

UNIVERSIDAD POLITÉCNICA SALESIANA

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN INFORMÁTICA PARA LA GESTIÓN

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL DE ASISTENCIA DE PERSONAL MEDIANTE
DISPOSITIVOS BIOMÉTRICOS Y EMISIÓN DE ROL DE PAGOS PARA
LA EMPRESA “ALPROSART”**

AUTOR: FRANCISCO JAVIER ORELLANA MALDONADO

TUTORA: ING. DORIS MEZA

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Francisco Javier Orellana Maldonado bajo mi dirección.

Ing. Doris Verónica Meza Bolaños
DIRECTORA DE TESIS

DECLARACIÓN

Yo, Francisco Javier Orellana Maldonado, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Francisco Javier Orellana Maldonado

1720961885

AGRADECIMIENTO

Agradezco a Dios y mis padres, por ser mi guía y mi apoyo incondicional durante toda mi vida.

A mis suegros por ayudarme de diferentes formas a superar todos los obstáculos.

Y a todas las personas que han puesto su grano de arena para la finalización de este proyecto.

DEDICATORIA

Para mis queridos padres Anita y Hugo por haberme apoyado día a día con sus palabras de aliento.

A mi amada esposa Lizeth quien con sus pensamientos, energía positiva y oraciones a Dios supo fortalecerme a cada instante para no decaer.

Contenido

CAPÍTULO 1	1
INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	1
1.2 OBJETIVOS DEL PROYECTO	4
1.2.1 OBJETIVO GENERAL	4
1.2.2 OBJETIVOS ESPECÍFICOS.....	4
1.3 JUSTIFICACIÓN	5
1.4 ALCANCE	6
1.5 METODOLOGÍA.....	9
1.5.1 RUP.....	9
1.5.2 UML.....	12
1.5.3 PROGRAMACIÓN POR CAPAS.....	19
1.5.4 TÉCNICAS DE RECOPIACIÓN DE INFORMACIÓN	20
1.5.5 ESTÁNDAR IEEE 830	21
1.6 HERRAMIENTAS DE DESARROLLO	26
CAPÍTULO 2.....	28
IDENTIFICACIÓN BIOMÉTRICA	28
2.1 SISTEMAS DE IDENTIFICACIÓN BIOMÉTRICA	28
2.1.1 CONCEPTOS BÁSICOS	28
2.1.2 COMPONENTES DE LOS DISPOSITIVOS BIOMÉTRICOS	28
2.2. CÓMO FUNCIONAN LOS SISTEMAS BIOMÉTRICOS.....	28
2.3. TÉCNICAS BIOMÉTRICAS.....	29
2.4. ARQUITECTURA DE UN SISTEMA BIOMÉTRICO DE IDENTIFICACIÓN PERSONAL	34
2.5. TÉCNICAS DE OBTENCIÓN DE LA HUELLA DACTILAR POR MEDIO DE DISPOSITIVOS BIOMÉTRICOS	35
CAPÍTULO 3	44
ANÁLISIS Y DISEÑO	44

3.1	ANÁLISIS	44
3.1.1	REQUERIMIENTOS	44
3.1.2	CASOS DE USO.....	51
3.1.3	HERRAMIENTAS DE HARDWARE Y SOFTWARE EMPLEADAS PARA EL DESARROLLO DEL PROYECTO	56
3.2	DISEÑO	60
3.2.1	MODELO DE BASE DE DATOS	60
3.2.2	DICCIONARIO DE DATOS.....	63
3.2.3	DIAGRAMA DE SECUENCIA.....	66
3.2.4	PROTOTIPO DE INTERFACES	69
CAPÍTULO 4.....		73
DESARROLLO.....		73
4.1	PROCESO PARA GENERAR LA BASE DE DATOS “ALPROSART”	74
CAPÍTULO 5		97
PRUEBAS E IMPLEMENTACIÓN		97
5.1	PRUEBAS	97
5.2	IMPLEMENTACIÓN	100
CAPÍTULO 6.....		102
CONCLUSIONES Y RECOMENDACIONES		102
CAPÍTULO 7.....		106
ANEXOS.....		106

Índice de Figuras

<i>Figura 1.1: Productos elaborados por ALPROSART</i>	1
<i>Figura 1.2: Estructura jerárquica ALPROSART</i>	2
<i>Figura 1.3: Fases del RUP con sus entregables al final de cada fase</i>	12
<i>Figura 1.4: Elementos de un diagrama de casos de uso</i>	13
<i>Figura 1.5: Diagrama de casos de uso, utilizando Include</i>	14
<i>Figura 1.6: Diagrama de casos de uso, utilizando Extend</i>	14
<i>Figura 1.7: Ejemplo de diagrama de caso de uso, utilizando Límites del Sistema</i>	15
<i>Figura 1.8: Composición de una clase</i>	16
<i>Figura 1.9: Ejemplo de diagrama de secuencia para el cálculo de remuneración</i>	18
<i>Figura 1.10: Arquitectura en 3 capas</i>	20
<i>Figura 2.1: Arquitectura de un sistema biométrico, usando huellas digitales.</i>	35
<i>Figura 2.2: Características de la huella dactilar</i>	36
<i>Figura 2.3: Clasificación de las huellas dactilares, de izquierda a derecha (Arch, Left Loop, Right Loop, Whorl)</i>	37
<i>Figura 2.4: Clasificación de las minucias</i>	37
<i>Figura 2.5: identificación de las minucias en una huella dactilar</i>	38
<i>Figura 2.6: Representación de los puntos Core y Delta en una huella dactilar</i>	38
<i>Figura 2.7: Método de lectura de huella dactilar por sensores ópticos</i>	40
<i>Figura 2.8: Método de lectura de huella dactilar por sensores capacitivos</i>	41
<i>Figura 2.9: Proceso de reconocimiento de huella dactilar</i>	42
<i>Figura 3.1: Diagrama General de Casos de Uso</i>	53
<i>Figura 3.2: Diagrama de Caso de Uso: Administrar Empleados</i>	53
<i>Figura 3.3: Diagrama de Casos de Uso: Registrar Hora de Entrada-Salida</i>	55
<i>Figura 3.4: Dispositivo Biométrico Digital Persona U.areU 4500</i>	58
<i>Figura 3.5: Modelo Conceptual de Base de Datos</i>	61
<i>Figura 3.6: Modelo Físico de Base de Datos</i>	62
<i>Figura 3.7: Diagrama de secuencia de inicio de sesión</i>	67
<i>Figura 3.8: Diagrama de secuencia de gestión de la información</i>	68
<i>Figura 3.9: Diagrama de secuencia de registro de entrada-salida</i>	69
<i>Figura 3.10: Formulario de inicio de sesión</i>	70
<i>Figura 3.11: Pantalla Principal</i>	71
<i>Figura 3.12: Formulario de ingreso de nuevo registro</i>	71
<i>Figura 3.13: Formulario de actualizar-anular</i>	72
<i>Figura 4.1: Carpetas generadas para el desarrollo del proyecto</i>	73
<i>Figura 4.2: Tablas y funciones de la base de datos ALPROSART</i>	74
<i>Figura 4.3: Función anular de la base de datos</i>	77
<i>Figura 4.4: Función dividir_descuentos_adelantos de la base de datos</i>	79
<i>Figura 4.5: Función registrar_entrada</i>	80

<i>Figura 4.6: Función registrar-salida.....</i>	81
<i>Figura 4.7: Función calcular_valor_total_descuentos</i>	82
<i>Figura 4.8: Función calcular_valor_total_horas_extras</i>	83
<i>Figura 4.9: Estructura del archivo de configuración para acceso de base de datos..</i>	84
<i>Figura 4.10: Método obtenerConeccion() para establecer la conexión con la base de datos</i>	85
<i>Figura 4.11: Método insertar().....</i>	86
<i>Figura 4.12: Método insertarAuditoria</i>	87
<i>Figura 4.13: Método authenticate()</i>	89
<i>Figura 4.14: Método registrar().....</i>	91
<i>Figura 4.15: Método verificar()</i>	93
<i>Figura 4.16: Formulario Principal</i>	94
<i>Figura 4.17: Menú Turnos</i>	95
<i>Figura 4.18: Formulario de creación de usuarios</i>	95
<i>Figura 5.1: Diagrama de Implementación</i>	101

Índice de Tablas

<i>Tabla 1.1: Plantilla de contenido de ERS, según estándar IEEE 830 (Puntos 1 y 2)</i>	23
<i>Tabla 1.2: Cuadro comparativo de alternativas de herramientas de desarrollo</i>	26
<i>Tabla 2.1: Comparación de técnicas biométricas</i>	33
<i>Tabla 3.1: Caso de Uso Administrar Empleados</i>	54
<i>Tabla 3.2: Caso de Uso Registrar Hora de Entrada – Salida</i>	56
<i>Tabla 3.3: Listado de Tablas de Base de Datos</i>	64
<i>Tabla 3.4: Listado de columnas de la tabla “EMPLEADO”</i>	66
<i>Tabla 5.1: Caso de prueba de inicio de sesión</i>	97
<i>Tabla 5.2: Caso de prueba de reasignación de clave</i>	98
<i>Tabla 5.3: Caso de prueba de administración de empleados</i>	99
<i>Tabla 5.4: Caso de prueba de registro de huella dactilar</i>	99
<i>Tabla 5.5: Caso de prueba de registro de entrada/salida por dispositivo biométrico</i>	99
<i>Tabla 5.6: Caso de prueba de generación de rol de pagos</i>	100

RESUMEN

En el Capítulo 1 se define la problemática principal en donde es detallada la actividad de la empresa auspiciante, su estructura jerárquica por medio de un diagrama organizacional, así como el proceso de registro de hora de entrada, salida y el proceso de cálculo de rol de pagos.

También se definen los objetivos generales y específicos del proyecto, los mismos que ayudarán a mejorar los procesos mencionados anteriormente. El alcance y los limitantes son otros de los puntos más importantes de este capítulo.

En el Capítulo 2 se describe el proceso de identificación biométrica. Además de las diferentes técnicas que se emplean para realizar la identificación como son: rostro, termograma facial, huellas dactilares, geometría y venas de las manos, iris, patrón de la retina, firma y voz.

En el Capítulo 3 se determinan los requerimientos funcionales y no funcionales del sistema a desarrollar, mediante un análisis de las encuestas y cuestionarios realizados a las personas que laboran en la empresa auspiciante.

Posteriormente en el Capítulo 4 se describen las funciones más importantes dentro del desarrollo del proyecto, tanto funciones, de base de datos como funciones JAVA, también detalla la estructura de los formularios que son presentados al usuario.

Luego de terminado el desarrollo del proyecto en el Capítulo 5 se realizan las pruebas necesarias para comprobar el correcto funcionamiento y la puesta en marcha del software desarrollado en el proyecto.

Finalmente se presentan las conclusiones y recomendaciones al concluir todas las fases del proyecto en el Capítulo 6.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

ALPROSART (Alimentos Procesados Artesanalmente) es una micro empresa la cual se encarga de la producción y distribución de productos alimenticios procesados artesanalmente, específicamente snack's como hojuelas de papas, rizos de papas y chifles, a las dos cadenas más grandes de supermercados del Ecuador, La Favorita (Megamaxi) y Santa María, además de varios mini markets de la ciudad de Quito.



*Figura 1.1: Productos elaborados por ALPROSART
Fuente: El Autor*

ALPROSART la conforman un conjunto de veinte personas las cuales están distribuidas para las distintas áreas de trabajo de la siguiente manera:

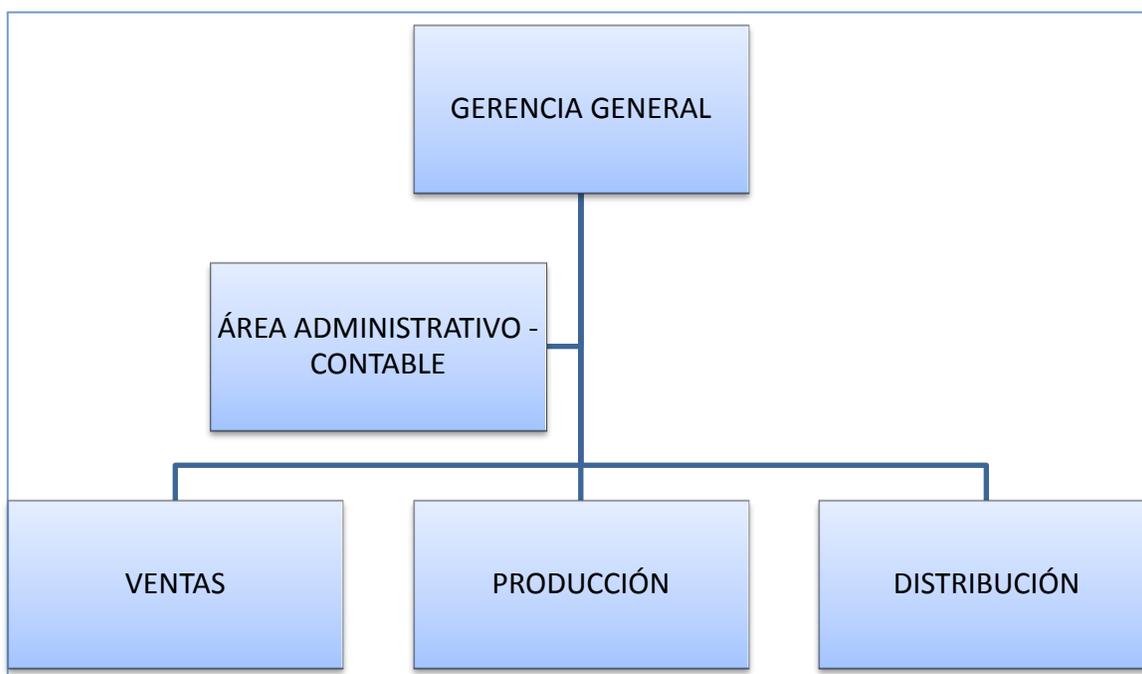
Gerencia General: una persona, encargada de las negociaciones con los diferentes proveedores.

Área Administrativo – Contable: una persona, encargada del pago de cuentas, roles de pago, control del personal, pago de servicios básicos.

Ventas: dos personas, encargadas de recibir los pedidos y comunicar al área de producción.

Producción: catorce personas, divididas en dos grupos los cuales van rotando cada semana en turnos de 7a.m. – 1p.m. y de 1p.m. – 7p.m. de lunes a domingo, los mismos que se encargan de elaborar los diferentes productos de la empresa. Al momento que el primer grupo ingresa a las instalaciones de la empresa empieza a correr el tiempo de trabajo, luego de transcurridas 6 horas laborables el segundo grupo debe estar listo para comenzar sus labores para evitar que la producción se detenga.

Distribución: dos personas, encargadas de distribuir el producto terminado desde la planta de producción a los clientes.



*Figura 1.2: Estructura jerárquica ALPROSART
Fuente: El Autor*

Actualmente en ALPROSART se realiza el control de la asistencia del personal de manera manual por parte de los propios empleados, es decir al momento de ingresar a la empresa el empleado anota en un cuaderno su número de cédula y su hora de

llegada, sin ninguna garantía que la hora registrada sea la hora real de llegada, sin llevar ningún tipo de registro de los permisos o atrasos que se puedan producir en el transcurso del mes, este proceso se repite para el registro de las horas de salida. Las horas extras son anotadas en un cuaderno por parte del administrador, para que un empleado labore horas extras debe comunicárselo al administrador para que éste al final del día verifique que en realidad el empleado estuvo laborando las horas que indicó y proceda a anotar los nombres y el número de horas extras que laboró. Cuando un empleado solicita laborar dos jornadas en el mismo día el administrador también registra esta novedad en su cuaderno para tenerlo en cuenta al momento de realizar el rol de pagos.

La forma en que se manejan las multas es muy similar al manejo de las horas extras debido a que es el administrador quien se encarga de controlar que los empleados no vayan en contra de las normas de la empresa. Cuando algún empleado comete una falta el administrador registra en su cuaderno esta novedad para tomarla en cuenta al momento de realizar el rol de pagos, por cada infracción que tenga un empleado se le descuenta la cantidad de \$5 USD del valor total de su salario.

Los empleados pueden tener bonificaciones a su sueldo por tareas adicionales que son solicitadas por el administrador o el gerente, las cuales después de cumplidas, el administrador se encarga de registrar en su cuaderno de apuntes para agregar un valor adicional al salario del empleado, dependiendo de la tarea encomendada.

La empresa realiza adelantos a los empleados cuando lo solicitan y en la mayoría de los casos el valor del adelanto es diferido en varios meses sin ningún tipo de interés, el administrador es el encargado de anotar el valor de la multa y en cuántos meses se difiere para realizar el descuento respectivo al rol de pago del empleado.

La problemática surge al momento de realizar los roles de pagos, debido a que no se sabe la cantidad exacta de dinero que se debe pagar a cada uno de los empleados ya que en el cuaderno donde se registran las horas de entradas y salidas existen muchas irregularidades e inconsistencias, además que en la mayoría de casos los empleados olvidan realizar este registro, también se da el caso de empleados que

recibieron un adelanto de su salario, pero al momento de recibir su rol de pagos no se ha realizado el descuento respectivo. Otro de los problemas identificados se da al momento del despido o renuncia de un empleado, debido a que cuando se tiene que realizar su liquidación en base a todos los salarios, no se sabe a ciencia cierta cuánto se tiene que pagar al mismo ya que no hay un registro completo de todas las remuneraciones que el empleado ha recibido.

La falta de un control confiable de asistencia provoca una gran demora en la obtención de informes y reportes necesarios para verificar que el personal está asistiendo regularmente a laborar, esto también conlleva a una pérdida de tiempo mayor al momento de realizar los roles de pago.

El descuido por parte de la persona encargada de realizar el pago de las obligaciones patronales y/o servicios básicos es otro de los problemas que se ha identificado en la empresa, lo que conlleva a que el patrón caiga en mora por falta de pago y por consiguiente pérdidas económicas significativas.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 OBJETIVO GENERAL

El objetivo general de la presente investigación es implementar un sistema informático para ALPROSART el cual permita optimizar y automatizar el control de la asistencia del personal de la empresa y la generación automática de roles de pago, por medio de dispositivos biométricos, ayudando a reducir las irregularidades en la asistencia del personal y las demoras al momento de realizar los roles de pago mes a mes. Además enviar alertas vía SMS y email cada vez que esté cerca el pago de una obligación patronal a la persona encargada de estos trámites para que se cumplan a tiempo.

1.2.2 OBJETIVOS ESPECÍFICOS

- i. Evitar la suplantación de identidad al momento de registrar la hora de entrada y salida del personal por medio de un dispositivo biométrico el cual identifica la huella digital que es única en cada persona.

- ii. Optimizar el tiempo y la exactitud de la información al momento de emitir roles de pagos, recibiendo toda esta información de una base de datos centralizada.
- iii. Reducir errores al momento de generar roles de pago.
- iv. Reducir las irregularidades en la asistencia por parte del personal, teniendo un registro exacto de los días y horas laborados.
- v. Minimizar las pérdidas económicas a la empresa al generar un rol de pagos, con el valor exacto a pagar.
- vi. Generar reportes con el historial de asistencia de cada empleado de una manera rápida y confiable.
- vii. Evitar multas por motivos de mora patronal alertando a la persona encargada antes que se cumpla el plazo de pago de las obligaciones patronales.
- viii. Evaluar la responsabilidad con respecto al cumplimiento del horario de entrada y salida de cada empleado, mediante métricas establecidas en el sistema.
- ix. Comprobar la evolución del personal mes a mes mediante reportes gráficos comparativos.

1.3 JUSTIFICACIÓN

La propuesta del sistema automatizado de control de asistencia del personal de ALPROSART surge a partir de las irregularidades presentadas al momento de registrar la asistencia del personal y de las dificultades que trae esto cuando se emiten los roles de pago. Los registros de entrada y salida son vulnerables a la modificación por parte de los propios empleados ya que son ellos los que realizan su registro de forma manual en un cuaderno, exponiéndose así a registrar una hora de entrada o de salida que no sea la real.

Por este motivo se propone realizar el control por medio de un dispositivo biométrico el cual va a reconocer la huella digital de cada persona, este tipo de tecnología es lo más usado actualmente, una de las razones por las que está en auge es que se garantiza que nadie podrá realizar el registro suplantando a otra persona porque la huella digital es única en cada uno de los trabajadores. Reduciendo la intervención

del recurso humano en la manipulación de esta información tan importante para la empresa.

Además, al tener un registro exacto de las horas laboradas de cada trabajador se facilita la generación de los roles de pago, evitando así que los empleados exijan que se les pague por horas no laboradas, también se registrará en el sistema cuando un empleado realice un adelanto, para tomarlo en cuenta al momento de generar su rol de pagos.

Al tener un registro digital de las horas de entrada y salida se evita la pérdida de datos ya que actualmente suelen desaparecer las hojas en donde se registra toda esta información, asegurando que los datos se encuentren íntegros en un solo sistema el cual solo sea manipulado por una persona de confianza.

Con todo esto se conseguirá que los empleados lleguen puntuales a sus estaciones de trabajo, se reducirá el tiempo con el que se emiten los roles de pago y se reducirán las pérdidas económicas por valores que ya han sido cancelados previamente a los empleados.

Además, con un sistema de alertas con fechas totalmente configurables se puede avisar a la persona encargada de los servicios patronales que tiene algún pago que está por llegar a su fecha límite de pago, previniendo inconvenientes por la falta de pago de los mismos.

1.4 ALCANCE

El objetivo principal del proyecto es desarrollar un sistema el cual agilite el manejo de la información de los roles de pago de los empleados de ALPROSART, a través de la obtención de los datos de la asistencia del personal registrados por medio de un dispositivo biométrico, dentro del sistema se encontrarán los siguientes módulos:

1.4.1 MÓDULO DE AUTENTICACIÓN

La autenticación al sistema se gestionará mediante un usuario y una contraseña generados por el administrador con ayuda de información proporcionada por el

empleado. Luego de que la autenticación sea exitosa se presentarán los módulos a los cuales tenga acceso el usuario al sistema, dependiendo su rol.

1.4.2 MÓDULO DE ADMINISTRACIÓN

Gestiona la información de los empleados, datos personales, el valor de su remuneración básica. Además de sus huellas digitales mediante un dispositivo biométrico, se registrarán las huellas de cada uno de los dedos de la mano derecha como de la mano izquierda, por cada empleado, para prevenir algún inconveniente en el caso de que el empleado no pueda usar alguno de sus dedos.

Además en este módulo se gestionarán las multas, adelantos, bonificaciones, vacaciones de los empleados, y demás información que afecte al cálculo del rol de pago de los empleados.

En este módulo se realizará la configuración de las fechas en las cuales se deban realizar los diferentes pagos patronales agregando una descripción de que es lo que se debe pagar.

1.4.3 MÓDULO DE AUDITORÍA

En el módulo de auditoría se visualizarán todas las acciones (modificaciones, consultas y eliminación de registros) que han realizado los usuarios en el sistema, para llevar un control sobre las mismas y en caso de detectar alguna anomalía, identificar quien y cuando lo hizo. La información presentada en este módulo no podrá ser modificada ni eliminada de manera alguna.

1.4.4 MÓDULO DE REPORTES

Este módulo contará con los siguientes reportes: atrasos y remuneraciones recibidas, por un empleado o por todos los empleados en un período de tiempo.

1.4.5 MÓDULO DE ROLES DE PAGO

El módulo de roles de pago se encargará de realizar el cálculo correspondiente a la remuneración que debe recibir cada uno de los empleados mes a mes tomando en cuenta las horas laboradas, atrasos, descuentos, horas extras, adelantos, bonificaciones y multas.

1.4.6 MÓDULO DE REGISTRO

En este módulo se realizará el registro de las horas de entrada y salida de los empleados desde el dispositivo biométrico, y también mostrará los mensajes personalizados para cada empleado ingresados desde el módulo de administración.

1.4.7 LIMITANTES

- El sistema se limitará al reconocimiento de huellas digitales no de retinas, manos o alguna otra parte del cuerpo.
- No se generará reportes que tengan que ver con la contabilidad de la empresa.
- El sistema solo generará roles de pago, no se realizará el pago de cheques ni ningún tipo de transferencia bancaria hacia las cuentas de los empleados.
- Las alertas que se envíen serán solamente a la persona registrada en el sistema como la encargada de realizar los pagos de las obligaciones patronales.
- No se generarán liquidaciones, pero será de ayuda para el proceso de liquidación, por medio del reporte de remuneraciones que ha recibido el empleado.

Los roles que manejará el sistema son los siguientes:

- Rol administrativo: tendrá acceso a los módulos de: administración, auditoría, autenticación, reportes, generación de roles de pago, y registro.
- Rol general: en este rol sólo se tendrá acceso al módulo de registro, y administración de datos personales básicos de la persona autenticada en el sistema.

El sistema contará con otro dispositivo biométrico para el reconocimiento de huellas de repuesto en caso de que se dañe el dispositivo principal, además habrá la posibilidad de registrar las horas de entrada y salida por medio de la cédula de identidad en el caso de que fallaran los 2 dispositivos.

1.5 METODOLOGÍA

1.5.1 RUP

RUP (*Rational Unified Process – Proceso Unificado Racional*) consiste en “metodologías adaptables al contexto y necesidades de cada organización¹” y “constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.²”

El objetivo de usar RUP es asegurar que el *software* a producirse sea de la mejor calidad y que satisfaga las necesidades del usuario, a partir de los recursos y presupuesto establecidos.

1.5.1.1 FASES DEL RUP

En los siguientes puntos se detallan las fases del RUP, y los entregables que se obtendrán para el presente proyecto luego de cada fase.

❖ FASE DE INICIO

En la fase de inicio se definen los objetivos del proyecto, el ámbito en el cual se va a desarrollar, las funcionalidades, las restricciones, estimaciones de costos y tiempos, con el objetivo de tener una visión clara de qué es lo que se va a realizar. Se establece un modelo inicial de casos de uso. En esta fase, el proyecto puede ser cancelado o replanteado si se considera que la propuesta inicial no es óptima.

Los entregables que se obtendrán luego de esta fase son los siguientes:

- Documento con la visión del proyecto.
- Caso de negocio inicial el cual incluye: contexto del negocio.
- Modelo de casos de uso, con una lista inicial de todos los casos de uso y los actores que puedan ser identificados.
- Lista de las funcionalidades y restricciones principales del sistema a desarrollar.
- Estándares para el prototipo inicial.

¹ Fuente de información: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

² Fuente de información: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

- Lista inicial de los riesgos y su forma de evaluación.
- Lista de requisitos funcionales y no funcionales.
- Prototipo inicial.

❖ **FASE DE ELABORACIÓN**

Se realiza un análisis de la funcionalidad del proyecto y del dominio del problema, con la finalidad de realizar una arquitectura básica, se complementa el modelo de casos de uso. En esta fase también se debe planificar las actividades necesarias y los recursos requeridos, especificando las características. El objetivo de esta etapa es determinar la mejor arquitectura. Se eliminan los elementos de mayor riesgo para el proyecto. Al final de esta fase se deben tener claro todos los casos de uso y los actores debidamente identificados.

La lista de entregables a obtener luego de la fase de elaboración son los siguientes:

- Generar una lista revisada de riesgos.
- Realizar la arquitectura del software.
- Revisar los requerimientos complementarios.
- Construir un prototipo inicial de la interfaz del usuario.
- Actualizar el plan del proyecto

❖ **FASE DE CONSTRUCCIÓN**

En la fase de construcción, se desarrollan todos los componentes y características del producto a través de iteraciones, una iteración es “un ciclo completo de desarrollo que resulta en una versión o *release* (interno o externo) de un producto ejecutable, un subconjunto del producto final que se encuentra bajo desarrollo y que crece incrementalmente en cada iteración hasta llegar al producto final³”, en cada iteración se debe realizar el proceso de análisis, diseño e implementación. Se va refinando la arquitectura del producto conforme se va construyendo. La documentación de la construcción del sistema como del uso del mismo es uno de los pasos más importantes en esta fase. Al final de esta fase se debe contar con:

³ Fuente de información: <http://www.conexionit.com/blog/metodologias/que-es-rup.html>

- Una lista revisada de riesgos.
- Administración de los recursos (herramientas, base de datos).
- Desarrollo completo de los componentes (prototipo funcional).
- Probar los componentes contra los criterios de evaluación definidos en la fase inicial.
- Actualizar el plan de proyecto

Aquí se puede determinar si la versión actual del producto de software es lo suficientemente madura y estable como para ser entregada al usuario final.

❖ **FASE DE TRANSICIÓN**

Esta es la última fase y consiste en el paso del producto de software a un ambiente de producción. Cuando el sistema ya sea entregado al usuario final, las tareas que se deben realizar son las siguientes: capacitación a los usuarios finales, corrección de errores, soporte, lanzamiento de versiones mejoradas.

Al final de esta fase se determina si los objetivos fueron alcanzados o no, muchas veces esto se puede medir de acuerdo al grado de satisfacción del usuario y/o una comparativa de los gastos reales versus los planeados.

Los entregables en la fase de transición son:

- Evaluación del usuario.
- Ajustes realizados al sistema.

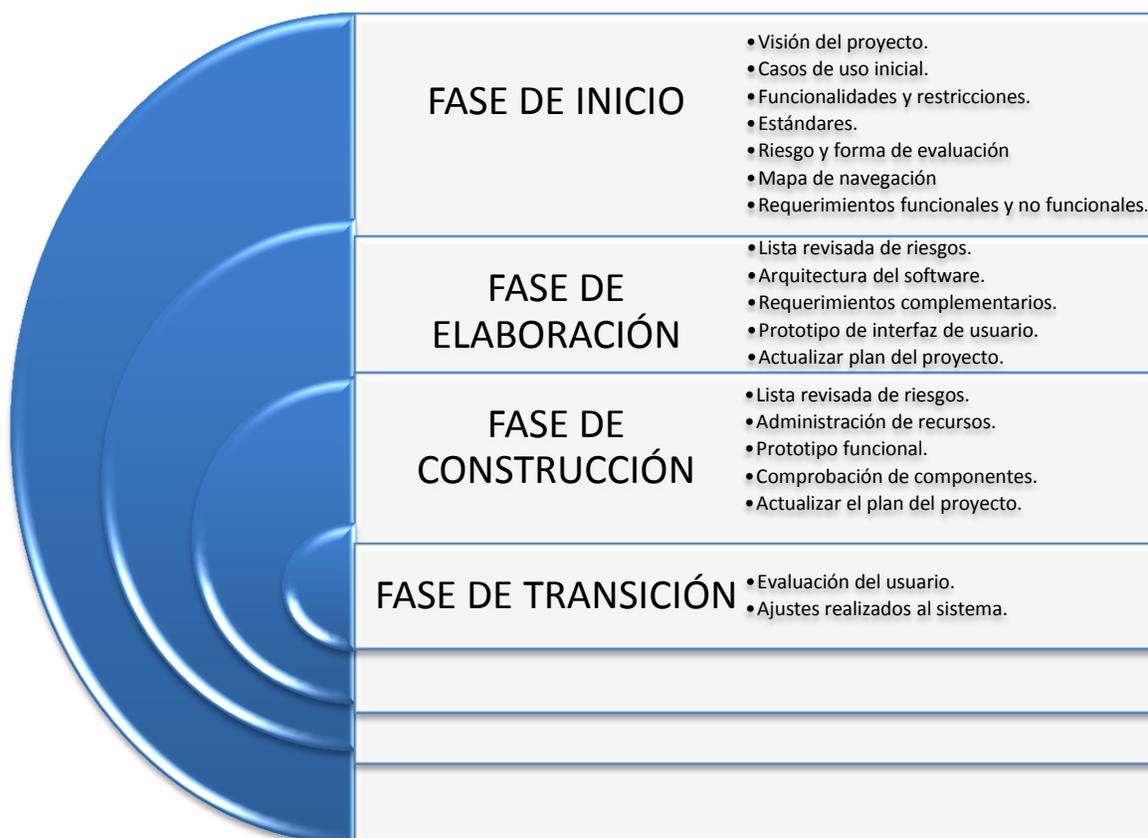


Figura 1.3: Fases del RUP con sus entregables al final de cada fase
Fuente: El Autor

1.5.2 UML

UML (*Unified Modeling Language* – Lenguaje de Modelado Unificado) es un lenguaje de propósito general para el modelado orientado a objetos, combina las notaciones que provienen de: (i) Modelado orientado a objetos, (ii) Modelado de datos, (iii) Modelado de componentes, (iv) Modelado de flujos de trabajo.

Cuando se habla de ingeniería se habla también de modelos, estos modelos son simplificaciones de la realidad, y ayudan a comprender de mejor manera que es lo que se va a desarrollar. Al momento de realizar un modelo, hay que centrarse en los detalles relevantes, por esta razón con un solo modelo no es suficiente al momento de representar un sistema.

1.5.2.1 DIAGRAMAS UML

A un sistema hay que verlo y comprenderlo desde varios frentes, es por esta razón que en UML recomienda el uso de distintos diagramas los cuales representan distintas vistas de un mismo sistema. Se presentará una descripción de cada uno de los diagramas UML a ser aplicados al proyecto.

❖ DIAGRAMA DE CASOS DE USO

Los diagramas de casos de uso describen el comportamiento del sistema al afrontar una tarea de negocio, desde el punto de vista del usuario. Ayudan a determinar los requerimientos funcionales del sistema, en otras palabras representan las funciones que el sistema puede ejecutar.

Entre los elementos básicos de un diagrama de casos de uso se tiene: (i) Actores, que representan a cualquier cosa que interactúe con el sistema ya sea una persona, otro sistema, una empresa. (ii) Caso de uso, es la representación de una tarea que debe llevarse a cabo con el apoyo del sistema que se está desarrollando. (iii) Asociaciones, se presentan cuando un actor interactúa con el sistema para realizar un caso de uso.



Figura 1.4: Elementos de un diagrama de casos de uso
Fuente: El Autor

En la Figura 1.4 se muestra la representación de un caso de uso en donde el actor (Administrador) interactúa con el caso de uso (Generar rol de pagos).

Existen dos tipos de asociaciones: (i) Include: se usa para especificar comportamiento común en dos o más casos de uso. (ii) Extend: esta relación se usa

para especificar que el comportamiento de un caso de uso es diferente dependiendo de algún factor.

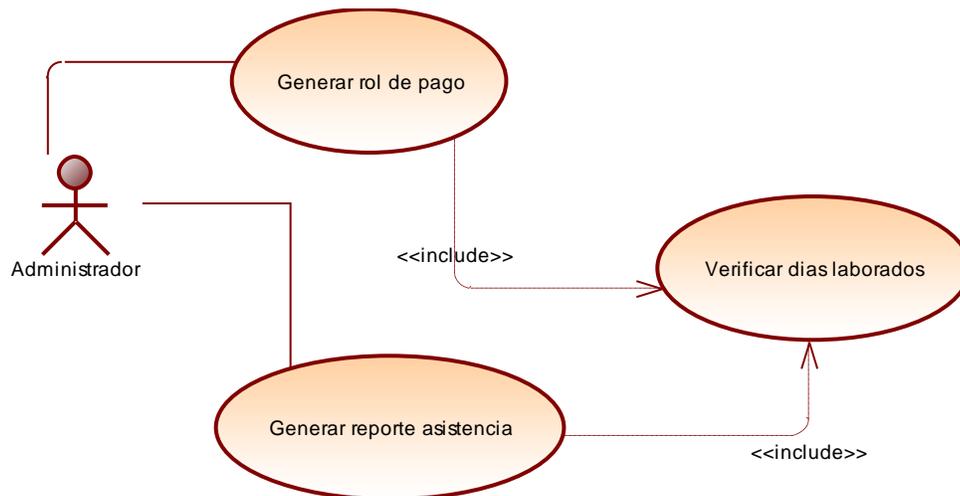


Figura 1.5: Diagrama de casos de uso, utilizando Include
Fuente: El Autor

En la Figura 1.5 se puede visualizar el uso de la asociación tipo *Include*, ya que tanto el caso de uso “Generar rol de pago” y “Generar reporte asistencia” realizan algo en común “Verificar días laborados”.

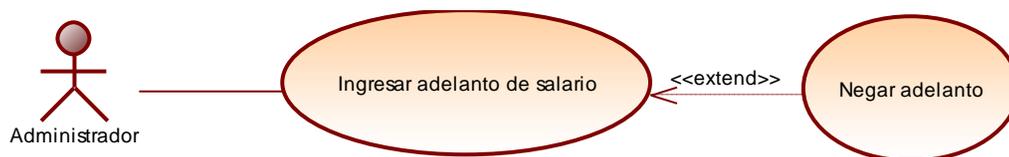
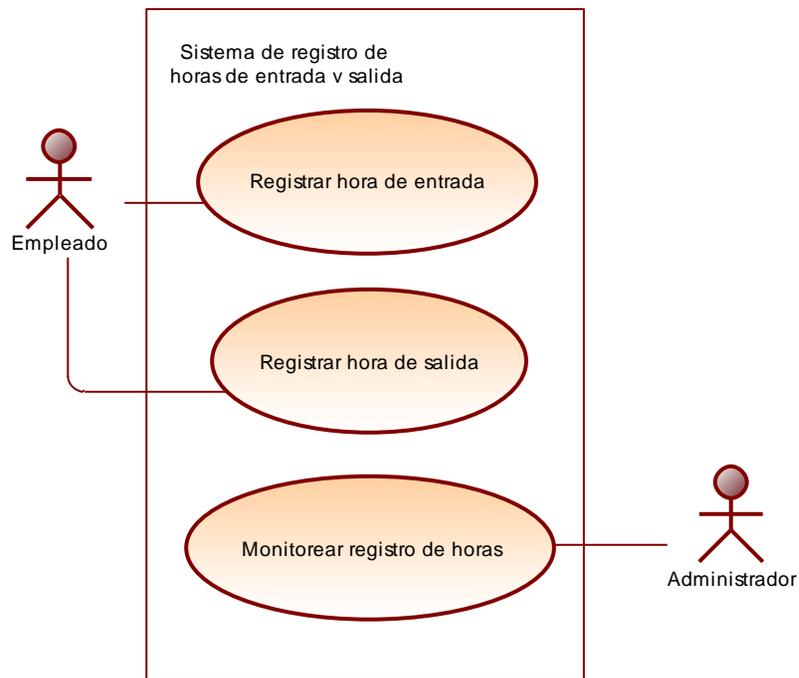


Figura 1.6: Diagrama de casos de uso, utilizando Extend
Fuente: El Autor

En la Figura 1.6 se tiene una representación de cómo se usa la asociación *Extend*, donde la flecha de asociación debe apuntar siempre al caso de uso original, en este caso “Ingresar adelanto de salario”.

Cuando se realizan los diagramas de caso de uso resulta útil dibujar los límites del sistema, cuando se pretende hacer un diagrama de casos de uso para parte del sistema, tal como se muestra en la Figura 1.7.



*Figura 1.7: Ejemplo de diagrama de caso de uso, utilizando Límites del Sistema
Fuente: El Autor*

❖ **DIAGRAMA DE CLASES**

En este tipo de diagrama se representan las clases a ser utilizadas en el sistema y las relaciones que existen entre sí. “Una clase es simplemente una representación de un tipo de objeto; piense en ella como un plano que describe el objeto. Así como un plano puede utilizarse para construir varios edificios, una clase puede utilizarse para crear varias copias de un objeto”⁴.

⁴ Fuente de información: [http://msdn.microsoft.com/es-es/library/ms172601\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/ms172601(v=vs.80).aspx)

En UML cada clase se representa por un rectángulo con tres divisiones. La primera división corresponde al nombre de la clase, la segunda a los atributos y la tercera a las operaciones, en la Figura 1.8 se distingue las 3 divisiones de una clase.

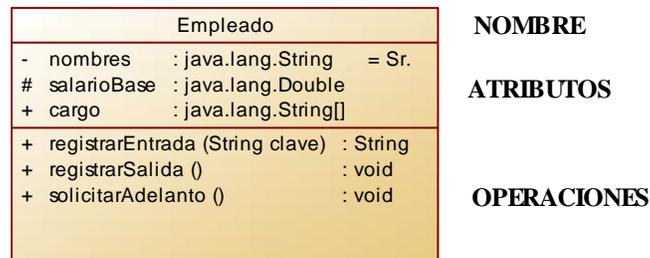


Figura 1.8: Composición de una clase
Fuente: El Autor

El nombre de la clase debe abarcar todo el conjunto de atributos y operaciones.

Los atributos de las clases poseen características las cuales ayudan a la manipulación y control de los mismos. Estas características son:

- Tipo: el tipo depende del lenguaje de programación que se vaya a utilizar. Por ejemplo: *int*, *String*, *Double*, *char*. Se lo coloca a continuación del nombre del atributo precedido de ":". En la Figura 1.8 se visualiza que el atributo "nombres" es de tipo *java.lang.String*.
- Valor Inicial: corresponde al valor que tendrá un atributo al momento de ser creado. En la figura 1.8 se visualiza que el valor inicial del atributo "nombres" es "Sr".
- Visibilidad: se refiere al nivel de acceso que tienen los atributos, pueden ser de tipo: (i) Público, es representado por el símbolo "+" e indica que el atributo puede ser accedido o modificado desde otras clases, (ii) Privado, es representado por el símbolo "-" e indica que no puede accesible desde otra clase, (iii) Protegido, es representado por el símbolo "#" e indica que sólo se

puede acceder por las clases que se derivan de la original. Se lo coloca antes del nombre del atributo, tal como se muestra en la *Figura 1.8*.

- Multiplicidad: indica si el atributo debe tener un valor único (1), si el atributo puede tener valor o no (0..1), si el atributo puede tener varios valores o ninguno (0..*), si el atributo puede tener varios valores (1..*), pero debe tener al menos uno (*), si el atributo puede tener varios valores ó si el atributo puede tener entre N y M valores (M..N). En UML se lo representa con el símbolo “[]”, luego del tipo de dato y en el interior se coloca el valor de la multiplicidad. Ejemplo: cargo:String [0..1]
- Modificadores: un atributo puede tener: (i) Modificadores de clase, cuando no es necesario un objeto para ejecutarlo, (ii) modificadores derivados, cuando es ejecutable a partir de otros atributos, (iii) Modificador transitorio, cuando tendrá valor sólo durante un tiempo determinado de ejecución, (iv) Modificador volátil, cuando no persiste.

Las operaciones son un servicio que una clase puede realizar, son también conocidos como “métodos”. Reciben o no parámetros y pueden enviar o no un parámetro de salida. Al igual que los atributos poseen “Visibilidad” y “Modificadores”. En la *Figura 1.8*, se puede observar que la operación “registrarEntrada” es de tipo pública (+), recibe un parámetro de entrada con nombre “clave” de tipo “String” y el resultado de esta operación es un parámetro de salida tipo “String”.

Existen 3 relaciones diferentes entre las clases: (i) Dependencias, es una relación de uso, es decir que una clase usa a la otra, (ii) Generalización, también conocida como herencia, en donde una o varias clases hija heredan todos los atributos y métodos de su padre, (iii) Asociación, cuando los elementos de una clase están relacionados con los elementos de otra.

❖ **DIAGRAMA DE SECUENCIA**

En un diagrama de secuencia se muestra una iteración ordenada mediante una secuencia temporal de eventos. Para la representación de un diagrama de secuencia en UML, el eje vertical representa el tiempo, y en el eje horizontal se colocan los

objetos y actores participantes en la interacción, sin un orden en particular. Cada objeto o actor tiene una línea vertical (línea de vida) en la parte superior de la línea de vida se coloca un rectángulo en donde se coloca el nombre del objeto, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o junto a las transiciones o activaciones a las que se refieren.

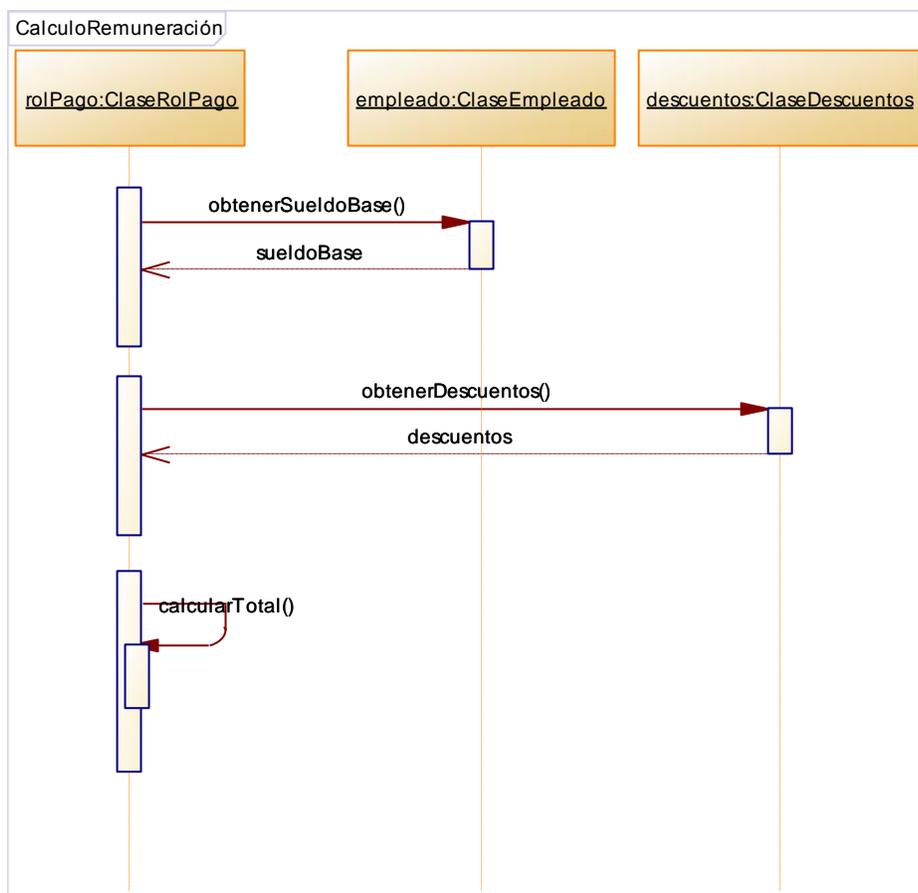


Figura 1.9: Ejemplo de diagrama de secuencia para el cálculo de remuneración
Fuente: El Autor

1.5.3 PROGRAMACIÓN POR CAPAS

La programación por capas es un estilo de programación donde el objetivo principal es separar a la lógica del negocio de la lógica de diseño, logrando con esto un desarrollo de sistemas en varios niveles agrupados según su funcionalidad, logrando así que al momento de realizar un cambio no se tenga que modificar todo el código, sino sólo al nivel requerido. El diseño que se implementa comúnmente es el de tres niveles (Arquitectura de 3 capas).

❖ CAPA DE PRESENTACIÓN

Esta capa es la que visualiza el usuario final, es la forma de presentación del sistema, captura y envía información al usuario, aquí se debe realizar un filtro previo para evitar que se envíen datos incorrectos al sistema por medio de validaciones. La capa de presentación se comunica directamente con la capa de negocio.

