispositivos de cada colaborador, para que estos puedan tener mayor facilidad desde cualquier lugar que la realicen y en el momento que deseen hacerla.

**Pregunta 9. ¿Cuáles son las operaciones de nómina que le interesan para una aplicación?**

Con esta consulta se pretende entender todo el alcance del prototipo y su factibilidad técnica. La respuesta es que las operaciones que les gustaría que se incluyeran en el prototipo son: cálculo de salarios, cálculo de aguinaldos, solicitud y aprobación de vacaciones, gestión de permisos, gestión de incapacidad, y cálculo de liquidaciones.

## **CAPÍTULO V. PROPUESTA**

### **Análisis**

#### **Análisis Detallado del Prototipo Desarrollado**

En este capítulo se describe la propuesta del prototipo. En esta sección se describe el alcance de cada uno de los módulos y cómo deben funcionar.

#### **Módulo Gestionar Nómina.**

Este módulo se encarga de realizar el cálculo de los salarios de los colaboradores de manera automática. A nivel de reporte se puede consultar los salarios desde un rango de fechas en cualquier momento que se desee pero a nivel de guardar el cálculo de planilla solo se mostrará el botón de guardar para los días 15 y 30 de cada mes, y si ese día es un domingo, entonces se habilitará un día antes, en el caso de Febrero son los días 15 y 28. Si el botón de guardar planilla está habilitado, al presionarlo se validará en base de datos si existe un cálculo salarial para esas fechas y de ser así no permitirá guardar un segundo cálculo.

Para realizar el cálculo de planilla se consultará para cada usuario activo el salario bruto, sobre esto se recalculará el salario neto tomando en cuenta las deducciones por ley y cargas sociales, así como las ausencias sin goce salarial y las incapacidades. También se aumentará el salario neto si existiesen horas extras aprobadas y comisiones durante el rango de fechas del cálculo salarial. Estos rubros conforman el salario neto

#### **Módulo Gestionar Vacaciones.**

Este módulo deberá agregar la cantidad de días de vacaciones a cada colaborador de manera automática, tomando en cuenta lo establecido por ley. Además, en este módulo el colaborador puede solicitar las vacaciones y obtener la lista de sus solicitudes antiguas y vigentes. Para que el pedido de vacaciones se pueda realizar el usuario debe seleccionar una fecha inicial y una fecha

final. El sistema este debe revisar si ya existen otros pedidos de vacaciones o permisos de ausencia en ese rango de fechas, y de existir no podrá crearse el pedido. El sistema también deberá validar que la fecha final sea mayor que la fecha inicial. Para el cálculo de la cantidad de días de vacaciones solicitadas debe tomarse en cuenta si en el rango de fechas seleccionado existen días donde el colaborador no labore, o que sean días feriados y de ser así esos días no deben sumarse a la cantidad de días de vacaciones solicitados.

Estas solicitudes de vacaciones aparecerán en el módulo de vacaciones del usuario administrador si la solicitud es de un supervisor y deberá mostrarse a los supervisores en caso de que el pedido haya sido realizado por un colaborador. En ambos casos se notificará por medio de una alerta de que existen pedidos pendientes, el usuario aprobador (administrador o supervisor) deberá proceder a su aprobación o rechazo. En ambos casos deberá cambiar el estado de la solicitud y se mostrará una alerta en la pantalla del usuario quien realizó la solicitud. Si el pedido fue aprobado entonces se deberán descontar esos días de vacaciones del usuario.

### **Módulo Gestionar Horas Extra.**

Este módulo calculará el monto de pago por horas extras laboradas que se deberá añadir al cálculo final del salario de cada colaborador. Para esto los usuarios deberán marcar cada día la hora de entrada y hora de salida utilizando la interfaz de marca entrada/salida. El sistema deberá realizar un cálculo en el que a la hora de salida se le resta la hora de entrada, si el resultado es igual o mayor que 10 horas entonces el sistema automáticamente creará un pedido de horas extra. Este pedido de hora extra deberá luego mostrarse en pantalla para el usuario supervisor o administrador, quien deberá aprobar o rechazar las horas extra. Si las horas extra fueran aprobadas deberán tomarse en cuenta para el cálculo de planilla.

### **Módulo Gestión de Liquidaciones.**

En este módulo el usuario de recursos humanos (RRHH) o el usuario administrador podrán ver la lista de usuario y también podrán realizar búsqueda por nombre de usuario. En una tabla con el nombre de usuario y el botón de “liquidar”, el usuario administrador o de RRHH presionará el botón y podrá seleccionar el tipo de liquidación que son: renuncia, despido sin responsabilidad patronal y despido con responsabilidad patronal. Luego se presionará el botón de liquidar y se realizará automáticamente el cálculo del monto de liquidación. Para ellos se calcularán los días que aún no han sido pagados de la quincena actual y los días de vacaciones pendientes. Además, si la liquidación es con responsabilidad patronal, entonces el sistema realizará un cálculo del promedio de los últimos meses de salario y con ello se calculará la cesantía y la pasantía. Se sumarán los subtotales para definir el monto final. Además, al liquidar el usuario este pasará a estado inactivo, y la bitácora del usuario llenará el campo de fecha de salida.

### **Módulo Gestionar Incapacidades.**

Este módulo deberá registrar las fechas de incapacidad de los colaboradores. Para ello usuario podrá llenar el formulario de incapacidad. Tanto el supervisor como el administrador tendrán la opción de búsqueda de usuario en el formulario, en caso de que la situación del empleado incapacitado no le permita realizar la gestión. Luego esas incapacidades deberán mostrarse para los usuarios de recursos humanos, quienes tendrán la posibilidad de aceptar o rechazar la incapacidad. En caso de ser aceptada la incapacidad se deberá tomar en cuenta para el cálculo de planilla.

### **Módulo Gestionar Aguinaldo.**

Este módulo permitirá al usuario de recursos humanos y al usuario administrador utilizar el sistema para realizar un cálculo automático del aguinaldo para cada colaborador, tomando en cuenta los salarios y la cantidad de meses laborados. El cálculo se realiza desde diciembre del año anterior a noviembre del año presente. También permitirá crear reportes del pago de aguinaldo de otros años y podrá descargarlo a Excel.

### **Módulo Gestionar Comisiones.**

En este módulo se registran las comisiones cada colaborador, que se tomarán en cuenta para el cálculo de salario. Para el registro de comisiones solamente dos usuarios tienen permiso de realizarlas: El usuario supervisor podrá registrar comisiones para los colaboradores y el usuario administrador podrá guardar las comisiones para los usuarios supervisores. Para rastreadores mundiales solamente existe la comisión por venta.

### **Módulo Gestionar Colaboradores.**

Este módulo se encarga del mantenimiento de los usuarios, para poder realizar inserción, edición y borrado de datos. A este módulo solo tendrá acceso el usuario administrador. Cuando se crea un nuevo usuario, el sistema creará automáticamente un horario de trabajo en el que se muestran preseleccionados los días de lunes a sábado, esto debido a que es el horario común en Rastreadores Mundiales, sin embargo, el usuario administrador podrá cambiar los días de trabajo para el colaborador de ser necesario. Una vez creado el usuario y el horario de trabajo, ambos pueden ser editados por el administrador. Por último, cuando se registra un nuevo usuario el sistema creará una nueva bitácora con el Id del usuario y la fecha de entrada.

### **Módulo Consultar.**

En los módulos se muestra un listado de sus objetos, es decir, el módulo de usuario muestra una lista de usuarios, el módulo de vacaciones muestra los pedidos de vacaciones existentes, y lo mismo sucede con los otros módulos. En el caso del usuario administrador, este tendrá acceso a la información de todos los usuarios. Para los supervisores solo podrán ver la información de los colaboradores. Además, cada usuario puede ver sus propios pedidos, y editarlos y modificarlos.

## **Módulo Reportes.**

En los módulos de planilla, aguinaldos, pedidos de vacaciones, permisos se tendrá la opción de generar un archivo Excel al presionar el botón de “Generar Excel”, el cual se descargará a la carpeta predeterminadas de descargas.

## **Módulo Seguridad**

Este módulo se encarga de realizar la autenticación de los usuarios mediante la validación de su usuario y contraseña. Además, como parte de la seguridad de la aplicación se ha utilizado la función de *Authorize* de C#, con el que se puede definir cuales roles de usuario pueden ejecutar acciones específicas dentro de la aplicación, esto con el fin de evitar que un usuario puede buscar la *URL* de una función y acceder a ella, aún cuando esta no se muestre en su interfaz gráfico de la pantalla de inicio.

## **Análisis Detallado del Hardware Requerido**

En esta sección se analiza cuál es el *hardware* requerido para el prototipo. En la Tabla 7, relativa a costos de operación en Azure IaaS, se especificó que se utilizaría infraestructura en la nube y no propiamente un servidor en la nube. La diferencia principal radica en que en IaaS el mantenimiento del sistema operativo no es necesario, ya que de esto se encarga el proveedor del servicio, en este caso Azure, por tanto, no se invirtió en algún servidor que albergue la aplicación o base de datos.

En cuanto a los dispositivos clientes, se acordó que lo mejor es utilizar los propios dispositivos móviles de los colaboradores, esto por cuanto es una aplicación web; además de que proporciona mayor comodidad a los propios colaboradores, quienes no necesitarían conectarse a una máquina de escritorio o portátil para realizar sus gestiones. De igual manera, Rastreadores Mundiales cuenta con más de una computadora de escritorio en sus instalaciones, esto en caso de que a algún colaborador téngase le presenten problemas al acceder desde su dispositivo móvil y, primordialmente, son estas computadoras de escritorio donde los usuarios administradores se conectarán para realizar sus gestiones diarias.

**Tabla 13**

*Dispositivos utilizados por usuarios administradores del prototipo*

Recurso	Valor
Tipo de Computadora	Escritorio
Procesador	Intel Core I5
Memoria RAM	8GB a 16GB
Disco Duro	200GB a 1TB
Tipo de procesador	64 bit

*Fuente:* Elaboración Propia (2024)

### **Análisis Detallado de los Elementos de Telecomunicaciones**

Al ser este un prototipo funcional de aplicación web, es necesario utilizar la red para conectarse. A nivel del servidor, al estar en Azure IaaS, la red es manejada por el proveedor, por lo que se garantiza que esta red es fiable. Con respecto a los usuarios administradores que utilizarán mayoritariamente las computadoras de escritorio de la compañía, en este caso sí es importante que haya una red estable para evitar interrupciones; en la actualidad el proveedor de esta red es la compañía Liberty. En cuanto a los dispositivos de los colaboradores que no son administradores, la fiabilidad dependerá del proveedor de internet del dispositivo o de la red inalámbrica a la que se conecten. En el caso que lo hagan desde la oficina, entonces es la misma red que utilizan los usuarios administradores, si lo hacen desde otro lugar la conexión depende de la red a la que se conecten.

### **Descripción Detallada de la Base de Datos**

Para el almacenamiento de datos se utilizará una base de datos Microsoft SQL Server, principalmente por ser una base de datos robusta y también por su integración con Visual Studio. Para el prototipo la conexión a base de datos se utilizará ADO, y para la gestión de tablas y procesos almacenados se utilizará SQL Server Management Studio que proporciona una interfaz gráfica que facilita las labores de base de datos.

### **Descripción de los Conocimientos de Recursos Humanos para el Uso del Prototipo**

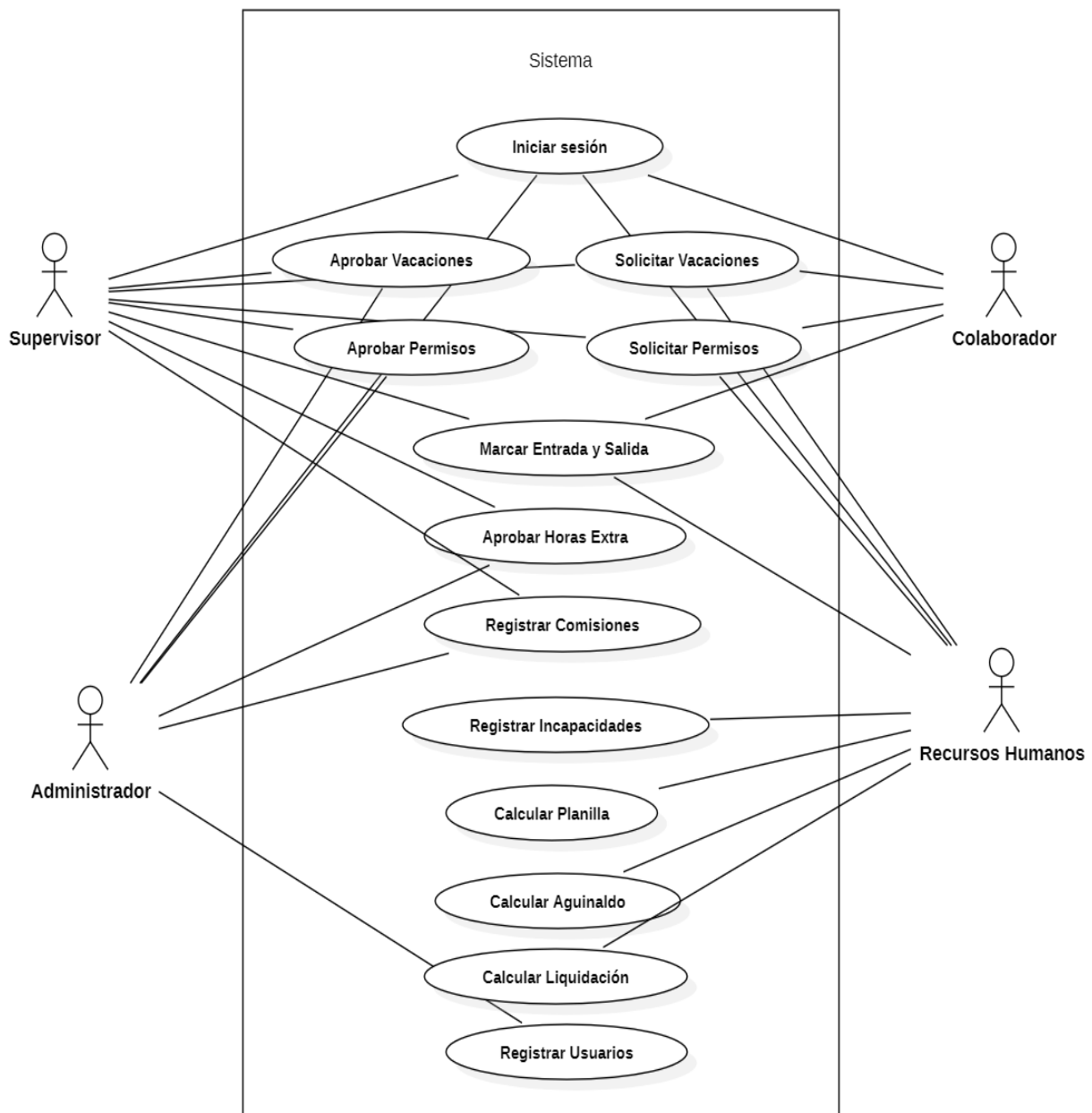
Con el fin de que el prototipo pueda pasar a producción en Rastreadores Mundiales, se prevé que deberá realizarse una capacitación para los usuarios de la aplicación. Con respecto al usuario colaborador, los módulos a los que tendrá a ingresar información son el de vacaciones y el de permisos, el resto de interacción es principalmente de reportería. En el caso de los supervisores también podrán ingresar horas extra y comisiones. Para los administradores de recursos humanos sí tienen acceso a todos los módulos. Por lo tanto, se plantea que se debe asegurar primero el correcto uso por parte del departamento de Recursos Humanos, luego de los supervisores y, por último, los del usuario final. Para que los usuarios puedan adquirir este conocimiento se planteará luego de finalizado el proyecto las mejores fechas para empezar con el proceso de capacitaciones.

## Casos de Uso

En esta sección se muestra el diagrama de casos de uso, y los casos de uso para cada módulo del prototipo.

**Figura 5.**

*Diagrama de Casos de Uso*



*Fuente:* Elaboración Propia (2024)

**Tabla 14.***Caso de uso Mantenimiento Usuarios*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa Rastreadores Mundiales, ubicada en Desamparados de San José</b>	
<b>Número caso de uso:</b> 001	<b>Nombre del caso de uso:</b> Mantenimiento Usuarios
<b>Fecha elaboración:</b>	09/08/2024
<b>Descripción caso de uso:</b>	El administrador ingresa, edita o desactiva empleados.
<b>Autor caso de uso:</b>	Usuario administrador
<b>Actores relacionados:</b>	Colaboradores, administradores
<b>Precondiciones:</b>	Usuario administrador se ha autenticado.
<b>Flujo básico del caso de uso</b>	
<ol style="list-style-type: none"> <li>1. El administrador ingresa al menú de Usuarios.</li> <li>2. El administrador ingresa un nuevo usuario <b>Sub Flujo 01.</b></li> <li>3. El administrador edita la información del usuario <b>Sub Flujo 02.</b></li> <li>4. El administrador elimina el usuario. <b>Sub Flujo 03.</b></li> <li>5. El sistema procesa el nuevo colaborador y verifica si existe.</li> </ol>	

6. Si el colaborador no existe, entonces el sistema lo guarda en base de datos.	
<b>Subflujos</b>	
<b>Subflujo 01. Registrar usuario</b>	El administrador presiona en el botón de “Agregar nuevo”. Ingresar los datos de nombre, correo, teléfono, fecha de entrada, el salario base, el puesto de trabajo. Si el correo ha sido registrado anteriormente, procede <b>Flujo alternativo 01</b> .
<b>Subflujo 02. Editar usuario</b>	El administrador presiona el botón de “editar” al lado del usuario que desea editar. Una pantalla con la información del usuario se despliega. El administrador edita las casillas disponibles. El administrador presiona el botón de “Guardar cambios”.
<b>Subflujo 03. Eliminar usuario</b>	El administrador presiona el botón de “Eliminar” al lado del usuario que desea eliminar. El usuario se desactiva en el sistema.
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1</b>	Si el colaborador existe, entonces el sistema deberá mostrar un mensaje de error.
<b>Requerimientos especiales</b>	

<b>Postcondiciones</b>	
El colaborador se registra, se edita o se desactiva del sistema.	

Fuente: Elaboración propia (2024).

**Tabla 15.**

*Caso de uso Autenticación de Usuario*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa Rastreadores Mundiales, ubicada en Desamparados de San José</b>	
<b>Número caso de uso:</b> 002	<b>Nombre del caso de uso:</b> Autenticación de usuario
<b>Fecha elaboración:</b>	09/08/2024
<b>Descripción caso de uso:</b>	El usuario se autentica en el sistema para acceder a sus funcionalidades.
<b>Autor caso de uso:</b>	Randall Fernández Meza
<b>Actores relacionados:</b>	Colaboradores, administradores
<b>Precondiciones:</b>	El usuario que se va a autenticar ya existe en el sistema y está activo.
<b>Flujo básico del caso de uso</b>	
<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación.</li> <li>2. La aplicación muestra la pantalla de inicio de sesión.</li> <li>3. El usuario ingresa su nombre de usuario (podría ser el</li> </ol>	

<p>correo electrónico) y contraseña.</p> <p>4. El sistema verifica las credenciales: Si la información es correcta corre el subflujo 01. Si la información es incorrecta, el sistema corre el flujo alternativo no. 2</p>	
<b>Subflujos</b>	
<b>Subflujo 01 Ingresar al sistema</b>	El sistema permite al usuario navegar por los menús del sistema.
<b>Subflujo 02 Cerrar Sesión</b>	El usuario presiona el botón de “Cerrar Sesión”.
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1</b>	Si el usuario olvidó su contraseña puede solicitar que la contraseña se restablezca mediante un botón en la pantalla de inicio de sesión.
<b>Flujo Alternativo No. 2</b>	Si el sistema verifica que las credenciales son incorrectas, se muestra un mensaje de error y se solicita al usuario que ingrese credenciales de nuevo. Si el error de credenciales en una cuenta sucede 5 veces entonces se ejecuta Flujo Alt No.3
<b>Flujo Alternativo No. 3</b>	Si el usuario ingresa la contraseña incorrecta más de 5 veces seguidas, entonces el sistema lo llevará a la pantalla de “Restablecer contraseña”.

<b>Postcondiciones</b>	
El usuario se ha autenticado y puede utilizar la aplicación.	

Fuente: Elaboración propia (2024).

**Tabla 16.**

*Caso de uso Restablecer Contraseña*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa Rastreadores Mundiales, ubicada en Desamparados de San José</b>	
<b>Número caso de uso: 003</b>	<b>Nombre del caso de uso:</b> Restablecer contraseña
<b>Fecha elaboración:</b>	09/08/2024
<b>Descripción caso de uso:</b>	Proceso mediante el cual un usuario que ha olvidado su contraseña puede solicitar una nueva contraseña y acceder nuevamente a su cuenta.
<b>Autor caso de uso:</b>	Colaborador registrado
<b>Actores relacionados:</b>	Colaboradores, administradores
<b>Precondiciones:</b>	El usuario que va a restablecer la contraseña ya existe en el sistema y está activado.
<b>Flujo básico del caso de uso</b>	
1. El usuario visita la página de inicio de sesión.	

2. El usuario hace clic en el enlace o botón “¿Olvidaste tu contraseña?”.
3. El sistema muestra un formulario donde el usuario debe ingresar su dirección de correo electrónico asociada con la cuenta.
4. El usuario ingresa su dirección de correo electrónico y envía la solicitud.
5. El sistema verifica si la dirección de correo electrónico existe en la base de datos.
6. Si la dirección de correo electrónico es válida, el sistema genera un token de restablecimiento de contraseña y lo envía al correo electrónico del usuario.
7. El usuario recibe el correo electrónico con el enlace para restablecer la contraseña.
8. El usuario hace clic en el enlace y es redirigido a una página donde puede ingresar una nueva contraseña.
9. El sistema valida el token y permite al usuario ingresar una nueva contraseña.
10. El usuario ingresa la nueva contraseña y confirma.

11. El sistema actualiza la contraseña en la base de datos y muestra un mensaje de éxito al usuario.	
<b>Subflujos</b>	
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1</b>	Si la dirección de correo electrónico no existe en la base de datos, el sistema muestra un mensaje de error y solicita al usuario que verifique la dirección ingresada.
<b>Requerimientos especiales</b>	
<b>Postcondiciones</b> El usuario puede iniciar sesión con la nueva contraseña.	

Fuente: Elaboración propia. (2024)

### Tabla 17.

*Caso de uso Calcular Planilla*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa</b>	
--	--

<b>Rastreadores Mundiales, ubicada en Desamparados de San José</b>	
<b>Número caso de uso: 004</b>	<b>Nombre del caso de uso:</b> Calcular Planilla
<b>Fecha elaboración:</b>	09/09/2024
<b>Descripción caso de uso:</b>	Describe el proceso mediante el cual se determina el monto de planilla tomando en cuenta varios factores como las cargas sociales, deducciones de ley, ausencias, horas extras, comisiones e incapacidades.
<b>Autor caso de uso:</b>	
<b>Actores relacionados:</b>	Administrador, Usuarios
<b>Precondiciones:</b>	El usuario administrador de nómina ha iniciado sesión en la aplicación. Los usuarios sobre los que se calculará el salario deberán estar activos. Se han registrado las ausencias, horas extras, comisiones e incapacidades del colaborador.
<b>Flujo básico del caso de uso</b>	
<ol style="list-style-type: none"> <li>1. El usuario administrador de nómina accede al módulo de nómina.</li> <li>2. El sistema muestra un formulario con año, mes y un <i>checkbox</i> si es primera o segunda quincena del mes.</li> </ol>	

<p>3. El usuario ingresa la información y presiona el botón de calcular planilla.</p> <p>4. El sistema recopila la siguiente información:</p> <ul style="list-style-type: none"> <li>-Salario base: monto acordado en el contrato,</li> <li>-Cargas sociales,</li> <li>-Deducciones de ley,</li> <li>-Horas extras,</li> <li>-Comisiones,</li> <li>-Incapacidades,</li> <li>-Ausencias justificadas e injustificadas.</li> </ul> <p>5. El sistema realiza los cálculos:</p> <ul style="list-style-type: none"> <li>-Salario bruto: este es la suma del salario base, horas extras, comisiones y viáticos.</li> <li>-Salario neto: esto es el salario bruto menos la cargas sociales y deducciones de ley.</li> </ul> <p>6. El usuario administrador de nómina verifica los resultados y confirma el salario calculado.</p> <p>7. El usuario guarda el cálculo de planilla en el sistema.</p>	
<b>Subflujos</b>	
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1</b>	Si el colaborador tiene ausencias injustificadas, se aplican descuentos adicionales al salario neto.

