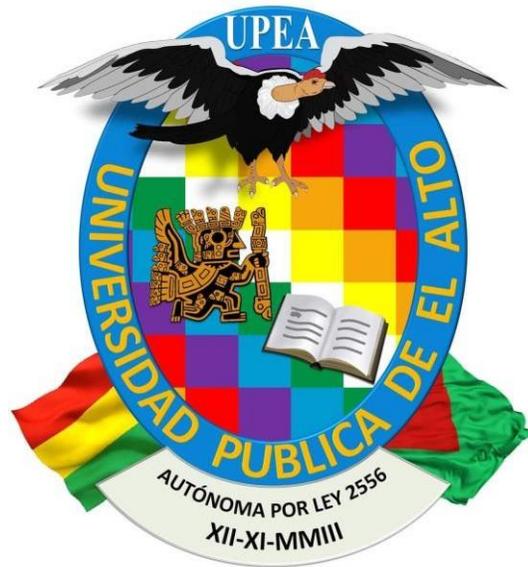


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE ACCESO
Y ASISTENCIA PARA EL PROCESO DE PLANILLAS MEDIANTE RFID”
CASO: ANTI MONEY LAUNDERING CONSULTANTS**

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Eddy Zeballos Quispe
Tutor Metodológico: M.Sc. Ing. Enrique Flores Baltazar
Tutor Revisor: Ing. Hernán Quispe Conurana
Tutor Especialista: Ing. Rodolfo Flores Cutile

EL ALTO – BOLIVIA

2022

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo **Eddy Zeballos Quispe** con **C.I. 6067034 LP.** y **R.U. 5336** mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE GRADO** titulada “**SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE ACCESO Y ASISTENCIA PARA EL PROCESO DE PLANILLAS MEDIANTE RFID**” es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados, Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, 21 noviembre del 2022.

.....

Eddy Zeballos Quispe

C.I. 6067034

zeballos.iglux@gmail.com

DEDICATORIA

Dedico el presente proyecto: A Dios por ser el inspirador en cada uno de mis pasos, por haberme dado la vida y fortaleza para con el presente proyecto.

A mis padres Cecilio y Severina por ser la guía en el sendero de cada acto que realizo, y por su apoyo incondicional.

AGRADECIMIENTOS

Agradezco profundamente a Dios, por guiarme en el sendero correcto de la vida, por ayudarme en todo lo que realizo en mi diario convivir.

A mis padres, por ser mi ejemplo para seguir adelante y por inculcarme valores, los cuales siempre los tendré presente.

A mi amigo de toda la vida Mijaíl Hinojosa (Longinus), por el apoyo incondicional.

Agradecer a mis distinguidos tutores:

A mi tutor metodológico Ing. Enrique Flores Baltazar, por su conocimiento, apoyo, tiempo y motivación brindada a mi persona.

A mi tutor revisor Ing. Hernán Quispe Conurana, por su acertada orientación y observaciones brindadas en la realización del presente proyecto.

A mi tutor especialista Ing. Rodolfo Flores Cutile, por su apoyo incondicional y por compartirme sus conocimientos y experiencias, durante el desarrollo del presente proyecto.

A la Universidad Pública de El Alto, a la carrera Ingeniería de Sistemas por acogerme en sus aulas en los años de estudio.

RESUMEN

En la actualidad, el área de recursos humanos de la empresa Anti Money Laundering Consultants A.M.L.C no cuenta con un software que le permita realizar las planillas, por tanto, su proceso es muy tedioso y nada eficiente. Además de no poseer un control de acceso para los empleados en las diferentes áreas.

Por lo cual se propuso elaborar un sistema de información para coadyuvar en el proceso de elaboración de las planillas y control de accesos con la finalidad de poder automatizar los procesos que se realizan en el área de recursos humanos.

Para todo eso se planteó métodos y metodologías para el proceso de diseño del sistema como se plantea herramientas de desarrollo.

En los siguientes capítulos desglosaremos todo los procesos y conceptos que se empleó para la elaboración del sistema y la documentación.

Palabras clave: sueldo, sistema, acceso, control, asistencia, Rfid.

SUMMARY

At present, the human resources area of the company Anti Money Laundering Consultants A.M.L.C. does not have a software that allows it to make payrolls, therefore, its process is very tedious and not efficient at all. In addition to not having an access control for employees in different areas.

Therefore, it was proposed to develop an information system to assist in the process of preparing payroll and access control in order to automate the processes carried out in the area of human resources.

For all this, methods and methodologies were proposed for the system design process as well as development tools.

In the following chapters, we will describe all the processes and concepts that were used for the development of the system and the documentation.

Key words: salary, system, access, control, attendance, Rfid.

ÍNDICE GENERAL

1	MARCO PRELIMINAR	1
1.1	Introducción.....	1
1.2	Antecedentes	2
1.2.1	Antecedentes institucionales	2
1.2.1.1	Misión.....	3
1.2.1.2	Visión	3
1.2.2	Antecedentes afines del proyecto	3
1.2.2.1	Internacional	3
1.2.2.2	Nacional.....	4
1.2.2.3	Local	4
1.3	Planteamiento del problema	5
1.3.1	Problema Principal.....	5
1.3.2	Formulación del problema	5
1.3.3	Problemas Secundarios	5
1.4	Objetivos	6
1.4.1	Objetivo General	6
1.4.2	Objetivos Específicos.....	6
1.5	Justificación.....	6
1.5.1	Justificación Técnica.....	6
1.5.2	Justificación Económica.....	7
1.5.3	Justificación Social.....	7
1.6	Metodología.....	7
1.6.1	Metodología de Desarrollo	7
1.6.2	Métricas de Calidad	8

1.6.3	Estimación de costos	8
1.6.4	Seguridad ISO/IEC 27000	9
1.6.5	Pruebas de software	9
1.7	Herramientas	10
1.8	Limites Y Alcances	11
1.8.1	Límites	11
1.8.2	Alcances	11
1.9	Aportes	12
2	MARCO TEORICO	13
2.1	Sistema	13
2.2	Web	13
2.2.1	La Web 2.0	14
2.3	Sistema Web	15
2.3.1	Aplicación Web	15
2.3.2	Single-Page Aplicación (SPA).....	16
2.3.3	API REST	17
2.4	Sistemas De Control De Acceso	19
2.4.1	Control de Acceso.....	19
2.4.2	Tecnología de Identificación por Radio Frecuencia (RFID)	20
2.4.2.1	Componentes del Sistema RFID	20
2.5	Administración De Recursos Humanos	22
2.5.1	Planilla	22
2.5.1.1	Cálculo de Salarios.....	23
2.5.2	Remuneración.....	26
2.5.3	Salario.....	27

2.6	Ingeniería De Software.....	27
2.7	Metodología De Desarrollo.....	29
2.7.1	Metodología Tradicional.....	29
2.7.2	Metodología Ágil	29
2.7.3	Metodología Kanban.....	29
2.7.3.1	Características.....	30
2.7.3.2	Roles Kanban	31
2.7.3.3	Tablero Kanban	32
2.7.3.4	Fases Kanban.....	33
2.8	UWE – Ingeniería Web Basada en UML	34
2.8.1	Modelo de Requerimientos	35
2.8.2	Modelo de contenido.....	36
2.8.3	Modelo de Navegación	37
2.8.4	Modelo de Proceso	38
2.8.5	Modelo de Presentación	39
2.9	Métricas De Calidad	40
2.9.1	Métricas de calidad establecida estándar ISO/IEC-25000.....	40
2.9.2	Estructura de la familia ISO/IEC 25000	41
2.9.3	Medición de Calidad ISO/IEC 25010	41
2.9.3.1	Adecuación funcional.....	42
2.9.3.2	Fiabilidad	42
2.9.3.3	Eficiencia en el desempeño	43
2.9.3.4	Facilidad de uso.....	43
2.9.3.5	Seguridad	44
2.9.3.6	Compatibilidad.....	44

2.9.3.7	Mantenibilidad.....	44
2.9.3.8	Portabilidad.....	45
2.10	Método De Estimación De Costo De Software.....	45
2.10.1	Modelo COCOMO.....	46
2.10.1.1	Modelos de Estimación de costo con COCOMO	47
2.11	Seguridad.....	51
2.11.1	Estándar ISO/IEC 27000.....	51
2.11.1.1	ISO 27001.....	52
2.11.1.2	ISO 27002.....	53
2.12	Pruebas De Software	53
2.12.1	Pruebas de Caja Blanca	53
2.12.2	Pruebas de Caja Negra.....	54
2.12.3	Pruebas de Estrés	55
2.13	Herramientas De Desarrollo.....	55
2.13.1	Php	55
2.13.2	JavaScript	56
2.13.3	Framework Laravel	56
2.13.4	Framework Vue.js	57
2.13.5	Framework Quasar	57
2.13.6	MariaDB.....	58
2.13.7	Módulo RFID-RC522	58
2.13.8	Módulo ESP8266MOD.....	59
3	MARCO APLICATIVO.....	61
3.1	Esquema Del Sistema	61
3.2	Aplicación De La Metodología De Desarrollo	61

3.2.1	Combinación de metodologías Kanban – UWE	61
3.2.2	Roles de Kanban	63
3.2.3	Fase de Análisis.....	63
3.2.3.1	Análisis de requerimientos.....	63
3.2.3.2	Caso de uso.....	64
3.2.3.3	Diagrama de contenido.....	74
3.2.3.4	Diseño de la base de datos	75
3.2.4	Fase de diseño	76
3.2.4.1	Modelo de Navegación	76
3.2.4.2	Modelo de Presentación	80
3.2.5	Fase de desarrollo	87
3.2.5.1	Modelo de Flujo de Proceso	87
3.2.6	Fase de implementación.....	90
3.2.7	Fase de producción	107
4	METRICAS DE CALIDAD, SEGURIDAD Y ESTIMACION DE COSTOS	109
4.1	Norma ISO/IEC 25010.....	109
4.1.1	Funcionalidad.....	109
4.1.2	Usabilidad	112
4.1.3	Fiabilidad	113
4.1.4	Mantenibilidad.....	115
4.1.5	Portabilidad	115
4.1.6	Calidad total.....	116
4.2	Estimación de costos.....	117
4.2.1	Análisis de costo	117
4.2.2	Costo del desarrollo del software del sistema.....	118

4.3	Seguridad	120
4.4	Pruebas	121
4.4.1	Prueba de Caja Blanca	122
4.4.2	Prueba de Caja Negra	123
4.4.3	Pruebas de estrés.....	125
5	CONCLUSIONES Y RECOMENDACIONES	126
5.1	Conclusiones.....	126
5.2	Recomendaciones.....	127

ÍNDICE DE FIGURAS

Figura 1.1 Organigrama de la empresa	2
Figura 2.1 Sistema de Gestión Api	17
Figura 2.2 Arquitectura REST	18
Figura 2.3 Arquitectura cliente-servidor	18
Figura 2.4 Formato JSON	19
Figura 2.5 Diagrama Sistema RFID	21
Figura 2.6 Ejemplo de formato de planilla se sueldos	26
Figura 2.7 Tablero Kanban	32
Figura 2.8 Panel Kanban	33
Figura 2.9 Modelo de Caso de Uso UWE	36
Figura 2.10 Modelo de contenido UWE	36
Figura 2.11 Modelo de navegación UWE	37
Figura 2.12 Nombres de estereotipos y sus iconos	37
Figura 2.13 Modelo de estructura del Proceso	38
Figura 2.14 Modelo de estructura del Flujo del Proceso	39
Figura 2.15 Nombre de estereotipos y sus iconos	40
Figura 2.16 Estructura de la familia ISO/IEC 25000	41
Figura 2.17 Características de la ISO 25010	42
Figura 2.19 Detalle de coeficientes de COCOMO II	50
Figura 2.20 Familia norma ISO/IED 27000	52
Figura 2.21 Caja Blanca	54
Figura 2.22 Caja Negra	54
Figura 3.1 Esquema del sistema	61

Figura 3.2 Tablero Kanban – UWE	62
Figura 3.3 Diagrama de caso de uso general	65
Figura 3.4 Diagrama de caso de uso Gestión Usuarios.....	67
Figura 3.5 Diagrama de caso de uso Gestión Empleados	68
Figura 3.6 Diagrama de caso de uso Contrato	69
Tabla 3.6 Especificación del caso de uso Contrato	70
Figura 3.7 Diagrama de caso de uso Gestión Planillas	70
Figura 3.8 Diagrama de caso de uso Control de Asistencias y Permisos	72
Figura 3.9 Diagrama de caso de uso Control de Asistencias	73
Figura 3.10 Diagrama de Contenido del sistema	75
Figura 3.11 Diseño de la base de datos	76
Figura 3.12 Diagrama de Navegación Gestión de Usuarios	77
Figura 3.13 Diagrama de Navegación Gestión de Empleados	78
Figura 3.14 Diagrama de Navegación Contrato.....	78
Figura 3.15 Diagrama de Navegación Planillas	79
Figura 3.16 Diagrama de Navegación Control de Asistencia.....	79
Figura 3.17 Diagrama de Presentación Usuario Admin	81
Figura 3.18 Diagrama de Presentación jefe RR.HH.	82
Figura 3.19 Diagrama de Presentación Usuario Técnico RR.HH.	83
Figura 3.20 Diagrama de Presentación Usuario Empleado	83
Figura 3.21 Diagrama de Presentación Gestión Usuarios	84
Figura 3.22 Diagrama de Presentación Gestión Empleados	85
Figura 3.23 Diagrama de Presentación Contrato.....	85
Figura 3.24 Diagrama de Presentación Planilla	86
Figura 3.25 Diagrama de Presentación Control de Asistencias	86

Figura 3.26 Diagrama de Flujo de Procesos Gestión de Usuarios	87
Figura 3.27 Diagrama de Flujo de Procesos Gestión de Empleados.....	88
Figura 3.28 Diagrama de Flujo de Procesos Contrato	88
Figura 3.29 Diagrama de Flujo de Proceso de Planilla	89
Figura 3.30 Diagrama de Flujo de Proceso de Control de Asistencias	90
Figura 3.31 Pantalla Login	91
Figura 3.32 Código fuente autenticación.....	91
Figura 3.33 Pantalla de bienvenida al Sistema.....	92
Figura 3.34 Pantalla de Gestión de Usuarios.....	92
Figura 3.35 Código fuente listado de usuarios según su rol	93
Figura 3.36 Pantalla de Creación de Usuario	93
Figura 3.37 Código fuente creación usuario	94
Figura 3.38 Pantalla de la lista de Roles.....	94
Figura 3.40 Pantalla de Creación de un Rol	95
Figura 3.41 Código fuente listado de roles	96
Figura 3.42 Pantalla de listado de permisos	96
Figura 3.43 Código fuente listado de permisos.....	97
Figura 3.44 Pantalla de Creación de permisos	97
Figura 3.45 Código fuente creación permisos.....	98
Figura 3.46 Pantalla de Gestión de Empleados.....	98
Figura 3.47 Código fuente métodos para la gestión de empleados	99
Figura 3.48 Pantalla de Creación de un Empleado.....	99
Figura 3.49 Código fuente creación empleados.....	100
Figura 3.50 Pantalla de listado de los contratos	100
Figura 3.51 Código fuente listado de contratos.....	101

Figura 3.52 Pantalla de la Creación de un Contrato	101
Figura 3.53 Código fuente creación contrato	102
Figura 3.54 Pantalla de Control (Regularización) de Asistencias	102
Figura 3.55 Código fuente listado asistencias.....	103
Figura 3.56 Pantalla de actualización de horas trabajadas.....	103
Figura 3.57 Código fuente actualización de horas trabadas	104
Figura 3.58 Captura del lector RFID	104
Figura 3.59 Captura de reporte de asistencia.....	105
Figura 3.60 Captura de reporte de Planillas.....	105
Figura 3.61 Captura de reporte de boleta de pago	106
Figura 3.62 Subida de la aplicación al server	107
Figura 3.63 Asignación de permisos.....	107
Figura 3.64 Creación de la base de datos	107
Figura 3.65 Asignación de permisos.....	108
Figura 3.66 Creación del subdominio	108
Figura 3.67 Prueba de verificación de la aplicación.....	108

ÍNDICE DE TABLAS

Tabla 2.1 Características Kanban.....	31
Tabla 2.2 Elementos tablero Kanban.....	32
Tabla 2.18 Cálculo de Punto función no ajustado	48
Tabla 3.1 Fases de trabajo, limite WIP y responsables	63
Tabla 3.2 Lista de requerimientos del sistema.....	64
Tabla 3.3 Descripción de caso de Uso General.....	65
Tabla 3.4 Especificación del caso de uso Gestión de Usuarios.....	67
Tabla 3.5 Especificación del caso de uso Gestión Empleados	69
Tabla 3.7 Especificación del caso de uso Gestión Planillas	71
Tabla 3.8 Especificación del caso de uso Control de Asistencias.....	72
Tabla 3.9 Especificación del caso de uso Asistencia.....	74
Tabla 4.1 Factor de ponderación	110
Tabla 4.2 Valores de ajuste de complejidad	111
Tabla 4.3 Encuesta sobre la usabilidad del sistema	112
Tabla 4.4 Valores de fiabilidad de cada módulo	114
Tabla 4.5 Cálculo de calidad global	117
Tabla 4.6 Conversión de puntos de función a KLDC	117
Tabla 4.7 Tabla de Constantes COCOMO	119
Tabla 4.8 Resumen COCOMO	120
Tabla 4.9 Medidas de seguridad.....	121
Tabla 4.10 Valores de entrada-proceso de inicio de sesión del usuario	124
Tabla 4.11 Pruebas de caja negra ingreso usuario.....	124

CAPÍTULO I

MARCO PRELIMINAR

1 MARCO PRELIMINAR

1.1 Introducción

En la actualidad las tecnologías de información han evolucionado de manera acelerada hecho que ha sido aprovechado por las empresas para mejorar el control de una gran cantidad de información, por lo cual cada vez son más las empresas que dedican esfuerzos a conseguir un buen sistema acorde a sus necesidades.

El grupo Anti Money Laundering Consultants (AMLC) es una empresa privada que tiene presencia en varios países de Latinoamérica que ofrece servicios de consultorías de gestión y administración de riesgos lavado de activos y financiamiento al terrorismo (LA/FT) acorde a las normativas citas por la ASFI y UIF a través de sus herramientas informática: AMLC COMPLIANCE “sistemas de prevención de lavado de activo y Financiamiento al Terrorismo”, para la detección y corrección de riesgos ocasionados por organizaciones criminales en los Sistemas Financieros y estatales, lo cual AMLC Bolivia requiere implementar un sistema acorde a las necesidades actuales.

En la actualidad, la empresa cuenta con un sistema de registro asistencia básico en Excel desarrollada con VBA siguiendo un procedimiento semi manual a la generación de reportes, planillas lo cual genera pérdida de tiempo y aumento de costos de operación.

El presente proyecto de grado, propone afrontar los inconvenientes que se da en el área de recursos humanos, implementando un Sistema Web para el control de asistencias para la mejora del proceso de planillas. Para el desarrollo del sistema se hará uso de la metodología UWE, que permite modelar de mejor manera una aplicación Web, también se utilizará la tecnología Radio-frequency Identification (RFID) que ha tenido mucho auge en los últimos años debido a la relativa reducción de precios en el mercado, al incremento en sus capacidades y a las ventajas que presenta frente a otras tecnologías de autoidentificación. El sistema web tendrá como arquitectura de desarrollo web API REST lo cual permitirá tener separado el Front-End y Back-End,

esto permitirá la evolución o refactorizar cualquiera de los dos.

1.2 Antecedentes

1.2.1 Antecedentes institucionales

La empresa Anti Money Laundering Consultants A.M.L.C. ubicada en la Av. Hernando Siles Nro. 420 zona de obrajes La Paz – Bolivia es una empresa orientada a brindar servicios de excelencia y a medida de las necesidades del cliente. La experiencia de sus miembros ha permitido focalizar sus esfuerzos en el desarrollo de Sistemas de prevención y metodologías de gestión de riesgos para la prevención de lavado de activos y financiamiento al terrorismo (LA/FT) ocasionados por organizaciones criminales en los Sistemas Financieros y estatales, así como implementar mecanismos para comprender la Gestión eficiente del riesgo que afecta a las organizaciones. Cuenta con presencia en Bolivia, Paraguay, Ecuador, Nicaragua, Honduras, Costa Rica, Panamá, El Salvador, México y apoyamos la relación institucional con países como Colombia, República Dominicana, Perú y Uruguay. Los trabajos de AMLC en Gestión de Riesgos y softwares de prevención están en más de 120 instituciones financieras en Latinoamérica.

Figura 1.1
Organigrama de la empresa



Nota: Organigrama de la empresa; (Amlc, 2020)

1.2.1.1 Misión

Poner a disposición a las instituciones financiera nacionales y extranjeras, nuestros recursos humanos y servicios para la administración eficiente de los riesgos a que están expuestas en sus diversas actividades, especialmente de Lavado de Activos, Financiera y de tecnología, ofreciendo un servicio personalizado.

Poner a disposición a las entidades financieras metodologías de riesgo acorde a las normativas y exigencias por parte de regulación de la Asfi y Uif, a través de una herramienta informática “AMLC COMPLIANCE”.

1.2.1.2 Visión

Convertirnos en el socio estratégico de nuestros clientes en el crecimiento de sus instituciones, productos o servicios, inculcando en nuestros socios, funcionarios y empleados el significado de la palabra SERVICIO para que resuelvan en forma rápida, oportuna y efectiva las necesidades de cada cliente.

1.2.2 Antecedentes afines del proyecto

1.2.2.1 Internacional

- Se cita el trabajo realizado por (Alvarado, 2008) que titula: “Sistema de Control de Acceso con RFID” Tiene como objetivos principales involucrar en el sistema tarjetas de radiofrecuencia para identificar al personal e implementar una serie de aplicaciones cliente-servidor que permita controlar y administrar la operación del sistema de accesos. El microcontrolador que se utilizó fue el MSP430F149 y como lenguaje de programación Java y gestor de base de datos MySQL; Instituto Politécnico Nacional, Departamento de Ingeniería Eléctrica; México.
- Se cita el trabajo realizado por (Cali, 2020) que titula: “Control de asistencia de estudiantes mediante dispositivos biométricos RFID en la universidad estatal del sur de Manabi”; Tiene como objetivos Implementar el sistema de control de asistencia de estudiantes mediante dispositivos biométricos RFID. Tesis elaborada mediante análisis de información aplicando técnicas como encuestas; Universidad Estatal del sur de Manabi; Ecuador.
- Se cita el trabajo realizado por (Vargas, 2013) que titula: “Sistema de Control de Acceso y Monitoreo con la Tecnología RFID para el Departamento de

Sistemas de la Universidad Politécnica Salesiana Sede Guayaquil” Proyecto de Grado que tiene como objetivo principal implementar dispositivos que permitan el uso seguro y adecuado en los laboratorios para así facilitar el control de ingreso de acuerdo a los horarios establecidos. Para lograr estos objetivos utiliza el PIC 18F44550 como cerebro principal de hardware., Guayaquil; Ecuador

1.2.2.2 Nacional

- Se cita el trabajo realizado por (Alcón, 2016) que titula: “Sistema de control y monitoreo de asistencia de personal mediante radiofrecuencia y NFC” Diseñó un software conjuntamente con Arduino lo cual permite realizar el control de asistencia de manera automática. Para el lograr el objetivo utiliza el Módulo RFID-RC522 como lector de los Tarjetas de identificación., UMSA; La Paz.
- Se cita el trabajo realizado por (Silva, 2016) que titula: “Sistema Biométrico para el control de personal y generación de planillas de sueldos, impositiva y patronal” Proyecto de Grado tiene como uno de su objetivo paritario diseñar una herramienta automatizada que controle las fechas de ingreso del personal a la institución. Para lograr estos objetivos se ha propuesto el uso de la tecnología de Arduino como plataforma principal, así mismo aplicando la metodología de desarrollo R.U.P., UMSA; La Paz.

1.2.2.3 Local

- Se cita el trabajo realizado por (Sánchez, 2020) que titula: “Sistema de control y Administración de Recursos Humanos para la incubadora de microempresas productivas” Se trata de un proyecto de grado basado en una metodología del modelo de prototipo evolutivo con la Programación Extrema (Xtreme Programming), desarrolló un software que le permite controlar y administrar todo referente a la administración de Recursos Humanos., UPEA; La Paz.
- Se cita el trabajo realizado por (Forra, 2020) que titula: “Sistema web para la venta de ladrillos y control personal” Proyecto donde uno de sus objetivos es la administración sobre las asistencias y salarios de todo el personal, a su vez aplica la metodología UWE., UPEA: La Paz.

1.3 Planteamiento del problema

La empresa Anti Money Laundering Consultants no cuenta con algún sistema de control de personal en cuanto a tiempos de asistencia y control de accesos de áreas.

Actualmente los responsables del área de Recursos Humanos conocen la importancia de disponer con un sistema que genere reportes de manera eficiente referentes a los tiempos tales como son: puntualidad y asistencia, permisos, personas que entran y salen de la empresa. Es por ello que el problema a plantear está enfocado al control de acceso y administración de recursos humanos concerniente a las asistencias de la empresa Anti Money Laundering Consultants AMLC.

1.3.1 Problema Principal

Realizado un análisis y posterior elaboración de un árbol de problemas (ver anexo), que ha permitido identificar la siguiente problemática:

La empresa Anti Money Laundering Consultants tiene dos problemas esenciales control de asistencia para la generación de planillas de manera eficiente y control de acceso en sus diferentes áreas de la empresa Anti Money Laundering Consultants, lo que provoca retrasos en los pagos de haberes y pérdida de activos.

1.3.2 Formulación del problema

Considerando lo anteriormente mencionado se formula el problema:

¿Cómo optimizar el control de asistencias para la generación de planillas además del control de accesos en la empresa Anti Money Laundering Consultants?

1.3.3 Problemas Secundarios

- El cálculo de horas trabajadas se realiza de forma semiautomática lo que ocasiona demora en la generación de planillas de sueldos del personal.
- La generación de boletas de pago semiautomática ocasiona demora en el pago al personal.
- El personal no cuenta con un control propio de sus retrasos y faltas, generando disconformidad en sus registros de asistencias.

- Seguridad de la información no es garantizada por los datos almacenados en documentos Excel, esto puede provocar pérdida de la data por motivos de corrupción del archivo o virus.
- No se cuenta con un control de acceso hacia las áreas, lo que puede ocasionar pérdida de activos.

Considerando lo anteriormente mencionado se formula el problema:

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar un Sistema Web de control de asistencias y control de acceso a las áreas de la empresa para el personal de AMLC, utilizando la tecnología RFID que permita generar la planilla de sueldos de manera eficiente y tener un control adecuado de accesos.

1.4.2 Objetivos Específicos

- Automatizar la generación de las planillas de sueldos con el cálculo de todos sus beneficios y descuentos de cada empleado.
- Automatizar la generación de boletas de pago de los empleados para el pago de los mismos.
- Realizar un módulo de reportes de las asistencias de los empleados para que tengan un seguimiento de los mismos.
- Diseñar una base de datos donde se administre la información, permitiendo la facilidad de resguardo de toda la data.
- Armar el hardware de comunicación de tarjetas(tag) RFID con el lector RFID-RC522, para el marcado de asistencias y control de acceso para los empleados. Tarjetas que serán asignadas al personal.

1.5 Justificación

1.5.1 Justificación Técnica

El proyecto se justifica técnicamente con la finalidad de automatizar los procesos de administración de asistencias, planillas, control de accesos entre otros, estará disponible desde cualquier ordenador dentro de la empresa.

La empresa Anti Money Laundering Consultants cuenta con una infraestructura

de equipamiento óptimo, además de contar con un servidor VPS y otro dedicado esto viabilizará la realización del presente proyecto.

1.5.2 Justificación Económica

El presente proyecto será desarrollado con software libre esto reducirá los costos por lo que no se tendrá que invertir en licencias de software en la empresa.

También el sistema minimizará el costo asociado al tiempo de obtención de información obteniendo así resultados eficientes con el menor esfuerzo del personal administrativo.

1.5.3 Justificación Social

El presente proyecto proporcionará a la empresa una herramienta que agilizará los procesos diarios, incrementando así la productividad lo cual beneficiará directamente el área de administración de recursos humanos. También beneficiará de gran manera al personal de la empresa, ya que podrán tener un historial de asistencias de manera automatizada.

1.6 Metodología

1.6.1 Metodología de Desarrollo

Para el desarrollo del software se hará uso de la metodología UWE, que permite modelar de mejor manera una aplicación Web, para el proceso de creación de aplicaciones detalla ésta, con una gran cantidad de definiciones, en el proceso de diseño lista que debe utilizarse. Procede de manera iterativa e incremental, coincidiendo con UML, incluyendo flujos de trabajo y puntos de control.

UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web, fue presentada por Nora Koch en el 2000.

Las fases o etapas a utilizar son:

- Captura, análisis y especificación de requisitos,
- Diseño del sistema.
- Codificación del software.
- Pruebas.
- La Instalación o Fase de Implementación.
- El Mantenimiento.

El modelo que propone UWE está compuesto por etapas/modelos o Submodelos: Modelo Lógico-Conceptual, Modelo de Navegación, Modelo de presentación, Interacción Temporal, Escenarios Web y Diagramas. (Cordova JR, 2013)

1.6.2 Métricas de Calidad

Serie ISO/IEC 25000 Es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La serie ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de la serie de normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, y la serie ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

- ISO/IEC 2500n - División de Gestión de Calidad
- ISO/IEC 2501n - División de Modelo de Calidad
- ISO/IEC 2502n - División de Medición de Calidad
- ISO/IEC 2503n - División de Requisitos de Calidad
- ISO/IEC 2504n - División de Evaluación de Calidad

1.6.3 Estimación de costos

Para la estimación de costos se aplicarán métricas para calcular el esfuerzo invertido, con el modelo COCOMO II (COConstructive COSt Model (MOdelo Constructivo de COstó) creado por Barry W. Boehm. En el que se define en tres modelos los cuales son:

- El modelo Composición de Aplicación
- El modelo Diseño Temprano
- El modelo Post-Arquitectura

Para el desarrollo del presente proyecto se utilizó el Modelo Post-Arquitectura de COCOMO II, dado que realiza las estimaciones con bastante precisión.

1.6.4 Seguridad ISO/IEC 27000

Un Sistema de Gestión de Seguridad de Información (SGSI) debe descansar en tres objetivos fundamentales que garanticen los datos que maneja y estos son:

- La Confidencialidad
- La Integridad
- La Disponibilidad

En lo que se conoce como el triángulo de la seguridad de los datos con el acrónimo de CID, Confidencialidad, Integridad y Disponibilidad.

La serie de normas internacionales ISO/IEC 27000 ofrecen una serie de recomendaciones de mejores prácticas, para la gestión de la seguridad de la información, y esta puede ser aplicada en cualquier organización sin importar el tamaño que tenga, está orientada a que estas puedan mantener un Sistema de Gestión de la Seguridad, la cual está compuesta de las siguientes normas 27001,27002, 27003, 27004 y 27005 al ser las que se encuentran relacionadas más íntimamente con la puesta en marcha de un SGSI.

1.6.5 Pruebas de software

Prueba de tipo de caja blanca: “Se refiere a las pruebas que se llevan a cabo en la interfaz del software. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software” (Pressman R., 2010).

Prueba de tipo de caja negra: “Se basa en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles” (Pressman R., 2010).

Prueba de tipo de estrés: "se enfrenta el sistema a situaciones anormales, es decir, enfrentarlo a peticiones de usuario con una frecuencia definida, para determinar su uso de recursos en dichas circunstancias y su reacción ante la misma. Busca sobrecargar el sistema, para hallar estados de inestabilidad o procesamiento inapropiado" (Ocampo A. y Correa T., 2011, p. 47).

1.7 Herramientas

Para el desarrollo del sistema web se tiene varias herramientas entre los más importantes se menciona a continuación:

Software:

Backend

- **PHP** (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.
- **Laravel** es un framework de código abierto para desarrollar aplicación y servicios web con PHP. Laravel es un framework de aplicación web con una sintaxis expresiva y elegante. Un framework web proporciona una estructura y un punto de partida para crear aplicaciones, lo que le permite concentrarse en crear algo increíble mientras nos preocupamos por los detalles.
- **MariaDB** es uno de los gestores de bases de datos más populares del mundo. Está hecho por los desarrolladores originales de MySQL y se garantiza que permanecerá como código abierto. Los usuarios notables incluyen Wikipedia, WordPress.com y Google.

Frontend

- **JavaScript** es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.
- **Vue.js** es un framework de JavaScript para construir interfaces de usuario. Se

basa en HTML, CSS y JavaScript estándar, y proporciona un modelo de programación declarativo y basado en componentes que lo ayuda a desarrollar interfaces de usuario de manera eficiente, ya sea simple o compleja.

- **Quasar** es un framework basado en Vue.js de código abierto con licencia del MIT, que le permite, como desarrollador web, crear rápidamente sitios web aplicaciones responsivos entre muchos: SPA (aplicación de una sola página), SSR (aplicación renderizada del lado del servidor), PWA (aplicación web progresiva).

Hardware:

- **RFID-RC522** (Identificador por radiofrecuencia) es un conjunto de tecnologías diseñadas para leer etiquetas (tags) a distancia de forma inalámbrica. Los lectores RFID pueden ser conectados a un autómata o procesador como Arduino.
- **ESP8266MOD** es una plataforma de desarrollo similar a Arduino especialmente orientada al Internet de las cosas (IoT). La placa NodeMcu v2 ESP8266 tiene como núcleo al SoM ESP-12E que a su vez está basado en el SoC Wi-Fi ESP8266, integra además el conversor USB-Serial TTL CP2102 y conector micro-USB necesario para la programación y comunicación a PC.

1.8 Límites Y Alcances

1.8.1 Límites

- El Sistema Web solo se aplicará en la empresa Anti Money Laundering Consultants
- El Sistema Web no estará relacionado con el área contable

1.8.2 Alcances

El presente proyecto de grado presenta una solución para la empresa Anti Money Laundering Consultants, por lo que se pretende desarrollar un Sistema web de control de asistencia y control de acceso a las áreas

El sistema web desarrollado tendrá los siguientes alcances:

- Módulo de Administración de Usuarios, se otorgará roles de acceso al sistema

y se aplicaran técnicas de seguridad en cuanto al ingreso al sistema

- Módulo Asignación de RFID al personal, se vinculará la tarjeta RFID al personal al crear al personal.
- Módulo de Asistencia, permitirá el marcado de asistencia con la tarjeta RFID.
- Módulo de reportes, en el cual se podrá ver los reportes de asistencias entre otros.

1.9 Aportes

El presente proyecto de grado es para solucionar los problemas con los que atraviesa el área de Recursos Humanos por la falta de control al personal disminuyendo así los pagos retrasados, exagerado uso de papel y dando solución a los procesos iterativos de las hojas de cálculo de Excel, por otro lado, la aplicación de la arquitectura de desarrollo web Api Rest permitirá obtener beneficios a niveles técnicos y de reutilización o cambios de lenguajes de programación debido a que su estructura descompuesta en Back-end y Front-end permiten fácilmente escalar.

2 MARCO TEORICO

En el siguiente capítulo se expone los fundamentos teóricos con relación a la metodología y herramientas a utilizar durante el desarrollo del sistema y la elaboración de la documentación. Esto permitirá tener una mejor referencia de lo que va a utilizar más adelante para la documentación del marco aplicativo.

2.1 Sistema

Se entiende por un sistema a un conjunto ordenado de elementos relacionados entre sí, de forma tal que un cambio en un elemento afecta al conjunto de todos ellos. Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación.

Según Seen (1999), es “Un conjunto de componentes que interactúan entre sí para lograr un objetivo común” (p. 363). También Montevilla (1999), “Un conjunto de dos o más elementos interrelacionados que conforman un todo” (p. 241). Por otra parte, Johansen (1993), lo define como “Un conjunto de partes coordinadas y en interacción para alcanzar un conjunto de objetivos” (p. 54).

2.2 Web

(World Wide Web, o www), es un conjunto de documentos (webs) interconectados por enlaces de hipertexto, disponibles en Internet que se pueden comunicar a través de la tecnología digital. Se entiende por “hipertexto” la mezcla de textos, gráficos y archivos de todo tipo, en un mismo documento.

La web fue creada en 1989 por Tim Berners Lee. Consistía en una forma de organizar la información usando como medio físico de comunicación la red internet y el protocolo HTTP. HTTP (Hypertext Transference Protocol) es el protocolo de transferencia de hipertexto que los navegadores utilizan para realizar peticiones a los servidores web y para recibir las respuestas de ellos. Es el protocolo que se utiliza para ver páginas web. (Ramos & Ramos, 2014, pág. 3)

El formato de esta información puede ser HTML y desde una web se puede entrar a otra con enlaces de hipertexto. Los recursos para la presentación de la información de una web pueden ser scripts, imágenes digitales, hojas en forma de cascada, entre Otros. Para que una web pueda estar en línea y ser vista por cualquier persona, requiere de un hospedaje, cosa que puede hacerse en un servidor remoto o local. La composición de una página web consiste básicamente en información que puede presentarse con módulos multimedia y texto, lo cual se complementa con hiperenlaces para conducir a otras secciones de dicha web o a otras webs. Según (Merino, 2013)

La Web es un sistema de servidores de internet que utilizan HTTP (Protocolo de transferencia de Hipertexto). Para transferir documentos formateados en HTML (Lenguaje de marcación de Hipertexto). Estos son visualizados al utilizar el lenguaje para navegadores Web. El hipertexto permite que un documento esté vinculado a otros documentos sobre la Web, a través de hiperenlaces, es posible moverse de un documento a otro utilizando los hipervínculos encontrados dentro de las páginas Web. (Martin & Martin, 2011)

2.2.1 La Web 2.0

Los sitios web en la actualidad no sólo tienen contenido estático; también permiten a las personas a colaborar y compartir información, además de crear nuevos servicios y contenido en línea aprovechando el potencial de los usuarios en la generación de contenidos. Estos servicios interactivos basados en internet de segunda generación se conocen como web 2.0.

“La Web 2.0 es un conjunto de tecnologías, estrategias de negocio y tendencias sociales” (Rossi, Pastor, Schwabe, & Olsina, 2008).