❖ CAPA DE NEGOCIO

En la capa de negocio es donde se ejecutan todos los procesos por medio de las peticiones del usuario, enviando una respuesta al mismo usuario luego de haber terminado el proceso. Aquí es donde se establecen todas las reglas que deben cumplirse, es por eso que también es denominada lógica del negocio. Esta capa se comunica con la capa de presentación para mostrar los resultados de los procesos invocados y con la capa de datos para almacenar o recuperar información necesaria para completar los procesos.

❖ CAPA DE DATOS

En la capa de datos es donde residen todos los datos del sistema, se forma por uno o varios gestores de bases de datos, los cuales se encargan de todas las transacciones para el almacenamiento o recuperación de información requeridas por la capa de negocio.

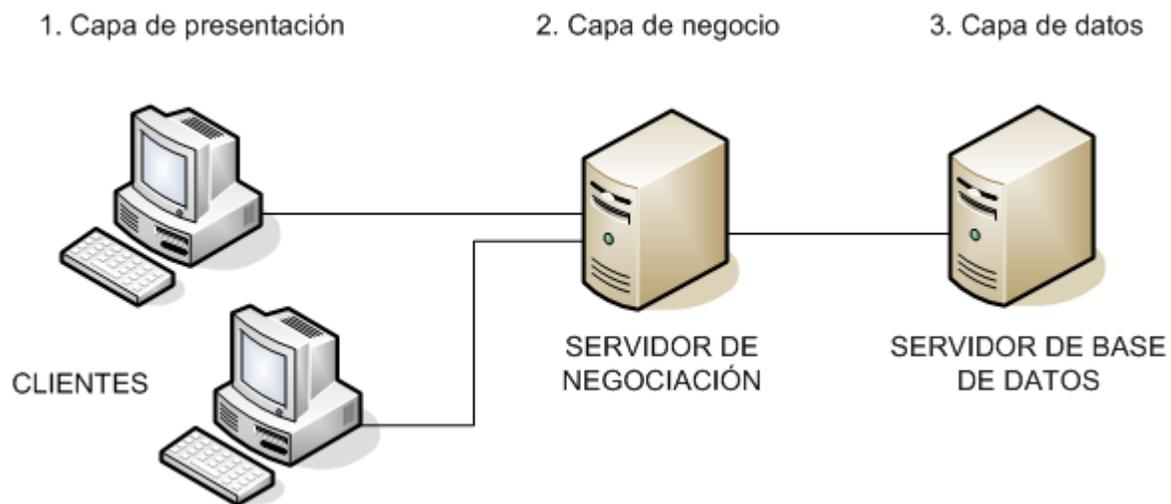


Figura 1.10: Arquitectura en 3 capas
Fuente: http://es.wikipedia.org/wiki/Archivo:Tres_capas.PNG

Las capas mencionadas anteriormente pueden estar en un mismo servidor, aunque no es lo más recomendable. Si el sistema crece y demanda más recursos, la capa de datos y negocio pueden estar ubicadas en un servidor separado de la capa de presentación. Lo más común en la arquitectura de 3 capas es que cada una de las capas esté en uno o más servidores independientes.

1.5.4 TÉCNICAS DE RECOPIACIÓN DE INFORMACIÓN

Para la recopilación de información se usará un método directo el cual consiste en la recopilación de los datos directamente de la fuente de información por parte del investigador. A fin de garantizar la obtención de todos aquellos datos de importancia para la resolución efectiva del problema.

❖ OBSERVACIÓN

Es la técnica con la cual se dirige a la fuente primaria de información y se ocupan de percibir todo el ambiente y los procesos concernientes al problema, con la finalidad de tener una visión clara del entorno del problema y de las fallas existentes y percibir las posibles soluciones que pueden tener en el mismo punto.

❖ ENTREVISTA

En esta técnica se entabla una conversación entre los investigadores y las personas que están en contacto con la fuente primaria de información, a fin de tratar de obtener datos relevantes acerca de la situación, sus características y posibles soluciones. En la mayoría de los casos es mucho más fácil programar una cita con un gerente de alto nivel que pedirle que llene un cuestionario. Al principio se recomienda una entrevista abierta, con preguntas y respuestas libres para conocer las ideas y necesidades de la persona que responde. En segunda instancia se debe realizar una entrevista estructurada con una serie de preguntas estandarizadas para obtener datos específicos sobre la investigación que se está desarrollando.

Para una entrevista exitosa la actitud del entrevistador representa un punto clave, el tacto, la imparcialidad, e incluso la manera de vestir ayudan a obtener una entrevista exitosa.

1.5.5 ESTÁNDAR IEEE 830

El presente proyecto tendrá todos los elementos mencionados en el estándar IEEE 830, pese a esto no se basará en el formato presentado en el mismo.

En la siguiente tabla se muestra una plantilla de cómo debe realizarse la especificación de requerimientos de software bajo el estándar IEEE 830.

FICHA DEL DOCUMENTO		
CONTENIDO		
1	INTRODUCCIÓN	Se brindará una descripción general de todo el documento de Especificación de Requerimientos de Software (ERS).
1.1	Propósito	Se especificará el propósito de la ERS y hacia quien va dirigido.
1.2	Ámbito del sistema	Se dará un nombre al futuro sistema, determinará el alcance, los limitantes,

	objetivos que se esperan alcanzar con el futuro sistema.
1.3 Definiciones, acrónimos y abreviaturas	Aquí se definirán todos los términos, definiciones, acrónimos y abreviaturas a usados en la ERS.
1.4 Referencias	Se mostrará una lista de todos los documentos a los cuales se hace referencia en la ERS.
1.5 Visión general del documento	Se describen de manera general el contenido y la organización del documento de ERS.
2 DESCRIPCIÓN GENERAL	Se describen los factores que afectarán al futuro sistema y sus requisitos, ayudando a definir con detalle los requerimientos en la siguiente sección.
2.1 Perspectiva del producto	Aquí se debe relacionar al futuro sistema con otros sistemas, en el caso de que sea totalmente independiente también se debe especificar en esta sección.
2.2 Funcionalidad del producto	En esta sección se describe de una manera general las funciones del futuro sistema, de una forma organizada.
2.3 Características de los usuarios	Se describen las características de los usuarios del futuro sistema, como por ejemplo: nivel de educación, experiencia, etc.
2.4 Restricciones	En esta sección se describen los limitantes que se imponen sobre los desarrolladores del futuro sistema. Políticas de la empresa, protocolos de

	comunicación, seguridades, lenguaje de programación, etc. Son unos de los limitantes que se pueden encontrar.
2.5 Suposiciones y dependencias	Especifica los factores que pueden afectar a los requisitos del nuevo sistema en el caso de que cambien, en caso de que esto suceda se debe cambiar los requerimientos de igual manera.
2.6 Evolución previsible del sistema	Se describen las mejoras en el sistema que se podrán realizar en el futuro.

Tabla 1.1: Plantilla de contenido de ERS, según estándar IEEE 830 (Puntos 1 y 2)

Fuente: El Autor

3 REQUERIMIENTOS ESPECÍFICOS

Esta es la sección más importante de la ERS, donde se detalla minuciosamente todos los requerimientos del futuro sistema, necesarios para que los desarrolladores puedan realizar su trabajo. Todo requisito aquí especificado describirá los comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Este documento deberá ser legible para cualquier persona que lo revise. Se deberá hacer mención a los documentos que tengan relevancia sobre los requerimientos. Los requerimientos deben ser identificados correctamente por medio de la asignación de un código, ó de una numeración adecuada.

Las características que deben poseer los requerimientos son:

- Correcto, todo requerimiento de la ERS debe reflejar una necesidad real, si se presenta una corrección en la ERS significa que el futuro sistema será el deseado.
- No ambiguo, cada requerimiento solo tiene una interpretación, de ser posible se debe usar gráficos explicativos, y en caso de que se usen términos que tengan más de un significado, se definirán estos términos con precisión en el

glosario. Expresiones como a veces, bien, adecuado, etc. introducen ambigüedad en los requerimientos.

- Completos, los requerimientos relevantes como irrelevantes deben ser incluidos en la ERS, hay que incluir la respuesta del sistema ante datos válidos e inválidos.
- Consistentes, ningún requisito debe contradecir a otro.
- Clasificados, no todos los requerimientos tienen el mismo nivel de importancia, se los debe clasificar por esenciales, condicionales u opcionales. Esto ayuda a implementar los requerimientos esenciales con mayor prioridad.
- Verificables, un requerimiento es verificable si existe un proceso finito y de bajo costo para verificar que el sistema cumple con el requerimiento.
- Modificable, una ERS es modificable si sus requerimientos están estructurados de una forma que se pueda realizar un cambio de manera fácil, consistente y completa.
- Explorable, un requerimiento es explorable si se conoce su origen, facilita la referencia de los componentes del diseño hacia los requerimientos.

3.1 Requerimientos comunes de las interfaces

Se describen los requerimientos que afectan a la interfaz de usuario, interfaces con otros sistemas, así como interfaces de comunicaciones.

3.2 Funciones

Se especifican todas las funciones que deberá realizar el sistema, agrupadas por:

- Tipo de usuario, cada grupo de usuarios tienen diferentes requerimientos.
- Objetos, los objetos son entidades del mundo real reflejadas en el sistema, para cada objeto debe detallarse sus atributos y funciones.
- Objetivos, se define como objetivo a un servicio que ofrece el sistema, para cada objetivo se debe especificar las funciones que sean necesarias para llevarlo a cabo.

- Estímulos, se detallan los estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.

3.3 Requerimientos no funcionales

Los requerimientos no funcionales especifican propiedades del sistema que pueden usarse para juzgar la operación del mismo.

3.3.1 Requerimientos de rendimiento

Se detallan los requerimientos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, también se especifican los requerimientos que puedan afectar el ingreso de información hacia la base de datos.

3.3.2 Atributos del Sistema

Se detallan los atributos de calidad del sistema: fiabilidad, mantenibilidad, seguridad, portabilidad.

3.3.3 Otros Requerimientos

Van detallados todos los requerimientos que no encajen en las secciones anteriores.

3.4 Restricciones de diseño

Indica todo aquello que restrinja las decisiones relativas al diseño, entre ellos, hardware, otro tipo de software, etc.

4 APÉNDICES

Se detalla todo tipo de información relevante a la ERS, pero que no forme parte de la misma, entre ellos: (i) formato de entrada y salida de datos, por pantalla y en datos, (ii) Resultados de análisis de costos, (iii) Restricciones sobre el lenguaje de programación.

1.6 HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo se han seleccionado a partir del análisis del siguiente cuadro comparativo (*Tabla 1.2*), considerando las posibilidades económicas y de infraestructura de la empresa auspiciante (ALPROSART).

Herramienta / Características	Licencia	Código Abierto	Plataforma	Documentación	Experiencia
Codificación					
Microsoft Visual Studio 2010	Pago	No	Windows	Completa	Media
Eclipse IDE 3.7.1	Libre	Si	Windows/Linux	Completa	Alta
Netbeans IDE 7.1	Libre	Si	Windows/Linux	Completa	Media
Diseño					
Agro UML 1.4	Libre	Si	Windows/Linux	Parcial	Baja
Power Designer 15	Pago	Si	Windows	Parcial	Media
Jdeveloper 11g	Libre	Si	Windows/Linux	Completa	Baja
Reportes					
Crystal Reports para Eclipse	Libre	No	Windows/Linux	Parcial	Media
Birth 2.6.2	Libre	Si	Windows/Linux	Parcial	Baja
JasperReports 1.0.7	Libre	Si	Windows/Linux	Parcial	Media
Base de Datos					
MySQL 5.5.20 GA	Libre	Si	Windows/Linux	Completa	Media
PostgreSQL 9.1.2	Libre	Si	Windows/Linux	Completa	Media
SQL Server 2010	Pago	No	Windows	Parcial	Alta
Servidor de Aplicaciones					
Apache	Libre	Si	Windows/Linux	Completa	Baja
JBOSS	Libre	Si	Windows/Linux	Completa	Alta
Framework					
IceFaces	Libre	Si	Windows/Linux	Parcial	Ninguna
Seam	Libre	Si	Windows/Linux	Completa	Alta

*Tabla 1.2: Cuadro comparativo de alternativas de herramientas de desarrollo
Fuente: El Autor*

Luego del análisis de la Tabla 1.2 se han seleccionado las siguientes herramientas. Estas podrán variar de versión dependiendo los requerimientos que serán especificados en los siguientes capítulos:

Diseño: Jdeveloper 11g

Codificación: Eclipse IDE 3.7.1, bajo el lenguaje de programación JAVA⁵.

Reportes: JasperReports 1.0.7

Base de datos: PostgreSQL 9.1.2

Servidor de aplicaciones: JBOSS

Framework⁶: Seam⁷

⁵ JAVA: Lenguaje de programación orientado a objetos.

⁶ Framework: Estructura conceptual y tecnológica que provee de una metodología de trabajo.

⁷ Seam: Framework desarrollado por JBOSS, utilizado para aplicaciones *Web*.

CAPÍTULO 2

IDENTIFICACIÓN BIOMÉTRICA

2.1 SISTEMAS DE IDENTIFICACIÓN BIOMÉTRICA

2.1.1 CONCEPTOS BÁSICOS

Una de las actividades que han sido automatizadas gracias a la tecnología es la capacidad de establecer la identidad de las personas a través de la biometría.

“La biometría es el estudio de métodos automáticos para el reconocimiento único de humanos basado en uno o más rasgos conductuales o físicos intrínsecos. El término se deriva de las palabras griegas bios (vida) y metrón (medida).⁸”

2.1.2 COMPONENTES DE LOS DISPOSITIVOS BIOMÉTRICOS

Los dispositivos biométricos poseen tres componentes en general; por un lado, disponen de un mecanismo automático que lee y captura una imagen digital o analógica de la característica a analizar. El segundo es un mecanismo encargado de manejar aspectos como la compresión, procesamiento, almacenamiento y comparación de los datos capturados frente a los datos almacenados en una base de datos, y el tercer componente es una interfaz que puede estar ubicada en el mismo o en otro sistema.

2.2. CÓMO FUNCIONAN LOS SISTEMAS BIOMÉTRICOS

En el caso de los sistemas basados en la identificación de la huella dactilar, el usuario inicialmente coloca su dedo sobre un sensor el cual realiza la lectura digital de su huella, luego el software guardará esta información como un modelo; la siguiente vez que el usuario trate de acceder al sistema deberá repetir el proceso de colocar el dedo sobre el sensor y el software se encargará de verificar que la huella leída por el sensor corresponda al modelo guardado anteriormente.

⁸ Fuente de información: <http://es.wikipedia.org/wiki/Biometría>

Este mismo principio rige para la identificación iris/retina, el rostro, la mano, etc. Las tasas de exactitud en la verificación dependen en gran medida a dos factores: el cambio que se puede producir en el cuerpo de las personas, debido a accidentes o al envejecimiento y a las condiciones ambientales, como por ejemplo: la humedad en el aire, suciedad y sudor; este último factor afecta en mayor medida a los sistemas basados en el reconocimiento de las huellas dactilares y la mano completa.

Para la selección de la parte del cuerpo más adecuada para la identificación biométrica se debería enfocar en las partes del cuerpo que son menos susceptibles a cambios y tenga una característica física robusta, que sea lo más distintiva posible en relación con el resto de la población, que sea una zona accesible, disponible y que sea aceptable por el usuario.

Finalmente se debe hacer una diferenciación entre los dispositivos que miden el comportamiento y los que miden una característica fisiológica. Los dispositivos que miden el comportamiento son los que se basan en el análisis dinámico de características no físicas como por ejemplo: la firma, requieren mayor cooperación por parte del usuario; y los que miden una característica fisiológica incluyen la huella dactilar, la geometría de la mano y el dedo, la termografía facial y la exploración del iris o la retina. El reconocimiento de la voz es un parámetro biométrico basado en ambos análisis, el fisiológico que determina la zona vocal y el de comportamiento analiza las palabras y el lenguaje usado.

2.3. TÉCNICAS BIOMÉTRICAS

En la actualidad existen sistemas biométricos que se basan en el reconocimiento de diversas características. Entre las técnicas más conocidas se encuentran:

2.3.1 ROSTRO

Las imágenes faciales son probablemente la característica biométrica más usada por las personas para la identificación personal debido a su gran aceptación por parte de los usuarios. Los sistemas basados en esta característica deben detectar si

en realidad existe un rostro en la imagen tomada por la cámara, localizar la imagen en la base de datos y reconocer la imagen desde un punto de vista general.

2.3.2 TERMOGRAMA FACIAL

El sistema vascular subyacente del rostro humano produce una única firma facial cuando el calor es emitido por la piel, estas firmas faciales pueden ser capturadas por una cámara infrarroja la cual es denominada termograma facial. Este sistema no es vulnerable ni a cirugías plásticas ni a máscaras ya que ninguna de estas alteraciones varía el flujo de la sangre a través de las venas, además son independientes a la luz del medio ambiente y no necesita mayor cooperación del usuario. Al termograma facial no se lo puede considerar como una característica biométrica permanente.

2.3.3 HUELLAS DACTILARES

La huella dactilar es el patrón de las crestas y valles sobre la superficie de la misma, formada por la acumulación de células muertas cornificadas que constantemente se mudan como escamas de la superficie expuesta. Esta formación es determinada en el período fetal y no varía a lo largo de toda la vida de la persona. Un problema que se presenta con la tecnología relacionada con las huellas dactilares es la aceptación por un usuario no familiarizado con estos sistemas ya que las huellas dactilares han sido usadas para investigaciones criminalísticas y trabajo policial hace mucho tiempo atrás, debido a esto este grupo de usuarios no se sienten cómodos al usar sus huellas para la identificación.

2.3.4 GEOMETRÍA DE LA MANO

La gran variedad de geometrías de manos incluyendo la forma, el largo, el ancho de los dedos, etc. puede ser usado como una característica biométrica. Este tipo de técnica es muy simple y de bajo costo y cuenta con un grado de aceptación por parte del usuario razonable. El problema que presentan los sistemas basados en el reconocimiento de la geometría de la mano, ocurre cuando se presenta una gran cantidad de modelos almacenados en la base de datos lo que dificulta la exactitud al momento de la identificación. Además del problema mencionado anteriormente se

puede mencionar que hoy en día es fácil falsificar la forma geométrica de la mano. Esta forma de identificación presenta una variante la cual consiste en la geometría de los dedos de la mano por medio de un reconocimiento en 3 dimensiones, aún se encuentra en desarrollo pero ya se ha determinado que tiene mayor exactitud que la identificación por la geometría de la mano.

2.3.5 VENAS DE LAS MANOS

Las venas de las manos proveen de un patrón robusto y único que se usa actualmente como característica biométrica para la identificación personal. Las imágenes digitalizadas de las venas pueden ser tomadas de manera fácil por una cámara infrarroja, en este caso el medio ambiente no interfiere al momento de tomar el modelo o de realizar el reconocimiento.

El patrón de las venas es prácticamente imposible de modificarlo incluso con cirugía.

El grado de aceptación por parte de los usuarios es muy alto y el grado de exactitud es muy alto, pero no se recomienda usarlo para grupos de usuarios muy grandes o que vayan a hacer uso del mismo durante muchos años ya que este patrón cambia de acuerdo a la edad de la persona.

2.3.6 IRIS

La formación de la textura del iris depende de las condiciones iniciales del mesodermo embrionario⁹ en donde se desarrolla, el iris es único para cada ser humano y nunca cambia a lo largo de su vida; además está aislado del medio ambiente y no puede ser modificado por ningún método. La técnica que usan los sistemas para el reconocimiento del iris es simple pero con gran exactitud.

Los usuarios no presentan una gran aceptación ante esta forma de identificación ya que el receptor de la imagen necesita proyectar un rayo de luz sobre el iris lo que no los hace sentir cómodos al realizar este proceso repetidamente debido a que las personas son muy cuidadosas con sus ojos.

⁹ Mesodermo embrionario: Tejido que cubre el nervio óptico.

Sin lugar a dudas este sistema es uno de los más efectivos pero también uno de los más costosos.

2.3.7 PATRÓN DE LA RETINA

El patrón de la retina, también conocido como venas de la retina posee características únicas para cada persona, las cuales son utilizadas para la identificación biométrica. Para tomar una imagen digital del patrón de la retina se dispara un rayo de luz de baja intensidad sobre el globo ocular. Al igual que el reconocimiento del iris el patrón de la retina es uno de los más confiables al momento de la identificación ya que no es posible su modificación por ningún método.

Sus inconvenientes son la poca aceptación de los usuarios y su alto costo en el mercado.

2.3.8 FIRMA

Cada persona tiene un único estilo de escritura, a la firma se la puede considerar como una especie de “huella digital” lo que hace que sea tomada en cuenta para la identificación de cada persona.

Existen dos formas de verificación de la firma: (i) dinámica y (ii) estática. La verificación estática se basa en las formas geométricas de la firma. La verificación dinámica hace uso de la verificación estática además del reconocimiento de la aceleración de la escritura, y la trayectoria de los trazos de la firma.

Es muy aceptada por el usuario e imposible de falsificar al momento de una verificación dinámica. Y los sistemas que usan esta característica para identificar a las personas deben poseer una gran precisión.

2.3.9 VOZ

Las características vocales de los humanos son determinadas por el tracto vocal, la boca, las cavidades nasales, y otros procesos del mecanismo humano al momento de hablar.

Estas características son únicas para cada ser humano y son llamadas comúnmente huellas de voz. Pueden ser verificadas de dos formas: (i) texto-dependientes y (ii) texto-independientes. La verificación texto-dependiente autentica al usuario basado en un texto predeterminado. La verificación texto-independiente autentica al usuario independientemente de la frase que pronuncie, este método de verificación tiene un mayor grado de complejidad.

No posee un gran porcentaje de exactitud ya que son sensibles a varios factores como por ejemplo el estado de ánimo y de salud del usuario lo que podría hacer variar a la huella de voz.

BIOMÉTRICOS	USO UNIVERSAL	UNICIDAD	PERMANENCIA	RENDIMIENTO	ACEPTACIÓN	SEGURIDAD
ROSTRO	Alta	Baja	Media	Baja	Alta	Baja
HUELLA DACTILAR	Media	Alta	Alta	Alta	Media	Alta
GEOMETRÍA DE LA MANO	Media	Media	Media	Media	Media	Media
VENAS DE LA MANO	Media	Media	Media	Media	Media	Alta
IRIS	Alta	Alta	Alta	Alta	Baja	Alta
PATRÓN DE LA RETINA	Alta	Alta	Media	Alta	Baja	Alta
FIRMA	Baja	Baja	Baja	Baja	Alta	Baja
VOZ	Media	Baja	Baja	Baja	Alta	Baja
TERMOGRAMA FACIAL	Alta	Alta	Baja	Media	Alta	Alta

Tabla 2.1: Comparación de técnicas biométricas

Fuente: El Autor

Una vez analizada la *Tabla 2.1* se ha decidido utilizar la técnica de la huella digital para el desarrollo del presente proyecto debido a su alta aceptación a nivel mundial y su alto nivel de seguridad, sin dejar a lado su bajo costo de implementación.

2.4. ARQUITECTURA DE UN SISTEMA BIOMÉTRICO DE IDENTIFICACIÓN PERSONAL

La arquitectura común de un sistema biométrico de identificación personal consta de dos módulos: (i) Módulo de inscripción y (ii) Módulo de identificación.

2.4.1 MÓDULO DE INSCRIPCIÓN

El módulo de inscripción se encarga de adquirir datos relativos al indicador biométrico elegido y entregar una presentación digital de los mismos. Luego de esto los datos adquiridos serán almacenados en una base de datos central o un dispositivo magnético, a estos datos comúnmente se los denomina *template*.

2.4.2 MÓDULO DE IDENTIFICACIÓN

El módulo de identificación es el encargado del reconocimiento de los usuarios, este proceso comienza cuando el lector biométrico captura la característica del usuario a ser identificado y la convierte en formato digital, para que seguidamente el extractor de características produzca una representación compacta con el mismo formato de los *templates*. La representación resultante se denomina *query* y es enviada al comparador de características el cual confronta a la *query* con uno o varios *templates* para obtener la identidad del usuario.

El módulo de identificación puede funcionar de dos modos: (i) modo de Identificación, (ii) modo de autenticación.

2.4.2.1 MODO DE IDENTIFICACIÓN

En el modo de identificación la *query* obtenida es comparada una a una con todos los *templates* de la base de datos hasta encontrar alguna coincidencia en caso de hallarse la coincidencia el usuario será identificado.

2.4.2.2 MODO DE AUTENTICACIÓN

En el modo de autenticación se compara la *query* obtenida solamente con el o los *templates* que correspondan a ese usuario, en este tipo de identificación se debe proveer adicionalmente algo que el usuario sepa por ejemplo, su número de cédula. De esta forma se verifica que el usuario es realmente quien dice ser. En el modo de

autenticación se aumenta notablemente la velocidad de respuesta del sistema debido a que no tiene que verificar toda la base de datos como en el modo de identificación.

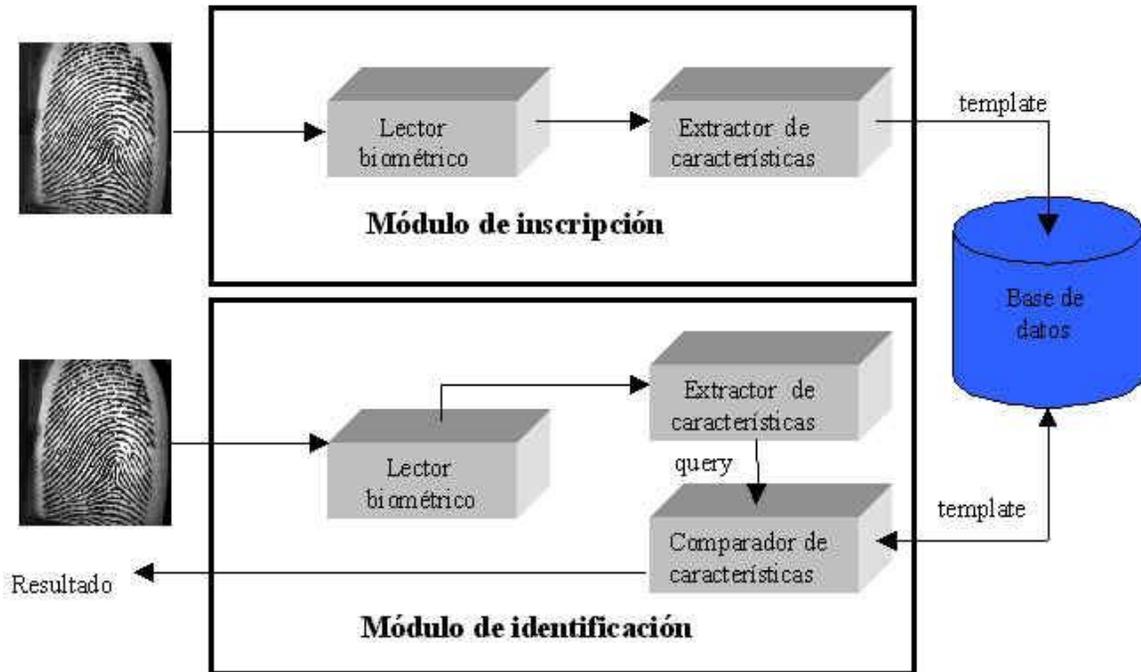


Figura 2.11: Arquitectura de un sistema biométrico, usando huellas digitales.
Fuente: http://www2.ing.puc.cl/~iing/ed429/sistemas_biometricos.htm

2.5. TÉCNICAS DE OBTENCIÓN DE LA HUELLA DACTILAR POR MEDIO DE DISPOSITIVOS BIOMÉTRICOS

2.5.1 INTRODUCCIÓN

Para poder entender de una mejor manera cómo se obtienen las características de una huella digital por medio de un dispositivo biométrico, se debe tener conocimiento de cómo se forman y en donde se ubican cada una de estas características.

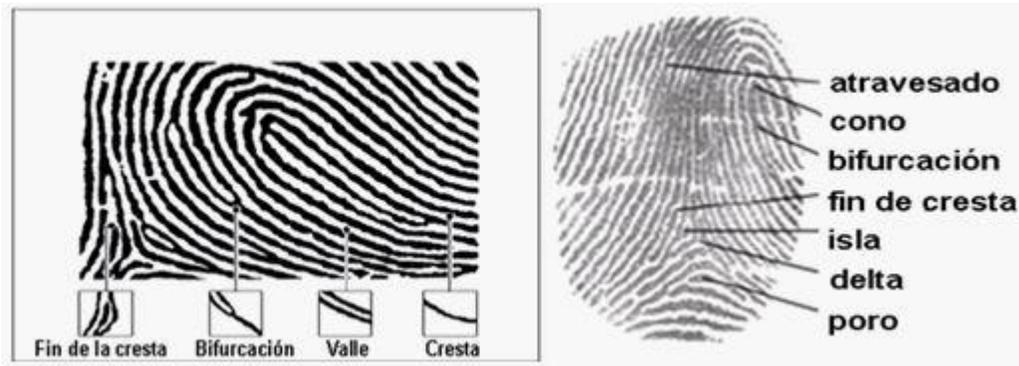


Figura 2.12: Características de la huella dactilar

Fuente: <http://www.monografias.com/trabajos57/huellas-lofoscopicas/huellas-lofoscopicas3.shtml>

A simple vista se puede identificar que la piel del ser humano no es del todo uniforme, posee ciertas rugosidades las mismas que son únicas para cada persona, en la huella dactilar se pueden identificar las siguientes rugosidades:

- **Papilas:** son rugosidades que nacen en la dermis y terminan en la epidermis, tienen diversas formas (cónicas, piramidales, hemisféricas). Se ha estimado por medio de estudios que el número de papilas por milímetro cuadrado es de 36 con una altura de 55 a 225 milésimas de milímetro.
- **Crestas:** las crestas están formadas por un conjunto de papilas formando bordes sobresalientes en la piel, estos bordes siguen las curvas de los surcos y forman un sinnúmero de formas en la yema de los dedos. Estas formas son denominadas dactilograma (en el caso de provenir de los dedos de la mano).
- **Surcos:** los surcos son los espacios que se forman entre papila y papila, es debido a los surcos que se pueden apreciar espacios en blanco al momento de imprimir una huella dactilar.
- **Poros:** los poros son pequeños orificios que situados en la punta de las crestas o cerca de su vértice, tienen la función de segregar sudor, poseen diversas formas (triangulares, circulares, ovoidales, etc.).

De acuerdo a la forma que posea el dactilograma las huellas dactilares se clasifican en:

- Right Loop: vuelta a la derecha (R)
- Left Loop: vuelta a la izquierda (L)
- Whorl: espiral (W)
- Arch: arco (A)

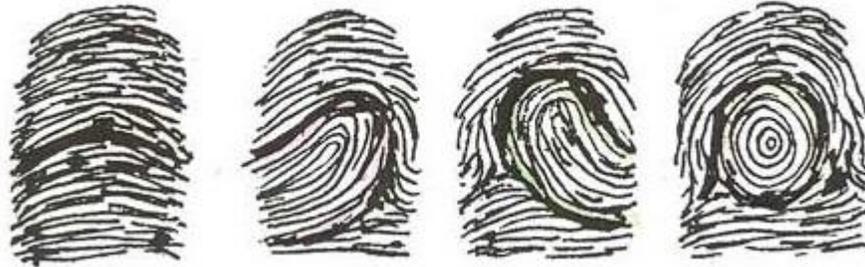


Figura 2.13: Clasificación de las huellas dactilares, de izquierda a derecha (Arch, Left Loop, Right Loop, Whorl)

Fuente: <http://hebesabiasque.blogspot.es/>

Además de las características presentadas anteriormente se debe mencionar otras complementarias las cuales son muy importantes en el proceso de la identificación biométrica:

- Minucias: las minucias son pequeñas discontinuidades formadas por el cruce y terminación de las crestas. En la huella dactilar se encuentran más de 50 minucias, las cuales han sido clasificadas en 7 tipos.

Características	
	Terminación
	Bifurcación
	Laguna
	Borde independiente
	Punto o isla
	Aguijón
	Cruce

Figura 2.14: Clasificación de las minucias

Fuente: kime25.tripod.com

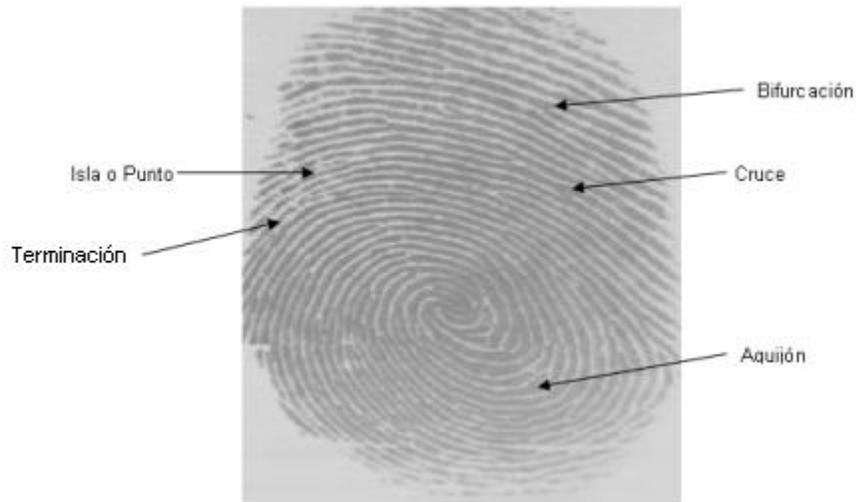


Figura 2.15: identificación de las minucias en una huella dactilar
Fuente: Modificación del Autor de: biometría.gov.ar

- Punto Core: es el punto de máxima curvatura de la cresta. Este punto es generalmente usado como punto de referencia para la codificación de minucias.
- Punto Delta: es el centro de un conjunto de crestas inferiores al Punto Core que dibuja varios triángulos concéntricos.



Figura 2.16: Representación de los puntos Core y Delta en una huella dactilar
Fuente: <http://gjorge.wordpress.com/2007/10/05/la-huella-dactilar/>

2.5.2 MÉTODOS PARA OBTENER LOS PATRONES DE LA HUELLA DACTILAR POR MEDIO DE LECTORES DIGITALES

Los lectores de la huella dactilar tienen dos tareas principales:

- Obtener una imagen digitalizada de la huella dactilar.
- Comparar el patrón de los valles y crestas de la imagen de la huella obtenida, con el patrón de las huellas almacenadas.

Existen dos métodos principales que realizan estas tareas: (i) por medio de lectores ópticos, (ii) por medio de lectores de capacitancia.

2.5.2.1 MÉTODO DE LECTORES ÓPTICOS

Los lectores ópticos funcionan como un dispositivo CCD¹⁰, como el que se emplea en las cámaras digitales. Tienen un arreglo de diodos¹¹ sensibles a la luz que generan una señal eléctrica en respuesta a ondas de luz.

Cada diodo graba un pixel¹², un pequeño punto que representa la luz que le es reflejada. Colectivamente, la luz y perfiles oscuros forman una imagen de la huella leída. El proceso de lectura comienza cuando se coloca el dedo sobre la ventana del lector, el cual tiene su propia fuente de iluminación, típicamente un arreglo de LED¹³s, para iluminar las crestas de la huella dactilar. El CCD genera, una imagen invertida del dedo, con áreas más oscuras que representan más luz reflejada obtenidas de las crestas de la huella y áreas más claras que representan menos luz reflejada obtenidas de los valles entre las crestas de la huella.

El procesador del lector se asegura que se ha obtenido una imagen digital lo suficientemente clara, antes de compararla con las muestras almacenadas. Revisa la oscuridad a través de los pixeles, o los valores generales en una muestra pequeña, y rechaza el resultado si la imagen general es demasiado oscura o demasiado clara. Si

¹⁰ CCD (Charged Coupled Device): Dispositivo de Carga Acoplada, es un circuito integrado que contiene un número determinado de condensadores enlazados o acoplados.

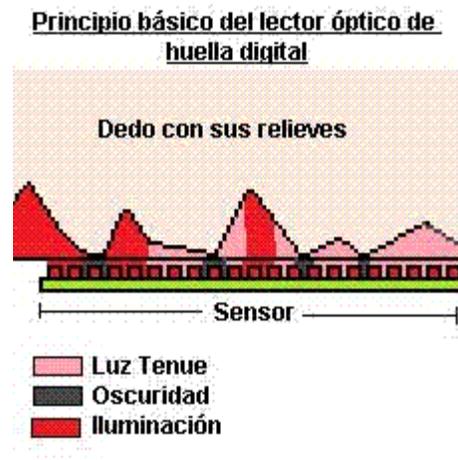
¹¹ DIODO: Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él.

¹² PIXEL: menor unidad en color que compone una imagen digital.

¹³ LED (Light-Emitting Diode): Diodo Emisor de Luz, es un diodo semiconductor que emite luz.

el resultado es rechazado, el lector ajusta su tiempo de exposición para permitir una cantidad mayor o menor de luz, luego de esto se realiza una nueva lectura de la huella.

Si el nivel de luz es adecuado, el lector revisa la definición de la imagen (que tan precisa es la imagen obtenida). El procesador busca varias líneas rectas que se mueven horizontal y verticalmente sobre la imagen, y si esta tiene buena definición, una línea que corre perpendicular a las crestas será hecha de secciones alternantes de pixeles muy claros y muy oscuros.



*Figura 2.17: Método de lectura de huella dactilar por sensores ópticos
Fuente: http://www.informaticamoderna.com/Lect_huella.htm*

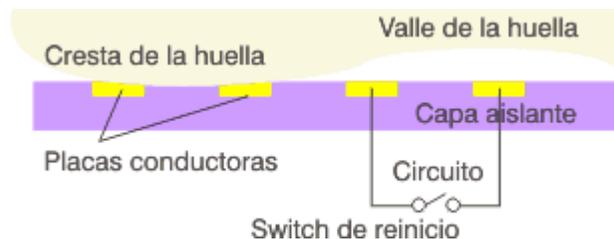
Para la comparación de las muestras la mayoría de los dispositivos biométricos usan los puntos de las minucias, para luego trazar líneas entre estos puntos formando una figura única en cada persona.

2.5.2.1 MÉTODO DE LECTORES DE CAPACITANCIA

Los lectores de capacitancia a diferencia de los lectores ópticos utilizan corriente eléctrica para obtener la imagen digitalizada de las crestas y los valles de la huella dactilar. El sensor de capacitancia está compuesto de uno o más chips que

contienen un conjunto de pequeñas celdas. Cada celda incluye dos placas conductoras, cubiertas con una capa aislante.

En el caso de que la placa conductora se encuentre con una cresta se tendrá un circuito cerrado (conectado), y si se encuentra con un valle se mantendrá el circuito abierto (desconectado), el resultado de esto es que cuando el capacitor se encuentre bajo una cresta posea un voltaje alto, y cuando el capacitor se encuentre con un valle posea un voltaje bajo.



*Figura 2.18: Método de lectura de huella dactilar por sensores capacitivos
Fuente: <http://www.soygik.com/como-funciona-un-lector-de-huellas-digital/>*

El software del sensor se encarga de construir una imagen digital en base a los diferentes valores de voltaje obtenidos.

La principal ventaja de usar un lector capacitivo es que requiere una verdadera forma de huella digital y no sólo un patrón de luz y oscuridad que haga la impresión visual de una huella digital como en los lectores ópticos, haciendo que el sistema tenga mayor seguridad.

2.5.3 PROCESO DE RECONOCIMIENTO DE HUELLAS DACTILARES

El proceso de reconocimiento de las huellas dactilares consta de 8 fases principales: (i) Adquisición de la imagen, (ii) Mejoramiento de la imagen, (iii) Imagen direccional, (iv) Red neuronal, (v) Segmentación, (vi) Postprocesamiento, (vii) Extracción de minucias, (viii) Verificación, independientemente del dispositivo que se use. En la Figura 2.8 se muestra el proceso que se realiza en cada una de estas etapas.

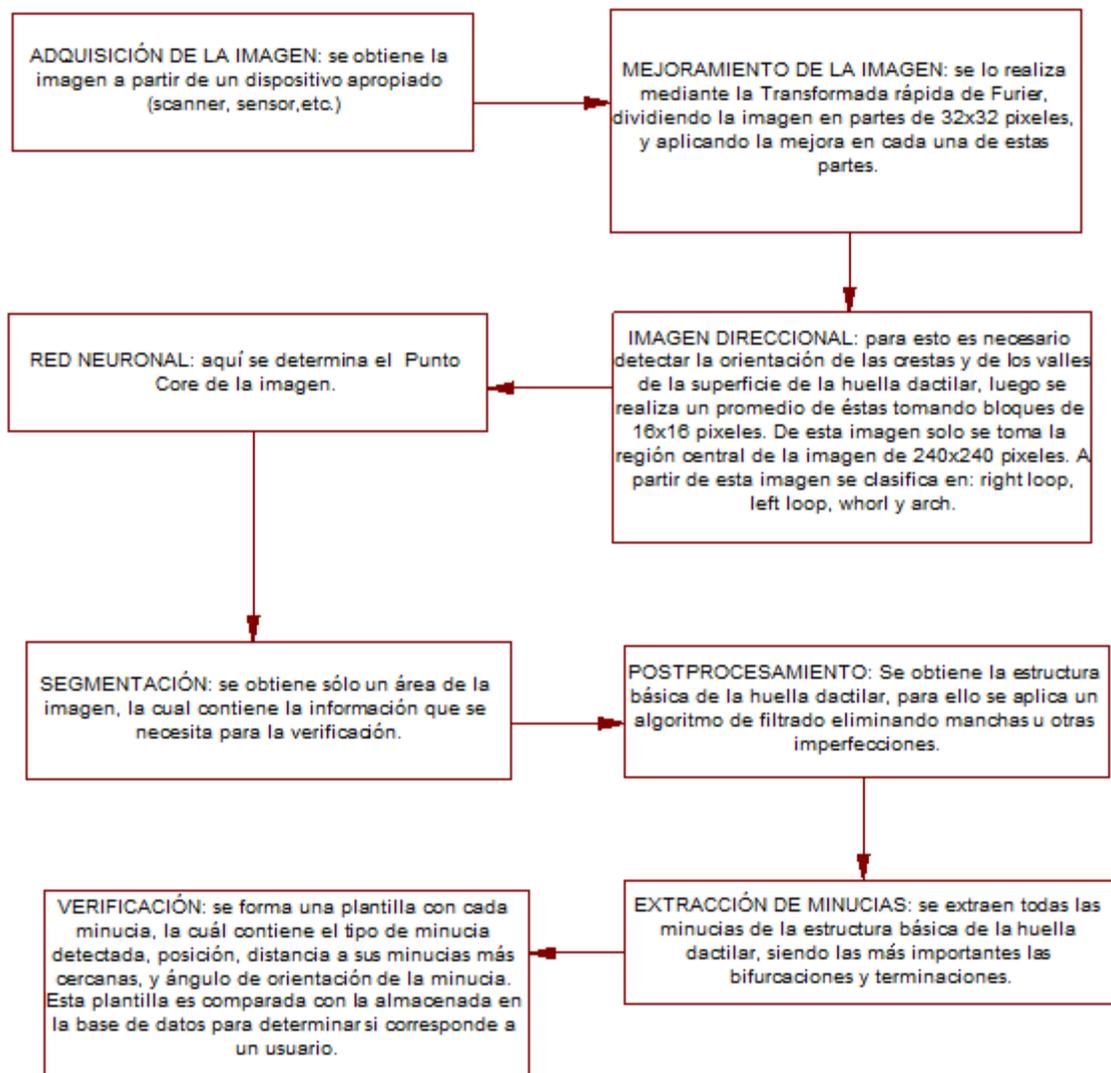


Figura 2.19: Proceso de reconocimiento de huella dactilar
Fuente: El Autor

2.5.4 VENTAJAS DE LA IDENTIFICACIÓN POR HUELLA DACTILAR

La identificación por huellas dactilares es conocida por tener un nivel de precisión alto y muy confiable, ya que las probabilidades de que dos o más personas tengan las mismas huellas son muy bajas.

Otras de sus ventajas son la permanencia e inmutabilidad, con el paso del tiempo, las únicas excepciones son cuando la persona ha sufrido un accidente o cirugía que

altere sus huellas. Es por ello que cada día se hace uso de las huellas dactilares con muchos fines en diversos tipos de empresas, dispositivos electrónicos, etc.

Además de las ventajas mencionadas anteriormente también se puede mencionar que un usuario nunca podrá “olvidar” su huella dactilar, a diferencia de un usuario y una contraseña los cuales pueden ser olvidados o utilizados por otras personas con fines inadecuados.

Finalmente cabe destacar su bajo costo de implementación y el fácil aprendizaje de uso por parte de los usuarios.

2.5.5 DESVENTAJAS DE LA IDENTIFICACIÓN POR HUELLA DACTILAR

A pesar de todas las ventajas que ofrece la identificación por medio de la huella dactilar también cabe destacar sus desventajas, las cuales no son muy significativas.

Al usar un sistema de identificación por huella dactilar se debe dar mantenimiento frecuentemente al dispositivo biométrico ya que al estar en contacto constantemente con los dedos de las manos, el lente tiende a ensuciarse de una manera más rápida, además se debe tener mucho cuidado al manipular el dispositivo, debido a que el sensor que poseen es muy frágil.

CAPÍTULO 3 ANÁLISIS Y DISEÑO

3.1 ANÁLISIS

3.1.1 REQUERIMIENTOS

Los requerimientos del proyecto se obtuvieron a partir de las entrevistas y cuestionarios realizados tanto al gerente como a los empleados de ALPROSART. En las siguientes tablas se presenta la tabulación de resultados obtenidos a partir de las encuestas y entrevistas realizadas que se encuentran en el Anexo I del documento.

Resultados de entrevista al Gerente y al Administrador

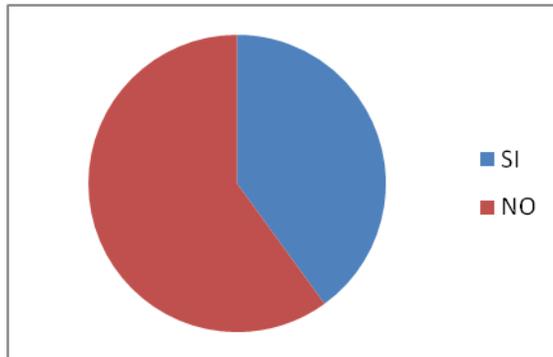
Luego de las entrevistas realizadas al Gerente General y al Administrador de ALPROSART, se llegó a la conclusión de que la manera en que se registra actualmente las horas de entrada o salida de los empleados no es segura y que son los propios empleados los que presentan inconformidad ante esta situación. Además de determinó que la forma en que se generan los roles de pagos actualmente toma demasiado tiempo y muchas veces no se toman en cuenta todos los valores para incluirlos en el mismo como son: multas, atrasos, adelantos, etc.

Adicionalmente no se cuenta con una forma efectiva de generar los reportes de atrasos solicitados por la gerencia.

