

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

TESIS DE GRADO

“DESARROLLAR E IMPLEMENTAR UN SISTEMA DE INFORMACIÓN WEB TRANSACCIONAL DE CONTROL DE TURNOS, ASISTENCIA Y SOLICITUDES DE NOVEDADES DE PERSONAL PARA LA EMPRESA HILANDERÍAS UNIDAS S.A.”

Previo a la obtención del Título de:

MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL

Presentado por:

José Antonio Caicedo Salazar
Julio Giovanni Litardo Unda

Guayaquil-Ecuador

2015



CIB - ESPOL

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

TESIS DE GRADO

“DESARROLLAR E IMPLEMENTAR UN SISTEMA DE INFORMACIÓN WEB TRANSACCIONAL DE CONTROL DE TURNOS, ASISTENCIA Y SOLICITUDES DE NOVEDADES DE PERSONAL PARA LA EMPRESA HILANDERÍAS UNIDAS S.A.”

Previo a la obtención del Título de:

MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL

Presentado por:

José Antonio Caicedo Salazar
Julio Giovanni Litardo Unda

Guayaquil-Ecuador

2015

espol

Biblioteca



D-100810

AGRADECIMIENTO

En especial a mi señora madre Luz Salazar Velasco, quien siempre me apoyó de forma incondicional en cada paso de mi vida, que gracias a su esfuerzo y sacrificio me he convertido en el profesional que soy, y ahora le dedico la culminación de mi maestría. A mi esposa quien me ha apoyado durante toda esta etapa, brindándome su tiempo, esfuerzo, comprensión y cariño.

José Antonio Caicedo Salazar

AGRADECIMIENTO

Al finalizar este arduo trabajo quiero agradecer en primer lugar a todas las personas que ayudaron a llevar a cabo el desarrollo de este trabajo, a nuestro director de tesis Msig Juan Carlos García, por su apoyo y orientación constante, a mi compañero y amigo de tesis, José Caicedo Salazar y un agradecimiento muy especial a mis padres, por haberme forjado en la persona que soy en la actualidad.

Julio Giovanni Litardo Unda

DEDICATORIA

Dedico este trabajo a la memoria de mi querida señora madre Luz Magdalena Salazar Velasco, quien me apoyó de forma incondicional en mi vida y desde el cielo sigue guiando mi camino. A mi amada esposa quien me ha apoyado en la culminación de esta etapa de mi carrera profesional, siempre ha estado presente cuando la necesite.

José Antonio Caicedo Salazar



DEDICATORIA

Dedico esta tesis sobre todo a Dios porque ha estado conmigo acompañándome en cada paso que doy, para aquella persona especial... que me falta... pero siempre está y también a mis padres, Alicia y Julio, quienes son pilares fundamentales en mi vida.

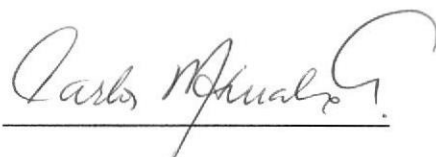
Julio Giovanni Litardo Unda

TRIBUNAL DE SUSTENTACIÓN



Msig. Juan Carlos García

DIRECTOR DE TESIS



PhD. Carlos Monsalve A.

MIEMBRO PRINCIPAL

Ing. Javier Tibau

MIEMBRO SUPLENTE

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**”.

(Reglamento de Graduación de la ESPOL).

José Antonio Caicedo Salazar

Julio Giovanni Litardo Unda



CIB - ESPOL

RESUMEN

El principal objetivo del presente trabajo de tesis es ofrecer un sistema en ambiente web de control de horarios, asistencias y solicitudes de novedades del personal para cualquier tipo de empresa, sea privada o pública, presentando cuadros resúmenes y estadísticas gráficas sobre el cumplimiento del personal.

El sistema facilita información real sobre el desempeño y cumplimiento del horario de trabajo fijado para el personal de una empresa, lo que mejora la eficiencia de las actividades del Departamento de Gestión de Talento Humano que mediante parámetros establecidos podrán tomar las decisiones adecuadas y oportunas.

El sistema se desarrolló aplicando los fundamentos de la metodología ICONIX, que propone el uso de ciertos diagramas de UML, esencialmente los casos de uso, para modelar el sistema tomando muy en cuenta los requerimientos del usuario. Para la implementación se empleó la arquitectura por capas, lo que permite que el sistema sea flexible a cualquier cambio de tecnología que se realice en un futuro posterior, por la integración de cada una de las capas a través de interfaces. Cabe especificar que se

utilizó software libre para el desarrollo, el servidor web y el servidor de base de datos, lo que implica cero costos por licenciamiento de software.

Finalmente se realizó un análisis económico del sistema, evaluando sus costos y beneficios, en términos de costos y reducción de tiempos.



CIB - ESPOL

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iv
TRIBUNAL DE SUSTENTACIÓN	vi
DECLARACIÓN EXPRESA	vii
RESUMEN	viii
ÍNDICE GENERAL	x
ABREVIATURAS	xv
ÍNDICE DE FIGURAS	xviii
ÍNDICE DE TABLAS	xx
ANEXOS	xxiii
INTRODUCCIÓN	xxiv
1 ANTECEDENTES	1
1.1 Descripción del Problema a resolver	2
1.2 Objetivo General	6
1.3 Objetivos Específicos	6
1.4 Resultados esperados	7
2 BASES TEÓRICAS Y METODOLÓGICAS	10
2.1 Arquitectura por capas	10
2.2 Patrón Modelo Vista Controlador (MVC)	15
2.3 Mapeo Objeto – Relacional (ORM)	19