<b>Flujo Alternativo No. 2</b>	Si el usuario desea, puede no guardar aún el cálculo de planilla y solamente obtener la información para consulta.
<b>Requerimientos especiales</b>	
Se han registrado las ausencias, horas extras, comisiones e incapacidades del colaborador	
<b>Postcondiciones</b> Se ha calculado y confirmado el monto de planilla.	

Fuente: Elaboración propia (2024).

**Tabla 18.**

*Caso de uso. Solicitar vacaciones*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa Rastreadores Mundiales.</b>	
<b>Número caso de uso: 005</b>	<b>Nombre del caso de uso:</b> Solicitar vacaciones
<b>Fecha elaboración:</b>	14/09/2024
<b>Descripción caso de uso:</b>	Describe el proceso de solicitud y aprobación de vacaciones.
<b>Autor caso de uso:</b>	Randall Fernández Meza
<b>Actores relacionados:</b>	Administrador, Usuarios
<b>Precondiciones:</b>	El usuario ha iniciado sesión en la aplicación.

	El usuario que aprobará la solicitud debe tener rol de administrador.
<b>Flujo básico del caso de uso</b>	
<ol style="list-style-type: none"> <li>1. El usuario ingresa al menú de “Solicitar vacaciones”.</li> <li>2. El usuario ingresa la solicitud de vacaciones.</li> </ol> <p><b>Subflujo 01</b></p> <ol style="list-style-type: none"> <li>3. La solicitud de vacaciones es gestionada por el administrador. <b>Subflujo 02</b></li> </ol>	
<b>Subflujos</b>	
<b>Subflujo 01. Ingresar solicitud.</b>	El usuario ingresa la fecha de inicio y fecha final que solicita para vacaciones.
<b>Subflujo 02. Solicitud registrada.</b>	La solicitud se muestra en el menú de vacaciones del usuario administrador.
<b>Subflujo 03. Aprobación administrador</b>	El usuario administrador cambiar el estado de las vacaciones a aprobada Flujo Alternativo 02 o rechazada Flujo Alternativo 03
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1. Datos incorrectos</b>	Si el usuario ingresó fechas pasadas, si elige una fecha de fin de vacaciones más reciente que la fecha de inicio de vacaciones, entonces el sistema muestra un mensaje de error.

<b>Flujo Alternativo No. 2. Aprobación de solicitud</b>	El sistema registra el cambio. Se descuentan los días tomados del total de vacaciones. El sistema envía un mensaje de aprobación de solicitud de vacaciones al usuario solicitante.
<b>Flujo Alternativo No. 3. Rechazo de solicitud</b>	El sistema registra la solicitud como rechazada. El sistema envía un mensaje de solicitud rechazada al usuario solicitante.
<b>Requerimientos especiales</b>	
El usuario debe de tener vacaciones acumuladas por su periodo de trabajo.	
<b>Postcondiciones</b> Se ha aprobado o rechazado la solicitud de vacaciones.	

Fuente: Elaboración propia.

### Tabla 19.

*Caso de uso. Solicitar horas extra*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa Rastreadores Mundiales.</b>	
<b>Número caso de uso: 006</b>	<b>Nombre del caso de uso:</b> Solicitar horas extra
<b>Fecha elaboración:</b>	15/09/2024
<b>Descripción caso de uso:</b>	Describe el proceso de solicitud y aprobación de horas extra.

<b>Autor caso de uso:</b>	Randall Fernández Meza
<b>Actores relacionados:</b>	Administrador, Usuarios
<b>Precondiciones:</b>	El usuario ha iniciado sesión en la aplicación. El usuario que creara la solicitud debe tener rol de administrador.
<b>Flujo básico del caso de uso</b>	
<ol style="list-style-type: none"> <li>1. El usuario administrador ingresa al menú de solicitar horas extra.</li> <li>2. El usuario administrador ingresa la solicitud de horas extra. <b>Subflujo 01</b></li> <li>3. La solicitud de horas extra es notificada al usuario colaborador. <b>Subflujo 02</b></li> </ol>	
<b>Subflujos</b>	
<b>Subflujo 01. Ingresar solicitud.</b>	El usuario administrador ingresa las horas extra al sistema, por cantidad de horas, fecha y usuario que hará las horas extra.
<b>Subflujo 02. Notificación al usuario</b>	La solicitud se muestra como una alerta cuando el usuario colaborador haya iniciado sesión.
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1. Datos incorrectos</b>	Si el usuario ingresó un monto mayor a 24 horas, entonces el sistema muestra un mensaje de error.

<b>Flujo Alternativo No. 2. Aprobación de solicitud</b>	El sistema registra la solicitud de horas extra. Se utiliza un contador mensual de horas extras tomando en cuenta si es día feriado y día laboral. El sistema envía una notificación al usuario que hará las horas extra.
<b>Requerimientos especiales</b>	
<b>Postcondiciones</b> Se ha ingresado las horas extra.	

Fuente: Elaboración propia (2024)

### Tabla 20.

*Caso de uso. Registrar comisiones*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa Rastreadores Mundiales.</b>	
<b>Número caso de uso: 008</b>	<b>Nombre del caso de uso:</b> Registrar Comisiones
<b>Fecha elaboración:</b>	16/09/2024
<b>Descripción caso de uso:</b>	Describe el registro de comisiones.
<b>Autor caso de uso:</b>	Randall Fernández Meza
<b>Actores relacionados:</b>	Administrador, Usuarios
<b>Precondiciones:</b>	El usuario ha iniciado sesión en la aplicación. El usuario que creará el registro de comisión debe tener rol de administrador.
<b>Flujo básico del caso de uso</b>	
1. El usuario administrador ingresa	

<p>al menú de “Registrar comisiones”.</p> <p>2. El usuario administrador registra las comisiones.</p>	
<b>Subflujos</b>	
<b>Subflujo 01. Registro de comisión</b>	El usuario administrador selecciona el usuario al que se le registrará la comisión. Se debe especificar la fecha y un identificador de factura que está asociada a la comisión.
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1. Datos incorrectos</b>	<p>Si el usuario ingresó un monto negativo, debe mostrarse un mensaje de error.</p> <p>Si el usuario no ingresó todos los datos, debe mostrarse un mensaje de error.</p>
<b>Requerimientos especiales</b>	
<b>Postcondiciones</b>	
La comisión queda registrada en el sistema. Se utilizará esta información para el cálculo de salario.	

Fuente: Elaboración propia (2024).

### Tabla 21.

*Caso de uso. Registrar incapacidades*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano</b>	
--	--

<b>en la empresa Rastreadores Mundiales.</b>	
<b>Número caso de uso:</b> 012	<b>Nombre del caso de uso:</b> Registrar Incapacidades
<b>Fecha elaboración:</b>	16/09/2024
<b>Descripción caso de uso:</b>	Describe el registro de incapacidades
<b>Autor caso de uso:</b>	Randall Fernández Meza
<b>Actores relacionados:</b>	Administrador, Usuarios
<b>Precondiciones:</b>	El usuario ha iniciado sesión en la aplicación. El usuario que aprobará la solicitud debe tener rol de administrador.
<b>Flujo básico del caso de uso</b>	
<ol style="list-style-type: none"> <li>1. El usuario administrador ingresa al menú de registrar incapacidad.</li> <li>2. El usuario administrador registra las fechas de incapacidad.</li> </ol> <p><b>Subflujo 01</b></p> <ol style="list-style-type: none"> <li>3. La incapacidad se registra para el usuario.</li> </ol>	
<b>Subflujos</b>	

<b>Subflujo 01. Registro de incapacidad</b>	El usuario administrador selecciona el usuario al que se le registrará la incapacidad. Se debe especificar la fecha de inicio y fecha final de incapacidad. Ingresar el tipo de incapacidad. Ingresar el documento oficial de incapacidad. Procede a registrar la información.
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1. Datos incorrectos</b>	Si el usuario ingresó una fecha inicial más reciente que la fecha final de incapacidad, se debe mostrar un mensaje de error.  Si el usuario no ingresó todos los datos, debe mostrarse un mensaje de error.
<b>Requerimientos especiales</b>	
El tipo de incapacidad definirá el monto que debe pagarse al colaborador utilizando los siguientes rubros.  Enfermedad: los primeros 3 días cubre el 60%.  Licencia de maternidad: hasta 120 días al 50%.  Licencia de paternidad: 8 días a 50%.  Accidente laboral no se cubre.	
<b>Postcondiciones</b>  La incapacidad queda registrada en el sistema. Se utilizará esta información para el cálculo de salario	

Fuente: Elaboración propia (2024).

**Tabla 22.***Caso de uso Calcular Aguinaldo*

<b>Prototipo: Prototipo funcional para la gestión del recurso humano en la empresa Rastreadores Mundiales, ubicada en Desamparados de San José</b>	
<b>Número caso de uso: 004</b>	<b>Nombre del caso de uso:</b> Calcular Aguinaldo
<b>Fecha elaboración:</b>	09/19/2024
<b>Descripción caso de uso:</b>	Describe el proceso mediante el cual se calcula el aguinaldo
<b>Autor caso de uso:</b>	
<b>Actores relacionados:</b>	Administrador, Usuarios
<b>Precondiciones:</b>	El usuario administrador de nómina ha iniciado sesión en la aplicación. Se han registrado los salarios de las quincenas previas.
<b>Flujo básico del caso de uso</b>	
<ol style="list-style-type: none"> <li>1. El usuario administrador de nómina accede al módulo de salarios.</li> <li>2. El usuario presiona el botón calcular aguinaldo.</li> <li>3. El sistema recopila el monto de los salarios de las quincenas registradas para cada colaborador.</li> <li>4. El sistema realiza los cálculos.</li> <li>5. El usuario administrador de nómina verifica los resultados y confirma el aguinaldo calculado.</li> </ol>	

6. El usuario guarda el cálculo de aguinaldo en el sistema.	
<b>Subflujos</b>	
<b>Flujos alternos</b>	
<b>Flujo Alternativo No. 1</b>	Si el usuario desea, puede no guardar aún el cálculo de planilla y solamente obtener la información para consulta.
<b>Requerimientos especiales</b>	
Se han registrado las quincenas de los colaboradores.	
<b>Postcondiciones</b> Se ha calculado y confirmado el monto de aguinaldo.	

Fuente: Elaboración propia (2024)

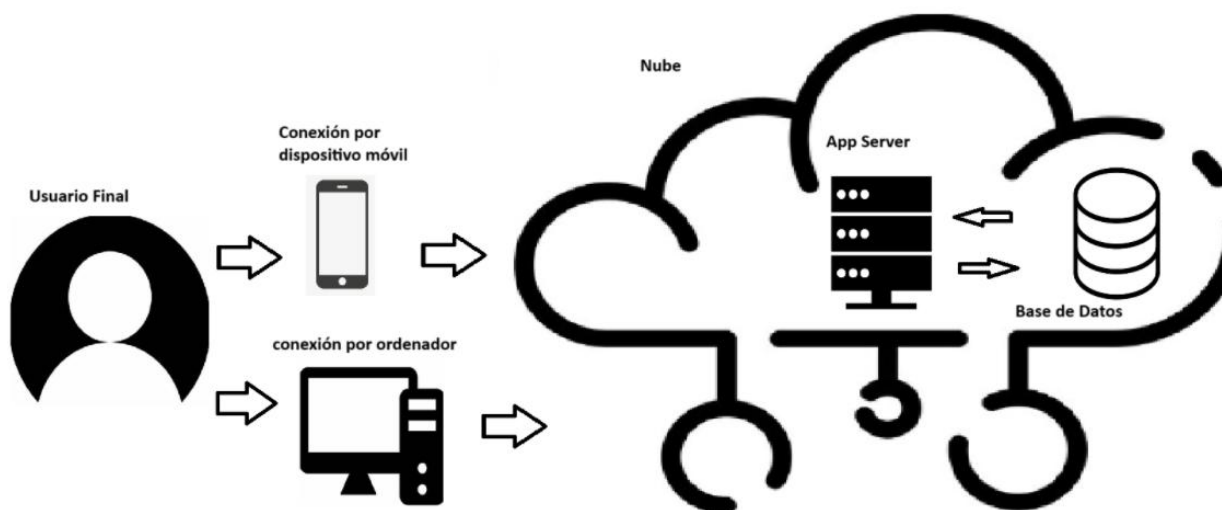
## Diseño

### Arquitectura del Sistema

Para acceder al sistema el usuario utilizaría su dispositivo móvil, o si lo prefiere alguno de los computadores disponibles en el local, estas computadoras se utilizan con mayor frecuencia por el Departamento de Recursos Humanos. Con cualquiera de los dispositivos que se utilice se redirigirá la petición de inicio de sesión al servidor de aplicaciones (*App Server*) y este se conectará a la base de datos. Ambos, el servidor de aplicación y la base de datos se encuentran en la nube. En la figura 6 se explica la arquitectura de sistema.

#### Figura 6.

##### *Arquitectura de sistema*



*Fuente: Elaboración propia, 2024*

### Arquitectura de Software

En este proyecto se está utilizando ASP Core con el modelo vista-controlador. En este modelo cuando se conecta a la aplicación, se accede a las vistas, en las vistas está el código css y html que mostrará al usuario los formularios con los que podrá interactuar. Además, se utiliza

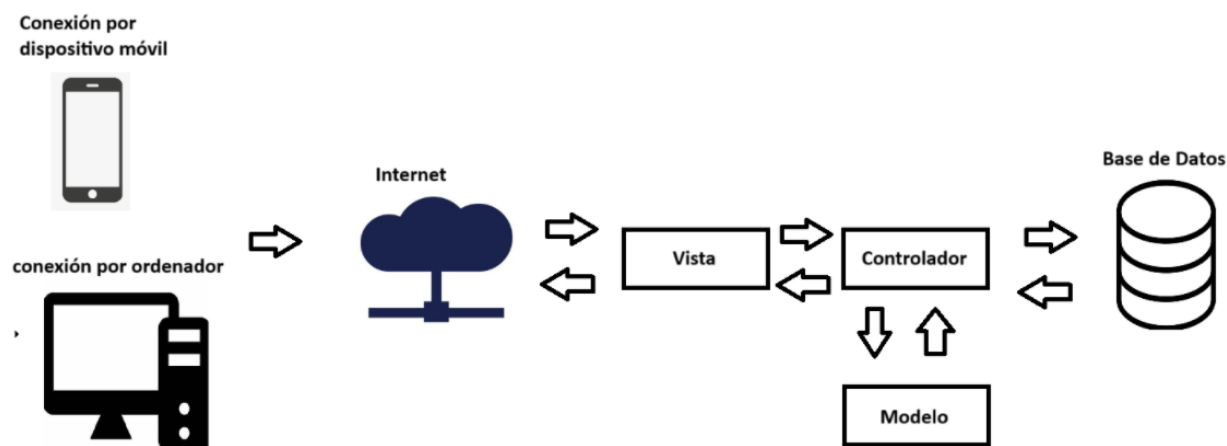
Razor, que es C# implementado directamente en la vista; adicionalmente, para algunas funcionalidades se utiliza JavaScript para poder interactuar con algunos campos, además de para poder ejecutar algunas funciones.

Las vistas se conectan a los controladores, en ellos está la lógica de negocio para ejecutar los diferentes métodos que se accionan desde la vista. Estos controladores dependen de un modelo, que son clases aparte que contienen los atributos del modelo, así como sus constructores. Por último, estos controladores por medio de una cadena de caracteres se conectan a la base de datos para sus métodos Get y Post, de obtención y envío de datos respectivamente. Cuando se hace un Get, y se obtiene información de la base de datos, mucha de esta información se puede enviar de vuelta a la vista y con esto hacerla visible de nuevo al usuario.

La elección del modelo vista-controlador es debido a que permite mantener el código ordenado y facilita la interacción de las diferentes clases (incluyendo los controladores) entre sí. Para el prototipo se eligió ADO, que es la utilización de consultas directas a la base de datos, en contraposición al *EntityFramework* que es el otro modelo de conexión utilizado. Esta decisión de utilizar ADO se toma con base en la experiencia del desarrollador del prototipo, pues para efectos prácticos se pueden obtener resultados similares. En la figura 7 se explica el modelo elegido.

**Figura 7.**

*Arquitectura de Software*



*Fuente: Elaboración Propia, 2024*

## Diccionario de Datos

Se especifica en detalle las tablas de la base de datos y sus columnas para aclarar la relación entre tablas. Se muestra una tabla para cada entidad

**Tabla 23**

Entidad Usuario

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idUsuario	int	sí			cédula
nombre	varchar(30)				nombre
apellido1	varchar(30)				apellidos
apellido2	varchar(30)				apellidos
telefono	varchar(15)				teléfono
correo	varchar(30)				email del usuario
contraseña	varchar(15)				contraseña del usuario
salarioBase	float				
fechaDeEntrada	date				
fechaDeSalida	date			sí	nulo si el usuario está activo
diasVacaciones	int				
rol	varchar(15)				define rol: colaborador, supervisor, administrador, rrhh, superadmin
estadoActivo	bit				indica si el usuario actualmente labora

*Fuente:* Elaboración Propia (2024)

**Tabla 24**

Entidad Vacación

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idVacaciones	Int	sí			id único
idUsuario	Int		sí		id del colaborador
fechaInicial	Date				inicio rango de fechas solicitadas
fechaFinal	Date				final rango de fechas solicitadas

Estado	varchar(15)				si ha sido aprobada o rechazado o pendiente
Días	Int				cantidad de días luego de calcular feriados y días que no se trabaja

*Fuente:* Elaboración Propia (2024)

**Tabla 25**

Entidad Permiso

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idPermiso	int	sí			id único
idUsuario	int		sí		id del colaborador
tipoPermiso	varchar(30)				informa el tipo de permiso
fechaDeSolicitud	date				
fechaInicial	date				inicio rango de fechas solicitadas
fechaFinal	date				final rango de fechas solicitadas
cantidadDias	int				cantidad de días luego de calcular feriados y días que no se trabaja
Estado	varchar(15)				si ha sido aprobada o rechazado o pendiente

*Fuente:* Elaboración Propia (2024)

**Tabla 26**

Entidad CalculoSalario

<b>Nombre de atributo</b>	<b>Tipo de dato</b>	<b>PK</b>	<b>FK</b>	<b>Null</b>	<b>Descripción</b>
idSalario	int	sí			id único
idUsuario	int		sí		id del colaborador
Anho	int				año para el que se calcula el salario
Mes	varchar(12)				mes para el que se calcula el salario
Quincena	bit				si es primera (0) o segunda quincena (1)
Monto	float				
ReducciónPermisosSinGoce	float				subtotal por ausencias sin goce salarial
AumentoPorComisiones	float				subtotal por comisiones
AumentoPorHorasExtras	float				subtotal por horas Extras
ReducciónIncapacides	float				subtotal por reducción Incapacidades
Fecha	float				Fecha que se paga la quincena

*Fuente:* Elaboración Propia (2024)

**Tabla 27**

Entidad PedidoHorasExtra

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
IdPedidoHorasExtra	Int	sí			id único
idUsuario	Int		sí		id del colaborador
fechaSolicitada	date				fecha que se laboró horas extras
cantidadHoras	float				cantidad de horas trabajadas luego de realizar el cálculo de la jornada laboral
estado	varchar(15)				si ha sido aprobada o rechazado o pendiente

*Fuente:* Elaboración Propia (2024)**Tabla 28**

Entidad Liquidación

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idLiquidacion	int	sí			id único
idUsuario	int		sí		id del colaborador
TipoLiquidacion	varchar(30)				Define si es renuncia o despido con o sin responsabilidad patronal
fechaLiquidacion	float				última fecha de laborar
montoLiquidacion	float				total por pagar

*Fuente:* Elaboración Propia (2024)**Tabla 29**

Entidad Comisión

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idComision	int	sí			id único
idUsuario	int		sí		id del colaborador
fecha	date				fecha que se aprobó la comisión
montoComision	float				monto total de comisión
idFactura	varchar(15)				Factura a la que hace referencia la comisión

*Fuente:* Elaboración Propia (2024)

**Tabla 30**

Entidad Incapacidad

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idIncapacidad	int	sí			id único
idUsuario	int		sí		id del colaborador
fechaInicial	Date				fecha que inicia incapacidad
fechaFinal	Date				fecha que termina la incapacidad
tipoIncapacidad	varchar(30)				tipo de incapacidad
Estado	varchar(15)				
cantidadDias	int				cantidad de días luego de calcular feriados y días que no se trabaja

*Fuente:* Elaboración Propia (2024)**Tabla 31**

Entidad Aguinaldo

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idAguinaldo	int	sí			
idUsuario	int		sí		
Anho	int				Indica el año para el que se calcula el aguinaldo
montoAguinaldo	float				

*Fuente:* Elaboración Propia (2024)**Tabla 32**

Entidad HorarioTrabajo

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idHorarioTrabajo	int	sí			
idUsuario	int		sí		
diaDeLaSemana	varchar(15)				Existe uno para cada día de la semana
esDíaDeTrabajo	bit				define sí es día de trabajo o día libre

*Fuente:* Elaboración Propia (2024)

**Tabla 33**

Entidad Bitacora

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idBitacora	int	sí			
idUsuario	int		sí		
FechaDeEntrada	date				fecha de ingreso de usuario a la compañía
FechaDeSalida	date			sí	fecha de salida del usuario, es nula mientras el usuario esté activo
EstadoActivo	bit				es false hasta que el usuario termine relación con la empresa

*Fuente:* Elaboración Propia (2024)**Tabla 34**

Entidad MarcaEntradaSalida

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idMarca	int	sí			
idUsuario	int		sí		
HoraEntrada	datetime				marca el inicio de jornada laboral
HoraSalida	datetime			sí	marca el final de la jornada, es nulo mientras la jornada no acabe
HorasTrabajo	float			sí	cálculo de horaSalida menos horaEntrada

*Fuente:* Elaboración Propia (2024)

**Tabla 35**

Entidad Notificación

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idMarca	int	sí			
idUsuario	int		sí		
TipoNotificacion	varchar(30)				define si es para vacacion, permiso, etc
fechaCreacion	date				fecha que se cambió el estado a la vacación, permiso, etc
haSidoVista	bit				si la notificación aparece una vez en pantalla debe pasar a true

*Fuente:* Elaboración Propia (2024)**Tabla 36**

Entidad Correo

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idCorreo	int	Sí			
idUsuario	int		sí		
Correo	varchar(30)				

*Fuente:* Elaboración Propia (2024)**Tabla 37**

Entidad Teléfono

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idTelefono	int	sí			
idUsuario	int		sí		

*Fuente:* Elaboración Propia (2024)

Tabla 38

## Entidad Rol

Nombre de atributo	Tipo de dato	PK	FK	Null	Descripción
idRol	int	Sí			
NombreRol	varchar(30)				

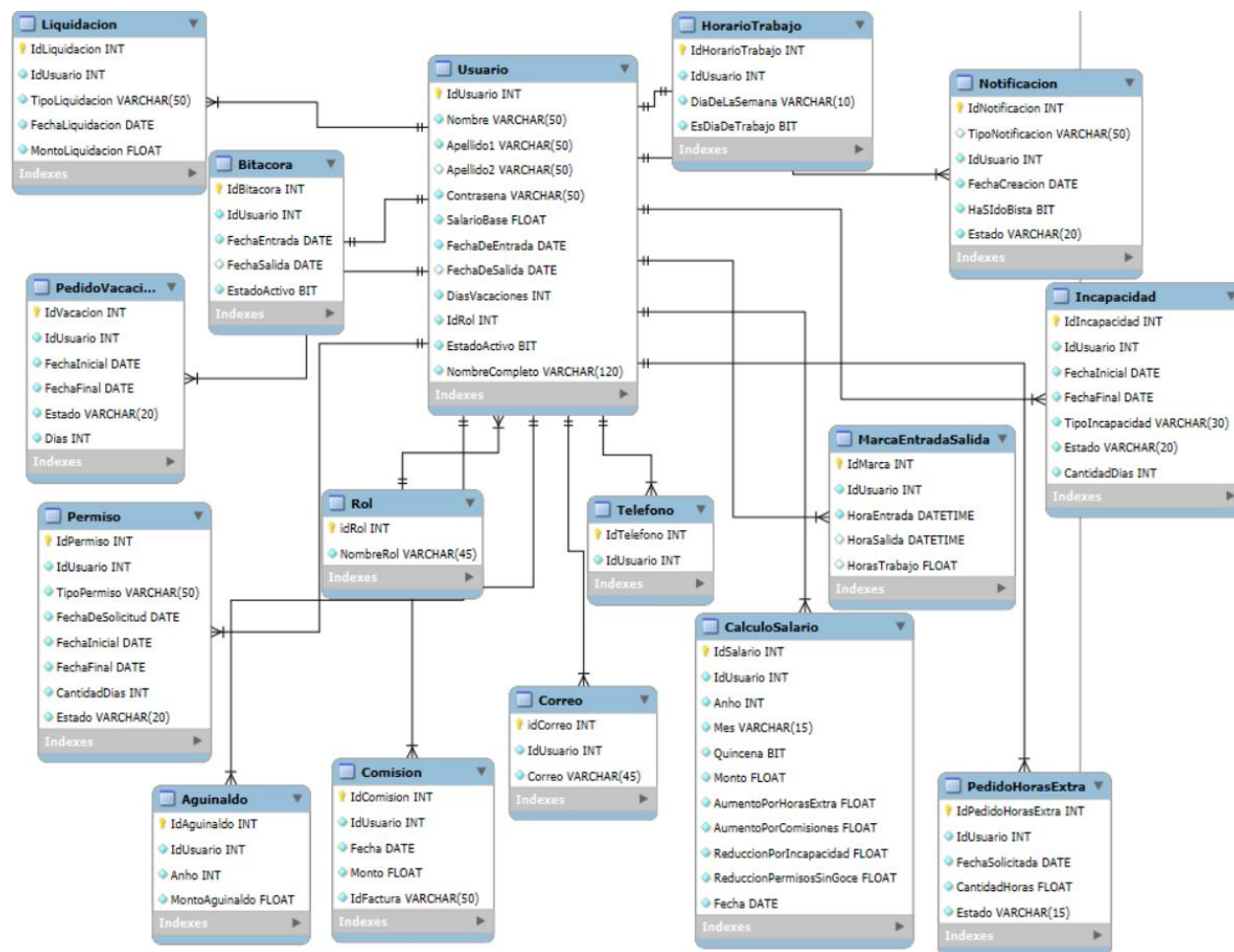
Fuente: Elaboración Propia (2024)

## Diseño de Bases de Datos

Se muestra el diagrama entidad relación para el prototipo.