Tabulación de resultados de encuesta al personal de producción y distribución

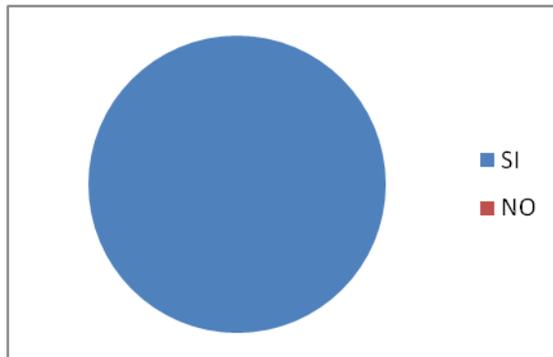
Pregunta 1: ¿Está conforme con la forma en la que usted registra sus horas de entrada y salida?

SI	40%
NO	60%



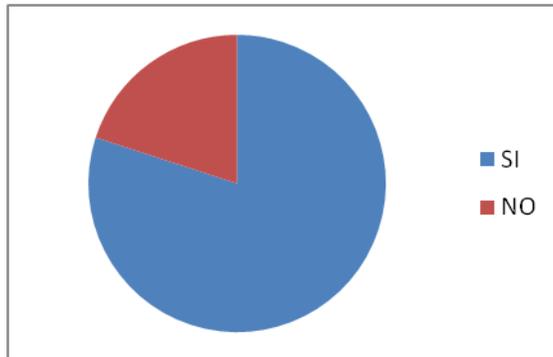
Pregunta 2: ¿Cree que se puede mejorar este proceso de registro?

SI	100%
NO	0%



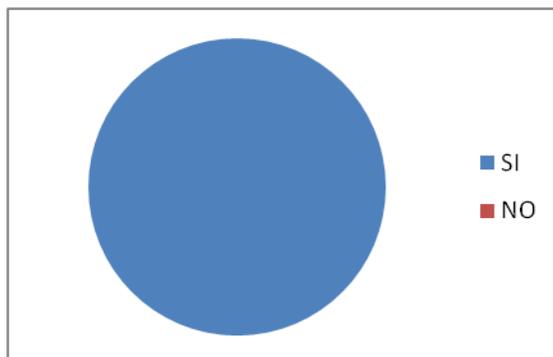
Pregunta 3: ¿Está conforme con la forma en que se realiza el cálculo de su rol de pago?

SI	80%
NO	20%



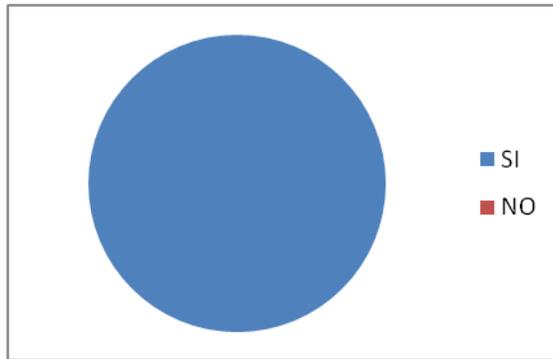
Pregunta 4: ¿Ha tenido dudas sobre si en realidad se han tomado en cuenta todas las horas laboradas para la emisión de su rol de pago?

SI	100%
NO	0%



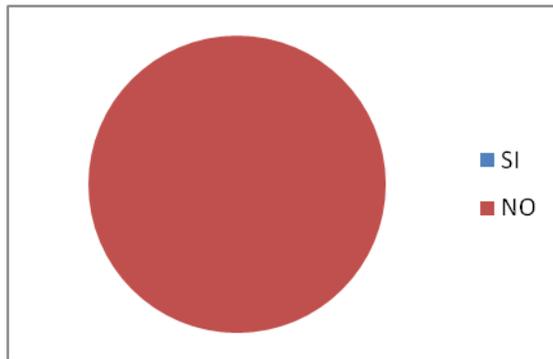
Pregunta 5: ¿Le gustaría saber exactamente cuántas horas ha laborado mensualmente?

SI	100%
NO	0%



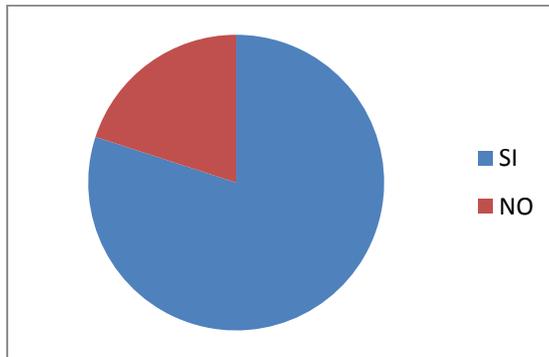
Pregunta 6: ¿Sabe usted qué es un lector de huellas dactilares?

SI	0%
NO	100%



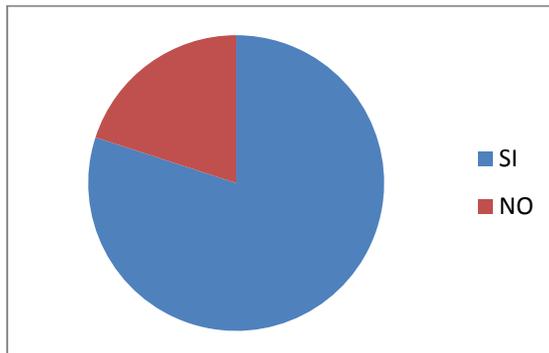
Pregunta 7: ¿Estaría de acuerdo en usar un lector de huellas dactilares para el registro de sus horas de entrada y salida? (Responda esta pregunta sólo si respondió SI a la pregunta 6)

SI	0%
NO	0%



Pregunta 8: ¿Estaría de acuerdo en que se use un sistema informático para el cálculo de su rol de pagos?

SI	80%
NO	20%



3.1.1.1 REQUERIMIENTOS FUNCIONALES

RF1. Administrar los datos personales de cada empleado, incluyendo su huella dactilar.

RF2. Administrar los cargos y departamentos de la empresa.

RF3. Registrar las horas de entrada y salida de los empleados por medio del lector biométrico.

RF4. Administrar turnos de trabajo de los empleados y días festivos.

RF5. Administrar multas, adelantos, bonificaciones, descuentos, permisos y vacaciones de los empleados.

RF6. Administrar calendario con las fechas de pago de obligaciones patronales y de servicios básicos, especificando que es lo que se tiene que pagar.

RF7. Generar rol de pago de cada empleado, detallando todos los valores que afecten al mismo.

RF8. Generar reportes de atrasos de uno o varios meses, por cada empleado y de todo el conjunto de empleados.

RF9. Generar reporte de remuneraciones recibidas por empleado en un período de tiempo.

3.1.1.2 REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales se dividieron en dos grupos, requerimientos de calidad y otros requerimientos no funcionales.

❖ REQUERIMIENTOS DE CALIDAD

- Desempeño.- garantizar el desempeño y la seguridad del sistema a los usuarios. La información que proporcione el sistema será accesible en cualquier momento y simultáneamente para todos los usuarios, sin afectar el tiempo de respuesta del mismo.
- Disponibilidad.- el sistema debe estar disponible las 24 horas del día, los 365 días del año, para todos los usuarios.
- Escalabilidad.- el sistema debe estar desarrollado de una manera que permita, agregar, modificar o eliminar, funcionalidades o módulos sin afectar en lo posible al código ya desarrollado.
- Facilidad de Uso e Ingreso de Información.- el sistema debe ser de fácil uso y aprendizaje, con una interfaz gráfica amigable para el usuario. Los mensajes

de error se deben presentar claramente al usuario, de manera que se pueda identificar el fallo y solucionarlo de manera ágil.

- Portabilidad.- el sistema debe funcionar sin ningún inconveniente, tanto en las plataformas de WINDOWS y LINUX.
- Mantenibilidad.- todo el sistema debe estar complemente documentado, cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados con sus respectivos manuales de usuario, para poder brindar el mantenimiento adecuado al mismo en el caso que se requiera. El sistema debe contar con un módulo de administración de usuarios para el ingreso al mismo.
- Seguridad.- el acceso al sistema debe estar restringido por un usuario y una contraseña asignada a cada uno de los usuarios. Además cada uno de los usuarios podrá visualizar sólo determinadas pantallas de acuerdo a un perfil asignado. El sistema debe registrar todas las acciones que realice un usuario para poder ser sometido a una auditoria en caso de que sea necesario.
- Validación de la información.- el sistema debe validar automáticamente la información ingresada por el usuario, tomando en cuenta, la obligatoriedad de los campos, longitud de caracteres permitida por cada campo de los formularios, manejo de tipo de datos, etc.

❖ OTROS REQUERIMIENTOS NO FUNCIONALES

- Arquitectura.- el sistema debe estar basado en la arquitectura de 3 capas (Capa de Presentación – Capa de Negocio – Capa de Datos), desarrollado como una aplicación Web dentro de una Intranet, la cual se pueda acceder desde un navegador de internet.
- Idioma.- el sistema debe tener interfaces gráficas en idioma español.
- Respaldos.- la información que se encuentre en la base de datos debe ser respaldada periódicamente. Estos respaldos deben ser generados automáticamente una vez por semana.

3.1.2 CASOS DE USO

Los casos de uso especificados que serán presentados proporcionan uno o más escenarios que indican cómo debería interactuar el sistema con el o los usuarios, para conseguir un objetivo específico, ayudando a determinar las características más importantes que el sistema debe tener.

Para el desarrollo de los casos de uso del proyecto se determinaron los actores, los procesos y sus relaciones, para luego realizar los respectivos diagramas de casos de uso.

3.1.2.1 PROCESOS

Los procesos identificados para el proyecto, basados en los requerimientos especificados, son los siguientes:

- Administrar empleados.
- Administrar cargos.
- Administrar departamentos.
- Administrar turnos de trabajo.
- Administrar días festivos.
- Administrar multas.
- Administrar descuentos.
- Administrar adelantos.
- Administrar bonificaciones.
- Administrar vacaciones.
- Administrar permisos.
- Administrar calendario de pago.
- Generar rol de pagos.
- Generar reporte de atrasos.
- Generar reporte de permisos.
- Generar reporte de remuneraciones recibidas.
- Registrar hora de entrada y salida por medio del dispositivo biométrico.

- Generar respaldo de base de datos.
- Administrar usuarios.
- Auditar acciones de usuarios
- Administrar accesos.

3.1.2.2 ACTORES

Los actores identificados que intervienen en los procesos especificados anteriormente son:

- Administrador: es el empleado que se encargará de administrar: datos personales del los empleados, cargos, departamentos, turnos de trabajo, días festivos, multas, atrasos, adelantos, bonificaciones, descuentos, usuarios, páginas de acceso al sistema, vacaciones, permisos, calendario de fechas de pago de obligaciones patronales, y de generar: rol de pagos por empleado, reporte de atrasos, reporte de permisos solicitados por el empleado.
- Empleados: son todas las personas que laboran en la empresa, los mismos que usarán el dispositivo biométrico para poder registrar sus horas de entrada y salida.

3.1.2.3 DIAGRAMA DE CASOS DE USO

En las siguientes figuras se presentan los casos de uso identificados, se muestra un caso de uso general del sistema, el caso de uso “Administrar Empleados” y el caso de uso “Registrar Hora de Entrada/Salida”, el resto de casos de uso con su respectiva especificación, se encuentran en el Anexo II.

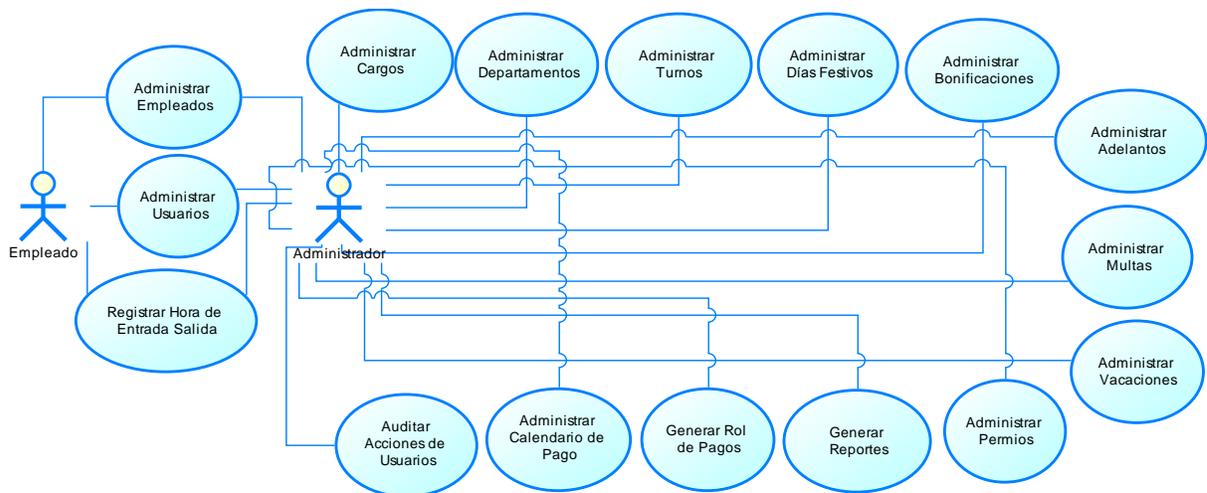


Figura 3.1: Diagrama General de Casos de Uso
Fuente: El Autor

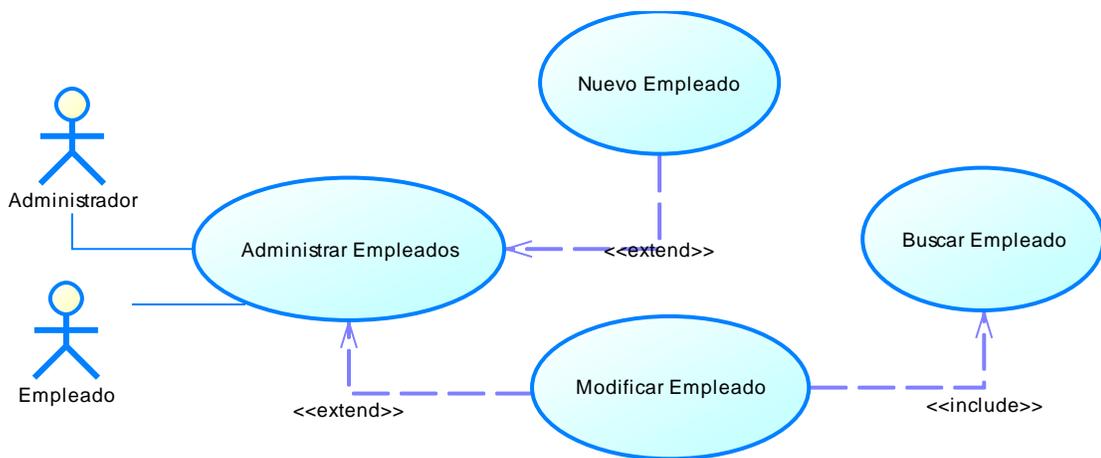


Figura 3.2: Diagrama de Caso de Uso: Administrar Empleados
Fuente: El Autor

CASO DE USO	Administrar Empleados
ACTORES	Administrador, Empleado
PROPÓSITO	Registrar los datos personales del empleado
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Son registrados los datos personales del empleado sin ningún inconveniente
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de Administración y luego al sub módulo de "Empleados", selecciona la opción "Nuevo Empleado" e ingresa todos los campos del formulario que presenta el sistema.

CURSO NORMAL DE EVENTOS
Paso 1. El administrador ingresa al módulo de administración
Paso 2. El sistema muestra el módulo de administración
Paso 3. El administrador ingresa al sub módulo de "Empleados"
Paso 4. El sistema muestra el sub módulo de "Empleados"
Paso 5. El administrador selecciona la opción "Nuevo Empleado"
Paso 6. El sistema muestra un formulario en blanco con los campos correspondientes a los datos personales del empleado
Paso 7. El administrador digita y/o selecciona cada uno de los datos correspondiente a los datos personales del empleado
Paso 8. El sistema valida y almacena la información ingresada
Paso 9. El administrador cierra el sub módulo de "Empleados"
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 5. El administrador selecciona la opción "Modificar Datos de Empleado"
Paso 8. El sistema muestra un mensaje de error porque la cédula del empleado ya existe en el sistema, o los datos ingresados son incorrectos, vuelve Paso 7.

Tabla 3.1: Caso de Uso Administrar Empleados

Fuente: El Autor

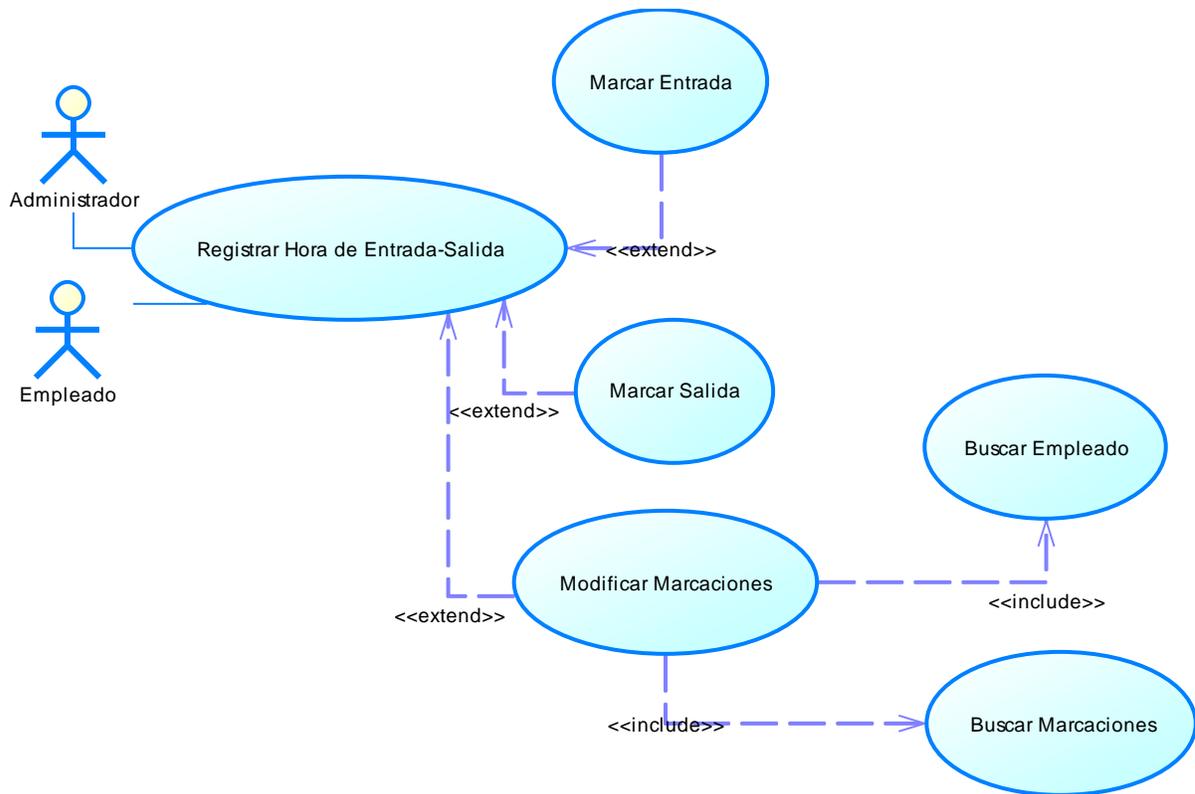


Figura 3.3: Diagrama de Casos de Uso: Registrar Hora de Entrada-Salida
Fuente: El Autor

CASO DE USO	Registrar Hora de Entrada / Salida
ACTORES	Administrador, Empleado
PROPÓSITO	Permite registrar la hora de entrada y salida de la jornada laboral de los empleados, por medio del lector biométrico
PRECONDICIÓN	Ingresar al sistema con usuario y clave válidos. El lector biométrico debe funcionar correctamente y el empleado debe colocar correctamente su huella dactilar sobre el sensor
POSTCONDICIÓN	Se verifica la huella dactilar y se registra la hora de entrada ó salida.
DESCRIPCIÓN	El empleado ingresa al sistema con su usuario y contraseña, selecciona el módulo de "Registro", selecciona "Marcar Entrada", aparece una ventana con la hora y fecha actual, el empleado coloca su dedo sobre el sensor del dispositivo biométrico, al obtener la huella dactilar el sistema verifica si esta existe en la base de datos, luego de esto se muestra una ficha con datos del empleado y se guarda la fecha y la hora exacta en la base de datos

CURSO NORMAL DE EVENTOS
Paso 1. El empleado ingresa al módulo de "Registro"
Paso 2. El sistema presenta el módulo de "Registro"
Paso 3. El empleado ingresa al sub módulo de "Marcar Entrada"
Paso 4. El sistema muestra el sub módulo de "Marcar Entrada"
Paso 5. El empleado coloca su dedo sobre el dispositivo biométrico
Paso 6. El sistema verifica la huella y registra la fecha y hora en que se colocó el dedo sobre el dispositivo biométrico
Paso 7. El sistema registra la hora y fecha en que el empleado colocó su huella dactilar
Paso 8. El sistema muestra una ventana con información correspondiente al empleado durante 10 segundos y luego se cierra
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve Paso 1.
Paso 3. El empleado selecciona la opción "Marcar Salida"
Paso 3. El administrador selecciona la opción "Modificar Marcaciones"
Paso 4. El sistema no muestra el módulo correctamente, vuelve Paso 3
Paso 5. El dispositivo biométrico no funciona, el empleado se comunica con el administrador para que habilite el registro de horas de entrada y salida por medio de número de cédula y contraseña del sistema
Paso 7. El sistema no registra la hora y fecha, vuelve Paso 6

*Tabla 3.2: Caso de Uso Registrar Hora de Entrada – Salida
Fuente: El Autor*

3.1.3 HERRAMIENTAS DE HARDWARE Y SOFTWARE EMPLEADAS PARA EL DESARROLLO DEL PROYECTO

Luego de haber establecido los requerimientos funcionales, no funcionales y los casos de uso del sistema ha ser desarrollado, se han tomado en cuenta las siguientes herramientas de hardware y software para el desarrollo del mismo.

3.1.3.1 HARDWARE

- Computador

Se usarán dos computadores, en uno se conectará el dispositivo biométrico para el registro de los empleados y en otro se conectará impresora y demás accesorios

necesarios para el funcionamiento correcto del sistema, este segundo computador lo usará el administrador del sistema, se utilizarán los computadores con los que cuenta la empresa actualmente, estos computadores poseen las siguientes características:

Computador 1

Marca: Compaq

Procesador: Intel Pentium 4, 2 Ghz

Memoria RAM: 2GB

Disco Duro: 100 GB

Computador 2

Marca: HP, T Series

Procesador: Intel Core 2 Duo, 3 GHz

Memoria RAM: 4GB

Disco Duro: 500GB

- Impresora

Para imprimir los reportes y demás comprobantes generados por el sistema, la empresa cuenta actualmente con una impresora de las siguientes características:

Marca: EPSON

Modo de impresión: Inyección

- Dispositivo Biométrico: Digital Persona U.are.U 4500

Para el reconocimiento e identificación biométrica se escogió el lector de Digital Persona (U.are.U 4500) debido a su bajo costo y alto grado de precisión, además de contar con un SDK¹⁴ de libre distribución y con el soporte necesario para el desarrollo de aplicaciones.

¹⁴ Software Development Kit: conjunto de herramientas de desarrollo de software que permite crear aplicaciones para un sistema.

CARACTERÍSTICAS DEL LECTOR BIOMÉTRICO DIGITAL PERSONA U.are.U 4500



Figura 3.4: Dispositivo Biométrico Digital Persona U.are.U 4500
Fuente: <http://www.digitalpersona.com/Biometrics/Hardware-Products/U-are-U-4500-Reader/4500-Reader/>

Diseñado para:

- Seguridad en computadores de escritorio.
- Seguridad en computadores móviles.
- Aplicaciones personalizadas.

Características:

- LED azul.
- Calidad de imagen excelente.
- Información de huella dactilar encriptada.
- Rechazo a huellas falsas.
- Sensor resistente.
- Trabaja correctamente con huellas mojadas, ásperas o secas.
- Compatible con Windows 7, Vista, XP Professional, 2000 y Windows Server 2000, 2003, 2008.

Especificaciones Técnicas:

- Resolución en píxeles: 512 dpi (promedio x, y sobre el área escaneada).
- Área de captura: 14.6 mm (ancho), 18.1 mm (alto).
- Escala de grises de 8 bits (256 niveles de gris).
- Compatible con especificaciones USB 1.0, 1.1 y 2.0 (Full Speed).
- Apto para uso interior y exterior.
- Peso: 105 gr.
- Temperatura de funcionamiento: 0 – 40 °C.
- Nivel de humedad para funcionamiento correcto: 20% - 80%.

3.1.3.2 SOFTWARE

Luego de realizar un segundo análisis de la Tabla 1.1 en el Capítulo 1 donde se proporcionan varias opciones de software para el desarrollo del proyecto, se ha realizado una variación de las herramientas escogidas en primera instancia.

- Diseño: Sybase Power Designer 15

Se optó por esta herramienta por ser de fácil uso y de gran funcionalidad para el modelado de los diferentes diagramas (casos de uso, base de datos, secuencia).

- Codificación: Eclipse Indigo

Para la codificación se usará esta herramienta, bajo el lenguaje de programación JAVA. Cabe recalcar que JAVA es uno de los lenguajes de programación más usados actualmente y posee una documentación completa disponible en la Web.

- Reportes: Jasper Reports para Eclipse

Se usará esta herramienta para el diseño y presentación de los reportes requeridos por la empresa. Su interfaz gráfica lo hace una de las herramientas de reportes más potente y usada actualmente.

- Motor de Base de Datos: PostgreSQL 8.4

PostgreSQL es uno de los motores de base de datos con mayor acogida en el mercado debido a su gran funcionalidad a pesar de ser de libre distribución.

- Servidor de Aplicaciones: JBOSS GA 5.1

JBOSS proporciona los servicios necesarios para alojar una aplicación Web, los cuales son configurados fácilmente por medio de archivos XML¹⁵.

- Framework: Seam 2.2

Proporciona una serie de componentes Web los cuales se adaptan fácilmente a las librerías que proporciona el servidor de aplicaciones JBOSS.

3.2 DISEÑO

3.2.1 MODELO DE BASE DE DATOS

El modelo conceptual de la base de datos se lo realizó representando los elementos que intervienen en la solución propuesta a los requerimientos de la empresa y sus relaciones.

¹⁵ XML: Extensible Markup Language, es un metalenguaje que permite la compatibilidad entre varios sistemas para compartir información de una manera segura y fiable.

● Modelo Físico

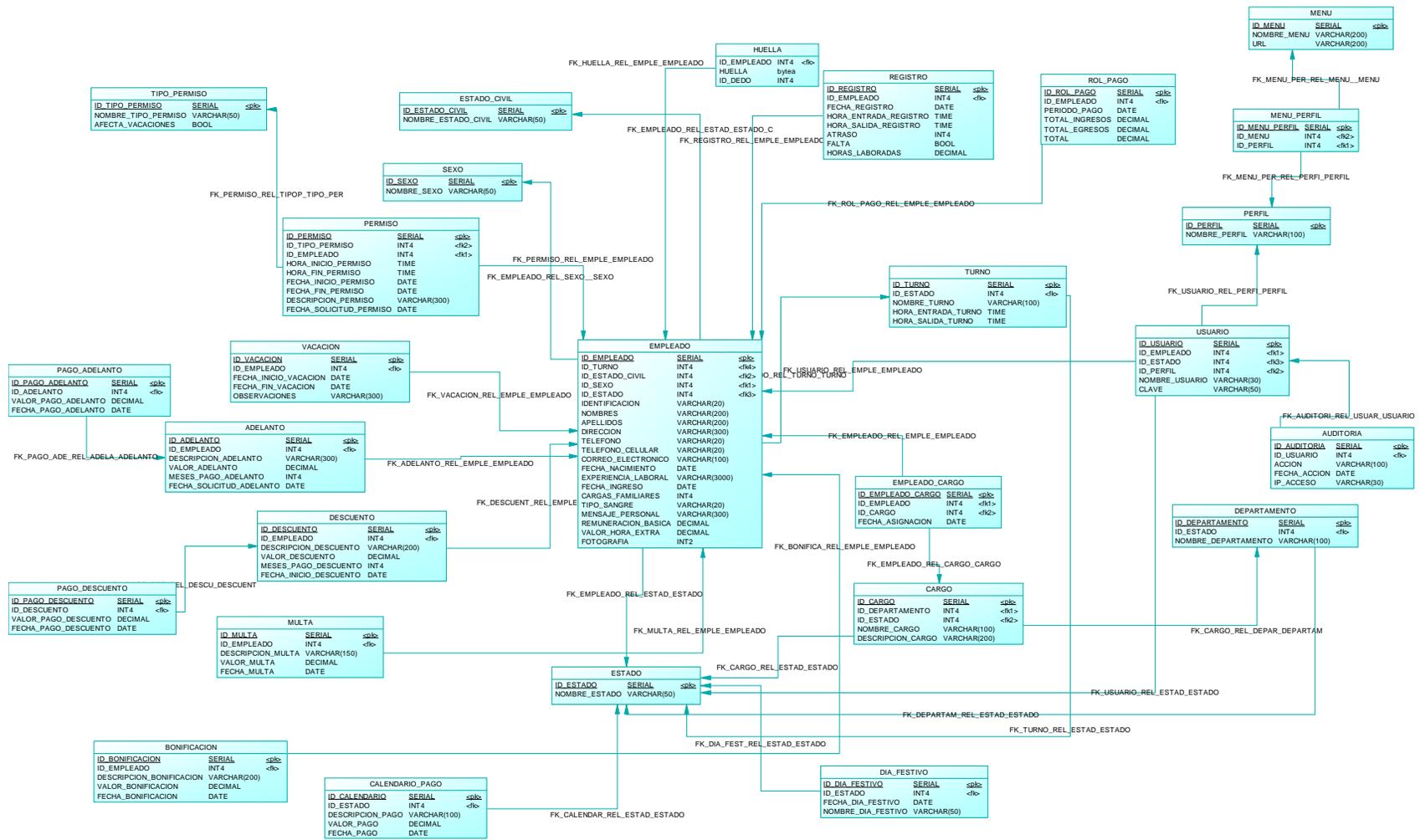


Figura 3.6: Modelo Físico de Base de Datos
Fuente: El Autor

3.2.2 DICCIONARIO DE DATOS

Para tener una idea más clara del diseño de la base de datos se especificarán los campos que componen cada una de las tablas y las relaciones que existen entre las mismas, en este capítulo se mostrará un listado de todas las tablas con una breve descripción y el detalle de la tabla “EMPLEADO”, el resto de las tablas se detallan en el ANEXO III.

Nombre	Descripción
ADELANTO	Tabla correspondiente a los adelantos de sueldo solicitados por un empleado
AUDITORIA	Tabla con información histórica de las acciones realizadas por los usuarios en el sistema
BONIFICACION	Tabla correspondiente a las bonificaciones recibidas por un empleado
CALENDARIO_PAGO	Tabla correspondiente a las fechas en que se debe cancelar una deuda que tenga la empresa
CARGO	Tabla de cargos que existen dentro de un determinado departamento
DEPARTAMENTO	Tabla de los departamentos que existen en la empresa
DESCUENTO	Tabla correspondiente a los descuentos que posee un empleado
DIA_FESTIVO	Tabla de días festivos a ser tomados en cuenta al momento de realizar el rol de pagos
EMPLEADO	Tabla que representa los datos más importantes para la empresa de un empleado
EMPLEADO_CARGO	Tabla histórica correspondiente a los cargos que ha desempeñado un empleado en la empresa
ESTADO	Catálogo de estados que indican si un registro está activo, inactivo, pagado, etc.
ESTADO_CIVIL	Catálogo de estado civil de una persona
MENU	Tabla correspondiente a las páginas que se disponen en el sistema

MENU_PERFIL	Tabla correspondiente a los menús que son asignados a un perfil
MULTA	Tabla correspondientes a las multas aplicadas a un empleado por faltas a las reglas de la empresa
PAGO_ADELANTO	Tabla correspondiente al desglose de los pagos en los que se ha diferido un adelanto
PAGO_DESCUENTO	Tabla correspondiente al desglose de los pagos que debe realizar un empleado por motivos de un descuento
PERFIL	Tabla de perfiles asignados a los usuarios, de los cuales dependen las acciones que pueden realizar en el sistema
PERMISO	Tabla correspondiente a los permisos solicitados por un empleado
REGISTRO	Tabla correspondiente a los registros de entrada y salida de un empleado
ROL_PAGO	Tabla donde se almacenan datos históricos de las remuneraciones recibidas por un empleado
SEXO	Catálogo correspondiente al sexo de una persona
TIPO_PERMISO	Catálogo correspondiente a los tipos de permiso
TURNO	Tabla correspondiente a los turnos de trabajo de la empresa
USUARIO	Tabla de usuarios asignados a un empleado
VACACION	Tabla correspondiente a las vacaciones solicitadas por un empleado
HUELLA	Tabla correspondiente a las huellas dactilares de un empleado.

*Tabla 3.3: Listado de Tablas de Base de Datos
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_EMPLEADO	SERIAL	X	X		Número identificador único de cada empleado, secuencial
ID_TURNO	INT4			X	Número identificador único de un turno,

					secuencial
ID_ESTADO_CIVIL	INT4			X	Número identificador único de un estado civil, secuencial
ID_SEXO	INT4			X	Número identificador único de un sexo, secuencial
ID_ESTADO	INT4			X	Número identificador único de un estado, secuencial
IDENTIFICACION	INT4	X			Número de identificación del empleado (cédula, pasaporte)
NOMBRES	VARCHAR(200)	X			Nombres del empleado
APELLIDOS	VARCHAR(200)	X			Apellidos de empleado
DIRECCION	VARCHAR(300)				Dirección domiciliaria del empleado
TELEFONO	VARCHAR(20)				Teléfono convencional de contacto del empleado
TELEFONO_CELULAR	VARCHAR(20)	X			Teléfono celular de contacto del empleado, se usará este número para enviar las alertas de pago vía mensaje de texto
CORREO_ELECTRONICO	VARCHAR(20)	X			Correo electrónico del empleado, se usará este correo para enviar las alertas de pago
FECHA_NACIMIENTO	DATE	X			Fecha de nacimiento del

					Empleado
EXPERIENCIA_LABORAL	VARCHAR(3000)				Experiencia laboral que posee el empleado
FECHA_INGRESO	DATE	X			Fecha en la que ingresó el empleado a la empresa
CARGAS_FAMILIARES	INT4				Número de cargas familiares del empleado
TIPO_SANGRE	VARCHAR(20)				Grupo sanguíneo al que pertenece el empleado
MENSAJE_PERSONAL	VARCHAR(300)				Mensaje que se muestra al momento que marca la entrada o salida el empleado
REMUNERACION_BASICA	DECIMAL	X			Sueldo base que percibe el empleado
VALOR_HORA_EXTRA	DECIMAL	X			Valor de la hora extra del empleado, debe ser mayor o igual al valor de la hora extra que estipula la ley

*Tabla 3.4: Listado de columnas de la tabla "EMPLEADO"
Fuente: El Autor*

3.2.3 DIAGRAMA DE SECUENCIA

Para el desarrollo del proyecto se han tomado en cuenta los siguientes diagramas de secuencia que representan los detalles de la implementación del sistema.

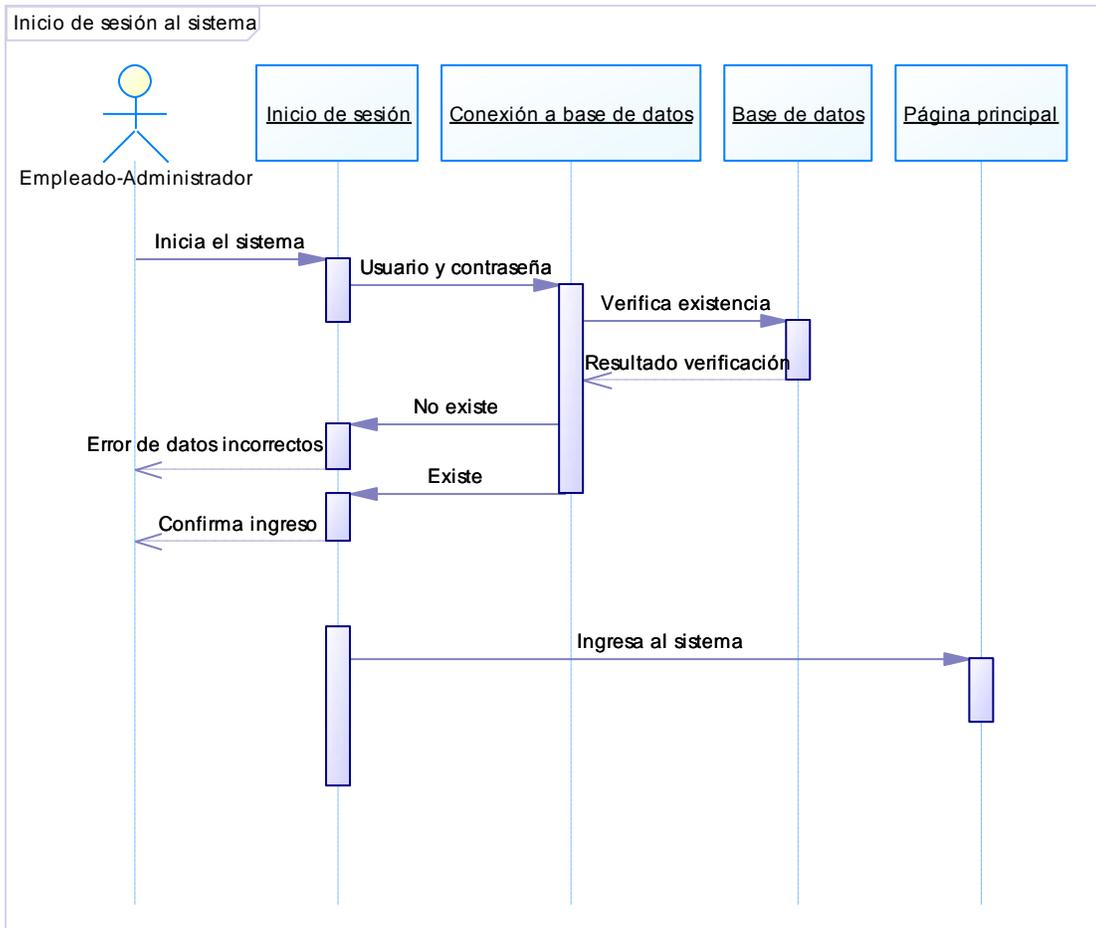


Figura 3.7: Diagrama de secuencia de inicio de sesión
Fuente: El Autor

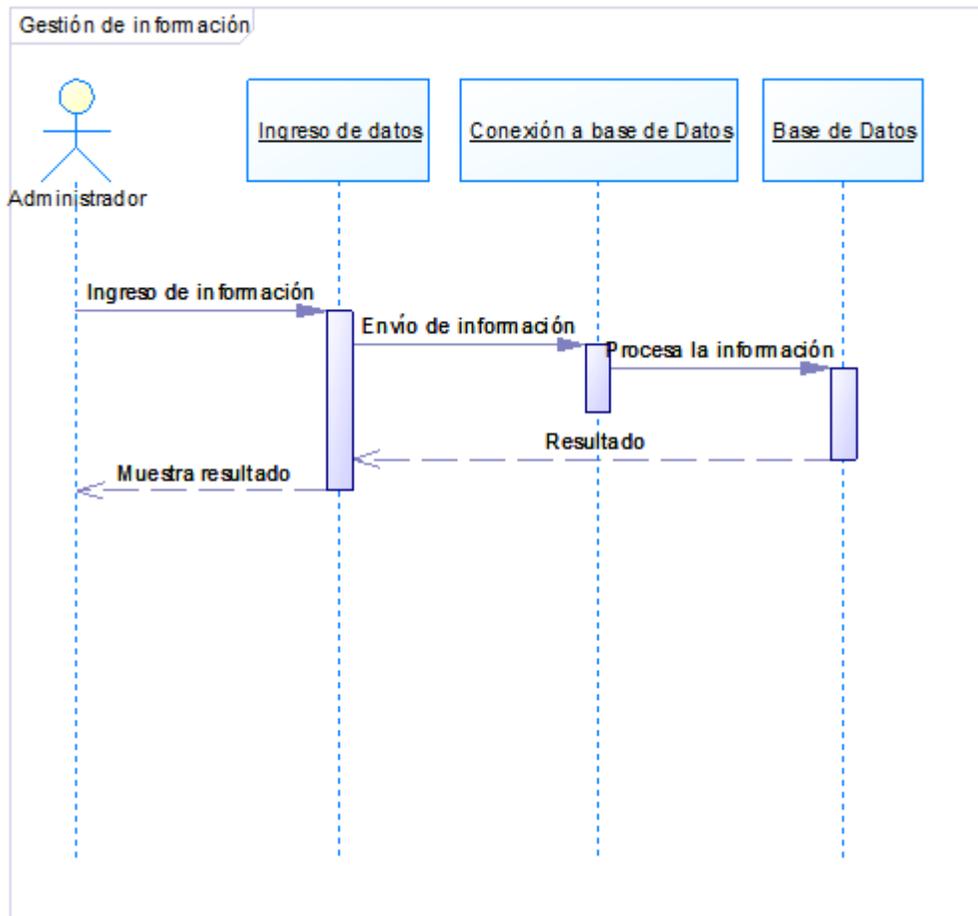


Figura 3.8: Diagrama de secuencia de gestión de la información
Fuente: El Autor

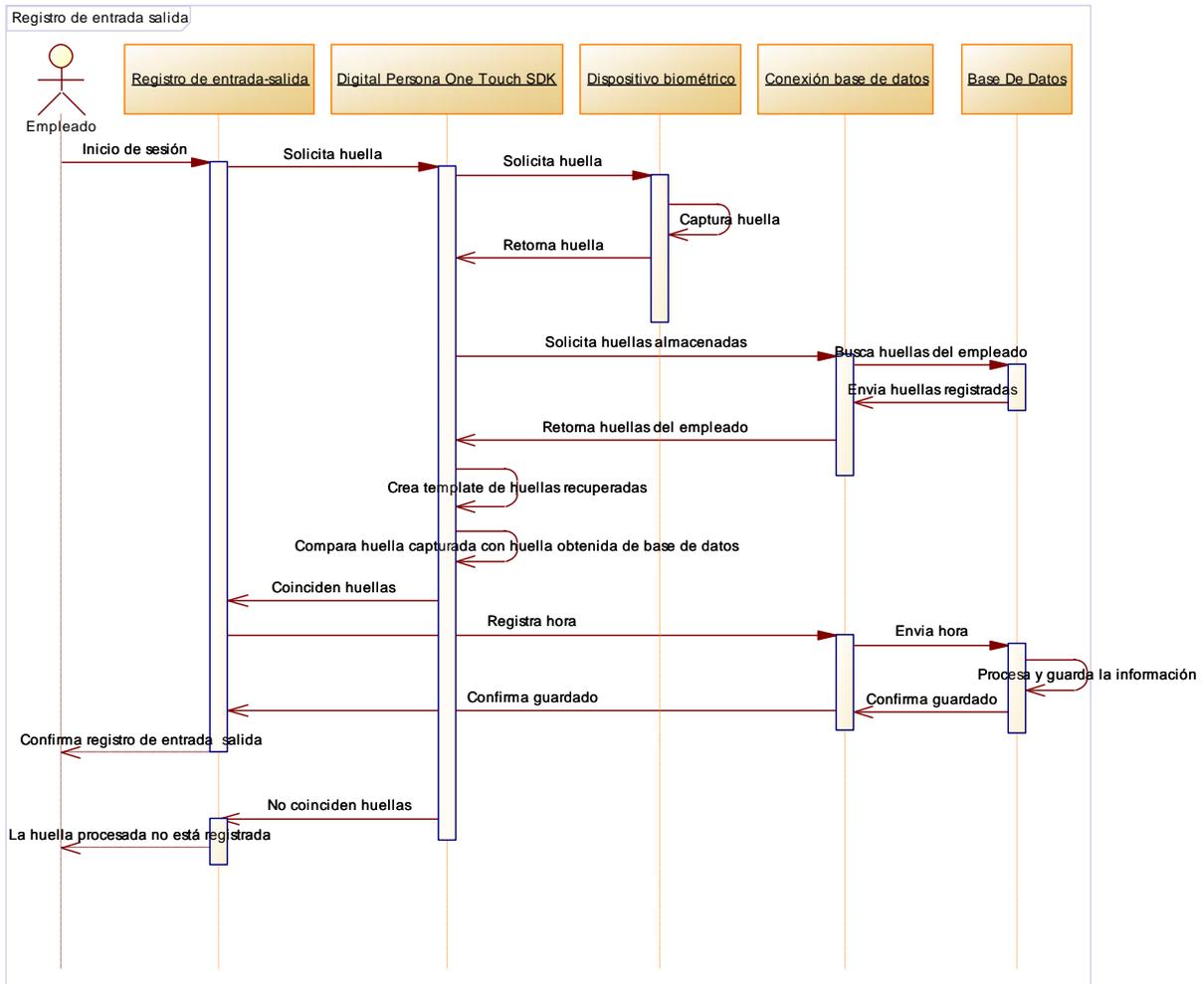


Figura 3.9: Diagrama de secuencia de registro de entrada-salida
Fuente: El Autor

3.2.4 PROTOTIPO DE INTERFACES

Son una muestra inicial de las pantallas / formularios del sistema, con la finalidad de que el usuario tenga una referencia gráfica de cómo se verá el sistema terminado y para que se tenga una idea macro de la aplicación. Estos prototipos son utilizados como guía para generar los formularios finales, facilitando cualquier cambio que se presente durante el desarrollo del proyecto.

A continuación se presentan los prototipos de los formularios que se ha diseñado para los procesos principales del sistema.

Formulario de inicio de sesión: Este formulario servirá para ingresar los datos de usuario y clave, necesarios para autenticarse en el sistema.

LOGO DE LA EMPRESA							
IMAGEN DE LA EMPRESA	<table><tr><td style="vertical-align: top;">USUARIO</td><td><input type="text"/></td></tr><tr><td style="vertical-align: top;">CLAVE</td><td><input type="password"/></td></tr><tr><td></td><td style="text-align: center;"><input type="button" value="INGRESAR"/></td></tr></table>	USUARIO	<input type="text"/>	CLAVE	<input type="password"/>		<input type="button" value="INGRESAR"/>
USUARIO	<input type="text"/>						
CLAVE	<input type="password"/>						
	<input type="button" value="INGRESAR"/>						

Figura 3.10: Formulario de inicio de sesión
Fuente: El Autor

Pantalla principal: será la pantalla que se muestre luego de haber iniciado sesión, en la parte superior derecha se mostrará que usuario se encuentra autenticado y en el panel central estará la misión y visión de la empresa.

