

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

**DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DEL PAGO DE
PLANILLA DEL GIMNASIO TITANIUM FITNESS, UBICADO EN ALAJUELA**

**MODALIDAD PROYECTO PARA OPTAR POR EL GRADO DE BACHILLERATO
EN INGENIERÍA INFORMÁTICA**

MAURO SAÚL PÉREZ CASTRO

**Daniel Álvarez Garro
TUTOR**

Sede Central

Abril, 2026

TABLA DE CONTENIDO

Dedicatoria	7
Agradecimientos	7
Carta del Tribunal Examinador	8
Carta de aprobación del tutor	9
Declaración jurada del estudiante	16
Carta de solicitud de defensa	17
Carta de revisión filológica	18
Tablas	19
CAPÍTULO I: INTRODUCCIÓN	20
Planteamiento del problema	20
<i>Errores y retrasos en el cálculo de la planilla</i>	21
<i>Falta de control sobre las horas extra trabajadas</i>	21
<i>Ausencia de un sistema para la solicitud y aprobación de permisos</i>	21
<i>Vacaciones e incapacidades sin control formal</i>	21
<i>Cálculo manual de aguinaldos, deducciones y liquidaciones</i>	21
<i>Registro manual de marcas de entrada y salida</i>	21
Objetivo general	22
Objetivos específicos	22
Justificación	22
Viabilidad Técnica de la Investigación	22
Requerimiento del sistema operativo	23
Requerimientos de <i>software</i>	23
Herramientas ofimáticas	23
Requerimientos de seguridad	24
Requerimientos de <i>hardware</i>	24

<i>Viabilidad Operativa de la Investigación</i>	24
<i>Viabilidad Económica</i>	25
<i>Viabilidad Legal</i>	26
Proyecciones	27
Alcance	27
<i>Alcance funcional</i>	28
Alcance metodológico	30
<i>Ciclo de vida</i>	30
Alcance tecnológico	31
CAPÍTULO II: MARCO REFERENCIAL	33
El <i>Software</i> y el Prototipado	34
Los Sistemas Web	35
Los Usuarios del Sistema	36
Módulo de <i>Software</i>	37
Metodología de Desarrollo: Kanban	38
Herramientas de Desarrollo	39
Entorno de Desarrollo Integrado (IDE): Visual Studio Code	40
Entorno de Desarrollo Local: XAMPP	40
Tecnologías del Lado del Cliente: HTML y CSS	41
Tecnologías del Lado del Servidor	41
<i>PHP</i>	41
<i>MySQL</i>	42
Gestión de Recursos Humanos y Planilla	43
Planilla de Sueldos	44
Permisos Laborales	45
Aguinaldo	46
Liquidación Laboral	46

Evaluación del Desempeño.....	47
Control de Asistencia	49
CAPÍTULO III: MARCO METODOLÓGICO	50
Enfoques de investigación	50
<i>Enfoque cuantitativo</i>	50
<i>Enfoque cualitativo</i>	50
<i>Enfoque mixto</i>	51
Enfoque de investigación seleccionado	51
Tipos de investigación	52
<i>Investigación exploratoria</i>	52
<i>Investigación descriptiva</i>	52
<i>Investigación aplicada (o tecnológica)</i>	52
Tipo de investigación seleccionado	53
Fuentes de información.....	53
Fuentes de información primaria	54
Fuentes de información secundarias	54
Fuentes de Información Terciaria	55
Variables o Unidades de Análisis.....	56
Variable Conceptual.....	56
Variable Operacional.....	57
Instrumentos para la Recolección de Datos	60
Proceso para la Recolección y Análisis de Datos	61
<i>Entrevista semiestructurada</i>	61
<i>Observación directa</i>	62
CAPÍTULO IV: ANÁLISIS DE RESULTADOS	63
Método de la Entrevista	63
Método de observación.....	74

Identificación de riesgos en el procedimiento manual.....	86
Conclusión	87
CAPÍTULO V: PROPUESTA.....	88
ANÁLISIS:	88
<i>Análisis detallado del Software por desarrollar:</i>	88
Gestión de horas extra:.....	88
Gestionar nómina:.....	88
Gestión de vacaciones:.....	88
Gestionar incapacidades:	89
Gestionar aguinaldos:.....	89
Gestionar deducciones:	89
Gestionar impuesto de renta:	89
Gestionar Asistencia:	89
Gestión de impuesto de CCSS:	89
Gestión de permisos:.....	90
Gestión de liquidaciones:.....	90
<i>Análisis detallado del hardware requerido, tanto para la programación del prototipo como para poner en producción el sistema desarrollado.....</i>	90
<i>Análisis detallado de los elementos relacionados con las telecomunicaciones que requiere el sistema para su funcionamiento:</i>	92
<i>Descripción detallada de las herramientas técnicas utilizadas para el desarrollo, motor de base de datos:.....</i>	93
<i>Descripción detallada del conocimiento básico que debe tener el recurso humano que operará el sistema.....</i>	94
DISEÑO:.....	114
Arquitectura del sistema:	114
Arquitectura del <i>software</i>	116
Diseño de entradas y salidas	118

Diseño físico de la base de datos	132
Diagrama de clases	136
Diagrama de secuencia	137
PROGRAMACIÓN	145
PRUEBAS.....	169
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES.....	184
Conclusión	184
Recomendaciones	185
ANEXOS	190
PREGUNTAS DE LA ENTREVISTA	190
GUÍA DE OBSERVACIÓN.....	191

Dedicatoria

Dedico este trabajo principalmente a mi familia, por el apoyo y la motivación que me han brindado durante todo este proceso. Su confianza en mí ha sido fundamental para seguir adelante y alcanzar esta meta.

También a todas las personas que de alguna manera estuvieron presentes durante esta etapa, brindándome consejos, ánimo y apoyo en los momentos en que más lo necesité.

Este logro representa el esfuerzo de varios años y el cierre de una etapa importante en mi formación profesional.

Agradecimientos

Agradezco a la institución y a los profesores que formaron parte de mi proceso de formación académica, ya que a lo largo de la carrera me brindaron los conocimientos necesarios para desarrollar este trabajo final de graduación.

De igual manera, agradezco al profesor **Daniel Álvarez Garro** por la orientación brindada durante el proceso del taller de investigación, la cual permitió guiar el desarrollo de este proyecto.

Finalmente, agradezco a mi familia por su apoyo y motivación durante toda mi etapa universitaria, ya que su respaldo fue fundamental para culminar esta importante meta académica.

Tablas

Tabla 1. Requerimientos de hardware	24
Tabla 2. Recursos y costos del sistema	26
Tabla 3. Módulos del sistema.....	29
Tabla 4. Unidad Análisis.	59
Tabla 5. Hardware detallado.	91
Tabla 6. Hardware requerido para la producción del prototipo	92
Tabla 7. Aspectos esenciales de las telecomunicaciones	93
Tabla 8. Caso de uso: Acceso al sistema (Login).....	96
Tabla 9. Caso de uso: Gestión de nómina	98
Tabla 10. Caso de uso: Gestión de vacaciones.....	100
Tabla 11. Caso de uso: Gestión de incapacidades	102
Tabla 12. Caso de uso: Gestión de permisos.....	104
Tabla 13. Caso de uso: Gestión de horas extra.....	106
Tabla 14. Caso de uso: Gestión de aguinaldos	108
Tabla 15. Caso de uso: Gestión de aguinaldos	110
Tabla 16. Caso de uso: Gestión de asistencia y registro de marcas	112

CAPÍTULO I: INTRODUCCIÓN

Este proyecto tiene como finalidad desarrollar un sistema web para la gestión del pago de planilla en el gimnasio Titanium Fitness, ubicado en Alajuela. Esta empresa ofrece servicios relacionados con la actividad física, como entrenamientos personalizados, clases grupales y asesoría en salud y bienestar. Cuenta con un equipo de colaboradores en distintas áreas, cuya gestión laboral es fundamental para el buen funcionamiento de la organización.

Actualmente, el registro de asistencia, el control de horas extra y el cálculo de la planilla se realizan de forma manual, lo cual ha ocasionado errores en los pagos, retrasos en la entrega de la planilla y poca claridad en la comunicación entre el personal y la administración. Además, las solicitudes de vacaciones o permisos no se gestionan mediante un sistema formal, lo que dificulta llevar un seguimiento ordenado de estas peticiones.

Ante esta situación, se plantea como solución el desarrollo de un sistema web que permita automatizar estos procesos de forma centralizada, utilizando herramientas como HTML, CSS, PHP y MySQL. El sistema será ejecutado en un entorno local, lo que facilita su implementación sin necesidad de adquirir infraestructura adicional.

Con esta herramienta se espera mejorar la organización interna del gimnasio en todo lo relacionado con la gestión de planilla, aumentar la precisión en el cálculo de salarios y ofrecer a los colaboradores una forma más transparente y sencilla de gestionar sus solicitudes y consultar su información laboral.

Planteamiento del problema

El gimnasio Titanium Fitness, ubicado en Alajuela, es una empresa dedicada al bienestar físico de sus clientes por medio de entrenamientos personalizados, clases grupales y asesorías especializadas. A pesar de contar con una operación bien estructurada en su área de servicios, en la parte administrativa aún existen retos importantes, especialmente en lo relacionado con la gestión del personal y el cálculo de planilla.

Actualmente, los procesos como el registro de asistencia, el control de horas extra, la gestión de permisos, vacaciones e incapacidades, así como el cálculo de deducciones, aguinaldos y liquidaciones, se realizan de forma manual. Esto ha generado distintos problemas que afectan tanto a la administración como a los colaboradores.

Debido a esta situación, se han identificado varios problemas que afectan directamente los procesos internos del gimnasio y que justifican el desarrollo de una solución tecnológica. A continuación, se describen los principales:

Errores y retrasos en el cálculo de la planilla

La planilla se genera de forma manual, lo que ha provocado errores en los montos pagados y atrasos en la entrega. Esto afecta la confianza del personal y genera retrabajo en el área administrativa.

Falta de control sobre las horas extra trabajadas

Las horas extra se registran informalmente, sin un sistema que las valide. Esto ha resultado en pagos incorrectos, ya sea por exceso o por defecto. Estos desaciertos han generado inconformidad en el personal.

Ausencia de un sistema para la solicitud y aprobación de permisos

Los permisos se solicitan verbalmente o por medio de mensajería, sin dejar un registro formal. Esto dificulta el control y puede ocasionar malentendidos o aprobaciones sin seguimiento.

Vacaciones e incapacidades sin control formal

No existe un historial digital que indique cuántos días de vacaciones o incapacidades ha tenido cada colaborador. Esto puede llevar a aprobar solicitudes sin verificar los saldos reales, o a errores en los pagos correspondientes.

Cálculo manual de aguinaldos, deducciones y liquidaciones

El cálculo de estos conceptos se realiza manualmente, lo que puede generar omisiones o errores. Esta situación pone en riesgo el cumplimiento de las obligaciones laborales y puede provocar reclamos por parte de los empleados.

Registro manual de marcas de entrada y salida

El control de asistencia se lleva por métodos manuales, lo cual da pie a registros incompletos o inexactos. Esto dificulta la generación de reportes confiables sobre puntualidad y horas trabajadas.

Todos estos problemas reflejan la necesidad de implementar una solución tecnológica que automatice los procesos más sensibles de la gestión de personal. Un sistema web permitiría

organizar la información de forma centralizada, reducir errores y ofrecer mayor claridad tanto a la empresa como a sus colaboradores.

Objetivo general

Desarrollar un sistema web para el pago de planilla en el gimnasio Titanium Fitness, con el fin de facilitar el control de salarios, el registro de asistencia y la generación de reportes, utilizando tecnologías como HTML, CSS, PHP y MySQL.

Objetivos específicos

- Analizar los requerimientos funcionales y técnicos necesarios para el desarrollo del sistema de gestión de planilla del gimnasio Titanium Fitness.
- Diseñar la estructura del sistema web, incluyendo módulos para el registro de asistencia, cálculo de salarios y generación de reportes.
- Programar el sistema utilizando tecnologías web como HTML, CSS, PHP y MySQL.
- Probar el funcionamiento del sistema mediante pruebas controladas que aseguren su correcto desempeño según los requerimientos definidos.

Justificación

A continuación, se presenta la justificación del proyecto desde las perspectivas de viabilidad técnica, operativa, económica y legal, con base en las condiciones reales del gimnasio Titanium Fitness, lugar donde se desarrollará e implementará el sistema.

Viabilidad Técnica de la Investigación

El sistema web para la gestión del pago de planilla en Titanium Fitness es completamente factible. Tanto el entorno de desarrollo como los recursos disponibles permiten llevar a cabo el proyecto sin restricciones técnicas.

El sistema será desarrollado en entorno local utilizando tecnologías ampliamente conocidas como PHP, HTML, CSS, JavaScript y MySQL, en conjunto con el paquete XAMPP

para simular un servidor web y base de datos desde la computadora del desarrollador. Este enfoque elimina la necesidad de adquirir licencias comerciales o servicios externos, facilitando la implementación y pruebas del sistema.

Requerimiento del sistema operativo

Durante el desarrollo se utiliza el sistema operativo Windows 11 Pro, el cual ofrece estabilidad, compatibilidad y buen rendimiento para ejecutar todas las herramientas necesarias del proyecto. Además, cuenta con el soporte necesario para los programas de desarrollo web y las herramientas administrativas que se requieren.

Requerimientos de *software*

Las herramientas que se emplearán son gratuitas, estables y ampliamente utilizadas en entornos de desarrollo web:

- **Visual Studio Code:** para la edición del código fuente.
- **PHP:** lenguaje de programación utilizado para ejecutar la lógica del sistema y procesar las solicitudes del servidor.
- **XAMPP:** para levantar el entorno local con Apache y MySQL.
- **phpMyAdmin:** para la gestión de la base de datos.
- **Google Chrome / Microsoft Edge:** para realizar pruebas de navegación del sistema.

Estas herramientas son livianas y no requieren características de *hardware* avanzadas, por lo que el equipo actual cumple con los requisitos.

Herramientas ofimáticas

Para la documentación y presentación del proyecto se utiliza el paquete Microsoft Office 365, que incluye Word, Excel y PowerPoint. Esta *suite* ya se encuentra instalada y activada en el equipo del desarrollador, lo que facilita la creación de informes, tablas, cronogramas y demás elementos de apoyo.

Requerimientos de seguridad

Durante el desarrollo del sistema se utiliza Microsoft Defender, solución de seguridad integrada en Windows 11 Pro, que protege el equipo de posibles amenazas locales. Además, al tratarse de una aplicación en entorno local, el riesgo de exposición a amenazas externas es prácticamente nulo durante la fase de desarrollo.

Requerimientos de *hardware*

El equipo de cómputo utilizado cumple con los requerimientos necesarios para el desarrollo fluido del sistema. A continuación, se detallan sus especificaciones:

Tabla 1.

Requerimientos de hardware

<i>Hardware</i>	Especificaciones
Monitor	Xiaomi G27Qi – 27" – 2K – IPS – 180Hz
Mouse	Hyperx Pulsefire Haste 2 Mini – Wireless
Teclado	Logitech G413 TKL SE
Equipo de cómputo	AMD Ryzen 7 5700
Memoria RAM	32GB
Almacenamiento	M.2 1TB
Fuente de poder	Aerocool Mirage 850W 80 Plus Gold

Fuente: Elaboración propia.

Viabilidad Operativa de la Investigación

El desarrollo del sistema web para la gestión de planilla en Titanium Fitness es operativamente viable, porque cuenta con el aval de la administración del gimnasio y se adapta de forma práctica al entorno actual de trabajo. El personal que lo utilice tiene conocimientos

básicos en el manejo de computadoras, lo que facilitará su uso sin necesidad de capacitaciones extensas.

La aplicación se enfoca en simplificar las labores administrativas relacionadas con el cálculo de salarios, el control de asistencia y la gestión de permisos, vacaciones e incapacidades. Se diseñó con una interfaz clara y sencilla, pensada para usuarios no técnicos, y se entregará con un manual digital que explica cómo utilizar cada una de sus funciones de forma detallada.

El sistema será operado principalmente por la persona encargada de planilla, sin que eso implique cambios en el personal actual ni en las funciones que ya desempeñan. Más bien, permitirá reducir el tiempo dedicado a tareas repetitivas y minimizar errores que actualmente ocurren por llevar los procesos de forma manual.

Con esta solución, Titanium Fitness podrá realizar sus gestiones internas con mayor orden, precisión y eficiencia, lo cual beneficiará tanto al área administrativa como a los colaboradores.

Viabilidad Económica

Desde el punto de vista económico, la implementación del sistema web para la gestión del pago de planilla en Titanium Fitness resulta viable, ya que no implica una inversión elevada ni gastos adicionales para la empresa.

El desarrollo del sistema será realizado por el estudiante responsable del proyecto, utilizando su equipo de cómputo personal con las características técnicas necesarias para llevar a cabo las fases de programación y prueba. Asimismo, se cuenta con una licencia activa de Microsoft Office 365 y con herramientas gratuitas de desarrollo como Visual Studio Code, XAMPP y phpMyAdmin, lo cual evita la necesidad de adquirir *software* adicional.

Por parte del gimnasio, se dispone de un equipo de escritorio funcional utilizado actualmente en tareas administrativas. Este equipo será destinado a ejecutar el sistema una vez finalizado, por lo que no se requiere inversión en *hardware* nuevo.

Además, al tratarse de un Trabajo Final de Graduación, la elaboración del sistema será realizada por mí como parte de mi proceso académico, por lo que no implica un costo de mano de obra.

En la Tabla 2 se detallan los recursos utilizados y su costo estimado, lo que evidencia que el proyecto no representa una carga económica para la organización.

Tabla 2.*Recursos y costos del sistema*

Recurso	Costo estimado (C)
Computadora del desarrollador (uso personal)	0 (recurso disponible)
Computadora del gimnasio (uso administrativo)	0 (recurso existente)
Licencia de Office 365	0 (recurso existente)
Visual Studio Code (gratuito)	0
XAMPP (gratuito)	0
phpMyAdmin (gratuito)	0
Mano de obra del estudiante desarrollador	0 (TFG, sin costo profesional)
Total estimado	0

Fuente: Elaboración propia.

Viabilidad Legal

Para garantizar que el desarrollo e implementación del sistema web de gestión de planilla en Titanium Fitness se realice de forma responsable y conforme a la legislación costarricense vigente, se consideran las siguientes normativas clave:

Ley N.º 8968: Protección de la persona frente al tratamiento de sus datos personales

Esta ley establece los principios de consentimiento, confidencialidad y seguridad en el manejo de información personal (Asamblea Legislativa de Costa Rica, 2011). Debido a que el sistema administrará datos sensibles como asistencia, vacaciones, incapacidades y pagos de los colaboradores, se aplicarán prácticas que aseguren su protección y eviten el acceso no autorizado.

Ley N.º 8148: Artículos 196 BIS, 217 BIS y 229 BIS del Código Penal

Esta normativa contempla sanciones para quienes accedan, modifiquen o divulguen datos electrónicos sin autorización (Asamblea Legislativa de Costa Rica, 2001a). El sistema será de uso exclusivo del personal autorizado de Titanium Fitness, por lo que se implementarán controles de acceso y buenas prácticas para prevenir riesgos relacionados con el uso indebido de la información.

Ley N.º 4573: Delitos informáticos

Esta ley refuerza la importancia de proteger los sistemas informáticos ante posibles acciones delictivas (Asamblea Legislativa de Costa Rica, 2001b). Aunque el sistema será implementado localmente, se contemplan medidas básicas de seguridad para minimizar vulnerabilidades que pudieran comprometer la integridad de la información.

Ley N.º 6683: Derechos de autor

Aunque este proyecto es una solución original desarrollada como parte de un proceso académico, se respetarán los derechos de autor evitando el uso indebido de recursos externos o *software* con licencias comerciales (Asamblea Legislativa de Costa Rica, 1982). Todas las herramientas empleadas son de uso libre o con versiones gratuitas.

En conclusión, el sistema propuesto no presenta restricciones legales para su implementación y, por el contrario, contribuye al cumplimiento de los principios de confidencialidad, seguridad y protección de datos administrativos en Titanium Fitness.

Proyecciones

Este proyecto busca generar un impacto positivo en la gestión administrativa de Titanium Fitness, mediante la automatización del cálculo de planilla y la organización de procesos clave como asistencia, permisos y vacaciones. La incorporación de un sistema web local permitirá dejar atrás los métodos manuales que actualmente se utilizan, los cuales son propensos a errores, duplicidad de información y pérdida de tiempo.

Este sistema pretende ser una herramienta confiable que facilite el registro de datos laborales y brinde mayor claridad en el manejo de los pagos y derechos del personal, sin necesidad de invertir en *software* comercial. Su enfoque práctico responde a las necesidades reales del gimnasio, porque permite un mayor control interno y simplifica las tareas administrativas.

Alcance

El sistema será desarrollado como una solución web de uso local, enfocada exclusivamente en el control de planilla y aspectos relacionados con el tiempo laboral del personal. Su implementación busca:

- Establecer controles automatizados para calcular planillas de forma exacta.
- Mejorar la organización de solicitudes y registros de permisos, vacaciones e incapacidades.
- Garantizar un respaldo digital de la información que antes se almacenaba en papel o en múltiples hojas de Excel.
- Brindar a la administración del gimnasio una visión clara y oportuna del estado laboral de cada colaborador.

Alcance funcional

El sistema cuenta con una estructura modular que cubre los principales procesos involucrados en la gestión de la planilla y el tiempo laboral. A continuación, se describen los módulos que lo componen:

Tabla 3.*Módulos del sistema*

Nombre del módulo	Descripción del módulo
Gestión de horas extra	Este módulo permitirá al colaborador solicitar las horas extra vía sistema a su jefatura inmediata. Si esta las aprueba, se notifica a RRHH por el sistema, y luego al colaborador. Si se rechaza, se notifica al colaborador directamente vía sistema.
Gestionar nómina	Este módulo permite realizar el cálculo automático de los salarios de los colaboradores, considerando deducciones, horas extra, incapacidades y otros ajustes correspondientes. Además, genera el comprobante de pago y facilita el registro de la planilla mensual.
Gestión de vacaciones	Este módulo permitirá que el colaborador solicite vacaciones vía sistema a su jefatura inmediata. Si esta las aprueba, se notifica a RRHH, y, posteriormente, al colaborador. En caso de rechazo, se notifica también vía sistema.
Gestionar incapacidades	Este módulo administrará los periodos de incapacidad de los empleados por enfermedad o accidente, y calculará las incapacidades correspondientes según la normativa costarricense.
Gestionar aguinaldos	Este módulo procesará los cálculos de los pagos de aguinaldos por cada colaborador conforme a la ley costarricense.
Gestionar Asistencia	Este módulo registrará las marcas de entrada y salida de cada colaborador, generando reportes detallados de asistencia y puntualidad.
Gestionar de liquidaciones	Este módulo se encargará de gestionar el pago de liquidaciones en el tiempo correcto según la ley correspondiente.
Gestión de permisos	El colaborador podrá solicitar permisos vía sistema a su jefatura inmediata. Si se aprueban, se notifica a RRHH y luego al colaborador. Si se rechazan, se notifica al colaborador por el sistema.
Evaluar empleados	Permite registrar una valoración general del desempeño de cada colaborador según criterios como asistencia, puntualidad y cumplimiento de tareas. Las evaluaciones quedan guardadas en el sistema para consulta interna de la administración.

Mantenimientos	Este módulo se encargará de realizar el borrado, inserción, modificación, actualización de datos
Consultas	Este módulo se encargará de generar información proporcionada de las diferentes tablas
Reportes	Este módulo se encargará de generar información proporcionada de las diferentes tablas y procesos, pero con un formato específico, según lo solicite el usuario. Podrá ser impreso o por pantalla
Seguridad	Este módulo se encargará de realizar la autenticación de contraseñas y definición de perfiles

Elaboración: Fuente propia.

Alcance metodológico

Para el desarrollo del sistema web de gestión de planilla en Titanium Fitness se utilizará la metodología Kanban, por tratarse de una herramienta visual y flexible que permite organizar las tareas de forma ordenada y progresiva. Esta metodología facilita el seguimiento del proyecto y ayuda a mantener un ritmo de trabajo constante, ideal para proyectos individuales como este Trabajo Final de Graduación.

Kanban divide las tareas en tres etapas principales: pendiente, en proceso y finalizado, las cuales se representan mediante un tablero visual. Esto permite ver claramente el estado actual del desarrollo, priorizar las actividades más urgentes y evitar acumulaciones de trabajo sin concluir.

Entre los principios fundamentales que guiarán el desarrollo del sistema se encuentran:

- Enfoque en la calidad del producto.
- Mejora continua durante cada fase.
- Reducción de tareas innecesarias.
- Adaptabilidad ante cambios o ajustes del sistema.

Ciclo de vida

- **Identificación de tareas clave:** Se inicia con un análisis general de los módulos que componen el sistema, dividiendo cada uno en actividades concretas y alcanzables. Estas tareas se convierten en tarjetas dentro del tablero Kanban

- **Planificación visual del tablero:** Se diseña un tablero organizado por columnas que representan el avance del trabajo, tales como: “Por hacer”, “En desarrollo” y “Completado”. Esto permite tener una vista clara del estado del proyecto en todo momento.
- **Asignación de prioridades:** Las tareas más importantes o que habilitan otras funciones se desarrollan primero. Esto garantiza un flujo lógico de avance y evita retrasos innecesarios por dependencias mal gestionadas.
- **Ejecución continua y revisión modular:** A medida que se van completando las tareas, cada módulo es probado de forma individual. Este enfoque permite detectar errores rápidamente y hacer ajustes sin afectar el resto del sistema.
- **Cierre y validación final:** Una vez completados todos los módulos y verificado su correcto funcionamiento, se procede con pruebas de integración. Al finalizar, el tablero refleja visualmente que todo el sistema está listo para entrega.

Alcance tecnológico

El sistema de gestión de planilla para Titanium Fitness será desarrollado con tecnologías accesibles y compatibles con los recursos disponibles en la organización. El enfoque está orientado a una solución sencilla, funcional y de uso local, sin necesidad de conexión a internet ni servicios externos.

A continuación, se detallan los principales componentes tecnológicos del proyecto:

- **Lenguaje de programación:** Se utilizará PHP para desarrollar la lógica del sistema del lado del servidor. Es una herramienta ampliamente utilizada en aplicaciones web por su flexibilidad, facilidad de aprendizaje y buen rendimiento en entornos pequeños y medianos.
- **Interfaz de usuario:** Las pantallas del sistema se construirán utilizando HTML y CSS, lo cual permitirá generar una presentación clara, ordenada y accesible desde cualquier navegador compatible.
- **Base de datos:** La información se almacenará en MySQL, un sistema de gestión de bases de datos confiable, gratuito y compatible con PHP, ideal para soluciones locales.

- **Entorno de desarrollo:** El desarrollo se realizará en Visual Studio Code, un editor ligero y personalizable que facilita la escritura de código y la organización del proyecto.
- **Implementación:** El sistema será una aplicación web de ejecución local, instalada en un equipo administrativo del gimnasio. No utilizará servicios en la nube ni dependerá de conexión a internet, lo que proporciona mayor control y seguridad sobre los datos.

CAPÍTULO II: MARCO REFERENCIAL

En este apartado se explican los conceptos necesarios para comprender el funcionamiento del sistema desarrollado.

Un prototipo puede entenderse como la primera versión funcional de un sistema. Aunque no es el producto final, sí incorpora las características esenciales para probar ideas, visualizar el diseño y evaluar si la solución planteada responde adecuadamente al problema. En el desarrollo de aplicaciones, los prototipos permiten explorar la experiencia de usuario, detectar posibles fallos y ajustar funciones antes de invertir tiempo y recursos en una versión completa.

Esta etapa inicial es clave porque permite validar la propuesta sin asumir grandes costos. Realizar cambios en un prototipo suele ser rápido y sencillo, mientras que modificarlos en un sistema ya finalizado implica un esfuerzo significativamente mayor. Además, el prototipo ayuda a comprender cómo interactúan los usuarios con la herramienta y qué elementos pueden mejorarse para que el sistema final sea más intuitivo y eficiente.

De acuerdo con Virket Agency (2021), un prototipo funciona como una representación preliminar del producto final y sirve para confirmar que el diseño cumple con los objetivos previstos. En otras palabras, permite experimentar, corregir y fortalecer la idea antes de avanzar hacia la programación formal.

En el ámbito del desarrollo de sistemas, el prototipo se relaciona estrechamente con el *software*, ya que es el *software* el que permite construir y ejecutar esa primera versión de prueba. Asimismo, el prototipo brinda información útil para mejorar el *software* y asegurar que sus funciones se ajusten a los requerimientos definidos.

El *software* puede definirse como el conjunto de instrucciones y programas que permiten que un equipo informático realice tareas específicas. Gracias a él es posible gestionar información, automatizar procedimientos, analizar datos o dar vida a aplicaciones completas. De acuerdo con el artículo “¿Qué es *software*?” (s. f.), estas herramientas pueden emplearse para controlar procesos, manipular datos y crear sistemas adaptados a distintas necesidades. El *software* desempeña diversas funciones relevantes dentro de un proyecto, entre ellas:

- **Administrar información:** implica almacenar, organizar y proteger datos para que puedan ser consultados y utilizados de manera eficiente.
- **Automatizar tareas:** permite transformar procesos manuales en acciones que el sistema ejecuta automáticamente mediante reglas predefinidas.

- **Desarrollar aplicaciones:** comprende actividades como diseñar, programar, probar y depurar un sistema para convertirlo en una herramienta funcional.
- **Facilitar la productividad:** integra herramientas que permiten mejorar la comunicación, el trabajo en equipo o la organización interna mediante plataformas digitales.

Para construir *software* se utilizan distintas herramientas de programación, las cuales permiten crear, editar y depurar código. Estas herramientas facilitan el proceso de desarrollo y ayudan a asegurar que el sistema funcione correctamente. Euroinnova Business School (2022) describe estas herramientas como programas que apoyan la creación y el mantenimiento de aplicaciones, desde el diseño inicial hasta la fase final de implementación.

Los entornos de desarrollo, los editores de código, los compiladores y los sistemas de control de versiones forman parte de este conjunto de herramientas que acompañan todo el ciclo de programación y garantizan un trabajo más organizado y eficiente.

El *Software* y el Prototipado

El desarrollo de cualquier solución informática parte del concepto de *software*, que se define como el conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora (Pérez & Merino, 2021). El *software* puede utilizarse para gestionar datos, automatizar procesos y crear aplicaciones, siendo el componente intangible que da vida al *hardware* y permite a los usuarios interactuar con la tecnología para resolver problemas específicos.

En las etapas iniciales del desarrollo de *software*, es común utilizar un prototipo. Un prototipo es una representación concreta, aunque inicial, de una idea. Se trata de una simulación que incorpora los elementos esenciales para ser funcional y permitir pruebas, brindando una apariencia y funcionalidades básicas. Como menciona Virket Agency (2021), "Es un primer modelo que sirve como representación o simulación del producto final que permitirá verificar que el diseño tenga las características necesarias para cumplir con el objetivo de negocio" (párr. 3).

La aplicación de un prototipo en este proyecto permitirá presentar a la administración del Gimnasio Titanium Fitness una versión tangible del sistema, facilitando la retroalimentación temprana sobre la interfaz y las funcionalidades. Asimismo, permite minimizar riesgos y costos a futuro, ya que es más económico y menos arriesgado hacer

modificaciones en esta fase que tener que comenzar desde cero en la etapa de programación. La conexión entre el *software* y el prototipo es intrínseca: el *software* es la herramienta con la que se construye el prototipo, y el prototipo, al mismo tiempo, sirve para probar y refinar el *software* antes de su versión final.

Los Sistemas Web

El sistema desarrollado en este proyecto corresponde a una aplicación web de uso local, es decir, un sistema que se ejecuta dentro de la propia red del gimnasio sin depender de Internet. Este tipo de solución, según Sistemas Perú Web (2024), permite que los usuarios accedan a la herramienta directamente desde un navegador, utilizando el mismo conjunto de lenguajes y tecnologías que se emplean en el desarrollo web tradicional, pero con la ventaja de que todos los datos se procesan y almacenan internamente.

A diferencia de un sitio web público accesible desde cualquier parte del mundo, un sistema web local funciona únicamente dentro de la infraestructura de la organización. Esto ofrece un mayor nivel de seguridad, ya que la información no viaja por redes externas y se mantiene protegida de accesos no autorizados. En el caso del gimnasio Titanium Fitness, esta característica resulta especialmente importante, porque se trabaja con datos sensibles como información personal de los colaboradores, registros de asistencia, detalles salariales y reportes administrativos.

Los sistemas web locales también destacan por su facilidad de uso. No requieren instalar programas adicionales en cada computadora; basta con ingresar al navegador y acceder al sistema mediante una dirección interna. Esto simplifica la administración tecnológica, reduce errores de instalación y permite que todos los usuarios trabajen con la misma versión del sistema en tiempo real.

Otra ventaja de este tipo de arquitectura es la centralización de la información, ya que todo se almacena en una única base de datos. Esto evita duplicidad de registros, facilita la actualización de datos y permite generar reportes más confiables. Para el proceso de planilla, este beneficio es clave porque garantiza que los cálculos se basen siempre en la misma información y que todos los módulos del sistema —como asistencia, horas extra o permisos— se conecten entre sí de manera coherente.

Además, al ser un sistema web desarrollado a la medida del gimnasio, permite automatizar procesos que antes se realizaban de forma manual o dispersa, como el control de

entradas y salidas, el registro de ausencias o la generación de cálculos salariales. Esta automatización no solo agiliza el trabajo administrativo, sino que también reduce la posibilidad de errores humanos.

Finalmente, optar por un sistema web local representa una solución económicamente eficiente para Titanium Fitness. Al no depender de servidores externos ni de servicios de alojamiento web, la empresa evita costos adicionales y mantiene un control total sobre su información y su infraestructura.

Los Usuarios del Sistema

Dentro del desarrollo del sistema para la gestión de planilla del gimnasio Titanium Fitness, el término “usuario” hace referencia a cualquier persona que interactúa con la plataforma para ejecutar una tarea dentro del proceso administrativo. En este proyecto, los colaboradores del gimnasio son considerados usuarios del sistema, ya que serán quienes utilicen la herramienta para acciones como visualizar información, registrar datos o solicitar permisos.

Un usuario, en términos informáticos, es toda persona que opera un sistema de *hardware* o *software* con un propósito definido, ya sea consultar información, ejecutar funciones o administrar procesos (Salesforce, s. f.). En esta misma línea, explica que los usuarios son quienes utilizan habitualmente programas o aplicaciones para realizar actividades dentro de un dispositivo o entorno digital.

El rol que desempeñan los usuarios en un proyecto informático es fundamental, especialmente durante la fase de análisis y pruebas. Su participación permite evaluar si la interfaz es intuitiva, si las funciones responden a las necesidades reales y si el flujo de trabajo se ajusta al proceso operativo del gimnasio. Un sistema puede estar técnicamente bien construido, pero si no responde a la experiencia y expectativas de quienes lo utilizarán a diario, su implementación será poco efectiva. Por lo tanto, la retroalimentación de los colaboradores resulta clave para ajustar la usabilidad, mejorar los módulos y garantizar una curva de aprendizaje adecuada.

Además, para asegurar un manejo ordenado y seguro de la información, los sistemas suelen establecer distintos tipos de usuarios según sus privilegios y responsabilidades dentro de la estructura organizacional. Esta diferenciación permite que cada persona acceda

únicamente a las funciones que necesita, lo que reduce riesgos y asegura que los procesos se ejecuten correctamente. Entre los roles más comunes se encuentran:

Usuarios Operacionales

Son aquellos que introducen información en el sistema o utilizan funciones básicas para registrar eventos relacionados con su trabajo. En el contexto del gimnasio, este rol lo desempeña un colaborador que reporta sus horas trabajadas, solicita vacaciones, registra ausencias o consulta datos personales. Su interacción directa con el sistema garantiza que la información registrada sea actual y confiable.

Usuarios Administradores

Corresponden a los encargados de gestionar la información global del sistema y supervisar su funcionamiento. Este tipo de usuario posee permisos avanzados para aprobar solicitudes, modificar datos, generar reportes, gestionar parámetros de la planilla y controlar los accesos de otros usuarios. En Titanium Fitness, este rol recaería en el encargado administrativo o en la persona responsable de la planilla, quien requiere un nivel mayor de control sobre los procesos.

La correcta definición de roles no solo facilita el uso del sistema, sino que también protege la integridad de los datos, delimita responsabilidades y mejora la organización interna del gimnasio. De esta forma, cada usuario realiza únicamente las tareas que le competen, asegurando un uso eficiente, ordenado y seguro de la plataforma desarrollada.

Módulo de *Software*

Un sistema complejo se organiza en módulos. Un módulo es un componente de *software* o parte de un programa que contiene una o más rutinas destinadas a ejecutar una función específica y claramente delimitada (Technopedia, 2020). Esta división del sistema en partes más pequeñas permite estructurar la aplicación de forma ordenada y comprensible, evitando que todo el funcionamiento recaiga en un solo bloque de código difícil de mantener. Gracias al enfoque modular, cada componente pretende resolver un único problema, lo que facilita su diseño, implementación y posterior revisión.

Una de las principales ventajas de trabajar con módulos es que el desarrollador puede concentrarse en un área puntual sin afectar directamente al resto del sistema. Esto mejora la organización del código, reduce la complejidad general y permite realizar cambios o correcciones de manera más segura. Asimismo, la modularidad agiliza las pruebas, debido a

que cada módulo puede evaluarse de forma independiente antes de integrarse al sistema completo. Esta independencia también contribuye al mantenimiento, porque cuando surge un problema, resulta más sencillo identificar en qué parte del sistema se encuentra y corregirlo sin comprometer otras funciones.

En el caso del sistema desarrollado para Titanium Fitness, la modularidad fue clave para dividir los diferentes procesos administrativos que conforman la gestión de planilla. Cada módulo aborda una actividad específica del gimnasio, permitiendo que el sistema sea más claro, ordenado y fácil de escalar. Algunos de los módulos implementados son Gestión de Vacaciones, donde se administran las solicitudes y acumulaciones de días; Cálculo de Horas Extra, encargado de procesar los montos adicionales según las horas trabajadas fuera de la jornada regular; y Generación de Reportes, que compila información relevante para el encargado de planilla, como resúmenes de asistencia y cálculos salariales.

Aunque cada módulo opera de manera independiente, todos se integran entre sí para conformar una solución completa y coherente. Esta integración asegura que los datos fluyan correctamente a través del sistema y que la información registrada en un módulo esté disponible para los demás cuando sea necesario. Gracias a ello, el sistema puede ampliarse en el futuro agregando nuevos módulos —como permisos, incapacidades o evaluaciones— sin afectar la estructura ya implementada.

Metodología de Desarrollo: Kanban

Para gestionar el ciclo de vida del proyecto se utiliza la metodología Kanban, un enfoque ágil basado en la visualización del trabajo mediante un tablero dividido en columnas que representan el estado de cada tarea. Usualmente, estas columnas se organizan en categorías como “Por hacer”, “En progreso” y “Completado”, que permiten observar de manera inmediata cómo avanza el proyecto y qué actividades requieren atención (Kanban University, 2023). El objetivo principal de Kanban es mantener un flujo de trabajo constante y evitar la acumulación excesiva de tareas en proceso. Este método promueve que las actividades se aborden de forma ordenada, concentrándose en finalizar lo que está en progreso antes de iniciar nuevas acciones.

Esto reduce la sobrecarga, mejora la calidad del trabajo y permite avanzar de manera más predecible, especialmente en proyectos desarrollados por una sola persona. Otra fortaleza de Kanban es su simplicidad. No exige reuniones formales, ciclos de desarrollo rígidos ni

planificación compleja. Más bien, se adapta a las necesidades del proyecto, permitiendo reorganizar tareas, ajustar prioridades y visualizar de forma clara qué etapas están completas y cuáles están pendientes. Esta flexibilidad lo convierte en una metodología ideal para proyectos pequeños o individuales, donde la claridad y la gestión del tiempo son fundamentales. En el sistema de gestión de planilla para Titanium Fitness, Kanban permitió dividir el trabajo en módulos —como el cálculo de horas extra, la gestión de vacaciones o la generación de reportes— y avanzar en ellos de forma progresiva y ordenada. Al visualizar el flujo de actividades, fue posible asegurarse de que cada módulo fuera completado antes de continuar con el siguiente, evitando confusiones y manteniendo una estructura definida durante todo el proceso de desarrollo. En conjunto, Kanban proporcionó una forma práctica, visual y eficiente de administrar el proyecto, permitiendo que el desarrollo del sistema avanzara de manera organizada y coherente con los objetivos establecidos.

Herramientas de Desarrollo

El lenguaje de programación y las herramientas utilizadas son elementos esenciales dentro del proceso de desarrollo de *software*, ya que determinan la manera en que se construye, organiza y mantiene una aplicación. La elección adecuada de estas herramientas influye directamente en la eficiencia del proyecto, la facilidad para detectar errores y la capacidad de extender o modificar el sistema en el futuro.

En el caso de este proyecto, se seleccionaron herramientas de uso libre y ampliamente adoptadas en la industria tecnológica, con el objetivo de asegurar un proceso de desarrollo accesible, estable y compatible con las necesidades del gimnasio Titanium Fitness. Estas herramientas permiten implementar el sistema de manera ordenada, trabajar con una estructura clara y realizar pruebas continuas durante el ciclo de vida del *software*.

Además, optar por tecnologías consolidadas y gratuitas facilita que el sistema pueda mantenerse en el tiempo sin depender de licencias costosas o plataformas cerradas. Esto asegura que, en caso de futuras mejoras, cualquier desarrollador pueda comprender la estructura del proyecto y continuar con su evolución sin enfrentar barreras técnicas o económicas.

Entorno de Desarrollo Integrado (IDE): Visual Studio Code

Para la escritura, depuración y mantenimiento del código fuente del proyecto, se utilizará Visual Studio Code (VS Code). Este entorno de desarrollo integrado, creado por Microsoft, ha alcanzado gran reconocimiento en la comunidad de desarrollo web por su rendimiento, extensibilidad y compatibilidad con múltiples sistemas operativos (Microsoft, 2023). Su arquitectura modular permite al desarrollador habilitar o deshabilitar funciones según lo requiera el proyecto.

Una de las grandes ventajas de VS Code es su extenso catálogo de extensiones. Estas permiten añadir funcionalidades especializadas como soporte para PHP, depuración avanzada, integración con Git y conexión con bases de datos MySQL. En el proyecto de gestión de planillas, estas extensiones facilitarán la edición de archivos, la depuración de errores y la conexión con el servidor local de forma eficiente (Microsoft, 2023).

Adicionalmente, VS Code incorpora herramientas como IntelliSense, que sugiere completado de código, documentación contextual y autocompletado inteligente. También incluye una terminal integrada, que permite ejecutar comandos del servidor o *scripts* desde el mismo editor sin cambiar de ventana. Estas características mejoran el flujo de trabajo y aumentan la productividad durante el desarrollo del sistema.

Entorno de Desarrollo Local: XAMPP

Para la ejecución local del sistema en fase de desarrollo, se empleará **XAMPP**, una distribución gratuita que integra Apache, MariaDB (MySQL), PHP y Perl (Apache Friends, s.f.). Su facilidad de instalación y configuración lo convierte en una excelente opción para simular un entorno de servidor completo sin necesidad de servidores remotos o configuraciones complicadas.

El uso de XAMPP garantiza que el sistema funcione de forma autónoma dentro de un entorno controlado, procesando las solicitudes del usuario y manejando los datos mediante Apache y MariaDB sin depender de servicios externos. Con esto se asegura la privacidad de la información y permite realizar pruebas de forma segura y eficiente (Apache Friends, s.f.).