Figura 8.

## Diagrama Entidad Relación



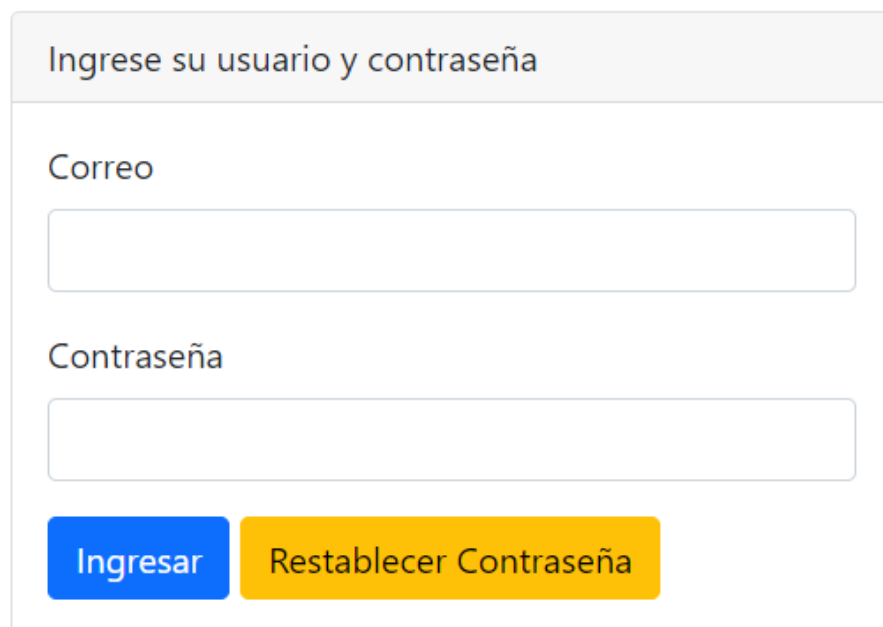
Fuente: Elaboración Propia (2024)

## Diseño de Entradas

Se muestran las imágenes de algunas pantallas de entrada que posee la aplicación actualmente.

### Figura 9.

*Pantalla de inicio de sesión.*



The image shows a login form with a light gray header containing the text "Ingrese su usuario y contraseña". Below the header are two input fields: "Correo" and "Contraseña". At the bottom of the form are two buttons: a blue button labeled "Ingresar" and a yellow button labeled "Restablecer Contraseña".

*Fuente:* Elaboración Propia (2024)

**Figura 10.***Crear nuevo usuario*

Crear Contacto

Cédula

Nombre


Apellido 1

Apellido 2

Teléfono

Correo

Salario Base

Fecha  

Rol

*Fuente: Elaboración Propia (2024)*

**Figura 11.***Cálculo de planilla*

Calcular Salario

Año

Mes

Es Primera Quincena

Salario Neto

Calcular Planilla

*Fuente:* Elaboración Propia (2024)

**Figura 12.***Cálculo de aguinaldo*

Calcular Aguinaldos

Año

Total Aguinaldos


Calcular Aguinaldo


*Fuente:* Elaboración Propia (2024)

**Figura 13.***Crear pedido de vacaciones*

Crear Pedido Vacaciones

Usted tiene 0 días de vacaciones

Fecha Inicial  


Fecha Final  


*Fuente:* Elaboración Propia (2024)

**Figura 14.***Crear pedido de permiso de ausencia*

Crear Permiso

Tipo Permiso

Fecha Inicial  


Fecha Final  

*Fuente:* Elaboración Propia (2024)

**Figura 15.***Registrar comisión*

### Registrar Comisión

Nombre Usuario

Fecha  

Monto


Id Factura


*Fuente:* Elaboración Propia (2024)

**Figura 16.***Registrar incapacidad*

### Crear Incapacidad

Nombre Usuario

Fecha Inicial  

Fecha Final  

Tipo Incapacidad


*Fuente: Elaboración Propia (2024)*

**Figura 17.***Cálculo de liquidación*

Cálculo de Liquidación

Nombre

Tipo Liquidación

Fecha Liquidación  

Monto Liquidación

[Calcular Liquidacion](#) [Volver a la lista](#)

*Fuente:* Elaboración Propia (2024)

**Figura 18.***Marcar entrada*

Marcar Entrada/Salida

**11:54:10**

Marcar Entrada

Marcar Entrada para otro Usuario

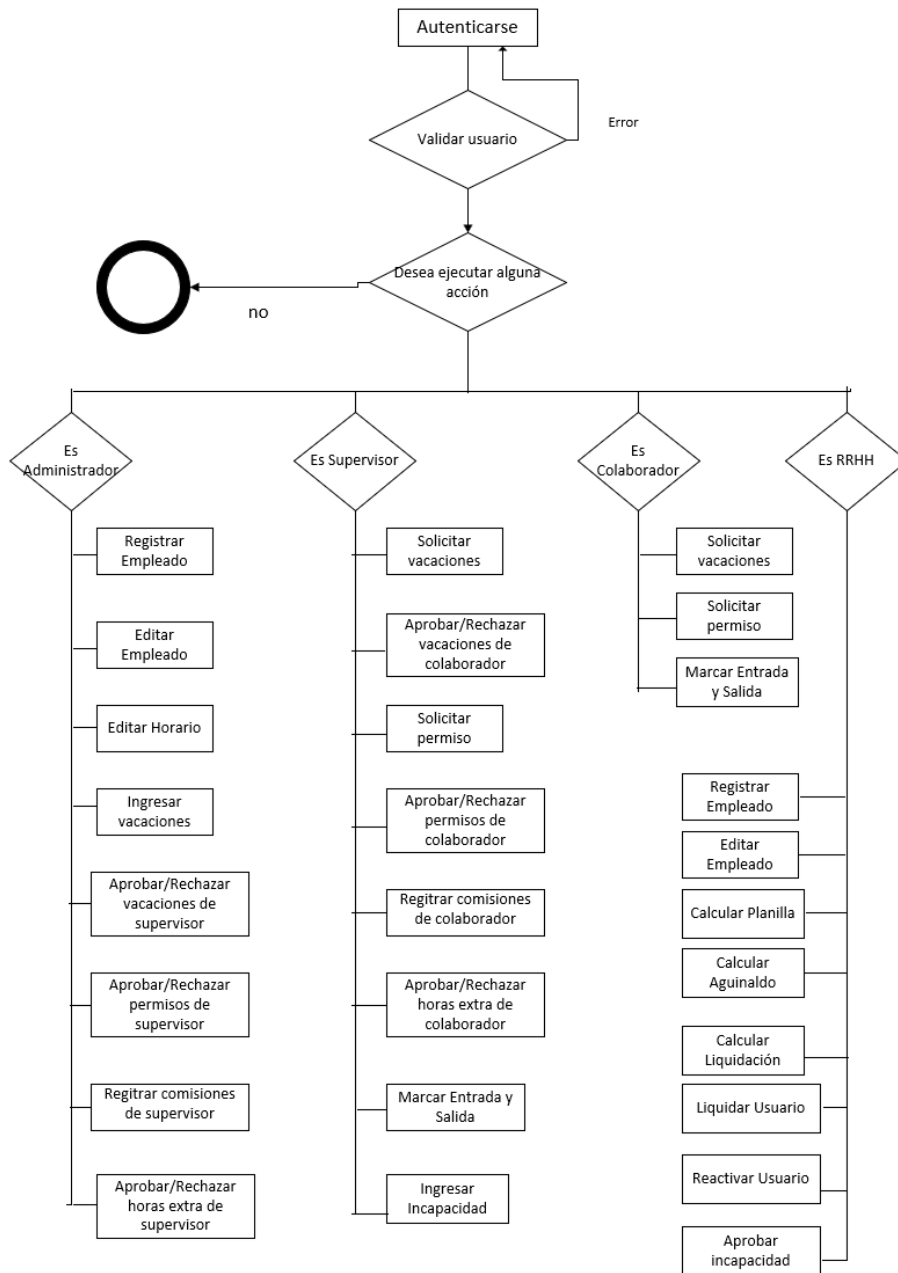
*Fuente:* Elaboración Propia (2024)

**Diseño de Procesos**

En la figura 9 se muestra el diagrama de flujo para el prototipo de gestión de nómina

**Figura 19.**

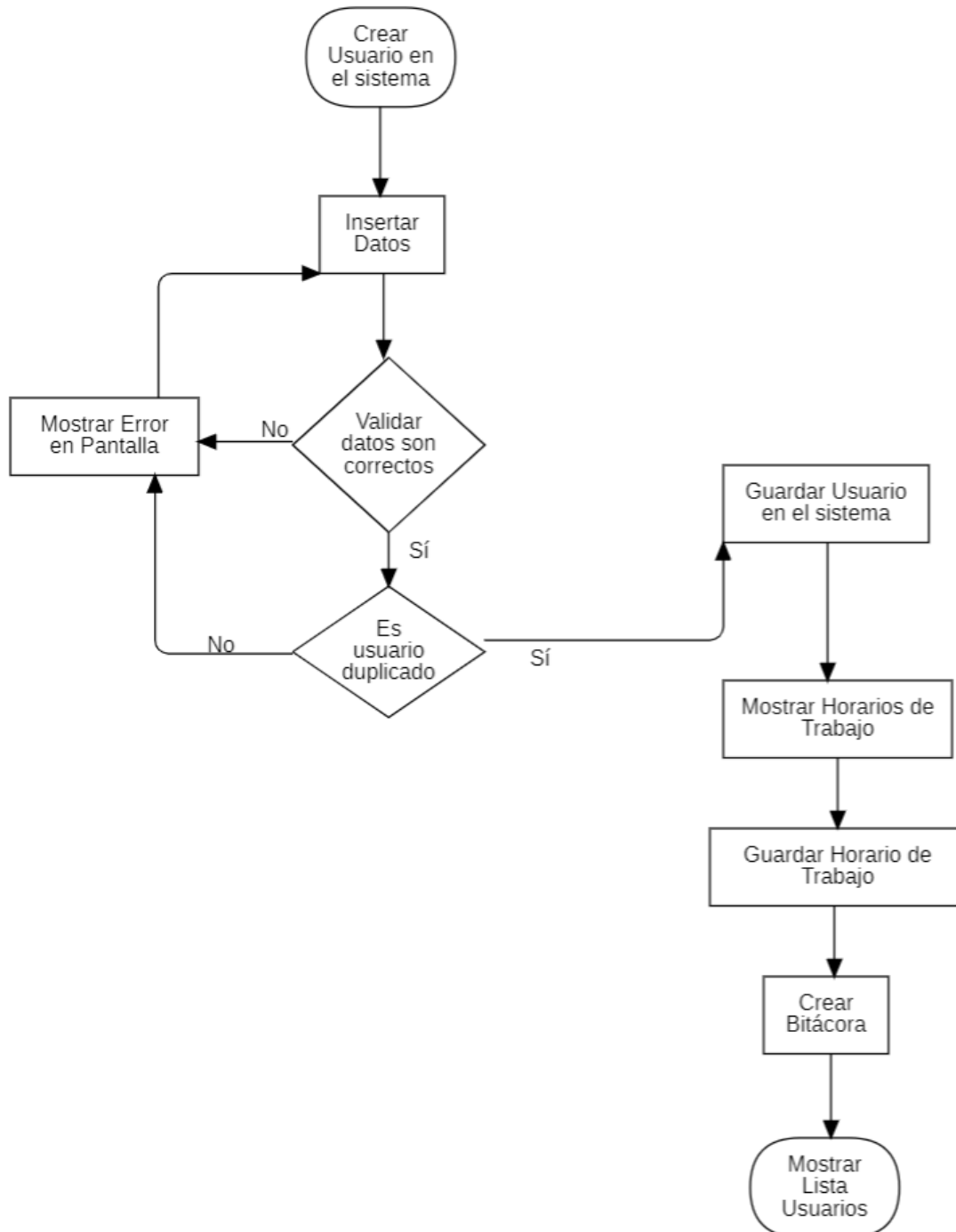
*Diagrama de flujo*



*Fuente: Elaboración Propia (2024)*

**Figura 20.**

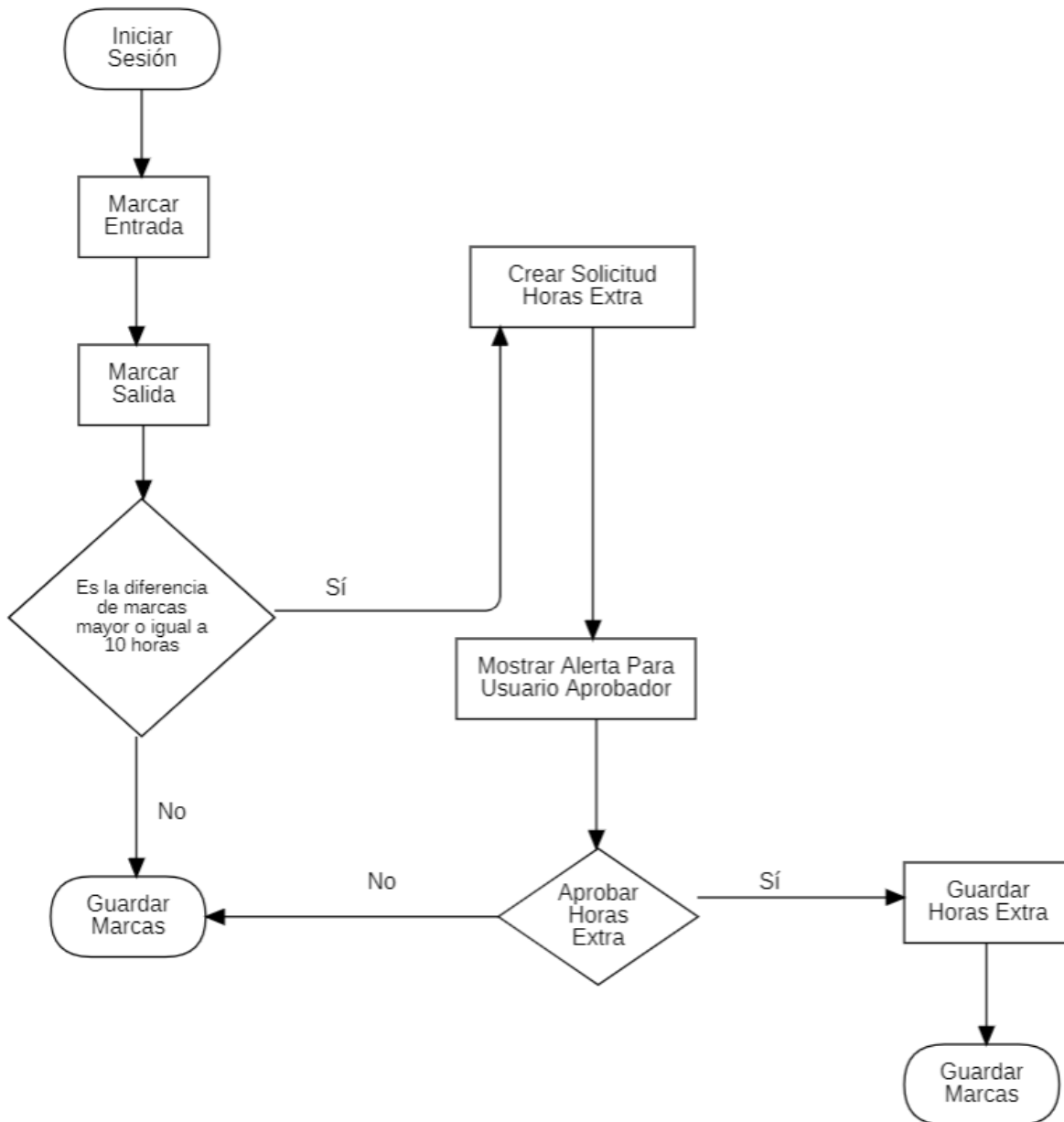
Diagrama de flujo para creación de usuario en sistema, edición de horario de trabajo y registro de bitácora



Fuente: Elaboración Propia (2024)

**Figura 21.**

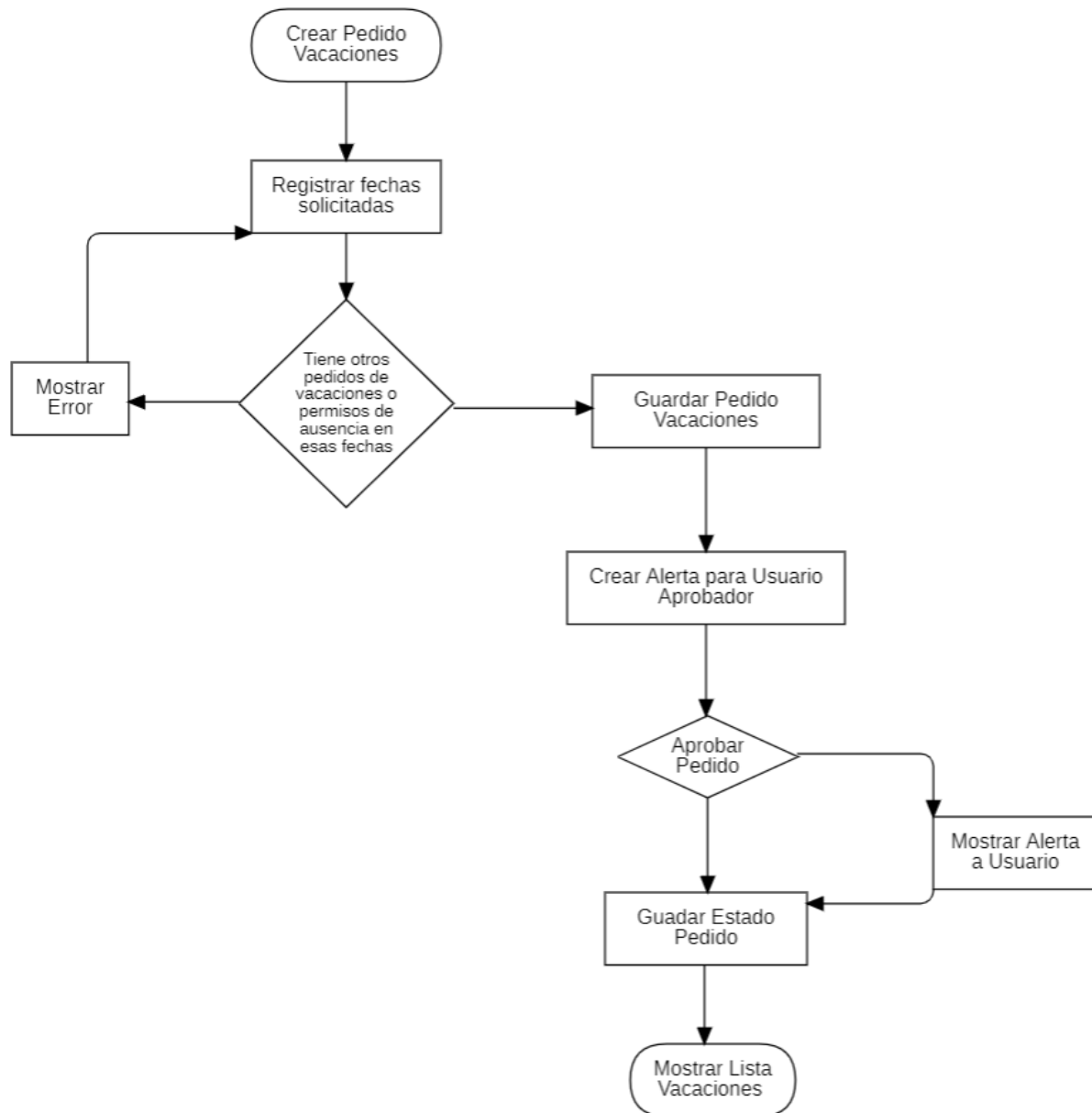
Diagrama de flujo para marcas de entrada y salida, y creación de solicitud de horas extra



Fuente: Elaboración Propia (2024)