						USUARIO
LOGO DE LA EMPRESA						
IMAGEN DE LA EMPRESA	ADMINISTRACIÓN	ROL DE PAGOS	REGISTRO	REPORTES	AUDITORÍA	
	MISIÓN VISIÓN					

Figura 3.11: Pantalla Principal
Fuente: El Autor

Formulario de nuevo registro: mostrará los campos que deben ser llenados para guardar un nuevo registro por ejemplo empleados, multas, adelantos, etc.

						USUARIO	
LOGO DE LA EMPRESA							
IMAGEN DE LA EMPRESA	ADMINISTRACIÓN	ROL DE PAGOS	REGISTRO	REPORTES	AUDITORÍA		
	CAMPO 1	<input type="text"/>					
	CAMPO 2	<input type="text"/>					
	CAMPO N	<input type="text"/>					
			GUARDAR				CANCELAR

Figura 3.12: Formulario de ingreso de nuevo registro
Fuente: El Autor

Formulario actualizar – anular: en este formulario se buscará por medio de parámetros específicos un registro ya sea para modificarlo o para eliminarlo permanentemente.

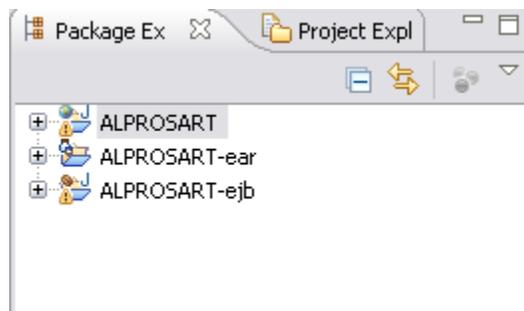
					USUARIO
LOGO DE LA EMPRESA					
IMAGEN DE LA EMPRESA	ADMINISTRACIÓN	ROL DE PAGOS	REGISTRO	REPORTES	AUDITORÍA
	CAMPO 1	<input type="text"/>		<input type="button" value="BUSCAR"/>	
	CAMPO 2				
	CAMPO N				
RESULTADOS / ACCIONES					
<input type="button" value="ACEPTAR/GUARDAR"/>			<input type="button" value="CANCELAR"/>		

Figura 3.13: Formulario de actualizar-anular
Fuente: El Autor

CAPÍTULO 4

DESARROLLO

Como fue definido en el Capítulo 3 para la codificación del sistema se usará el IDE Eclipse (Indigo Build.20120216-1857). Una vez instalado y configurado se creó un proyecto “Seam Web Project” al que se lo nombró “ALPROSART”, luego se generarán las siguientes carpetas:



*Figura 4.1: Carpetas generadas para el desarrollo del proyecto
Fuente: El Autor*

En donde:

- ALPROSART: es el directorio donde van a estar todas las páginas Web del sistema.
- ALPROSART-ear: es el directorio donde se encuentran todos los archivos .jar, los mismo que conforman el ejecutable del sistema.
- ALPROSART-ejb: en este directorio se colocará todo el código fuente del sistema.

4.1 PROCESO PARA GENERAR LA BASE DE DATOS

“ALPROSART”

A partir de los modelos conceptual y físico que fueron diseñados en Power Designer 15 se obtuvo el script de la base de datos “ALPROSART”, la misma que se utilizará para almacenar toda la información del sistema a desarrollar.

La *Figura 4.1* muestra las tablas y funciones creadas a partir del script obtenido.

También fueron creadas funciones para que cumplan tareas determinadas, las mismas que ayudarán al mejor rendimiento del servidor de aplicaciones.

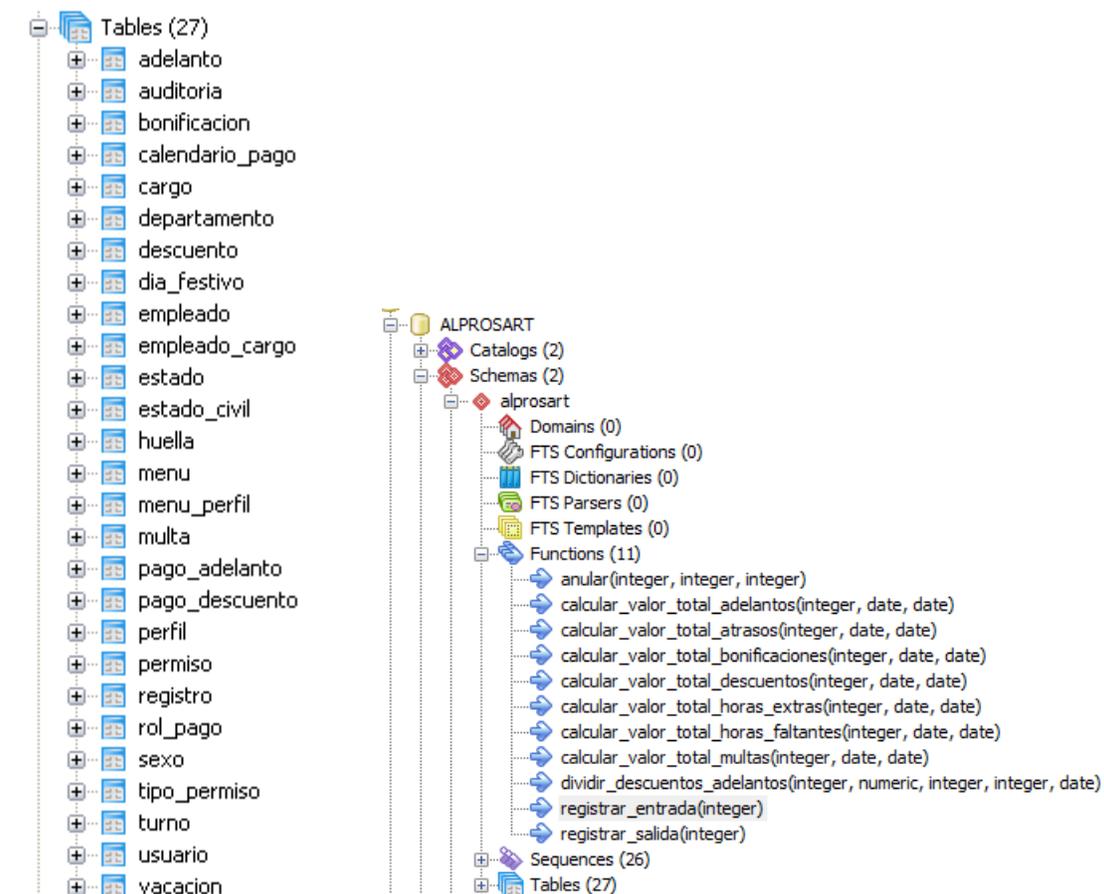


Figura 4.2: Tablas y funciones de la base de datos “ALPROSART”

Fuente: El Autor

4.2 EXPLICACIÓN DETALLADA DE LA LÓGICA DE LOS MÓDULOS, FUNCIONES Y CLASES

En los siguientes puntos se describen los métodos, clases y funciones más relevantes utilizados en el desarrollo del sistema.

- **Funciones de la base de datos**

Las funciones desarrolladas en la base de datos están escritas en lenguaje Pg/PISQL.

- I. Función anular:

La función anular recibe tres parámetros de tipo *integer* (números enteros) y retorna un valor tipo *boolean* (verdadero, falso). El proceso que realiza la función es el siguiente:

En el cuerpo de la función se declaran alias para cada parámetro que recibe la función:

- a) *f_id*: alias para el parámetro 1, representa al número identificador del registro que se requiere borrar.
- b) *f_id_empleado*: alias para el parámetro 2, representa al número identificador del empleado al que pertenece el registro a borrar.
- c) *f_tabla*: alias para el parámetro 3, representa a la tabla de donde se va a borrar el registro.

También se declara la variable *resultado* con valor inicial *false*, es el valor que devuelve la función, *true* si se han eliminado registros o *false* si no se borró información.

El primer paso es verificar la tabla de dónde se va a borrar el registro por medio de la sentencia *if... then*, este proceso utiliza el valor de *f_tabla*, donde:

- 1 = tabla ALPROSART.DESCUENTO
- 2 = tabla ALPROSART.ADELANTO
- 3 = tabla ALPROSART.MULTA

- 4 = tabla ALPROSART.BONIFICACION
- 5 = tabla ALPROSART.VACACIONES
- 6 = tabla ALPROSART.PERMISO

Una vez identificado de que tabla se va a borrar la información, se comprueba si el valor de *f_id* está relacionado con el valor de *f_id_empleado* mediante la sentencia *if exists ... then* y la sentencia *SELECT*.

Finalmente si la verificación es exitosa procede a borrar la información de la tabla con el id especificado, cambiando el valor del resultado a *true*.

```

CREATE OR REPLACE FUNCTION anular(integer, integer, integer)
  RETURNS boolean AS
$BODY$
DECLARE
f_id ALIAS FOR $1;
f_id_empleado ALIAS FOR $2;
f_tabla ALIAS FOR $3;

resultado boolean:=false;

BEGIN
  if f_tabla = 1 then --anula descuento
    if exists (select id_descuento from alprostart.descuento where id_descuento = f_id and id_empleado = f_id_empleado) then
      delete from alprostart.pago_descuento where id_descuento = f_id;
      delete from alprostart.descuento where id_descuento = f_id and id_empleado = f_id_empleado;
      resultado := true;
    end if;
  end if;

  if f_tabla = 2 then --anula adelanto
    if exists (select id_adelanto from alprostart.adelanto where id_adelanto = f_id and id_empleado = f_id_empleado) then
      delete from alprostart.pago_adelanto where id_adelanto = f_id;
      delete from alprostart.adelanto where id_adelanto = f_id and id_empleado = f_id_empleado;
      resultado := true;
    end if;
  end if;

  if f_tabla = 3 then --anula multa
    if exists (select id_multa from alprostart.multa where id_multa = f_id and id_empleado = f_id_empleado) then
      delete from alprostart.multa where id_multa = f_id and id_empleado = f_id_empleado;
      resultado := true;
    end if;
  end if;

  if f_tabla = 4 then --anula bonificacion
    if exists (select id_bonificacion from alprostart.bonificacion where id_bonificacion = f_id and id_empleado = f_id_empleado) then
      delete from alprostart.bonificacion where id_bonificacion = f_id and id_empleado = f_id_empleado;
      resultado := true;
    end if;
  end if;

  if f_tabla = 5 then -- anula vacaciones
    if exists (select id_vacacion from alprostart.vacacion where id_vacacion = f_id and id_empleado = f_id_empleado) then
      delete from alprostart.vacacion where id_vacacion = f_id and id_empleado = f_id_empleado;
      resultado := true;
    end if;
  end if;

  if f_tabla = 6 then -- anula permiso
    if exists (select id_permiso from alprostart.permiso where id_permiso = f_id and id_empleado = f_id_empleado) then
      delete from alprostart.permiso where id_permiso = f_id and id_empleado = f_id_empleado;
      resultado := true;
    end if;
  end if;

  RETURN resultado;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION anular(integer, integer, integer) OWNER TO alprostart;

```

Figura 4.3: Función anular de la base de datos
Fuente: El Autor

II. Función dividir_descuentos_adelantos

La función dividir_descuentos_adelantos recibe cuatro parámetros tres tipo *integer* (números enteros), uno tipo *numeric* (números decimales) y uno tipo *date* (fecha).

En el cuerpo de la función se declaran alias para cada parámetro que recibe la función:

- a) *f_id*: alias para el parámetro 1, representa al número identificador del registro que se requiere borrar.
- b) *f_valor*: alias para el parámetro 2, representa al valor total del descuento o adelanto.
- c) *f_numero_meses*: alias para el parámetro 3, representa el número de meses en los que se va a dividir el descuento o adelanto.
- d) *f_tabla*: alias para el parámetro 4, representa a la tabla en la que se va a realizar el ingreso de la información.
- e) *f_fecha_pago*: alias para el parámetro 5, representa a la fecha desde cuándo se va a realizar el pago del descuento o adelanto.

También se declara la variable *calculo* de tipo *numeric*, que es utilizada para asignar el valor de cada cuota de pago.

El primer paso es dividir *f_valor* / *f_numero_meses*, el resultado se asigna a la variable *calculo*.

Luego se realiza un contador con la sentencia *for...in...loop* desde 1 hasta el número de meses (*f_numero_meses*). Dentro del contador se verifica la tabla a donde se va a realizar el ingreso de la información, donde:

- 1 = tabla ALPROSART.PAGO_DESCUENTO
- 2 = tabla ALPROSART.PAGO_ADELANTO

Una vez verificada la tabla, se realiza el ingreso de la información mediante la sentencia *INSERT*, los valores a ingresar a la tabla son:

- *id*: número identificador único del registro.
- *calculo*: valor a cancelar calculado a partir de *f_valor* / *f_numero_meses*.
- *fecha_pago*: fecha que es calculada a partir de *f_fecha_pago* más el valor actual del contador transformado a meses.

```

CREATE OR REPLACE FUNCTION dividir_descuentos_adelantos ( integer, numeric(10,2), integer, integer, date ) returns void as $$
DECLARE
f_id ALIAS for $1;
f_valor ALIAS for $2;
f_numero_meses ALIAS for $3;
f_tabla ALIAS for $4;
f_fecha_pago ALIAS for $5;
calculo numeric (10,2);
BEGIN
    calculo:=(f_valor/f_numero_meses);
    for cont in 1..f_numero_meses loop
        begin
            if f_tabla = 1 then
                insert into alprosart.pago_descuento (id_pago_descuento,id_descuento,valor_pago_descuento, fecha_pago_descuento)
                values (nextval('alprosart.pago_descuento_id_pago_descuento_seq'),f_id,calculo, f_fecha_pago
                + cast (cont-1||' months' as interval) );
            end if;
            if f_tabla = 2 then
                insert into alprosart.pago_adelanto (id_pago_adelanto,id_adelanto,valor_pago_adelanto, fecha_pago_adelanto)
                values (nextval('alprosart.pago_adelanto_id_pago_adelanto_seq'),f_id,calculo, f_fecha_pago
                + cast (cont-1||' months' as interval) );
            end if;
        end;
    end loop;
END;
$$ LANGUAGE plpgsql;

```

Figura 4.4: Función *dividir_descuentos_adelantos* de la base de datos
Fuente: El Autor

III. Función *registrar_entrada*

La función *registrar_entrada* recibe un parámetro tipo *Integer* el cual corresponde al código identificador del empleado y retorna una variable tipo *boolean* la cual indica si se insertó (*true*) o no (*false*) el registro, el proceso que realiza es el siguiente:

En el cuerpo de la función se declara el alias para el parámetro tipo *Integer* que recibe la función en este caso es *f_id*.

Se declara la variable *resultado* tipo *boolean* que sirve para determinar si el registro se insertó correctamente o no.

Luego se verifica si la hora actual es mayor a 10 minutos (tiempo de gracia que tiene el empleado para registrar la entrada), con respecto a la hora de entrada del empleado, dependiendo que turno tenga asignado.

En el caso de que este valor sea verdadero se inserta un nuevo registro en la tabla *ALPROSART.registro* con la información de *id_registro*, *id_empleado*, *fecha_registro*, *hora_entrada_registro* y con el valor de *atraso* igual a *true*.

Caso contrario si el valor es falso se inserta la misma información cambiando el valor de *atraso* igual a *false*.

Si la información se ingresó correctamente se asigna a la variable resultado el valor de *true*, y se devuelve este valor. Todo este proceso se lo visualiza en la *Figura 4.5*.

```
create or replace function registrar_entrada(integer) returns boolean as $$
declare
    f_id ALIAS for $1;
    resultado boolean:= false;
begin
    --verifica si la hora actual es mayor a 10 minutos con respecto a la hora de entrada normal del empleado
    if (SELECT NOW()::TIME - ( SELECT t.hora_entrada_turno FROM alprostart.turno t, alprostart.empleado e
        WHERE e.id_empleado=f_id AND e.id_turno=t.id_turno )) > '00:10:00' then
        --si es verdadero inserta un nuevo registro en la tabla alprostart.registro con el valor e en atraso igual a true
        INSERT INTO alprostart.registro (id_registro,id_empleado,fecha_registro,hora_entrada_registro,atraso)
            VALUES ( nextval('alprostart.registro_id_registro_seq'), f_id, now()::date, now()::time,true);
        resultado:=true;
    else
        --si es falso inserta un nuevo registro en la tabla alprostart.registro con el valor e en atraso igual a false
        INSERT INTO alprostart.registro (id_registro,id_empleado,fecha_registro,hora_entrada_registro,atraso)
            VALUES ( nextval('alprostart.registro_id_registro_seq'), f_id, now()::date, now()::time,false);
        resultado:=true;
    end if;
    return resultado;
end;
$$ language plpgsql;
```

Figura 4.5: Función registrar_entrada
Fuente: El Autor

IV. Función registrar_salida

La función *registrar_salida* recibe un parámetro tipo *Integer* el cual corresponde al código identificador del empleado y retorna una variable tipo *boolean* la cual indica si se insertó (*true*) o no (*false*) el registro, el proceso que realiza es el siguiente:

En el cuerpo de la función se declara el alias para el parámetro que recibe la función, en este caso será *f_id*. También son declaradas las variables *resultado* tipo *boolean* que indica si el registro se insertó correctamente y la variable *f_horas_laboradas* tipo *time* a la cual se le asigna el valor de las horas laboradas del empleado en el día.

Luego se ingresa la hora de salida del empleado en el registro correspondiente al día en curso.

En el siguiente paso se calcula las horas laboradas, restando la hora de salida del empleado menos la hora de entrada y asignándolo a la variable *f_horas_laboradas*.

Finalmente se ingresa el valor de las horas laboradas al registro del día en curso y se devuelve el valor de la variable *resultado*.

Todo el proceso se visualiza en la *Figura 4.6*.

```
create or replace function registrar_salida(integer) returns boolean as $$
declare
    f_id ALIAS for $1;
    resultado boolean:= false;
    f_horas_laboradas time;
begin
    --ingreso de la hora de salida
    UPDATE alproart.registro SET hora_salida_registro = NOW()::TIME WHERE id_empleado=f_id AND fecha_registro=NOW()::DATE;
    --calculo de horas laboradas
    f_horas_laboradas= (SELECT hora_salida_registro FROM alproart.registro WHERE id_empleado=f_id AND fecha_registro=NOW()::DATE)
    - (SELECT hora_entrada_registro FROM alproart.registro WHERE id_empleado=f_id AND fecha_registro=NOW()::DATE);
    --ingreso de horas laboradas
    UPDATE alproart.registro SET horas_laboradas = f_horas_laboradas WHERE id_empleado=f_id AND fecha_registro=NOW()::DATE;
    resultado:=true;
    return resultado;
end;
$$ language plpgsql;
```

Figura 4.6: Función registrar_salida
Fuente: El Autor

V. Función *calcular_valor_total_descuentos*

La función *calcular_valor_total_descuentos* recibe un parámetro tipo *Integer* el cual corresponde al código identificador del empleado, y dos parámetros tipo *Date* los cuales determinan el período del cual se va a realizar el cálculo y retorna una variable tipo *Numeric* la cual indica el valor total de descuentos en el período especificado, el proceso que realiza es el siguiente:

El primer paso es declarar los alias para los parámetros que recibe, *f_id_empleado* para el primer parámetro tipo *Integer*, *f_fecha_inicio_mes* para el segundo parámetro tipo *Date* y *f_fecha_fin_mes* para el tercer parámetro tipo *Date*. También es declarada una variable tipo *Numeric* llamada *total_descuentos*.

Luego se realiza la sumatoria de los registros del campo *valor_pago_descuento* de la tabla *ALPROSART.pago_descuento* que se encuentren entre las fechas y empleado especificados. Este valor es asignado a la variable *total_descuentos*. El código fuente de la función se muestra en la *Figura 4.7*.

```

--FUNCION PARA CALCULAR EL VALOR A SER DESCONTADO POR DESCUENTOS EN EL MES
CREATE OR REPLACE FUNCTION calcular_valor_total_descuentos(INTEGER,DATE,DATE) RETURNS numeric AS $$
DECLARE
    f_id_empleado ALIAS FOR $1;
    f_inicio_mes ALIAS FOR $2;
    f_fin_mes ALIAS FOR $3;
    total_descuentos numeric (10,2);
BEGIN
    total_descuentos = 0;
    --sumatoria de descuentos correspondientes a cada mes, valor devuelto en USD
    total_descuentos = SUM (pd.valor_pago_descuento) FROM alprosart.pago_descuento pd, alprosart.descuento d
    WHERE d.id_empleado = f_id_empleado AND pd.id_descuento = d.id_descuento AND pd.fecha_pago_descuento
    BETWEEN f_inicio_mes AND f_fin_mes;
    RETURN total_descuentos;
END;
$$ language plpgsql;

```

Figura 4.7: Función *calcular_valor_total_descuentos*
Fuente: El Autor

VI. Función *calcular_valor_total_horas_extras*

La función *calcular_valor_total_horas_extras* recibe un parámetro tipo *Integer* el cual corresponde al código identificador del empleado, y dos parámetros tipo *Date* los cuales determinan el período del cual se va a realizar el cálculo y retorna una variable tipo *Numeric* la cual indica el valor total a ser acreditado al rol de pagos del empleado por horas extras en el período especificado, el proceso que realiza es el siguiente:

El primer paso es declarar los alias para los parámetros que recibe, *f_id_empleado* para el primer parámetro tipo *Integer*, *f_fecha_inicio_mes* para el segundo parámetro tipo *Date* y *f_fecha_fin_mes* para el tercer parámetro tipo *Date*. También se declaran dos variables tipo *Interval* llamadas *total_horas_laboradas* y *total_horas_extras*, una variable tipo *Decimal* llamada *valor_hora* y la variable *valor_total* de tipo *Numeric* que es el valor que va a ser devuelto por la función.

Luego de esto se calcula la sumatoria del campo *horas_laboradas* de la tabla *ALPROSART.registro* en el rango de fechas y con el código de empleado especificado. Este valor es asignado a la variable *total_horas_laboradas*.

Luego se verifica si el *total_horas_laboradas* es mayor a las 168 horas al mes que deben laborar los empleados.

Si la verificación es verdadera se asigna a la variable *valor_hora* el valor de hora extra que corresponde al empleado. Seguidamente se calcula el *total_horas-extras* restando el total de horas laboradas menos el número de horas mínimo al mes que debe laborar un empleado, en este caso es 168.

Finalmente a la variable *valor_total* se le asigna el resultado de multiplicar *valor_hora* por *total_horas_extras*, y este será el valor que devuelve la función. El código fuente completo de la función se muestra en la *Figura 4.8*.

```
--FUNCION PARA CALCULAR EL VALOR A SER ACREDITADO POR HORAS EXTRAS EN EL MES
CREATE OR REPLACE FUNCTION calcular_valor_total_horas_extras(INTEGER,DATE,DATE) RETURNS numeric AS $$
DECLARE
    f_id_empleado ALIAS FOR #1;
    f_inicio_mes ALIAS FOR #2;
    f_fin_mes ALIAS FOR #3;
    total_horas_laboradas interval;
    total_horas_extras interval;
    valor_hora decimal;
    valor_total numeric(10,2);
BEGIN
    valor_total = 0;
    --sumatoria de horas_extras correspondientes a cada mes
    total_horas_laboradas = SUM (horas_laboradas) FROM alprosart.registro WHERE id_empleado = f_id_empleado
    AND fecha_registro BETWEEN f_inicio_mes AND f_fin_mes;
    --604800 corresponde al valor en segundos de las 168 horas mínimo que deben trabajar al mes
    IF total_horas_laboradas > '604800'::interval THEN
        --se extrae el valor de la hora extra del empleado
        valor_hora = valor_hora_extra from alprosart.empleado WHERE id_empleado = f_id_empleado;
        --se calcula el número de horas extras
        total_horas_extras = total_horas_laboradas - '604800'::interval;
        --se multiplica el valor de la hora extra por el valor en número de horas del total de horas extras
        valor_total = valor_hora * ((EXTRACT(EPOCH FROM total_horas_extras))/3600);
    END IF;
    RETURN valor_total;
END;
$$ language plpgsql;
```

Figura 4.8: Función calcular_valor_total_horas_extras
Fuente: El Autor

- **Conexión con la Base de Datos**

El acceso a la información de la base de datos se lo realiza mediante JDBC¹⁶, los datos necesarios para establecer la conexión se encuentran en un archivo XML llamado “*ALPROSART-ds.xml*” ubicado en el directorio */deploy* del servidor de aplicaciones JBoss. La *Figura 4.9* muestra la manera en la que debe estar estructurado el archivo XML de configuración.

¹⁶ Java Database Connectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE datasources (View Source for full doctype...)>
- <datasources>
- <local-tx-datasource>
  <jndi-name>ALPROSARTDataSource</jndi-name>
  <use-java-context>true</use-java-context>
  <connection-url>jdbc:postgresql:ALPROSART</connection-url>
  <driver-class>org.postgresql.Driver</driver-class>
  <user-name>alprosart_t</user-name>
  <password>12345678</password>
</local-tx-datasource>
</datasources>

```

Figura 4.9: Estructura del archivo de configuración para acceso de base de datos
Fuente: El Autor

Los parámetros incluidos en el archivo de configuración son:

- a) *jndi-name*: nombre del *datasource* el cual se usa para realizar la conexión.
- b) *connection-url*: dirección de conexión a la base de datos, en el caso de que el servidor de base de datos se encuentre en otro equipo se debe incluir la dirección IP¹⁷ del mismo.
- c) *driver-class*: nombre del driver que será utilizado para la conexión.
- d) *user-name*: usuario de base de datos con el que se va a realizar la conexión.
- e) *password*: contraseña del usuario de base de datos establecido en la línea anterior.

Para establecer la conexión con la base de datos se creó el paquete *ec.com.alprosart.dao*, en este paquete fue creada la clase *ConexionBDD.java* en la que está el método *obtenerConexion*, este método retorna un objeto de tipo *java.sql.Connection*.

En el método *obtenerConexion* se declararon las siguientes variables:

- *connection*: objeto tipo *java.sql.Connection* que va a ser devuelto por el método.

¹⁷IP: número que identifica un dispositivo en una red.

- *initialContext*: objeto tipo *javax.naming.InitialContext.InitialContext* que obtiene el contexto inicial de los objetos.
- *dataSource*: objeto tipo *javax.sql.DataSource* que contiene el objeto *java.sql.Connection* a ser devuelto por el método.

El proceso que se realiza para obtener la conexión es el siguiente:

1. Se instancia la variable *initialContext*.
2. Se hace *lookup* al origen de datos *ALPROSARTDataSource* declarado en el archivo "*ALPROSART-ds.xml*". Luego se realiza *cast* del objeto obtenido a un objeto tipo *javax.sql.DataSource* y se lo asigna a la variable *dataSource*.
3. De la variable *dataSource* se obtiene el objeto *java.sql.Connection* con el método *getConnection()*, y se lo asigna a la variable *connection*.
4. Se retorna la variable *connection*.

```

public Connection obtenerConexion(){
    Connection connection = null;
    InitialContext initialContext;
    DataSource dataSource = null;
    try {
        initialContext = new InitialContext();
        dataSource = (DataSource) initialContext.lookup("java:ALPROSARTDataSource");
        connection = dataSource.getConnection();
    } catch (NamingException e1) {
        e1.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return connection;
}

```

Figura 4.10: Método *obtenerConexion()* para establecer la conexión con la base de datos
Fuente: El Autor

- **Métodos y Clases JAVA**

Los métodos y clases que se describen a continuación constituyen los más importantes del sistema desarrollado.

- I. Clase *TransaccionesBDD*

La clase pública *TransaccionesBDD* consta de todas las funcionalidades que permiten comunicarse con la base de datos entre los métodos más importantes de esta clase están:

Método insertar

El método insertar recibe una cadena de caracteres, en este caso la cadena es llamada *query*, para ejecutar la instrucción indicada se debe obtener un objeto tipo *java.sql.connection* y por medio de un objeto tipo *java.sql.PreparedStatement* que recibe la cadena de caracteres como parámetro. Si no hubo problemas en la ejecución de la instrucción la función retornará *true* caso contrario *false*.

```
public boolean insertar(String query){
    coneccionBDD = new ConeccionBDD();
    connection = coneccionBDD.obtenerConeccion();
    try {
        preparedStatement = connection.prepareStatement(query);
        preparedStatement.execute();
        preparedStatement.close();
        connection.close();

        auditoria.setAccion(query.replace("'", " '"));
        insertarAuditoria(auditoria);
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    } finally {
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    return true;
}
```

Figura 4.11: Método insertar()

Fuente: El Autor

Método insertarAuditoria()

Este método recibe un objeto tipo *Auditoria* que es una entidad especificada en el paquete *ec.com.alprosart.entidades*, este tipo de objeto tiene atributos los cuales serán insertados en la base de datos como son: *id_auditoria* (número secuencial), *id_usuario* (identificador del usuario que realiza la acción), *acción* (acción realizada por el usuario), *fecha_accion* (fecha en la que se realizó la acción), *ip_acceso* (dirección ip del computador donde se realizó la acción). Aquí son declaradas una

variable auxiliar de tipo *String* llamada *ip* y una variable tipo *Usuario* que es una entidad igualmente especificada en el paquete de entidades.

Se obtiene el usuario que está autenticado en el sistema por medio de la variable de sesión *usuario* y la dirección *IP* por medio de *ServletRequest*, una vez obtenidos estos valores, son asignados a las propiedades de la variable *auditoria*. A partir de esta variable se construye una instrucción SQL que registrará toda acción realizada en el sistema.

```
public void insertarAuditoria(Auditoria auditoria){
    String ip;
    Usuario usuario;
    //obtiene la variable de sesión usuario
    usuario = (Usuario) UtilsWeb.getSession().getAttribute("usuario");
    //obtiene la dirección IP del cliente por medio de ServletRequest
    HttpServletRequest req = (HttpServletRequest) FacesContext.getCurrentInstance().getExternalContext().getRequest();
    ip = req.getRemoteAddr();
    //Construye el objeto auditoria
    auditoria.setFechaAccion(new java.util.Date());
    auditoria.setIpAcceso(ip);
    auditoria.setIdUsuario(usuario.getIdUsuario());
    //forma la query de insert
    query = "INSERT INTO alprosart.auditoria VALUES(nextval('alprosart.auditoria_id_auditoria_seq'),"+auditoria.getIdUsua
    coneccionBDD = new ConeccionBDD();
    connection = coneccionBDD.obtenerConeccion();
    try {
        preparedStatement = connection.prepareStatement(query);
        preparedStatement.execute();
        preparedStatement.close();
        connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Figura 4.12: Método *insertarAuditoria*
Fuente: El Autor

II. Clase *Authenticator*

La clase *Authenticator* proporciona los métodos necesarios para iniciar y cerrar sesión en el sistema. El método más importante de esta clase es:

Método *authenticate()*

Este método retorna un valor tipo *boolean* el cual indica si se autenticó correctamente el usuario en el sistema.

El proceso que realiza el método es el siguiente:

Consulta de la tabla *ALPROSART.usuario* si existen registros con el usuario y clave proporcionada.

En el caso de que existan registros, obtiene los resultados y los asigna a una variable tipo *Usuario*.

Luego verifica si el estado del usuario es *Activo* o *Inactivo*, en el caso que sea inactivo retorna *false* e indica al empleado que su usuario se encuentra inactivo. En el caso de que se encuentre activo selecciona la dirección de todas las páginas a las que tiene acceso en la tabla *ALPOSART.menu_perfil*, de acuerdo el perfil que esté asignado al usuario.

Luego se verifica si el perfil del usuario es administrador o empleado, en el caso de que sea administrador asigna a una variable tipo *boolean* llamada *administrador* el valor de *true*, caso contrario *false*.

Una vez obtenidos todos los parámetros mencionados anteriormente se los inserta en la sesión actual del usuario, estos valores servirán para validaciones y habilitar opciones.

En la *Figura 4.13* se muestra el código fuente completo de la función.

```

public boolean authenticate() {
    transaccionesBDD = new TransaccionesBDD();
    usuario = new Usuario();
    recursos = new ArrayList<String>();
    try {
        // selecciona los datos del usuario con los datos proporcionados en
        // la pantalla de login
        query = "SELECT * FROM alprostart.usuario u WHERE u.nombre_usuario = '"
            .concat(credentials.getUsername())
            .concat("' AND u.clave = '"
            .concat(SHA1.getHash(credentials.getPassword()))
            .concat("'";
        resultSet = transaccionesBDD.consultar(query);
        if (resultSet.next()) {
            usuario.setIdUsuario(resultSet.getInt(1));
            usuario.setIdEmpleado(resultSet.getInt(2));
            usuario.setIdEstado(resultSet.getInt(3));
            usuario.setIdPerfil(resultSet.getInt(4));
            usuario.setNombreUsuario(resultSet.getString(5));
            usuario.setClave(resultSet.getString(6));
            // verifica si el usuario se encuentra activo
            if (usuario.getIdEstado() == Constantes.ESTADO_INACTIVO) {
                falloLogueo(Constantes.USUARIO_INACTIVO);
                return false;
            }
            // selecciona la url de las páginas a las que tiene acceso el
            // usuario
            query = "SELECT m.url FROM alprostart.menu m, alprostart.menu_perfil mp WHERE m.id_menu = mp.id_menu AND mp.id_pe
                .concat(usuario.getIdPerfil().toString());
            resultSet = transaccionesBDD.consultar(query);
            while (resultSet.next()) {
                recursos.add(resultSet.getString(1));
            }
            // verifica si el usuario tiene perfil de administrador o
            // empleado
            if (usuario.getIdPerfil() == Constantes.PERFIL_ADMINISTRADOR) {
                administrador = true;
            } else if (usuario.getIdPerfil() == Constantes.PERFIL_EMPLEADO) {
                administrador = false;
            }
            // inserta variables en la sesión actual del usuario
            UtilsWeb.getSession().setAttribute("administrador",
                administrador);
            UtilsWeb.getSession().setAttribute("usuario", usuario);
            UtilsWeb.getSession().setAttribute("recursos", recursos);
            return true;
        }
        falloLogueo(Constantes.USUARIO_CLAVE_INCORRECTA);
        return false;
        // inserta variables en la sesión actual del usuario
        UtilsWeb.getSession().setAttribute("administrador",
            administrador);
        UtilsWeb.getSession().setAttribute("usuario", usuario);
        UtilsWeb.getSession().setAttribute("recursos", recursos);
        return true;
    }
    falloLogueo(Constantes.USUARIO_CLAVE_INCORRECTA);
    return false;
} catch (SQLException e) {
    e.printStackTrace();
}
return false;
}

```

Figura 4.13: Método *authenticate()*

Fuente: El Autor

III. Clase *huellaManager*

La clase *huellaManager* tiene los métodos necesarios para registrar y verificar las huellas dactilares, estos métodos son proporcionados por el SDK¹⁸ de DigitalPersona

¹⁸ SDK: Software Development Kit, son un conjunto de herramientas de desarrollo de software.

y fueron modificados para que muestren mensajes al usuario en español y para trabajar con la base de datos APROSART en lugar de archivos planos.

Método registrar()

La funcionalidad del método registrar es la de almacenar la información de una huella dactilar en la base de datos.

Inicialmente se crean variables las cuales sirven para especificar el nombre del dedo de donde se van a extraer las características.

Una vez creadas estas variables se obtiene la muestra de la huella dactilar llamando al método *getSample()* el cual retorna un objeto tipo *Sample* propio del SDK.

Para extraer las características de la muestra se usa el método *createFeatureSet* del objeto *DPFPFeatureExtraction*.

Luego se procede a crear la plantilla de la huella dactilar obtenida con el método *getTemplate()*.

Para almacenar esta información en la base de datos, se procede a serializar la plantilla obtenida para luego asignar esta información a un arreglo de bytes con el resto de la información necesaria para guardarla en la tabla *ALPROSART.huella*.

Para comprender de mejor manera el método *registrar()* en la *Figura 4.14* se muestra el código fuente completo.

```

public void registrar() {
    try {
        facesMessages.clear();
        DPFPFingerIndex finger = DPFPFingerIndex.values()[idDedo];
        DPFPFeatureExtraction featureExtractor = DPFPGlobal
            .getFeatureExtractionFactory().createFeatureExtraction();
        DPFPEnrollment enrollment = DPFPGlobal.getEnrollmentFactory()
            .createEnrollment();

        while (enrollment.getFeaturesNeeded() > 0) {
            DPFPsample sample = getSample();
            if (sample == null)
                continue;

            DPFPFeatureSet featureSet;
            try {
                featureSet = featureExtractor.createFeatureSet(sample,
                    DPFPDataPurpose.DATA_PURPOSE_ENROLLMENT);
            } catch (DPFPImageQualityException e) {
                facesMessages.add("La calidad de la imagen es mala".concat(
                    e.getCaptureFeedback().toString()).concat(
                    ". Inténtelo nuevamente."));
                continue;
            }

            enrollment.addFeatures(featureSet);
        }

        DPFPTemplate template = enrollment.getTemplate();
        datosHuella = template.serialize();
        boolean resultado;
        transaccionesBDD = new TransaccionesBDD();
        resultado = transaccionesBDD.guardarHuella(datosHuella, empleado.getIdEmpleado(), idDedo);
        if(resultado){
            facesMessages.add("La ".concat(fingerprintName(finger)).concat(
                " ha sido registrado correctamente"));
        }else{
            facesMessages.add("No se ha podido ingresar la huella en la base de datos");
        }

    } catch (DPFPImageQualityException e) {
        facesMessages.add("El registro de la huella ha fallado");
    } catch (InterruptedException e) {
        throw new RuntimeException(e);
    }
}

```

Figura 4.14: Método registrar()

Fuente: El Autor

Método verificar()

El método verificar se lo utiliza para marcar la entrada y salida de los empleados, el procedimiento que se realiza es el siguiente, cabe recalcar que para realizar la verificación el usuario debe estar autenticado en el sistema y sus huellas dactilares deben estar registradas.

Inicialmente, se obtiene el código del empleado a partir de la variable de sesión *usuario*, con este código se realiza la consulta de todas las huellas dactilares que el empleado tenga registradas.

Luego se obtiene la muestra de la huella dactilar con el método *getSample()*.

Para realizar la verificación se crean instancias de los objetos *DPFPFeatureExtraction*, *DPFPFeatureSet*, *DPDPVerification*. En este último objeto se ajusta el nivel de verificación de acuerdo a las necesidades, para este proyecto se estableció un nivel medio de verificación.

Finalmente se compara uno a uno, el resultado obtenido de la consulta a la base de datos contra la huella obtenida para la verificación hasta encontrar alguna coincidencia, este proceso se realiza con el método *verify()*.

La *Figura 4.15* muestra el código fuente completo del método *verificar()*.

```

public boolean verificar() {
    transaccionesBDD = new TransaccionesBDD();
    Usuario usuario = new Usuario();
    usuario = (Usuario) UtilsWeb.getSession().getAttribute("usuario");
    query = "SELECT h.huella FROM alprosart.huella h WHERE h.id_empleado="+ usuario.getIdEmpleado();
    resultSet = transaccionesBDD.consultar(query);
    try {
        DPFPSSample sample = getSample();
        if (sample == null){
            throw new Exception();
        }
        DPFPFeatureExtraction featureExtractor = DPFPGlobal.getFeatureExtractionFactory().createFeatureExtraction();
        DPFPFeatureSet featureSet = featureExtractor.createFeatureSet(sample, DPFPDataPurpose.DATA_PURPOSE_VERIFICATION);
        DPFPVerification matcher = DPFPGlobal.getVerificationFactory().createVerification();
        matcher.setFARRequested(DPFPVerification.MEDIUM_SECURITY_FAR);
        while(resultSet.next()){
            datosHuella = resultSet.getBytes(1);
            DPFPTemplate template = DPFPGlobal.getTemplateFactory().createTemplate(datosHuella);
            if (template != null) {
                DPFPVerificationResult result = matcher.verify(featureSet,template);
                if (result.isVerified()) {
                    facesMessages.add(Severity.INFO, "Identidad verificada exitosamente.");
                    return true;
                }
            }
        }
        facesMessages.add(Severity.ERROR, "No se ha podido verificar la identidad");
        return false;
    } catch (Exception e) {
        facesMessages.add(Severity.ERROR, "Error al realizar la verificación");
        return false;
    }
}

} catch (DPFPImageQualityException e) {
    facesMessages.add("El registro de la huella ha fallado");
} catch (InterruptedException e) {
    throw new RuntimeException(e);
}
}
}

```

Figura 4.15: Método verificar()
Fuente: El Autor

4.3 FORMULARIOS DEL SISTEMA

- Formulario Principal (template.xhtml)

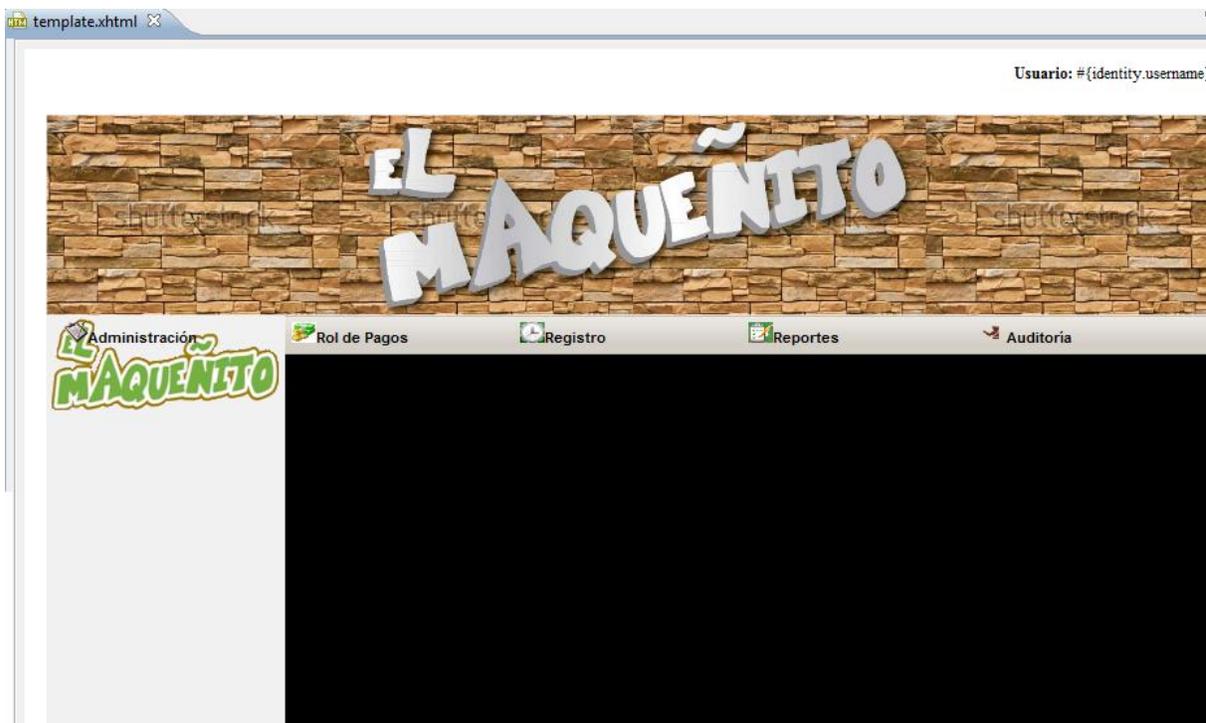


Figura 4.16: Formulario Principal
Fuente: El Autor

El formulario principal que se muestra en la *figura 4.16* consta de los siguientes controles:

Cabecera: es un control tipo *h:panelGroup*. En este se ubica en la parte superior derecha un control *h:outputText* donde se coloca el nombre del usuario autenticado. Y una imagen que corresponde a un logo de la empresa.

Menú Principal: El menú principal está formado por un componente *h:panelGroup* en este panel se incluye una página donde se encuentran todos los controles del menú principal (*menú.xhtml*). Las opciones del menú se presentan al usuario dependiendo el perfil asignado esto se controla con la propiedad *rendered* de cada control, en la *Figura 4.17* se muestra el uso de esta propiedad.

```

<rich:menuGroup value="Turnos" rendered="#{administrador}">
  <rich:menuItem submitMode="ajax" value="Nuevo Turno" action="#{turnoManager.nuevoTurno()}" >

  </rich:menuItem>
  <rich:menuItem submitMode="ajax" value="Modificar Turno" action="#{turnoManager.modificarTurno()}">

  </rich:menuItem>
</rich:menuGroup>

```

Figura 4.17: Menú Turnos
Fuente: El Autor

Panel lateral: en el panel lateral se ubica la imagen del logo de la empresa en tamaño pequeño.

Contenedor Central: aquí se despliegan todas las páginas del sistema (*body*), en la Figura 4.16 es representado por el área de color negro.

- **Formulario para crear nuevos usuarios (Usuario.xhtml)**

Figura 4.18: Formulario de creación de usuarios
Fuente: El Autor

En el formulario de creación de usuarios se muestra la mayoría de componentes que son presentados generalmente en el resto de formularios del sistema, en la *Figura 4.18* se muestran estos componentes y se detalla en qué consiste cada uno:

1. *h:selectOneMenu*: este componente muestra una lista desplegable con varias opciones de las que sólo se puede escoger una.
2. *rich:message*: muestra mensajes de validación asignado para un componente específico.
3. *h:inputText*: es un cuadro de texto donde se ingresa cualquier valor el cual puede ser validado o no.
4. *h:inputSecret*: cuadro de texto utilizado generalmente para el ingreso de contraseñas.
5. *rich:dataTable*: tabla que muestra un grupo de datos.
6. *rich:dataScroller*: componente que sirve para navegar entre las páginas de un *rich:dataTable*.
7. *rich:messages*: muestra mensajes de validación o de error en general.
8. *h:commandbutton*: botón que es usado para ejecutar una acción en particular.
9. *h:outputText*: etiqueta que muestra un texto determinado.

CAPÍTULO 5

PRUEBAS E IMPLEMENTACIÓN

5.1 PRUEBAS

Una vez concluido el desarrollo del sistema se lo presentó a las autoridades de ALPROSART junto con los jefes de cada área previa a la implementación del sistema en el ambiente de producción.

Luego se realizó el ingreso de los datos iniciales para el uso del sistema como son: cargos, departamentos, estados, estado civil, menú, menú por perfil, perfil, sexos, turnos y un empleado con su respectivo usuario administrador.

Para comprobar el correcto funcionamiento del sistema se realizaron distintos casos de prueba, los resultados de los procesos más importantes del sistema se presentan en las siguientes tablas, el resto de casos de pruebas se encuentran en el Anexo IV.