2.3 Sistema Web

Los sistemas web y aplicaciones basados en web (WebApps), ofrecen un complejo arreglo de contenido y funcionalidad a una amplia población de usuarios finales.

Un sistema web es una aplicación que no depende de una plataforma, Sino que está desarrollada para ser implementada en un servidor web con características y capacidades mayores a una página web.

Las características más importantes de un sistema web detallados en R. Pressman son las siguientes.

Disponibilidad, aunque la expectativa de disponibilidad total es poco razonable, los usuarios e WEDApps populares con frecuencia demandan acceso sobre una base de "24/7/365".

Inmediatez, tienen una inmediatez que no se puede encontrar en otro tipo de software, los sistemas web con frecuencia muestran un tiempo para comercializar puede ser cuestión de unos días o semanas.

Seguridad, los Sistemas Web, están disponibles mediante el acceso a la red, es difícil, si no imposible, limitar la población de usuarios finales que pueden tener acceso al sistema. Con la finalidad de proteger el contenido confidencial y ofrecer modos seguros de transición de datos, Se deben implementar fuertes medidas de seguridad a lo largo de la infraestructura que sustenta. un Sistema Web y dentro de la aplicación misma.

Estética, la presentación de un Sistema Web y la disposición de sus elementos es muy importante. "Cuando una aplicación se diseña para comercializar o vender productos o ideas, la estética puede tener tanto en el éxito como el diseño técnico" (Cotana, 2020).

2.3.1 Aplicación Web

Las aplicaciones web es un tipo especial de aplicación cliente/servidor, donde el cliente (el navegador o explorador) como el servido (servidor web) y el protocolo de comunicación (HTTP) están estandarizados y o han de ser creados por el programador de la aplicación. Las aplicaciones web son populares debido a lo práctico del

navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Las principales ventajas son:

- El problema de gestionar el código en el cliente se reduce drásticamente; todos los cambios tanto de funcionalidad e interfaz se realizarán cambiando el código que resida en el servidor.
- Ahorro de tiempo de ejecución, se desarrollan las tareas sin la necesidad de instalar algún programa.
- Se evita la gestión de versiones, actualizaciones inmediatas.
- Consumo de recursos bajos porque muchas de las tareas la realizan desde el servidor.
- Independencia de plataforma, se puede ejecutar en distintas plataformas (sistema operativo y hardware), solo se necesita disponer de un navegador web.
- Los virus no dañan los datos porque son almacenados en el servidor de base de datos.

(Subramanian)

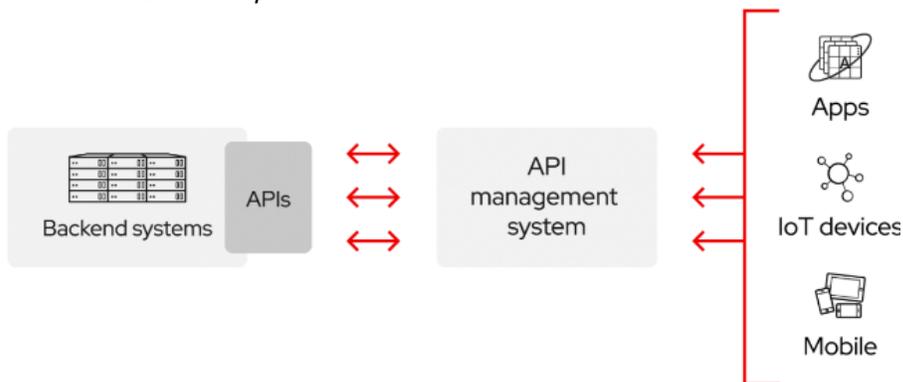
2.3.2 Single-Page Aplicación (SPA)

Un single-page application (SPA), o aplicación de página única es una aplicación web o es un sitio web que cabe en una sola página, y cuyo propósito es dar una experiencia más fluida a los usuarios similar a una aplicación de escritorio. En un SPA todos los códigos de HTML, JavaScript, y CSS se cargan de una vez o los recursos necesarios se cargan dinámicamente como lo requiera la página y se van agregando, normalmente como respuesta de las acciones del usuario. La página no tiene que cargar otra vez en ningún punto del proceso tampoco se transfiere a otra página, aunque las tecnologías modernas pueden permitir la navegabilidad en páginas lógicas dentro de la aplicación. La interacción con las aplicaciones de página única puede involucrar comunicaciones dinámicas con el servidor web que está detrás. (Red Hat)

2.3.3 API REST

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas. API significa interfaz de programación de aplicaciones.

Figura 2.1
Sistema de Gestión Api



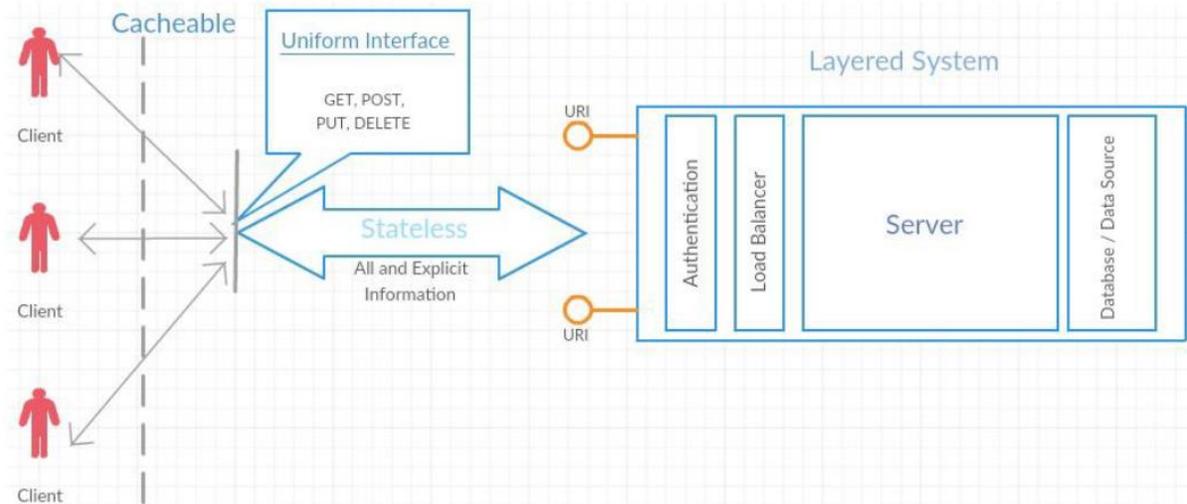
Nota: Comunicación del sistema de gestión Api; (Red Hat)

El propósito de una API es intercambiar datos entre diferentes sistemas, la mayoría de las veces estos intercambios de datos tienen como objetivo automatizar procesos y creación de nuevas funcionalidades. En el ámbito web, podríamos decir que una API es un servicio backend que se utiliza para conectar dos aplicaciones.

Las API brindan acceso a los recursos mientras mantiene la seguridad y el control. Su seguridad depende de una gestión adecuada incluido el uso de puertas de enlace API. Para conectarse a las API y crear aplicaciones que usen los datos o la funcionalidad que brindan, puede usar una plataforma de integración distribuida que conecte cosas, como sistemas heredados e Internet de las cosas (IoT).

Las APIs REST se distinguen por que se basan en el protocolo de aplicación HTTP. Es decir, utilizan códigos y métodos de respuesta HTTP para una función específica y ampliamente reconocida. Y nos permite a través de la URI, la estructuración de los recursos disponibles. (Red Hat)

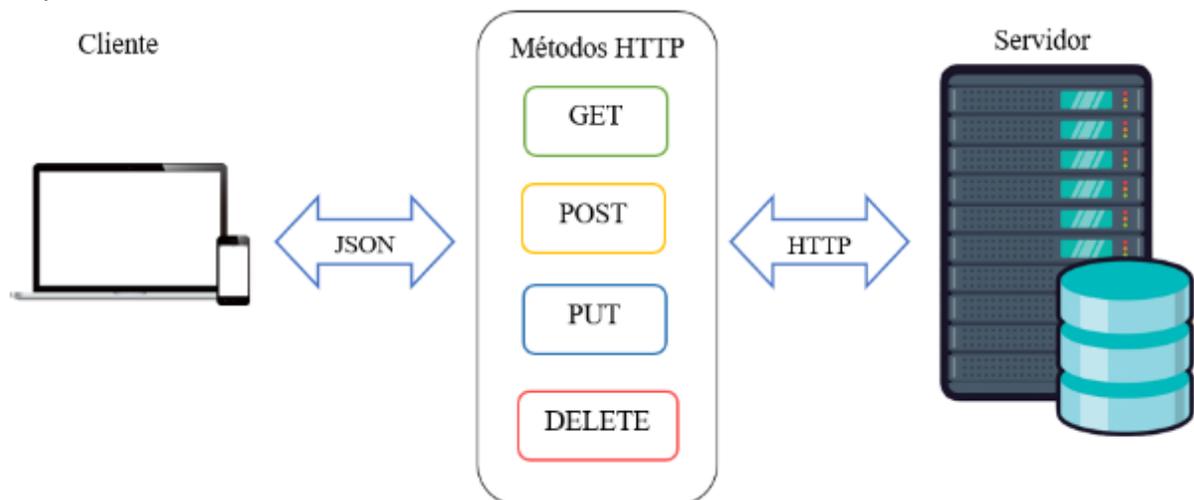
Figura 2.2
Arquitectura REST



Nota: comportamiento en la arquitectura REST; (H. Subramanian y P. Raj, 2019)

La petición se envía desde el cliente al servidor vía HTTP en forma de URL web con sus respectivas cabeceras. Usando un método HTTP, después, se envía una respuesta desde el servidor en forma de recurso, que puede ser algo como HTML, XML, JSON.

Figura 2.3
Arquitectura cliente-servidor



Nota: Ejemplo de petición cliente-servidor; (Giancarlo Culcay, 2022)

Entre las ventajas de una API REST está la posibilidad de crear un cliente / servidor en diferentes lenguajes de programación. Esto nos da la capacidad de enviar y recibir información en diferentes formatos, aunque JSON (JavaScript Object Notation) se usa comúnmente (Figura 2.4).

Figura 2.4
Formato JSON

```
GET /users/1

[
  {
    "id":1,
    "name":"Root",
    "email":"root@admin.com",
    "created_at":"2022-09-26",
    "updated_at":"Sep 26, 2022",
    "cellphone":"73252051",
    "_roles":[
      1
    ],
    "roles":[
      {
        "id":1,
        "name":"root",
        "description":"Administrador del Sistema, Acceso Total",
        "created_at":"2022-09-27T02:16:52.000000Z",
        "updated_at":"2022-09-27T02:16:52.000000Z",
        "pivot":{
          "user_id":1,
          "role_id":1
        }
      }
    ]
  }
]
```

Nota: Ejemplo de respuesta JSON de una API.

2.4 Sistemas De Control De Acceso

2.4.1 Control de Acceso

El control de acceso físico puede permitir a un empleado ingresar a determinadas áreas dependiendo su jerarquía, a los proveedores y visitantes al departamento asignado, evitando así que acceden a áreas restringidas, en caso de intentarlo el sistema alerta al personal de seguridad. Un control de acceso consiste en proveer la habilidad para controlar, monitorear y restringir el movimiento de personas, bienes o vehículos en, dentro o alrededor de cualquier edificio o lugar, esto constituye

uno de los servicios de seguridad que es indispensable para una empresa ya que es el primer filtro para evitar que una situación de riesgo se presente dentro de las instalaciones. Se trata de una de las principales medidas de seguridad que se tienen que tomar en todas las compañías grandes, medianas o pequeñas. El control de acceso es uno de los sistemas de protección que, si se aplican correctamente, es de los más efectivos pues controla todo el flujo de personas que entran y salen del lugar en cuestión. Si a esto se le agrega un poco de tecnología, se puede obtener un sistema de registro de asistencia y al mismo tiempo un control de acceso a donde determinados empleados pueden o no ingresar.

2.4.2 Tecnología de Identificación por Radio Frecuencia (RFID)

Las siglas RFID hacen referencia a Radio Frequency IDentification, lo que en español conocemos como Identificación por Radiofrecuencia.

“RFID es posible hacer lectura simultaneas de objetos, productos, vehículos o personas, haciendo más eficiente los procesos de identificación, y para esto no es necesario que la etiqueta del producto este en línea con el lector” (Morales, 2012, pág. 22).

2.4.2.1 Componentes del Sistema RFID

El sistema RFID se compone principalmente por los siguientes elementos:

1. Etiqueta RFID, también llamada tag o transpondedor: Consta de un microchip, una antena y un transductor.

- El microchip almacena los datos y la pequeña antena posibilita la comunicación por radiofrecuencia con el lector.
- El transductor es el que convierte la información que transmite la antena.
- Esta etiqueta se inserta o adhiere en un objeto por ejemplo a un vehículo, llave, tarjeta, paquete, producto, planta, etc., animal o persona portando la información en su chip.

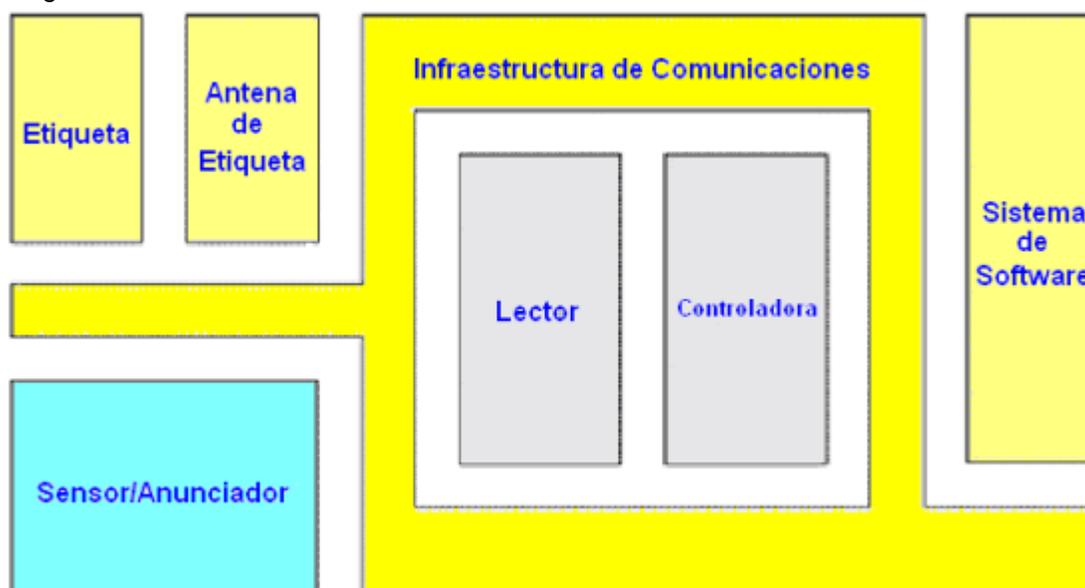
2. Lector de RFID o transceptor: compuesto por una antena, un transceptor y un decodificador. Transmite a la etiqueta la energía suficiente para excitarla.

- El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta, la cual contiene la información de identificación de esta, extrae la información y se la pasa al subsistema de procesamiento de datos.

3. Subsistema de procesamiento de datos o Middleware RFID: recoge, filtra y maneja los datos, es decir, proporciona los medios de proceso y almacenamiento de datos, haciendo de intermediario entre el lector y las aplicaciones empresariales.

4. Programadores RFID: son los dispositivos que realizan la escritura de información sobre la etiqueta RFID, es decir, codifican la información en un microchip situado dentro de una etiqueta RFID. La programación de las etiquetas se realiza una única vez si las etiquetas son de sólo lectura, o varias veces si son de lectura/escritura.

Figura 2.5
Diagrama Sistema RFID



Nota: Diagrama esquemático de un sistema RFID; (Morales, 2012)

2.5 Administración De Recursos Humanos

Dentro de una empresa o institución existe un departamento de recursos humanos, ya que es el único departamento que administra las personas que componen la empresa.

Las tareas propias de esta área son las siguientes: selección de los trabajadores, la promoción de los mismos y su formación, así como distintos procesos de organización. Todo ello con el objetivo de optimizar la productividad de una empresa.

Si bien el producto o servicio que gestiona una entidad es importante, la mayoría de empresas coinciden en hacer hincapié en el factor humano. En consecuencia, se establecen herramientas para evaluar todos los parámetros que forman parte de la dinámica de los recursos humanos. (Navarro, 2018)

2.5.1 Planilla

Se conoce como al documento en el que se registra su remuneración por los servicios que presta el trabajador a la empresa, están sujetos al régimen laboral de la actividad privada, como tal las empresas están obligadas a llevar planillas de pago de remuneración de los trabajadores.

Son registros auxiliares elaborados por la Contabilidad para llevar el control de la remuneración al personal permanente y eventual de una empresa.

Son documentos exigidos por diversos organismos que llevan el control del pago de salarios y los beneficios que conllevan: CNS, SIN, AFP, etc. La Planilla de Sueldos y Salarios de acuerdo a la estructura difundida por el Ministerio de Trabajo debe contener la siguiente información:

- Encabezado: Nombre y Razón Social del Empleador, Numero de registro del Empleador en el Ministerio de Trabajo, Numero de Identificación Tributaria,

Numero de registro de Empleador en la Caja de Salud, además del periodo al que corresponde.

- Estructura de la planilla: Carnet de identidad, Nombres y Apellidos, Nacionalidad, Fecha de Nacimiento, Sexo, Cargo, Fecha de Ingreso, Días pagados, Horas pagados, Haber Básico, Bono de Antigüedad, Horas Extras, Bonos, Dominicales, Total ganado, Descuentos por AFPs, RC-IVA y Otros. Finalmente, el líquido pagable y las firmas.
 - ✓ Haber básico: Es importante señalar que este importe anualmente es ajustado por el ministerio de Trabajo.
 - ✓ Bono antigüedad: Este bono se cancela por la antigüedad que tiene un trabajador en una empresa, de acuerdo al Artículo 60 del D.S. 21060, que sigue continúa aplicándose los porcentajes aplicables sobre el salario mínimo nacional.
 - ✓ Otros bonos: En cuanto a los otros bonos como ser el bono de producción es aplicable a empresas productivas. Por otro lado, según el D.S.211137 de 30 de noviembre de 1985 que reglamenta al D.S. 21060, los bonos en el sector público quedaron eliminados. Otro que merece un especial análisis, es el pago por las horas extraordinarias y trabajos en feriados que se pagan con el recargo del 100%, de acuerdo al artículo 55 de la L.G.T. El trabajo nocturno en las mismas condiciones con el recargo entre el 25% al 50%. En cuanto al trabajo en domingo se paga el triple, de acuerdo al D.S.3691 de 3 de abril de 1954. (Ministerio de Trabajo)

2.5.1.1 Cálculo de Salarios

Sueldo: Podemos definir qué sueldo es son los pagos que se realizan a los trabajadores según su cargo y las horas de trabajo que realiza cada individuo. El horario define el monto que se pagara al trabajador según el cargo que desempeña.

Haber básico = sueldo

Incremento salarial: para ejecutar el cálculo se realiza la siguiente operación X (es el incremento de la presente gestión en %) multiplicado por (el haber mensual) y dividido entre 100(el total del haber mensual). Se lo calcula basándose en la siguiente formula.

$$\text{Nuevo Sueldo} = (X \% * \text{Haber básico}) / 100\%$$

Bonos: el bono de antigüedad se debe colocar la fecha de ingreso y la fecha final a la que se quiere calcular el tiempo de antigüedad en el trabajo. El bono de antigüedad se lo calcula basándose en la siguiente formula.

$$\text{BA} = \text{Salario Mínimo Nacional} * 3 * \text{Porcentaje}$$

Art 60° Substitución de toda otra forma porcentual de aplicación del bono de antigüedad, se establece la siguiente escala única aplicable a todos los sectores laborales.

Horas extras: Es la jornada que va desde las 20:00 hasta las 6:00 del día siguiente duración máxima es de 7 horas.

Art. 29° A los fines del artículo 41 de la ley, se consideran feriados declarados por ley: los domingos; el 1° de enero; lunes y martes de carnaval; jueves y viernes santo; el 1° de mayo; el día de Corpus Christi; el 6 de agosto; el 1° y 2 de noviembre; el 25 de diciembre. Los días declarados de duelo o de regocijo no se consideran feriados para los efectos de la ley.

Art. 30° Durante los días domingos y feriados no podrán realizarse otros trabajos que los especificados en el Decreto Supremo de 30 de agosto de 1927.

Art. 31° Los trabajadores que hubiesen trabajado en días feriados, por estar comprendidos sus servicios en algunas de las excepciones indicadas en el artículo 30, tendrán derecho, a elección del patrono, a una compensación de descanso con otro día de la semana en curso o para ser pagados con un 100% de recargo sobre el salario normal.

Art. 32° El descanso de dos horas a la mitad del día feriado, a que se refiere el artículo 42 de la ley, no tendrá lugar cuando el trabajo se efectúe por equipos, por

razones de interés público o por la naturaleza de la tarea. (Mi Trabajo Bolivia, 2020)

Por otro lado, el Reglamento de la L.G.T señala que se debe llevar siempre un registro de las horas extras, aprobado por la Inspección General del Trabajo. Para ello, dividimos tu salario entre 30 días del mes y luego entre la duración de la jornada. Entre 8, en este caso:

$$\text{Sueldo por hora} = (\text{Sueldo básico} / 30 \text{ días}) / 8 \text{ horas}$$

Salario por horas trabajadas para el valor de horas extras multiplicamos por 2.

$$\text{Sueldo por hora} * 2$$

Ahora, este valor lo multiplicamos por las horas extras acumuladas

$$\text{Horas extras} = (\text{Sueldo por hora} * 2) * \text{Horas extras acumuladas}$$

Descuentos laborales: Por lo general los aportes de Ley son de carácter social, para las empresas grandes, medianas o pequeñas, mediante estas normas legales los empleadores y trabajadores están en la obligación de realizar aportes o pagos emergentes de la relación laboral a las instituciones correspondientes; así lo dispone el Art. 42 del D.R.L.G.T. (Mi Trabajo Bolivia, s.f.) Constituyen deducciones legales realizadas por la empresa en proporción al total ganado mensual a cuenta de otros organismos como ser:

- **aporte de Vejez:** 10.00% s/ total ganado (Art. 88 Ley 065 de Pensiones)
- **Riesgo Común:** 1,71% s/ Total ganado (Art. 83 Ley 065 de Pensiones)
- **Comisión:** 0,50% s/ Total ganado (Art. 151 Ley 065 de Pensiones)
- **AFP- Aporte Laboral Solidario:** 0,50% s/ Total ganador (art. 94 Ley 065 de Pensiones)
- **Sub total** = 12.71%
- **Aporte Patronal Solidario:** 3% s/Total Ganado (Ley 065 de Pensiones) Las Empresas Minero Metalúrgicas, además tienen la obligación de aportar el, 2% sobre el Total Ganado, para el Fondo Solidario Minero, creado por la Ley 065. Por lo que, con este aporte, se tendría en este tipo de empresa, un total aporte patronal del 18.71%. (Mi Trabajo Bolivia, s.f.)

Aguinaldo: Aquí podrás calcular cuánto recibirás por concepto del aguinaldo de Navidad, la misma modalidad se aplica para el segundo aguinaldo “Esfuerzo por Bolivia”.

$$\text{Aguinaldo} = (\text{Total ganado} / 12) * (\text{meses trabajados})$$

Para aquellos empleados y obreros que no hubiesen trabajado los doce meses completos, su pago se realizará por duodécimas en proporción al tiempo trabajado en la gestión, con la condición que los empleados hayan trabajado más de tres meses calendario y los obreros más de un mes calendario. (Ferrere, 2019)

$$\text{Aguinaldo} = ((\text{Total ganado} / 12) * (\text{meses trabajados})) + ((\text{Total ganado} / 365) * (\text{días trabajados}))$$

Figura 2.6

Ejemplo de formato de planilla de sueldos

NOMBRE O RAZÓN SOCIAL										N° DE NIT		Página		de																		
N° IDENTIFICADOR DEL EMPLEADOR ANTE EL MINISTERIO DE TRABAJO										N° DE EMPLEADOR (Caja de Salud)																						
PLANILLA DE SUELDOS Y SALARIOS																																
(En Bolivianos)																																
N°	Documento de identidad	Apellidos y nombres	País de nacionalidad	Fecha de nacimiento	Sexo (V/M)	Ocupación que desempeña	Fecha de ingreso	Horas pagadas (Día)	Días pagados (Mes)	(1) Haber básico	(2) Bono de Antigüedad	(3) Bono de producción	(4) Subsidio de frontera	(5) Trabajo extraordinario y nocturno	(6) Pago dominical y domingo trabajado	(7) Otros bonos	(8) TOTAL GANADO Suma (1 a 7)	(9) Aporte a las AFPs	(10) FC-IVA	(11) Otros descuentos	(12) TOTAL DESCUENTOS Suma (9 a 11)	Mes	Año	(14) Firma								
																									(13) LÍQUIDO PAGABLE (12-8)							
1																	0.00				0.00											
2																	0.00				0.00											
3																	0.00				0.00											
20																	0.00				0.00											
TOTALES										0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00									
Personas con discapacidad																																
N°	CI	Fecha de nacimiento	Nombre completo	Figura en planilla	N° del dependiente laboral relacionado	Relación del dependiente laboral con la persona con discapacidad																										
1																																
2																																
3																																
NOMBRE DEL EMPLEADOR O REPRESENTANTE LEGAL					N° DE DOCUMENTO DE IDENTIDAD					FIRMA					FECHA																	

Nota: Ejemplo de formato de una planilla; (Ministerio de Trabajo, 2022)

2.5.2 Remuneración

Podemos mencionar que remuneración es el pago en efectivo o especie del esfuerzo que una persona realiza dicho trabajo.

La remuneración o salario según el reglamento de la ley general de trabajo del decreto supremo N°224 de 23 de agosto de 1943, en su capítulo IV de las remuneraciones indica en sus artículos y tomado en cuenta como base estos términos.

Art. 39º, Remuneración o salario es el que percibe el empleado o trabajador en dinero, en pago de su trabajo, incluyéndose en esta denominación, las comisiones y participaciones en los beneficios, cuando éstos invistan carácter permanente.

2.5.3 Salario

Es la forma de pago que se puede realizar aun empleado o trabajador que realiza función dentro de una institución o empresa por la cual se realiza el pago.

Existen diferentes tipos de sistema de salario las cuales podemos listar y mencionar los siguientes:

- Salario por tiempo
- Salario a destajo(tarea)o por pieza
- Salario nominal
- Salario real
- Salario básico
- Salario justo
- Salario mínimo nacional
- Salario mínimo departamental
- Salario mínimo vital
- Salario minino vital con escala móvil.

Salario mínimo Nacional, El Salario Mínimo Nacional dispuesto a través del Decreto Supremo 4501 es de 2.164 bolivianos, que representa un incremento del 2% con relación al establecido en la gestión 2019, deberá ser pagado de manera retroactiva de enero a abril y cancelado hasta el 31 de mayo de este año por los empleadores, según establece la Resolución Ministerial 512/21 emitida por el Ministerio de Trabajo.

2.6 Ingeniería De Software

“La ingeniería de software es una disciplina que se interesa por todos los aspectos en la elaboración de un software desde las primeras etapas de la

especificación del sistema hasta el mantenimiento después que se pone en operación” (Someryille, 2011).

Es el proceso utilizado para crear, implantar y mantener aplicaciones y Sistemas Web de alta calidad. Esta breve definición nos lleva a abordar un aspecto clave de Cualquier proyecto como es determinar qué tipo de proceso es más adecuado en función de las características del mismo. El desarrollo de aplicaciones Web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y Sistemas de información. Uno de los aspectos más tenidos en cuenta, en el desarrollo de sitios web es sin duda alguna el diseño gráfico y la organización estructural del contenido. En la actualidad la web está sufriendo grandes cambios, que han obligado a expertos en el tema a utilizar herramientas y técnicas basadas en la ingeniería del Software, para poder garantizar el buen funcionamiento y administración de los sitios web. Para garantizar el buen funcionamiento y mantenimiento de los sitios web, este debe contar con ciertos atributos y características que en conjunto forman un concepto muy importante, para alcanzar el éxito en cualquier organización, herramienta, y todo aquello que se pueda considerar como servicio. Dicho concepto es la calidad, que con atributos como, usabilidad, navegabilidad, seguridad, Mantenibilidad, entre otros, hace posible por un lado la eficiencia del artefacto web y por ende la satisfacción del usuario final. (López, 2010)

En este sentido, la ingeniería de la Web hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones Web Complejas y de gran dimensión en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones.

2.7 Metodología De Desarrollo

2.7.1 Metodología Tradicional

Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

2.7.2 Metodología Ágil

Por definición, las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

Los objetivos del modelo ágil incluyen:

- Mejorar la satisfacción del cliente.
- Ser adaptable y abierto al cambio.
- Cooperación en equipos de desarrollo.
- Creación eficiente de software de trabajo.
- Racionalización del proceso de desarrollo.
- Lanzamiento de productos y actualizaciones más rápido.

2.7.3 Metodología Kanban

La palabra Kanban significa “tablero” o “tarjeta visual” en japonés. Éste se refiere a un sistema de tarjetas que ayuda a visualizar el estado en el que está cada actividad o tarea. Fue concebido para las líneas de producción de Toyota en la década del setenta y hace algunos años, fue adoptado en el mercado de TI con el fin de aplicarlo para el desarrollo de software. Su objetivo es organizar y gestionar de manera

general, la forma en la que se van completando las tareas. En los últimos años, se ha utilizado en la gestión de proyectos de desarrollo de Software, asociado a la metodología Lean y ha logrado un amplio grado de aceptación y utilización dentro del mercado.

El término viene del japonés: Kan, visual, y ban, tarjeta. Kanban permite visualizar el flujo de trabajo en una barra de tareas a través de tarjetas. Propone distribuir las mismas en una serie de columnas. Kanban trabaja con tableros que pueden ser tanto físicos como digitales y permite una visualización clara de todas las tareas a realizar, en qué etapa está cada una y quién es el encargado de las mismas. (Brown & Rocher, 2013)

El Kanban es un sistema de gestión del trabajo en curso, que sirve principalmente para asegurar una producción continua y sin sobrecargas en el equipo de producción multimedia. El Kanban es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir. (Fitzgerald, 1999)

El Kanban es un sistema de gestión del trabajo en curso (WIP), que sirve principalmente para asegurar una producción continua y sin sobrecargas en el equipo de producción multimedia. El Kanban es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir. (Avison & Fitzgerald, 1999)

2.7.3.1 Características

Según Ibarra Guzmán & Castañeda Islas (2014) las principales características de Kanban se lo pueden observar en la tabla.

Tabla 2.1
Características Kanban

Características	Descripción
Visualiza el flujo de trabajo	Divide el trabajo en bloques, escribe cada elemento en una tarjeta y se coloca en el tablero. Utiliza columnas con nombre para ilustrar donde este cada elemento en el flujo de trabajo.
Limita el trabajo en curso	Asigna limites concretos al número de elementos que pueden estar en progreso en cada estado del flujo de trabajo.
Mide el tiempo de ciclo medio para completar un elemento	Optimiza el proceso para que el lead time sea tan pequeño y predecible como sea posible.

Nota: Descripción de las características kanban; (Isla & Castañeda, 2014)

2.7.3.2 Roles Kanban

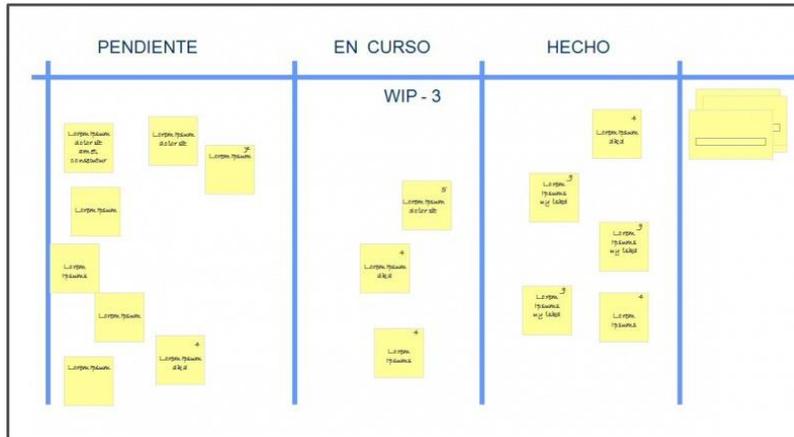
Kanban, a diferencia de otras metodologías ágiles como Scrum, no prescribe roles o reuniones. Pero existen una serie de roles que pueden ser adoptados en Kanban, estos han sido creados a partir de la observación en organizaciones que han utilizado el método y lo han evolucionado de forma colaborativa. (Palacios, 2016)

- **Service Request Manager:** Se encarga de gestionar la demanda y los requisitos dentro del sistema Kanban, manejando las relaciones con los stakeholders y fomentando la transparencia dentro del sistema en torno a la priorización del trabajo. Alternativamente, este se puede llamar Product Owner.
- **Service Delivery Manager:** Es responsable del flujo de trabajo dentro de un sistema Kanban y/o determinados ítems de trabajo y facilita el Kanban Meeting y el Delivery Planning. Algunos nombres alternativos son Flow Manager, Delivery Manager o incluso Flow Master.

2.7.3.3 Tablero Kanban

Los tableros Kanban, que se desprenden de la metodología del mismo nombre, son herramientas visuales para la organización de las tareas, figura 2.7.

Figura 2.7
Tablero Kanban



Nota: Ejemplo de un tablero kanban; (Velásquez & Cornejo, 2015)

Un tablero de Kanban en su concepción más simple es un diagrama o rectángulo dividido en columnas, las cuales mantienen un número finito de tareas en ejecución (WIP) y las tarjetas, las cuales emulan las actividades que realiza cada miembro del equipo de trabajo. (Cornejo Velásquez, 2015)

Tabla 2.2
Elementos tablero Kanban

Elementos	Descripción
Tareas o actividades a realizar	Se representan en el tablero mediante tarjetas haciendo alusión al origen etimológico de la palabra Kanban. Las tarjetas que representan cada tarea o actividad se mueven a través de las columnas describiendo el estado en el que se encuentra.
Columnas	Estas representan las fases o etapas por las que debe pasar una actividad antes de considerarse terminada. Definir el número de columnas que debe tener un tablero Kanban siempre dependerá de las características del proyecto al que se quiera aplicar y del manual de procesos que se tenga en cada ente organizacional, ya que, como se mencionó anteriormente, la

intención del Kanban no es contaminar ni cambiar el flujo y las formas de trabajo sino aportar una representación visual al mismo.

WIP

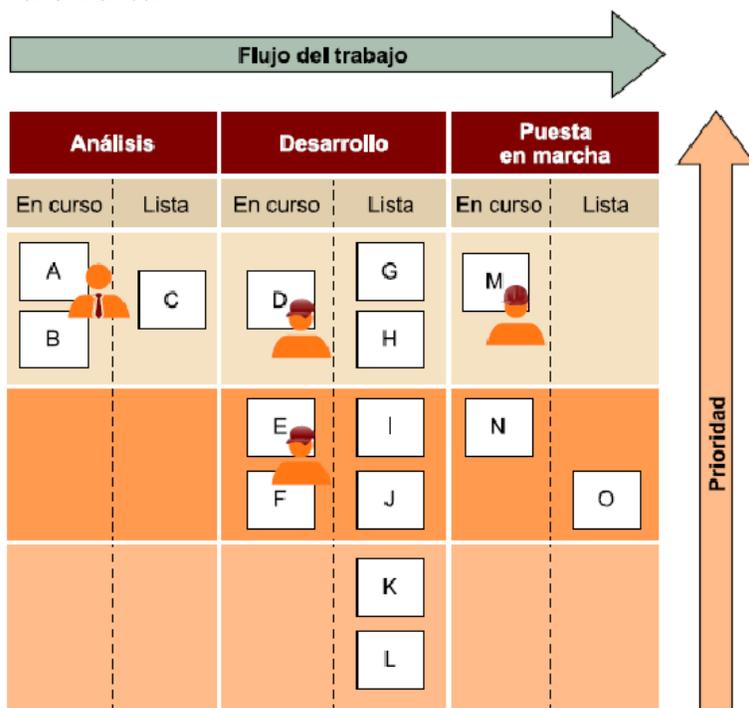
No es propiamente un elemento tangible dentro del tablero, pero sin el cual no se podría tener un control sobre las actividades máximas soportadas por el equipo de trabajo. Esto es tal vez lo que más dificultad presenta al momento de implantar un tablero Kanban o la metodología por si misma debido a que nunca se tiene definido el número finito de tareas o actividades que se deben realizar por cada fase que compone el proceso de desarrollo del proyecto

Nota: Elementos de un tablero kanban; (Sundén & Hammarberg, 2014)

2.7.3.4 Fases Kanban

En el tablero Kanban constituido por 3 columnas, que representan las diferentes fases por las que una tarea tiene que pasar para ser terminada.

Figura 2.8
Panel Kanban



Nota: Ejemplo de las fases de un tablero kanban; (Bermejo, 2010)

En la imagen se puede observar un panel constituido por tres columnas, que representan las diferentes fases por las que una tarea tiene que fluir para ser desarrollada (análisis, desarrollo y puesta en marcha).

Cada fase se encuentra subdividida en dos estados, que son en curso y lista, para pasar a la siguiente fase; esta división está representada por la línea discontinua de cada fase. El estado en curso significa que el equipo está actualmente trabajando en esta tarea, en esta fase y el estado lista significa que el equipo ya ha acabado el trabajo que tenía que ejecutar en esta fase y la tarea está esperando a que el sistema pueda asumirla para la siguiente fase. Esta división nos ayuda a localizar atascos en el proceso de producción. (Morales, 2015)

Las filas podrán representar diferentes proyectos en los que la empresa trabaja, siendo lo habitual que las filas indiquen prioridad, donde las tareas más superiores son las más prioritarias.

2.8 UWE – Ingeniería Web Basada en UML

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. (Maximilians, 2016)

De acuerdo a lo definido anteriormente la metodología UWE nos proporciona una extensión del uso de los modelos UML aplicado al análisis y diseño de sistemas orientados a la web, nos proporciona una guía de la manera de encarar este tipo de proyectos.