**Figura 22.**

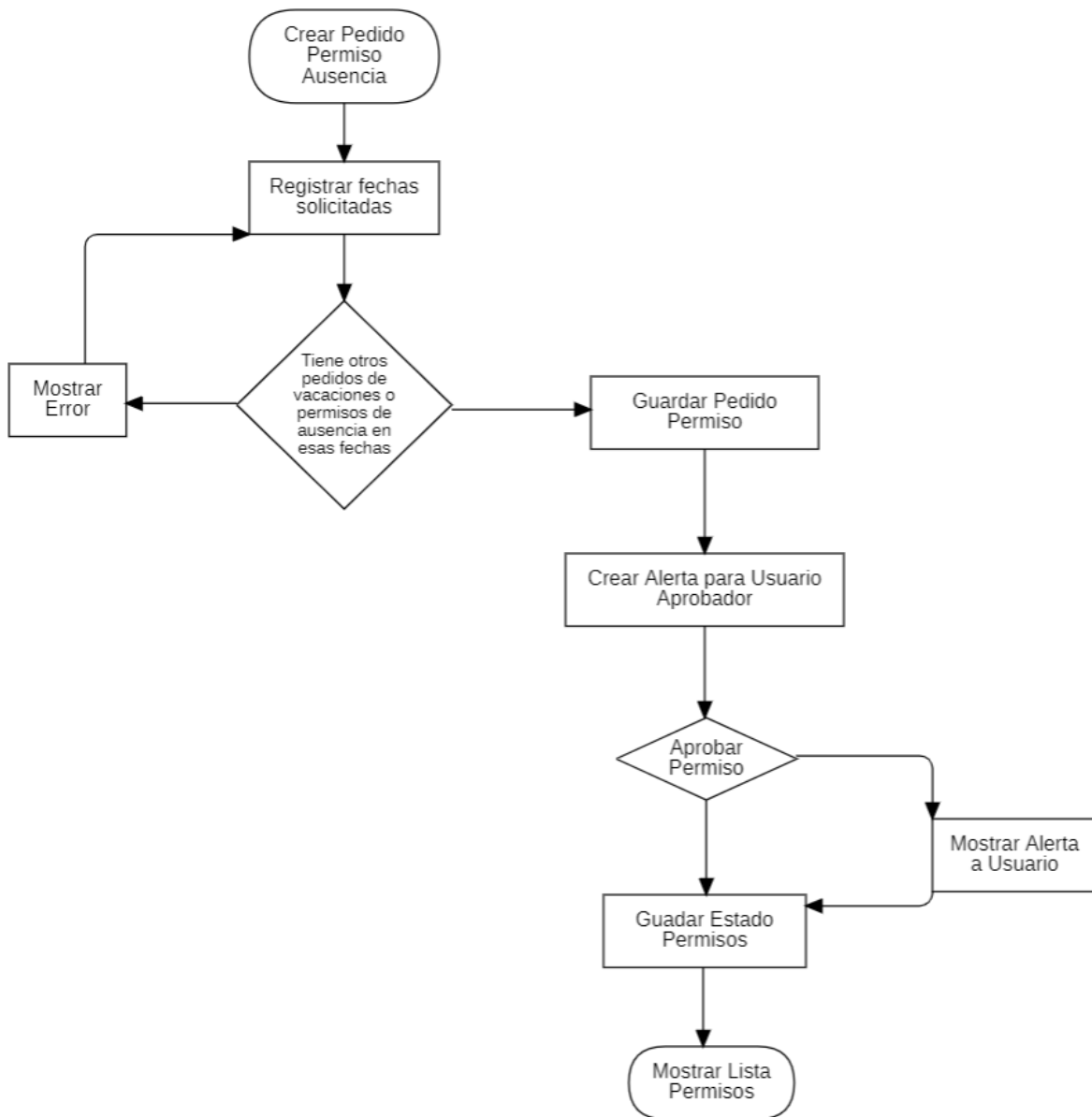
*Diagrama de flujo de pedidos de vacaciones*



*Fuente:* Elaboración Propia (2024)

**Figura 23.**

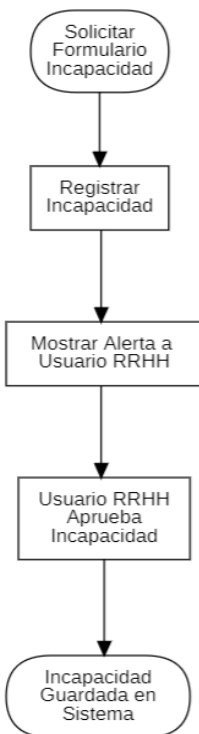
*Diagrama de flujo de permisos de ausencias*



*Fuente: Elaboración Propia (2024)*

**Figura 24.**

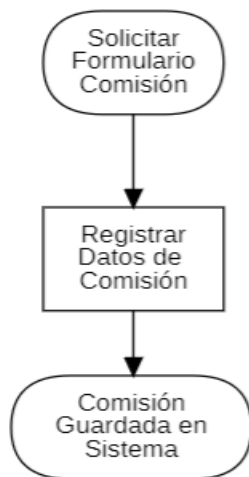
*Diagrama de flujo registrar incapacidades*



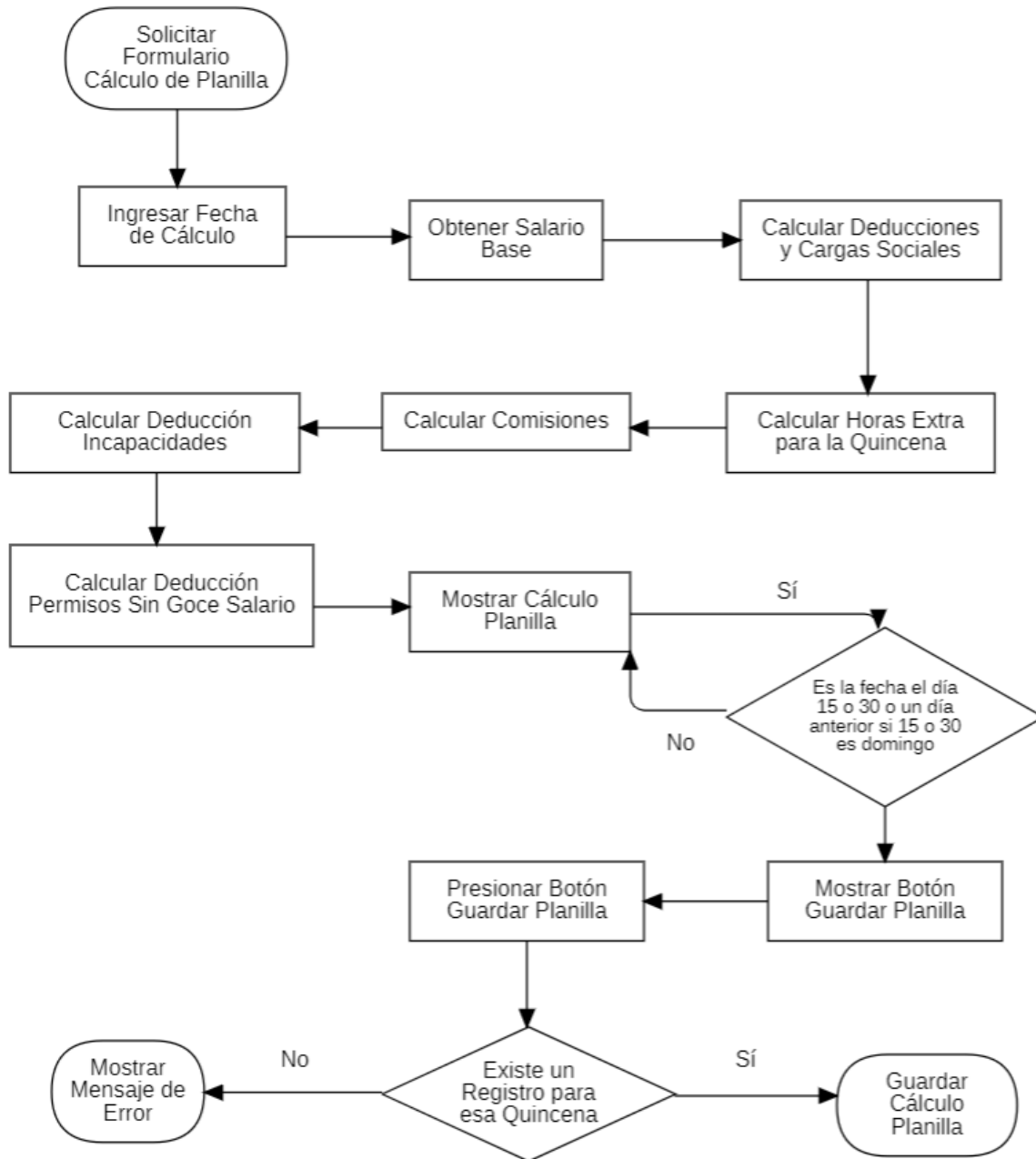
*Fuente: Elaboración Propia (2024)*

**Figura 25.**

*Diagrama de flujo de comisiones*

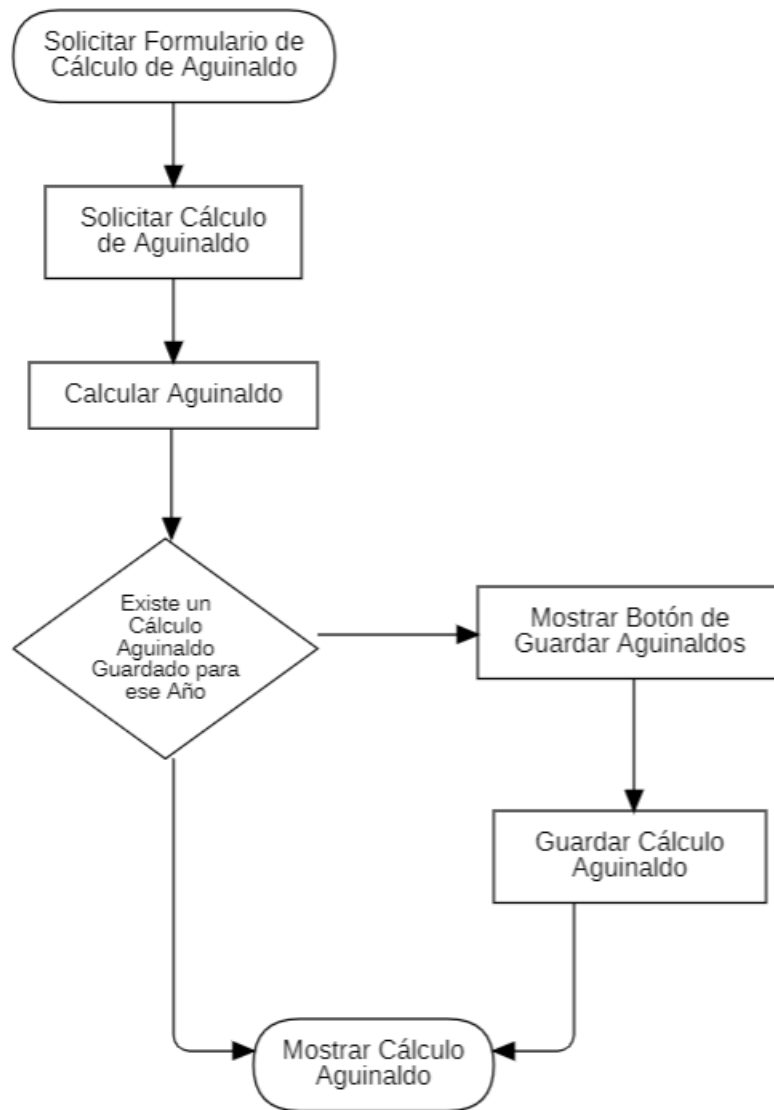


*Fuente: Elaboración Propia (2024)*

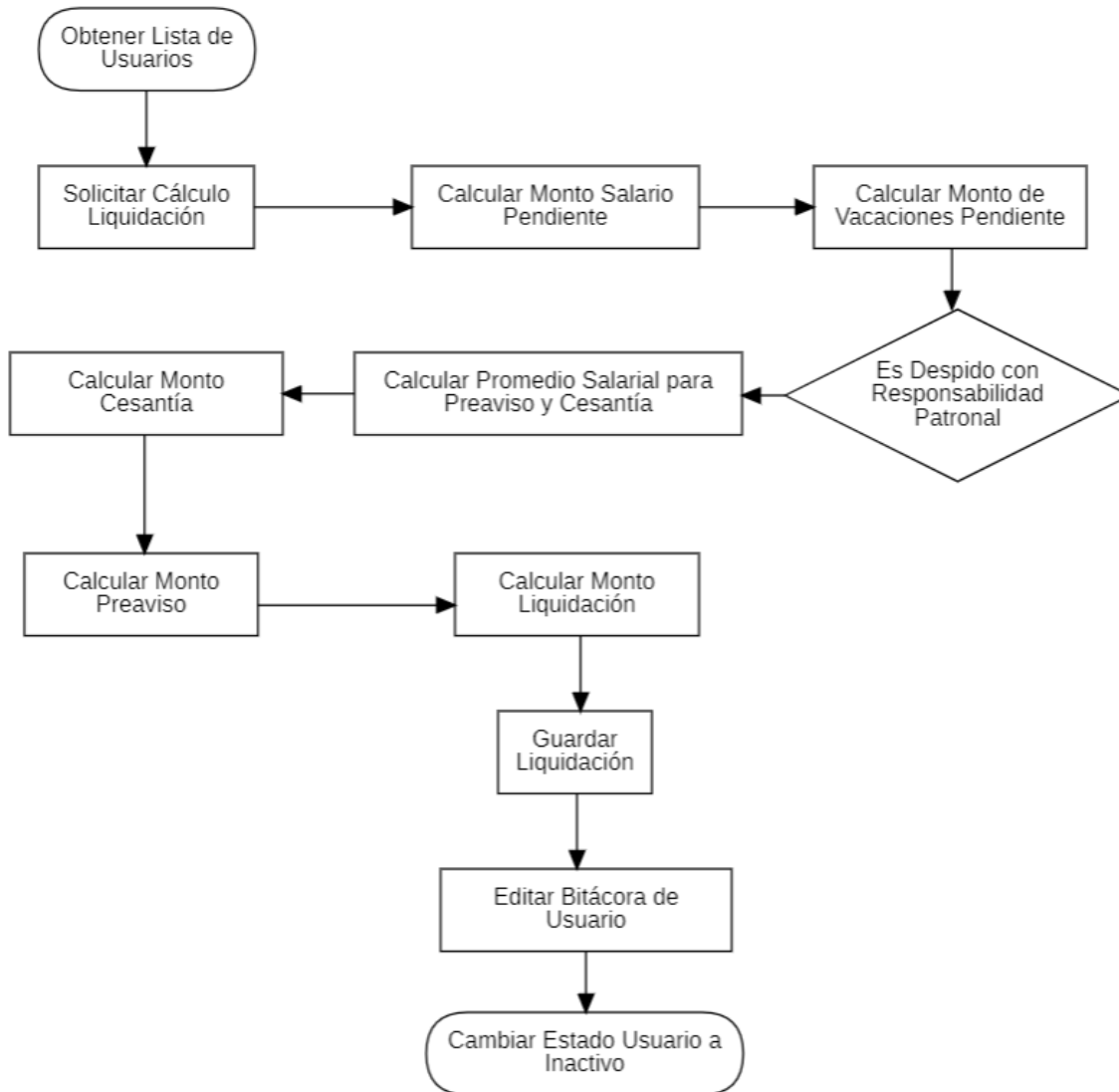
**Figura 26.***Diagrama de flujo de cálculo de planilla**Fuente: Elaboración Propia (2024)*

**Figura 27.**

*Diagrama de flujo de cálculo de aguinaldo*



*Fuente:* Elaboración Propia (2024)

**Figura 28.***Diagrama de flujo de liquidación*

*Fuente:* Elaboración Propia (2024)

## Diseño de salidas

### Figura 29.

*Obtener lista de usuarios*

Lista Usuarios

[Crear Nuevo](#)

Búsqueda por nombre

Nombre	Apellido 1	Apellido 2	Telefono	Correo	Fecha de Entrada	
Romy	Ortiz	Fernández	10101010	romy@gmail.com	04/11/2024	<a href="#">Editar</a> <a href="#">Obtener Horario</a> <a href="#">Desactivar</a>
Randall	Fernández	Meza	60638426	randall@gmail.com	07/06/2024	<a href="#">Editar</a> <a href="#">Obtener Horario</a> <a href="#">Desactivar</a>
Jenni	Hernandez	Poveda	7070707	jenni@gmail.com	05/10/2024	<a href="#">Editar</a> <a href="#">Obtener Horario</a> <a href="#">Desactivar</a>
Sonia	Meza	Madrigal	10101010	sonia@gmail.com	22/10/2024	<a href="#">Editar</a> <a href="#">Obtener Horario</a> <a href="#">Desactivar</a>
Jose	Fernandez	Bustamante	20202020	jose@gmail.com	05/11/2024	<a href="#">Reactivar</a>

*Fuente: Elaboración Propia (2024)*

### Figura 30.

*Obtener Lista de vacaciones*

Obtener Lista Vacaciones de todos los usuarios

[Ir a mis pedidos de vacaciones](#)

Nombre Usuario	Fecha Inicial	Fecha Final	Días	Estado
Sonia Meza Madrigal	11/11/2024	11/11/2024	1	Aprobado
Jose Fernandez Bustamante	18/11/2024	18/11/2024	1	Aprobado
Jose Fernandez Bustamante	21/11/2024	21/11/2024	1	Aprobado
Sonia Meza Madrigal	24/12/2024	26/12/2024	3	Aprobado
Jenni Hernandez Poveda	28/10/2024	28/10/2024	1	Rechazado
Jenni Hernandez Poveda	30/10/2024	31/10/2024	2	Rechazado
Jenni Hernandez Poveda	02/11/2024	03/11/2024	2	Rechazado
Sonia Meza Madrigal	03/11/2024	03/11/2024	1	Rechazado
Jose Fernandez Bustamante	19/11/2024	19/11/2024	1	Rechazado

*Fuente: Elaboración Propia (2024)*

**Figura 31.***Obtener lista de permisos*

Lista Permisos

[Ir a mis pedidos de permisos](#)

Búsqueda por nombre

Nombre Usuario	Tipo Permiso	Fecha Inicial	Fecha Final	Estado	
Jenni Hernandez Poveda	Con goce salarial	28/12/2024	02/01/2025	Pendiente	<a href="#">Aprobar</a> <a href="#">Rechazar</a>
Jose Fernandez Bustamante	Con goce salarial	05/11/2024	05/11/2024	Rechazado	
Jose Fernandez Bustamante	Sin goce salarial	08/11/2024	08/11/2024	Rechazado	
Jenni Hernandez Poveda	Con goce salarial	01/11/2024	01/11/2024	Aprobado	
Sonia Meza Madrigal	Con goce salarial	03/11/2024	03/11/2024	Aprobado	
Jose Fernandez Bustamante	Con goce salarial	04/11/2024	04/11/2024	Aprobado	
Sonia Meza Madrigal	Con goce salarial	18/11/2024	18/11/2024	Aprobado	

*Fuente: Elaboración Propia (2024)***Figura 32.***Obtener lista de comisiones*

Obtener Lista de Comisiones de todos los usuarios

[Ir a mis comisiones](#)

Nombre Usuario	ID Factura	Fecha	Monto	
Jenni Hernandez Poveda	A12121212	06/11/2024	500	<a href="#">Editar</a> <a href="#">Eliminar</a>
Jenni Hernandez Poveda	RM003	05/11/2024	9000	<a href="#">Editar</a> <a href="#">Eliminar</a>
Sonia Meza Madrigal	A000003	03/11/2024	5500	<a href="#">Editar</a> <a href="#">Eliminar</a>
Jose Fernandez Bustamante	A000004	03/11/2024	1000	<a href="#">Editar</a> <a href="#">Eliminar</a>
Jenni Hernandez Poveda	A00001	02/11/2024	9000	<a href="#">Editar</a> <a href="#">Eliminar</a>

[Guardar Comision](#)

*Fuente: Elaboración Propia (2024)*

**Figura 33.***Obtener lista horas extra*

Lista Solicitudes Horas Extra			
<a href="#">Ir a mis horas extra</a>			
Nombre Usuario	Fecha de Solicitud	Cantidad de Horas Extra	Estado
Jose Fernandez Bustamante	04/11/2024	6	Aprobado
Jenni Hernandez Poveda	03/11/2024	2	Aprobada
Jenni Hernandez Poveda	02/11/2024	2	Aprobada

*Fuente: Elaboración Propia (2024)***Figura 34.***Obtener lista de incapacidades*

Lista Incapacidades					
<a href="#">Crear Nuevo</a>					
Nombre Usuario	Fecha Inicial	Fecha Final	Dias	Tipo Incapacidad	
Jenni Hernandez Poveda	31/10/2024	31/10/2024	1	Enfermedad	<a href="#">Editar</a> <a href="#">Eliminar</a>
Jenni Hernandez Poveda	05/11/2024	05/11/2024	1	Enfermedad	<a href="#">Editar</a> <a href="#">Eliminar</a>
Sonia Meza Madrigal	03/11/2024	03/11/2024	0	Enfermedad	<a href="#">Editar</a> <a href="#">Eliminar</a>

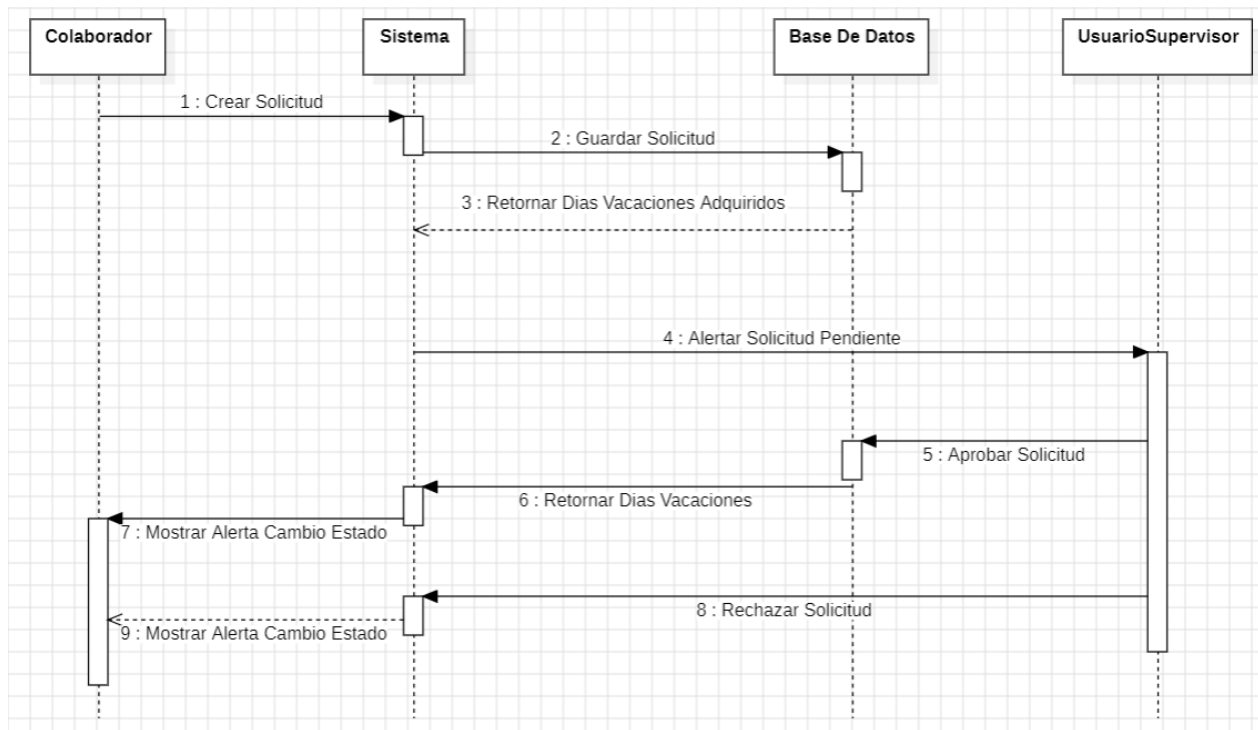
*Fuente: Elaboración Propia (2024)*

## Diagramas de Secuencia

Se muestran los diagramas de secuencia.

