

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE: INGENIERÍA EN SISTEMAS DE INFORMACIÓN



TRABAJO DE TITULACIÓN

TEMA: SISTEMA WEB PARA EL REGISTRO Y CONTROL DE ASISTENCIA.

AUTOR:

MARCEL TOMÁS BOHÓRQUEZ TINIZARAY

QUITO DM, JUNIO DE 2024

# **DEDICATORIA**

Quisiera dedicar este trabajo de titulación a mis pilares de vida, mis padres Gissella Tinizaray y Marcelo Bohórquez y mi hermana Karen Bohórquez que en mi vida me brindaron la oportunidad de crecer y la autonomía de hacerlo a mi manera, en mi vida gracias a su amor, su apoyo, sus lecciones y su compañía incondicional me han llevado a lograr muchas metas y a trazar mis propósitos de vida, les dedico mi trabajo de titulación y todos mis éxitos en el futuro.

Un agradecimiento eterno a mi familia por tenerlos en mi vida, por todo lo que aportan a mi crecimiento y ser mejor persona y profesional cada día de mi vida.

# **AGRADECIMIENTO**

Mi más sincero agradecimiento a la Pontificia Universidad Católica del Ecuador, que fue mi hogar durante estos cuatro años, hogar donde aprendí y crecí tanto como pude, tanto personalmente como profesionalmente. Un agradecimiento especial a mis compañeros de carrera, Ariana Núñez, Sebastián Miranda, Richard Molina, Jhoel Muela y Daniel Bonilla, mucho más que colegas dentro de mi formación profesional tuve el privilegio de encontrar una familia y una amistad sincera, apoyo incondicional que me permitió mantener mi confianza y seguridad de que estaba en el lugar correcto con las personas correctas. Un agradecimiento especial a Annie Bermeo, un pilar fundamental para conocer que además de los conocimientos técnicos que se deben aprender, lo que lo mueve a uno como persona es el amor que uno tiene por las cosas y por las personas, amor a la vida y cada momento que me ha llevado a estar aquí.

Finalmente agradezco nuevamente a mis padres y mi hermana que día tras día me impulsan y gracias a ellos, a su apoyo incondicional es que he podido construir mis sueños y aspiraciones. Con todo mi corazón gracias totales.

## **RESUMEN**

El proyecto de titulación "Sistema Web para el Registro y Control de Asistencia" presenta el desarrollo de una aplicación web destinada a gestionar y automatizar el control de asistencia de empleados administrativos. El trabajo se justifica en la necesidad de mejorar la precisión y eficiencia del registro de asistencia, reemplazando métodos manuales propensos a errores que afectan la productividad. La metodología utilizada es el prototipado, permitiendo iteraciones sucesivas de diseño y retroalimentación para afinar funcionalidades y usabilidad. El sistema integra la gestión de usuarios, roles, horarios, cargos, empleados, asistencias y justificaciones, garantizando un registro preciso de entradas y salidas, y facilitando la generación de reportes detallados para la toma de decisiones estratégicas en la gestión de personal.

# ÍNDICE GENERAL

CAPÍTULO I .....	1
INTRODUCCIÓN .....	1
1.    MARCO DE REFERENCIA .....	1
1.1    JUSTIFICACIÓN .....	1
1.2    PLANTEAMIENTO DEL PROBLEMA .....	1
1.3    OBJETIVO GENERAL .....	2
1.4    OBJETIVOS ESPECÍFICOS .....	2
1.5    ALCANCE .....	3
CAPÍTULO II .....	4
FUNDAMENTACIÓN TEÓRICA.....	4
2.    MARCO TEÓRICO Y CONCEPTUAL .....	4
2.1    ANTECEDENTES .....	4
2.2    MARCO CONCEPTUAL .....	5
2.2.1    CONTROL DE REGISTRO DE ASISTENCIA.....	5
2.2.2    APLICACIONES WEB .....	6
2.3    MARCO TEORICO .....	7
2.3.1    VISUAL STUDIO.....	7
2.3.3    POSTGRESQL.....	8
2.3.4    SQL SERVER .....	8
2.3.5    MYSQL .....	9
2.3.6    TABLA COMPARATIVA DE BASES DE DATOS .....	9
2.3.7    PYTHON .....	10
2.3.8    JAVA.....	11
2.3.9    PHP.....	11
2.3.10    TABLA COMPARATIVA DE LENGUAJES DE PROGRAMACIÓN.....	12
2.3.11    HTML .....	13
2.3.12    JAVASCRIPT .....	14
2.3.13    CSS.....	15
2.3.14    ADMINISTRACIÓN DE RECURSOS HUMANOS.....	15
2.3.15    AUTOMATIZACIÓN DE PROCESOS .....	16
2.3.16    INGENIERÍA DE SOFTWARE.....	16
2.3.17    DESARROLLO DE SOFTWARE.....	17

2.3.18	ANÁLISIS DE SISTEMAS .....	17
2.3.19	DISEÑO DE SISTEMAS.....	17
2.3.20	DISEÑO DE INTERFACES DE USUARIO.....	18
2.3.21	SEGURIDAD DE LA INFORMACIÓN.....	18
2.3.22	ARQUITECTURA DE SOFTWARE.....	19
2.3.23	ARQUITECTURA CLIENTE-SERVIDOR.....	19
2.3.24	ARQUITECTURA ORIENTADA A EVENTOS .....	20
2.3.25	ARQUITECTURA MVC (MODELO-VISTA-CONTROLADOR) .....	20
2.3.25.1	MODELO.....	21
2.3.25.2	VISTA .....	21
2.3.25.3	CONTROLADOR.....	21
2.3.26	TABLA COMPARATIVA DE ARQUITECTURAS.....	22
2.3.27	METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	22
2.3.28	METODOLOGÍA SCRUM .....	23
2.3.29	METODOLOGÍA KANBAN .....	23
2.3.30	METODOLOGÍA PROTOTIPADO.....	24
2.3.31	TABLA DE COMPARACIÓN METODOLOGÍAS DE DESARROLLO .....	24
CAPÍTULO III.....		26
METODOLOGÍA .....		26
3.	METODOLOGÍA DE DESARROLLO DEL PLAN DE TESIS .....	26
3.1	PROTOTIPADO .....	26
3.1.1	TEORÍA DE LA METODOLOGÍA DE PROTOTIPADO .....	27
3.1.2	ROLES Y RESPONSABILIDADES .....	27
3.1.3	ARTEFACTOS DE LA METODOLOGÍA PROTOTIPADO.....	28
3.1.4	EVENTOS DEL PROTOTIPADO .....	29
3.1.5	CICLO ITERATIVO DEL PROTOTIPADO .....	29
3.2	REQUERIMIENTOS .....	30
3.2.1	REQUERIMIENTOS FUNCIONALES .....	30
3.2.2	REQUERIMIENTOS NO FUNCIONALES.....	32
3.2.3	CASOS DE USO.....	33
3.2.3.1	COMPONENTES DE UN CASO DE USO .....	33
3.2.3.2	DIAGRAMA DE CASO DE USO NIVEL GENERAL .....	34
3.2.3.3	RF01 ACCESO AL SISTEMA WEB.....	35
3.2.3.4	RF02 ADMINISTRACIÓN DE USUARIOS .....	36
3.2.3.5	RF02.1 ADMINISTRACIÓN DE ROLES DE USUARIO .....	37

3.2.3.6	RF03 ADMINISTRACIÓN DE CARGOS DEL EMPLEADO .....	38
3.2.3.8	RF05 ADMINISTRACIÓN DE EMPLEADOS .....	40
3.2.3.9	RF05.1 ADMINISTRACIÓN DE ROLES DE EMPLEADOS .....	41
3.2.3.10	RF06 ADMINISTRACIÓN DE ASISTENCIAS .....	42
3.2.3.11	RF07 ADMINISTRACIÓN DE JUSTIFICACIONES .....	43
3.2.3.12	RF08 REGISTRO DE ASISTENCIA ENTRADA – SALIDA .....	45
3.2.4	PLANIFICACIÓN DE LAS ITERACIONES DE PROTOTIPADO.....	45
CAPÍTULO IV .....		48
DESARROLLO .....		48
4.	DESARROLLO DEL SISTEMA WEB .....	48
4.1	ITERACIÓN 1 - DESARROLLO DE LA GESTIÓN DE USUARIOS E INICIO DE SESIÓN .....	48
4.1.1	TABLA DE ACTIVIDADES ITERACIÓN 1 .....	49
4.1.2	CAPTURAS DE PANTALLA ITERACIÓN 1 .....	50
4.1.2.1	INICIO DE SESIÓN .....	50
4.1.2.2	GESTIÓN DE USUARIOS .....	52
4.1.2.3	GESTIÓN DE ROLES DE USUARIOS .....	57
4.1.3	PRUEBAS Y RESULTADOS ITERACIÓN 1 .....	60
4.2	ITERACIÓN 2 - DESARROLLO DE LA GESTIÓN DE CARGOS Y HORARIOS.....	62
4.2.1	TABLA DE ACTIVIDADES ITERACIÓN 2 .....	63
4.2.2.1	GESTIÓN DE CARGOS DE EMPLEADOS .....	64
4.2.2.2	GESTIÓN DE HORARIOS DE EMPLEADOS.....	67
4.2.3	PRUEBAS Y RESULTADOS ITERACIÓN 2 .....	71
4.3	ITERACIÓN 3 - DESARROLLO DE LA GESTIÓN DE EMPLEADOS Y SUS ROLES .....	74
4.3.1	TABLA DE ACTIVIDADES ITERACIÓN 3 .....	75
4.3.2.1	GESTIÓN DE EMPLEADOS .....	75
4.3.2.2	GESTIÓN DE ROLES DE EMPLEADOS .....	79
4.3.3	PRUEBAS Y RESULTADOS ITERACIÓN 3 .....	83
4.4	ITERACIÓN 4 - DESARROLLO DE LA GESTIÓN DE ASISTENCIAS Y JUSTIFICACIONES.....	85
4.4.1	TABLA DE ACTIVIDADES ITERACIÓN 4 .....	86
4.4.2.1	GESTIÓN DE ASISTENCIAS .....	87
4.4.2.2	GESTIÓN DE JUSTIFICACIONES .....	91
4.4.3	PRUEBAS Y RESULTADOS ITERACIÓN 4 .....	94
CAPÍTULO V .....		97
CONCLUSIONES Y RECOMENDACIONES.....		97

5. CONCLUSIONES ..... 97

6. RECOMENDACIONES..... 98

# ÍNDICE DE FIGURAS

Figura 1 .....	34
Caso de uso nivel general. ....	34
Figura 2 .....	35
Acceso al sistema web. ....	35
Figura 4 .....	36
Administración de usuarios.....	36
Figura 3 .....	37
Administración de roles de usuarios .....	37
Figura 5 .....	38
Administración de cargos .....	38
Figura 6 .....	39
Administración de horarios .....	39
Figura 8 .....	40
Administración de empleados .....	40
Figura 7 .....	41
Administración de roles de usuarios .....	41
Figura 9 .....	42
Administración de asistencias .....	42
Figura 10 .....	43
Administración de justificaciones .....	43
Figura 11 .....	45
Registro de asistencia Entrada - Salida .....	45
Figura 12 .....	50
Pantalla Inicio Sesión.....	50
Figura 13 .....	50
Pantalla Inicio Sesión ingresando usuario y contraseña .....	50
Figura 14 .....	51
Pantalla Inicio Sesión mostrando contraseña.....	51
Figura 15 .....	51
Pantalla Inicio Sesión, caso usuario o contraseña incorrectas .....	51
Figura 16 .....	52
Pantalla Inicio Sesión, caso campos vacíos .....	52
Figura 17 .....	52

Pantalla Lista de Usuarios.....	52
Figura 18.....	53
Pantalla Botones de acción .....	53
Figura 19.....	53
Pantalla Modificación de usuario.....	53
Figura 19.....	54
Notificación modificación de usuario correcta .....	54
Figura 20.....	54
Pantalla registro de usuario.....	54
Figura 21.....	55
Pantalla registro de usuario ingresando su información .....	55
Figura 22.....	55
Notificación registro de usuario correcto.....	55
Figura 23.....	56
Pantalla eliminación de usuario .....	56
Figura 24.....	56
Notificación eliminación de usuario correcta .....	56
Figura 25.....	57
Pantalla listado de roles de usuario.....	57
Figura 26.....	57
Pantalla modificación de roles de usuario.....	57
Figura 27.....	58
Notificación modificación de roles de usuario correcta.....	58
Figura 28.....	58
Pantalla registro de roles de usuario .....	58
Figura 29.....	59
Notificación registro de roles de usuario correcto .....	59
Figura 30.....	59
Pantalla eliminación de roles de usuario.....	59
Figura 31.....	60
Notificación eliminación de roles de usuario correcta.....	60
Figura 31.....	64
Pantalla lista de cargos de los empleados .....	64
Figura 32.....	64
Pantalla modificación de cargos de los empleados .....	64

Figura 33 .....	65
Notificación modificación de cargos de los empleados correcta .....	65
Figura 34 .....	65
Pantalla registro de cargos de los empleados.....	65
Figura 35 .....	66
Notificación registro de cargos de los empleados correcto.....	66
Figura 36 .....	66
Pantalla eliminación de cargos de los empleados .....	66
Figura 37 .....	67
Notificación eliminación de cargos de los empleados correcta .....	67
Figura 38 .....	68
Pantalla listado de horarios de los empleados.....	68
Figura 39 .....	68
Pantalla modificación de los horarios de los empleados.....	68
Figura 40 .....	68
Notificación modificación de los horarios de los empleados correcta.....	68
Figura 41 .....	69
Pantalla registro de los horarios de los empleados.....	69
Figura 42 .....	69
Notificación registro de los horarios de los empleados correcto .....	69
Figura 43 .....	70
Pantalla eliminación de los horarios de los empleados .....	70
Figura 44 .....	70
Notificación eliminación de los horarios de los empleados correcta .....	70
Figura 45 .....	75
Pantalla lista de empleados .....	76
Figura 46 .....	76
Pantalla modificación de empleado .....	76
Figura 47 .....	77
Notificación modificación de empleado correcta .....	77
Figura 48 .....	77
Pantalla registro de empleado .....	77
Figura 49 .....	78
Notificación registro de empleado correcto .....	78
Figura 50 .....	78

Pantalla eliminación de empleado.....	78
Figura 51 .....	79
Notificación eliminación de empleado correcta.....	79
Figura 52 .....	79
Pantalla listado de roles de empleado .....	79
Figura 53 .....	80
Pantalla modificación de roles de empleado.....	80
Figura 54 .....	80
Notificación modificación de roles de empleado correcta.....	80
Figura 55 .....	81
Pantalla registro de roles de empleado.....	81
Figura 56 .....	81
Notificación registro de roles de empleado correcto.....	81
Figura 57 .....	82
Pantalla eliminación de roles de empleado .....	82
Figura 58 .....	82
Notificación eliminación de roles de empleado correcto.....	82
Figura 59 .....	87
Pantalla registro de asistencia del empleado.....	87
Figura 60 .....	88
Notificación registro de entrada.....	88
Figura 61 .....	89
Notificación registro de salida .....	89
Figura 62 .....	89
Pantalla de listado de asistencias .....	89
Figura 63 .....	90
Pantalla de eliminación de asistencias .....	90
Figura 64 .....	90
Notificación eliminación de asistencias correcta .....	90
Figura 65 .....	91
Pantalla de listado de justificaciones .....	91
Figura 66 .....	91
Pantalla de modificación de justificaciones .....	91
Figura 67 .....	92
Notificación modificación de justificaciones correcta.....	92

Figura 68 .....	92
Pantalla listado de asistencias, caso estado justificado .....	92
Figura 69 .....	93
Pantalla listado de asistencias, caso exportación de datos en formato .csv .....	93
Figura 70 .....	93
Archivo Excel exportado .....	93

# ÍNDICE TABLAS

Tabla 1. Tabla comparativa de bases de datos .....	10
Tabla 2. Tabla comparativa de lenguajes de programación.....	13
Tabla 3. Tabla comparativa de arquitecturas .....	22
Tabla 4. Tabla comparativa de metodologías de desarrollo .....	25
Tabla 5. Tabla Roles y Responsabilidades del Prototipado .....	28
Tabla 6. Tabla Requerimientos Funcionales.....	31
Tabla 7. Tabla componentes de un caso de uso .....	34
Tabla 8. Tabla actividades Iteración 1 .....	49
Tabla 9. Tabla actividades Iteración 2 .....	64
Tabla 10. Tabla actividades Iteración 3 .....	75
Tabla 11. Tabla actividades Iteración 4 .....	87

# CAPÍTULO I

## INTRODUCCIÓN

### 1. MARCO DE REFERENCIA

#### 1.1 JUSTIFICACIÓN

Las empresas están en constante evolución, buscando optimizar sus procesos y mejorar la productividad. El control de asistencia de los empleados administrativos es crucial para garantizar el cumplimiento de horarios y la eficiencia operativa. El registro manual de asistencia puede ser laborioso y propenso a errores. Según la Organización Internacional del Trabajo (OIT), hasta el 20% de las empresas que utilizan métodos manuales de registro de asistencia experimentan errores significativos, lo que lleva a una pérdida estimada del 15% en la productividad laboral (Organización Internacional del Trabajo, 2019).

Investigaciones de la revista Harvard Business Review indican que aproximadamente el 30% de las decisiones estratégicas en la gestión del personal se ven afectadas negativamente por la falta de precisión en los datos de asistencia (Harvard Business Review, 2022). Por esta razón, desarrollar un prototipo de sistema web para el registro y control de asistencia para empleados administrativos es de gran valor. Este sistema automatizaría el proceso de registro, mejorando la precisión de los datos y proporcionando información detallada sobre el comportamiento de los empleados. La implementación de un sistema automatizado de registro de asistencia puede reducir los errores manuales en un 25% y aumentar la eficiencia en la gestión del personal, lo que apoya la toma de decisiones estratégicas basadas en datos precisos y actualizados (McKinsey & Company, 2019).

#### 1.2 PLANTEAMIENTO DEL PROBLEMA

El problema principal es la falta de un sistema automatizado para el registro y control de asistencia de los empleados. Esta carencia provoca ineficiencia en el proceso de registro,

imprecisión en los datos y una capacidad limitada para obtener información relevante sobre el comportamiento de los empleados. Solucionar este problema mejoraría la eficiencia y precisión en el control de asistencia y proporcionaría datos completos y detallados que facilitarían la toma de decisiones estratégicas para optimizar las operaciones de la empresa.

A partir de la identificación del problema, se plantean las siguientes preguntas de investigación:

¿Cómo se puede automatizar el registro y control de asistencia de los empleados administrativos de una empresa?

Preguntas secundarias:

- ¿Cuáles son los beneficios de la automatización del registro y control de asistencia de los empleados administrativos?
- ¿Cuáles son los requisitos específicos del sistema web para el registro y control de asistencia de los empleados administrativos?
- ¿Cuál es el impacto esperado de la automatización del registro y control de asistencia en la eficiencia, productividad y toma de decisiones de las empresas?

### 1.3 OBJETIVO GENERAL

Desarrollar un prototipo de sistema web para el registro y control de asistencia de empleados administrativos de una empresa, que permita automatizar el proceso, mejorar la precisión de los datos y proporcionar información detallada sobre el comportamiento de los empleados.

### 1.4 OBJETIVOS ESPECÍFICOS

- Identificar las necesidades de los empleados administrativos relacionados con el registro y control de asistencia.
- Diseñar un prototipo de sistema web que facilite a los empleados administrativos registrar su asistencia de manera rápida y precisa.

- Construir un prototipo de sistema web basado en una plataforma intuitiva, efectiva para sus usuarios potenciales.
- Implementar un sistema que provea información detallado en base a los requerimientos del usuario.

## 1.5 ALCANCE

Este proyecto tiene como meta desarrollar un sistema web dedicado al registro y seguimiento de la asistencia del personal administrativo. Utilizaremos tecnologías web actuales, como PHP, HTML, CSS y JavaScript, garantizando así un sistema moderno y adaptable. Para responder ágilmente a las necesidades del usuario, emplearemos una metodología flexible con un enfoque evolutivo de prototipado.

Entre las limitaciones identificadas se encuentra el tiempo de desarrollo, crucial para satisfacer las necesidades del proyecto. Además, debido al enfoque acelerado del desarrollo de prototipos, la cantidad de documentación podría verse afectada. Esto podría incidir en nuestra capacidad para justificar exhaustivamente las decisiones de diseño y desarrollo.

El sistema propuesto ofrecerá un registro automatizado y preciso de la asistencia, generación de reportes detallados para facilitar la toma de decisiones, y un control eficaz para los usuarios. Todo esto se realizará dentro de un marco de tiempo definido para el desarrollo del sistema.

# CAPÍTULO II

## FUNDAMENTACIÓN TEÓRICA

### 2. MARCO TEÓRICO Y CONCEPTUAL

#### 2.1 ANTECEDENTES

La implementación de sistemas automatizados de registro de asistencia ha emergido como una solución clave para mejorar la precisión de los datos y la eficiencia del tiempo en diversos contextos laborales. Investigaciones recientes han arrojado luz sobre los beneficios de estos sistemas en distintos ámbitos, subrayando su relevancia para nuestro estudio actual sobre el desarrollo de un sistema web para el registro y control de asistencia de empleados administrativos. El estudio de (Almaraz, 2023) sobre el impacto de los sistemas automatizados de registro de asistencia en el rendimiento académico en educación superior subraya la importancia de estos sistemas en la mejora del desempeño y la productividad, lo cual es relevante para nuestro contexto administrativo. Esta investigación encontró una correlación positiva entre la adopción de tales sistemas y el mejoramiento del desempeño y la productividad, destacando la necesidad de implementar sistemas eficaces de registro de asistencia en instituciones educativas y la correcta gestión de horarios.

En el ámbito laboral, (Alshammari, 2021) examinó cómo los sistemas basados en web transforman el seguimiento de la asistencia de los empleados y mejoran la eficiencia organizacional. Este estudio reveló que la implementación de sistemas automatizados no solo redujo los errores administrativos asociados con los métodos manuales, sino que también optimizó la gestión del tiempo del personal de recursos humanos, permitiéndoles concentrarse en tareas más estratégicas. Este hallazgo es crucial para nuestro objetivo de mejorar la eficiencia operativa y la precisión en la gestión del tiempo en entornos laborales.

En el sector educativo, (García, 2018) profundizó en cómo las herramientas basadas en web pueden mejorar el monitoreo de la asistencia del personal en entornos de docencia. Este estudio mostró que la implementación de sistemas automatizados no solo mejoró la precisión del registro de asistencia, sino que también permitió una asignación más eficiente de recursos y una respuesta

más ágil a las necesidades de los alumnos. Estos hallazgos son aplicables a nuestro objetivo de desarrollar un sistema que mejore la gestión del personal y la atención a los empleados administrativos.

(Hassan, 2020) destacó la importancia de la flexibilidad y accesibilidad en los sistemas web de registro de asistencia. Su estudio resalta cómo la capacidad de acceder al sistema desde cualquier ubicación y en cualquier momento mejora la eficiencia y la comodidad para los empleados, lo que a su vez puede traducirse en una mayor satisfacción laboral y productividad. Este aspecto es vital para nuestro diseño de un sistema web que no solo sea eficiente, sino también fácil de usar y accesible.

Además de los estudios técnicos, (Wilson, 2015) exploró los beneficios internos de la implementación de sistemas automatizados de registro de asistencia, señalando que estos pueden influir positivamente en la moral y el compromiso de los empleados. Este estudio encontró que al introducir un sistema que monitorea y registra de manera precisa las horas de trabajo, se fomenta una cultura de responsabilidad y transparencia en relación con el tiempo laboral. Este hallazgo es importante para nuestro objetivo de no solo mejorar la eficiencia operativa, sino también contribuir al bienestar y la motivación de los empleados.

En conjunto, estos estudios respaldan la necesidad y la pertinencia de abordar la problemática planteada en esta tesis. La combinación de evidencia sobre los beneficios de la automatización, la flexibilidad y la influencia positiva en la moral de los empleados proporciona un sólido fundamento para el desarrollo de un sistema web para el registro y control de asistencia de empleados administrativos. Este sistema no solo tiene el potencial de mejorar la eficiencia operativa y la precisión de los datos, sino también de contribuir al bienestar y la motivación de los empleados, aspectos cruciales para el éxito a largo plazo de cualquier organización.

## 2.2 MARCO CONCEPTUAL

### 2.2.1 CONTROL DE REGISTRO DE ASISTENCIA

El control de registro de asistencia se refiere al proceso de monitoreo y registro de la presencia y el tiempo laboral de los empleados en una organización. En este sentido, se fundamenta en la teoría de la gestión del tiempo y la productividad laboral. Según (Taylor, 1911), la gestión

científica del trabajo busca optimizar la eficiencia mediante la estandarización de métodos y la supervisión del rendimiento. En el contexto del control de registro de asistencia, esta teoría implica la necesidad de establecer procedimientos claros y eficientes para registrar y gestionar el tiempo de trabajo de manera sistemática.

Además, el control de registro de asistencia se relaciona con la teoría del compromiso organizacional, que sugiere que la implicación y la responsabilidad de los empleados hacia su trabajo pueden influir en su desempeño y lealtad hacia la organización (Meyer & Allen, 1911). En este sentido, un sistema efectivo de registro de asistencia puede fomentar un mayor compromiso al proporcionar una estructura clara para la rendición de cuentas y la transparencia en relación con el tiempo laboral.

## 2.2.2 APLICACIONES WEB

La realización de aplicaciones web se fundamenta en la necesidad de adaptarse a la realidad tecnológica actual. Estas herramientas basadas en la web ofrecen funcionalidades y servicios a través de un navegador, abarcando diversas tecnologías y lenguajes de programación como HTML, CSS, y JavaScript. Su accesibilidad desde cualquier dispositivo con conexión a Internet las hace extremadamente versátiles y convenientes en una era digital donde la movilidad y la accesibilidad son esenciales para el desarrollo de actividades humanas.

Según (Nielsen, 2000), la usabilidad y la accesibilidad son aspectos fundamentales en el diseño de aplicaciones web. La usabilidad se refiere a la facilidad con la que los usuarios pueden aprender a usar una aplicación y alcanzar sus objetivos de manera eficiente, mientras que la accesibilidad se refiere a la capacidad de la aplicación para ser utilizada por personas con diferentes capacidades, incluidas aquellas con discapacidades. Esto implica la necesidad de crear interfaces intuitivas y eficientes que permitan a los usuarios interactuar de manera efectiva con la aplicación. La implementación de principios de diseño centrado en el usuario y pruebas de usabilidad son prácticas esenciales para garantizar que una aplicación web cumpla con estos estándares.

Las aplicaciones web se enmarcan en la teoría de la tecnología de la información (TI), que considera cómo las tecnologías informáticas pueden utilizarse para facilitar la gestión y el intercambio de información en las organizaciones (Laudon & Laudon, 2020). En este contexto, el

desarrollo de aplicaciones web para el control de registro de asistencia representa una aplicación práctica de la TI para mejorar la eficiencia y la precisión en la gestión de recursos humanos.

## 2.3 MARCO TEORICO

### 2.3.1 VISUAL STUDIO

Visual Studio es un Entorno de Desarrollo Integrado (IDE) desarrollado por Microsoft, utilizado principalmente para el desarrollo de aplicaciones para sistemas operativos Windows, aplicaciones web, servicios web y aplicaciones móviles. La herramienta es conocida por su capacidad de integración con una amplia variedad de herramientas de desarrollo y servicios, su soporte para múltiples lenguajes de programación y su conjunto de características diseñadas para mejorar la productividad de los desarrolladores.

Entre sus características principales se incluye el soporte multilenguaje, lo que permite a los desarrolladores trabajar en diferentes tipos de proyectos dentro del mismo entorno, y la depuración avanzada, que ofrece herramientas robustas para detectar y solucionar problemas de manera eficiente. Su integración con sistemas de control de versiones como Git y Subversion, y sus herramientas para pruebas de unidades, pruebas de integración y análisis estático de código, mejoran significativamente la productividad y la calidad del código. Además de la eficiencia y productividad aumentan al integrar múltiples herramientas dentro de un solo entorno, y su estrecha integración con el ecosistema de Microsoft facilita la creación de aplicaciones que aprovechan al máximo los servicios y tecnologías. (Microsoft, s.f.)

### 2.3.2 XAMPP

XAMPP es una distribución gratuita y de código abierto que permite instalar fácilmente un servidor web Apache junto con bases de datos MySQL, el lenguaje de programación PHP y Perl. La herramienta es conocida por su fácil instalación y configuración, y no requiere conocimientos avanzados para poner en marcha un servidor local. Al ser una aplicación de código abierto y multiplataforma, puede ser instalado en diferentes sistemas operativos, incluyendo Windows, Linux y macOS. Viene con todos los componentes necesarios para el desarrollo web, incluyendo

el servidor Apache, la base de datos MySQL, y los lenguajes de programación PHP y Perl, todos preconfigurados y listos para usar. Además, incluye un panel de control intuitivo que permite a los usuarios iniciar y detener servicios, configurar componentes y acceder a herramientas adicionales como phpMyAdmin para la gestión de bases de datos. En el contexto del proyecto, XAMPP permite configurar un entorno de desarrollo local rápidamente, facilitando el desarrollo y prueba de aplicaciones web antes de desplegarlas en un entorno de producción. (Apache Friends., s.f.)