Además, XAMPP incluye herramientas como phpMyAdmin para la administración de bases de datos y muestra registros de errores en tiempo real, lo que facilita la depuración del sistema. Estas funciones permiten al desarrollador validar funcionalidades y corregir errores antes de llevar el sistema al entorno final del gimnasio.

Tecnologías del Lado del Cliente: HTML y CSS

Para estructurar el contenido y elementos visuales del sistema, se utilizará HTML (HyperText Markup Language). Este lenguaje define la semántica y el esqueleto de las páginas web, organizando textos, formularios, imágenes y enlaces de forma accesible y lógica (MDN Web Docs, s.f.-a). En el proyecto, HTML estructurará las vistas de inicio de sesión, registros y reportes.

Fuente:StickPNG

Por otro lado, CSS (Cascading Style Sheets) se encargará de la presentación visual del contenido. CSS permite separar el diseño de la estructura, controlando colores, tipografías, márgenes y disposición responsiva de los elementos. Esto ayuda a mantener consistencia visual en todas las páginas del sistema y mejora la experiencia del usuario (MDN Web Docs, s.f.-b).

La combinación de HTML y CSS permite desarrollar interfaces limpias, funcionales e intuitivas que complementan la lógica *backend* escrita en PHP. Con este enfoque, el sistema ofrecerá una experiencia visual profesional y coherente, acorde con los estándares actuales de desarrollo web.

Tecnologías del Lado del Servidor

PHP

Fuente:Seeklogo

El sistema de gestión de planillas estará desarrollado con PHP, un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo web del lado del servidor. PHP destaca por su capacidad de generar contenido dinámico, interactuar con bases de datos y manejar peticiones HTTP de manera eficiente (The PHP Group, s. f.). Su sintaxis sencilla y su integración nativa con HTML lo convierten en una herramienta versátil para construir aplicaciones web funcionales y adaptables.

Una de las principales razones para seleccionar PHP en este proyecto es su compatibilidad con MySQL y otros sistemas de bases de datos, lo cual facilita la comunicación entre la capa lógica y la capa de almacenamiento. Gracias a esta integración, el sistema podrá registrar, modificar y consultar datos de forma rápida y segura. Además, al tratarse de un lenguaje interpretado, PHP permite realizar ajustes al código sin necesidad de recompilar, lo que agiliza la etapa de pruebas y mantenimiento (The PHP Group, s. f.).

Otro aspecto relevante de PHP es su amplia comunidad de desarrolladores y la cantidad de recursos disponibles, lo cual asegura soporte técnico constante y acceso a bibliotecas actualizadas. Esta característica favorece la estabilidad del sistema a largo plazo y permite incorporar mejoras o nuevas funcionalidades sin necesidad de recurrir a herramientas propietarias.

MySQL

Para el manejo de la información, el sistema empleará MySQL, un sistema de gestión de bases de datos relacional (RDBMS) de código abierto desarrollado originalmente por MySQL AB y actualmente mantenido por Oracle Corporation (Oracle, s.f.). MySQL es reconocido por su rendimiento, estabilidad y compatibilidad con múltiples plataformas, lo que lo convierte en una opción ideal para aplicaciones que requieren fiabilidad y eficiencia en la administración de datos.

En este proyecto, MySQL almacenará los registros del personal, cálculos de horas extra, vacaciones y demás información relacionada con la gestión de planillas. Su modelo relacional permite estructurar los datos mediante tablas interconectadas, garantizando la consistencia e integridad de la información. Además, su lenguaje SQL facilita la ejecución de consultas precisas, reportes y operaciones de mantenimiento con bajo consumo de recursos del servidor (Oracle, s. f.).

Finalmente, la elección de MySQL responde también a su compatibilidad directa con PHP, lo que permite un intercambio de datos fluido entre el sistema web y la base de datos. Esta combinación ha sido probada en numerosos entornos empresariales, garantizando un

desempeño estable, escalable y con bajo costo de implementación para proyectos de pequeño y mediano tamaño, como el del gimnasio Titanium Fitness.

phpMyAdmin

Para la administración de la base de datos utilizada en este proyecto se empleó phpMyAdmin, una herramienta web ampliamente reconocida para la gestión de servidores MySQL y MariaDB. phpMyAdmin permite realizar tareas de administración mediante una interfaz gráfica intuitiva que facilita operaciones como la creación y modificación de tablas, la ejecución de consultas SQL, la importación y exportación de datos y la visualización estructurada de los registros almacenados (phpMyAdmin, 2023). Al funcionar directamente sobre un navegador web, esta herramienta reduce la complejidad asociada al uso de líneas de comando y proporciona un entorno accesible tanto para desarrolladores como para administradores de sistemas. Según la documentación oficial, phpMyAdmin fue diseñado para ofrecer una administración completa del motor MySQL a través de un panel centralizado, lo que permite gestionar permisos, optimizar tablas y supervisar el estado del servidor de manera eficiente (phpMyAdmin Documentation, 2023). En el contexto del gimnasio Titanium Fitness, phpMyAdmin resultó esencial durante el desarrollo del sistema, porque permitió manipular la base de datos de forma rápida y segura, mantener un control preciso sobre los datos almacenados —como registros de asistencia, información de colaboradores y cálculos de planilla— y realizar ajustes estructurales conforme avanzaba el proyecto. Su integración dentro del entorno XAMPP también contribuyó a un flujo de trabajo más ágil, factor que permitió administrar MySQL sin necesidad de herramientas externas o configuraciones complejas.

Gestión de Recursos Humanos y Planilla

El sistema se enfoca en automatizar tareas del departamento de Recursos Humanos (RRHH), que es el área que gestiona todo lo relacionado con las personas que trabajan en una empresa, desde su reclutamiento hasta su desvinculación (Factorial HR, s. f.). Como menciona Andrés (2021), "Un sistema de administración de Recursos Humanos es una herramienta tecnológica creada para facilitar el trabajo de los profesionales que gestionan el personal y el talento de una empresa" (párr. 3).

Además de estas funciones, el departamento de Recursos Humanos también se encarga de administrar procesos como la capacitación, el control de asistencia, el seguimiento de desempeño y la gestión de beneficios laborales, los cuales requieren un manejo preciso y actualizado de información. En este contexto, los sistemas digitales se han convertido en un respaldo esencial para la automatización de tareas repetitivas y la reducción de errores humanos. Según Oracle (2024), los sistemas de información de Recursos Humanos (HRIS) permiten centralizar datos, automatizar trámites administrativos y mejorar la eficiencia en procesos como la nómina, los permisos, las vacaciones y la administración del personal.

La incorporación de este tipo de herramientas en organizaciones como Titanium Fitness no solo contribuye a agilizar tareas operativas, sino que también fortalece la capacidad del gimnasio para organizar su información, garantizar la trazabilidad de cada proceso y facilitar la toma de decisiones basada en datos confiables. En conjunto, un sistema de RRHH automatizado ofrece mayor transparencia, consistencia y control sobre los procesos internos que impactan directamente en la gestión de la planilla.

Planilla de Sueldos

La planilla de sueldos es el documento mediante el cual los patronos reportan a la Caja Costarricense de Seguro Social (CCSS) los salarios de sus trabajadores activos (FODESAF, s.f.). Es el registro financiero fundamental que detalla la remuneración de un empleado, incluyendo salario bruto, bonificaciones, aportes y deducciones. Además de cumplir una función legal, la planilla sirve como soporte administrativo para garantizar que los pagos se realicen de manera correcta y que se apliquen las obligaciones patronales estipuladas por la normativa costarricense.

La CCSS establece que la planilla constituye la base para calcular las cuotas obrero-patronales, los seguros sociales y las contribuciones obligatorias, por lo que su correcta elaboración es esencial para asegurar el cumplimiento de las responsabilidades laborales y fiscales de la empresa (CCSS, 2023). Asimismo, la planilla permite mantener un registro histórico de los movimientos salariales y constituye una herramienta clave para auditorías internas, procesos contables y la toma de decisiones relacionadas con la gestión del recurso humano.

En el ámbito organizacional, la planilla también cumple una función operativa relevante, ya que refleja información sobre horas laboradas, horas extra, permisos,

incapacidades y otros conceptos que afectan el cálculo salarial mensual. Plataformas especializadas en gestión de nómina destacan que un sistema automatizado reduce errores, mejora la precisión de los cálculos y facilita la generación de reportes para la administración y la contabilidad (ADP, 2024). Esto es especialmente importante en organizaciones que dependen de registros actualizados y precisos para garantizar transparencia y eficiencia en sus procesos internos.

Permisos Laborales

Los permisos laborales son autorizaciones para ausentarse de las labores. En Costa Rica, este servicio está destinado a situaciones como enfermedad grave o fallecimiento de un familiar (Dirección de Gestión del Talento Humano, s. f.). Estos permisos permiten que la persona trabajadora pueda atender asuntos personales o familiares urgentes sin comprometer su estabilidad laboral ni sus derechos adquiridos.

De acuerdo con el Código de Trabajo de Costa Rica, los trabajadores tienen derecho a solicitar ausencias justificadas en casos debidamente comprobados, y el patrono está obligado a conceder ciertos permisos cuando están respaldados por la normativa. Por ejemplo, la legislación costarricense establece licencias remuneradas por fallecimiento de familiares directos, con plazos definidos según el grado de parentesco, garantizando que el colaborador cuente con el tiempo necesario para atender estos procesos personales (Asamblea Legislativa, 2024).

Asimismo, el Ministerio de Trabajo y Seguridad Social ha señalado que, en situaciones no reguladas expresamente por la ley, el otorgamiento de permisos queda sujeto al criterio del patrono. Sin embargo, si se concede un permiso de manera habitual, este debe aplicarse bajo criterios de igualdad para evitar discriminación laboral y mantener un trato coherente entre los colaboradores (MTSS, 2023). Esto subraya la importancia de documentar adecuadamente cada solicitud de permiso y sus justificaciones dentro de la organización.

Contar con un sistema digital para gestionar permisos laborales, como el desarrollado para Titanium Fitness, facilita el registro, seguimiento y aprobación de estas solicitudes. Además, permite que la administración del gimnasio mantenga un control preciso de las ausencias, realice cálculos de planilla correctos y garantice el cumplimiento de las obligaciones

legales asociadas. De esta manera, el sistema contribuye a una gestión más transparente, organizada y alineada con la normativa laboral vigente en Costa Rica.

Aguinaldo

El aguinaldo es un salario adicional que todo empleador debe pagar a sus trabajadores dentro de los primeros veinte días de diciembre de cada año. En Costa Rica se considera un “salario trece”, porque corresponde a una doceava parte de todos los salarios ordinarios y extraordinarios devengados durante el periodo comprendido entre el 1° de diciembre del año anterior y el 30 de noviembre del año en curso (MTSS, s.f.-b). Este derecho laboral está regulado por la Ley de Aguinaldo a los Trabajadores del Sector Privado y constituye una obligación patronal independiente del salario mensual.

La finalidad del aguinaldo es brindar un apoyo económico adicional a los trabajadores en una época del año que suele implicar mayores gastos familiares. De acuerdo con la normativa vigente, para su cálculo se deben sumar todos los salarios percibidos por el trabajador durante el periodo anual, incluyendo horas extra, comisiones, incentivos y otros pagos salariales, y dividir el resultado entre doce (Asamblea Legislativa de Costa Rica, 1959). La fórmula puede expresarse de la siguiente manera: **Aguinaldo = (Suma total de salarios percibidos en el periodo) / 12**

Este beneficio no está sujeto a cargas sociales ni a impuestos sobre la renta, lo que significa que el monto debe ser pagado en su totalidad al trabajador. En el contexto del gimnasio Titanium Fitness, la automatización del cálculo del aguinaldo dentro del sistema de planilla garantiza precisión en los montos, evita errores manuales y facilita la generación de reportes para fines administrativos y contables.

Liquidación Laboral

Es el pago de los derechos adquiridos por el trabajador al finalizar la relación laboral (preaviso, cesantía, vacaciones y aguinaldo proporcional). Es una compensación que la empresa debe proporcionar cuando la terminación del contrato no es atribuible al empleado (Time, s. f.).

Además de representar un derecho laboral, la liquidación cumple una función de protección económica para el trabajador, ya que le permite enfrentar el periodo posterior a la

finalización del contrato con un respaldo financiero según el tiempo de servicio prestado. Este pago incluye tanto montos acumulados como indemnizaciones, por lo que su cálculo debe realizarse de manera ordenada, transparente y tomando en cuenta cada uno de los componentes generados durante la relación laboral.

Generalmente, la liquidación contempla varios elementos que reflejan el historial laboral del empleado: días de vacaciones pendientes, aguinaldo proporcional, pago de preaviso en caso de no haberse otorgado trabajo y la cesantía, que varía según los años laborados. Cada uno de estos rubros se calcula de forma independiente y luego se integran en un único monto, el cual debe entregarse al trabajador junto con un desglose claro para evitar confusiones o reclamos posteriores.

En las empresas modernas, gestionar adecuadamente la liquidación laboral es fundamental, ya que un cálculo incorrecto puede generar conflictos, afectar la relación con el personal o incluso derivar en sanciones legales. Por ello, muchas organizaciones optan por automatizar este proceso mediante sistemas internos que realizan cálculos precisos y reducen el margen de error.

En el caso del gimnasio Titanium Fitness, el sistema desarrollado facilita el cálculo de estos componentes de manera automática, garantizando que cada liquidación se realice con exactitud y uniformidad. Esto no solo agiliza el proceso administrativo, sino que también fortalece la transparencia en la gestión del recurso humano y asegura que cada trabajador reciba lo que le corresponde según su historial laboral.

Evaluación del Desempeño

Según el blog SYDLE (2022), la evaluación del desempeño es un proceso que se lleva a cabo para “identificar las cualidades inherentes de los trabajadores, sus habilidades y el nivel de competencia para su futuro crecimiento laboral” (párr. 2). Esta herramienta de Recursos Humanos permite medir el rendimiento de los colaboradores y tomar decisiones informadas sobre promociones, bonificaciones y capacitaciones.

Además de su función de medición, la evaluación del desempeño cumple un papel estratégico dentro de las organizaciones. A través de este proceso, es posible detectar fortalezas individuales, áreas de mejora y necesidades específicas de formación, lo que facilita la planificación del desarrollo profesional de cada colaborador. Además, ayuda a identificar

patrones dentro del equipo, como problemas de comunicación, falta de coordinación o sobrecargas de trabajo que puedan afectar la productividad general.

Otro beneficio importante es que la evaluación del desempeño fomenta una cultura de retroalimentación continua. Cuando se realiza de forma clara y estructurada, permite que los trabajadores comprendan cómo está siendo su aporte al cumplimiento de los objetivos de la empresa y qué acciones pueden implementar para mejorar su rendimiento. Esto genera mayor compromiso y una alineación más precisa entre las funciones del colaborador y las metas del gimnasio Titanium Fitness.

En el contexto del sistema desarrollado, contar con un módulo que registre y organice la evaluación del desempeño facilita a los encargados de la planilla y administración tener acceso a un historial confiable del progreso de cada trabajador. De esta manera, las decisiones sobre ascensos, reconocimientos o necesidad de capacitación se basan en datos objetivos y no en percepciones subjetivas. Además, centralizar esta información ayuda a mantener transparencia y a fortalecer los procesos internos de gestión del talento humano.

Gestión de nómina

La gestión de nómina consiste en calcular y administrar los pagos del personal, incluyendo salarios, deducciones y beneficios. Una adecuada administración garantiza el cumplimiento legal y la transparencia laboral (Dessler, 2020). Además, el uso de sistemas automatizados facilita el control de datos, reduce errores y mejora la eficiencia administrativa (Bizneo HR 2023).

La nómina es uno de los procesos administrativos más importantes dentro de cualquier organización, ya que de su precisión depende no solo el pago correcto a los colaboradores, sino también el cumplimiento de obligaciones legales relacionadas con cargas sociales, retenciones y reportes institucionales. Un manejo erróneo de la nómina puede generar inconformidades, retrasos en pagos e incluso sanciones legales, por lo que su gestión requiere estructuras claras y procedimientos bien establecidos.

Además de los cálculos salariales, la gestión de nómina implica registrar ausencias, aplicar políticas internas, controlar horas extra, administrar beneficios y llevar un registro histórico de los movimientos laborales. Todo esto demanda un manejo ordenado y actualizado de la información, especialmente en organizaciones que cuentan con varios colaboradores o turnos rotativos, como es el caso del gimnasio Titanium Fitness.

Implementar un sistema automatizado permite estandarizar el proceso y reducir la dependencia de cálculos manuales, lo que disminuye significativamente la probabilidad de errores. También facilita la generación de reportes para auditorías internas, análisis de costos laborales y toma de decisiones administrativas. En Titanium Fitness, este tipo de herramienta asegura que la información del personal se procese de manera coherente y confiable, permitiendo que la administración mantenga un control efectivo y transparente de todos los componentes de la planilla.

Control de Asistencia

El control de asistencia es el proceso mediante el cual una organización registra y supervisa las horas de trabajo de sus colaboradores, incluyendo entradas, salidas, ausencias y horas extra. En el contexto de la gestión de personal, la administración del tiempo y la asistencia se define como el seguimiento sistemático de las horas laboradas, las ausencias y los horarios programados, con el fin de garantizar registros precisos para la planilla y el cumplimiento de las políticas internas de la empresa (Darwinbox, 2024; Time and attendance, s. f.).

Un sistema de control de asistencia adecuado permite identificar con claridad si los trabajadores cumplen con sus horarios, detectar patrones de retrasos o ausencias recurrentes y contar con información confiable para la toma de decisiones. Además, contribuye a la transparencia en la relación laboral, ya que tanto la administración como los colaboradores pueden respaldar las horas registradas en caso de dudas o revisiones internas.

Desde la perspectiva de la gestión de nómina, el control de asistencia es un insumo fundamental. A partir de estos registros se calculan salarios, recargos por horas extra, deducciones por ausencias injustificadas y otros conceptos asociados a la jornada de trabajo. Cuando este proceso se realiza de forma manual, aumenta el riesgo de errores de digitación o interpretación, lo que puede derivar en pagos incorrectos y en posibles conflictos laborales.

Por esta razón, muchas organizaciones optan por sistemas digitalizados que automatizan el registro de entradas y salidas y almacenan la información en una base de datos centralizada. En el caso del gimnasio Titanium Fitness, integrar un módulo de control de asistencia dentro del sistema desarrollado permite registrar las horas trabajadas de cada colaborador, así como sus permisos y ausencias, de manera ordenada y accesible. Esto facilita la generación de la planilla, reduce el trabajo administrativo repetitivo y mejora la trazabilidad de la información relacionada con el tiempo de trabajo.

CAPÍTULO III: MARCO METODOLÓGICO

En este capítulo se presenta el marco metodológico que guía el desarrollo del sistema web de gestión de planilla para el gimnasio Titanium Fitness. Su propósito es explicar de manera clara cómo se llevó a cabo el proceso de investigación, desde la elección del enfoque y los métodos utilizados, hasta las técnicas aplicadas para recopilar y analizar la información. Todo esto con el objetivo de asegurar que el sistema propuesto responda a las necesidades reales del gimnasio y contribuya a mejorar su gestión administrativa.

Enfoques de investigación

Los métodos de investigación son estrategias, procesos o técnicas empleadas para recolectar y analizar información con el propósito de descubrir nuevos conocimientos o comprender mejor un fenómeno. Según Zita (2021), los métodos de investigación son las herramientas que los investigadores utilizan para obtener y analizar datos, entre ellos los cuestionarios, entrevistas, estudios de caso o grupos de enfoque. La elección del enfoque es una decisión esencial que orienta el proceso investigativo y define la naturaleza del estudio, así como el tipo de información recopilada.

Enfoque cuantitativo

El enfoque cuantitativo se centra en la recolección y el análisis de datos numéricos para identificar patrones o comprobar hipótesis. Ortega (2023) explica que este enfoque emplea procedimientos estructurados de recopilación y análisis de datos a través de herramientas estadísticas y matemáticas, con el fin de cuantificar un problema de investigación. Su naturaleza es deductiva, debido a que parte de una teoría para confirmarla o refutarla mediante la observación objetiva de datos medibles.

Enfoque cualitativo

A diferencia del enfoque anterior, el método cualitativo busca comprender la realidad desde una perspectiva subjetiva, analizando experiencias, percepciones y significados. Según *QuestionPro* (s.f.), "...la investigación cualitativa implica recopilar y analizar datos no numéricos para comprender conceptos, opiniones o experiencias, y obtener una comprensión más profunda del problema de investigación" (párr. 1). Esta definición resalta que el propósito central de este enfoque no es medir, sino interpretar la realidad desde la experiencia humana.

Asimismo, Canive (2019) afirma que “la investigación cualitativa se basa en la observación de comportamientos naturales y en la interpretación de significados obtenidos del contexto”. Esto refuerza la idea de que el proceso cualitativo se centra en el entendimiento profundo del fenómeno estudiado, permitiendo que las conclusiones emerjan directamente de los datos recolectados. En consecuencia, este enfoque resulta el más apropiado para analizar las experiencias y percepciones del personal del gimnasio Titanium Fitness respecto a la gestión manual de la planilla.

Enfoque mixto

El enfoque mixto combina los métodos cuantitativo y cualitativo en un mismo estudio. Ortega (2023) indica que este enfoque permite integrar ambos tipos de información cuando se busca una comprensión más completa del fenómeno investigado. Su principal fortaleza radica en la triangulación de datos: los resultados cuantitativos identifican tendencias generales, mientras que los cualitativos explican en profundidad las causas y significados de dichas tendencias.

Enfoque de investigación seleccionado

Para este proyecto, se ha seleccionado el enfoque cualitativo. La razón principal para esta elección es que el objetivo inicial de la investigación no es medir variables numéricas, sino comprender a fondo la problemática y las necesidades del personal del Gimnasio Titanium Fitness en relación con el proceso manual de pago de planilla. Para respaldar esta elección, se recurre a autores como Hernández-Sampieri y Mendoza (2023), quienes explican que el enfoque cualitativo busca "comprender un fenómeno social complejo" dentro de su contexto natural, enfocándose en las perspectivas y significados que los propios participantes le otorgan a sus experiencias. Esta definición se alinea perfectamente con la necesidad de entender a fondo la problemática del gimnasio desde la visión de quienes la viven a diario, permitiendo capturar una realidad subjetiva que los datos numéricos no podrían revelar.

Este enfoque se materializó a través de técnicas como entrevistas abiertas con el encargado de la planilla y la observación directa de los procedimientos manuales. Estos métodos permitieron obtener una comprensión profunda de los errores, limitaciones y cuellos de botella del proceso actual, identificando los puntos críticos que el nuevo sistema debe solucionar. Además, la flexibilidad del método cualitativo resulta esencial en la etapa de

levantamiento de requerimientos, ya que permite adaptar las preguntas y profundizar en temas que surgen espontáneamente, como la gestión de permisos o el cálculo de horas extra. Esta adaptabilidad es crucial para un proyecto de desarrollo de *software* a la medida, garantizando que la solución final responda a las verdaderas y, a veces, latentes necesidades del gimnasio.

Tipos de investigación

La investigación también puede clasificarse según su propósito o nivel de profundidad. De acuerdo con la Universidad de La Sabana (s. f.), una investigación es un proceso sistemático mediante el cual una o más personas realizan procedimientos para generar conocimiento. A continuación, se presentan tres tipos de investigación relevantes para este proyecto.

Investigación exploratoria

La investigación exploratoria constituye el primer acercamiento científico a un problema. Se utiliza cuando este no ha sido suficientemente estudiado, con el fin de obtener una comprensión inicial del tema e identificar variables o factores clave (Universidad de La Sabana, s. f.). No busca conclusiones definitivas, sino sentar las bases para estudios posteriores.

Investigación descriptiva

La investigación descriptiva se aplica cuando se pretende caracterizar un fenómeno y detallar sus propiedades principales. Según la Universidad de La Sabana (s. f.), este tipo de estudio describe cómo es y cómo se manifiesta una situación concreta, ofreciendo una visión detallada de la realidad sin centrarse necesariamente en sus causas.

Investigación aplicada (o tecnológica)

La investigación aplicada tiene como propósito resolver un problema específico mediante la aplicación práctica del conocimiento existente. En el ámbito de la ingeniería y la informática, este tipo de investigación se traduce en el desarrollo de productos o sistemas que atienden necesidades reales. A diferencia de la investigación pura, busca generar soluciones tangibles y funcionales más que teorías generales.

Tipo de investigación seleccionado

El tipo de investigación seleccionado para este proyecto es la investigación aplicada, debido a que su finalidad es crear una solución tecnológica concreta que resuelva los problemas administrativos detectados en Titanium Fitness. El propósito no es generar una nueva teoría sobre la gestión de planillas, sino desarrollar un sistema funcional que optimice el cálculo de salarios, horas extras, permisos, vacaciones e incapacidades.

Esta investigación integra los conocimientos teóricos de sistemas de información y desarrollo web con la práctica de ingeniería, aplicándolos a una necesidad real del gimnasio. De esta manera, el éxito del proyecto se midió por la funcionalidad, eficiencia y utilidad del sistema desarrollado, más que por la producción de conocimiento teórico.

Fuentes de información

Para el desarrollo de esta investigación, las principales fuentes de información corresponden a fuentes primarias y secundarias, las cuales permiten obtener una visión completa del contexto actual del gimnasio Titanium Fitness. En el caso de las fuentes primarias, se incluyó la información suministrada directamente por el personal administrativo, principalmente el encargado de la gestión de planillas. A través de entrevistas y observación directa, se recopilaron datos relevantes sobre los procedimientos actuales de cálculo de salarios, control de horas extras, vacaciones y permisos, lo cual sirvió de base para la elaboración del sistema propuesto.

Asimismo, se utilizaron fuentes secundarias compuestas por documentos técnicos, libros, artículos académicos y recursos en línea de instituciones oficiales. Entre ellas se incluyen publicaciones del Ministerio de Trabajo y Seguridad Social (MTSS) y la Caja Costarricense de Seguro Social (CCSS), que proporcionan información sobre legislación laboral y normativa vigente en materia de prestaciones, incapacidades y permisos (MTSS, s.f.-a; CCSS, s.f.). Estas fuentes permitieron fundamentar el diseño del sistema en aspectos legales y administrativos, asegurando el cumplimiento de la normativa costarricense.

De acuerdo con Hernández-Sampieri y Mendoza (2023), las fuentes de información pueden clasificarse según su origen en primarias, secundarias y terciarias. Sin embargo, en este

estudio no se recurrió a fuentes terciarias, debido a que la información requerida proviene directamente del contexto organizacional y de documentación técnica especializada. Según Gallay (2020), “...las fuentes de información son documentos de distintos tipos y orígenes que proveen datos y recursos sobre un tema o área determinada, y según su nivel de información pueden clasificarse como: fuentes primarias, fuentes secundarias y fuentes terciarias” (párr. 2). Este enfoque permitió integrar información empírica y documental, garantizando la validez y relevancia de los datos empleados en el proyecto.

Fuentes de información primaria

Las fuentes de información primaria son aquellas que proporcionan datos originales y directos, obtenidos de primera mano y sin ningún tipo de interpretación previa. Este tipo de fuentes permite conocer con mayor precisión la realidad del entorno en estudio y comprender cómo se desarrollan los procesos en la práctica.

En este proyecto, las fuentes de información primaria se utilizaron para obtener información directa del personal administrativo del gimnasio Titanium Fitness, quienes participan activamente en el registro de asistencia, el control de horas extra y el proceso de pago de planillas. Esta información permitió identificar cómo se realizan actualmente dichos procesos y cuáles son las principales necesidades y problemáticas que deben ser atendidas mediante el sistema propuesto.

De acuerdo con el artículo *Clasificación general de las fuentes de información* (s. f.), “...las fuentes primarias contienen información original que ha sido publicada por primera vez y que no ha sido filtrada, interpretada o evaluada por nadie más” (párr. 1). En este proyecto, este tipo de fuente sirvió como base para definir los requerimientos funcionales del sistema, tales como el registro de asistencia, la gestión de permisos, el cálculo de horas extra y la generación de reportes de planilla, asegurando que la solución se ajuste a la realidad operativa del gimnasio.

Fuentes de información secundarias

Las fuentes de información secundarias corresponden a aquellas que contienen información previamente recopilada, analizada o interpretada por otros autores. Estas fuentes

se emplearon para complementar la información obtenida mediante las fuentes primarias y para sustentar teóricamente el desarrollo del proyecto.

Según el artículo *Clasificación general de las fuentes de información* (s. f.), "...las fuentes secundarias contienen información primaria, sintetizada y reorganizada, y se utilizan para confirmar hallazgos y ampliar el contenido de una investigación" (párr. 2). En el contexto de este trabajo, las fuentes secundarias se utilizaron para respaldar el marco teórico y el marco legal del sistema de gestión de planillas.

En particular, se consultaron documentos y publicaciones del Ministerio de Trabajo y Seguridad Social (MTSS) y de la Caja Costarricense de Seguro Social (CCSS), con el fin de asegurar que el sistema cumpla con la normativa laboral costarricense relacionada con salarios, deducciones, vacaciones, incapacidades y cargas sociales. Asimismo, se emplearon portales especializados como Bizneo HR y BG&A Corporate Attorneys para apoyar la definición de buenas prácticas en la gestión administrativa, las cuales fueron consideradas durante el diseño del sistema propuesto para el gimnasio Titanium Fitness.

Fuentes de Información Terciaria

Las fuentes de información terciarias se caracterizan por reunir y organizar información proveniente de fuentes primarias y secundarias, ofreciendo una visión general que facilita el acceso a contenidos relevantes. Este tipo de fuente se utilizó principalmente como apoyo para la contextualización del proyecto y la localización de referencias confiables.

De acuerdo con el artículo *Clasificación general de las fuentes de información* (s. f.), "las fuentes terciarias son guías físicas o virtuales que contienen información sobre las fuentes secundarias y facilitan el acceso a repertorios de referencia" (párr. 3). En este proyecto, las fuentes terciarias sirvieron para identificar documentos, artículos y recursos relacionados con la gestión de planillas y el desarrollo de sistemas informáticos, los cuales posteriormente se analizaron con mayor profundidad mediante fuentes secundarias.

El uso de este tipo de fuentes permite fortalecer la base documental del proyecto y orientar adecuadamente el desarrollo del sistema de gestión de planillas para el gimnasio Titanium Fitness.

Variables o Unidades de Análisis

En una investigación, las variables o unidades de análisis representan los elementos o características que se seleccionan para ser examinados con el fin de obtener información útil que contribuya al cumplimiento de los objetivos del estudio. Constituyen la base sobre la cual se recopilan, observan y analizan los datos que permiten describir o explicar un fenómeno determinado.

Definir de manera precisa la unidad de análisis desde el inicio del proceso investigativo resulta esencial, debido a que esto orienta la forma como se recolecta la información, los aspectos objeto de estudio y los métodos aplicados para el análisis de esta. Una correcta elección de la unidad de análisis garantiza resultados confiables y válidos, permite generar conclusiones pertinentes a partir de los datos obtenidos. De esta manera, se facilita la interpretación de los hallazgos y la comprensión del problema que se desea abordar.

Según el artículo El protocolo de investigación IV: las variables de estudio (López & López, 2015:

...las variables en un estudio de investigación constituyen todo aquello que se mide, la información que se recolecta o los datos que se obtienen con la finalidad de responder a las preguntas de investigación, las cuales se especifican en los objetivos. Su selección es un componente esencial del protocolo de investigación. (párr. 3)

En este proyecto, las variables corresponden a los factores analizados en relación con la gestión de planillas del gimnasio Titanium Fitness, tales como la precisión en el cálculo de salarios, el control de horas extras, la administración de permisos y el registro de vacaciones. Estas variables permitieron evaluar la eficiencia del sistema propuesto y comprobar si su implementación mejora los procesos administrativos y el manejo de la información del personal.

Variable Conceptual

Una variable conceptual se refiere a una idea teórica o general que se pretende analizar dentro del marco de una investigación. Este tipo de variable describe un concepto abstracto o una característica que no se puede medir de forma directa, pero que sirve como guía para comprender el fenómeno que se estudia. En el caso de este proyecto, las variables conceptuales

permiten abordar aspectos relacionados con la mejora en la gestión administrativa, el control de planillas y la eficiencia operativa del gimnasio Titanium Fitness.

Como indica Tomas (2022):

...las variables conceptuales se basan en definiciones teóricas o de diccionarios especializados que describen la esencia o las características principales de un fenómeno o atributo. Estas definiciones se ajustan a las necesidades prácticas de la investigación para facilitar su análisis y comprensión. (párr. 10)

De esta manera, la variable conceptual en esta investigación representa los elementos teóricos vinculados con la automatización de procesos y la optimización del manejo de la información dentro del sistema propuesto.

Variable Operacional

La variable operacional define los procedimientos e indicadores utilizados para medir una variable dentro de una investigación. Permite transformar una idea teórica en un elemento concreto y medible, estableciendo cómo se debe recopilar la información, qué valores podrá tomar y bajo qué criterios se debe evaluar. Según Arias González (2021), "...la variable operacional especifica los indicadores, las dimensiones y las técnicas que se emplearán para medir de forma objetiva un fenómeno determinado" (p. 45).

En este proyecto, la variable operacional se aplicó para observar aspectos como la eficiencia en el cálculo de planillas, el control de horas extras y la precisión en los registros administrativos del gimnasio Titanium Fitness, a través de indicadores verificables que permitan evaluar el funcionamiento del sistema propuesto.

Variable Instrumental

La variable instrumental se relaciona con los medios, recursos o herramientas que se utilizan dentro de una investigación para obtener información sobre la variable principal. Su función es apoyar la medición de los datos, definiendo cómo se debe recopilar la información necesaria para su análisis. Estos instrumentos pueden variar según el tipo de estudio y de los objetivos que se busquen alcanzar.

Según el artículo Variables instrumentales: definición y ejemplos (2021):

una variable instrumental es una tercera variable introducida en el análisis de regresión que está correlacionada con la variable predictora, pero no correlacionada con la variable de respuesta. Al usar esta variable, es posible estimar el verdadero efecto causal que alguna variable predictora tiene sobre una variable de respuesta. (párr. 3)

Tabla 4. Unidad de Análisis

Objetivos	Variable	Variable Conceptual	Variable Operacional	Variable Instrumental
Analizar los requerimientos funcionales y técnicos necesarios para el desarrollo del sistema de gestión de planilla del gimnasio Titanium Fitness.	Requerimientos del sistema	Según Pressman (2020), “los requerimientos funcionales y no funcionales representan las necesidades específicas que debe cumplir un sistema para garantizar su correcto desempeño y satisfacer las expectativas del usuario final” (párr. 4).	Entrevistas al personal administrativo y observación directa de los procesos actuales de gestión de planilla.	Guía de entrevistas Guía de observación
Diseñar la estructura del sistema web, incluyendo módulos para el registro de asistencia, cálculo de salarios y generación de reportes.	Diseño del sistema web	Según Sommerville (2019), “el diseño de <i>software</i> consiste en definir la arquitectura, los componentes y la interfaz del sistema, asegurando la coherencia entre los módulos y la facilidad de mantenimiento” (párr. 3).	Elaboración de diagramas de flujo, diagramas entidad–relación y modelos estructurales del sistema web.	Draw.io Visual Studio Code MySQL Workbench
Programar el sistema utilizando tecnologías web como HTML, CSS, PHP y MySQL.	Implementación del sistema	Según Kendall y Kendall (2020), “la implementación es la fase en la que los programas se desarrollan, prueban y documentan, integrando los componentes del sistema en un entorno operativo funcional” (párr. 2).	Desarrollo de los módulos del sistema (asistencia, salarios y reportes) utilizando las tecnologías definidas.	Visual Studio Code PHP MySQL Server

Probar el funcionamiento del sistema mediante pruebas controladas que aseguren su correcto desempeño según los requerimientos definidos.	Pruebas de funcionamiento	Según Sommerville (2019), “la prueba de <i>software</i> es el proceso de ejecutar un sistema con el propósito de encontrar defectos y verificar que cumple con los requisitos especificados” (párr. 5).	Ejecución de pruebas funcionales, de integración y de rendimiento sobre los módulos desarrollados.	Guía de pruebas funcionales Visual Studio Code
--	---------------------------	---	--	---

Fuente: Elaboración propia.

En el caso de esta investigación, la variable instrumental se asocia con los métodos e instrumentos empleados para evaluar el funcionamiento del sistema de gestión de planillas del gimnasio Titanium Fitness, como entrevistas al personal, formularios de registro y pruebas internas del sistema, que facilitan la obtención de datos precisos y confiables sobre su desempeño.

Instrumentos para la Recolección de Datos

Para garantizar la calidad y fiabilidad de esta investigación, se optó por utilizar la entrevista semiestructurada como principal método para la recolección de información. Las entrevistas se llevaron a cabo de manera presencial con el personal administrativo del gimnasio Titanium Fitness, brindando un espacio adecuado para dialogar sobre los procesos actuales de gestión de planillas, cálculo de salarios y control de horas extras. Este método permitió obtener información detallada sobre la experiencia, las dificultades y las expectativas del personal respecto al sistema actual y al desarrollo del nuevo *software* propuesto.

De acuerdo con Hernández-Sampieri y Mendoza (2022), la entrevista “...consiste en una conversación profesional con propósito, en la cual el investigador busca obtener información relevante y significativa por medio de la interacción directa con los participantes” (p. 128). En el enfoque cualitativo, esta técnica resulta esencial, porque facilita la exploración de percepciones, actitudes y opiniones, favoreciendo un diálogo abierto que aporta datos profundos sobre la realidad estudiada.

Asimismo, se aplicó la técnica de observación directa, que complementó los resultados obtenidos en las entrevistas. Esta herramienta fue útil para analizar de manera práctica cómo se desarrollan las tareas administrativas, cómo se registran las horas laborales y de qué forma se procesan los pagos. Según Ortega (2023), "...la observación es una técnica utilizada en diversos ámbitos del conocimiento, que implica la recolección sistemática de información sobre un fenómeno o situación determinada mediante la percepción y registro de datos a través de los sentidos" (párr. 2). Gracias a esta técnica, se logró obtener una visión más completa de los procesos internos, validando los requerimientos necesarios para el diseño del sistema de planillas propuesto.

Proceso para la Recolección y Análisis de Datos

Para garantizar la calidad y fiabilidad de esta investigación, se optó por utilizar la **entrevista semiestructurada** y la **observación directa** como principales métodos para la recolección de información. Ambos instrumentos se aplicaron con el propósito de obtener datos cualitativos que reflejen la experiencia y percepción del personal administrativo del gimnasio Titanium Fitness respecto al sistema actual de pago de planillas y los requerimientos para el nuevo *software* propuesto.

Entrevista semiestructurada

Observar y evaluar los procesos administrativos relacionados con la gestión de planilla en el gimnasio Titanium Fitness, con el fin de identificar prácticas actuales, detectar oportunidades de mejora y recopilar información relevante para el desarrollo del sistema web propuesto.

La entrevista semiestructurada se utilizó como el instrumento principal para recolectar información relevante sobre los procesos administrativos del gimnasio. Se realizaron de forma presencial con el personal encargado de la gestión de planillas, cálculo de salarios y control de horas extras, brindando un espacio de diálogo donde los participantes pudieron expresar sus experiencias, dificultades y expectativas sobre la implementación del nuevo sistema.

De acuerdo con Hernández-Sampieri y Mendoza (2022), la entrevista "consiste en una conversación profesional con propósito, en la cual el investigador busca obtener información relevante y significativa por medio de la interacción directa con los participantes" (p. 128). En

el enfoque cualitativo, esta técnica resulta esencial, porque facilita la exploración de percepciones, actitudes y opiniones, lo que favorece un diálogo abierto que aporta datos profundos sobre la realidad estudiada.

Observación directa

La técnica de observación directa se aplicó como complemento a las entrevistas, con el fin de analizar de forma práctica cómo se desarrollan las tareas administrativas, cómo se registran las horas laborales y de qué manera se procesan los pagos dentro del gimnasio. Esta herramienta permitió observar de primera mano el funcionamiento del sistema actual, identificando los puntos de mejora en la gestión de la planilla.

Según Ortega (2023), “...la observación es una técnica utilizada en diversos ámbitos del conocimiento, que implica la recolección sistemática de información sobre un fenómeno o situación determinada mediante la percepción y registro de datos a través de los sentidos” (párr. 2). Gracias a esta técnica, se logró obtener una visión más completa de los procesos internos, validando los requerimientos necesarios para el desarrollo del sistema propuesto.

CAPÍTULO IV: ANÁLISIS DE RESULTADOS

El análisis de resultados se realizó mediante la aplicación de una entrevista estructurada a cinco empleados del gimnasio Titanium Fitness. El instrumento estuvo conformado por diez preguntas cerradas de respuesta Sí/No y una pregunta abierta opcional, con el objetivo de evaluar la percepción del personal sobre el proceso actual de gestión de planilla.

Durante la aplicación de la entrevista se indagó sobre diversos aspectos relacionados con la administración de la planilla, tales como la existencia de un proceso definido, la organización en el registro de asistencia, la validación de horas extra, la eficiencia en el cálculo de salarios y deducciones, así como la presencia de errores o inconvenientes en el procedimiento actual. Además, se consultó sobre el tiempo que demanda el proceso, la necesidad de implementar un sistema web automatizado, la disposición del personal para utilizarlo y la importancia de recibir capacitación para su correcto uso.

Asimismo, se utilizó una guía de observación con el propósito de analizar directamente cómo se ejecuta el proceso de gestión de planilla dentro del gimnasio. Esta observación permitió comprender la forma en que se registran las horas trabajadas, cómo se calculan los pagos correspondientes y qué herramientas se emplean actualmente para llevar el control administrativo.

El objetivo de esta etapa fue identificar posibles debilidades en el proceso manual, tales como riesgos de error en los cálculos, duplicidad de información o consumo excesivo de tiempo en tareas administrativas. Con base en los hallazgos obtenidos, se busca fundamentar la necesidad del desarrollo de un sistema web local que permita optimizar la gestión de la planilla, mejorar la precisión de los cálculos y centralizar la información en una base de datos.

Método de la Entrevista

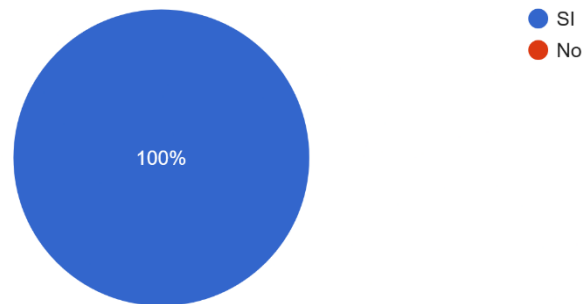
A continuación, se presentan varios gráficos que muestran la información recopilada durante las entrevistas:

Figura 1.

Existencia de un proceso definido para la gestión de planilla en Titanium Fitness.

1. ¿Existe actualmente un proceso definido para la gestión de planilla en el gimnasio?

5 respuestas



Fuente: Elaboración propia.

De acuerdo con los resultados obtenidos, el 100% de los empleados entrevistados indicó que actualmente sí existe un proceso definido para la gestión de planilla en el gimnasio. Esto refleja que la organización cuenta con un procedimiento establecido para el cálculo y control de los pagos correspondientes al personal.

Sin embargo, aunque todos los colaboradores reconocen la existencia de dicho proceso, esto no implica necesariamente que este se encuentre automatizado o que sea completamente eficiente. La presencia de un procedimiento manual puede generar dependencia de registros físicos o herramientas básicas, lo cual podría incrementar el riesgo de errores y el tiempo invertido en tareas administrativas.