**Figura 35.**

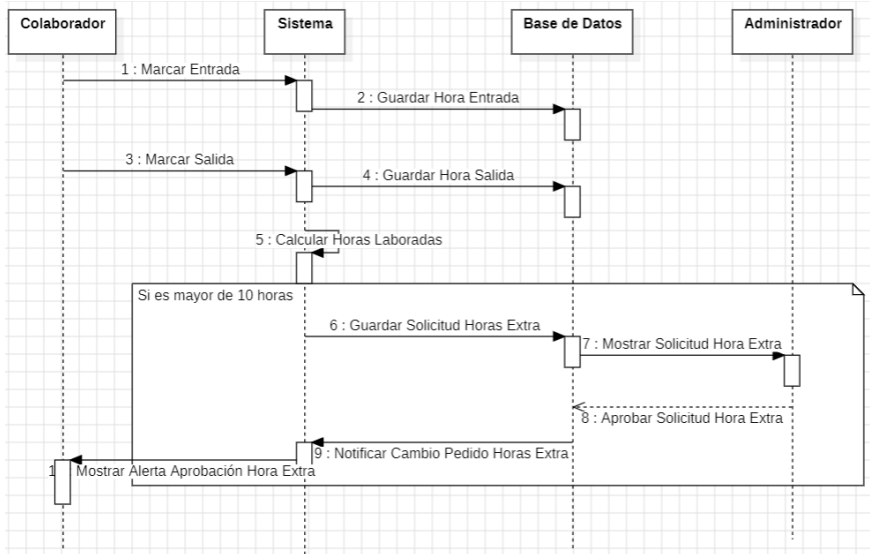
*Diagrama de Secuencia. Solicitud de vacaciones*



*Fuente:* Elaboración Propia (2024)

**Figura 36.**

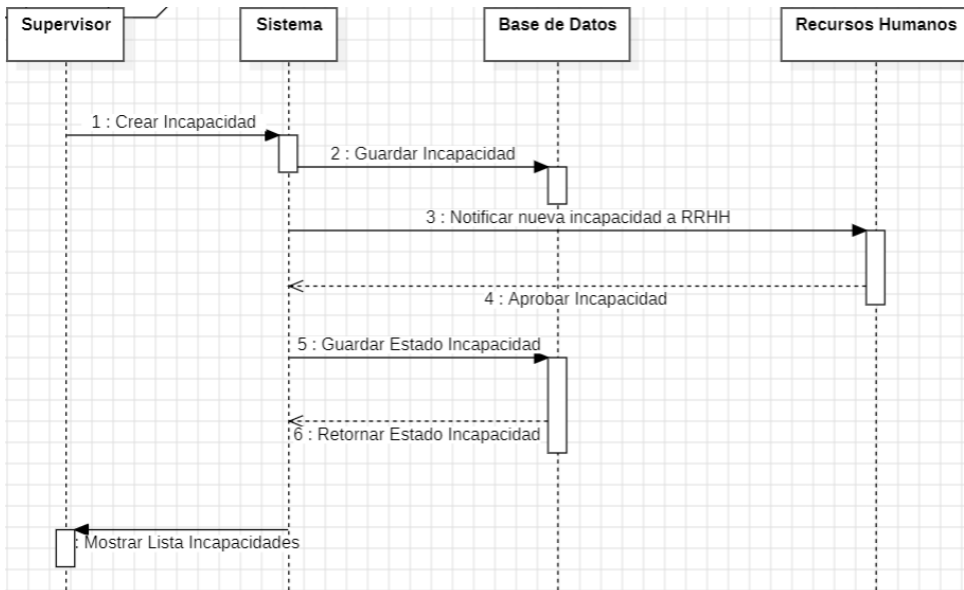
*Diagramas de Secuencia. Marcar Entrada – Salida. Creación de solicitud Horas Extr*



*Fuente: Elaboración Propia (2024)*

**Figura 37.**

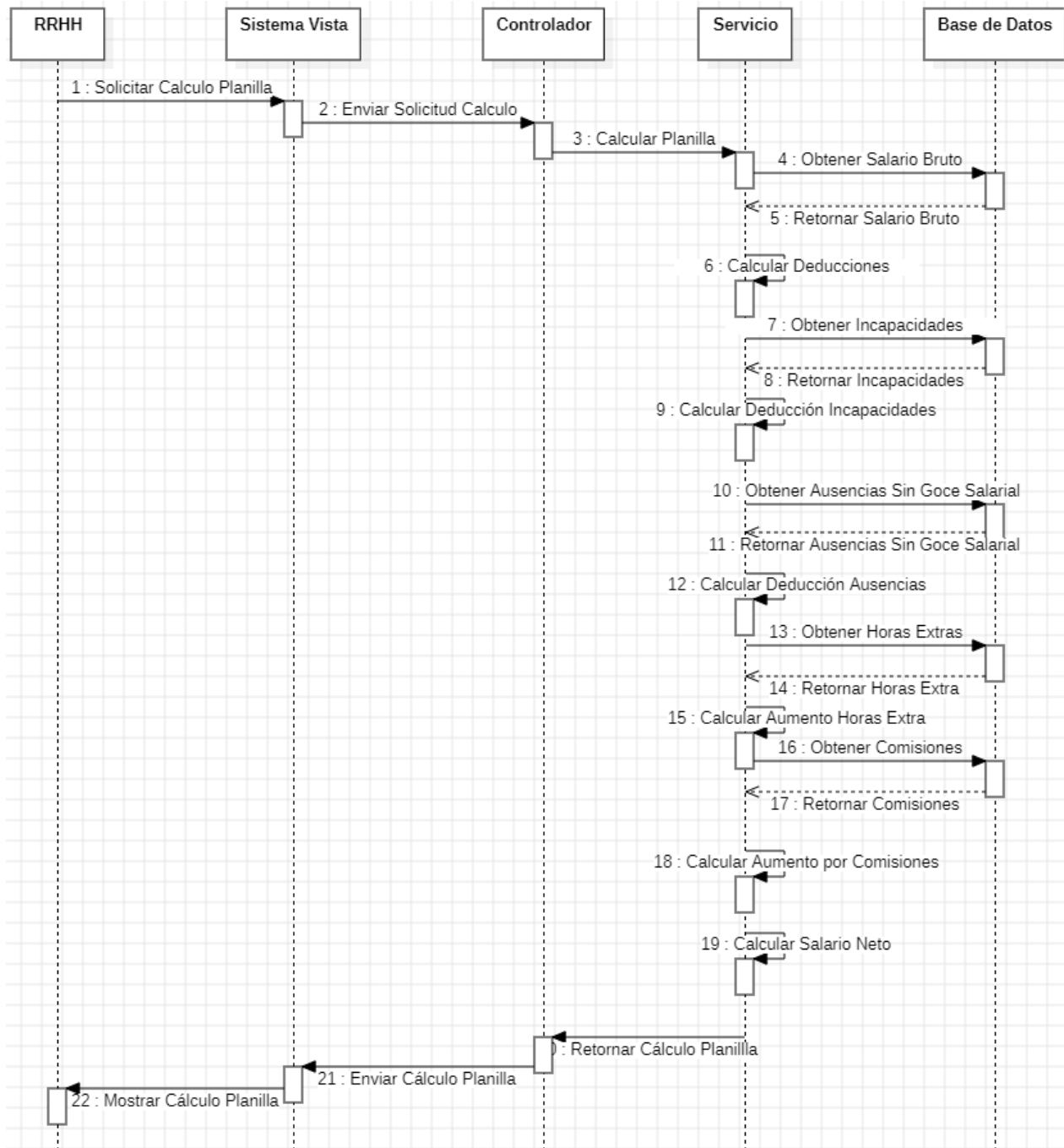
*Diagrama de Secuencia. Registrar Incapacidades*



*Fuente: Elaboración Propia (2024)*

**Figura 38.**

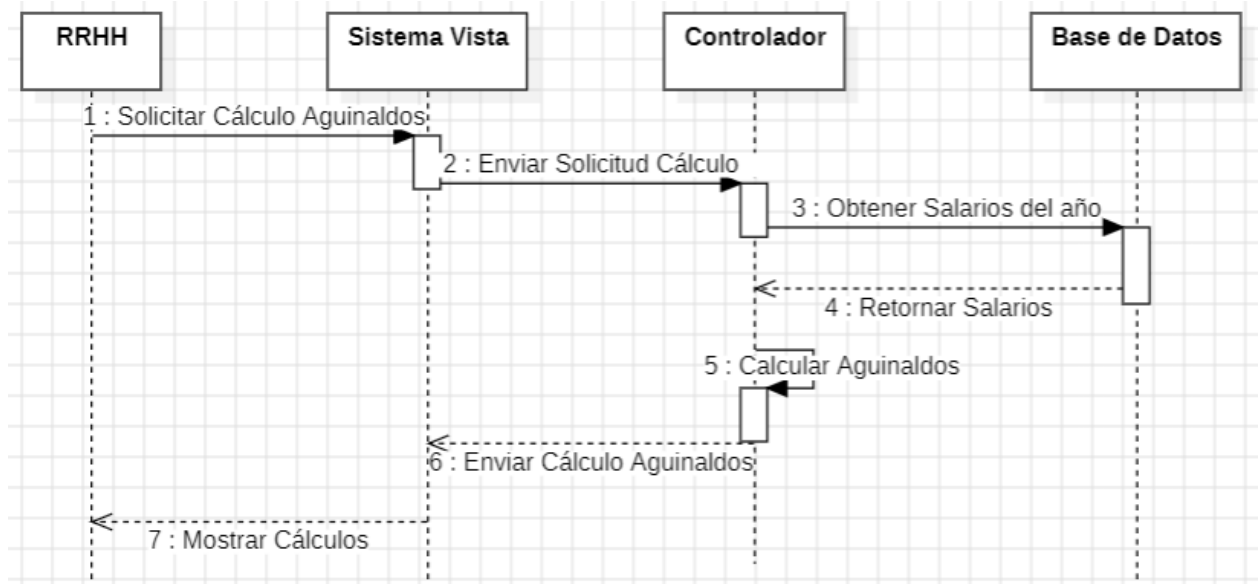
*Diagrama de Secuencia. Calcular Planilla*



*Fuente: Elaboración Propia (2024)*

**Figura 39.**

*Diagrama de Secuencia. Calcular Aguinaldos*



*Fuente: Elaboración Propia (2024)*

## Programación

### Entradas

#### Figura 40.

#### Código para guardar Vacaciones

```

1 referencia
public bool GuardarPedidoVacaciones(VacacionModelo vacacion, usuarioModelo usuario)
{
    bool respuesta;
    int IdUsuario = usuario.IdUsuario;
    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            string _estado = "Pendiente";
            int cantidadDias = 0;
            _conexion.Open();

            for (DateTime diaSemana = vacacion.FechaInicial; diaSemana <= vacacion.FechaFinal; diaSemana = diaSemana.AddDays(1))
            {
                string query = "Select EsDiaDeTrabajo from HorarioTrabajo where IdUsuario = @IdUsuario and DiaDeLaSemana = @DiaDeLaSemana";
                SqlCommand cmd2 = new SqlCommand(query, _conexion);
                string _diaDeLaSemana = diaSemana.DayOfWeek.ToString();
                switch (_diaDeLaSemana)
                {
                    case "Monday": _diaDeLaSemana = "Lunes"; break;
                    case "Tuesday": _diaDeLaSemana = "Martes"; break;
                    case "Wednesday": _diaDeLaSemana = "Miercoles"; break;
                    case "Thursday": _diaDeLaSemana = "Jueves"; break;
                    case "Friday": _diaDeLaSemana = "Viernes"; break;
                    case "Saturday": _diaDeLaSemana = "Sabado"; break;
                    case "Sunday": _diaDeLaSemana = "Domingo"; break;
                    default: throw new InvalidOperationException("error, dia no existe");
                }
                cmd2.Parameters.AddWithValue("IdUsuario", IdUsuario);
                cmd2.Parameters.AddWithValue("DiaDeLaSemana", _diaDeLaSemana);
                using (var dr = cmd2.ExecuteReader())
                {
                    if (dr.Read())
                    {
                        bool esDiaDeTrabajo = dr.GetBoolean(dr.GetOrdinal("EsDiaDeTrabajo"));
                        if (esDiaDeTrabajo && !EsDiaFeriado(diaSemana))
                        {
                            cantidadDias++;
                        }
                    }
                }
            }
        }

        SqlCommand cmd = new SqlCommand("sp_pedidoVacacion_Guardar", _conexion);
        cmd.Parameters.AddWithValue("IdUsuario", IdUsuario);
        cmd.Parameters.AddWithValue("FechaInicial", vacacion.FechaInicial);
        cmd.Parameters.AddWithValue("FechaFinal", vacacion.FechaFinal);
        cmd.Parameters.AddWithValue("Estado", _estado);
        cmd.Parameters.AddWithValue("Dias", cantidadDias);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.ExecuteNonQuery();
    }
    respuesta = true;
}
catch (Exception e)
{
    string error = e.Message;
    respuesta = false;
}
return respuesta;
}

```

Fuente: Elaboración Propia (2024)

**Figura 41.***Código para editar Incapacidades*

```

1 reference
public bool EditarIncapacidad(IncapacidadModelo incapacidad)
{
    bool respuesta;

    try
    {
        var conexion = new Conexion();
        int cantidadDias = 0;

        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            _conexion.Open();

            SqlCommand cmd = new SqlCommand("sp_incapacidad_Editar", _conexion);
            cmd.Parameters.AddWithValue("IdIncapacidad", incapacidad.IdIncapacidad);
            cmd.Parameters.AddWithValue("TipoIncapacidad", incapacidad.TipoIncapacidad);
            cmd.Parameters.AddWithValue("FechaInicial", incapacidad.FechaInicial);
            cmd.Parameters.AddWithValue("FechaFinal", incapacidad.FechaFinal);
            cmd.Parameters.AddWithValue("Estado", incapacidad.Estado);

            TimeSpan _cantidadDiasAux = incapacidad.FechaFinal - incapacidad.FechaInicial;
            int _cantidadDias = (int)_cantidadDiasAux.TotalDays;
            _cantidadDias = _cantidadDias + 1;
            cmd.Parameters.AddWithValue("CantidadDias", _cantidadDias);

            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}

```

*Fuente:* Elaboración Propia (2024)

## Figura 42.

### Código para marcar entrada

```
[HttpGet]
0 references
public IActionResult MarcarEntrada() {
    var IdUsuarioClaim = User.FindFirst(ClaimTypes.NameIdentifier)?.Value;
    int IdUsuario = int.Parse(IdUsuarioClaim);

    var usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
    ViewBag.UsuarioRol = usuario.Rol;

    using (var _conexion = new SqlConnection(new Conexion().getConexionSQL())) {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand("SELECT COUNT(*) FROM MarcaEntradaSalida WHERE IdUsuario = @IdUsuario AND CAST(HoraEntrada AS DATE) = CAST(GETDATE() AS DATE)", _conexion);
        cmd.Parameters.AddWithValue("IdUsuario", IdUsuario);
        int count = (int)cmd.ExecuteScalar();
        ViewBag.EntradaExistente = count > 0; }
    return View();
}

[HttpPost]
0 references
public IActionResult MarcarEntrada(MarcaEntradaSalidaModelo marca, int IdUsuario) {

    if (IdUsuario==0) {
        var IdUsuarioClaim = User.FindFirst(ClaimTypes.NameIdentifier)?.Value;
        IdUsuario = int.Parse(IdUsuarioClaim);
    }
    usuarioModelo usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
    marca.Usuario = usuario;
    marca.IdUsuario = IdUsuario;
    marca.HoraEntrada = DateTime.Now;
    marca.HoraSalida = null;
    marca.HorasTrabajo = null;
    marca.Dia= DateTime.Now;

    var respuesta = marcaDatos.MarcarEntrada(marca);
    if (respuesta)
    {
        return Json(new { success = true });
    }
    else
    {
        return Json(new { success = false, message = "Error al marcar entrada." });
    }
}
}
```

*Fuente:* Elaboración Propia (2024)

**Figura 43.***Código para calcular planilla*

```

[Authorize(Roles = "Administrador, SuperAdmin, RRHH")]
0 references
public IActionResult CalcularPlanilla()
{
    SalarioModelo salarioModelo = new SalarioModelo
    {
        EnumMes = new List<SelectListItem> {
            new SelectListItem { Value = "Enero", Text = "Enero" },
            new SelectListItem { Value = "Febrero", Text = "Febrero" },
            new SelectListItem { Value = "Marzo", Text = "Marzo" },
            new SelectListItem { Value = "Abril", Text = "Abril" },
            new SelectListItem { Value = "Mayo", Text = "Mayo" },
            new SelectListItem { Value = "Junio", Text = "Junio" },
            new SelectListItem { Value = "Julio", Text = "Julio" },
            new SelectListItem { Value = "Agosto", Text = "Agosto" },
            new SelectListItem { Value = "Septiembre", Text = "Septiembre" },
            new SelectListItem { Value = "Octubre", Text = "Octubre" },
            new SelectListItem { Value = "Noviembre", Text = "Noviembre" },
            new SelectListItem { Value = "Diciembre", Text = "Diciembre" },
        },
        EnumAño = new List<SelectListItem>{
            new SelectListItem { Value = "2024", Text = "2024" },
            new SelectListItem { Value = "2025", Text = "2025" },
            new SelectListItem { Value = "2026", Text = "2026" },
            new SelectListItem { Value = "2027", Text = "2027" },
            new SelectListItem { Value = "2028", Text = "2028" },
        }
    };

    int mesActual = DateTime.Now.Month;
    string nombreMesActual = salarioModelo.EnumMes[mesActual - 1].Text;
    salarioModelo.Mes = nombreMesActual;

    int añoActual = DateTime.Now.Year;
    salarioModelo.Año = añoActual;

    int díaActual = DateTime.Now.Day;
    if (díaActual < 15)
    {
        salarioModelo.EsPrimeraQuincena = true;
    }
    else {
        salarioModelo.EsPrimeraQuincena = false;
    }
}

ViewBag.SalarioModelo = salarioModelo;
var listaSalarios = new List<SalarioModelo>();

return View(listaSalarios);
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 44.***Código para calcular liquidación*

```
[HttpPost]
0 references
public IActionResult CalcularLiquidacion(int IdUsuario, string TipoLiquidacion, DateTime FechaLiquidacion)
{
    usuarioModelo usuarioModelo = new usuarioModelo();
    liquidacionModelo liquidacionModelo = new liquidacionModelo();

    double montoLiquidacion = liquidacionDatos.CalcularLiquidacion(IdUsuario, TipoLiquidacion, FechaLiquidacion);

    liquidacionModelo.IdUsuario = IdUsuario;
    liquidacionModelo.TipoLiquidacion = TipoLiquidacion;
    liquidacionModelo.FechaLiquidacion = FechaLiquidacion;
    liquidacionModelo.Usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
    liquidacionModelo.MontoPorLiquidacion = montoLiquidacion;

    if (montoLiquidacion >= 0)
    {
        ViewBag.IdUsuario = IdUsuario;
        ViewBag.TipoLiquidacion = TipoLiquidacion;
        ViewBag.FechaLiquidacion = FechaLiquidacion;

        liquidacionModelo.MontoPorLiquidacion = montoLiquidacion;
        ViewBag.MontoLiquidacion = montoLiquidacion;
        liquidacionModelo.Usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
        return View(liquidacionModelo);
    }
    else
        return View(liquidacionModelo);
}
```

## Salidas

### Figura 45.

*Código para obtener lista de pedidos horas extras*

```

public List<HorasExtraModelo> ObtenerListaPedidoHorasExtra(usuarioModelo usuarioLoggeado)
{
    var Lista = new List<HorasExtraModelo>();
    var conexion = new Conexion();
    usuarioModelo usuario = new usuarioModelo();

    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand("sp_pedidosHorasExtra_Listar", _conexion);
        cmd.CommandType = CommandType.StoredProcedure;

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);

                Lista.Add(new HorasExtraModelo()
                {
                    IdPedidoHorasExtra = Convert.ToInt32(dr["IdPedidoHorasExtra"]),
                    IdUsuario = idUsuario,
                    CantidadHoras = Convert.ToDouble(dr["CantidadHoras"]),
                    FechaSolicitada = Convert.ToDateTime(dr["FechaSolicitada"]),
                    Estado = dr["Estado"].ToString(),
                    Usuario = usuarioDatos.ObtenerUsuario(idUsuario),
                });
            }
        }

        if (usuarioLoggeado.Rol == "Supervisor")
        {
            Lista = Lista.Where(c => c.Usuario.Rol == "Colaborador").ToList();
        }

        return Lista;
    }
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 46.**

*Código para obtener lista de comisiones para usuario final*

```
public List<ComisionModelo> ObtenerListaComisionesPorUsuario(int IdUsuario)
{
    var Lista = new List<ComisionModelo>();
    usuarioModelo usuario = new usuarioModelo();
    string query = "SELECT * FROM Comision WHERE IdUsuario = @IdUsuario";
    var conexion = new Conexion();
    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        SqlCommand cmd = new SqlCommand(query, _conexion);
        cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
        _conexion.Open();
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                Lista.Add(new ComisionModelo()
                {
                    IdUsuario = idUsuario,
                    Fecha = Convert.ToDateTime(dr["Fecha"]),
                    Monto = Convert.ToDouble(dr["Monto"]),
                    IdFactura = dr["IdFactura"].ToString()
                });
            }
        }
    }
    return Lista;
}
```

*Fuente: Elaboración Propia (2024)*

**Figura 47.**

*Código para obtener pedidos de vacaciones por usuario*

```
public List<VacacionModelo> ObtenerListaVacacionesPorUsuario(int IdUsuario)
{
    var Lista = new List<VacacionModelo>();
    usuarioModelo usuario = new usuarioModelo();
    string query = "SELECT * FROM PedidoVacacion WHERE IdUsuario = @IdUsuario";
    var conexion = new Conexion();
    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        SqlCommand cmd = new SqlCommand(query, _conexion);
        cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
        _conexion.Open();
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                Lista.Add(new VacacionModelo()
                {
                    FechaInicial = Convert.ToDateTime(dr["FechaInicial"]),
                    FechaFinal = Convert.ToDateTime(dr["FechaFinal"]),
                    Estado = dr["Estado"].ToString(),
                    CantidadDias = Convert.ToInt32(dr["Dias"]),
                    IdUsuario = idUsuario,
                    IdPedidoVacaciones = Convert.ToInt32(dr["IdVacacion"])
                });
            }
        }
    }
    return Lista;
}
```

*Fuente: Elaboración Propia (2024)*

**Figura 48.**

*Código para obtener lista horarios trabajo*

```
public List<HorarioTrabajoModelo> ObtenerListaHorarioTrabajo()
{
    var Lista = new List<HorarioTrabajoModelo>();
    var conexion = new Conexion();
    usuarioModelo usuario = new usuarioModelo();
    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand("sp_horarioTrabajo_Listar", _conexion);
        cmd.CommandType = CommandType.StoredProcedure;

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                Lista.Add(new HorarioTrabajoModelo()
                {
                    IdHorarioTrabajo = Convert.ToInt32(dr["IdHorarioTrabajo"]),
                    IdUsuario = idUsuario,
                    DiaDeLaSemana = dr["DiaDeLaSemana"].ToString(),
                    EsDiaDeTrabajo = Convert.ToBoolean(dr["EsDiaDeTrabajo"]),
                    Usuario = usuarioDatos.ObtenerUsuario(idUsuario),
                    //Usuario = ObtenerUsuario(idUsuario),
                });
            }
        }
        return Lista;
    }
}
```