### 2.3.3 POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos relacionales de objetos (ORDBMS) de código abierto, conocido por su robustez, flexibilidad y conformidad con estándares SQL. Es altamente compatible con SQL estándar, lo que facilita la migración y la interoperabilidad con otros sistemas. Permite a los desarrolladores añadir nuevas funciones a través de extensiones, procedimientos almacenados y tipos de datos personalizados. PostgreSQL soporta transacciones completas ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), lo que garantiza la integridad y confiabilidad de los datos, y proporciona un soporte robusto para datos en formato JSON. Es altamente escalable tanto vertical como horizontalmente, soportando grandes volúmenes de datos y operaciones concurrentes, y ofrece múltiples métodos de replicación y técnicas de clustering para garantizar la alta disponibilidad y la recuperación ante desastres. La fiabilidad y estabilidad de PostgreSQL son cruciales para aplicaciones críticas que requieren alta disponibilidad y consistencia de datos, y su capacidad para manejar grandes volúmenes de datos y altas cargas de trabajo lo convierte en una excelente elección para aplicaciones de gran escala. (PostgreSQL Global Development Group., s.f.)

### 2.3.4 SQL SERVER

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) desarrollado por Microsoft. Se integra perfectamente con otras herramientas de Microsoft, como Visual Studio, Azure y Office, facilitando la creación y gestión de soluciones integrales. Ofrece un rendimiento optimizado con características como particionamiento de tablas y consultas paralelas, e incluye funciones avanzadas de seguridad, como la autenticación integrada de Windows, encriptación de

datos en reposo y en tránsito, y auditoría detallada. Proporciona soluciones de alta disponibilidad como Always On Availability Groups, replicación transaccional y clustering de conmutación por error. Además, incluye herramientas como SQL Server Analysis Services (SSAS) y SQL Server Integration Services (SSIS) para análisis de datos y ETL (Extracción, Transformación y Carga). (Microsoft., s.f.)

### 2.3.5 MYSQL

MySQL, se define como un sistema de gestión de base de datos relacional (SGBD) de código abierto, se destaca por su facilidad de uso y su perfecta integración con proyectos de desarrollo web. Su integración con XAMPP lo convierte en una herramienta popular para la creación rápida de prototipos y el desarrollo ágil. Al ser de código abierto, esta ofrece ventajas como una comunidad de usuarios activa, flexibilidad para la personalización del software y costos de licencia reducidos. Sin embargo, también presenta desafíos como la necesidad de soporte y mantenimiento autogestionados.

Otra característica importante de la herramienta es que es escalable y puede manejar grandes volúmenes de datos y altas demandas de procesamiento a través de funciones como la replicación, el particionamiento y la optimización de consultas. Su compatibilidad con los estándares SQL facilita el desarrollo y la migración de aplicaciones. (MySQL, s.f.)

### 2.3.6 TABLA COMPARATIVA DE BASES DE DATOS

<b>Funcionalidades</b>	<b>MySQL</b>	<b>PostgreSQL</b>	<b>SQL Server</b>
<b>Tipos de datos</b>	Básicos (INT, VARCHAR, etc.), JSON, XML	Avanzados (JSON, XML, Hstore, etc.)	Básicos y avanzados (JSON, XML, etc.)
<b>Compatibilidad con lenguajes de programación</b>	Amplia (PHP, Python, Java, Node.js, etc.)	Amplia (PHP, Python, Java, Node.js, etc.)	.NET, Java, Python, Node.js, etc.

<b>Disponibilidad de herramientas</b>	Amplia (MySQL Workbench, HeidiSQL, etc.)	Amplia (pgAdmin, DBeaver, etc.)	SQL Server Management Studio (SSMS), Azure Data Studio
<b>Rendimiento</b>	Buen rendimiento en lectura y escritura	Excelente rendimiento en lectura y escritura soporta operaciones concurrentes eficientemente	Alto rendimiento, optimizado para transacciones complejas y grandes volúmenes de datos
<b>Escalabilidad</b>	Buena escalabilidad vertical	Excelente escalabilidad horizontal y vertical	Alta escalabilidad vertical y horizontal con particionamiento y replicación
<b>Uso práctico</b>	Pequeños y medianos proyectos	Proyectos medianos y grandes	Grandes proyectos empresariales

Tabla 1. Tabla comparativa de bases de datos

*Fuente. (PostgreSQL Global Development Group., s.f.), (MySQL, s.f.) y (Microsoft., s.f.)*

### 2.3.7 PYTHON

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, conocido por su simplicidad y legibilidad. Es ampliamente utilizado en una variedad de aplicaciones, desde desarrollo web hasta análisis de datos y machine learning. Su sintaxis clara y legible facilita el desarrollo rápido y la comprensión del código, y cuenta con una amplia gama de bibliotecas y frameworks que simplifican el desarrollo de aplicaciones, incluyendo Django y Flask para desarrollo web, y NumPy, Pandas y TensorFlow para análisis de datos y machine learning. Una gran y activa comunidad de desarrolladores contribuye al desarrollo continuo de Python y proporciona soporte a través de foros, documentación y recursos en línea. Python es compatible

con múltiples sistemas operativos, incluyendo Windows, macOS y Linux, lo que facilita la portabilidad de las aplicaciones, y puede integrarse con otros lenguajes y tecnologías, permitiendo su uso en entornos heterogéneos. Su sintaxis clara y la disponibilidad de numerosas bibliotecas permiten un desarrollo rápido y eficiente, y su flexibilidad permite su uso en una amplia variedad de aplicaciones, desde scripts simples hasta aplicaciones complejas de machine learning. (Python Software Foundation, s.f.)

### 2.3.8 JAVA

Java es un lenguaje de programación de alto nivel, orientado a objetos y de propósito general, desarrollado por Sun Microsystems, ahora propiedad de Oracle. Este lenguaje de programación está completamente basado en el paradigma de la programación orientada a objetos, lo que facilita la modularidad y la reutilización del código. El bytecode de Java puede ejecutarse en cualquier plataforma que tenga una Máquina Virtual Java (JVM), lo que garantiza la portabilidad del código. Aunque es un lenguaje interpretado, Java ofrece un rendimiento cercano al de lenguajes compilados gracias a la optimización en tiempo de ejecución (JIT) de la JVM. Incluye características de seguridad robustas, como el manejo de permisos y la ejecución en una máquina virtual aislada del sistema operativo, y cuenta con un extenso conjunto de bibliotecas estándar y frameworks que facilitan el desarrollo de aplicaciones empresariales, web, móviles y de big data. La programación orientada a objetos facilita la creación de código modular y reutilizable, mejorando la mantenibilidad y escalabilidad de las aplicaciones, y la capacidad de ejecutar aplicaciones Java en cualquier plataforma con JVM aumenta la flexibilidad y el alcance de las aplicaciones desarrolladas. La amplia gama de bibliotecas y frameworks disponibles en Java acelera el desarrollo y permite crear soluciones robustas y escalables para diversas necesidades empresariales. (Deitel & Deitel, 2023)

### 2.3.9 PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo web del lado del servidor. Es conocido por su simplicidad, flexibilidad y rápida curva de aprendizaje. PHP se integra fácilmente con HTML, permitiendo a los

desarrolladores insertar scripts PHP directamente en páginas web. Es fácil de aprender y usar, lo que lo hace accesible para desarrolladores principiantes y expertos por igual, y se puede integrar con diversas bases de datos, incluyendo MySQL, PostgreSQL, Oracle y SQLite, proporcionando flexibilidad en la gestión de datos. PHP es altamente extensible y cuenta con una amplia gama de bibliotecas y frameworks, como Laravel y Symfony, que facilitan el desarrollo de aplicaciones web robustas y escalables. La gran y activa comunidad de PHP proporciona soporte, documentación y recursos en línea. La sintaxis sencilla y la integración con HTML permiten un desarrollo rápido y eficiente de aplicaciones web, y al ser de código abierto, PHP es una solución rentable para proyectos web de cualquier tamaño. Su capacidad de integrarse con múltiples bases de datos y tecnologías hace de PHP una opción versátil para el desarrollo de aplicaciones web dinámicas y complejas. (Zandstra, 2021)

### 2.3.10 TABLA COMPARATIVA DE LENGUAJES DE PROGRAMACIÓN

Característica	PHP	Python	JavaScript
<b>Manejo de bases de datos</b>	Se integra perfectamente con una amplia gama de bases de datos, como MySQL, PostgreSQL, MariaDB, Oracle, y MSSQL.	Ofrece soporte robusto para diversas bases de datos (MySQL, PostgreSQL, SQLite, etc.) con bibliotecas como SQLAlchemy y Django ORM.	Puede conectarse a bases de datos mediante Node.js, con soporte para MySQL, PostgreSQL, MongoDB, y más, aunque requiere configuración adicional.
<b>Seguridad</b>	Cuenta con características de seguridad integradas, como protección contra CSRF, XSS, y SQL Injection mediante herramientas como PDO.	Se considera seguro debido a su tipado dinámico y la robustez de su ecosistema; frameworks como Django y Flask incluyen medidas de seguridad avanzadas.	Tiene vulnerabilidades conocidas, pero se pueden mitigar usando buenas prácticas de codificación y herramientas como Node.js Security Project.

<b>Escalabilidad</b>	Puede escalarse vertical y horizontalmente, y es adecuado para aplicaciones web de tamaño medio a grande.	Destaca por su escalabilidad horizontal gracias a frameworks como Django y Flask, y es adecuado para aplicaciones de ciencia de datos y desarrollo web.	Es ideal para aplicaciones web en tiempo real y altamente escalables, especialmente con Node.js y frameworks como Express.js.
<b>Entorno de ejecución</b>	Se ejecuta en el servidor (back-end), ideal para tareas como procesar formularios, acceder a bases de datos y generar contenido dinámico.	Puede ejecutarse en el servidor (back-end) y en el lado del cliente (front-end) con herramientas como Jupyter y Brython, proporcionando gran versatilidad.	Se ejecuta principalmente en el lado del cliente (front-end), permitiendo una experiencia de usuario interactiva y dinámica; también puede ejecutarse en el servidor con Node.js.
<b>Rendimiento</b>	Rendimiento eficiente en tareas web comunes; puede ser mejorado con cachés como APCu y OPcache.	Buen rendimiento general; excelente en tareas de computación científica y análisis de datos con bibliotecas optimizadas como NumPy y Pandas.	Alto rendimiento en aplicaciones web en tiempo real y altamente concurrentes con Node.js; V8 engine de Google mejora su velocidad.
<b>Popularidad y Comunidad</b>	Amplia popularidad y una comunidad activa con muchos recursos, bibliotecas y frameworks disponibles.	Muy popular, especialmente en ciencia de datos, desarrollo web y automatización, con una comunidad grande y activa.	Extremadamente popular para desarrollo web, tanto en el front-end (React, Angular) como en el back-end (Node.js), con una comunidad masiva y vibrante.

Tabla 2. Tabla comparativa de lenguajes de programación

Fuente. (Python Software Foundation, s.f.), (Deitel & Deitel, 2023) y (Zandstra, 2021)

### 2.3.11 HTML

HTML (Lenguaje de Marcado de Hipertexto) es un lenguaje estándar para los desarrolladores, utilizado prioritariamente para la creación y diseño de páginas web. Este proporciona una estructura básica para crear páginas web, permitiendo a los desarrolladores definir la jerarquía y disposición del contenido. Utiliza una serie de elementos y etiquetas para definir

diferentes tipos de contenido, como texto, imágenes, enlaces y formularios. La sintaxis es sencilla y fácil de entender, lo que facilita su aprendizaje y uso por parte de desarrolladores de todos los niveles. HTML es la base de la web y se utiliza en combinación con CSS y JavaScript para crear páginas web interactivas y visualmente atractivas. Es compatible con todos los navegadores web modernos, garantizando que las páginas web se muestren correctamente en diferentes plataformas. Su simplicidad permite a los desarrolladores crear rápidamente páginas web estructuradas y bien organizadas, y su compatibilidad con otros lenguajes y tecnologías permite la creación de sitios web dinámicos e interactivos. Al ser el lenguaje estándar para el diseño web, es fundamental para cualquier proyecto de desarrollo web, proporcionando la base sobre la cual se construyen las páginas web. (Robbins, 2012)

### 2.3.12 JAVASCRIPT

JavaScript es un lenguaje de programación ampliamente utilizado para desarrollar contenido interactivo en páginas web. A diferencia de HTML y CSS, que se encargan de la estructura y la presentación, JavaScript añade interactividad y comportamiento dinámico a los elementos web. Utiliza una sintaxis sencilla basada en objetos y funciones, lo que permite a los desarrolladores crear efectos como menús desplegables, carruseles de imágenes, validación de formularios y actualizaciones de contenido en tiempo real sin necesidad de recargar la página. JavaScript se puede integrar directamente en el código HTML o incluirse como un archivo separado, y es compatible con todos los navegadores web modernos. Además, su versatilidad se ve ampliada por una gran cantidad de bibliotecas y frameworks, como jQuery, React y Angular, que simplifican el desarrollo de aplicaciones web complejas. La capacidad de manipular el Document Object Model (DOM) permite a JavaScript responder a eventos del usuario y modificar el contenido y el estilo de la página de manera dinámica. Esta capacidad de crear experiencias interactivas y responsivas es fundamental en el desarrollo web moderno, mejorando significativamente la usabilidad y la experiencia del usuario. JavaScript no solo es crucial para el frontend, sino que también se utiliza en el backend a través de plataformas como Node.js, lo que lo convierte en una herramienta integral para desarrolladores web. (Robbins, 2012)

### 2.3.13 CSS

CSS (Hojas de Estilo en Cascada) es un lenguaje utilizado para describir la presentación de documentos HTML o XML. CSS permite a los desarrolladores separar el contenido de la presentación, facilitando el mantenimiento y actualización del diseño de una página web. Utiliza una sintaxis sencilla para definir estilos para elementos HTML, como colores, fuentes, márgenes, alineaciones y disposición del contenido. CSS permite crear diseños responsivos que se adaptan a diferentes tamaños de pantalla y dispositivos, mejorando la experiencia del usuario en móviles y tabletas. La capacidad de reutilizar estilos y crear hojas de estilo externas facilita la consistencia y el mantenimiento del diseño en múltiples páginas web. CSS es compatible con todos los navegadores web modernos y se puede utilizar en combinación con HTML y JavaScript para crear sitios web dinámicos e interactivos. La separación del contenido y la presentación permite a los desarrolladores actualizar fácilmente el diseño sin afectar el contenido, y la capacidad de crear diseños responsivos mejora la accesibilidad y usabilidad de las páginas web en diferentes dispositivos. CSS es esencial para el desarrollo web moderno, proporcionando las herramientas necesarias para crear diseños atractivos y funcionales. (Robbins, 2012)

### 2.3.14 ADMINISTRACIÓN DE RECURSOS HUMANOS

La administración de recursos humanos (ARH) es esencial para el éxito de cualquier organización, ya que se encarga de gestionar el capital humano desde la adquisición de empleados hasta su mantenimiento a lo largo de su carrera en la empresa. Según (Mathis & Jackson, 2011), ARH busca asegurar que las metas organizacionales se alcancen de manera eficiente y efectiva, lo cual implica una gestión integral de todos los aspectos relacionados con los empleados, desde el reclutamiento y la selección hasta el desarrollo y la retención del talento. La automatización del registro de asistencia se ha convertido en una herramienta fundamental para la ARH, sustituyendo el registro manual y mejorando la precisión, eficiencia e integridad de los datos. Esta automatización permite la recopilación de información en tiempo real sobre la asistencia de los empleados, facilitando el análisis de patrones para detectar problemas y tomar decisiones informadas. Para las empresas, esta tecnología simplifica las tareas administrativas, mejora la

precisión de los datos y facilita la toma de decisiones estratégicas. Para los empleados, promueve un ambiente laboral más transparente y equitativo, reduce errores y manipulaciones en el registro de horas trabajadas, y fomenta la confianza y la satisfacción laboral, contribuyendo así a un entorno de trabajo más positivo y productivo.

### 2.3.15 AUTOMATIZACIÓN DE PROCESOS

La automatización de procesos implica la aplicación de tecnología para realizar tareas de manera automatizada, reduciendo en la medida de lo posible la intervención humana con la finalidad de impulsar la eficiencia de dichos procesos. (Davenport, 2018) señala que la automatización de procesos no solo reduce la carga de trabajo manual, sino que también mejora la precisión y la velocidad de las tareas realizadas, permitiendo a las empresas responder de manera más ágil y efectiva a las demandas del mercado. La automatización puede aplicarse a una amplia variedad de procesos, desde la producción y el control de calidad hasta la gestión de la cadena de suministro y los servicios al cliente, proporcionando beneficios significativos en términos de eficiencia operativa y competitividad.

### 2.3.16 INGENIERÍA DE SOFTWARE

Según (Pressman, 2014), la ingeniería de software es la disciplina que se ocupa de todos los aspectos de la producción de software, desde las etapas iniciales de la especificación del sistema hasta el mantenimiento del sistema después de que se ha puesto en operación. Esto implica el uso de buenas prácticas de ingeniería para un desarrollo de alta calidad en todo el ciclo de vida del software desarrollado. La ingeniería de software abarca una amplia gama de actividades, incluyendo el análisis de requisitos, el diseño, la codificación, las pruebas y el mantenimiento, todas ellas orientadas a garantizar que el software cumpla con los estándares de calidad y las expectativas del usuario. Además, la ingeniería de software también aborda aspectos como la gestión de proyectos, la documentación y la mejora continua, asegurando que el software se desarrolle de manera eficiente y se mantenga relevante y funcional a lo largo del tiempo.

### 2.3.17 DESARROLLO DE SOFTWARE

El desarrollo de software se define como la creación de un software de alto nivel de calidad realizado en un plazo y presupuesto determinado. (Sommerville, 2015) destaca la importancia del tiempo, la calidad y el costo durante todo el proceso del desarrollo de software, subrayando que un buen desarrollo de software debe equilibrar estos tres factores para lograr resultados satisfactorios. El desarrollo de software implica una serie de etapas, desde la concepción inicial y la planificación hasta la implementación, las pruebas y el despliegue, cada una de las cuales requiere una cuidadosa gestión y coordinación. Un enfoque meticuloso en el desarrollo de software no solo garantiza que el producto final sea funcional y eficiente, sino que también asegura que se entregue a tiempo y dentro del presupuesto, cumpliendo con las expectativas de los clientes y usuarios finales.

### 2.3.18 ANÁLISIS DE SISTEMAS

(Kendall & Kendall, 2014) presentan al análisis de sistemas como el proceso de entender y especificar en detalle qué debe hacerse en un sistema de información. Este proceso es fundamental para detectar y documentar los requerimientos del sistema, estableciendo una base sólida para el desarrollo y diseño de futuros proyectos. El análisis de sistemas implica una exploración profunda de las necesidades y objetivos de los usuarios, así como de las restricciones y limitaciones técnicas, para desarrollar una visión clara y detallada del sistema a implementar. Este análisis detallado permite identificar posibles problemas y oportunidades de mejora, asegurando que el sistema desarrollado sea eficiente, efectivo y alineado con los objetivos organizacionales. Además, un análisis de sistemas bien realizado facilita la comunicación entre los desarrolladores y los usuarios, garantizando que todos los involucrados comprendan y estén de acuerdo con los requisitos y expectativas del proyecto.

### 2.3.19 DISEÑO DE SISTEMAS

El diseño de sistemas se centra en definir la arquitectura técnica del sistema, según (Whitten, Bentley, & Dittman, 2007). Esto implica una representación detallada del

funcionamiento del sistema, incluyendo la estructura, la funcionalidad y la interacción de los componentes. El diseño de sistemas es una etapa crucial en el desarrollo de software, ya que establece el plan y la estructura sobre los cuales se construirá el sistema final. Un buen diseño de sistemas considera tanto los requisitos funcionales como los no funcionales, asegurando que el sistema sea robusto, escalable y mantenible. Además, un diseño bien elaborado facilita la implementación y el mantenimiento del sistema, reduciendo el riesgo de errores y problemas técnicos a lo largo de su ciclo de vida. La definición clara de la arquitectura técnica también permite una mejor gestión de los recursos y una implementación más eficiente, asegurando que el sistema cumpla con los objetivos establecidos y proporcione valor a la organización.

### 2.3.20 DISEÑO DE INTERFACES DE USUARIO

(Cooper, 2014) define el diseño de interfaces de usuario como el proceso de crear una forma de comunicación entre el usuario y la computadora, subrayando la importancia de diseñar interfaces gráficas que sean intuitivas, atractivas e inclusivas. Una buena interfaz de usuario facilita la interacción del usuario con el sistema, mejorando la usabilidad y la satisfacción del usuario. El diseño de interfaces de usuario implica considerar una variedad de factores, como la disposición visual, la navegación, la accesibilidad y la respuesta del sistema, para crear una experiencia de usuario fluida y agradable. Un diseño de interfaz bien ejecutado no solo mejora la eficiencia y la productividad del usuario, sino que también puede influir positivamente en la percepción y la aceptación del sistema. Además, una interfaz de usuario intuitiva y bien diseñada puede reducir la necesidad de capacitación y soporte, mejorando la adopción y el uso del sistema.

### 2.3.21 SEGURIDAD DE LA INFORMACIÓN

(Stallings & & Brown, 2014) indican que la seguridad de la información involucra la protección de la información y los sistemas informáticos contra el acceso no autorizado. Destacan la importancia de implementar medidas de seguridad para proteger los datos de vulnerabilidades, amenazas y riesgos, garantizando la confidencialidad, integridad y disponibilidad del sistema. La seguridad de la información es un aspecto crítico de la gestión de TI, ya que las amenazas a la seguridad pueden tener consecuencias graves para las organizaciones, incluyendo la pérdida de

datos, la interrupción del negocio y daños a la reputación. Las medidas de seguridad pueden incluir el uso de firewalls, cifrado, autenticación y controles de acceso, así como la implementación de políticas y procedimientos de seguridad. Además, la seguridad de la información también implica la educación y concienciación de los empleados sobre las mejores prácticas de seguridad, asegurando que todos en la organización comprendan y cumplan con las medidas de seguridad establecidas. Un enfoque proactivo y comprensivo en la seguridad de la información ayuda a proteger los activos de la organización y a mantener la confianza de los clientes y socios comerciales.

### 2.3.22 ARQUITECTURA DE SOFTWARE

La arquitectura de software es la estructura o estructuras que se tienen en el sistema, comprendiendo componentes, sus propiedades visibles externamente, y las relaciones que existen entre ellos. (Bass, Clements, & Kazman, 2012) subrayan la importancia de definir una arquitectura sólida que guíe el desarrollo del sistema y garantice su coherencia y mantenibilidad. Una buena arquitectura de software proporciona un marco claro y organizado para el desarrollo, facilitando la integración de nuevos componentes y la adaptación a los cambios. Además, una arquitectura bien diseñada mejora la escalabilidad y el rendimiento del sistema, asegurando que pueda crecer y evolucionar con las necesidades de la organización. La arquitectura de software también juega un papel crucial en la gestión de la complejidad del sistema, permitiendo a los desarrolladores centrarse en componentes específicos sin perder de vista el panorama general. Al proporcionar una base sólida y coherente, una buena arquitectura de software contribuye a la calidad y la sostenibilidad a largo plazo del sistema, facilitando su mantenimiento y evolución.

### 2.3.23 ARQUITECTURA CLIENTE-SERVIDOR

La arquitectura Cliente-Servidor describe un modelo de diseño de software en el que las responsabilidades se dividen entre dos tipos de programas: el Cliente, que solicita servicios o recursos, y el Servidor, que los proporciona. (Bass, Clements, & Kazman, 2012) explican que esta estructura mejora el rendimiento al distribuir la carga entre ambos, centraliza la seguridad para proteger datos y recursos, y simplifica el desarrollo de aplicaciones distribuidas. En este modelo, el cliente interactúa directamente con el usuario, mientras que el servidor gestiona los datos y los

procesos necesarios para cumplir con las solicitudes del cliente. Esta separación de responsabilidades permite una mayor flexibilidad y escalabilidad, ya que los servidores pueden manejar múltiples clientes simultáneamente y los clientes pueden ser actualizados o reemplazados sin afectar al servidor. Además, la arquitectura Cliente-Servidor facilita la implementación de medidas de seguridad centralizadas, protegiendo los datos sensibles y asegurando que solo los usuarios autorizados puedan acceder a los recursos. Este enfoque también permite una gestión más eficiente de los recursos y una mejor capacidad de respuesta a las demandas del usuario, contribuyendo a la eficiencia y la efectividad del sistema en su conjunto.

### 2.3.24 ARQUITECTURA ORIENTADA A EVENTOS

La arquitectura orientada a eventos se fundamenta en la comunicación asincrónica entre componentes mediante eventos, teniendo como sus principales ventajas un bajo acoplamiento entre los componentes, lo que facilita modificaciones y mantenimiento, además de una alta escalabilidad al distribuir eventos eficientemente y una gran flexibilidad adaptable a distintos tipos de aplicaciones (Bellemare, 2014). En una arquitectura orientada a eventos, los componentes del sistema se comunican enviando y recibiendo eventos, lo que permite que funcionen de manera independiente y respondan de forma dinámica a los cambios y a las demandas del entorno. Este enfoque es especialmente útil en sistemas distribuidos y aplicaciones en tiempo real, donde la capacidad de responder rápidamente a eventos y de escalar según sea necesario es crucial. La arquitectura orientada a eventos también facilita la integración de nuevos componentes y la adaptación a nuevas necesidades, ya que los cambios en un componente no afectan directamente a los demás. Al reducir la dependencia entre componentes, esta arquitectura mejora la resiliencia y la robustez del sistema, permitiendo que continúe funcionando correctamente incluso en presencia de fallos o cambios inesperados.

### 2.3.25 ARQUITECTURA MVC (MODELO-VISTA-CONTROLADOR)

El MVC es una arquitectura de desarrollo de software que se ha convertido en un estándar común para el desarrollo de aplicaciones basadas en la web. (Fowler, 2006) destaca la relevancia

de adoptar un patrón arquitectónico para estructurar y organizar el código de manera efectiva en aplicaciones web. El modelo en la arquitectura MVC representa la estructura de datos de la aplicación y define la lógica de negocio. Es responsable de interactuar con la base de datos, realizar operaciones de lectura y escritura, y procesar la lógica empresarial. Al separar la lógica de negocio de la interfaz de usuario, el modelo permite una mayor reutilización del código y facilita la implementación de cambios en la aplicación. Esta separación de responsabilidades también mejora la capacidad de mantenimiento y la escalabilidad del sistema, ya que los desarrolladores pueden trabajar en diferentes partes de la aplicación de manera independiente. Además, el uso del patrón MVC promueve una mayor organización y claridad en el código, lo que facilita la colaboración entre los miembros del equipo y la comprensión del sistema por parte de nuevos desarrolladores.

### 2.3.25.1 MODELO

El modelo en la arquitectura MVC representa la estructura de datos de la aplicación y define la lógica de negocio. Es responsable de interactuar con la base de datos, realizar operaciones de lectura y escritura, y procesar la lógica empresarial. Al separar la lógica de negocio de la interfaz de usuario, el modelo permite una mayor reutilización del código y facilita la implementación de cambios en la aplicación.

### 2.3.25.2 VISTA

La vista en la arquitectura MVC es responsable de presentar la interfaz de usuario al usuario final. Muestra la información al usuario de una manera que sea fácil de entender y permite la interacción con la aplicación. La vista recibe datos del controlador y los muestra al usuario de acuerdo con el diseño y la estructura definidos. Al separar la presentación de la lógica de negocio, la vista permite una mayor flexibilidad en el diseño y la personalización de la interfaz de usuario.

### 2.3.25.3 CONTROLADOR

El controlador en la arquitectura MVC es un intermediario que existe entre el modelo y la vista, es responsable de recibir las solicitudes del usuario, procesarlas y coordinar la interacción

entre el modelo y la vista. El controlador interpreta las acciones del usuario y realiza las operaciones correspondientes en el modelo, luego pasa los datos necesarios a la vista para su presentación al usuario.

### 2.3.26 TABLA COMPARATIVA DE ARQUITECTURAS

Arquitectura	MVC	Cliente-Servidor	Orientada a eventos
<b>Enfoque</b>	Separación de preocupaciones: Modelo, Vista y Controlador	Comunicación entre dos entidades: Cliente y Servidor	Flujo de información asincrónico mediante eventos
<b>Escalabilidad</b>	Buena escalabilidad horizontal	Buena escalabilidad vertical	Buena escalabilidad horizontal y vertical
<b>Flexibilidad</b>	Alta flexibilidad para cambios en la interfaz	Flexibilidad para cambios en la lógica de negocio	Alta flexibilidad para cambios en el flujo de información
<b>Uso práctico</b>	Aplicaciones web, aplicaciones de escritorio	Aplicaciones web, aplicaciones distribuidas	Sistemas de mensajería, sistemas de streaming

Tabla 3. Tabla comparativa de arquitecturas

*Fuente. (Bass, Clements, & Kazman, 2012), (Bellemare, 2014) y (Fowler, 2006)*

Se ha seleccionado MVC para el sistema de registro de asistencia por su capacidad para separar claramente las responsabilidades del código en capas definidas, esto facilita la adaptación del sistema a nuevas necesidades, ya que permite ajustes en la interfaz de usuario sin afectar la lógica de negocio. Además, la simplicidad de MVC lo hace fácil de entender e implementar, destacando su capacidad de organización y adaptabilidad.

### 2.3.27 METODOLOGÍA DE DESARROLLO DE SOFTWARE

En el contexto de los Sistemas de Información (SI), la metodología se refiere al enfoque sistemático utilizado para planificar, diseñar, implementar y gestionar sistemas de información. (Pressman, 2014) describe una metodología de desarrollo de software como "un conjunto estándar de prácticas, técnicas y pautas para el desarrollo de software." Esto subraya la importancia de seguir un enfoque sistemático y disciplinado para planificar, implementar y evaluar el proceso de desarrollo de software.