Modelo de UWE

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro de marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

- Un Modelo de Requerimientos que captura los requerimientos del sistema, en este caso es aplicable el modelo de casos de uso.
- Un Modelo conceptual para el contenido (modelo de contenido).
- Un Modelo de Proceso.
- Un Modelo de Navegación que comprende la estructura de la navegación.
- Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo de flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).

2.8.1 Modelo de Requerimientos

El primer paso para el desarrollo de un sistema web que se especificará con UWE es realizar la identificación de los requerimientos y plasmarlos en un modelo de requerimientos. Los requerimientos pueden ser documentados en diferentes niveles de detalle para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UML. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados. Los casos de uso fueron propuestos por el Proceso de Desarrollo de Software Unificado (RUP) para capturar los

requerimientos del sistema. (Schmuller, 2010)

Figura 2.9
Modelo de Caso de Uso UWE

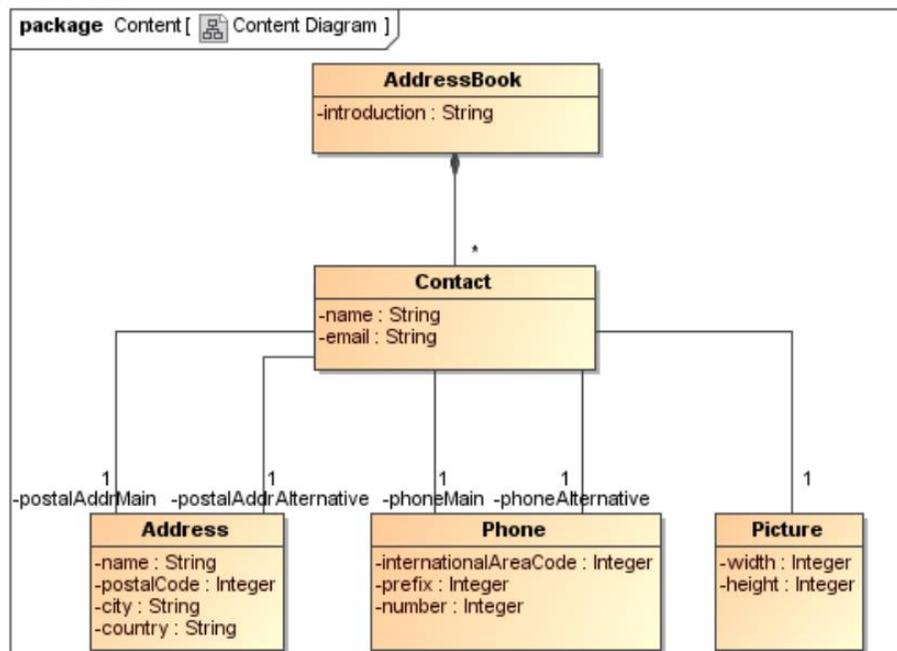


Nota: Ejemplo de un caso de uso; (Schmuller, 2010)

2.8.2 Modelo de contenido

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio Web. Su objetivo es construir un modelo de contenido del dominio de la aplicación considerando los requisitos reflejados en los casos de uso.

Figura 2.10
Modelo de contenido UWE



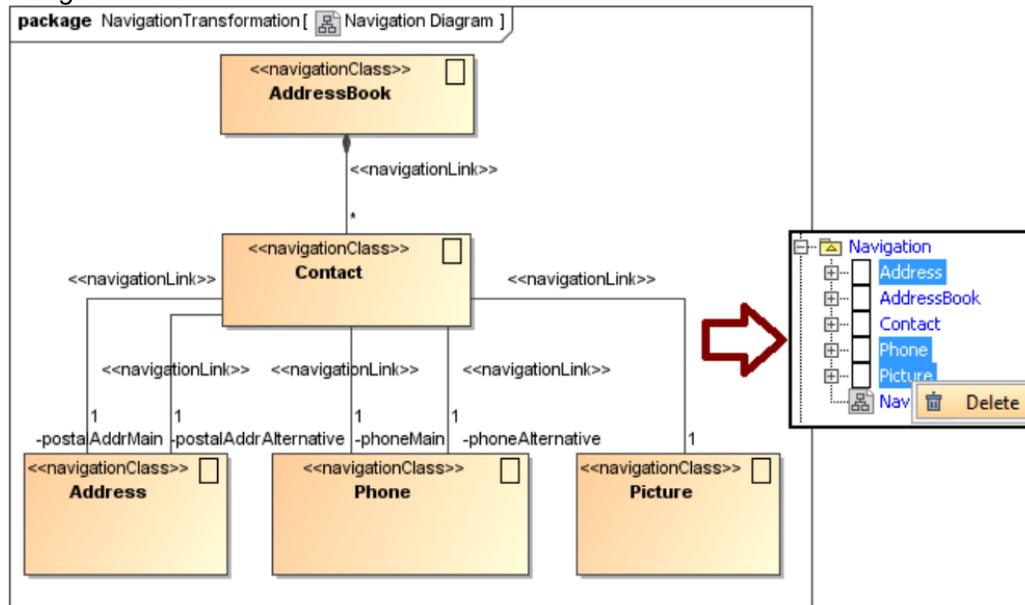
Nota: Ejemplo de contenido UWE; (Web Engineering Group, 2020)

2.8.3 Modelo de Navegación

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos (nodes) y enlaces (links).

Nodos son unidades de navegación y están conectados por medio de enlaces. Las mismas pueden ser presentadas en diferentes páginas o en una misma.

Figura 2.11
Modelo de navegación UWE



Nota: Ejemplo de navegación UWE; (Web Engineering Group, 2020)

En este modelo se especifica la relación interna del sitio web, es decir cómo se relaciona cada página web con las demás, con lo cual, en definitiva, es como se navega por el sitio web. Para ello, se utilizan unidades de navegación llamadas nodos conectados por enlaces de navegación.

Figura 2.12
Nombres de estereotipos y sus iconos

nombres de estereotipos y sus iconos	
 clase de navegación	 menú
 índice	 pregunta
 visita guiada	 clase de proceso
 nodo externo	

Nota: Listado de nombre de los estereotipos; (Web Engineering Group, 2020)

2.8.4 Modelo de Proceso

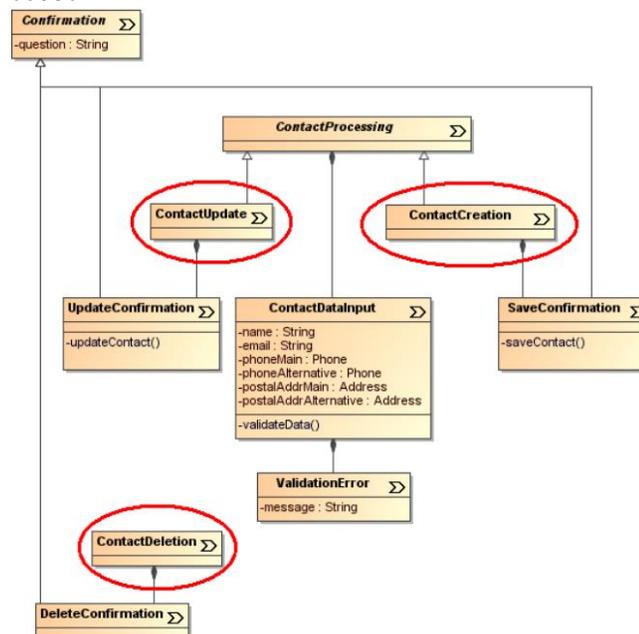
En este modelo se definen las acciones que realizan las clases del proceso (operacionales) específicas en el modelo de navegación, como se explicó anteriormente, el modelo del proceso se divide en dos partes:

- **Modelo de estructura del proceso:** en el cual se incluyen las relaciones entre las clases de proceso, se crea un diagrama de clases donde cada una se presenta con un estereotipo de clases de proceso.
- **Modelo de flujo del proceso:** la conducta de un proceso es representado mediante un diagrama de actividades UML, describiendo el flujo de una clase del proceso, lo que sucede cuando un usuario navega hacia una clase de proceso.

(Web Engineering Group, 2020)

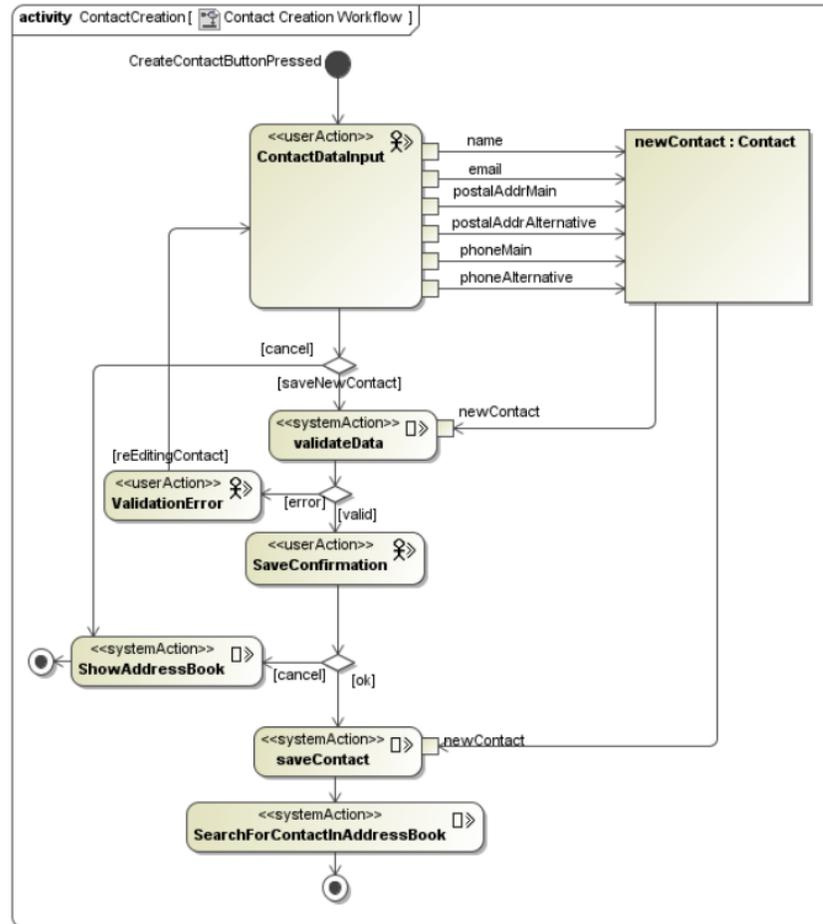
Figura 2.13

Modelo de estructura del Proceso



Nota: Ejemplo de estructura del Modelo de Proceso; (Web Engineering Group, 2020)

Figura 2.14
Modelo de estructura del Flujo del Proceso



Nota: Modelo de estructura del Flujo del Proceso; (Web Engineering Group, 2020)

2.8.5 Modelo de Presentación

El modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenece a una página web, para esto se describe la utilización del diagrama de Presentación. En este modelo presentan las clases de navegación de procesos que pertenecen a cada página web. Las notaciones de UWE son más claras y están mejor documentadas. UWE tiene como ventaja adicional que es un profile de UML, es decir, un modelo navegacional en un diagrama de clases UML con estereotipos.

Figura 2.15

Nombre de estereotipos y sus iconos

nombres de estereotipos y sus iconos	
 grupo de presentación	 página de presentación
 texto	 entrada de texto
 ancla	 fileUpload
 botón	 imagen
 formulario	 componente de cliente
 alternativas de presentación	 selección

Nota: Listado de estereotipos del Modelo de Presentación; (Web Engineering Group, 2020)

2.9 Métricas De Calidad

La calidad de un producto viene determinada por la satisfacción de las exigencias del usuario para el cual fueron diseñados, cumpliendo con sus expectativas y necesidades. Asegurar la calidad del software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisface los requerimientos del cliente o usuario.

Así también Pressman (2010) afirma que la calidad de software se define como: “El proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan” (p. 340).

Por lo tanto, podemos definir que la calidad de software cumple con los requisitos de acuerdo al modelo de calidad aplicado de esta manera satisfaciendo las necesidades del usuario o cliente.

2.9.1 Métricas de calidad establecida estándar ISO/IEC-25000

Las normas ISO/IEC 25000 también llamadas SQuare (Requisitos y Evaluación de Calidad de Productos de Software) están conformadas por las normas ISO/IEC 9126 e ISO/IEC 14598, surgen para crear modelos, métricas, procesos y herramientas de evaluación de calidad del software como producto, por medio de la especificación de los requisitos. Es una familia de normas que tiene por objetivo la creación de un

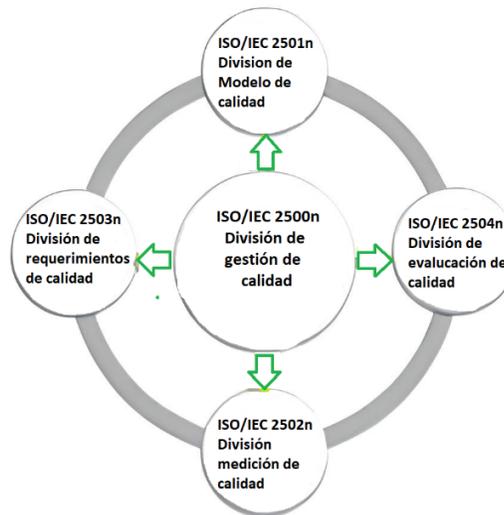
marco de trabajo común para evaluar la calidad del producto software. (ISO/IEC 25000, 2005)

2.9.2 Estructura de la familia ISO/IEC 25000

La ISO/IEC 25000 se encuentra compuesta por varias divisiones, entre las que se destacan las siguientes

Figura 2.16

Estructura de la familia ISO/IEC 25000



Nota: Estructura de la familia ISO/IEC 25000; (ISO/IEC 25000, 2005)

2.9.3 Medición de Calidad ISO/IEC 25010

Esta norma nos ayuda a validar las funcionalidades básicas de los sistemas informáticos, adoptando las características de calidad a la hora de su evaluación.

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad. (NORMA ISO 25010, 2019)

Figura 2.17
Características de la ISO 25010



Nota: Características de calidad de producto software; (Norma ISO 25010, 2019)

2.9.3.1 Adecuación funcional

Representa la capacidad del producto o sistema software para proporcionar las funciones necesarias para satisfacer al usuario. Esta característica se divide en las siguientes sub-características:

- **Completitud funcional:** Capacidad del sistema software para proporcionar un conjunto de funcionalidades apropiadas para cubrir todas las tareas y objetivos determinados por el usuario.
- **Exactitud funcional:** Capacidad del sistema software para proporcionar los resultados correctos con el grado necesario de precisión.

(NORMA ISO 25010, 2019)

2.9.3.2 Fiabilidad

Capacidad del producto o sistema software para realizar las funciones específicas cuando se utiliza bajo ciertas condiciones y periodos de tiempo determinadas. Esta característica se divide en las siguientes sub - características:

- **Madurez:** Capacidad del sistema para satisfacer las necesidades de fiabilidad durante el funcionamiento normal.
- **Disponibilidad:** Capacidad de un sistema de estar operativo y accesible para su uso cuando se necesite.
- **Tolerancia a Fallos:** Capacidad de un sistema para operar cuando se presenten fallos.
- **Recuperabilidad:** Capacidad de un sistema para restablecer el estado del

sistema y recuperar datos que se hayan afectado, en caso de fallo.

2.9.3.3 Eficiencia en el desempeño

Capacidad de un producto o sistema software de proporcionar un rendimiento apropiado, respecto a la cantidad recursos utilizados bajo determinadas condiciones.

Esta característica se divide en las siguientes sub-características:

- **Comportamiento Temporal:** Capacidad de un sistema software para proporcionar los tiempos de respuesta y procesamiento apropiados.
- **Utilización de Recursos:** Capacidad en que un sistema software utiliza las cantidades y tipos de recursos adecuados.
- **Capacidad:** Capacidad de un sistema software de cumplir con los requisitos determinados.

(NORMA ISO 25010, 2019)

2.9.3.4 Facilidad de uso

Capacidad del sistema software para que sea entendido, aprendido, agradado y usado por el usuario. Esta característica se divide en las siguientes sub-características:

Capacidad de reconocer su adecuación: Capacidad del sistema software que permite al usuario entender si el software es adecuado para sus necesidades.

- **Capacidad para ser entendido:** Capacidad del sistema, que permite al usuario entender si el software es adecuado para alcanzar sus objetivos determinados.
- **Operatividad:** Capacidad de un sistema software que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores del usuario:** Capacidad en que el sistema brinda la protección necesaria contra errores que realizan los usuarios.
- **Estética de la Interfaz del usuario:** Capacidad en que la interfaz de usuario llega a satisfacer y agradar al usuario.
- **Accesibilidad técnica:** Capacidad del sistema software para que se permita ser utilizado por usuarios con determinadas discapacidades.

(NORMA ISO 25010, 2019)

2.9.3.5 Seguridad

Capacidad de proteger la información y los datos, de manera que personas o sistemas no autorizados puedan tener acceso para consultas o actualizaciones. Esta característica se divide en las siguientes sub-características:

- **Confidencialidad:** Capacidad de proteger la información y el acceso a datos no autorizados, ya sea de manera accidental o intencional.
- **Integridad:** Capacidad de un producto, sistema o componente software para evitar accesos no autorizados a datos o programas de computación.
- **No – repudio:** Capacidad para demostrar que los eventos han ocurrido, de manera que dichos eventos no puedan ser refutados posteriormente.
- **Responsabilidad:** Capacidad de dar seguimiento a las acciones que fueron realizadas por una entidad.
- **Autenticidad:** Capacidad de demostrar la identidad de un sujeto o un recurso.

(NORMA ISO 25010, 2019)

2.9.3.6 Compatibilidad

Capacidad de dos o más sistemas software, para llevar a cabo sus funciones intercambiando información mientras comparten el mismo entorno. Esta característica se divide en las siguientes sub-características:

- **Co – existencia:** Capacidad de un sistema software para coexistir en un entorno en el cual comparten recursos comunes con otro software independiente.
- **Interoperabilidad:** Capacidad de dos o más sistemas software para intercambiar información y utilizar dicha información.

(NORMA ISO 25010, 2019)

2.9.3.7 Mantenibilidad

Capacidad del sistema software para ser modificado o actualizado debido a necesidades evolutivas y correctivas. Esta característica se divide en las siguientes sub-características:

- **Modularidad:** Capacidad de un sistema software que cuando sea modificado no afecte a otras funcionalidades del sistema.
- **Reusabilidad:** Capacidad de un activo (Información, Software, Hardware, Usuarios) para ser utilizado en más de un sistema o en la construcción de otros activos.
- **Capacidad de ser analizado:** Facilidad con la que se puede llevar a cabo un análisis del impacto de una determinada modificación en el sistema.
- **Capacidad de ser modificado:** Capacidad del sistema para permitir que sea modificado sin causar daños o reducir la calidad del producto existente.
- **Capacidad de ser probado:** Facilidad de realizar pruebas a un sistema o componente software, para determinar si se han cumplido con los requerimientos establecidos.

(NORMA ISO 25010, 2019)

2.9.3.8 Portabilidad

Capacidad de un sistema o componente software de ser trasladado de un entorno a otro sin que esto afecte la funcionalidad de cada sistema. Esta característica se subdivide en las siguientes sub-características:

- **Adaptabilidad:** Capacidad de un sistema software de ser adaptado a distintos entornos.
- **Capacidad de ser instalado:** Capacidad de un sistema para que pueda ser fácilmente instalado y/o desinstalado.
- **Capacidad de ser reemplazado:** Capacidad del sistema software para ser utilizado en lugar de otro sistema en el mismo entorno y cumpliendo con el mismo objetivo.

(NORMA ISO 25010, 2019)

2.10 Método De Estimación De Costo De Software

La elaboración de software conlleva varios procesos que son muy importantes para obtener un buen producto, y en este caso nos referimos particularmente al proceso de estimación de costos del software. “Una de las tareas de mayor

importancia en la administración de proyectos de software es la estimación de costos. Si bien es una de las primeras actividades, inmediatamente posterior al establecimiento de los requerimientos, se ejecutan regularmente a medida que el proyecto progresa con el fin de ajustar la precisión en la estimación” (Gómez, Migani, & Otazú, 2008, p.4).

2.10.1 Modelo COCOMO

Modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste). El modelo original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria.

CARACTERISTICAS

- Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.
- Representa el más extenso modelo empírico para la estimación de software.
- Existen herramientas automáticas que estiman costos basados en COCOMO.

Objetivos para la construcción de COCOMO II

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

Modelos de COCOMO II

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son:

1. **Modelo de composición de aplicación.** Utilizado durante las primeras etapas de la Ingeniería del software, donde el prototipado de las interfaces de usuario, la interacción del sistema y del software, la evaluación del rendimiento, y la evaluación de la madurez de la tecnología son de suma importancia.
2. **Modelo de fase de diseño previo.** Utilizado una vez que se han estabilizado los requisitos y que se ha establecido la arquitectura básica del software.
3. **Modelo de fase posterior a la arquitectura.** Utilizado durante la construcción del software.

(Mercado, 2018)

2.10.1.1 Modelos de Estimación de costo con COCOMO

En la estimación del tamaño de software COCOMO utiliza tres técnicas:

- Puntos Objeto,
- Puntos Función No Ajustados
- Líneas de Código Fuente.

Además, se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reusó, reingeniería, conversión y mantenimiento.

Puntos objeto: El procedimiento para determinar Puntos Objeto en un proyecto software se resume en:

1. Determinar Cantidad de Objetos: Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.
2. Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil).
3. Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de

complejidad.

4. Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

Tabla 2.18

Cálculo de Punto función no ajustado

Para Pantallas			
Cantidad y fuente de las tablas de datos			
Cantidad de Vistas Contenidas	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3 - 5 cliente)	Total 8 + (> 3 servidor > 5 cliente)
< 3	Simple	Simple	Media
3 - 7	Simple	Media	Difícil
> 8	Media	Difícil	Difícil

Para Reportes			
Cantidad y fuente de las tablas de datos			
Cantidad de Vistas Contenidas	Total < 4 (< 2 servidor < 3 cliente)	Total < 8 (< 2 - 3 servidor < 3-5 cliente)	Total 8 + (> 3 servidor > 5 cliente)
0 o 1	Simple	Simple	Media
2 o 3	Simple	Media	Difícil
4 +	Media	Difícil	Difícil

Nota: Cálculo de punto de función no ajustado; (García, 2012).

Puntos Función:

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post- Arquitectura.

Los puntos función están basados en información disponible en las etapas tempranas del ciclo de vida del desarrollo de software. COCOMO II considera solamente UFP (Puntos Función no ajustados).

$$FP = UFP \times TCF$$

Donde:

UFP: Puntos Función no Ajustados

TCF: Factor de Complejidad Técnica

Para calcular los UFP, se deben identificar los siguientes elementos:

- **Entradas Externas (Inputs):** Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno.
- **Salidas Externas (Outputs):** Salida de datos de usuario o de control que deja el límite del sistema de software.
- **Archivo Lógicos Internos (Archivos):** Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software.
- **Archivos Externos de Interface (Interfaces):** Archivos transferidos o compartidos entre sistemas de software.
- **Solicitudes Externas (Queries):** Combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa.

Una vez identificados los elementos se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto. Se asigna un peso a cada ítem según el tipo y el grado de complejidad correspondiente. Finalmente, los UFP son calculados sumando los pesos de todos los ítems identificados.

Líneas de Código Fuente (SLOC):

El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa. Definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes.

A los efectos de COCOMO II, se eliminan las categorías de software que consumen poco esfuerzo. Así no están incluidas librerías de soporte, sistemas operativos, librerías comerciales, etc., ni tampoco el código generado con generadores de código fuente.

Conversión de Puntos Función a Líneas de Código Fuente (SLOC):

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a líneas de código fuente considerando el lenguaje de implementación.

Por otro lado, el modelo COCOMO (CONstructive COst MOdel) Calcula esfuerzo y coste en función del tamaño del programa (LDC). COCOMO está definido para tres tipos de proyectos de Software:

1. **Modo orgánico:** Proyectos pequeños y sencillos, con equipos de experiencia en la aplicación y requisitos poco rígidos.
2. **Modo semi-acoplado:** Proyectos intermedios (más complejos), con equipos que poseen variados niveles de experiencia y requisitos más rígidos.
3. **Modo empotrado:** Proyectos que deben ser desarrollados en un conjunto de Hardware, Software y restricciones muy grandes. Según (Modelos de estimación, 2012).

Figura 2.19

Detalle de coeficientes de COCOMO II

PARA PUNTOS DE FUNCIÓN:			COCOMO BASICO				
DOMINIO	COMPLEJIDAD	PESO	Modo (Tipo de Proyecto)	a	b	c	d
Salidas	Alta	7	Orgánico	2.4	1.05	2.5	0.38
	Media	5					
	Baja	4					
Entradas	Alta	6	Semiacoplado	3.0	1.12	2.5	0.35
	Media	4					
	Baja	3					
Consultas	Alta	7	Empotrado	3.6	1.20	2.5	0.32
	Media	5					
	Baja	4					
Archivo Interno	Alta	15	$E = a * (KLDC)^b$ $D = c * (E)^d$				
	Media	10					
	Baja	7					
Archivo Externo / Interfaces	Alta	10	COCOMO INTERMEDIO				
	Media	7					
	Baja	5					
			$E = a * (KLDC)^b * FAE$ $D = 2.5 * (E)^{0.38}$				

Nota: Detalle de coeficientes de COCOMO II; (Modelos de estimación, 2012).

2.11 Seguridad

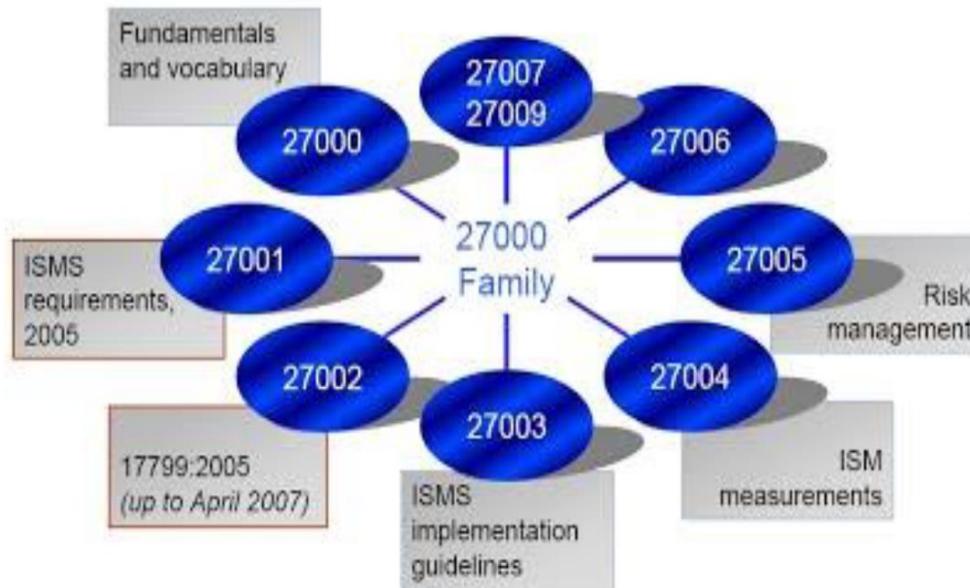
La seguridad de la información es ampliamente discutido y evidenciado en el medio corporativo, ya que la información se considera uno de los bienes más valiosos para las organizaciones, independientemente de su segmento o porte. La importancia de la información amplió la necesidad del desarrollo de estructura estandarizada para implantación y operación de los conceptos de seguridad de la información. En el marco de esta demanda, organizaciones mundiales (ISO/IEC) iniciaron el desarrollo de normas, originando la familia ISO 27000, que estandariza actividades relacionadas con la implantación y operación de Sistemas de Gestión de Seguridad de la Información (SGSI).

2.11.1 Estándar ISO/IEC 27000

La familia ISO/IEC 27000 se la conoce como serie ISO 27000, se desarrolla y publica por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC). La familia ISO 27000 contiene un conjunto de buenas prácticas para el establecimiento, implementación, mantenimiento y mejora de Sistemas de Gestión de la Seguridad de la información. (ISO/IEC 27000, 2018)

Esta norma proporciona una visión general de las normas que componen la serie 27000, ver Figura 14, indicando para cada una de ellas su alcance de actuación y el propósito de su publicación. Recoge todas las definiciones para la serie de normas 27000 y aporta las bases de por qué es importante la implantación de un SGSI, una introducción a los Sistemas de Gestión de Seguridad de la Información, una breve descripción de los pasos para el establecimiento, monitorización, mantenimiento y mejora de un SGSI.

Figura 2.20
Familia norma ISO/IED 27000



Nota: Familia norma ISO/IEC 27000; (ISO/IEC serie 27000)

2.11.1.1 ISO 27001

La ISO27001 desarrolla los conceptos de “calidad de la seguridad” y “madurez” como principales parámetros que dan la definición de indicadores y métricas. Los pasos que tienen que seguir para adoptar un conjunto de métricas son:

- Reconoce los elementos a medir a partir del índice del estándar ISO 1779.
- Define los niveles de madurez para cada elemento medible.
- Define los niveles de calidad para cada nivel de madurez de elemento.

Con ello obtendremos los valores de información que dará del nivel de evolución del control y calidad con la que se lleva a cabo la implementación, valores que pueden ser una imagen real del nivel de la seguridad de la empresa.

Se implanta de la ISO-27001 las actividades:

- Definición del alcance del sistema de Gestión de Seguridad de la Información.
- Define la política de Seguridad.
- Define los métodos y criterios para la Gestión y evaluación del riesgo.
- Identificación de los riesgos.

- Evaluación de los tratamientos de riesgos posibles.
- Desarrollo de la aplicabilidad de controles y requisitos.

(ISO/IEC 27000, 2018)

2.11.1.2 ISO 27002

Es una guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información. El estándar internacional ISO/IEC 27002 va orientado a la seguridad de la información de instituciones, empresas u organizaciones, de modo que las probabilidades de ser afectados por robo, daño o pérdida de información se minimicen al máximo.

2.12 Pruebas De Software

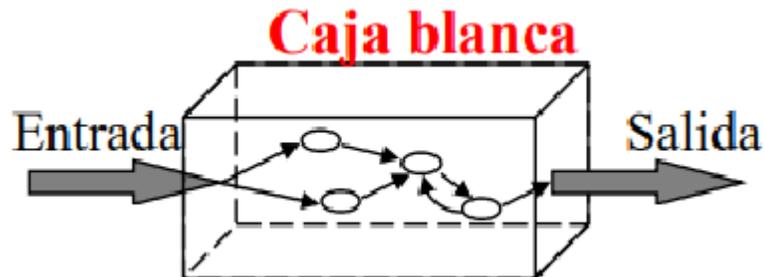
Las pruebas de seguridad están diseñadas para probar la vulnerabilidad en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor y de vuelta y el ambiente del lado del servidor. Cada uno de estos dominios puede recibir ataques y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo. (Pressman, 2005)

2.12.1 Pruebas de Caja Blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño 29 procedimental para obtener los casos de prueba. Basadas en información sobre cómo el software ha sido diseñado o codificado. (Blanco Bueno, 2007) Mediante los métodos de prueba de caja blanca, el ingeniero de software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa, ejecuten todos los ciclos en sus límites y con sus límites operacionales y ejerciten las estructuras internas de datos para

asegurar su validez. (Blanco Bueno, 2007)

Figura 2.21
Caja Blanca



Nota: Representación de E/S Caja Blanca; (Blanco Bueno, 2007)

2.12.2 Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas pruebas de comportamientos, se centran en los requisitos funcionales del software. Es decir, la prueba de caja negra permite al ingeniero del software obtener conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. (Blanco Bueno, 2007)

Figura 2.22
Caja Negra



Nota: Representación de E/S Caja Negra; (Blanco Bueno, 2007)

Las pruebas de caja negra intentan encontrar errores de las siguientes categorías.

- Funciones incorrectas o ausentes.
- Errores de interfaz.

- Errores de estructuras de datos o en acceso a base de datos externas.
- Errores de rendimiento.
- Errores de inicio y fin.

2.12.3 Pruebas de Estrés

Las pruebas de estrés, son utilizadas normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecutan una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema. Esto ayuda a los administradores determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

2.13 Herramientas De Desarrollo

Para el desarrollo del software se utilizarán las siguientes herramientas

2.13.1 Php

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts. (php.net, 2022)

2.13.2 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. (Eguiluz, 2019, p.5)

2.13.3 Framework Laravel

Laravel es un framework de código abierto para desarrollar aplicación y servicios web con PHP.

Laravel es un framework de aplicación web con una sintaxis expresiva y elegante. Un framework web proporciona una estructura y un punto de partida para crear aplicaciones, lo que le permite concentrarse en crear algo increíble mientras nos preocupamos por los detalles.

Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web. Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas.

Ventajas:

- Reducción de costos y tiempos en el desarrollo y mantenimiento.
- Curva de aprendizaje relativamente Baja (en comparación con otros framework PHP).
- Flexible y adaptable no solo al MVC Tradicional (Modelo vista controlador) sino que para reducir código propone usar “Routes with clousures”.

- Buena y abundante documentación sobre todo en el sitio oficial.
- Posee una amplia comunidad y foros.

2.13.4 Framework Vue.js

Vue es un framework de JavaScript para construir interfaces de usuario. Se basa en HTML, CSS y JavaScript estándar, y proporciona un modelo de programación declarativo y basado en componentes que lo ayuda a desarrollar interfaces de usuario de manera eficiente, ya sea simple o compleja.

composición API

Con la composición API, definimos la lógica de un componente utilizando funciones API importadas. En SFC, la API de composición se usa normalmente con `<script setup>`. El `setup` atributo es una pista que hace que Vue realice transformaciones en tiempo de compilación que nos permiten usar la API de composición con menos repeticiones. Por ejemplo, las importaciones y las variables/funciones de nivel superior declaradas en `<script setup>` se pueden usar directamente en la plantilla. (vuejs.org, 2022)

2.13.5 Framework Quasar

Quasar es un framework basado en Vue.js de código abierto con licencia del MIT, que le permite, como desarrollador web, crear rápidamente sitios web aplicaciones responsivos entre muchos:

- SPA (aplicación de una sola página).
- SSR (aplicación renderizada del lado del servidor) (+ adquisición de cliente PWA opcional).
- PWA (aplicación web progresiva).
- BEX (extensión del navegador).
- Aplicaciones móviles (Android, iOS, ...) a través de Cordova o Capacitor.
- Aplicaciones de escritorio multiplataforma (usando Electron).

El lema de Quasar es: escribir el código una vez y desplegarlo simultáneamente

como un sitio web, una aplicación móvil y/o una aplicación Electron. Sí, una base de código para todos ellos, que lo ayuda a desarrollar una aplicación en un tiempo récord mediante el uso de una CLI de última generación y el respaldo de las mejores prácticas, los componentes web ultrarrápidos de Quasar.

Cuando use Quasar, no se necesitará bibliotecas pesadas adicionales como Hammer.js, Moment.js o Bootstrap. Tiene esas necesidades cubiertas internamente. (Quasar.dev, 2022)

2.13.6 MariaDB

MariaDB Server es uno de los servidores de bases de datos más populares del mundo. Está hecho por los desarrolladores originales de MySQL y se garantiza que permanecerá como código abierto. Los usuarios notables incluyen Wikipedia, WordPress.com y Google.

MariaDB convierte los datos en información estructurada en una amplia gama de aplicaciones, desde banca hasta sitios web. Originalmente diseñado como un reemplazo directo mejorado para MySQL, MariaDB se usa porque es rápido, escalable y robusto, con un rico ecosistema de motores de almacenamiento, complementos y muchas otras herramientas que lo hacen muy versátil para una amplia variedad de casos de uso.

MariaDB se desarrolla como software de código abierto y como base de datos relacional proporciona una interfaz SQL para acceder a los datos. Las últimas versiones de MariaDB también incluyen funciones GIS y JSON.

(mariadb.org, 2022)

2.13.7 Módulo RFID-RC522

El RFID RC522 (Identificador por radiofrecuencia) es un conjunto de tecnologías diseñadas para leer etiquetas (tags) a distancia de forma inalámbrica. Los lectores RFID pueden ser conectados a un autómata o procesador como Arduino.

Las etiquetas RFID están disponibles en una gran variedad de formatos, tales como pegatinas adheribles, tarjetas, llaveros, pueden integrarse en un determinado producto o, incluso, insertarse bajo la piel en un animal o humano.

Especificaciones y características:

- Modelo: MF522-ED.
- Voltaje de alimentación: 3.3V
- Corriente de operación: 13-26mA
- Im máxima: 30mA.
- Frecuencia de operación: 13.56Mhz
- Distancia de lectura: 0 a 60mm.
- Protocolo de comunicación: SPI.
- Velocidad de datos máxima: 10Mbit/s
- Dimensiones: 40 x 60 mm
- Temperatura de operación: -20 a 80°C
- Humedad de operación: 5%-95%.
- Máxima velocidad de SPI: 10Mbit/s
- Incluye pines, llavero y tarjeta
- Corriente de reposo/stand by (Isb): 10-13mA a 3.3V
- Corriente en sleep-mode (Ism): <80uA
- Peso: 20 g

(uelectronics, 2022)

2.13.8 Módulo ESP8266MOD

NodeMCU ESP8266 es una plataforma de desarrollo similar a Arduino especialmente orientada al Internet de las cosas (IoT). La placa NodeMcu v2 ESP8266 tiene como núcleo al SoM ESP-12E que a su vez está basado en el SoC Wi-Fi ESP8266, integra además el conversor USB-Serial TTL CP2102 y conector micro-USB necesario para la programación y comunicación a PC. NodeMcu v2 ESP8266 está diseñado especialmente para trabajar montado en protoboard o soldado sobre una placa. Posee un regulador de voltaje de 3.3V en placa, esto permite alimentar la placa directamente del puerto micro-USB o por los pines 5V y GND.

La plataforma ESP8266 permite el desarrollo de aplicaciones en diferentes lenguajes como: Arduino, Lua, MicroPython, C/C++, Scratch. Al trabajar dentro del entorno Arduino podremos utilizar un lenguaje de programación conocido y hacer uso de un IDE sencillo de utilizar, además de hacer uso de toda la información sobre proyectos y librerías disponibles en internet.