*Fuente: Elaboración Propia (2024)*

## Procesos

### Figura 49.

#### Código para marcar entrada de jornada laboral

```

UsuarioDatos usuarioDatos = new UsuarioDatos();
MarcaEntradaSalidaDatos marcaDatos = new MarcaEntradaSalidaDatos();
[HttpGet]
0 references
public IActionResult MarcarEntrada() {
    var IdUsuarioClaim = User.FindFirst(ClaimTypes.NameIdentifier)?.Value;
    int IdUsuario = int.Parse(IdUsuarioClaim);

    var usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
    ViewBag.UsuarioRol = usuario.Rol;

    using (var _conexion = new SqlConnection(new Conexion().getConexionSQL())) {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand("SELECT COUNT(*) FROM MarcaEntradaSalida WHERE IdUsuario = @IdUsuario AND CAST(HoraEntrada AS DATE) = CAST(GETDATE() AS DATE)", _conexion);
        cmd.Parameters.AddWithValue("IdUsuario", IdUsuario);
        int count = (int)cmd.ExecuteScalar();
        ViewBag.EntradaExistente = count > 0;
    }
    return View();
}

[HttpPost]
0 references
public IActionResult MarcarEntrada(MarcaEntradaSalidaModelo marca, int IdUsuario) {

    if (IdUsuario==0) {
        var IdUsuarioClaim = User.FindFirst(ClaimTypes.NameIdentifier)?.Value;
        IdUsuario = int.Parse(IdUsuarioClaim);
    }
    usuarioModelo usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
    marca.Usuario = usuario;
    marca.IdUsuario = IdUsuario;
    marca.HoraEntrada = DateTime.Now;
    marca.HoraSalida = null;
    marca.HorasTrabajo = null;
    marca.Dia= DateTime.Now;

    var respuesta = marcaDatos.MarcarEntrada(marca);
    if (respuesta)
    {
        return Json(new { success = true });
    }
    else
    {
        return Json(new { success = false, message = "Error al marcar entrada." });
    }
}

```

Fuente: Elaboración Propia (2024)

## Figura 50.

*Código para calcular reducciones de salario para permisos sin goce salarial*

```

//*****Lista de Permisos*****//
var ListaPermisos = new List<permisoModelo>();
double reduccionPermisosSinGoce = 0;

string queryPermisos = "SELECT * FROM Permiso WHERE IdUsuario = @IdUsuario and TipoPermiso=@TipoPermiso and Estado=@Estado and FechaSolicitada >= @fechaInicial AND FechaSolicitada <= @fechaFinal;";

try
{
    double monto;
    int permisosSinGoceSalarialContador = 0;
    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand(queryPermisos, _conexion);
        cmd.Parameters.AddWithValue("@fechaInicial", fechaInicial);
        cmd.Parameters.AddWithValue("@fechaFinal", fechaFinal);
        cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
        cmd.Parameters.AddWithValue("@TipoPermiso", "Ausencia con Goce salarial");
        cmd.Parameters.AddWithValue("@Estado", "Aprobada");
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                ListaPermisos.Add(new permisoModelo()
                {
                    IdPermiso = Convert.ToInt32(dr["IdPermiso"]),
                    //IdUsuario = Convert.ToInt32(dr["IdUsuario"]),
                    IdUsuario = idUsuario,
                    TipoPermiso = dr["TipoPermiso"].ToString(),
                    FechaDeSolicitud = Convert.ToDateTime(dr["FechaDeSolicitud"]),
                    FechaInicial = Convert.ToDateTime(dr["FechaInicial"]),
                    FechaFinal = Convert.ToDateTime(dr["FechaFinal"]),
                    Usuario = ObtenerUsuario(idUsuario),

                    Estado = dr["Estado"].ToString()
                });
                permisosSinGoceSalarialContador++;
                //Debug.WriteLine(permisosSinGoceSalarialContador);
                reduccionPermisosSinGoce = montoSalario / 30 * -permisosSinGoceSalarialContador;
            }
        }
    }
}
catch (Exception e)
{
    string error = e.Message;
    respuesta = false;
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 51.**

*Código para obtener rango de fechas para cálculos salariales*

```

1 reference
public Tuple<DateTime, DateTime> ObtenerRangoFechas(string mes, int anho, bool esPrimeraQuincena)
{
    int _mes = 0;

    switch (mes)
    {
        case "Enero":
            _mes = 1;
            break;
        case "Febrero":
            _mes = 2;
            break;
        case "Marzo":
            _mes = 3;
            break;
        case "Abril":
            _mes = 4;
            break;
        case "Mayo":
            _mes = 5;
            break;
        case "Junio":
            _mes = 6;
            break;
        case "Julio":
            _mes = 7;
            break;
        case "Agosto":
            _mes = 8;
            break;
        case "Septiembre":
            _mes = 9;
            break;
        case "Octubre":
            _mes = 10;
            break;
        case "Noviembre":
            _mes = 11;
            break;
        case "Diciembre":
            _mes = 12;
            break;

        default:
            throw new InvalidOperationException("ese mes no existe");
    }

    DateTime fechaInicial = new DateTime(anho, _mes, esPrimeraQuincena ? 1 : 16);
    DateTime fechaFinal = new DateTime(anho, _mes, esPrimeraQuincena ? 15 : DateTime.DaysInMonth(anho, _mes));

    return new Tuple<DateTime, DateTime>(fechaInicial, fechaFinal);
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 52.***Código para guardar bitácora de usuario al ser contratado*

```

public bool GuardarBitacoraUsuarioActivo(usuarioModelo usuario, bool EstadoActivo)
{
    BitacoraModelo bitacora = new BitacoraModelo();
    bitacora.IdUsuario = usuario.IdUsuario;
    bool respuesta;
    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            _conexion.Open();

            if (EstadoActivo)
            {
                bitacora.FechaEntrada = usuario.FechaDeEntrada;
                bitacora.FechaSalida = null;
                bitacora.EstadoActivo = true;

                SqlCommand cmd = new SqlCommand("sp_bitacora_Guardar", _conexion);
                cmd.Parameters.AddWithValue("IdUsuario", bitacora.IdUsuario);
                cmd.Parameters.AddWithValue("FechaEntrada", bitacora.FechaEntrada);
                cmd.Parameters.AddWithValue("FechaSalida", bitacora.FechaSalida ?? (object)DBNull.Value);
                cmd.Parameters.AddWithValue("EstadoActivo", bitacora.EstadoActivo);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.ExecuteNonQuery();
            }
            else
            {
                SqlCommand cmdEntrada = new SqlCommand("SELECT FechaEntrada FROM Bitacora WHERE IdUsuario = @IdUsuario AND FechaSalida IS NULL", _conexion);
                cmdEntrada.Parameters.AddWithValue("@IdUsuario", bitacora.IdUsuario);
                using (var dr = cmdEntrada.ExecuteReader())
                {
                    if (dr.Read())
                    {
                        bitacora.FechaEntrada = dr.GetDateTime(dr.GetOrdinal("FechaEntrada"));
                        bitacora.FechaSalida = DateTime.Today;
                        bitacora.EstadoActivo = false;
                    }
                }

                SqlCommand cmd = new SqlCommand("sp_bitacora_DesactivarUsuario", _conexion);
                cmd.Parameters.AddWithValue("IdUsuario", bitacora.IdUsuario);
                //cmd.Parameters.AddWithValue("FechaEntrada", bitacora.FechaEntrada);
                cmd.Parameters.AddWithValue("FechaSalida", bitacora.FechaSalida ?? (object)DBNull.Value);
                cmd.Parameters.AddWithValue("EstadoActivo", bitacora.EstadoActivo);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.ExecuteNonQuery();
            }
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 53.**

*Código para calcular salarios de meses previos para realizar el cálculo de cesantía y preaviso*

```
//-----Calcular salario meses previos para sacar salario promedio de cesantía y preaviso
double promedioSalario = 0;
double totalSalario = 0;
DateTime fechaInicioCalculo = FechaLiquidacion.AddMonths(-6);
int anho = FechaLiquidacion.Year;
int mesNumero = FechaLiquidacion.Month - 1;
string[] meses = { "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre" };

using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
{
    _conexion.Open();
    //variables auxiliares para cálculo
    int diasInicio = fechaInicioCalculo.Day;
    int dia1 = diasInicio;
    int dia2 = 0;

    // Calcular salario primera quincena parcial
    if (diasInicio > 0 && diasInicio <= 15)
    {
        dia2 = 15;
    }
    if (diasInicio > 15 && diasInicio < 31)
    {
        dia2 = 30;
    }
    salarioModelo = salarioDatos.CrearCalculoSalario(meses[fechaInicioCalculo.Month - 1], fechaInicioCalculo.Year, IdUsuario, false, dia1, dia2);
    double salarioParcialInicio = (double)salarioModelo.SalarioNeto;
    totalSalario += salarioParcialInicio;

    // Calculo del salario de meses completos
    for (int mes = fechaInicioCalculo.Month + 1; mes < FechaLiquidacion.Month; mes++)
    {
        string mesString = meses[(mes - 1) % 12];
        string query = "SELECT Monto FROM CalculoSalario WHERE IdUsuario = @IdUsuario AND Anho = @Anho AND Mes = @Mes";

        using (var cmd = new SqlCommand(query, _conexion))
        {
            cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
            cmd.Parameters.AddWithValue("@Anho", FechaLiquidacion.Year);
            cmd.Parameters.AddWithValue("@Mes", mesString);

            using (var dr = cmd.ExecuteReader())
            {
                while (dr.Read())
                {
                    double salario = Convert.ToDouble(dr["Monto"]);
                    totalSalario += salario;
                }
            }
        }
    }

    // Calcular salario segunda quincena parcial
    int diasFin = FechaLiquidacion.Day;
    if (diasInicio > 0 && diasInicio <= 15)
    {
        dia1 = 1;
    }
    if (diasInicio > 16 && diasInicio < 31)
    {
        dia1 = 16;
    }
    double salarioParcialFin = (double)salarioModelo.SalarioNeto;
    totalSalario += salarioParcialFin;
}
double promedioSalarioDiario = totalSalario / 180;
```

*Fuente: Elaboración Propia (2024)*

**Figura 54.***Código para calcular planilla*

```

public List<SalarioModelo> CalcularPlanilla(string Mes, int Anho, Boolean esPrimerQuincena)
{
    //auxDia no se usa para el cálculo salario
    int dial = 0;
    int dia2 = 0;
    SalarioDatos salario = new SalarioDatos();
    double totalPlanilla = 0;
    double montoPorUsuario = 0;

    usuarioModelo usuario = new usuarioModelo();
    UsuarioDatos usuarioDatos = new UsuarioDatos();
    SalarioModelo salarioModelo = new SalarioModelo();
    var Lista = new List<SalarioModelo>();
    var conexion = new Conexion();

    string query = "SELECT * FROM Usuario where EstadoActivo=1;";

    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand(query, _conexion);

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                salarioModelo = salario.CrearCalculoSalario(Mes, Anho, idUsuario, esPrimerQuincena, dial, dia2);
                usuario = usuarioDatos.ObtenerUsuario(idUsuario);

                montoPorUsuario = (double)salarioModelo.SalarioNeto;
                Lista.Add(new SalarioModelo()
                {
                    IdUsuario = idUsuario,
                    SalarioNeto = montoPorUsuario,
                    AumentoPorHorasExtra = salarioModelo.AumentoPorHorasExtra,
                    AumentoPorComisiones = salarioModelo.AumentoPorComisiones,
                    ReduccionIncapacidades = salarioModelo.ReduccionIncapacidades,
                    ReduccionPermisosSinGoce = salarioModelo.ReduccionPermisosSinGoce,
                    Usuario = usuario
                });
                Debug.WriteLine("El monto de salario para " + usuario.Nombre + usuario.Apellido1 + "es de: "
                    + montoPorUsuario + "con un total de aumento por horas extra de:" + salarioModelo.AumentoPorHorasExtra);
                totalPlanilla = totalPlanilla + montoPorUsuario;
            }
        }
    }
    return Lista;
}

```

*Fuente:* Elaboración Propia (2024)

**Figura 55.***Código cliente para mostrar alertas*

```

function ObtenerPedidosPendiente()
{ $.ajax({ url: '/Usuario/ObtenerPendientes',
type: 'GET',
success: function (response) {
    if (response.hayPendientes) {
        mostrarAlerta('Tiene nuevas solicitudes');
    }
}, error: function (error) {
    console.log('Error', error);
}
});
}
setInterval(ObtenerPedidosPendiente, 30000);

function ObtenerAlertasParaUsuario() {
$.ajax(
{
    url: '/Notificacion/ObtenerAlertasParaUsuario',
    type: 'GET',
    dataType: 'json',
    success: function (response)
    {
        if (response.hayPendientes)
        {
            var primeraNotificacion = response.notificaciones[0];
            var message = `Su ${primeraNotificacion.tipoNotificacion} ha sido ${primeraNotificacion.estado}`;
            mostrarAlerta(message);
            response.notificaciones.forEach(function (notificacion)
            {
                var idNotificacion = notificacion.IdNotificacion;
                MarcarNotificacionComoLeida(notificacion.idNotificacion);
            });
        }
    },
    error:
    function (error)
    {
        console.log('Error', error);
    }
});
}
}

```

*Fuente: Elaboración Propia (2024)*

## Validaciones

### Figura 56.

*Código para revisar si existen pedidos de vacaciones en las mismas fechas en las cuales se está creando una nueva solicitud de vacaciones*

```

public bool ExistePedidoVacacionOverlap(VacacionModelo vacacion, usuarioModelo usuario)
{
    bool existeOverlap;
    int IdUsuario = usuario.IdUsuario;

    var query = @"
SELECT COUNT(*)
FROM PedidoVacacion
WHERE IdUsuario = @IdUsuario
AND (
    (@FechaInicial >= FechaInicial AND @FechaInicial <= FechaFinal)
OR (@FechaFinal >= FechaInicial AND @FechaFinal <= FechaFinal)
OR (@FechaInicial <= FechaInicial AND @FechaFinal >= FechaFinal)
)";

    using (var conexion = new SqlConnection(new Conexion().getConexionSQL()))
    {
        using (var command = new SqlCommand(query, conexion))
        {
            command.Parameters.AddWithValue("@IdUsuario", IdUsuario);
            command.Parameters.AddWithValue("@FechaInicial", vacacion.FechaInicial);
            command.Parameters.AddWithValue("@FechaFinal", vacacion.FechaFinal);

            conexion.Open();
            var count = (int)command.ExecuteScalar();
            existeOverlap = count > 0;
        }
    }

    return existeOverlap;
}

```

*Fuente: Elaboración Propia (2024)*

## Módulos

**Figura 57.**

*Código para mantenimiento usuarios*

```
public bool EditarUsuario(usuarioModelo usuario)
{
    bool respuesta;

    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            _conexion.Open();
            SqlCommand cmd = new SqlCommand("sp_usuario_Editar", _conexion);
            cmd.Parameters.AddWithValue("idUsuario", usuario.IdUsuario);
            cmd.Parameters.AddWithValue("nombre", usuario.Nombre);
            cmd.Parameters.AddWithValue("apellido1", usuario.Apellido1);
            cmd.Parameters.AddWithValue("apellido2", usuario.Apellido2);
            cmd.Parameters.AddWithValue("telefono", usuario.Telefono);
            cmd.Parameters.AddWithValue("correo", usuario.Correo);
            cmd.Parameters.AddWithValue("contrasena", usuario.Contrasena);
            cmd.Parameters.AddWithValue("salarioBase", usuario.SalarioBase);
            cmd.Parameters.AddWithValue("fechaDeEntrada", usuario.FechaDeEntrada);
            cmd.Parameters.AddWithValue("fechaDeSalida", usuario.FechaDeSalida ?? (object)DBNull.Value);
            cmd.Parameters.AddWithValue("EstadoActivo", usuario.EstadoActivo);
            cmd.Parameters.AddWithValue("rol", usuario.Rol);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}
```

*Fuente: Elaboración Propia (2024)*

**Figura 58.***Código para hacer el cálculo de planilla, parte 1*

```

public double CrearCalculoSalario(string Mes, int Anho, int IdUsuario, Boolean esPrimerQuincena, int dial, int dia2)
{
    var conexion = new Conexion();
    bool respuesta = false;

    UsuarioDatos usuarioDatos = new UsuarioDatos();
    usuarioModelo usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
    usuario.IdUsuario = IdUsuario;

    double montoSalario = 0;
    double montoSalarioPorDia = 0;
    montoSalario = usuario.SalarioBase / 2;
    montoSalarioPorDia = usuario.SalarioBase / 15;

    var rangoFechas = ObtenerRangoFechas(Mes, Anho, esPrimerQuincena);
    DateTime _fechaInicial = rangoFechas.Item1;
    DateTime _fechaFinal = rangoFechas.Item2;

    if (dial > 0 && dial <= 15) {
        esPrimerQuincena = true;
        _fechaInicial = _fechaInicial.AddDays(dial - 1);
        if (dial == 1) {
            _fechaFinal = new DateTime(_fechaFinal.Year, _fechaFinal.Month, dia2);
            montoSalario = montoSalarioPorDia / 2 * (dia2 - dial + 1);
        }

        montoSalario = montoSalarioPorDia * (15 - dial + 1);
        Debug.WriteLine(montoSalario);
    }
    if (dial > 15 && dial < 32)
    {
        esPrimerQuincena = false;
        _fechaInicial = new DateTime(_fechaFinal.Year, _fechaFinal.Month, dial);
        _fechaFinal = new DateTime(_fechaFinal.Year, _fechaFinal.Month, dia2);
        if (dial == 16)
        {
            _fechaFinal = new DateTime(_fechaFinal.Year, _fechaFinal.Month, dia2);
            montoSalario = montoSalarioPorDia / 2 * (dia2 - dial + 1);
        }
        Debug.WriteLine(_fechaInicial);
    }

    string fechaInicial = _fechaInicial.ToString("yyyy-MM-dd");
    string fechaFinal = _fechaFinal.ToString("yyyy-MM-dd");
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 59.**

*Código para hacer el cálculo de planilla, parte 2 (Permisos sin goce salarial)*

```

//*****Lista de Permisos*****//
var ListaPermisos = new List<PermisoModelo>();
double reduccionPermisosSinGoce = 0;

string queryPermisos = "SELECT * FROM Permiso WHERE IdUsuario = @IdUsuario and TipoPermiso=@TipoPermiso and Estado=@Estado and FechaSolicitada >= @fechaInicial AND FechaSolicitada <= @fechaFinal;";

try
{
    double monto;
    int permisosSinGoceSalarialContador = 0;
    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand(queryPermisos, _conexion);
        cmd.Parameters.AddWithValue("@fechaInicial", fechaInicial);
        cmd.Parameters.AddWithValue("@fechaFinal", fechaFinal);
        cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
        cmd.Parameters.AddWithValue("@TipoPermiso", "Ausencia con Goce salarial");
        cmd.Parameters.AddWithValue("@Estado", "Aprobada");
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                ListaPermisos.Add(new PermisoModelo()
                {
                    IdPermiso = Convert.ToInt32(dr["IdPermiso"]),
                    //IdUsuario = Convert.ToInt32(dr["IdUsuario"]),
                    IdUsuario = idUsuario,
                    TipoPermiso = dr["TipoPermiso"].ToString(),
                    FechaDeSolicitud = Convert.ToDateTime(dr["FechaDeSolicitud"]),
                    FechaInicial = Convert.ToDateTime(dr["FechaInicial"]),
                    FechaFinal = Convert.ToDateTime(dr["FechaFinal"]),
                    Usuario = ObtenerUsuario(idUsuario),

                    Estado = dr["Estado"].ToString()
                });
                permisosSinGoceSalarialContador++;
                reduccionPermisosSinGoce = montoSalario / 30 * -permisosSinGoceSalarialContador;
            }
        }
    }
}
catch (Exception e)
{
    string error = e.Message;
    respuesta = false;
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 60.**

*Código para hacer el cálculo de planilla, parte 3 (Comisiones)*

```

//*****Lista Comisiones*****//
var ListaComisiones = new List<ComisionModelo>();
double aumentoPorComisiones = 0;
string queryComisiones = "SELECT * FROM Comision WHERE IdUsuario = @IdUsuario and Fecha >= @fechaInicial AND Fecha <= @fechaFinal;";

try
{
    double monto=0;
    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand(queryComisiones, _conexion);
        cmd.Parameters.AddWithValue("@fechaInicial", fechaInicial);
        cmd.Parameters.AddWithValue("@fechaFinal", fechaFinal);
        cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                monto = Convert.ToDouble(dr["Monto"]);
                ListaComisiones.Add(new ComisionModelo()
                {
                    IdUsuario = idUsuario,
                    Monto = Convert.ToDouble(dr["Monto"]),
                    Fecha = Convert.ToDateTime(dr["Fecha"]),
                    Usuario = ObtenerUsuario(idUsuario),
                });
                aumentoPorComisiones = aumentoPorComisiones + monto;
            }
        }
    }
}
catch (Exception e)
{
    string error = e.Message;
    respuesta = false;
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 61.****Código para hacer el cálculo de planilla, parte 4 (Horas Extra, parcial 1)**

```

//*****Lista Horas Extra*****
var ListaHorasExtra = new List<HorasExtraModelo>();
double aumentoPorHorasExtra = 0;

string queryHorasExtra = "SELECT * FROM PedidoHorasExtra WHERE IdUsuario = @IdUsuario and Estado='Aprobada' and FechaSolicitada >= @fechaInicial AND FechaSolicitada <= @fechaFinal;";

try
{
    int cantidadHoras = 0;
    int contadorHoras = 0;
    double montoHorasExtra = 0;

    using (var _conexion = new SqlConnection(conexion.getConnectionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand(queryHorasExtra, _conexion);
        cmd.Parameters.AddWithValue("@fechaInicial", fechaInicial);
        cmd.Parameters.AddWithValue("@fechaFinal", fechaFinal);
        cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                cantidadHoras = Convert.ToInt32(dr["CantidadHoras"]);
                DateTime fechaSolicitada = Convert.ToDateTime(dr["FechaSolicitada"]);

                ListaHorasExtra.Add(new HorasExtraModelo()
                {
                    IdUsuario = idUsuario,
                    CantidadHoras = cantidadHoras,
                    FechaSolicitada = fechaSolicitada
                });

                string _diaDeLaSemana = fechaSolicitada.DayOfWeek.ToString();
                switch (_diaDeLaSemana)
                {
                    case "Monday": _diaDeLaSemana = "Lunes"; break;
                    case "Tuesday": _diaDeLaSemana = "Martes"; break;
                    case "Wednesday": _diaDeLaSemana = "Miercoles"; break;
                    case "Thursday": _diaDeLaSemana = "Jueves"; break;
                    case "Friday": _diaDeLaSemana = "Viernes"; break;
                    case "Saturday": _diaDeLaSemana = "Sabado"; break;
                    case "Sunday": _diaDeLaSemana = "Domingo"; break;
                    default: throw new InvalidOperationException("error, dia no existe");
                }

                // Validar si es día de trabajo
                bool esDiaDeTrabajo = false;
                bool esDiaFeriado = EsDiaFeriado(fechaSolicitada);

                using (var _conexion2 = new SqlConnection(conexion.getConnectionSQL()))
                {
                    _conexion2.Open();
                    string query = "Select EsDiaDeTrabajo from HorarioTrabajo where IdUsuario = @IdUsuario and DiaDeLaSemana = @DiaDeLaSemana";
                    SqlCommand cmd2 = new SqlCommand(query, _conexion2);
                    cmd2.Parameters.AddWithValue("@IdUsuario", idUsuario);
                    cmd2.Parameters.AddWithValue("@DiaDeLaSemana", _diaDeLaSemana);

                    using (var dr2 = cmd2.ExecuteReader())
                    {
                        if (dr2.Read())
                        {
                            esDiaDeTrabajo = dr2.GetBoolean(dr2.GetOrdinal("EsDiaDeTrabajo"));
                        }
                    }
                }

                // Multiplicador
                double multiplicador = 1.5;
            }
        }
    }
}

```

*Fuente:* Elaboración Propia (2024)