2.3.1	Componentes de un ORM.....	20
2.3.2	Ventajas de un ORM.....	21
2.3.3	Desventajas de un ORM.....	23
2.4	Hibernate JPA.....	23
2.4.1	Introducción.....	23
2.4.2	Java Persistence API (JPA).....	26
2.4.3	Arquitectura.....	27
2.5	Framework Java para aplicaciones web JSF (Java Server Faces).....	32
2.5.1	Características de la Tecnología JSF.....	34
2.5.2	Implementaciones JSF.....	35
2.5.3	El ciclo de vida del proceso de solicitudes JSF.....	38
2.5.4	Java Beans, Beans Administrados (Managed Beans).....	44
2.5.5	Navegación.....	45
2.6	Framework Ajax: PrimeFaces.....	47
2.7	Servidor Web Apache.....	51
2.8	Framework Spring.....	53
2.9	Metodologías de Desarrollo Ágil.....	56
2.9.1	Metodología ICONIX.....	59
2.9.2	Características Principales.....	62
2.9.3	Tareas de ICONIX.....	63
3	ANÁLISIS DE REQUISITOS.....	68
3.1	Descripción de la situación actual.....	68

3.2	Problemas y especificación de requerimientos.....	73
3.2.1	Especificación de Requerimientos.....	73
3.2.2	Requisitos Funcionales	73
3.2.3	Requisitos No Funcionales.....	81
3.3	Sistema propuesto.....	83
3.3.1	Lista de Actores del Negocio.....	83
3.3.2	Modelo de Casos de Uso	86
3.3.3	Prototipo de interfaz de usuario.....	90
4	ANÁLISIS Y DISEÑO PRELIMINAR.....	95
4.1	Descripción de los Casos de Uso.....	95
4.1.1	Descripción de Caso de Uso Crear Turno Normal	96
4.1.2	Descripción de Caso de Uso Modificar Turno Normal.....	99
4.1.3	Descripción de Caso de Uso Crear Turno Frecuencia Mensual..	101
4.1.4	Descripción de Caso de Uso Crear Turno Rotativo Mensual	105
4.1.5	Descripción de Caso de Uso Asignar Turno.....	108
4.1.6	Descripción de Caso de Uso Cargar Marcaciones	110
4.1.7	Descripción de Caso de Uso Ingresar Solicitud.....	112
4.1.8	Descripción de Caso de Uso Aprobar Solicitud	115
4.2	Diagrama de Robustez	117
4.3	Diagramas de Clases (Modelo Conceptual)	121
5	DISEÑO DETALLADO.....	122

5.1	Entorno de software.....	122
5.1	Diagramas de secuencias	124
5.2	Diagramas de Clases (Diseño).....	128
5.3	Esquema de base de datos	131
5.4	Diccionario de datos	132
5.5	Seguridad	134
6	IMPLEMENTACIÓN.....	136
6.1	Estándares de codificación.....	136
6.2	Diagrama de paquetes	139
6.3	Diagrama de componentes.....	140
6.4	Diagrama de despliegue.....	143
6.5	Pruebas y Análisis de Resultados	145
6.5.1	Pruebas Unitarias	145
6.5.2	Pruebas basadas en casos de uso.....	148
7	INSTALACIÓN.....	155
7.1	Requerimientos de Instalación	155
7.1.1	Requerimiento de Hardware.....	156
7.1.2	Requerimiento de Software	157
7.2	Capacitación.....	158
7.3	Mantenimiento	160
8	ESTUDIO DE FACTIBILIDAD.....	162
8.1	Costos del Sistema.....	162

8.2	Beneficios tangibles e intangibles.....	165
8.2.1	Beneficios tangibles.....	166
8.2.2	Beneficios Intangibles.....	174
8.3	Análisis Económico.....	175
CONCLUSIONES		178
RECOMENDACIONES.....		180
GLOSARIO		181
BIBLIOGRAFÍA.....		189
ANEXOS.....		191



ABREVIATURAS

AJAX:	Asynchronous JavaScript And XML, Java Script Asíncrono y XML
AOP:	Aspect Oriented Programming, Programación Orientada a Aspectos
AWT:	Abstract Window Toolkit, Kit de herramientas de Ventana Abstracta
API:	Application Programming Interface, Interfaz de Programación de Aplicaciones
CM:	Costo Manual
CSI:	Costo Sistema Informático
CRUD:	Create Read Update Delete, Crear Leer Actualizar Eliminar
CSS:	Cascading Style Sheet, Hojas de Estilo en Cascada
DAO:	Data Access Object, Objeto de Acceso a Datos
DVD-ROM:	Digital Versatile Disc – ROM, Disco Versátil Digital de sólo lectura.
EJB:	Enterprise JavaBean
GB:	GigaByte
GTH:	Gestión de Talento Humano
HTML:	HyperText Markup Language, Lenguaje de Marcas de Hipertexto
HTTP:	Hypertext Transfer Protocol, Protocolo de Transferencia de

	Hipertexto
HTTPD:	HTTP Daemon
IE:	Internet Explorer
IT:	Information Technology, Tecnología de la Información.
JAR	Java Archive
JCP:	Java Community Process, Proceso de la Comunidad Java
JDBC:	Java DataBase Connectivity, Conector Java para Base de Datos
JEE:	Java Enterprise Edition, Java Edición Empresarial
JPA:	Java Persistence API, API de Persistencia de Java
JPQL:	Java Persistence Query Language, Lenguaje de Consulta de Persistencia Java
JSE:	Java Standard Edition, Java Edición Estándar
JSF:	Java Server Faces
JSP:	Java Server Pages
JSR:	Java Specification Request
MVC:	Model View Controller, Modelo Vista Controlador
ORM:	Object/Relational Mapping, Mapeo Objeto Relacional
PC:	Personal Computer, Computador Personal
POJO:	Plain Old Java Object, Antiguo Objeto Plano de Java
RAM:	Random Access Memory, Memoria de Acceso Aleatorio
RDBMS:	Relational Database Management System, Sistema de Gestión

	de Base de Datos Relacional.
RF:	Requisito Funcional
RNF:	Requisito No Funcional
ROM:	Read Only Memory, Lectura Sólo en Memoria
RUP:	Rational Unified Proccess, Proceso Unificado de Rational
SQL:	Structured Query Language, Lenguaje de Consulta Estructurado
SWT:	Standard Widget Toolkit,
TB:	TeraByte
UI:	User Interface, Interface de Usuario
UML:	Unified Modeling Language, Lenguaje Unificado de Modelado
VA:	Valor del Ahorro
VGA:	Video Graphics Array, Adaptador Gráficos de Videos
XHTML:	eXtensible HTML, HTML Extensible
XP:	eXtreme Programming, Programación Extrema
XML:	eXtensible Markup Language, Lenguaje de Marcas Extensible
XUL:	XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario

ÍNDICE DE FIGURAS

Figura 2.1: Arquitectura de tres capas	13
Figura 2.2: Patrón Modelo Vista Controlador	17
Figura 2.3: ORM Mapeo Objeto – Relacional	20
Figura 2.4: Arquitectura Externa JPA.....	28
Figura 2.5: Elementos de la Arquitectura JPA	29
Figura 2.6: Modelo de Componentes del Servidor	38
Figura 2.7: Ciclo de Vida de una página JavaServer Faces	40
Figura 2.8: Archivo de configuración face-config.xml.....	46
Figura 2.9: Principales Módulos de Spring.....	55
Figura 2.10: Metodología Tradicional vs. Metodología Ágil.....	58
Figura 2.11: Metodología ICONIX.....	61
Figura 4.1: Diagrama de Robustez Crear Turno Normal	117
Figura 4.2: Diagrama de Robustez Asignar Turno	118
Figura 4.3: Diagrama de Robustez Crear Turno Frecuencia Mensual	118
Figura 4.4: Diagrama de Robustez Crear Turno Rotativo Mensual	119
Figura 4.5: Diagrama de Robustez Cargar Marcaciones	119
Figura 4.6: Diagrama de Robustez Ingresar Solicitud.....	120
Figura 4.7: Diagrama de Robustez Aprobar Solicitud	120
Figura 4.8: Modelo de Dominio Gestión de Marcaciones	121
Figura 4.9: Modelo de Dominio Gestión de Seguridades.....	121

Figura 5.1: Diagrama Entidad – Relación 131

Figura 8.1: Resumen de Costos del Desarrollo del Sistema 165

Figura 8.2: Costos Manual vs. Sistema Información 177



CIB - ESPOL

ÍNDICE DE TABLAS

Tabla 1: Arquitectura JPA interfaces del paquete javax.persistence	30
Tabla 2: Implementaciones JavaServer Faces	35
Tabla 3: Comparativa de implementaciones JSF	49
Tabla 4: Requisitos Funcionales del Sistema	74
Tabla 5: Requisitos No Funcionales del Sistema	81
Tabla 6: Actores del Negocio	83
Tabla 7: Diagrama de Casos de Uso del Sistema de Control de Turnos y Solicitudes.....	86
Tabla 8: Diagrama de Casos de Uso Gestión de Marcaciones.....	87
Tabla 9: Diagrama de Casos de Uso Gestión de Solicitudes.....	88
Tabla 10: Diagrama de Casos de Uso Gestión de Seguridades.....	89
Tabla 11: Interfaz de usuario crear turno normal	90
Tabla 12: Interfaz de usuario crear turno frecuencia mensual	91
Tabla 13: Interfaz de usuario crear turno variable mensual	92
Tabla 14: Interfaz de usuario asignar turno.....	93
Tabla 15: Interfaz de usuario carga de marcaciones	94
Tabla 16: Descripción Caso de Uso Crear Turno Normal	96
Tabla 17: Descripción Caso de Uso Modificar Turno Normal	99
Tabla 18: Descripción Caso de Uso Crear Turno Frecuencia Mensual	101
Tabla 19: Descripción Caso de Uso Crear Turno Rotativo Mensual.....	105

Tabla 20: Descripción Caso de Uso Asignar Turno	108
Tabla 21: Descripción Caso de Uso Cargar Marcaciones	110
Tabla 22: Descripción Caso de Uso Ingresar Solicitud	112
Tabla 23: Descripción Caso de Uso Aprobar Solicitud	115
Tabla 24: Tecnología de Implementación	123
Tabla 25: Diagrama de Secuencias Crear Turnos	124
Tabla 26: Diagrama de Secuencias Carga de Marcaciones	125
Tabla 27: Diagrama de Secuencias Asignar Turno.....	126
Tabla 28: Diagrama de Secuencias Aprobar Solicitud.....	127
Tabla 29: Diagrama de Clases Gestión de Marcaciones	128
Tabla 30: Diagrama de Clases Gestión de Solicitudes	129
Tabla 31: Diagrama de Clases Gestión de Seguridades	130
Tabla 32: Lista de Tablas.....	132
Tabla 33: Estándares de Codificación	138
Tabla 34: Diagrama de Paquetes	140
Tabla 35: Diagrama de Componentes	142
Tabla 36: Diagrama de Despliegue.....	144
Tabla 37: Prueba unitaria clase entidad.....	146
Tabla 38: Prueba unitaria clase capa de datos	147
Tabla 39: Prueba unitaria clase capa de negocio	148
Tabla 40: Prueba PR_RF-001.....	149
Tabla 41: Prueba PR_RF-004.....	150

Tabla 42: Prueba PR_RF-005.....	151
Tabla 43: Prueba PR_RF-006.....	152
Tabla 44: Prueba PR_RF-007.....	153
Tabla 45: Requerimiento de Hardware Servidor Web y Servidor Base de Datos	156
Tabla 46: Requerimiento de Hardware Computadoras Clientes	157
Tabla 47: Requerimiento de Software Servidor Web y Servidor Base de Datos	158
Tabla 48: Requerimiento de Software Computadoras Clientes	158
Tabla 49: Costos del personal de desarrollo.....	163
Tabla 50: Costos del equipo para desarrollar	164
Tabla 51: Costos Varios.....	164
Tabla 52: Resumen de Costos del Desarrollo del Sistema	165
Tabla 53: Beneficios de tiempo	170
Tabla 54: Beneficios Económicos por Actividad	173
Tabla 55: Costos Manual vs. Sistema Información en proyección de 5 años	176

ANEXOS

Anexo 1: Descripción de campos de las tablas.....	191
Anexo 2: Entrevistas realizadas a los actores del sistema.....	213

INTRODUCCIÓN

En la actualidad está teniendo mucha demanda y aceptación las metodologías ágiles para el proceso de desarrollo de software. Las metodologías ágiles; que a diferencia de las metodologías tradicionales o pesadas que establecen de manera rigurosa las tareas, herramientas y notaciones en el desarrollo; se enfocan en una mayor interacción del usuario con el desarrollador donde se le va mostrando versiones funcionales del software en períodos cortos de tiempo, para que pueda evaluar y sugerir cambios al software según se va desarrollando.