Especificaciones:

- Voltaje de Alimentación: 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC (No usar 5V)
- Placa: NodeMCU v2 (Amica)
- Chip conversor USB-serial: CP2102
- SoM: ESP-12E (Ai-Thinker)
- SoC: ESP8266 (Espressif)
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Instruction RAM: 32KB
- Data RAM: 96KB
- Memoria Flash Externa: 4MB
- Pines Digitales GPIO: 17 (4 pueden configurarse como PWM a 3.3V)
- Pin Analógico ADC: 1 (0-1V)
- Puerto Serial UART: 2
- Certificación FCC
- Antena en PCB
- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado

(uelectronics, 2022)

CAPITULO III

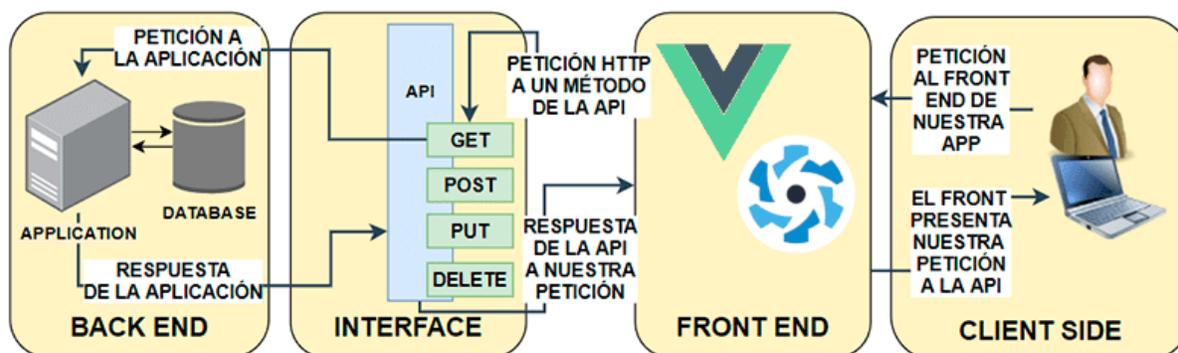
3 MARCO APLICATIVO

El análisis para el desarrollo de cualquier sistema es necesario contar con una metodología de desarrollo de software, por lo siguiente en este capítulo se describe el desarrollo del “Sistema De Información Web Para El Control De Acceso Y Asistencia Para El Proceso De Planillas Mediante Rfid” aplicando la combinación de la metodología ágil Kanban con la metodología UWE.

3.1 Esquema Del Sistema

En el siguiente grafico muestra el comportamiento del sistema con los usuarios involucrados con el sistema.

Figura 3.1
Esquema del sistema



Nota: Esquema del sistema.

3.2 Aplicación De La Metodología De Desarrollo

3.2.1 Combinación de metodologías Kanban – UWE

La metodología Kanban nos permite visualizar el flujo de trabajo y la carga de trabajo, lo cual nos permite saber en qué fases tenemos cuellos de botella.

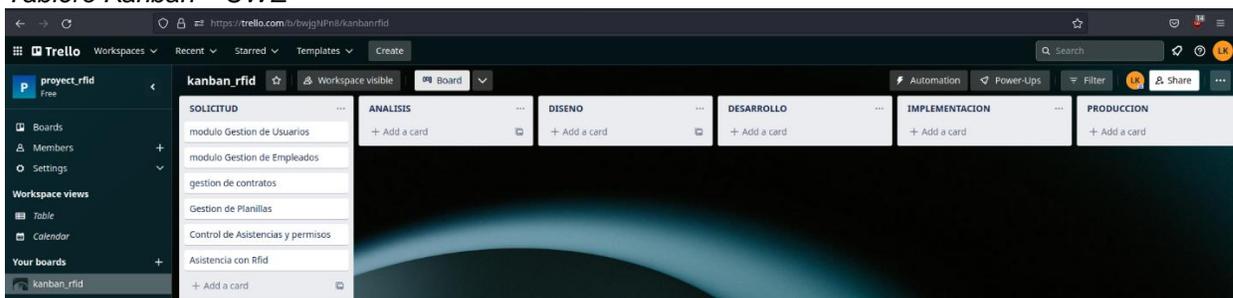
Para la construcción del tablero Kanban tendremos 6 columnas, estas fases nos permitirán tener un flujo de trabajo para la finalización del desarrollo del sistema, estas son descritas a continuación.

- **Solicitud:** Esta fase representa todos los requerimientos que se necesita para el desarrollo del sistema.

- **Análisis:** En esta fase se hace el análisis del sistema, hallando los actores involucrados, se hace el uso del análisis de requerimientos y modelo de contenidos de la metodología UWE.
- **Diseño:** En esta fase se realiza el diseño del sistema web, todo referente al diseño y comportamiento de la interfaz y navegación en diferentes dispositivos, se hace uso del modelo de navegación y de presentación de la metodología UWE.
- **Desarrollo:** Fase donde se realiza el desarrollo del sistema, se hace uso del modelo de flujo de proceso de la metodología UWE.
- **Implementación:** En esta fase se implementa el sistema, ya la complicación de los módulos del sistema con cero errores.
- **Producción:** En esta columna es donde estará los módulos finalizados subidos al servidor ya para el funcionamiento en la empresa.

Para administrar los procesos en el tablero Kanban se utilizará la herramienta visual Trello que facilita el trabajo colaborativo. En la siguiente figura 3.2 vemos el tablero Kanban donde las columnas análisis, diseño, desarrollo e implementación se subdividen en dos columnas en curso y hecho; la columna en curso es para las tareas que se están desarrollando y la columna hecha es cuando ya está finalizada.

Figura 3.2
Tablero Kanban – UWE



Nota: Tablero kanban realizada en trello.com.

En la columna solicitud están todos los módulos a realizar para lograr el objetivo de terminar el desarrollo del sistema; estos módulos son: administración de usuarios que implica roles y permisos, empleados, contratos, permisos, Rfid, asistencias, generación de planillas, control de asistencias.

3.2.2 Roles de Kanban

La tabla 3.1 se muestra las fases de trabajo encontrados en el tablero de Kanban con UWE, así también los límites de WIP (Work in Progress) y los responsables del módulo asignado a la persona o equipo en kanban.

Tabla 3.1
Fases de trabajo, limite WIP y responsables

FASE	WIP	RESPONSABLE
Solicitud		
Análisis	3	Eddy Zeballos
Diseño	3	
Desarrollo	2	
Implementación	2	
Producción		

Nota: Fases de trabajo, limite WIP y responsables.

3.2.3 Fase de Análisis

3.2.3.1 Análisis de requerimientos

Lista de requerimientos del sistema

El área de recursos humanos requiere un sistema de información que tenga las siguientes características.

Tabla 3.2*Lista de requerimientos del sistema*

REF.	FUNCION	CATEGORIA (Oculto – Evidente)
R1	Autenticación de usuarios y contraseñas para ingresar al sistema	Evidente
R2	Registro de usuarios para el manejo del sistema, además de asignarle roles	Evidente
R3	Registrar nuevos empleados al sistema,	Evidente
R4	Asignar o revocar las tarjetas tag Rfid al empleado	Evidente
R5	Registrar las asistencias con las tarjetas Rfid y guardarlos en una base de datos	Oculto
R6	Registrar datos sobre el contrato como ser sueldo, fecha de inicio y fin de contrato, cargo; estos deben ser almacenado una base de datos	Oculto
R7	Los empleados podrán ver sus asistencias	Evidente
R8	En caso de faltas y atrasos hacer los respectivos descuentos y ser visualizados en su boleta de pago	Evidente
R9	control de accesos a las áreas mediante el tag Rfid	Evidente
R10	Generar planillas de sueldos	Evidente
R11	Generar boleta de pago	Evidente

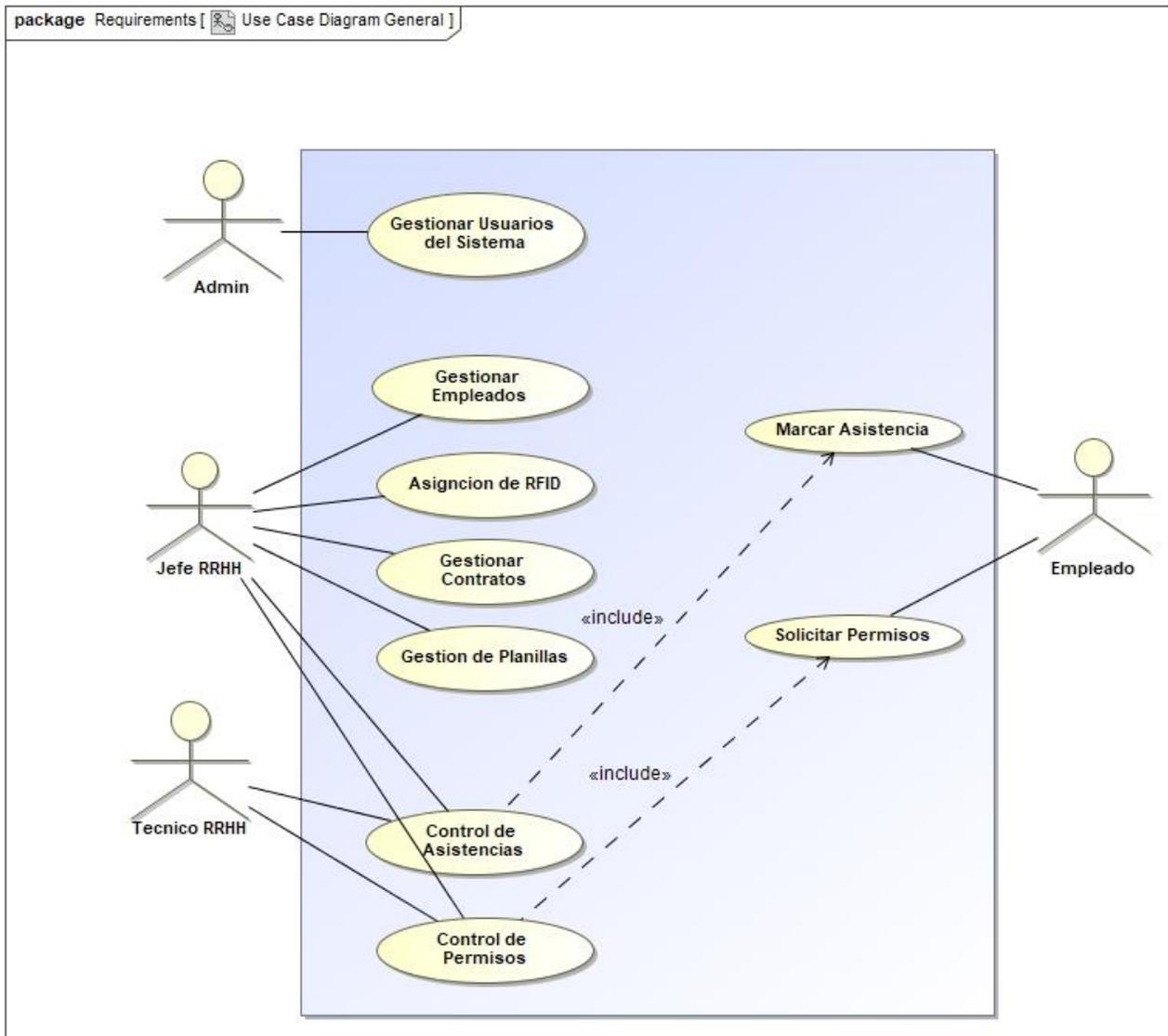
Nota: Lista de requerimientos del sistema.

3.2.3.2 Caso de uso

Diagrama de caso de uso general

El modelo de caso de uso nos muestra el comportamiento del sistema frente a los actores, en este diagrama mostraremos un proceso general de todos los actores que intervienen en el sistema.

Figura 3.3
Diagrama de caso de uso general



Nota: Diagrama de caso de uso general.

Tabla 3.3
Descripción de caso de Uso General

ACTOR	EVENTO	RESPUESTA
Admin	Gestionar Usuarios	El Admin podrá crear usuarios, roles y permisos; estos usuarios son para el manejo del sistema.
RR.HH.	Empleados	El jefe de RR.HH. podrá gestionar la creación o modificación de usuarios.
	Rfid	El jefe de RR.HH. puede asignar o revocar las tarjetas

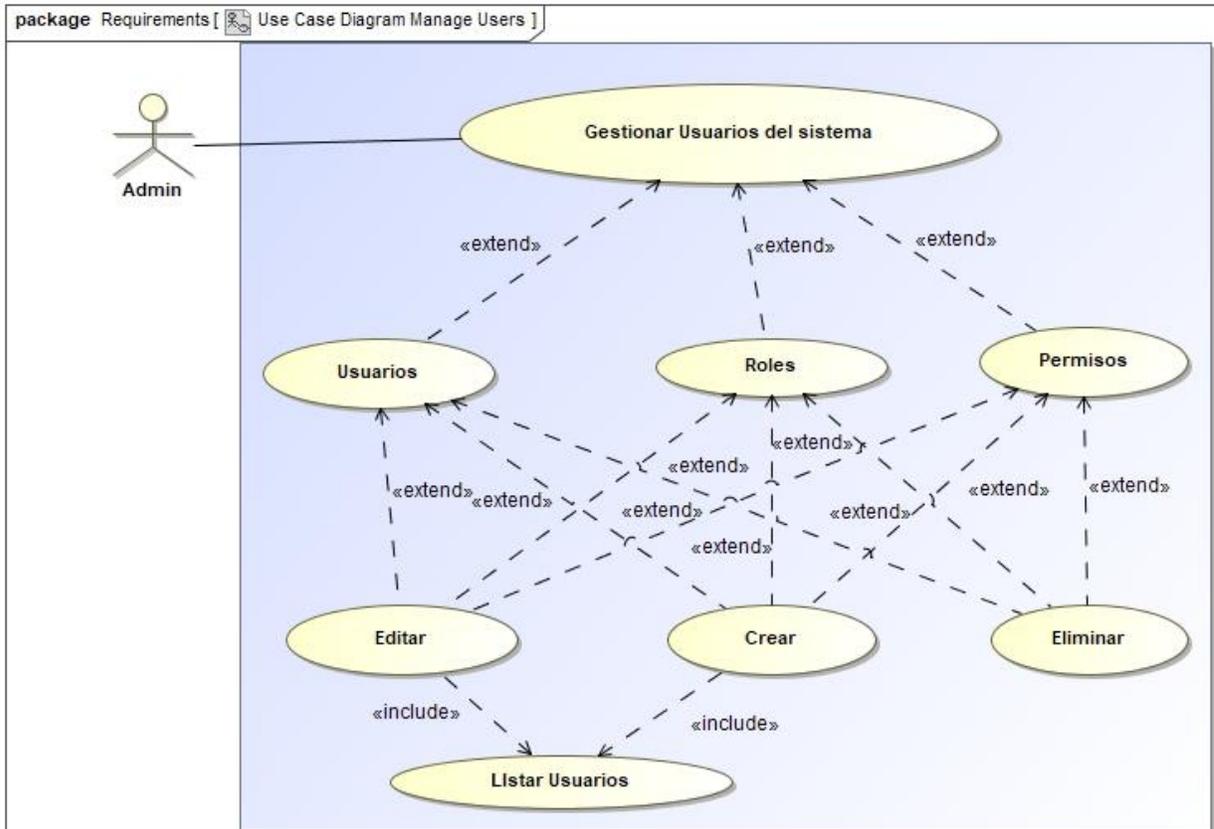
		tag Rfid a los empleados.
	Contratos	El jefe de RR.HH. podrá gestionar la creación o modificación de contratos.
	Planillas	El jefe de RR.HH. podrá gestionar la generación de planillas.
	Asistencias	El jefe de RR.HH. podrá gestionar las asistencias de los empleados.
	Permisos	El jefe de RR.HH. podrá gestionar los permisos de los empleados.
Técnico RR.HH.	Asistencias	El técnico de RR.HH. podrá ingresar al módulo según su rol.
	Permisos	El técnico de RR.HH. podrá ingresar al módulo según su rol.
Empleado	Asistencias	El empleado podrá ver sus estados de asistencia.
	Permisos	El empleado podrá solicitar y ver el estado de sus permisos

Nota: Descripción de caso de Uso General.

Diagrama de caso de uso Gestión de Usuarios

Figura 3.4

Diagrama de caso de uso Gestión Usuarios



Nota: Diagrama de caso de uso Gestión Usuarios.

Tabla 3.4

Especificación del caso de uso Gestión de Usuarios

Caso de uso:	Gestión Usuarios
Actores:	Admin (Encargado del Sistema)
Descripción:	Permite crear, listar, editar, eliminar para los usuarios, roles y permisos
Precondiciones:	El actor debe ser un usuario registrado a su vez tener el rol Admin. El rol Admin debe estar registrado. Los permisos deben estar registrados.
Flujo Normal:	<ul style="list-style-type: none"> El actor ingresa a la opción Gestión de Usuarios, donde

muestra el listado de usuarios con las opciones de crear, editar, eliminar.

- El actor al crear o editar usuario podrá asignar o quitar un rol.
- El actor ingresa a la opción de Roles y Permisos, donde muestra un listado con las opciones crear, editar, eliminar.
- El actor al crear o editar role podrá asignar o quitar un permiso.
- Los permisos pueden ser creados o editados

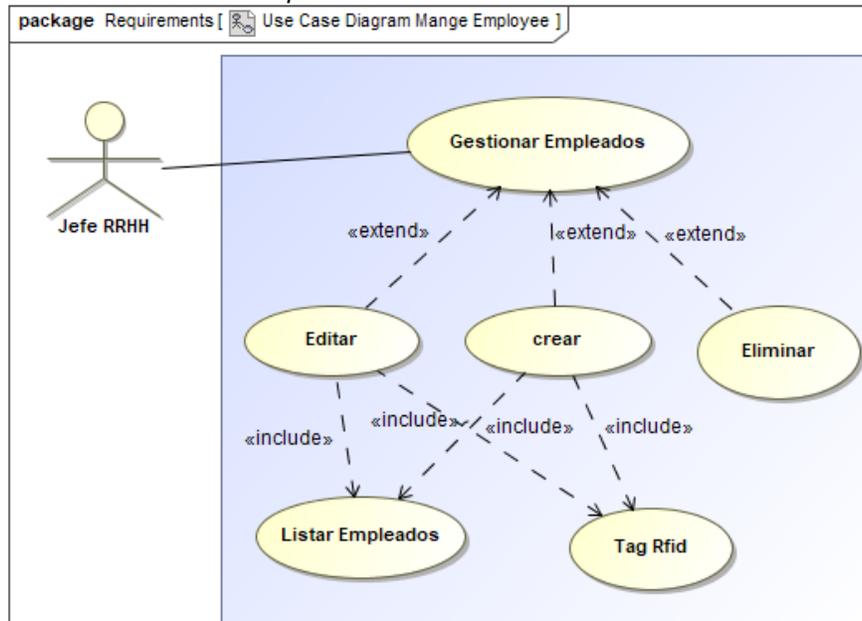
Postcondiciones: El sistema ejecuta las opciones elegidas por el actor y muestra los cambios realizados.

Nota: Especificación del caso de uso Gestión de Usuarios.

Diagrama de caso de uso Gestión de Empleados

Figura 3.5

Diagrama de caso de uso Gestión Empleados



Nota: caso de uso Gestión Empleados.

Tabla 3.5

Especificación del caso de uso Gestión Empleados

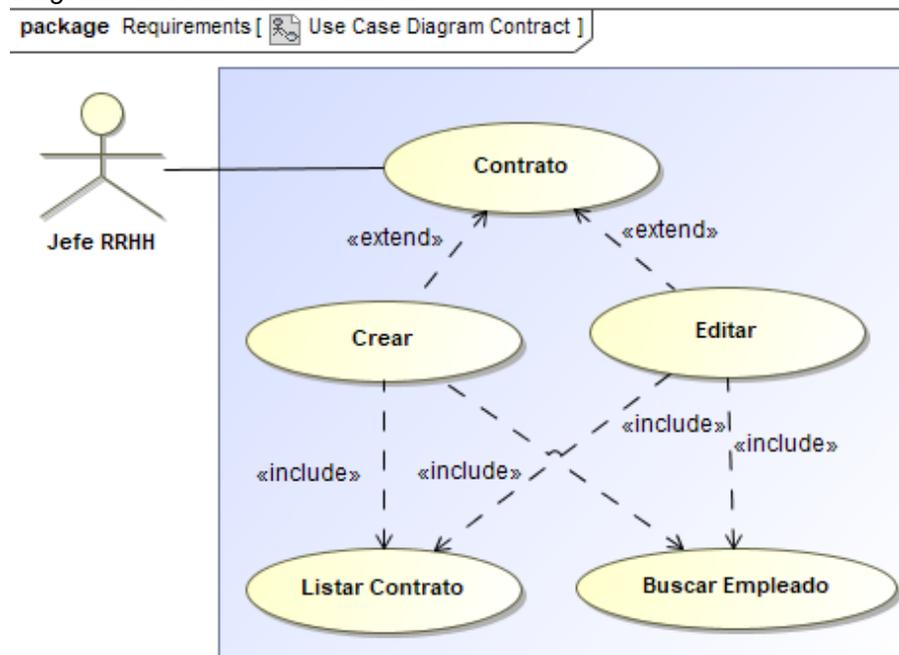
Caso de uso:	Gestión Empleados
Actores:	Jefe de RR.HH.
Descripción:	Permite crear, listar, editar, eliminar para los Empleados, roles y permisos
Precondiciones:	El actor debe ser un usuario registrado a su vez tener un rol con los permisos de crear, editar.
Flujo Normal:	<ul style="list-style-type: none">• El actor ingresa a la opción Gestión de Empleados, donde muestra el listado de empleados con las opciones de crear, editar, eliminar.• El actor al crear o editar empleado podrá asignar, actualizar o quitar la tarjeta tag Rfid.
Postcondiciones:	El sistema ejecuta las opciones elegidas por el actor y muestra los cambios realizados.

Nota: Especificación del caso de uso Gestión Empleados.

Diagrama de caso de uso Contrato

Figura 3.6

Diagrama de caso de uso Contrato



Nota: Diagrama de caso de uso Contrato.

Tabla 3.6

Especificación del caso de uso Contrato

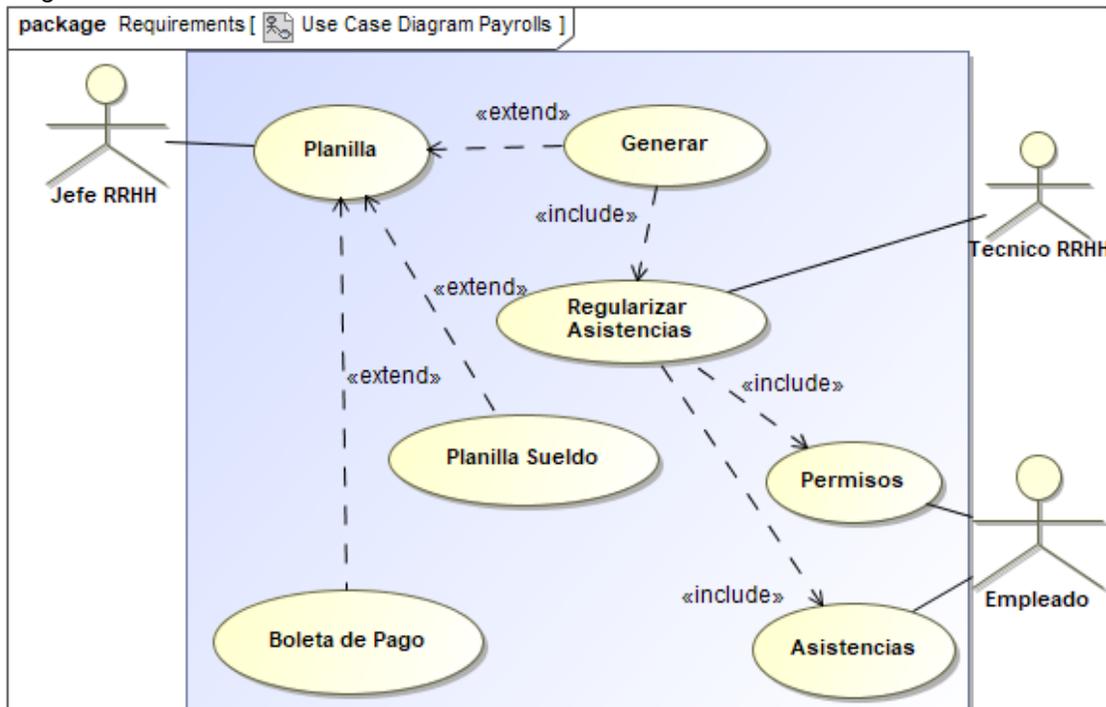
Caso de uso:	Contrato
Actores:	Jefe de RR.HH.
Descripción:	Permite crear, listar, editar contrato
Precondiciones:	El actor debe ser un usuario registrado a su vez tener un rol con los permisos de crear, editar.
Flujo Normal:	<ul style="list-style-type: none">• El actor ingresa a la opción Contratos, donde muestra el listado de contratos de los empleados con las opciones de crear, editar.• El actor al crear contrato el sistema le mostrará todos los empleados sin contratos y deberá escoger alguno.
Postcondiciones:	El sistema ejecuta las opciones elegidas por el actor y muestra los cambios realizados.

Nota: Especificación del caso de uso Contrato.

Diagrama de caso de uso Gestión Planillas

Figura 3.7

Diagrama de caso de uso Gestión Planillas



Nota: Diagrama de caso de uso Gestión Planillas.

Tabla 3.7

Especificación del caso de uso Gestión Planillas

Caso de uso:	Gestión Planillas
---------------------	--------------------------

Actores:	Jefe de RR.HH.
-----------------	----------------

Descripción:	Permite generar, editar las planillas
---------------------	---------------------------------------

Precondiciones:	Los actores deben ser usuarios registrados a su vez tener un rol con los permisos de crear, editar. Para la generación de las planillas debe existir Asistencias por parte de los empleados.
------------------------	---

Flujo Normal:	<ul style="list-style-type: none">• El actor ingresa a la opción Planillas, donde muestra el listado de planillas.• El actor al generar la planilla antes debe haber regularizado las asistencias y permisos.• Una vez generado la planilla podrá generar las boletas de pago.
----------------------	--

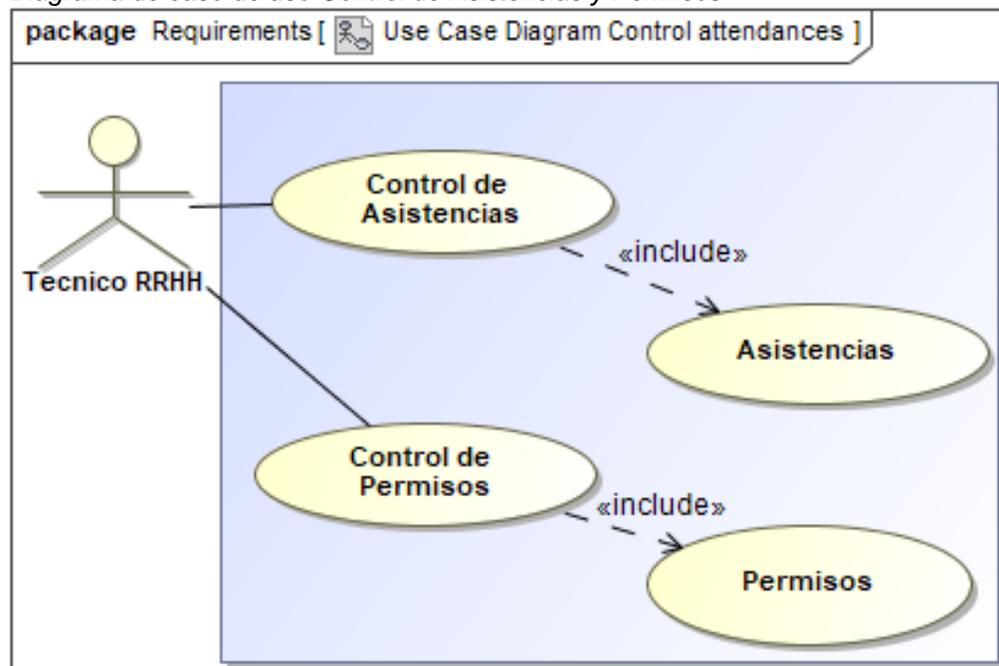
Postcondiciones:	El sistema ejecuta las opciones elegidas por el actor y muestra los cambios realizados.
-------------------------	---

Nota: Especificación del caso de uso Gestión Planillas.

Diagrama de caso de uso Control de Asistencias y Permisos

Figura 3.8

Diagrama de caso de uso Control de Asistencias y Permisos



Nota: Diagrama de caso de uso Control de Asistencias y Permisos.

Tabla 3.8

Especificación del caso de uso Control de Asistencias

Caso de uso:	Control de asistencias
Actores:	Técnico de RR.HH.
Descripción:	El actor regulariza las asistencias y permisos para la generación de planillas
Precondiciones:	Los actores deben ser usuarios registrados a su vez tener un rol con los permisos de crear, editar.
Flujo Normal:	<ul style="list-style-type: none"> • El actor ingresa a la opción Regularizar Asistencias. • El Actor seleccionar el mes y año para buscar las todas las asistencias de los empleados. • Una vez generado los cálculos internos de horas trabajadas de acuerdo a las asistencias. El actor deberá seleccionar un empleado.

- Una vez seleccionado el empleado se visualizará todas las asistencias de ese empleado en un calendario.
- El actor deberá verificar si existe faltas, que se justifican por motivos fortuitos, estos casos el actor contrastara con la debida justificación y modificara las horas trabajadas.

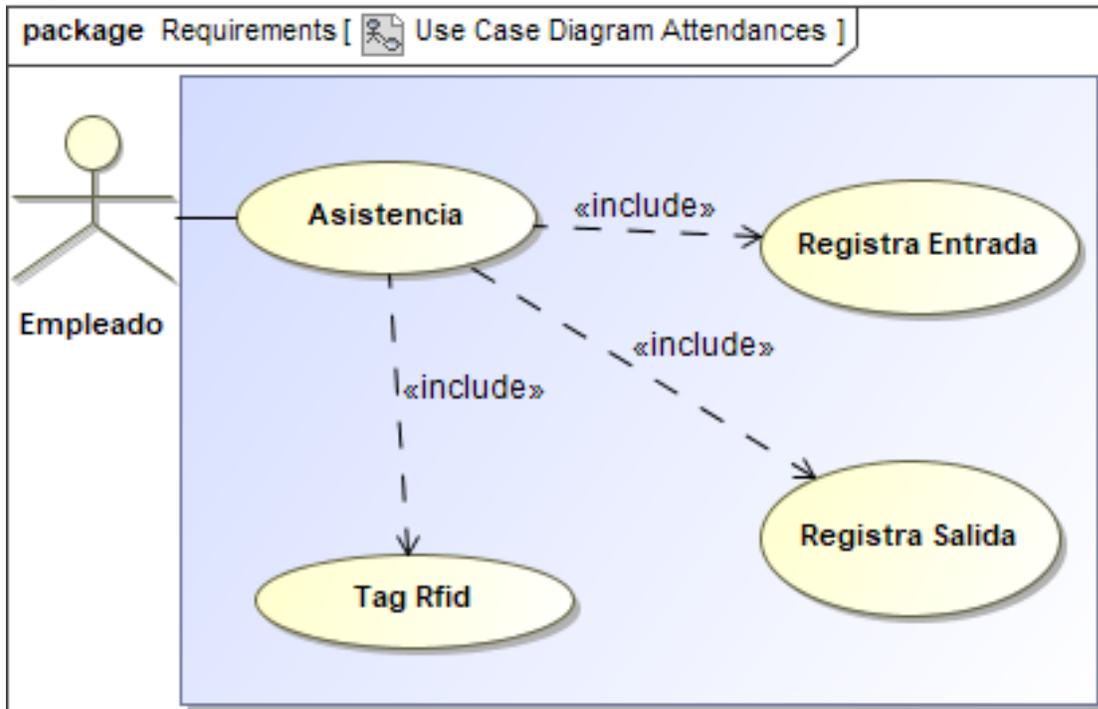
Postcondiciones: El sistema ejecuta las opciones elegidas por el actor y muestra los cambios realizados.

Nota: Especificación del caso de uso Control de Asistencias.

Diagrama de caso de uso Asistencia

Figura 3.9

Diagrama de caso de uso Control de Asistencias



Nota: Diagrama de caso de uso Control de Asistencias.

Tabla 3.9

Especificación del caso de uso Asistencia

Caso de uso:	Asistencia
Actores:	Empleado
Descripción:	El actor realiza su asistencia
Precondiciones:	Los actores deben ser usuarios registrados a su vez tener el rol de empleado, además de tener asignado su tarjeta tag Rfid.
Flujo Normal:	<ul style="list-style-type: none">• El actor al momento de marcar la asistencia de tener su tarjeta Rfid.• Para Registro de entrada deberá marcar antes de las 11:00 a.m.• Para la salida tarde deberá marcar después de las 18:00 p.m.
Postcondiciones:	El sistema ejecuta las opciones elegidas por el actor y muestra los cambios realizados.

Nota: Especificación del caso de uso Asistencia.

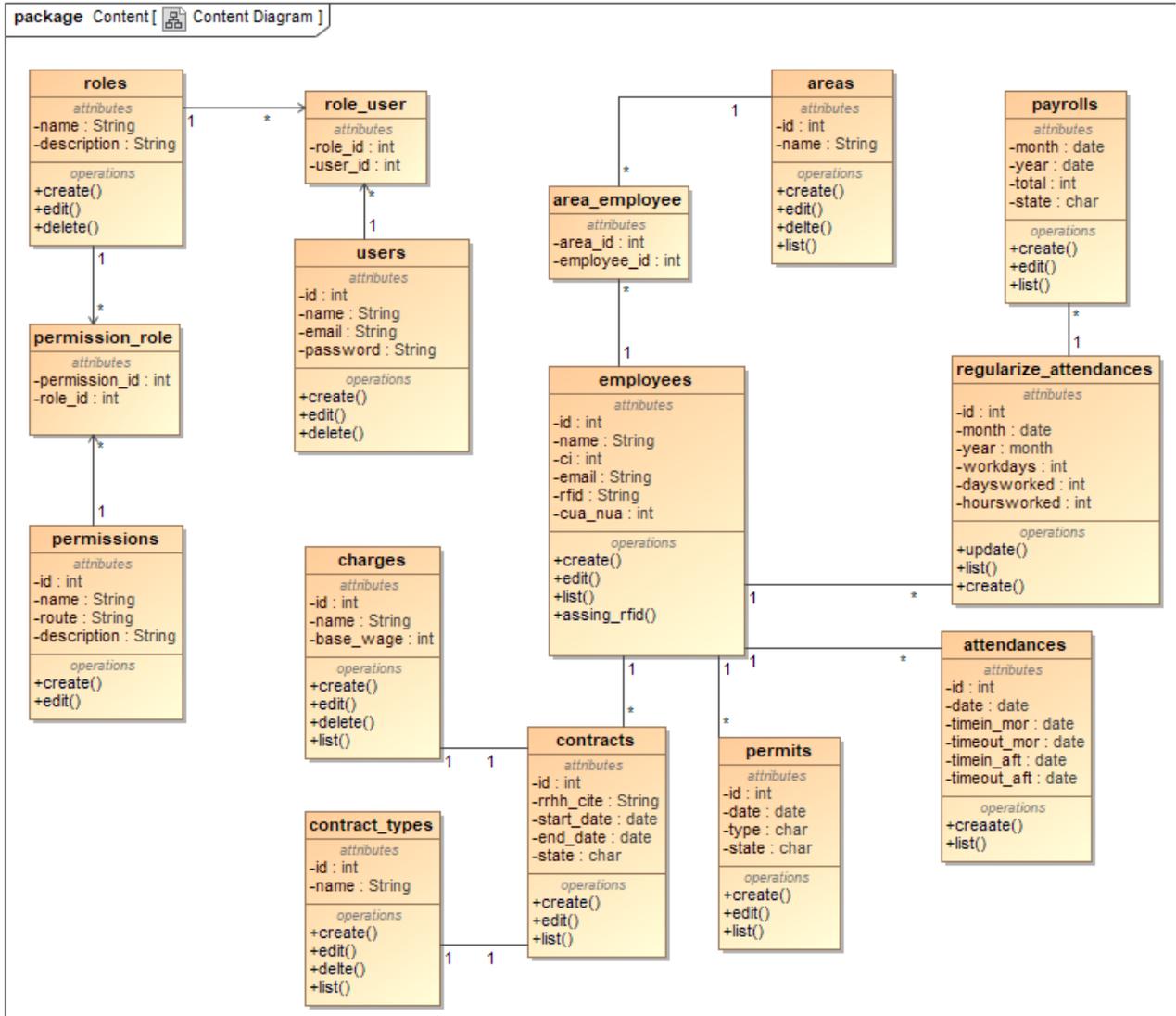
3.2.3.3 Diagrama de contenido

En el sistema el diagrama de contenido mostrara las relaciones entre las entidades y la estructura de loa datos que se encuentran alojados en el sistema, las clases identificadas describirán como se encuentras relacionadas.

En este modelo de diagrama de contenido se muestra los siguientes módulos que se muestran en la figura 3.10.