**Figura 62.**

*Código para hacer el cálculo de planilla, parte 5 (Horas Extra, parcial 2)*

```
        // Multiplicador
        double multiplicador = 1.5;
        if (!esDiaDeTrabajo || esDiaFeriado)
        {
            multiplicador = 2;
        }

        montoHorasExtra += cantidadHoras * multiplicador;
        contadorHoras += cantidadHoras;
    }

    aumentoPorHorasExtra = montoSalario / 30 / 8 * montoHorasExtra;
}
}
}
catch (Exception e)
{
    string error = e.Message;
    respuesta = false;
}
```

*Fuente: Elaboración Propia (2024)*

**Figura 63.***Código para hacer el cálculo de planilla, parte 6 (Incapacidades)*

```

//*****Lista Incapacidades*****
var ListaIncapacidades = new List<IncapacidadModelo>();
double reduccionIncapacidades = 0;

string queryIncapacidades = "SELECT * FROM Incapacidad WHERE IdUsuario = @IdUsuario and Estado='Aprobada' and FechaInicial >= @fechaInicial AND FechaFinal <= @fechaFinal;";

try
{
    int cantidadDias = 0;
    int contadorHoras = 0;
    double montoHorasAPagar = 0;
    double monto = 0;
    double pagoPorHora = 0;
    // int permisosSinGoceSalarialContador = 0;
    using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
    {
        _conexion.Open();
        SqlCommand cmd = new SqlCommand(queryIncapacidades, _conexion);
        cmd.Parameters.AddWithValue("@fechaInicial", fechaInicial);
        cmd.Parameters.AddWithValue("@fechaFinal", fechaFinal);
        cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                int idUsuario = Convert.ToInt32(dr["IdUsuario"]);
                cantidadDias = Convert.ToInt32(dr["CantidadDias"]);
                ListaIncapacidades.Add(new IncapacidadModelo()
                {
                    //IdUsuario = Convert.ToInt32(dr["IdUsuario"]),
                    IdUsuario = idUsuario,
                    CantidadDias = Convert.ToInt32(dr["CantidadDias"]),
                    FechaInicial = Convert.ToDateTime(dr["FechaInicial"]),
                    FechaFinal = Convert.ToDateTime(dr["FechaFinal"])
                    //Usuario = ObtenerUsuario(idUsuario),
                });
                if (cantidadDias <= 3)
                {
                    contadorHoras = cantidadDias*8;
                    montoHorasAPagar = 0;
                    monto = montoSalario / 30 / 8 * montoHorasAPagar;
                }
                else {
                    contadorHoras = cantidadDias * 8;
                    montoHorasAPagar = contadorHoras * 0.6;
                    monto = montoSalario / 30 / 8 * montoHorasAPagar;
                }

                reduccionIncapacidades = reduccionIncapacidades + monto;
                /* Debug.WriteLine(contadorHoras);
                Debug.WriteLine(montoHorasAPagar);
                Debug.WriteLine(monto);
                Debug.WriteLine(reduccionIncapacidades);
                */
            }
            montoHorasExtra = contadorHoras * 1.5;
            //aumentoPorHorasExtra = montoSalario / 30 / 8 * montoHorasExtra;
            //Debug.WriteLine(aumentoPorHorasExtra);
        }
    }
}
catch (Exception e)
{
    string error = e.Message;
    respuesta = true;
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 64.**

*Código para hacer el cálculo de planilla, parte 7 (subtotales)*

```

double salarioBruto = 0;
salarioBruto = montoSalario + reduccionPermisosSinGoce + aumentoPorComisiones + aumentoPorHorasExtra - reduccionIncapacidades;
double deducciones = 0;
deducciones = salarioBruto / 100 * 10.67;
double salarioNeto = 0;
salarioNeto = salarioBruto - deducciones;

return salarioNeto;

```

*Fuente: Elaboración Propia (2024)*

**Figura 65.**

*Código para hacer el cálculo de horas laboradas para ingresar horas extra*

```

public bool MarcarSalida(MarcaEntradaSalidaModelo marca)
{
    bool respuesta;
    try
    {
        var conexion = new Conexion();
        string tiempo = marca.HoraEntrada.ToString("HH:mm:ss");

        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            _conexion.Open();
            //Calcular si aplica para hora extra
            TimeSpan diferencia = (TimeSpan)(marca.HoraSalida - marca.HoraEntrada);
            double horasRedondeadas = Math.Floor(diferencia.TotalHours);
            if (horasRedondeadas > 10)
            {
                int cantidadHorasExtra = (int)(horasRedondeadas - 9);
                HorasExtraDatos horasExtra = new HorasExtraDatos();
                HorasExtraModelo horasExtraModelo = new HorasExtraModelo();
                horasExtraModelo.IdUsuario = marca.IdUsuario;
                horasExtraModelo.FechaSolicitada = marca.HoraEntrada;
                horasExtraModelo.CantidadHoras = cantidadHorasExtra;
                horasExtraModelo.Estado = "Pendiente";
                horasExtra.GuardarPedidoHorasExtra(horasExtraModelo);
            }

            SqlCommand cmd = new SqlCommand("sp_marcarSalida_Guardar", _conexion);
            cmd.Parameters.AddWithValue("IdMarca", marca.IdMarca);
            cmd.Parameters.AddWithValue("HoraSalida", marca.HoraSalida);
            cmd.Parameters.AddWithValue("HorasTrabajo", marca.HorasTrabajo);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 66.**

*Código para hacer el cálculo de liquidación, parte 1 (Vacaciones y aguinaldo)*

```
public double CalcularLiquidacion(int IdUsuario, string TipoLiquidacion, DateTime FechaLiquidacion)
{
    var conexion = new Conexion();
    bool respuesta = false;

    UsuarioDatos usuarioDatos = new UsuarioDatos();
    SalarioDatos salarioDatos = new SalarioDatos();
    usuarioModelo usuario = usuarioDatos.ObtenerUsuario(IdUsuario);
    double montoLiquidacion = 0;

    //-----Calculo Vacaciones-----
    double montoVacaciones = 0;
    string queryVacaciones = "SELECT * FROM Usuario WHERE IdUsuario = @IdUsuario;";
    double montoSalarioPorDia = 0;
    try
    {
        double salarioBase = 0;
        int cantidadDiasVacaciones = 0;
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            _conexion.Open();
            SqlCommand cmd = new SqlCommand(queryVacaciones, _conexion);
            cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
            using (var dr = cmd.ExecuteReader())
            {
                while (dr.Read())
                {
                    salarioBase = Convert.ToDouble(dr["SalarioBase"]);
                    cantidadDiasVacaciones = Convert.ToInt32(dr["DiasVacaciones"]);
                    montoSalarioPorDia = salarioBase / 30;
                    montoVacaciones = montoSalarioPorDia * cantidadDiasVacaciones;
                }
            }
        }
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }

    //-----Calculo Aguinaldo-----
    double montoAguinaldo = 0;
    int Anho = FechaLiquidacion.Year;
    montoAguinaldo = salarioDatos.CalcularAguinaldoPorUsuario(Anho, IdUsuario);
}
```

*Fuente: Elaboración Propia (2024)*

**Figura 67.**

*Código para hacer el cálculo de liquidación, parte 2 (cálculo salario promedio)*

```
//-----Calcular salario meses previos para sacar salario promedio de cesantia y preaviso
double promedioSalario = 0;
double totalSalario = 0;
DateTime fechaInicioCalculo = FechaLiquidacion.AddMonths(-6);
int anho = FechaLiquidacion.Year;
int mesNumero = FechaLiquidacion.Month - 1;
string[] meses = { "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre" };

using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
{
    _conexion.Open();
    //variables auxiliares para cálculo
    int diasInicio = fechaInicioCalculo.Day;
    int dia1 = diasInicio;
    int dia2 = 0;

    // Calcular salario primera quincena parcial
    if (diasInicio > 0 && diasInicio <= 15)
    {
        dia2 = 15;
    }
    if (diasInicio > 15 && diasInicio < 31)
    {
        dia2 = 30;
    }

    double salarioParcialInicio = salarioDatos.CrearCalculoSalario(meses[fechaInicioCalculo.Month - 1], fechaInicioCalculo.Year, IdUsuario, false, dia1, dia2);
    totalSalario += salarioParcialInicio;

    // Calculo del salario de meses completos
    for (int mes = fechaInicioCalculo.Month + 1; mes < FechaLiquidacion.Month; mes++)
    {
        string mesString = meses[(mes - 1) % 12];
        string query = "SELECT Monto FROM CalculoSalario WHERE IdUsuario = @IdUsuario AND Anho = @Anho AND Mes = @Mes";

        using (var cmd = new SqlCommand(query, _conexion))
        {
            cmd.Parameters.AddWithValue("@IdUsuario", IdUsuario);
            cmd.Parameters.AddWithValue("@Anho", FechaLiquidacion.Year);
            cmd.Parameters.AddWithValue("@Mes", mesString);

            using (var dr = cmd.ExecuteReader())
            {
                while (dr.Read())
                {
                    double salario = Convert.ToDouble(dr["Monto"]);
                    totalSalario += salario;
                }
            }
        }
    }

    // Calcular salario segunda quincena parcial
    int diasFin = FechaLiquidacion.Day;
    if (diasInicio > 0 && diasInicio <= 15)
    {
        dia1 = 1;
    }
    if (diasInicio > 16 && diasInicio < 31)
    {
        dia1 = 16;
    }

    double salarioParcialFin = salarioDatos.CrearCalculoSalario(meses[FechaLiquidacion.Month - 1], FechaLiquidacion.Year, IdUsuario, false, dia1, diasFin);
    totalSalario += salarioParcialFin;
}
}
```

*Fuente: Elaboración Propia (2024)*

**Figura 68.**

*Código para hacer el cálculo de liquidación, parte 3 (cesantía)*

```
//-----Calcular Cesantía-----
double montoCesantia = 0;
DateTime fechaEntrada = usuario.FechaDeEntrada;
int cantidadMeses = 12 * ( FechaLiquidacion.Year - fechaEntrada.Year) + FechaLiquidacion.Month - fechaEntrada.Month;
if (FechaLiquidacion.Day < fechaEntrada.Day)
{
    cantidadMeses--;
}
double diasCalculo = 0;
double montoPorDia = usuario.SalarioBase / 30;
switch (cantidadMeses)
{
    case int n when (n < 2):
        diasCalculo = 0;
        break;
    case int n when (n >= 3 && n < 6):
        diasCalculo = 7;
        break;
    case int n when (n >= 6 && n < 12):
        diasCalculo = 14;
        break;
    case int n when (n >= 12 && n < 24):
        diasCalculo = 19.5;
        break;
    case int n when (n >= 24 && n < 36):
        diasCalculo = 20;
        break;
    case int n when (n >= 36 && n < 48):
        diasCalculo = 20.5;
        break;
    case int n when (n >= 48 && n < 60):
        diasCalculo = 21;
        break;
    case int n when (n >= 60 && n < 72):
        diasCalculo = 21.24;
        break;
    case int n when (n >= 72 && n >84):
        diasCalculo = 21.5;
        break;
    case int n when (n >= 84 && n < 96):
        diasCalculo = 22;
        break;
    case int n when (n >= 96 && n < 108):
        diasCalculo = 22;
        break;
    case int n when (n >= 108 && n < 120):
        diasCalculo = 22;
        break;
    case int n when (n >= 120 && n < 132):
        diasCalculo = 21.5;
        break;
    case int n when (n >= 132 && n < 144):
        diasCalculo = 21;
        break;
    case int n when (n >= 144 && n < 156):
        diasCalculo = 20.5;
        break;
    case int n when (n >= 156):
        diasCalculo = 20.5;
        break;
}

montoCesantia = promedioSalarioDiario * diasCalculo;
```

*Fuente: Elaboración Propia (2024)*

**Figura 69.**

*Código para hacer el cálculo de liquidación, parte 4 (preaviso y subtotales)*

```

//-----Calcular Preaviso-----
double montoPreaviso = 0;

double diasCalculoPreaviso = 0;
switch (cantidadMeses)
{
    case int n when (n <= 0 ):
        diasCalculoPreaviso = 0;

        break;
    case int n when (n > 3 && n <= 6):
        diasCalculoPreaviso = 7;
        break;
    case int n when (n > 6 && n <= 12):
        diasCalculoPreaviso = 14;
        break;
    case int n when (n > 12 ):
        diasCalculoPreaviso = 30;
        break;
}

montoPreaviso = promedioSalarioDiario * diasCalculoPreaviso;

//-----Calculo de subtotales-----

if (TipoLiquidacion == "Despido sin Responsabilidad Patronal" || TipoLiquidacion == "Renuncia") {
    montoLiquidacion = montoAguinaldo + montoVacaciones;
}
else {
    montoLiquidacion = montoAguinaldo + montoVacaciones + montoCesantia + montoPreaviso;
}

return montoLiquidacion;
}

1 reference
public bool GuardarLiquidacion(LiquidacionModelo liquidacion)
{
    bool respuesta;
    UsuarioDatos usuario = new UsuarioDatos();
    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            _conexion.Open();
            SqlCommand cmd = new SqlCommand("sp_liquidacion_Guardar", _conexion);
            //ya voy bien por aqui, pero debo cambiar el idusuario
            cmd.Parameters.AddWithValue("IdUsuario", liquidacion.IdUsuario);
            cmd.Parameters.AddWithValue("TipoLiquidacion", liquidacion.TipoLiquidacion);
            cmd.Parameters.AddWithValue("FechaLiquidacion", liquidacion.FechaLiquidacion);
            cmd.Parameters.AddWithValue("MontoLiquidacion", liquidacion.MontoPorLiquidacion);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 70.***Código para ingresar vacaciones*

```

public bool GuardarPedidoVacaciones(VacacionModelo vacacion, usuarioModelo usuario)
{
    bool respuesta;
    int IdUsuario = usuario.IdUsuario;
    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            string _estado = "Pendiente";
            int cantidadDias = 0;
            _conexion.Open();

            for (DateTime diaSemana = vacacion.FechaInicial; diaSemana <= vacacion.FechaFinal; diaSemana = diaSemana.AddDays(1))
            {
                string query = "Select EsDiaDeTrabajo from HorarioTrabajo where IdUsuario = @IdUsuario and DiaDeLaSemana = @DiaDeLaSemana";
                SqlCommand cmd2 = new SqlCommand(query, _conexion);
                string _diaDeLaSemana = diaSemana.DayOfWeek.ToString();
                switch (_diaDeLaSemana)
                {
                    case "Monday": _diaDeLaSemana = "Lunes"; break;
                    case "Tuesday": _diaDeLaSemana = "Martes"; break;
                    case "Wednesday": _diaDeLaSemana = "Miercoles"; break;
                    case "Thursday": _diaDeLaSemana = "Jueves"; break;
                    case "Friday": _diaDeLaSemana = "Viernes"; break;
                    case "Saturday": _diaDeLaSemana = "Sabado"; break;
                    case "Sunday": _diaDeLaSemana = "Domingo"; break;
                    default: throw new InvalidOperationException("error, dia no existe");
                }
                cmd2.Parameters.AddWithValue("IdUsuario", IdUsuario);
                cmd2.Parameters.AddWithValue("DiaDeLaSemana", _diaDeLaSemana);
                using (var dr = cmd2.ExecuteReader())
                {
                    if (dr.Read())
                    {
                        bool esDiaDeTrabajo = dr.GetBoolean(dr.GetOrdinal("EsDiaDeTrabajo"));
                        if (esDiaDeTrabajo && !EsDiaFeriado(diaSemana))
                        {
                            cantidadDias++;
                        }
                    }
                }
            }
            SqlCommand cmd = new SqlCommand("sp_pedidoVacacion_Guardar", _conexion);
            cmd.Parameters.AddWithValue("IdUsuario", IdUsuario);
            cmd.Parameters.AddWithValue("FechaInicial", vacacion.FechaInicial);
            cmd.Parameters.AddWithValue("FechaFinal", vacacion.FechaFinal);
            cmd.Parameters.AddWithValue("Estado", _estado);
            cmd.Parameters.AddWithValue("Dias", cantidadDias);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}

```

*Fuente: Elaboración Propia (2024)*

Figura 71.

Código para solicitud de permisos

```

public bool GuardarPermiso(permisoModelo permiso, usuarioModelo usuario)
{
    bool respuesta;
    int IdUsuario = usuario.IdUsuario;
    NotificacionDatos notificacion = new NotificacionDatos();
    string tipoPedido = "Permiso";
    int cantidadDias = 0;
    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionsSQL()))
        {
            _conexion.Open();

            // Revisión de días
            for (DateTime diaSemana = permiso.FechaInicial; diaSemana <= permiso.FechaFinal; diaSemana = diaSemana.AddDays(1))
            {
                // Revisión de horario de trabajo
                string query = "Select EsDiaDeTrabajo from HorarioTrabajo where IdUsuario = @IdUsuario and DiaDeLaSemana = @DiaDeLaSemana";
                SqlCommand cmd2 = new SqlCommand(query, _conexion);
                string _diaDeLaSemana = diaSemana.DayOfWeek.ToString();
                switch (_diaDeLaSemana)
                {
                    case "Monday": _diaDeLaSemana = "Lunes"; break;
                    case "Tuesday": _diaDeLaSemana = "Martes"; break;
                    case "Wednesday": _diaDeLaSemana = "Miercoles"; break;
                    case "Thursday": _diaDeLaSemana = "Jueves"; break;
                    case "Friday": _diaDeLaSemana = "Viernes"; break;
                    case "Saturday": _diaDeLaSemana = "Sabado"; break;
                    case "Sunday": _diaDeLaSemana = "Domingo"; break;
                    default: throw new InvalidOperationException("error, dia no existe");
                }
                cmd2.Parameters.AddWithValue("IdUsuario", IdUsuario);
                cmd2.Parameters.AddWithValue("DiaDeLaSemana", _diaDeLaSemana);
                using (var dr = cmd2.ExecuteReader())
                {
                    if (dr.Read())
                    {
                        bool esDiaDeTrabajo = dr.GetBoolean(dr.GetOrdinal("EsDiaDeTrabajo"));
                        if (esDiaDeTrabajo && !EsDiaFeriado(diaSemana))
                        {
                            cantidadDias++;
                        }
                    }
                }
            }

            // Guardar Permiso
            string _estado = "Pendiente";
            DateTime FechaDeSolicitud = DateTime.Today;
            SqlCommand cmd = new SqlCommand("sp_permiso_Guardar", _conexion);
            cmd.Parameters.AddWithValue("IdUsuario", IdUsuario);
            cmd.Parameters.AddWithValue("TipoPermiso", permiso.TipoPermiso);
            cmd.Parameters.AddWithValue("FechaDeSolicitud", FechaDeSolicitud);
            cmd.Parameters.AddWithValue("FechaInicial", permiso.FechaInicial);
            cmd.Parameters.AddWithValue("FechaFinal", permiso.FechaFinal);
            cmd.Parameters.AddWithValue("CantidadDias", cantidadDias);
            cmd.Parameters.AddWithValue("Estado", _estado);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}

```

Fuente: Elaboración Propia (2024)

**Figura 72.***Código para ingresar incapacidad*

```

public bool GuardarIncapacidad(IncapacidadModelo incapacidad, string nombre)
{
    bool respuesta;
    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            string _estado = "Pendiente";
            int cantidadDias = 0;
            int IdUsuario2 = ObtenerUsuarioPorNombre(nombre);
            _conexion.Open();
            for (DateTime diaSemana = incapacidad.FechaInicial;
                diaSemana <= incapacidad.FechaFinal;
                diaSemana = diaSemana.AddDays(1))
            {
                string query = "Select EsDiaDeTrabajo from HorarioTrabajo where IdUsuario = @IdUsuario and DiaDeLaSemana = @DiaDeLaSemana";
                SqlCommand cmd2 = new SqlCommand(query, _conexion);
                string _diaDeLaSemana = diaSemana.DayOfWeek.ToString();
                DayOfWeek _diaDeLaSemanaAux;

                switch (_diaDeLaSemana)
                {
                    case "Monday":
                        _diaDeLaSemana = "Lunes";
                        break;
                    case "Tuesday":
                        _diaDeLaSemana = "Martes";
                        break;
                    case "Wednesday":
                        _diaDeLaSemana = "Miercoles";
                        break;
                    case "Thursday":
                        _diaDeLaSemana = "Jueves";
                        break;
                    case "Friday":
                        _diaDeLaSemana = "Viernes";
                        break;
                    case "Saturday":
                        _diaDeLaSemana = "Sabado";
                        break;
                    case "Sunday":
                        _diaDeLaSemana = "Domingo";
                        break;
                    default:
                        throw new InvalidOperationException("error, dia no existe");
                }
                cmd2.Parameters.AddWithValue("IdUsuario", IdUsuario2);
                cmd2.Parameters.AddWithValue("DiaDeLaSemana", _diaDeLaSemana);
                using (var dr = cmd2.ExecuteReader())
                {
                    if (dr.Read())
                    {
                        bool esDiaDeTrabajo = dr.GetBoolean(dr.GetOrdinal("EsDiaDeTrabajo"));
                        if (esDiaDeTrabajo)
                        {
                            cantidadDias++;
                        }
                    }
                }
            }
            int IdUsuario = ObtenerUsuarioPorNombre(nombre);
            SqlCommand cmd = new SqlCommand("sp_incapacidad_Guardar", _conexion);
            cmd.Parameters.AddWithValue("IdUsuario", IdUsuario);
            cmd.Parameters.AddWithValue("TipoIncapacidad", incapacidad.TipoIncapacidad);
            cmd.Parameters.AddWithValue("FechaInicial", incapacidad.FechaInicial);
            cmd.Parameters.AddWithValue("FechaFinal", incapacidad.FechaFinal);
            cmd.Parameters.AddWithValue("Estado", _estado);
            cmd.Parameters.AddWithValue("CantidadDias", cantidadDias);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
    }
}

```

*Fuente: Elaboración Propia (2024)*

**Figura 73.**

*Código para ingresar comisiones*