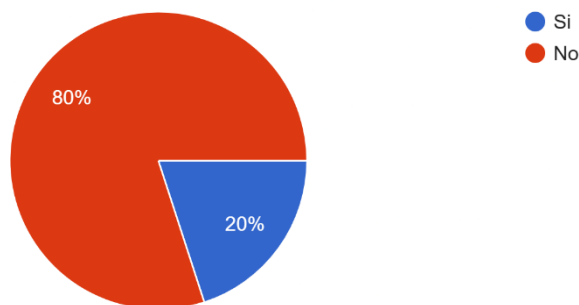
A partir de este resultado, se evidencia que el gimnasio dispone de una base organizativa en cuanto a la gestión de planilla; sin embargo, surge la oportunidad de fortalecer y modernizar este proceso mediante la implementación de un sistema web local que permita optimizar los cálculos, centralizar la información y mejorar el control administrativo.

Figura 2.

Percepción sobre la organización del registro de asistencia en Titanium Fitness.

2.¿El registro de asistencia del personal se realiza de forma organizada y consistente?

5 respuestas



Fuente: Elaboración propia.

De acuerdo con la Figura 2, la mayoría de los empleados, específicamente el 80% (4 de 5), considera que el registro de asistencia no se realiza de forma organizada y consistente. Únicamente un 20% (1 de 5) indicó que percibe un control adecuado en este proceso.

Estos resultados permiten identificar una debilidad importante dentro de la gestión administrativa del gimnasio, ya que el control de asistencia constituye la base para el cálculo correcto de salarios, horas extra y otros rubros asociados a la planilla. La falta de organización en este registro puede provocar inconsistencias en los datos y aumentar el riesgo de errores en los pagos.

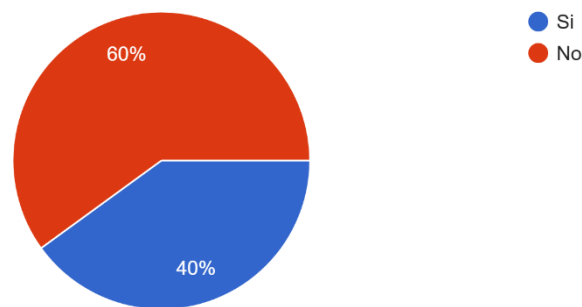
Con base en lo anterior, se evidencia la necesidad de fortalecer el control y la sistematización del registro de asistencia mediante una herramienta tecnológica que permita almacenar la información de manera estructurada y segura. La implementación de un sistema web local contribuiría a mejorar la precisión en los cálculos y a optimizar el proceso de gestión de planilla.

Figura 3.

Claridad en el proceso de registro y validación de horas extra en Titanium Fitness.

3. ¿El proceso de registro y validación de horas extra se encuentra claramente establecido?

5 respuestas



Fuente: Elaboración propia.

Según los resultados mostrados en la Figura 3, el 60% de los empleados entrevistados (3 de 5) considera que el proceso de registro y validación de horas extra no se encuentra claramente establecido. Por su parte, el 40% (2 de 5) indicó que sí percibe un procedimiento definido para este control.

Estos datos reflejan que existe una percepción dividida entre los colaboradores; sin embargo, la mayoría evidencia que no hay un mecanismo formal o suficientemente claro para gestionar las horas extra. Esta situación puede generar inconsistencias en el registro de información y posibles errores en el cálculo de pagos adicionales.

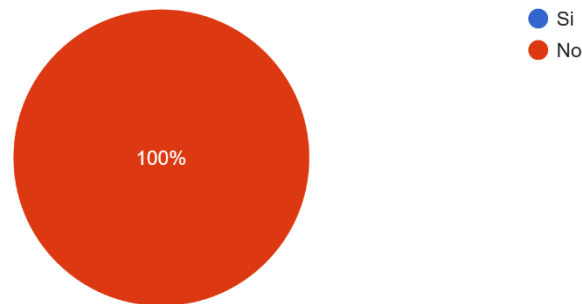
A partir de los resultados obtenidos, se hace evidente la importancia de fortalecer el control interno relacionado con el registro de horas extra, por medio de procedimientos más claros y estructurados. La incorporación de una herramienta tecnológica permitiría formalizar este proceso, garantizando que cada hora adicional trabajada sea registrada, revisada y procesada correctamente antes de su inclusión en la planilla. De esta manera, un sistema web local contribuiría a optimizar la gestión administrativa, reducir inconsistencias y brindar mayor seguridad en el cálculo de los pagos correspondientes.

Figura 4.

Percepción sobre la eficiencia en el cálculo de salarios y deducciones en Titanium Fitness.

4. ¿Considera que el cálculo de salarios y deducciones se realiza de manera eficiente?

5 respuestas



Fuente: Elaboración propia.

De acuerdo con la Figura 4, el 100% de los empleados entrevistados (5 de 5) considera que el cálculo de salarios y deducciones no se realiza de manera eficiente. Ninguno de los participantes manifestó que el proceso actual cumpla con criterios adecuados de eficiencia.

Este resultado evidencia una debilidad significativa en la gestión administrativa del gimnasio, ya que el cálculo salarial constituye uno de los procesos más sensibles dentro de la planilla. La percepción generalizada de ineficiencia sugiere posibles demoras, errores en los cálculos o falta de automatización en el procedimiento actual.

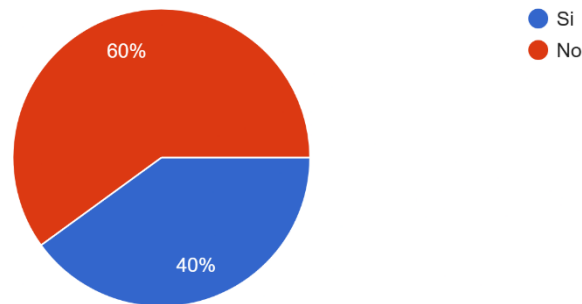
A partir de estos resultados, se evidencia la urgencia de optimizar el proceso de cálculo salarial dentro del gimnasio, incorporando herramientas tecnológicas que reduzcan la dependencia de procedimientos manuales. La implementación de un sistema web local permitiría estandarizar los cálculos, asegurar la correcta aplicación de deducciones y generar mayor control sobre la información procesada. De esta forma, se contribuiría a fortalecer la transparencia y confianza en la gestión de la planilla.

Figura 5.

Experiencia de errores o inconvenientes en el proceso actual de planilla en Titanium Fitness.

5. ¿Ha experimentado errores o inconvenientes en el proceso actual de planilla?

5 respuestas



Fuente: Elaboración propia.

De acuerdo con la Figura 5, el 40% de los empleados entrevistados (2 de 5) manifestó haber experimentado errores o inconvenientes en el proceso actual de planilla. Por otro lado, el 60% (3 de 5) indicó no haber tenido problemas relacionados con este procedimiento.

Aunque la mayoría señala no haber enfrentado inconvenientes, el hecho de que una parte significativa del personal haya experimentado errores evidencia que el proceso no es completamente confiable. En materia de gestión de planilla, incluso un porcentaje reducido de inconsistencias puede afectar la percepción de transparencia y confianza en los pagos.

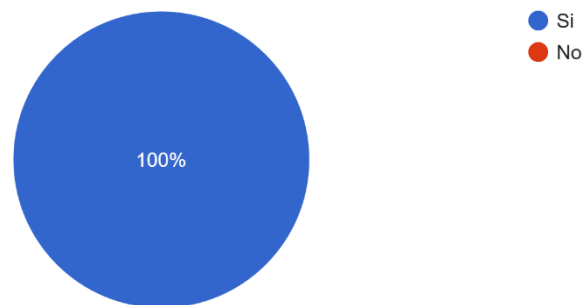
A partir de los hallazgos obtenidos, se identifica la necesidad de fortalecer los mecanismos de control en el proceso de elaboración de la planilla, con el fin de disminuir la probabilidad de inconsistencias. La incorporación de un sistema web local permitiría establecer validaciones automáticas en los cálculos y mantener un registro ordenado de la información, facilitando la detección temprana de posibles errores. De esta manera, se contribuiría a consolidar un proceso más seguro, transparente y alineado con las buenas prácticas administrativas.

Figura 6.

Percepción sobre el tiempo invertido en el proceso actual de gestión de planilla en Titanium Fitness.

6. ¿El proceso actual de gestión de planilla consume más tiempo del necesario?

5 respuestas



Fuente: Elaboración propia.

Según los resultados presentados en la Figura 6, el 100% de los empleados entrevistados (5 de 5) considera que el proceso actual de gestión de planilla consume más tiempo del necesario. Ninguno de los participantes indicó que el procedimiento sea ágil o adecuado en términos de tiempo.

Este resultado evidencia que el proceso vigente presenta ineficiencias operativas, probablemente relacionadas con la utilización de métodos manuales o registros no automatizados. Cuando una actividad administrativa demanda más tiempo del requerido, se incrementa la carga de trabajo y se reduce la productividad en otras funciones del gimnasio.

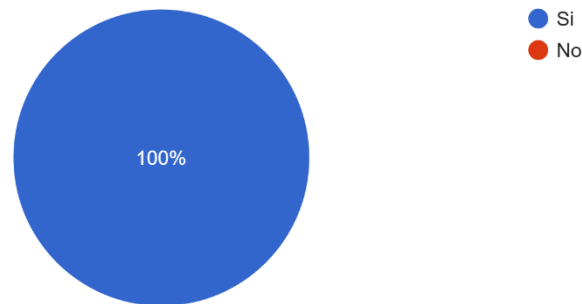
A partir de estos resultados, se hace necesario replantear la forma en que se gestiona actualmente la planilla, incorporando herramientas que permitan agilizar las tareas administrativas y optimizar los tiempos de ejecución. La implementación de un sistema web local facilitaría la automatización de los cálculos y el registro de información, disminuyendo la carga operativa y permitiendo que el personal administrativo pueda dedicar mayor tiempo a otras funciones estratégicas dentro del gimnasio. De esta manera, se promovería una gestión más eficiente y organizada.

Figura 7.

Necesidad de implementar un sistema web para la gestión de planilla en Titanium Fitness.

7.¿Considera necesaria la implementación de un sistema web para la gestión de planilla?

5 respuestas



Fuente: Elaboración propia.

De acuerdo con la Figura 7, el 100% de los empleados entrevistados (5 de 5) considera necesaria la implementación de un sistema web para la gestión de planilla. Ninguno de los participantes manifestó que el proceso actual deba mantenerse sin modificaciones tecnológicas.

Este resultado refleja una aceptación total por parte del personal hacia la modernización del proceso administrativo. La percepción generalizada de necesidad sugiere que los colaboradores reconocen las limitaciones del sistema actual y visualizan en una herramienta web una oportunidad de mejora en términos de organización, precisión y eficiencia.

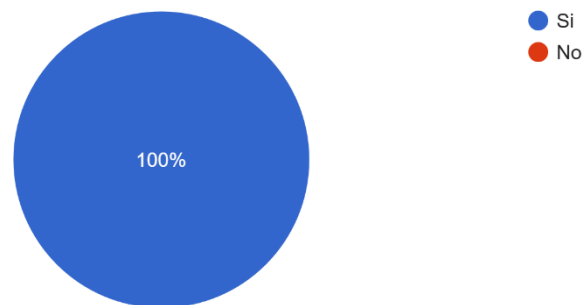
A partir de estos resultados, se consolida el fundamento práctico de este proyecto, al evidenciar que la propuesta responde a una necesidad real identificada por el personal del gimnasio. La implementación del sistema web local no surge únicamente como una iniciativa tecnológica, sino como una solución orientada a mejorar la organización, el control y la eficiencia en la gestión de la planilla. De esta manera, el desarrollo planteado en esta investigación se alinea directamente con las expectativas y requerimientos detectados en el diagnóstico realizado.

Figura 8.

Percepción sobre la reducción de errores mediante un sistema automatizado en Titanium Fitness.

8. ¿Cree que un sistema automatizado ayudaría a reducir errores administrativos?

5 respuestas



Fuente: Elaboración propia.

Según los resultados presentados en la Figura 8, el 100% de los empleados entrevistados (5 de 5) considera que un sistema automatizado contribuiría a reducir errores administrativos en la gestión de planilla. No se registraron respuestas negativas en esta consulta.

Este resultado evidencia que el personal reconoce la relación directa entre la automatización de procesos y la disminución de fallos en tareas administrativas. La intervención manual en cálculos y registros suele aumentar el riesgo de equivocaciones, especialmente cuando se manejan múltiples variables como horas trabajadas, deducciones y pagos adicionales.

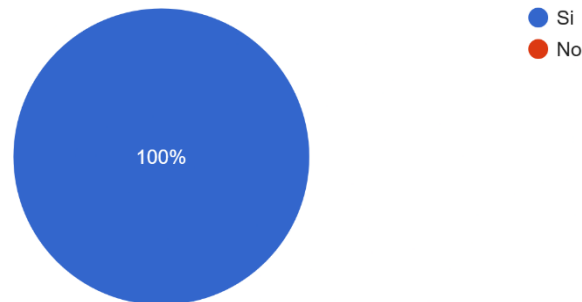
A partir de estos hallazgos, se evidencia que la automatización del proceso de planilla no solo representa una mejora tecnológica, sino una estrategia para fortalecer la calidad y consistencia en la gestión administrativa. La implementación de un sistema web local permitiría estandarizar los procedimientos, reducir la dependencia de cálculos manuales y establecer controles más precisos en cada etapa del proceso. De esta manera, se favorecería una administración más confiable y alineada con buenas prácticas organizacionales.

Figura 9.

Disposición del personal para utilizar un nuevo sistema web de gestión de planilla en Titanium Fitness.

9. ¿Estaría dispuesto(a) a utilizar un nuevo sistema web para la gestión de planilla?

5 respuestas



Fuente: Elaboración propia.

De acuerdo con la Figura 9, el 100% de los empleados entrevistados (5 de 5) manifestó estar dispuesto(a) a utilizar un nuevo sistema web para la gestión de planilla. No se registraron respuestas negativas en esta consulta.

Este resultado refleja una actitud positiva del personal hacia la incorporación de herramientas tecnológicas en el proceso administrativo del gimnasio. La disposición total para utilizar el sistema propuesto facilita su futura implementación, ya que reduce la resistencia al cambio y favorece la adaptación a nuevas metodologías de trabajo.

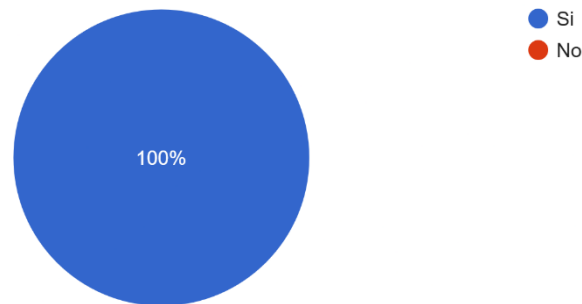
A partir de estos resultados, se evidencia que las condiciones internas del gimnasio son favorables para la implementación del sistema propuesto. La apertura del personal hacia el uso de una herramienta digital facilita el proceso de transición y reduce posibles obstáculos asociados al cambio organizacional. En este sentido, la aceptación manifestada por los colaboradores respalda la factibilidad operativa del proyecto y contribuye a garantizar una adopción efectiva del sistema web local.

Figura 10.

Necesidad de capacitación para el uso del nuevo sistema web en Titanium Fitness.

10. ¿Considera necesaria una capacitación para el uso del nuevo sistema?

5 respuestas



Fuente: Elaboración propia.

De acuerdo con los resultados presentados en la Figura 10, el 100% de los empleados entrevistados (5 de 5) considera necesaria una capacitación para el uso del nuevo sistema web de gestión de planilla. No se registraron respuestas negativas en esta consulta.

Este resultado demuestra que, aunque el personal se encuentra dispuesto a utilizar la nueva herramienta tecnológica, reconoce la importancia de recibir una orientación adecuada para su correcta implementación. La capacitación permitirá asegurar un uso eficiente del sistema, reducir posibles errores durante la etapa inicial y facilitar la adaptación al nuevo proceso automatizado.

A partir de estos resultados, se identifica la necesidad de planificar adecuadamente la etapa de implementación del sistema, incorporando acciones formativas que faciliten su correcta utilización. La capacitación del personal no solo permitirá maximizar el aprovechamiento de las funcionalidades del sistema web local, sino que también contribuirá a fortalecer la seguridad en el manejo de la información y la correcta aplicación de los procedimientos establecidos. De esta manera, se favorecerá una integración efectiva del sistema dentro de la dinámica operativa del gimnasio.

Método de observación

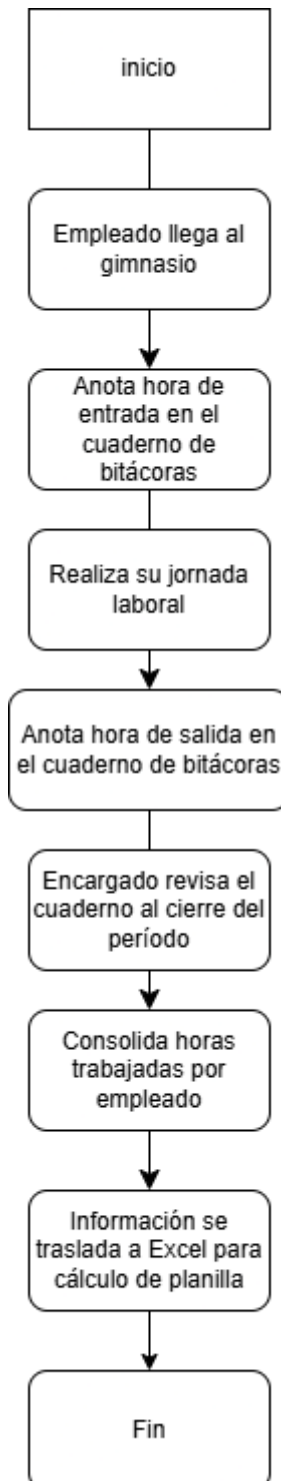
A continuación, se describe la información obtenida mediante la aplicación del método de observación en el gimnasio Titanium Fitness. A través de este método, se analizaron diversos procedimientos relacionados con la gestión administrativa y el cálculo de la planilla. Se identificaron los siguientes procesos:

- Proceso manual de registro de asistencia del personal.
- Proceso manual de registro y validación de horas extra.
- Proceso manual de cálculo de salarios y deducciones.
- Proceso manual de gestión de permisos y ausencias.
- Flujo general del proceso administrativo de planilla.
- Herramientas utilizadas actualmente para el cálculo y control.
- Identificación de errores frecuentes durante el procedimiento.

Estos procesos fueron observados con el objetivo de identificar posibles debilidades, demoras o inconsistencias que puedan afectar la eficiencia y confiabilidad de la gestión de planilla, con el fin de proponer mejoras a través del desarrollo de un sistema web local.

Figura 11.

Proceso manual de registro de asistencia del personal en Titanium Fitness.



Fuente: Elaboración propia

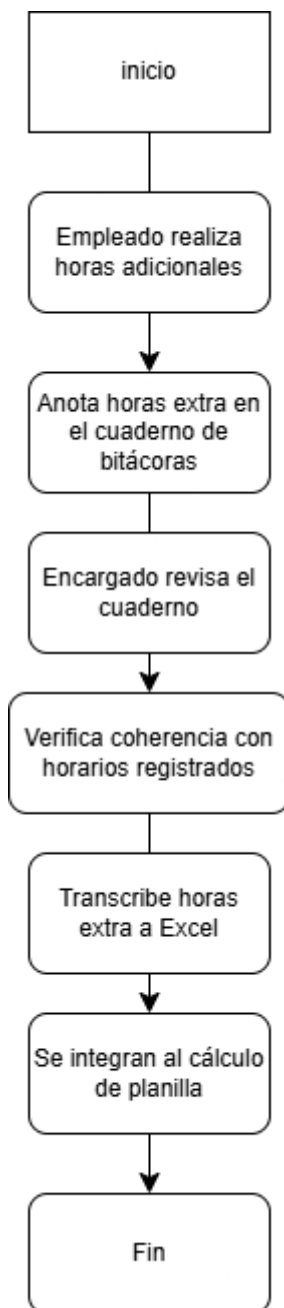
Según la Figura 11, durante la aplicación del método de observación, se identificó que el registro de asistencia del personal en el gimnasio Titanium Fitness se realiza mediante un cuaderno de bitácoras. Cada colaborador anota manualmente su hora de entrada al inicio de la jornada laboral y su hora de salida al finalizarla, dejando constancia física de las horas trabajadas.

Posteriormente, al cierre del período correspondiente, el encargado administrativo revisa el cuaderno y consolida manualmente la información de cada empleado. Estos datos son posteriormente trasladados a una hoja de cálculo en Excel para continuar con el proceso de cálculo de la planilla.

A partir de la observación realizada, se determinó que este procedimiento depende de registros físicos y de la transcripción manual de información, lo que puede generar errores por omisión, dificultades en la lectura de anotaciones o inconsistencias en los horarios registrados. Asimismo, el proceso requiere tiempo adicional para la revisión y traslado de datos. En este contexto, la implementación de un sistema web local permitiría registrar la asistencia de manera digital y centralizada, reducir la duplicidad de tareas y mejorar el control previo al cálculo salarial.

Figura 12.

Proceso manual de registro y validación de horas extra en Titanium Fitness.



Fuente: Elaboración propia

Según la Figura 12, durante la aplicación del método de observación, se identificó que el registro de horas extra en el gimnasio Titanium Fitness se realiza mediante anotaciones manuales en el cuaderno de bitácoras. Cuando un colaborador extiende su jornada laboral, las horas adicionales son registradas en el mismo medio utilizado para el control de asistencia.

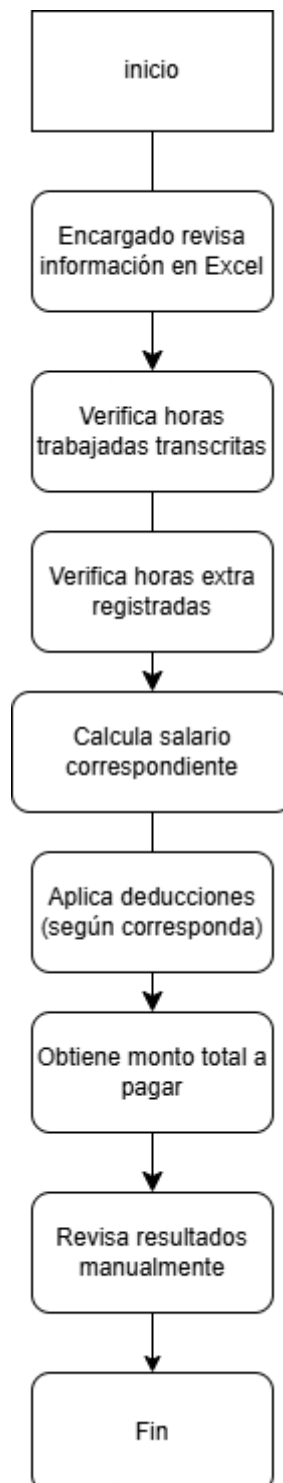
Posteriormente, el encargado administrativo revisa el cuaderno al cierre del período, verificando que las horas extra registradas coincidan con los horarios de entrada y salida

anotados. Una vez confirmada su validez, estas horas son transcritas a una hoja de cálculo en Excel, donde se integran al cálculo salarial correspondiente.

A partir de la observación realizada, se determinó que este procedimiento requiere una revisión manual constante y depende de la correcta anotación en el cuaderno, así como de la transcripción precisa a Excel. Esta doble manipulación de la información puede generar omisiones o errores de digitación que afecten la exactitud del cálculo final. En este contexto, la implementación de un sistema web local permitiría registrar y validar las horas extra de manera estructurada, reduciendo la intervención manual y mejorando el control administrativo.

Figura 13.

Proceso manual de cálculo de salarios y deducciones en Titanium Fitness.



Fuente: Elaboración propia

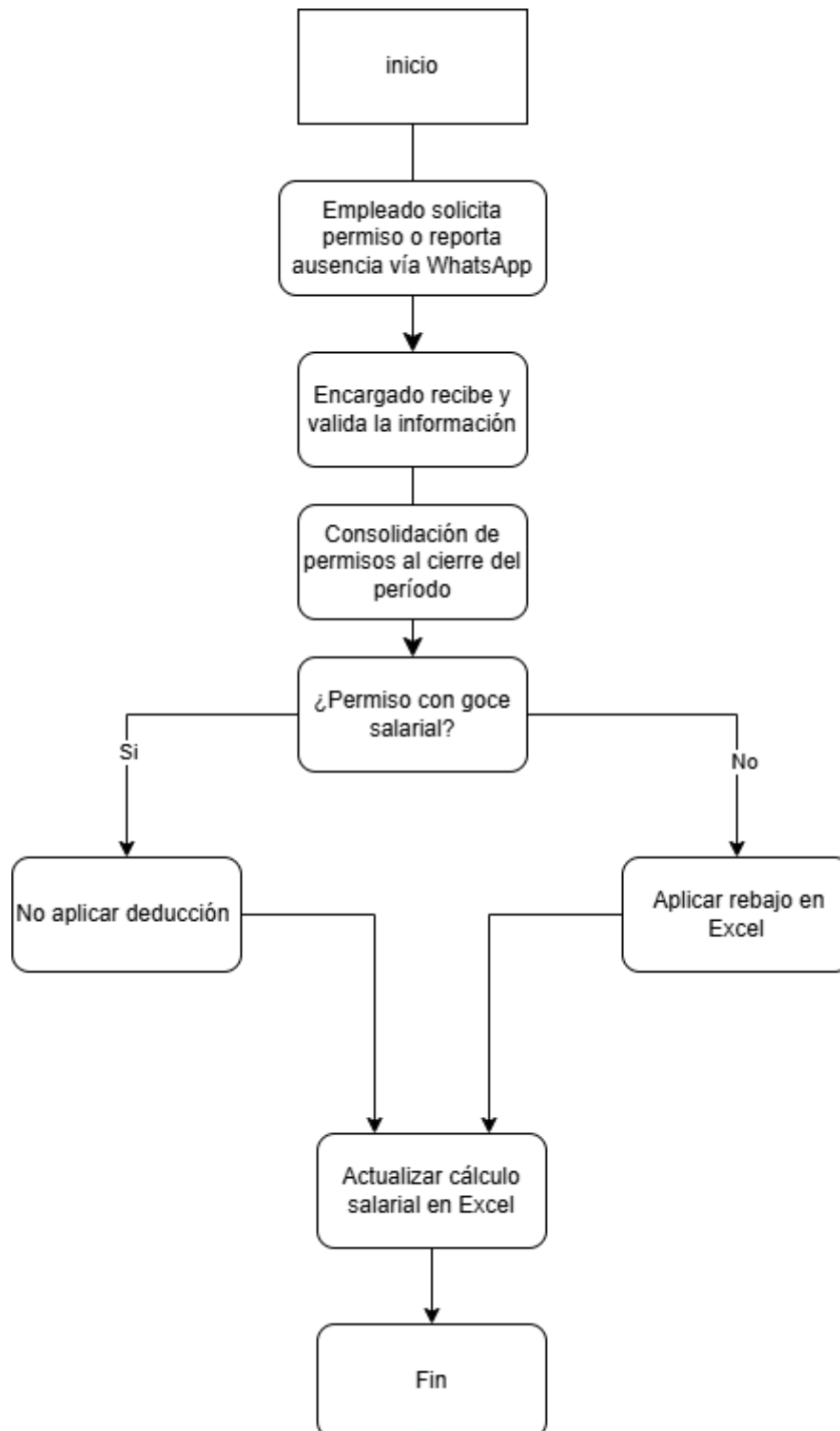
De acuerdo con la Figura 13, durante la aplicación del método de observación se identificó que el cálculo de salarios en el gimnasio Titanium Fitness se realiza mediante una hoja de cálculo en Excel. Una vez transcritas las horas trabajadas y las horas extra desde el cuaderno de bitácoras, el encargado administrativo verifica la información ingresada.

Posteriormente, se realizan los cálculos correspondientes para determinar el salario según las horas trabajadas, incorporando las horas adicionales cuando corresponda y aplicando las deducciones respectivas. Finalmente, se obtiene el monto total por pagar y se revisan los resultados antes de efectuar el pago.

A partir de la observación realizada, se determinó que, aunque Excel facilita el uso de fórmulas para automatizar operaciones matemáticas, el proceso continúa dependiendo de la correcta transcripción y verificación manual de datos. Esta dinámica puede generar errores por digitación, inconsistencias en la información o retrasos en la revisión. En este contexto, la implementación de un sistema web local permitiría integrar el registro de asistencia con el cálculo automático de salarios y deducciones, reduciendo la intervención manual y mejorando la precisión y eficiencia del proceso administrativo.

Figura 14.

Proceso manual de gestión de permisos y ausencias en Titanium Fitness.



Fuente: Elaboración propia

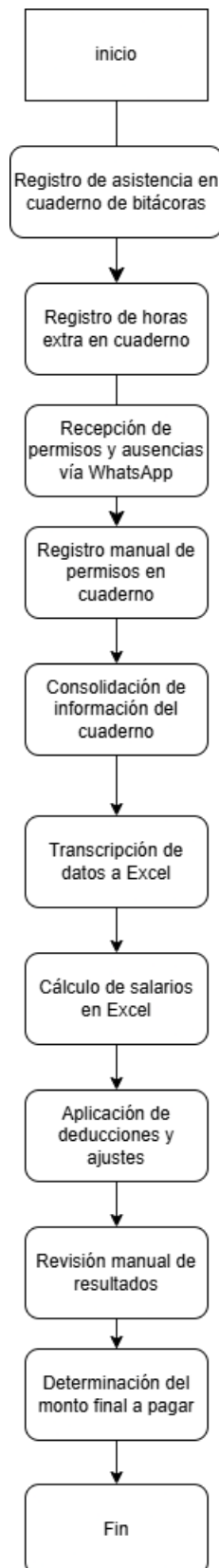
Según la Figura 14, durante la aplicación del método de observación se determinó que la gestión de permisos y ausencias en el gimnasio Titanium Fitness se realiza mediante comunicación a través de la plataforma WhatsApp. El colaborador informa su situación al encargado administrativo, quien recibe el mensaje y registra manualmente la información en el cuaderno de bitácoras utilizado para el control general de asistencia.

Al cierre del período de pago, el encargado consolida los permisos registrados y analiza si estos corresponden a permisos con goce salarial o sin goce salarial. En caso de tratarse de un permiso con goce salarial, no se aplica deducción alguna en el cálculo final. Por el contrario, si el permiso no contempla goce salarial, se aplica el rebajo correspondiente en la hoja de cálculo en Excel, donde se efectúa el cálculo de la planilla.

Desde una perspectiva técnica, se identificó que este procedimiento involucra múltiples medios de registro (mensajería digital, registro físico y hoja de cálculo), lo que genera fragmentación de la información y aumenta el riesgo de inconsistencias durante la transcripción de datos. Asimismo, la falta de integración entre los canales utilizados limita la trazabilidad del proceso y dificulta la automatización de los ajustes salariales. En este contexto, la implementación de un sistema web local permitiría centralizar el registro de permisos y ausencias, integrándolo directamente con el módulo de cálculo de planilla para fortalecer el control administrativo.

Figura 15.

Flujo general del proceso administrativo de planilla en Titanium Fitness.



Fuente: Elaboración propia

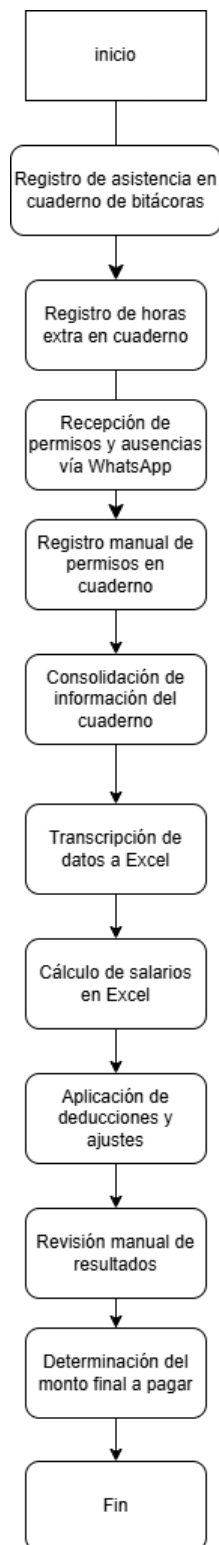
De acuerdo con la Figura 15, durante la aplicación del método de observación se analizó de manera integral el flujo general del proceso administrativo de planilla en el gimnasio Titanium Fitness. El procedimiento inicia con el registro manual de asistencia y horas extra en el cuaderno de bitácoras, así como con la recepción de permisos y ausencias comunicados a través de WhatsApp, los cuales posteriormente son registrados también de forma manual.

Al cierre del período de pago, el encargado consolida toda la información contenida en el cuaderno y transcribe los datos a una hoja de cálculo en Excel. En esta etapa se realizan los cálculos salariales correspondientes, incorporando horas extra y aplicando las deducciones o ajustes según cada caso. Finalmente, se revisan los resultados obtenidos antes de determinar el monto final a pagar a cada colaborador.

Desde una perspectiva técnica, se identificó que el proceso actual se caracteriza por la utilización de múltiples medios de registro y por la dependencia de transcripciones manuales entre sistemas no integrados. Esta dinámica genera fragmentación de la información, aumenta el riesgo de errores por digitación y demanda tiempo adicional en cada período de pago. En este contexto, la implementación de un sistema web local permitiría centralizar el registro de datos, automatizar los cálculos y mejorar la trazabilidad y eficiencia del proceso administrativo de planilla.

Figura 16.

Herramientas utilizadas actualmente para el cálculo y control de planilla en Titanium Fitness.



Fuente: Elaboración propia

De acuerdo con la Figura 16, durante la aplicación del método de observación se identificó que la gestión de planilla en el gimnasio Titanium Fitness se apoya en múltiples herramientas que operan de forma independiente entre sí. El registro de asistencia y de horas extra se realiza mediante un cuaderno de bitácoras, mientras que la gestión de permisos y ausencias se efectúa a través de la aplicación WhatsApp. Posteriormente, la información recopilada es consolidada manualmente y transcrita a una hoja de cálculo en Microsoft Excel, donde se ejecutan los cálculos salariales y la aplicación de deducciones correspondientes.

Desde un enfoque técnico, se determinó que la utilización de herramientas no integradas genera fragmentación de la información y dependencia de procesos manuales para la transferencia de datos entre medios físicos y digitales. Esta dinámica incrementa la probabilidad de errores por digitación, omisiones en el registro y duplicidad de información, además de demandar mayor tiempo administrativo para la consolidación y validación de datos.

En este contexto, la implementación de un sistema web local permitiría centralizar el registro de asistencia, horas extra, permisos y ausencias en una única plataforma, integrando directamente estos datos con el módulo de cálculo de planilla. Esto contribuiría a mejorar la trazabilidad, consistencia y eficiencia del proceso administrativo dentro del gimnasio.

Identificación de riesgos en el procedimiento manual

Durante la aplicación del método de observación no se identificaron fallas críticas en la gestión actual de planilla en el gimnasio Titanium Fitness; sin embargo, se detectaron diversos puntos vulnerables derivados de la naturaleza manual del proceso y de la utilización de herramientas no integradas entre sí.

Entre los principales riesgos identificados se encuentran:

- Dependencia de anotaciones manuales en el cuaderno de bitácoras para el registro de asistencia y horas extra.
- Transcripción manual de información desde el cuaderno hacia la hoja de cálculo en Excel.
- Gestión de permisos y ausencias a través de WhatsApp sin integración automática con el cálculo salarial.
- Necesidad de verificación manual de fórmulas y datos antes de confirmar los pagos.

Desde una perspectiva técnica, estos factores no implican errores constantes, pero sí representan riesgos potenciales que pueden generar inconsistencias en caso de omisión, digitación incorrecta o retraso en la consolidación de datos. Además, la fragmentación de la

información en distintos medios dificulta la trazabilidad y el control integral del proceso administrativo.

En este contexto, la implementación de un sistema web local permitiría centralizar el registro de información, automatizar los cálculos y reducir la intervención manual. De este modo se fortalecería la confiabilidad y eficiencia en la gestión de planilla del gimnasio.

Conclusión

El análisis de resultados desarrollado en este capítulo permitió comprender de manera integral el funcionamiento actual del proceso de gestión de planilla en el gimnasio Titanium Fitness. Mediante la aplicación de la entrevista y la guía de observación, se obtuvo información relevante tanto desde la percepción de los colaboradores como desde la revisión técnica del procedimiento administrativo vigente.

Los resultados evidencian que, aunque el gimnasio cuenta con un proceso establecido para el control de asistencia, registro de horas extra y cálculo salarial, este se apoya en herramientas no integradas y en procedimientos manuales que requieren transcripción constante de información entre distintos medios. Esta dinámica genera riesgos potenciales de inconsistencias, incrementa el tiempo administrativo y limita la eficiencia del proceso.

Asimismo, la entrevista reflejó una percepción generalizada de necesidad de mejora, destacando la disposición del personal hacia la implementación de una solución tecnológica que permita optimizar la gestión de planilla. Estos hallazgos no solo describen la situación actual, sino que proporcionan el sustento técnico y práctico para el desarrollo del sistema web local propuesto en esta investigación.

En este sentido, los resultados obtenidos constituyen la base diagnóstica del Trabajo Final de Graduación, ya que permiten definir con mayor precisión los requerimientos del sistema, orientar el diseño de sus funcionalidades y asegurar que la solución propuesta responda a necesidades reales identificadas en el entorno organizacional de Titanium Fitness.

CAPÍTULO V: PROPUESTA

En este apartado se describen los aspectos fundamentales relacionados con el análisis del sistema propuesto, detallando cómo se llevará a cabo el desarrollo del *software* y su posterior implementación. Se analizan los módulos que conforman el sistema, los requerimientos de *hardware* necesarios, el manejo y almacenamiento de la información, así como el perfil del recurso humano requerido para su correcta operación. Asimismo, se explica la forma en que el sistema será utilizado dentro del gimnasio en situaciones reales, con el fin de garantizar su funcionalidad y viabilidad.

ANÁLISIS:

Análisis detallado del Software por desarrollar:

Gestión de horas extra:

Este módulo permitirá al colaborador realizar solicitudes de horas extra a través del sistema, las cuales serán enviadas a su jefatura inmediata para su revisión. La jefatura tendrá la opción de aprobar o rechazar la solicitud. En caso de aprobación, el sistema notificará al área de recursos humanos y posteriormente al colaborador. Si la solicitud es rechazada, el sistema notificará directamente al colaborador, para mantener un control básico del proceso.

Gestionar nómina:

Este módulo se encargará de gestionar el pago de la nómina de los colaboradores de forma automática, asegurando que los pagos se realicen en el tiempo establecido y de manera correcta. Su propósito es apoyar el proceso administrativo de la empresa mediante un manejo ordenado de la información relacionada con los pagos.

Gestión de vacaciones:

Este módulo permitirá que el colaborador solicite vacaciones a través del sistema, las cuales serán enviadas a su jefatura inmediata para su aprobación o rechazo. En caso de ser aprobadas, el sistema notificará al área de recursos humanos y posteriormente al colaborador. Si la solicitud es rechazada, el sistema notificará al colaborador mediante el mismo medio.

Gestionar incapacidades:

Este módulo administrará los períodos de incapacidad de los colaboradores por motivos de enfermedad o accidente. El sistema permitirá llevar un registro de dichas incapacidades y realizar el cálculo correspondiente, de acuerdo con la normativa costarricense aplicable.

Gestionar aguinaldos:

Este módulo procesará el cálculo del pago de aguinaldos de cada colaborador, conforme a lo establecido por la legislación costarricense. Su finalidad es apoyar el cumplimiento oportuno de esta obligación laboral.

Gestionar deducciones:

Este módulo se encargará de aplicar correctamente las deducciones de ley correspondientes al salario de los colaboradores, basándose en la normativa vigente, con el fin de evitar errores en los montos retenidos.

Gestionar impuesto de renta:

Este módulo será el encargado de realizar el cálculo del impuesto sobre la renta correspondiente a cada colaborador, conforme a lo establecido en la legislación aplicable.

Gestionar Asistencia:

Este módulo permitirá registrar las marcas de entrada y salida de los colaboradores, con el objetivo de llevar un control básico de la asistencia y la puntualidad del personal.

Gestión de impuesto de CCSS:

Este módulo se encargará de gestionar las deducciones de la CCSS de forma automática y basándose en la ley correspondiente.

Gestión de permisos:

Este módulo permitirá que los colaboradores soliciten permisos a través del sistema a su jefatura inmediata. Las solicitudes podrán ser aprobadas o rechazadas, y el sistema notificará al colaborador y al área de recursos humanos según corresponda.

Gestión de liquidaciones:

Este módulo se encargará de gestionar el pago de las liquidaciones de los colaboradores en el tiempo correspondiente, conforme a lo establecido por la normativa legal vigente.

Análisis detallado del hardware requerido, tanto para la programación del prototipo como para poner en producción el sistema desarrollado

Para el desarrollo del prototipo del sistema web de gestión de planilla, es necesario contar con un equipo de cómputo que permita ejecutar de forma adecuada las herramientas utilizadas durante el proceso de programación. Se requiere una computadora de escritorio o portátil con capacidad suficiente para realizar tareas de desarrollo, pruebas y validación del sistema, garantizando un desempeño estable durante el trabajo.

Durante el proceso de desarrollo se utilizan diversas herramientas técnicas que demandan recursos del equipo, por lo que se recomienda disponer de un procesador de gama media y una configuración que permita trabajar de manera fluida. Asimismo, el uso de un editor de código como Visual Studio Code facilita la organización, edición y depuración del código fuente del sistema.

Para la ejecución y prueba del sistema en un entorno local, se emplea XAMPP como servidor local, el cual permite simular el funcionamiento del sistema web antes de su implementación. La gestión de la base de datos se realiza mediante MySQL, por medio de la herramienta phpMyAdmin para la administración y consulta de la información almacenada.

En cuanto a las tareas de documentación y presentación del proyecto, se utiliza la suite Office 365, la cual incluye aplicaciones como Word, Excel y PowerPoint, necesarias para la elaboración de documentación técnica, registro de información y presentación de resultados.

Adicionalmente, se dispone de una conexión a internet con una velocidad aproximada de 300 Mbps, la cual facilita la descarga de actualizaciones de *software*, el acceso a recursos en línea y la realización de pruebas complementarias. Sin embargo, el sistema está diseñado

para funcionar en un entorno local, por lo que su operación no depende de una conexión permanente a internet.

Tabla 5.

Hardware detallado.

Elemento	Detalle	Costo estimado
Computadora de desarrollo (Uso personal)	Equipo de cómputo con procesador de gama media, memoria RAM suficiente y almacenamiento de estado sólido para ejecutar las herramientas de desarrollo.	€0
Plataforma de desarrollo	Visual Studio Code como editor de código y XAMPP como servidor local para pruebas del sistema web.	€0
Gestión de base de datos	MySQL como motor de base de datos, administrado mediante phpMyAdmin.	€0
Herramientas ofimáticas (Uso personal)	Licencia de Office 365 utilizada para la elaboración de documentación y presentaciones del proyecto.	€0
Conexión a internet	Plan Liberty 300Mbps	€30.00 por mes

Fuente: Elaboración propia.

Para la puesta en producción del prototipo funcional del sistema web de gestión de planilla, se consideran los recursos de *hardware* necesarios para garantizar su correcto funcionamiento en un entorno local. Debido a la naturaleza del sistema y a su alcance, no se requiere una infraestructura tecnológica compleja, ya que el sistema está diseñado para ser utilizado como apoyo a los procesos administrativos del gimnasio.