- Modulo Gestión de Usuarios
- Modulo Gestión de Empleados
- Modulo Contratos
- Modulo Planillas
- Modulo Control (Regularización) de asistencias y permisos
- Modulo Asistencias con Rfid

Figura 3.10
Diagrama de Contenido del sistema

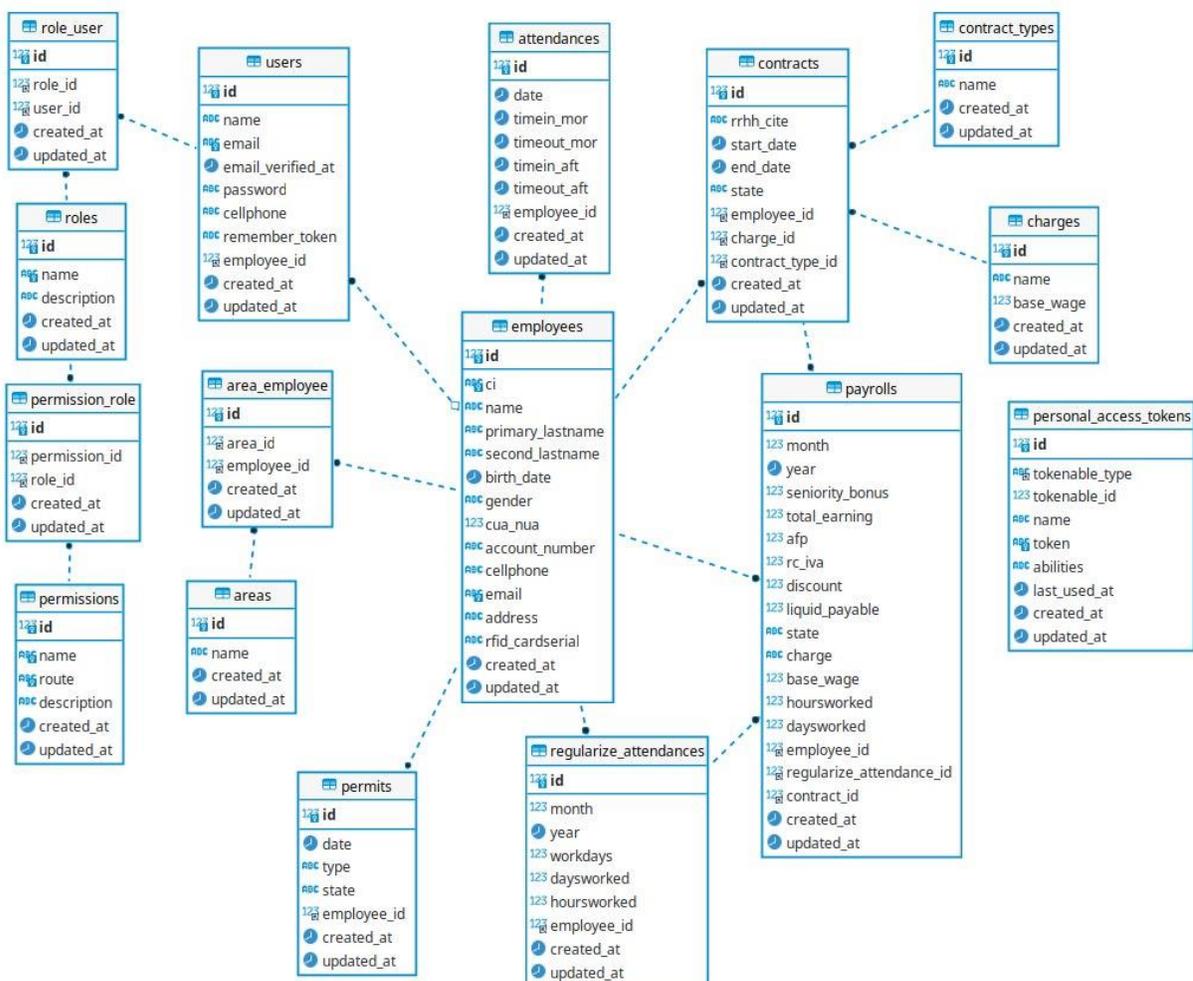


Nota: Diagrama de Contenido del sistema.

3.2.3.4 Diseño de la base de datos

Se diseño la base de datos analizando todos los datos de los requerimientos para su cumplimiento y el adecuado funcionamiento del sistema.

Figura 3.11
Diseño de la base de datos



Nota: Diseño de la base de datos amlc.

3.2.4 Fase de diseño

3.2.4.1 Modelo de Navegación

En este diagrama se representará los siguientes:

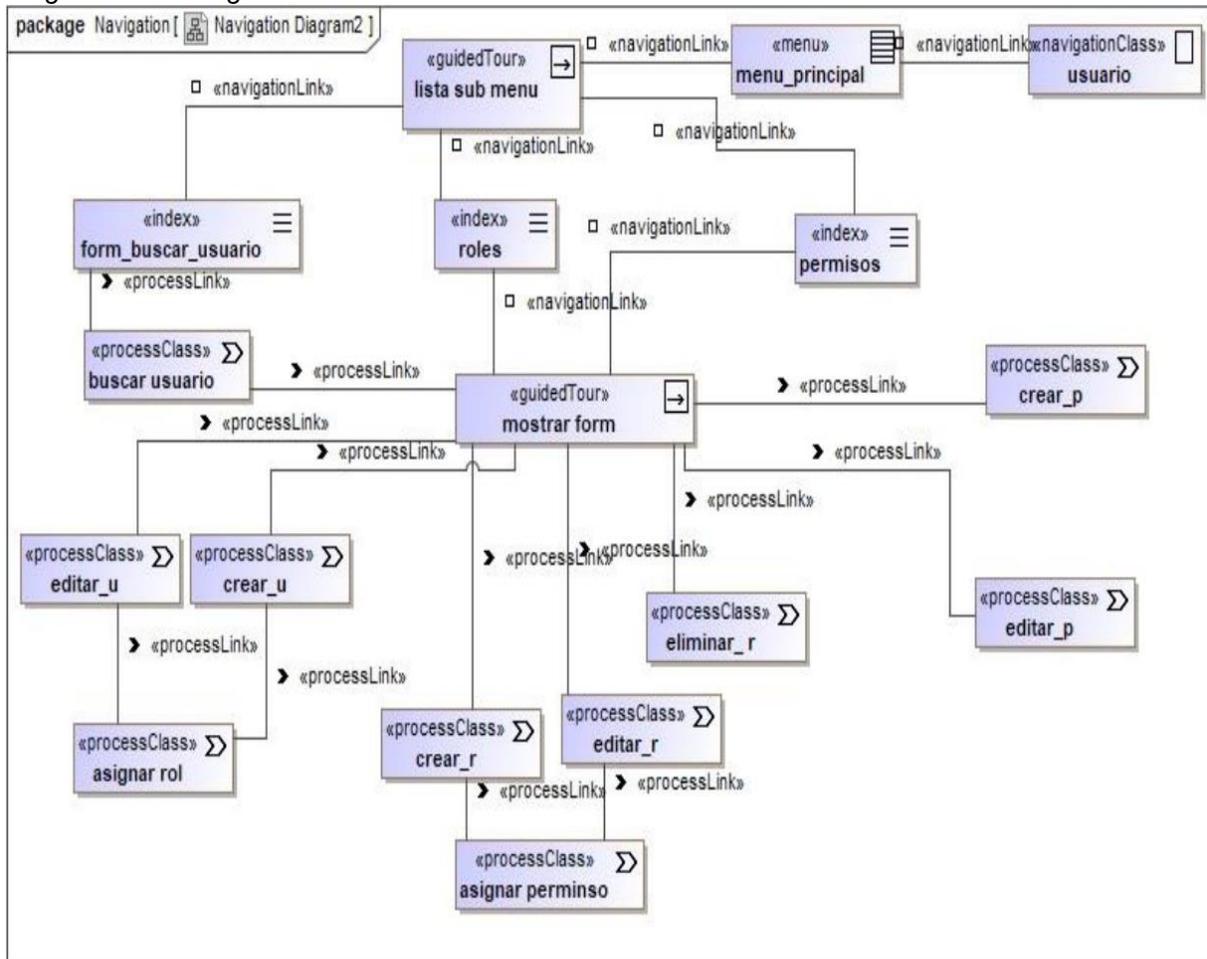
- Modulo Gestión de Usuarios
- Modulo Gestión de Empleados
- Modulo Contrato

- Modulo Asistencias con Rfid
- Modulo Planillas
- Modulo Control de asistencias y permisos
- Modulo Control de acceso con Rfid

Diagrama de navegación Gestión de Usuarios

Figura 3.12

Diagrama de Navegación Gestión de Usuarios

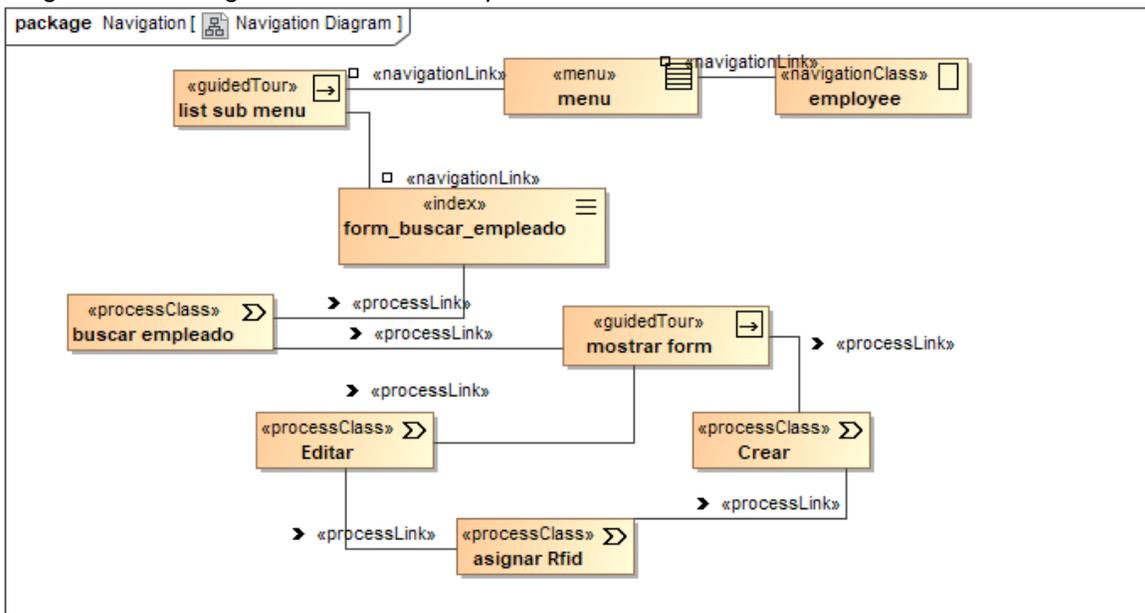


Nota: Diagrama de Navegación Gestión de Usuarios.

Diagrama de navegación Gestión de Empleados

Figura 3.13

Diagrama de Navegación Gestión de Empleados

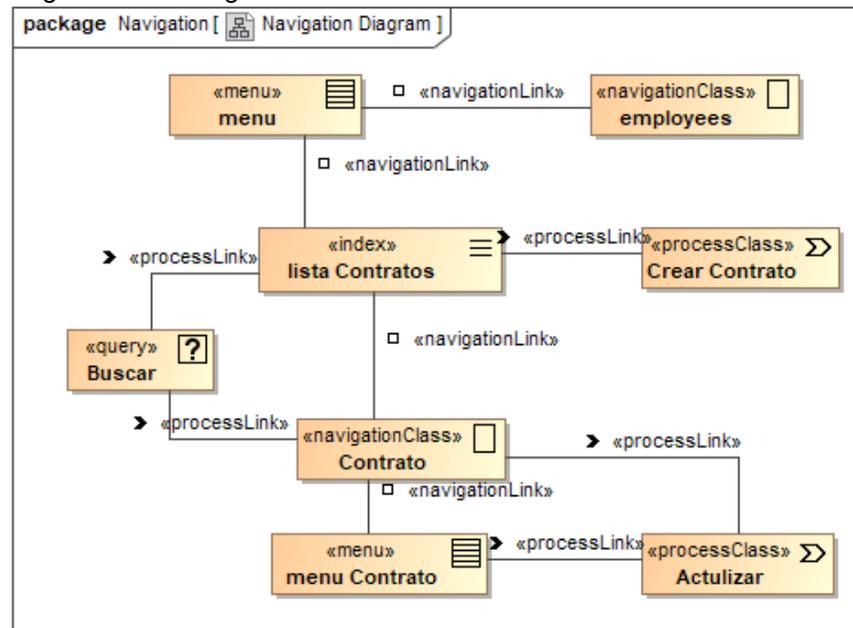


Nota: Diagrama de Navegación Gestión de Empleados.

Diagrama de navegación Contrato

Figura 3.14

Diagrama de Navegación Contrato

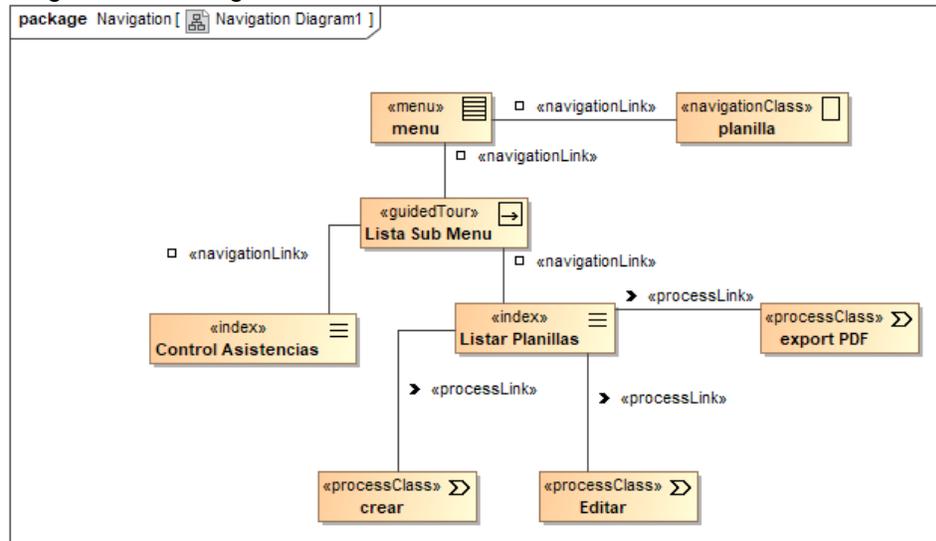


Nota: Diagrama de Navegación Contrato.

Diagrama de navegación Planillas

Figura 3.15

Diagrama de Navegación Planillas

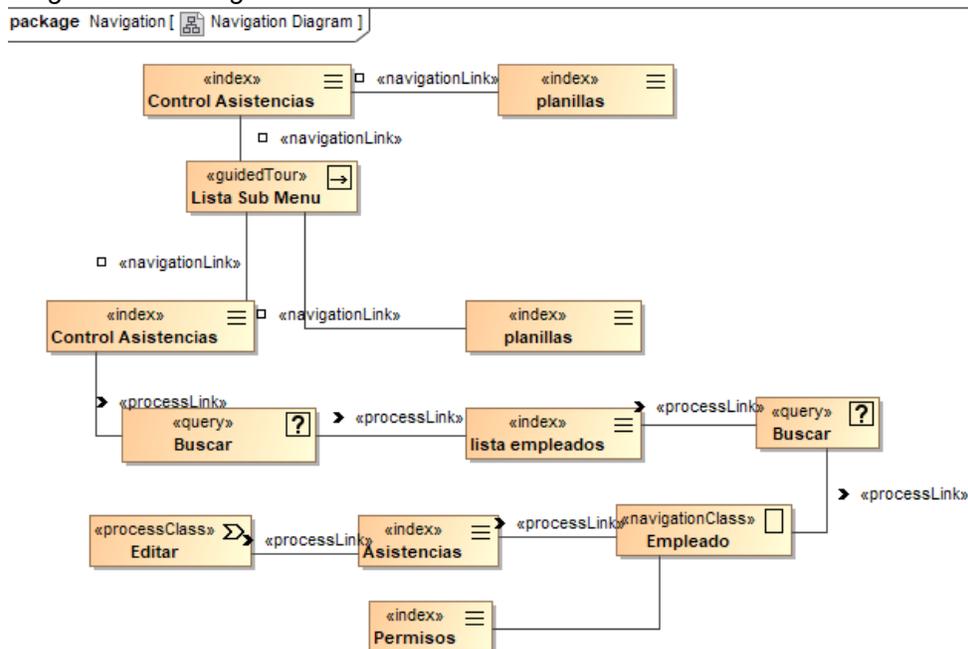


Nota: Diagrama de Navegación Planillas.

Diagrama de navegación Control de Asistencia

Figura 3.16

Diagrama de Navegación Control de Asistencia



Nota: Diagrama de Navegación Control de Asistencia.

3.2.4.2 Modelo de Presentación

En este modelo se muestra el diagrama de presentación para los módulos siguientes:

- Modulo Gestión de Gestión Usuarios
- Modulo Gestión de Gestión Empleados
- Modulo Contrato
- Modulo Asistencias con Rfid
- Modulo Planillas
- Modulo Control de asistencias y permisos

Diagrama de presentación General

En este diagrama de presentación representaremos los accesos que tiene cada usuario al sistema como ser:

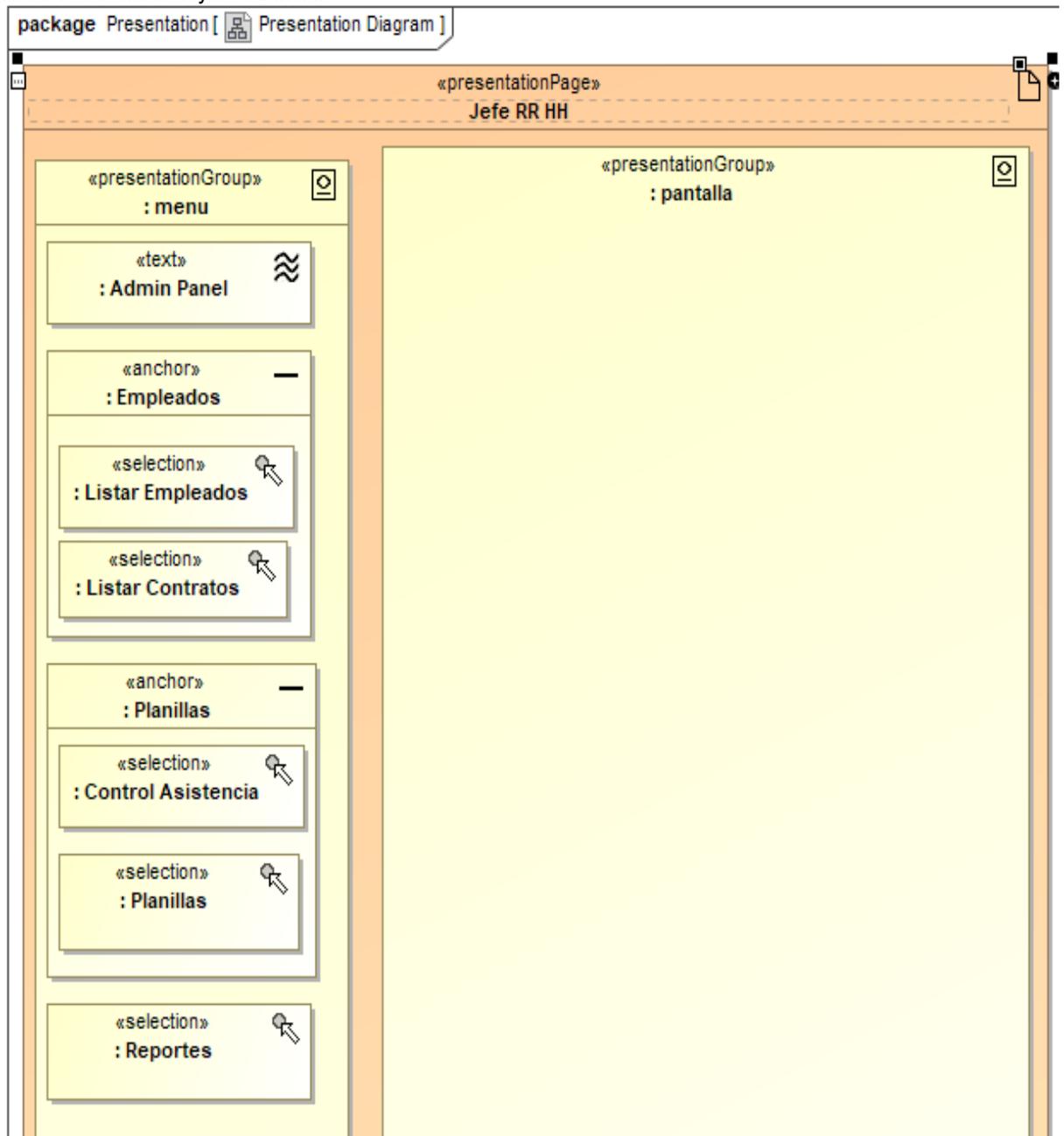
- Admin
- Jefe RRHH
- Técnico RRHH
- Empleado

Figura 3.17
Diagrama de Presentación Usuario Admin



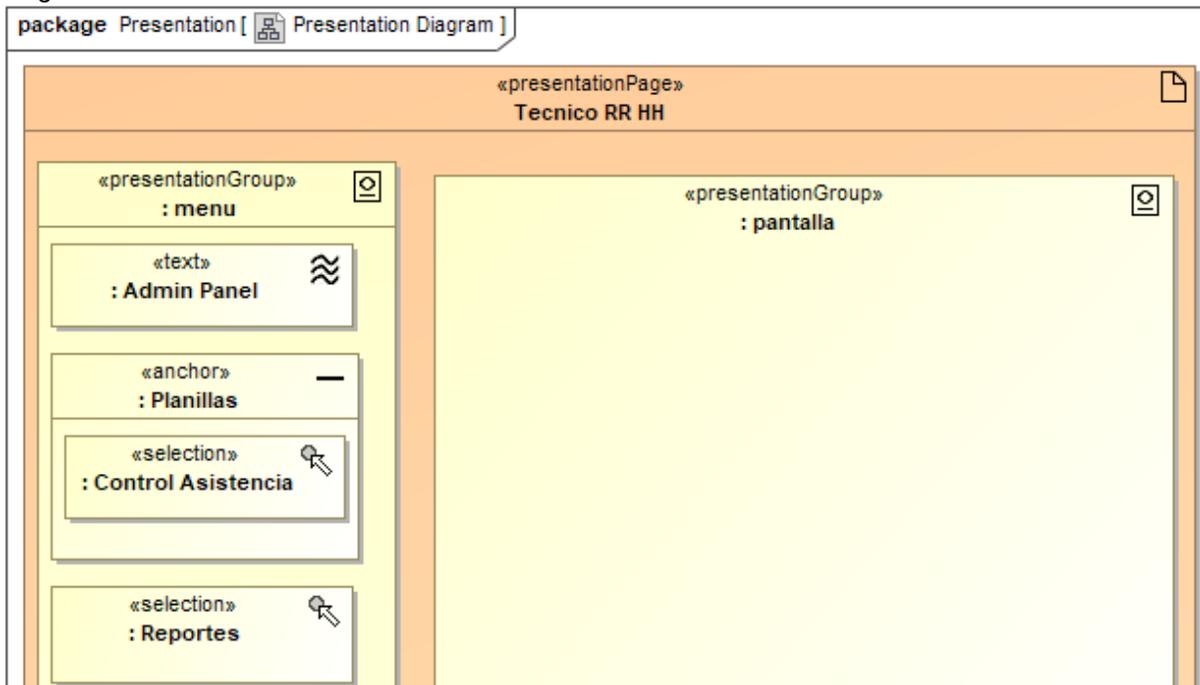
Nota: Diagrama de Presentación Usuario Admin.

Figura 3.18
Diagrama de Presentación jefe RR.HH.



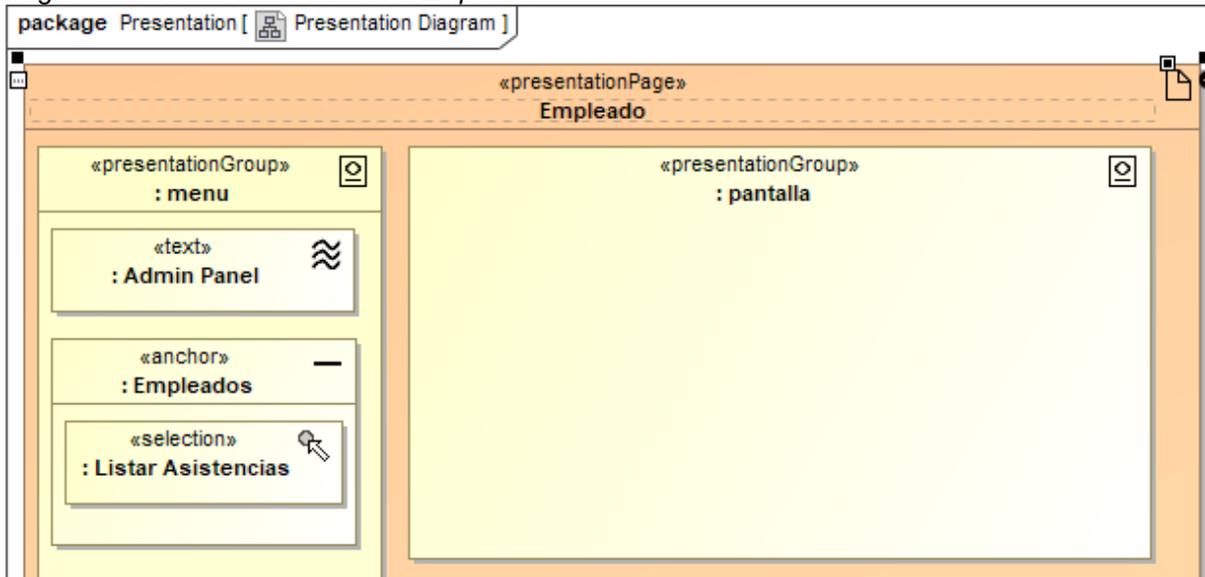
Nota: Diagrama de Presentación jefe RR.HH.

Figura 3.19
Diagrama de Presentación Usuario Técnico RR.HH.



Nota: Diagrama de Presentación Usuario Técnico RR.HH.

Figura 3.20
Diagrama de Presentación Usuario Empleado

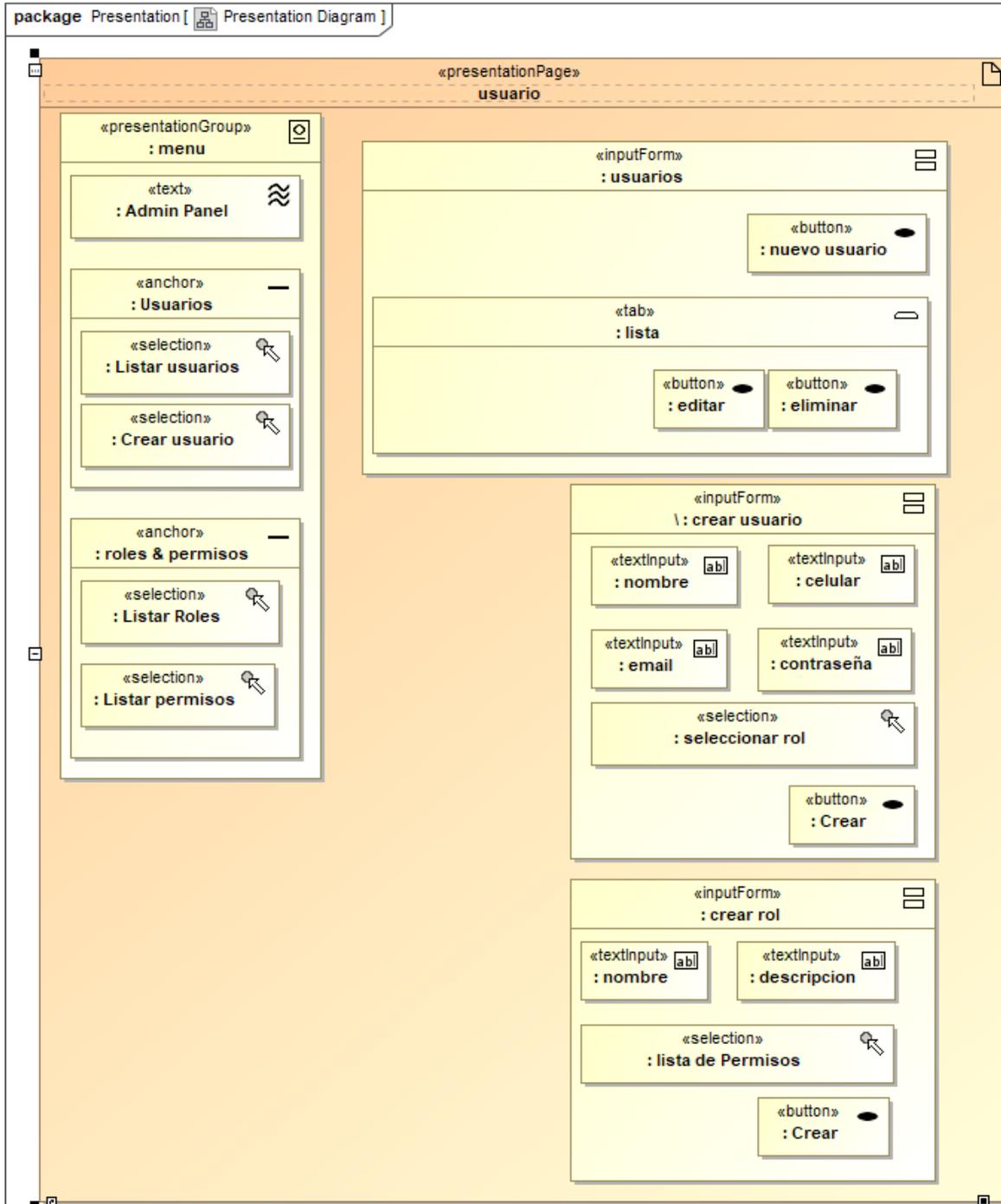


Nota: Diagrama de Presentación Usuario Empleado.

Diagrama de presentación Gestión Usuarios

Figura 3.21

Diagrama de Presentación Gestión Usuarios

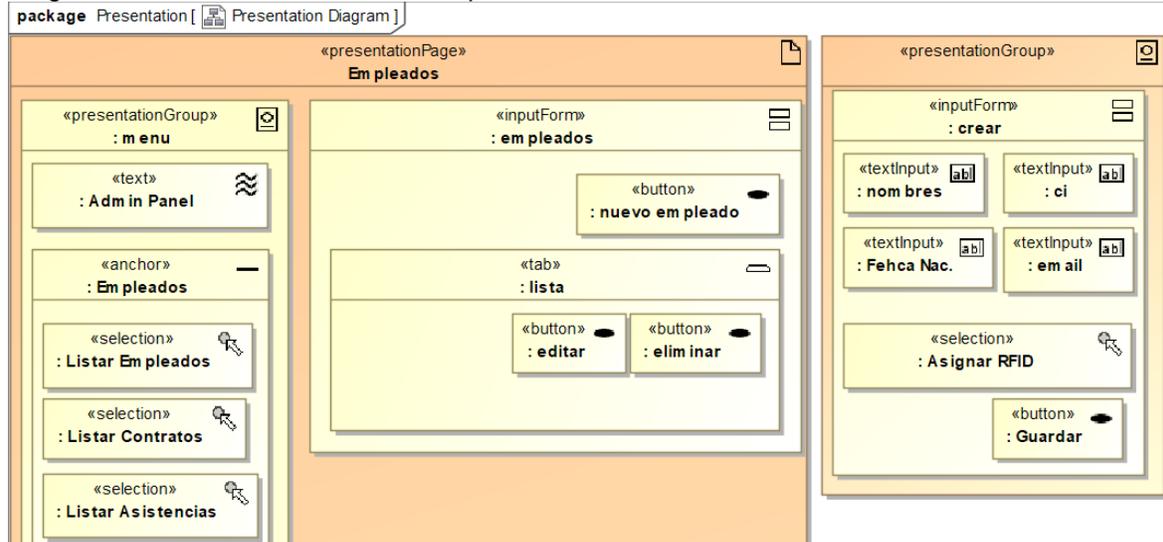


Nota: Diagrama de Presentación Gestión Usuarios.

Diagrama de presentación Gestión Empleados

Figura 3.22

Diagrama de Presentación Gestión Empleados

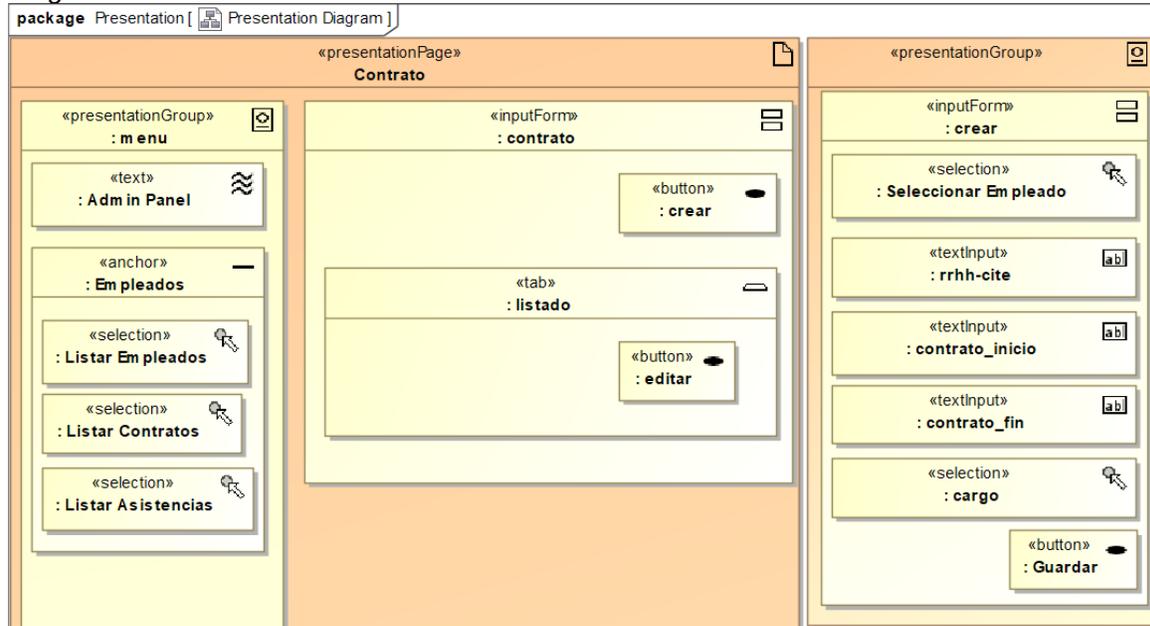


Nota: Diagrama de Presentación Gestión Empleados.

Diagrama de presentación Contrato

Figura 3.23

Diagrama de Presentación Contrato

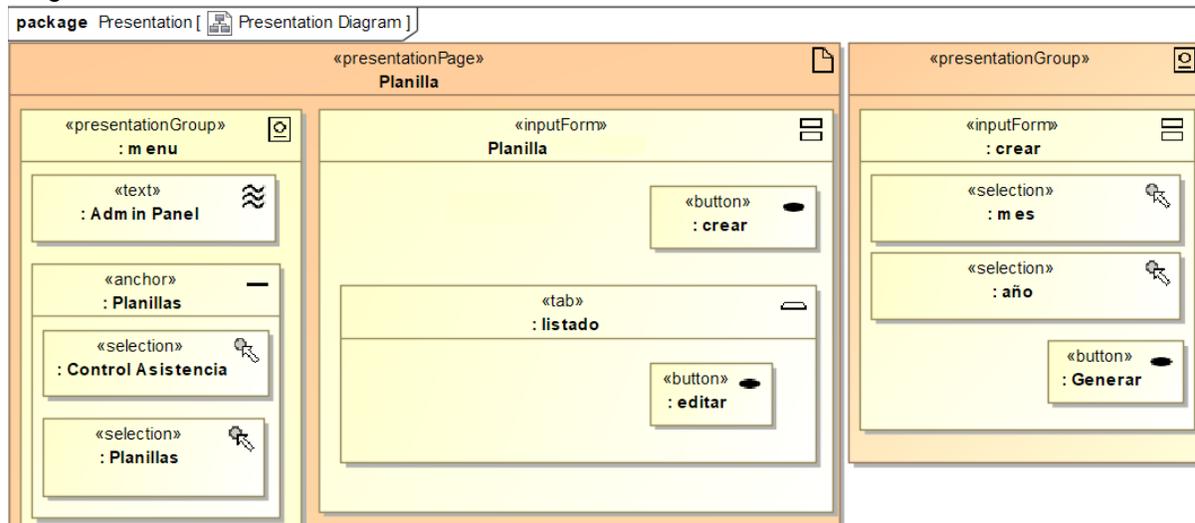


Nota: Diagrama de Presentación Contrato.

Diagrama de presentación Planilla

Figura 3.24

Diagrama de Presentación Planilla

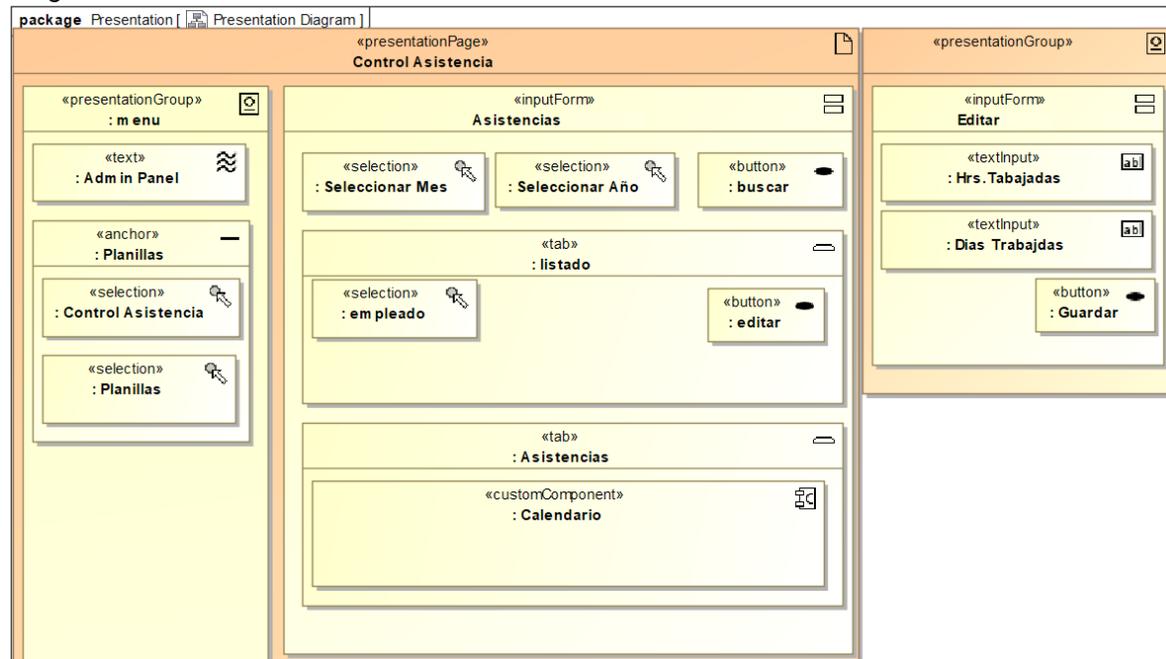


Nota: Diagrama de Presentación Planilla.