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que solamente se puede ingresar al sistema con un usuario y clave válidos	El usuario debe estar creado	usuario= vmancheno , clave= incorrecta	El sistema deniega el acceso	Correcto
		usuario= incorrecto , clave= 1234	El sistema deniega el acceso	Correcto
		usuario= inactivo , clave= 1234	El sistema deniega el acceso	Correcto
		usuario= vmancheno , clave= 1234	El sistema permite el acceso	Correcto

*Tabla 5.1: Caso de prueba de inicio de sesión
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que solamente se puede reasignar la contraseña con un usuario y número de cédula válidos	El usuario debe estar creado	usuario= vmancheno , identificación= incorrecta	El sistema muestra mensaje de error	Correcto
		usuario= incorrecto , identificación= 1721837779	El sistema muestra mensaje de error	Correcto
		usuario= incorrecto , identificación= incorrecta	El sistema muestra mensaje de error	Correcto
		usuario= vmancheno , identificación= 1721837779	El sistema muestra mensaje de envío de correo con nueva clave	Correcto

*Tabla 5.2: Caso de prueba de reasignación de clave
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede crear un usuario	El usuario debe estar autenticado con perfil de administrador	Sin datos	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Datos sin formato, ejemplo correos, fechas, número de cargas familiares	El sistema muestra mensaje de error indicando que el formato es incorrecto	Correcto
		Identificación duplicada	El sistema muestra mensaje de error indicando que la identificación ya existe	Correcto
		Datos correctos	El sistema ingresa el empleado y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede modificar un empleado	El empleado debe estar creado	Sin datos	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Datos sin formato, ejemplo correos, fechas, número de cargas familiares	El sistema muestra mensaje de error indicando que el formato es incorrecto	Correcto
		Identificación duplicada	El sistema muestra mensaje de error indicando que la identificación ya existe	Correcto

		Datos correctos	El sistema modifica los datos y redirige a página de modificación exitosa	Correcto
--	--	-----------------	---	----------

*Tabla 5.3: Caso de prueba de administración de empleados
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede registrar las huellas dactilares de un empleado	El empleado debe estar creado, el lector biométrico debe estar conectado	Empleado= vacío, Nombre dedo= vacío	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Verónica Mancheno, Nombre dedo= Pulgar Derecho, Calidad de imagen= mala	El sistema muestra mensaje de error de mala calidad de la imagen	Correcto
		Empleado=Verónica Mancheno, Nombre dedo= Pulgar Derecho, Calidad de imagen= buena	El sistema registra la huella y muestra mensaje de información de registro exitoso	Correcto
		Empleado=Verónica Mancheno, Nombre dedo= Pulgar Derecho, Calidad de imagen= buena	El sistema muestra mensaje de error de huella ya registrada	Correcto

*Tabla 5.4: Caso de prueba de registro de huella dactilar
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el empleado puede registrar su entrada/ salida por medio del lector biométrico	El empleado debe estar autenticado en el sistema, el lector biométrico debe estar conectado	Huella dactilar= no registrada	El sistema muestra mensaje de error indicando que no se pudo verificar la identidad	Correcto
		Huella dactilar= huella de empleado diferente al autenticado	El sistema muestra mensaje de error indicando que no se pudo verificar la identidad	Correcto
		Huella dactilar= correcta	El sistema muestra un formulario con datos personales del empleado	Correcto

*Tabla 5.5: Caso de prueba de registro de entrada/salida por dispositivo biométrico
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador pueda generar un nuevo rol de pago	El usuario debe estar autenticado	Empleado=no seleccionado, año=no seleccionado, mes=no seleccionado	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, año=2012, mes=Enero	El sistema genera el rol de pagos y muestra la información generada con la opción de guardar.	Correcto
		Empleado=Veronica Mancheno, año=2012, mes=Enero	El sistema muestra mensaje de error indicando que ya existe un rol de pagos generado para el período y empleado seleccionado	Correcto

Tabla 5.6: Caso de prueba de generación de rol de pagos

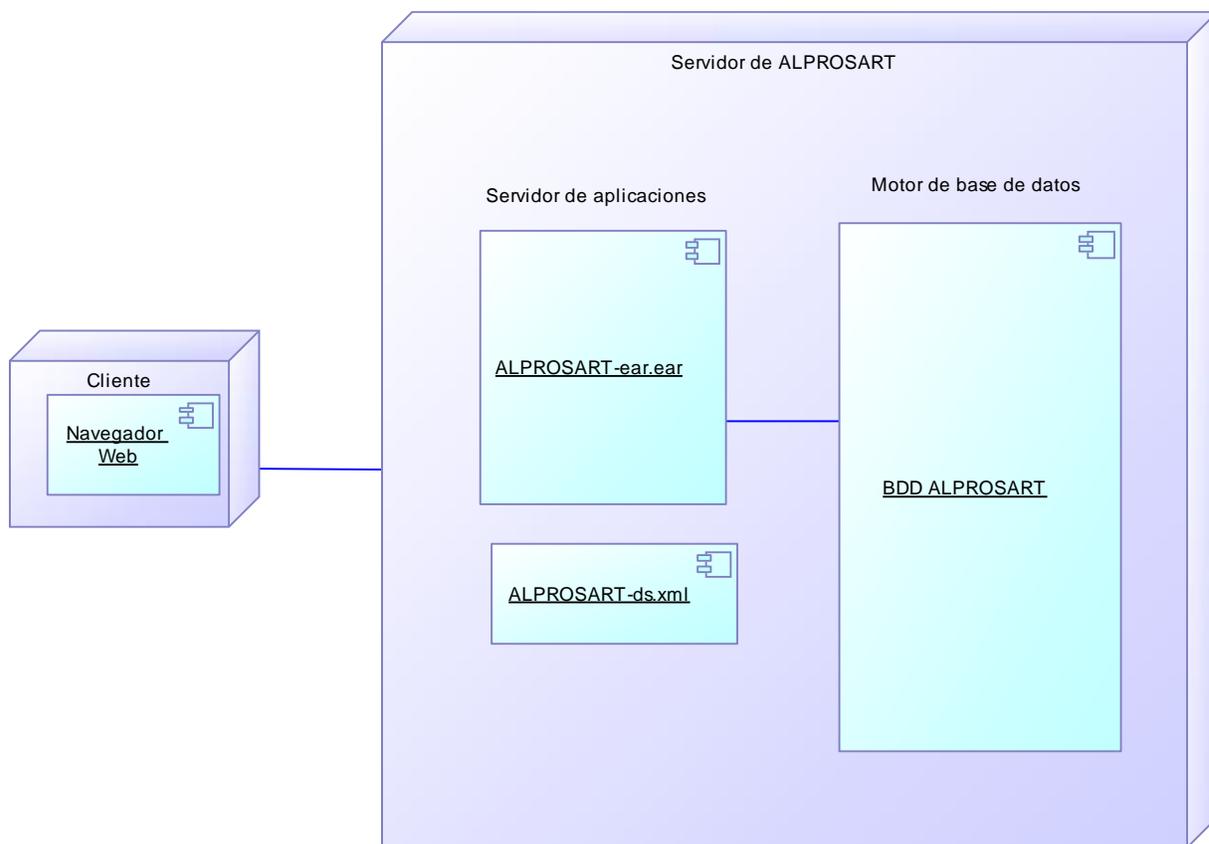
Fuente: El Autor

5.2 IMPLEMENTACIÓN

Al finalizar el sistema y realizar las correcciones de los errores encontrados en el sistema, se procedió con la implementación en el ambiente de producción de ALPROSART. El procedimiento que se realizó fue el siguiente:

- Instalar el JDK, el servidor de base de datos PostgreSQL y el servidor de aplicaciones JBoss.
- Crear la base de datos ALPROSART, el usuario propietario de la base de datos y el usuario transaccional *alprosart_t* que es el que accede a la base de datos desde el sistema desarrollado.
- Establecer la dirección del sistema como página de inicio en el explorador de internet.
- Crear el directorio para los respaldos de la base de datos.
- Capacitar al personal para el acceso a la aplicación.

En el Anexo V se encuentra el Manual de Instalación donde se explica de manera detallada el proceso de instalación del sistema para su correcto funcionamiento.



*Figura 5.1: Diagrama de Implementación
Fuente: El Autor*

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

Luego de la finalización del proyecto se llegaron a las siguientes conclusiones y recomendaciones

6.1 CONCLUSIONES.

- i. Para la fase del levantamiento de información, el uso de entrevistas y cuestionarios permitieron conocer más a fondo los procesos que realiza la empresa para el registro de asistencia del personal y generación de los roles de pago de los empleados. Esta información fue útil para el diseño adecuado del sistema con el fin de mejorar los procesos.
- ii. El uso del estándar IEEE 830 ayudó a mantener un acuerdo mutuo entre el cliente y el desarrollador (tesista), sobre los requerimientos del sistema que se desarrolló. Además fue de gran ayuda al momento de desarrollar ya que se tenía un panorama claro de qué es lo que el cliente deseaba.
- iii. La constante comunicación con los usuarios del sistema durante el desarrollo del mismo ayudó a obtener información de vital importancia para la elaboración de un producto que satisfaga sus necesidades.
- iv. El dispositivo biométrico de DigitalPersona U.are.U 4500 posee una alta capacidad de capturar imágenes sin distorsión y en un tiempo mínimo de respuesta; gracias a las librerías que incluye su SDK, lo cual ayudó al registro de las horas de entrada – salida del personal de ALPROSART y evitando que exista suplantación de identidad. Obteniendo como resultado un menor índice de atrasos desde su implementación.
- v. EL SDK de DigitalPersona es uno de los pocos que existen de libre distribución y con la suficiente documentación para facilitar el desarrollo de aplicaciones; además al momento de instalarlo se instalan ejemplos los cuales sirvieron de base para este proyecto.

- vi. Tener información de todos los empleados centralizada en una base de datos, facilita el manejo y visualización de la misma, cuando sea necesario.
- vii. Con la implementación de un lector biométrico para el registro de la entrada y salida de los empleados se redujo el porcentaje de atrasos de los empleados.
- viii. Las multas por mora de los pagos que debe realizar la empresa se redujeron considerablemente ya que el sistema desarrollado alertaba al empleado encargado de realizar estos pagos desde dos días antes de que se venza la fecha de pago.
- ix. La problemática de realizar los roles de pago manualmente se solucionó ya que el sistema lo genera automáticamente, tomando en cuenta los valores de ingresos y egresos del empleado y detallándolos en el mismo rol. Con esto también se redujeron los reclamos por parte de los empleados que no sabían la forma en la que su remuneración era calculada.
- x. Por medio del reporte de atrasos y horas laboradas finalmente se pudo evaluar la asistencia y cumplimiento de los horarios de trabajo de los empleados.
- xi. El IDE de desarrollo Eclipse es muy potente ya que permitió crear un sistema funcional y de fácil manejo, además permite integrar una gran variedad de herramientas que ayudaron al desarrollo del proyecto como JasperReports y Seam Framework.
- xii. El motor de base de datos PostgreSQL fue la mejor opción para el almacenamiento de la información ya que se encontró toda la documentación necesaria para el desarrollo del sistema en la Web y además cuenta con casi todas las funcionalidades de un motor de base de datos con licencia.

6.2 RECOMENDACIONES

- i. Previo al uso del sistema se debe capacitar a todos los empleados de la empresa, sobre el uso del sistema y los conceptos básicos de la identificación biométrica para asegurar el correcto uso y aclarar las dudas del personal.
- ii. Para el correcto uso del sistema y de sus funcionalidades se recomienda usar Internet Explorer 8 ya que es compatible en su totalidad con los componentes de los formularios del sistema. Se recomienda no utilizar Internet Explorer 9 para acceder al sistema debido a que algunos de componentes de los formularios pierden funcionalidad con este navegador, otra alternativa es utilizar Mozilla Firefox 13 o superior.
- iii. Es necesario realizar una limpieza periódica del dispositivo biométrico debido a la humedad de los dedos y al polvo que se encuentra en el ambiente, la limpieza debe ser realizada con un paño seco y con el dispositivo apagado; lo que garantiza que siempre se obtenga una imagen de buena calidad.
- iv. Se recomienda que previo al uso del dispositivo biométrico los dedos de la mano estén limpios para evitar dificultades en la captura de la huella y en el proceso de verificación.
- v. Para obtener un modelo de base de datos óptimo es recomendable analizar detalladamente las relaciones que existen entre las tablas para evitar que existan registros repetidos en varias tablas.
- vi. El uso de la metodología RUP no es muy recomendable para proyectos grandes, en el caso del presente proyecto se recomendaría usar el Modelo en Cascada que se define como: *“Una secuencia de actividades, donde la estrategia principal es seguir el proceso del desarrollo de software hacia puntos de revisión bien definidos mediante entregas calendarizadas”*.¹⁹
- vii. Si se necesita adquirir un lector de huellas dactilares, se recomiendan los dispositivos biométricos de la empresa DigitalPersona, en este caso el lector UareU 4500, por las ventajas encontradas y porque funciona correctamente

¹⁹ Fuente de información: <http://elo-ge-ma.blogspot.com/2009/10/sintesis-modelos-del-proceso-del.html>

con el SDK que provee el mismo fabricante que es gratuito y evita la compra de licencias.

- viii. Se recomienda cambiar la clave del usuario periódicamente, para evitar accesos no autorizados al sistema.
- ix. Es recomendable para evitar cualquier tipo de demora en el desarrollo del proyecto y en la implantación del mismo, utilizar versiones estables y probadas de las herramientas y/o librerías que han de ser utilizadas en el proyecto.
- x. Para evitar problemas en la implantación del sistema, es recomendable, desarrollarlo en un entorno de ser posible igual o similar al entorno de producción.
- xi. Para futuros desarrollos sobre el sistema se recomienda migrar a JBoss EAP7, esta es la última versión disponible que cuenta con mayores funcionalidades de administración.

CAPÍTULO 7

ANEXOS

ANEXO I: ENTREVISTAS, CUESTIONARIOS Y TABULACIÓN DE RESULTADOS

ENTREVISTA AL GERENTE

1. ¿De qué forma se controlan actualmente las horas de entrada y salida de los empleados?

2. ¿Qué datos ingresa el personal para el registro de las horas de entrada y salida?

3. ¿Cuál es la jornada laboral de los empleados?

4. ¿Cuánto tiempo disponen los empleados para almorzar?

5. ¿La hora de entrada y salida del almuerzo, es registrado?

6. ¿Se comprueba de alguna forma que los empleados hayan registrado las horas de entrada y salida?

7. ¿Ha tenido quejas por parte de los empleados al momento de recibir su rol de pagos, debido a cálculos mal realizados u horas trabajadas no contabilizadas?

8. ¿La persona encargada de las obligaciones patronales y servicios básicos de la empresa, realiza el pago a tiempo?

9. ¿Le gustaría tener un mayor control de las horas de entrada y salida de sus empleados, así como evitar caer en mora por falta de pago de obligaciones patronales y servicios básicos, con ayuda de la tecnología?

10. ¿Está informado de los atrasos, multas y permisos que tiene cada empleado?

ENTREVISTA AL ADMINISTRADOR

1. ¿Cree usted que el registro de horas de entrada y salida actual es confiable?

2. ¿Al momento de realizar los roles de pago de los empleados, se le dificulta el cálculo de las horas que laboró cada empleado?

3. ¿Ha olvidado incluir multas, atrasos u otros valores que afecten el rol de pago de cada empleado?

4. ¿Cuándo la gerencia solicita un reporte de asistencia de los empleados, es generado a tiempo?

5. ¿Ha olvidado realizar el pago de las obligaciones patronales o servicios básicos de la empresa?

6. ¿Le gustaría recibir un recordatorio cada vez que deba cumplir un pago de obligaciones patronales o de servicios básicos de la empresa?

7. ¿Posee un registro histórico de las horas laboradas de cada empleado?

8. ¿Piensa usted que haciendo uso de la tecnología se puede mejorar el control de ingreso y salida de los empleados?

ENCUESTA AL PERSONAL DE PRODUCCIÓN Y DISTRIBUCIÓN

Cargo: _____

Edad: _____

Fecha: _____

OBJETIVO: El objetivo de la encuesta es recolectar la información necesaria para realizar un estudio sobre el proceso de registro de horas de entrada y salida del personal de “ALPROSART” así como la forma en la que se generan los roles de pagos, para automatizar y optimizar estos procesos. La información que se obtenga de la encuesta será confidencial para el uso del mencionado estudio.

Lea las preguntas detenidamente, y escoja solamente una opción por pregunta marcándola con una **X**.

1. ¿Está conforme con la forma en la que usted registra sus horas de entrada y salida?

SI

NO

2. ¿Cree que se puede mejorar este proceso de registro?

SI

NO

3. ¿Está conforme con la forma en que se realiza el cálculo de su rol de pago?

SI

NO

4. ¿Ha tenido dudas sobre si en realidad se han tomado en cuenta todas las horas laboradas para la emisión de su rol de pago?

SI

NO

5. ¿Le gustaría saber exactamente cuántas horas ha laborado mensualmente?

SI

NO

6. ¿Sabe usted qué es un lector de huellas dactilares?

SI

NO

7. ¿Estaría de acuerdo en usar un lector de huellas dactilares para el registro de sus horas de entrada y salida? (Responda esta pregunta sólo si respondió SI a la pregunta 6)

SI

NO

8. ¿Estaría de acuerdo en que se use un sistema informático para el cálculo de su rol de pagos?

SI

NO

ANEXO II: ESPECIFICACIÓN Y DIAGRAMAS DE CASOS DE USO

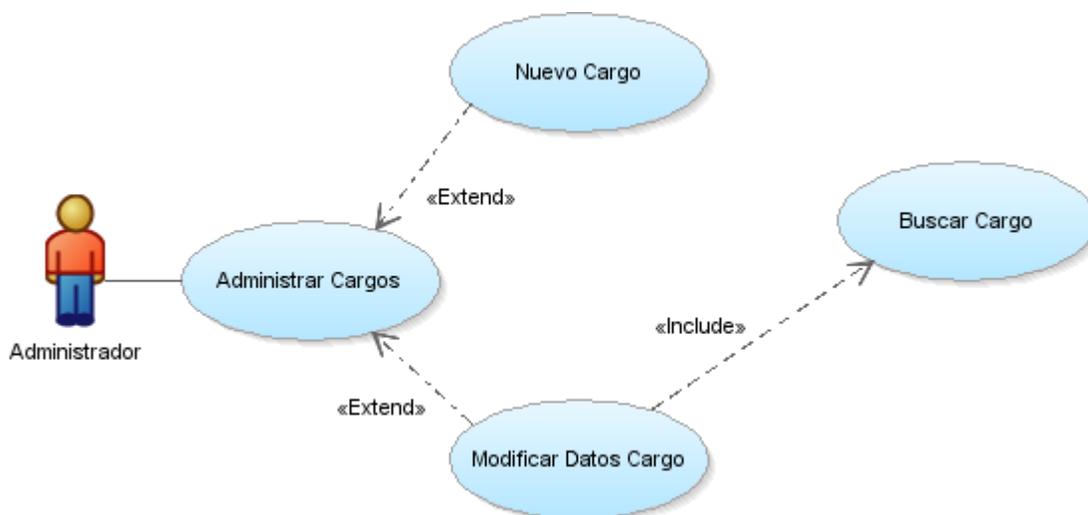
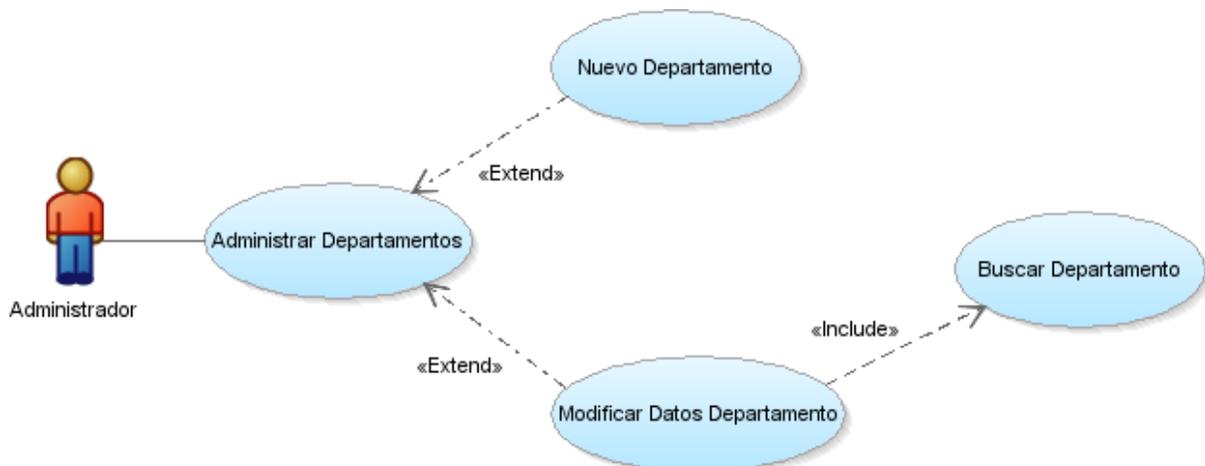


Figura 7.2.1: Diagrama de Caso de Uso: Administrar Cargos
Fuente: El Autor

CASO DE USO	Administrar Cargos
ACTORES	Administrador
PROPÓSITO	Ingresar los cargos que existen en la empresa, para asignarlos a cada empleado
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error el nombre del cargo
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Cargos", selecciona la opción "Nuevo Cargo" e ingresa todos los campos del formulario que presenta el sistema.
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Administración"	
Paso 2. El sistema muestra el módulo de "Administración"	
Paso 3. El administrador ingresa al sub módulo de "Cargos"	
Paso 4. El sistema muestra el sub módulo de "Cargos"	
Paso 5. El administrador selecciona la opción "Nuevo Cargo"	
Paso 6. El sistema muestra un formulario con los campos correspondientes a los datos de un cargo	

Paso 7. El administrador digita los datos correspondientes al nuevo cargo
Paso 8. El sistema valida y almacena la información ingresada
Paso 9. El administrador cierra el sub módulo de "Cargos"
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 5. El administrador selecciona la opción "Modificar Cargo"
Paso 8. El sistema muestra un mensaje de error porque el nombre del cargo ya existe en el sistema, vuelve Paso 7.

*Tabla 7.2.1: Caso de Uso Administrar Cargos
Fuente: El Autor*

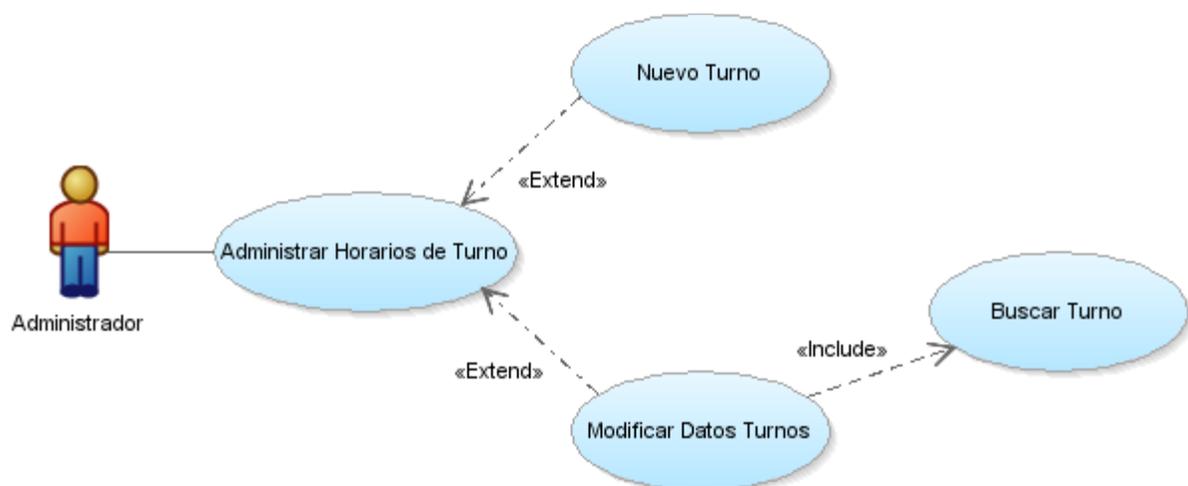


*Figura 7.2.2: Diagrama de Casos de Uso: Administrar Departamentos
Fuente: El Autor*

CASO DE USO	Administrar Departamentos
ACTORES	Administrador
PROPÓSITO	Ingresar los departamentos que existen en la empresa
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error el nombre del departamento
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Departamentos", selecciona la opción "Nuevo Departamento" e ingresa todos los campos del formulario que presenta el sistema.

CURSO NORMAL DE EVENTOS
Paso 1. El administrador ingresa al módulo de "Administración"
Paso 2. El sistema muestra el módulo de "Administración"
Paso 3. El administrador ingresa al sub módulo de "Departamentos"
Paso 4. El sistema muestra el sub módulo de "Departamentos"
Paso 5. El administrador selecciona la opción "Nuevo Departamento"
Paso 6. El sistema muestra un formulario con los campos correspondientes a los datos de un departamento
Paso 7. El administrador digita y/o selecciona la información correspondiente a un departamento
Paso 8. El sistema valida y almacena la información ingresada
Paso 9. El administrador cierra el sub módulo de "Departamentos"
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 5. El administrador selecciona la opción "Modificar Departamento"
Paso 8. El sistema muestra un mensaje de error porque el nombre del departamento ya existe en el sistema, vuelve Paso 7.

*Tabla 7.2.2: Caso de Uso Administrar Departamentos
Fuente: El Autor*



*Figura 7.2.3: Diagrama de Casos de Uso: Administrar Horarios de Turno
Fuente: El Autor*

CASO DE USO	Administrar Horarios de Turno
ACTORES	Administrador
PROPÓSITO	Registrar los horarios de turno para los diferentes grupos de empleados de la empresa
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los horarios a trabajar
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Turnos", ingresa los datos de los turnos de trabajo que maneja la empresa y la información se guarda correctamente.
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Administración"	
Paso 2. El sistema muestra el módulo de "Administración"	
Paso 3. El administrador ingresa al sub módulo de "Turnos"	
Paso 4. El sistema muestra el sub módulo de "Turnos"	
Paso 5. El administrador selecciona la opción "Nuevo Turno"	
Paso 6. El sistema muestra un formulario con los campos correspondientes a los datos de un turno.	
Paso 7. El administrador digita los datos correspondientes al nuevo turno	
Paso 8. El sistema valida y almacena la información ingresada	
Paso 9. El administrador cierra el sub módulo de "Turnos"	
CURSO ALTERNATIVO	
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.	
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.	
Paso 5. El administrador selecciona la opción "Modificar Datos Turno"	
Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 7.	

Tabla 7.2.3: Caso de Uso Administrar Horarios de Turno

Fuente: El Autor

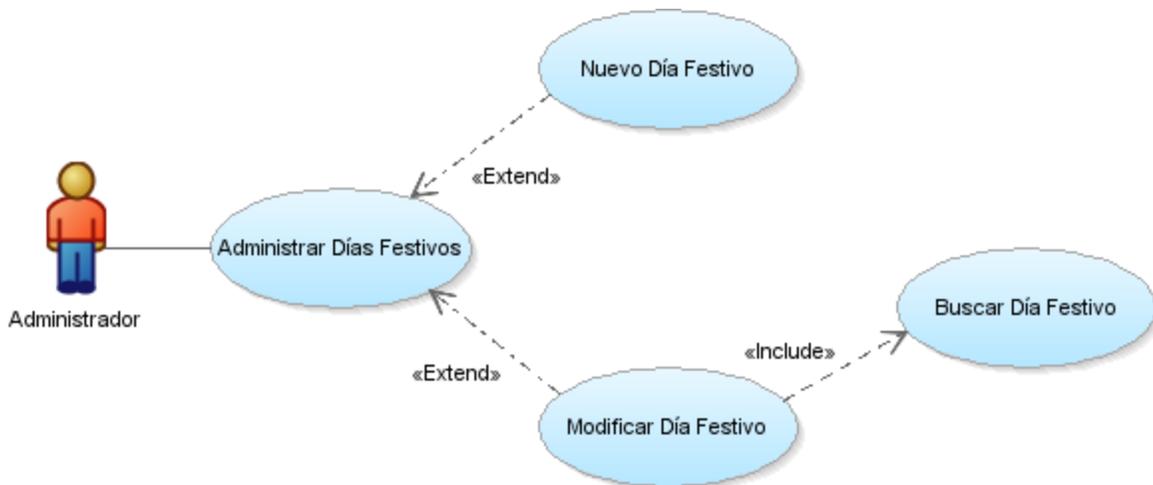


Figura 7.2.4: Diagrama de Casos de Uso: Administrar Días Festivos
Fuente: El Autor

CASO DE USO	Administrar Días Festivos
ACTORES	Administrador
PROPÓSITO	Registrar el nombre y seleccionar la fecha de los diferentes días considerados como festivos ya sea a nivel local, o nacional, para ser considerados como no laborables.
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los días festivos
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Días Festivos", selecciona la opción "Nuevo Día Festivo", ingresa el nombre del día festivo y selecciona la fecha en el calendario correspondiente, y se guarda correctamente la información
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Administración"	
Paso 2. El sistema muestra el módulo de "Administración"	
Paso 3. El administrador ingresa al sub módulo de "Días Festivos"	
Paso 4. El sistema muestra el sub módulo de "Días Festivos"	
Paso 5. El administrador selecciona la opción "Nuevo Día Festivo"	
Paso 6. El sistema muestra un formulario con los campos correspondientes a los datos de un día festivo	
Paso 7. El administrador digita el nombre del día festivo y selecciona la fecha	
Paso 8. El sistema valida y almacena la información ingresada	

Paso 9. El administrador cierra el sub módulo de "Días Festivos"
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 5. El administrador selecciona la opción "Modificar Día Festivo"
Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 7.

Tabla 7.2.4: Caso de Uso Administrar Días Festivos

Fuente: El Autor

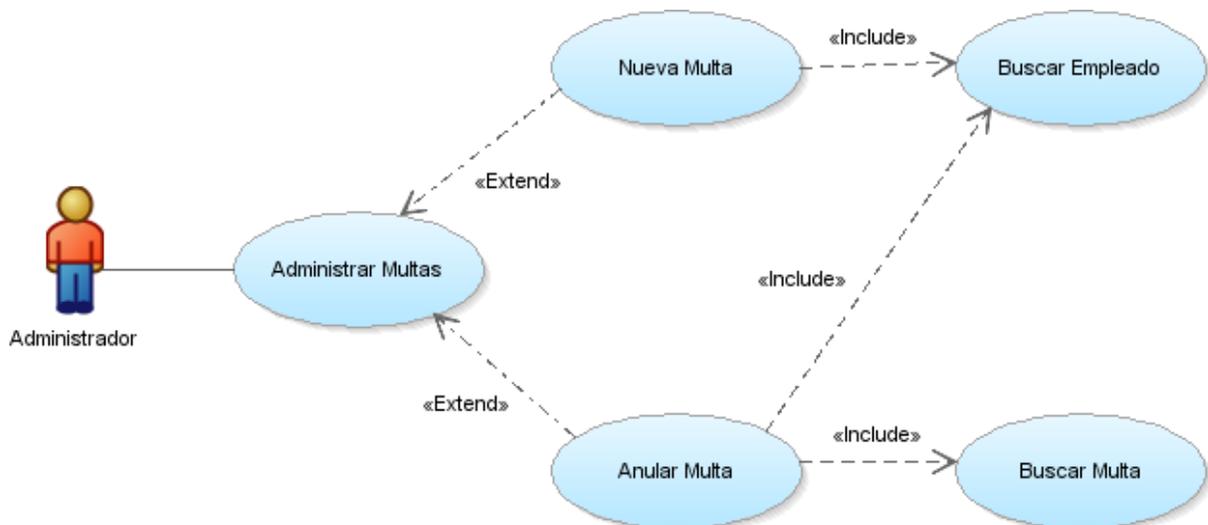


Figura 7.2.5: Diagrama de Casos de Uso: Administrar Multas

Fuente: El Autor

CASO DE USO	Administrar Multas
ACTORES	Administrador, Empleado
PROPÓSITO	Registrar las multas por faltas al reglamento de la empresa
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los datos de la multa ingresados

DESCRIPCIÓN	El administrador aplica una multa a un empleado, ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Multas", selecciona "Nueva Multa", selecciona al empleado, digita el motivo de la multa, el monto a descontar del sueldo y selecciona la fecha en la que el empleado cometió la falta, la información se guarda adecuadamente y se imprime correctamente una notificación de la multa y es entregada al empleado.
CURSO NORMAL DE EVENTOS	
Paso 1. El empleado comete una falta	
Paso 2. El administrador ingresa al módulo de "Administración"	
Paso 3. El sistema muestra el módulo de "Administración"	
Paso 4. El administrador ingresa al sub módulo de "Multas"	
Paso 5. El sistema muestra el sub módulo de "Multas"	
Paso 6. El administrador selecciona "Nueva Multa"	
Paso 7. El administrador selecciona el empleado, digita la descripción de la multa y digita el monto a ser descontado del sueldo del empleado	
Paso 8. El sistema valida y guarda la información	
Paso 9. El administrador selecciona la opción imprimir para entregar al empleado la notificación de la multa	
Paso 10. El sistema muestra la vista previa de impresión, y procede a imprimir	
Paso 11. El administrador obtiene la notificación, la firma y se la entrega al empleado	
Paso 12. El administrador cierra el sub módulo de "Multas"	
CURSO ALTERNATIVO	
Paso 3. El sistema no muestra el módulo correctamente, vuelve a Paso 2.	
Paso 5. El sistema no muestra el módulo correctamente, vuelve a Paso 4.	
Paso 6. El administrador selecciona "Anular Multa"	
Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 6	
Paso 10. No se imprime la notificación, vuelve Paso 9	
Paso 11. El documento es de mala calidad, vuelve Paso 9	

Tabla 7.2.5: Caso de Uso Administrar Multas

Fuente: El Autor

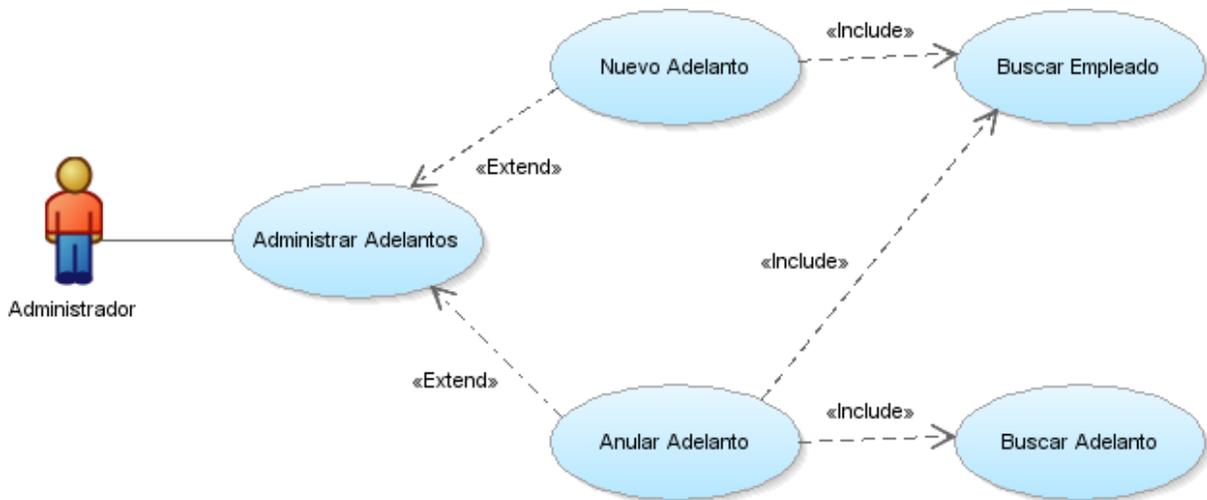


Figura 7.2.6: Diagrama de Casos de uso: Administrar Adelantos
Fuente: El Autor

CASO DE USO	Administrar Adelantos
ACTORES	Administrador, Empleado
PROPÓSITO	Registrar los adelantos de sueldo solicitados por los empleados a la empresa
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los datos del adelanto ingresado
DESCRIPCIÓN	El empleado solicita un adelanto al administrador, el administrador lo aprueba, ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Adelantos", selecciona "Nuevo Adelanto", selecciona al empleado, especifica los detalles del adelanto, el monto equivalente al adelanto, y el número de cuotas (meses) en las que se va a cancelar el mismo, la información se guarda adecuadamente y se imprime correctamente un comprobante del adelanto realizado y es entregado al empleado.
CURSO NORMAL DE EVENTOS	
Paso 1. El empleado solicita un adelanto	
Paso 2. El administrador aprueba el adelanto	
Paso 3. El administrador ingresa al módulo de "Administración"	
Paso 4. El sistema muestra el módulo de "Administración"	
Paso 5. El administrador ingresa al sub módulo de "Adelantos"	
Paso 6. El sistema muestra el sub módulo de "Adelantos"	

Paso 7. El administrador selecciona "Nuevo Adelanto"
Paso 8. El administrador selecciona el empleado, digita la descripción del adelanto, el monto equivalente al adelanto y el número de meses en los que va a ser dividido el adelanto para ser descontado del sueldo del empleado
Paso 9. El sistema valida y guarda la información
Paso 10. El administrador selecciona la opción imprimir para entregar al empleado la notificación de la multa
Paso 11. El sistema muestra la vista previa de impresión, y procede a imprimir
Paso 12. El administrador obtiene el comprobante del adelanto, la firma el administrador y el empleado, y este comprobante es archivado por el administrador
Paso 13. El administrador cierra el sub módulo de "Adelantos"
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 7. El administrador selecciona "Anular Adelanto"
Paso 9. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 8
Paso 11. No se imprime el comprobante, vuelve Paso 10
Paso 12. El documento es de mala calidad, vuelve Paso 10

Tabla 7.2.6: Caso de Uso Administrar Adelantos
Fuente: El Autor

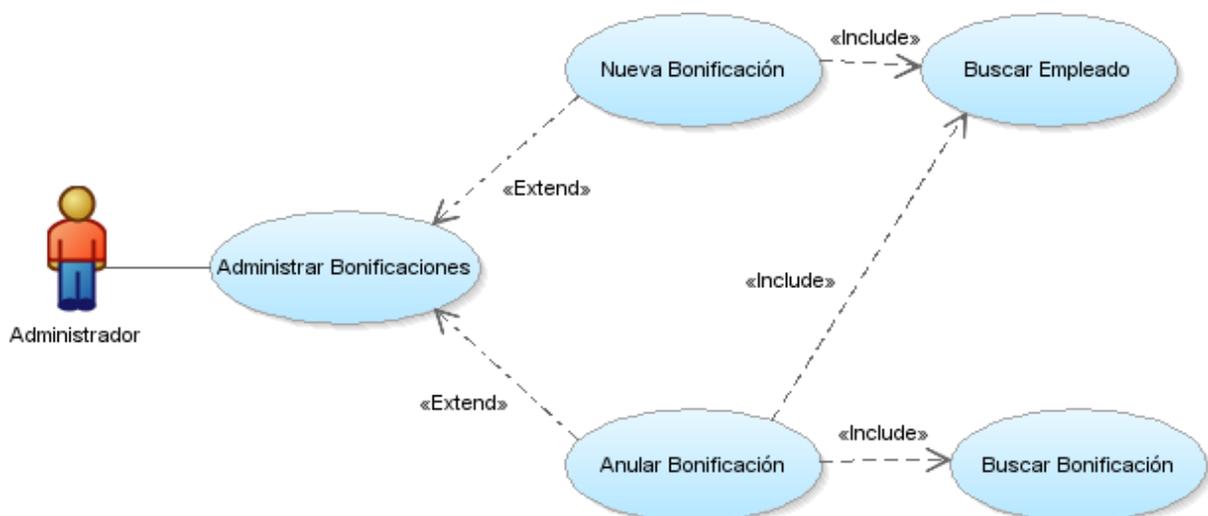


Figura 7.2.7: Diagrama de Casos de Uso: Administrar Bonificaciones
Fuente: El Autor

CASO DE USO	Administrar Bonificaciones
ACTORES	Administrador, Empleado
PROPÓSITO	Registrar las bonificaciones obtenidas por los empleados al realizar una tarea adicional
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los datos de la bonificación ingresada
DESCRIPCIÓN	El empleado realiza una tarea adicional solicitada por el administrador o el gerente, el administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Bonificaciones", selecciona la opción "Nueva Bonificación" selecciona al empleado, digita una descripción del porque de la bonificación y el monto equivalente a la bonificación y selecciona la fecha que se realizó la tarea adicional, la información se guarda adecuadamente y se genera un comprobante de ingreso para el empleado.
CURSO NORMAL DE EVENTOS	
Paso 1. El empleado realiza una tarea solicitada por una de las autoridades de la empresa	
Paso 2. El administrador ingresa al módulo de "Administración"	
Paso 3. El sistema muestra el módulo de "Administración"	
Paso 4. El administrador ingresa al sub módulo de "Bonificaciones"	
Paso 5. El sistema muestra el sub módulo de "Bonificaciones"	
Paso 6. El administrador selecciona "Nueva Bonificación"	
Paso 7. El administrador selecciona el empleado, digita la descripción de la bonificación, el monto equivalente a la bonificación y selecciona la fecha de la bonificación	
Paso 8. El sistema valida y guarda la información	
Paso 9. El administrador selecciona la opción imprimir para entregar al empleado un comprobante del ingreso de su bonificación	
Paso 10. El sistema muestra la vista previa de impresión, y procede a imprimir	
Paso 11. El administrador obtiene el comprobante de ingreso de bonificación, la firma el administrador y el empleado, y este comprobante es entregado al empleado.	
Paso 12. El administrador cierra el sub módulo de "Bonificaciones"	
CURSO ALTERNATIVO	
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.	
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.	
Paso 6. El administrador selecciona "Anular Bonificación"	

Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 7
Paso 10. No se imprime el comprobante, vuelve Paso 9
Paso 11. El documento impreso es de mala calidad, vuelve Paso 9

Tabla 7.2.7: Caso de Uso Administrar Bonificaciones
Fuente: El Autor

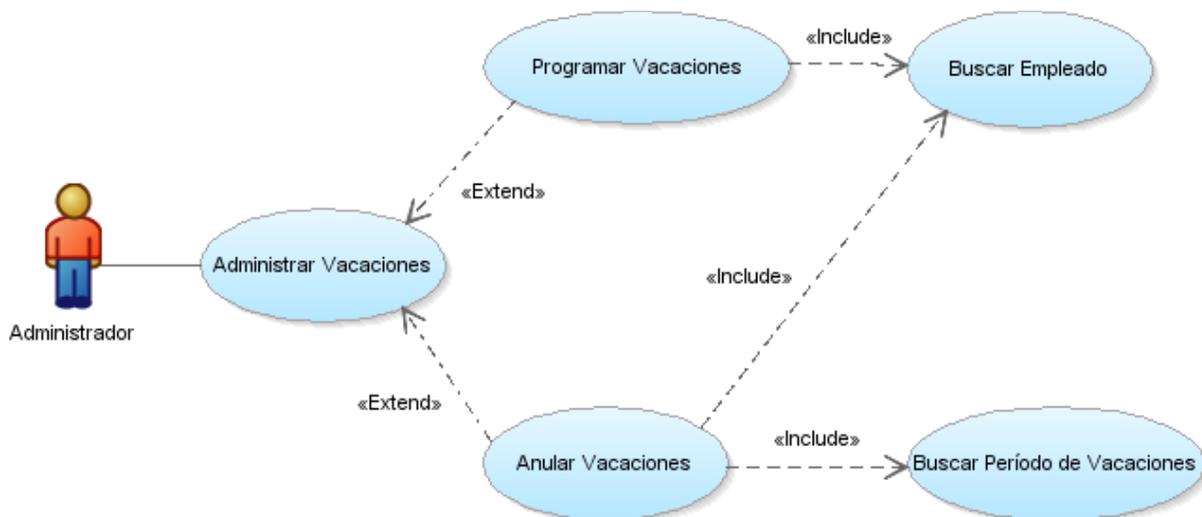


Figura 7.2.8: Diagrama de Casos de Uso: Administrar Vacaciones
Fuente: El Autor

CASO DE USO	Administrar Vacaciones
ACTORES	Administrador, Empleado
PROPÓSITO	Registrar los días de vacaciones que un empleado haya solicitado
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los datos de las vacaciones ingresados
DESCRIPCIÓN	El empleado solicita un tiempo de vacaciones al administrador, la solicitud es aprobada, el administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Vacaciones", selecciona la opción "Programar Vacaciones", selecciona al empleado, la fecha de inicio y fin de las mismas, luego esta información se guarda y se imprime de forma adecuada una autorización de vacaciones para que el administrador y el empleado lo firme.