Un proceso es ágil cuando el desarrollo de software es incremental (se realizan entregas pequeñas de software, con ciclos rápidos), es cooperativo (usuario y desarrolladores trabajan juntos), es sencillo (es fácil de aprender y modificar, bien documentado), y es adaptable (permite realizar cambios de último momento sin afectar mayormente a todo el proceso).

Por ello se desarrolló el Sistema de Información Web Transaccional de Control de Turnos, Asistencia y Solicitudes de Novedades del Personal, aplicando los fundamentos de la metodología ICONIX para el análisis y diseño del sistema; guiada a través de casos de uso y siguiendo un ciclo de

vida iterativo e incremental. Para la codificación se empleó Java, por ser una plataforma muy amplia y estable que ofrece una variedad de opciones para el desarrollo de aplicaciones web entre ellas Java Server Faces por ser un estándar, tiene componentes configurables, permite la creación de interfaces de usuario avanzadas, entre las principales. Java Server Faces cuenta con una variedad de implementaciones que facilitan el desarrollo de aplicaciones utilizando esta tecnología, entre ellas se encuentra PrimeFaces.

Con el presente trabajo se alcanzó a modelar y desarrollar un sistema web para el control de turnos, asistencias y solicitudes de personal para la empresa Hilanderías Unidas. El sistema web ofrecerá una serie de opciones que facilitará el registro del personal, los horarios, las marcaciones que el personal genera diariamente a través del reloj biométrico, las solicitudes de novedades del personal (permisos, vacaciones, licencias, etc.) todo esto con el objetivo de facilitar las actividades del departamento de Gestión del Talento Humano, en relación al control y seguimiento de los empleados ya que podrá utilizar gráficas estadísticas y aplicar así procedimientos preventivos y correctivos hacia el personal.

CAPÍTULO 1

1 ANTECEDENTES

La empresa Hilanderías Unidas Hiunsa S.A. fue constituida el 9 de julio de 1.991 y está ubicada en el kilómetro 7,5 vía a Daule de la ciudad de Guayaquil-Ecuador. La etapa pre operacional tuvo una duración de 2 años y a partir del año 1.994 se inicia la producción de hilos para la industria Textil y de la confección y así se empieza a comercializar el hilo producido con los más altos estándares de calidad.

Es una empresa dedicada a la hilatura y fabricación de hilados e hilos para tejedura o costura, para el comercio o para procesamiento posterior, texturización, retorcido, plegado, cableado y remojo de hilaturas filamentosas de toda clase de fibras animales, vegetales, sintéticas.

Actualmente se ha incrementado el número de empleados en la empresa, tanto en el área Administrativa y mucho más en el área de Producción, y es en esta última área donde los empleados tienen diversidad de turnos. La empresa cuenta con relojes biométricos pero no dispone de un software que le permita obtener de una manera más rápida y eficaz de consultas de las marcaciones realizadas por los empleados.

De igual manera, el jefe del área de Producción, debido a los repentinos cambios de turnos que se producen para los operarios, estos no pueden ser notificados a tiempo al Departamento de Gestión del Talento Humano debido a que el trámite es realizado por notificación escrita, esto genera que al realizar el cierre para el rol de pagos se reflejen descuentos a los empleados por supuestas faltas.

1.1 Descripción del Problema a resolver

En la actualidad, un gran porcentaje de las empresas tienen la necesidad de llevar un control más detallado de las actividades de sus empleados. Es por este motivo, que en la mayoría de estas han incorporado en sus instalaciones el uso de relojes biométricos



o de tarjeta magnética, para llevar un control de las entradas y salidas de los empleados.

Este tipo de controles, genera un archivo de texto, los cuales son descargados del dispositivo diariamente. Estos archivos son tarjados con respecto al horario de trabajo del empleado, para de esta forma establecer si ha tenido alguna novedad, y consolidar sus atrasos o sus horas extras, y de esta manera, entregar un informe al departamento financiero al cierre del mes, y efectuar su correspondiente acreditación o descuento del sueldo.

Este proceso ha generado un gran malestar entre los empleados, debido a que desconocen de los descuentos al final del mes, es decir cuando se les entrega el rol correspondiente al mes devengado. Por este motivo es muy común que los últimos días para el cierre correspondiente, los empleados realicen masivas consultas vía telefónica o personalmente al departamento de Gestión del Talento Humano, para constatar con anterioridad, cuanto será descontado del rol.

El departamento de Gestión de Talento Humano, realiza mensualmente el pago de los sueldos de los empleados, este proceso se realiza los 24 de cada mes. Para realizar el rol de pago, se dedica durante dos días anteriores el asistente de Gestión del Talento Humano, de tarjar las marcaciones realizadas por los empleados. Para de esta forma definir las horas extras, atrasos, faltas y poderlas incluir en el rol del empleado.

Uno de los principales problemas han sido las quejas reiteradas por parte del personal, por descuentos indebidos y reiterados errores en la correcta acreditación de horas extras realizadas. Básicamente esto se debe a la ausencia de informes tanto para el personal de administración como para los empleados quienes a su vez en los días de pagos saturan tanto el correo como la línea telefónica, para realizar consultas sobre sus acreditaciones o descuentos.

Además, la gerencia está solicitando, que se debe entregar memorándums a las personas, que incurren en faltas injustificadas, atrasos de manera reiterativa, y que se incluyan los artículos del código de trabajo.