### 2.3.28 METODOLOGÍA SCRUM

La metodología Scrum es un marco ágil que organiza el desarrollo de software en ciclos cortos e iterativos llamados sprints, que suelen durar entre dos y cuatro semanas. Scrum define roles específicos como el Scrum Master, el Product Owner y el equipo de desarrollo, y establece eventos regulares, incluyendo reuniones diarias, revisiones de sprint y retrospectivas. El objetivo principal de Scrum es entregar incrementos de producto funcionales al final de cada sprint, permitiendo una rápida adaptación a los cambios y una retroalimentación constante. Este enfoque estructurado fomenta la colaboración, la transparencia y la mejora continua, asegurando que el equipo pueda responder eficientemente a las necesidades del cliente y del mercado. (Schwaber, 2020)

### 2.3.29 METODOLOGÍA KANBAN

La metodología Kanban es un enfoque visual y flexible para la gestión de proyectos y desarrollo de software que se centra en mejorar la eficiencia del flujo de trabajo. Utiliza un tablero Kanban dividido en columnas que representan las diferentes etapas del proceso de trabajo, y tarjetas que representan tareas específicas. Una de las características clave de Kanban es la limitación del trabajo en progreso (WIP) para evitar la sobrecarga y asegurar un flujo constante de trabajo. Esto permite identificar rápidamente cuellos de botella y áreas de mejora. Kanban promueve la transparencia y la colaboración continua, facilitando ajustes inmediatos en el proceso para mejorar la entrega de valor. (Sommerville, 2015)

### 2.3.30 METODOLOGÍA PROTOTIPADO

La metodología de desarrollo de software prototipado es un enfoque iterativo y experimental para la creación de aplicaciones, en el cual se construyen prototipos del software con el propósito de explorar y refinar los requisitos y funcionalidades antes de la implementación final. Este método permite a los desarrolladores y usuarios finales interactuar con una versión inicial del software, identificando rápidamente errores, omisiones y mejoras necesarias. La principal ventaja del prototipado radica en su capacidad para proporcionar retroalimentación temprana y continua, lo que facilita ajustes y cambios en etapas tempranas del desarrollo, reduciendo el riesgo de errores costosos y asegurando una mayor satisfacción del usuario. Existen varios tipos de prototipos, incluyendo los desechables, que se crean para comprender mejor los requisitos y luego se descartan, y los evolutivos, que se mejoran continuamente hasta convertirse en el producto final. En resumen, el prototipado promueve una colaboración más estrecha entre los desarrolladores y los usuarios, mejorando la calidad y relevancia del software producido. (Pressman, 2014)

### 2.3.31 TABLA DE COMPARACIÓN METODOLOGÍAS DE DESARROLLO

<b>Característica</b>	<b>Prototipado</b>	<b>Scrum</b>	<b>Kanban</b>
<b>Enfoque</b>	Iterativo y experimental	Iterativo e incremental	Visual y continuo
<b>Objetivo Principal</b>	Refinar requisitos y funcionalidades	Entregar incrementos de producto funcionales	Mejorar la eficiencia del flujo de trabajo
<b>Duración de Ciclos</b>	No tiene ciclos fijos	Sprints de 2 a 4 semanas	Continuo
<b>Roles Definidos</b>	No roles específicos formalizados	Scrum Master, Product Owner, Equipo de Desarrollo	No roles específicos formalizados

<b>Gestión del Trabajo</b>	Creación de prototipos sucesivos	Backlog, Sprints, Tablero Scrum	Tablero Kanban, Límites de Trabajo en Progreso (WIP)
<b>Retroalimentación</b>	Continua con cada versión del prototipo	Al final de cada sprint y durante las reuniones diarias	Continua mediante la visualización del flujo de trabajo
<b>Flexibilidad</b>	Alta flexibilidad para cambios	Moderada, permite cambios entre sprints	Alta flexibilidad para cambios continuos
<b>Adecuado Para</b>	Proyectos con requisitos iterativos o en evolución constante	Proyectos que requieren entregas frecuentes y organizadas	Proyectos que necesitan optimización de procesos

Tabla 4. Tabla comparativa de metodologías de desarrollo

*Fuente. (Schwaber, 2020), (Sommerville, 2015) y (Pressman, 2014)*

Se decidió usar la metodología de desarrollo de software prototipado para el desarrollo de la tesis debido a la naturaleza iterativa y evolutiva de los requisitos del proyecto. Esta metodología permite la creación de versiones preliminares del software, facilitando la exploración y refinamiento de funcionalidades a través de iteraciones sucesivas. Al interactuar con prototipos, el director del proyecto puede proporcionar retroalimentación continua, lo cual es crucial para identificar y corregir errores, omisiones y mejorar características desde etapas tempranas del desarrollo. Esto no solo reduce el riesgo de errores en etapas avanzadas, sino que también asegura que el producto final cumpla con las expectativas del alcance final del proyecto.

## CAPÍTULO III

### METODOLOGÍA

#### 3. METODOLOGÍA DE DESARROLLO DEL PLAN DE TESIS

##### 3.1 PROTOTIPADO

La metodología del prototipado es una práctica fundamental en el desarrollo de software y diseño de productos, permitiendo validar conceptos y diseños antes de la implementación completa del producto, esta metodología es una aproximación iterativa que se centra en la creación de prototipos tempranos y funcionales del producto final para recopilar comentarios de los usuarios y mejorar el diseño del software.

El prototipado ofrece varios beneficios, como mejorar la comunicación entre desarrolladores, usuarios y partes interesadas, lo que ayuda a evitar malentendidos y garantiza una correcta aproximación al producto que se quiere realizar. Además, permite detectar errores y problemas de usabilidad en etapas tempranas del desarrollo, ahorrando tiempo y recursos monetarios a largo plazo. La naturaleza iterativa del prototipado proporciona mayor flexibilidad a los desarrolladores para adaptarse a cambios en los requisitos o condiciones del mercado.

El prototipado se desarrollará en cinco fases: planificación, diseño del prototipo inicial, desarrollo iterativo, pruebas del prototipo y refinamiento del prototipo. En la fase de planificación, se definen los objetivos y se establecen los requisitos clave. Luego, se crea una versión preliminar del producto en la etapa de diseño del prototipo inicial. A medida que se desarrolla, se somete a pruebas reales para evaluar su funcionalidad y usabilidad. Finalmente, se realizan ajustes iterativos en el prototipo para lograr un diseño óptimo.

La metodología del prototipado se desarrollará en un proceso iterativo, donde cada fase será exhaustivamente documentada en el informe del proyecto de titulación. Estas iteraciones permitirán una revisión constante del progreso del prototipo, asegurando su alineación con los objetivos del proyecto y su capacidad para satisfacer las necesidades del usuario final. Cada iteración del prototipo será cuidadosamente evaluada por el director del proyecto, quien actuará como el principal responsable de la supervisión y guía del desarrollo del proyecto. Esta evaluación

se basará en criterios predefinidos, que incluirán aspectos como la calidad del diseño, la adecuación a los requisitos del usuario y la integración efectiva de la retroalimentación obtenida en las etapas anteriores.

### 3.1.1 TEORÍA DE LA METODOLOGÍA DE PROTOTIPADO

La teoría del prototipado se basa en la premisa fundamental de la iteración y la retroalimentación continua, una filosofía que reconoce la dinámica inherente a los proyectos de desarrollo de software. Este enfoque destaca la necesidad de adaptarse ágilmente a los cambios en los requisitos y especificaciones del cliente, una habilidad crucial en un entorno tecnológico en constante evolución (Jhonson, 2020). El prototipado emerge como una herramienta indispensable en este contexto, al permitir la experimentación temprana con ideas y soluciones. Esta práctica facilita a los equipos la identificación y resolución proactiva de problemas de diseño y funcionalidad en las primeras etapas del ciclo de desarrollo. No solo eso, sino que también promueve la colaboración interdisciplinaria entre diseñadores, desarrolladores y usuarios finales, cultivando un ambiente propicio para la creación de productos más centrados en las necesidades reales del usuario (Brown, 2018). Concluyendo, el prototipado no solo es una estrategia eficiente para la validación de conceptos, sino que también funciona como un catalizador para la innovación y la mejora continua en el proceso de desarrollo de software.

### 3.1.2 ROLES Y RESPONSABILIDADES

Los roles y responsabilidades en la metodología de prototipado son fundamentales para asegurar la eficacia y el éxito del proyecto. El diseñador de prototipos es el encargado de traducir los requisitos y conceptos del proyecto en versiones preliminares del producto, ya sea mediante prototipos de baja fidelidad o prototipos de alta fidelidad. El diseñador de prototipos debe poseer habilidades sólidas en diseño de interacción y experiencia de usuario, así como una comprensión profunda de las necesidades y expectativas del usuario final (White, 2021).

Por otro lado, el desarrollador es responsable de implementar los prototipos y proporcionar retroalimentación técnica sobre su viabilidad y factibilidad técnica, dentro de los desarrolladores

pueden incluir desarrolladores de software, ingenieros de hardware, especialistas en experiencia de usuario, entre otros. Finalmente, los usuarios finales del producto desempeñan un papel crucial al proporcionar retroalimentación sobre la usabilidad y utilidad de los prototipos, contribuyendo así a la mejora continua del producto (Lowdermilk, 2020).

<b>Rol</b>	<b>Responsabilidades</b>
<b>Diseñador</b>	Crea prototipos que representan la apariencia, la funcionalidad y la experiencia del usuario del software.
<b>Desarrollador</b>	Implementa los prototipos utilizando herramientas de desarrollo de software.
<b>Usuario final</b>	Prueba y evalúa los prototipos, proporcionando retroalimentación sobre la usabilidad, la funcionalidad y la experiencia general.

Tabla 5. Tabla Roles y Responsabilidades del Prototipado

Fuente. (Lowdermilk, 2020)

### 3.1.3 ARTEFACTOS DE LA METODOLOGÍA PROTOTIPADO

La metodología del prototipado implica la generación de una variedad de artefactos destinados a visualizar y comunicar eficazmente ideas y conceptos de diseño. Los prototipos de baja fidelidad representan versiones esquemáticas y simplificadas del producto, diseñadas para explorar y validar la arquitectura y el flujo de la interfaz de usuario. Estos modelos iniciales, caracterizados por su rapidez y economía en la producción, resultan ideales para la fase inicial del proceso de diseño, ya que permiten una exploración ágil y eficiente de múltiples enfoques y soluciones.

En contraste, los prototipos de alta fidelidad se distinguen por su nivel de detalle y realismo, incorporando diseños visuales avanzados y funcionalidades interactivas más sofisticadas. Estas representaciones más acabadas del producto son especialmente útiles para evaluar aspectos específicos del diseño, como la estética y la usabilidad. Además, ofrecen la posibilidad de realizar pruebas de usuario más exhaustivas y detalladas, lo que contribuye a la identificación y resolución temprana de posibles problemas y mejoras potenciales en el producto final (Green, 2019). En

resumen, la combinación estratégica de prototipos de baja y alta fidelidad en el proceso de diseño proporciona un enfoque completo y efectivo para el desarrollo de productos centrados en el usuario y orientados a la excelencia en la experiencia del usuario (Green, 2019).

### 3.1.4 EVENTOS DEL PROTOTIPADO

El proceso de prototipado se compone de una serie de eventos y actividades estratégicas orientadas a la generación de ideas, la validación de soluciones y la obtención de retroalimentación. Las sesiones de lluvia de ideas y diseño representan encuentros colaborativos donde el equipo de desarrollo despliega su creatividad para concebir ideas y conceptos para el prototipado, empleando herramientas como mapas de empatía, mapas de recorrido y bocetos. Estas sesiones se erigen como pilares fundamentales para establecer una visión compartida del producto y para delinear claramente los objetivos y requisitos del prototipo, asegurando así una alineación colectiva en el equipo. En paralelo, las pruebas de usuario constituyen eventos cruciales en los cuales los usuarios finales interactúan con los prototipos y ofrecen sus opiniones y percepciones sobre su experiencia. Estas pruebas pueden llevarse a cabo tanto de manera presencial como remota, abarcando una variedad de actividades que incluyen entrevistas, encuestas y tareas de usabilidad. La información obtenida durante estas pruebas se revela como un recurso inestimable para identificar posibles deficiencias en el diseño y la funcionalidad, proporcionando así una guía valiosa para el proceso continuo de iteración y mejora del producto (Red, 2022).

### 3.1.5 CICLO ITERATIVO DEL PROTOTIPADO

El proceso de prototipado sigue un ciclo iterativo compuesto por diversas etapas que abarcan desde la planificación inicial hasta la entrega final del producto. En la etapa inicial de planificación, se delimitan con precisión los objetivos, el alcance y los requisitos del prototipo, sentando así las bases fundamentales para el desarrollo subsiguiente. Posteriormente, durante la fase de diseño y desarrollo, se da vida al prototipo, aprovechando herramientas y técnicas especializadas en el diseño centrado en el usuario para garantizar su idoneidad y utilidad. Una vez que el prototipo toma forma, se inicia una fase crucial de pruebas y evaluación, donde se recolecta la valiosa retroalimentación de los usuarios y se identifican áreas susceptibles de mejora.

Esta retroalimentación actúa como el motor que impulsa el proceso de iteración y refinamiento en la siguiente etapa del ciclo. Es aquí donde se materializa el verdadero potencial del prototipado, ya que cada iteración busca perfeccionar y enriquecer la experiencia del usuario, adaptándola de manera continua a sus necesidades y expectativas.

Este ciclo de iteración y mejora se repite diligentemente hasta alcanzar un nivel óptimo de validez y usabilidad del prototipo, momento en el cual se procede a su validación y entrega al cliente o usuario final. Sin embargo, este proceso no concluye con la entrega, ya que la retroalimentación continua y la adaptación a las necesidades cambiantes del mercado garantizan que el producto evolucione de manera constante para mantener su relevancia y utilidad a lo largo del tiempo (Snyder, 2003).

## 3.2 REQUERIMIENTOS

### 3.2.1 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales, abarcan un conjunto de funciones esenciales que el sistema debe cumplir. Estas funciones engloban operaciones específicas, servicios del sistema, comportamientos que se esperan del mismo, así como cualquier interacción prevista con los usuarios. Son, en esencia, la principal guía que define la utilidad y el propósito del sistema en cuestión (Sommerville, 2015).

Dentro del desarrollo de software los requerimientos funcionales poseen un rol dinámico y vital ya que los requisitos no solo orientan, sino que también moldean todas las etapas del ciclo de vida del software. Desde la concepción inicial hasta la implementación y mantenimiento continuo, los requerimientos funcionales sirven como un faro constante para el equipo de desarrollo, proporcionando una dirección clara y coherente.

Es crucial destacar que los requerimientos funcionales no son meramente directrices estáticas, sino que actúan como un marco fundamental sobre el cual se construye todo el entramado del software. Siguiendo la analogía de la construcción estos requisitos funcionales son los cimientos sobre los cuales se erigen todas las características y funcionalidades del sistema. Constituyen, por tanto, el pilar sobre el cual descansa la arquitectura del software, definiendo su estructura y finalidad última. (Pressman, 2014)

	<b>Descripción</b>	<b>Prioridad</b>	<b>Comentario</b>
<b>RF01</b>	El sistema debe proporcionar una funcionalidad segura y eficiente para el acceso de los usuarios al sistema web, garantizando que solo usuarios autorizados puedan iniciar sesión.	Alta	La asistencia se podrá registrar con la cédula del empleado.
<b>RF02</b>	El sistema debe permitir la gestión de los usuarios, incluyendo la capacidad de agregar, editar, eliminar y consultar información detallada de cada empleado.	Alta	El sistema contará con notificaciones al usuario indicando si la acción ejecutada se realizó correctamente.
<b>RF03</b>	El sistema debe permitir la gestión de los cargos de los empleados, incluyendo la capacidad de agregar, editar, eliminar y consultar información detallada de cada cargo.	Alta	El sistema contará con notificaciones al usuario indicando si la acción ejecutada se realizó correctamente.
<b>RF04</b>	El sistema debe permitir la gestión de los horarios de los empleados, incluyendo la capacidad de agregar, editar, eliminar y consultar información detallada de cada horario.	Alta	El sistema contará con notificaciones al usuario indicando si la acción ejecutada se realizó correctamente.
<b>RF05</b>	El sistema debe permitir la gestión de los empleados, incluyendo la capacidad de agregar, editar, eliminar y consultar información detallada de cada empleado.	Alta	El sistema contará con notificaciones al usuario indicando si la acción ejecutada se realizó correctamente.
<b>RF06</b>	El sistema debe permitir la consulta y revisión de las asistencias de los empleados, incluyendo la capacidad de exportación de datos y búsqueda por filtros.	Alta	Se implementarán filtros por fechas y empleados para una búsqueda más organizada.
<b>RF07</b>	El sistema debe proporcionar una gestión de justificaciones a las asistencias de los empleados que no cumplan con la cantidad de horas trabajadas establecidas en su horario.	Alta	El sistema contará con notificaciones al usuario indicando si la acción ejecutada se realizó correctamente.
<b>RF08</b>	El sistema debe permitir a los empleados registrar su asistencia mediante la entrada de su número de identificación, almacenando la hora de entrada y salida en una base de datos.	Alta	La asistencia se podrá registrar con la cédula del empleado

Tabla 6. Tabla Requerimientos Funcionales

*Elaboración propia.*

### 3.2.2 REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales desempeñan un papel crucial al ampliar y complementar los requerimientos funcionales. No se limitan a definir funciones específicas del sistema, sino que también abordan atributos de calidad, restricciones y otros criterios que influyen en cómo se ejecutan y perciben estas funciones. Esencialmente, mientras los requerimientos funcionales delimitan lo que el sistema debe hacer, los no funcionales definen cómo debe hacerlo, y de esta manera, desempeñan un papel igualmente significativo en la conformación de la experiencia del usuario y la calidad del software.

En el contexto de este análisis, es fundamental examinar a fondo la importancia y el impacto de los requerimientos no funcionales en el desarrollo de software y en la satisfacción del usuario. Dentro de lo destacado de los requerimientos no funcionales es como delinea las cualidades esenciales del sistema y los criterios mediante los cuales se evalúa su desempeño. Desde la usabilidad hasta la seguridad, pasando por la eficiencia y la confiabilidad, los requerimientos no funcionales abarcan una amplia gama de atributos que, en última instancia, moldean la experiencia del usuario y determinan la percepción general de la calidad del software (Sommerville, *Software engineering* (9th ed.), 2011).

Los requerimientos funcionales para este sistema web son los siguientes:

- El tiempo de carga de las páginas para el registro de asistencia debe ser de menos de 2 segundos.
- Las contraseñas de los usuarios deben estar protegidas mediante un mecanismo de hash utilizando el algoritmo MD5.
- El sistema debe ser capaz de manejar hasta 500 usuarios simultáneamente sin degradar el rendimiento.
- Los flujos de trabajo para el registro de asistencia, la gestión de empleados y la administración de cargos y horarios deben completarse en menos de 5 pasos.
- El código del sistema web debe estar bien documentado y organizado para facilitar su comprensión y mantenimiento.
- El sistema web debe ser compatible con los últimos navegadores web, incluyendo Chrome, Firefox, Edge y Safari.

### 3.2.3 CASOS DE USO

Los casos de uso constituyen representaciones detalladas de las interacciones entre los actores, ya sean usuarios o sistemas externos, y un sistema dado con el propósito de alcanzar un objetivo particular. Cada caso de uso es una narrativa que describe cómo se lleva a cabo una tarea específica dentro del sistema, mostrando los pasos que un usuario o actor sigue para completar dicha tarea, manteniendo una secuencia precisa de eventos que se desencadenan en respuesta a las acciones del usuario o actor, detallando así cómo se comporta el sistema en diferentes situaciones. El enfoque principal de los casos de uso está centrado en el usuario, lo que facilita la comprensión de los requerimientos del sistema y su posterior implementación. Al centrarse en las acciones y objetivos del usuario, los casos de uso ayudan a definir con claridad las funcionalidades necesarias del sistema, así como los diferentes caminos que pueden tomar los usuarios para lograr sus objetivos. Esta comprensión detallada de las interacciones usuario-sistema es fundamental para diseñar interfaces intuitivas y experiencias de usuario satisfactorias (Cockburn, 2001).

Es importante resaltar sus beneficios en la comunicación, ya que, al presentar escenarios específicos de uso del sistema, los casos de uso permiten a los desarrolladores, diseñadores y partes interesadas comprender mejor el comportamiento esperado del sistema y detectar posibles problemas o áreas de mejora desde una etapa temprana del proceso de desarrollo de software.

#### 3.2.3.1 COMPONENTES DE UN CASO DE USO

<b>Componente</b>	<b>Descripción</b>
<b>Nombre</b>	Título descriptivo que refleja la acción del usuario.
<b>Actores</b>	Entidades que interactúan con el sistema (usuarios, roles, sistemas externos).
<b>Descripción</b>	Resumen breve del objetivo del caso de uso.
<b>Flujo principal</b>	Pasos secuenciales que el actor y el sistema realizan para lograr el objetivo.

<b>Flujos alternativos</b>	Secuencias de eventos que reemplazan al flujo principal en situaciones excepcionales.
<b>Precondiciones</b>	Condiciones que deben cumplirse antes de iniciar el caso de uso.
<b>Postcondiciones</b>	Condiciones que se cumplen tras la ejecución exitosa del caso de uso.

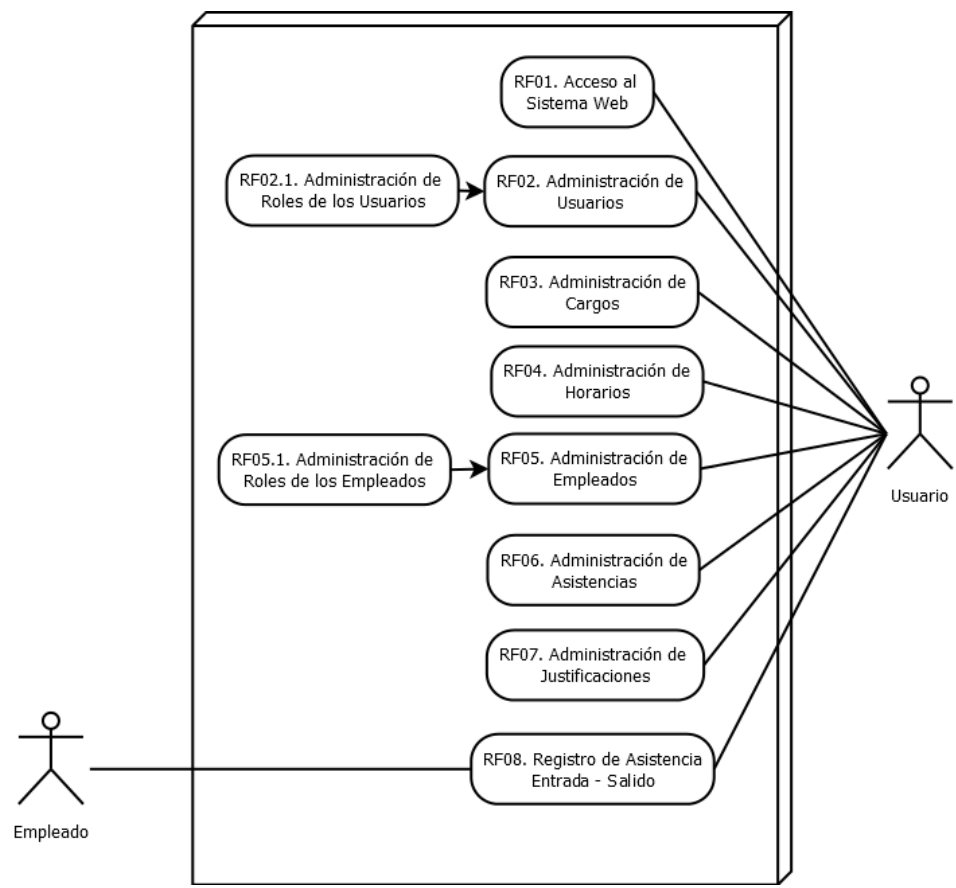
Tabla 7. Tabla componentes de un caso de uso

Fuente. (Cockburn, 2001)

### 3.2.3.2 DIAGRAMA DE CASO DE USO NIVEL GENERAL

Figura 1

Caso de uso nivel general.

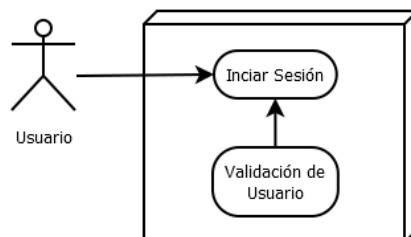


Nota. Elaboración propia.

### 3.2.3.3 RF01 ACCESO AL SISTEMA WEB

Figura 2

Acceso al sistema web.



Nota. Elaboración propia.

**Descripción:** Permite el ingreso al sistema web mediante el inicio de sesión para los usuarios administradores.

**Actores:** Usuario

**Flujo Principal:**

1. Los usuarios abrirán el sistema web en su dispositivo.
2. La pantalla de inicio mostrará las ventana de inicio de sesión.
3. Aparecerá un formulario solicitando el nombre de usuario y la contraseña.
4. El usuario ingresará su nombre de usuario y contraseña en los campos correspondientes.
5. El usuario hará clic en el botón "Iniciar Sesión" para proceder.
6. El sistema verificará las credenciales ingresadas contra la base de datos.

**Flujo Alternativo:**

7. Si las credenciales son incorrectas:

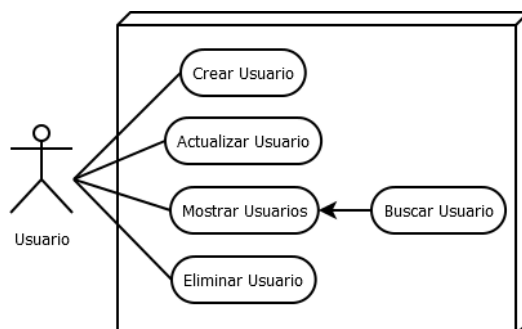
El sistema mostrará una notificación de error.

7. Si las credenciales son correctas:
8. El sistema almacenará la información de la sesión del usuario de forma segura.
9. Una vez que la validación sea correcta, el sistema redirigirá al usuario a la pantalla principal de la aplicación.

### 3.2.3.4 RF02 ADMINISTRACIÓN DE USUARIOS

Figura 4

Administración de usuarios



Nota. Elaboración propia.

**Descripción:** Permite la que el usuario pueda crear, actualizar, mostrar, eliminar y buscar un usuario que este en el sistema web.

**Actores:** Usuario

**Flujo Principal:**

1. El actor ingresará a la ventana de usuarios en el sistema.
2. El sistema muestra la información de los usuarios existentes en el sistema.
3. El actor dará clic en el botón “Registrar”.
4. El sistema muestra la ventana de registro de usuarios.
5. El actor llena la información necesaria del usuario para que pueda ser creado.

**Flujo Alternativo:**

5. Si la información no es del formato correcto:

El sistema mostrará una notificación de error.

6. El actor podrá modificar la información del usuario para que pueda ser actualizado.

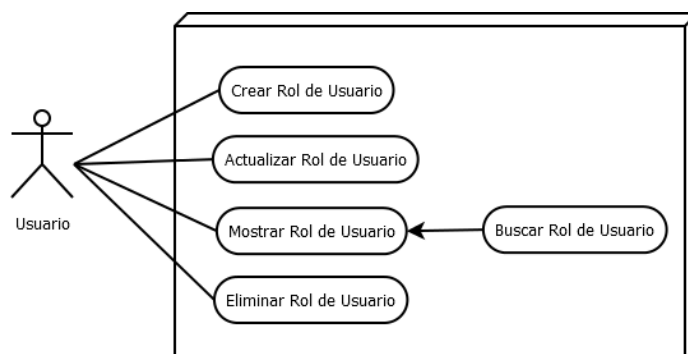
**Precondiciones:** El usuario debe estar registrado en el sistema.

7. El actor podrá visualizar a los usuarios ya creados en el sistema.
8. El actor podrá buscar a un específico grupo de usuarios en el sistema en base a sus características.
9. El actor podrá eliminar usuarios ya creados en el sistema.

### 3.2.3.5 RF02.1 ADMINISTRACIÓN DE ROLES DE USUARIO

Figura 3

Administración de roles de usuarios



Nota. Elaboración propia.

**Descripción:** Permite la que el usuario pueda crear, actualizar, mostrar, eliminar y buscar un rol de usuario que este en el sistema web.

**Actores:** Usuario

**Flujo Principal:**

1. El actor ingresará a la ventana de usuarios en el sistema.
2. El sistema muestra la información de los usuarios existentes en el sistema.
3. El actor dará clic en el botón “Roles”.
4. El sistema muestra la ventana de roles de usuarios.
5. El actor dará clic en el botón “Registrar”.
6. El sistema muestra la ventana de registro de roles de usuarios.
7. El actor llena la información necesaria del rol del usuario para que pueda ser creado.

**Flujo Alternativo:**

7. Si la información no es del formato correcto:

El sistema mostrará una notificación de error.

8. El actor podrá modificar la información del rol del usuario para que pueda ser actualizado.

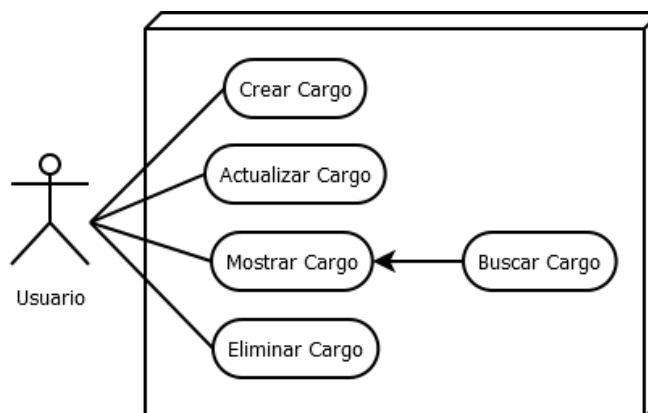
**Precondiciones:** El rol de usuario debe estar registrado en el sistema.