La empresa cuenta con un equipo de cómputo portátil que cumple con las características técnicas necesarias para la ejecución del sistema, permitiendo el acceso al mismo por parte del personal autorizado. El uso de este equipo existente facilita la implementación del prototipo, sin necesidad de realizar inversiones adicionales en *hardware*, factor que contribuye a la viabilidad técnica y económica del proyecto.

Adicionalmente, el sistema puede operar dentro de una red local básica, utilizando el equipo de conectividad provisto por la empresa proveedora del servicio de internet. El acceso

a internet se considera un recurso de apoyo para tareas administrativas generales, sin que el funcionamiento del sistema dependa de una conexión permanente.

Tabla 6.

Hardware requerido para la producción del prototipo

Elemento	Detalle	Costo estimado
Equipo de cómputo	Computadora portátil HP Victus 15 con procesador AMD Ryzen 5, memoria RAM y unidad de estado sólido, utilizada para la ejecución del sistema web en un entorno local.	€369 000
Conectividad de red	Router provisto por la empresa proveedora del servicio de internet para permitir la comunicación en red local.	€0
Sistema operativo	Sistema operativo Windows 11 Pro, incluido en el equipo.	€0
Conexión a internet	Conexión a internet mediante plan Liberty 300 Mbps, utilizada como recurso de apoyo para tareas administrativas generales.	€30 000 por mes

Fuente: Elaboración propia.

Análisis detallado de los elementos relacionados con las telecomunicaciones que requiere el sistema para su funcionamiento:

Los elementos de telecomunicaciones considerados para el funcionamiento del sistema web de gestión de planilla fueron definidos de acuerdo con las necesidades reales del prototipo funcional. Puesto que el sistema está diseñado para operar en un entorno local, no se requiere una infraestructura de telecomunicaciones compleja ni configuraciones avanzadas para su implementación.

La comunicación entre los equipos se realiza mediante una red local básica, utilizando conexión inalámbrica (wifi) o conexión por cable (Ethernet), según la disponibilidad del entorno. Esta configuración permite el acceso al sistema por parte del personal autorizado de manera estable y sencilla.

El sistema se ejecuta de forma local dentro de la red interna de la empresa, permitiendo su operación y acceso desde los equipos autorizados. Para el apoyo a las labores administrativas generales, la empresa cuenta con una conexión a internet mediante un plan Liberty de 300 Mbps, el cual resulta adecuado para dichas actividades y no interfiere con el funcionamiento del sistema.

En cuanto a la configuración de puertos y el uso de equipos especializados, no se requiere ninguna configuración adicional específica, debido a que el sistema se ejecuta en un entorno controlado y de carácter interno. Esta característica contribuye a una implementación sencilla y a la reducción de costos asociados a las telecomunicaciones.

Tabla 7.

Aspectos esenciales de las telecomunicaciones

Elemento	Descripción	Costo estimado
Tipo de red	Red local básica, utilizando conexión inalámbrica (wifi) o conexión por cable (Ethernet).	N/A
Uso de internet	Conexión a internet mediante plan Liberty 300 Mbps, utilizada como recurso de apoyo para tareas administrativas generales.	€30 000 por mes
Puertos	No se requiere configuración específica de puertos para el funcionamiento del sistema.	N/A
Equipos de red	Router provisto por la empresa proveedora del servicio de internet.	€0
Seguridad	Firewall Windows (Incluido en el sistema)	€0

Fuente: Elaboración propia

Descripción detallada de las herramientas técnicas utilizadas para el desarrollo, motor de base de datos:

Para el desarrollo del sistema web de gestión de planilla, se optó por utilizar MySQL como motor de base de datos, debido a su confiabilidad y a su adecuada integración con tecnologías web. Este motor permite el almacenamiento y la gestión estructurada de la

información generada por el sistema, lo que facilita el manejo de los datos relacionados con los colaboradores y los procesos administrativos.

Para la administración de la base de datos se emplea la herramienta phpMyAdmin, la cual permite gestionar las tablas, campos y registros de manera visual y sencilla, apoyando las tareas de desarrollo y prueba del sistema. Esta herramienta resulta adecuada para el alcance del prototipo funcional, ya que permite trabajar de forma eficiente sin incurrir en costos adicionales.

El desarrollo de la lógica del sistema se realiza utilizando el lenguaje de programación PHP, el cual permite procesar la información y establecer la comunicación entre la interfaz del sistema y la base de datos. Para la escritura y organización del código fuente se utiliza Visual Studio Code como editor de desarrollo, debido a su facilidad de uso y compatibilidad con las tecnologías empleadas.

Asimismo, se utiliza XAMPP como entorno de desarrollo local, el cual integra los componentes necesarios para ejecutar y probar el sistema web en un entorno controlado antes de su puesta en producción. Las herramientas utilizadas cuentan con licenciamiento gratuito, lo que contribuye a reducir los costos asociados al desarrollo del proyecto y favorece su viabilidad.

El uso de estas herramientas técnicas resulta suficiente para cumplir con los objetivos del prototipo funcional, lo que permite el desarrollo y validación del sistema sin requerir infraestructuras complejas ni tecnologías especializadas.

Descripción detallada del conocimiento básico que debe tener el recurso humano que operará el sistema

El sistema web de gestión de planilla está dirigido al personal administrativo del gimnasio, que tendrá a su cargo la operación y gestión de la información relacionada con los colaboradores. Para el uso adecuado del sistema, se requiere que el recurso humano cuente con conocimientos básicos en cuanto al uso de computadoras y del sistema operativo Windows.

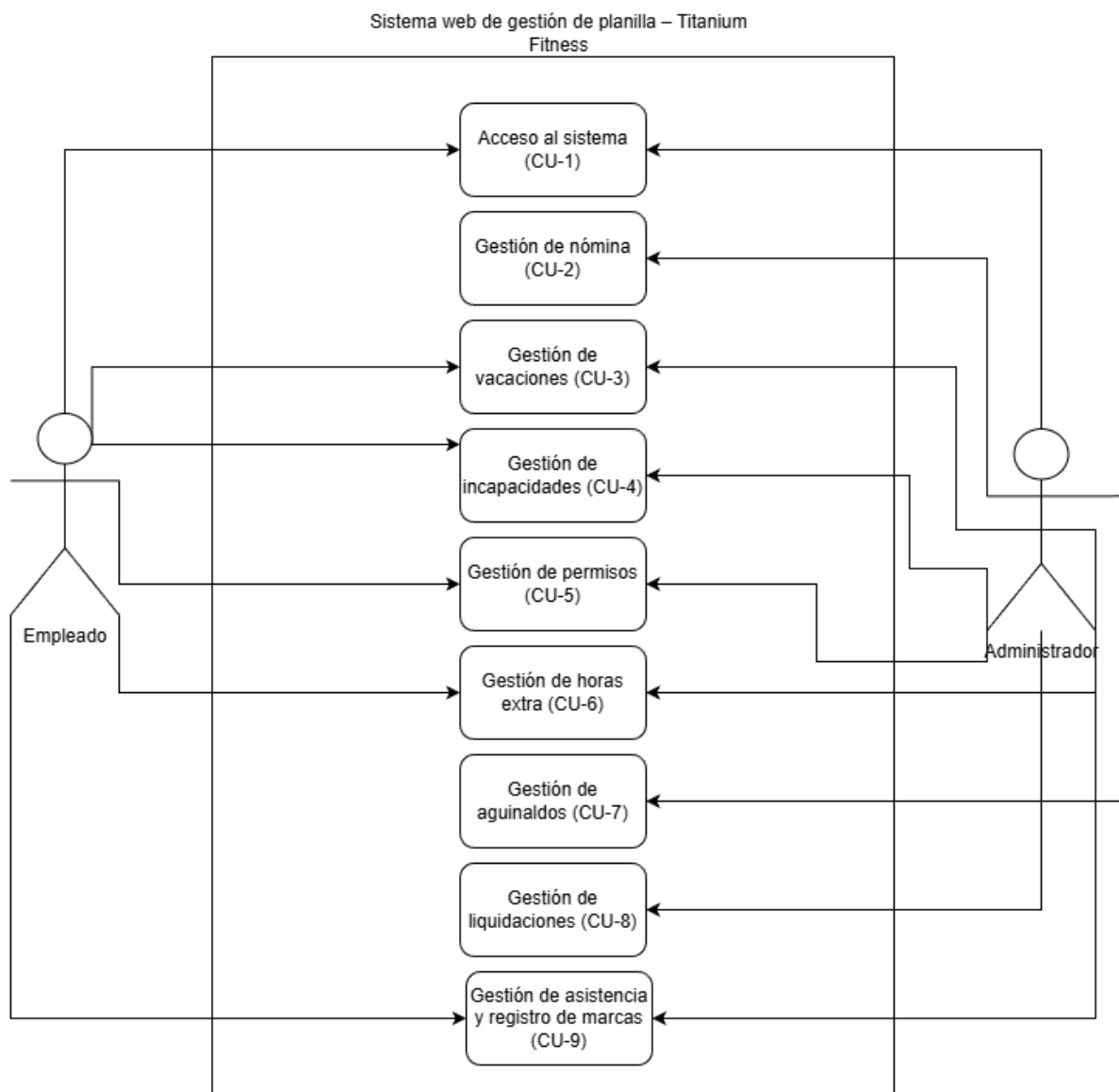
Asimismo, el usuario del sistema debe poseer habilidades básicas para la navegación en entornos web, tales como el uso de internet, el ingreso de información mediante formularios y la consulta de datos en pantalla. Estos conocimientos permiten una interacción sencilla con el sistema y facilitan su adopción en las labores administrativas diarias.

Debido a que el sistema fue diseñado con una interfaz intuitiva y de fácil uso, no se requiere que el personal cuente con conocimientos técnicos avanzados ni experiencia previa en sistemas

de planilla. El uso del sistema se limita a las funciones definidas dentro del alcance del prototipo funcional.

Elaboración de los casos de uso

A continuación, se presentan los casos de uso del sistema, los cuales describen de manera estructurada las principales funcionalidades del prototipo y la forma en que los usuarios interactúan con el sistema durante su operación. Estos casos de uso permiten representar las acciones que el sistema realiza en respuesta a las solicitudes de los usuarios, sirviendo como base para comprender su funcionamiento general.



Fuente: Elaboración propia

Tabla 8.

Caso de uso: Acceso al sistema (Login)

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 1	Nombre del Caso de Uso: Acceso al sistema
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso describe el proceso mediante el cual los usuarios autorizados ingresan al sistema web, validando su identidad a través de credenciales previamente registradas, con el objetivo de acceder a las funcionalidades habilitadas según su rol dentro del sistema.
Autor Caso de uso:	Mauro Pérez Castro
Actores relacionados	Administrador, Colaborador.
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe encontrarse registrado en el sistema con un rol asignado. • El sistema debe estar disponible para su acceso dentro de la red interna.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. El usuario accede a la interfaz de acceso del sistema. 2. El usuario ingresa sus credenciales correspondientes. 3. El sistema procesa la información ingresada y verifica su validez. 4. Si la información es correcta, el sistema habilita el acceso y muestra la pantalla principal correspondiente al rol del usuario. 5. El usuario continúa con el uso normal del sistema. 	
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Verificación de credenciales	<ul style="list-style-type: none"> • El sistema compara las credenciales ingresadas con los datos almacenados en la base de datos para confirmar la identidad del usuario.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Datos de acceso inválidos	<ul style="list-style-type: none"> • Si las credenciales ingresadas no coinciden con un registro válido, el sistema informa al usuario mediante un mensaje. • El usuario puede intentar ingresar nuevamente sus datos.

Requerimientos especiales
<ul style="list-style-type: none">• El acceso al sistema debe realizarse de manera segura para proteger la información del usuario.• La interfaz de acceso debe ser clara y de fácil comprensión.
<i>Postcondiciones</i>
<ul style="list-style-type: none">• Cuando el acceso es exitoso, el usuario puede utilizar las funcionalidades disponibles según su rol.• Si el acceso no es válido, el sistema no permite el ingreso hasta que se proporcionen credenciales correctas.

Fuente: Elaboración propia

Tabla 9.

Caso de uso: Gestión de nómina

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 2	Nombre del Caso de Uso: Gestión de nómina
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso describe el proceso mediante el cual el administrador gestiona la información relacionada con la nómina de los colaboradores del gimnasio, utilizando los datos registrados en el sistema para llevar a cabo el control del pago correspondiente.
Autor Caso de uso:	Mauro Pérez Castro
Actores relacionados	Administrador.
Precondiciones:	<ul style="list-style-type: none"> • El administrador debe haber iniciado sesión en el sistema. • Los colaboradores deben encontrarse registrados en el sistema. • La información necesaria para la gestión de la nómina debe estar disponible en el sistema.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. El administrador accede al módulo de gestión de nómina desde el sistema. 2. El sistema muestra la información disponible de los colaboradores. 3. El administrador selecciona el período de pago correspondiente. 4. El sistema procesa la información registrada para la nómina. 5. El administrador revisa los datos generados. 6. El sistema registra la información de la nómina. 	
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Procesamiento de la información de nómina	<ul style="list-style-type: none"> • El sistema utiliza los datos disponibles para gestionar la información correspondiente a la nómina del período seleccionado.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Información incompleta	<ul style="list-style-type: none"> • Si el sistema detecta que la información requerida no está completa, se muestra un mensaje informativo al administrador. • El administrador puede completar o corregir la información necesaria.

Requerimientos especiales	
<ul style="list-style-type: none">• El acceso a este caso de uso debe estar restringido únicamente al administrador.• El sistema debe presentar la información de manera clara para su revisión.	
<i>Postcondiciones</i>	
<ul style="list-style-type: none">• La información de la nómina queda registrada en el sistema.• El administrador puede continuar con otras tareas relacionadas con la gestión de la planilla.	

Fuente: Elaboración propia

Tabla 10.

Caso de uso: Gestión de vacaciones

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 3	Nombre del Caso de Uso: Gestión de vacaciones
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso permite la gestión de las solicitudes de vacaciones de los colaboradores del gimnasio, mediante el registro de la solicitud en el sistema y su posterior revisión para aprobación o rechazo, de acuerdo con los lineamientos establecidos por la empresa.
Autor Caso de uso:	Mauro Pérez Castro
Actores relacionados	Empleado, jefe y Recursos Humanos.
Precondiciones:	<ul style="list-style-type: none"> • El empleado debe encontrarse registrado en el sistema. • El sistema debe estar disponible para el registro de solicitudes
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. El empleado ingresa al módulo de gestión de vacaciones. 2. El sistema muestra la información relacionada con las vacaciones disponibles del empleado. 3. El empleado registra la solicitud de vacaciones indicando el período deseado. 4. El sistema envía la solicitud al jefe inmediato y al área de Recursos Humanos para su revisión. 5. El sistema registra la aprobación de la solicitud. 6. El sistema notifica al empleado sobre el resultado de la solicitud. 	
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Revisión de solicitud de vacaciones	<ul style="list-style-type: none"> • El jefe inmediato revisa la solicitud registrada por el empleado antes de emitir una decisión.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Solicitud rechazada	<ul style="list-style-type: none"> • Si la solicitud no es aprobada, el sistema registra el rechazo. • El sistema notifica al empleado sobre la decisión tomada.

Requerimientos especiales
<ul style="list-style-type: none">• El sistema debe permitir el registro y seguimiento de solicitudes de vacaciones.
<i>Postcondiciones</i>
<ul style="list-style-type: none">• La solicitud de vacaciones queda registrada con su estado correspondiente.• El empleado queda informado sobre la resolución de su solicitud.

Fuente: Elaboración propia

Tabla 11.

Caso de uso: Gestión de incapacidades

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 4	Nombre del Caso de Uso: Gestión de incapacidades
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso describe el proceso mediante el cual el administrador gestiona la información relacionada con la nómina de los colaboradores del gimnasio, utilizando los datos registrados en el sistema para llevar a cabo el control del pago correspondiente.
Autor Caso de uso:	Mauro Pérez Castro
Actores relacionados	Empleado, Recursos Humanos.
Precondiciones:	<ul style="list-style-type: none"> • El empleado debe encontrarse registrado en el sistema. • El sistema debe estar disponible para el registro de incapacidades.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. El empleado informa la incapacidad correspondiente. 2. Recursos Humanos accede al módulo de gestión de incapacidades. 3. El sistema muestra el formulario de registro de incapacidad. 4. Recursos Humanos registra la información de la incapacidad (fechas y tipo). 5. El sistema almacena la información registrada. 6. El sistema asocia la incapacidad al expediente del empleado. 	
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Registro de incapacidad	<ul style="list-style-type: none"> • Recursos Humanos verifica la información proporcionada y completa el registro de la incapacidad en el sistema.
SF-02 – Actualización de incapacidad	<ul style="list-style-type: none"> • En caso de cambios en el período de incapacidad, el sistema permite actualizar la información registrada.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Información incompleta	<ul style="list-style-type: none"> • En caso de cambios en el período de incapacidad, el sistema permite actualizar la información registrada.
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe permitir registrar y consultar incapacidades de los empleados. 	

- | |
|---|
| <ul style="list-style-type: none">• El acceso a este módulo debe estar restringido a personal autorizado |
| <p style="text-align: center;"><i>Postcondiciones</i></p> |
| <ul style="list-style-type: none">• La incapacidad queda registrada en el sistema.• La información queda disponible para su consulta y control administrativo. |

Fuente: Elaboración propia

Tabla 12.

Caso de uso: Gestión de permisos

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 5	Nombre del Caso de Uso: Gestión de permisos
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso permite gestionar las solicitudes de permisos realizadas por los empleados del gimnasio, mediante el registro de la solicitud en el sistema y su posterior revisión para aprobación o rechazo, según las disposiciones internas de la empresa.
Autor Caso de uso:	Mauro Pérez Castro
Actores relacionados	Empleado, jefe y Recursos Humanos.
Precondiciones:	<ul style="list-style-type: none"> • El empleado debe encontrarse registrado en el sistema. • El sistema debe estar disponible para el registro de incapacidades.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. El empleado accede al módulo de gestión de permisos. 2. El sistema muestra el formulario para el registro de la solicitud. 3. El empleado ingresa la información del permiso solicitado (motivo y período). 4. El sistema envía la solicitud al jefe inmediato y al área de Recursos Humanos para su revisión. 5. El sistema registra la aprobación del permiso. 6. El sistema notifica al empleado sobre el resultado de la solicitud. 	
Sub Flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Revisión de solicitud de permiso	<ul style="list-style-type: none"> • El jefe revisa la solicitud registrada antes de emitir una decisión.
SF-02 – Rechazo de solicitud de permiso	<ul style="list-style-type: none"> • Si la solicitud no es aprobada, el sistema registra el rechazo. • El sistema notifica al empleado sobre la decisión tomada.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Información incompleta	<ul style="list-style-type: none"> • Si la información ingresada no es suficiente, el sistema solicita completar los datos requeridos.

Requerimientos especiales
<ul style="list-style-type: none">• El sistema debe permitir el registro y seguimiento de solicitudes de permisos. El acceso a este módulo debe estar restringido a personal autorizado
<i>Postcondiciones</i>
<ul style="list-style-type: none">• La solicitud de permiso queda registrada con su estado correspondiente.• El empleado queda informado sobre la resolución de su solicitud.

Fuente: Elaboración propia

Tabla 13.

Caso de uso: *Gestión de horas extra*

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 6	Nombre del Caso de Uso: Gestión de horas extra
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso describe el proceso mediante el cual se gestionan las solicitudes de horas extra realizadas por los empleados del gimnasio, permitiendo su registro en el sistema y su posterior revisión para aprobación o rechazo, de acuerdo con los lineamientos establecidos por la empresa.
Autor caso de uso:	Mauro Pérez Castro
Actores relacionados	Empleado, jefe y Recursos Humanos.
Precondiciones:	<ul style="list-style-type: none"> • El empleado debe encontrarse registrado en el sistema. • El sistema debe estar disponible para el registro de solicitudes de horas extra.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. El empleado accede al módulo de gestión de horas extra. 2. El sistema muestra el formulario para el registro de horas extra. 3. El empleado ingresa la información correspondiente a las horas extra realizadas. 4. El sistema envía la solicitud al jefe inmediato y al área de Recursos Humanos para su revisión. 5. El sistema registra la aprobación de las horas extra. 6. El sistema notifica al empleado sobre el resultado de la solicitud. 	
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Revisión de solicitud de horas extra	<ul style="list-style-type: none"> • El jefe revisa la solicitud registrada antes de emitir una decisión.
SF-02 – Rechazo de solicitud de horas extra	<ul style="list-style-type: none"> • Si la solicitud no es aprobada, el sistema registra el rechazo. • El sistema notifica al empleado sobre la decisión tomada.

Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Información incompleta	<ul style="list-style-type: none"> • Si la información ingresada no es suficiente, el sistema solicita completar los datos requeridos.
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe permitir el registro y control de horas extra. 	
<i>Postcondiciones</i>	
<ul style="list-style-type: none"> • La solicitud de horas extra queda registrada con su estado correspondiente. • El empleado queda informado sobre la resolución de su solicitud. 	

Fuente: Elaboración propia

Tabla 14.

Caso de uso: Gestión de aguinaldos

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 7	Nombre del Caso de Uso: Gestión de aguinaldos
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso permite gestionar el proceso de cálculo y registro del aguinaldo correspondiente a los empleados del gimnasio, utilizando la información almacenada en el sistema y de acuerdo con la normativa aplicable.
Autor caso de uso:	Mauro Pérez Castro
Actores relacionados	Recursos Humanos, empleado.
Precondiciones:	<ul style="list-style-type: none"> • El empleado debe encontrarse registrado en el sistema. • El sistema debe contar con la información salarial necesaria para el cálculo del aguinaldo.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. Recursos Humanos accede al módulo de gestión de aguinaldos. 2. El sistema muestra la lista de empleados registrados. 3. Recursos Humanos selecciona el período correspondiente al aguinaldo. 4. El sistema procesa la información disponible para el cálculo del aguinaldo. 5. Recursos Humanos revisa la información generada. 6. El sistema registra el aguinaldo correspondiente a cada empleado. 	
Sub Flujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Cálculo del aguinaldo	<ul style="list-style-type: none"> • El sistema utiliza la información salarial registrada para determinar el monto del aguinaldo correspondiente.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Información incompleta	<ul style="list-style-type: none"> • Si el sistema detecta que la información requerida no está completa, se solicita la revisión de los datos antes de continuar.

Requerimientos especiales
<ul style="list-style-type: none">• El acceso a este módulo debe estar restringido al personal autorizado.
<i>Postcondiciones</i>
<ul style="list-style-type: none">• El aguinaldo queda registrado en el sistema.• La información queda disponible para su consulta administrativa.

Fuente: Elaboración propia

Tabla 15.

Caso de uso: Gestión de aguinaldos

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 8	Nombre del Caso de Uso: Gestión de aguinaldos
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso permite gestionar el proceso de cálculo y registro del aguinaldo correspondiente a los empleados del gimnasio, utilizando la información almacenada en el sistema y de acuerdo con la normativa aplicable.
Autor caso de uso:	Mauro Pérez Castro
Actores relacionados	Recursos Humanos, empleado.
Precondiciones:	<ul style="list-style-type: none"> • El empleado debe encontrarse registrado en el sistema. • El sistema debe contar con la información salarial necesaria para el cálculo del aguinaldo.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. Recursos Humanos accede al módulo de gestión de aguinaldos. 2. El sistema muestra la lista de empleados registrados. 3. Recursos Humanos selecciona el período correspondiente al aguinaldo. 4. El sistema procesa la información disponible para el cálculo del aguinaldo. 5. Recursos Humanos revisa la información generada. 6. El sistema registra el aguinaldo correspondiente a cada empleado. 	
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Cálculo del aguinaldo	<ul style="list-style-type: none"> • El sistema utiliza la información salarial registrada para determinar el monto del aguinaldo correspondiente.
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Información insuficiente	<ul style="list-style-type: none"> • Si el sistema detecta que la información requerida no está completa, se solicita la revisión de los datos antes de continuar.
Requerimientos especiales	
<ul style="list-style-type: none"> • El acceso a este módulo debe estar restringido al personal autorizado 	

<i>Postcondiciones</i>
<ul style="list-style-type: none">• El aguinaldo queda registrado en el sistema.• La información queda disponible para su consulta administrativa.

Fuente: Elaboración propia

Tabla 16.

Caso de uso: Gestión de asistencia y registro de marcas

Prototipo: Prototipo funcional del sistema web para la gestión de planilla del gimnasio Titanium Fitness.	
Número Caso de Uso: 9	Nombre del Caso de Uso: Gestión de asistencia y registro de marcas
Fecha de elaboración:	28/12/2025
Descripción del Caso de Uso	Este caso de uso describe el proceso mediante el cual se registran y gestionan las marcas de entrada y salida de los empleados del gimnasio, permitiendo el control de la asistencia y la puntualidad mediante la información almacenada en el sistema.
Autor Caso de uso:	Mauro Pérez Castro
Actores relacionados	Empleado y administrador.
Precondiciones:	<ul style="list-style-type: none"> • El empleado debe encontrarse registrado en el sistema. • El sistema debe estar disponible para el registro de marcas de asistencia.
Flujo Básico del caso de uso	
Este caso de uso comienza cuando el actor hace algo. Un actor siempre inicia casos de uso. El caso de uso describe lo que el actor hace y lo que el sistema hace en respuesta. Esto es descrito en forma de un diálogo entre el actor y el sistema.	
Detalle del Flujo Básico	
<ol style="list-style-type: none"> 1. El empleado accede al módulo de registro de asistencia. 2. El sistema muestra la opción para registrar marca de entrada o salida. 3. El empleado registra la marca correspondiente. 4. El sistema almacena la marca con la fecha y hora correspondiente. 5. Recursos Humanos puede consultar los registros de asistencia almacenados. 	
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el Flujo Básico.	
SF-01 – Registro de marca de entrada	<ul style="list-style-type: none"> • El sistema registra la hora de ingreso del empleado.
SF-02 – Registro de marca de salida	<ul style="list-style-type: none"> • El sistema registra la hora de salida del empleado
Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA-01 – Error en el registro de marca	<ul style="list-style-type: none"> • Si ocurre un inconveniente durante el registro, el sistema muestra un mensaje informativo y permite reintentar la acción.
Requerimientos especiales	
<ul style="list-style-type: none"> • El sistema debe permitir el registro de marcas de entrada y salida. 	

- | |
|---|
| <ul style="list-style-type: none">• Los registros deben almacenarse de forma segura para su consulta posterior. |
|---|

<p style="text-align: center;"><i>Postcondiciones</i></p>
--

- | |
|--|
| <ul style="list-style-type: none">• La marca de asistencia queda registrada en el sistema.• La información queda disponible para el control administrativo. |
|--|

Fuente: Elaboración propia

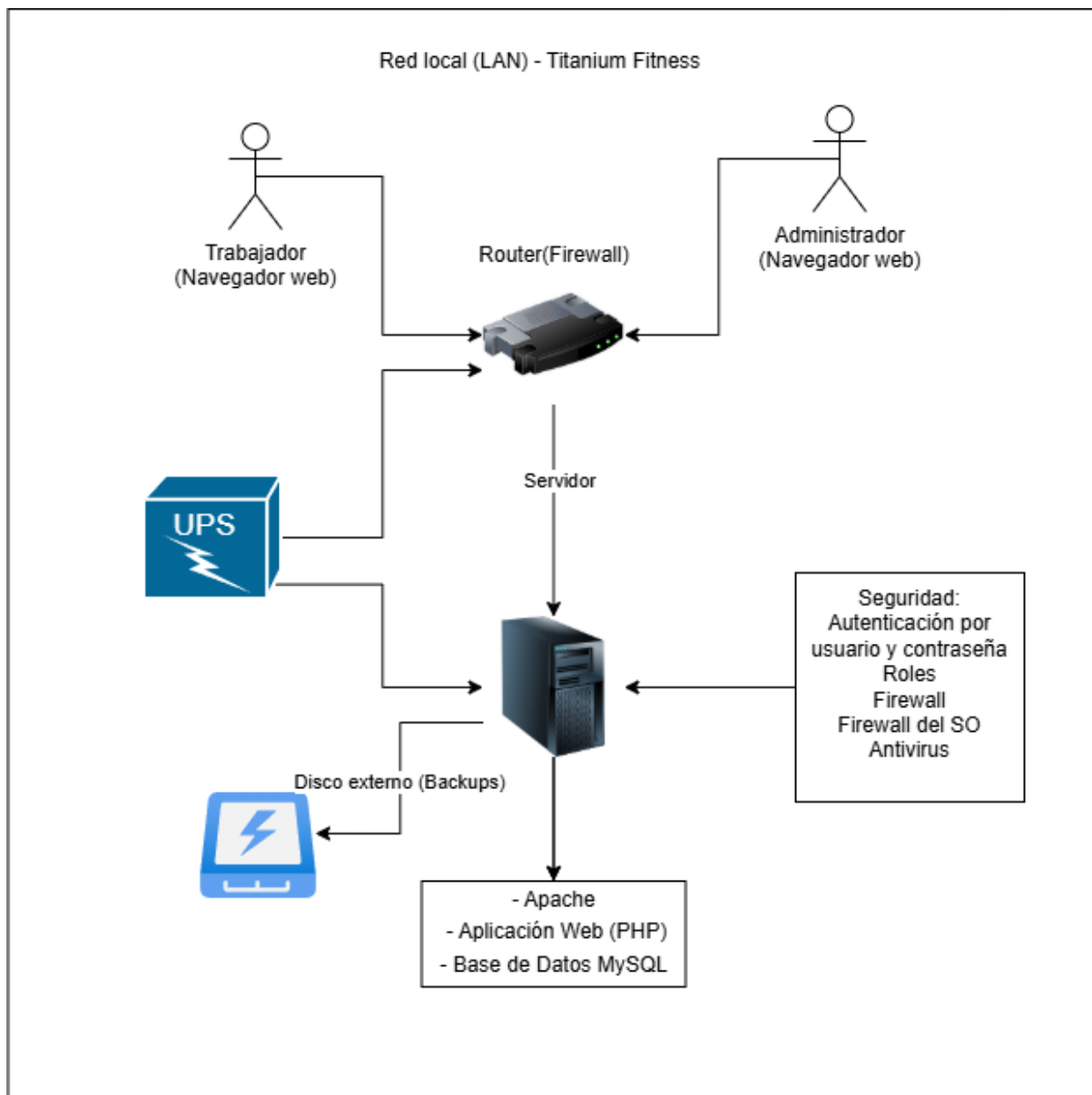
DISEÑO:

A continuación, se describen los elementos principales del prototipo desarrollado para Titanium Fitness, incluyendo aspectos relacionados con su arquitectura, el diseño de interfaces, la organización de la base de datos, los procesos principales y las salidas generadas por el sistema. Además, se incorporan los diagramas utilizados para representar el funcionamiento del prototipo, con el propósito de facilitar la comprensión de su estructura, la interacción entre usuarios y módulos, y su integración dentro del entorno operativo del gimnasio.

Arquitectura del sistema:

En esta sección se presenta la arquitectura del sistema propuesto, la cual corresponde a una solución web de ejecución local dentro de la red interna (LAN) de Titanium Fitness. La arquitectura permite que los usuarios autorizados accedan al sistema mediante un navegador web, mientras que el procesamiento y almacenamiento de la información se realiza en un servidor local.

El servidor utiliza el entorno XAMPP, integrando Apache como servidor web, la aplicación desarrollada en PHP y la base de datos MySQL. Esta configuración permite centralizar la información relacionada con la gestión de planilla y mantener el control de los datos dentro de la organización.

Figura 17.*Arquitectura del sistema.*

Fuente: Elaboración propia

La Figura 17 muestra el modelo cliente-servidor implementado en la red local del gimnasio. El trabajador y el administrador realizan solicitudes desde sus dispositivos conectados a la LAN, las cuales son dirigidas a través del *router* hacia el servidor local. El servidor procesa dichas solicitudes mediante la aplicación en PHP y gestiona el almacenamiento de la información en la base de datos MySQL.

Esta estructura garantiza disponibilidad interna, control administrativo y resguardo de la información sin depender de servicios en la nube

Arquitectura del *software*

En esta sección se describe la organización interna del sistema desarrollado para la gestión de planilla en Titanium Fitness. El prototipo no utiliza un *framework* específico ni un patrón MVC formal, sino que se encuentra estructurado mediante una organización modular basada en archivos PHP independientes por funcionalidad.

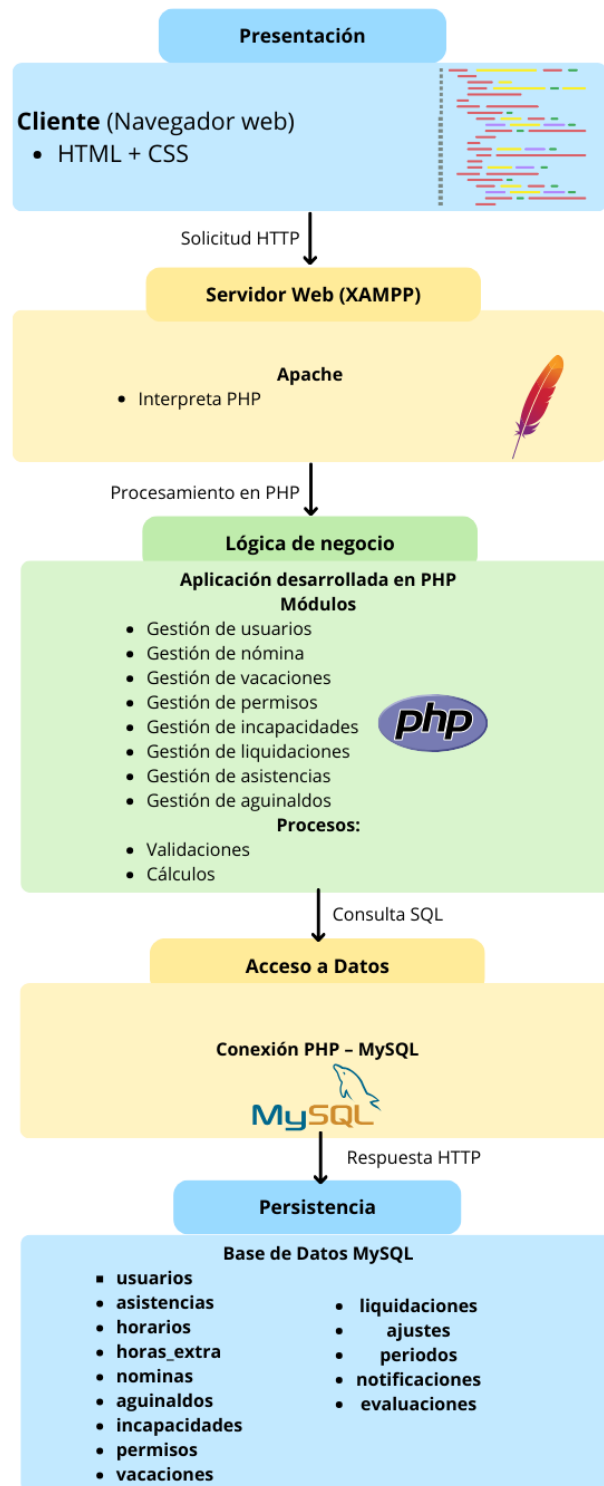
El sistema presenta una estructura simple en la que los archivos principales se encuentran ubicados en el directorio raíz del proyecto. Cada módulo funcional (usuarios, nóminas, vacaciones, incapacidades, permisos, asistencias, liquidaciones, entre otros) está implementado mediante archivos PHP individuales que contienen tanto la lógica de procesamiento como la interacción con la base de datos.

Adicionalmente, el sistema incluye carpetas específicas para la organización de ciertos recursos:

- **Carpeta *assets*:** Contiene los recursos visuales del sistema, como hojas de estilo (CSS) y elementos gráficos utilizados en la interfaz.
- **Carpeta *inc*:** Incluye archivos reutilizables, tales como la conexión a la base de datos y validaciones de sesión.
- **Carpeta *php*:** Contiene archivos auxiliares relacionados con procesos internos del sistema.

La lógica del sistema se desarrolla en PHP, mientras que la presentación utiliza HTML y CSS. La información es almacenada y gestionada en una base de datos MySQL, lo que permite la ejecución de consultas SQL para la manipulación de los datos.

Esta estructura modular permite una organización clara del sistema, facilitando su comprensión y mantenimiento.

Figura 18.*Arquitectura del software*

Fuente: Elaboración propia

La Figura 18 representa el flujo interno de funcionamiento del sistema. El usuario, ya sea administrador o trabajador, realiza una solicitud HTTP desde el navegador web hacia la

aplicación. Dicha solicitud es recibida y procesada por los módulos funcionales implementados en PHP, según la operación requerida.

Durante este proceso, la aplicación utiliza la capa de acceso a datos para establecer la conexión con la base de datos MySQL y ejecutar las consultas necesarias (inserción, actualización, consulta o eliminación de registros). Una vez procesada la información, el sistema genera la respuesta correspondiente y la envía nuevamente al usuario a través de la interfaz web, completando así el ciclo de comunicación cliente-servidor.

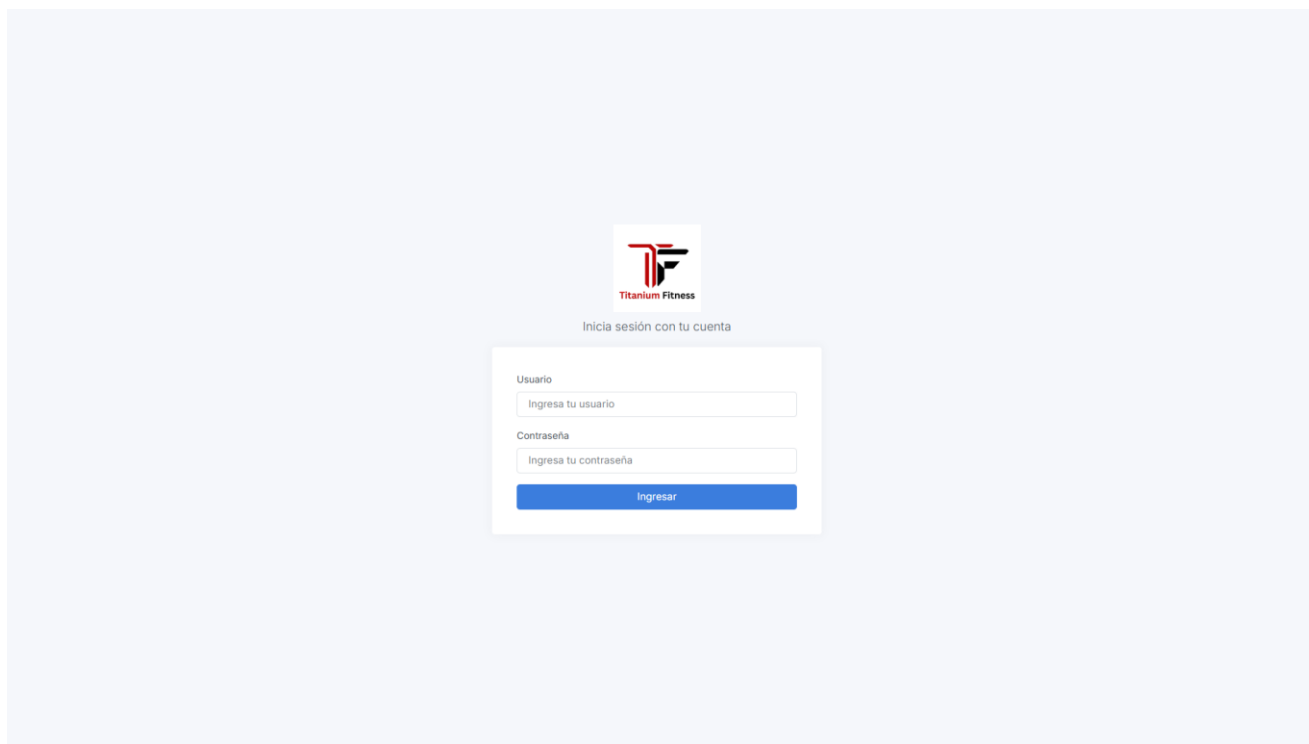
Diseño de entradas y salidas

En esta sección se presenta el diseño de las interfaces del prototipo desarrollado para la gestión de planilla en Titanium Fitness. Las interfaces fueron diseñadas bajo un criterio de simplicidad, claridad visual y coherencia gráfica, manteniendo una estructura uniforme en todas las pantallas del sistema.

Se procuró que los formularios y elementos visuales presenten una distribución ordenada, para facilitar la interacción del usuario y reducir la posibilidad de errores durante el ingreso de datos. Asimismo, se aplicó un diseño consistente en cuanto a colores, tipografía y disposición de botones, con el objetivo de garantizar una experiencia de uso intuitiva.

Figura 19.

Interfaz de login – versión escritorio.



Fuente: Elaboración propia

La interfaz de inicio de sesión permite validar la identidad del usuario antes de acceder al sistema. Esta pantalla constituye el primer punto de interacción entre el usuario y la aplicación.

El formulario está compuesto por los campos de nombre de usuario y contraseña, así como un botón para iniciar sesión. El sistema verifica que las credenciales ingresadas coincidan con los registros almacenados en la base de datos.

En caso de autenticación correcta, el usuario es redirigido al panel principal del sistema según su rol (administrador o trabajador). Si los datos ingresados no son válidos, el sistema muestra un mensaje de error correspondiente, para impedir el acceso no autorizado.

Figura 20.*Interfaz de registro de incapacidades*

The screenshot shows the 'Titanium Fitness' web application interface. On the left is a dark sidebar menu with options: Inicio, Horas extra, Nóminas, Vacaciones, Aguardados, Incapacidades (highlighted), Asistencias, Liquidaciones, Permisos, Mantenimiento, Usuarios, and Ajustes. The main content area is titled 'Incapacidades' and features a 'Crear nuevo' section. This section contains a form with four fields: 'Empleado' (a dropdown menu with 'Seleccionar'), 'Tipo' (a dropdown menu with 'Seleccionar'), 'Fecha inicio' (a date input field with a calendar icon and placeholder 'dd/mm/aaaa'), and 'Fecha fin' (a date input field with a calendar icon and placeholder 'dd/mm/aaaa'). A blue 'Guardar' button is positioned below the 'Empleado' field. The top right corner shows a user profile icon and the name 'admin'. The bottom left corner of the main area contains the text 'Titanium Fitness'.

Fuente: Elaboración propia.

La Figura 20 muestra la interfaz destinada al registro de incapacidades de los empleados dentro del sistema. Esta pantalla permite al administrador ingresar una nueva incapacidad asociada a un trabajador específico.

El formulario incluye los campos de selección de empleado, tipo de incapacidad, fecha de inicio y fecha de finalización. Estos campos permiten registrar la información necesaria para el control y posterior procesamiento de los periodos de incapacidad.

Una vez completados los datos, el sistema permite guardar la información mediante el botón correspondiente. Al ejecutar esta acción, la aplicación valida los datos ingresados y los almacena en la base de datos MySQL, para asegurar la trazabilidad y el control administrativo de las incapacidades registradas.

La estructura de la interfaz mantiene coherencia visual con el resto del sistema, utilizando un diseño organizado y una disposición clara de los campos, lo cual facilita la correcta introducción de información y reduce la posibilidad de errores en el registro.

Figura 21.

Interfaz de solicitud de horas extra.