CURSO NORMAL DE EVENTOS
Paso 1. El empleado solicita vacaciones
Paso 2. El administrador ingresa al módulo de "Administración"
Paso 3. El sistema muestra el módulo de "Administración"
Paso 4. El administrador ingresa al sub módulo de "Vacaciones"
Paso 5. El sistema muestra el sub módulo de "Vacaciones"
Paso 6. El administrador selecciona "Programar Vacaciones"
Paso 7. El administrador selecciona el empleado, la fecha de inicio y fin de las vacaciones, y digita una observación adicional si es necesario.
Paso 8. El sistema valida y guarda la información
Paso 9. El administrador selecciona la opción imprimir para generar una autorización de vacaciones
Paso 10. El sistema muestra la vista previa de impresión, y procede a imprimir
Paso 11. El administrador obtiene la autorización de vacaciones, y es firmada por el administrador y el empleado
Paso 12. El administrador cierra el sub módulo de "Vacaciones"
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 6. El administrador selecciona "Anular Vacaciones"
Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, o debido a que el empleado ya no dispone de vacaciones, vuelve Paso 7
Paso 10. No se imprime la autorización, vuelve Paso 9
Paso 11. El documento impreso es de mala calidad, vuelve Paso 9

Tabla 7.2.8: Caso de Uso Administrar Vacaciones

Fuente: El Autor

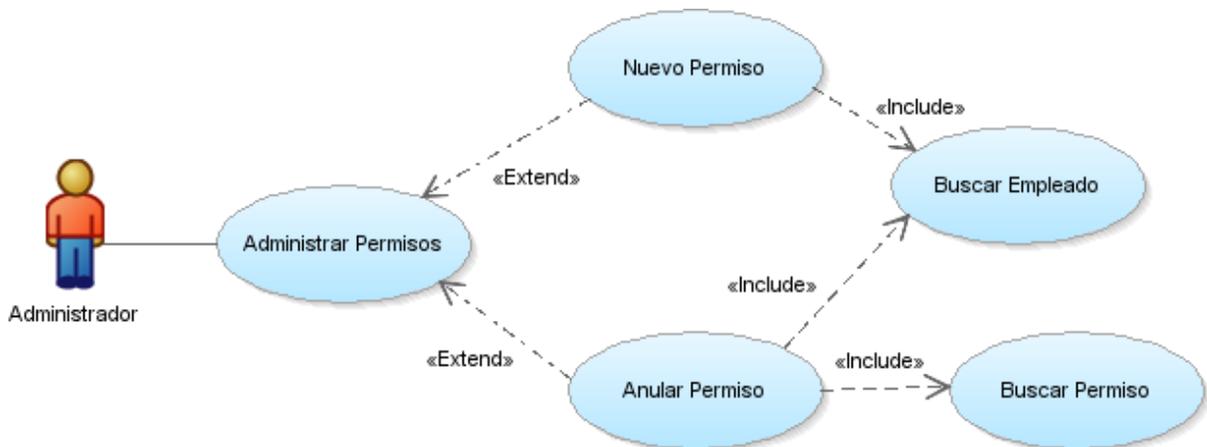


Figura 7.2.9: Diagrama de Casos de Uso: Administrar Permisos
Fuente: El Autor

CASO DE USO	Administrar Permisos
ACTORES	Administrador, Empleado
PROPÓSITO	Registra un permiso que un empleado haya solicitado ya sea para un día o por horas
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los datos de los permisos solicitados
DESCRIPCIÓN	El empleado solicita un permiso al administrador, el administrador se lo concede, el administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y después al sub módulo de "Permisos", luego a la opción "Nuevo Permiso", selecciona al empleado, selecciona el tipo de permiso, digita una descripción del permiso, selecciona las horas y/o días de permiso, luego la información se guarda adecuadamente y se imprime correctamente una autorización del permiso para que sea firmado por el administrador y el empleado
CURSO NORMAL DE EVENTOS	
Paso 1. El empleado solicita permiso	
Paso 2. El administrador ingresa al módulo de "Administración"	
Paso 3. El sistema muestra el módulo de "Administración"	
Paso 4. El administrador ingresa al sub módulo de "Permisos"	
Paso 5. El sistema muestra el sub módulo de "Permisos"	
Paso 6. El administrador selecciona la opción "Nuevo Permiso"	

Paso 7. El administrador selecciona el empleado, selecciona el tipo de permiso, digita la descripción del permiso, selecciona las horas y/o días de permiso
Paso 8. El sistema valida y guarda la información
Paso 9. El administrador selecciona la opción imprimir para generar una autorización de permiso
Paso 10. El sistema muestra la vista previa de impresión, y procede a imprimir
Paso 11. El administrador obtiene la autorización de permiso, y es firmada por el administrador y el empleado
Paso 12. El administrador cierra el sub módulo de "Permisos"
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 6. El administrador selecciona la opción "Anular Permiso"
Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 7
Paso 10. No se imprime la autorización, vuelve Paso 9
Paso 11. El documento impreso es de mala calidad, vuelve Paso 9

Tabla 7.2.9: Caso de Uso Administrar Permisos

Fuente: El Autor

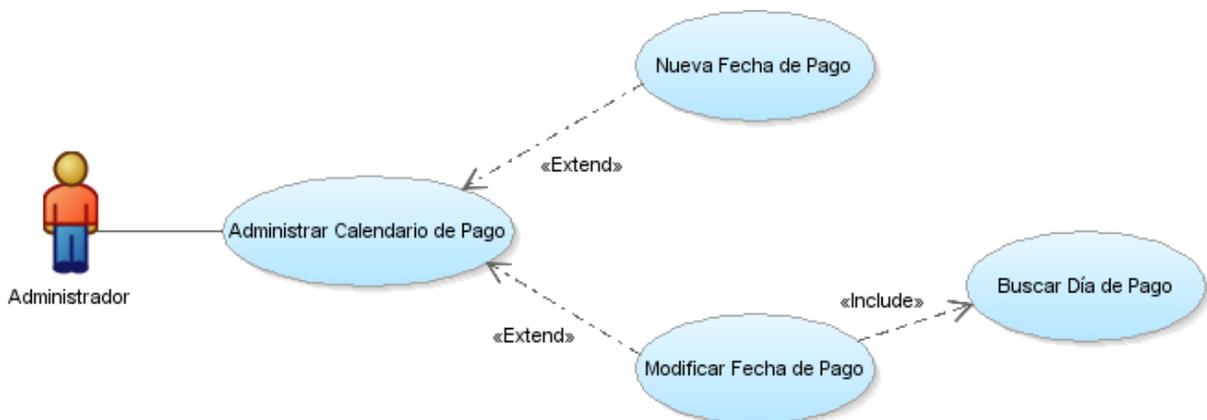


Figura 7.2.10: Diagrama de Casos de Uso: Administrar Calendario de Pago

Fuente: El Autor

CASO DE USO	Administrar Calendario de Pago
ACTORES	Administrador
PROPÓSITO	Registrar las fechas en las que el administrador deba realizar pagos por obligaciones patronales o de servicios básicos
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los datos de las fechas de pago de obligaciones patronales o servicios básicos
DESCRIPCIÓN	El administrador recibe una factura que debe ser pagada o a su vez revisa en la página web del IESS ²⁰ , que debe realizar un pago patronal. El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Calendario de Pagos", selecciona la opción "Nueva Fecha de Pago", digita la descripción del pago que se debe efectuar, digita el valor del pago y selecciona la fecha en que se debe realizar, y finalmente esta información se guarda adecuadamente
CURSO NORMAL DE EVENTOS	
Paso 1. El empleado solicita vacaciones	
Paso 2. El administrador ingresa al módulo de "Administración"	
Paso 3. El sistema muestra el módulo de "Administración"	
Paso 4. El administrador ingresa al sub módulo de "Calendario de Pagos"	
Paso 5. El sistema muestra el sub módulo de "Calendario de Pagos"	
Paso 6. El administrador selecciona la opción "Nueva Fecha de Pago"	
Paso 7. El administrador digita la descripción del pago, el valor del pago y selecciona la fecha en que se debe realizar el pago	
Paso 8. El sistema valida y guarda la información	
Paso 9. El administrador cierra el sub módulo de "Calendario de Pagos"	
CURSO ALTERNATIVO	
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.	
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.	
Paso 6. El administrador selecciona la opción "Modificar Fecha de Pago"	
Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 7	

Tabla 7.2.10: Caso de Uso Administrar Calendario de Pago

Fuente: El Autor

²⁰ IESS: Instituto Ecuatoriano de Seguridad Social

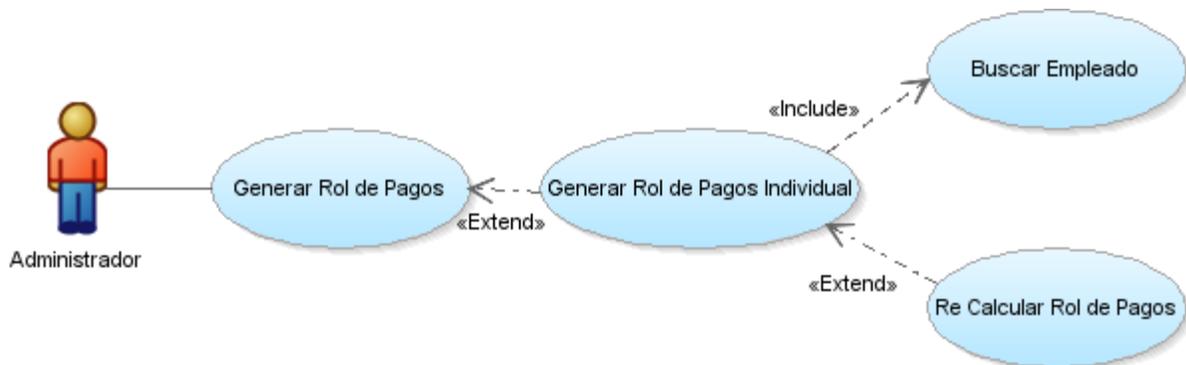


Figura 7.2.11: Diagrama de Casos de Uso: Generar Rol de Pagos
Fuente: El Autor

CASO DE USO	Generar Rol de Pagos
ACTORES	Administrador
PROPÓSITO	Generar el rol de pagos de un empleado correspondiente a un mes o varios meses
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se genera, presenta e imprime correctamente la información correspondiente al rol de pagos
DESCRIPCIÓN	El administrador desea generar el rol de pagos de un empleado ingresa al módulo de "Rol de Pagos", luego al sub módulo "Generar Rol de Pagos Individual", selecciona al empleado, y digita el período del cual se quiere generar el rol de pagos, se presenta la información solicitada y se imprime.
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Rol de Pagos"	
Paso 2. El sistema muestra el módulo de "Rol de Pagos"	
Paso 3. El administrador ingresa al sub módulo de "Generar Rol de Pagos Individual"	
Paso 4. El sistema muestra el sub módulo de "Generar Rol de Pagos Individual"	
Paso 5. El administrador selecciona al empleado y digita el período del cual se va a generar el rol de pago	
Paso 6. El sistema presenta el rol de pagos con los datos proporcionados	
Paso 7. El administrador selecciona la opción imprimir	
Paso 8. El sistema muestra la vista previa de impresión, y procede a imprimir	
Paso 9. El administrador obtiene el rol de pagos y se lo entrega al empleado correspondiente	

CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 6. El sistema encuentra un rol de pagos ya generado para el empleado seleccionado en el período indicado, se muestra la opción de “Re Calcular Rol de Pagos”
Paso 6. No existen resultados con la información proporcionada por el administrador, vuelve Paso 5.
Paso 8. No se imprime el rol de pagos, vuelve Paso 7
Paso 9. El documento impreso es de mala calidad, vuelve Paso 7

Tabla 7.2.11: Generar Rol de Pagos

Fuente: El Autor

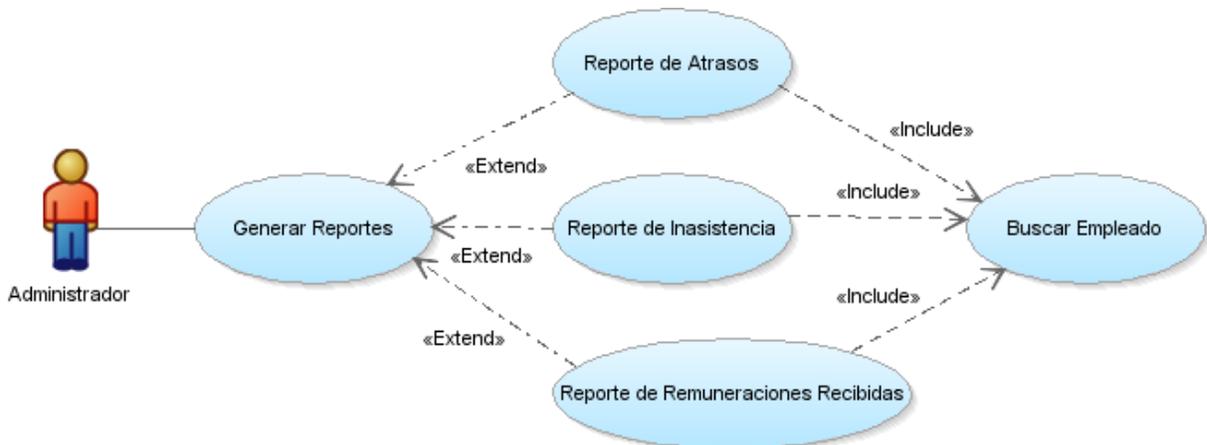


Figura 7.2.12: Diagrama de Casos de Uso: Generar Reportes

Fuente: El Autor

CASO DE USO	Generar Reportes
ACTORES	Administrador
PROPÓSITO	Generar un reporte de atrasos o remuneraciones recibidas de un empleado o de todos los empleados, correspondiente a un mes o varios meses
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se genera, presenta e imprime correctamente un reporte de atrasos

DESCRIPCIÓN	El administrador desea generar un reporte, ingresa al módulo de "Reportes", luego selecciona el tipo de reporte, selecciona al empleado o todos los empleados, y digita el período del cual se quiere generar el reporte, se presenta la información solicitada y se imprime.
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Reportes"	
Paso 2. El sistema muestra el módulo de "Reportes"	
Paso 3. El administrador selecciona el tipo de reporte	
Paso 4. El administrador selecciona al empleado o selecciona la opción "Todos" para incluir a todos los empleados en el reporte y digita el período del cual desea generar el reporte	
Paso 5. El sistema presenta el reporte con los datos proporcionados	
Paso 6. El administrador selecciona la opción imprimir	
Paso 7. El sistema muestra la vista previa de impresión, y procede a imprimir	
Paso 8. El administrador obtiene el reporte impreso	
Paso 9. El administrador cierra el sub módulo de "Reporte de Atrasos"	
CURSO ALTERNATIVO	
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.	
Paso 5. No existen resultados con la información proporcionada por el administrador, vuelve Paso 4.	
Paso 7. No se imprime el reporte, vuelve Paso 6	
Paso 8. El documento impreso es de mala calidad, vuelve Paso 6	

*Tabla 7.2.12: Caso de Uso Generar Reportes
Fuente: El Autor*

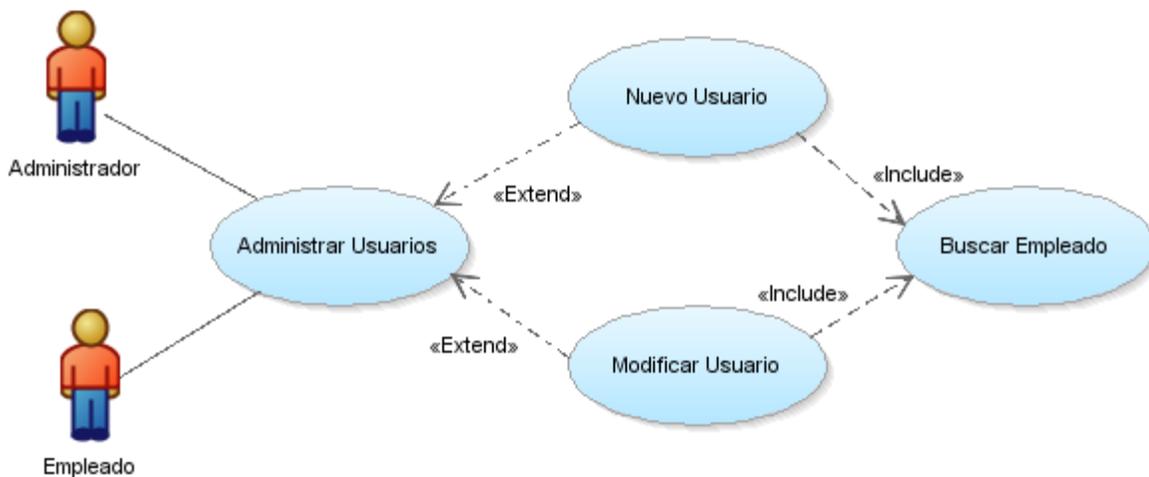


Figura 7.2.13: Diagrama de Casos de Uso: Administrar Usuarios
Fuente: El Autor

CASO DE USO	Administrar Usuarios
ACTORES	Administrador, Empleado
PROPÓSITO	Registrar los datos de un usuario
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Son registrados los datos del usuario del empleado sin ningún inconveniente
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de Administración y luego al sub módulo de "Usuarios", selecciona la opción "Nuevo Usuario", selecciona al empleado e ingresa todos los campos del formulario que presenta el sistema.
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Administración"	
Paso 2. El sistema muestra el módulo de "Administración"	
Paso 3. El administrador ingresa al sub módulo de "Usuarios"	
Paso 4. El sistema muestra el sub módulo de "Usuarios"	
Paso 5. El administrador selecciona la opción "Nuevo Usuario"	
Paso 6. El administrador selecciona al empleado que desea asignar un usuario	
Paso 7. El administrador digita el nombre del usuario, el perfil a ser asignado, el empleado digita su clave dos veces para verificar que sea correcta	
Paso 8. El sistema valida y almacena la información ingresada	
Paso 9. El administrador cierra el sub módulo de "Usuarios"	

CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.
Paso 5. El administrador selecciona la opción "Modificar Usuario"
Paso 8. El sistema muestra un mensaje de error porque el nombre de usuario ya existe en el sistema, o los datos ingresados son incorrectos, vuelve Paso 7.

Tabla 7.2.13: Caso de Uso Administrar Usuarios

Fuente: El Autor



Figura 7.2.14: Diagrama de Casos de Uso: Auditar Acciones de Usuarios

Fuente: El Autor

CASO DE USO	Auditar Acciones de Usuarios
ACTORES	Administrador
PROPÓSITO	Revisar todas las acciones realizadas por los usuarios en el sistema
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Son recuperados correctamente los datos de las acciones realizadas por los usuarios en el sistema
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de Auditoría y selecciona la fecha de inicio y fin de la auditoría a realizar, finalmente se muestran las acciones que han realizado los usuarios en el sistema en el período de tiempo seleccionado
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Auditoría"	
Paso 2. El sistema muestra el módulo de "Auditoría"	
Paso 3. El administrador selecciona la fecha de inicio y fin de la auditoría	

Paso 4. El sistema muestra las acciones que han realizado los usuarios en las fechas seleccionadas
Paso 5. El administrador cierra el módulo de Auditoría
CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve Paso 1
Paso 4. No existe información en el período seleccionado, vuelve Paso 3

Tabla 7.2.14: Caso de Uso Auditar Acciones de Usuarios
Fuente: El Autor

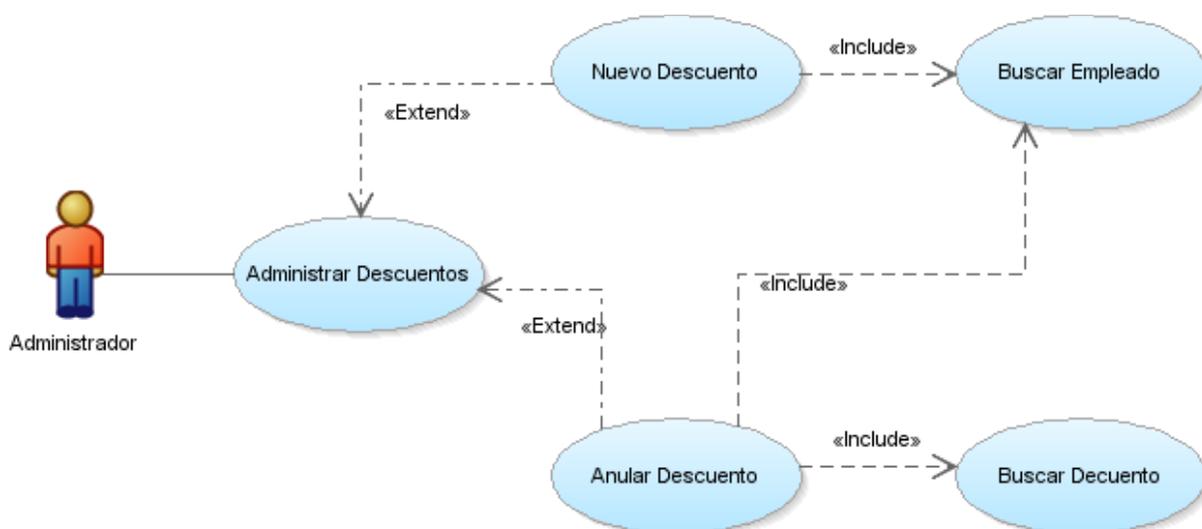


Figura 7.2.15: Diagrama de Casos de Uso: Administrar Descuentos
Fuente: El Autor

CASO DE USO	Administrar Descuentos
ACTORES	Administrador, Empleado
PROPÓSITO	Registrar los descuentos particulares de un empleado
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Se registra sin ningún error los datos del descuento ingresados

DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de "Administración" y luego al sub módulo de "Descuentos", selecciona "Nuevo Descuento", selecciona al empleado, digita una descripción del descuento, el monto a descontar del sueldo, selecciona el mes y el año de inicio que va a realizar el descuento y el número de meses en los que va a ser cancelado, y la información se guarda adecuadamente.
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Administración"	
Paso 2. El sistema muestra el módulo de "Administración"	
Paso 3. El administrador ingresa al sub módulo de "Descuentos"	
Paso 4. El sistema muestra el sub módulo de "Descuentos"	
Paso 5. El administrador selecciona "Nuevo Descuento"	
Paso 6. El administrador selecciona el empleado, digita una descripción del descuento, el monto a ser descontado, el mes y el año en el que se va a realizar el descuento del sueldo del empleado y digita el número de meses en los que va a ser cancelado el valor del mismo.	
Paso 7. El sistema valida y guarda la información	
Paso 8. El administrador cierra el sub módulo de "Descuentos"	
CURSO ALTERNATIVO	
Paso 2. El sistema no muestra el módulo correctamente, vuelve a Paso 1.	
Paso 4. El sistema no muestra el módulo correctamente, vuelve a Paso 3.	
Paso 6. El administrador selecciona "Anular Descuento"	
Paso 8. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 7	

*Tabla 7.2.15: Caso de Uso Administrar Descuentos
Fuente: El Autor*

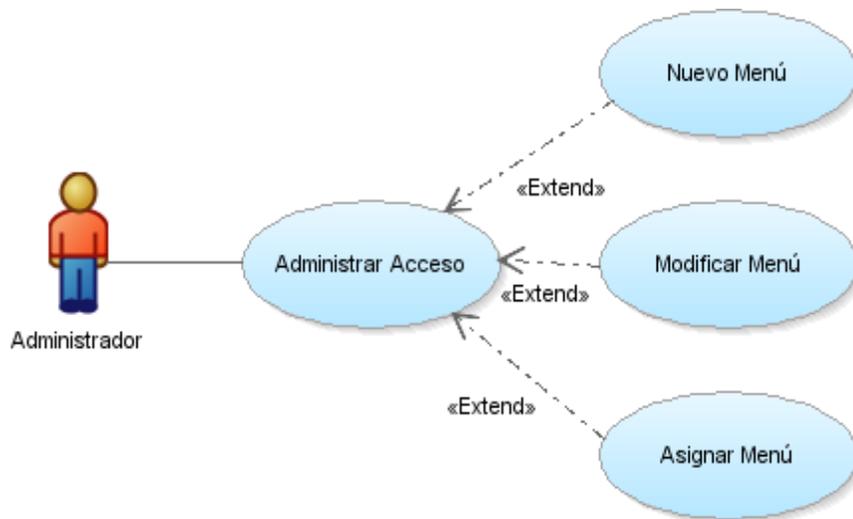


Figura 7.2. 16: Caso de Uso Administrar Acceso
Fuente: El Autor

CASO DE USO	Administrar Acceso
ACTORES	Administrador
PROPÓSITO	Administrar el acceso que tienen los usuarios a las páginas del sistema
PRECONDICIÓN	Ingresar al sistema con usuario y clave de administrador
POSTCONDICIÓN	Son ingresadas y asignadas las páginas a las que un usuario tiene acceso al sistema según su perfil.
DESCRIPCIÓN	El administrador ingresa al sistema con su usuario y contraseña, se dirige al módulo de Administración, luego al sub módulo de Acceso, selecciona la opción "Nuevo Menú", digita el nombre del menú y la URL de acceso al mismo.
CURSO NORMAL DE EVENTOS	
Paso 1. El administrador ingresa al módulo de "Administración"	
Paso 2. El sistema muestra el módulo de "Administración"	
Paso 3. El administrador ingresa al sub módulo de "Acceso"	
Paso 4. El sistema muestra el sub módulo de "Acceso"	
Paso 5. El administrador selecciona la opción "Nuevo Menú"	
Paso 6. El administrador digita el nombre del menú y la URL de acceso.	
Paso 7. El sistema valida y guarda la información.	
Paso 8. El administrador cierra el sub módulo de "Acceso".	

CURSO ALTERNATIVO
Paso 2. El sistema no muestra el módulo correctamente, vuelve Paso 1
Paso 4. El sistema no muestra el módulo correctamente, vuelve Paso 3
Paso 5. El administrador selecciona la opción "Modificar Menú"
Paso 5. El administrador selecciona la opción "Asignar Menú"
Paso 7. El sistema muestra un mensaje de error porque los datos ingresados son incorrectos, vuelve Paso 8

*Tabla 7.2.16: Caso de Uso Administrar Acceso
Fuente: El Autor*

ANEXO III: LISTADO DE COMPONENTES DE BASE DE DATOS

- **TABLAS**

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_ADELANTO	SERIAL	X	X		Número identificador único de un adelanto, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
DESCRIPCION_ADELANTO	VARCHAR(300)				Descripción del por qué de la solicitud del adelanto
VALOR_ADELANTO	DECIMAL	X			Valor total del adelanto a ser descontado del rol de pagos
MESES_PAGO_ADELANTO	INT4	X			Número de meses en los que se va a cancelar el adelanto

Tabla 7.3.1: Columnas de la tabla "ADELANTO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_AUDITORIA	SERIAL	X	X		Número identificador único de registro de auditoría, secuencial
ID_USUARIO	INT4			X	Número identificador único de un usuario, secuencial
ACCION	VARCHAR(100)	X			Descripción de la acción realizada por el usuario
FECHA_ACCION	DATE	X			Fecha en la que el usuario realizó la acción
IP_ACCESO	VARCHAR(30)				Dirección IP desde donde se realizó la acción

Tabla 7.3.2: Columnas de la tabla "AUDITORIA"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_BONIFICACION	SERIAL	X	X		Número identificador único de una bonificación, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
DESCRIPCION_BONIFICACION	VARCHAR(200)	X			Descripción correspondiente al porqué de la bonificación
VALOR_BONIFICACION	DECIMAL	X			Valor total que recibirá el empleado por la bonificación
FECHA_BONIFICACION	DATE	X			Fecha en la que se aplicó la bonificación

*Tabla 7.3.3: Columnas de la tabla "BONIFICACION"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_CALENDARIO	SERIAL	X	X		Número identificador único de una fecha de calendario, secuencial
ID_ESTADO	INT4			X	Número identificador único de un estado, secuencial
DESCRIPCION_PAGO	VARCHAR(100)	X			Descripción del pago que se debe realizar
VALOR_PAGO	DECIMAL	X			Valor total que debe cancelar la empresa
FECHA_PAGO	DATE	X			Fecha máxima en la que se debe realizar el pago

*Tabla 7.3.4: Columnas de la tabla "CALENDARIO_PAGO"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_CARGO	SERIAL	X	X		Número identificador único de un cargo, secuencial
ID_DEPARTAMENTO	INT4			X	Número identificador único de un departamento, secuencial
ID_ESTADO	INT4			X	Número identificador único de un estado, secuencial
NOMBRE_CARGO	VARCHAR(100)	X			Nombre del cargo
DESCRIPCION_CARGO	VARCHAR(200)	X			Describe brevemente las tareas que corresponden al cargo

Tabla 7.3.5: Columnas de la tabla "CARGO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_DEPARTAMENTO	SERIAL	X	X		Número identificador único de un departamento, secuencial
ID_ESTADO	INT4			X	Número identificador único de un estado, secuencial
NOMBRE_DEPARTAMENTO	VARCHAR(100)	X			Nombre o descripción del departamento

Tabla 7.3.6: Columnas de la tabla "DEPARTAMENTO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_DESCUENTO	SERIAL	X	X		Número identificador único de un descuento, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
DESCRIPCION_DESCUENTO	VARCHAR(200)	X			Descripción del porqué del descuento
VALOR_DESCUENTO	DECIMAL	X			Valor total del descuento a ser aplicado en el rol de pagos
MESES_PAGO_DESCUENTO	INT4	X			Número de meses en los que se va a cancelar el descuento
FECHA_INICIO_DESCUENTO	DATE	X			Fecha inicial en la que se va a aplicar el descuento

*Tabla 7.3.7: Columnas de la tabla "DESCUENTO"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_DIA_FESTIVO	SERIAL	X	X		Número identificador único de un día festivo, secuencial
ID_ESTADO	INT4			X	Número identificador único de un estado, secuencial
FECHA_DIA_FESTIVO	DATE	X			Fecha a la que corresponde el día festivo
NOMBRE_DIA_FESTIVO	VARCHAR(50)	X			Nombre con el que se identifica el día festivo

*Tabla 7.3.8: Columnas de la tabla "DIA_FESTIVO"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_EMPLEADO_CARGO	SERIAL	X	X		Número identificador único de la relación entre empleado y un cargo, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
ID_CARGO	INT4			X	Número identificador único de un cargo, secuencial
FECHA_ASIGNACION	DATE	X			Fecha en la que fue designado el empleado en un cargo determinado

*Tabla 7.3.9: Columnas de la tabla "EMPLEADO_CARGO"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_ESTADO	SERIAL	X	X		Número identificador único de un estado, secuencial
NOMBRE_ESTADO	VARCHAR(50)	X			Nombre que describe a un estado

*Tabla 7.3.10: Columnas de la tabla "ESTADO"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_ESTADO_CIVIL	SERIAL	X	X		Número identificador único de un estado civil, secuencial
NOMBRE_ESTADO_CIVIL	VARCHAR(50)	X			Nombre que identifica a un estado civil

*Tabla 7.3.11: Columnas de la tabla "ESTADO_CIVIL"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_MULTA	SERIAL	X	X		Número identificador único de una multa, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
DESCRIPCION_MULTA	VARCHAR(150)	X			Descripción del porqué se aplicó la multa
VALOR_MULTA	DECIMAL	X			Valor total de la multa a ser descontado del rol de pagos del empleado
FECHA_MULTA	DATE	X			Fecha en la que se aplicó la multa al empleado

*Tabla 7.3.12: Columnas de la tabla "MULTA"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_PERFIL	SERIAL	X	X		Número identificador único de un perfil, secuencial
NOMBRE_PERFIL	VARCHAR(100)	X			Nombre del perfil a ser asignado a un usuario

*Tabla 7.3.13: Columnas de la tabla "PERFIL"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_PERMISO	SERIAL	X	X		Número identificador único de un permiso, secuencial
ID_TIPO_PERMISO	INT4			X	Número identificador único de un tipo de permiso, secuencial

ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
HORA_INICIO_PERMISO	TIME				Hora inicial del permiso
HORA_FIN_PERMISO	TIME				Hora en la que termina el permiso
FECHA_INICIO_PERMISO	DATE				Fecha inicial del permiso
FECHA_FIN_PERMISO	DATE				Fecha final del permiso
DESCRIPCION_PERMISO	VARCHAR(300)	X			Descripción del porqué del permiso solicitado

Tabla 7.3.14: Columnas de la tabla "PERMISO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_REGISTRO	SERIAL	X	X		Número identificador único de un registro de hora de entrada/salida, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
FECHA_REGISTRO	DATE	X			Fecha correspondiente al registro
HORA_ENTRADA_REGISTRO	TIME				Hora de entrada del empleado
HORA_SALIDA_REGISTRO	TIME				Hora de salida del empleado
ATRASO	BOOL				Valor verdadero si el empleado se atrasó
HORAS_LABORADAS	DECIMAL				Número total de horas laboradas por el empleado

Tabla 7.3.15: Columnas de la tabla "REGISTRO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_ROL_PAGO	SERIAL	X	X		Número identificador único de un rol de pago, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
PERIODO_PAGO	DATE	X			Período de pago al que corresponde el rol de pago
TOTAL_INGRESOS	DECIMAL	X			Sumatoria de los ingresos que tenga el empleado en el período de pago
TOTAL_EGRESOS	DECIMAL	X			Sumatoria de los egresos que tenga el empleado en el período de pago
TOTAL	DECIMAL	X			Valor total que recibirá el empleado en el período de pago, corresponde al total de ingresos menos el total de egresos

Tabla 7.3.16: Columnas de la tabla "ROL_PAGO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_SEXO	SERIAL	X	X		Número identificador único de un sexo, secuencial
NOMBRE_SEXO	VARCHAR(50)	X			Nombre que identifica al sexo de una persona

Tabla 7.3.17: Columnas de la tabla "SEXO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_TIPO_PERMISO	SERIAL	X	X		Número identificador único de un tipo de permiso, secuencial
NOMBRE_TIPO_PERMISO	VARCHAR(50)	X			Nombre identificador correspondiente al tipo de permiso
AFECTA_VACACIONES	BOOL	X			Indica si afecta al período de vacaciones o no

*Tabla 7.3.18: Columnas de la tabla "TIPO_PERMISO"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_TURNO	SERIAL	X	X		Número identificador único de un turno, secuencial
ID_ESTADO	INT4			X	Número identificador único de un estado, secuencial
NOMBRE_TURNO	VARCHAR(100)	X			Nombre que identifica y/o describe brevemente a un turno
HORA_ENTRADA_TURNO	TIME	X			Hora en la que el empleado debe ingresar
HORA_SALIDA_TURNO	TIME	X			Hora en la que el empleado debe salir

*Tabla 7.3.19: Columnas de la tabla "TURNO"
Fuente: El Autor*

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_USUARIO	SERIAL	X	X		Número identificador único de un usuario, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
ID_ESTADO	INT4			X	Número identificador único de un estado, secuencial
ID_PERFIL	INT4			X	Número identificador único de un perfil, secuencial
NOMBRE_USUARIO	VARCHAR(30)	X			Nombre de usuario que se utiliza para iniciar sesión en el sistema
CLAVE	VARCHAR(50)	X			Clave de acceso al sistema, encriptada

Tabla 7.3.20: Columnas de la tabla "USUARIO"

Fuente: El Autor

Nombre	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea	Descripción
ID_VACACION	SERIAL	X	X		Número identificador único de un período de vacaciones, secuencial
ID_EMPLEADO	INT4			X	Número identificador único de cada empleado, secuencial
FECHA_INICIO_VACACION	DATE	X			Fecha correspondiente al inicio de las vacaciones de un empleado
FECHA_FIN_VACACION	DATE	X			Fecha correspondiente al fin de las vacaciones de un empleado
OBSERVACIONES	VARCHAR(300)				Observaciones adicionales a la solicitud de vacaciones

Tabla 7.3.21: Columnas de la tabla "VACACION"

Fuente: El Autor

Nombre	Descripción	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea
ID_MENU	Número identificador único de un menú, secuencial	SERIAL	X	X	
NOMBRE_MENU	Nombre identificador de un menú	VARCHAR(200)	X		
URL	URL de acceso al menú	VARCHAR(200)	X		

*Tabla 7.3.22: Columnas de la tabla "MENU"
Fuente: El Autor*

Nombre	Descripción	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea
ID_MENU_PERFIL		SERIAL	X	X	
ID_MENU	Número identificador único de un menú, secuencial	INT4			X
ID_PERFIL	Número identificador único de un perfil, secuencial	INT4			X

*Tabla 7.3.23: Columnas de la tabla "MENU_PERFIL"
Fuente: El Autor*

Nombre	Descripción	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea
ID_PAGO_ADELANTO	Número identificador único del pago de un adelanto, secuencial	SERIAL	X	X	
ID_ADELANTO	Número identificador único de un adelanto, secuencial	INT4			X
VALOR_PAGO_ADELANTO	Valor que debe ser descontado al empleado en su rol de pagos por motivo del adelanto	DECIMAL	X		
FECHA_PAGO_ADELANTO	Período en el que se va a realizar el descuento por el adelanto al empleado en su rol de pagos	DATE	X		

*Tabla 7.3.24: Columnas de la tabla "PAGO_ADELANTO"
Fuente: El Autor*

Nombre	Descripción	Tipo de Dato	Mandatorio	Clave Primaria	Clave Foránea
ID_PAGO_DESCUENTO	Número identificador único del pago de un descuento, secuencial	SERIAL	X	X	
ID_DESCUENTO	Número identificador único de un descuento, secuencial	INT4			X
VALOR_PAGO_DESCUENTO	Valor a ser descontado en el rol de pagos del empleado por motivo de un descuento	DECIMAL	X		
FECHA_PAGO_DESCUENTO	Fecha en la que se va realizar el descuento en el rol de pagos del empleado	DATE	X		

*Tabla 7.3.25: Columnas de la tabla "PAGO_DESCUENTO"
Fuente: El Autor*

ANEXO IV: CASOS DE PRUEBA DEL SISTEMA

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo turno	El usuario debe estar autenticado	Nombre del turno=vacio, hora de entrada=vacio, hora de salida=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre del turno=TARDE, hora de entrada=aaaa, hora de salida=bbbb	El sistema muestra mensaje de error indicando que el formato para horas es hh:mm	Correcto
		Nombre del turno=TARDE, hora de entrada=12:00, hora de salida=19:00	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede ingresar un nuevo turno	El turno debe estar creado	Nombre del turno=vacio, hora de entrada=vacio, hora de salida=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre del turno=TARDE, hora de entrada=aaaa, hora de salida=bbbb	El sistema muestra mensaje de error indicando que el formato para horas es hh:mm	Correcto
		Nombre del turno=TARDE, hora de entrada=12:00, hora de salida=20:00	El sistema guarda la información y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.1: Caso de prueba de administración de turnos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo día festivo	El usuario debe estar autenticado	Nombre del día festivo=vacio, fecha de celebración=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre del día festivo=NAVIDAD, fecha de celebración=25/12/2012	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto

Comprobar que el administrador puede modificar un día festivo	El día festivo debe estar creado	Nombre del día festivo=vacio, fecha de celebración=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre del día festivo=NAVIDAD, fecha de celebración=01/12/2013	El sistema guarda la información y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.2: Caso de prueba de administración de días festivos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar una nueva multa	El usuario debe estar autenticado, el empleado a aplicar la multa debe estar registrado	Empleado=no seleccionado, detalles=vacio, fecha=vacio, valor= vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, detalles=Faltas al reglamento, fecha=2012/05/01, valor=aaaaa	El sistema muestra un mensaje de error de formato numérico incorrecto	Correcto
		Empleado=Veronica Mancheno, detalles=Faltas al reglamento, fecha=2012/05/01, valor=20	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede anular una multa	La multa debe estar creada	Empleado=no seleccionado, confirmación de código de multa=no seleccionado	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, confirmación de código de multa=incorrecto	El sistema muestra un mensaje de error indicando que el código no existe o no está relacionado al empleado seleccionado	Correcto
		Empleado=Veronica Mancheno, confirmación de código de multa=correcto	El sistema anula la multa y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.3: Caso de prueba de administración de multas
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo descuento	El usuario debe estar autenticado, el empleado a aplicar el descuento debe estar registrado	Empleado=no seleccionado, detalles=vacio, fecha=vacio, valor= vacio, meses a diferir=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, detalles=Préstamo, fecha=2012-05-01, valor=aaaa, meses a diferir=bbbb	El sistema muestra un mensaje de error de formato numérico incorrecto	Correcto
		Empleado=Veronica Mancheno, detalles=Préstamo, fecha=2012-05-01, valor=300, meses a diferir=5	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede anular un descuento	El descuento debe estar creado	Empleado=no seleccionado, confirmación de código de descuento=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica mancheno, confirmación de código de descuento=incorrecto	El sistema muestra un mensaje de error indicando que el código no existe o no está relacionado al empleado seleccionado	Correcto
		Empleado=Veronica mancheno, confirmación de código de descuento=correcto	El sistema anula el descuento y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.4: Caso de prueba de administración de descuentos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo adelanto	El usuario debe estar autenticado, el empleado a aplicar el	Empleado=no seleccionado, detalles=vacio, fecha solicitud=vacio, fecha a pagar=vacio, valor= vacio, meses a diferir=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto

	adelanto debe estar registrado	Empleado=Veronica Mancheno, detalles=Adelanto de sueldo, fecha solicitud=2012/03/01, fecha a pagar=2012/04/01, valor=bbbb, meses a diferir=cccc	El sistema muestra un mensaje de error de formato numérico incorrecto	Correcto
		Empleado=Veronica Mancheno, detalles=Adelanto de sueldo, fecha solicitud=2012/03/01, fecha a pagar=2012/04/01, valor=50, meses a diferir=2	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede anular un adelanto	El adelanto debe estar creado	Empleado=no seleccionado, confirmación de código de adelanto=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, confirmación de código de adelanto=incorrecto	El sistema muestra un mensaje de error indicando que el código no existe o no está relacionado al empleado seleccionado	Correcto
		Empleado=Veronica Mancheno, confirmación de código de adelanto=correcto	El sistema anula el adelanto y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.5: Caso de prueba de administración de adelantos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar una nueva bonificación	El usuario debe estar autenticado, el empleado a aplicar la bonificación debe estar registrado	Empleado=no seleccionado, detalles=vacio, valor=vacio, fecha=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Verónica Mancheno, detalles=Bono, valor=xxx, fecha=2012/05/01	El sistema muestra un mensaje de error de formato numérico incorrecto	Correcto

		Empleado=Verónica Mancheno, detalles=Bono, valor=45.55, fecha=2012/05/01	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede anular una bonificación	La bonificación debe estar creada	Empleado=no seleccionado, confirmación de código de adelanto=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, confirmación de código de bonificación=incorrecto	El sistema muestra un mensaje de error indicando que el código no existe o no está relacionado al empleado seleccionado	Correcto
		Empleado=Veronica Mancheno, confirmación de código de bonificación=correcto	El sistema anula la bonificación y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.6: Caso de prueba de administración de bonificaciones
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede programar un nuevo período de vacaciones	El usuario debe estar autenticado, el empleado que solicita las vacaciones debe estar registrado	Empleado=no seleccionado, fecha inicio=vacio, fecha fin=vacio, observaciones=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, fecha inicio=2012/05/01, fecha fin=2012/05/15, observaciones= Vacaciones del 2012	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede anular un período de vacaciones	El período de vacaciones debe estar creado	Empleado=no seleccionado, confirmación de código de período de vacaciones=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, confirmación de código de período de vacaciones=incorrecto	El sistema muestra un mensaje de error indicando que el código no existe o no está relacionado al empleado seleccionado	Correcto

		Empleado=Veronica mancheno, confirmación de código de período de vacaciones=correcto	El sistema anula el período de vacaciones y redirige a página de modificación exitosa	Correcto
--	--	--	---	----------

*Tabla 7.4.7: Caso de prueba de administración de períodos de vacaciones
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo permiso	El usuario debe estar autenticado, el empleado que solicita el permiso debe estar registrado	Permiso por=no seleccionado, empleado=no seleccionado, tipo permiso=no seleccionado, fecha inicio=vacio, fecha fin=vacio, observaciones= vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Permiso por=Horas, empleado=Veronica Mancheno, tipo permiso= Personal, hora inicio=gggg, hora fin= aaaa, observaciones= Estudios	El sistema muestra mensaje de error indicando que el formato de las horas debe ser hh:mm	Correcto
		Permiso por=Horas, empleado=Veronica Mancheno, tipo permiso= Personal, hora inicio=10:00, hora fin= 11:00, observaciones= Estudios	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
		Permiso por=Días, empleado=Veronica Mancheno, tipo permiso= Personal, fecha inicio= 2012/05/01, fecha fin= 2012/05/03, observaciones= Estudios	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede anular un permiso	El permiso debe estar creado	Empleado=no seleccionado, confirmación de código de permiso=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Incorrecto, no muestra el mensaje
		Empleado=Veronica Mancheno, confirmación de código de permiso=incorrecto	El sistema muestra un mensaje de error indicando que el código no existe o no está relacionado al empleado seleccionado	Correcto

		Empleado=Veronica Mancheno, confirmación de código de permiso=correcto	El sistema anula el permiso y redirige a página de modificación exitosa	Correcto
--	--	--	---	----------

*Tabla 7.4.8: Caso de prueba de administración de permisos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar una nueva fecha de pago	El usuario debe estar autenticado	Detalles=vacio, valor=vacio, fecha pago= vacio, estado=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Detalles=Pagar aportes de IESS, valor=jjj7, fecha pago= 2012/04/12, estado=Pendiente	El sistema muestra mensaje de error indicando formato numérico incorrecto	Correcto
		Detalles=Pagar aportes de IESS, valor=570, fecha pago= 2012/04/12, estado=Pendiente	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede modificar una fecha de pago	La fecha de pago debe estar creada	Detalles=vacio, valor=vacio, fecha pago= vacio, estado=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Detalles=Pagar aportes de IESS, valor=jjj7, fecha pago= 2012/04/12, estado=Pendiente	El sistema muestra mensaje de error indicando formato numérico incorrecto	Correcto
		Detalles=Pagar aportes de IESS, valor=700, fecha pago= 2012/04/12, estado=Pagado	El sistema modifica la información y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.9: Caso de prueba de administración de calendario de pagos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo cargo	El usuario debe estar autenticado	Nombre del cargo=vacio, departamento= no seleccionado, funciones= vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre del cargo=Contador, departamento= Contabilidad, funciones= Reportes contables	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede modificar un cargo	El cargo debe estar creado	Cargo=no seleccionado, nombre cargo=vacio, departamento=no seleccionado, funciones, vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre del cargo=Contador, departamento= Contabilidad y Auditoría, funciones= Reportes contables	El sistema modifica la información y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.10: Caso de prueba de administración de cargos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo departamento	El usuario debe estar autenticado	Nombre del departamento=vacio	El sistema muestra mensaje de error indicando que el valor es requerido	Correcto
		Nombre del departamento= Sistemas	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede modificar un departamento	El departamento debe estar creado	Departamento=no seleccionado, nombre departamento=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto

		Departamento=sistemas, nombre departamento= Tecnología	El sistema modifica la información y redirige a página de modificación exitosa	Correcto
--	--	--	--	----------

*Tabla 7.4.11: Caso de prueba de administración de departamentos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo usuario	El usuario debe estar autenticado	Empleado=no seleccionado, nombre usuario= vacío, clave=vacio, perfil=no seleccionado	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, nombre usuario=vmancheno, clave=1234, perfil=Administrador	El sistema muestra mensaje de error indicando que el nombre de usuario ya existe	Correcto
		Empleado=Veronica Mancheno, nombre usuario=vmancheno2, clave=1234, perfil=Administrador	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede modificar un usuario	El usuario debe estar creado	Empleado=no seleccionado, nombre usuario= vacío, clave=vacio, perfil=no seleccionado	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, nombre usuario=vmancheno, clave=1234, perfil=Administrador	El sistema muestra mensaje de error indicando que el nombre de usuario ya existe	Correcto
		Empleado=Veronica Mancheno, nombre usuario=vmancheno2, clave=1234, perfil=Empleado	El sistema modifica la información y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.12: Caso de prueba de administración de usuarios
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador puede ingresar un nuevo menú	El usuario debe estar autenticado	Nombre menu= vacio, url=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre menu= Reportes, url=/pages/reportes/Reporte1.seam	El sistema guarda la información y redirige a página de ingreso exitoso	Correcto
Comprobar que el administrador puede modificar un menú	El menú debe estar creado	Nombre menu= vacio, url=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Nombre menu= Reportes, url=/pages/reportes/Reporte2.seam	El sistema modifica la información y redirige a página de modificación exitosa	Correcto

*Tabla 7.4.13: Caso de prueba de administración de menús
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador pueda modificar el registro de entrada/salida	El usuario debe estar autenticado	Empleado=no seleccionado, fecha marcación= vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, fecha marcación= 2012/05/20, hora entrada=aaaa, hora salida=bbbb, atraso= si, horas laboradas= cccc	El sistema muestra mensaje de error indicando que el formato de las horas debe ser hh:mm	Correcto

		Empleado=Veronica Mancheno, fecha marcación= 2012/05/20, hora entrada=07:00, hora salida=08:00, atraso= no, horas laboradas= 01:00	El sistema modifica la información y redirige a página de modificación exitosa	Correcto
--	--	--	--	----------

*Tabla 7.4.14: Caso de prueba de modificación de registros de horas de entrada/salida
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador pueda consultar los registros de auditoría	El usuario debe estar autenticado	Empleado=no seleccionado, fecha inicio=vacio, fecha fin=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Veronica Mancheno, fecha inicio=2012/01/01, fecha fin=2012/12/31	El sistema muestra los registros de auditoría correspondientes al empleado en formato de tabla	Correcto

*Tabla 7.4.15: Caso de prueba de consulta de registros de auditoría
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador pueda generar el reporte de atrasos	El usuario debe estar autenticado	Empleado=no seleccionado, fecha inicio=vacio, fecha fin=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Todos, fecha inicio=2012/01/01, fecha fin=2012/12/31	El sistema muestra el reporte de atrasos correspondiente a todos los empleados en el período seleccionado.	Correcto

		Empleado=Veronica Mancheno, fecha inicio=2012/01/01, fecha fin=2012/12/31	El sistema muestra el reporte de atrasos correspondiente al empleado en el periodo seleccionado.	Correcto
--	--	---	--	----------

*Tabla 7.4.16: Caso de prueba de generación de reporte de atrasos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador pueda generar el reporte de permisos	El usuario debe estar autenticado	Empleado=no seleccionado, fecha inicio=vacio, fecha fin=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto
		Empleado=Todos, fecha inicio=2012/01/01, fecha fin=2012/12/31	El sistema muestra el reporte de permisos correspondiente a todos los empleados en el periodo seleccionado.	Correcto
		Empleado=Veronica Mancheno, fecha inicio=2012/01/01, fecha fin=2012/12/31	El sistema muestra el reporte de permisos correspondiente al empleado en el periodo seleccionado.	Correcto

*Tabla 7.4.17: Caso de prueba de generación de reporte de permisos
Fuente: El Autor*

Caso de prueba	Prerrequisito	Datos de prueba	Resultados esperados	Resultados obtenidos
Comprobar que el administrador pueda generar el reporte de rol de pagos	El usuario debe estar autenticado	Empleado=no seleccionado, fecha inicio=vacio, fecha fin=vacio	El sistema muestra mensaje de error indicando que los valores son requeridos	Correcto

		Empleado=Todos, fecha inicio=2012/01/01, fecha fin=2012/12/31	El sistema muestra el reporte de rol de pagos correspondiente a todos los empleados en el período seleccionado.	Correcto
		Empleado=Veronica Mancheno, fecha inicio=2012/01/01, fecha fin=2012/12/31	El sistema muestra el reporte de rol de pagos correspondiente al empleado en el período seleccionado.	Correcto

*Tabla 7.4.18: Caso de prueba de generación de reporte de rol de pagos
Fuente: El Autor*

ANEXO V: MANUAL DE USUARIO

SISTEMA DE CONTROL DE ASISTENCIA DE PERSONAL MEDIANTE DISPOSITIVOS BIOMÉTRICOS Y EMISIÓN DE ROL DE PAGOS

Manual de Usuario

El sistema de control de asistencia y emisión de rol de pagos necesita de un navegador Web para su uso, se recomienda usar Internet Explorer 8 para su correcto funcionamiento y despliegue de componentes, aunque también puede hacer uso del navegador de su preferencia.