9. El actor podrá visualizar a los roles de usuarios ya creados en el sistema.
10. El actor podrá buscar a un especifico grupo de roles de usuarios en el sistema en base a sus características.
11. El actor podrá eliminar roles de usuarios ya creados en el sistema.

### 3.2.3.6 RF03 ADMINISTRACIÓN DE CARGOS DEL EMPLEADO

Figura 5

Administración de cargos



Nota. Elaboración propia.

**Descripción:** Permite la que el usuario pueda crear, actualizar, mostrar, eliminar y buscar un cargo que este en el sistema web.

**Actores:** Usuario

**Flujo Principal:**

1. El actor ingresará a la ventana de cargos en el sistema.
2. El sistema muestra la información de los cargos existentes en el sistema.
3. El actor dará clic en el botón “Registrar”
4. El sistema muestra la ventana de registro de cargos
5. El actor llena la información necesaria del cargo para que pueda ser creado.

**Flujo Alterno:**

5. Si la información no es del formato correcto:

El sistema mostrará una notificación de error.

6. El actor podrá modificar la información del cargo para que pueda ser actualizado.

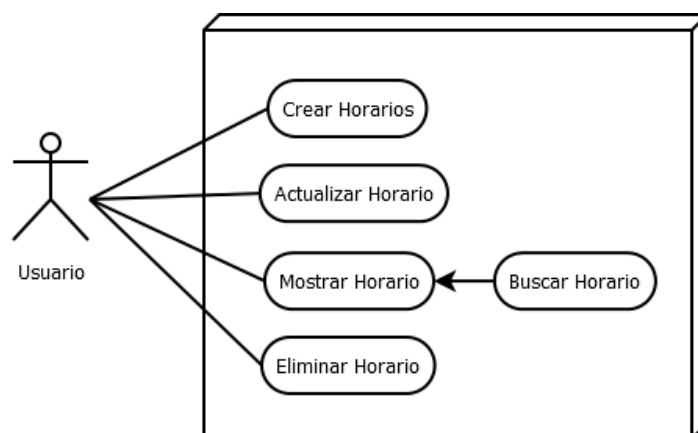
**Precondiciones:** El cargo debe estar registrado en el sistema.

7. El actor podrá visualizar a los cargos ya creados en el sistema.
8. El actor podrá buscar a un específico grupo de cargos en el sistema en base a sus características.
9. El actor podrá eliminar cargos ya creados en el sistema.

### 3.2.3.7 RF04 ADMINISTRACIÓN DE HORARIOS

Figura 6

Administración de horarios



Nota. Elaboración propia.

**Descripción:** Permite la que el usuario pueda crear, actualizar, mostrar, eliminar y buscar un horario que este en el sistema web.

**Actores:** Usuario

**Flujo Principal:**

1. El actor ingresará a la ventana de horario en el sistema.
2. El sistema muestra la información de los horarios existentes en el sistema.
3. El actor dará clic en el botón “Registrar”
4. El sistema muestra la ventana de registro de horarios
5. El actor llena la información necesaria del horario para que pueda ser creado.

**Flujo Alternativo:**

5. Si la información no es del formato correcto:

El sistema mostrará una notificación de error.

6. El actor podrá modificar la información del horario para que pueda ser actualizado.

**Precondiciones:** El horario debe estar registrado en el sistema.

7. El actor podrá visualizar a los horarios ya creados en el sistema.

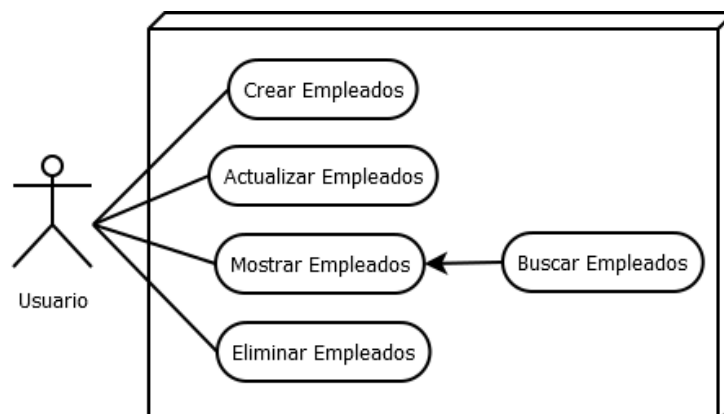
8. El actor podrá buscar a un específico grupo de horarios en el sistema en base a sus características.

9. El actor podrá eliminar horarios ya creados en el sistema.

### 3.2.3.8 RF05 ADMINISTRACIÓN DE EMPLEADOS

Figura 8

Administración de empleados



Nota. Elaboración propia.

**Descripción:** Permite la que el usuario pueda crear, actualizar, mostrar, eliminar y buscar un empleado que este en el sistema web.

**Actores:** Usuario

**Flujo Principal:**

1. El actor ingresará a la ventana de empleados en el sistema.
2. El sistema muestra la información de los empleados existentes en el sistema.
3. El actor dará clic en el botón “Registrar”

4. El sistema muestra la ventana de registro de empleados
5. El actor llena la información necesaria del cargo para que pueda ser creado.

**Flujo Alternativo:**

5. Si la información no es del formato correcto:

El sistema mostrará una notificación de error.

6. El actor selecciona el cargo, rol y horario del empleado de los creados previamente en el sistema y su base de datos.
7. El sistema cuenta con una gestión interna para los roles de empleados
8. El actor podrá modificar la información del cargo para que pueda ser actualizado.

**Precondición:** El empleado debe estar registrado en el sistema.

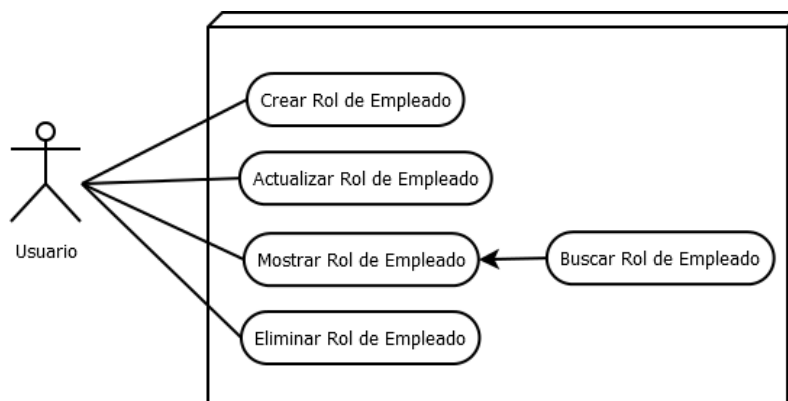
9. El actor podrá visualizar a los empleados ya creados en el sistema.
10. El actor podrá buscar a un específico grupo de empleados en el sistema en base a sus características.
11. El actor podrá eliminar empleados ya creados en el sistema.

**Precondición:** Los roles, cargos y horarios de los empleados deben estar registrados previamente en el sistema.

### 3.2.3.9 RF05.1 ADMINISTRACIÓN DE ROLES DE EMPLEADOS

Figura 7

Administración de roles de usuarios



Nota. Elaboración propia.

**Descripción:** Permite la que el usuario pueda crear, actualizar, mostrar, eliminar y buscar un rol de empleado que este en el sistema web.

**Actores:** Usuario

**Flujo Principal:**

1. El actor ingresará a la ventana de empleados en el sistema.
2. El sistema muestra la información de los empleados existentes en el sistema.
3. El actor dará clic en el botón “Roles”.
4. El sistema muestra la ventana de roles de empleados.
5. El actor dará clic en el botón “Registrar”.
6. El sistema muestra la ventana de registro de roles de empleados.
7. El actor llena la información necesaria del rol del empleado para que pueda ser creado.

**Flujo Alternativo:**

7. Si la información no es del formato correcto:

El sistema mostrará una notificación de error.

8. El actor podrá modificar la información del rol del empleado para que pueda ser actualizado.

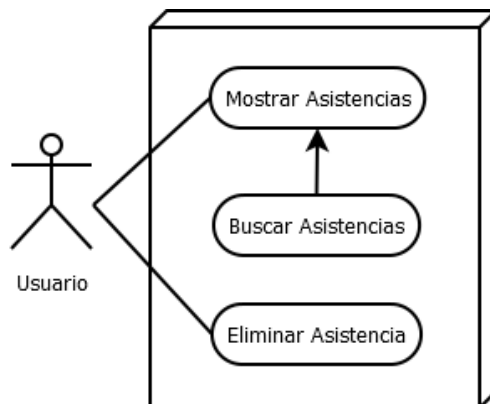
**Precondiciones:** El rol del empleado debe estar registrado en el sistema.

9. El actor podrá visualizar a los roles de empleados ya creados en el sistema.
10. El actor podrá buscar a un específico grupo de roles de empleados en el sistema en base a sus características.
11. El actor podrá eliminar roles de empleados ya creados en el sistema.

### 3.2.3.10 RF06 ADMINISTRACIÓN DE ASISTENCIAS

Figura 9

Administración de asistencias



Nota. Elaboración propia.

**Descripción:** Permite que el usuario pueda mostrar, eliminar y buscar una asistencia que este en el sistema web.

**Actores:** Usuario

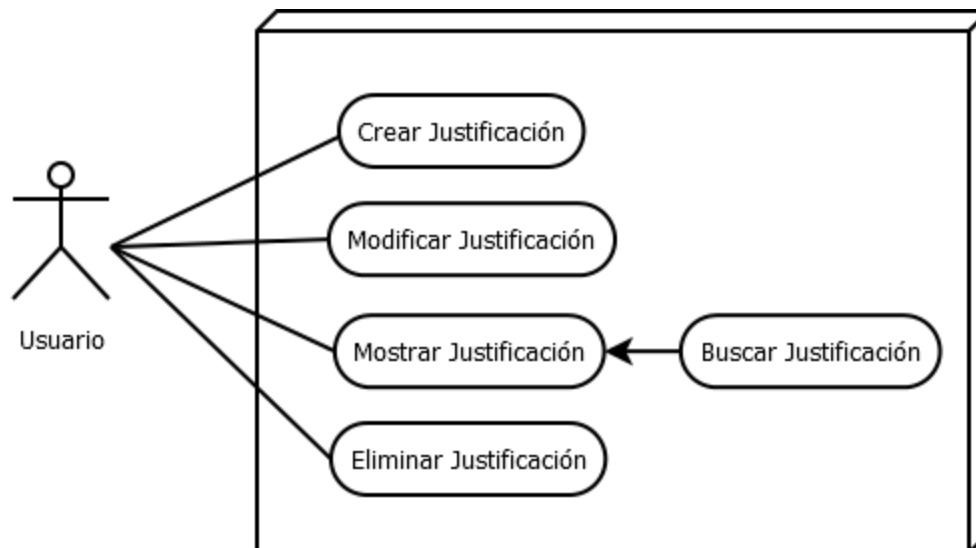
**Flujo Principal:**

1. El actor ingresará a la ventana de asistencias en el sistema.
2. El sistema muestra la información de las asistencias existentes en el sistema.
3. El actor podrá visualizar las horas que debería trabajar el empleado según su horario.
4. El actor podrá visualizar las horas trabajadas del empleado según su entrada y salida.
5. El sistema dará un indicante que muestre si el empleado cumplió sus horas por trabajar.
6. El sistema podrá generar una justificación de la asistencia de ser necesario.
7. El actor podrá buscar a un específico grupo de asistencia en el sistema en base a sus características.
8. El actor podrá eliminar asistencias ya creadas en el sistema.

### 3.2.3.11 RF07 ADMINISTRACIÓN DE JUSTIFICACIONES

Figura 10

Administración de justificaciones



Nota. Elaboración propia.

**Descripción:** Permite la que el usuario pueda generar, actualizar, mostrar, eliminar y buscar un horario que este en el sistema web.

**Actores:** Usuario

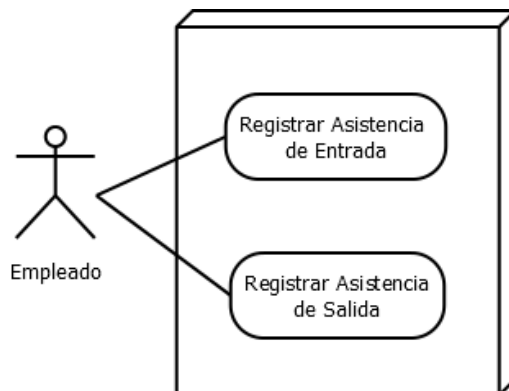
**Flujo Principal:**

1. El actor generará una justificación en la ventana de asistencias en el sistema.
2. El sistema muestra la notificación de que la justificación se ha creado.
3. El actor dará clic en el botón “Justificaciones”.
4. El sistema muestra la ventana de justificaciones.
5. El actor da clic en el botón de modificación.
6. El sistema muestra la ventana de modificación de justificación.
7. El actor modificará la información de la justificación según el caso.
8. El actor guardará los cambios dando clic en el botón “Modificar”.
9. El actor podrá visualizar a las justificaciones ya creadas en el sistema.
10. El actor podrá buscar a un específico grupo de justificaciones en el sistema en base a sus características.
11. El actor podrá eliminar justificaciones ya creadas en el sistema.

### 3.2.3.12 RF08 REGISTRO DE ASISTENCIA ENTRADA – SALIDA

Figura 11

Registro de asistencia Entrada - Salida



Nota. Elaboración propia.

**Descripción:** Permite que el empleado pueda registrar su asistencia mediante el botón de entrada y de salida en el sistema web.

**Actores:** Empleado

**Flujo Principal:**

1. El empleado ingresa su número de identificación.
2. El empleado da “clic” en el botón “Entrada”.
3. El sistema registra la asistencia en la hora exacta en que se realiza la entrada.
4. El sistema muestra una notificación de que la entrada se ha registrado correctamente.
5. El empleado da “clic” en el botón “Salida”.
6. El sistema registra la asistencia en la hora exacta en que se realiza la salida.
7. El sistema muestra una notificación de que la salida se ha registrado correctamente.

**Precondiciones:** El empleado debe estar registrado en el sistema.

El empleado no puede agregar una salida antes que la entrada.

**Postcondiciones:** La hora de entrada/salida queda registrada en la base de datos.

### 3.2.4 PLANIFICACIÓN DE LAS ITERACIONES DE PROTOTIPADO

Para el correcto desarrollo del proyecto de titulación con la creación del prototipo de sistema web, el proceso será conducido por una sola persona. Por lo tanto, acorde a los roles de la metodología de desarrollo de software de prototipado, tanto el rol de diseñador como desarrollador serán asumidos por mi persona, Marcel Bohórquez. Mientras que el rol de usuario final será asumido por el tutor designado, ya que al mismo se deberá presentar constantemente los avances obtenidos tras cada iteración y este proporcionará constantemente una retroalimentación con la finalidad de potenciar al prototipo.

Una vez teniendo tanto los roles establecidos como los requisitos definidos, el desarrollo del sistema web se estructuró el proceso en cuatro iteraciones de prototipado, iteraciones definidas a continuación:

### **1. Iteración 1 - Desarrollo de la Gestión de Usuarios e Inicio de Sesión:**

Durante esta primera iteración de desarrollo, el enfoque principal se centra en la creación de la infraestructura básica del sistema, incluyendo la autenticación de usuarios y la gestión de sus roles y permisos. Se implementa un registro seguro para permitir el acceso seguro al sistema. Además, se diseñan las interfaces de usuario iniciales para el inicio de sesión.

### **2. Iteración 2 - Desarrollo de la Gestión de Cargos y Horarios:**

En esta iteración, se aborda la gestión de cargos y horarios de los empleados dentro del sistema. Se desarrolla la funcionalidad para crear, modificar y eliminar los cargos y horarios del empleado de manera eficiente. Esto incluye la creación de la vista para su gestión, así como la integración de estas funcionalidades con el sistema de usuarios establecido en la iteración anterior. La prioridad es garantizar la coherencia y la usabilidad en la gestión de cargos.

### **3. Iteración 3 - Desarrollo de la Gestión de Empleados y Roles de Empleados:**

La tercera iteración se centra en la gestión de los empleados y sus roles dentro del sistema. Se implementa la funcionalidad para agregar, editar y eliminar empleados y sus roles, así como la asignación de cargos y horarios, y gestionar su información personal. Se diseñan interfaces intuitivas para la gestión eficiente de los datos de los empleados y sus roles, permitiendo una fácil navegación y manipulación de la información.

#### **4. Iteración 4 - Desarrollo de la Gestión de Asistencias y Justificaciones:**

En esta última fase del desarrollo, se implementa la funcionalidad principal del sistema, la gestión de asistencias de los empleados y las justificaciones en caso de incumplimiento de las horas asignadas. Se desarrollan los mecanismos necesarios para registrar las entradas y salidas de los empleados, además de funcionalidades completas para la gestión de las justificaciones. Se enfoca en la robustez y la precisión del sistema para garantizar un registro preciso y confiable de las asistencias.

Cada ciclo de prototipado se extenderá a lo largo de tres semanas, lo que facilitará un avance constante y eficiente. Este enfoque ágil permitirá adaptaciones y mejoras continuas a lo largo del proceso de desarrollo. Como resultado, se obtendrá el prototipo de un sistema web para el registro y control de la asistencia de los empleados en entornos empresariales.

## CAPÍTULO IV

### DESARROLLO

El presente capítulo detalla el desarrollo del sistema web de registro de asistencias, basado en la metodología de prototipado mencionada anteriormente. Se describen las iteraciones acordadas, cada una con sus respectivas fases de diseño, implementación y evaluación. Este enfoque se asegurará que los requisitos se cumplan de manera efectiva y eficiente. A través de este capítulo se proporciona una visión clara del proceso de desarrollo y de cómo se alcanzaron los objetivos del proyecto.

#### 4. DESARROLLO DEL SISTEMA WEB

##### 4.1 ITERACIÓN 1 - DESARROLLO DE LA GESTIÓN DE USUARIOS E INICIO DE SESIÓN

La primera iteración del proyecto se centró en desarrollar las funcionalidades de gestión de usuarios, sus roles e inicio de sesión para el sistema web de registro de asistencias. Se comenzó con un análisis exhaustivo de los requisitos funcionales y no funcionales relacionados con el manejo de usuarios, incluyendo el registro, la gestión de usuarios y la autenticación.

Con base en este análisis, se diseñaron las interfaces de usuario necesarias, tales como formularios de registro, pantallas de inicio de sesión, notificaciones para el usuario y vistas para la administración de usuarios. Para obtener retroalimentación temprana, se crearon prototipos de estas interfaces y se presentaron al tutor designado.

Posteriormente, se implementaron las funcionalidades básicas del sistema, como la creación de roles de usuario, el almacenamiento seguro de contraseñas mediante la técnica de cifrado MD5 y el establecimiento de sesiones de usuario. Para garantizar el correcto funcionamiento del sistema, se realizaron pruebas exhaustivas durante el desarrollo.

Al finalizar la primera iteración, se llevó a cabo una revisión completa del sistema con el tutor designado. Esta revisión permitió identificar posibles mejoras y establecer las bases para las siguientes iteraciones del desarrollo.

## 4.1.1 TABLA DE ACTIVIDADES ITERACIÓN 1

<b>Funcionalidad</b>	<b>Descripción</b>	<b>Actor</b>	<b>Criterio de Aceptación</b>
<b>Diseño de Interfaz para el Inicio de Sesión y Ventana de Usuarios</b>	El usuario contará con una interfaz intuitiva para ingresar su usuario y contraseña y tener acceso al sistema, además de la ventana para la gestión de usuarios.	Usuario	Interfaces intuitivas con validaciones y notificaciones para el correcto ingreso de sesión.
<b>Visualización y modificación de Usuarios y sus roles.</b>	El usuario podrá visualizar todos los usuarios y sus roles creados y editar los campos correspondientes.	Usuario	Visualizar los usuarios dentro de su vista y editar al correctamente.
<b>Creación y eliminación de Usuarios y sus roles.</b>	El usuario tiene la potestad de ingresar cuantos usuarios y roles se consideren necesarios y eliminarlos según el caso.	Usuario	Verificar los usuarios que han sido añadidos y eliminados.

Tabla 8. Tabla actividades Iteración 1

*Elaboración propia.*

## 4.1.2 CAPTURAS DE PANTALLA ITERACIÓN 1

### 4.1.2.1 INICIO DE SESIÓN

Figura 12

Pantalla Inicio Sesión

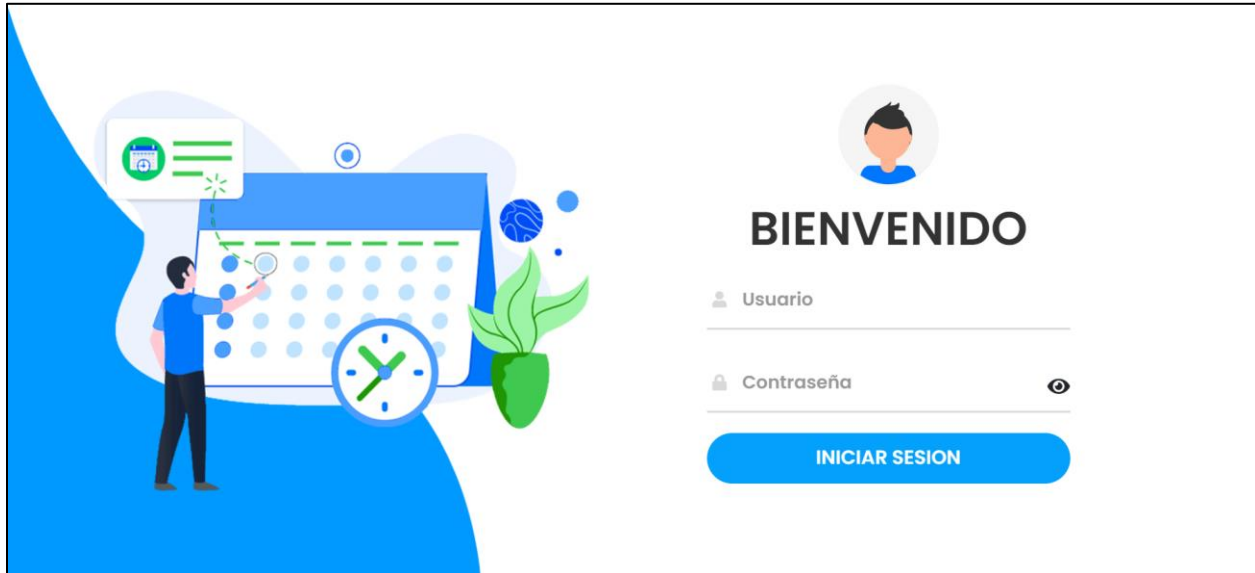


Figura 13

Pantalla Inicio Sesión ingresando usuario y contraseña

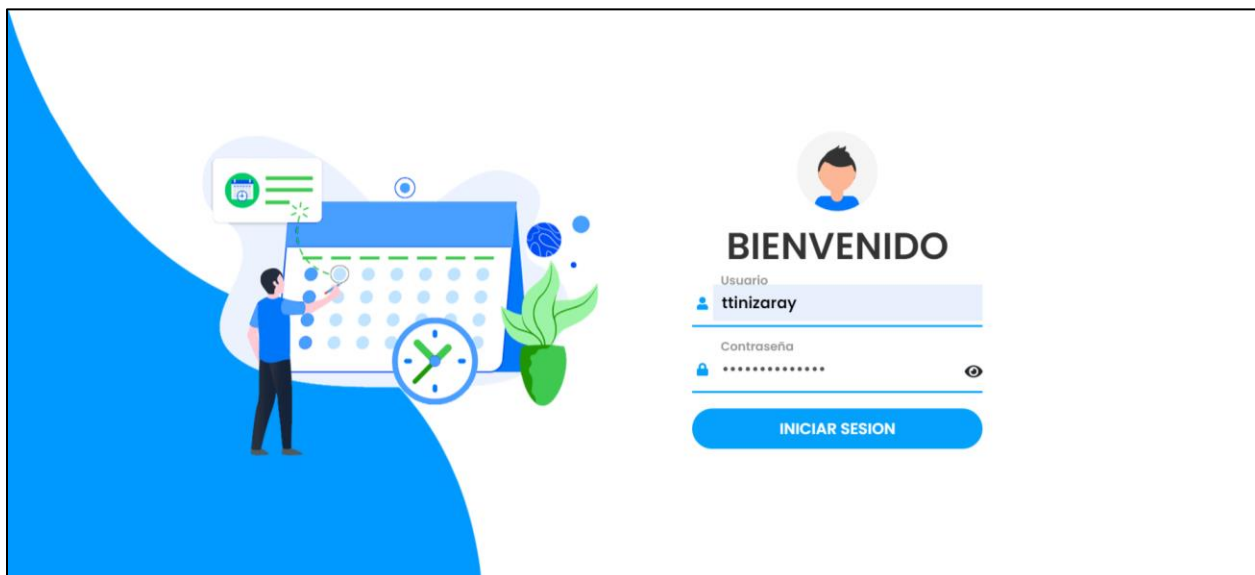


Figura 14

Pantalla Inicio Sesión mostrando contraseña

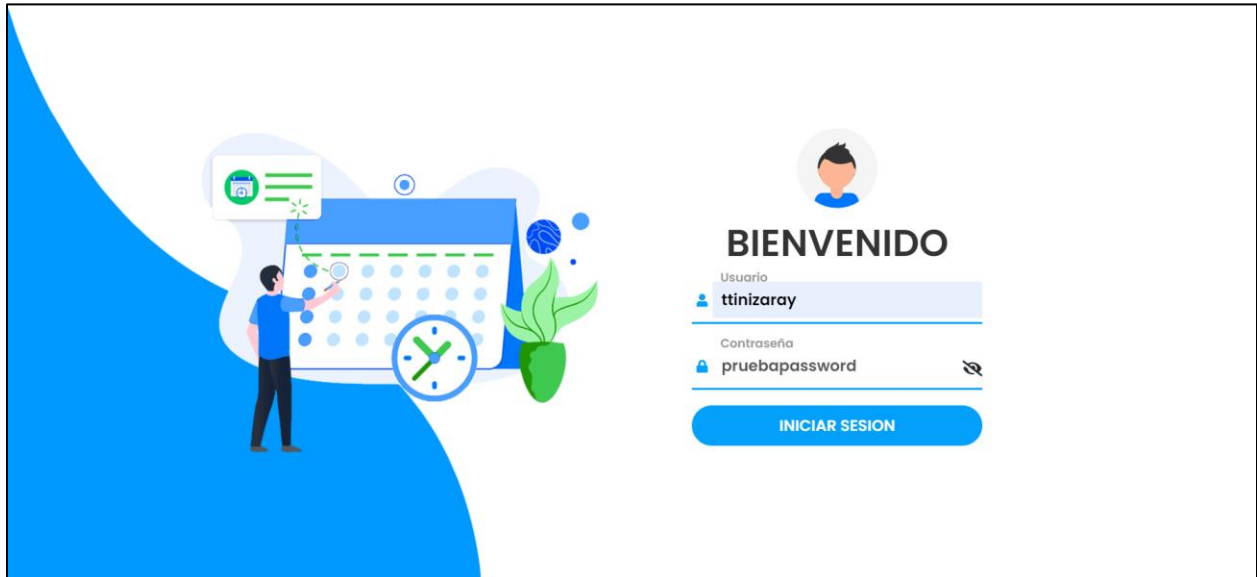


Figura 15

Pantalla Inicio Sesión, caso usuario o contraseña incorrectas

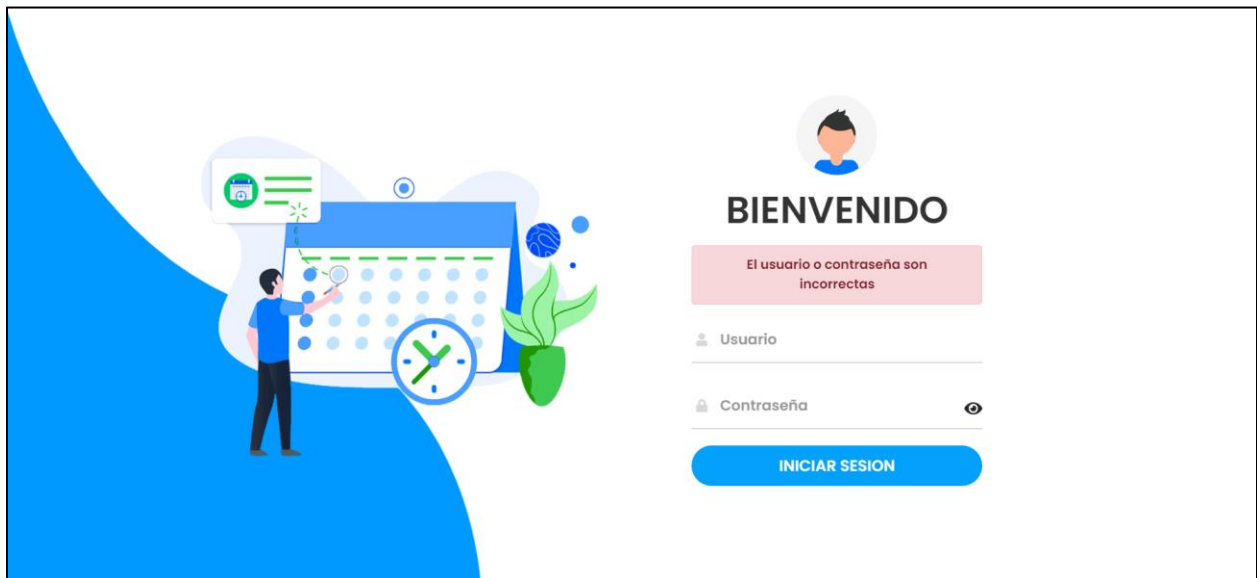
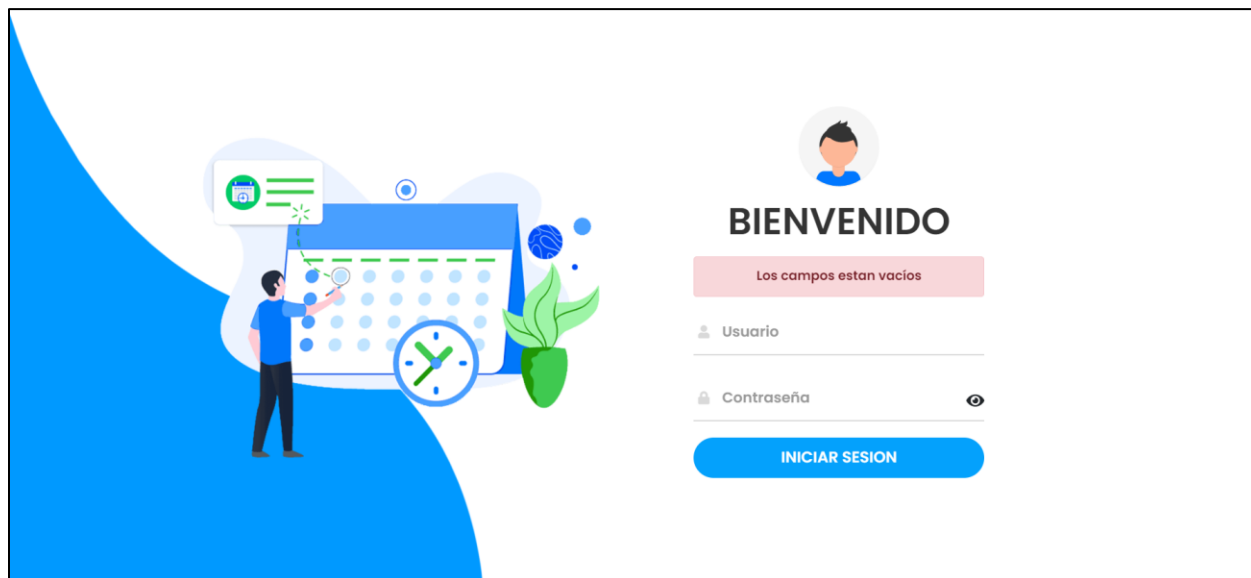


Figura 16

Pantalla Inicio Sesión, caso campos vacíos



## 4.1.2.2 GESTIÓN DE USUARIOS

Figura 17

Pantalla Lista de Usuarios

The image shows a dashboard for user management. The top navigation bar includes a menu icon, the user role 'Administrador', and a profile icon. The main content area is titled 'LISTA DE USUARIOS' and features a '+ Registrar' button. A search bar is labeled 'Buscar:'. A table displays a list of users with columns for ID, NOMBRE, USUARIO, and ROL. A 'ROLES' button is located to the right of the table. At the bottom, there is a pagination control showing 'Registros del 1 al 5 de 5 registros' and a 'Mostrar iconos ocultos' button.