The screenshot displays the 'Solicitud de horas extra' (Request for extra hours) interface. On the left is a dark sidebar with the 'Titanium Fitness' logo and a menu containing: Inicio, Notificaciones, Horas extra, Vacaciones, Asistencias, and Permisos. The main content area is titled 'Solicitud de horas extra' and features a 'Crear nuevo' (Create new) section. This section contains three input fields: 'Fecha' with a date format hint 'dd/mm/aaaa', 'Horas', and 'Motivo'. Below these fields is a blue 'Guardar' (Save) button. In the top right corner, there is a user profile icon labeled 'Empleado' with a dropdown arrow. The footer of the page shows the 'Titanium Fitness' logo.

Fuente: Elaboración propia.

La Figura 21 muestra la interfaz destinada al registro de solicitudes de horas extra por parte del empleado. Esta pantalla permite que el trabajador ingrese la información correspondiente a las horas adicionales laboradas durante una fecha específica.

El formulario está compuesto por los campos de fecha, cantidad de horas y motivo, los cuales permiten documentar de manera clara y estructurada el trabajo extraordinario realizado. Una vez completada la información, el usuario puede registrar la solicitud mediante el botón "Guardar".

Al enviarse los datos, el sistema valida la información ingresada, la almacena en la base de datos MySQL y queda disponible para su posterior revisión y aprobación administrativa. Este módulo permite mantener un control formal de las horas extraordinarias y facilita su consideración en los procesos de cálculo de planilla.

Figura 22.*Interfaz de solicitud de permiso.*

The screenshot displays the 'Titanium Fitness' application interface. On the left, a dark sidebar contains navigation icons and labels: 'Inicio', 'Notificaciones', 'Horas extra', 'Vacaciones', 'Asistencias', and 'Permisos'. The main content area is titled 'Solicitud de permiso' and features a form for creating a new request. The form includes three input fields: 'Fecha inicio' (with a date format 'dd/mm/aaaa'), 'Fecha fin' (with a date format 'dd/mm/aaaa'), and 'Motivo'. A blue 'Guardar' button is positioned below the 'Fecha inicio' field. The top right corner shows a user profile icon labeled 'Empleado'.

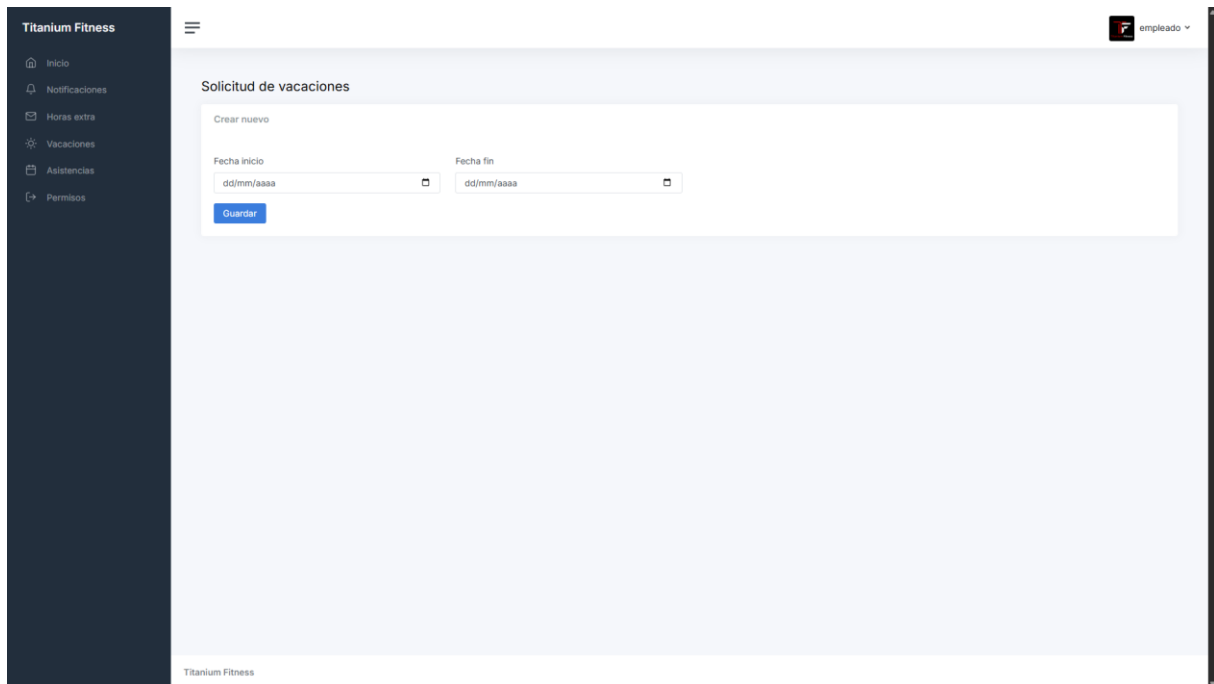
Fuente: Elaboración propia.

La Figura 22 presenta la interfaz destinada al registro de solicitudes de permiso por parte del empleado. Esta pantalla permite que el trabajador ingrese formalmente un periodo de ausencia autorizado, indicando la fecha de inicio, la fecha de finalización y el motivo correspondiente.

El formulario está estructurado con campos de selección de fechas y un campo de texto para describir el motivo del permiso, lo cual permite documentar de manera clara y organizada la información requerida. Una vez completados los datos, el usuario puede registrar la solicitud mediante el botón "Guardar".

Figura 23.

Interfaz de solicitud de vacaciones.



The screenshot shows a web application interface for Titanium Fitness. On the left is a dark sidebar with a menu containing: Inicio, Notificaciones, Horas extra, Vacaciones, Asistencias, and Permisos. The main content area is titled 'Solicitud de vacaciones' and features a 'Crear nuevo' section. This section contains two date selection fields: 'Fecha inicio' and 'Fecha fin', both with a 'dd/mm/aaaa' placeholder and a calendar icon. Below these fields is a blue 'Guardar' button. In the top right corner, there is a user profile icon labeled 'Empleado' with a dropdown arrow. The footer of the page contains the text 'Titanium Fitness'.

Fuente: Elaboración propia.

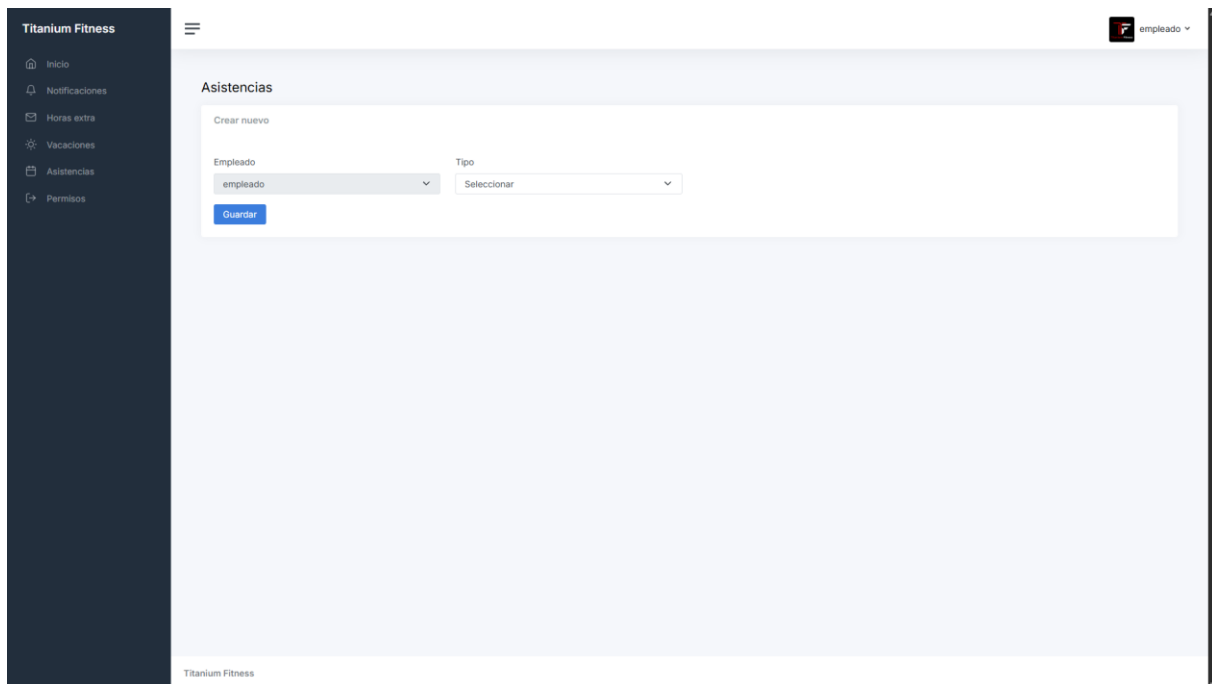
La Figura 23 muestra la interfaz destinada al registro de solicitudes de vacaciones por parte del empleado. Esta pantalla permite que el trabajador ingrese el periodo de vacaciones solicitado, indicando la fecha de inicio y la fecha de finalización.

El formulario presenta campos de selección de fechas que facilitan la introducción correcta del rango solicitado. Una vez completada la información, el usuario puede registrar la solicitud mediante el botón "Guardar".

Al enviarse los datos, el sistema valida la información ingresada, la almacena en la base de datos MySQL y permite que la administración gestione y controle los periodos de ausencia por vacaciones. Esta funcionalidad contribuye a mantener un registro organizado de los días solicitados, para evitar inconsistencias en la planificación operativa y en los procesos relacionados con la planilla.

Figura 24.

Interfaz de registro de asistencias.

The image shows a web application interface for 'Titanium Fitness'. On the left is a dark sidebar with a menu containing 'Inicio', 'Notificaciones', 'Horas extra', 'Vacaciones', 'Asistencias', and 'Permisos'. The main content area is titled 'Asistencias' and features a 'Crear nuevo' section. This section contains two dropdown menus: 'Empleado' (with 'empleado' selected) and 'Tipo' (with 'Seleccionar' selected). Below these is a blue 'Guardar' button. The top right corner shows a user profile icon and the name 'empleado'. The bottom left corner of the main area has the text 'Titanium Fitness'.

La Figura 24 muestra la interfaz destinada al registro de asistencias de los empleados. Esta pantalla permite documentar las marcas correspondientes a la jornada laboral, específicamente las horas de entrada y salida.

El formulario incluye un campo de selección de empleado y un campo denominado “Tipo”, en el cual se puede elegir entre las opciones de entrada o salida. De esta forma, el sistema permite registrar de manera estructurada cada marca realizada durante la jornada.

Una vez ingresada la información, el usuario puede almacenarla mediante el botón "Guardar". El sistema valida los datos, los registra en la base de datos MySQL y permite llevar un control organizado de la asistencia del personal.

Este módulo contribuye al seguimiento de la jornada laboral y facilita la verificación de cumplimiento horario, además de servir como insumo para procesos administrativos relacionados con la gestión de personal.

Figura 25.*Interfaz de gestión de usuarios.*

Nombres	Apellidos	Usuario	Sueldo base	Rol	Acción
Mauro	Perez	admin	€ 2,000.00	admin	[Edit] [Delete]
Adrian	Chaves	empleado	€ 1,000.00	empleado	[Edit] [Delete]
Emmanuel	Solera	empleado 1	€ 1,000.00	empleado	[Edit] [Delete]
Isai	Andrades	empleado 2	€ 4,000.00	empleado	[Edit] [Delete]
Francisco	Murillo	empleado 3	€ 23,000.00	empleado	[Edit] [Delete]
Isaac	Sanchez	empleado 4	€ 3,400.00	empleado	[Edit] [Delete]

Fuente: Elaboración propia.

La Figura 25 muestra la interfaz destinada a la gestión de usuarios del sistema. Esta pantalla permite al administrador visualizar, registrar, editar y eliminar los usuarios que tendrán acceso a la aplicación.

La interfaz presenta una tabla estructurada que incluye información relevante como nombres, apellidos, nombre de usuario, sueldo base y rol asignado. El rol permite diferenciar entre administrador y empleado, determinando así los permisos y funcionalidades disponibles para cada tipo de usuario dentro del sistema.

Adicionalmente, la pantalla incluye un botón para la creación de nuevos usuarios y opciones de acción para editar o eliminar registros existentes. Estas funcionalidades permiten mantener actualizado el control de acceso y la información necesaria para los procesos de planilla.

Este módulo es fundamental para la administración del sistema, ya que centraliza la gestión de credenciales y datos laborales básicos que posteriormente son utilizados en los cálculos de nómina y demás procesos relacionados con la gestión del personal.

Figura 26.

Interfaz de gestión de permisos.

Usuario	Fecha inicio	Fecha fin	Motivo	Estado	Fecha solicitud	Fecha revisión	Revisor	Acción
empleado 1	2026-02-13	2026-02-14	personal	pendiente	2026-02-11 02:04:26			+ -

Fuente: Elaboración propia.

La Figura 26 muestra la interfaz destinada a la gestión y revisión de solicitudes de permisos. Esta pantalla permite al administrador visualizar las solicitudes realizadas por los empleados y tomar una decisión respecto a su aprobación o rechazo.

La información se presenta en una tabla estructurada que incluye datos como el usuario solicitante, fecha de inicio, fecha de finalización, motivo, estado de la solicitud, fecha de registro, fecha de revisión y el revisor correspondiente. Esta organización permite un seguimiento claro y ordenado de cada solicitud.

El estado de cada permiso puede visualizarse de forma inmediata (por ejemplo, pendiente), lo que facilita la identificación de solicitudes que requieren revisión. Asimismo, la columna de acciones permite aprobar o rechazar la solicitud mediante los botones correspondientes.

Este módulo contribuye al control administrativo de ausencias autorizadas, asegurando que cada solicitud quede registrada y validada antes de ser considerada en los procesos internos del sistema, incluyendo la gestión de asistencia y planilla.

Figura 27.*Interfaz de gestión de liquidaciones.*

Empleado	Motivo	Salario pendiente	Vacaciones	Agüinaldo	Preaviso	Cesantía	Total	Fecha	Acción
admin	despido_cor_resp	€ 1,333.33	€ 333.33	€ 166.67	€ 2,000.00	€ 2,000.00	€ 5,833.33	2026-02-11 04:17:30	[Acción]

Fuente: Elaboración propia.

La Figura 27 muestra la interfaz correspondiente a la gestión de liquidaciones laborales. Este módulo permite registrar y visualizar los cálculos asociados a la finalización de la relación laboral de un empleado.

La información se presenta en una tabla estructurada que incluye datos como el empleado, motivo de la liquidación, salario pendiente, vacaciones acumuladas, agüinaldo proporcional, preaviso, cesantía y el total calculado. Además, se muestra la fecha de registro y la opción de acción correspondiente.

El sistema realiza los cálculos necesarios con base en los datos almacenados previamente, permitiendo automatizar el proceso de liquidación y reducir errores asociados a cálculos manuales. Esto contribuye a garantizar mayor precisión y transparencia en la determinación de los montos correspondientes.

Figura 28.

Interfaz de gestión de asistencias.

Empleado	Tipo	Fecha	Acción
empleador 1	salida	2026-02-13 13:12:16	
empleador 1	entrada	2026-02-13 08:15:23	

Fuente: Elaboración propia.

La Figura 28 muestra la interfaz destinada a la gestión y visualización de registros de asistencia. Esta pantalla permite consultar las marcas de entrada y salida registradas por los empleados.

La información se presenta en una tabla estructurada que incluye el nombre del empleado, el tipo de marca (entrada o salida) y la fecha y hora en que fue registrada. Esta organización permite llevar un control cronológico de la jornada laboral.

Adicionalmente, la columna de acción permite eliminar registros en caso de error, garantizando la correcta administración de la información almacenada.

Este módulo contribuye al control de cumplimiento horario y facilita la supervisión administrativa del personal. Asimismo, los datos registrados pueden servir como insumo para otros procesos internos del sistema, como el control de horas extra o validaciones relacionadas con la gestión de planilla.

Figura 29.

Interfaz de gestión de aguinaldos.

Empleado	Periodo	Total salarios	Meses laborados	Monto aguinaldo	Fecha	Acción
admin	2026	€ 4,000.00	2	€ 2,000.00	2026-02-11 08:24:42	

Fuente: Elaboración propia.

La Figura 29 muestra la interfaz correspondiente al módulo de gestión de aguinaldos del sistema. Esta pantalla permite al administrador registrar, calcular y consultar el aguinaldo de los empleados según el periodo seleccionado.

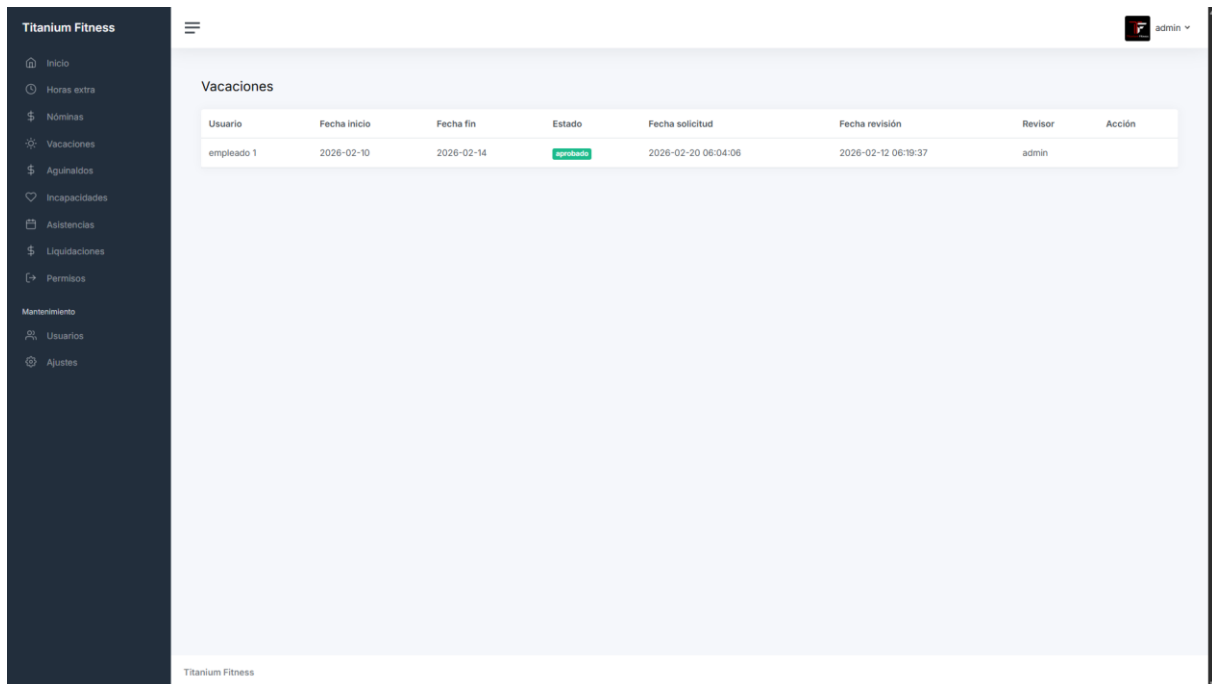
La información se presenta en una tabla estructurada que incluye los siguientes datos: empleado, periodo, total de salarios acumulados, meses laborados, monto de aguinaldo calculado, fecha de registro y la acción disponible.

El sistema realiza automáticamente el cálculo del monto del aguinaldo con base en los salarios acumulados durante el periodo indicado, lo cual reduce errores manuales y mejora la precisión del proceso. El botón “Crear nuevo” permite generar un nuevo cálculo, mientras que la opción de acción facilita la eliminación de registros cuando sea necesario.

Este módulo contribuye al cumplimiento adecuado de las obligaciones laborales relacionadas con el pago del aguinaldo, garantizando un control organizado y transparente dentro del gimnasio.

Figura 30.

Interfaz de gestión de vacaciones.



Usuario	Fecha inicio	Fecha fin	Estado	Fecha solicitud	Fecha revisión	Revisor	Acción
empleado 1	2026-02-10	2026-02-14	aprobado	2026-02-20 06:04:06	2026-02-12 06:19:37	admin	

Fuente: Elaboración propia.

La Figura 30 muestra la interfaz correspondiente al módulo de gestión de vacaciones del sistema. Esta pantalla permite al administrador visualizar y controlar las solicitudes de vacaciones realizadas por los empleados.

La información se presenta en una tabla organizada que incluye los siguientes campos: usuario, fecha de inicio, fecha de finalización, estado de la solicitud, fecha de solicitud, fecha de revisión, revisor y acción correspondiente.

El campo estado indica si la solicitud ha sido aprobada, rechazada o se encuentra pendiente, lo que facilita el seguimiento del proceso. Además, se registra el usuario revisor (administrador) y la fecha en que se realizó la revisión, garantizando trazabilidad y control administrativo.

Este módulo permite mantener un historial claro de las vacaciones otorgadas, lo que contribuye a una mejor planificación operativa dentro del gimnasio y a evitar conflictos en la programación del personal.

Figura 31.*Interfaz de gestión de nóminas.*

Empleado	Periodo	Sueldo base	Salario bruto	Total deducciones	Neto a pagar	Fecha	Acción
admin	Enero 2026	€ 2,000.00	€ 2,000.00	€ 206.80	€ 1,793.20	2026-02-15 06:44:14	
empleado 3	Enero 2026	€ 23,000.00	€ 23,000.00	€ 2,378.20	€ 20,621.80	2026-02-15 06:20:30	

Fuente: Elaboración propia.

La Figura 31 muestra la interfaz correspondiente al módulo de gestión de nóminas del sistema. Esta pantalla permite al administrador generar, consultar y administrar los registros de planilla mensual de los empleados.

La información se presenta en una tabla estructurada que incluye los siguientes campos: empleado, periodo, sueldo base, salario bruto, total de deducciones, monto neto a pagar, fecha de registro y acción correspondiente.

El sistema realiza automáticamente los cálculos necesarios para determinar el salario bruto, aplicar las deducciones correspondientes y obtener el monto neto a pagar, lo que disminuye la probabilidad de errores manuales y mejora la precisión del proceso.

El botón “Crear nuevo” permite generar una nueva nómina para un periodo específico, mientras que la opción de acción permite eliminar registros cuando sea necesario.

Este módulo constituye uno de los componentes centrales del sistema, ya que integra la información proveniente de otros módulos como horas extra, incapacidades y deducciones, permitiendo una gestión integral y automatizada de la planilla en Titanium Fitness.

Diseño físico de la base de datos

En esta sección se presenta el diseño físico de la base de datos implementada para el sistema de gestión de planilla de Titanium Fitness. Se describe la estructura real creada en MySQL, incluyendo las tablas, sus campos, tipos de datos y relaciones establecidas entre ellas.

El diseño físico fue desarrollado considerando los requerimientos funcionales del sistema, con el objetivo de garantizar la integridad, coherencia y organización de la información. Cada tabla representa una entidad clave dentro del sistema, como usuarios, nóminas, vacaciones, incapacidades, permisos, asistencias, liquidaciones y aguinaldos.

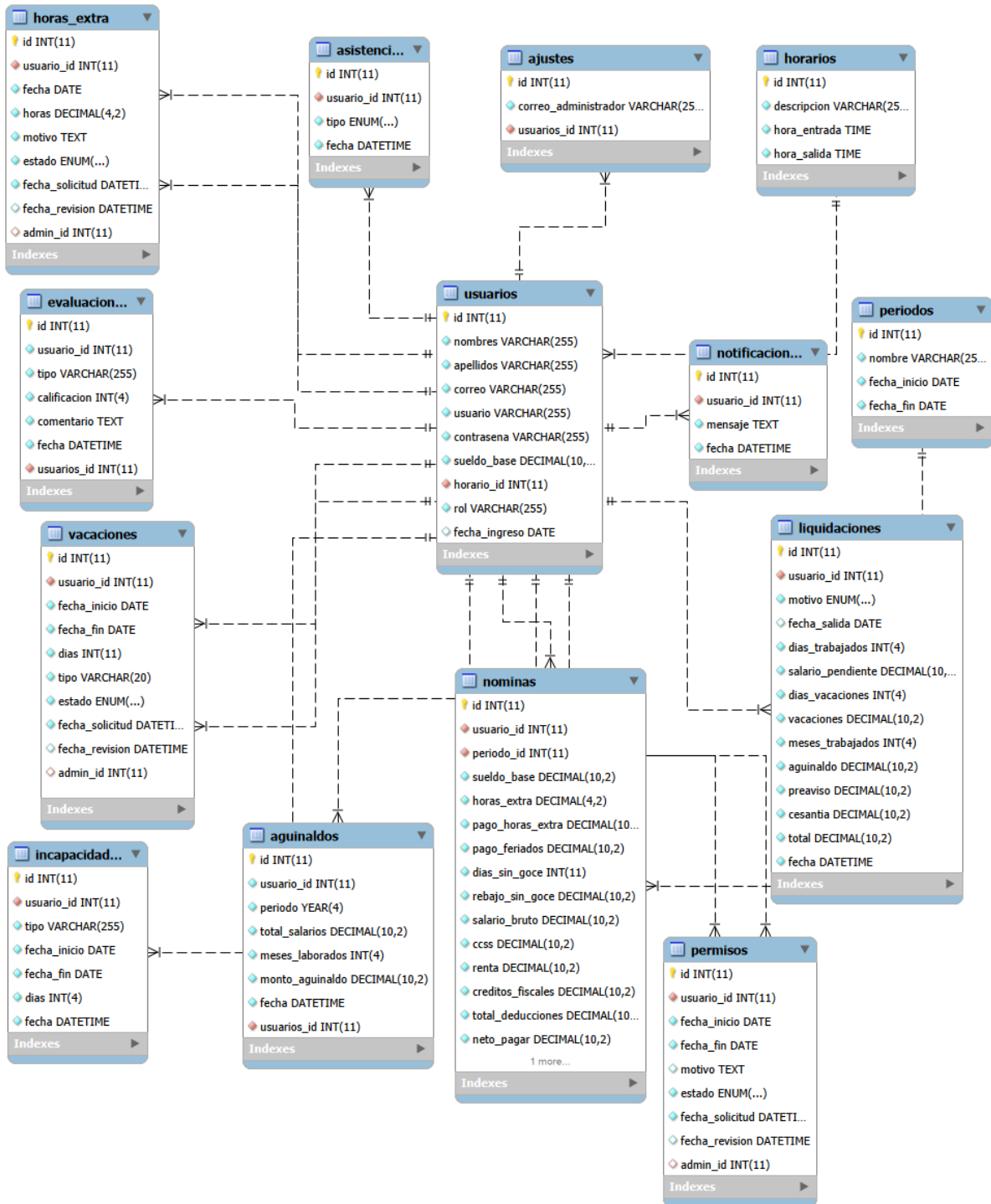
Las relaciones entre tablas se establecieron mediante claves primarias y claves foráneas, permitiendo mantener la integridad referencial y evitar inconsistencias en los datos. Por ejemplo, los registros de nómina se relacionan con la tabla de usuarios, mientras que los módulos de vacaciones, incapacidades y permisos se vinculan directamente con el empleado correspondiente.

Asimismo, se definieron tipos de datos adecuados para cada campo (VARCHAR, INT, DECIMAL, DATE, DATETIME, entre otros), para asegurar un almacenamiento eficiente y estructurado de la información. En los campos relacionados con montos salariales se utilizaron tipos DECIMAL para mantener precisión en los cálculos financieros.

Este diseño físico permite que el sistema opere de manera estable dentro del entorno local (XAMPP – MySQL), facilitando el almacenamiento, consulta y actualización de datos en tiempo real dentro de la red interna del gimnasio.

Figura 32.

Diseño físico de la base de datos.



Fuente: Elaboración propia.

A continuación, en la figura 32 se presenta el diccionario de datos, el cual constituye un recurso fundamental para el desarrollo del proyecto. Este documento tiene como propósito detallar y clarificar los términos técnicos y conceptos específicos utilizados a lo largo de la investigación, facilitando así una mejor comprensión de la información presentada.

Figura 33.

Diccionario de datos

No	REPORT	ServerName	DatabaseName	TableName	SchemaName	ColumnName	DataType	Max.Length	IsNull	IsIdentity	KeyConstraint
1	DATADICIONARY	Mauro	empresa	aguinaldos	empresa	id	int	NULL	NO	YES	PRIMARY KEY
2	DATADICIONARY	Mauro	empresa	aguinaldos	empresa	usuario_id	int	NULL	NO	NO	
3	DATADICIONARY	Mauro	empresa	aguinaldos	empresa	periodo	year	NULL	NO	NO	
4	DATADICIONARY	Mauro	empresa	aguinaldos	empresa	total_salarios	decimal	NULL	NO	NO	
5	DATADICIONARY	Mauro	empresa	aguinaldos	empresa	meses_laborados	int	NULL	NO	NO	
6	DATADICIONARY	Mauro	empresa	aguinaldos	empresa	monto_aguinaldo	decimal	NULL	NO	NO	
7	DATADICIONARY	Mauro	empresa	aguinaldos	empresa	fecha	datetime	NULL	NO	NO	
8	DATADICIONARY	Mauro	empresa	ajustes	empresa	id	int	NULL	NO	YES	PRIMARY KEY
9	DATADICIONARY	Mauro	empresa	ajustes	empresa	correo_administrador	varchar	255	NO	NO	
10	DATADICIONARY	Mauro	empresa	asistencias	empresa	id	int	NULL	NO	YES	PRIMARY KEY
11	DATADICIONARY	Mauro	empresa	asistencias	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
12	DATADICIONARY	Mauro	empresa	asistencias	empresa	tipo	enum	7	NO	NO	
13	DATADICIONARY	Mauro	empresa	asistencias	empresa	fecha	datetime	NULL	NO	NO	
14	DATADICIONARY	Mauro	empresa	evaluaciones	empresa	id	int	NULL	NO	YES	PRIMARY KEY
15	DATADICIONARY	Mauro	empresa	evaluaciones	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
16	DATADICIONARY	Mauro	empresa	evaluaciones	empresa	tipo	varchar	255	NO	NO	
17	DATADICIONARY	Mauro	empresa	evaluaciones	empresa	calificacion	int	NULL	NO	NO	
18	DATADICIONARY	Mauro	empresa	evaluaciones	empresa	comentario	text	65535	NO	NO	
19	DATADICIONARY	Mauro	empresa	evaluaciones	empresa	fecha	datetime	NULL	NO	NO	
20	DATADICIONARY	Mauro	empresa	horarios	empresa	id	int	NULL	NO	YES	PRIMARY KEY
21	DATADICIONARY	Mauro	empresa	horarios	empresa	descripcion	varchar	255	NO	NO	
22	DATADICIONARY	Mauro	empresa	horarios	empresa	hora_entrada	time	NULL	NO	NO	
23	DATADICIONARY	Mauro	empresa	horarios	empresa	hora_salida	time	NULL	NO	NO	
24	DATADICIONARY	Mauro	empresa	horas_extra	empresa	id	int	NULL	NO	YES	PRIMARY KEY
25	DATADICIONARY	Mauro	empresa	horas_extra	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
26	DATADICIONARY	Mauro	empresa	horas_extra	empresa	fecha	date	NULL	NO	NO	
27	DATADICIONARY	Mauro	empresa	horas_extra	empresa	horas	decimal	NULL	NO	NO	
28	DATADICIONARY	Mauro	empresa	horas_extra	empresa	motivo	text	65535	NO	NO	
29	DATADICIONARY	Mauro	empresa	horas_extra	empresa	estado	enum	9	NO	NO	
30	DATADICIONARY	Mauro	empresa	horas_extra	empresa	fecha_solicitud	datetime	NULL	NO	NO	
31	DATADICIONARY	Mauro	empresa	horas_extra	empresa	fecha_revision	datetime	NULL	YES	NO	
32	DATADICIONARY	Mauro	empresa	horas_extra	empresa	admin_id	int	NULL	YES	NO	FOREIGN KEY / INDEX
33	DATADICIONARY	Mauro	empresa	incapacida...	empresa	id	int	NULL	NO	YES	PRIMARY KEY
34	DATADICIONARY	Mauro	empresa	incapacida...	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
35	DATADICIONARY	Mauro	empresa	incapacida...	empresa	tipo	varchar	255	NO	NO	
36	DATADICIONARY	Mauro	empresa	incapacida...	empresa	fecha_inicio	date	NULL	NO	NO	
37	DATADICIONARY	Mauro	empresa	incapacida...	empresa	fecha_fin	date	NULL	NO	NO	
38	DATADICIONARY	Mauro	empresa	incapacida...	empresa	dias	int	NULL	NO	NO	
39	DATADICIONARY	Mauro	empresa	incapacida...	empresa	fecha	datetime	NULL	NO	NO	
40	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	id	int	NULL	NO	YES	PRIMARY KEY
41	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
42	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	motivo	enum	16	NO	NO	
43	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	fecha_salida	date	NULL	YES	NO	
44	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	dias_trabajados	int	NULL	NO	NO	
45	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	salario_pendiente	decimal	NULL	NO	NO	
46	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	dias_vacaciones	int	NULL	NO	NO	
47	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	vacaciones	decimal	NULL	NO	NO	
48	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	meses_trabajados	int	NULL	NO	NO	
49	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	aguinaldo	decimal	NULL	NO	NO	
50	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	preaviso	decimal	NULL	NO	NO	
51	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	cesantia	decimal	NULL	NO	NO	
52	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	total	decimal	NULL	NO	NO	

Fuente: Elaboración propia.

Figura 34.

Diccionario de datos

No	REPORT	ServerName	DatabaseName	TableName	SchemaName	ColumnName	DataType	MaxLength	IsNull	IsIdentity	KeyConstraint
53	DATADICIONARY	Mauro	empresa	liquidaciones	empresa	fecha	datetime	NULL	NO	NO	
54	DATADICIONARY	Mauro	empresa	nominas	empresa	id	int	NULL	NO	YES	PRIMARY KEY
55	DATADICIONARY	Mauro	empresa	nominas	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
56	DATADICIONARY	Mauro	empresa	nominas	empresa	periodo_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
57	DATADICIONARY	Mauro	empresa	nominas	empresa	sueldo_base	decimal	NULL	NO	NO	
58	DATADICIONARY	Mauro	empresa	nominas	empresa	horas_extra	decimal	NULL	NO	NO	
59	DATADICIONARY	Mauro	empresa	nominas	empresa	pago_horas_extra	decimal	NULL	NO	NO	
60	DATADICIONARY	Mauro	empresa	nominas	empresa	pago_feridos	decimal	NULL	NO	NO	
61	DATADICIONARY	Mauro	empresa	nominas	empresa	dias_sin_goce	int	NULL	NO	NO	
62	DATADICIONARY	Mauro	empresa	nominas	empresa	rebajo_sin_goce	decimal	NULL	NO	NO	
63	DATADICIONARY	Mauro	empresa	nominas	empresa	salario_bruto	decimal	NULL	NO	NO	
64	DATADICIONARY	Mauro	empresa	nominas	empresa	ccss	decimal	NULL	NO	NO	
65	DATADICIONARY	Mauro	empresa	nominas	empresa	renta	decimal	NULL	NO	NO	
66	DATADICIONARY	Mauro	empresa	nominas	empresa	creditos_fiscales	decimal	NULL	NO	NO	
67	DATADICIONARY	Mauro	empresa	nominas	empresa	total_deducciones	decimal	NULL	NO	NO	
68	DATADICIONARY	Mauro	empresa	nominas	empresa	neto_pagar	decimal	NULL	NO	NO	
69	DATADICIONARY	Mauro	empresa	nominas	empresa	fecha	datetime	NULL	NO	NO	
70	DATADICIONARY	Mauro	empresa	notificacio...	empresa	id	int	NULL	NO	YES	PRIMARY KEY
71	DATADICIONARY	Mauro	empresa	notificacio...	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
72	DATADICIONARY	Mauro	empresa	notificacio...	empresa	mensaje	text	65535	NO	NO	
73	DATADICIONARY	Mauro	empresa	notificacio...	empresa	fecha	datetime	NULL	NO	NO	
74	DATADICIONARY	Mauro	empresa	periodos	empresa	id	int	NULL	NO	YES	PRIMARY KEY
75	DATADICIONARY	Mauro	empresa	periodos	empresa	nombre	varchar	255	NO	NO	
76	DATADICIONARY	Mauro	empresa	periodos	empresa	fecha_inicio	date	NULL	NO	NO	
77	DATADICIONARY	Mauro	empresa	periodos	empresa	fecha_fin	date	NULL	NO	NO	
78	DATADICIONARY	Mauro	empresa	permisos	empresa	id	int	NULL	NO	YES	PRIMARY KEY
79	DATADICIONARY	Mauro	empresa	permisos	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
80	DATADICIONARY	Mauro	empresa	permisos	empresa	fecha_inicio	date	NULL	NO	NO	
81	DATADICIONARY	Mauro	empresa	permisos	empresa	fecha_fin	date	NULL	NO	NO	
82	DATADICIONARY	Mauro	empresa	permisos	empresa	motivo	text	65535	YES	NO	
83	DATADICIONARY	Mauro	empresa	permisos	empresa	estado	enum	9	NO	NO	
84	DATADICIONARY	Mauro	empresa	permisos	empresa	fecha_solicitud	datetime	NULL	NO	NO	
85	DATADICIONARY	Mauro	empresa	permisos	empresa	fecha_revision	datetime	NULL	YES	NO	
86	DATADICIONARY	Mauro	empresa	permisos	empresa	admin_id	int	NULL	YES	NO	FOREIGN KEY / INDEX
87	DATADICIONARY	Mauro	empresa	usuarios	empresa	id	int	NULL	NO	YES	PRIMARY KEY
88	DATADICIONARY	Mauro	empresa	usuarios	empresa	nombres	varchar	255	NO	NO	
89	DATADICIONARY	Mauro	empresa	usuarios	empresa	apellidos	varchar	255	NO	NO	
90	DATADICIONARY	Mauro	empresa	usuarios	empresa	correo	varchar	255	NO	NO	
91	DATADICIONARY	Mauro	empresa	usuarios	empresa	usuario	varchar	255	NO	NO	
92	DATADICIONARY	Mauro	empresa	usuarios	empresa	contrasena	varchar	255	NO	NO	
93	DATADICIONARY	Mauro	empresa	usuarios	empresa	sueldo_base	decimal	NULL	NO	NO	
94	DATADICIONARY	Mauro	empresa	usuarios	empresa	horario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
95	DATADICIONARY	Mauro	empresa	usuarios	empresa	rol	varchar	255	NO	NO	
96	DATADICIONARY	Mauro	empresa	usuarios	empresa	fecha_ingreso	date	NULL	YES	NO	
97	DATADICIONARY	Mauro	empresa	vacaciones	empresa	id	int	NULL	NO	YES	PRIMARY KEY
98	DATADICIONARY	Mauro	empresa	vacaciones	empresa	usuario_id	int	NULL	NO	NO	FOREIGN KEY / INDEX
99	DATADICIONARY	Mauro	empresa	vacaciones	empresa	fecha_inicio	date	NULL	NO	NO	
100	DATADICIONARY	Mauro	empresa	vacaciones	empresa	fecha_fin	date	NULL	NO	NO	
101	DATADICIONARY	Mauro	empresa	vacaciones	empresa	dias	int	NULL	NO	NO	
102	DATADICIONARY	Mauro	empresa	vacaciones	empresa	tipo	varchar	20	NO	NO	
103	DATADICIONARY	Mauro	empresa	vacaciones	empresa	estado	enum	9	NO	NO	
104	DATADICIONARY	Mauro	empresa	vacaciones	empresa	fecha_solicitud	datetime	NULL	NO	NO	
105	DATADICIONARY	Mauro	empresa	vacaciones	empresa	fecha_revision	datetime	NULL	YES	NO	
106	DATADICIONARY	Mauro	empresa	vacaciones	empresa	admin_id	int	NULL	YES	NO	FOREIGN KEY / INDEX

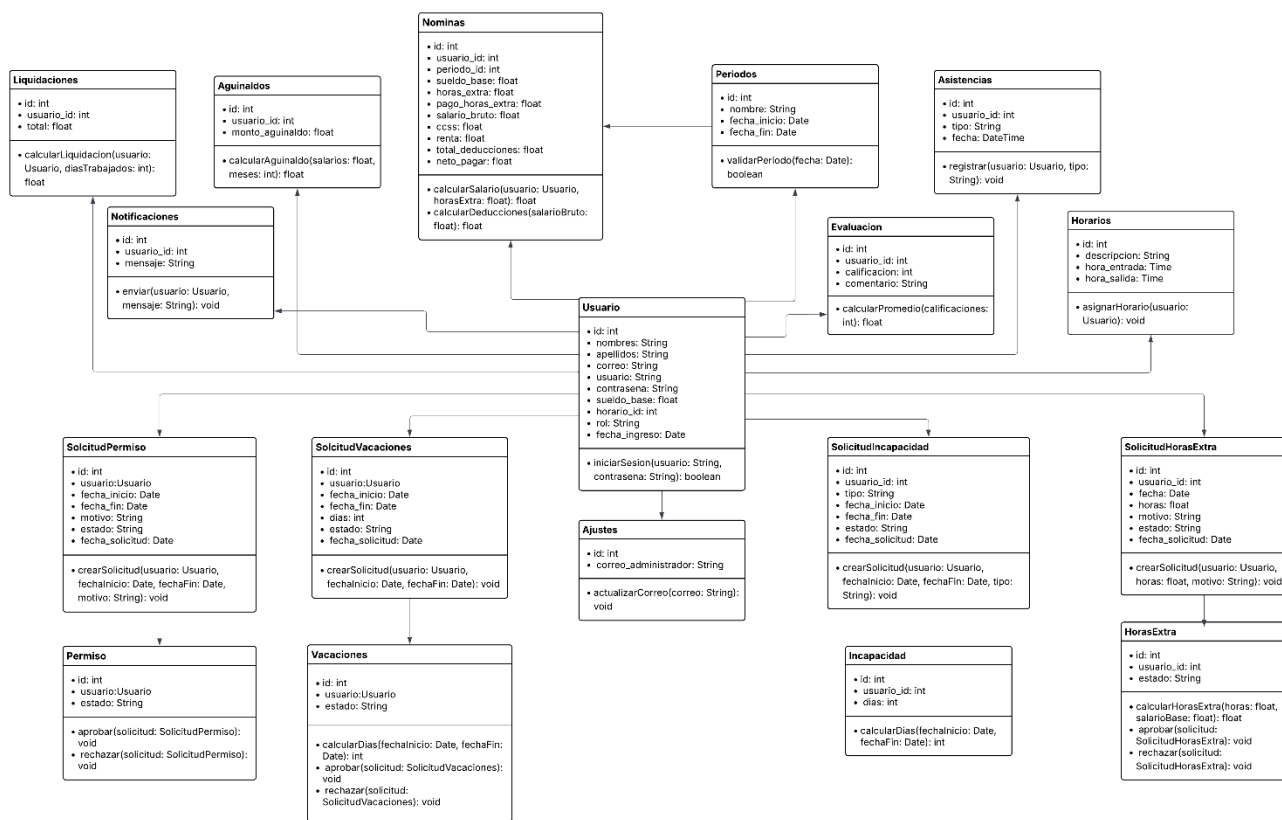
Fuente: Elaboración propia.

Diagrama de clases

El diagrama de clases representa un elemento importante dentro del desarrollo del sistema propuesto para el gimnasio Titanium Fitness. En esta sección se describe la estructura interna de la solución, utilizando este tipo de diagrama como apoyo para visualizar la organización del sistema. A través de él, se logra comprender cómo están definidas las entidades, así como las relaciones e interacciones que existen entre ellas. Este análisis permite tener una visión clara del funcionamiento del sistema y facilita su desarrollo e implementación.

Figura 35.