Existen descontentos por parte de diversas áreas de la organización, ya que hay empleados que trabajan en horarios rotativos, y estos a veces son desconocidos por el asistente de Gestión del Talento Humano, motivo por el cual algunos empleados reportan ausencia, cuando si han cubierto el horario respectivo.

La falta de reportes estadísticos por área, departamento y un análisis minucioso del empleado, en cuanto a su asistencia y cumplimiento de horas en la institución.

El Sistema de Información Web Transaccional de Control de Turnos, Asistencia y Solicitudes de Novedades del Personal, permitirá descongestionar el área de Gestión del Talento Humano, cuando se realice el cierre de mes consolidando las asistencias de los empleados automáticamente. Además permitirá que las solicitudes por novedades, sean manejadas de manera digital, reduciendo el consumo de papel, almacenamiento de estos, pérdidas y deterioros que se dan con el transcurrir del tiempo.

La aplicación del Sistema de Información Web Transaccional de Control de Turnos, Asistencia y Solicitudes de Novedades del Personal es para cualquier tipo de empresa que disponga de relojes biométricos o tarjetas, para realizar el control de asistencia.

1.2 Objetivo General

Desarrollar e Implementar un Sistema de Información Web Transaccional de control de turnos, asistencia y solicitudes de novedades de personal para la empresa Hilanderías Unidas S.A.

1.3 Objetivos Específicos

- Identificar los diferentes problemas referentes al control de asistencia de los empleados y proveer de un sistema en línea que le permita al departamento de Gestión de Talento Humano realizar un seguimiento de los turnos, asistencia y novedades (faltas injustificadas, atrasos, etc.) de los empleados.
- Desarrollar el sistema utilizando la programación por capas, separando la capa de datos de la capa de lógica de negocios y de la capa de presentación al usuario, lo que permite que el desarrollo se puede llevar a cabo en varios niveles y, en caso de

que sobrevenga algún cambio, sólo se realiza en la capa requerida sin tener que realizar cambios a las demás capas.

- Definir los escenarios de uso del sistema por parte de los usuarios, para facilitar la comprensión del diseño del sistema mediante el empleo de diagramas UML (Diagramas de Casos de Uso, Clases, secuencias, actividades y estados).
- Diseñar el modelo de negocios y el esquema de base de datos del sistema, así como las interfaces gráficas de usuario las que serán intuitivas y agradables al usuario final, mediante el uso de herramientas libres aplicables a la Web.
- Realizar la implantación general, la instalación del servidor de aplicación, en el cual residirá los componentes del sistema y capacitar oportuna y adecuadamente a los usuarios, para su óptima utilización.

1.4 Resultados esperados

La automatización de los procesos de asistencia y novedades, que se reportan diariamente en el departamento de Gestión del Talento Humano (GTH) de una empresa, dentro de este campo se enmarcará el seguimiento y la temporización y responsabilidades de las diferentes tipos de solicitudes que se manejan en el área de GTH.



El uso de diferentes herramientas libres tanto a nivel de programación, como es el uso de Java, almacenamiento a través de MySQL y un sistema operativo ampliamente aceptado como es Linux.

El sistema a realizar permitirá introducirnos en diferentes metodologías ampliamente utilizadas en la actualidad, como son las metodologías ágiles, las cuales permitirán una mejor interacción tanto con el equipo interno de desarrollo, así como con el cliente, programando entregas de prototipos a corto plazo, lo cual nos permitirá conocer adecuadamente los requerimientos de los clientes.

Con la implementación del Sistema de Información Web Transaccional de Control de Turnos, Asistencia y Solicitudes de Novedades del Personal se espera lograr lo siguiente:

- Disponer de un control de asistencia, para realizar un seguimiento de las actividades del empleado.
- Consultas de sus marcaciones en línea por parte de los empleados.

- Generación automática del informe final entregado cada mes a la gerencia de recursos humanos.
- Generación de informes estadísticos por empleado, área y/o departamento.
- Definir los diferentes tipos de solicitudes.
- Realizar de manera digital, los diferentes tipos de solicitudes de novedad.
- Realizar el seguimiento respectivo, y envío de alertas cuando una solicitud se ha demorado más del tiempo debido.
- Realizar la aprobación de las solicitudes para que afecten a las marcaciones.
- Generación de horarios y turnos, para el personal

CAPÍTULO 2

2 BASES TEÓRICAS Y METODOLÓGICAS

Aquí se argumenta desde el punto de vista tecnológico, la arquitectura a utilizar para el desarrollo del sistema por capas: capa de presentación, de negocios y de datos; ¿por qué desarrollar por capas? El uso del patrón Modelo – Vista – Controlador en el desarrollo y comprender la especificación Java Server Faces (JSF) 2.0 y la implementación de la misma a través de Prime Faces en el desarrollo Java Web.

2.1 Arquitectura por capas

El estilo de arquitectura por capas, se enfoca en cómo organizar el modelo de diseño de la aplicación en capas. Se basa en una distribución jerárquica de los roles y responsabilidades bien definidas de cada componente, lo que significa que los

componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada. [1]

La idea de hacer una separación en capas es que cada una de las mismas cumpla con un rol y tenga responsabilidades bien definidas. [2]

Algunos autores definen el estilo por capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior.

La Arquitectura por capas tiene las siguientes características:

- Las capas de una aplicación pueden estar en la misma máquina o pueden estar distribuidas entre varios equipos.
- La mayoría de interacciones entre los servicios ocurre solamente entre capas vecinas.

- Los componentes de cada capa se comunican con otros componentes en otras capas mediante interfaces muy bien definidas.

En las aplicaciones distribuidas modernas se pueden encontrar diversas variantes del estilo por capas: arquitecturas de dos capas, arquitecturas de tres capas y arquitecturas de n capas. La arquitectura de tres capas es la más utilizada en las aplicaciones web, es la arquitectura común de la mayoría de los sistemas de información, abarcando una interfaz para el usuario y el almacenamiento de datos persistentes. Una vista clásica de esta arquitectura plantea tres capas verticales: Presentación (Interfaz de usuario), Lógica de Aplicaciones (tareas y reglas que rigen el proceso de negocio) y Almacenamiento (medio de almacenamiento persistente).