ID	NOMBRE	USUARIO	ROL
1	Ana García	agarcia2003	Operativo
8	Luis Torres	ltorres2001	Operativo
9	William Sarmiento	wsarmiento2004	Administrador
14	Rodrigo Ramallo	rramallo2001	Administrador
17	Tomás Tinizaray	ttinizaray2003	Administrador

Figura 18

## Pantalla Botones de acción

The screenshot displays the 'LISTA DE USUARIOS' (User List) interface. On the left, a sidebar contains navigation options: ASISTENCIA, USUARIOS (selected), EMPLEADOS, CARGOS, and HORARIOS. The main area features a '+ Registrar' button, a search bar, and a 'ROLES' button. A table lists users with columns for ID, NOMBRE, USUARIO, and ROL. The first user, Ana García (ID 1), is highlighted with a red circle. Below the table, there are pagination controls showing 'Registros del 1 al 5 de 5 registros' and buttons for 'Anterior', '1', and 'Siguiete'.

ID	NOMBRE	USUARIO	ROL
1	Ana García	agarcia2003	Operativo
8	Luis Torres	ltorres2001	Operativo
9	William Sarmiento	wsarmiento2004	Administrador
14	Rodrigo Ramallo	rramallo2001	Administrador
17	Tomás Tinizaray	ttinizaray2003	Administrador

Figura 19

Pantalla Modificación de usuario

The screenshot shows the 'Modificar Usuario' (Edit User) modal form overlaid on the user list. The form contains the following fields: 'Nombre' (Ana), 'Apellido' (García), 'Usuario' (agarcia2003), and a dropdown menu for 'Rol'. The dropdown menu is open, showing options: Operativo, Administrador, Operativo, and Gerencial. The background shows the same user list as in Figure 18, with the first user highlighted.

Figura 19  
Notificación modificación de usuario correcta

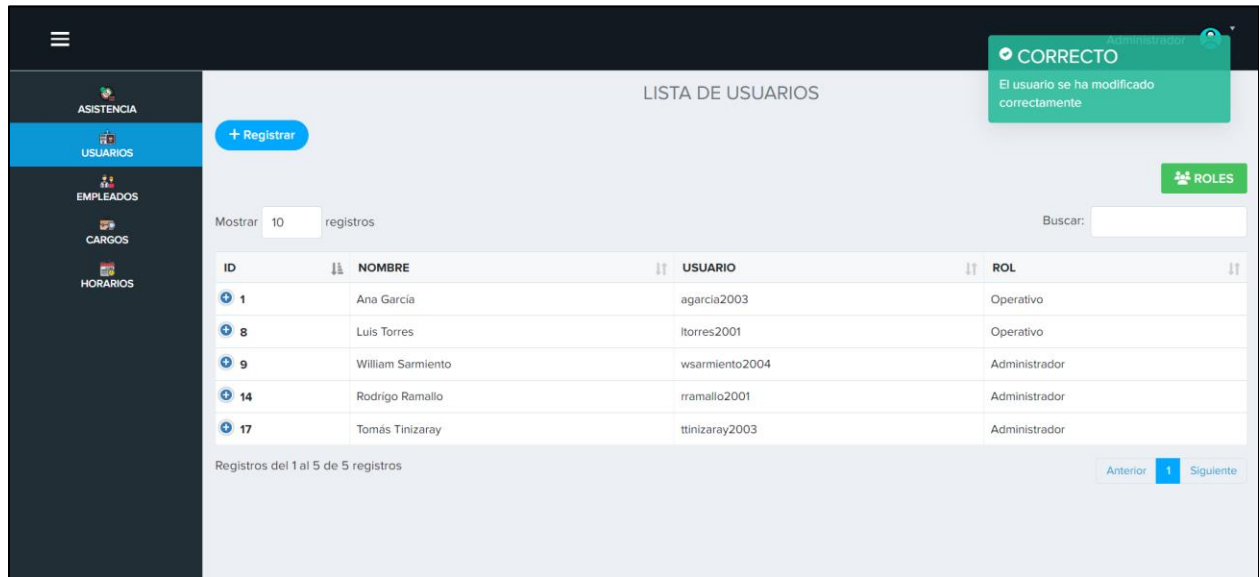


Figura 20  
Pantalla registro de usuario

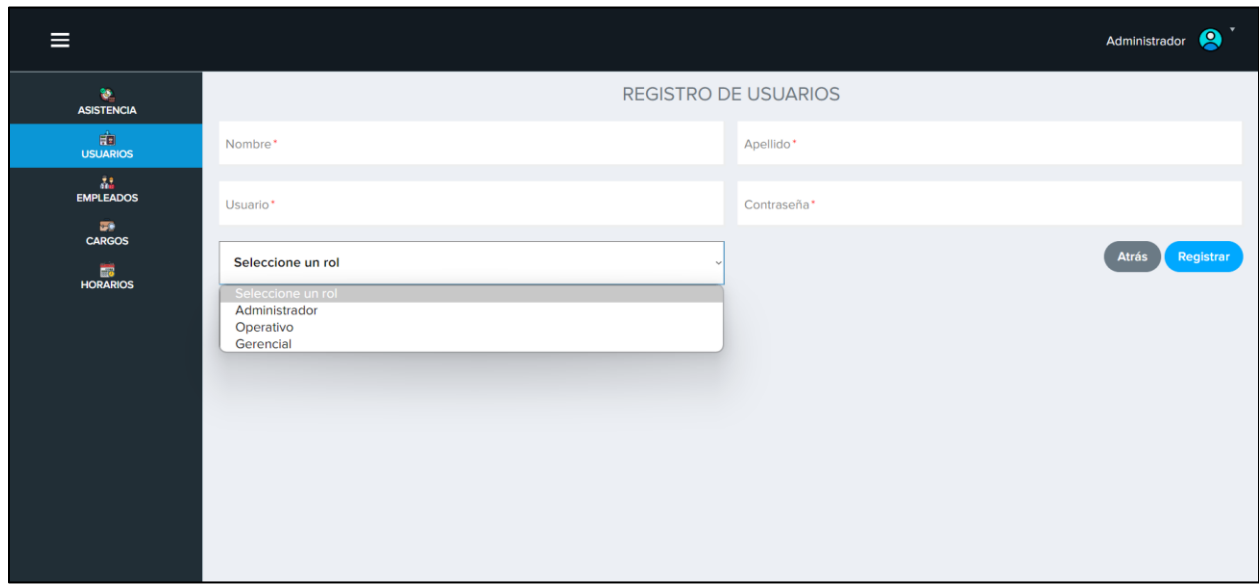
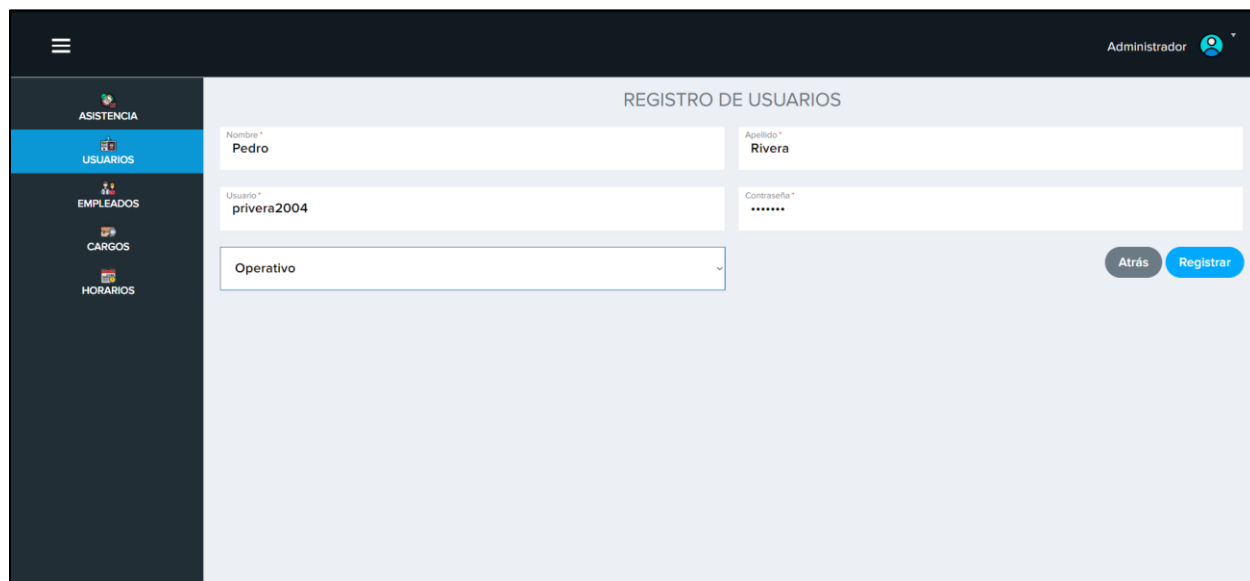


Figura 21

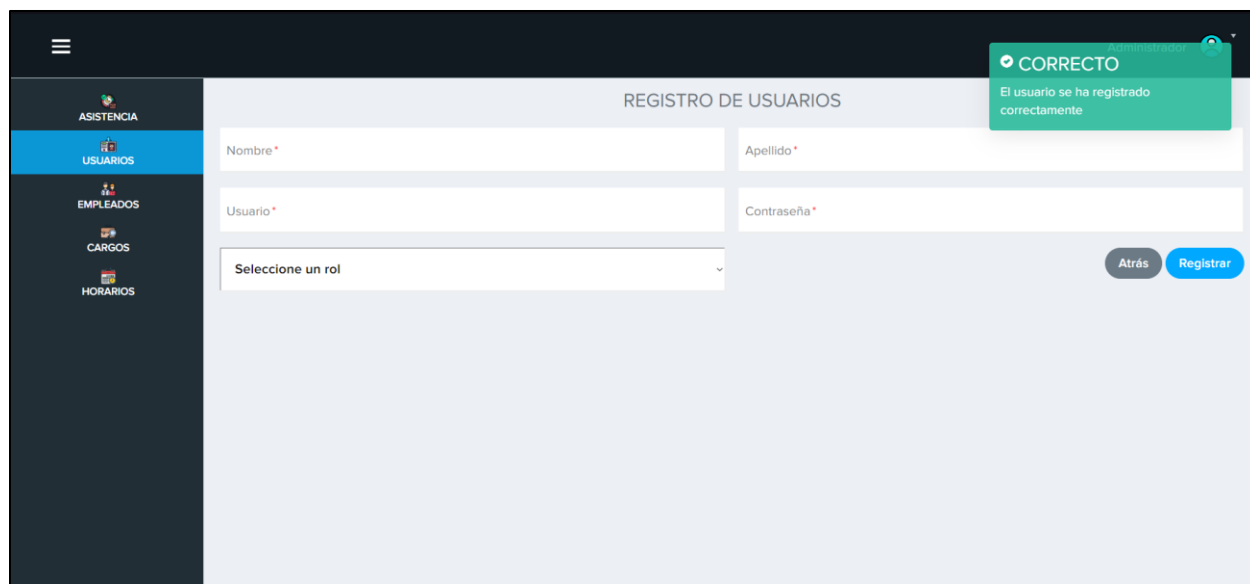
Pantalla registro de usuario ingresando su información



The screenshot shows a web application interface for user registration. The page title is "REGISTRO DE USUARIOS". On the left, there is a dark sidebar with a menu containing "ASISTENCIA", "USUARIOS" (highlighted in blue), "EMPLEADOS", "CARGOS", and "HORARIOS". The main content area contains a registration form with the following fields: "Nombre\*" with the value "Pedro", "Apellido\*" with the value "Rivera", "Usuario\*" with the value "privera2004", and "Contraseña\*" with masked characters "\*\*\*\*\*". Below these fields is a dropdown menu for "Operativo" with a downward arrow. At the bottom right of the form, there are two buttons: "Atrás" (grey) and "Registrar" (blue). In the top right corner, the user is identified as "Administrador" with a profile icon.

Figura 22

Notificación registro de usuario correcto



This screenshot shows the same user registration form as in Figure 21, but with a success notification. A green toast message is displayed in the top right corner, containing a checkmark icon, the word "CORRECTO", and the text "El usuario se ha registrado correctamente". The form fields are now empty: "Nombre\*", "Apellido\*", "Usuario\*", and "Contraseña\*". The dropdown menu is labeled "Seleccione un rol" with a downward arrow. The "Atrás" and "Registrar" buttons remain at the bottom right. The "Administrador" user name is still visible in the top right corner.

Figura 23  
Pantalla eliminación de usuario

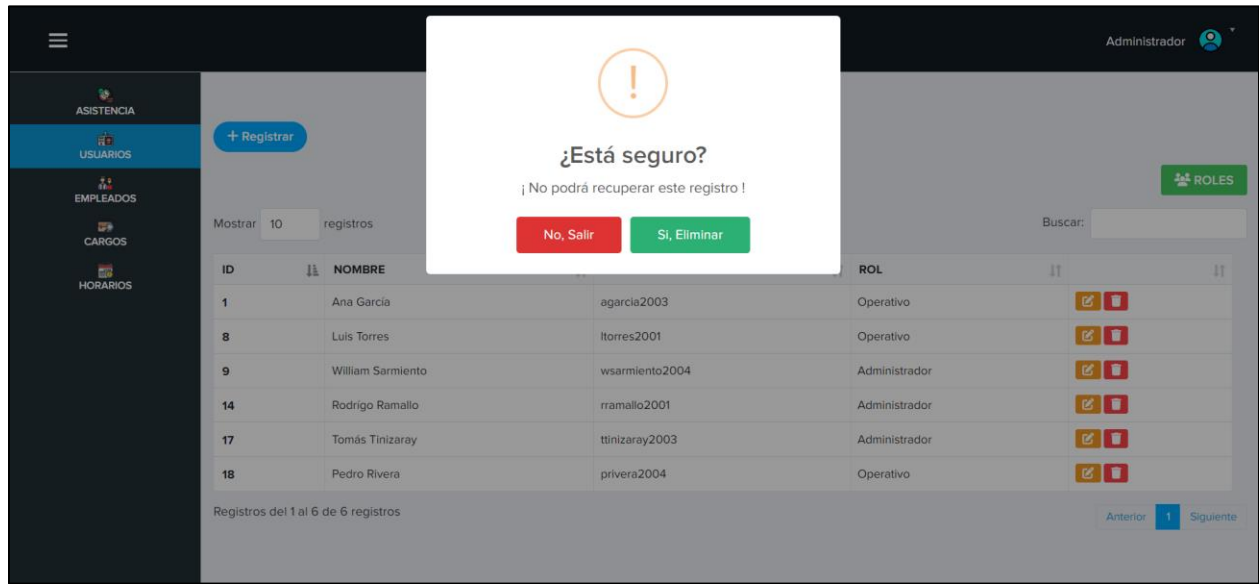
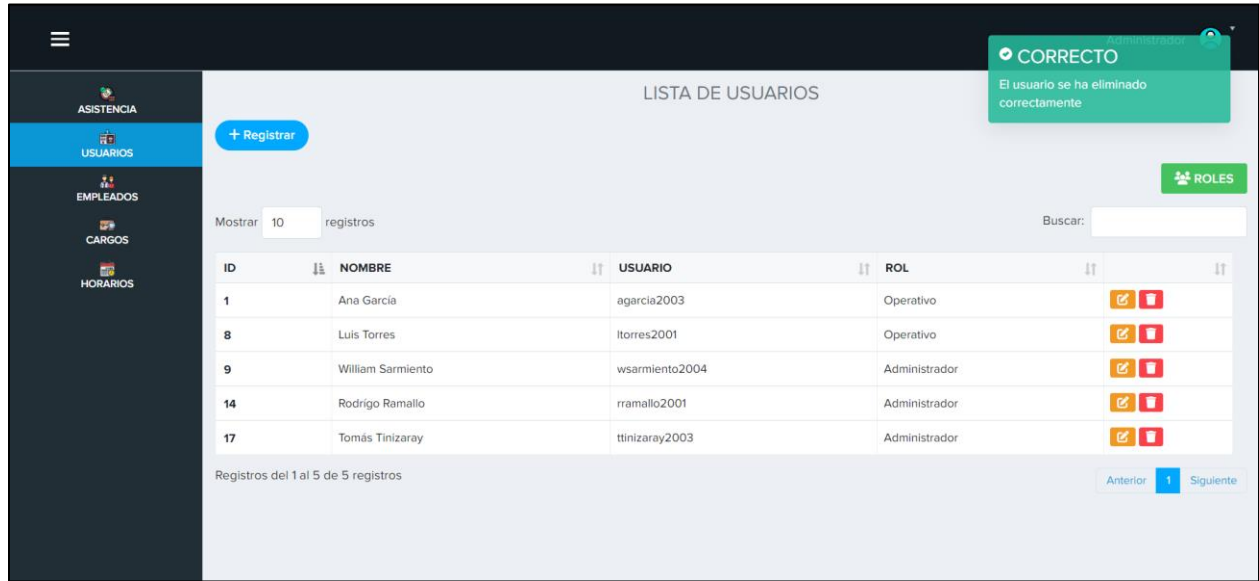


Figura 24  
Notificación eliminación de usuario correcta



### 4.1.2.3 GESTIÓN DE ROLES DE USUARIOS

Figura 25

Pantalla listado de roles de usuario

Administrador

#### LISTA DE ROLES DE USUARIO

+ Registrar

Mostrar 10 registros

Buscar:

ID	DESCRIPCION	
1	Administrador	
2	Operativo	
7	Gerencial	

Registros del 1 al 3 de 3 registros

Anterior 1 Siguiete

Figura 26

Pantalla modificación de roles de usuario

Administrador

#### Modificar Rol

Nombre  
Administrador

Atrás Modificar

ID	DESCRIPCION	
1	Administrador	
2	Operativo	
7	Gerencial	

Registros del 1 al 3 de 3 registros

Anterior 1 Siguiete

Figura 27

Notificación modificación de roles de usuario correcta

The screenshot displays the 'LISTA DE ROLES DE USUARIO' (User Role List) interface. A green notification box in the top right corner indicates a successful update: 'CORRECTO' and 'El rol Administrador se ha modificado correctamente'. The interface includes a sidebar with navigation options: ASISTENCIA, USUARIOS (selected), EMPLEADOS, CARGOS, and HORARIOS. A '+ Registrar' button is visible. The main content area shows a table with columns 'ID' and 'DESCRIPCION'. The table lists three roles: ID 1 (Administrador), ID 2 (Operativo), and ID 7 (Gerencial). Each row has edit and delete icons. Below the table, it shows 'Registros del 1 al 3 de 3 registros' and navigation buttons for 'Anterior', '1', and 'Siguinte'. A search bar is also present.

ID	DESCRIPCION	
1	Administrador	
2	Operativo	
7	Gerencial	

Figura 28

Pantalla registro de roles de usuario

The screenshot displays the 'REGISTRO DE ROLES DE USUARIO' (User Role Registration) interface. The page title is 'REGISTRO DE ROLES DE USUARIO'. The sidebar is identical to the previous screenshot, with 'USUARIOS' selected. The main content area features a form with a 'Descripción\*' label and the text 'Rol de Empleado Prueba'. At the bottom right, there are 'Atrás' and 'Registrar' buttons.

Figura 29

Notificación registro de roles de usuario correcto

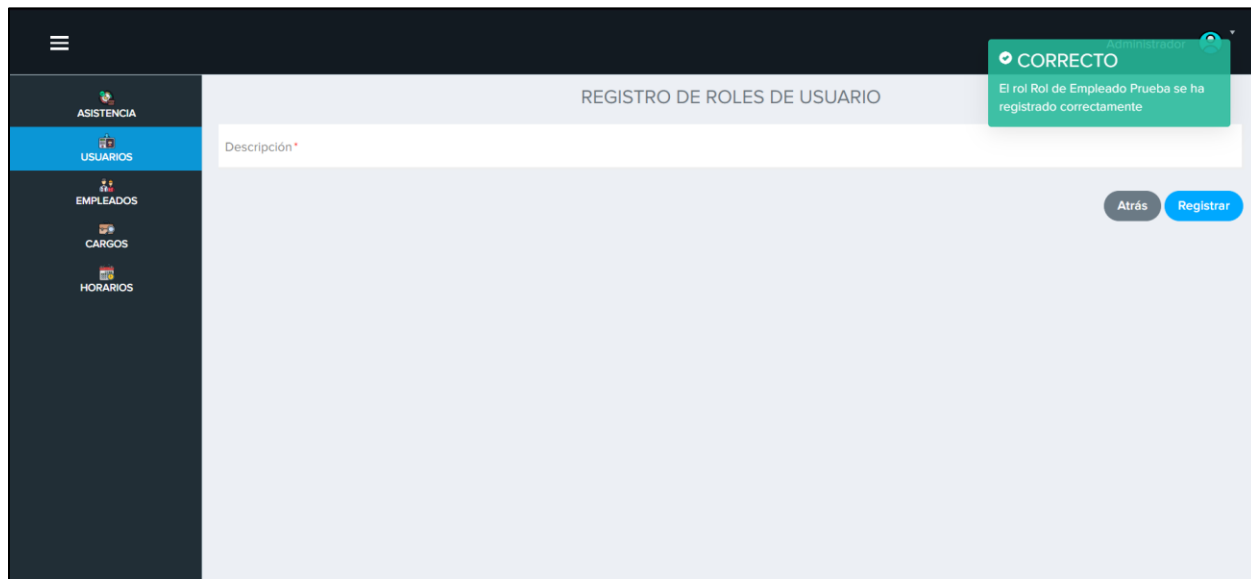


Figura 30

Pantalla eliminación de roles de usuario

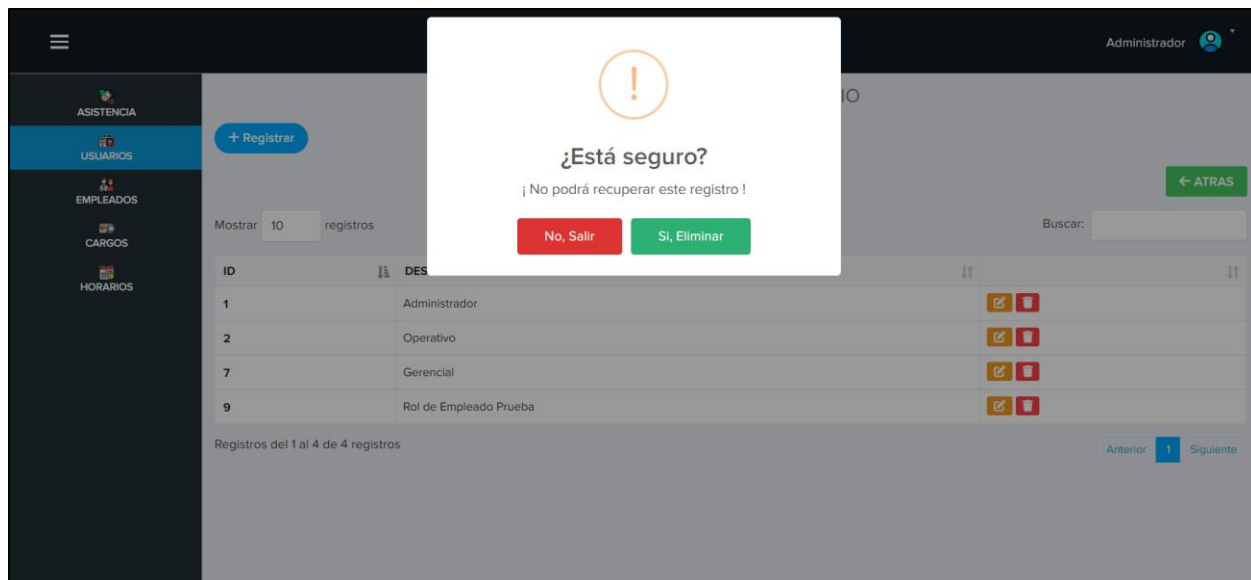


Figura 31

Notificación eliminación de roles de usuario correcta



### 4.1.3 PRUEBAS Y RESULTADOS ITERACIÓN 1

#### PRUEBAS DE USABILIDAD:

**Objetivo:** Evaluar la facilidad de uso y la intuición de las interfaces de usuario.

Para este estudio, se seleccionaron diez usuarios con distintos niveles de experiencia tecnológica. Se les asignaron tareas específicas, tales como registrarse, iniciar sesión y cambiar su información de perfil. Se utilizaron herramientas de grabación de pantalla, se administraron cuestionarios de satisfacción post-prueba y el uso de la técnica “Thinking Aloud” para recolectar datos cualitativos y cuantitativos.

#### Resultados:

- Positivos: Los usuarios encontraron las interfaces intuitivas y fáciles de navegar, destacando la claridad de los formularios y la rapidez del proceso de registro.
- Mejoras: Algunos usuarios sugirieron reorganizar los botones y mejorar los mensajes de error para que fueran más específicos y útiles.

**Acciones Derivadas:** Tras recibir la retroalimentación de los usuarios, se implementó un filtro general en la visualización de la pantalla de usuario y sus roles, permitiendo clasificar por el texto

que escriba el usuario. Además, se optimizó la navegación en la interfaz de usuario reorganizando los menús y botones, donde se sugirió usar iconos en lugar de palabras textuales para la eliminación y edición. Estas mejoras resultaron en una experiencia de usuario más fluida y eficiente.

### **PRUEBAS UNITARIAS:**

**Objetivo:** Verificar que cada componente individual del sistema funcione correctamente.

Se desarrollaron y ejecutaron casos de prueba automatizados utilizando la herramienta Junit, cada caso de prueba se centró en una función específica, como la validación de entradas en el formulario de registro, el cifrado de contraseñas y la autenticación de usuarios.

Las pruebas cubrieron todas las rutas de código posibles para asegurar que cada función se comportara como se esperaba bajo diferentes condiciones.

#### **Resultados:**

Todas las pruebas unitarias pasaron, confirmando que los componentes aislados funcionan correctamente. No se encontraron errores en el código individual de los módulos probados.

### **PRUEBAS DE FUNCIONALIDAD:**

**Objetivo:** Asegurar que las funciones del sistema cumplan con los requisitos especificados.

Para las pruebas de funcionalidad se desarrollaron escenarios de uso que replicaban situaciones del mundo real, tales como el registro de un nuevo usuario, el inicio de sesión con credenciales correctas e incorrectas, y el inicio de sesión con los campos vacíos. Se compararon los resultados obtenidos con los resultados esperados para cada escenario de uso.

#### **Resultados:**

Las pruebas confirmaron que las funcionalidades operaron conforme a los requisitos. Todos los casos de prueba se completaron con éxito, permitiendo la creación, edición, búsqueda y eliminación de usuarios y roles sin errores.

### **PRUEBAS DE INTEGRACIÓN:**

**Objetivo:** Comprobar la interacción correcta entre diferentes módulos del sistema.

Se realizaron pruebas que implicaban la secuencia completa de registro de usuarios, inicio de sesión y asignación de roles. Se utilizó un entorno de pruebas que replicaba el entorno de producción, con el uso de la herramienta Jenkins para la integración continua.