Diagrama de clases



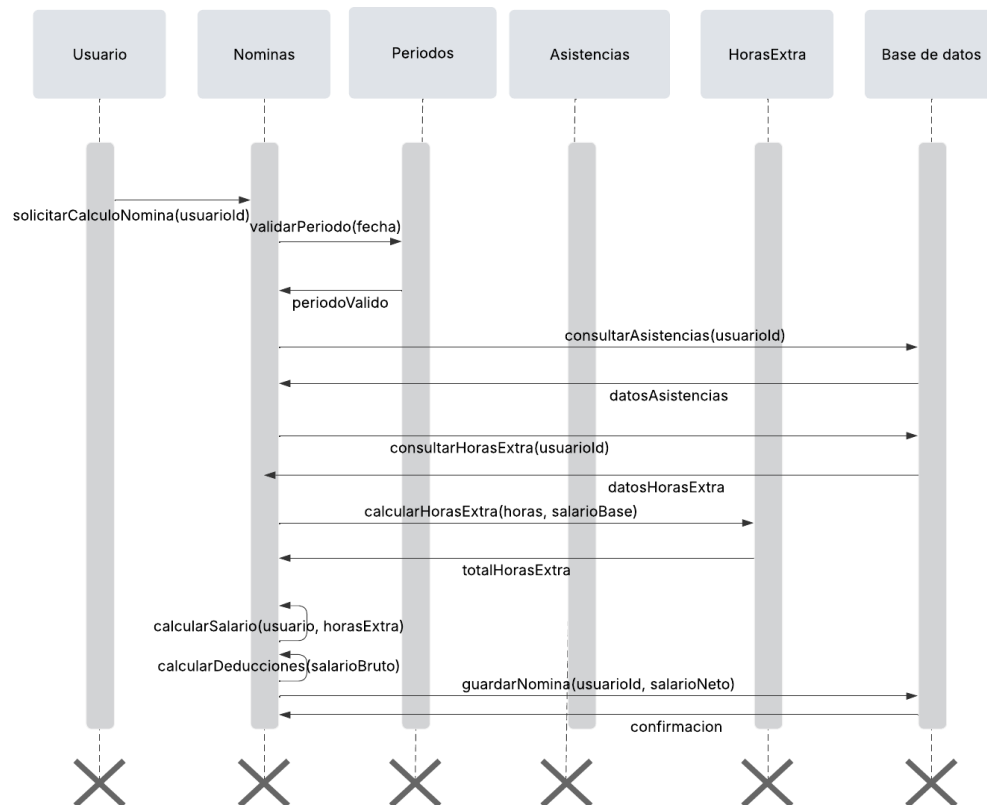
Fuente: Elaboración propia.

Diagrama de secuencia

En el desarrollo del sistema de gestión de planilla para el gimnasio Titanium Fitness, los diagramas de secuencia permiten visualizar de forma clara cómo interactúan el administrador y el sistema en los distintos procesos. Estos diagramas muestran el flujo de información desde que se realiza una acción hasta que el sistema genera una respuesta, facilitando la comprensión y validación del funcionamiento general. A continuación, se presentan los diagramas correspondientes a los principales procesos del sistema.

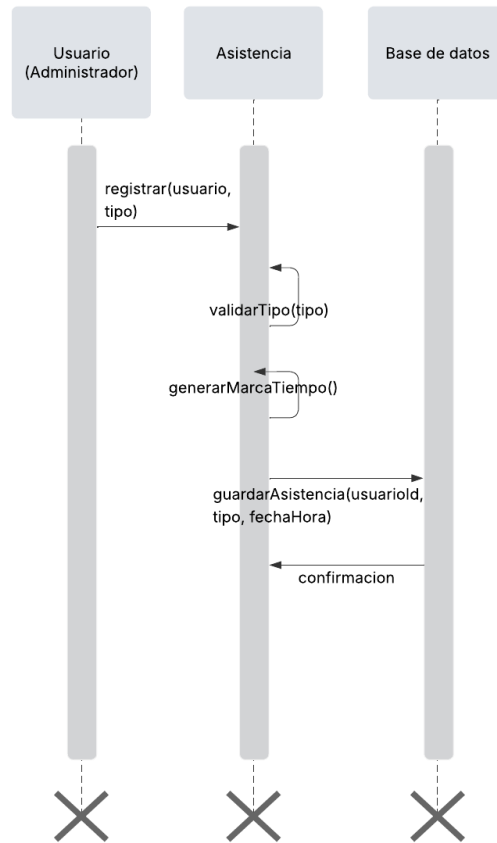
Figura 36.

Diagrama de secuencia de nomina



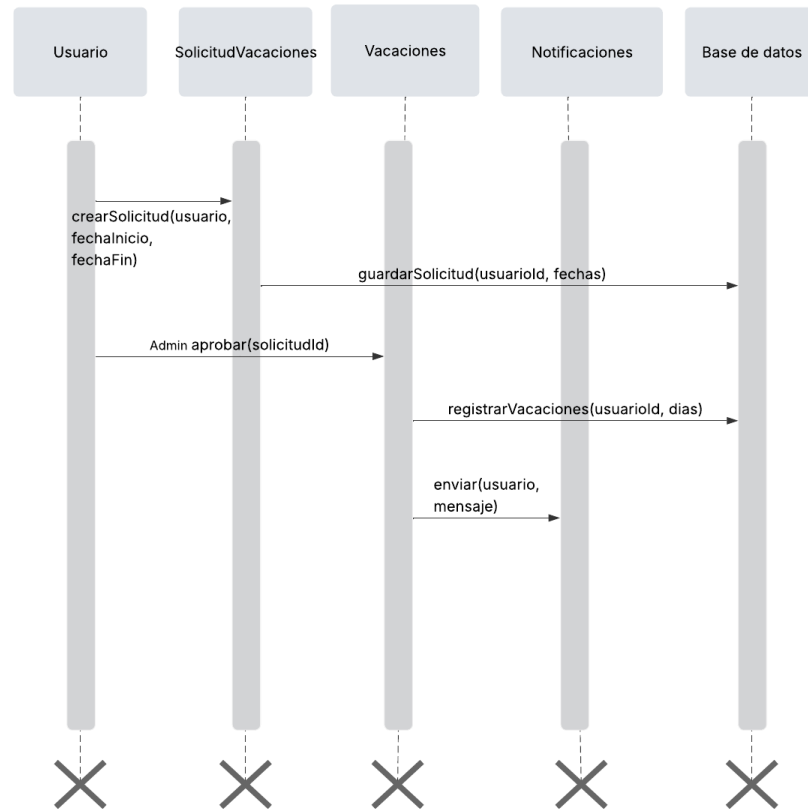
Fuente: Elaboración propia.

El diagrama de secuencia del cálculo de nómina ilustra la interacción entre los distintos componentes del sistema para obtener datos de asistencias y horas extra, calcular el salario correspondiente, aplicar deducciones y finalmente almacenar la información en la base de datos.

Figura 37.*Diagrama de secuencia de asistencia*

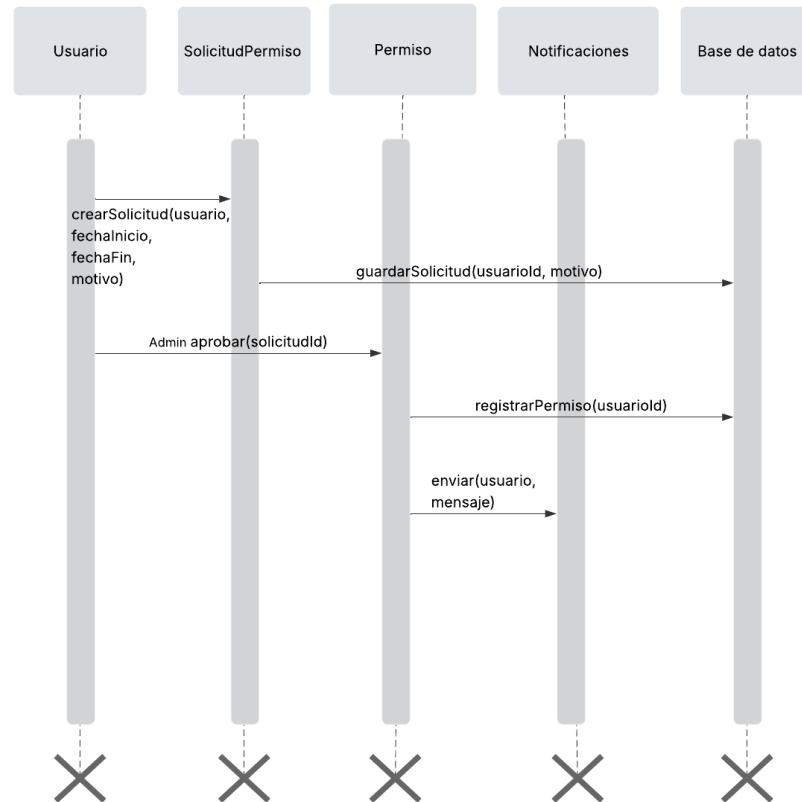
Fuente: Elaboración propia.

Este diagrama muestra el proceso de registro de asistencias, en el cual el usuario realiza una marcación que es validada y almacenada en la base de datos, garantizando el control adecuado de entradas y salidas.

Figura 38.*Diagrama de secuencia de vacaciones*

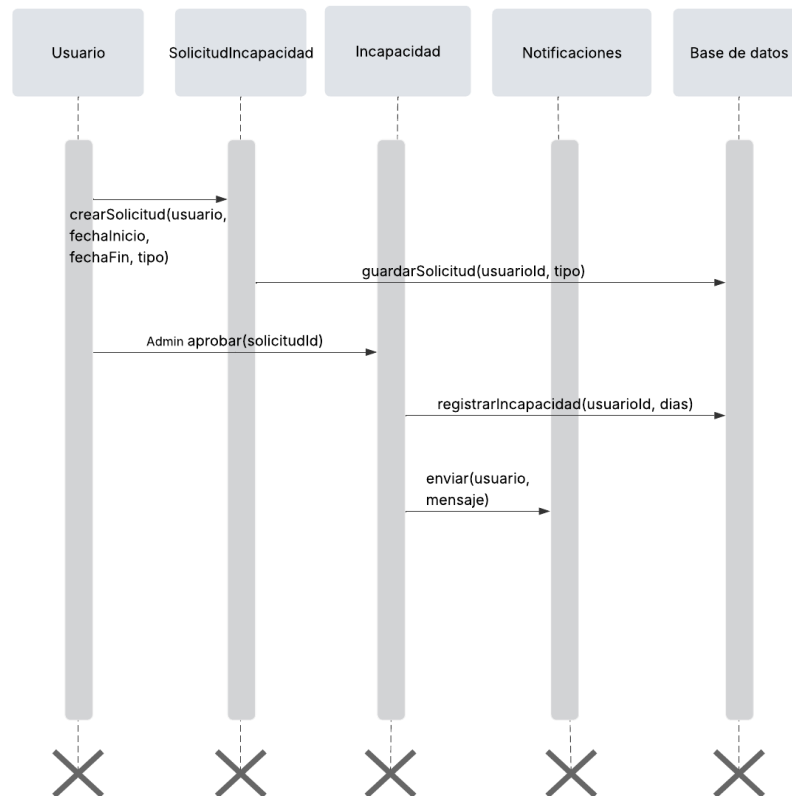
Fuente: Elaboración propia.

Este diagrama representa el proceso mediante el cual un usuario solicita vacaciones, las cuales son registradas, evaluadas por un administrador y, en caso de aprobación, se almacenan y se notifica al usuario.

Figura 39.*Diagrama de secuencia de permisos*

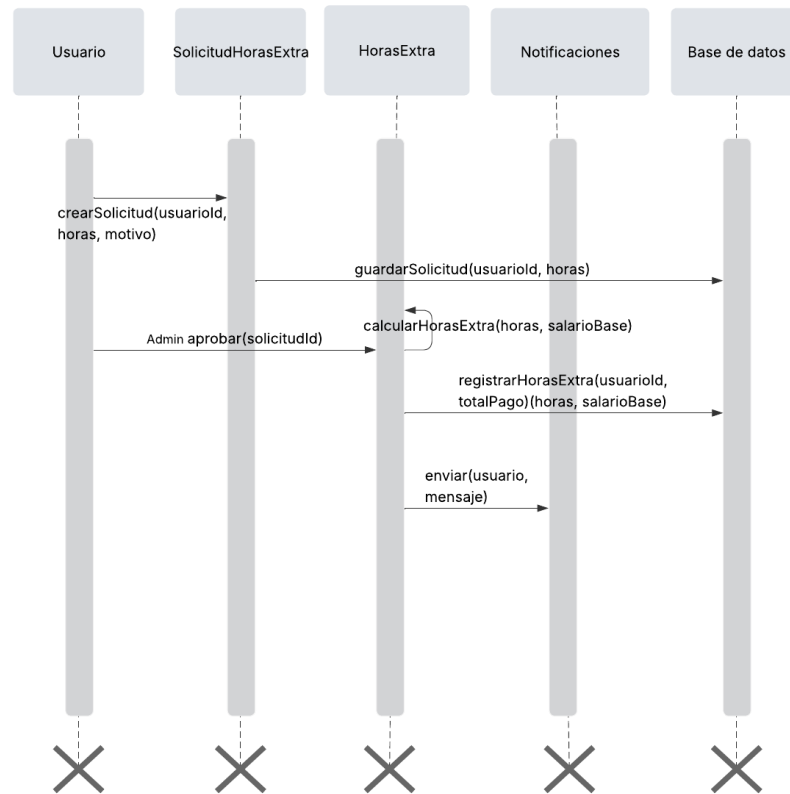
Fuente: Elaboración propia.

El diagrama muestra el flujo de solicitud de permisos, donde el usuario registra su solicitud, esta es evaluada por un administrador y, posteriormente, se almacena en el sistema y se notifica el resultado.

Figura 40.*Diagrama de secuencia de incapacidad*

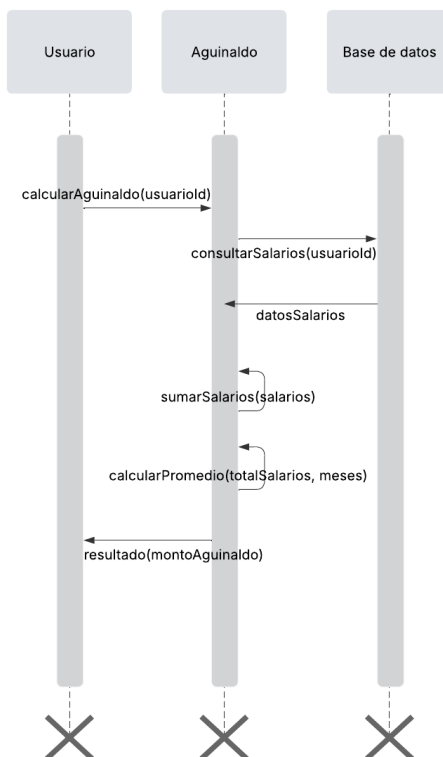
Fuente: Elaboración propia.

Este diagrama describe el proceso de registro y aprobación de incapacidades, donde el usuario envía la solicitud, el administrador la valida y se procede a almacenar la información junto con la notificación correspondiente.

Figura 41.*Diagrama de secuencia de horas extra*

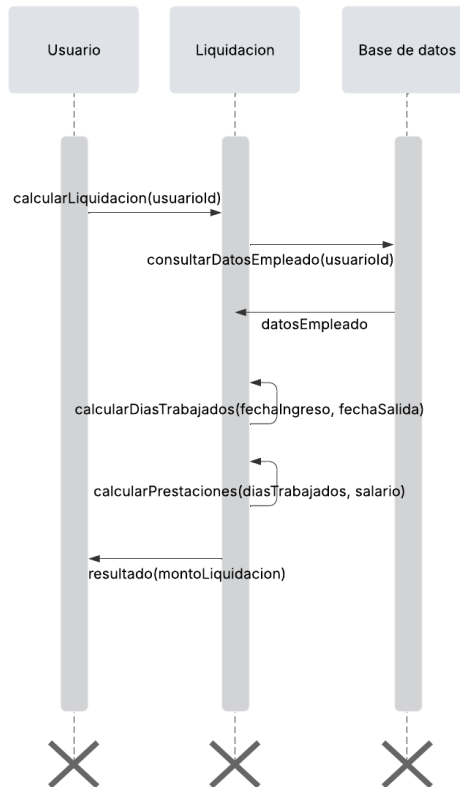
Fuente: Elaboración propia.

El diagrama de secuencia de horas extra muestra cómo el usuario solicita el registro de horas adicionales, las cuales son aprobadas, calculadas y almacenadas, generando además una notificación al usuario.

Figura 42.*Diagrama de secuencia de aguinaldo*

Fuente: Elaboración propia.

Este diagrama representa el proceso de cálculo del aguinaldo, donde se consultan los salarios del usuario, se realizan los cálculos correspondientes y se obtiene el monto final.

Figura 43.*Diagrama de secuencia de liquidación*

Fuente: Elaboración propia.

El diagrama ilustra el proceso de cálculo de liquidación, en el cual se consultan los datos del empleado, se determinan los días trabajados y se calculan las prestaciones correspondientes para obtener el monto final.

PROGRAMACIÓN

Dentro del desarrollo del sistema, la programación constituye un componente fundamental para garantizar el correcto funcionamiento de los módulos implementados. En esta etapa se definieron las estructuras de control, validaciones, consultas a la base de datos y procesamiento de la información mediante el lenguaje PHP, integrando HTML y CSS para la construcción de las interfaces.

El sistema fue desarrollado bajo una arquitectura modular, donde cada funcionalidad (horas extra, vacaciones, incapacidades, permisos, nóminas, liquidaciones y aguinaldos) cuenta con su propia lógica de procesamiento. Se implementaron validaciones tanto del lado del servidor como en la interfaz, con el fin de asegurar la consistencia y confiabilidad de los datos ingresados.

A continuación, se presentan ejemplos representativos del código fuente que evidencian la interacción entre la interfaz de usuario y la base de datos, así como la estructura interna de los formularios y su procesamiento.

Figura 44.

Fragmento de código del módulo de aguinaldos (aguinaldos.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6
7 // Consulta todos los aguinaldos registrados
8 // u = tabla aguinaldos
9 // u = tabla usuarios
10 // Hace un JOIN para traer también el nombre de usuario
11 $stmt = $conn->query("
12     SELECT a.*, u.usuario AS usuario
13     FROM aguinaldos a
14     JOIN usuarios u ON a.usuario_id = u.id
15     ORDER BY fecha DESC
16 ");
17
18 // Guarda todos los registros en un arreglo asociativo
19 $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
20
21 // Si se presionó el botón "eliminar"
22 if(isset($_POST['eliminar'])){
23
24     // Captura el id del registro que se quiere eliminar
25     $id = $_POST['id'];
26
27     // Prepara un DELETE seguro usando parámetro
28     $stmt = $conn->prepare("DELETE FROM aguinaldos WHERE id = :id");
29
30     // Ejecuta la eliminación
31     $stmt->execute(['id' => $id]);
32
33     // Redirige para evitar memoria al refrescar
34     header("Location: aguinaldos.php?msg=RegistroEliminado");
35 }
36
37
38 ?>
39 <!DOCTYPE html>
40 <html lang="es">
41 <head>
42 <?php include_once 'inc/head.php' ?>
43 </head>
44 <body>
45 <div class="wrapper">
46 <?php include_once 'inc/sidebar.php' ?>
47
48 <div class="main">
49
50 <?php include_once 'inc/header.php' ?>
51
52 <main class="content">
53 <div class="container-fluid p-0">
54
55 <h1 class="h3 mb-3">Aguinaldos</h1>
56
57 <div class="row">
58 <div class="col-12">
59 <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
60 <div class="card">
61 <div class="card-body">
62 <a href="crean_aguinaldo.php" class="btn btn-primary">Crear nuevo</a>
63 </div>
64 <table class="table table-hover my-0">
65 <thead>
66 <tr>
67 <th>Empleado</th>

```

Fuente: Elaboración propia.

La Figura 44 presenta un fragmento del archivo `aguinaldos.php`, correspondiente al módulo de gestión de aguinaldos del sistema desarrollado para Titanium Fitness. En este archivo se implementa la lógica encargada de consultar, mostrar y eliminar registros almacenados en la base de datos.

En la parte inicial del código se incluyen los archivos `connection.php` y `verify.php`. El primero permite establecer la conexión con la base de datos MySQL mediante PDO, mientras que el segundo verifica que el usuario tenga una sesión activa, reforzando la seguridad del sistema.

Posteriormente, se ejecuta una consulta SQL que obtiene todos los registros de la tabla `aguinaldos`, realizando un *JOIN* con la tabla `usuarios` para recuperar el nombre del usuario asociado a cada registro. Esta consulta se ordena por fecha, en orden descendente, que permite visualizar los registros más recientes, en primer lugar.

Los resultados de la consulta se almacenan en un arreglo asociativo mediante el método `fetchAll(PDO::FETCH_ASSOC)`, que facilita su posterior recorrido y la visualización en la tabla HTML del módulo.

Asimismo, el código implementa la funcionalidad de eliminación de registros utilizando sentencias preparadas (*prepare*) con parámetros enlazados. Este mecanismo fortalece la seguridad del sistema al prevenir ataques de inyección SQL. Tras ejecutar la eliminación, se realiza una redirección mediante `header()` para evitar el reenvío del formulario al actualizar la página.

Finalmente, el archivo integra la estructura HTML del módulo, incluyendo componentes reutilizables como encabezado, menú lateral y tabla de visualización, conservando la uniformidad en el diseño general del sistema.

Figura 45.

Fragmento de código del módulo de asistencias (*asistencias.php*).

```

asistencias.php
1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6
7 $where = '';
8
9 $rol = $_SESSION['rol'];
10 $usuario_id = $_SESSION['id'];
11
12 if($rol == 'empleado'){
13     $where = "AND usuario_id = ($usuario_id)";
14 }
15
16 $stmt = $conn->query("SELECT a.*, u.usuario AS usuario FROM asistencias a JOIN usuarios u ON a.usuario_id = u.id WHERE 1 $where ORDER BY fecha DESC");
17 $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
18
19 if(isset($_POST['eliminar'])){
20     $id = $_POST['id'];
21
22     $stmt = $conn->prepare("DELETE FROM asistencias WHERE id = :id");
23     $stmt->execute(['id' => $id]);
24
25     header("Location: asistencias.php?msg=Registro+eliminado");
26 }
27
28 ?>
29 <!DOCTYPE html>
30 <html lang="es">
31 <head>
32     <?php include_once 'inc/head.php' ?>
33 </head>
34 <body>
35     <div class="wrapper">
36         <?php include_once 'inc/sidebar.php' ?>
37
38         <div class="main">
39
40             <?php include_once 'inc/header.php' ?>
41
42             <main class="content">
43                 <div class="container-fluid p-0">
44
45                     <h1 class="h3 mb-3">Asistencias</h1>
46
47                     <div class="row">
48                         <div class="col-12">
49                             <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
50                             <div class="card">
51                                 <?php if($rol == 'empleado'): ?>
52                                 <div class="card-body">
53                                     <a href="crear_asistencia.php" class="btn btn-primary">Crear nuevo</a>
54                                 </div>
55                                 <?php endif; ?>
56                                 <table class="table table-hover my-0">
57                                     <thead>
58                                         <tr>
59                                             <th>Empleado</th>
60                                             <th>Tipo</th>
61                                             <th>Fecha</th>
62                                             <th>Acción</th>
63                                         </tr>
64                                     </thead>
65                                     <tbody>
66                                         <?php foreach($rows as $row): ?>

```

Fuente: Elaboración propia.

En este fragmento se desarrolla la lógica encargada de consultar y administrar los registros de asistencia en el sistema.

Primero se establece la conexión con la base de datos y se valida que el usuario tenga una sesión activa. Luego se obtiene el rol y el identificador del usuario desde la sesión. Esta información es importante porque determina qué datos puede visualizar cada persona.

Si el usuario tiene rol de empleado, el sistema agrega una condición adicional en la consulta para que solo pueda ver sus propios registros. En cambio, si es administrador, la consulta devuelve todos los registros almacenados. De esta manera se controla el acceso a la información desde la lógica del sistema.

La consulta realiza una unión (*JOIN*) entre la tabla de asistencias y la tabla de usuarios para poder obtener el nombre del empleado asociado a cada registro. Los datos se ordenan por fecha en orden descendente y se almacenan en un arreglo asociativo para su posterior uso.

Además, se implementa el proceso de eliminación de registros. Cuando se recibe la solicitud por método POST, se captura el identificador del registro y se ejecuta una sentencia preparada para eliminarlo. El uso de sentencias preparadas permite mayor seguridad al evitar inyección SQL. Finalmente, se realiza una redirección para evitar el reenvío del formulario al actualizar la página.

Este proceso demuestra cómo el sistema maneja la consulta, el control de acceso por rol y la eliminación segura de datos dentro del módulo de asistencias.

Figura 46.

Proceso de creación y registro de aguinaldo (crear_aguinaldo.php).

```

1  <?php
2  require_once 'php/connection.php';
3  require_once 'php/verify.php';
4  $error = '';
5  $queryU = $conn->query("SELECT * FROM usuarios ORDER BY apellidos, nombres");
6  $usuarios = $queryU->fetchAll(PDO::FETCH_ASSOC);
7
8  if(isset($_POST['guardar'])){
9      $usuario_id = $_POST['usuario_id'];
10     $periodo = $_POST['periodo'];
11     $fecha_inicio = ($periodo - 1)." -12-01";
12     $fecha_fin = ($periodo)-11-30";
13     $stmt = $conn->prepare("
14         SELECT
15             COUNT(*) AS meses,
16             SUM(salario_bruto) AS salarios
17         FROM nominas
18         WHERE usuario_id = :usuario_id
19         AND fecha BETWEEN :fecha_inicio AND :fecha_fin
20     ");
21
22     $stmt->execute([
23         'usuario_id' => $usuario_id,
24         'fecha_inicio' => $fecha_inicio,
25         'fecha_fin' => $fecha_fin
26     ]);
27
28     $historial = $stmt->fetch(PDO::FETCH_ASSOC);
29
30     $meses_laborados = $historial['meses'];
31
32     $total_salarios = $historial['salarios'];
33
34     if($meses_laborados > 0){
35
36         $monto_aguinaldo = $total_salarios / $meses_laborados;
37
38         $stmt = $conn->prepare("
39             INSERT INTO aguinaldos
40             (usuario_id, periodo, total_salarios, meses_laborados, monto_aguinaldo, fecha)
41             VALUES
42             (:usuario_id, :periodo, :total_salarios, :meses_laborados, :monto_aguinaldo, :fecha)
43         ");
44
45         $stmt->execute([
46             'usuario_id' => $usuario_id,
47             'periodo' => $periodo,
48             'total_salarios' => $total_salarios,
49             'meses_laborados' => $meses_laborados,
50             'monto_aguinaldo' => $monto_aguinaldo,
51             'fecha' => date('Y-m-d H:i:s') // Fecha actual del registro
52         ]);
53
54         header('Location: aguinaldos.php?msg=Registro+creado');
55     }else{
56
57         // Si no tiene nóminas en ese periodo muestra error
58         $error = 'El empleado no tiene nominas registradas';
59     }
60 }
61
62 }

```

Fuente: Elaboración propia.

En este fragmento se implementa el proceso para generar un aguinaldo a partir de la información registrada en las nóminas del sistema.

Primero se carga la lista de usuarios desde la base de datos para seleccionar el empleado al que se le va a calcular el aguinaldo. Luego, cuando se envía el formulario (acción guardar), el sistema recibe el *usuario_id* y el periodo.

Con el periodo ingresado, se construye el rango de fechas que se va a usar para el cálculo (del 1 de diciembre del año anterior al 30 de noviembre del año seleccionado). Después se ejecuta una consulta que obtiene dos datos clave desde la tabla de nóminas:

- la cantidad de meses con nómina registrada (COUNT(*))
- y la suma de los salarios brutos (SUM(salario_bruto))

Con esos resultados, el sistema calcula el aguinaldo dividiendo el total de salarios entre los meses laborados. Antes de guardar, se valida que el empleado realmente tenga meses registrados en ese periodo, para evitar cálculos incorrectos.

Finalmente, si todo está correcto, se inserta el registro en la tabla aguinaldos mediante una sentencia preparada, guardando el usuario, periodo, total de salarios, meses laborados, monto calculado y la fecha del registro. Al finalizar, se redirige al listado de aguinaldos con un mensaje de confirmación. En caso de que no existan nóminas en ese periodo, se muestra un mensaje de error para indicar que el empleado no tiene nóminas registradas.

Figura 47.

Proceso de registro de asistencia (*crear_asistencia.php*).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4 $error = '';
5
6 // Consulta todos los usuarios ordenados por apellidos y nombres
7 $queryU = $conn->query("SELECT * FROM usuarios ORDER BY apellidos, nombres");
8
9 // Convierte el resultado en un arreglo asociativo (lista de usuarios)
10 $usuarios = $queryU->fetchAll(PDO::FETCH_ASSOC);
11
12 // Consulta todos los periodos (por ejemplo: quincena, semana, etc.) ordenados por nombre
13 $queryP = $conn->query("SELECT * FROM periodos ORDER BY nombre ASC");
14
15 // Convierte el resultado en un arreglo asociativo (lista de periodos)
16 $periodos = $queryP->fetchAll(PDO::FETCH_ASSOC);
17
18 // Si se presionó el botón "guardar" del formulario
19 if(isset($_POST['guardar'])){
20
21     // Toma el id del usuario logueado desde la sesión (quien está registrando la asistencia)
22     $usuario_id = $_SESSION['id'];
23
24     // Toma el tipo de asistencia enviado por el formulario (ej: entrada / salida)
25     $tipo = $_POST['tipo'];
26
27     // Prepara un INSERT seguro para registrar la asistencia
28     $stmt = $conn->prepare("INSERT INTO asistencias (usuario_id, tipo, fecha) VALUES (:usuario_id, :tipo, :fecha)");
29
30     // Ejecuta el INSERT enviando los valores como parámetros
31     $stmt->execute([
32         'usuario_id' => $usuario_id,
33         'tipo' => $tipo,
34
35         // Guarda la fecha y hora actual del registro
36         'fecha' => date('Y-m-d H:i:s')
37     ]);
38
39     // Redirige para evitar que al recargar se vuelva a insertar el mismo registro
40     header('Location: asistencias.php?msg=Registro+creado');
41 }
42
43
44 >
45 <!DOCTYPE html>
46 <html lang="es">
47 <head>
48     <?php include_once 'inc/head.php' ?>
49 </head>
50 <body>
51     <div class="wrapper">
52         <?php include_once 'inc/sidebar.php' ?>
53
54         <div class="main">
55             <?php include_once 'inc/header.php' ?>
56
57             <main class="content">
58                 <div class="container-fluid p-0">
59
60                     <h1 class="h3 mb-3">Asistencias</h1>
61
62                     <div class="row">
63                         <div class="col-12">
64                             <div class="card">
65                                 <div class="card-header">
66                                     <h5 class="card-title mb-0">Crear nuevo</h5>
67

```

Fuente: Elaboración propia.

En este fragmento se desarrolla la lógica para registrar una nueva marca de asistencia en el sistema.

Al inicio se establece la conexión con la base de datos y se valida la sesión del usuario. Luego se realiza una consulta para obtener la lista de usuarios y los periodos disponibles, los cuales se cargan desde la base de datos y se almacenan en arreglos asociativos. Esta información sirve como base para el registro.

Cuando el formulario es enviado mediante la acción guardar, el sistema toma el identificador del usuario desde la sesión, es decir, la persona que está realizando la marca. Además, recibe el tipo de asistencia enviado desde el formulario, que puede ser entrada o salida.

Con estos datos se prepara una sentencia INSERT utilizando parámetros para registrar la asistencia en la tabla correspondiente. Además, se guarda automáticamente la fecha y hora actual del sistema utilizando la función *date()*, lo que permite llevar un control preciso del momento en que se realizó la marca.

Una vez ejecutado el registro, el sistema redirige nuevamente al módulo de asistencias. Esta redirección se realiza para evitar que, al actualizar la página, se vuelva a insertar el mismo registro.

Este proceso refleja cómo el sistema gestiona el registro de entradas y salidas de manera controlada, asegurando que cada marca quede asociada al usuario autenticado y registrada con su respectiva fecha y hora.

Figura 48.

Proceso de registro de incapacidades (crear_incapacidad.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4 $error = '';
5
6 // Consulta todos los usuarios para llenar el <select> de empleados
7 $queryU = $conn->query("SELECT * FROM usuarios ORDER BY apellidos, nombres");
8
9 // Convierte el resultado en un arreglo asociativo (lista de usuarios)
10 $usuarios = $queryU->fetchAll(PDO::FETCH_ASSOC);
11
12 // Si se presionó el botón "guardar" del formulario
13 if(isset($_POST['guardar'])){
14
15     // Captura el empleado seleccionado
16     $usuario_id = $_POST['usuario_id'];
17
18     // Captura el tipo de incapacidad
19     $tipo = $_POST['tipo'];
20
21     // Captura la fecha de inicio y fin desde el formulario
22     $fecha_inicio = $_POST['fecha_inicio'];
23     $fecha_fin = $_POST['fecha_fin'];
24
25     // Convierte las fechas (string) a objetos DateTime para poder calcular diferencia
26     $inicio = new DateTime($fecha_inicio);
27     $fin = new DateTime($fecha_fin);
28
29     // Calcula los días entre fechas:
30     // diff()->days devuelve la diferencia en días SIN incluir el día inicial,
31     // por eso se suma +1 para contar ambos días (inicio y fin)
32     $dias = $inicio->diff($fin)->days + 1;
33
34     // Prepara el INSERT para guardar la incapacidad
35     $stmt = $conn->prepare("
36         INSERT INTO incapacidades
37         (usuario_id, tipo, fecha_inicio, fecha_fin, dias, fecha)
38         VALUES
39         (:usuario_id, :tipo, :fecha_inicio, :fecha_fin, :dias, :fecha)
40     ");
41
42     //
43
44
45 }
46 <!DOCTYPE html>
47 <html lang="es">
48 <head>
49 <?php include_once 'inc/head.php' ?>
50 </head>
51 <body>
52 <div class="wrapper">
53 <?php include_once 'inc/sidebar.php' ?>
54
55 <div class="main">
56
57 <?php include_once 'inc/header.php' ?>
58
59 <main class="content">
60 <div class="container-fluid p-0">
61
62 <h1 class="h3 mb-3">Incapacidades</h1>
63
64 <div class="row">
65 <div class="col-12">
66 <div class="card">
67 <div class="card-header">

```

Fuente: Elaboración propia.

En este fragmento se implementa el proceso para registrar una incapacidad dentro del sistema.

Primero se establece la conexión con la base de datos y se valida que el usuario tenga una sesión activa. Después se consulta la tabla de usuarios para obtener la lista de empleados y seleccionar a quién se le va a registrar la incapacidad.

Cuando se presiona guardar, el sistema captura los datos principales: el empleado seleccionado (`usuario_id`), el tipo de incapacidad (`tipo`) y las fechas de inicio y fin. Con esas fechas, el sistema realiza el cálculo de los días de incapacidad.

Para calcular los días, se convierten las fechas a objetos `DateTime` y se obtiene la diferencia entre ambas. Luego se suma 1, porque se está contando tanto el día de inicio como el día final (es decir, se registran días completos dentro del rango).

Con el total de días ya calculado, se prepara una sentencia `INSERT` con parámetros para guardar el registro en la tabla `incapacidades`, incluyendo el usuario, el tipo, las fechas, la cantidad de días y la fecha del registro. Este uso de consultas preparadas ayuda a mantener el proceso más seguro y ordenado al momento de enviar datos a la base.

Figura 49.

Proceso de registro de liquidación (crear_liquidacion.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4 $error = '';
5
6 // Consulta todos los usuarios para llenar el <select> de empleados
7 // Ordena por apellidos y nombres
8 $queryU = $conn->query("SELECT * FROM usuarios ORDER BY apellidos, nombres");
9
10 // Trae todos los registros como arreglo asociativo (cada usuario es un array)
11 $usuarios = $queryU->fetchAll(PDO::FETCH_ASSOC);
12
13 // Si se presionó el botón "Guardar" (submit) se ejecuta el registro de liquidación
14 if(isset($_POST['guardar']))){
15     $usuario_id = $_POST['usuario_id'];
16     $motivo = $_POST['motivo'];
17     $dias_trabajados = $_POST['dias_trabajados'];
18     $dias_vacaciones = $_POST['dias_vacaciones'];
19     $meses_trabajados = $_POST['meses_trabajados'];
20     $stmt = $conn->prepare("SELECT sueldo_base FROM usuarios WHERE id = :usuario_id");
21     $stmt->execute(['usuario_id' => $usuario_id]);
22     $sueldo_base = $stmt->fetchColumn();
23     $valor_dia = $sueldo_base / 30;
24     $salario_pendiente = $valor_dia * $dias_trabajados;
25     $vacaciones = $valor_dia * $dias_vacaciones;
26     $aguinaldo = ($sueldo_base / 12) * $meses_trabajados;
27     $preaviso = ($motivo == 'despido_con_resp') ? $sueldo_base : 0;
28     $cesantia = ($motivo == 'despido_con_resp') ? $sueldo_base : 0;
29     $total = $salario_pendiente + $vacaciones + $aguinaldo + $preaviso + $cesantia;
30     $stmt = $conn->prepare("
31     INSERT INTO liquidaciones
32     (usuario_id, motivo, dias_trabajados, salario_pendiente, dias_vacaciones, vacaciones, meses_trabajados, aguinaldo, preaviso, cesantia, total, fecha)
33     VALUES
34     (:usuario_id, :motivo, :dias_trabajados, :salario_pendiente, :dias_vacaciones, :vacaciones, :meses_trabajados, :aguinaldo, :preaviso, :cesantia, :total, :fecha)
35     ");
36     $stmt->execute([
37         'usuario_id' => $usuario_id,
38         'motivo' => $motivo,
39         'dias_trabajados' => $dias_trabajados,
40         'salario_pendiente' => $salario_pendiente,
41         'dias_vacaciones' => $dias_vacaciones,
42         'vacaciones' => $vacaciones,
43         'meses_trabajados' => $meses_trabajados,
44         'aguinaldo' => $aguinaldo,
45         'preaviso' => $preaviso,
46         'cesantia' => $cesantia,
47         'total' => $total,
48         'fecha' => date("Y-m-d H:i:s")
49     ]);
50     header('Location: liquidaciones.php?msg=Registro+creado');
51 }
52 }
53 }
54 }
55 <!DOCTYPE html>
56 <html lang="es">
57 <head>
58 <?php include_once 'inc/head.php' ?>
59 </head>
60 <body>
61 <div class="wrapper">
62 <?php include_once 'inc/sidebar.php' ?>
63
64 <div class="main">
65
66 <?php include_once 'inc/header.php' ?>
67

```

Fuente: Elaboración propia.

En este proceso se desarrolla la lógica necesaria para calcular y registrar una liquidación de un empleado.

Primero se establece la conexión con la base de datos y se valida la sesión activa. Luego se consulta la tabla de usuarios para obtener la lista de empleados disponibles, ordenados por apellidos y nombres.

Cuando se presiona el botón Guardar, el sistema captura los datos ingresados en el formulario: el empleado seleccionado, el motivo de la liquidación, los días trabajados, los días de vacaciones pendientes y los meses trabajados.

Posteriormente, el sistema consulta el sueldo base del empleado desde la base de datos. A partir de ese valor se calcula el salario diario, dividiendo el sueldo base entre 30 días. Con ese dato se determinan:

- El salario pendiente, multiplicando el salario diario por los días trabajados.
- El monto correspondiente a vacaciones, multiplicando el salario diario por los días de vacaciones.
- El aguinaldo proporcional, calculado como el sueldo base dividido entre 12 y multiplicado por los meses trabajados.

Además, se evalúa el motivo de la liquidación. Si el motivo corresponde a un despido con responsabilidad patronal, se incluyen los montos de preaviso y cesantía; en caso contrario, estos valores se establecen en cero.

Finalmente, se calcula el total sumando salario pendiente, vacaciones, aguinaldo, preaviso y cesantía. Con todos los valores definidos, se prepara una sentencia INSERT para guardar la liquidación en la base de datos junto con la fecha del registro.

Este proceso concentra la parte más importante del módulo, porque automatiza los cálculos que normalmente se realizarían de forma manual, reduciendo errores y asegurando mayor claridad en el desglose de la liquidación.

Figura 50.