Para ingresar tiene que abrir el navegador Web y acceder a la dirección del sistema, las direcciones pueden variar dependiendo las configuraciones de red de ALPROSART.

Una vez se ingresa a la dirección del sistema se presenta la pantalla de inicio de sesión.

- **Inicio de sesión**

En la pantalla principal debe introducir el usuario y la clave proporcionados por el administrador, a continuación debe hacer clic en el botón *Ingresar*.

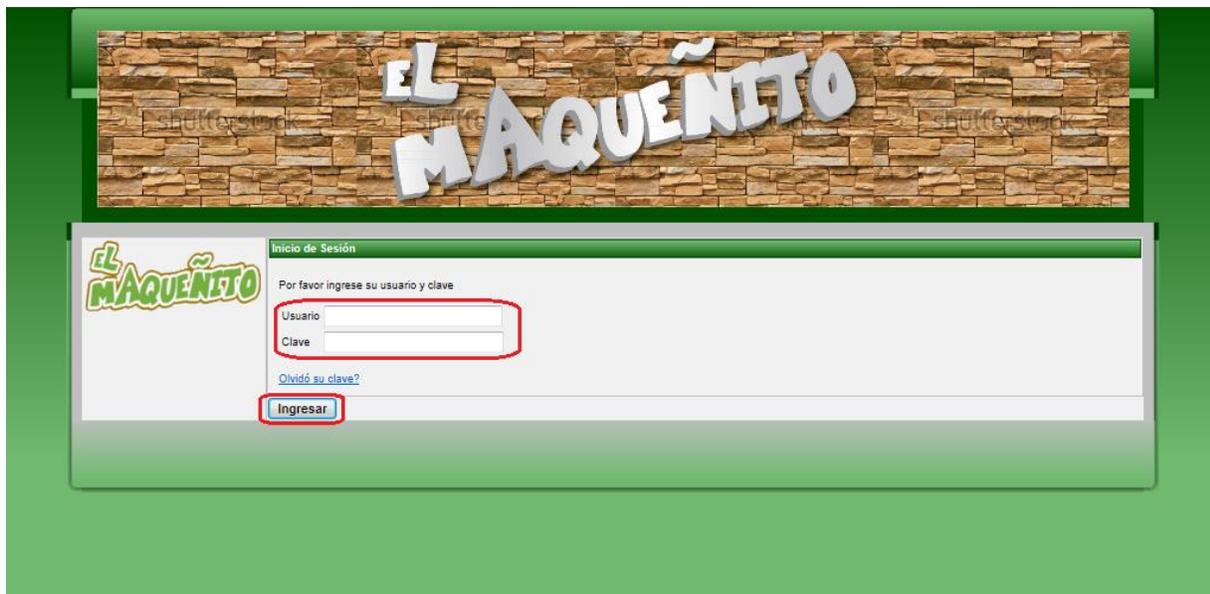
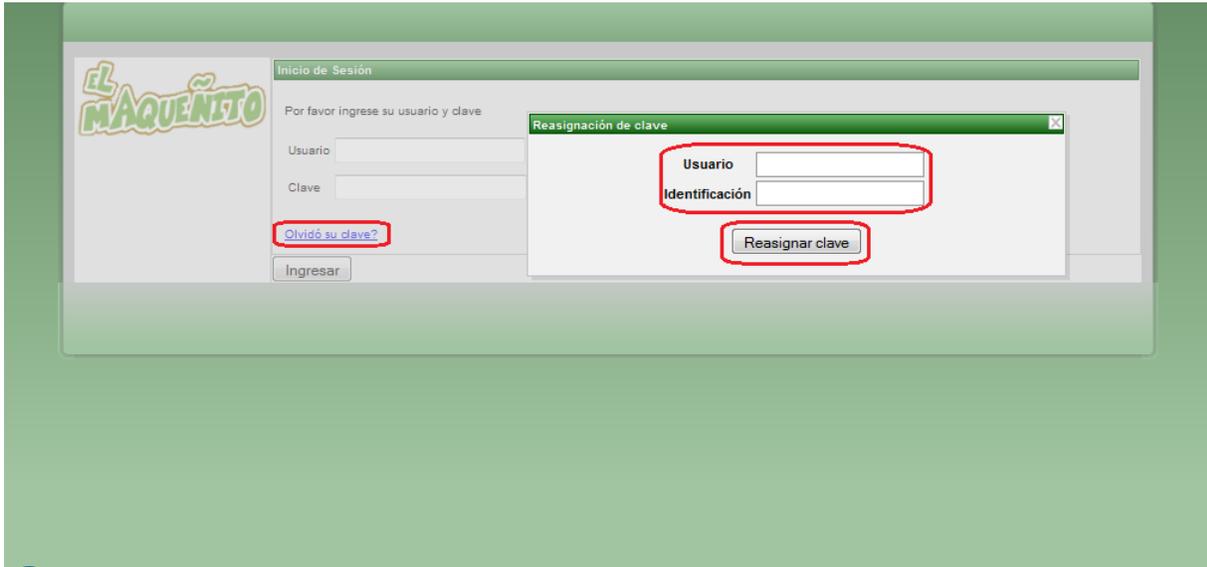


Figura 7.5.1: Formulario de Inicio de sesión
Fuente: El Autor

- **Reasignación de clave**

Si no recuerda su clave debe hacer clic en el link *Olvidó su clave?*. En la nueva ventana debe ingresar su usuario e identificación personal, luego dar clic en el botón *Reasignar clave*. Luego recibirá un correo electrónico con su nueva clave.



*Figura 7.5.2: Formulario de Reasignación de clave
Fuente: El Autor*

Mientras tenga una sesión activa en el sistema usted podrá visualizar en el panel superior los siguientes elementos: Link de salida del sistema, con el cual podrá cerrar su sesión en cualquier momento, el nombre de usuario que está autenticado en el sistema; y dependiendo del perfil que tenga asignado el usuario podrá visualizar en el menú principal las opciones de *Administración y Registro* para usuarios con perfil de *Empleado*, ver *Figura 7.5.3*, y *Administración, Rol de Pagos, Registro, Reportes, Auditoría* para usuarios con perfil de *Administrador*, ver *Figura 7.5.4*.



Figura 7.5.3: Opciones visualizadas por el usuario con perfil Empleado
Fuente: El Autor



Figura 7.5.4: Opciones visualizados por el usuario con perfil Administrador
Fuente: El Autor

- **Administración**

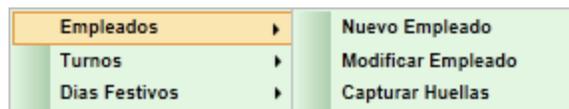
Es la primera opción del menú principal, permite la administración de todos los elementos del sistema, es decir se pueden crear, consultar, modificar y eliminar información del sistema. Si su usuario no tiene perfil de Administrador sólo podrá

- Empleados ▶
- Turnos ▶
- Dias Festivos ▶
- Multas ▶
- Descuentos ▶
- Adelantos ▶
- Bonificaciones ▶
- Vacaciones ▶
- Permisos ▶
- Calendario de Pago ▶
- Cargos ▶
- Departamentos ▶
- Usuarios ▶
- Menu ▶

acceder a la modificación de cierta información de su usuario o de sus datos personales.

- **Administración de empleados**

Para acceder a esta opción debe dar clic en el menú de Administración y luego seleccionar Empleados; en esta opción se desplegarán tres sub menús: Nuevo Empleado, Modificar Empleado y Capturar Huellas.



*Figura 7.5.5: Opciones del menú Administrar Empleados
Fuente: El Autor*

Nuevo Empleado / Modificar Empleado: Al seleccionar Nuevo Empleado se presenta un formulario con los campos que deben ser completados, en la Tabla 7.5.1 se describe cada uno de los campos.

CAMPO	OBLIGATORIO	DESCRIPCIÓN	NÚMERO DE CARACTERES MÁXIMO
NOMBRES	SI	Nombres del empleado	200
APELLIDOS	SI	Apellidos del empleado	200
IDENTIFICACIÓN	SI	Número de cédula o pasaporte del empleado	20
FECHA NACIMIENTO	SI	Fecha de nacimiento del empleado	-
ESTADO CIVIL	SI	Estado civil del empleado	-
CARGAS FAMILIARES	SI	Número de cargas familiares del empleado	-
SEXO	SI	Sexo del empleado	-
TIPO DE SANGRE	SI	Tipo de Sangre del empleado	20
DIRECCIÓN	SI	Dirección del domicilio del empleado	300
TELÉFONO CONVENCIONAL	SI	Teléfono convencional de contacto	20
TELÉFONO CELULAR	SI	Teléfono celular de contacto	20
CORREO ELECTRÓNICO	SI	Correo electrónico de contacto	100
EXPERIENCIA LABORAL	SI	Experiencia laboral del empleado	3000
FECHA DE INGRESO A LA EMPRESA	SI	Fecha en la que firmó el contrato inicial	-
DEPARTAMENTO	SI	Departamento al que pertenece el empleado	-
CARGO	SI	Cargo asignado al empleado	-
REMUNERACIÓN BÁSICA	SI	Sueldo básico del empleado, se utilizará este valor para generar el rol de pagos	-

VALOR POR HORA EXTRA	SI	Monto a ser acreditado por cada hora extra del empleado, se utilizará este valor para generar el rol de pagos	-
TURNO DE TRABAJO	SI	Turno en el que labora el empleado, se debe asignar correctamente el turno caso contrario al registrar la hora de entrada puede marcarse como atraso.	-
MENSAJE PERSONAL	SI	Mensaje que se le presentará al empleado cada vez que marque su hora de entrada o salida	300
FOTOGRAFÍA	NO	Fotografía del empleado, en formato jpeg	-
ESTADO	SI	Estado del empleado, por defecto Activo, en el caso que el empleado salga de la empresa cambiarlo a Inactivo	-

*Tabla 7.5.1: Campos del formulario de nuevo empleado
Fuente: El Autor*

NOMBRES:

APELLIDOS:

IDENTIFICACIÓN:

FECHA DE NACIMIENTO:

ESTADO CIVIL: -- Seleccione --

CARGAS FAMILIARES:

SEXO: -- Seleccione --

TIPO DE SANGRE:

DIRECCIÓN:

TELÉFONO CONVENCIONAL:

TELÉFONO CELULAR:

CORREO ELECTRÓNICO:

EXPERIENCIA LABORAL:

FECHA DE INGRESO A LA EMPRESA:

DEPARTAMENTO: -- Seleccione --

CARGO: -- Seleccione --

REMUNERACIÓN BÁSICA:

VALOR POR HORA EXTRA:

TURNO DE TRABAJO: -- Seleccione --

MENSAJE PERSONAL:

FOTOGRAFÍA Examinar...

ESTADO: ACTIVO

*Figura 7.5.6: Formulario de Ingreso de Nuevo Empleado
Fuente: El Autor*

En el caso de que se seleccione la opción Modificar Empleado se debe seleccionar primero el empleado que se desea modificar, como se muestra en la Figura 7.5.7. Luego de que se haya seleccionado el empleado, se cargará toda la información del empleado seleccionado y se modificará la información requerida.

SELECCIONE UN EMPLEADO:	-- Seleccione --
INFORMACIÓN DEL EMPLEADO:	-- Seleccione --
CÓDIGO DEL EMPLEADO:	CUENCA ACARO DIANA CECILIA
NOMBRES:	ESPINOZA LAVAYEN ARMANDO AGUSTIN
APELLIDOS:	ESTRADA GOMEZ DAVID ALEXANDER
IDENTIFICACIÓN:	GUAMAN CUENCA MARGARITA SOLEDAD
FECHA DE NACIMIENTO:	MALAN CHUNGA RUBEN DARIO
ESTADO CIVIL:	MANCHENO GUAMAN VALERIA SOLEDAD
CARGAS FAMILIARES:	MANCHENO GUAMÁN VERÓNICA LIZETH
SEXO:	MANCHENO ROSERO LEE VINICIO
	ORELLANA MALDONADO FRANCISCO JAVIER
	ORTEGA CANGO JOSÉ MÁXIMO
	SANCHEZ CHICAIZA CARLOS DAVID
	SANCHEZ IZA OSCAR DAMIAN
	SANCHEZ IZA JESUS DAVID
	SANCHEZ IZA HOLGUER FRANCISCO

Figura 7.5.7: Selección de empleado en el formulario Modificar Empleado
Fuente: El Autor

Una vez los campos se encuentren llenados correctamente ya sea en el caso de Nuevo Empleado o Modificar Empleado se debe hacer clic sobre el botón **Guardar** para que los cambios realizados se apliquen, si no hubo errores se presentará una página donde se indica que el proceso se realizó exitosamente.

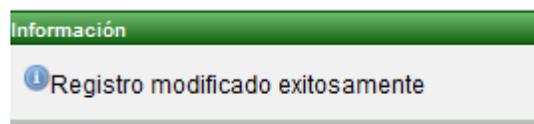


Figura 7.5.8: Mensaje de ingreso / modificación correcto
Fuente: El Autor

Capturar Huellas: Al seleccionar la opción Capturar Huellas se muestra un formulario en donde se podrán registrar los diez dedos de la mano del empleado, primero debe seleccionar al empleado y luego seleccionar el dedo que se va a registrar, el lector biométrico debe estar conectado correctamente. Luego debe presionar el botón

Capturar para iniciar el registro biométrico. El dedo debe cubrir la mayor parte del sensor para una mejor calidad de imagen. El dedo debe ser colocado por cuatro ocasiones sobre el sensor y debe mantenerlo un mínimo de dos segundos. Para su ayuda en el formulario se muestra una figura con los nombres de los dedos de cada mano.

El sistema mostrará un mensaje indicando si el registro fue exitoso o no.

Mano Izquierda	Mano Derecha
1: Meñique	6: Pulgar
2: Anular	7: Índice
3: Medio	8: Medio
4: Índice	9: Anular
5: Pulgar	10: Meñique

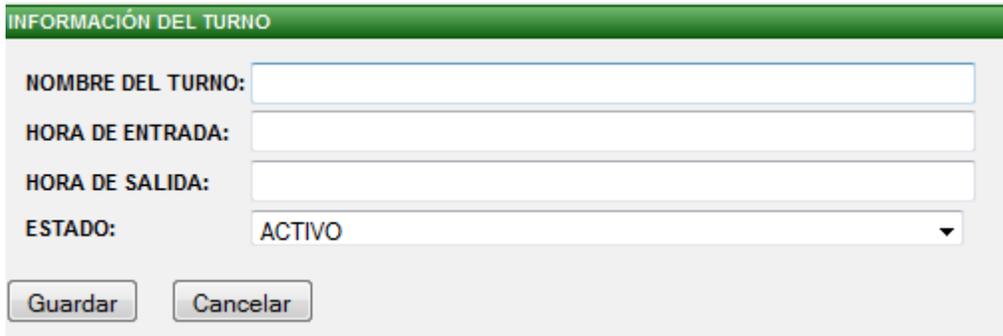
Figura 7.5.9: Formulario de registro de huellas dactilares
Fuente: El Autor

- **Administración de Turnos**

Para acceder a la administración de turnos debe seleccionar el menú de Administración y seleccionar la opción Turnos, se presentarán las siguientes opciones:

Nuevo Turno: debe llenar todos los campos que presenta el formulario: Nombre del turno debe ser un nombre descriptivo del turno, Hora de entrada y Hora de salida deben estar en el formato hh:mm, el Estado por defecto es Activo. Una vez llenados

todos los campos correctamente presiona el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8 si no existieron errores.



INFORMACIÓN DEL TURNO

NOMBRE DEL TURNO:

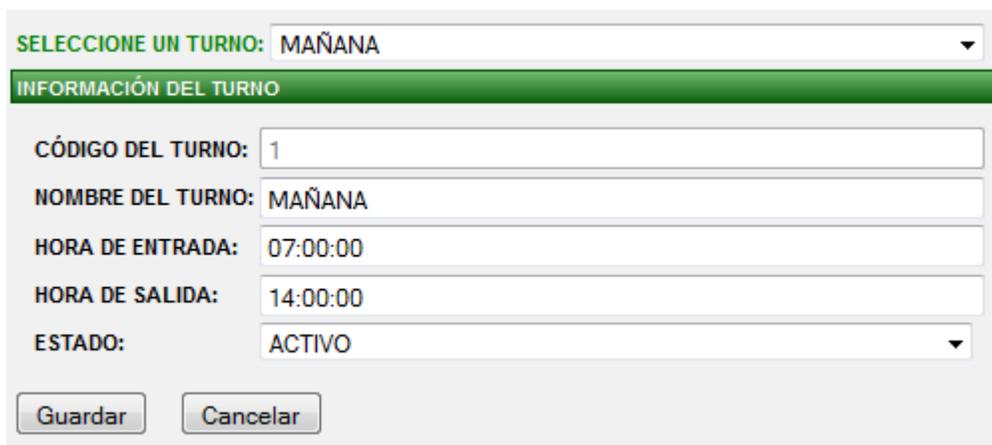
HORA DE ENTRADA:

HORA DE SALIDA:

ESTADO:

*Figura 7.5.10: Formulario de Nuevo Turno
Fuente: El Autor*

Modificar Turno: en el formulario que se presenta primero debe seleccionar el turno que desea modificar, luego debe cambiar la información requerida con el formato establecido. Para que los cambios se realicen debe presionar el botón **Guardar**.



SELECCIONE UN TURNO:

INFORMACIÓN DEL TURNO

CÓDIGO DEL TURNO:

NOMBRE DEL TURNO:

HORA DE ENTRADA:

HORA DE SALIDA:

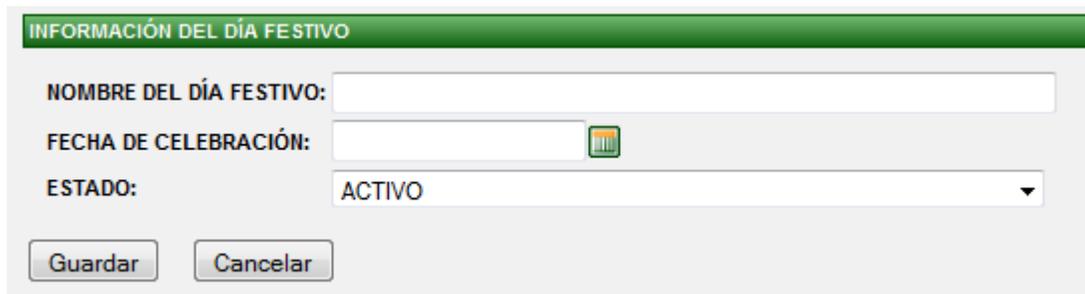
ESTADO:

*Figura 7.5.11: Formulario de Modificación de Turno
Fuente: El Autor*

- **Administrar Días Festivos**

Para acceder a la administración de días festivos debe seleccionar el menú de Administración y seleccionar la opción Días Festivos, se presentarán las siguientes opciones:

Nuevo Día Festivo: En el formulario que se presenta debe completar los datos de: Nombre del día festivo, debe ser una descripción breve del día festivo o el nombre común; Fecha de celebración, debe escoger el día exacto que se celebra; estado, por defecto es Activo. Una vez llenados todos los campos correctamente presiona el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8, si no existieron errores.



The image shows a web form titled "INFORMACIÓN DEL DÍA FESTIVO". It has three main input fields: "NOMBRE DEL DÍA FESTIVO" (a text box), "FECHA DE CELEBRACIÓN" (a date picker with a calendar icon), and "ESTADO" (a dropdown menu currently showing "ACTIVO"). At the bottom of the form are two buttons: "Guardar" and "Cancelar".

*Figura 7.5.12: Formulario de Nuevo Día Festivo
Fuente: El Autor*

Modificar Día Festivo: en el formulario que se presenta primero debe seleccionar el día festivo que desea modificar, luego debe cambiar la información requerida con el formato establecido. Para que los cambios se realicen debe presionar el botón **Guardar**.

SELECCIONE UN DIA FESTIVO: NAVIDAD

INFORMACIÓN DEL DÍA FESTIVO

CÓDIGO DEL DÍA FESTIVO: 1

NOMBRE DEL DÍA FESTIVO: NAVIDAD

FECHA DE CELEBRACIÓN: ago 1, 2012 

ESTADO: ACTIVO

Guardar Cancelar

Figura 7.5.13: Formulario de Modificación de Día Festivo
Fuente: El Autor

- **Administrar Multas, Administrar Bonificaciones**

Se agruparon estos dos sub menús ya que su manejo es similar al igual que sus formularios, tienen las opciones de:

Nuevo: en el formulario que presente el sistema deben ser completados los campos de: Empleado, debe seleccionar al empleado; Detalles, debe proporcionar un motivo de la multa o bonificación; Valor, debe ingresar el valor correspondiente a la multa o bonificación; Fecha, tiene que escoger la fecha en la que va a ser aplicada la multa o bonificación. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.

INFORMACIÓN DE LA BONIFICACIÓN

EMPLEADO: GUAMAN CUENCA MARGARITA SOLEDAD

DETALLES / MOTIVOS DE LA BONIFICACIÓN:

VALOR:

FECHA DE LA BONIFICACIÓN: 

Guardar Cancelar

Figura 7.5.14: Formulario de Nueva Bonificación
Fuente: El Autor

Anular: para anular una multa o bonificación primero debe seleccionar al empleado, luego se listarán los datos correspondientes al empleado, para anular uno de los registros debe ingresar el código asignado a la multa o bonificación y luego presionar el botón **Anular**.

CÓDIGO DE LA MULTA	DETALLES / MOTIVOS DE LA MULTA	FECHA DE LA MULTA	VALOR
4	FALTA DE RESPETO	1/08/2012	20.0

Figura 7.5.15: Formulario de Anulación de Multa
Fuente: El Autor

- **Administrar Adelantos, Administrar Descuentos**

Se agruparon estos dos sub menús ya que su manejo es similar al igual que sus formularios, tienen las opciones de:

Nuevo: En el formulario de Nuevo Adelanto o Nuevo Descuento deben ser completados los campos de: Empleado, debe seleccionar al empleado al que se le va a aplicar el descuento o adelanto; Detalles, debe proporcionar una descripción o motivo breve; Valor, cantidad a ser debitada al empleado; Fecha Inicio, debe seleccionar la fecha desde cuándo se va a realizar el débito; Número de meses a diferir, número de meses en los que se va a dividir la deuda.

Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.

INFORMACIÓN DEL DESCUENTO

EMPLEADO: ORELLANA MALDONADO FRANCISCO JAVIER

DETALLES DEL DESCUENTO:

VALOR:

FECHA DE INICIO DEL DESCUENTO:

NÚMERO DE MESES A DIFERIR:

Guardar Cancelar

Figura 7.5.16: Formulario de Nuevo Descuento
Fuente: El Autor

Anular: para anular un descuento o adelanto primero debe seleccionar al empleado, luego se listarán los datos correspondientes al empleado, para anular uno de los registros debe ingresar el código asignado a la multa o bonificación y luego presionar el botón **Anular**.

Descuentos - Anular

SELECCIONE UN EMPLEADO: ORELLANA MALDONADO FRANCISCO JAVIER

INFORMACIÓN DEL DESCUENTO

CÓDIGO DEL DESCUENTO	DETALLES DEL DESCUENTO	VALOR	NÚMERO DE MESES A DIFERIR	FECHA DE INICIO
5	PAGO DE UNIFORMES	100.0	5	1/08/2012

Confirme el código del descuento que desea anular:

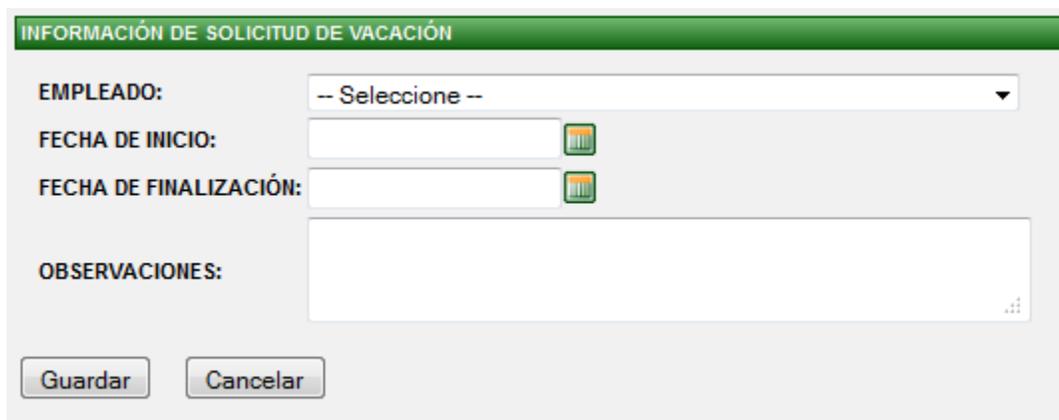
Anular Cancelar

Figura 7.5.17: Formulario de Anulación de Descuento
Fuente: El Autor

- **Administrar Vacaciones**

Para acceder a la administración de vacaciones debe seleccionar el menú de Administración y seleccionar la opción Vacaciones, se presentarán las siguientes opciones:

Programar Vacaciones: en el formulario que se presenta debe completar los datos de: Empleado, debe seleccionar al empleado que solicita las vacaciones; Fecha inicio y Fecha de finalización, debe seleccionar las fechas de inicio y fin de las vacaciones; y Observaciones, debe realizarse cualquier observación que sea necesaria. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.



El formulario, titulado "INFORMACIÓN DE SOLICITUD DE VACACIÓN", contiene los siguientes campos:

- EMPLEADO:** Un menú desplegable con el texto "-- Seleccione --".
- FECHA DE INICIO:** Un campo de texto con un icono de calendario.
- FECHA DE FINALIZACIÓN:** Un campo de texto con un icono de calendario.
- OBSERVACIONES:** Un área de texto grande para ingresar comentarios.

En la parte inferior del formulario se encuentran dos botones: "Guardar" y "Cancelar".

Figura 7.5.18: Formulario de Nuevo Período de Vacaciones
Fuente: El Autor

Anular Vacaciones: para anular un período de vacaciones debe seleccionar al empleado, luego se listarán los datos correspondientes al empleado, para anular uno de los registros debe ingresar el código asignado al período de vacaciones y luego presionar el botón **Anular**.

Vacaciones - Anular

SELECCIONE UN EMPLEADO: ORELLANA MALDONADO FRANCISCO JAVIER

INFORMACIÓN DE SOLICITUD DE VACACIÓN

CÓDIGO DE SOLICITUD DE VACACIÓN	FECHA DE INICIO	FECHA DE FINALIZACIÓN	OBSERVACIONES
1	1/08/2012	15/08/2012	VACACIONES DEL PERÍODO 2012

Confirmo el código de la vacación que desea anular: 1

Anular Cancelar

Figura 7.5.19: Formulario de Anulación de Período de Vacaciones
Fuente: El Autor

- **Administrar Permisos**

Para acceder a la administración de permisos debe seleccionar el menú de Administración y seleccionar la opción Permisos, se presentarán las siguientes opciones:

Nuevo Permiso: en el formulario de nuevo permiso deben completarse los siguientes campos: Permiso por, tiene que seleccionar si el permiso es por horas o por días; Empleado, debe seleccionar al empleado que solicita el permiso; Tipo de Permiso, debe seleccionar si el permiso es Personal o Empresarial; en el caso de que el permiso sea por días, debe seleccionar la fecha de inicio y fin del permiso, en el caso de que sea por horas debe especificar la hora de inicio y la hora de finalización en el formato hh:mm; Fecha de Solicitud, debe seleccionar la fecha en la que se solicita el permiso; Observaciones, debe ingresar algún tipo de observación adicional del permiso.

Figura 7.5.20: Formulario de Nuevo Permiso
Fuente: El Autor

Anular Permiso: para anular un permiso debe seleccionar al empleado, luego se listarán los datos correspondientes al empleado, para anular uno de los registros debe ingresar el código asignado al permiso y luego presionar el botón **Anular**.

CÓDIGO DEL PERMISO	OBSERVACIONES	HORA DE INICIO	HORA DE FINALIZACIÓN	FECHA DE INICIO	FECHA DE FINALIZACIÓN	FECHA DE SOLICITUD
1	ESTUDIOS	08:09:00	16:00:00			2/08/2012

Figura 7.5.21: Formulario de Anulación de Permiso
Fuente: El Autor

- **Administrar Calendario de Pagos**

Para acceder a la administración de calendario de pagos debe seleccionar el menú de Administración y seleccionar la opción Calendario de Pagos, se presentarán las siguientes opciones:

Nueva Fecha de Pago: en el formulario que presenta el sistema deben completarse los campos de: Detalles de Pago, debe detallarse a que se debe el pago; Valor, debe especificarse el monto del pago; Fecha de Pago, tiene que seleccionar la fecha máxima de pago; Estado, por defecto se deja el valor Pendiente. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.



The image shows a web form titled "INFORMACIÓN DE FECHA DE PAGO". It contains the following elements:

- DETALLES DEL PAGO:** A large text input area.
- VALOR:** A text input field.
- FECHA DE PAGO:** A date selection field with a calendar icon.
- ESTADO:** A dropdown menu with "PENDIENTE" selected.
- Buttons:** "Guardar" and "Cancelar" buttons at the bottom.

Figura 7.5.22: Formulario de Nueva Fecha de Pago
Fuente: El Autor

Modificar Fecha de Pago: en el formulario presentado por el sistema debe seleccionar el pago que desea modificar dando clic en la columna acción, en la nueva ventana que se abrirá podrá modificar toda la información del pago. Para que los cambios se realicen debe hacer clic en el botón **Guardar** y en el listado se visualizarán los cambios realizados.

Calendario de Pagos - Modificar

INFORMACIÓN DE FECHA DE PAGO

CÓDIGO DE FECHA DE PAGO	DETALLES DEL PAGO	VALOR	FECHA DE PAGO	ESTADO	ACCIÓN
1	IESS PRUEBA	3000.0	12/07/2012	PAGADO	

Cancelar

Modificar Datos de Calendario Pago

CÓDIGO DE FECHA DE PAGO:

DETALLES DEL PAGO:

VALOR:

FECHA DE PAGO: 

ESTADO:

Guardar Cancelar

Figura 7.5.23: Formulario de Modificación de Fecha de Pago
Fuente: El Autor

- **Administración de Cargos**

Para acceder a la administración de cargos debe seleccionar el menú de Administración y seleccionar la opción Cargos, se presentarán las siguientes opciones:

Nuevo Cargo: debe llenar todos los campos que presenta el formulario: Nombre del cargo debe ser un nombre descriptivo del cargo; Departamento, tiene que seleccionar al departamento que pertenecerá el cargo; Descripción, debe ingresar una breve descripción de las funciones que realizará el empleado asignado al cargo; Estado por defecto es Activo. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.

Figura 7.5.24: Formulario de Nuevo Cargo
Fuente: El Autor

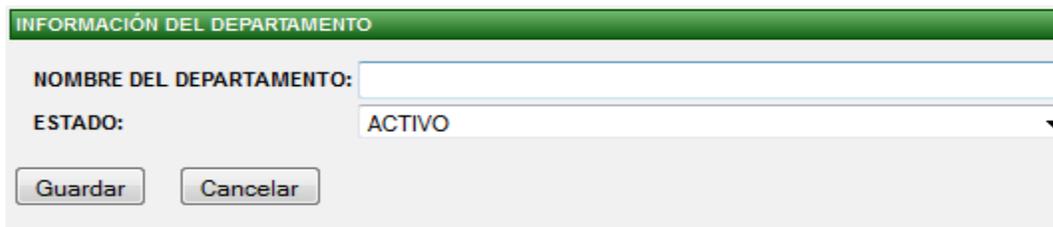
Modificar Cargo: en el formulario que se presenta primero debe seleccionar el cargo que desea modificar, luego debe cambiar la información requerida con el formato establecido. Para que los cambios se realicen debe presionar el botón **Guardar**.

Figura 7.5.25: Formulario de Modificación de Cargo
Fuente: El Autor

- **Administrar Departamentos**

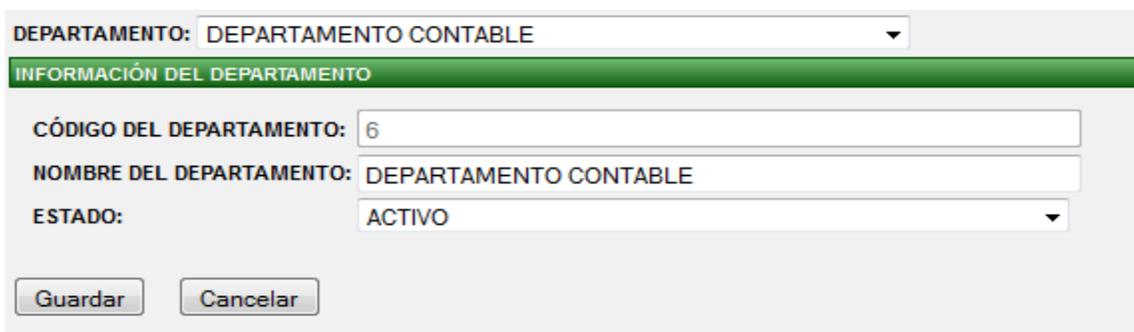
Para acceder a la administración de departamentos debe seleccionar el menú de Administración y seleccionar la opción departamentos, se presentarán las siguientes opciones:

Nuevo Departamento: debe llenar todos los campos que presenta el formulario: Nombre del cargo debe ser un nombre descriptivo del departamento; Estado por defecto es Activo. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.



*Figura 7.5.26: Formulario de Nuevo Departamento
Fuente: El Autor*

Modificar Departamento: en el formulario que se presenta primero debe seleccionar el departamento que desea modificar, luego debe cambiar la información requerida con el formato establecido. Para que los cambios se realicen debe presionar el botón **Guardar**.



*Figura 7.5.27: Formulario de Modificación de Departamento
Fuente: El Autor*

- **Administrar Usuarios**

Para acceder a la administración de usuarios debe seleccionar el menú de Administración y seleccionar la opción usuarios, se presentarán las siguientes opciones:

Nuevo Usuario: debe llenar todos los campos del formulario que presenta el sistema: Empleado, al seleccionar el empleado se listarán todos los usuarios que ya están creados para su uso; Nombre de usuario, debe digitar el identificador de usuario que servirá para iniciar sesión; Clave, debe ser digitada por el propio empleado para evitar confusiones; Perfil, debe escoger entre Administrador o Empleado; Estado, por defecto tiene el valor de Activo. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.

CÓDIGO DE USUARIO	NOMBRE DE USUARIO	PERFIL	ESTADO
4	forellana	EMPLEADO	ACTIVO

Figura 7.5.28: Formulario de Nuevo Usuario
Fuente: El Autor

Modificar Usuario: en el formulario presentado por el sistema debe seleccionar el usuario que desea modificar dando clic en la columna acción, en la nueva ventana que se abrirá podrá modificar toda la información del usuario. Para que los cambios

se realicen debe hacer clic en el botón **Guardar** y en el listado se visualizarán los cambios realizados.

The screenshot shows a web application interface for user management. At the top, there is a table titled "USUARIOS" with the following data:

CÓDIGO DE USUARIO	NOMBRE DE USUARIO	EMPLEADO	PERFIL	ESTADO	ACCIÓN
3	vmancheno	MANCHENO GUAMÁN VERÓNICA LIZETH	ADMINISTRADOR	ACTIVO	
4	forellana	ORELLANA MALDONADO FRANCISCO JAVIER	EMPLEADO	ACTIVO	

Below the table, there is a modal form titled "Usuario" for editing user 4. The form contains the following fields:

- CÓDIGO DE USUARIO: 4
- EMPLEADO: ORELLANA MALDONADO FRANCISCO JAVIER
- NOMBRE DE USUARIO: forellana
- CLAVE: [masked with dots]
- PERFIL: EMPLEADO
- ESTADO: ACTIVO

Buttons for "Guardar" and "Cancelar" are located at the bottom of the modal. A red double-headed arrow points to the "ACCIÓN" column of the table, indicating the edit action.

Figura 7.5.29: Formulario de Modificación de Usuario
Fuente: El Autor

- **Administrar Menús**

Para acceder a la administración de menús debe seleccionar el menú de Administración y seleccionar la opción Menú, se presentarán las siguientes opciones:

Nuevo Menú: debe llenar todos los campos que presenta el formulario: Nombre del menú debe ser un nombre descriptivo de la función del menú; URL, dirección de acceso a la página. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.

The screenshot shows a form titled "INFORMACIÓN DEL MENU" with the following fields:

- NOMBRE DEL MENU: [text input field]
- URL DEL MENU: [text input field]

Buttons for "Guardar" and "Cancelar" are located at the bottom of the form.

Figura 7.5.30: Formulario de Nuevo Menú
Fuente: El Autor

Modificar Menú: en el formulario presentado por el sistema debe seleccionar el menú que desea modificar dando clic en la columna acción, en la nueva ventana que se abrirá podrá modificar toda la información del menú. Para que los cambios se realicen debe hacer clic en el botón **Guardar** y en el listado se visualizarán los cambios realizados.

CÓDIGO MENU	NOMBRE DEL MENU	URL DEL MENU	ACCIÓN
1	ADELANTOS	/pages/adelantos/Adelanto.seam	[Icon]
2	BONIFICACIONES	/pages/bonificaciones/Bonificacion.seam	[Icon]
3	CALENDARIO DE PAGOS		
4	CARGOS		
5	DEPARTAMENTOS		
6	DESCUENTOS		
7	ANULAR DESCUENTOS		
8	DIAS FESTIVOS		
9	EMPLEADOS		
10	MULTAS		

Modificar Datos de Menu

CÓDIGO MENU:

NOMBRE DEL MENU:

URL DEL MENU:

Figura 7.5.31: Formulario de Modificación de Menú
Fuente: El Autor

Asignar Menú: El sub menú tiene dos opciones, *Asignar Menú Administrador* y *Asignar Menú Empleado*, el formulario es similar para las dos opciones, se despliegan dos listados de menús. El listado superior muestra los menús asignados al perfil, el listado inferior muestra los menús disponibles para ser asignados al perfil. Para eliminar un menú del listado superior debe dar clic en el símbolo menos (-) de la columna Acción, luego de esto se mostrará una nueva ventana con los datos del menú y debe presionar el botón **Remove**.

Para asignar un nuevo menú al listado de menús asignados, debe dar clic en el símbolo más (+) de la columna Acción, luego de esto se mostrará una nueva ventana con los datos del menú y debe presionar el botón **Asignar**.

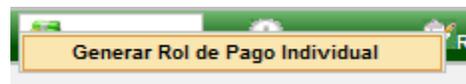
MENUS ASIGNADOS			
CÓDIGO MENU	NOMBRE DEL MENU	URL DEL MENU	ACCIÓN
16	EDITAR EMPLEADOS	/pages/empleados/EmpleadoEditable.seam	-
19	INFORMACIÓN ÉXITO REGISTROS MODIFICADOS	/pages/info/ModificacionCorrecta.seam	-
27	USUARIOS MODIFICAR	/pages/usuarios/UsuarioEditable.seam	-
29	MARCAR ENTRADA	/pages/marcaciones/Entrada.seam	-
34	MARCAR SALIDA	/pages/marcaciones/Salida.seam	-
35	DATOS BÚSQUEDA MACACIONES	/pages/marcaciones/DatosSeleccionMarcacion.seam	-
41	INGRESO EXITOSO	/pages/info/IngresoCorrecto.seam	-

MENUS DISPONIBLES			
CÓDIGO MENU	NOMBRE DEL MENU	URL DEL MENU	ACCIÓN
1	ADELANTOS	/pages/adelantos/Adelanto.seam	+
2	BONIFICACIONES	/pages/bonificaciones/Bonificacion.seam	+
3	CALENDARIO DE PAGOS	/pages/calendarioPagos/CalendarioPago.seam	+
4	CARGOS	/pages/cargos/Cargo.seam	+
5	DEPARTAMENTOS	/pages/departamentos/Departamento.seam	+
6	DESCUENTOS	/pages/descuentos/Descuento.seam	+
7	ANULAR DESCUENTOS	/pages/descuentos/DescuentoEditable.seam	+
8	DIAS FESTIVOS	/pages/diasFestivos/DiaFestivo.seam	+
9	EMPLEADOS	/pages/empleados/Empleado.seam	+
10	MULTAS	/pages/multas/Multa.seam	+

Figura 7.5.32: Formulario de Asignación de Menú
Fuente: El Autor

- **Rol de Pagos**

Es la segunda opción del menú principal, esta opción solamente será visualizada por el usuario Administrador.



- **Generar Rol de Pagos Individual**

Para acceder a este menú debe seleccionar la opción Rol de Pagos e ingresar al sub menú Generar Rol de Pagos Individual. En el formulario que presenta el sistema debe escoger el empleado, el mes y el año del cual se generará el rol de pagos, una vez seleccionados los parámetros debe presionar el botón **Generar**.

Si la información se generó correctamente se mostrará en pantalla el detalle de los valores calculados y se mostrará el botón **Guardar**, si la información está correcta

presione este botón para guardar el rol generado, cabe mencionar que una vez guardada la información no podrá ser modificada, se mostrará un mensaje como el de la Figura 7.5.8 si no existieron errores al guardar la información.

La impresión del rol de pagos se podrá realizar ingresando al menú de Reportes.

Figura 7.5.33: Formulario para generar el rol de pagos
Fuente: El Autor

RESULTADOS	
ROL DE PAGOS CORRESPONDIENTE AL PERIODO DE: 2012 - 7	
APELLIDOS Y NOMBRES:	ORELLANA MALDONADO FRANCISCO JAVIER
IDENTIFICACIÓN:	1720961885
DIRECCIÓN:	ANONAS Y CALLE B
TELÉFONO CELULAR:	084279212
INGRESOS:	
REMUNERACIÓN BÁSICA:	500.0
APOORTE IESS EMPLEADOR**:	55.75
BONIFICACIONES:	0.0
HORAS EXTRAS:	0.0
TOTAL INGRESOS:	500.0
(**) Valor de Aporte IESS Empleador es informativo, el monto es depositado por el empleador en las oficinas del IESS	
EGRESOS:	
APOORTE IESS EMPLEADO:	46.75
ADELANTOS:	0.0
ATRASOS:	10.0
DESCUENTOS:	0.0
HORAS FALTANTES:	148.82
MULTAS:	0.0
TOTAL EGRESOS:	195.57
TOTAL A RECIBIR: 304.43	
Guardar	

Figura 7.5.34: Rol de Pagos generado
Fuente: El Autor

- **Registro**

Es la tercera opción del menú principal, desde esta opción se puede acceder a Marcar Entrada, Marcar Salida, Marcar Entrada sin Biometría, Marcar Salida sin Biometría, Modificar Marcaciones. El sub menú Modificar Marcaciones sólo puede ser accedido por usuarios con perfil Administrador.



- **Marcar Entrada, Marcar Salida, Marcar Entrada sin Biometría, Marcar Salida sin Biometría**

Para que el empleado marque la hora de entrada o salida necesita del dispositivo biométrico, o en su defecto el número de identificación del usuario autenticado en el caso que no se use biometría. Las opciones de marcar entrada/salida sin biometría serán habilitadas únicamente cuando no se disponga del dispositivo biométrico.

Para marcar la entrada o salida usando el dispositivo biométrico, el empleado debe seleccionar el menú Registro y luego seleccionar la opción Marcar Entrada o Marcar Salida, cuando el sistema muestre la nueva ventana debe hacer clic sobre la imagen del reloj y colocar cualquiera de los dedos registrados previamente sobre el dispositivo biométrico para el proceso de verificación. Una vez verificada la identidad, se mostrará una ficha con información del empleado.

En el caso de Marcar Entrada / Salida sin biometría el empleado debe ingresar su número de identificación en el campo de texto mostrado en el formulario correspondiente y presionar el botón **Marcar Entrada / Salida**. Luego de validar que la identificación corresponda al usuario autenticado, se mostrará la ficha con la información del empleado.

Nota: cada empleado debe realizar este proceso con su propio usuario, caso contrario no se podrá verificar la identidad.