Lo esencial de esta arquitectura consiste en aislar la lógica de la aplicación y convertirla en una capa intermedia bien definida y lógica del software. Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la

aplicación. Se pueden construir y desplegar varias interfaces de usuario sin cambiar para nada la lógica.

Arquitectura de Tres Capas

En los últimos años se ha dinamizado mucho la construcción de nuevas aplicaciones en el ámbito empresarial (sobre todo), la arquitectura más usada por las grandes empresas desarrolladoras son las fundamentadas en niveles (3 capas), que son: la capa de presentación, la capa de negocio y la capa de acceso a datos (persistencia), como se muestra en la siguiente figura.

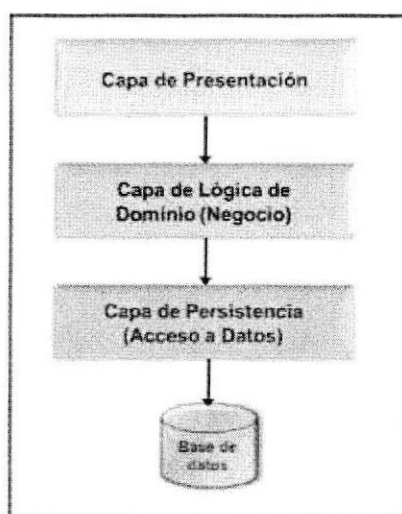


Figura 2.1: Arquitectura de tres capas

Fuente: Freire, T. (2008). Estudio de la Técnica ORM (Mapeo Objeto - Relacional). [Figura].

Recuperado de <http://repositorio.utn.edu.ec/bitstream/123456789/571/1/Tesis.pdf>.

Debido a que esta arquitectura permite tener total independencia entre cada capa permite también depender de los elementos de las capas inferiores pero no con las capas superiores.

Capa de Presentación

El principal objetivo de esta capa es la de permitir al desarrollador diseñar la interfaz que visualizará el usuario y de esa manera pueda (el usuario) interactuar con el sistema, la tarea fundamental de esta capa es la de comunicar y receptor información del usuario, sin embargo, otra tarea es la de poder interpretar cualquier mensaje que llegue desde el browser.

Capa de Lógica de Dominio o de Negocio

Esta capa representa los conceptos de negocio como reglas o estados, es donde se establecen y aplican todas estas reglas del negocio que deben cumplirse y sobre todo mantiene sus conceptos principales del proceso del negocio.

Capa de Persistencia o de Acceso a Datos

Esta capa se encarga de obtener y mantener la persistencia de los datos encapsulándolos. Aunque es independiente de la capa de lógica de dominio ofrece servicios para sincronizarse con ella. Ya que es necesario reconocer de antemano todos los objetos que deben sobrevivir a la ejecución de la aplicación hace que esta capa se convierta en la parte más delicada en el desarrollo del sistema.

2.2 Patrón Modelo Vista Controlador (MVC)

La idea del patrón MVC es separar los datos (modelo) y la lógica de negocio (controlador) de una aplicación de la interfaz de usuario (vista) en tres componentes separados e independientes y que cualquier cambio que se dé en el Modelo se reflejen instantemente en las Vistas, este patrón se basa en la idea de reutilización de código.

El patrón MVC se lo puede observar habitualmente en aplicaciones web en la que:

- La vista es la página web;
- El modelo es el RDBMS y la lógica de negocio.

- El controlador es el encargado de admitir los eventos de entrada desde la vista.

Las aplicaciones web ha sido sin lugar a dudas el tipo donde más ha sido aplicado este patrón de diseño, esto debido a que se necesita separar la lógica de control de la interface de usuario y las acciones de presentación y validación de los datos presentados en la interface.

Este tipo de patrón permite desacoplar ciertos aspectos del formato de la presentación y poder hacer uso máximo de la reutilización de componentes. Esta separación físicamente se establece con dos archivos generalmente uno para la renderización de los componentes gráficos y otros para las acciones del usuario con los componentes.

Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio

producido en el Modelo se refleje automáticamente en cada una de las Vistas. [3]

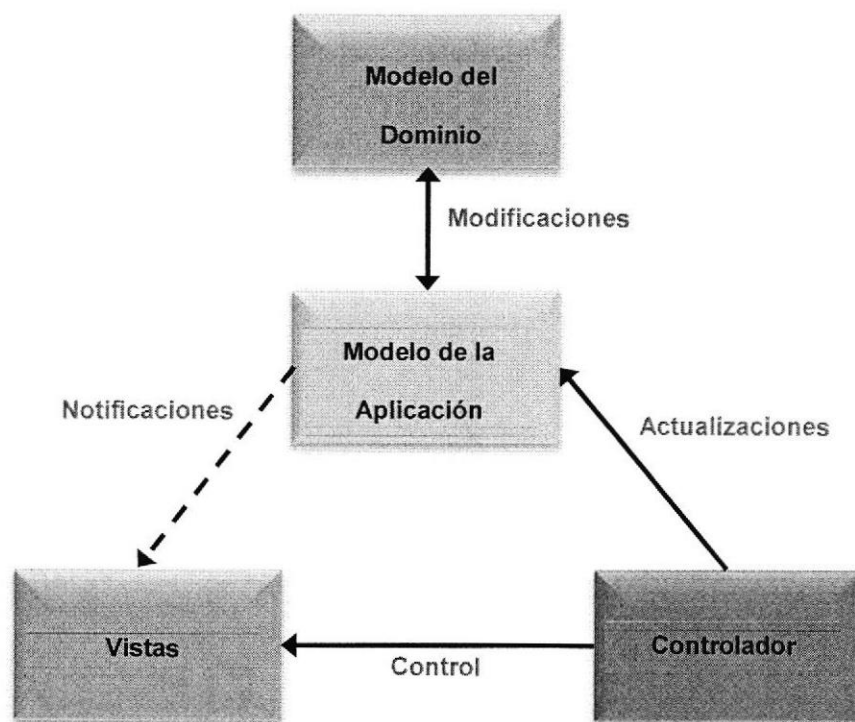


Figura 2.2: Patrón Modelo Vista Controlador

Fuente: Bascón, T. (2004). El patrón de diseño Modelo-Vista-Controlador (MVC)

y su implementación en Java Swing. [Figura]. Recuperado de

[http://www.ucbca.edu.bo/Publicaciones/revistas/actanova/documentos/v2n4/v2.](http://www.ucbca.edu.bo/Publicaciones/revistas/actanova/documentos/v2n4/v2.n4.bascon.pdf)

n4.bascon.pdf.