Proceso de generación de nómina (crear_nomina.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4 $error = '';
5 $queryU = $conn->query("SELECT * FROM usuarios ORDER BY apellidos, nombres");
6 $usuarios = $queryU->fetchAll(PDO::FETCH_ASSOC);
7 $queryP = $conn->query("SELECT * FROM periodos ORDER BY nombre ASC");
8 $periodos = $queryP->fetchAll(PDO::FETCH_ASSOC);
9 if(isset($_POST['guardar'])){
10     $usuario_id = $_POST['usuario_id'];
11     $periodo_id = $_POST['periodo_id'];
12     $stmt = $conn->prepare("SELECT sueldo_base FROM usuarios WHERE id = :usuario_id");
13     $stmt->execute(['usuario_id' => $usuario_id]);
14     $sueldo_base = $stmt->fetchColumn();
15     $stmt = $conn->prepare("SELECT * FROM periodos WHERE id = :periodo_id");
16     $stmt->execute(['periodo_id' => $periodo_id]);
17     $periodo = $stmt->fetch(PDO::FETCH_ASSOC);
18     $stmt = $conn->prepare("
19         SELECT SUM(horas)
20         FROM horas_extra
21         WHERE usuario_id = :usuario_id
22             AND estado = 'aprobado'
23             AND fecha BETWEEN :periodo_inicio AND :periodo_fin
24     ");
25     $stmt->execute([
26         'usuario_id' => $usuario_id,
27         'periodo_inicio' => $periodo['fecha_inicio'],
28         'periodo_fin' => $periodo['fecha_fin']
29     ]);
30     $horas_extra = $stmt->fetchColumn() ? $stmt->fetchColumn() : 0;
31     $valor_hora = $sueldo_base / 240;
32     $pago_horas_extra = $horas_extra * ($valor_hora * 1.5);
33     $salario_bruto = $sueldo_base + $pago_horas_extra;
34     $ccss = $salario_bruto * 0.0034;
35     $banco_popular = $salario_bruto * 0.01;
36     $total_deducciones = $ccss + $banco_popular;
37     $neto_pagar = $salario_bruto - $total_deducciones;
38     $stmt = $conn->prepare("
39         INSERT INTO nominas
40         (usuario_id, periodo_id, sueldo_base, horas_extra, pago_horas_extra, salario_bruto, ccss, banco_popular, total_deducciones, neto_pagar, fecha)
41         VALUES
42         (:usuario_id, :periodo_id, :sueldo_base, :horas_extra, :pago_horas_extra, :salario_bruto, :ccss, :banco_popular, :total_deducciones, :neto_pagar, :fecha)
43     ");
44     $stmt->execute([
45         'usuario_id' => $usuario_id,
46         'periodo_id' => $periodo_id,
47         'sueldo_base' => $sueldo_base,
48         'horas_extra' => $horas_extra,
49         'pago_horas_extra' => $pago_horas_extra,
50         'salario_bruto' => $salario_bruto,
51         'ccss' => $ccss,
52         'banco_popular' => $banco_popular,
53         'total_deducciones' => $total_deducciones,
54         'neto_pagar' => $neto_pagar,
55         'fecha' => date("Y-m-d H:i:s")
56     ]);
57     header("Location: nominas.php?msg=registrocreado");
58 }
59 }
60 }
61 }
62 }
63 <DOCTYPE html
64 <html lang="es"
65 <head
66 <?php include_once 'inc/head.php' ?>
67 </head

```

En este fragmento se implementa el proceso para generar una nómina y dejarla registrada en la base de datos.

En primer lugar, se realiza la conexión a la base de datos y se valida la sesión activa. Después, el sistema consulta la lista de usuarios y los periodos disponibles, porque la nómina se genera para un empleado específico dentro de un rango de fechas definido.

Cuando se presiona el botón “Guardar”, el sistema captura el empleado seleccionado (*usuario_id*) y el periodo (*periodo_id*). Con esa información, se consulta el sueldo base del empleado y también se obtiene el detalle del periodo, principalmente sus fechas de inicio y fin.

A partir del periodo seleccionado, el sistema consulta si existen horas extra aprobadas para ese empleado dentro del rango de fechas. Si no se encuentran registros, se asigna cero horas extra para evitar errores en el cálculo.

Luego se realizan los cálculos principales:

- Se calcula el valor de la hora a partir del sueldo base (dividido entre 240).
- Se determina el pago de horas extra aplicando el factor correspondiente (por ejemplo, 1.5).

- Se obtiene el salario bruto sumando el sueldo base y el pago de horas extras.
- Se calculan las deducciones (por ejemplo, CCSS y Banco Popular) como porcentajes del salario bruto.
- Finalmente, se calcula el neto a pagar restando el total de deducciones al salario bruto.

Con todos los montos listos, se prepara una sentencia *INSERT* para guardar la nómina en la tabla correspondiente, incluyendo el empleado, el periodo, sueldo base, horas extra, deducciones, neto a pagar y la fecha/hora de creación del registro.

Al finalizar, el sistema redirige al listado de nóminas para confirmar que el registro fue creado y evitar que se genere nuevamente si se recarga la página

Figura 51.

Proceso de registro de solicitud de horas extra (crear_solicitud_he.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $error = '';
6
7 if(isset($_POST['guardar'])){
8     $fecha = $_POST['fecha'];
9     $horas = $_POST['horas'];
10    $motivo = trim($_POST['motivo']);
11    $usuario_id = $_SESSION['id'];
12    $fecha_solicitud = date('Y-m-d H:i:s');
13
14    $stmt = $conn->prepare("INSERT INTO horas_extra (usuario_id, fecha, horas, motivo, fecha_solicitud) VALUES (:usuario_id, :fecha, :horas, :motivo, :fecha_solicitud)");
15    $stmt->execute([
16        'usuario_id' => $usuario_id,
17        'fecha' => $fecha,
18        'horas' => $horas,
19        'motivo' => $motivo,
20        'fecha_solicitud' => $fecha_solicitud
21    ]);
22
23    $stmt = $conn->query("SELECT correo_administrador FROM ajustes");
24    $correo_administrador = $stmt->fetchColumn();
25
26    $usuario = $_SESSION['usuario'];
27
28    $mensaje = "<h3>Solicitud de horas extras</h3>";
29    $mensaje .= "<p>El usuario {$usuario} ha solicitado {$horas} horas extras del día {$fecha} con el motivo {$motivo}</p>";
30    $mensaje .= "<p>Por favor aprobar o rechazar en el sistema.</p>";
31
32    enviarCorreo($correo_administrador, $mensaje);
33
34    header("Location: solicitudes_he.php?msg=Solicitud+creada");
35
36 }
37
38 ?>
39 <!DOCTYPE html>
40 <html lang="es">
41 <head>
42     <?php include_once 'inc/head.php' ?>
43 </head>
44 <body>
45     <div class="wrapper">
46         <?php include_once 'inc/sidebar.php' ?>
47     </div>
48     <div class="main">
49         <?php include_once 'inc/header.php' ?>
50     </div>
51     <main class="content">
52         <div class="container-fluid p-0">
53             <h1 class="h3 mb-3">Solicitud de horas extra</h1>
54             <div class="row">
55                 <div class="col-12">
56                     <div class="card">
57                         <div class="card-header">
58                             <h5 class="card-title mb-0">Crear nuevo</h5>
59                         </div>
60                         <div class="card-body">
61                             <div class="text-danger mb-2 fw-bold"><?php echo $error ?></div>
62                             <form method="POST">
63                                 <div class="row">
64                                     <div class="col-md-3">

```

Fuente: Elaboración propia.

En este archivo se controla el proceso para que un trabajador pueda registrar una solicitud de horas extra.

Primero se carga la conexión a la base de datos y se valida que el usuario tenga sesión activa. Cuando el formulario se envía (al presionar Guardar), el sistema toma los datos ingresados: fecha, cantidad de horas y motivo. Además, se obtiene automáticamente el *id* del usuario desde la sesión y se genera la fecha y hora de solicitud con la fecha actual del sistema.

Con esa información, se prepara un *INSERT* para guardar la solicitud en la tabla *horas_extras*, registrando quién la solicitó, qué día, cuántas horas pidió, el motivo y cuándo se creó la solicitud.

Después de guardar el registro, el sistema consulta en la tabla **ajustes** el **correo del administrador** (*correo_administrador*). Con ese correo se arma un mensaje de notificación indicando el usuario que solicitó las horas extra, la cantidad de horas, la fecha y el motivo, y se envía el correo mediante la función *enviarCorreo(...)*.

Finalmente, se redirige al listado de solicitudes con un mensaje de confirmación, para evitar que al recargar la página se vuelva a registrar la misma solicitud.

Figura 52.

Proceso de registro de solicitud de permiso (crear_solicitud_permiso.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $error = '';
6
7 if(isset($_POST['guardar'])){
8     $fecha_inicio = $_POST['fecha_inicio'];
9     $fecha_fin = $_POST['fecha_fin'];
10    $motivo = $_POST['motivo'];
11
12    $usuario_id = $_SESSION['id'];
13    $fecha_solicitud = date("Y-m-d H:i:s");
14
15    $stmt = $conn->prepare("INSERT INTO permisos (usuario_id, fecha_inicio, fecha_fin, motivo, fecha_solicitud) VALUES (:usuario_id, :fecha_inicio, :fecha_fin, :motivo, :fecha_solicitud)");
16    $stmt->execute([
17        'usuario_id' => $usuario_id,
18        'fecha_inicio' => $fecha_inicio,
19        'fecha_fin' => $fecha_fin,
20        'motivo' => $motivo,
21        'fecha_solicitud' => $fecha_solicitud
22    ]);
23
24    $stmt = $conn->query("SELECT correo_administrador FROM ajustes");
25    $correo_administrador = $stmt->fetchColumn();
26
27    $usuario = $_SESSION['usuario'];
28
29    $mensaje = "<div>Solicitud de permiso/</div>";
30    $mensaje .= "<p>El usuario ($usuario) ha solicitado permiso desde el día ($fecha_inicio) hasta el día ($fecha_fin) con motivo ($motivo).</p>";
31    $mensaje .= "<p>Por favor aprobar o rechazar en el sistema.</p>";
32
33    enviarCorreo($correo_administrador, $mensaje);
34
35    header('Location: solicitudes_permisos.php?msg=Solicitud+creada');
36
37 }
38
39 ?>
40 <!DOCTYPE html>
41 <html lang="es">
42 <head>
43     <?php include_once 'inc/head.php' ?>
44 </head>
45 <body>
46     <div class="wrapper">
47         <?php include_once 'inc/sidebar.php' ?>
48     <div class="main">
49         <?php include_once 'inc/header.php' ?>
50     <main class="content">
51         <div class="container-fluid p-0">
52             <h1 class="h3 mb-3">Solicitud de permiso</h1>
53             <div class="row">
54                 <div class="col-12">
55                     <div class="card">
56                         <div class="card-header">
57                             <h5 class="card-title mb-0">Crear nuevo</h5>
58                         </div>
59                         <div class="card-body">
60                             <div class="text-danger mb-2 fw-bold"><?php echo $error ?></div>
61                             <form method="POST">
62                                 <div class="row">

```

Fuente: Elaboración propia.

Este archivo se encarga de registrar una solicitud de permiso realizada por un trabajador.

Al inicio se incluye la conexión a la base de datos y la validación de sesión. Cuando el usuario envía el formulario (botón “Guardar”), el sistema toma los datos ingresados: fecha de inicio, fecha de fin y motivo. Además, obtiene el *id* del usuario desde la sesión y genera automáticamente la fecha y hora de solicitud con la fecha actual.

Con esos datos, se ejecuta un *INSERT* en la tabla permisos, almacenando el usuario que solicita, el rango de fechas, el motivo y el momento exacto en que se registró la solicitud.

Luego, el sistema consulta en la tabla ajustes el correo del administrador y construye un mensaje donde se indica quién solicitó el permiso, desde qué fecha hasta cuál fecha y el motivo. Ese mensaje se envía mediante la función `enviarCorreo(...)`, para notificar al responsable y facilitar la revisión.

Finalmente, se redirige al listado de solicitudes de permisos con un mensaje de confirmación, evitando que se duplique el registro si la página se recarga.

Figura 53.

Proceso de registro de solicitud de vacaciones (crear_solicitud_vacaciones.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $error = '';
6
7 if(isset($_POST['guardar'])){
8     $fecha_inicio = $_POST['fecha_inicio'];
9     $fecha_fin = $_POST['fecha_fin'];
10
11     $usuario_id = $_SESSION['id'];
12     $fecha_solicitud = date('Y-m-d H:i:s');
13
14     $stmt = $conn->prepare("INSERT INTO vacaciones (usuario_id, fecha_inicio, fecha_fin, fecha_solicitud) VALUES (:usuario_id, :fecha_inicio, :fecha_fin, :fecha_solicitud)");
15     $stmt->execute([
16         'usuario_id' => $usuario_id,
17         'fecha_inicio' => $fecha_inicio,
18         'fecha_fin' => $fecha_fin,
19         'fecha_solicitud' => $fecha_solicitud
20     ]);
21
22     $stmt = $conn->query("SELECT correo_administrador FROM ajustes");
23     $correo_administrador = $stmt->fetchColumn();
24
25     $usuario = $_SESSION['usuario'];
26
27     $mensaje = "<h3>Solicitud de vacaciones</h3>";
28     $mensaje .= "<p>El usuario ($usuario) ha solicitado vacaciones desde el día ($fecha_inicio) hasta el día ($fecha_fin).</p>";
29     $mensaje .= "<p>Por favor aprobar o rechazar en el sistema.</p>";
30
31     enviarCorreo($correo_administrador, $mensaje);
32
33     header('Location: solicitudes_vacaciones.php?msg=Solicitud+creada');
34
35 }
36
37 ?>
38 <!DOCTYPE html>
39 <html lang="es">
40 <head>
41 <?php include_once 'inc/head.php' ?>
42 </head>
43 <body>
44 <div class="wrapper">
45 <?php include_once 'inc/sidebar.php' ?>
46
47 <div class="main">
48
49 <?php include_once 'inc/header.php' ?>
50
51 <main class="content">
52 <div class="container-fluid p-0">
53
54 <h1 class="h3 mb-3">Solicitud de vacaciones</h1>
55
56 <div class="row">
57 <div class="col-12">
58 <div class="card">
59 <div class="card-header">
60 <h5 class="card-title mb-0">Crear nuevo</h5>
61 </div>
62 <div class="card-body">
63 <div class="text-danger mb-2 fu-bold"><?php echo $error ?></div>
64 <form method="POST">
65 <div class="row">
66 <div class="col-md-3">
67 <div class="mb-3">

```

Fuente: Elaboración propia.

Este archivo se encarga de guardar una solicitud de vacaciones hecha por el trabajador.

Primero se carga la conexión a la base de datos y se valida que la sesión esté activa. Cuando el usuario presiona “Guardar”, el sistema toma las fechas ingresadas (fecha de inicio y fecha de fin). Luego obtiene el id del usuario desde la sesión y genera automáticamente la fecha y hora de la solicitud con la fecha actual.

Con esa información se ejecuta un *INSERT* en la tabla vacaciones, almacenando quién solicitó, el rango de fechas solicitado y el momento exacto en que se registró la solicitud.

Después, el sistema consulta en la tabla ajustes el correo del administrador y arma un mensaje donde se indica el usuario que solicitó vacaciones y el periodo solicitado. Ese mensaje se envía con la función `enviarCorreo(...)` para notificar al responsable y que pueda aprobar o rechazar desde el sistema.

Finalmente, se redirige al listado de solicitudes de vacaciones con un mensaje de confirmación, evitando que se registre la misma solicitud si la página se recarga.

Figura 54.

Proceso de creación de usuarios (crear_usuario.php).

```

1  crear_usuario.php
2  <?php
3  require_once 'php/connection.php';
4  require_once 'php/verify.php';
5  $error = '';
6
7  // Verifica si el formulario fue enviado (porque existe el botón/flag "guardar" en el POST)
8  if(isset($_POST['guardar'])){
9
10     // Captura lo que el usuario escribió en el formulario (inputs)
11     $nombres = $_POST['nombres'];
12     $apellidos = $_POST['apellidos'];
13     $usuario = $_POST['usuario'];
14     $contrasena = $_POST['contrasena'];
15     $rol = $_POST['rol'];
16
17     // Prepara una consulta para buscar si ya existe un usuario con ese "usuario"
18     // Usan usuario evita concatenar strings y ayuda a prevenir SQL Injection (siempre que se use correctamente)
19     $stmt = $conn->prepare("SELECT * FROM usuarios WHERE usuario = :usuario");
20
21     // Ejecuta la consulta enviando el valor real de $usuario al parámetro :usuario
22     $stmt->execute(['usuario' => $usuario]);
23
24     // Si no encontró registros (o sea, no existe ese usuario en la base)
25     if($stmt->rowCount() == 0){
26
27         // Prepara el INSERT para crear un nuevo usuario en la tabla usuarios
28         $stmt = $conn->prepare("INSERT INTO usuarios (nombres, apellidos, usuario, contrasena, rol) VALUES (:nombres, :apellidos, :usuario, :contrasena, :rol)");
29
30         // Ejecuta el INSERT pasando cada valor a su parámetro correspondiente
31         $stmt->execute([
32             'nombres' => $nombres,
33             'apellidos' => $apellidos,
34             'usuario' => $usuario,
35             'contrasena' => $contrasena,
36             'rol' => $rol
37         ]);
38
39         // Redirecciona a usuarios.php enviando un mensaje en la URL
40         header('location: usuarios.php?msg=Usuario+creado');
41
42     } else {
43         // Si sí encontró registros, significa que ya existe ese "usuario"
44         // Guarda un mensaje de error para mostrarlo arriba del formulario
45         $error = 'El campo usuario ya se encuentra registrado.';
46     }
47 }
48
49 // Cierra el bloque PHP antes de iniciar el HTML
50 ?>
51 <!DOCTYPE html>
52 <html lang="es">
53 <head>
54     <php include_once 'inc/head.php' ?>
55     <!-- Inserta contenido común del <head> (título, meta tags, CSS, etc.) -->
56 </head>
57 <body>
58     <div class="wrapper">
59         <php include_once 'inc/sidebar.php' ?>
60         <!-- Carga el menú lateral del sistema -->
61
62         <div class="main">
63
64             <php include_once 'inc/header.php' ?>
65             <!-- Carga el header superior (barra, usuario logueado, etc.) -->
66
67             <main class="content">

```

Fuente: Elaboración propia.

Este archivo se encarga de registrar un nuevo usuario dentro del sistema. Al inicio se incluye la conexión a la base de datos y la validación de sesión. Cuando el administrador envía el formulario (botón “Guardar”), el sistema toma los datos ingresados: nombres, apellidos, usuario, contraseña y rol.

Antes de guardar, se hace una verificación importante: se consulta la tabla usuarios para confirmar si ya existe un registro con el mismo nombre de usuario. Si no se encuentra ninguno, se ejecuta el *INSERT* para crear el usuario con los datos capturados.

Finalmente, el sistema redirige al listado de usuarios mostrando un mensaje de confirmación. En caso de que el usuario ya exista, no se inserta nada y se guarda un mensaje de error para indicarle al administrador que ese usuario ya está registrado.

Figura 55.

Proceso de edición de usuarios (editar_usuario.php).

```

1  <?php
2  require_once 'php/connection.php';
3  require_once 'php/verify.php';
4
5  $error = '';
6
7  // Verifica que venga un id por GET (por ejemplo: editar_usuario.php?id=5) y que no venga vacío
8  if(isset($_GET['id']) && !empty($_GET['id'])) {
9
10     // Guarda el id recibido por la URL
11     $id = $_GET['id'];
12
13     // Prepara una consulta para traer el usuario de la base de datos según el id
14     $stmt = $conn->prepare("SELECT * FROM usuarios WHERE id = :id");
15
16     // Ejecuta la consulta pasando el id al parámetro :id
17     $stmt->execute(['id' => $id]);
18
19     // Obtiene el registro con un arreglo asociativo (columnas => valores)
20     $row = $stmt->fetch(PDO::FETCH_ASSOC);
21
22 } else {
23     // Si no viene id, redirige a la lista de usuarios para evitar entrar a editar sin referencia
24     header("Location: usuarios.php");
25 }
26
27 if(isset($_POST['guardar'])) {
28
29     // Captura lo que el usuario escribió en el formulario (inputs)
30     $nombres = $_POST['nombres'];
31     $apellidos = $_POST['apellidos'];
32     $usuario = $_POST['usuario'];
33     $rol = $_POST['rol'];
34
35     // Prepara la consulta UPDATE para actualizar los datos del usuario (excepto contraseña en este caso)
36     $stmt = $conn->prepare("UPDATE usuarios SET nombres = :nombres, apellidos = :apellidos, usuario = :usuario, rol = :rol WHERE id = :id");
37
38     // Ejecuta el UPDATE pasando los valores a sus parámetros
39     $stmt->execute([
40         'nombres' => $nombres,
41         'apellidos' => $apellidos,
42         'usuario' => $usuario,
43         'rol' => $rol,
44         'id' => $id
45     ]);
46     header("Location: usuarios.php?msg=Usuarioactualizado");
47 }
48
49
50
51 ?>
52 <!DOCTYPE html>
53 <html lang="es">
54 <head>
55     <?php include_once 'inc/head.php' ?>
56
57 </head>
58 <body>
59     <div class="wrapper">
60         <?php include_once 'inc/sidebar.php' ?>
61
62         <div class="main">
63
64             <?php include_once 'inc/header.php' ?>
65
66

```

Fuente: Elaboración propia.

Este archivo permite modificar la información de un usuario ya existente en el sistema. Primero se valida que en la URL venga un *id* válido. Si no se recibe ese parámetro, el sistema redirige automáticamente al listado de usuarios para evitar errores. Cuando el *id* está presente,

se realiza una consulta a la base de datos para obtener los datos actuales del usuario, acceder a ellos y modificarlos.

Si el formulario es enviado (botón Guardar), el sistema captura los nuevos valores ingresados: nombres, apellidos, usuario y rol. Luego se prepara una sentencia *UPDATE* para modificar esos datos en la tabla usuarios, utilizando el *id* como referencia para asegurarse de que se actualice el registro correcto.

Finalmente, después de ejecutar la actualización, el sistema redirige al listado de usuarios mostrando un mensaje de confirmación, para indicar que el usuario fue actualizado correctamente

Figura 56.

Proceso de listado y eliminación de incapacidades (incapacidades.php).

```

1  <?php
2  require_once 'php/connection.php';
3  require_once 'php/verify.php';
4
5  $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6
7  $stmt = $conn->query("SELECT i.*, u.usuario AS usuario FROM incapacidades i JOIN usuarios u ON i.usuario_id = u.id ORDER BY fecha DESC");
8  $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
9
10 if(isset($_POST['eliminar'])){
11     $id = $_POST['id'];
12
13     $stmt = $conn->prepare("DELETE FROM incapacidades WHERE id = :id");
14     $stmt->execute(['id' => $id]);
15
16     header('Location: incapacidades.php?msg=Registro+eliminado');
17 }
18
19 >
20 <!DOCTYPE html>
21 <html lang="es">
22 <head>
23     <?php include_once 'inc/head.php' ?>
24 </head>
25 <body>
26     <div class="wrapper">
27         <?php include_once 'inc/sidebar.php' ?>
28
29         <div class="main">
30
31             <?php include_once 'inc/header.php' ?>
32
33             <main class="content">
34                 <div class="container-fluid p-0">
35
36                     <h1 class="h3 mb-3">Incapacidades</h1>
37
38                     <div class="row">
39                         <div class="col-12">
40                             <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
41                             <div class="card">
42                                 <div class="card-body">
43                                     <a href="crear_incapacidad.php" class="btn btn-primary">Crear nuevo</a>
44                                 </div>
45                                 <table class="table table-hover my-0">
46                                     <thead>
47                                         <tr>
48                                             <th>Empleado</th>
49                                             <th>Tipo</th>
50                                             <th>Fecha Inicio</th>
51                                             <th>Fecha Fin</th>
52                                             <th>Días</th>
53                                             <th>Fecha</th>
54                                             <th>Acción</th>
55                                         </tr>
56                                     </thead>
57                                     <tbody>
58                                         <?php foreach($rows as $row): ?>
59                                             <tr>
60                                                 <td><?php echo $row['usuario'] ?></td>
61                                                 <td><?php echo $row['tipo'] ?></td>
62                                                 <td><?php echo $row['fecha_inicio'] ?></td>
63                                                 <td><?php echo $row['fecha_fin'] ?></td>
64                                                 <td><?php echo $row['dias'] ?></td>
65                                                 <td><?php echo $row['fecha'] ?></td>
66                                                 <td>
67                                                     <form method="POST" class="d-inline-block">

```

Fuente: Elaboración propia.

Este archivo se encarga de mostrar todas las incapacidades registradas en el sistema y permitir su eliminación.

Primero se incluye la conexión a la base de datos y la validación de sesión. Luego se ejecuta una consulta que obtiene los registros de la tabla incapacidades, realizando un *JOIN*

con la tabla usuarios para poder mostrar el nombre del empleado asociado a cada registro. Los resultados se ordenan por fecha descendente, mostrando primero los más recientes.

Todos los registros obtenidos se almacenan en un arreglo asociativo para poder recorrerlos posteriormente y mostrarlos en la tabla.

Además, el archivo incluye la lógica para eliminar un registro. Cuando se presiona el botón Eliminar, se recibe el id mediante *POST* y se ejecuta una sentencia *DELETE* sobre la tabla incapacidades, utilizando una consulta preparada para mayor seguridad.

Una vez eliminado el registro, el sistema redirige nuevamente al listado con un mensaje de confirmación, para evitar que se vuelva a ejecutar la eliminación si la página se recarga.

Este proceso permite mantener actualizada la información de incapacidades y facilita la administración de los registros dentro del sistema.

Figura 57.

Proceso de listado y eliminación de liquidaciones (*liquidaciones.php*).

```

liquidaciones.php
1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6
7 $stmt = $conn->query("SELECT l.*, u.usuario AS usuario FROM liquidaciones l JOIN usuarios u ON l.usuario_id = u.id ORDER BY fecha DESC");
8 $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
9
10 if(isset($_POST['eliminar'])){
11     $id = $_POST['id'];
12
13     $stmt = $conn->prepare("DELETE FROM liquidaciones WHERE id = :id");
14     $stmt->execute(['id' => $id]);
15
16     header("Location: liquidaciones.php?msg=Registro+eliminado");
17 }
18
19 ?>
20 <!DOCTYPE html>
21 <html lang="es">
22 <head>
23     <?php include_once 'inc/head.php' ?>
24 </head>
25 <body>
26     <div class="wrapper">
27         <?php include_once 'inc/sidebar.php' ?>
28
29         <div class="main">
30             <?php include_once 'inc/header.php' ?>
31
32             <main class="content">
33                 <div class="container-fluid p-0">
34                     <h1 class="h3 mb-3">Liquidaciones</h1>
35
36                     <div class="row">
37                         <div class="col-12">
38                             <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
39                             <div class="card">
40                                 <div class="card-body">
41                                     <a href="crear_liquidacion.php" class="btn btn-primary">Crear nuevo</a>
42                                 </div>
43                             <table class="table table-hover my-0">
44                                 <thead>
45                                     <tr>
46                                         <th>Empleado</th>
47                                         <th>Motivo</th>
48                                         <th>Salario pendiente</th>
49                                         <th>Vacaciones</th>
50                                         <th>Aguinaldo</th>
51                                         <th>Previsión</th>
52                                         <th>Cesantía</th>
53                                         <th>Total</th>
54                                         <th>Fecha</th>
55                                         <th>Acción</th>
56                                     </tr>
57                                 </thead>
58                                 <tbody>
59                                     <?php foreach($rows as $row): ?>
60                                         <tr>
61                                             <td><?php echo $row['usuario'] ?></td>
62                                             <td><?php echo $row['motivo'] ?></td>
63                                             <td><?php echo moneda($row['salario_pendiente']) ?></td>
64                                             <td><?php echo moneda($row['vacaciones']) ?></td>
65

```

Fuente: Elaboración propia.

Este archivo se encarga de mostrar todas las liquidaciones registradas y permitir su eliminación cuando sea necesario.

Al inicio se valida la sesión y se establece la conexión con la base de datos. Luego se ejecuta una consulta que obtiene los registros de la tabla liquidaciones, realizando un *JOIN* con la tabla usuarios para mostrar el nombre del empleado asociado a cada liquidación. Los resultados se ordenan por fecha descendente, mostrando primero los registros más recientes.

Los datos obtenidos se almacenan en un arreglo asociativo para recorrerlos y presentarlos en la tabla correspondiente.

Además, el archivo incluye la lógica para eliminar una liquidación. Cuando se presiona el botón Eliminar, el sistema recibe el *id* del registro mediante *POST* y ejecuta una sentencia *DELETE* utilizando una consulta preparada, lo que permite mayor seguridad al evitar inyecciones SQL.

Una vez completada la eliminación, el sistema redirige nuevamente al listado con un mensaje de confirmación. Esto también evita que la acción se repita si el usuario actualiza la página.

Figura 58.

Proceso de autenticación de usuarios (login.php).

```

login.php
1 <?php
2 require_once 'php/connection.php';
3
4 $error = '';
5
6 if(isset($_POST['ingresar'])){
7     $usuario = trim($_POST['usuario']);
8     $contrasena = trim($_POST['contrasena']);
9
10    $stmt = $conn->prepare("SELECT * FROM usuarios WHERE usuario = :usuario");
11    $stmt->execute([
12        'usuario' => $usuario
13    ]);
14    $row = $stmt->fetch(PDO::FETCH_ASSOC);
15
16    if($stmt->rowCount()){
17
18        if(password_verify($contrasena, $row['contrasena'])){
19            $_SESSION['online'] = true;
20            $_SESSION['id'] = $row['id'];
21            $_SESSION['usuario'] = $row['usuario'];
22            $_SESSION['rol'] = $row['rol'];
23            header('location: index.php');
24        }else{
25            $error = 'Usuario o contraseña incorrecta';
26        }
27    }
28    }else{
29        $error = 'Usuario no se encuentra registrado';
30    }
31 }
32 }
33 }
34
35 <!DOCTYPE html>
36 <html lang="es">
37 <head>
38 <php include_once 'inc/head.php' ?>
39 </head>
40 <body>
41 <main class="d-flex w-100">
42 <div class="container d-flex flex-column">
43 <div class="row vh-100">
44 <div class="col-sm-10 col-md-8 col-lg-6 col-xl-5 mx-auto d-table h-100">
45 <div class="d-table-cell align-middle">
46
47 <div class="text-center mt-4">
48 
49 <p class="lead">
50 Inicia sesión con tu cuenta
51 </p>
52 </div>
53
54 <div class="card">
55 <div class="card-body">
56 <div class="text-center">
57 <div class="text-danger mb-2 fw-bold"><php echo $error ?></div>
58 <form method="POST">
59 <div class="mb-3">
60 <label class="form-label">Usuario</label>
61 <input class="form-control form-control-lg" type="text" name="usuario" placeholder="Ingresa tu usuario" required>
62 </div>
63 <div class="mb-3">
64 <label class="form-label">Contraseña</label>
65 <input class="form-control form-control-lg" type="password" name="contrasena" placeholder="Ingresa tu contraseña" required>
66 </div>
67 <div class="d-grid gap-2 mt-3">
68 <button href="#" type="submit" class="btn btn-lg btn-primary" name="ingresar">Ingresar</button>

```

Fuente: Elaboración propia.

Cuando el usuario presiona el botón “Ingresar”, el sistema captura el nombre de usuario y la contraseña ingresados en el formulario. Ambos valores se limpian con *trim()* para evitar espacios innecesarios al inicio o al final.

Luego se realiza una consulta preparada a la tabla usuarios, buscando un registro que coincida con el nombre de usuario ingresado. Si se encuentra un registro, se valida la contraseña utilizando *password_verify*, comparando la contraseña ingresada con la que está almacenada en la base de datos (en formato encriptado).

Si la contraseña es correcta, el sistema crea las variables de sesión necesarias, como el estado de conexión (online), el id del usuario, el nombre de usuario y su rol. Después redirige al usuario al archivo principal del sistema.

Si la contraseña no coincide, se muestra un mensaje indicando que el usuario o la contraseña son incorrectos. Y, si no se encuentra ningún registro con ese nombre de usuario, se informa que el usuario no está registrado.

Figura 59.

Proceso de listado y eliminación de nóminas (nominas.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6
7 $stmt = $conn->query("SELECT n.*, u.usuario AS usuario, p.nombre AS periodo FROM nominas n JOIN usuarios u ON n.usuario_id = u.id JOIN periodos p ON n.periodo_id = p.id ORDER BY fecha DESC");
8 $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
9
10 if(isset($_POST['eliminar'])){
11     $id = $_POST['id'];
12
13     $stmt = $conn->prepare("DELETE FROM nominas WHERE id = :id");
14     $stmt->execute(['id' => $id]);
15
16     header('Location: nominas.php?msg=RegistroEliminado');
17 }
18
19 >
20 <!DOCTYPE html>
21 <html lang="es">
22 <head>
23     <?php include_once 'inc/head.php' ?>
24 </head>
25 <body>
26     <div class="wrapper">
27         <?php include_once 'inc/sidebar.php' ?>
28
29         <div class="main">
30
31             <?php include_once 'inc/header.php' ?>
32
33             <main class="content">
34                 <div class="container-fluid p-0">
35
36                     <h1 class="h3 mb-3">Nóminas</h1>
37
38                     <div class="row">
39                         <div class="col-12">
40                             <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
41                             <div class="card">
42                                 <div class="card-body">
43                                     <a href="crear_nomina.php" class="btn btn-primary">Crear nuevo</a>
44                                 </div>
45                                 <table class="table table-hover my-0">
46                                     <thead>
47                                         <tr>
48                                             <th>Empleado</th>
49                                             <th>Periodo</th>
50                                             <th>Sueldo base</th>
51                                             <th>Salario bruto</th>
52                                             <th>Total deducciones</th>
53                                             <th>Neto a pagar</th>
54                                             <th>Fecha</th>
55                                             <th>Acción</th>
56                                         </tr>
57                                     </thead>
58                                     <tbody>
59                                         <?php foreach($rows as $row) ?>
60                                         <tr>
61                                             <td><?php echo $row['usuario'] ?></td>
62                                             <td><?php echo $row['periodo'] ?></td>
63                                             <td><?php echo moneda($row['sueldo_base']) ?></td>
64                                             <td><?php echo moneda($row['salario_bruto']) ?></td>
65                                             <td><?php echo moneda($row['total_deducciones']) ?></td>
66                                             <td><?php echo moneda($row['neto_pagar']) ?></td>
67

```

Fuente: Elaboración propia.

Al inicio se valida la sesión y se establece la conexión con la base de datos. Luego se ejecuta una consulta que obtiene los registros de la tabla nóminas, realizando un JOIN con la tabla usuarios para mostrar el nombre del empleado y otro *JOIN* con la tabla periodos para mostrar el nombre del periodo correspondiente. Los resultados se ordenan por fecha descendente, mostrando primero las nóminas más recientes.

Los datos obtenidos se almacenan en un arreglo asociativo para poder recorrerlos y presentarlos en la tabla.

Además, se incluye la lógica para eliminar una nómina. Cuando se presiona el botón Eliminar, el sistema recibe el id mediante *POST* y ejecuta una sentencia *DELETE* utilizando una consulta preparada para mayor seguridad.

Después de eliminar el registro, el sistema redirige nuevamente al listado mostrando un mensaje de confirmación. Esto evita que la eliminación se repita si la página se actualiza. Este proceso permite llevar un control ordenado de las nóminas generadas y facilita la administración de los registros dentro del sistema.

Figura 60.

Proceso de gestión y actualización de permisos (permisos.php).

```

permisos.php
1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6
7 $stmt = $conn->query("SELECT p.*, e.usuario AS empleado, s.usuario AS supervisor FROM permisos p JOIN usuarios e ON p.usuario_id = e.id LEFT JOIN usuarios s ON p.admin_id = s.id");
8 $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
9
10 if(isset($_POST['actualizar'])){
11     $id = $_POST['id'];
12     $usuario_id = $_POST['usuario_id'];
13     $estado = $_POST['estado'];
14
15     $stmt = $conn->prepare("UPDATE permisos SET estado = :estado, fecha_revision = :fecha_revision, admin_id = :admin_id WHERE id = :id");
16     $stmt->execute([
17         'id' => $id,
18         'estado' => $estado,
19         'fecha_revision' => date('Y-m-d H:i:s'),
20         'admin_id' => $_SESSION['id']
21     ]);
22
23     if($estado == 'aprobado'){
24         $mensaje = "Tu solicitud de permiso #{$id} fue aprobada";
25     }elseif($estado == 'rechazado'){
26         $mensaje = "Tu solicitud de permiso #{$id} fue rechazada";
27     }
28
29     $stmt = $conn->prepare("INSERT INTO notificaciones (usuario_id, mensaje, fecha) VALUES (:usuario_id, :mensaje, :fecha)");
30     $stmt->execute([
31         'usuario_id' => $usuario_id,
32         'mensaje' => $mensaje,
33         'fecha' => date('Y-m-d H:i:s')
34     ]);
35
36     header('Location: permisos.php?msg=Solicitudactualizada');
37 }
38
39 ?>
40 <!DOCTYPE html>
41 <html lang="es">
42 <head>
43     <?php include_once 'inc/head.php' ?>
44 </head>
45 <body>
46     <div class="wrapper">
47         <?php include_once 'inc/sidebar.php' ?>
48
49         <div class="main">
50
51             <?php include_once 'inc/header.php' ?>
52
53             <main class="content">
54                 <div class="container-fluid p-0">
55                     <h1 class="h3 mb-3">Permisos</h1>
56
57                     <div class="row">
58                         <div class="col-12">
59                             <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
60                             <div class="card">
61                                 <table class="table table-hover my-0">
62                                     <thead>
63                                         <tr>
64                                             <th>Usuario</th>
65                                             <th>Fecha inicio</th>
66                                             <th>Fecha fin</th>
67

```

Fuente: Elaboración propia.

Al inicio se establece la conexión con la base de datos y se valida la sesión activa. Luego se ejecuta una consulta que obtiene los registros de la tabla permisos, realizando un *JOIN* con la tabla usuarios para mostrar el nombre del empleado que realizó la solicitud. Además, se hace un *LEFT JOIN* con la misma tabla de usuarios para identificar al administrador que revisó la solicitud, en caso de que ya haya sido gestionada.

Los resultados se almacenan en un arreglo asociativo para ser mostrados en la tabla correspondiente.

Cuando el administrador decide aprobar o rechazar una solicitud, se envía el formulario con el estado seleccionado. El sistema recibe el id del permiso y el nuevo estado, y ejecuta una sentencia *UPDATE* donde se actualizan tres datos importantes:

- El estado (aprobado o rechazado).
- La fecha de revisión.
- El id del administrador que realizó la acción.

Después de actualizar el permiso, el sistema genera automáticamente una notificación para el usuario. Para ello, se construye un mensaje indicando si la solicitud fue aprobada o rechazada y se guarda en la tabla notificaciones, junto con la fecha correspondiente. De esta forma, el empleado puede visualizar la respuesta dentro del sistema.

Finalmente, se redirige al listado de permisos, para mostrar un mensaje de confirmación.

Figura 61.

Proceso de visualización de solicitudes de horas extra del usuario (*solicitudes_he.php*).

```

1 solicitudes_he.php
2 <?php
3 require_once 'php/connection.php';
4 require_once 'php/verify.php';
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6 $usuario_id = $_SESSION['id'];
7
8 $stat = $conn->prepare("SELECT he.f, e.usuario AS empleado, s.usuario AS supervisor FROM horas_extra he JOIN usuarios e ON he.usuario_id = e.id LEFT JOIN usuarios s ON he.admin_id = s.id WHERE he.usuario_id = :
9 $stat->execute([
10 'usuario_id' => $usuario_id
11 ]);
12 $rows = $stat->fetchAll(PDO::FETCH_ASSOC);
13 >
14 <!DOCTYPE html>
15 <html lang="es">
16 <head>
17 <?php include_once 'inc/head.php' ?>
18 </head>
19 <body>
20 <div class="wrapper">
21 <?php include_once 'inc/sidebar.php' ?>
22
23 <div class="main">
24
25 <?php include_once 'inc/header.php' ?>
26
27 <main class="content">
28 <div class="container-fluid p-0">
29
30 <h1 class="h3 mb-3">Solicitudes de horas extra</h1>
31
32 <div class="row">
33 <div class="col-12">
34 <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
35 <div class="card">
36 <div class="card-body">
37 <a href="crear_solicitud_he.php" class="btn btn-primary">Crear nuevo</a>
38 </div>
39 <table class="table table-hover my-0">
40 <thead>
41 <tr>
42 <th></th>
43 <th>Usuario</th>
44 <th>Fecha</th>
45 <th>Horas</th>
46 <th>Motivo</th>
47 <th>Fecha solicitud</th>
48 <th>Fecha revisión</th>
49 <th>Revisor</th>
50 </tr>
51 </thead>
52 <tbody>
53 <?php foreach($rows as $row) ?>
54 <tr>
55 <td><?php echo $row['id'] ?></td>
56 <td><?php echo $row['empleado'] ?></td>
57 <td><?php echo $row['fecha'] ?></td>
58 <td><?php echo $row['horas'] ?></td>
59 <td><?php echo $row['motivo'] ?></td>
60 <td>
61 <?php if($row['estado'] == 'pendiente') ?>
62 <span class="badge bg-primary"><?php echo $row['estado'] ?></span>
63 <?php elseif($row['estado'] == 'aprobado') ?>
64 <span class="badge bg-success"><?php echo $row['estado'] ?></span>
65 <?php elseif($row['estado'] == 'rechazado') ?>
66 <span class="badge bg-danger"><?php echo $row['estado'] ?></span>
67

```

Fuente: Elaboración propia.

Al inicio se valida la sesión y se obtiene el id del usuario almacenado en la variable de sesión. Con ese dato, se prepara una consulta que busca en la *tabla horas_extra* todos los registros asociados a ese usuario específico.

La consulta incluye un *JOIN* con la tabla usuarios para mostrar el nombre del empleado y un *LEFT JOIN* adicional para identificar al administrador que revisó la solicitud, en caso de que ya haya sido gestionada. El filtro *WHERE he.usuario_id = :usuario_id* garantiza que el usuario solo pueda ver sus propias solicitudes, lo cual refuerza la seguridad del sistema.

Los resultados se almacenan en un arreglo asociativo y luego se recorren para mostrarlos en la tabla.

En la vista se presenta información como:

- Fecha de la solicitud.
- Cantidad de horas.

- Motivo.
- Estado (pendiente, aprobado o rechazado).
- Fecha de revisión.
- Nombre del revisor.

El estado se muestra de forma diferenciada, lo que permite al usuario identificar fácilmente si su solicitud aún está pendiente o si ya fue revisada.

Este proceso brinda transparencia al empleado, ya que puede dar seguimiento a sus solicitudes sin necesidad de consultar directamente con el administrador, mejorando así la comunicación interna dentro del sistema.

Figura 62.

Proceso de visualización de solicitudes de permisos del usuario (solicitudes_permisos.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6 $usuario_id = $_SESSION['id'];
7
8 $stmt = $conn->prepare("SELECT p.*, e.usuario AS empleado, s.usuario AS supervisor FROM permisos p JOIN usuarios e ON p.usuario_id = e.id LEFT JOIN usuarios s ON p.admin_id = s.id WHERE p.usuario_id = :usuario_id ORDER
9 $stmt->execute([
10     'usuario_id' => $usuario_id
11 ]);
12 $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
13
14 <DOCTYPE html>
15 <html lang="es">
16 <head>
17     <?php include_once 'inc/head.php' ?>
18 </head>
19 <body>
20     <div class="wrapper">
21         <?php include_once 'inc/sidebar.php' ?>
22
23     <div class="main">
24
25         <?php include_once 'inc/header.php' ?>
26
27         <main class="content">
28             <div class="container-fluid p-0">
29
30                 <h1 class="h3 mb-3">Solicitudes de permisos</h1>
31
32                 <div class="row">
33                     <div class="col-12">
34                         <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
35                         <div class="card">
36                             <div class="card-body">
37                                 <a href="crear_solicitud_permiso.php" class="btn btn-primary">Crear nuevo</a>
38                             </div>
39                             <table class="table table-hover my-0">
40                                 <thead>
41                                     <tr>
42                                         <th></th>
43                                         <th>Usuario</th>
44                                         <th>Fecha inicio</th>
45                                         <th>Fecha fin</th>
46                                         <th>Motivo</th>
47                                         <th>Estado</th>
48                                         <th>Fecha solicitud</th>
49                                         <th>Fecha revisión</th>
50                                         <th>Revisor</th>
51                                     </tr>
52                                 </thead>
53                                 <tbody>
54                                     <?php foreach($rows as $row) ?>
55                                         <tr>
56                                             <td><?php echo $row['id'] ?></td>
57                                             <td><?php echo $row['empleado'] ?></td>
58                                             <td><?php echo $row['fecha_inicio'] ?></td>
59                                             <td><?php echo $row['fecha_fin'] ?></td>
60                                             <td><?php echo $row['motivo'] ?></td>
61                                             <td>
62                                                 <?php if($row['estado'] == 'pendiente') ?>
63                                                 <span class="badge bg-primary"><?php echo $row['estado'] ?></span>
64                                                 <?php elseif($row['estado'] == 'aprobado') ?>
65                                                 <span class="badge bg-success"><?php echo $row['estado'] ?></span>
66                                                 <?php elseif($row['estado'] == 'rechazado') ?>
67                                                 <span class="badge bg-danger"><?php echo $row['estado'] ?></span>

```

Fuente: Elaboración propia.

Al inicio se valida la sesión activa y se obtiene el id del usuario desde la variable de sesión. Con ese dato se prepara una consulta a la tabla permisos, filtrando los registros mediante la condición *WHERE* `p.usuario_id = :usuario_id`. Esto garantiza que cada empleado solo pueda visualizar la información que le corresponde.

La consulta realiza un *JOIN* con la tabla usuarios para mostrar el nombre del empleado y un *LEFT JOIN* adicional para identificar al administrador que revisó la solicitud, en caso de que ya haya sido aprobada o rechazada.

Los resultados se almacenan en un arreglo asociativo y posteriormente se recorren para mostrarlos en la tabla del sistema.

En la vista se presenta información como:

- Fecha de inicio y fecha final del permiso.
- Motivo.
- Estado actual (pendiente, aprobado o rechazado).
- Fecha en que se realizó la solicitud.
- Fecha de revisión.
- Nombre del revisor.

El estado se muestra de forma diferenciada, lo que permite identificar rápidamente si la solicitud aún está en espera o si ya fue gestionada.

Figura 63.