**Resultados:**

La integración entre los módulos de gestión de usuarios, roles de usuario e inicio de sesión fue exitosa. Los datos se transfirieron y se procesaron correctamente entre módulos, sin pérdida de información ni errores de sincronización.

**PRUEBAS DE RENDIMIENTO:**

**Objetivo:** Evaluar la capacidad del sistema bajo carga y su tiempo de respuesta.

Para comprobar el rendimiento se utilizó JMeter para simular la carga del sistema con múltiples usuarios accediendo y operando simultáneamente. Las pruebas incluyeron picos de tráfico y cargas sostenidas durante periodos prolongados, donde se midió el tiempo de respuesta, la tasa de errores y el uso de recursos del servidor.

**Resultados:**

El sistema mantuvo un rendimiento aceptable bajo las condiciones de prueba. Los tiempos de respuesta fueron inferiores a dos segundos para la mayoría de las operaciones, y no se registraron fallos o cuellos de botella significativos.

**RESULTADOS OBTENIDOS:**

Al finalizar la primera iteración, se implementaron y validaron con éxito las funcionalidades básicas del sistema. Las pruebas demostraron que el sistema era intuitivo, funcional y capaz de manejar múltiples usuarios sin degradación significativa del rendimiento. Se identificaron algunas áreas para mejoras menores en la interfaz de usuario, que se abordarán en iteraciones futuras.

## 4.2 ITERACIÓN 2 - DESARROLLO DE LA GESTIÓN DE CARGOS Y HORARIOS

La segunda iteración del proyecto se enfocó en desarrollar las funcionalidades de gestión de cargos y horarios de empleados dentro del sistema web de registro de asistencias. En esta fase, se definieron con precisión los requisitos específicos para la creación, actualización, búsqueda y eliminación de cargos y horarios. Con base en estos requisitos, se diseñaron interfaces de usuario intuitivas y eficientes para la gestión de cargos y horarios, facilitando la navegación y la realización de tareas administrativas.

De igual forma, se implementó la lógica del controlador para manejar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de los cargos y horarios. Estas funcionalidades se integraron con la base de datos para garantizar el almacenamiento y la recuperación segura de la información. Durante esta iteración, se realizaron pruebas exhaustivas para verificar la integridad de los datos y asegurar que las operaciones de gestión de cargos y horarios se realizaran de manera segura y correcta. Se evaluaron diferentes escenarios para identificar y corregir posibles errores o fallos en el sistema.

#### 4.2.1 TABLA DE ACTIVIDADES ITERACIÓN 2

<b>Funcionalidad</b>	<b>Descripción</b>	<b>Actor</b>	<b>Criterio de Aceptación</b>
<b>Diseño de interfaz para la ventana de Cargos y Horarios de los empleados.</b>	El usuario contará con una interfaz intuitiva para ingresar los cargos y horarios necesarios para los empleados.	Usuario	Interfaces intuitivas con validaciones y notificaciones para el correcto ingreso de cargos.
<b>Visualización y modificación de Cargos y Horarios de los empleados.</b>	El usuario podrá visualizar todos los cargos y horarios creados y editar los campos correspondientes.	Usuario	Visualizar los cargos dentro de su vista y editar al correctamente.

<p><b>Creación y eliminación de Cargos y Horarios de los empleados.</b></p>	<p>El usuario tiene la potestad de ingresar cuantos cargos y horarios se consideren necesarios y eliminarlos según el caso.</p>	<p>Usuario</p>	<p>Verificar los cargos que han sido añadidos y eliminados.</p>
---	---	----------------	---

Tabla 9. Tabla actividades Iteración 2

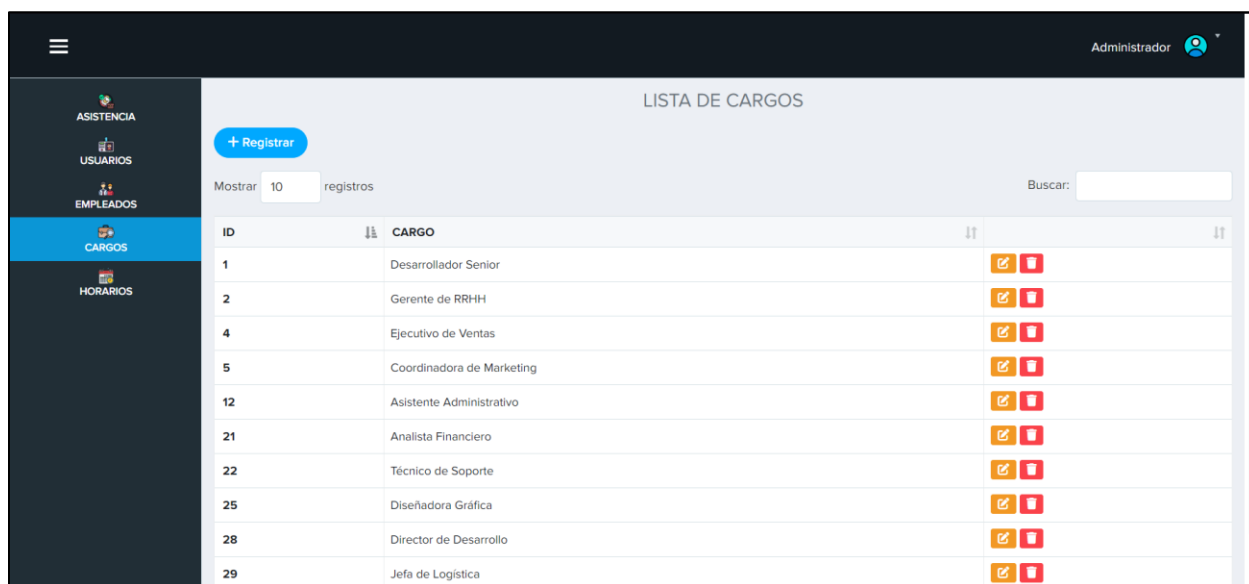
*Elaboración propia.*

## 4.2.2 CAPTURAS DE PANTALLA ITERACIÓN 2

### 4.2.2.1 GESTIÓN DE CARGOS DE EMPLEADOS

Figura 31

Pantalla lista de cargos de los empleados



ID	CARGO		
1	Desarrollador Senior		
2	Gerente de RRHH		
4	Ejecutivo de Ventas		
5	Coordinadora de Marketing		
12	Asistente Administrativo		
21	Analista Financiero		
22	Técnico de Soporte		
25	Diseñadora Gráfica		
28	Director de Desarrollo		
29	Jefa de Logística		

Figura 32

Pantalla modificación de cargos de los empleados

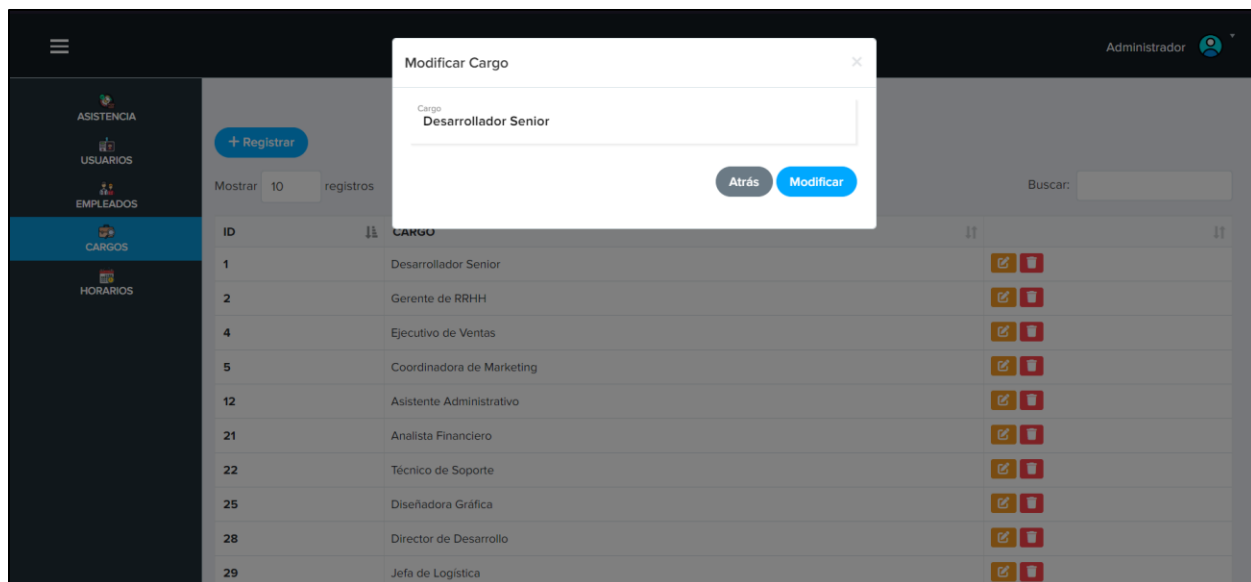


Figura 33

Notificación modificación de cargos de los empleados correcta

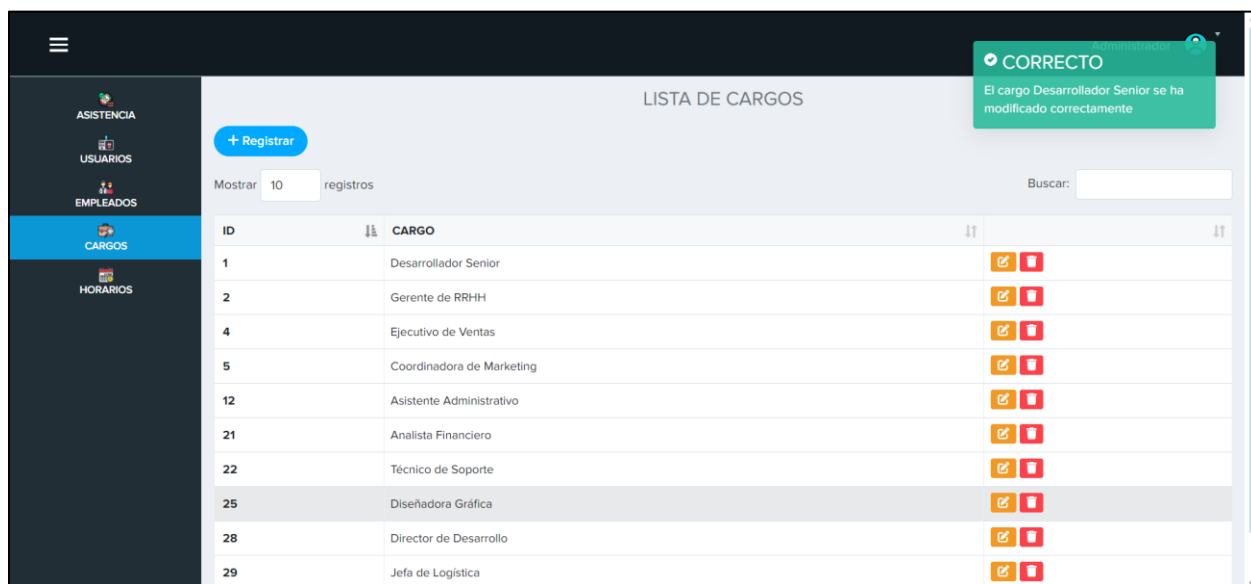


Figura 34

Pantalla registro de cargos de los empleados

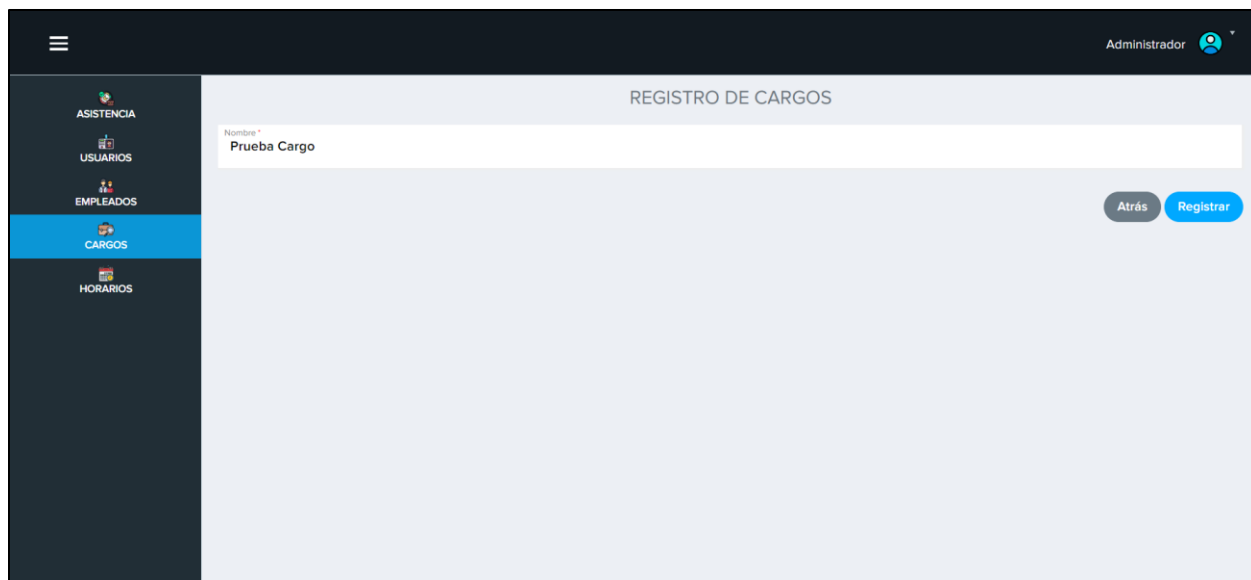


Figura 35

Notificación registro de cargos de los empleados correcto

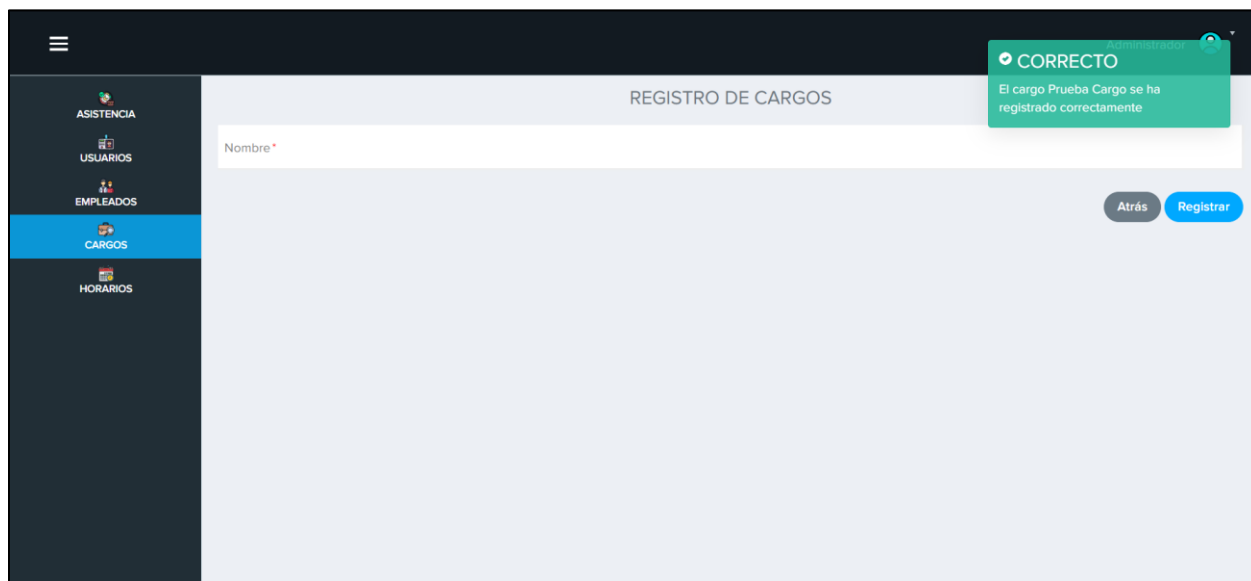


Figura 36

Pantalla eliminación de cargos de los empleados

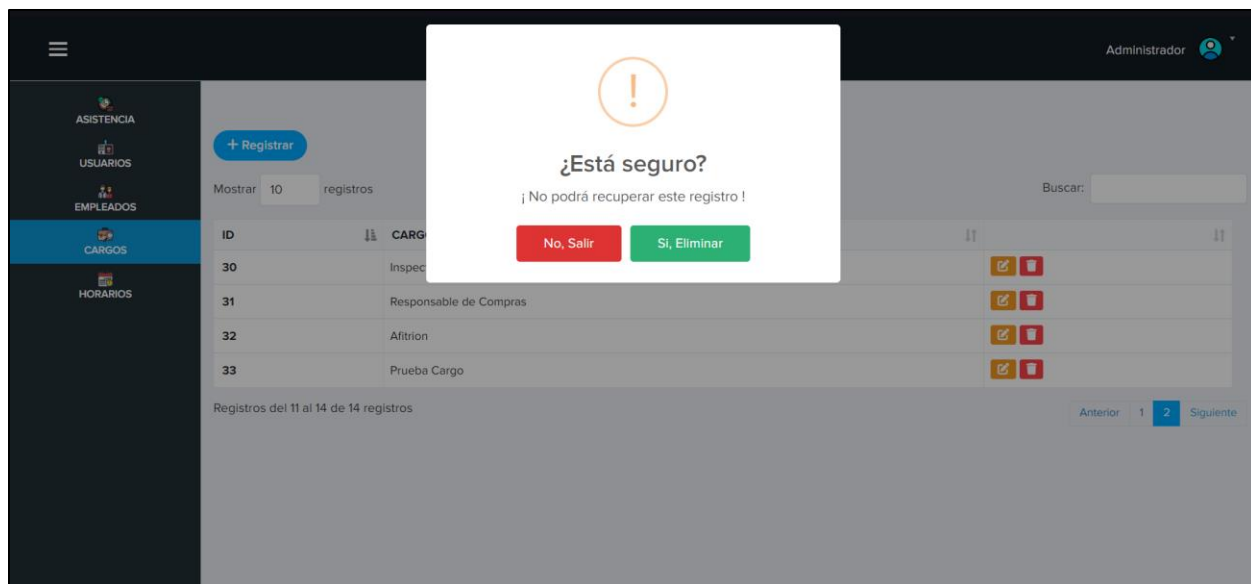
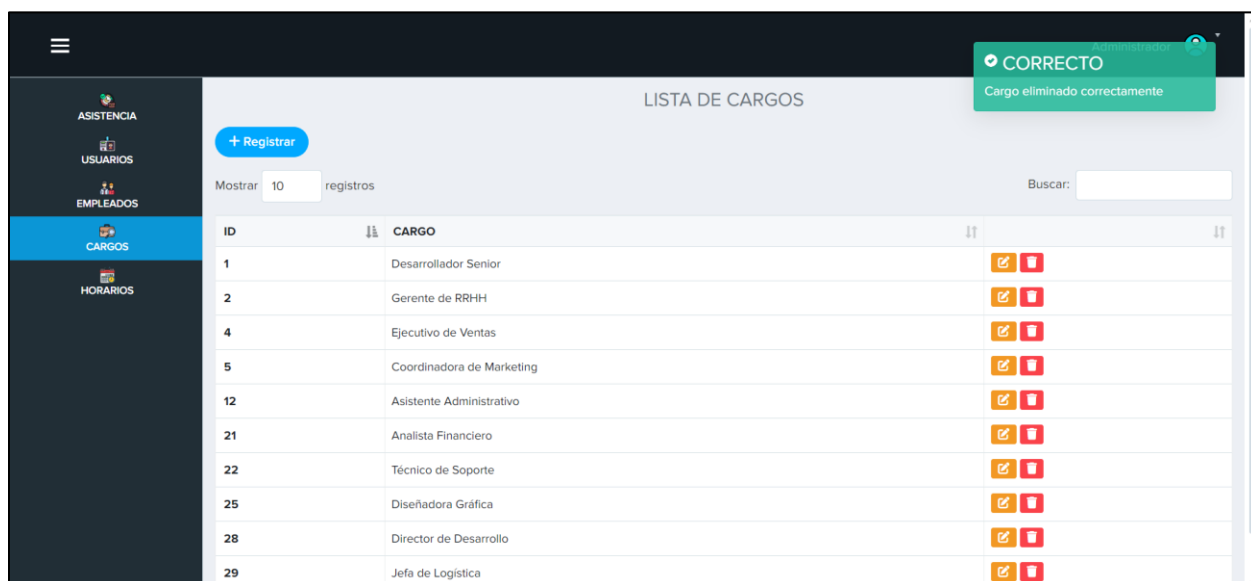


Figura 37

Notificación eliminación de cargos de los empleados correcta



#### 4.2.2.2 GESTIÓN DE HORARIOS DE EMPLEADOS

Figura 38

Pantalla listado de horarios de los empleados

ID	DESCRIPCIÓN	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
1	Ventas Matutino	08:00:00 - 16:00:00	09:00:00 - 17:00:00	08:30:00 - 16:30:00	09:00:00 - 17:00:00	08:00:00 - 15:00:00
4	Ventas Vespertino	14:00:00 - 20:00:00	16:00:00 - 22:00:00	15:00:00 - 21:00:00	15:00:00 - 21:00:00	09:00:00 - 14:00:00
6	Marketing Matutino	07:00:00 - 17:00:00	07:00:00 - 18:00:00	07:00:00 - 19:00:00	07:00:00 - 20:00:00	09:00:00 - 16:00:00
12	Marketing Vespertino	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00
13	Sistemas Matutino	07:00:00 - 16:00:00	07:00:00 - 17:00:00	07:00:00 - 18:00:00	07:00:00 - 18:00:00	07:00:00 - 16:00:00

Figura 39

Pantalla modificación de los horarios de los empleados

Martes

09:00 a. m. 05:00 p. m.

Miércoles

04:30 p. m.

Jueves

05:00 p. m.

Viernes

08:00 a. m. 03:00 p. m.

Atrás Modificar

Figura 40

Notificación modificación de los horarios de los empleados correcta

**CORRECTO**  
El horario Ventas Matutino se ha modificado correctamente

HORARIO DE LOS EMPLEADOS

+ Registrar

Mostrar 10 registros Buscar:

ID	DESCRIPCIÓN	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
1	Ventas Matutino	08:00:00 - 16:00:00	08:00:00 - 17:00:00	08:30:00 - 16:30:00	09:00:00 - 17:00:00	08:00:00 - 15:00:00
4	Ventas Vespertino	14:00:00 - 20:00:00	16:00:00 - 22:00:00	15:00:00 - 21:00:00	15:00:00 - 21:00:00	09:00:00 - 14:00:00
6	Marketing Matutino	07:00:00 - 17:00:00	07:00:00 - 18:00:00	07:00:00 - 19:00:00	07:00:00 - 20:00:00	09:00:00 - 16:00:00
12	Marketing Vespertino	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00
13	Sistemas Matutino	07:00:00 - 16:00:00	07:00:00 - 17:00:00	07:00:00 - 18:00:00	07:00:00 - 18:00:00	07:00:00 - 16:00:00

Registros del 1 al 5 de 5 registros Anterior 1 Siguiente

Figura 41

Pantalla registro de los horarios de los empleados

Administrador

REGISTRO DE HORARIOS

Descripción  
Horario Prueba

Lunes

Hora de Entrada: 07:30 a. m.  Hora de Salida: 03:30 p. m.

Martes

Hora de Entrada: 04:30 p. m.  Hora de Salida: 08:30 p. m.

Miércoles

Hora de Entrada: 08:00 a. m.  Hora de Salida: 06:00 p. m.

Figura 42

Notificación registro de los horarios de los empleados correcto

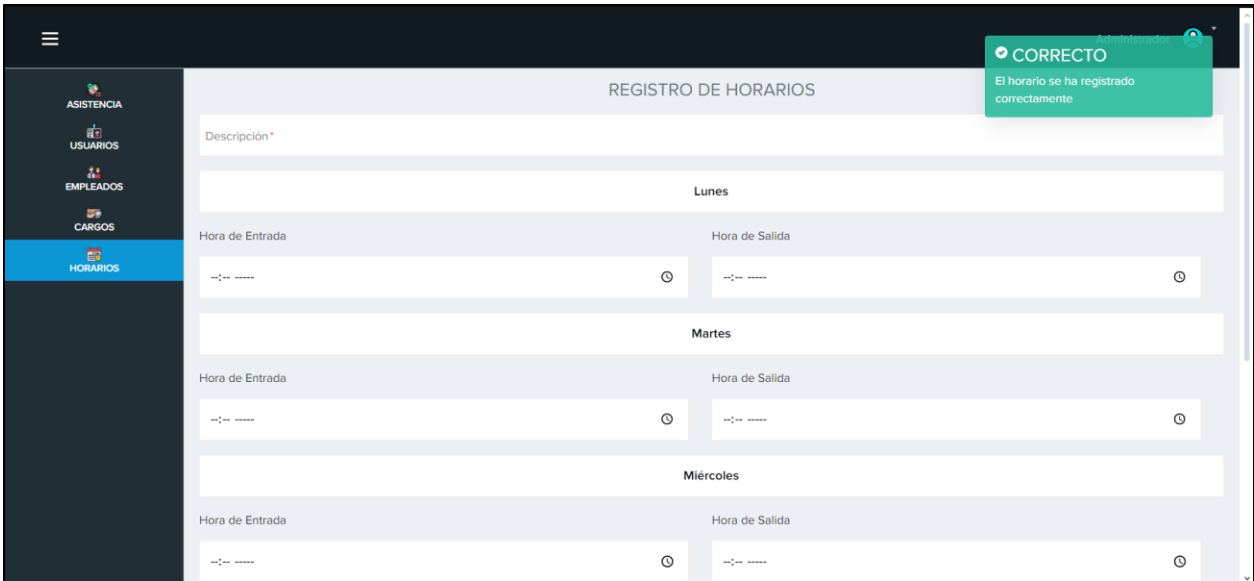


Figura 43  
Pantalla eliminación de los horarios de los empleados

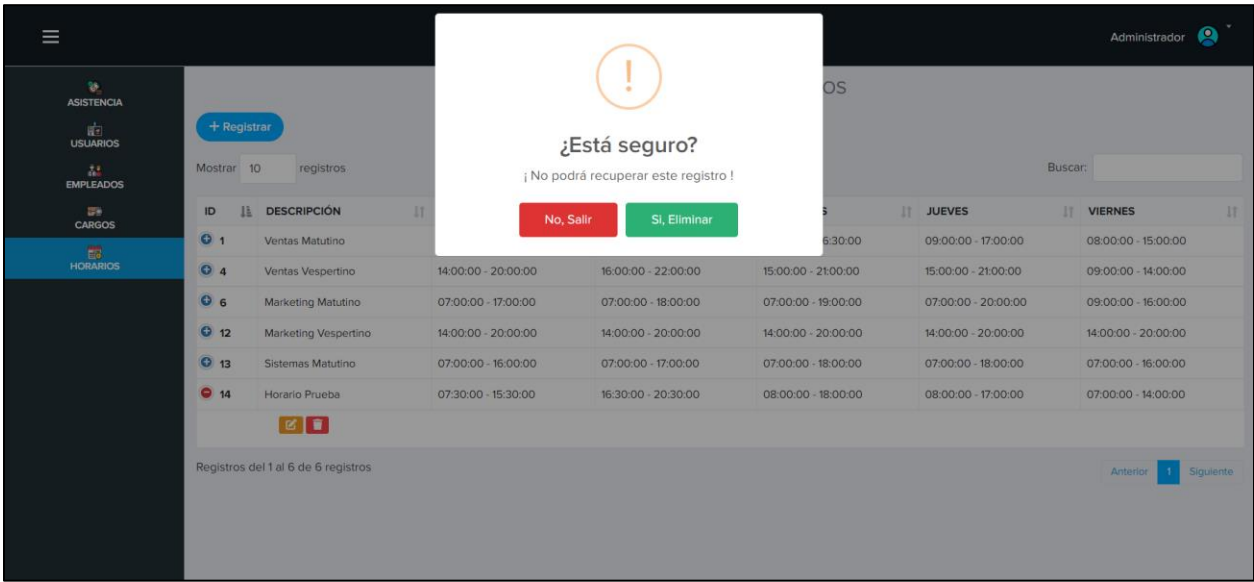


Figura 44  
Notificación eliminación de los horarios de los empleados correcta

ID	DESCRIPCIÓN	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
1	Ventas Matutino	08:00:00 - 16:00:00	08:00:00 - 17:00:00	08:30:00 - 16:30:00	09:00:00 - 17:00:00	08:00:00 - 15:00:00
4	Ventas Vespertino	14:00:00 - 20:00:00	16:00:00 - 22:00:00	15:00:00 - 21:00:00	15:00:00 - 21:00:00	09:00:00 - 14:00:00
6	Marketing Matutino	07:00:00 - 17:00:00	07:00:00 - 18:00:00	07:00:00 - 19:00:00	07:00:00 - 20:00:00	09:00:00 - 16:00:00
12	Marketing Vespertino	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00	14:00:00 - 20:00:00
13	Sistemas Matutino	07:00:00 - 16:00:00	07:00:00 - 17:00:00	07:00:00 - 18:00:00	07:00:00 - 18:00:00	07:00:00 - 16:00:00

## 4.2.3 PRUEBAS Y RESULTADOS ITERACIÓN 2

### PRUEBAS DE USABILIDAD:

**Objetivo:** Asegurar que la gestión de cargos y horarios sea fácil de usar.

En la segunda iteración se realizaron pruebas de usabilidad con diez usuarios, quienes completaron tareas específicas utilizando el sistema, para ello se utilizaron herramientas de grabación de pantalla, se administraron cuestionarios de satisfacción post-prueba y el uso de la técnica “Thinking Aloud” para recolectar datos cualitativos y cuantitativos.