```
public bool GuardarComision(ComisionModelo comision, string nombre)
{
    bool respuesta;
    try
    {
        var conexion = new Conexion();
        using (var _conexion = new SqlConnection(conexion.getConexionSQL()))
        {
            string _estado = "Pendiente";
            int IdUsuario = ObtenerUsuarioPorNombre(nombre);
            _conexion.Open();
            SqlCommand cmd = new SqlCommand("sp_comision_Guardar", _conexion);
            cmd.Parameters.AddWithValue("IdUsuario", IdUsuario);
            cmd.Parameters.AddWithValue("Fecha", comision.Fecha);
            cmd.Parameters.AddWithValue("Monto", comision.Monto);
            cmd.Parameters.AddWithValue("IdFactura", comision.IdFactura);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.ExecuteNonQuery();
        }
        respuesta = true;
    }
    catch (Exception e)
    {
        string error = e.Message;
        respuesta = false;
    }
    return respuesta;
}
```

*Fuente: Elaboración Propia (2024)*

## Pruebas

En esta sección se describirán las pruebas realizadas para validar el funcionamiento de los módulos de la aplicación por medio de la confirmación de comportamiento de las entradas y salidas de la aplicación.

### Prueba de autenticación.

Con esta prueba se valida que tanto el campo de usuario como de contraseña deben contener información, además, estos datos deben compararse contra la base de datos y su tabla de usuario para revisar si la contraseña y usuario coinciden.

**Tabla 39**

*Prueba de autenticación*

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 01	<b>Nombre del caso de prueba:</b>	Autenticar
<b>Descripción</b>	Prueba para validar la autenticación		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	15/10/2024
<b>Módulo</b>	Inicio de sesión		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
1	Llenar formulario con datos y enviarlo	El usuario y contraseña se envían	Prueba exitosa

2	Validación de credenciales	El sistema debe validar que el usuario existe, que la contraseña existe y que son para el mismo usuario	Prueba exitosa
3	Ingresar a la aplicación	El sistema redirecciona a la pantalla de permisos	Prueba exitosa
4	Cierre de sesión al presionar en el botón "salir"	El usuario presiona el botón "salir" y debe redireccionarse a la página de inicio de sesión, además si el usuario ingresa una URL de la aplicación manualmente, esta no debe redireccionarlo	Prueba exitosa

*Fuente:* Elaboración Propia (2024)

### **Prueba cálculo de nómina.**

Con esta prueba se valida que el cálculo de salario se hace con base en un salario base para cada usuario, además de realizarse en un rango predeterminando que será por quincena. Para ese rango de tiempo se debe revisar para cada usuario si existen horas extras aprobadas o comisiones que aumenten el salario, así como permisos sin goce salarial e incapacidades que disminuyen al salario, y que dependiendo del tipo de incapacidad y cantidad de días el cálculo variará.

**Tabla 40.***Prueba Cálculo de nómina*

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 02	<b>Nombre del caso de prueba:</b>	Calcular Planilla
<b>Descripción</b>	Prueba para validar el cálculo de salarios		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	17/10/2024
<b>Módulo</b>	Cálculo de planilla		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
1	Seleccionar el año, mes y quincena para el que se quiere calcular la planilla	El sistema realizara el cálculo para la fecha seleccionada	Prueba exitosa
2	Cálculo solo para usuarios activos	El sistema debe realizar el cálculo solamente para los usuarios que se encuentran activos.	Prueba exitosa
3	Validación de que se tomen en cuenta las horas extras, comisiones, incapacidades y permisos sin goce salarial.	El sistema debe mostrar subtotales para cada rubro durante la ejecución del cálculo e imprimirlos en consola.	Prueba exitosa

*Fuente: Elaboración Propia (2024)*

### Prueba de cálculo de aguinaldo.

Con esta prueba se validará que se tomen en cuenta los salarios para cada usuario del mes de diciembre del año anterior hasta el fin de noviembre del año actual.

**Tabla 41**

*Prueba del cálculo de aguinaldo*

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 03	<b>Nombre del caso de prueba:</b>	Calcular Aguinaldo
<b>Descripción</b>	Prueba para validar el cálculo de salarios		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	19/10/2024
<b>Módulo</b>	Cálculo de aguinaldo		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
1	Calcular el aguinaldo solo para los usuarios activos	El sistema realizará el cálculo de aguinaldo solo para los usuarios activos	Prueba exitosa
2	Calcular el aguinaldo en el rango de fechas correctas	El sistema debe mostrar en consola el monto para cada mes desde el 01 de diciembre del año anterior hasta el 30 de noviembre del presente año	Prueba exitosa
3	Guardar cálculo de aguinaldo	El sistema guardará en base de datos el monto de aguinaldo para cada usuario y deberá mostrar en pantalla la lista de usuario y el monto	Prueba exitosa

*Fuente:* Elaboración Propia (2024)

### **Prueba de creación de pedido vacaciones y aprobación.**

Con esta prueba se debe validar que la fecha inicial sea anterior a la fecha final de los días que se piden vacaciones. Además, el rango de fechas no puede incluir otros días donde se haya pedido vacaciones o permisos de ausencia. Se debe calcular la cantidad de días de vacaciones a pedir, excluyendo días donde el usuario no trabaja, además de feriados. Por último, el usuario debe tener una cantidad de días de vacaciones igual o mayor que la cantidad de días solicitadas. Cuando se cree el pedido de vacación entonces para el usuario aprobador cuando inicie sesión se le deberá mostrar una alerta de que tiene pedidos pendientes. Una vez que se apruebe o rechace, entonces esto deberá crear una notificación que mostrará una alerta para el usuario que solicitó las vacaciones cuando este inicie sesión. Si la vacación fue aprobada entonces debe mostrarse para el personal de recursos humanos.

**Tabla 42**

*Prueba de pedido de vacaciones*

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 04	<b>Nombre del caso de prueba:</b>	Gestión Vacaciones
<b>Descripción</b>	Prueba para validar creación y aprobación/rechazo de un pedido de vacaciones		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	22/10/2024
<b>Módulo</b>	Gestión de pedido vacaciones		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>

1	El usuario llena el pedido de vacaciones	El sistema debe revisar que contenga una fecha inicial y otra final. La fecha inicial debe ser anterior a la fecha final.	Prueba exitosa
2	Revisar si existen vacaciones o permisos para ese usuario en esas fechas	El sistema revisará si ya este usuario pidió vacaciones o permiso de ausencia para esa fecha. Si es así, no permitirá la creación del pedido de vacaciones.	Prueba exitosa
3	Calcular cantidad de días	El sistema calculará en el rango de fechas seleccionado, si existen días donde el usuario no trabaja, además si existen días feriados, en ambos casos no se debe sumar esos días a la cantidad de días solicitados	Prueba exitosa
4	Validar cantidad de días solicitados contra los días de vacaciones	El sistema revisará si el usuario tiene la suficiente cantidad de días de vacaciones para poder realizar el pedido.	Prueba exitosa
5	Mostrar notificación a usuario aprobador	Cuando el usuario aprobador inicie sesión entonces a los 5 minutos se le debe mostrar una alerta que tiene pedidos pendientes, y está alerta seguirá apareciendo cada 5 minutos hasta que se apruebe o rechace el pedido	Prueba exitosa
6	Mostrar notificación al usuario de cambio de estado de pedido de vacaciones	El sistema mostrará al usuario que solicitó las vacaciones una notificación cuando el pedido cambie a estado aprobado o rechazado	Prueba exitosa

7	Mostrar notificación a recursos humanos	El sistema mostrará una notificación a los usuarios con rol de RRHH cuando un pedido de vacaciones fue aprobado	Prueba exitosa
---	---	---	----------------

*Fuente:* Elaboración Propia (2024)

### Prueba de creación de permisos de ausencias.

Con esta prueba se debe validar que la fecha inicial sea anterior a la fecha final de los días que se piden con permisos de ausencia. Además, el rango de fechas no puede incluir otros días donde se haya pedido vacaciones o permisos de ausencia. Cuando se cree el pedido de permiso de ausencia entonces para el usuario aprobador cuando inicie sesión se le deberá mostrar una alerta de que tiene pedidos pendientes. Una vez que se apruebe o rechace, entonces esto deberá crear una notificación que mostrará una alerta para el usuario que solicitó las vacaciones cuando este inicie sesión. Si la vacación fue aprobada entonces debe mostrarse para el personal de recursos humanos.

**Tabla 43**

*Prueba de gestión de permisos de ausencias*

<b>Número del caso de prueba:</b>	Prueba 05	<b>Nombre del caso de prueba:</b>	Gestión de permisos
<b>Descripción</b>	Prueba para validar creación y aprobación/rechazo de un permiso de ausencia		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	23/10/2024
<b>Módulo</b>	Gestión de permiso de ausencia		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
1	El usuario llena el pedido de vacaciones	El sistema debe revisar que contenga una fecha inicial y otra final. La fecha inicial debe ser anterior a la fecha final.	Prueba exitosa

2	Revisar si existen vacaciones o permisos para ese usuario en esas fechas	El sistema revisará si ya este usuario pidió vacaciones o permiso de ausencia para esa fecha. Si es así, no permitirá la creación del pedido de vacaciones.	Prueba exitosa
3	Mostrar notificación a usuario aprobador	Cuando el usuario aprobador inicie sesión entonces a los 5 minutos se le debe mostrar una alerta que tiene pedidos pendientes, y está alerta seguirá apareciendo cada 5 minutos hasta que se apruebe o rechace el pedido	Prueba exitosa
4	Mostrar notificación al usuario de cambio de estado de pedido de vacaciones	El sistema mostrará al usuario que solicitó las vacaciones una notificación cuando el pedido cambie a estado aprobado o rechazado	Prueba exitosa
5	Mostrar notificación a recursos humanos	El sistema mostrará una notificación a los usuarios con rol de RRHH cuando un pedido de vacaciones fue aprobado	Prueba exitosa

### Prueba de comisión.

En este módulo se validará que la comisión que se registrará cuenta con todos los datos necesarios, además que debe de mostrar una notificación para el usuario al que se le registro la comisión.

**Tabla 44**

*Prueba de creación de comisión*

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 06	<b>Nombre del caso de prueba:</b>	Gestión de comisiones
<b>Descripción</b>	Prueba para validar creación de comisiones		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	23/10/2024
<b>Módulo</b>	Comisiones		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
<b>1</b>	El usuario supervisor o administrador llena el formulario de comisión	El sistema debe verificar que todos los campos requeridos son llenados, sino debe mostrar un error.	Prueba exitosa
<b>2</b>	Notificación de usuario	El sistema creará una notificación para el usuario al que se la ha añadido una comisión.	Prueba exitosa
<b>3</b>	Mostrar en lista de comisiones	El usuario al que se le creó la comisión puede verla en la lista de comisiones	Prueba exitosa

*Fuente:* Elaboración Propia (2024)

### Prueba de creación de registro de horas extra.

En este módulo se calculará la cantidad de horas laboradas para un usuario, si esas horas sobrepasan las 9 horas desde que se marca la entrada y la salida entonces se debe generar un pedido de horas extra automáticamente, para el que los usuarios aprobadores recibirán una notificación

**Tabla 45**

*Prueba de registro de horas extras*

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 07	<b>Nombre del caso de prueba:</b>	Gestión de horas extra
<b>Descripción</b>	Prueba para validar la cantidad de horas laboradas y si es igual o mayor a 10 horas crear un pedido de horas extra		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	25/10/2024
<b>Módulo</b>	Horas Extra		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
1	El usuario debe marcar hora de entrada	El sistema debe mostrar solamente la posibilidad de marcar hora de entrada.	Prueba exitosa
2	El usuario debe marcar hora de salida	El sistema mostrará el botón de marcar salida solo si este usuario ya marcó hora de entrada para ese día	Prueba exitosa
3	Cálculo de cantidad de horas	Si la cantidad de horas resultantes del cálculo de hora de entrada y hora de salida es igual o mayor a 10 entonces se debe crear un pedido de horas extra.	Prueba exitosa

4	Mostrar notificación a usuario aprobador	El sistema mostrará una notificación al usuario aprobador cada 5 minutos mientras hayan pedido de horas extra pendientes	Prueba exitosa
---	--	--	----------------

*Fuente:* Elaboración Propia (2024)

### Prueba para creación de incapacidades.

En esta prueba se valida que la creación de incapacidad cuente con la información correcto, también se debe generar una notificación a recursos humanos.

**Tabla 46**

*Prueba para creación de incapacidad*

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 08	<b>Nombre del caso de prueba:</b>	Gestión incapacidades
<b>Descripción</b>	Prueba para validar creación de incapacidad		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	27/10/2024
<b>Módulo</b>	Incapacidades		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
1	El usuario supervisor deberá crear el registro de incapacidad	El sistema debe revisar que todos los campos han sido llenados correctamente	Prueba exitosa
2	Notificación a recursos humanos	Se debe mostrar una alerta a recursos humanos de la nueva incapacidad, y deberá mostrarse hasta que sea aprobada	Prueba exitosa

### Prueba para liquidación de usuario.

En esta prueba se calculará el pago que debe realizarse al usuario liquidado tomando en cuenta sus vacaciones y salario de la quincena que se liquida, además si es despido con responsabilidad patronal deberá sumarse el monto de pago por cesantía y preaviso.

Tabla 47

## Prueba de liquidación

<b>Prototipo Funcional para la Gestión del Recurso Humano para Rastreadores Mundiales</b>			
<b>Número del caso de prueba:</b>	Prueba 09	<b>Nombre del caso de prueba:</b>	Gestión Liquidación
<b>Descripción</b>	Prueba para validar cálculo de liquidación		
<b>Autor</b>	Randall Fernández Meza	<b>Fecha de prueba</b>	30/10/2024
<b>Módulo</b>	Liquidaciones		
<b>Actividades de prueba</b>			
<b>No.</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado actual</b>
1	El usuario de RRHH selecciona el usuario que se debe liquidar	El sistema realiza un cálculo inicial de la liquidación del usuario tomando en cuenta los días de vacaciones pendientes y el salario no devengado de la quincena actual	Prueba exitosa
2	Validación tipo despido o renuncia	Si es un despido con responsabilidad patronal entonces el sistema debe calcular el monto a pagar por preaviso y liquidación	Prueba exitosa
3	Guardar liquidación	El sistema guarda la información de liquidación en base de datos, también modifica la tabla bitácora para mostrar una fecha de salida. Al usuario le quedará en 0 los días de vacaciones y su Estado pasará a inactivo	Prueba exitosa

4	Usuario no disponible	El usuario no está disponible para ninguna gestión, con la excepción de la reactivación del usuario	Prueba exitosa
---	-----------------------	---	----------------

*Fuente:* Elaboración Propia (2024)

## CAPÍTULO VI, CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

Se analizó la situación actual del proceso de gestión de nómina para la empresa Rastreadores Mundiales, concluyendo que en la operativa actual existen problemas asociados a la cantidad de tiempo de trabajo que se consume en el cálculo de planilla y de aguinaldo. Además, el proceso actual implica cálculos manuales de salario lo que puede resultar en errores de salarios o aguinaldos a pagar.

Se diseñó un prototipo funcional mediante la evaluación de los requerimientos de Rastreadores Mundiales, específicamente para tareas como cálculos de salario, de liquidación y de aguinaldo. También para el procesamiento de vacaciones, permisos de ausencia, aprobación de horas extra y registro de comisiones.

Se programó en ASP Core MVC utilizando el modelo vista-controlador, y los lenguajes de C# para el código de servidor, Javascript y Razor para el código de cliente y con HTML y CSS para las vistas de usuario.

Se realizaron pruebas exhaustivas en la aplicación para poder validar la integridad de los datos y el correcto funcionamiento del procesamiento de datos.

Se desarrolló un prototipo funcional para Rastreadores Mundiales, basándose en el análisis de la situación actual, el diseño y programación del prototipo, así como las pruebas de su correcto funcionamiento.

## Recomendaciones

Es fundamental que la transición no sea abrupta, sino que se desdoble desde que el prototipo entre a producción, se recomienda que por un lapso se siga realizando los mismos procesos actuales que ejecuta el personal de recursos humanos en paralelo, esto para asegurar que los resultados con la metodología anterior y con el prototipo son los mismos. Los responsables de la ejecución de esta recomendación es el personal de recursos humanos, y la fecha de inicio de la propuesta es desde el 6 de enero de 2025, con una duración de 3 meses.

Se recomienda capacitar al personal para el uso adecuado de la aplicación. Con recursos humanos se estima que se puede explicar el funcionamiento en un plazo de dos semanas y con el resto del personal en una semana. Si después de este tiempo existen dudas sobre el uso de la aplicación, la comunicación queda abierta para apoyarlos a resolver las preguntas que tengan. Los responsables de la ejecución de esta recomendación son todo el personal de la empresa y mi persona, empezando el 6 de enero de 2025, para el personal de recursos humanos la duración es de dos semanas y el resto del personal una semana.

La evaluación constante sobre la facilidad de uso, eficiencia y constancia de resultados de los diferentes módulos es recomendada para así poder identificar posibles áreas de mejora y asegurar que el prototipo cumple con los objetivos para el que fue desarrollado. Esto también para considerar la posibilidad de mejoras en el futuro, ya sea por modificación de módulos existentes o con la posibilidad de expandir las funcionalidades como reportería, entre otros. Los responsables de esta tarea son el administrador de la empresa y recursos humanos, la fecha de inicio de la propuesta es desde el 6 de enero, con una duración de 6 meses.

Por último, aunque Rastreadores Mundiales prefiera el uso de la nube para albergar el prototipo, se recomienda mantener una visión adaptativa en función de optimizar el uso de recursos en cuanto a tipo de licencias y tipo de servidor. En caso de que la empresa esté buscando desarrollar aplicaciones para resolver otras necesidades, tendría sentido considerar otro tipo de licenciamiento de base de datos, así como evaluar las posibilidades de seguridad para la aplicación. El responsable de esta tarea es el administrador, la fecha de inicio propuesta es el 6 de enero, con una duración de 6 meses.

## Referencias

- Agudelo, M., Alveiro, C., Saavedra B., Ramiro M. (2016). El Recurso Humano Como Elemento Fundamental Para La Gestión de Calidad y la Competitividad Organizacional. *Visión de Futuro* vol. 20, num 2
- Alfaro, J. (1 de junio de 2024). Estas son las deducciones y cargas salariales en Costa Rica: ¿son muchas o pocas?. *El Financiero*.
- Álvarez, M., Álvarez, S., Castro, J., Peña, J., Rousset, D. y Vargas, J. (s.f.) *Manual de CSS3*. <http://www.desarrolloweb.com/manuales/css3.htm>
- Arias, J., Cáceres, M. y Covinos, M. (2020). Formulación de los objetivos específicos desde el alcance correlacional en trabajos de investigación. *Ciencia Latina Revista Multidisciplinar*. Vol. 4. N. 2. [https://doi.org/10.37811/cl\\_rcm.v4i2.73](https://doi.org/10.37811/cl_rcm.v4i2.73)
- Ávila, J., Becerril A., Camacho, N., Leo, G., Vega, A. y Vega, G. (2014) Paradigmas en la Investigación. Enfoque Cuantitativo y Cualitativo. *European Scientific Journal*. Vol. 10, No. 15.
- Azuero, A. (2018) *Significatividad del marco metodológico en el desarrollo de proyectos de investigación*. Universidad Católica de Cuenca, Ecuador. <http://dx.doi.org/10.35381/r.k.v4i8.274>
- Bergero, F., Meza, E., Ruiz, E., Wolf, G. (2014) *Sistemas Operativos*. Iniciativa Latinoamericana de Libros de Texto Abierto.
- Bilbao, J. y Escobar, P. (2020). *Investigación y Educación Superior*. Estados Unidos, Lulu.
- Bustamante, Castillo R., Loaiza R., Martel C., Medina M. y Rojas R. (2023) *Metodología de Investigación. Técnicas e Instrumentos de Investigación*. DOI <https://doi.org/10.35622/inudi.b.080>

- Carballos, M. y Guelmes, E. (2016). Algunas Consideraciones Acerca de las Variables en las Investigaciones que se Desarrollan en Educación. *Revista Universidad y Sociedad* [seriada en línea], 8 (1). pp.140-150. Recuperado de <http://rus.ucf.edu.cu>
- Carvajal, A. (2020). *Gestión actual del talento humano: contexto universitario*. Universidad Estatal a Distancia, San José, Costa Rica. 10.22458
- Código Penal. *Ley 4573 de 1973. (Costa Rica)*.
- Código de Trabajo de Costa Rica (1943). *Ley 2 de 1943. 27 de Agosto de 1943*
- Córdoba. (2023). *Modelo de Desarrollo Cascada – Qué es, Cómo Funciona y Más. Aprende Informáticas*  
<https://aprendeinformaticas.com/modelo-de-desarrollo-cascada/>
- Gómez, A., Honores, J., Valarezo, M. y Vines, L. (2018) *Comparación de Tendencias Tecnológicas en Aplicaciones Web*
- González, A. (2015) *Entorno de Desarrollo Integrado para el lenguaje*
- Gutiérrez, J., Marcillo, J., Marcillo, K., Pisco, A., Quimis, O. y Regalado, J. (2017) *Fundamentos Sobre la Gestión de Base de Datos. Área e Innovación y Desarrollo*
- Hadida, S. y Trolio, F. (2020). *La Agilidad en las Organizaciones: Trabajo Comparativo Entre Metodologías Ágiles y de Cascada en un Contexto de Ambigüedad y Transformación Digital*. Streb UCEMA, Argentina
- Henríquez, E. y Zepeda, M. (2003) Preparación e un Proyecto de Investigación. *Ciencia y Enfermería IX* (2): 23-28, I.S.S.N. 0717 - 2079
- Hernández, R. y Mendoza, C. (2018). *Metodología de la Investigación*. México: McGraw Hill.

Lerma, R., Murcia, J., Mifsud, E. (2013) *Aplicaciones Web*. México: McGraw Hill.

Luján, S. (2002) *Programación de Aplicaciones Web. Historia, Principios Básicos y Clientes Web*. Editorial Club Universitario.

Microsoft (2023) *Descripción del programa de instalación en IIS 7*. <https://learn.microsoft.com/es-es/iis/install/installing-iis-7/understanding-setup-in-iis>

Microsoft (2024) *¿Qué es ASP.NET?* <https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet>

Ministerio de Trabajo de Costa Rica (2024). *Lista de salarios mínimos del sector privado* <https://www.mtss.go.cr/temas-laborales/salarios/lista-salarios.html>

Ministerio de Trabajo de Costa Rica (2024). *Temas Laborales* <https://www.mtss.go.cr/temas-laborales/>

Montes, W. (2015) *ASP.NET MVC 6 - Una Guía Introductoria*

Morales, M. (2008). El Salario y la Previsión Social entre el Derecho Social y el Fiscal. *Revista Latinoamericana de Derecho Social*, ISSN-e 1870-4670, N°. 7, 2008

Rivas, C., Corona, V., Gutiérrez, J. y Hernández, L. (2015). Metodologías actuales de desarrollo de software. *Revista Tecnología e Innovación*. Vol.2 No.5 980-986

Sáenz, K. y Támez, G. (2014) *Métodos y Técnicas Cualitativas y Cuantitativas Aplicables a la Investigación en Ciencias Sociales*. México, Tirant Humanidades

Suárez, E. (2008) *¿Qué es una base de datos relacional?* Fundación Universitaria Cafam, Colombia

Tamayo, M. (1980) *Metodología Formal de la Investigación Científica*. Universidad de Santo Tomás, Colombia.

Torres, M. (s.f.) *Métodos de Recolección de Datos para una Investigación*. Universidad Rafael Landívar, Guatemala.

## Apéndices

### Apéndice 1.

#### GUÍA DE OBSERVACIÓN

**Nombre de la Empresa:** Rastreadores Mundiales

**Actividad de la Empresa:** Venta e Instalación de Sistemas de Geolocalización para Vehículos

**OBJETIVO:** Observar y evaluar las actividades relacionadas con el proyecto de Prototipo Funcional para Gestión de Nómina.

No	Aspectos por observar	Cumple	No Cumple	Oportunidad de mejora	Detalle de Observación
1	¿Qué herramienta se utiliza para la gestión de nómina?				
2	¿Están todos los datos sobre la gestión de nómina centralizados en un solo sistema?				
3	¿Hay una documentación confiable de horas extras?				

4	¿Cuántos salarios se pueden calcular en una hora?				
5	¿Se cometieron errores durante el cálculo de nómina?				
6	¿Hubo confusiones con las vacaciones de los colaboradores?				

## Apéndice 2

### GUÍA DE ENTREVISTA

**Organización:** Rastreadores Mundiales

**Nombre del Entrevistado:** Erick Mora

**Cargo:** Gerente

#### **Preguntas:**

Pregunta 1: ¿Utiliza una sola herramienta para la gestión de nómina como cálculos de salarios, aguinaldos, liquidaciones?

Pregunta 2 ¿Se comenten errores frecuentes en el cálculo de salario?

Pregunta 3 ¿Se han presentando errores en el cálculo de horas extras?

Pregunta 4. ¿Cuánto tiempo cree que se toma el cálculo de salarios?

Pregunta 5 ¿Tienen sistemas informáticos, de qué tipo de sistemas son y dónde los tienen?

Pregunta 6. ¿Tiene alguna preferencia sobre la nube o infraestructura física?

Pregunta 7. ¿Es viable utilizar un sistema PaaS en donde el mantenimiento del sistema operativo lo hace el proveedor de la nube, pero por un costo mayor?

Pregunta 8 ¿Estaría de acuerdo que los colaboradores usen su propio teléfono para gestionar sus operaciones como pedir vacaciones?

Pregunta 9. ¿Cuáles son las operaciones de nómina que le interesan para una aplicación?