Diagrama de presentación Control de Asistencias

Figura 3.25

Diagrama de Presentación Control de Asistencias



Nota: Diagrama de Presentación Control de Asistencias.

3.2.5 Fase de desarrollo

3.2.5.1 Modelo de Flujo de Proceso

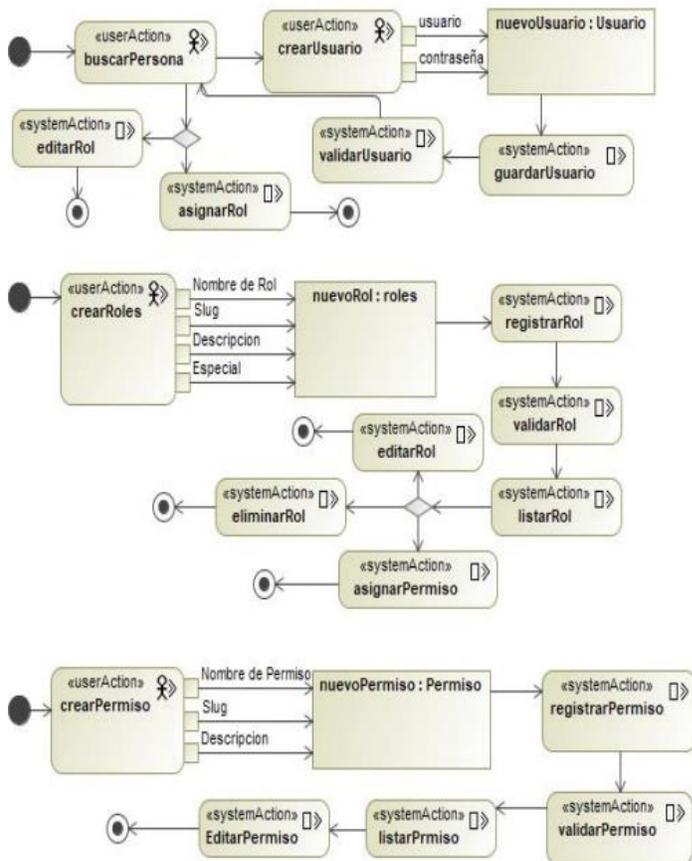
En este modelo se muestra el diagrama de flujo procesos para los módulos siguientes:

- Modulo Gestión de Gestión Usuarios
- Modulo Gestión de Gestión Empleados
- Modulo Contrato
- Modulo Asistencias con Rfid
- Modulo Planillas
- Modulo Control de asistencias y permisos

Diagrama de flujo de procesos de gestión de usuarios

Figura 3.26

Diagrama de Flujo de Procesos Gestión de Usuarios

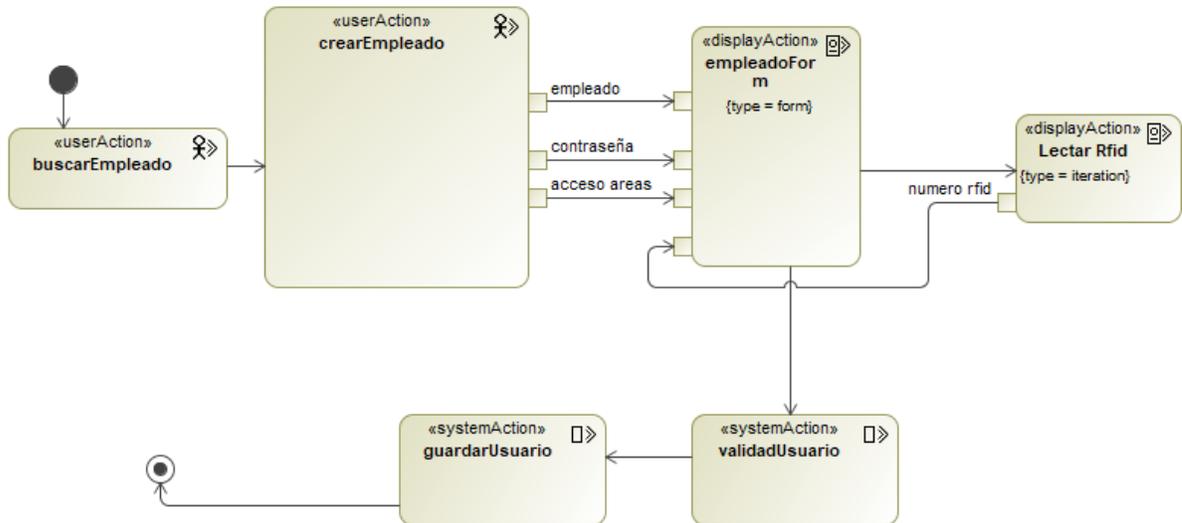


Nota: Diagrama de Flujo de Procesos Gestión de Usuarios.

Diagrama de flujo de procesos de gestión de empleados

Figura 3.27

Diagrama de Flujo de Procesos Gestión de Empleados

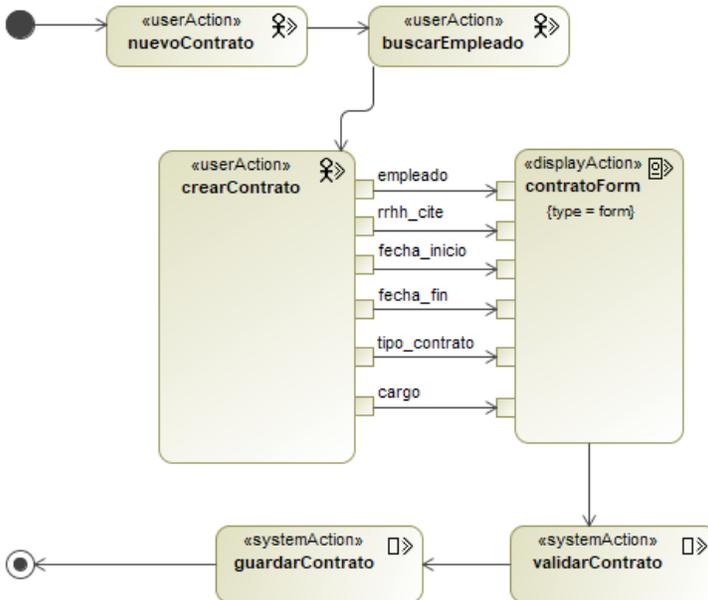


Nota: Diagrama de Flujo de Procesos Gestión de Empleados.

Diagrama de flujo de procesos contrato

Figura 3.28

Diagrama de Flujo de Procesos Contrato

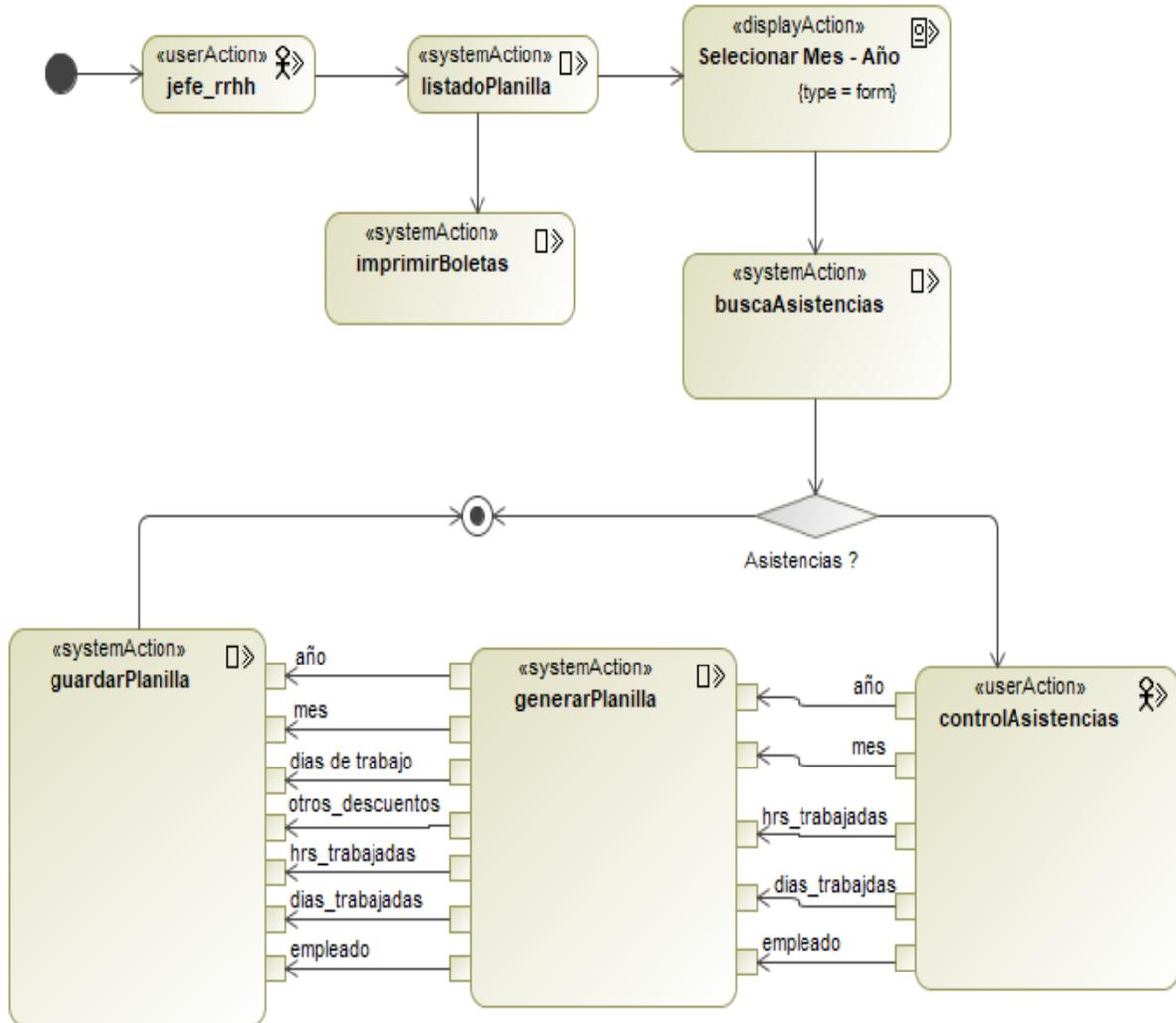


Nota: Diagrama de Flujo de Procesos Contrato.

Diagrama de flujo de procesos planilla

Figura 3.29

Diagrama de Flujo de Proceso de Planilla

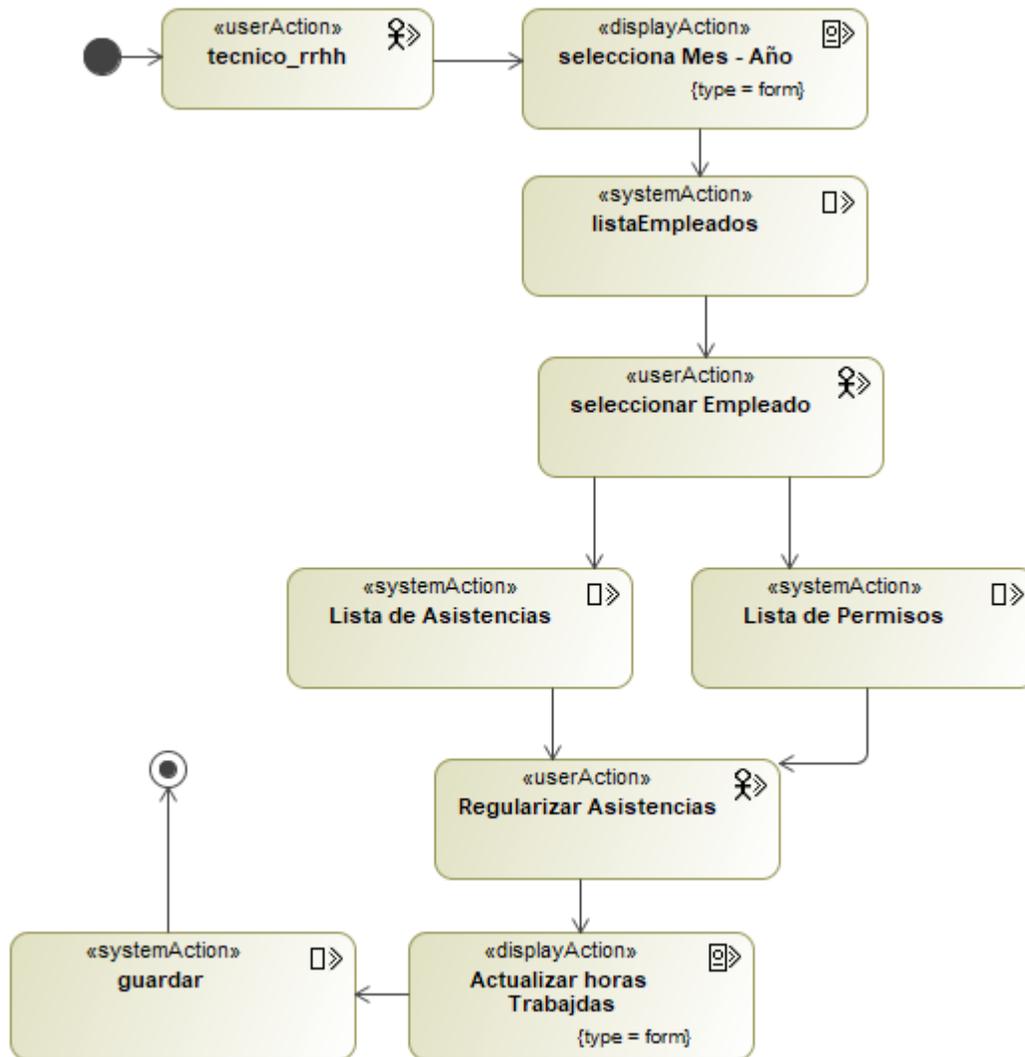


Nota: Diagrama de Flujo de Proceso de Planilla.

Diagrama de flujo de procesos control de asistencia

Figura 3.30

Diagrama de Flujo de Proceso de Control de Asistencias



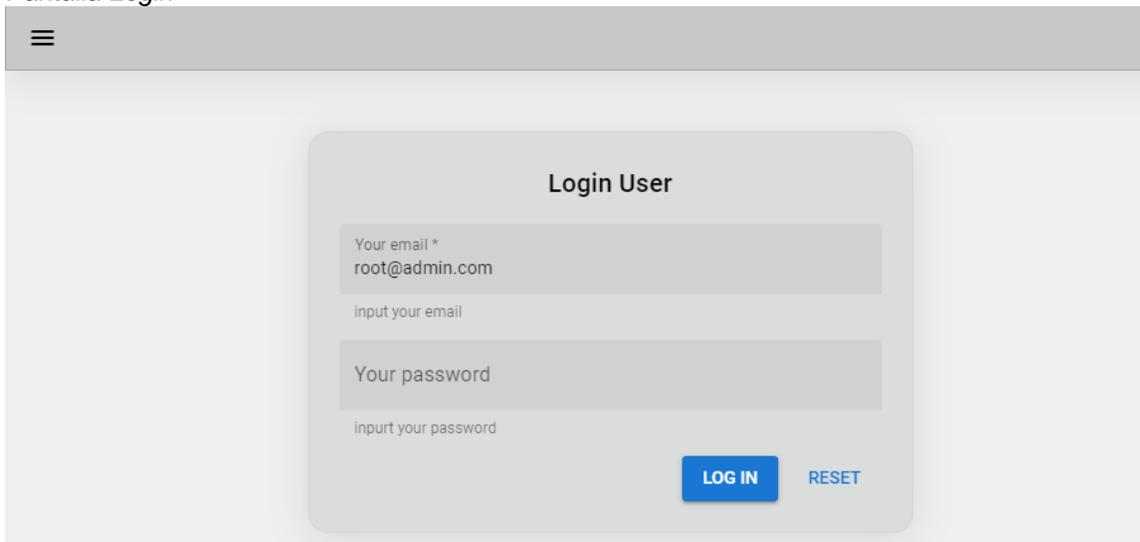
Nota: Diagrama de Flujo de Proceso de Control de Asistencias.

3.2.6 Fase de implementación

En este modelo se muestra la implementación en el sistema mostrando el desarrollo de las interfaces de los diferentes módulos y sus elementos contruidos de acuerdo al seguimiento de la metodología UWE.

Módulo Login

Figura 3.31
Pantalla Login



Nota: Pantalla Autenticación de usuarios.

Figura 3.32

Código fuente autenticación

```
public function login(Request $request) {
    $fields = $request->validate([
        'email' => 'required|string',
        'password' => 'required|string'
    ]);
    $user = User::where('email', $fields['email'])->first();

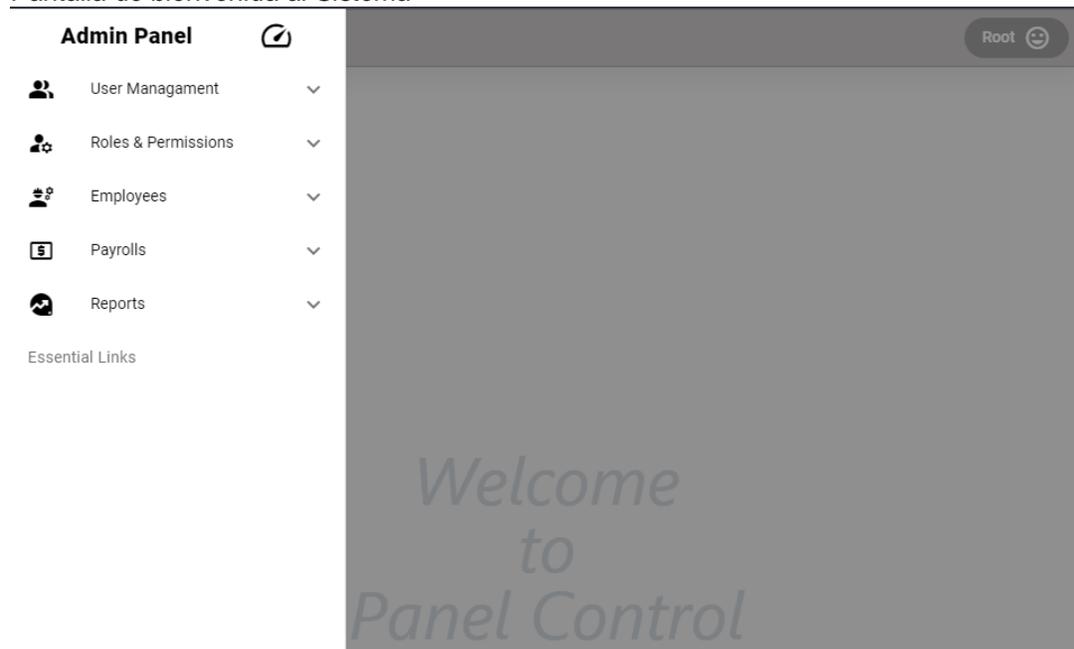
    if(!$user || !Hash::check($fields['password'], $user->password)) {
        return response([
            'message' => 'Bad creds'
        ], 401);
    }

    $scope = array('role_index', 'role_delete', );
    $token = $user->createToken('myapptoken',$scope)->plainTextToken;

    $response = [
        'user' => $user,
        'token' => $token,
    ];
    return response($response, 201);
}
```

Nota: Código fuente php de autenticación.

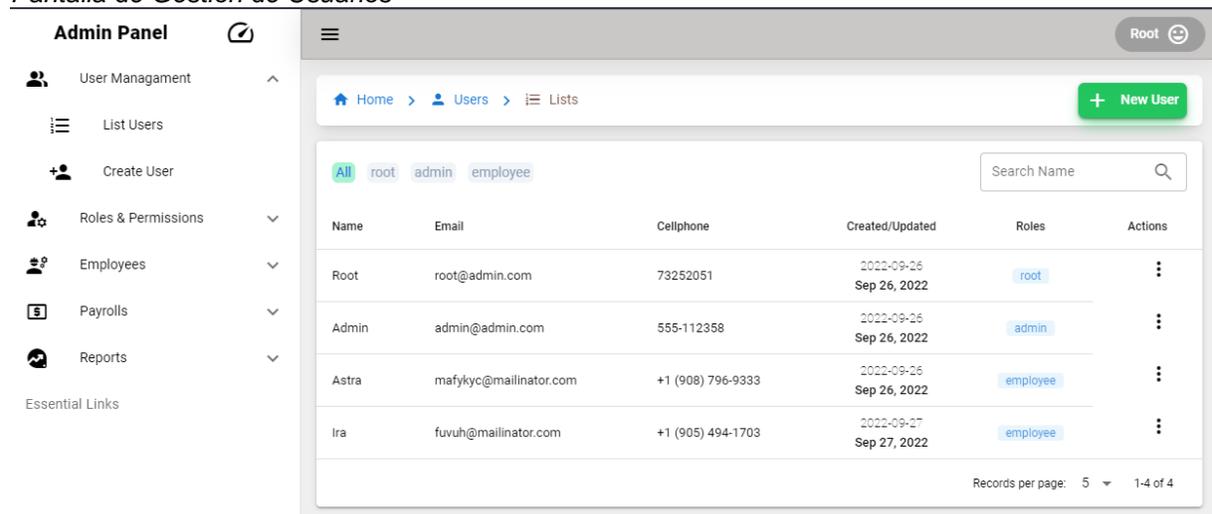
Figura 3.33
Pantalla de bienvenida al Sistema



Nota: Pantalla de bienvenida que muestra todos los módulos.

Modulo Gestión de Usuarios

Figura 3.34
Pantalla de Gestión de Usuarios



Nota: Pantalla de la Gestión de Usuarios.

Figura 3.35

Código fuente listado de usuarios según su rol

```
public function index(Request $request)
{
    abort_if(Gate::denies('user_index'), Response::HTTP_FORBIDDEN, '403 Forbidden');

    if(!$request->user()) {
        return response()->json(['error' => 'User does not exist'], 500);
    }

    $u = new User;

    if( $u->isAdmin(Auth::user()) ) $users = User::all();
    else $users = $u->getUsers_notAdmins();

    return response( UserResource::collection($users),201);
}
```

Nota: Código fuente, según el rol logeado obtiene los usuarios.

Figura 3.36

Pantalla de Creación de Usuario

The screenshot shows the 'Create New User' form in an Admin Panel. The form is titled 'Create New User' and is located in the 'Users' section. It contains the following fields and sections:

- Your full Name:** Input your full name
- Your cellphone *:** Input your cellphone
- Your email *:** Input your email
- Your password:** Input your password
- Password Confirmation:** Input your password confirmation
- Assign Roles:** A section with three toggle switches for assigning roles:
 - root:** Administrador del Sistema, Acceso Total
 - admin:** Administrador del Sistema
 - employee:** Solo puede ver sus Asistencias
- Create User:** A green button at the bottom right of the form.

Nota: Adición de un nuevo Usuario.

Figura 3.37

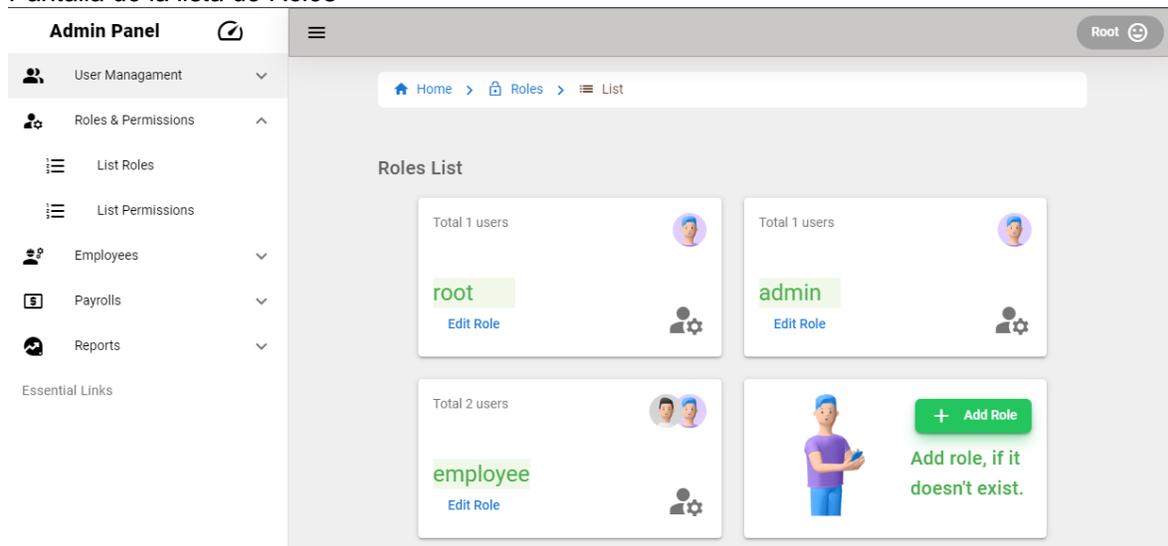
Código fuente creación usuario

```
public function store(Request $request)
{
    abort_if(Gate::denies('user_create'), Response::HTTP_FORBIDDEN, '403 Forbidden');
    $this->validate($request, [
        'name' => 'required|max:100',
        'email' => 'required|email|max:100|unique:users',
        'password' => 'required|confirmed',
        'cellphone' => 'max:50',
    ]);
    $user = User::create([
        'name' => $request->name,
        'email' => $request->email,
        'password' => $request->password,
        'cellphone' => $request->cellphone,
        'create_at' => Carbon::now(),
        'update_at' => Carbon::now(),
    ]);
    $user->roles()->attach($request->_roles);
    $response = [
        'message' => 'successfully created',
        'user' => new UserResource($user),
    ];
    return response($response, 201);
}
```

Nota: Código fuente del método store que crea el usuario.

Figura 3.38

Pantalla de la lista de Roles



Nota: Listado de todos los Roles existentes.

Figura 3.39

Código fuente listado de roles

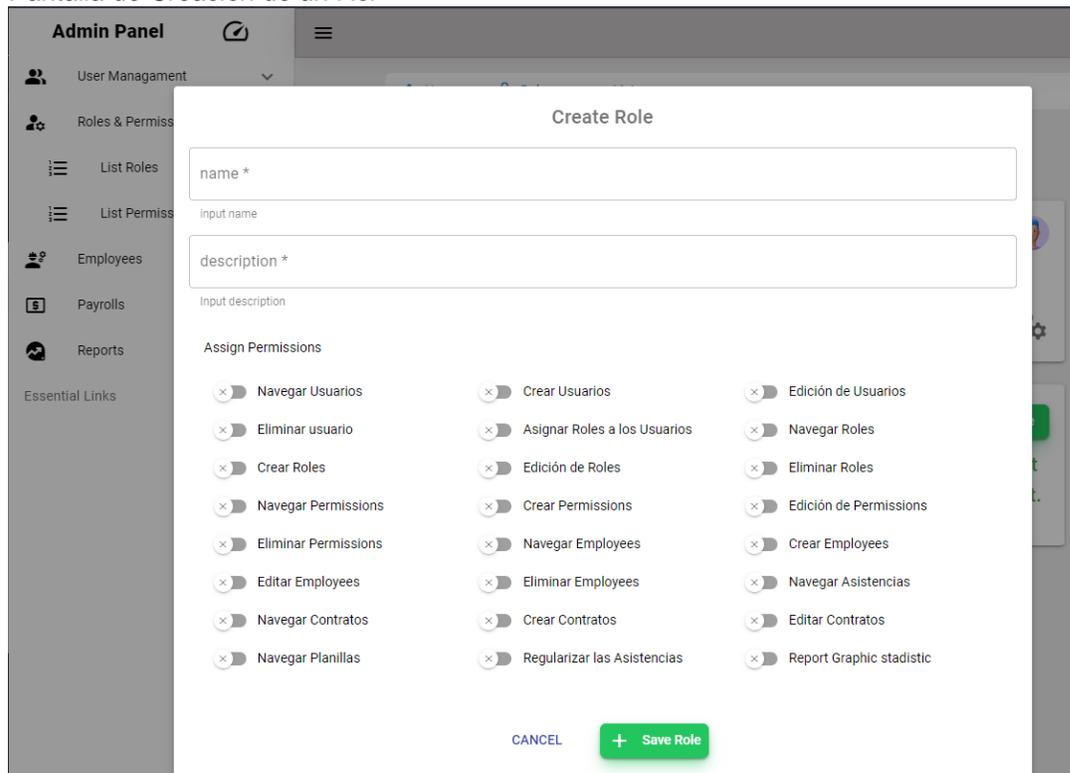
```
getRoles(){
  this.setResetState_roles()

  return new Promise( (resolve,reject) => {
    Role.indexRoles().then( response =>{
      console.log(response.data)
      this.roles = response.data
      resolve(response)
    })
    .catch(error =>{
      console.log(error);
      reject(error)
    })
  })
}
```

Nota: Código fuente javascript obteniendo los roles.

Figura 3.40

Pantalla de Creación de un Rol



Nota: Creación de un rol con los permisos a elección.

Figura 3.41

Código fuente listado de roles

```
public function store(Request $request) {  
    abort_if(Gate::denies('role_create'), Response::HTTP_FORBIDDEN, '403 Forbidden');  
    $this->validate($request, [  
        'name' => 'required|max:100',  
        'description' => 'required|max:100',  
    ]);  
    $role = Role::create([  
        'name' => $request->name,  
        'description' => $request->description,  
        'create_at' => Carbon::now(),  
        'update_at' => Carbon::now(),  
    ]);  
    $role->permissions()->attach($request->_permissions);  
    $response = [  
        'message' => 'successfully created',  
        'role' => new RoleResource($role),  
    ];  
    return response($response, 201);  
}
```

Nota: Código fuente método store que crea rol.

Figura 3.42

Pantalla de listado de permisos

Name	Assigned To	Created	Actions
Navegar Usuarios	root	2022-09-27T02:16:52.000000Z	Edit Delete
Crear Usuarios	root	2022-09-27T02:16:52.000000Z	Edit Delete
Edición de Usuarios	root	2022-09-27T02:16:52.000000Z	Edit Delete
Eliminar usuario	root	2022-09-27T02:16:52.000000Z	Edit Delete
Asignar Roles a los Usuarios	root	2022-09-27T02:16:52.000000Z	Edit Delete

Nota: Listado de todos los permisos del sistema.

Figura 3.43

Código fuente listado de permisos

```
getPermissions(){
    return new Promise( (resolve,reject) => {
        Permission.indexPermissions().then( response =>{
            console.log(response.data)
            this.permissions = response.data
            resolve(response)
        })
        .catch(error =>{
            console.log(error)
            reject(error)
        })
    })
}
```

Nota: Código fuente javascript que obtiene los permisos.

Figura 3.44

Pantalla de Creación de permisos

The screenshot shows an Admin Panel interface. On the left is a sidebar with a menu including 'User Management', 'Roles & Permissions', 'List Roles', 'List Permissions', 'Employees', 'Payrolls', 'Regularize A', 'List Payrolls', and 'Reports'. The main content area is titled 'Permissions' and contains a table with columns 'Name', 'Assigned To', 'Created', and 'Actions'. A modal window titled 'Create Permission' is open in the foreground, featuring three text input fields: 'name *', 'route *', and 'description *'. Below the inputs are two buttons: 'CANCEL' and '+ Save Permission'. The background also shows a breadcrumb trail 'Home > Permissions > Lists' and a '+ New Permission' button.

Nota: Adición de un nuevo permiso.

Figura 3.45

Código fuente creación permisos

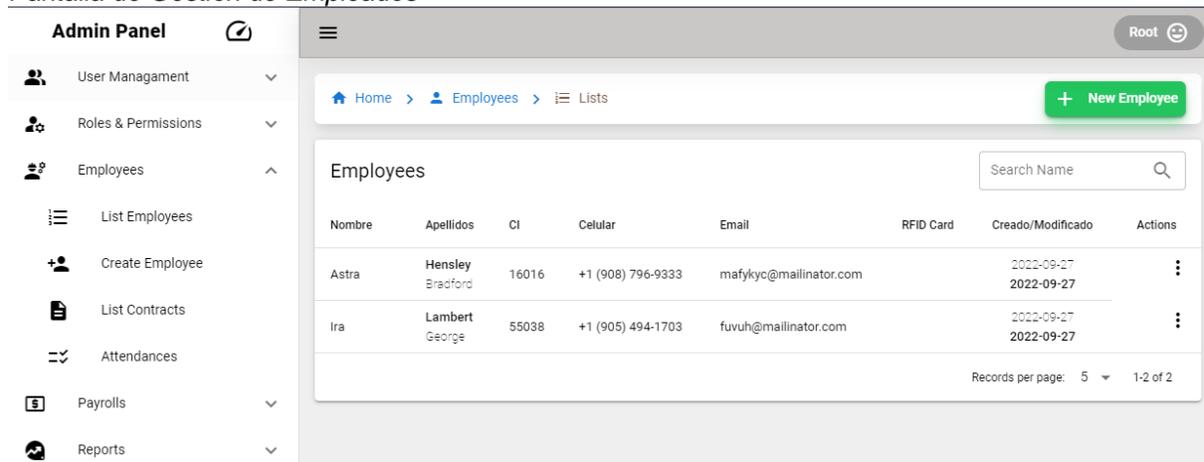
```
public function store(Request $request)
{
    abort_if(Gate::denies('permission_create'), Response::HTTP_FORBIDDEN, '403 Forbidden');
    $this->validate($request, [
        'name' => 'required|max:50',
        'route' => 'required|max:50',
        'description' => 'required|max:200',
    ]);
    $permission = Permission::create([
        'name' => $request->name,
        'route' => $request->route,
        'description' => $request->description
    ]);
    $response = [
        'message' => 'successfully created',
        'permission' => $permission,
    ];
    return response($response, 201);
}
```

Nota: Código fuente método store para la creación permiso.

Modulo Gestión de Empleados

Figura 3.46

Pantalla de Gestión de Empleados



Nota: Pantalla del listado de empleados y operación que esta tiene.

Figura 3.47

Código fuente métodos para la gestión de empleados

```
export default {
  indexEmployees() {
    return Api().get('api/employees');
  },
  storeEmployee(employee){
    return Api().post('api/employees',employee);
  },
  createEmployee() {
    return Api().get('api/employees/create');
  },
  editEmployee(id){
    return Api().get(`api/employees/${id}/edit`);
  },
  updateEmployee(id,employee){
    return Api().put(`api/employees/${id}`,employee);
  },
  deleteEmployee(id){
    return Api().delete(`api/employees/${id}`);
  },
  get_RFID_cardSerial(pair){
    return Api().get(`api/esp/${pair}`)
  }
}
```

Nota: Código fuente consumiendo las Apis.

Figura 3.48

Pantalla de Creación de un Empleado

The screenshot shows a web application interface for creating a new employee. On the left is a sidebar menu titled 'Admin Panel' with options like 'User Management', 'Roles & Permissions', 'Employees', 'List Employees', 'Create Employee', 'List Contracts', 'Attendances', 'Payrolls', and 'Reports'. The main content area is titled 'Create New Employee' and contains a form with the following fields: 'Your Name' (input your name), 'Your C.I.' (input your C.I.), 'Your last name' (input your primary lastname) and 'Your last name, second' (input your second lastname), 'Your email *' (input your email) and 'Your cellphone' (input your Cellphone), 'Your address' (input your Address), and '2000-01-31' (input your Birth Date). Below the form is a section 'Assign RFID Card' with a message 'the user has no RFID card assigned' and a button 'ASSIGN RFID CARD?'. At the bottom right is a green 'Create Employee' button.

Nota: La adición de un nuevo empleado con la asignación del TAG Rfid.

Figura 3.49

Código fuente creación empleados

```
public function store(Request $request)
{
    $rules = [ ...
    ];

    $this->validate($request, $rules);

    $employee = Employee::create([ ...
    ]);

    try {
        $employee = Employee::where('ci',$request->ci)->first();
    } catch (\Throwable $th) {
    }

    if($employee) {
        $user = User::create([
            'name'=>$request->name,
            'email'=>$request->email,
            'password'=>$request->ci,
            'cellphone'=>$request->cellphone,
            'employee_id' => $employee->id,

            'create_at'=>Carbon::now(),
            'update_at'=>Carbon::now(),
        ]);
        $user->roles()->attach([3]);
    }
}
```

Nota: Código fuente php método store para la creación.

Modulo Contratos

Figura 3.50

Pantalla de listado de los contratos

The screenshot displays the 'Admin Panel' interface. On the left is a sidebar menu with the following items: User Management, Roles & Permissions, Employees, List Employees, Create Employee, List Contracts, Attendances, Payrolls, and Reports. The main content area is titled 'Contracts' and features a search bar labeled 'Search Name'. Below the search bar is a table with the following data:

Nro. Cite	CI	Nombre	Apellidos	Inicio	Conclusion	State	Creado/Modificado	Actions
2022/001	16016	Astra	Hensley Bradford	2022-09-01	2022-12-31	CURRENT	2022-09-30 2022-09-30	⋮

At the bottom right of the table, it indicates 'Records per page: 5' and '1-1 of 1'. A '+ New Contract' button is located in the top right corner of the main content area.

Nota: Muestra el listado de todos los contratos.

Figura 3.51

Código fuente listado de contratos

```
getContracts() {  
    return new Promise( (resolve, reject) => {  
        Contract.indexContracts().then( response => {  
            console.log(response.data)  
            this.contracts = response.data  
            resolve(response)  
        }).catch(error => {  
            console.log(error)  
            reject(error)  
        })  
    })  
},
```

Nota: Código fuente javascript método para el listado de contratos.

Figura 3.52

Pantalla de la Creación de un Contrato

The screenshot displays a web application interface for creating a contract. A modal window titled "Create Contract" is open, containing the following fields:

- Select Employee:** A dropdown menu with a red error icon and the text "Please type something".
- rrhh-cite *:** A text input field containing "rrhh-cite *".
- Input rrhh-cite:** A label for the text input field.
- input your start date:** A date selection field with a calendar icon.
- input your end date:** A date selection field with a calendar icon.
- Select Contract Type:** A dropdown menu.
- Select Charge:** A dropdown menu.