Figura 7.5.35: Pantalla para Marcar Entrada / Salida
Fuente: El Autor

	DATOS PERSONALES NOMBRES: FRANCISCO JAVIER APELLIDOS: ORELLANA MALDONADO
	INFORMACIÓN DEL EMPLEADO TURNO DE TRABAJO: <input type="text" value="MAÑANA"/> HORA DE ENTRADA: 01:37:00
	MENSAJE PERSONAL: <input type="text" value="REVISAR PEDIDOS"/>

Identidad verificada exitosamente.

Figura 7.5.36: Pantalla para Marcar Entrada / Salida
Fuente: El Autor

- **Modificar Marcaciones**

Para acceder a este menú debe seleccionar la opción Registro e ingresar al sub menú Modificar Marcaciones. En el formulario que presenta el sistema debe seleccionar el empleado y la fecha que se desea modificar, una vez seleccionados debe hacer clic en el botón **Buscar Marcaciones**, después de esto el sistema presenta un nuevo formulario donde se muestra toda la información de las horas de entrada y salida con la información proporcionada previamente. Puede modificar los campos de hora de entrada, hora de salida, horas laboradas, siempre y cuando conserve el formato de hh:mm:ss, también podrá cambiar el valor de atraso entre SI

y NO. Una vez llenados todos los campos correctamente presionar el botón **Guardar** y se mostrará un mensaje como el de la Figura 7.5.8; si no existieron errores.

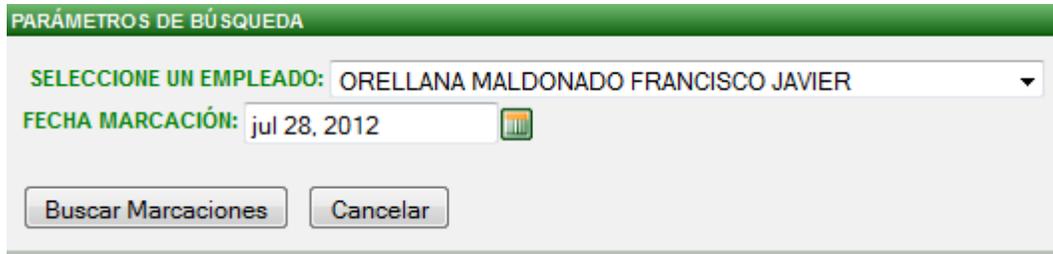


Figura 7.5.37: Formulario de búsqueda de marcaciones de entrada / salida
Fuente: El Autor

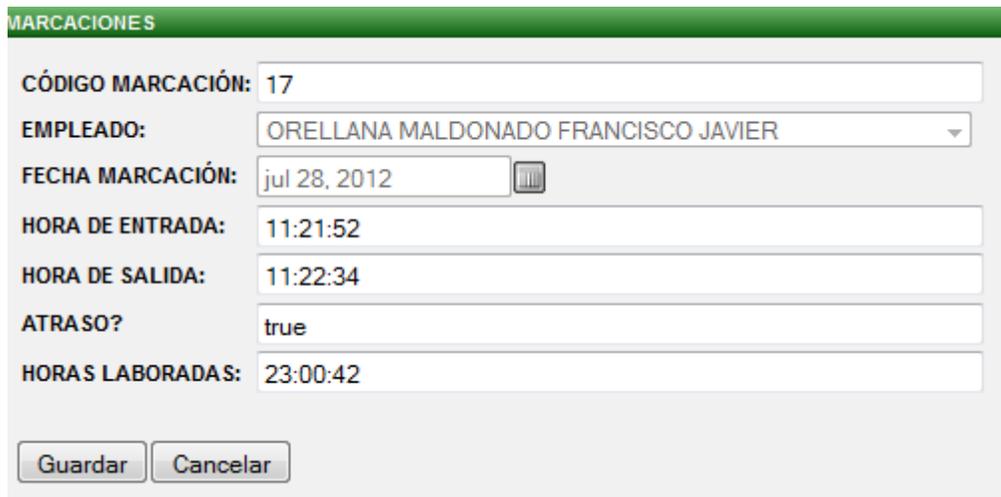


Figura 7.5.38: Resultado de búsqueda de marcaciones de entrada / salida
Fuente: El Autor

- **Reportes**

Es la cuarta opción del menú principal, que sólo puede ser accedida por un usuario con perfil Administrador, desde esta opción se puede acceder a Reporte de Atrasos, Reporte de Horas Laboradas, Reporte de Permisos y Reporte de Rol de Pago.



Los formularios de los reportes son similares, una vez abierto el formulario para ingresar los parámetros para generar los reportes, debe seleccionar el empleado, la fecha inicial y la fecha final que corresponde al período del cual se seleccionará la información. Una vez establecidos los parámetros debe hacer clic sobre el link **Generar Reporte** y el sistema mostrará el reporte en formato PDF con los parámetros establecidos previamente.

Parametros de busqueda

Empleado: ORELLANA MALDONADO FRANCISCO JAVIER

Fecha Inicial: 1/7/2012

Fecha Final: 31/7/2012

[Generar Reporte](#)

Figura 7.5.39: Formulario de parámetros para generar reportes
Fuente: El Autor

05/08/2012

ROL DE PAGOS INDIVIDUAL

CORRESPONDIENTE AL PERÍODO DE: 01/07/12

APELLIDOS: ORELLANA MALDONADO
 NOMBRES: FRANCISCO JAVIER
 IDENTIFICACIÓN: 1720961885

INGRESOS	DETALLE
REMUNERACIÓN BÁSICA	500.0
APORTE IEES EMPLEADOR **	55.75
BONIFICACIONES	0.0
HORAS EXTRAS	0.0
TOTAL INGRESOS	500.0
<small>(*) Valor de Aporte IEES Empleador es informativo, el monto es depositado por el empleador en las oficinas del IEES.</small>	
EGRESOS	
APORTE IEES EMPLEADO	46.75
ADELANTOS	0.0
ATRASOS	10.0
DESCUENTOS	0.0
HORAS FALTANTES	148.82
MULTAS	0.0
TOTAL EGRESOS	195.57
TOTAL A RECIBIR	304.43

Figura 7.5.40: Reporte generado de rol de pagos
Fuente: El Autor

- **Auditoría**

Es la quinta opción del menú principal, que sólo puede ser accedida por un usuario con perfil Administrador, desde esta opción se puede acceder a Consulta de Registros.



- **Consulta de Registros**

Para acceder a este menú debe seleccionar la opción Auditoría e ingresar al submenú Consulta de Registros. En el formulario que presenta el sistema debe seleccionar el empleado, la fecha inicial y la fecha final que corresponde al período del cual se seleccionará la información. Una vez establecidos los parámetros debe hacer clic en el botón **Buscar Registros**. Luego de realizar la búsqueda, el sistema muestra los registros auditados de todos los usuarios del empleado seleccionado.

Un formulario web con un encabezado verde que dice "Criterios de Búsqueda". Contiene tres campos de entrada: "SELECCIONE UN EMPLEADO:" con un menú desplegable que muestra "MANCHENO GUAMÁN VERÓNICA LIZETH"; "FECHA DE INICIO:" con un campo de texto que muestra "jul 1, 2012" y un icono de calendario; y "FECHA DE FINALIZACIÓN:" con un campo de texto que muestra "ago 5, 2012" y un icono de calendario. En la parte inferior hay un botón que dice "Buscar Registros".

*Figura 7.5.41: Criterios de búsqueda de registros de auditoría
Fuente: El Autor*

CÓDIGO DE AUDITORÍA	ACCIÓN REALIZADA	FECHA ACCIÓN	IP ACCESO
4	INSERT INTO alprostart.empleado VALUES (nextval(alprostart.empleado_id_empleado_seq),1,2,1,1, 1234567899 , VINICIO , MANCHENO , CALLE B LOTE 90 Y CALLE D , 023281814 , 083912344 , alprostart@hotmail.com , Mon Jul 02 00:00:00 COT 2012 , aasd , Sun Jul 01 00:00:00 COT 2012 , 4, ORH- , hola ,4000,0,40,0.?)	16/07/2012	127.0.0.1
5	INSERT INTO alprostart.empleado_cargo VALUES (nextval(alprostart.empleado_cargo_id_empleado_cargo_seq),(SELECT MAX (id_empleado) FROM alprostart.empleado),4,CURRENT_DATE)	16/07/2012	127.0.0.1
6	UPDATE alprostart.empleado SET id_turno=1, id_estado_civil=2, idsexo=1,id_estado=1, identificacion= 1721837779 ,nombres= VERÓNICA LIZETH , apellidos= MANCHENO GUAMÁN ,direccion= CALLE B E15-B Y ANONAS, CONJUNTO PRADERAS DEL EDEN, DPTO #3, telefono= 023281814 ,telefono_celular= 095169418 , correo_electronico= lizethmancheno@yahoo.com, fecha_nacimiento= Thu Jan 07 00:00:00 COT 1988 , experiencia_laboral= SECRETARIA EN ALPROSART ,fecha_ingreso= Tue Jan 01 00:00:00 COT 2008 ,cargas_familiares = 1,tipo_sangre= ORH+ ,mensaje_personal= LIMPIAR OFICINA ,remuneracion_basica=298.5,valor_hora_extra=1.8 WHERE id_empleado=2	16/07/2012	127.0.0.1
7	INSERT INTO alprostart.empleado_cargo VALUES (nextval(alprostart.empleado_cargo_id_empleado_cargo_seq),2,4,CURRENT_DATE)	16/07/2012	127.0.0.1
8	INSERT INTO alprostart.turno VALUES (nextval(alprostart.turno_id_turno_seq),1, TARDE , 13:30 , 19:30)	16/07/2012	127.0.0.1
9	UPDATE alprostart.turno SET id_estado=1,nombre_turno= MAÑANA ,hora_entrada_turno= 07:00:00 ,hora_salida_turno= 13:00:00 WHERE id_turno=1	16/07/2012	127.0.0.1
11	INSERT INTO alprostart.descuento VALUES (nextval(alprostart.descuento_id_descuento_seq),3, PRUEBA DESCUENTO AUDIT ,300,0,3, Wed Aug 01 00:00:00 COT 2012)	16/07/2012	127.0.0.1
12	UPDATE alprostart.empleado SET id_turno=1, id_estado_civil=2, idsexo=1,id_estado=1, identificacion= 1234567899 ,nombres= VINICIO , apellidos= MANCHENO ,direccion= CALLE B LOTE 90 Y CALLE D ,telefono= 023281814 ,telefono_celular= 083912344 , correo_electronico= alprostart@hotmail.com , fecha_nacimiento= Mon Jul 02 00:00:00 COT 2012 , experiencia_laboral= aasd ,fecha_ingreso= Sun Jul 01 00:00:00 COT 2012 ,cargas_familiares = 4,tipo_sangre= ORH+ ,mensaje_personal= hola ,remuneracion_basica=4000.0,valor_hora_extra=40.0 WHERE id_empleado=11	16/07/2012	127.0.0.1
13	INSERT INTO alprostart.empleado_cargo VALUES (nextval(alprostart.empleado_cargo_id_empleado_cargo_seq),11,4,CURRENT_DATE)	16/07/2012	127.0.0.1
14	INSERT INTO alprostart.cargo VALUES (nextval(alprostart.cargo_id_cargo_seq),3,1, GERENTE , guiar)	16/07/2012	127.0.0.1

Figura 7.5.42: Resultados de búsqueda de registros de auditoría
Fuente: El Autor

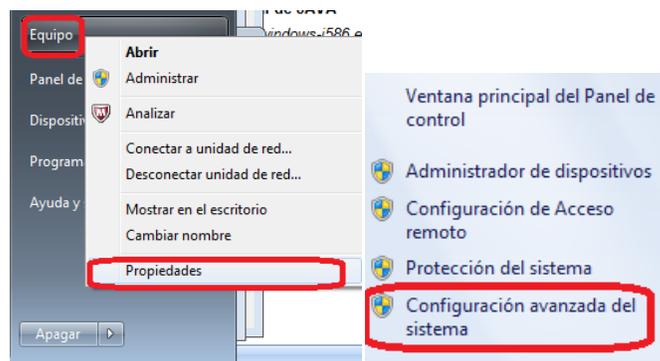
ANEXO VI: MANUAL DE INSTALACIÓN

El manual de instalación se detalla para la configuración que se tiene en el servidor de ALPROSART; en el CD adjunto se encuentran todos los instaladores y aplicativos necesarios para el funcionamiento del sistema.

- **Instalación de máquina virtual de JAVA**

Ejecutar el archivo *jdk-6u25-windows-x64.exe* ubicado en el directorio *Instaladores* del CD adjunto, siga las instrucciones del asistente de instalación.

Luego definir la variable de entorno *JAVA_HOME*. Para crearla debe hacer clic derecho en Equipo, luego clic en Propiedades, seleccionar la opción Configuración avanzada del sistema, en la pestaña Opciones Avanzadas, hacer clic en el botón Variables de entorno, en el panel Variables de sistema hacer clic en Nueva. En la nueva ventana ingresar el Nombre de la variable: *JAVA_HOME* y en Valor de la variable escriba el directorio donde se instaló la máquina virtual de JAVA: *C:\Program Files\Java\jdk1.6.0_25*.



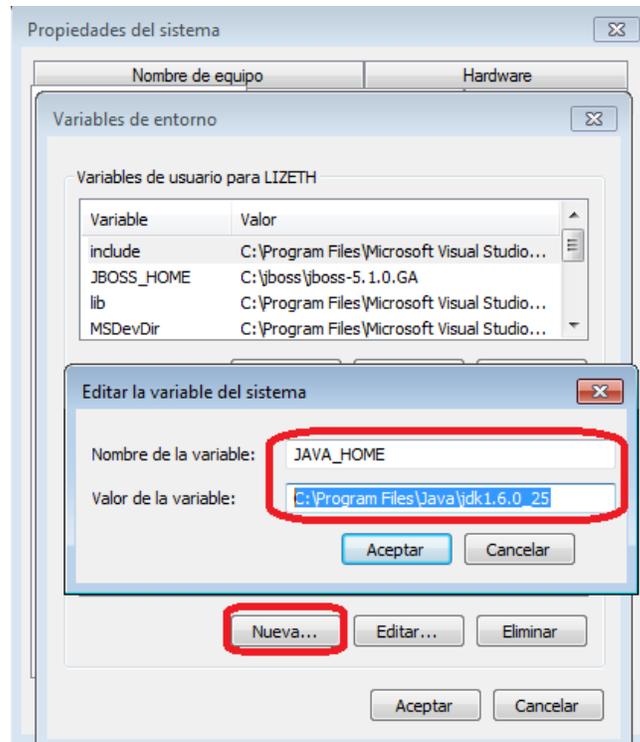


Figura 7.6.1: Pasos para crear variable JAVA_HOME
Fuente: El Autor

- **Instalación de motor de base de datos PostgreSQL**

Ejecutar el archivo *postgresql-8.4.11-1.exe* ubicado en el directorio *Instaladores* del CD adjunto, siga las instrucciones del asistente de instalación, debe mantener los directorios de instalación por defecto, cuando el asistente solicite la clave de *super-usuario postgres* debe introducir una contraseña segura ya que este usuario administra todos los objetos de la base de datos, en el puerto debe dejar el valor por defecto 5432.

- **Instalación de servidor de aplicaciones JBoss**

Copiar la carpeta *jboss* ubicada en el directorio *Instaladores* del CD adjunto, cópiela al disco *C:* del servidor de ALPROSART, dentro del nuevo directorio *C:/jboss* se encuentra una carpeta llamada *jboss-5.1.0.GA* a este directorio se le llamará *JBOSS_HOME*.

Dentro del directorio *JBOSS_HOME/servers/default/lib* se encuentran todas las librerías para el manejo del dispositivo biométrico, conexión a la base de datos, y generación de reportes.

Dentro del directorio *JBOSS_HOME/servers/default/deploy* se encuentran los archivos de configuración (*ALPROSART-ds.xml* y *mail-service.xml*) y el archivo .ear (*ALPROSART-ear.ear*) que contiene toda la aplicación.

- **Creación de la base de datos**

Abrir el administrador de PostgreSQL (pgAdmin III), y conectarse con la cuenta de super-usuario postgres con la clave ingresada en la instalación. Primero debe crear el usuario propietario de la base de datos ALPROSART y el usuario transaccional que será utilizado para la conexión desde el sistema.

Para esto debe hacer clic derecho en la opción *Login Roles* y seleccionar *New Login Role*. El usuario propietario tiene como nombre *alprosart* y el usuario transaccional tiene como nombre *alprosart_t*, la clave del usuario transaccional tiene que ser la misma que se especifica en el archivo *ALPROSART-ds.xml* ubicado en el directorio *JBOSS_HOME/servers/default/deploy*.

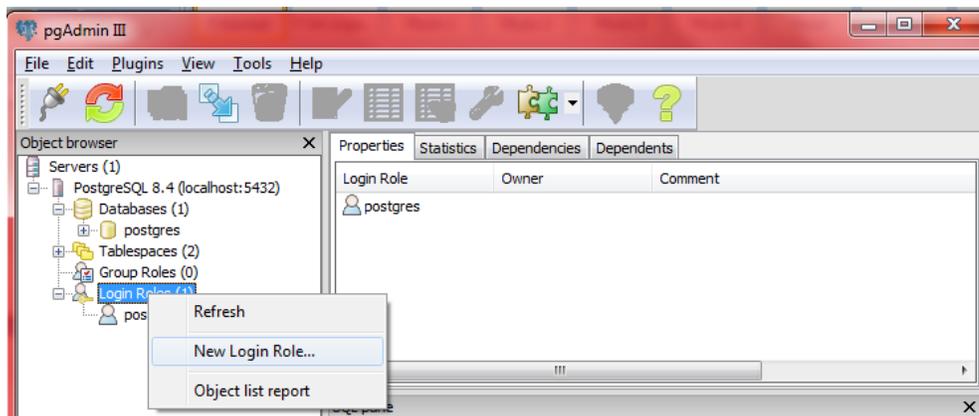
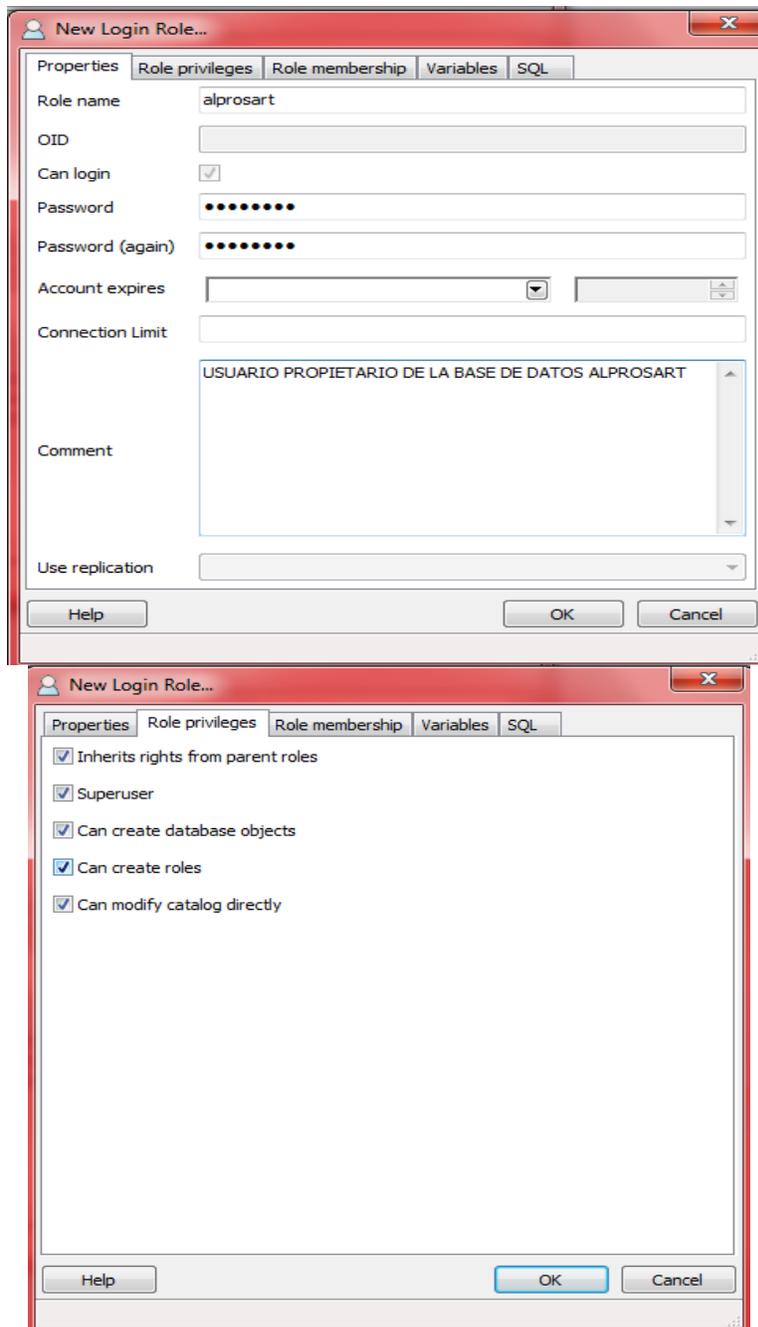


Figura 7.6.2: Creación de usuarios de base de datos
Fuente: El Autor



*Figura 7.6.3: Privilegios del usuario alprosart
Fuente: El Autor*

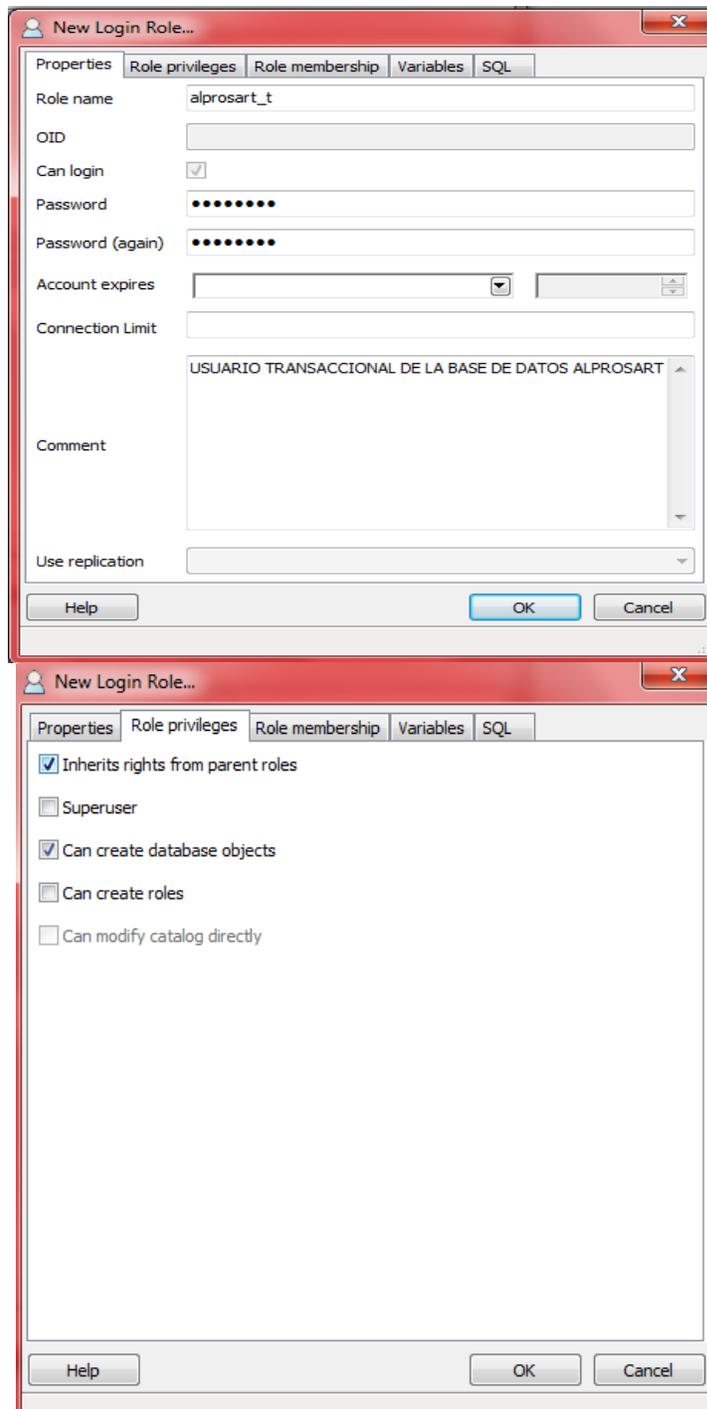


Figura 7.6.4: Privilegios del usuario alprostart_t
Fuente: El Autor

Una vez creados los usuarios, debe crear la base de datos ALPROSART, haciendo clic derecho en la opción *Databases* y luego seleccionar *New Database*, en la nueva ventana llenar la información de *Name* y *Owner* como se muestra en la Figura 7.6.5.

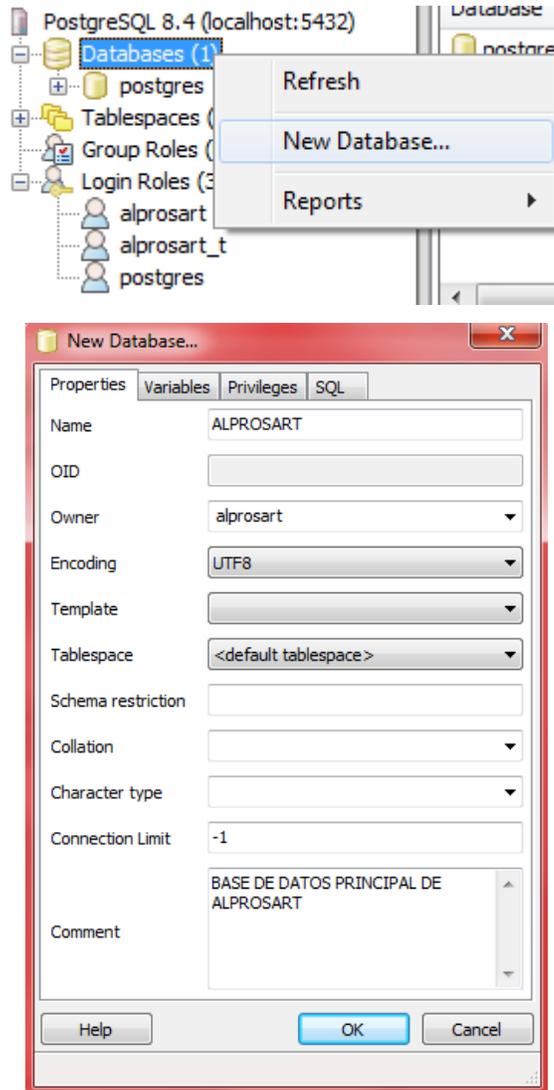


Figura 7.6.5: Creación de base de datos ALPROSART
Fuente: El Autor

Cuando la base de datos haya sido creada, debe restaurar los datos iniciales para el primer uso del sistema. El primer paso es hacer clic derecho sobre la base de datos creada y seleccionar la opción *Restore*. En la nueva ventana debe escoger el archivo

Base de datos *alprostart.backup* que se encuentra en la carpeta *Base de Datos* del CD adjunto y presionar el botón *OK* para que se inicie el proceso.

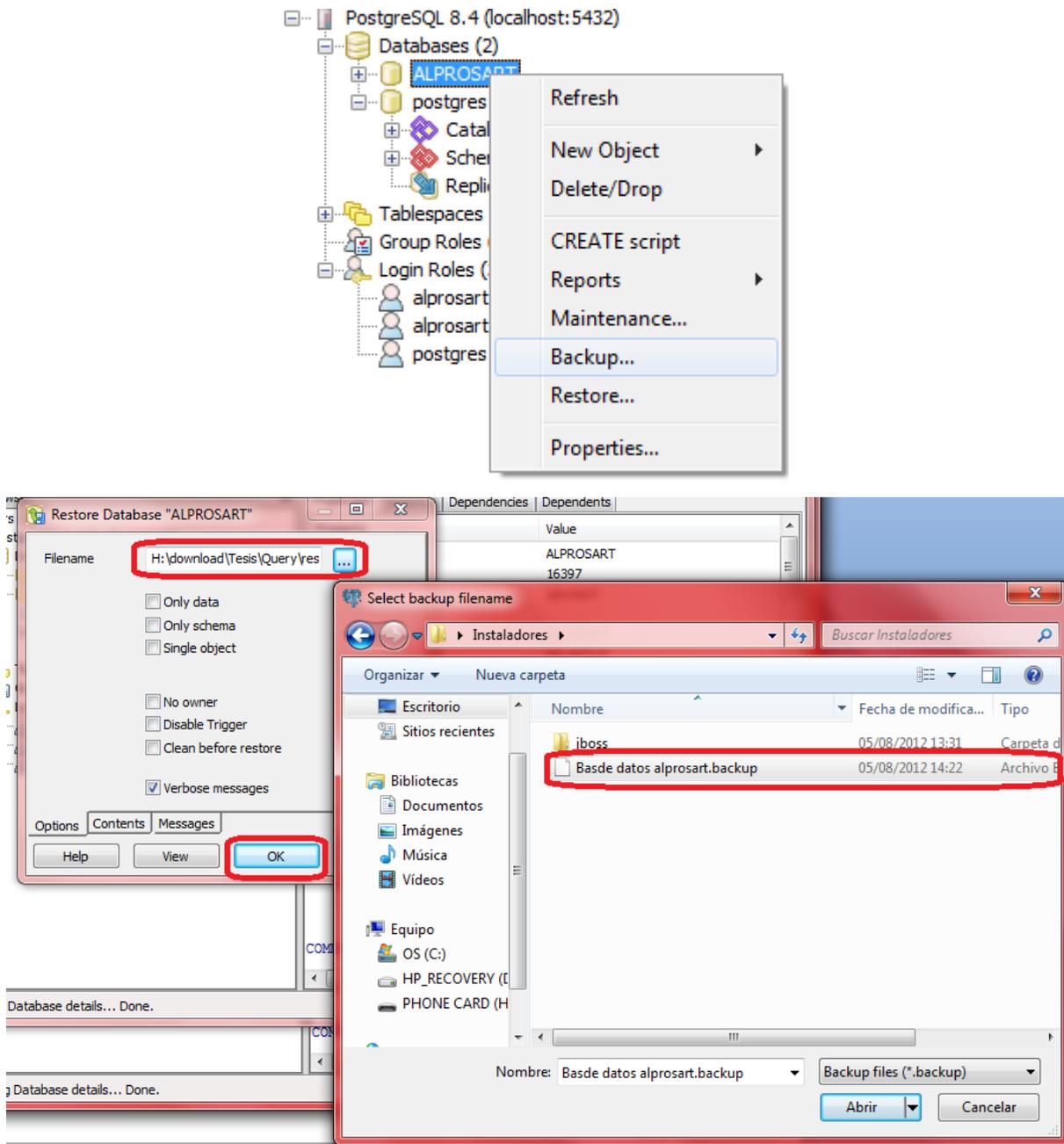


Figura 7.6.6: Restauración de base de datos ALPROSART
Fuente: El Autor

Luego que termine el proceso todos los objetos de la base de datos habrán sido restaurados y estará listo para el primer uso.

- **Creación de tarea programada para respaldo de base de datos**

Para generar los respaldos de la base de datos automáticamente debe copiar la carpeta *Respaldos* del directorio */Base de Datos* a la raíz del disco *C:/* del servidor de ALPROSART, en este directorio se almacenarán todos los respaldos generados semanalmente.

Dentro de esta carpeta se encuentra el archivo *script_respaldoBDD.bat* el cual se encargará la tarea de respaldo. Para que este archivo se ejecute automáticamente debe generar una nueva tarea programada de Windows.

En *Equipo* debe dar clic derecho y seleccionar la opción *Administrar*, luego seleccionar la opción *Programador de Tareas* y en el panel derecho seleccionar *Crear tarea básica*.

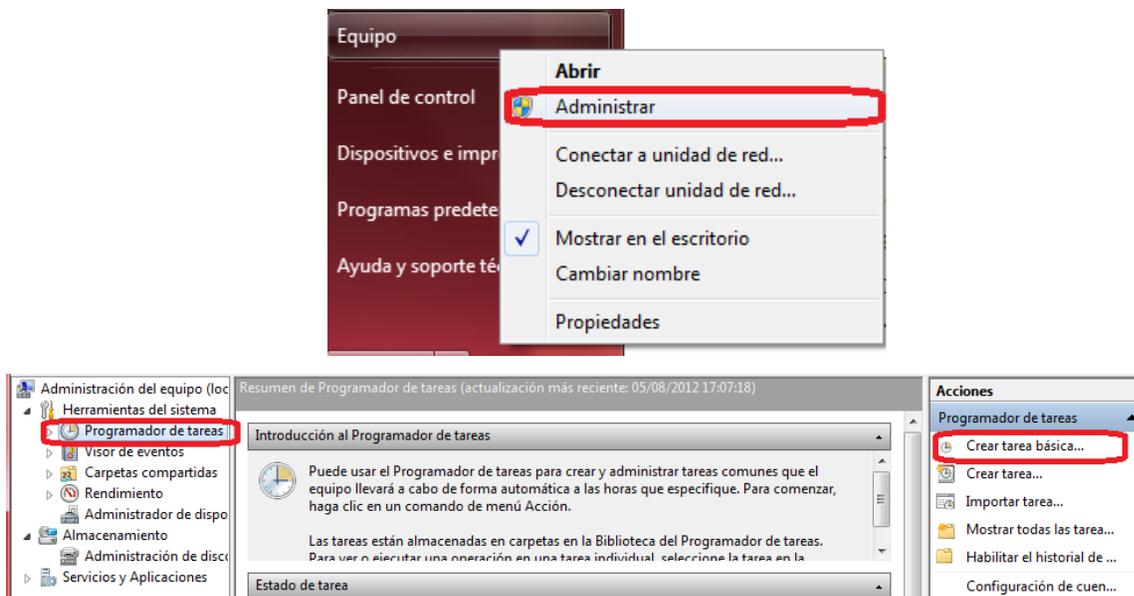
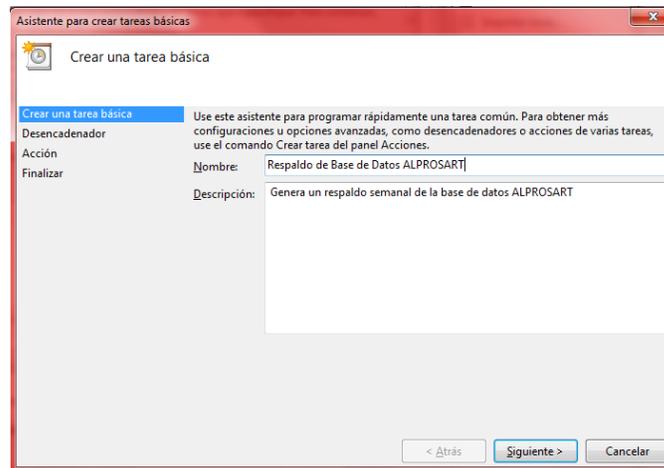


Figura 7.6.7: Primer paso para crear una nueva tarea programada en Windows 7

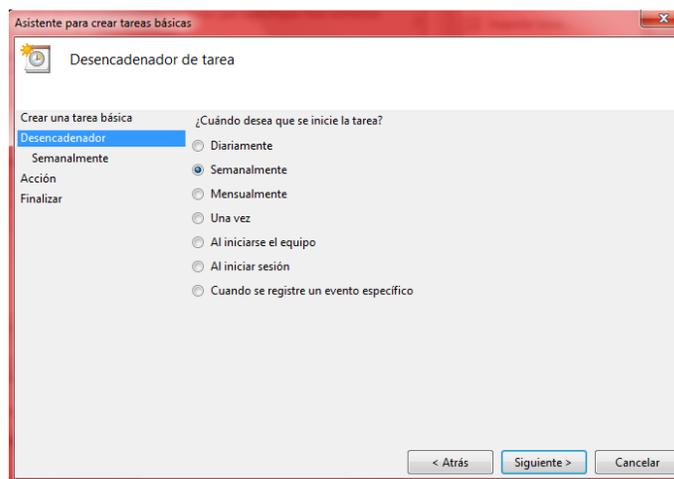
Fuente: El Autor

En el asistente para crear tareas básicas, debe proporcionar un nombre y una descripción de la tarea.



*Figura 7.6.8: Segundo paso para crear una nueva tarea programada en Windows 7
Fuente: El Autor*

El siguiente paso es especificar la frecuencia con que se ejecutará la tarea debe seleccionar *Semanalmente*.



*Figura 7.6.9: Tercer paso para crear una nueva tarea programada en Windows 7
Fuente: El Autor*

El siguiente paso es especificar el día, la hora y desde cuando se ejecutará la tarea.

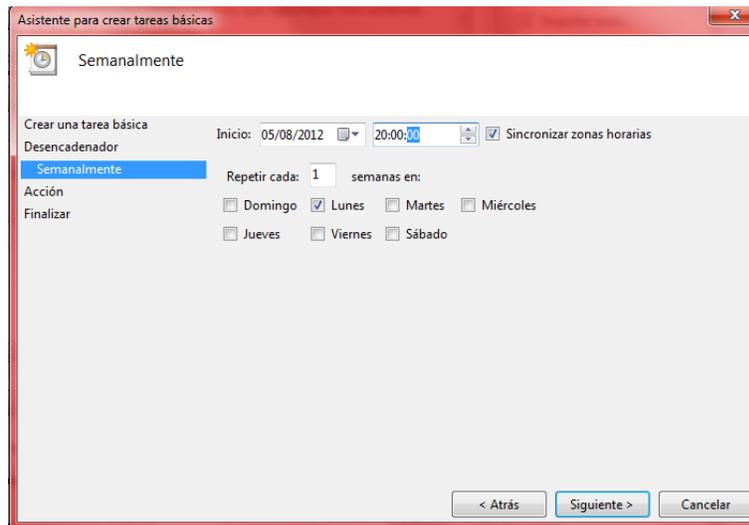


Figura 7.6.10: Cuarto paso para crear una nueva tarea programada en Windows 7
Fuente: El Autor

El siguiente paso es especificar qué tipo de acción se realizará, debe seleccionar *Iniciar un programa*.

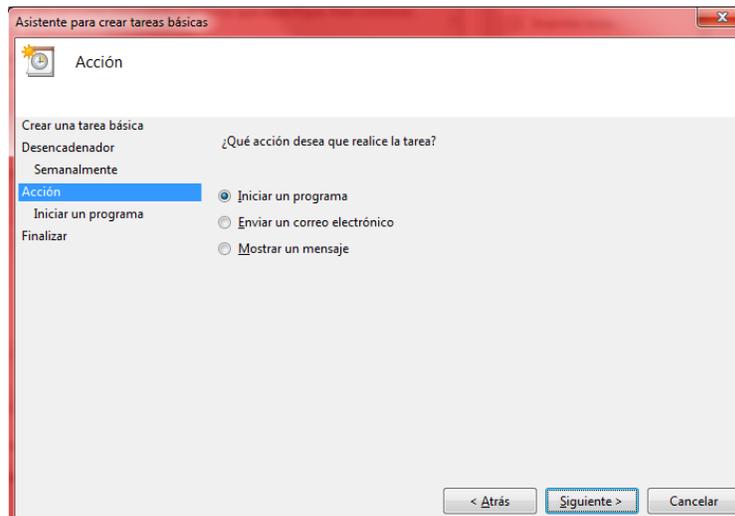


Figura 7.6.11: Quinto paso para crear una nueva tarea programada en Windows 7
Fuente: El Autor

Luego seleccionar el programa que se ejecutará, en este caso es el archivo *script_respaldoBDD.bat* luego de esto el asistente se cerrará y creará la tarea programada.

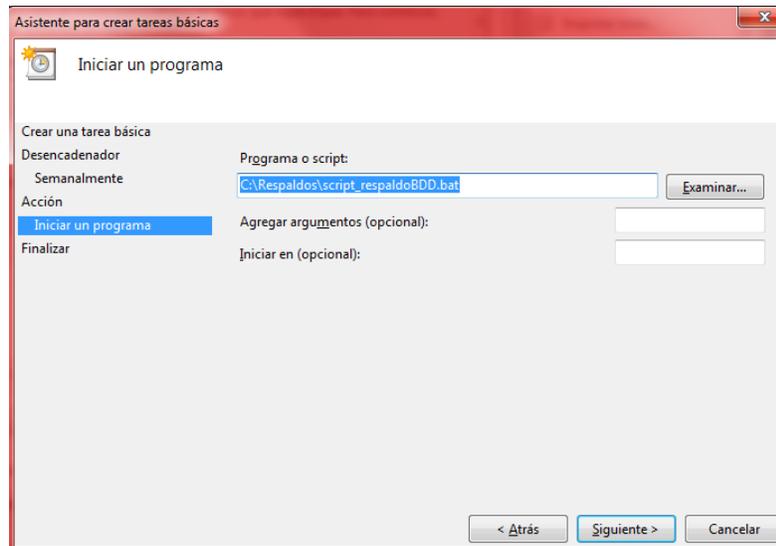


Figura 7.6.12: Sexto paso para crear una nueva tarea programada en Windows 7
Fuente: El Autor

- **Instalación de drivers del dispositivo biométrico**

Para instalar los drivers del dispositivo biométrico debe ingresar al directorio *Instaladores\DPOTWSDK161\SDK* del CD adjunto donde tiene que ejecutar el archivo *Setup.exe* y seguir las instrucciones del asistente de instalación.

- **Inicio del servidor de aplicaciones JBoss**

Abrir una nueva consola con las teclas *Windows + R* y luego escribir el comando *cmd* y hacer clic sobre el botón *Aceptar*.

Para iniciar el servidor debe ubicarse en el directorio *JBOSS_HOME/bin* desde la consola con el comando *cd* y luego ejecutar el siguiente comando *run.bat -b 0.0.0.0* luego presione *Enter*.

- **Ingreso al sistema**

Abra su navegador Web y en la barra de direcciones escribir <http://localhost:8080/ALPROSART> , debe ingresar el usuario *vmancheno* y la clave *1234* para iniciar sesión en el sistema.

BIBLIOGRAFÍA

LIBROS

- Bernd Bruegge y Allen H. Dutoit. Ingeniería de software orientada a objetos. 1ra edición. México. Pearson Educación S.A. 2002. 576 páginas.
- J. Rumbaugh, I. Jacobson, G. Booch. Lenguaje unificado de modelado UML 2da edición. España. Addison Wesley. 2006. 548 páginas.
- Martin, Rober c. UML para programadores JAVA. 1ra edición. España. Pearson Educación S.A. 2004. 250 páginas.
- M. Piattini, E. Marcos, C. Calero y B. Vela. Tecnología y diseño de bases de datos. 2da edición. México. Alfaomega. 2007. 980 páginas.
- Leobardo López Román. Metodología de la programación orientada a objetos. 1ra edición. México. Alfaomega. 2006. 456 páginas.
- Connolly, T. y Begg, C. Sistemas de Bases de Datos. Un enfoque práctico para diseño, implementación y gestión. 1ra edición. España. Ra-Ma. 2005. 1320 páginas.
- Steve McConnell. Desarrollo y Gestión de Proyectos. 3ra edición. España. McGraw-Hill. 1997. 691 páginas.
- Peter H. Gregory, CISA, CISSP y Michael A. Simon. Biometrics for dummies. 1ra edición. USA. Wiley. 2008. 292 páginas.
- Introducción a RichFaces, [en línea]. Jose Manuel Sánchez Suárez. Dirección URL:
<<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflntro>>. [Consulta: 7 enero 2012].
- Sistemas biométricos: matching de huellas dactilares mediante transformada de hough generalizada, [en línea]. Domingo Morales L. y Javier Ruiz del Solar. Dirección URL:
<http://www2.ing.puc.cl/~iing/ed429/sistemas_biometricos.htm>. [Consulta: 10 enero 2012].

- Automatic personal identification using fingerprints, [en línea]. Lin Hong. Dirección URL: http://biometrics.cse.msu.edu/Publications/Thesis/LinHong_FpIdentification_PhD98.pdf. [Consulta: 20 enero 2012].
- Metodología RUP – desarrollo de software de calidad, [en línea]. Fabián Bermeo. Dirección URL: <http://fabianbermeop.blogspot.com/2010/12/metodologia-rup-desarrollo-de-software.html>. [Consulta: 1 febrero 2012].
- Metodología de desarrollo RUP, [en línea]. Anónimo. Dirección URL: <http://www.conexionit.com/blog/metodologias/que-es-rup.html>. [Consulta: 2 febrero 2012].
- Diagrama de casos de uso, [en línea]. Comunidad. Dirección URL: http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso. [Consulta: 2 febrero 2012].
- Diagramas de casos de uso, [en línea]. Jesús Cáceres Tello. Dirección URL: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>. [Consulta: 8 febrero 2012].
- Diagramas de estado, [en línea]. Mari Carmen Otero Vidal. Dirección URL: <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r33019.PDF>. [Consulta: 13 febrero 2012].
- Introducción a UML, [en línea]. Marcos Fouces Lago. Dirección URL: <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>. [Consulta: 10 febrero 2012].
- Técnicas de recopilación de información, [en línea]. Ramón Cab chan. Dirección URL: <http://www.mitecnologico.com/Main/TecnicasDeRecopilacionDeInformacion>. [Consulta: 11 febrero 2012].
- Diagramas de estado, [en línea]. Marianella Martínez. Dirección URL: <http://www.clikear.com/manuales/uml/diagramasestado.aspx>. [Consulta: 13 febrero 2012].

- Conceptos básicos en un diagrama de estados, [en línea]. Pablo Figueroa. Dirección URL: <http://webdocs.cs.ualberta.ca/~pfiguero/soo/uml/estados01.html>. [Consulta: 12 febrero 2012].
- Diagrama de secuencia UML 2, [en línea]. Sparx Systems Pty Ltd. Dirección URL: http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html. [Consulta: 20 febrero 2012].
- Clasificación de los sistemas biométricos, [en línea]. Abril Hernández Briones. Dirección URL: <http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/recohuella.html>. [Consulta: 1 marzo 2012].
- Píxel, [en línea]. Comunidad. Dirección URL: <http://es.wikipedia.org/wiki/P%C3%ADxel>. [Consulta: 3 marzo 2012].
- Cómo funcionan los lectores de huella digital, [en línea]. TEC Electrónica, S.A. de C.V. Dirección URL: <http://www.tec-mex.com.mx/promos/bit/bit0903-bio.htm>. [Consulta: 10 marzo 2012].
- Kit de desarrollo de software, [en línea]. Comunidad. Dirección URL: http://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software. [Consulta: 16 marzo 2012].
- Java database connectivity, [en línea]. Comunidad. Dirección URL: http://es.wikipedia.org/wiki/Java_Database_Connectivity. [Consulta: 20 marzo 2012].
- Respaldo periódico en PostgreSQL, [en línea]. Comunidad. Dirección URL: http://www.lawebdelprogramador.com/foros/PostgreSQL/1071545Respaldo_periodico_en_postgres.html. [Consulta: 29 marzo 2012].