### Resultados:

- **Positivos:** Los usuarios encontraron la interfaz de gestión de cargos y horarios clara y fácil de usar, destacando la simplicidad de la creación y edición de horarios.
- **Mejoras:** Sugirieron mejoras en la visualización de horarios, especialmente en la representación gráfica de los mismos.

**Acciones Derivadas:** Tras recibir la retroalimentación de los usuarios, se implementó un filtro general en la visualización de la pantalla de cargos y horarios, permitiendo clasificar por el texto que escriba el usuario. Además, se optimizó la navegación en la interfaz de usuario reorganizando los menús y botones, donde se sugirió usar iconos en lugar de palabras textuales para la eliminación y edición, además de usar un input que permita al usuario elegir la hora que desee para el horario en lugar de escribirla, esto con la finalidad de tener funciones más accesibles. Estas mejoras resultaron en una experiencia de usuario más fluida y eficiente.

### **PRUEBAS UNITARIAS:**

**Objetivo:** Validar la funcionalidad de cada componente individual de gestión de cargos y horarios.

En las pruebas unitarias se desarrollaron casos de prueba automatizados para las funciones de creación, actualización, eliminación y búsqueda de cargos y horarios, utilizando JUnit. Las pruebas cubrieron todas las posibles combinaciones de datos de entrada y situaciones de error.

### **Resultados:**

Las funciones pasaron todas las pruebas unitarias, asegurando la correcta funcionalidad de los componentes. No se encontraron defectos significativos.

### **PRUEBAS DE FUNCIONALIDAD:**

**Objetivo:** Verificar que las funciones cumplan con los requisitos definidos.

Para la funcionalidad se validaron los escenarios de uso relacionados con la gestión de cargos y horarios, utilizando casos de prueba que replicaban situaciones del mundo real. Con esto se logró una comparación de los resultados obtenidos con los resultados esperados y revisión por pares.

### **Resultados:**

Las funcionalidades cumplieron con los requisitos. Todas las pruebas de funcionalidad se completaron con éxito, permitiendo la correcta gestión de cargos y horarios de empleados.

### **PRUEBAS DE INTEGRACIÓN:**

**Objetivo:** Evaluar la interacción entre el módulo de gestión de cargos/horarios y otros módulos del sistema.

En la integración se realizaron pruebas que incluían la asignación de cargos y horarios a usuarios registrados previamente, utilizando un entorno de pruebas controlado gracias a la herramienta Jenkins para la integración continua.

#### **Resultados:**

La integración fue exitosa. Los datos sobre cargos y horarios se sincronizaron correctamente con los módulos de gestión de usuarios y roles, sin inconsistencias ni errores.

### **PRUEBAS DE RENDIMIENTO:**

**Objetivo:** Analizar el rendimiento del sistema durante la gestión de múltiples cargos y horarios.

Para lograr medir el rendimiento se realizó una simulación de múltiples operaciones de gestión realizadas simultáneamente utilizando JMeter. Se midieron tiempos de respuesta y utilización de recursos del servidor bajo diferentes cargas. Dentro de las métricas a revisar se consideró tiempos de respuesta, tasas de error y consumo de CPU/memoria.

#### **Resultados:**

El sistema mantuvo un rendimiento estable y tiempos de respuesta rápidos bajo la carga simulada. No se identificaron cuellos de botella ni degradación significativa del rendimiento.

## RESULTADOS OBTENIDOS

La segunda iteración culminó con éxito, proporcionando una funcionalidad robusta para la gestión de cargos y horarios. Las pruebas confirmaron que el sistema era funcional, fácil de usar y capaz de manejar las operaciones administrativas sin problemas significativos. Las sugerencias de mejora se documentaron para su implementación en futuras iteraciones.

### 4.3 ITERACIÓN 3 - DESARROLLO DE LA GESTIÓN DE EMPLEADOS Y SUS ROLES

En la tercera etapa del proyecto, se abordó el desarrollo de las funcionalidades de gestión de empleados y sus roles en el sistema. Primero, se identificaron y definieron de manera precisa los requisitos necesarios para manejar la información relevante de los empleados, incluyendo datos personales y detalles laborales. Con base en estos requisitos, se diseñaron formularios y vistas intuitivas para facilitar la entrada y visualización de la información de los empleados y sus roles. Estos formularios y vistas fueron cuidadosamente elaborados para proporcionar una experiencia de usuario fluida. La implementación incluyó la creación de modelos de datos robustos en el controlador y la integración de estos con la interfaz de usuario. Se desarrollaron funcionalidades para añadir, editar y eliminar registros de empleados y sus roles, así como para visualizar listas y detalles individuales de manera clara y organizada. Durante esta iteración, se realizaron pruebas exhaustivas para garantizar la precisión y consistencia de los datos de los empleados. Se pusieron a prueba diferentes escenarios para identificar y corregir posibles errores o fallos en el sistema. Finalmente, una vez completadas estas tareas, se presentó a los usuarios la funcionalidad de gestión de empleados y sus roles para obtener retroalimentación valiosa y realizar los ajustes necesarios. Se recogieron sugerencias y comentarios para identificar oportunidades de mejora en futuras iteraciones del desarrollo.

### 4.3.1 TABLA DE ACTIVIDADES ITERACIÓN 3

<b>Funcionalidad</b>	<b>Descripción</b>	<b>Actor</b>	<b>Criterio de Aceptación</b>
<b>Diseño de interfaz para la ventana de Empleados y Roles de Empleado.</b>	El usuario contará con una interfaz intuitiva para ingresar los empleados y sus roles necesarios según el caso.	Usuario	Interfaces intuitivas con validaciones y notificaciones para el correcto ingreso de los datos.
<b>Visualización y modificación de Empleados y sus roles.</b>	El usuario podrá visualizar todos los empleados y sus roles creados y editar los campos correspondientes.	Usuario	Visualizar los empleados dentro de su vista y editar al correctamente.
<b>Creación y eliminación de Empleados y sus roles.</b>	El usuario tiene la potestad de ingresar cuantos empleados y sus roles se consideren necesarios y eliminarlos según el caso.	Usuario	Verificar los empleados que han sido añadidos y eliminados.

Tabla 10. Tabla actividades Iteración 3

*Elaboración propia.*

### 4.3.2 CAPTURAS DE PANTALLA ITERACIÓN 3

#### 4.3.2.1 GESTIÓN DE EMPLEADOS

Figura 45

## Pantalla lista de empleados

Administrador

LISTA DE EMPLEADOS

+ Registrar

Mostrar 10 registros

Buscar:

ROLES

ID	NOMBRE	CEDULA	CARGO	TIPO DE CONTRATO	ROL	HORARIO
1	Fermin López	1739681255	Ejecutivo de Ventas	Tiempo Parcial	Diseño	Ventas Matutino
16	Paula Cubarsí	17224231384	Coordinadora de Marketing	Tiempo Parcial	Marketing	Marketing Vespertino
17	Joao Cancelo	1727274860	Desarrollador Senior	Tiempo Completo	Ingeniero de Software	Sistemas Matutino
18	Rebeca Villavicencio	1727245563	Analista Financiero	Tiempo Completo	Finanzas	Ventas Vespertino
24	Andreas Christensen	1727453620	Ejecutivo de Ventas	Tiempo Completo	Ventas	Ventas Matutino
25	Samuel Vega	1752815652	Ejecutivo de Ventas	Tiempo Completo	Compras	Ventas Matutino
26	Ferran Torres	1737475630	Asistente Administrativo	Tiempo Parcial	Administración	Ventas Matutino
27	Hector Fort	1789453420	Técnico de Soporte	Tiempo Parcial	Soporte Técnico	Ventas Matutino
28	Ronald Araujo	1743236760	Desarrollador Senior	Tiempo Completo	Ingeniero de Software	Sistemas Matutino
29	Luciana Sosa	1789563250	Jefa de Logística	Tiempo Completo	Logística	Ventas Vespertino

Figura 46

## Pantalla modificación de empleado

Administrador

Modificar Empleado

Nombre  
Javier

Apellido  
Hernández

Cedula  
1746374840

Responsable de Compras

Tiempo Completo

Ventas

Ventas Vespertino

Atrás Modificar

Mostrar 10 registros

Buscar:

ROLES

ID	NOMBRE	ROL	HORARIO
30	Javier Hernández	Ventas	Ventas Vespertino
31	Carlos Puyol	Ingeniero de Software	Sistemas Matutino

Registros del 11 al 12 de 12 registros

Anterior 1 2 Siguiente

Figura 47

Notificación modificación de empleado correcta

The screenshot shows the 'LISTA DE EMPLEADOS' (Employee List) page. A green notification box in the top right corner displays the message: 'CORRECTO El empleado se ha modificado correctamente.' (Correct. The employee has been modified correctly.) The page features a sidebar with navigation options: ASISTENCIA, USUARIOS, EMPLEADOS (highlighted), CARGOS, and HORARIOS. A '+ Registrar' button is visible. The table below lists 10 employees with columns for ID, NOMBRE, CEDULA, CARGO, TIPO DE CONTRATO, ROL, and HORARIO.

ID	NOMBRE	CEDULA	CARGO	TIPO DE CONTRATO	ROL	HORARIO
1	Fermin López	1739681255	Ejecutivo de Ventas	Tiempo Parcial	Diseño	Ventas Matutino
16	Paula Cubarsi	17224231384	Coordinadora de Marketing	Tiempo Parcial	Marketing	Marketing Vespertino
17	Joao Cancelo	1727274860	Desarrollador Senior	Tiempo Completo	Ingeniero de Software	Sistemas Matutino
18	Rebeca Villavicencio	1727245563	Analista Financiero	Tiempo Completo	Finanzas	Ventas Vespertino
24	Andreas Christensen	1727453620	Ejecutivo de Ventas	Tiempo Completo	Ventas	Ventas Matutino
25	Samuel Vega	1752815652	Ejecutivo de Ventas	Tiempo Completo	Compras	Ventas Matutino
26	Ferran Torres	1737475630	Asistente Administrativo	Tiempo Parcial	Administración	Ventas Matutino
27	Hector Fort	1789453420	Técnico de Soporte	Tiempo Parcial	Soporte Técnico	Ventas Matutino
28	Ronald Araujo	1743236760	Desarrollador Senior	Tiempo Completo	Ingeniero de Software	Sistemas Matutino
29	Luciana Sosa	1789563250	Jefa de Logística	Tiempo Completo	Logística	Ventas Vespertino

Figura 48

Pantalla registro de empleado

The screenshot shows the 'REGISTRO DE EMPLEADOS' (Employee Registration) form. The form fields are: Nombre (Luis), Apellido (Suárez), Cédula (1737584930), Cargo (Ejecutivo de Ventas), Tipo de Contrato (Tiempo Completo), Rol (Ventas), and Horario (Ventas Matutino). A dropdown menu for 'Horario' is open, showing options: Ventas Matutino, Ventas Vespertino, Marketing Matutino, Marketing Vespertino, and Sistemas Matutino. The 'Registrar' button is highlighted in blue, and an 'Atrás' button is also visible.

Figura 49

Notificación registro de empleado correcto

The screenshot shows a web application interface for employee registration. A dark sidebar on the left contains navigation items: ASISTENCIA, USUARIOS, EMPLEADOS (highlighted), CARGOS, and HORARIOS. The main content area is titled 'REGISTRO DE EMPLEADOS' and contains a form with the following fields: Nombre\*, Apellido\*, Cédula\*, Cargo..., Tipo de Contrato..., Rol..., and Horario... At the bottom right of the form are 'Atrás' and 'Registrar' buttons. A green notification box in the top right corner displays a checkmark icon, the text 'CORRECTO', and 'Empleado registrado correctamente'. The user's role 'Administrador' is visible in the top right corner.

Figura 50

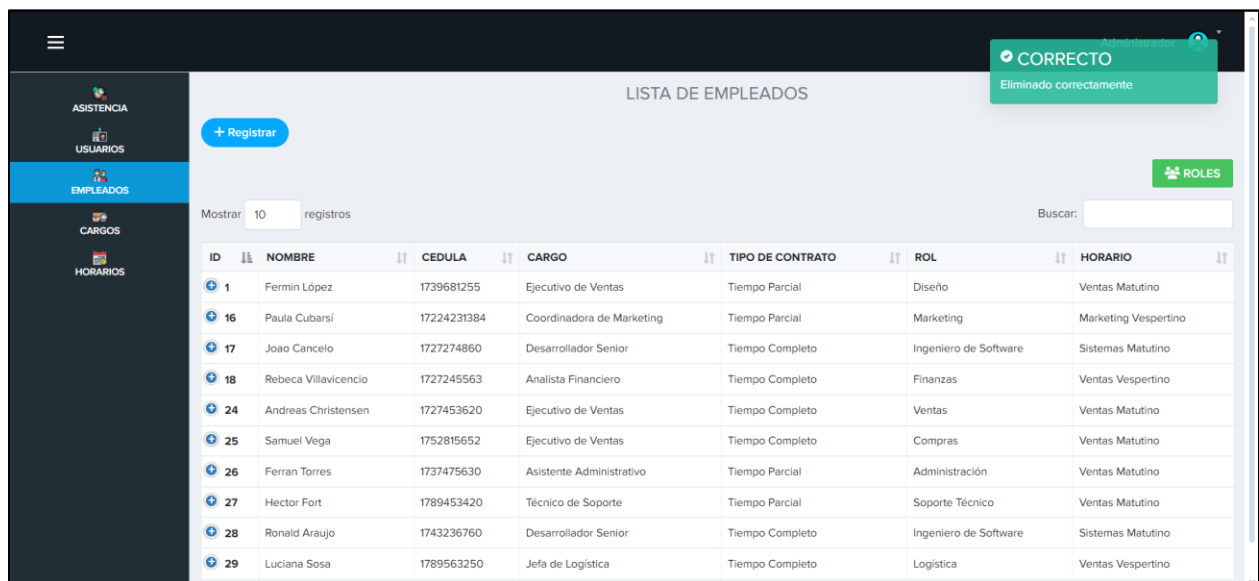
Pantalla eliminación de empleado

The screenshot shows the 'EMPLEADOS' list page. A confirmation dialog is centered on the screen, asking '¿Está seguro?' with the warning '¡ No podrá recuperar este registro !'. The dialog has two buttons: 'No, Salir' (red) and 'Si, Eliminar' (green). The background shows a table of employees with columns for ID, NOMBRE, CÉDULA, CARGO, TIEMPO, ROL, and HORARIO. A '+ Registrar' button is visible in the top left of the main area. The user's role 'Administrador' is in the top right. A search bar labeled 'ROLES' and 'Buscar:' is also present.

ID	NOMBRE	CÉDULA	CARGO	TIEMPO	ROL	HORARIO
30	Javier Hernández	1746374840	Responsable de Compras	Tiempo Completo	Ventas	Ventas Vespertino
31	Carlos Puyol	1745347820	Desarrollador Senior	Tiempo Completo	Ingeniero de Software	Sistemas Matutino
32	Luis Suárez	1737584930	Ejecutivo de Ventas	Tiempo Completo	Ventas	Ventas Matutino

Figura 51

Notificación eliminación de empleado correcta



### 4.3.2.2 GESTIÓN DE ROLES DE EMPLEADOS

Figura 52

Pantalla listado de roles de empleado

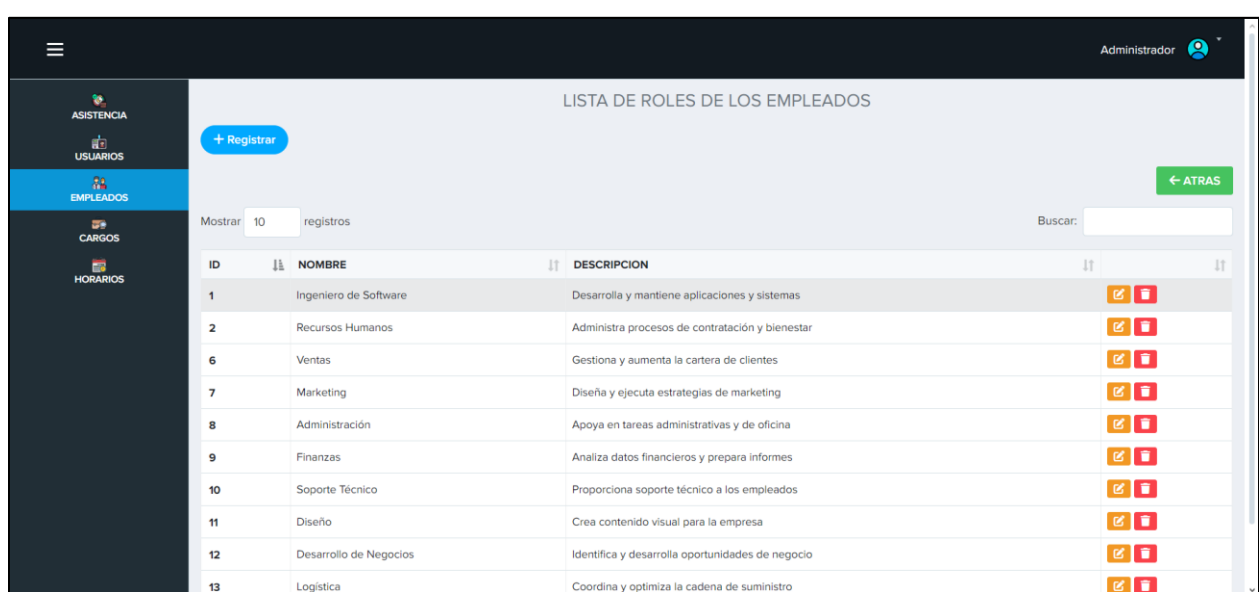


Figura 53

## Pantalla modificación de roles de empleado

Modificar Rol

Nombre  
Logística

Descripción  
Coordina y optimiza la cadena de suministro

Atrás Modificar

ID	NOMBRE	DESCRIPCION
1	Ingeniero de Software	Desarrolla y mantiene aplicaciones y sistemas
2	Recursos Humanos	Administra procesos de contratación y bienestar
6	Ventas	Gestiona y aumenta la cartera de clientes
7	Marketing	Diseña y ejecuta estrategias de marketing
8	Administración	Apoya en tareas administrativas y de oficina
9	Finanzas	Analiza datos financieros y prepara informes
10	Soporte Técnico	Proporciona soporte técnico a los empleados
11	Diseño	Crea contenido visual para la empresa
12	Desarrollo de Negocios	Identifica y desarrolla oportunidades de negocio
13	Logística	Coordina y optimiza la cadena de suministro

Figura 54

## Notificación modificación de roles de empleado correcta

LISTA DE ROLES DE LOS EMPLEADOS

+ Registrar

Mostrar 10 registros

Buscar:

CORRECTO  
El rol Logística se ha modificado correctamente

← ATRAS

ID	NOMBRE	DESCRIPCION
1	Ingeniero de Software	Desarrolla y mantiene aplicaciones y sistemas
2	Recursos Humanos	Administra procesos de contratación y bienestar
6	Ventas	Gestiona y aumenta la cartera de clientes
7	Marketing	Diseña y ejecuta estrategias de marketing
8	Administración	Apoya en tareas administrativas y de oficina
9	Finanzas	Analiza datos financieros y prepara informes
10	Soporte Técnico	Proporciona soporte técnico a los empleados
11	Diseño	Crea contenido visual para la empresa
12	Desarrollo de Negocios	Identifica y desarrolla oportunidades de negocio
13	Logística	Coordina y optimiza la cadena de suministro

Figura 55

Pantalla registro de roles de empleado

REGISTRO DE ROLES DE EMPLEADO

Nombre \*  
Rol de Empleado Prueba

Descripción \*  
Registro de prueba de Rol de Empleado

Atrás Registrar

Figura 56

Notificación registro de roles de empleado correcto

REGISTRO DE ROLES DE EMPLEADO

Nombre \*  
Rol de Empleado Prueba

Descripción \*  
Registro de prueba de Rol de Empleado

Atrás Registrar

**CORRECTO**  
El rol de empleado Rol de Empleado Prueba se ha registrado correctamente

Figura 57

Pantalla eliminación de roles de empleado

The screenshot shows a web application interface for managing employee roles. A central modal dialog box is displayed with a warning icon and the text: "¿Está seguro? ¡ No podrá recuperar este registro !". Below the text are two buttons: "No, Salir" (red) and "Si, Eliminar" (green). The background interface includes a sidebar with menu items: ASISTENCIA, USUARIOS, EMPLEADOS (highlighted), CARGOS, and HORARIOS. The main content area shows a table of roles with columns for ID, NOMBRE, and DESCRIPCION. The table lists roles 14, 15, and 17. A search bar and pagination controls are also visible.

ID	NOMBRE	DESCRIPCION
14	Calidad	Asegura la calidad de los productos y servicios
15	Compras	Gestiona las adquisiciones y proveedores
17	Rol de Empleado Prueba	Registro de prueba de Rol de Empleado

Figura 58

Notificación eliminación de roles de empleado correcto

The screenshot shows the same web application interface as Figure 57, but with a success notification. A green toast message in the top right corner reads: "CORRECTO Rol de empleado eliminado correctamente". The background interface is the same, showing the table of roles. The table lists roles 1 through 13. A search bar and pagination controls are also visible.

ID	NOMBRE	DESCRIPCION
1	Ingeniero de Software	Desarrolla y mantiene aplicaciones y sistemas
2	Recursos Humanos	Administra procesos de contratación y bienestar
6	Ventas	Gestiona y aumenta la cartera de clientes
7	Marketing	Diseña y ejecuta estrategias de marketing
8	Administración	Apoya en tareas administrativas y de oficina
9	Finanzas	Analiza datos financieros y prepara informes
10	Soporte Técnico	Proporciona soporte técnico a los empleados
11	Diseño	Crea contenido visual para la empresa
12	Desarrollo de Negocios	Identifica y desarrolla oportunidades de negocio
13	Logística	Coordina y optimiza la cadena de suministro

### 4.3.3 PRUEBAS Y RESULTADOS ITERACIÓN 3

#### **PRUEBAS DE USABILIDAD:**

**Objetivo:** Evaluar la facilidad de uso de las interfaces para la gestión de empleados y sus roles.

Para las pruebas de usabilidad se realizaron pruebas con diez usuarios gestionando empleados y sus roles. Se utilizaron herramientas de grabación de pantalla , se administraron cuestionarios de satisfacción post-prueba y el uso de la técnica “Thinking Aloud” para recolectar datos cualitativos y cuantitativos.

#### **Resultados:**

- **Positivos:** Las interfaces fueron consideradas intuitivas, con una navegación clara y eficiente.
- **Mejoras:** Se identificaron algunas mejoras en la organización de la información de los empleados, sugiriendo un diseño más modular y categorizado.

**Acciones Derivadas:** Tras recibir la retroalimentación de los usuarios, se implementó un filtro general en la visualización de la pantalla de empleados y roles de los empleados, permitiendo clasificar por el texto que escriba el usuario. Además, se optimizó la navegación en la interfaz de usuario reorganizando los menús y botones, donde se sugirió usar iconos en lugar de palabras textuales para la eliminación y edición, con la finalidad de tener funciones más accesibles. Estas mejoras resultaron en una experiencia de usuario más fluida y eficiente.

#### **PRUEBAS UNITARIAS:**

**Objetivo:** Validar cada componente individual de la gestión de empleados y roles.

En las pruebas unitarias se desarrollaron casos de prueba automatizados para la adición, edición y eliminación de empleados y roles, utilizando Junit, estas pruebas cubrieron todos los posibles escenarios de manejo de datos y situaciones de error.

**Resultados:**

Las pruebas unitarias confirmaron el correcto funcionamiento de cada componente individual. No se encontraron defectos significativos.

**PRUEBAS DE FUNCIONALIDAD:**

**Objetivo:** Asegurar que las funciones cumplan con los requisitos definidos.

Para medir la funcionalidad se validaron los escenarios de uso para la gestión de empleados y roles, utilizando casos de prueba que replicaban situaciones del mundo real, con esto se logró comparar los resultados obtenidos con los resultados esperados y revisión por pares.

**Resultados:**

Las funcionalidades cumplieron con los requisitos. Todas las pruebas de funcionalidad se completaron con éxito, permitiendo la correcta gestión de empleados y roles.

**PRUEBAS DE INTEGRACIÓN:**

**Objetivo:** Verificar la correcta interacción entre los módulos de gestión de empleados, roles y otros módulos del sistema.

En la integración se manejaron pruebas que involucraron la gestión de empleados en conjunto con la asignación de roles y cargos, utilizando un entorno de pruebas controlado gracias al uso de la herramienta Jenkins para la integración continua.

**Resultados:**

La integración fue exitosa. Los datos sobre empleados y roles se sincronizaron correctamente con los módulos de gestión de cargos y horarios, sin inconsistencias ni errores.

## **PRUEBAS DE RENDIMIENTO:**

**Objetivo:** Evaluar el rendimiento del sistema bajo la carga de múltiples operaciones de gestión de empleados.

Con las pruebas de rendimiento se realizó una simulación de múltiples operaciones de gestión realizadas simultáneamente utilizando JMeter. Se midieron tiempos de respuesta y utilización de recursos del servidor bajo diferentes cargas, teniendo como principales métricas los tiempos de respuesta, tasas de error y consumo de CPU/memoria.

### **Resultados:**

El sistema mostró un rendimiento adecuado con tiempos de respuesta rápidos y sin fallos bajo la carga simulada. No se identificaron cuellos de botella ni degradación significativa del rendimiento.

## **RESULTADOS OBTENIDOS**

La tercera iteración resultó en la implementación efectiva de la gestión de empleados y roles de empleado. Las pruebas confirmaron que el sistema era intuitivo, funcional y capaz de manejar múltiples operaciones administrativas sin problemas significativos. Las mejoras sugeridas en la organización de la información de los empleados se planificaron para la siguiente iteración.

### **4.4 ITERACIÓN 4 - DESARROLLO DE LA GESTIÓN DE ASISTENCIAS Y JUSTIFICACIONES**

La cuarta etapa del proyecto se centró en desarrollar la funcionalidad central del sistema: la gestión de asistencias y justificaciones. Esta funcionalidad incluye el registro y seguimiento de las asistencias de los empleados, la capacidad de marcar entradas y salidas, y la generación de justificaciones y su detalle en la asistencia.

Primero, se definieron los requisitos específicos para el registro y seguimiento de asistencias, asegurando que el sistema cumpliera con las necesidades de la organización. Se consideraron aspectos como la facilidad de uso, la precisión del registro y la exportación de la información.

Con base en estos requisitos, se diseñó una interfaz principal intuitiva y fácil de usar para que los empleados pudieran registrar sus asistencias de manera rápida y eficiente. La interfaz se elaboró cuidadosamente para minimizar el tiempo y esfuerzo necesarios para registrar la entrada y salida del trabajo. En el controlador del sistema, se implementaron las funcionalidades necesarias para almacenar, gestionar y proteger los registros de asistencia.

Adicionalmente, se incluyó la funcionalidad de exportar la información sea completa o filtrada en un archivo “.xls”. Estas herramientas permiten a los usuarios exportar información valiosa sobre las tendencias de asistencia, identificar posibles problemas y tomar decisiones informadas para la gestión de recursos humanos.

Para garantizar el correcto funcionamiento del sistema, se realizaron pruebas exhaustivas durante esta iteración. Se verificó que todas las partes del sistema, desde la interfaz de usuario hasta el almacenamiento de datos, funcionaran de manera cohesiva y sin errores.

Finalmente, se llevó a cabo una revisión completa del sistema con los usuarios para verificar que cumpliera con los requisitos establecidos y proporcionara las funcionalidades esperadas. Se recogieron comentarios y sugerencias valiosos para realizar los ajustes necesarios y optimizar el sistema según el alcance establecido en el proyecto de titulación.

#### 4.4.1 TABLA DE ACTIVIDADES ITERACIÓN 4

<b>Funcionalidad</b>	<b>Descripción</b>	<b>Actor</b>	<b>Criterio de Aceptación</b>
<b>Diseño de interfaz para la ventana de Asistencias y Justificaciones de los empleados.</b>	El empleado contará con una interfaz intuitiva para ingresar su asistencia de entrada y salida y de ser necesario una justificación.	Empleado	Interfaces intuitivas con validaciones y notificaciones para el correcto ingreso de la asistencia y generación de las justificaciones.

<b>Visualización de Asistencias y Justificaciones de los empleados.</b>	El usuario podrá visualizar todas las asistencias y justificaciones creadas.	Usuario	Visualizar los asistencias de los empleados dentro de su vista y de ser necesario generar y editar su justificación.
<b>Creación de las Asistencias de los empleados.</b>	El empleado tiene la potestad de ingresar su asistencia de entrada y de salida diariamente, y el usuario de generar justificaciones de ser necesarias.	Empleado	Verificar los registros de asistencia y justificaciones que han sido añadidos.