Proceso de visualización de solicitudes de vacaciones del usuario (solicitudes_vacaciones.php).

```

1 <?php
2 require_once 'php/connection.php';
3 require_once 'php/verify.php';
4
5 $msg = isset($_GET['msg']) ? $_GET['msg'] : '';
6 $usuario_id = $_SESSION['id'];
7
8 $stmt = $conn->prepare("SELECT v.*, e.usuario AS empleado, s.usuario AS supervisor FROM vacaciones v JOIN usuarios e ON v.usuario_id = e.id LEFT JOIN usuarios s ON v.admin_id = s.id WHERE v.usuario_id = :usuario_id ORDER B
9
10 $stmt->execute([
11     'usuario_id' => $usuario_id
12 ]);
13
14 $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
15
16 <!-- DOCTYPE html -->
17 <html lang="es" -->
18 <head -->
19     <?php include_once 'inc/head.php' ?>
20 </head -->
21 <body -->
22     <div class="wrapper" -->
23         <?php include_once 'inc/sidebar.php' ?>
24
25         <div class="main" -->
26
27             <?php include_once 'inc/header.php' ?>
28
29             <main class="content" -->
30                 <div class="container-fluid p-0" -->
31
32                     <h1 class="h3 mb-3">Solicitudes de vacaciones</h1>
33
34                     <div class="row" -->
35                         <div class="col-12" -->
36                             <div class="text-info mb-2 fw-bold"><?php echo $msg ?></div>
37                             <div class="card" -->
38                                 <div class="card-body" -->
39                                     <a href="crear_solicitud_vacaciones.php" class="btn btn-primary">Crear nuevo</a>
40                                 </div>
41                                 <table class="table table-hover my-0" -->
42                                     <thead -->
43                                         <tr>
44                                             <th>#</th>
45                                             <th>Usuario</th>
46                                             <th>Fecha inicio</th>
47                                             <th>Fecha fin</th>
48                                             <th>Estado</th>
49                                             <th>Fecha solicitud</th>
50                                             <th>Fecha revisión</th>
51                                             <th>Revisor</th>
52                                         </tr>
53                                     </thead>
54                                     <tbody -->
55                                         <?php foreach($rows as $row): ?>
56                                             <tr>
57                                                 <td><?php echo $row['id'] ?></td>
58                                                 <td><?php echo $row['empleado'] ?></td>
59                                                 <td><?php echo $row['fecha_inicio'] ?></td>
60                                                 <td><?php echo $row['fecha_fin'] ?></td>
61                                                 <td>
62                                                     <?php if($row['estado'] == 'pendiente'): ?>
63                                                         <span class="badge bg-primary"><?php echo $row['estado'] ?></span>
64                                                     <?php elseif($row['estado'] == 'aprobado'): ?>
65                                                         <span class="badge bg-success"><?php echo $row['estado'] ?></span>
66                                                     <?php elseif($row['estado'] == 'rechazado'): ?>
67                                                         <span class="badge bg-danger"><?php echo $row['estado'] ?></span>
68                                                     <?php endif; ?>
69                                                 </td>
70                                             </tr>
71                                         </tbody>
72                                     </table>
73                                 </div>
74                             </div>
75                         </div>
76                     </div>
77                 </main>
78             </div>
79         </div>
80     </body>
81 </html -->

```

Fuente: Elaboración propia.

Al inicio se valida la sesión activa y se obtiene el id del usuario desde la variable de sesión. Con ese dato se prepara una consulta a la tabla vacaciones, filtrando los registros mediante la condición *WHERE v.usuario_id = :usuario_id*. Esto asegura que el usuario únicamente pueda visualizar sus propias solicitudes.

La consulta realiza un *JOIN* con la tabla usuarios para mostrar el nombre del empleado y un *LEFT JOIN* adicional para identificar al administrador que revisó la solicitud, en caso de que ya haya sido gestionada.

Los resultados se almacenan en un arreglo asociativo y luego se recorren para mostrarlos en la tabla del sistema.

En la vista se presenta información como:

- Fecha de inicio y fecha final solicitada.
- Estado actual de la solicitud (pendiente, aprobado o rechazado).
- Fecha en que se realizó la solicitud.
- Fecha en que fue revisada.
- Nombre del revisor.

El estado se muestra de forma diferenciada para que el usuario pueda identificar fácilmente si su solicitud aún está pendiente o si ya recibió una respuesta.

PRUEBAS

En esta sección se describen las pruebas realizadas al sistema de gestión de planilla desarrollado para Titanium Fitness. Estas pruebas permiten verificar que cada módulo funcione correctamente, validando tanto el comportamiento esperado como los resultados obtenidos en el entorno local del sistema.

Caso de prueba CP-01

Creación de usuario

Objetivo:

Verificar que el sistema permita registrar correctamente un nuevo usuario dentro del módulo de gestión de usuarios.

Módulo:

Gestión de usuarios.

Prioridad:

Alta, ya que el registro de usuarios es fundamental para el funcionamiento del sistema.

Precondiciones:

- El administrador debe haber iniciado sesión correctamente.
- El sistema debe estar conectado a la base de datos local.

Datos de prueba utilizados:

- Nombres: Diego
- Apellidos: Murillo
- Usuario: DiegoM
- Contraseña: Prueba123
- Sueldo Base: ₡ 100,000.00
- Rol: Empleado

Descripción del procedimiento:

Primero se ingresó al módulo “Usuarios” desde el menú lateral del sistema. Posteriormente, se seleccionó la opción “Crear nuevo”, lo que permitió visualizar el formulario de registro.

Se completaron todos los campos obligatorios con datos válidos y se presionó el botón “Guardar”. El sistema procesó la información y realizó el registro en la base de datos.

Resultado esperado:

El sistema debe almacenar correctamente la información del nuevo usuario y redirigir al listado general mostrando un mensaje de confirmación.

Resultado obtenido:

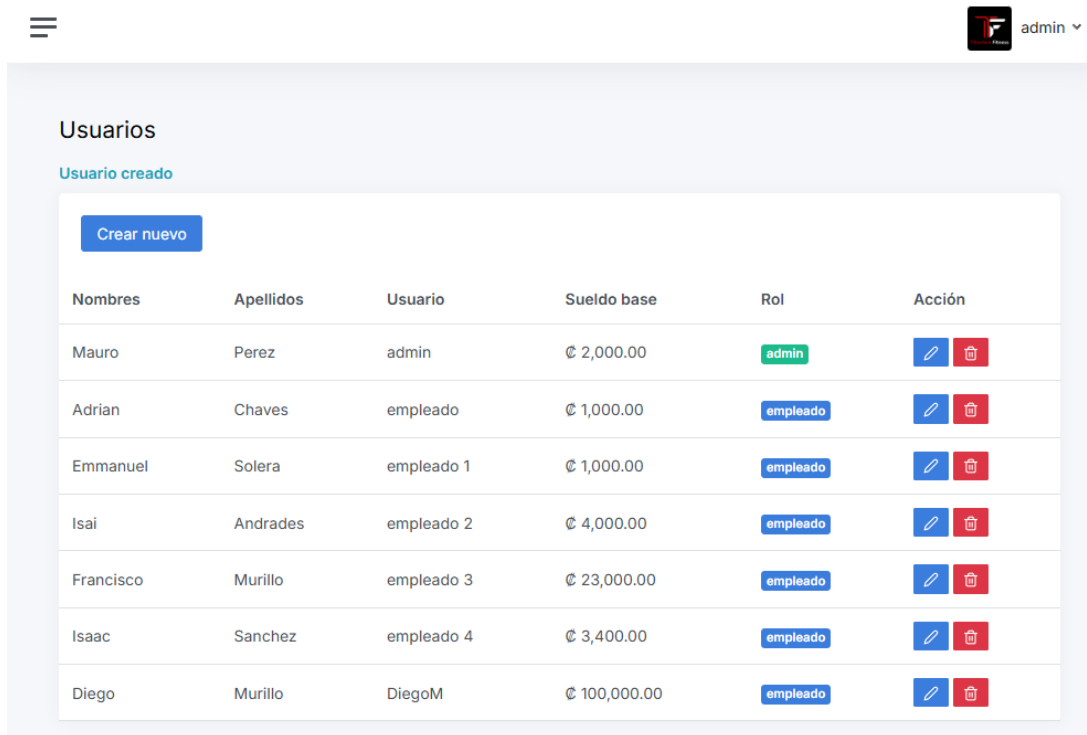
El usuario fue creado exitosamente, se almacenó en la base de datos y se visualizó en el listado de usuarios con el mensaje correspondiente.

Conclusión:

La funcionalidad cumple con lo esperado.

Estado del caso de prueba:

Pasa.

Figura 64.*Prueba módulo de creación de usuario*


The screenshot shows a web application interface for user management. At the top right, there is a user profile icon labeled 'admin'. Below it, the title 'Usuarios' is displayed. A blue button labeled 'Crear nuevo' is positioned above a table. The table has six columns: 'Nombres', 'Apellidos', 'Usuario', 'Sueldo base', 'Rol', and 'Acción'. The table contains eight rows of user data. Each row in the 'Acción' column has two icons: a blue pencil for editing and a red trash can for deleting.

Nombres	Apellidos	Usuario	Sueldo base	Rol	Acción
Mauro	Perez	admin	€ 2,000.00	admin	[Edit] [Delete]
Adrian	Chaves	empleado	€ 1,000.00	empleado	[Edit] [Delete]
Emmanuel	Solera	empleado 1	€ 1,000.00	empleado	[Edit] [Delete]
Isai	Andrades	empleado 2	€ 4,000.00	empleado	[Edit] [Delete]
Francisco	Murillo	empleado 3	€ 23,000.00	empleado	[Edit] [Delete]
Isaac	Sanchez	empleado 4	€ 3,400.00	empleado	[Edit] [Delete]
Diego	Murillo	DiegoM	€ 100,000.00	empleado	[Edit] [Delete]

Fuente: Elaboración propia.

Caso de prueba CP-02**Creación de solicitud de vacaciones****Objetivo:**

Verificar que el sistema permita registrar correctamente una solicitud de vacaciones por parte del usuario y que esta quede almacenada en la base de datos con estado pendiente.

Módulo:

Gestión de vacaciones.

Prioridad:

Alta, debido a que el control de vacaciones forma parte esencial de la administración del personal.

Precondiciones:

- El usuario debe haber iniciado sesión correctamente en el sistema.
- El sistema debe estar funcionando en el entorno local con conexión activa a la base de datos.

Datos de prueba utilizados:

- Fecha de inicio: 2026-02-10
- Fecha final: 2026-02-15

Descripción del procedimiento:

Primero se ingresó al sistema con un usuario tipo empleado. Desde el menú principal se accedió al módulo “Solicitud de vacaciones” y se seleccionó la opción “Crear nuevo”.

Una vez visualizado el formulario, se ingresaron las fechas correspondientes al período de vacaciones y se presionó el botón “Guardar”. El sistema procesó la información ingresada y ejecutó el registro en la base de datos.

Posteriormente, se verificó que la solicitud apareciera en el listado de solicitudes con el estado correspondiente.

Resultado esperado:

El sistema debe registrar la solicitud en la tabla de vacaciones, asignarle el estado “pendiente”, guardar la fecha de solicitud automáticamente y mostrarla en el listado del usuario.

Resultado obtenido:

La solicitud fue almacenada correctamente en la base de datos, se visualizó en el listado de solicitudes y se mostró con estado pendiente.

Conclusión:

La funcionalidad cumple con el comportamiento esperado y permite gestionar adecuadamente las solicitudes de vacaciones.

Estado del caso de prueba:

Pasa.

Figura 65.*Prueba módulo de solicitud de vacaciones*

Solicitudes de vacaciones

Solicitud creada

Crear nuevo

#	Usuario	Fecha inicio	Fecha fin	Estado	Fecha solicitud	Fecha revisión	Revisor
2	DiegoM	2026-02-10	2026-02-15	pendiente	2026-02-20 17:00:43		

Fuente: Elaboración propia.

Caso de prueba CP-03**Creación de solicitud de horas extra****Objetivo:**

Verificar que el sistema permita registrar correctamente una solicitud de horas extra por parte del empleado y que la información quede almacenada en la base de datos con estado pendiente.

Módulo:

Gestión de horas extra.

Prioridad:

Alta, ya que las horas extra influyen directamente en el cálculo de la nómina.

Precondiciones:

- El usuario debe haber iniciado sesión con rol de empleado.
- El sistema debe estar conectado correctamente a la base de datos.

Datos de prueba utilizados:

- Fecha: 2026-02-05
- Cantidad de horas: 3
- Motivo: Personal

Descripción del procedimiento:

Se inició sesión con un usuario tipo empleado y se ingresó al módulo “Horas extra”. Posteriormente, se seleccionó la opción “Crear nuevo”, lo que permitió visualizar el formulario de registro.

Se completaron los campos correspondientes (fecha, cantidad de horas y motivo) y se presionó el botón “Guardar”. El sistema procesó la información y ejecutó el registro en la base de datos.

Después del registro, se verificó que la solicitud apareciera en el listado personal del usuario con el estado correspondiente.

Resultado esperado:

El sistema debe almacenar la solicitud en la tabla de horas extra, asignarle estado “pendiente”, registrar automáticamente la fecha de creación y mostrarla en el listado del usuario.

Resultado obtenido:

La solicitud fue registrada correctamente, se almacenó en la base de datos y se visualizó en el listado con estado pendiente.

Conclusión:

La funcionalidad cumple con lo esperado y permite registrar horas extra de forma controlada.

Estado del caso de prueba:

Pasa.

Figura 66.*Prueba módulo de solicitud de horas extra*

Solicitudes de horas extra

Solicitud creada

Crear nuevo

#	Usuario	Fecha	Horas	Motivo	Estado	Fecha solicitud	Fecha revisión	Revisor
9	DiegoM	2026-02-05	3.00	Personal	pendiente	2026-02-20 17:05:08		

Fuente: Elaboración propia.

Caso de prueba CP-04**Registro de asistencia (entrada y salida)****Objetivo:**

Verificar que el sistema permita registrar correctamente las marcas de asistencia de tipo entrada y salida, asociándolas al usuario autenticado y almacenándolas con la fecha y hora correspondiente.

Módulo:

Gestión de asistencias.

Prioridad:

Alta, ya que el control de asistencia es fundamental para la gestión del personal y puede influir en otros procesos administrativos.

Precondiciones:

- El usuario debe haber iniciado sesión correctamente en el sistema.
- El sistema debe estar conectado a la base de datos local.

Datos de prueba utilizados:

- Tipo: Entrada
- Tipo: Salida

Descripción del procedimiento:

Se inició sesión con un usuario tipo empleado y se ingresó al módulo “Asistencias”.

Primero se seleccionó el tipo **Entrada** y se presionó el botón “Guardar”. El sistema registró automáticamente la fecha y hora actual, asociando la marca al usuario autenticado.

Posteriormente, se repitió el procedimiento seleccionando el tipo **Salida** y presionando nuevamente “Guardar”. El sistema volvió a registrar la fecha y hora correspondiente.

Finalmente, se verificó que ambos registros aparecieran en el listado de asistencias.

Resultado esperado:

El sistema debe almacenar correctamente cada marca en la base de datos, diferenciando el tipo (entrada o salida), asociándola al usuario y registrando automáticamente la fecha y hora.

Resultado obtenido:

Las marcas de entrada y salida se registraron correctamente, se almacenaron en la base de datos y se visualizaron en el listado correspondiente.

Conclusión:

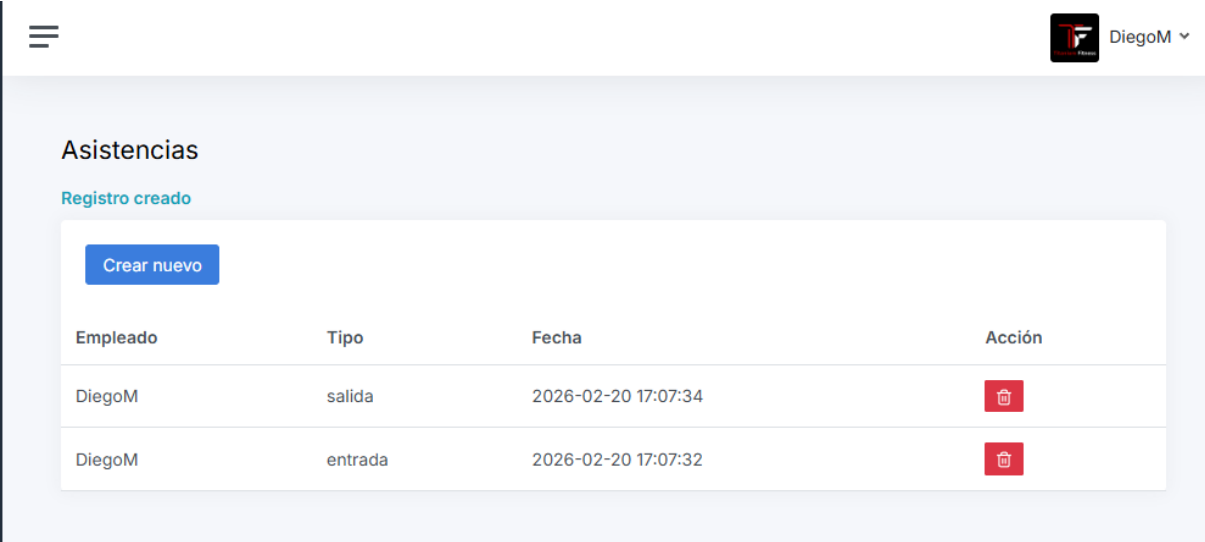
El módulo funciona adecuadamente y permite llevar un control básico de asistencia mediante el registro de entradas y salidas.

Estado del caso de prueba:



Pasa.

Figura 67.

Prueba módulo de Asistencia



The screenshot shows a web interface for 'Asistencias'. At the top right, there is a user profile for 'DiegoM'. Below the title 'Asistencias', there is a message 'Registro creado' and a blue button labeled 'Crear nuevo'. The main content is a table with the following data:

Empleado	Tipo	Fecha	Acción
DiegoM	salida	2026-02-20 17:07:34	
DiegoM	entrada	2026-02-20 17:07:32	

Fuente: Elaboración propia.

Caso de prueba CP-05**Creación de solicitud de permiso****Objetivo:**

Verificar que el sistema permita registrar correctamente una solicitud de permiso por parte del empleado y que esta quede almacenada con estado pendiente para su posterior revisión.

Módulo:

Gestión de permisos.

Prioridad:

Alta, ya que el control de permisos influye en la administración de ausencias del personal.

Precondiciones:

- El usuario debe haber iniciado sesión con rol de empleado.
- El sistema debe estar conectado correctamente a la base de datos.

Datos de prueba utilizados:

- Fecha de inicio: 2026-02-12
- Fecha final: 2026-02-12
- Motivo: Cita médica personal

Descripción del procedimiento:

Se inició sesión con un usuario tipo empleado y se ingresó al módulo “Solicitud de permisos”. Luego se seleccionó la opción “Crear nuevo” para visualizar el formulario.

Se completaron los campos de fecha de inicio, fecha final y motivo, y posteriormente se presionó el botón “Guardar”. El sistema procesó la información ingresada y registró la solicitud en la base de datos.

Finalmente, se verificó que la solicitud apareciera en el listado personal del usuario con el estado correspondiente.

Resultado esperado:

El sistema debe almacenar la solicitud en la tabla de permisos, asignarle el estado “pendiente”, registrar automáticamente la fecha de solicitud y permitir su posterior revisión por parte del administrador.

Resultado obtenido:

La solicitud fue registrada correctamente, se almacenó en la base de datos y se visualizó en el listado con estado pendiente.

Conclusión:

La funcionalidad cumple con lo esperado y permite gestionar adecuadamente las solicitudes de permiso.

Estado del caso de prueba:

Pasa.

Figura 68.

Prueba módulo de solicitud de permisos

#	Usuario	Fecha inicio	Fecha fin	Motivo	Estado	Fecha solicitud	Fecha revisión	Revisor
2	DiegoM	2026-02-12	2026-02-12	Cita médica personal	pendiente	2026-02-20 17:13:46		

Fuente: Elaboración propia.

Caso de prueba CP-06**Cálculo de liquidación****Objetivo:**

Verificar que el sistema calcule correctamente los montos correspondientes a una liquidación laboral según los datos ingresados y el motivo seleccionado.

Módulo:

Gestión de liquidaciones.

Prioridad:

Alta, debido a que la liquidación implica cálculos sensibles relacionados con derechos laborales del empleado.

Precondiciones:

- El administrador debe haber iniciado sesión correctamente.
- El empleado debe estar previamente registrado en el sistema con un salario base definido.
- El sistema debe estar conectado a la base de datos local.

Datos de prueba utilizados:

- Empleado: Murillo, Diego
- Motivo: Despido con responsabilidad
- Días trabajados: 15

- Días de vacaciones: 5
- Meses trabajados: 12

Descripción del procedimiento:

Se inició sesión con un usuario administrador y se ingresó al módulo “Liquidaciones”. Posteriormente, se seleccionó el empleado correspondiente y el motivo de finalización laboral.

Luego se ingresaron los datos solicitados: días trabajados pendientes, días de vacaciones acumulados y meses trabajados para el cálculo del aguinaldo proporcional.

Al presionar el botón “Guardar”, el sistema realizó automáticamente los cálculos correspondientes, considerando el salario base del empleado y aplicando las fórmulas definidas en el sistema para salario pendiente, vacaciones, aguinaldo, preaviso y cesantía, según el motivo seleccionado.

Finalmente, se verificó que el registro quedara almacenado en la base de datos y que los montos se mostraran correctamente en el listado de liquidaciones.

Resultado esperado:

El sistema debe calcular correctamente cada componente de la liquidación (salario pendiente, vacaciones, aguinaldo, preaviso y cesantía cuando corresponda), generar el total final y guardar el registro en la base de datos.

Resultado obtenido:

El sistema realizó los cálculos de manera correcta, generó el total correspondiente y almacenó la liquidación en la base de datos, mostrando la información en el listado general.

Conclusión:

El módulo cumple con el comportamiento esperado y realiza los cálculos de forma automática y consistente.

Estado del caso de prueba:

Pasa.

Figura 69.*Prueba módulo cálculo de liquidación.*

Liquidaciones

Registro creado

Crear nuevo

Empleado	Motivo	Salario pendiente	Vacaciones	Aguinaldo	Preaviso	Cesantia	Total	Fecha	Acción
DiegoM	despido_con_resp	€ 50,000.00	€ 16,666.67	€ 100,000.00	€ 100,000.00	€ 100,000.00	€ 366,666.67	2026-02-20 17:20:31	

Fuente: Elaboración propia.

Caso de prueba CP-07**Cálculo de nómina****Objetivo:**

Verificar que el sistema genere correctamente una nómina mensual para un empleado, realizando los cálculos automáticos de salario bruto, deducciones y salario neto.

Módulo:

Gestión de nóminas.

Prioridad:

Alta, ya que este módulo es el eje principal del sistema de gestión de planilla.

Precondiciones:

- El administrador debe haber iniciado sesión en el sistema.
- El empleado debe estar registrado con un salario base definido.
- El período debe existir previamente en el sistema.
- El sistema debe estar conectado correctamente a la base de datos.

Datos de prueba utilizados:

- Empleado: Murillo Diego (DiegoM)
- Período: Enero 2026
- Salario base: €100,000.00

Descripción del procedimiento:

Se ingresó al módulo “Nóminas” y se seleccionó la opción “Crear nuevo”. Posteriormente, se eligió al empleado “Murillo Diego (DiegoM)” y el período correspondiente a “Enero 2026”. Al presionar el botón “Guardar”, el sistema consultó automáticamente el salario base del empleado y verificó si existían horas extra aprobadas dentro del período seleccionado.

Seguidamente, el sistema realizó los cálculos correspondientes:

- Determinó el salario bruto.
- Calculó las deducciones
- Generó el total de deducciones.
- Calculó el salario neto a pagar.

Finalmente, el registro fue almacenado en la base de datos y se mostró en el listado general de nóminas con el detalle completo de los montos.

Resultado esperado:

El sistema debe generar automáticamente la nómina con los cálculos correctos y almacenar el registro mostrando el salario bruto, total de deducciones y salario neto.

Resultado obtenido:

La nómina fue generada correctamente, los cálculos coincidieron con los valores esperados y el registro se visualizó en el listado con la información completa.

Conclusión:

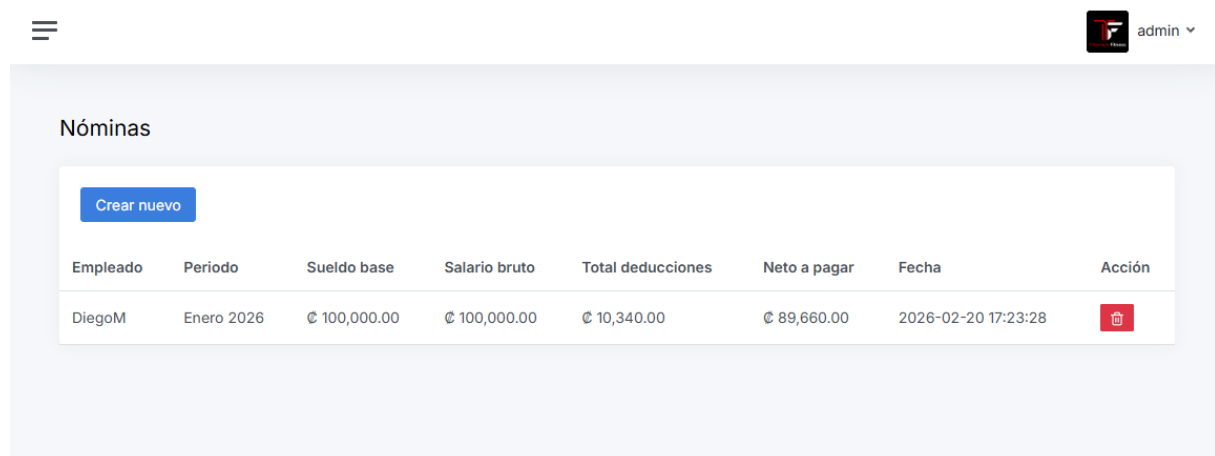
El módulo cumple con su función principal, realizando los cálculos de manera automática y registrando la información correctamente.

Estado del caso de prueba:


Pasa.

Figura 70.

Prueba módulo cálculo de nómina.



The screenshot shows a web interface for the 'Nóminas' (Payroll) module. At the top left is a hamburger menu icon, and at the top right is a user profile icon labeled 'admin'. Below the header, there is a 'Nóminas' title and a 'Crear nuevo' button. A table displays the following data:

Empleado	Periodo	Sueldo base	Salario bruto	Total deducciones	Neto a pagar	Fecha	Acción
DiegoM	Enero 2026	€ 100,000.00	€ 100,000.00	€ 10,340.00	€ 89,660.00	2026-02-20 17:23:28	

Fuente: Elaboración propia.

Caso de prueba CP-08**Cálculo de aguinaldo****Objetivo:**

Comprobar que el sistema calcule y registre correctamente el aguinaldo de un empleado a partir del período seleccionado, tomando como base las nóminas generadas dentro de ese rango.

Módulo:

Gestión de aguinaldos.

Prioridad:

Alta, porque el aguinaldo es un cálculo obligatorio y debe quedar registrado correctamente.

Precondiciones:

- El usuario administrador debe haber iniciado sesión.
- El empleado debe existir en el sistema.
- Deben existir nóminas registradas para ese empleado en el período que se va a calcular.
- El sistema debe estar operando en el entorno local con acceso a la base de datos.

Datos de prueba utilizados:

- Empleado: Murillo Diego (DiegoM)
- Período: 2026

Descripción del procedimiento:

Se ingresó al módulo “Aguinaldos” y se seleccionó al empleado “Murillo Diego (DiegoM)”. Luego se indicó el período **2026** en el formulario de creación.

Al presionar el botón “**Guardar**”, el sistema consultó las nóminas registradas del empleado dentro del rango correspondiente al cálculo, sumó los salarios brutos encontrados, contabilizó los meses con registros y, con base en esa información, calculó automáticamente el monto del aguinaldo.

Finalmente, el sistema guardó el resultado en la base de datos y el registro quedó disponible en el listado general de aguinaldos.

Resultado esperado:

El sistema debe calcular el aguinaldo de forma automática según el historial de nóminas del período seleccionado, registrar el total de salarios, los meses laborados y el monto final del aguinaldo, y almacenar el registro correctamente.

Resultado obtenido:

El sistema realizó el cálculo automáticamente y guardó el aguinaldo con la información correspondiente, mostrando el registro en pantalla.

Conclusión:

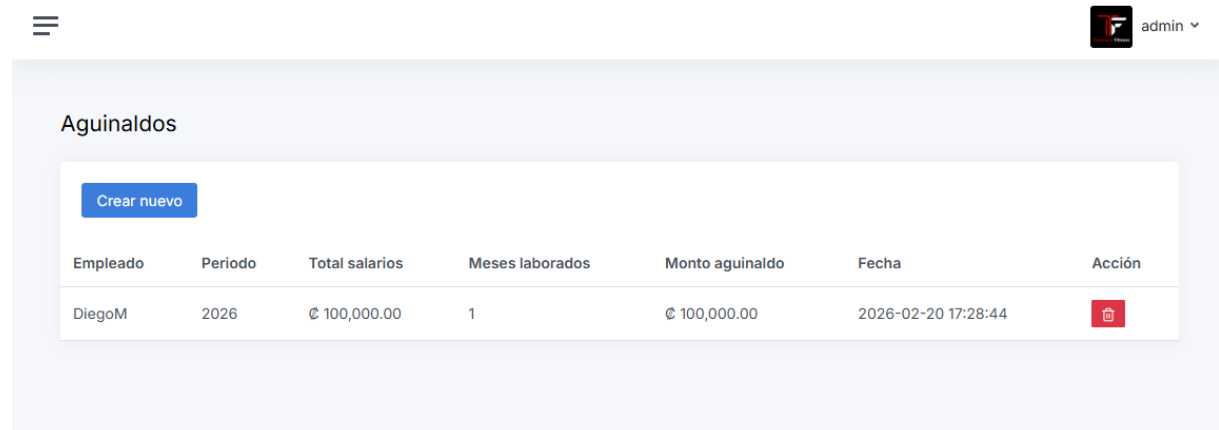
El módulo calcula y registra el aguinaldo de manera correcta a partir de los datos disponibles en nóminas.

Estado del caso de prueba:


Pasa.

Figura 71.

Prueba módulo cálculo de aguinaldo.



admin

Empleado	Periodo	Total salarios	Meses laborados	Monto aguinaldo	Fecha	Acción
DiegoM	2026	€ 100,000.00	1	€ 100,000.00	2026-02-20 17:28:44	

Fuente: Elaboración propia.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

Conclusión

Con base en el análisis detallado del proceso de cálculo y gestión de planilla en el gimnasio Titanium Fitness, se concluye que la implementación de un sistema automatizado es fundamental para atender las deficiencias identificadas en los procedimientos manuales. Entre los principales inconvenientes detectados se encuentran la falta de control centralizado de la información, posibles errores en los cálculos salariales, atrasos en la gestión de datos y dificultad para supervisar correctamente los registros relacionados con horas extra, vacaciones, incapacidades y liquidaciones.

Los objetivos específicos establecidos para el desarrollo del prototipo funcional fueron alcanzados de manera satisfactoria, porque contribuyen a optimizar la administración interna del gimnasio Titanium Fitness:

- **Analizar y evaluar:** Se identificaron y estudiaron los requerimientos reales del gimnasio en cuanto al manejo de la planilla, con el fin de comprender las necesidades administrativas y operativas que debía cubrir el sistema.
- **Diseñar y adaptar:** Se diseñó una aplicación web orientada a la gestión de planilla, considerando los procesos específicos del gimnasio, lo que asegura que la solución se ajusta a su dinámica de trabajo.
- **Programar e implementar:** Se desarrolló e implementó el prototipo funcional utilizando tecnologías web, para automatizar los cálculos salariales y organizar la información del personal de manera estructurada.
- **Probar y validar:** Se realizaron pruebas para verificar el correcto funcionamiento del sistema, para asegurar la precisión en los cálculos, el almacenamiento adecuado de los datos y la estabilidad general de la aplicación.

Estos resultados se traducen en un sistema que no solo resuelve los problemas detectados inicialmente, sino que también fortalece el control administrativo, mejora la eficiencia en la gestión de la planilla y aporta mayor seguridad en el manejo de la información financiera del gimnasio.

De esta manera, el proyecto evidencia cómo una solución tecnológica desarrollada a la medida puede generar mejoras significativas en pequeñas empresas, para contribuir a la modernización de sus procesos y al crecimiento organizacional.

Recomendaciones

A partir del análisis realizado durante el desarrollo del sistema web de gestión de planilla para el gimnasio Titanium Fitness, se plantean algunas recomendaciones que pueden contribuir a mejorar su utilización dentro de la organización y a mantener un adecuado control de la información administrativa relacionada con el personal.

En primer lugar, se recomienda que la administración del gimnasio utilice el sistema desarrollado como la herramienta principal para la gestión de la planilla del personal. La utilización de un único sistema permitirá mantener la información organizada y evitar que se sigan utilizando registros manuales paralelos que podrían generar inconsistencias en los datos. Esta práctica debería comenzar a aplicarse desde el momento en que el sistema sea implementado oficialmente en el gimnasio y mantenerse como parte del proceso administrativo habitual.

De igual forma, se recomienda que el administrador del gimnasio establezca un procedimiento sencillo de respaldo de la base de datos del sistema. Debido a que la aplicación funciona en un entorno local, es importante resguardar periódicamente la información para evitar pérdidas de datos en caso de fallos técnicos en el equipo donde se encuentra instalado el sistema. Como medida preventiva, se sugiere realizar una copia de seguridad de la base de datos al menos una vez por semana en un dispositivo externo, iniciando esta práctica durante el primer mes posterior a la implementación del sistema.

Asimismo, se considera conveniente que el administrador del gimnasio realice una revisión mensual de la nómina generada por el sistema antes de efectuar los pagos al personal. Aunque el sistema automatiza gran parte de los cálculos, esta revisión permite confirmar que la información registrada sea correcta y que los resultados obtenidos correspondan a los datos ingresados. Este procedimiento puede iniciarse a partir del primer mes de uso del sistema y mantenerse como una práctica regular dentro de la gestión administrativa.

Se recomienda que la administración del gimnasio realice una revisión general del funcionamiento del sistema aproximadamente una vez al año. Esta revisión permitiría identificar posibles ajustes o mejoras que puedan surgir con el uso del sistema y con los cambios en las necesidades administrativas del gimnasio. De esta forma se podría asegurar que el sistema continúe siendo una herramienta útil para el manejo de la planilla.

Finalmente, se recomienda que el administrador del sistema revise periódicamente las medidas de seguridad relacionadas con el acceso a la aplicación y con el equipo donde se

encuentra instalado el sistema. Esto incluye verificar las contraseñas de acceso y mantener protegido el equipo utilizado para su operación. Esta revisión podría realizarse una vez al año como parte de las prácticas básicas de control de la información.

REFERENCIAS

- ADP. (2024). *¿Qué es la nómina y cómo se gestiona correctamente?*
<https://www.adp.com/resources/articles-and-insights/articles/w/what-is-payroll.aspx>
- Apache Friends. (s.f.). *XAMPP*. <https://www.apachefriends.org>
- Asamblea Legislativa de Costa Rica. (s.f.). *Ley de Aguinaldo a los Trabajadores del Sector Privado (Ley N.º 2412)*.
https://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTC&nValor1=1&nValor2=4376&nValor3=4939&strTipM=TC
- Asamblea Legislativa de Costa Rica. (2024). *Ley para crear la licencia remunerada por muerte de familiares*.
https://pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?nValor1=1&nValor2=103384&nValor3=143663¶m1=NRTC&strTipM=TC
- Caja Costarricense de Seguro Social. (2023). *Proceso de registro y presentación de planillas*.
<https://www.ccss.sa.cr/planillas>
- Darwinbox. (2024, 12 de febrero). *What is Time and Attendance Management in HR?*
<https://blog.darwinbox.com/time-and-attendance-management-in-hr>
- Dirección de Gestión del Talento Humano. (s.f.). *Permisos con goce de salario autorizados por jefaturas inmediatas*. <https://dgth.mep.go.cr/permisos-con-goce-de-salario-autorizados-por-jefes-inmediatos/>
- Euroinnova Business School. (2022, 4 de mayo). *Herramientas de programación*.
<https://www.euroinnova.edu.es/blog/herramientas-de-programacion>
- FODESAF. (s.f.). *Componentes de una planilla de la CCSS*.
https://fodesaf.go.cr/gestion_de_cobros/campus_virtual/Componentes_de_una_planilla_de_la_CCSS.html
- Kanban University. (2023). *What is Kanban? Core principles & practices*.
<https://kanban.university/kanban-guide/>
- Microsoft Azure. (s.f.). *Visual Studio Code*. <https://azure.microsoft.com/es-es/products/visual-studio-code>
- Ministerio de Trabajo y Seguridad Social. (2023). *Criterio jurídico sobre permisos no regulados expresamente (DAJ-AER-OFP-1315-2023)*.
https://www.mtss.go.cr/elministerio/estructura/direccion-asuntos-juridicos/criterios_juridicos/documentos/DAJ-AER-OFP-1315-2023.pdf

- Ministerio de Trabajo y Seguridad Social de Costa Rica. (s.f.-b). *Asuntos laborales: Aguinaldo*. <https://www.mtss.go.cr/temas-laborales/aguinaldo/aguinaldo.html>
- Oracle. (2024). *What is an HRIS? Human Resource Information System explained*. <https://www.oracle.com/human-capital-management/what-is-hris/>
- Oracle. (s.f.). *¿Qué es MySQL?* <https://www.mysql.com/what-is-mysql/>
- phpMyAdmin. (2023). *phpMyAdmin: A web interface for MySQL and MariaDB*. <https://www.phpmyadmin.net/>
- phpMyAdmin Documentation. (2023). *phpMyAdmin official documentation*. <https://docs.phpmyadmin.net/>
- Salesforce. (s.f.). *¿Qué es un usuario?* <https://www.salesforce.com/es/glossary/user/>
- Sistemas Perú Web. (2024, 6 de abril). *¿Qué es un sistema web?* <https://sistemasperuweb.com/web/que-es-un-sistema-web/>
- Techopedia. (2020, 14 de julio). *Module*. <https://www.techopedia.com/definition/3807/module>
- Time. (s.f.). *Liquidación laboral: Qué es y cómo se calcula*. <https://www.tiemda.com/blog/liquidacion-laboral/>
- Virket Agency. (2021, 13 de diciembre). *¿Qué es un prototipo para diseño web?* <https://virket.agency/blog/disenio-ux-ui/que-es-un-prototipo-para-diseno-web/>
- Canive, M. (2019, 13 de diciembre). *Investigación cualitativa: qué es, características y métodos*. Aiteco. <https://aiteco.com/investigacion-cualitativa/>
- Ortega, C. (2023a, noviembre). *Investigación cuantitativa: ¿Qué es, tipos y características?*. QuestionPro. <https://www.questionpro.com/es/investigacion-cuantitativa.html>
- Ortega, C. (2023b, noviembre). *Investigación mixta: qué es y cómo realizarla*. QuestionPro. <https://www.questionpro.com/es/investigacion-mixta.html>
- Ortega, C. (2023). *La observación como técnica de investigación*. QuestionPro. <https://www.questionpro.com/es/observacion.html>
- QuestionPro. (s.f.). *Investigación cualitativa: Definición, tipos, métodos y ejemplos*. <https://www.questionpro.com/es/investigacion-cualitativa.html>
- Universidad de la Sabana. (s.f.). *Introducción a la Investigación: guía interactiva*. <https://www.unisabana.edu.co/portaldenoticias/al-dia/detalles/noticia/como-hacer-una-investigacion-conoce-sus-tipos-1/>
- Zita, A. (2021, 22 de octubre). *Métodos de investigación: qué son, tipos y ejemplos*. Toda Materia. <https://www.todamateria.com/metodos-de-investigacion/>

- Hernández-Sampieri, R., & Mendoza, C. P. (2023). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta* (2.^a ed.). McGraw-Hill Interamericana. <https://www.mheducation.com.mx/metodologia-de-la-investigacion-las-rutas-cuantitativa-cualitativa-y-mixta-9786071516988-latam>
- Kendall, K. E., & Kendall, J. E. (2020). *Análisis y diseño de sistemas* (10.^a ed.). Pearson Educación. <https://www.pearson.com/es-latam>
- Pressman, R. S. (2020). *Ingeniería del software: Un enfoque práctico* (8.^a ed.). McGraw-Hill Education. <https://www.mheducation.com/highered/product/ingenieria-del-software-enfoque-practico-pressman>
- Sommerville, I. (2019). *Ingeniería de software* (10.^a ed.). Pearson Educación. <https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000006331>

ANEXOS

Anexo A

GUÍA DE ENTREVISTA

Objetivo de la entrevista:

Recopilar información cualitativa del personal administrativo de Titanium Fitness sobre los procesos actuales de gestión de planilla, con el fin de identificar necesidades, dificultades y oportunidades de mejora para el desarrollo del sistema web propuesto.

Contexto:

Esta entrevista forma parte del Trabajo Final de Graduación de la carrera de Ingeniería Informática. Se aplicará al personal encargado de tareas administrativas como registro de asistencia, control de horas extra, permisos y cálculo de salarios.

Indicaciones:

- La participación es voluntaria.
- La información brindada será confidencial y utilizada únicamente con fines académicos.
- No se solicitarán datos personales sensibles.
- El entrevistado puede responder de manera libre y con ejemplos si lo considera necesario.

PREGUNTAS DE LA ENTREVISTA

1. ¿Existe actualmente un proceso definido para la gestión de planilla en el gimnasio?
R/ Sí / No
2. ¿El registro de asistencia del personal se realiza de forma organizada y consistente?
R/ Sí / No
3. ¿El proceso de registro y validación de horas extra se encuentra claramente establecido?
R/ Sí / No
4. ¿Considera que el cálculo de salarios y deducciones se realiza de manera eficiente?
R/ Sí / No
5. ¿Ha experimentado errores o inconvenientes en el proceso actual de planilla?

- R/ Sí / No
6. ¿El proceso actual de gestión de planilla consume más tiempo del necesario?
R/ Sí / No
7. ¿Considera necesaria la implementación de un sistema web para la gestión de planilla?
R/ Sí / No
8. ¿Cree que un sistema automatizado ayudaría a reducir errores administrativos?
R/ Sí / No
9. ¿Estaría dispuesto(a) a utilizar un nuevo sistema web para la gestión de planilla?
R/ Sí / No
10. ¿Considera necesaria una capacitación para el uso del nuevo sistema?
R/ Sí / No
11. En caso de considerar necesario un nuevo sistema, ¿qué aspectos le gustaría que este mejore?
R/ (opcional)

Anexo B

GUÍA DE OBSERVACIÓN

Objetivo:

Observar y evaluar los procesos administrativos relacionados con la gestión de planilla en el gimnasio Titanium Fitness, con el fin de identificar prácticas actuales, detectar oportunidades de mejora y recopilar información relevante para el desarrollo del sistema web propuesto.

Contexto:

Esta guía de observación se aplicará en las instalaciones del gimnasio Titanium Fitness, durante la revisión de los procesos administrativos que involucran la gestión de asistencia, control de horas extra, permisos y elaboración de planilla. La información recolectada permitirá identificar necesidades operativas y validar los requerimientos para el desarrollo del sistema web propuesto en el Trabajo Final de Graduación.

No	Aspectos por observar	Cumple	No Cumple	Oportunidad de mejora	Detalle de Observación
1	Registro de asistencia del personal				
2	Proceso de registro y validación de horas extra				
3	Cálculo de salarios y deducciones				
4	Gestión de permisos y ausencias				
5	Flujo general del proceso administrativo				