At the bottom of the modal, there are two buttons: "CANCEL" and "+ Save Role".

The background shows the "Admin Panel" sidebar with options like "User Management", "Roles & Permissions", "Employees", "List Employees", "Create Employee", "List Contracts", "Attendances", "Payrolls", and "Reports". The main content area shows a "Contracts" page with a "New Contract" button and a table of contracts.

State	Creado/Modificado	Actions
CURRENT	2022-09-30 2022-09-30	⋮

Records per page: 5 1-1 of 1

Nota: Al adicionar un contrato, muestra los empleados sin contrato.

Figura 3.53

Código fuente creación contrato

```
createContract() {  
  return new Promise( (resolve, reject) => {  
    Contract.storeContract(this.contract).then( response => {  
      console.log(response.data)  
  
      this.contracts.push(response.data.contract)  
  
      let indexEmp = employeeStore.employees.findIndex( element => element.id === response.data.contract.employee_id )  
      employeeStore.employees[indexEmp].contracts.push( this.contract )  
  
      this.setReset_stateContract()  
  
      resolve(response)  
    })  
  })  
}
```

Nota: Código fuente javascript, para la creación de un contrato asignándole a un empleado.

Modulo Control de Asistencia

Figura 3.54

Pantalla de Control (Regularización) de Asistencias

The screenshot displays the 'Regularize Attendances' interface. On the left is an 'Admin Panel' sidebar with icons and labels for 'User Management', 'Roles & Permissions', 'Employees', 'Payrolls', 'Regularize Attendances', 'List Payrolls', and 'Reports'. The main area has a breadcrumb trail: 'Home > Employees > Regularize Attendances'. There are filters for 'Select Month' (September) and 'Select Year' (2022), along with a 'Search' button and an 'Export PDF' button. Below this is a table titled 'Regularize Attendances' with a search bar and columns: 'Nombre', 'Apellidos', 'Month', 'Year', 'WorkDays', 'daysworked', 'hoursworked', and 'Actions'. One record is shown: Ira Lambert George, 9, 2022, 30, 30, 240.0. Below the table, it says '1 record selected.' and 'Records per page: 2 / 1-1 of 1'. At the bottom is a calendar for 'October 2022' with a 'TO DAY' indicator and navigation buttons for 'BACK' and 'NEXT'. The calendar shows dates from 25 to 5, with the 31st highlighted. On the 4th, there are time slots: 07:45:33, 13:45:33, and 14:30:00.

Nota: Muestra a todos los empleados, al seleccionar un empleado se listará sus asistencias, para la regularización por casos fortuitos de la falta de asistencia.

Figura 3.55

Código fuente listado asistencias

```
getRegularizeAttendances(month, year) {  
  return new Promise( (resolve, reject) => {  
    RegularizeAttendace.indexRegularizeAttendances(month, year).then( response => {  
      console.log(response.data)  
      this.regularize_attendances = response.data  
      resolve(response)  
    }).catch(error => {  
      console.log(error)  
      reject(error)  
    })  
  })  
},
```

Nota: Código fuente javascript listado de asistencia para la regularización.

Figura 3.56

Pantalla de actualización de horas trabajadas

The screenshot displays an 'Admin Panel' interface. A modal window titled 'Ira LambertUpdate' is open, allowing for the regularization of attendance. The modal contains three input fields: 'work days *' with the value 30, 'Days Worked *' with the value 30, and 'Hours Worked *' with the value 240.0. Below these fields are 'CANCEL' and '+ Save Regularize' buttons. The background shows a table with the following data:

Nombre	Apellidos	Month	Year	WorkDays	daysworked	hoursworked	Actions
Ira	Lambert George	9	2022	30	30	240.0	[Edit]

At the bottom of the table, it indicates '1 record selected.' and 'Records per page: 2'.

Nota: Actualización de las horas trabajadas de un empleado de acuerdo falta por casos fortuitos de que no realizo la marcación ej.: corte de luz.

Figura 3.57

Código fuente actualización de horas trabajadas

```
updateRegularizeAttendance(id){
  return new Promise( (resolve, reject) => {
    RegularizeAttendance.updateRegularizeAttendance(id, this.regularize_attendance).then( response => {
      console.log(response.data)

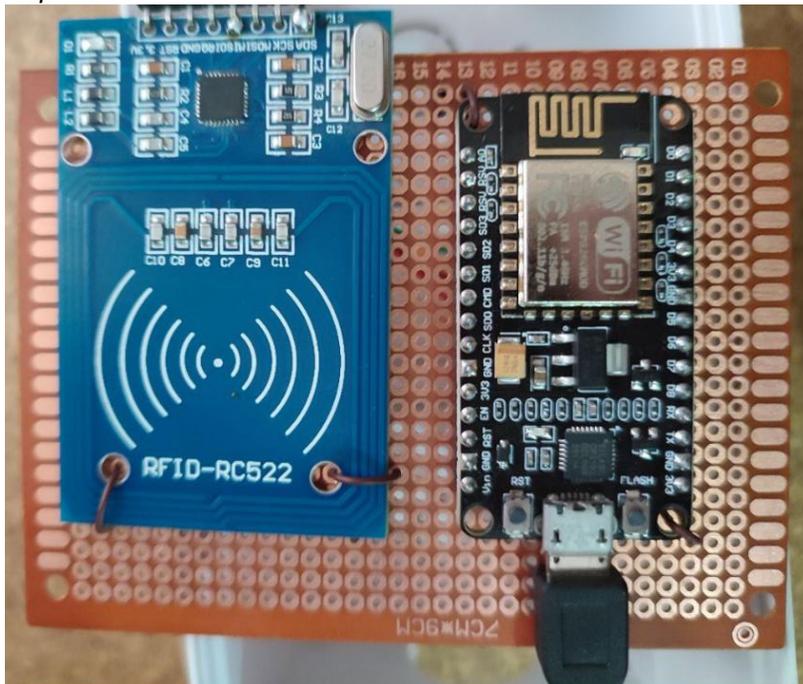
      let index = this.regularize_attendances.findIndex( element => element.id == id )
      this.regularize_attendances[index] = this.regularize_attendance

      this.setReset_stateRegularizeAttendance()
      resolve(response.data)
    }).catch(error =>{
      console.log(error)
      reject(error.response.data)
    })
  })
},
```

Nota: Código fuente javascript método para guardar la modificación de horas trabajadas.

Figura 3.58

Captura del lector RFID



Nota: Circuito que consta del lector Rfid y comunicación a la red Wi-Fi.

Figura 3.59
Captura de reporte de asistencia



Anti Money Laundering Consultants
human resources department

Attendance Records

Attendance Records

Full Name: Esther Carroll Joyce

Attendance Month: October 2022

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
25	26	27	28	29	30	1
2	3 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	4 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	5 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	6 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	7 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	8
9	10 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	11 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	12 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	13 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	14 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	15
16	17 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	18 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	19 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	20 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	21 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	22
23	24 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	25 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	26 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	27 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	28 • 08:14:47 • 13:14:52 • 13:50:55 • 18:14:58	29

Nota: Reporte pdf de las asistencias de un mes de un empleado.

Figura 3.60
Captura de reporte de Planillas



PLANILLA DE HABERES
MES NOVIEMBRE DE 2022
(Expresa en Bolivianos)

NIT 6067034013

Documento de Identidad	Apellidos y Nombres	Nacionalidad	Sexo	Ocupacion	Fecha de Ingreso	Horas Pagadas(dia)	Dias Pagados(Mes)	Haber Basico	Bono Antigüedad	Total Ganado	Aportes AFP	Total Descuento	Liquido Pagable	
2942732	EstherCarrollJoyce	Boliviana	Femelle	Soporte	2022-11-01	8	21	4500	338	3488	503	503	2985	
99383454	JessamineMendezFlores	Boliviana	Femelle	Jefe Area Juridica	2022-07-01	8	21	8000	338	5938	856	856	5081	
5969684	AngelFloresMendoza	Boliviana	Male	Tecnico RRHH	2022-11-21	8	22	4000	338	3271	472	472	2799	
77959483	JavierCarrilloLopez	Boliviana	Male	Desarrollador	2022-11-16	8	21	6000	338	4538	654	654	3883	
									22500	1352	17235	2485	2485	14748

Nota: Reporte pdf de la planilla.

Figura 3.61

Captura de reporte de boleta de pago



N.0001
BOLETA DE PAGO
Mes Noviembre 2022

Apellidos y Nombres	Nro. C.I.	Fecha de Ingreso
Angel, Flores Mendoza	5969684	2022-11-21
Cua/Nua	Cargo	Sueldo Base
10534	Tecnico RRHH	Bs.4000

Ingresos		
Sueldo Basico	22 Dia(s)	Bs.2933
Bono Antigüedad	-	Bs.338
Total Ingresos		Bs.3271

Descuentos		
A.F.P.	14.42%	Bs.472
Total Descuentos		Bs.472

TOTAL LIQUIDO PAGABLE	Bs.2799
------------------------------	----------------

Angel, Flores Mendoza
RECIBI CONFORME

11/17/2022, 10:29:21 PM

Nota: Reporte pdf de una boleta de pago de un empleado.

3.2.7 Fase de producción

En esta fase se vera la subida al servidor vps, creación de la base de datos, creación del subdominio entre otros pasos para la producción.

Figura 3.62

Subida de la aplicación al server

```
Archivo Editar Ver Buscar Terminal Ayuda
sadakiyo@sadakiyo:/media/sadakiyo/HP$ scp admin-panel-amlc.zip ezeballos@142.44.241.223:/home/ezeballos
ezeballos@142.44.241.223's password:
admin-panel-amlc.zip
```

Nota: Subida de la ampliación al servidor mediante ssh.

Figura 3.63

Asignación de permisos

```
[ezeballos@gnolab ~]$ sudo chown -R nginx:nginx /var/www/gnolab.com/admin-panel-amlc/
[ezeballos@gnolab ~]$ sudo chmod -R 775 /var/www/gnolab.com/admin-panel-amlc
[ezeballos@gnolab ~]$
```

Nota: Asignación de permisos 775 al todo el contenido del folder de la aplicación.

Figura 3.64

Creación de la base de datos

```
[ezeballos@gnolab ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ;
Your MariaDB connection id is 598
Server version: 10.5.17-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database admin_panel_amlc;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> █
```

Nota: Creación de la base de datos admin_panel_amlc.

Figura 3.65
Asignación de permisos

```
MariaDB [(none)]> grant all privileges on admin_panel_amlc.* to 'amlc'@'%' identified by '██████████';
```

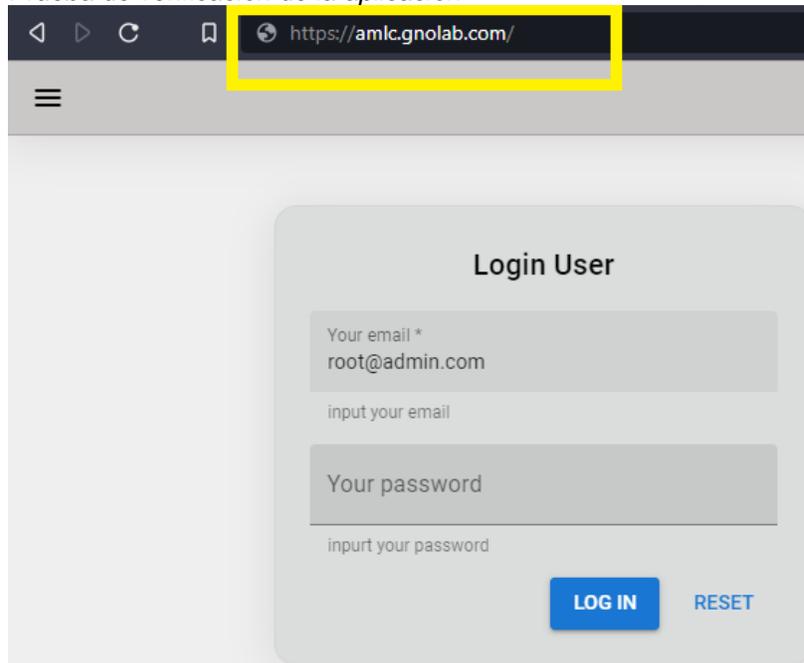
Nota: Asignamos los privilegios al usuario amlc hacia la base de datos creada; (Elaboración propia)

Figura 3.66
Creación del subdominio

Tipo ▲	Nombre	Contenido	Estado de proxy	TTL	Acciones
CNAME	amlc	gnolab.com	Redirigido por proxy	Automático	Editar ▶
A	gnolab.com	██████████	Redirigido por proxy	Automático	Editar ▶
CNAME	██████████	██████████	Redirigido por proxy	Automático	Editar ▶
CNAME	██████████	██████████	Redirigido por proxy	Automático	Editar ▶
CNAME	██████████	██████████	Redirigido por proxy	Automático	Editar ▶
MX	██████████	██████████	Solo DNS	Automático	Editar ▶

Nota: Creación del subdominio para la aplicación

Figura 3.67
Prueba de verificación de la aplicación



Nota: Verificación de acceso al servidor

CAPITULO IV

4 METRICAS DE CALIDAD, SEGURIDAD Y ESTIMACION DE COSTOS

4.1 Norma ISO/IEC 25010

Para valorar la calidad de los productos de software o sistemas que se desarrollan se proporcionan información adecuada sobre los datos referentes de la misma a la calidad del producto, permitiendo una visión más profunda sobre el cumplimiento de los objetivos del proyecto. (Pressman, 2002)

En todo sistema se debe conocer el funcionamiento y calidad del sistema por lo tanto para su evaluación de calidad del sistema se utiliza la norma ISO 25010.

4.1.1 Funcionalidad

Métrica que calcula la funcionalidad del sistema, realizando, el cálculo del punto función, para lo cual se evaluarán las siguientes características que debe cumplir el sistema.

- **Número de entradas de usuario:** Conteo del número de entrada de usuario que proporciona diferentes datos al sistema.
- **Número de salidas de usuario:** Información que elabora el sistema para que pueda ser mostrada al usuario.
- **Número de peticiones de usuarios:** Se refiere a las entradas interactivas entre el usuario y el sistema.
- **Número de archivos:** Conteo de cada archivo maestro lógico, es decir, las tablas existentes en la base de datos.
- **Número de interfaces externas:** Conteo de todas las interfaces legibles que se utilizan para transmitir la información.

Para calcular los puntos de función se emplea la siguiente formula:

$$PF = Cuenta\ Total * (0.65 + 0.01 * \sum F_i)$$

Donde:

PF: Medida de la adecuación funcional.

Cuenta Total: Es la suma de los siguientes datos: número de estradas, número de salidas, número de peticiones, numero de archivos y de numero de interfaces externas.

0,65: Confiabilidad del proyecto, varia del 1% al 100% (0 a 1).

0.01: Error mínimo aceptable de complejidad.

Fi: Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

Analizando todas las interfaces que tiene el sistema se obtuvieron los siguientes datos:

Tabla 4.1

Factor de ponderación

Parámetro de medición	Cuenta	Factor de ponderación	Total
Número de entradas de usuario	20	4	80
Numero de salidas de usuario	26	5	125
Número de peticiones de usuario	26	5	130
Numero de archivo	15	10	150
Numero de interfaces externos	1	5	5
Cuenta total			490

Nota: Parámetros de medida con factor de ponderación.

La cuenta total de los puntos función obtenida se debe ajustar en función a las características ambientales del sistema. Los valores de ajuste de complejidad Fi basados en las respuestas a las preguntas formuladas de la siguiente tabla.

Tabla 4.2

Valores de ajuste de complejidad

N.º	Factores	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
		0	1	2	3	4	5	
1	Comunicación de datos					x		4
2	Procesamiento distribuido				x			3
3	Objetivo de rendimiento						x	5
4	Configuración del equipamiento	x						1
5	Tasa de transacciones					x		4
6	Entrada de datos en línea						x	5
7	Interface con el usuario	x						1
8	Actualización en línea	x						1
9	Procesamiento complejo				x			4
10	Reusabilidad del código					x		5
11	Facilidad de implementación				x			4
12	Facilidad de operación				x			4
13	Instalaciones múltiples			x				3
14	Facilidad de cambios					x		5
Factor de Complejidad								49

Nota: Valores de ajuste de complejidad en cuanto a las preguntas formuladas.

Una vez que se consiguió los valores correspondientes a las variables de la fórmula de los puntos de función se procedió a realizar el cálculo de mismo.

$$PF = Cuenta\ Total * (0.65 + 0.01 * \sum F_i)$$

$$PF = 490 * (0.65 + 0.01 * 49)$$

$$PF = 490 * (1.14)$$

$$PF = 558.6$$

Para comparar los puntos función con su valor máximo, se calculó los puntos función

con valores de ajuste de complejidad al máximo siendo un total de 70:

$$PF_{ideal} = 490 * (0.65 + 0.01 * 70)$$

$$PF_{ideal} = 661.5$$

Después de haber calculado ambos valores se tiene que la funcionalidad real es:

$$Adecuacion\ funcional = \frac{pf}{PF_{ideal}} * 100\%$$

$$Adecuacion\ funcional = \frac{pf}{PF_{ideal}} * 100\%$$

$$Adecuacion\ funcional = 84.44 \%$$

Por tanto, se tiene una funcionalidad del 84,44 %.

4.1.2 Usabilidad

Característica que muestra el esfuerzo necesario para la manipulación del sistema, en la siguiente tabla muestra los valores obtenidos para cada pregunta:

Tabla 4.3

Encuesta sobre la usabilidad del sistema

N.º	PREGUNTA	RESPUESTA		% de SI
		SI	NO	
1	¿Aprendió a usar rápido el sistema?	4	1	80
2	¿Las vistas de pantalla que vio fueron de su agrado?	5	0	100
3	¿Las pantallas que vio fueron fáciles de comprender?	5	0	100
4	¿El sistema responde rápido a sus solicitudes?	5	0	100
5	¿El sistema le facilita el trabajo?	5	0	100
6	¿El sistema reduce su tiempo de	5	0	100

	trabajo?			
7	¿Es fácil navegar por las distintas opciones?	5	0	100
8	¿Las operaciones que se realizan no son complicadas?	5	0	100
9	¿El sistema le proporciono las respuestas requeridas?	4	1	80
10	¿El sistema no presento errores?	4	1	80
Resultados de la usabilidad es de:				94%

Nota: Encuesta sobre la usabilidad del sistema.

4.1.3 Fiabilidad

Característica que mide la capacidad del software en su funcionamiento tomando en cuenta las fallas que pueden ocurrir en un tiempo determinado. Para el cálculo de fiabilidad de cada módulo se tiene la siguiente ecuación:

$$R(t) = e^{-\lambda t}$$

Donde:

R(t): Fiabilidad de un componente o subsistema t.

λ : Tasa de constantes de fallo (λ = número de fallas de acceso/ número total de accesos al sistema).

t: Periodo de operación de tiempo.

$e^{-\lambda t}$: Probabilidad de falla de un componente o subsistema ene l tiempo t.

$$e = 2,72$$

Luego de realizar pruebas de cada módulo ene un tiempo de 4 horas continuas se obtiene la siguiente tabla:

Tabla 4.4

Valores de fiabilidad de cada módulo

N.º	Modulo	λ	t	R(t)
1	Modulo Gestión de Usuarios	0.02	4 horas	0.92
2	Modulo Gestión de Empleados	0.01	4 horas	0.96
3	Modulo Planilla	0.02	4 horas	0.92
4	Modulo Control de Asistencia	0.03	4 horas	0.89
5	Modulo Contrato	0.02	4 horas	0.92
6	Modulo Área	0.01	4 horas	0.96

Nota: Lista de valores de fiabilidad de cada módulo.

Para el cálculo de la fiabilidad del sistema completo se tiene la siguiente formula:

$$Fiabilidad = R_s * R_p$$

Donde:

$$R_s = R_1 = 0.92 \quad \text{y} \quad R_p = \frac{\sum_{i=2}^5 (R_i * P_i)}{\sum_{i=2}^5 P_i}$$

La fórmula de R_p , la variable P_i es la participación en el equipo de desarrollo del módulo y como la participación fue al 100% por tanto $P_i = 1$, del cual si tiene el siguiente resultado:

$$R_p = \frac{\sum_{i=2}^5 R_i}{6} = \frac{0.92+0.96+0.92+0.89+0.92+0.96}{6} = \frac{5.57}{6} = 0.93$$

Reemplazando los valores calculados en la fórmula: $Fiabilidad = R_s * R_p$ se tiene:

$$Fiabilidad = 0.92 * 0.93 = 0.8556 * 100 = 85.56\%$$

Es decir, que la fiabilidad del sistema es de 85.56 %

4.1.4 Mantenibilidad

Es la cualidad del software para ser modificado, corregido o mejorado. Para obtener la calidad de mantenimiento se utilizará el índice de madurez (IMS), para determinar la estabilidad del producto. El índice de madurez se calcula con la siguiente formula:

$$IMS = \frac{M_t - (F_a + F_b + F_c)}{M_t}$$

Donde:

M_t: Número de módulos en la versión actual.

F_a: Número de módulos en la versión actual que se han cambiado.

F_b: Número de módulos en la versión actual que se han añadido.

F_c: Número de módulos en la versión anterior que se han borrado en la versión actual.

Recopilando la información necesaria, se obtuvo que:

M_t = 5 por el número de módulos actuales en el sistema.

F_a, *F_b*, *F_c* = 0 no se cuenta con estos valores.

Con los valores obtenidos se realiza el cálculo de IMS, para el cual reemplazamos los valores la fórmula:

$$IMS = \frac{6 - (0 + 0 + 0)}{6} = \frac{6}{6} = 1$$

$$IMS = 1 * 100\% = 100\%$$

Con el resultado obtenido se concluye que el sistema tiene un índice de madurez del 100%.

4.1.5 Portabilidad

Característica que refiere a la capacidad que tiene el software para ser llevado de un entorno a otro, tomando en cuenta la facilidad de instalación, ajuste y adaptación al cambio. Se cuenta con la siguiente fórmula para obtener el grado de portabilidad:

$$GP = 1 - \frac{ET}{ER}$$

Donde:

GP: Grado de portabilidad.

Si: **GP** > 0, la portabilidad es más rentable que el re-desarrollo.

GP < 0, el re-desarrollo es más rentable que la portabilidad.

GP = 1, la portabilidad es perfecta.

ET: Recursos necesarios para llevar el sistema a otro entorno.

ER: Recursos necesarios para crear el sistema en el entorno residente.

Los recursos necesarios para llevar el sistema a otro entorno son: Un servidor vps con Sistema operativo GNU/Linux, 1 Gb min de almacenamiento. Por lo tanto, **ET** = 1.

Los recursos necesarios para crear el sistema son: IDE de desarrollo, framework (Laravel, Quasar, vuejs, platformio). Por lo tanto, **ER** = 5. Con los valores obtenidos se reemplazan los valores en la ecuación, para la obtención de grado de portabilidad:

$$GP = 1 - \frac{1}{5}$$

$$GP = 1 - 0.2$$

$$GP = 0.8 * 100\% = 80 \%$$

Por lo tanto, el grado de portabilidad es 0.8, lo que indica que el sistema es más rentable que el re-desarrollo.

4.1.6 Calidad total

Una vez calculadas las características anteriores (adecuación funcional, usabilidad, fiabilidad, mantenibilidad y portabilidad) conforme la norma ISO/IEC 25010, se realiza el cálculo de la calidad total, obteniendo la media de todos estos valores.

Tabla 4.5*Cálculo de calidad global*

CARACTERISTICAS	VALOR EN %
Adecuación funcional	84.44
Usabilidad	94
Fiabilidad	85.56
Mantenibilidad	100
Portabilidad	80
Calidad Total es de:	88.8 %

Nota: Sumatoria total del cálculo de calidad.

4.2 Estimación de costos

Consiste en determinar, con un grado de certeza de los recursos de hardware y software, costo, tiempo y esfuerzo, necesarios para el desarrollo del sistema.

4.2.1 Análisis de costo

Se tiene los siguientes aspectos:

$$E = \frac{\text{Esfuerzo hombre}}{\text{mes}}$$

KLDC = Número estimado de miles de líneas de código

El factor de conversión de *KLDC* se muestra en la siguiente tabla el cual especifica el nivel que pertenece el lenguaje PHP que fue utilizado para el desarrollo del proyecto.

Tabla 4.6*Conversión de puntos de función a KLDC*

Lenguaje	Nivel	Factor LDC/PF
C	2.5	128
Java	6	53
Php	11	29
Visual C++	9.5	34

Nota: Lista de puntos de función, según el lenguaje de programación; (T. Chambi, 2007)

Como se indica en la Tabla 4.6. el lenguaje de programación en que se basa el sistema es PHP por lo que: $Factor\ LDC/PF = 29$.

El Punto Función es: $PF = 558.6$ que fue obtenido anteriormente.

4.2.2 Costo del desarrollo del software del sistema

Las líneas de código obtenidas son:

$$LDC = 7090$$

Convirtiendo las LDC en KLDC:

$$KLDC = \frac{LDC}{1000}$$

$$KLDC = \frac{7090}{1000}$$

$$KLDC = 7.09$$

Aplicando las fórmulas de esfuerzo tiempo calendario y personal requerido:

$$E = a(KLDC)^b$$

$$Tdev = C(E)^b$$

Donde:

- E: Es el esfuerzo en personas por mes.
- Tdev: Es el tiempo de desarrollo en meses cronológicos.
- KLDC: Es el número de líneas de código distribuidos en miles.

Tabla 4.7

Tabla de Constantes COCOMO

MODO	A	B	C	D
Orgánico	2.40	1.05	2.50	0.38
Semi-acoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Nota: Tabla de constantes COCOMO; (Pressman, 2010)

Calculando el esfuerzo:

$$E = 3(7.09)^{1.12}$$

$$E = 26.91$$

Realizando el cálculo del tiempo de desarrollo:

$$Tdev = 2.5(26.91)^{0.35}$$

$$Tdev = 7.9 \text{ Meses}$$

Por lo tanto, el número de programadores es:

$$NP = \frac{E}{Tdev}$$

$$NP = \frac{26.91}{7.9} = 3$$

Si en caso de que el salario que percibiera un programador es de 350\$, entonces se tiene lo siguiente:

El número de programadores por salario de programador es:

$$CT = sueldoMes * NP * Tdev$$

$$CT = 350 * 3 * 7.9$$

$$CT = 8295 \text{ \$us}$$

Por lo tanto, se requiere 3 personas, con 7.9 meses de trabajo por lo que el costo total

del sistema es de 8.295 \$us.

El costo por línea de código es:

$$\text{Costo LDC} = \frac{CT}{LDC}$$

$$\text{Costo LDC} = \frac{8295}{7090}$$

$$\text{Costo LDC} = 1.17 \text{ \$us}$$

Tabla 4.8

Resumen COCOMO

CARACTERISTICAS	VALOR
Líneas de código LDC	7090
Costo por línea de código	1.17 \$us.
Número de programadores	3
Costo total del sistema	8295 \$us.

Nota: Resumen de los cálculos obtenidos.

4.3 Seguridad

La norma ISO 27002 proporciona diferentes recomendaciones de las mejores prácticas en la gestión de la seguridad de la información a todos los responsables para iniciar, implementar o mantener sistemas de gestión de la seguridad de la información. Considerando las recomendaciones más relevantes especificadas en la citada norma con respecto a la presentación de las características de confiabilidad, integridad y disponibilidad de la información se incorpora las siguientes medidas de seguridad del sistema.

Tabla 4.9

Medidas de seguridad

Recomendaciones ISO/IEC 27002	Medidas de seguridad incorporada en el sistema
Control de acceso (token de autenticación)	<p>Se implementó como elemento importante la autenticación del usuario que consta de email y contraseña, el usuario deberá estar previamente autenticado con el token generado por la librería Sanctum para realizar cualquier acción, caso contrario será restringido.</p> <pre data-bbox="521 695 1430 892">\$token = \$user->createToken('myapptoken')->plainTextToken \$response = ['user' => \$user, 'token' => \$token]; return response(\$response, 201);</pre>
Control de acceso (RFID)	<p>Se restringe el acceso físico a las áreas críticas a toda persona no autorizada, para reducir el riesgo de accidentes fraudulentos</p>
Encriptación	<p>Es uno de los algoritmos de cifrado más utilizados y seguros para la encriptación de contraseña, es uno del dato de suma importancia para el ingreso al sistema. de este modo se está utilizando lo que es el algoritmo de B-CRIPPT una encriptación de seguridad para el sistema.</p> <pre data-bbox="521 1360 1398 1459">public function setPasswordAttribute(\$password) { \$this->attributes['password'] = bcrypt(\$password); }</pre>
Seguridad Lógica	<p>Los respaldos o back-up de la base de datos del sistema se realiza automáticamente una vez a la semana, con la creación de un archivo .bash</p>

Nota: Listado de medidas de seguridad según el ISO/IEC 27002.

4.4 Pruebas

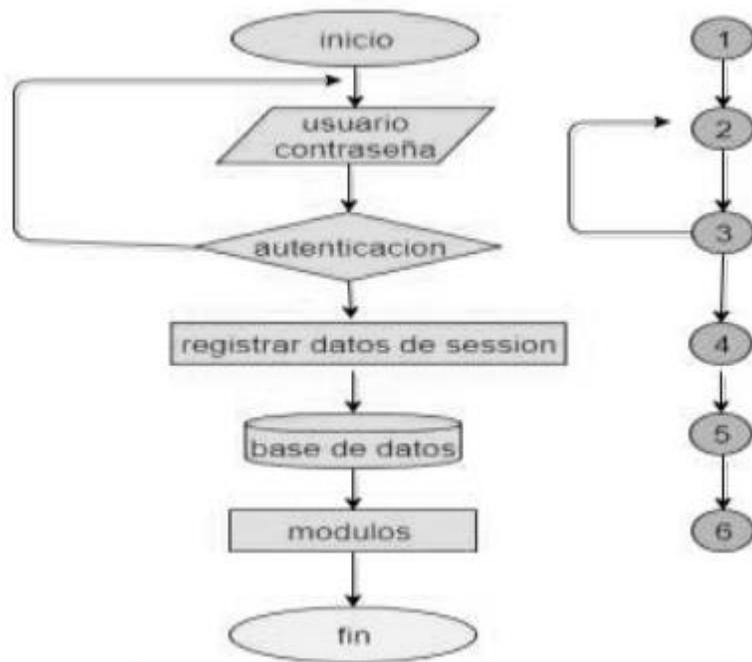
Para el presente proyecto nos basaremos en las dinámicas que proponen las pruebas de caja blanca y pruebas de caja negra, así mismo se contemplan pruebas de estrés

que ayudaran a verificar la saturación del sistema.

4.4.1 Prueba de Caja Blanca

Figura 4.1

Flujograma del sistema



Nota: Flujograma del sistema.

La complejidad se obtiene en base al grafo de flujo se define como:

$$V(G)=E-N+2$$

Dónde: E= número de aristas y N= número de nodos.

$$V(G)=6-6+2=2$$

Por lo tanto, las existencias son de 2 caminos independientes, para los que se realizan las pruebas.

Caso de prueba camino 1

Valor (usuario='admin', contraseña='r00t&^*##(\$)') = entrada validada ya que los datos ingresados existen en la base de datos, rol como administrador.

Ingresa al sistema, visualiza los módulos. Resultados esperados correctos.

Caso de prueba camino 2

Valor (usuario='UsuarioGuest', contraseña='ninguno') = entrada no validada ya que los datos ingresados no existen en la base de datos, vuelva a introducir datos.

Actualiza datos de entrada.

Resultados esperados correctos

4.4.2 Prueba de Caja Negra

La prueba de caja negra consiste en probar cada una de las funciones del sistema que fueron especificadas en el capítulo II.

Con este tipo de prueba se debe buscar que las funciones sean operativas y optimas, además se debe agotar al sistema de tal manera buscar la mayor cantidad de errores. Estas pruebas están referidas que algunos del usuario deben probar el sistema las cuales son llevadas sobre el interfaz del sistema actuando como una caja negra introduciendo datos y estudiando las salidas para ver si son las esperadas.

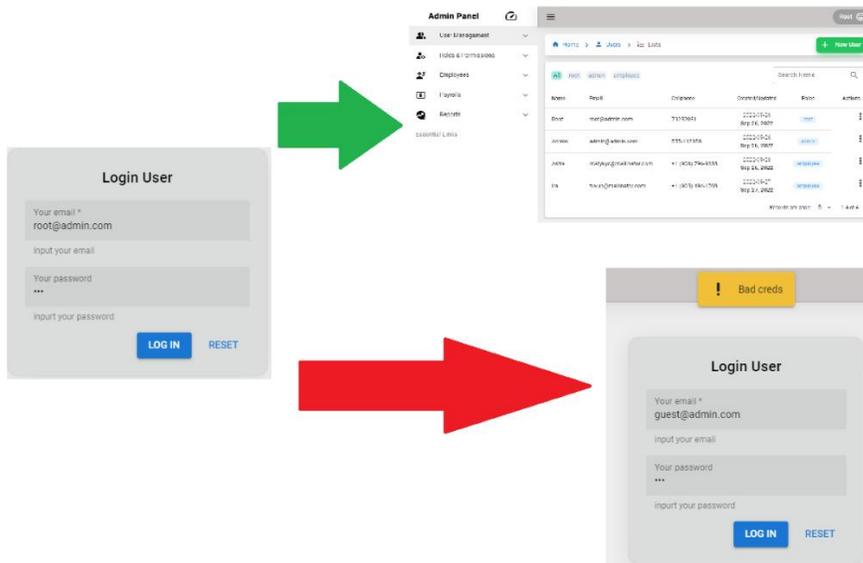
Son pruebas sobre la interfaz del software, a continuación, se muestran algunas pruebas relevantes.

Prueba de autenticación

Si el usuario que desea ingresar al sistema es el correcto desplegara la interfaz gráfica de ingreso de la bienvenida al usuario, caso contrario se mostrara un mensaje de acceso denegado o erróneo.

Figura 4.2

Autenticación positiva y errónea



Nota: Autenticación positiva y errónea.

Tabla 4.10

Valores de entrada-proceso de inicio de sesión del usuario

Campo	Ingreso exitoso	Ingreso fallido
Usuario	Nombre del usuario	Espacio vacío
Contraseña	Contraseña del usuario	Espacio vacío

Nota: Valores de entrada-proceso de inicio del usuario, que puede ser exitoso o fallido.

Tabla 4.11

Pruebas de caja negra ingreso usuario

Entradas		Salida	Resultado
Usuario	Contraseña		
Correcto/ Incorrecto		Ingrese usuario y contraseña	Si el nombre está lleno y la contraseña está vacío el usuario debe completar el campo vacío.
	Correcto/ Incorrecto	Ingrese usuario y contraseña	Si la contraseña esta y el nombre esta vacía el usuario debe completar el campo.
Correcto/ Incorrecto	Incorrecto	El nombre y contraseña no coinciden	Si el nombre es correcto/incorrecto y la contraseña es incorrecto el usuario debe ingresar nuevamente los datos.

Incorrecto	Correcto/ Incorrecto	El nombre y contraseña coinciden	y no	Si la contraseña es correcta/incorrecta y el nombre es incorrecto el usuario debe ingresar nuevamente los datos.
Incorrecto	Incorrecto	El nombre y contraseña coincide	y no	Si ambos campos son incorrectos el usuario deberá ingresar nuevamente los datos.
Correcto	Correcto	El usuario ingresa al sistema		Si ambos campos son correctos el usuario ingresa al sistema.

Nota: Prueba de caja negra de Login, con todas las posibilidades de entradas.

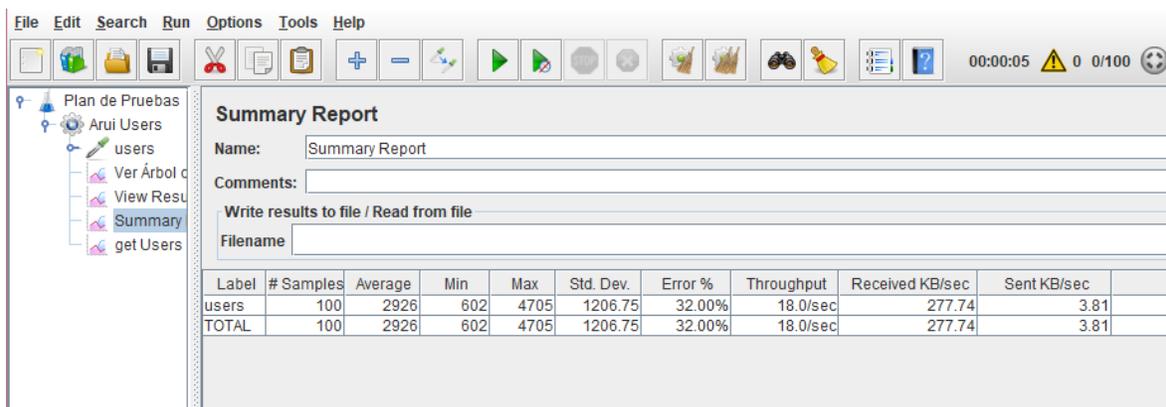
Realizando esta prueba se ha podido constatar que el proceso de autenticación cumple con sus funciones correctamente.

4.4.3 Pruebas de estrés

Probar un sistema enfatizando aspectos como la robustez, disponibilidad, manejo de errores y otros. Bajo una carga pesada, el objetivo de estas pruebas es de asegurar que el sistema no colapsara cuando disponga de pocos recursos o exista una gran concurrencia de usuarios, para el mismo recurriremos a la aplicación open source **Apache JMeter**.

Figura 4.3

Prueba de estrés con Apache JMeter



Nota: Prueba de estrés con 100 usuarios haciendo petición en un lapso de 1 segundo, donde se puede observar los resultados.

CAPITULO V

5 CONCLUSIONES Y RECOMENDACIONES

En este capítulo se mostrará las conclusiones y recomendaciones que contiene el presente documento. Para llevar a cabo este trabajo se recabo información exhaustiva del proceso de desarrollo de software, áreas de conocimiento de ingeniería de sistemas y otras disciplinas dentro de la informática, se leyó bibliografía de libros, proyectos de grado similares y páginas web relacionadas con el tema.

5.1 Conclusiones

Una vez de culminado con el desarrollo y la implementación del sistema web para la empresa AMLC, se logró dar solución al problema principal, así también logrando cumplir con todos los objetivos específicos.