Tabla 11. Tabla actividades Iteración 4

*Elaboración propia.*

## 4.4.2 CAPTURAS DE PANTALLA ITERACIÓN 4

### 4.4.2.1 GESTIÓN DE ASISTENCIAS

Figura 59

Pantalla registro de asistencia del empleado



Figura 60

Notificación registro de entrada



Figura 61

Notificación registro de salida



Figura 62

Pantalla de listado de asistencias

ASISTENCIA DE EMPLEADOS

Todos los empleados

ID	EMPLEADO	C.I.	DÍA	DESCRIPCION	HORARIO	HORAS	TIMBRE	TOTAL	OBSERVACIÓN	ESTADO	
95	Carlos Puyol	1745347820	Viernes	Sistemas Matutino	07:00:00 - 16:00:00	09:00	2024-06-07 08:00:00 - 2024-06-07 17:45:00	09:45:00	Correcto		
84	Javier Hernández	1746374840	Viernes	Ventas Vespertino	09:00:00 - 14:00:00	05:00	2024-06-07 07:30:00 - 2024-06-07 17:30:00	10:00:00	Correcto		
83	Luciana Sosa	1789563250	Jueves	Ventas Vespertino	15:00:00 - 21:00:00	06:00	2024-06-06 09:00:00 - 2024-06-06 18:00:00	09:00:00	Correcto		
104	Javier Hernández	1746374840	Jueves	Ventas Vespertino	15:00:00 - 21:00:00	06:00	2024-06-06 07:30:00 - 2024-06-06 16:45:00	09:15:00	Correcto		
82	Ronald Araujo	1743236760	Jueves	Sistemas Matutino	07:00:00 - 18:00:00	11:00	2024-06-06 07:00:00 - 2024-06-06 16:30:00	09:30:00	Incorrecto	Justificado	
93	Luciana Sosa	1789563250	Miércoles	Ventas Vespertino	15:00:00 - 21:00:00	06:00	2024-06-05 09:00:00 - 2024-06-05 18:15:00	09:15:00	Correcto		
81	Hector Fort	1789453420	Miércoles	Ventas Matutino	08:30:00 - 16:30:00	08:00	2024-06-05 08:45:00 - 2024-06-05 17:45:00	09:00:00	Correcto		
80	Ferran Tones	1737475630	Miércoles	Ventas Matutino	08:30:00 - 16:30:00	08:00	2024-06-05 07:15:00 - 2024-06-05 16:45:00	09:30:00	Correcto		
79	Samuel Vega	1752815652	Martes	Ventas Matutino	08:00:00 - 17:00:00	09:00	2024-06-04 08:00:00 - 2024-06-04 18:30:00	10:30:00	Correcto		
92	Ronald Araujo	1743236760	Martes	Sistemas Matutino	07:00:00 - 17:00:00	10:00	2024-06-04 07:00:00 - 2024-06-04 16:45:00	09:45:00	Incorrecto	Injustificado	

1 2 3 Siguierte

Figura 63  
Pantalla de eliminación de asistencias

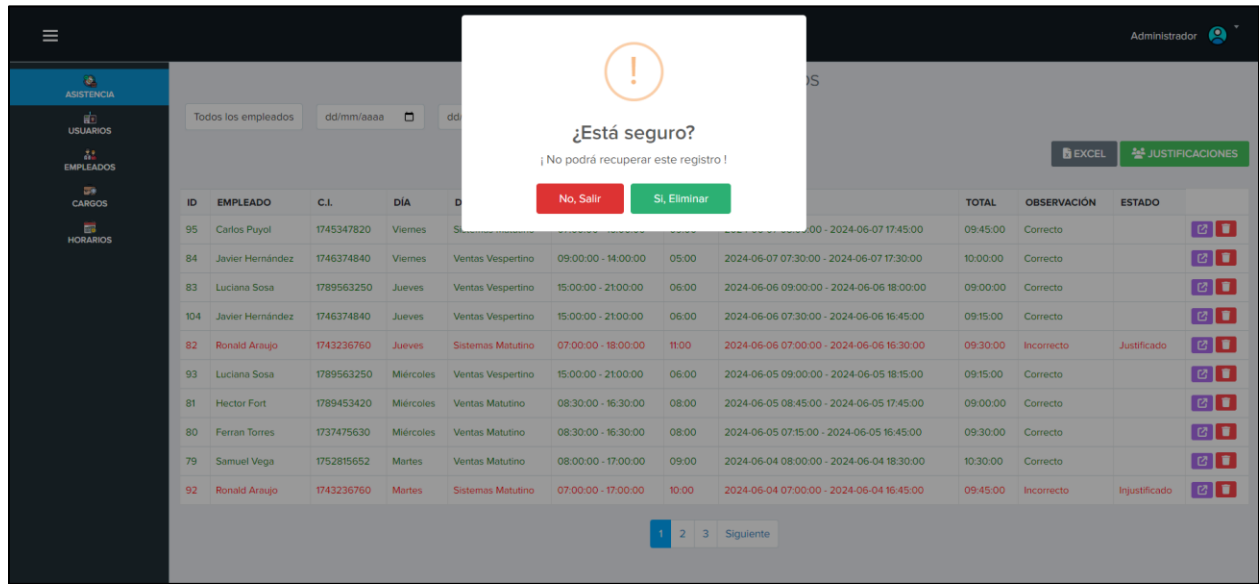
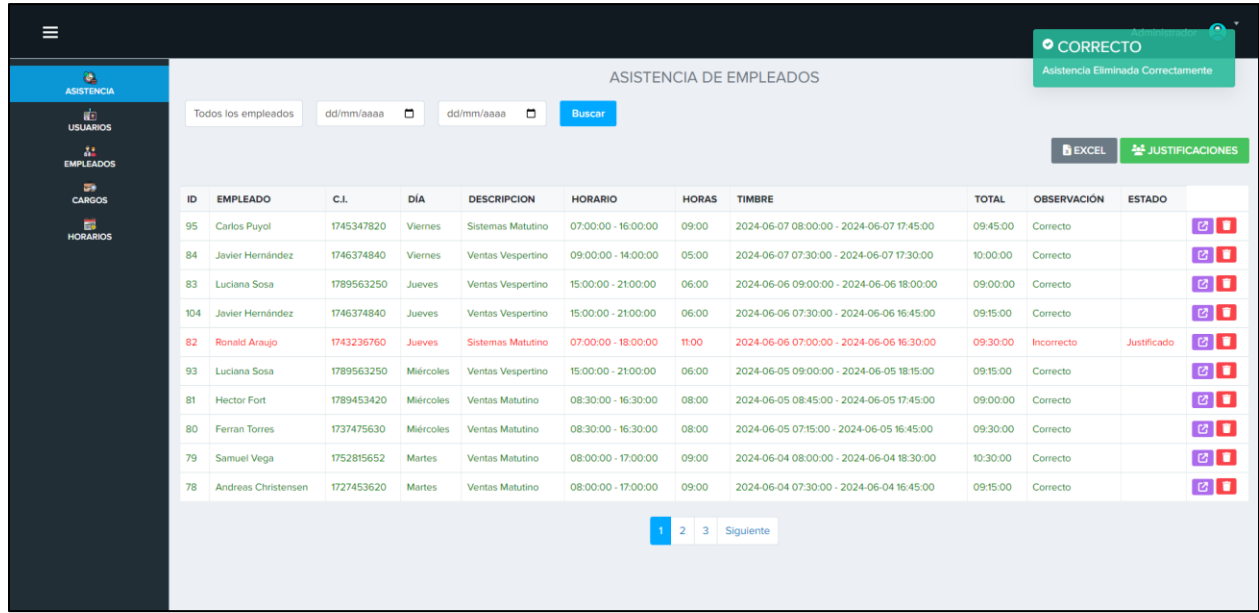


Figura 64  
Notificación eliminación de asistencias correcta



## 4.4.2.2 GESTIÓN DE JUSTIFICACIONES

Figura 65

Pantalla de listado de justificaciones

Administrador

LISTA DE JUSTIFICACIONES

Todos los empleados dd/mm/aaaa dd/mm/aaaa Buscar

Mostrar 10 registros Buscar:

ID	EMPLEADO	ASISTENCIA	FECHA	ESTADO	MOTIVO	COMENTARIO
29	Ronald Araujo	2024-06-06 07:00:00 - 2024-06-06 16:30:00	2024-06-13	Justificado	Cita Médica	Intoxicación estomacal
32	Ronald Araujo	2024-06-04 07:00:00 - 2024-06-04 16:45:00	2024-06-13	Injustificado	Sin Justificación	Salió temprano sin autorización
35	Joao Cancelo	2024-05-31 07:05:14 - 2024-05-31 12:05:14	2024-06-13	Justificado	Permiso por situación familiar	Permiso autorizado
37	Joao Cancelo	2024-05-29 07:00:00 - 2024-05-29 16:45:00	2024-06-13			

Registros del 1 al 4 de 4 registros

Anterior 1 Siguiente

← ATRAS

Figura 66

Pantalla de modificación de justificaciones

Administrador

Modificar Justificación

Nombre: Joao Cancelo

Entrada: 2024-05-29 07:00:00 - 2024-05-29 16:45:00

Fecha: 13/06/2024

Estado: Justificado

Motivo: Cita Médica

Comentario: Migraña Fuerte

Atrás Modificar

← ATRAS

Anterior 1 Siguiente

Figura 67

Notificación modificación de justificaciones correcta

The screenshot shows the 'LISTA DE JUSTIFICACIONES' (Justification List) interface. A green notification box at the top right states 'CORRECTO La justificación se ha modificado correctamente'. The interface includes a search bar with filters for 'Todos los empleados', date pickers, and a 'Buscar' button. Below the search bar, it indicates 'Mostrar 10 registros' and a search input field. The main content is a table with the following data:

ID	EMPLEADO	ASISTENCIA	FECHA	ESTADO	MOTIVO	COMENTARIO	
29	Ronald Araujo	2024-06-06 07:00:00 - 2024-06-06 16:30:00	2024-06-13	Justificado	Cita Médica	Intoxicación estomacal	
32	Ronald Araujo	2024-06-04 07:00:00 - 2024-06-04 16:45:00	2024-06-13	Injustificado	Sin Justificación	Salió temprano sin autorización	
35	Joao Cancelo	2024-05-31 07:05:14 - 2024-05-31 12:05:14	2024-06-13	Justificado	Permiso por situación familiar	Permiso autorizado	
37	Joao Cancelo	2024-05-29 07:00:00 - 2024-05-29 16:45:00	2024-06-13	Justificado	Cita Médica	Migraña Fuerte	

At the bottom of the table, it says 'Registros del 1 al 4 de 4 registros' and includes navigation buttons for 'Anterior' and 'Siguiente'.

Figura 68

Pantalla listado de asistencias, caso estado justificado

The screenshot shows the 'ASISTENCIA DE EMPLEADOS' (Employee Attendance) interface. It features a search bar with filters for 'Todos los empleados', date pickers, and a 'Buscar' button. There are buttons for 'EXCEL' and 'JUSTIFICACIONES'. The main content is a table with the following data:

ID	EMPLEADO	C.I.	DÍA	DESCRIPCION	HORARIO	HORAS	TIMBRE	TOTAL	OBSERVACIÓN	ESTADO	
72	Ferran Torres	1737475630	Jueves	Ventas Matutino	09:00:00 - 17:00:00	08:00	2024-05-30 07:15:00 - 2024-05-30 17:30:00	10:15:00	Correcto		
71	Samuel Vega	1752815652	Miércoles	Ventas Matutino	08:30:00 - 16:30:00	08:00	2024-05-29 08:00:00 - 2024-05-29 18:45:00	10:45:00	Correcto		
70	Andreas Christensen	1727453620	Miércoles	Ventas Matutino	08:30:00 - 16:30:00	08:00	2024-05-29 07:30:00 - 2024-05-29 17:00:00	09:30:00	Correcto		
88	Joao Cancelo	1727274860	Miércoles	Sistemas Matutino	07:00:00 - 18:00:00	11:00	2024-05-29 07:00:00 - 2024-05-29 16:45:00	09:45:00	Incorrecto	Justificado	
69	Rebeca Villavicencio	1727245563	Martes	Ventas Vespertino	16:00:00 - 22:00:00	06:00	2024-05-28 09:45:00 - 2024-05-28 19:30:00	09:45:00	Correcto		
68	Joao Cancelo	1727274860	Martes	Sistemas Matutino	07:00:00 - 17:00:00	10:00	2024-05-28 07:00:00 - 2024-05-28 17:15:00	10:15:00	Correcto		
87	Paula Cubarsi	17224231384	Martes	Marketing Vespertino	14:00:00 - 20:00:00	06:00	2024-05-28 08:30:00 - 2024-05-28 17:15:00	08:45:00	Correcto		
67	Paula Cubarsi	17224231384	Lunes	Marketing Vespertino	14:00:00 - 20:00:00	06:00	2024-05-27 08:30:00 - 2024-05-27 16:00:00	07:30:00	Correcto		
86	Fermin López	1739681255	Lunes	Ventas Matutino	08:00:00 - 16:00:00	08:00	2024-05-27 07:15:00 - 2024-05-27 16:00:00	08:45:00	Correcto		
66	Andreas Christensen	1727453620	Lunes	Ventas Matutino	08:00:00 - 16:00:00	08:00	2024-05-27 07:15:00 - 2024-05-27 15:45:00	08:30:00	Correcto		

At the bottom of the table, it says 'Anterior 1 2 3' with the number 3 highlighted.

Figura 69

Pantalla listado de asistencias, caso exportación de datos en formato .csv

The screenshot shows a web application interface for employee attendance. At the top, there's a navigation menu on the left with options like 'ASISTENCIA', 'USUARIOS', 'EMPLEADOS', 'CARGOS', and 'HORARIOS'. The main area is titled 'ASISTENCIA DE EMPLEADO'. Below the title, there are search filters for 'Todos los empleados', date ranges, and a 'Buscar' button. A notification in the top right corner says 'asistencia.csv Descarga finalizada'. Below the filters is a table with columns: ID, EMPLEADO, C.I., DÍA, DESCRIPCION, HORARIO, HORAS, TIMBRE, TOTAL, OBSERVACIÓN, and ESTADO. The table lists several employees with their respective attendance details. At the bottom of the table, there are pagination controls showing '1 2 3 Siguinte'.

Figura 70

Archivo .csv exportado

ID	EMPLEADO	C.I.	DÍA	DESCRIPCION	HORARIO	HORAS	TIMBRE	TOTAL	OBSERVACIÓN	ESTADO
95	Carlos Puyol	1745347820	Viernes	Sistemas Matutino	07:00:00 - 16:00:00	09:00	2024-06-07 08:00:00 - 2024-06-07 17:45:00	09:45:00	Correcto	
84	Javier Hernández	1746374840	Viernes	Ventas Vespertino	09:00:00 - 14:00:00	05:00	2024-06-07 07:30:00 - 2024-06-07 17:30:00	10:00:00	Correcto	
83	Luciana Sosa	1789563250	Jueves	Ventas Vespertino	15:00:00 - 21:00:00	06:00	2024-06-06 09:00:00 - 2024-06-06 18:00:00	09:00:00	Correcto	
104	Javier Hernández	1746374840	Jueves	Ventas Vespertino	15:00:00 - 21:00:00	06:00	2024-06-06 07:30:00 - 2024-06-06 16:45:00	09:15:00	Correcto	
82	Ronald Araujo	1743236760	Jueves	Sistemas Matutino	07:00:00 - 18:00:00	11:00	2024-06-06 07:00:00 - 2024-06-06 16:30:00	09:30:00	Incorrecto	Justificado
93	Luciana Sosa	1789563250	Miércoles	Ventas Vespertino	15:00:00 - 21:00:00	06:00	2024-06-05 09:00:00 - 2024-06-05 18:15:00	09:15:00	Correcto	
81	Hector Fort	1789453420	Miércoles	Ventas Matutino	08:30:00 - 16:30:00	08:00	2024-06-05 08:45:00 - 2024-06-05 17:45:00	09:00:00	Correcto	
80	Ferran Torres	1737475630	Miércoles	Ventas Matutino	08:30:00 - 16:30:00	08:00	2024-06-05 07:15:00 - 2024-06-05 16:45:00	09:30:00	Correcto	
79	Samuel Vega	1752815652	Martes	Ventas Matutino	08:00:00 - 17:00:00	09:00	2024-06-04 08:00:00 - 2024-06-04 18:30:00	10:30:00	Correcto	
92	Ronald Araujo	1743236760	Martes	Sistemas Matutino	07:00:00 - 17:00:00	10:00	2024-06-04 07:00:00 - 2024-06-04 16:45:00	09:45:00	Incorrecto	Injustificado
77	Rebeca Villavicencio	1727245563	Lunes	Ventas Vespertino	14:00:00 - 20:00:00	06:00	2024-06-03 09:45:00 - 2024-06-03 19:15:00	09:30:00	Correcto	
91	Hector Fort	1789453420	Lunes	Ventas Matutino	08:00:00 - 16:00:00	08:00	2024-06-03 08:45:00 - 2024-06-03 17:30:00	08:45:00	Correcto	
59	Joao Cancelo	1727274860	Lunes	Sistemas Matutino	07:00:00 - 16:00:00	09:00	2024-06-03 07:00:00 - 2024-06-03 17:00:00	10:00:00	Correcto	
75	Luciana Sosa	1789563250	Viernes	Ventas Vespertino	09:00:00 - 14:00:00	05:00	2024-05-31 09:00:00 - 2024-05-31 19:00:00	10:00:00	Correcto	
74	Ronald Araujo	1743236760	Viernes	Sistemas Matutino	07:00:00 - 16:00:00	09:00	2024-05-31 07:00:00 - 2024-05-31 16:45:00	09:45:00	Correcto	
90	Andreas Christensen	1727453620	Viernes	Ventas Matutino	08:00:00 - 15:00:00	07:00	2024-05-31 07:30:00 - 2024-05-31 16:00:00	08:30:00	Correcto	

### 4.4.3 PRUEBAS Y RESULTADOS ITERACIÓN 4

#### **PRUEBAS DE USABILIDAD:**

**Objetivo:** Evaluar la facilidad de uso de las interfaces para la gestión de asistencias y justificaciones.

Para las pruebas de usabilidad se realizaron pruebas con diez usuarios utilizando las funciones de registro de asistencia y justificaciones. Se utilizaron herramientas de grabación de pantalla, se administraron cuestionarios de satisfacción post-prueba y el uso de la técnica “Thinking Aloud” para recolectar datos cualitativos y cuantitativos.

#### **Resultados:**

- **Positivos:** Las interfaces fueron consideradas intuitivas y fáciles de usar.
- **Mejoras:** Los usuarios sugirieron mejoras en la visualización de justificaciones, como la adición de filtros extra a los presentados.

**Acciones Derivadas:** Tras recibir la retroalimentación de los usuarios, se implementaron filtros adicionales en la visualización de la pantalla de asistencia y justificaciones, permitiendo clasificar por fecha y empleado. Además, se optimizó la navegación en la interfaz de usuario reorganizando los menús y botones, donde se sugirió añadir un color más llamativo al botón de generación de justificaciones, con la finalidad de tener funciones más accesibles. Estas mejoras resultaron en una experiencia de usuario más fluida y eficiente.

#### **PRUEBAS UNITARIAS:**

**Objetivo:** Verificar cada componente individual de la gestión de asistencias y justificaciones.

En las pruebas unitarias se desarrollaron casos de prueba automatizados para el registro de entrada y salida, así como para la creación y modificación de justificaciones, utilizando JUnit. Las pruebas cubrieron todas las posibles combinaciones de datos de entrada y situaciones de error.

**Resultados:**

Todas las funciones individuales pasaron las pruebas unitarias, asegurando su correcto funcionamiento. No se encontraron defectos significativos.

**PRUEBAS DE FUNCIONALIDAD:**

**Objetivo:** Asegurar que las funciones cumplan con los requisitos especificados.

Para la funcionalidad se aseguró la validación de los escenarios de uso para el registro de asistencias y la gestión de justificaciones, utilizando casos de prueba que replicaban situaciones del mundo real, teniendo así una comparación de los resultados obtenidos con los resultados esperados y revisión por pares.

**Resultados:**

Las funcionalidades operaron conforme a los requisitos. Todas las pruebas de funcionalidad se completaron con éxito, permitiendo el registro preciso de asistencias y la gestión adecuada de justificaciones.

**PRUEBAS DE INTEGRACIÓN:**

**Objetivo:** Comprobar la interacción correcta entre los módulos de gestión de asistencias, justificaciones y otros módulos del sistema.

Para la integración se realizó pruebas que implicaban el registro de asistencias y la justificación de ausencias, integrándose con la gestión de empleados y horarios, utilizando un entorno de pruebas controlado con el uso de la herramienta Jenkins.

**Resultados:**

La integración fue exitosa. Los datos sobre asistencias y justificaciones se sincronizaron correctamente con los módulos de gestión de empleados y horarios, sin inconsistencias ni errores.

### **PRUEBAS DE RENDIMIENTO:**

**Objetivo:** Evaluar el rendimiento del sistema bajo la carga de múltiples operaciones de gestión de asistencias y justificaciones.

Para medir el rendimiento se realizó una simulación de múltiples registros de asistencia y justificaciones realizadas simultáneamente utilizando JMeter. Se midieron tiempos de respuesta y utilización de recursos del servidor bajo diferentes cargas, teniendo como métricas principales los tiempos de respuesta, tasas de error y consumo de CPU/memoria.

### **Resultados:**

El sistema mantuvo un rendimiento adecuado con tiempos de respuesta rápidos y sin fallos bajo la carga simulada. No se identificaron cuellos de botella ni degradación significativa del rendimiento.

### **RESULTADOS OBTENIDOS**

La cuarta iteración culminó con la implementación exitosa de la gestión de asistencias y justificaciones. Las pruebas confirmaron que el sistema es completamente funcional, fácil de usar y capaz de manejar múltiples registros y justificaciones sin problemas significativos. Además, se identificaron áreas para mejoras menores en la visualización de justificaciones, las cuales se abordaron de manera efectiva, asegurando así una experiencia de usuario optimizada y un rendimiento robusto del sistema. Esta fase final demuestra la solidez y fiabilidad del sistema desarrollado, cumpliendo con todos los objetivos planteados y garantizando su operatividad en entornos reales.

# CAPÍTULO V

## CONCLUSIONES Y RECOMENDACIONES

### 5. CONCLUSIONES

La elección de la metodología de prototipado fue crítica para el éxito del proyecto, gracias a su ciclo de iteración rápida y la incorporación de retroalimentación continua. Esta metodología no solo facilitó la adaptación a cambios y mejoras constantes, sino que también mejoró significativamente las habilidades técnicas en gestión de proyectos y desarrollo ágil. La capacidad de ajustarse rápidamente a los desafíos y a las necesidades del usuario fue fortalecida, demostrando la importancia de la flexibilidad y la adaptabilidad en el desarrollo de sistemas de la información.

El uso de PHP como lenguaje de programación fue vital para lograr una adecuada flexibilidad e integración con las demás herramientas del proyecto. La decisión de desarrollar el proyecto de titulación en PHP permitió una integración fluida con XAMPP y el sistema gestor de bases de datos MySQL, simplificando así el control de los datos y la implementación de la arquitectura MVC, alcanzando una programación cómoda y organizada que cumplió con los tiempos establecidos para cada iteración propuesta con la metodología del prototipado.

La creación de casos de uso basados en los requerimientos funcionales fue fundamental para el desarrollo del sistema, ya que los casos de uso permitieron identificar con precisión las necesidades y expectativas del usuario final, lo cual fue esencial para garantizar que el proyecto cumpliera con todos los requisitos establecidos. De esta forma se aseguró que cada aspecto del proyecto estuviera alineado con los objetivos y expectativas de los usuarios, resultando en un sistema de información funcional, confiable y capaz de satisfacer las demandas del usuario final en su entorno operativo.

La realización del proyecto logró consolidar conocimientos teóricos, mediante su aplicación en un contexto práctico. La revisión de diversas tecnologías y mejores prácticas enriqueció el enfoque del proyecto, asegurando que se adoptaran las soluciones más efectivas y adecuadas.

## 6. RECOMENDACIONES

Es recomendable mantener un constante monitoreo y evaluación para identificar áreas de mejora y ajustar el sistema según las necesidades cambiantes de la organización. Esto incluye la recolección de retroalimentación, el análisis de métricas de desempeño y la implementación de mejoras tecnológicas y funcionales periódicas. Un enfoque proactivo en la mejora continua garantiza que el sistema se mantenga relevante, eficiente y alineado con los objetivos estratégicos de la organización.

Se recomienda optimizar la identificación de los empleados mediante funcionalidades avanzadas, como el uso de huellas dactilares o el reconocimiento facial. Esta mejora puede implementarse utilizando sensores especializados para la detección de huellas dactilares y reconocimiento facial. La incorporación de estos sensores mejorará significativamente la experiencia del usuario, además de reducir el tiempo necesario para registrar la entrada y salida de los empleados. Al automatizar este proceso, se minimiza la posibilidad de errores humanos y se agiliza la gestión del personal. Además, la implementación de estas tecnologías contribuirá a aumentar la seguridad, ya que se reduce el riesgo de fraude o suplantación de identidad.

A corto plazo, se recomienda optimizar las interfaces de usuario basándose en la retroalimentación continua proporcionada por los empleados y los usuarios que estén probando el sistema de manera constante. A mediano plazo, se sugiere explorar la integración de nuevas funcionalidades, como la gestión avanzada de permisos y roles, así como la adición de características que beneficien otras áreas además de Recursos Humanos. Por ejemplo, incluir funcionalidades en el área de nómina para dar soporte a la compensación salarial basada en las horas trabajadas y gestionar horas extras de manera eficiente.

## BIBLIOGRAFÍA

- Almaraz, A. (2023). *The Impact of Automated Attendance Systems on Academic Performance in Higher Education*. Journal of College and University Teaching.
- Alshammari, N. (2021). *The Impact of Automated Attendance Systems on Student Engagement and Performance in Higher Education*. International Journal of Technology in Education and Science.
- Apache Friends. (s.f.). *XAMPP: Apache Friends*. . <https://www.apachefriends.org/index.html>.
- Bass, L., Clements, P., & Kazman, R. (2012). *Architecture of Software Systems: The Essentials (3rd ed.)*. Addison-Wesley.
- Bellemare, A. (2014). *Building Event-Driven Microservices: Leveraging Organizational Data at Scale*. O'Reilly Media.
- Black, A. (2020). *User-Centered Design: A Practical Guide*. Wiley.
- Brown, R. (2018). *Prototyping for Designers: Developing the Best Digital and Physical Products*. O'Reilly Media.
- Cockburn, A. (2001). *Writing effective use cases*. Addison-Wesley.
- Cooper, A. (2014). *About Face: The Essentials of User Interface Design (4th ed.)*. Wiley.
- Davenport, T. H. (2018). *Process Automation: Putting Bots to Work*. Harvard Business Review Press.
- Deitel, P. J., & Deitel, H. (2023). *Java: How to Program (20th Anniversary Edition)*. Pearson Education.
- Fowler, M. (2006). *Patterns of Enterprise Application Architecture (1st ed.)*. Addison-Wesley.
- García, S. (2018). *Improving Employee Attendance Tracking Through Web-Based Systems*. Tourism Management Perspectives.
- Green, S. (2019). *Design Thinking: A Practical Guide*. Routledge.
- Harvard Business Review. (2022). *How Accurate Attendance Tracking Drives Strategic Decisions*. Harvard Business Review Digital Articles.
- Hassan, S. (2020). *A Web-Based Attendance Management System for Higher Education Institutions*. [https://www.researchgate.net/publication/303023739\\_Web-based\\_Attendance\\_Management\\_System](https://www.researchgate.net/publication/303023739_Web-based_Attendance_Management_System): Gomal University.
- Jhonson, T. (2020). *Agile Prototyping: A Practical Guide*. Packt Publishing.
- Kendall, K. E., & Kendall, J. E. (2014). *Systems Analysis and Design (9th ed.)*. Pearson.
- Laudon, K. C., & Laudon, J. P. (2020). *Management information systems: Managing digital organizations in a global economy (19th ed.)*. Pearson Education.
- Mathis, R. L., & Jackson, J. H. (2011). *Administración de recursos humanos (14a ed.)*. Cengage Learning.
- McKinsey & Company. (2019). *The Future of Work in America: People and Places, Today and Tomorrow*. <https://www.mckinsey.com/featured-insights/future-of-work/the-future-of-work-in-america-people-and-places-today-and-tomorrow>.

- Meyer, J., & Allen, M. (1911). *Commitment to the organization and organizational citizenship behaviors: A theoretical and empirical review*. Journal of management.
- Microsoft. (s.f.). *Visual Studio*. <https://visualstudio.microsoft.com/>.
- Microsoft. (s.f.). *Microsoft SQL Server*. <https://www.microsoft.com/es-es/sql-server/sql-server-2022>.
- MySQL. (s.f.). *MySQL: The world's most popular open source database*. <https://www.mysql.com>.
- Nielsen, J. (2000). *Designing web usability: The practice of simplifying truth*. New Riders Press.
- Organización Internacional del Trabajo. (2019). *Digitalización, plataformas y el futuro del trabajo: Desafíos y oportunidades*. <https://www.ilo.org>.
- PostgreSQL Global Development Group. (s.f.). *PostgreSQL: The Database for Modern Applications*. <https://www.postgresql.org/>.
- Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach (8th ed.)*. McGraw-Hill Education.
- Python Software Foundation. (s.f.). *The Python Programming Language*. <https://www.python.org/>.
- Red, E. (2022). *User Experience Design: Creating Designs Users Really Love*. O'Reilly Media.
- Rjeib, H. D. (2018). *Attendance and Information System using RFID and*. International Journal of Advanced Computer Science and Applications.
- Robbins, J. N. (2012). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. O'Reilly Media, Inc.
- Schwaber, K. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Scrum.org.
- Smith, A. (2018). *Optimizing Workforce Management through Automated Attendance Systems*. Journal of Business Efficiency.
- Snyder, C. (2003). *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann.
- Sommerville, I. (2011). *Software engineering (9th ed.)*. Pearson Education.
- Sommerville, I. (2015). *Software Engineering (10th ed.)*. Pearson.
- Stallings, W., & Brown, L. (2014). *Information Security: Principles and Practices (6th ed.)*. Pearson.
- Taylor, F. W. (1911). *The principles of scientific management*. Harper & Brothers.
- White, L. (2021). *Lean UX: Designing Great Products with Agile Teams*. O'Reilly Media.
- Whitten, J., Bentley, L., & Dittman, K. (2007). *Systems Analysis and Design (8th ed.)*. McGraw-Hill.
- Wilson, K. (2015). *The Benefits of Web-Based Attendance Systems*. Education and Information Technologies.
- Zandstra, M. (2021). *PHP 8 Objects, Patterns, and Practice*. O'Reilly Media.