- **El Modelo**

El Modelo es el conjunto de clases que se encarga del acceso a la data (base de datos) representando la información del mundo real que el sistema debe operar, a parte gestiona los accesos concediendo sólo los privilegios necesario a cada usuario, y a la información que éste va a manipular, pero teniendo en cuenta que siempre estará regido por las especificaciones dictada desde la lógica de negocio. En otras palabras el Modelo, opera el comportamiento y los datos del dominio de la aplicación, responde a los requerimientos de información acerca de su estado (usualmente desde la vista) y a las instrucciones para cambiar de estado (usualmente desde el controlador).

- **La Vista**

La Vista es el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Muestra los datos de forma apropiada y representará lo que el usuario verá en la pantalla (usualmente es conocida como interface de usuario) para que él pueda interactuar con el sistema.

- **El Controlador**

El Controlador representa el mediador entre la vista (respondiendo a eventos o acciones frecuentemente solicitadas por el cliente) e invocando peticiones al modelo (solicitando información).

2.3 Mapeo Objeto – Relacional (ORM)

El ORM es una poderosa técnica de programación que nos ayuda a enlazar las entidades de un Sistema de Gestión de Base de Datos Relacional (RDBMS) y los conceptos de orientación a objetos, creando una base de datos orientada a objetos virtual, simplificando las tareas de acceso y manipulación de datos en tiempo de programación, a esto se lo conoce como una persistencia de objetos.

Como es conocido, la RDBMS sólo puede almacenar tipos de datos primitivos, por lo que el ORM ayuda precisamente a olvidar completamente de cómo transformar los objetos en registros almacenados y viceversa, abstrayendo al programador de la base de datos y lo centra en el desarrollo de la aplicación.

El mapeo Objeto/Relacional es “la persistencia automatizada y transparente de las tablas en una Base de Datos relacional, usando metadatos que definen el mapeo entre los objetos y la Base de Datos”. [4]

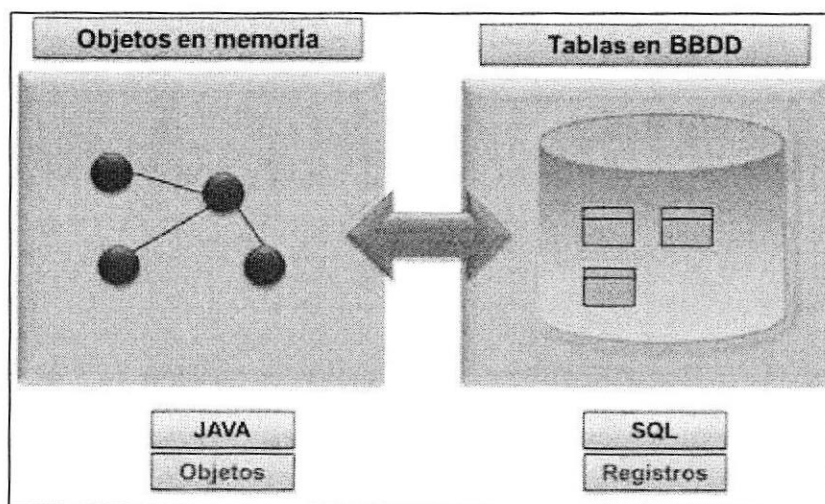


Figura 2.3: ORM Mapeo Objeto – Relacional

2.3.1 Componentes de un ORM

Los componentes de una solución ORM son los siguientes:

- Una API que permite realizar las cuatro operaciones CRUD sobre objetos de clases persistentes que son: creación, recuperación/lectura, actualización y eliminación.

- Provee una API o un lenguaje que ayuda a definir consultas sobre esas clases persistentes.
- Provee una opción que ayuda a definir el mapeo de los metadatos.
- Provee una técnica para que la implementación del ORM interactúe con objetos transaccionales para que pueda llevar a cabo consultas, agrupaciones u otras funciones de optimización.

2.3.2 Ventajas de un ORM

1 Facilidad y rapidez en el desarrollo.

La mayoría de las herramientas ORM actuales permiten la creación del modelo directamente desde el esquema de la base de datos, obteniendo el modelo adecuado.

2 Abstracción de la base de datos utilizada.

Al usar un ORM, se logra una total independencia del repositorio de datos, lo que ayudaría a minimizar el impacto si se diera el caso de cambiar el motor de base de datos, ya que sería más fácil.

3 Reutilización.

Permite acceder a todos los métodos de un objeto de datos desde cualquier parte de la aplicación, incluso desde sistemas diferentes.

4 Seguridad.

Algunos ORM suelen implementar sistemas de seguridad evitando ataques contra la capa de acceso a datos.

5 Mantenimiento del código.

Facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos, haciendo que el código sea mucho más sencillo y limpio.

6 Lenguaje propio para realizar las consultas.

Todos los ORM usan su propio lenguaje para realizar consultas.



2.3.3 Desventajas de un ORM

1 Tiempo empleado en aprender el lenguaje del ORM.

Debido a la complejidad de la herramienta la curva de aprendizaje sobre la misma suele ser un tanto dilatada porque hay que emplear tiempo en aprender el correcto funcionamiento y poder explotarla de manera óptima.