- Se automatizo la generación de planillas de sueldos con el cálculo de sus beneficios y descuentos de cada empleado.
- Se automatizo la generación de boletas de pago de los empleados
- Se desarrollo un módulo de reportes de asistencias para los empleados para que tengan un seguimiento de los mismos.
- Se diseño una base de datos en la cual se registran todos los datos que intervienen para el funcionamiento del sistema web.
- Se armo el circuito de comunicación RFID para el marcado de asistencias y control de acceso para los empleados.

De esta manera se concluyó con el objetivo general del proyecto satisfactoriamente, con el desarrollo de sistema web de control de asistencia y control de acceso a las áreas de la empresa para el personal, utilizando la tecnología RFID, esto permite generar la planilla de sueldos de manera eficiente y tener un control adecuado del personal.

5.2 Recomendaciones

Si bien el sistema cuenta con un mayor nivel de confiabilidad y seguridad, es necesario realizar acciones para permitir mantener la madurez del sistema, para ello se recomienda:

- Se recomienda implementar el módulo de contabilidad para mayor integración de información centralizada.
- Se recomienda que el circuito de lectura Rfid este fuera de alcance de altas temperaturas y un mantenimiento preventivo al circuito.
- Se recomienda implementar un lector de Rfid industrial para el acceso de áreas.
- Se recomienda capacitar a los nuevos administrativos.

BIBLIOGRAFÍA

Alvarado (2008). *Sistema de Control de Acceso con RFID (Trabajo de Grado)*. Instituto Politécnico Nacional, México.

Cali (2020). *Control de asistencia de estudiantes mediante dispositivos biométricos RFID en la universidad estatal del sur de Manabí (Tesis de Grado)*. Universidad Estatal del sur de Manabí, Ecuador.

Vargas (2013). *Sistema de Control de Acceso y Monitoreo con la Tecnología RFID para el Departamento de Sistemas de la Universidad Politécnica Salesiana Sede Guayaquil (Proyecto de Grado)*. Universidad Politécnica Salesiana, Ecuador.

Alcón (2016). *Sistema de control y monitoreo de asistencia de personal mediante radiofrecuencia y NFC (Proyecto de Grado)*. UMSA, La Paz.

Silva (2016). *Sistema Biométrico para el control de personal y generación de planillas de sueldos, impositiva y patronal (Proyecto de Grado)*. UMSA, La Paz.

Sánchez (2020). *Sistema de control y Administración de Recursos Humanos para la incubadora de microempresas productivas (Proyecto de Grado)*. UPEA, La Paz.

Forra (2020). *Sistema web para la venta de ladrillos y control personal (Proyecto de Grado)*. UPEA, La Paz.

Pressman, S. R. (2010). *Ingeniería del software*. Mexico: McGRAW-HILL

Seen (1999). *Requerimientos para aplicaciones web*. In XIII Workshop de Investigadores en Ciencias de la Computación.

Sommerville, I., (2005), *Ingeniería del Software*. Séptima Edición, Madrid, Pearson Educación.

Montevilla (1999). *Introducción a las Tecnologías Web*. Madrid: Informática Tributaria.

Ramos & Ramos (2014). *Metodologías De Desarrollo En Aplicaciones Web*. Universidad Técnica de Machala.

Martin & Martin (2011). *Programación web dinámicos*. Madrid.

Morales, D. S. (2012). Prototipo de Control de Acceso Peatonal al Campus de la Corporación Universitaria. Caldas Antioquia.

WEB

Cordova JR. (2013). Metodología UWE-UML. Recuperado el 10 de noviembre de 2019, de: <http://www.enubes.com/desarrollo-de-aplicaciones-web.html>.

Red Hat,(2021) .aplicación de interfaces recuperado de <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>. [Último acceso: 01 Julio 2021].

M. Pérez, «Geek Theory,». Recuperado de: <https://geekytheory.com/que-es-una-api-rest-y-para-que-se-utiliza> .[Último acceso: 29 junio 2021].

Mi Trabajo Bolivia. (2020). Clases de salario. Obtenido de Mi Trabajo Bolivia: <https://mitrabajobolivia.com/clases-de-salario/>

Mi Trabajo Bolivia. (s.f.). Pago de Salario y Aportes de Ley. Obtenido de Mi Trabajo Bolivia: <https://mitrabajobolivia.com/pago-de-salario-y-aportes-de-ley/>

Ministerio de trabajo. (13 de mayo de 2019). Salario mínimo nacional. Obtenido de Ministerio de trabajo: https://www.mintrabajo.gob.bo/images/normativa/RM/rm_425_2019.pdf

Ministerio de trabajo empleo y previsión social. (20 de octubre de 2016). Reglamento de la ley general de trabajo. Obtenido de Ministerio de trabajo: <https://www.mintrabajo.gob.bo/images/normativa/LEY/13-REGLAMENTO DE LEY GENERAL DEL TRABAJO.pdf>

ISO 25000 Calidad del producto Software. (2019). Recuperado de <https://iso25000.com/index.php/normas-iso-25000>

ISO/IEC 25010. (2019). Normas ISO 25010 Recuperado de <http://www.iso25000.com/index.php/normas-iso-25000/iso-25010>

ISO 27001: Los activos de información. (2015-03-30). Recuperado de <https://www.pmq-ssi.com/2015/03/iso-27001-los-activos-de-informacion/>

CAPÍTULO II

MARCO TEORICO

CAPÍTULO III

MARCO APLICATIVO

CAPÍTULO IV

METRICAS DE CALIDAD, SEGURIDAD Y ESTIMACION DE COSTOS

CAPÍTULO V

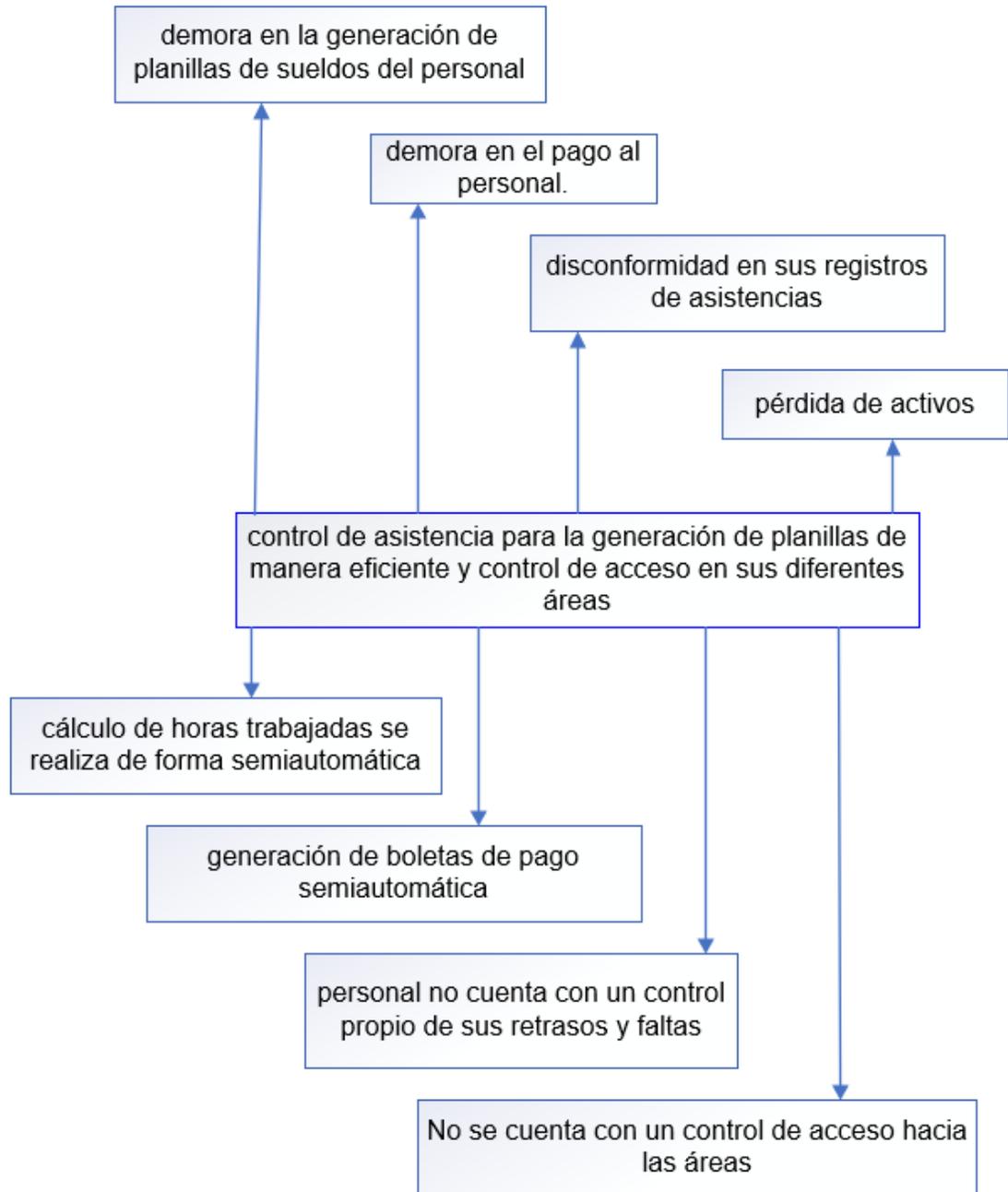
CONCLUSIONES

Y

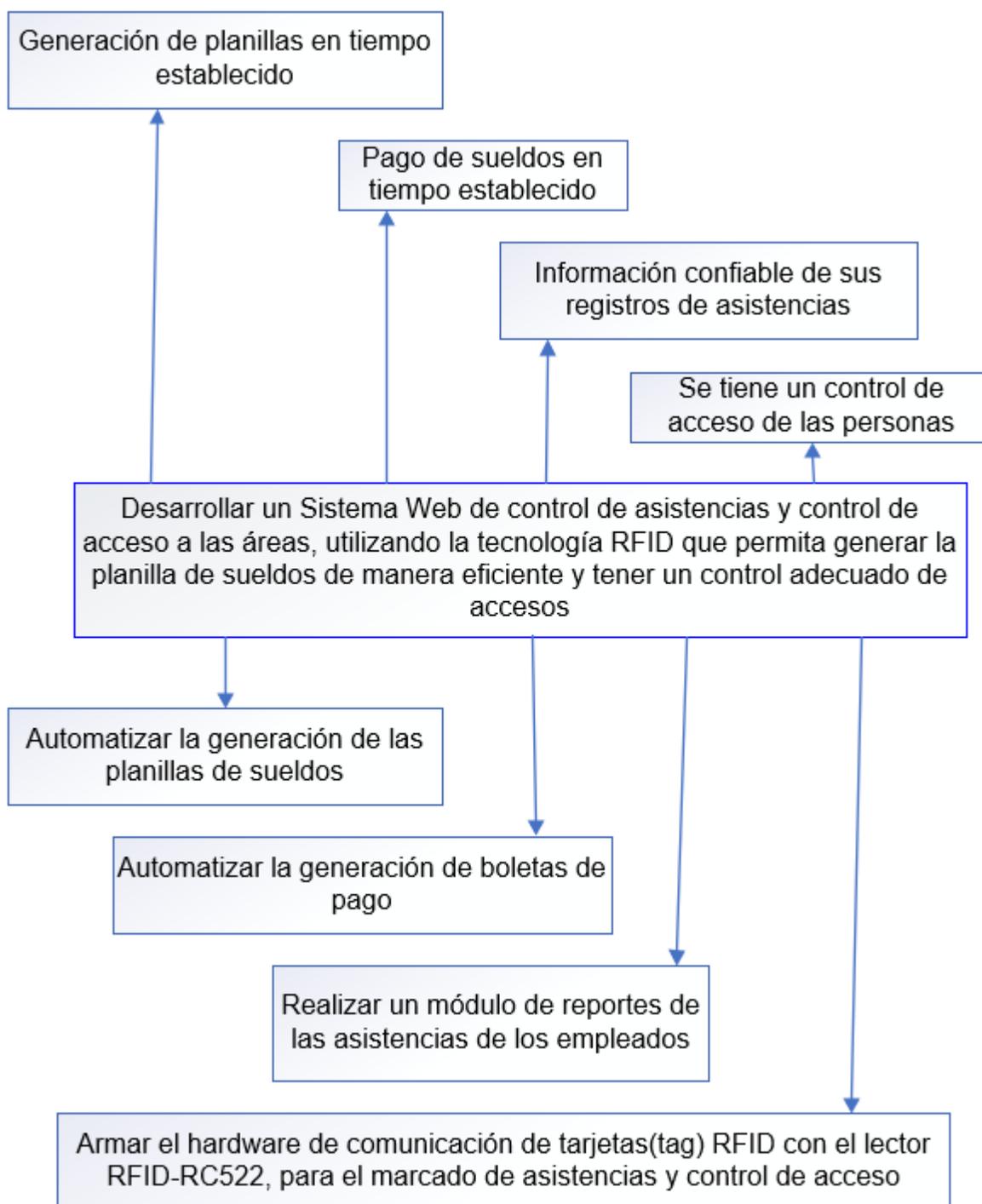
RECOMENDACIONES

ANEXOS

ANEXO A: ARBOL DE PROBLEMAS



ANEXO B: ARBOL DE OBJETIVOS



ANEXO C: CUESTIONARIO

Nombres y apellidos:

Cargo:

Marca en las casillas en SI o NO

Tu sinceridad es muy importante en cuanto es facilidad de uso del sistema

PREGUNTAS	RESPUESTA	
	SI	NO
¿Aprendió a usar rápido el sistema?		
¿Las vistas de pantalla que vio fueron de su agrado?		
¿Las pantallas que vio fueron fáciles de comprender?		
¿El sistema responde rápido a sus solicitudes?		
¿El sistema le facilita el trabajo?		
¿El sistema reduce su tiempo de trabajo?		
¿Es fácil navegar por las distintas opciones?		
¿Las operaciones que se realizan no son complicadas?		
¿El sistema le proporciono las respuestas requeridas?		
¿El sistema no presento errores?		

La Paz, 14 de noviembre de 2022

Señor:
M.Sc. Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE ACCESO Y ASISTENCIA PARA EL PROCESO DE PLANILLAS MEDIANTE RFID
CASO: ANTI MONEY LAUNDERING CONSULTANTS
MODALIDAD: PROYECTO DE GRADO
Univ. EDDY ZEBALLOS QUISPE
Registro Universitario: 5336
Cedula de Identidad: 6067034 LP

De tal forma cabe recalcar que el SISTEMA/PROYECTO fue **IMPLEMENTADO** satisfactoriamente y se realizó las capacitaciones respectivas en la empresa.

En cuanto certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


Franco Rojas
GERENTE GENERAL
AMLC BOLIVIA S.R.L.


Anti Money Laundering Consultants

El Alto, 18 de noviembre de 2022

Señor:
M. Sc. Ing. David Carlos Mamani Quispe
DIRECTOR DE CARRERA
INGENIERÍA DE SISTEMAS
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido director:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE ACCESO Y ASISTENCIA PARA EL PROCESO DE PLANILLAS MEDIANTE RFID
CASO: ANTI MONEY LAUNDERING CONSULTANTS
MODALIDAD: PROYECTO DE GRADO
Univ. EDDY ZEBALLOS QUISPE
Registro Universitario: 5336
Cedula de Identidad: 6067034 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


M.Sc. Ing. Enrique Flores Baltazar
Tutor Metodológico

El Alto, 16 de noviembre de 2022

Señor:
M.Sc. Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

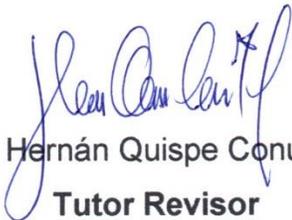
REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE ACCESO Y ASISTENCIA PARA EL PROCESO DE PLANILLAS MEDIANTE RFID
CASO: ANTI MONEY LAUNDERING CONSULTANTS
MODALIDAD: PROYECTO DE GRADO
Univ. EDDY ZEBALLOS QUISPE
Registro Universitario: 5336
Cedula de Identidad: 6067034 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


Ing. Hernán Quispe Conurana
Tutor Revisor

El Alto, 16 de noviembre de 2022

Señor:

M.Sc. Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO

TALLER DE GRADO II

Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE ACCESO Y ASISTENCIA PARA EL PROCESO DE PLANILLAS MEDIANTE RFID

CASO: ANTI MONEY LAUNDERING CONSULTANTS

MODALIDAD: PROYECTO DE GRADO

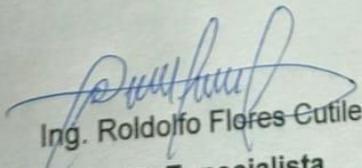
Univ. EDDY ZEBALLOS QUISPE

Registro Universitario: 5336

Cedula de Identidad: 6067034 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,


Ing. Roldolfo Flores Cutile
Tutor Especialista

MANUAL TÉCNICO

El presente manual técnico describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación del sistema desarrollado para la empresa.

REQUERIMIENTOS DEL SISTEMA

Requerimientos mínimos de hardware

- Procesador de 4 Núcleos, frecuencia 3.6 GHz.
- Memoria RAM 2 GB. DDR 4
- Almacenamiento 500 GB

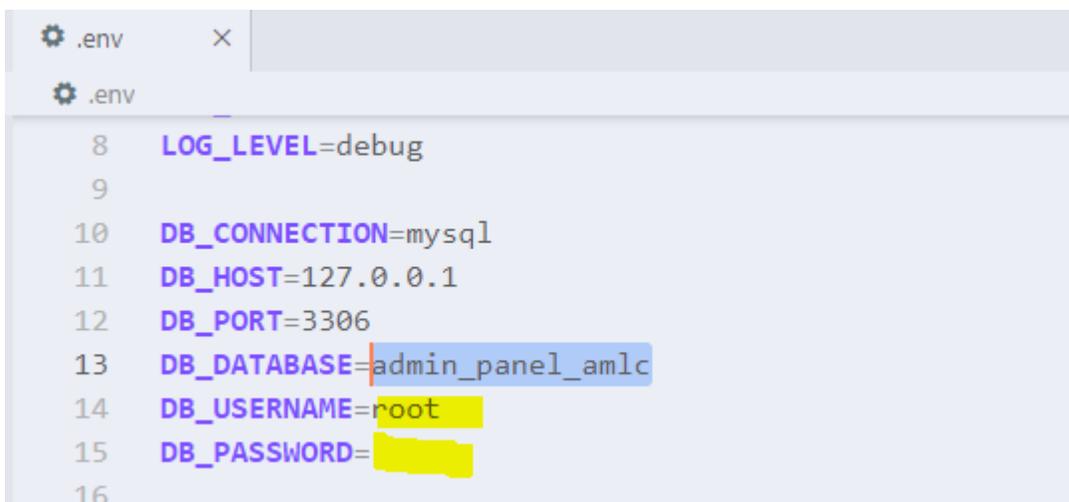
Requerimientos mínimos de software

- Cualquier distribución GNU/Linux con kernel ≥ 5.2
- Php $\geq 7.4.x$
- Conexión a Internet

INSTALACION DEL SISTEMA

Instalación BACKEND

Crear la base de datos con el nombre de **admin_panel_amlc**, una vez creada modificar el archivo “.env” con las credenciales que correspondan.



```
.env
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=admin_panel_amlc
14 DB_USERNAME=root
15 DB_PASSWORD=
16
```

Después situarse dentro de la carpeta del proyecto, luego lanzar los siguientes comandos:

```
>php artisan migrate
```

```
>php artisan db:seed
```

Instalación FRONTEND

Verificar el archivo Api.js que tenga el dominio deseado.

```
JS Api.js x
src > adminapp > apis > JS Api.js > BaseApi
1 import axios from "axios";
2
3 let BaseApi = axios.create({
4   baseUrl: "https://apiamlc.gnolab.com/" //server
5 });
```

Luego si modifico el dominio, se necesita la compilación del proyecto con el siguiente comando.

```
>quasar build
```

Una vez generado, copiar la carpeta **dist** a su directorio web ej: /var/www

Instalación RFID

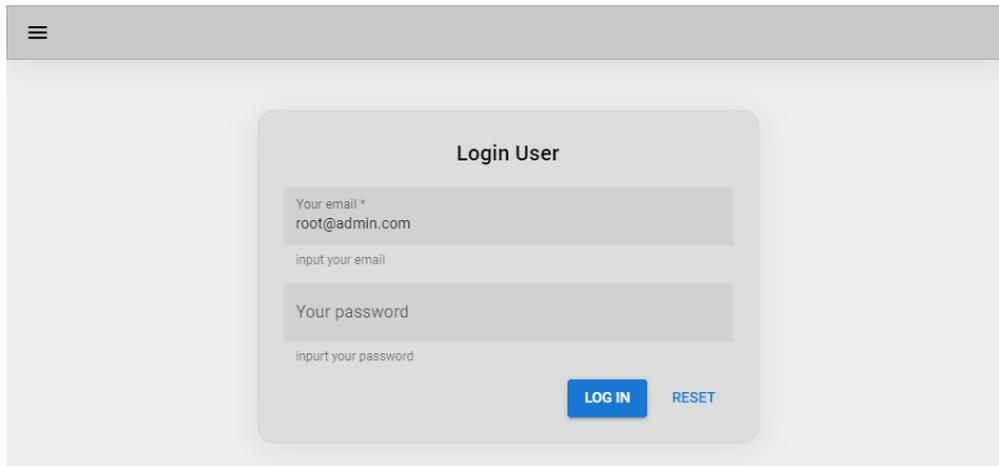
Para cambiar el dominio y la red, modificar el archivo main.cpp cambiando los datos deseados. Una vez modificado compilarlo y grabarlo al módulo ESP8266.

Ya grabado el programa al módulo, solo queda alimentarlo con 3v – 5v

```
// Credentials to connect to the wifi network
const char *ssid = "XXXXXXXXXX";
const char *password = "XXXXXXXXXX";
/*
The ip or server address. If you are on localhost, put your computer's IP (for example http://192.168.1.65)
If the server is online, put the server's domain for example https://parzibyte.me
*/
const String SERVER_ADDRESS = "http://192.168.100.5/admin-panel-amlc/public/api/esp";
void setup()
{
  // ...
}
```

MANUAL USUARIO

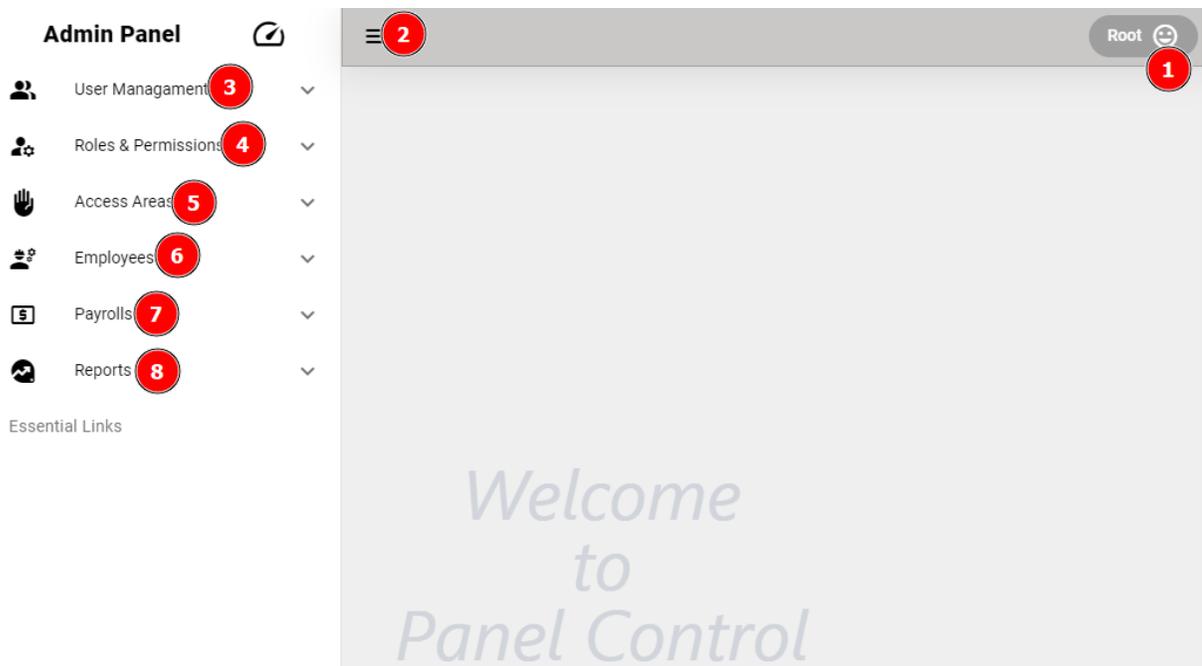
Login del Sistema



The screenshot shows a login form titled "Login User". It contains two input fields: "Your email *" with the value "root@admin.com" and "Your password". Below the password field, there is a "LOG IN" button and a "RESET" link.

1. ingrese el correo del usuario con lo que se registró en el sistema.
2. Ingrese la contraseña con lo que se registró en el sistema, para el empleado por defecto es su ci.
3. Ingresar al sistema, si los datos son correctos, caso contrario el sistema no le permitirá el ingreso al sistema.

Panel principal del administrador



The screenshot shows the Admin Panel interface. The left sidebar contains a menu with the following items: User Management (3), Roles & Permissions (4), Access Areas (5), Employees (6), Payrolls (7), and Reports (8). The main content area displays "Welcome to Panel Control". The top right corner shows "Root" with a user icon (1) and a hamburger menu icon (2).

1. Muestra al usuario logeado, también tiene un submenú para el logout.
2. Muestra o esconde a todo el menú.
3. Sub menú de gestión de usuarios.
4. Sub menú de gestión de roles y permisos, que se asigna a un usuario.
5. Sub menú de gestión de acceso de áreas, que posee un empleado.
6. Sub menú de gestión de empleados.
7. Sub menú de gestión de planillas y boletas de pago.
8. Sub menú de reporte gráfico.

Gestión de usuarios

The screenshot shows a user management interface. At the top right, the user 'Root' is logged in. The breadcrumb navigation shows 'Home > Users > Lists'. A green '+ New User' button is highlighted with a red circle '1'. Below the breadcrumb, there are filter tabs: 'All', 'root', 'admin', and 'employee', with 'employee' highlighted and circled in red '3'. A search box labeled 'Search Name' is circled in red '4'. The main table lists users with columns: Name, Email, Cellphone, Created/Updated, Roles, and Actions. The 'Root' user is highlighted, and its 'Actions' menu is open, showing 'Edit' and 'Delete' options, circled in red '2'. The 'Admin' user is also visible in the table.

Name	Email	Cellphone	Created/Updated	Roles	Actions
Root	root@admin.com	73252051	2022-11-18 Nov 18, 2022	root	⋮
Admin	admin@admin.com	555-112358	2022-11-18 Nov 18, 2022	admin	✎ Edit 🗑 Delete

1. Crea un nuevo usuario.
2. Edición y Borrado del usuario.
3. Filtrado de usuario por el rol que tiene.
4. Búsqueda del usuario por el nombre.

Creación de un nuevo usuario

Formulario de creación de usuario con los siguientes campos:

- Your full Name (input your full name)
- Your cellphone * (input your cellphone)
- Your email * (input your email) **1**
- Your password (input your password)
- Password Confirmation (input your password confirmation)

Sección "Assign Roles" **2**:

- root: Administrador del Sistema, Acceso Total
- admin: Administrador del Sistema
- employee: Solo puede ver sus Asistencias

Botón "Create User" verde.

1. Llenar todos los campos requeridos.
2. Asignación de uno o varios roles al usuario.

Gestión de roles

Muestra el listado de todos los roles creados.

Roles List

Total 1 users root Edit Role 1	Total 1 users admin Edit Role
Total 0 users employee Edit Role	 2 + Add Role Add role, if it doesn't exist.

1. Edición del rol.
2. Adición de un nuevo rol.

Adición de un nuevo rol

Create Role

name * **1**

Input name

description * **2**

Input description

Assign Permissions

<input type="checkbox"/> Navegar Usuarios	<input type="checkbox"/> Crear Usuarios	<input type="checkbox"/> Edición de Usuarios
<input type="checkbox"/> Eliminar usuario	<input type="checkbox"/> Asignar Roles a los Usuarios	<input type="checkbox"/> Navegar Roles
<input type="checkbox"/> Crear Roles	<input type="checkbox"/> Edición de Roles	<input type="checkbox"/> Eliminar Roles
<input type="checkbox"/> Navegar Permissions	<input type="checkbox"/> Crear Permissions	<input type="checkbox"/> Edición de Permissions
<input type="checkbox"/> Eliminar Permissions 3	<input type="checkbox"/> Navegar Employees	<input type="checkbox"/> Crear Employees
<input type="checkbox"/> Editar Employees	<input type="checkbox"/> Eliminar Employees	<input type="checkbox"/> Navegar Asistencias
<input type="checkbox"/> Navegar Contratos	<input type="checkbox"/> Crear Contratos	<input type="checkbox"/> Editar Contratos
<input type="checkbox"/> Navegar Planillas	<input type="checkbox"/> Regularizar las Asistencias	<input type="checkbox"/> Report Graphic statistic
<input type="checkbox"/> Navegar Acceso Areas		

[CANCEL](#) [+ Save Role](#)

1. Campo nombre del rol.
2. Campo de descripción del rol, debe especificar una explicación corta de que consiste ese rol.
3. Listado de permisos de todo el sistema, donde deberá seleccionar los permisos que desee.

Gestión de acceso de áreas

Root

Home > Areas > Lists **1** [+ New Area](#)

Search Name

Name	Created/Updated	Actions
Area Dpto. Administrativo	2022-11-18 2022-11-18	2 3
Area Dpto. Juridica	2022-11-18 2022-11-18	
Area Dpto. Sistemas	2022-11-18 2022-11-18	

Records per page: 5 1-3 of 3

1. Crea una nueva área de acceso.
2. Edita el área seleccionada.
3. Elimina el área seleccionada.

Gestión de empleados

Home > Employees > Lists Root

1 + New Employee

Employees

Search Name

Nombre	Apellidos	CI	Celular	Email	RFID Card	Creado/Modificado	Actions
Martha	Higadera Mendez	586854	7568684	mhigadera@gmail.com		2022-11-21 2022-11-21	2 ⋮

Records per page: 5 1-1 of 1

1. Crear un nuevo empleado.
2. Edición y eliminación de un empleado seleccionado.

Adición de un empleado

Create New Employee

1

Your Name: Jose
Your C.I.: 5965864

Your last name: Perez
Your last name, second: Lopez

Your email*: jperezlopez@gmail.com
Your cellphone: 69894854

Your Cua/Nua Number: 56434 **2**
Your Account Number: 100043453 **3**

Your address: C. Mejillones N.2234 Z. Alto Tejar, La Paz

1982-01-02 **4** M F

Assign RFID Card
the user has no RFID card assigned **5** ASSIGN RFID CARD ?

Assign Access to Areas **6**

Area Dpto. Administrativo
 Area Dpto. Juridica
 Area Dpto. Sistemas

Create Employee

1. Datos personales del empleado.
2. Número de afiliación de seguro AFP cua o nua según corresponda.
3. Número de cuenta bancaria.
4. Fecha de nacimiento.
5. Asignación de una Etiqueta tag de Rfid.
 - Dar clic botón Asignar Rfid Card.
 - Aparecerá el mensaje **Esperando para la lectura del Card.**
 - Acercar el Tag al lector.
 - Dar nuevo clic, de ahí se observará el mensaje **Card usable** cuando la tarjeta tag está disponible, caso contrario **Card no disponible.**
6. Asignar los accesos de área que desee

Gestión de contratos

Nro. Cite	CI	Nombre	Apellidos	Inicio	Conclusion	State	Creado/Modificado	Actions
2022/001	586854	Martha	Higadera Mendez	2022-11-01	2023-11-30	CURRENT	2022-11-21 2022-11-21	

1. Adicionar un contrato a un empleado.
2. Edición y eliminación de un contrato.

Adición de un contrato

Create Contract

Select Employee
Martha Higadera Mendez(586854) **1**

rrhh-cite *
2022/001 **2**

Input rrhh-cite

2022-11-01 **3**

input your start date

2023-11-30 **4**

input your end date

Select Contract Type
Eventual **5**

Select Charge
Abogado 6000 Bs. **6**

CANCEL **+ Save Role**

1. Seleccionar al empleado que desea asignar el empleado.
2. Llenar con el numero cite que corresponda al empleado.
3. Fecha de inicio de contrato.
4. Fecha de fin de contrato.
5. Seleccione el tipo de contrato.
6. Seleccione el cargo

Gestión de Asistencias

Home > Employees > Attendances 4 Export PDF

Employees

Search Name 3

Nombre	Apellidos	CI	Celular	Email
<input checked="" type="checkbox"/> 1 Esther	Carroll Joyce	2942732	+1 (312) 892-1815	ducy@mailinator.com
<input type="checkbox"/> Jessamine	Mendez Flores	99383454	+1 (382) 356-2901	hyhym@mailinator.com

1 record selected. Records per page: 2 1-2 of 6

TO DAY
BACK NEXT 2

October 2022

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
25	26	27	28	29	30	1
2	3 08:14:47 13:14:52 - 13:50:55 18:14:58	4 08:14:47 13:14:52 - 13:50:55 18:14:58	5 08:14:47 13:14:52 - 13:50:55 18:14:58	6 08:14:47 13:14:52 - 13:50:55 18:14:58	7 08:14:47 13:14:52 - 13:50:55 18:14:58	8
9	10 08:14:47 13:14:52 - 13:50:55 18:14:58	11 08:14:47 13:14:52 - 13:50:55 18:14:58	12 08:14:47 13:14:52 - 13:50:55 18:14:58	13 08:14:47 13:14:52 - 13:50:55 18:14:58	14 08:14:47 13:14:52 - 13:50:55 18:14:58	15
16	17 08:14:47 13:14:52 - 13:50:55 18:14:58	18 08:14:47 13:14:52 - 13:50:55 18:14:58	19 08:14:47 13:14:52 - 13:50:55 18:14:58	20 08:14:47 13:14:52 - 13:50:55 18:14:58	21 08:14:47 13:14:52 - 13:50:55 18:14:58	22
23	24 08:14:47 13:14:52 -	25 08:14:47 13:14:52 -	26 08:14:47 13:14:52 -	27 08:14:47 13:14:52 -	28 08:14:47 13:14:52 -	29

1. Selección del empleado para el listado de sus asistencias.
2. Muestra el calendario de las asistencias del empleado seleccionado, donde puede navegar por los meses del año.
3. Búsqueda del empleado por el campo nombre.
4. Exportar las asistencias del empleado seleccionado.

Regularizar asistencias

La regularización de asistencias se refiere a cualquier situación de no marcado por factores externos. Se deberá corregir las horas trabajadas de los empleados.

Home > Employees > Regularize Attendances Export PDF

Select Month: October 1 Select Year: 2022 2 Search

Regularize Attendances

Search Name

	Nombre	Apellidos	Month	Year	WorkDays	daysworked	hoursworked	Actions
<input checked="" type="checkbox"/>	Esther	Carroll Joyce	10	2022	30	21	168	4
<input type="checkbox"/>	Jessamine	Mendez Flores	10	2022	30	21	168	

1 record selected. Records per page: 2 1-2 of 4 < >

TO DAY

< BACK 3

NEXT >

October 2022

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
25	26	27	28	29	30	1
2	3 08:14:47 13:14:52 - 13:50:55 18:14:58	4 08:14:47 13:14:52 - 13:50:55 18:14:58	5 08:14:47 13:14:52 - 13:50:55 18:14:58	6 08:14:47 13:14:52 - 13:50:55 18:14:58	7 08:14:47 13:14:52 - 13:50:55 18:14:58	8
9	10 08:14:47 13:14:52 - 13:50:55 18:14:58	11 08:14:47 13:14:52 - 13:50:55 18:14:58	12 08:14:47 13:14:52 - 13:50:55 18:14:58	13 08:14:47 13:14:52 - 13:50:55 18:14:58	14 08:14:47 13:14:52 - 13:50:55 18:14:58	15
16	17 08:14:47 13:14:52 - 13:50:55	18 08:14:47 13:14:52 - 13:50:55	19 08:14:47 13:14:52 - 13:50:55	20 08:14:47 13:14:52 - 13:50:55	21 08:14:47 13:14:52 - 13:50:55	22

1. Seleccionar el mes y año.
2. Al dar clic, mostrará un mensaje preguntando si deseas cargar las asistencias de los empleados, al dar ok calculara de forma interna las horas y días trabajadas, si no existen asistencias le saldrá una notificación de la no existencia de asistencias.
3. Muestra el calendario de las asistencias del empleado seleccionado, donde puede navegar por los meses del año.
4. Modificación de los días trabajadas del empleado.

Esther Carroll Update

work days * 30	Days Worked * 21	Hours Worked * 168
input work days	input days worked	hours worked

CANCEL + Save Regularize

Regularize Attendances

Gestión de planillas

Para la generación de planillas es necesario las asistencias de los empleados, dicho esto antes de deberá ir al anterior paso de regularización de empleados, en ese modulo hace de forma interna los cálculos de días de trabajo.

Home > Payrolls > List 3 Export PDF

Select Month: October 1 Select Year: 2022 2 Search

Payrolls										Search Name
Name	Apellidos	Month	Year	seniority_bonus	total_earning	afp	discount	liquid_payable	Created/Updated	Actions
Esther	Carroll Joyce	10	2022	338	3488	503	503	2985	2022-11-18 2022-11-18	4 ☰
Jessamine	Mendez Flores	10	2022	338	5938	856	856	5081	2022-11-18 2022-11-18	☰
Angel	Flores Mendoza	10	2022	338	3271	472	472	2799	2022-11-18 2022-11-18	☰
Javier	Carrillo Lopez	10	2022	338	4538	654	654	3883	2022-11-18 2022-11-18	☰

Records per page: 5 1-4 of 4

1. Seleccionar el mes y año.
2. Al dar clic, mostrará un mensaje preguntando si deseas crear la planilla de los empleados, al dar ok aparecerá los empleados con sus respectivos cálculos, caso contrario de no existir asistencias aparecerá un mensaje de aviso de eso.
3. Es el reporte de la planilla en formato pdf.
4. Es la generación de la boleta de pago en formato pdf.