2 Aplicaciones pueden afectar su rendimiento.

Esto sucede porque cualquier consulta que se haga hacia la base de datos, la aplicación necesariamente debe primero transformar al lenguaje propio de la herramienta, posteriormente leer cada uno de los registros y finalmente crear las entidades (objetos).

2.4 Hibernate JPA

2.4.1 Introducción

En la arquitectura de una aplicación el framework de persistencia u ORM se ha convertido en un elemento crucial en el desarrollo de una aplicación, ya que, nos ayuda a equilibrar los paradigmas de la programación orientada a objetos con las de las base de datos relacionales.

Como se conoce la diferencia entre los lenguajes de programación orientados a objetos (como por ejemplo Java) y las bases de datos relacionales es que el primero soluciona problemas del negocio utilizando objetos (éstos tienen estado y comportamiento), mientras que los segundos tienen tablas (compuestas por filas y columnas) para resolver problemas de negocio, lo que necesariamente obliga a realizar una correlación entre el sistema de base de datos relacionales y el lenguaje orientado a objetos para guardar un objeto dentro de una tabla de base de datos y viceversa.

La herramienta Java que nos ayuda a realizar dicha correlación (convertir un objeto a tabla de base de datos y viceversa) de manera simple y transparente es el API de Persistencia (JPA), que no es otra cosa que una abstracción del Conector para Base de Datos Java (JDBC), esta transformación es lo que se conoce como ORM y es configurado usando metadatos (empleando XML o anotaciones).

JPA ofrece una interface común que es implementada por un proveedor de persistencia ORM (como Hibernate, Eclipse Link, etc.), actualmente existen numerosos proveedores de persistencia que han evolucionado su framework, para facilitar la transformación del modelo de objetos al relacional y viceversa, con lo que se puede elegir en cualquier momento el proveedor que más se adecue a los requerimientos y necesidades del desarrollo de la aplicación. Es el proveedor quién realiza el trabajo, pero siempre funcionando bajo la API de JPA.

Hibernate ORM permite a los desarrolladores escribir aplicaciones más fácilmente cuyos datos sobreviven al proceso de solicitud. Como framework ORM, Hibernate se ocupa de la persistencia de datos que se aplica a las bases de datos relacionales (vía JDBC). Hibernate es también una implementación de la especificación JPA. Como tal, se puede utilizar fácilmente en cualquier entorno que soporte JPA incluyendo aplicaciones Java SE, servidores de aplicaciones Java EE, etc. [5]

2.4.2 Java Persistence API (JPA)

La API de persistencia Java es el estándar de transformación objeto/relacional que proporciona a los programadores Java administrar datos relacionales en las aplicaciones Java mediante anotaciones o con descriptores XML. [6]

La API de persistencia (JPA) provee un modelo de persistencia basado en Plain Old Java Object: clases Java simples (POJO's) para mapear bases de datos relacionales en Java. El objetivo principal del JPA es no desaprovechar las ventajas del uso de un lenguaje de orientación a objetos al interactuar con una base de datos relacionales.

JPA es una especificación de Sun Microsystem, que fue desarrollada para la plataforma Java EE por expertos de EJB 3.0, e incluida en el estándar de EJB 3.0 como parte de la Java Specification Request JSR 220. Pero su uso no se limita a los componentes software EJB. Se puede utilizar en

aplicaciones web (Java EE) y aplicaciones clientes (Java SE).

Está construida alrededor de tres áreas principales:

1. API de persistencia Java, definida en el paquete `javax.persistence`.
2. Metadatos de transformación objeto/relacional (Anotaciones).
3. Lenguaje de consultas de Persistencia de Java (JPQL), permite especificar consultas a bases de datos relacionales en forma parecida a SQL desde un programa Java.

2.4.3 Arquitectura

La arquitectura de JPA, vista externa, se muestra en la siguiente figura.

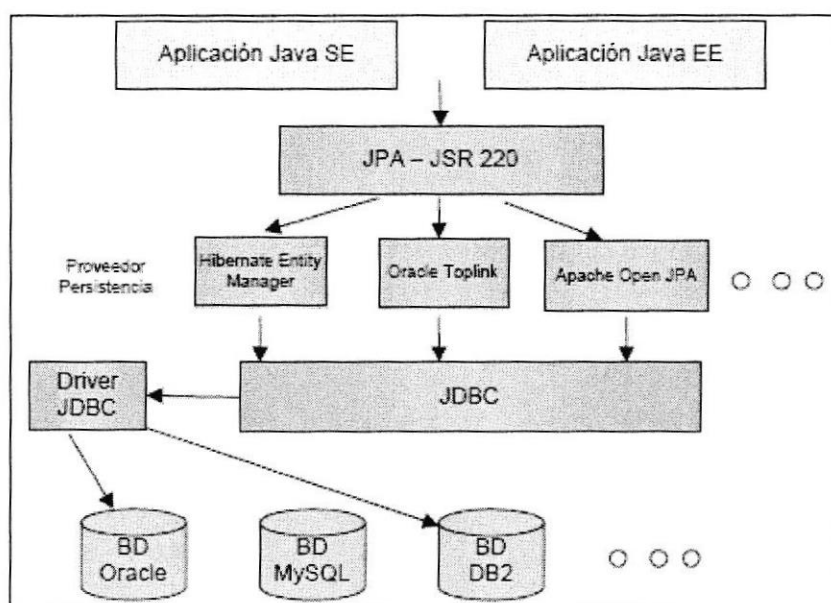


Figura 2.4: Arquitectura Externa JPA

Fuente: Freire, T. (2008). Estudio de la Técnica ORM (Mapeo Objeto - Relacional). [Figura].

Recuperado de <http://repositorio.utn.edu.ec/bitstream/123456789/571/1/Tesis.pdf>.

JPA se basa en los siguientes elementos para facilitar la persistencia:

- ✓ Entidades
- ✓ Contexto de Persistencia
- ✓ Unidad de Persistencia
- ✓ Administrador de Entidades

La siguiente figura muestra la relación entre los principales elementos de la arquitectura JPA dentro del paquete *javax.persistence*.

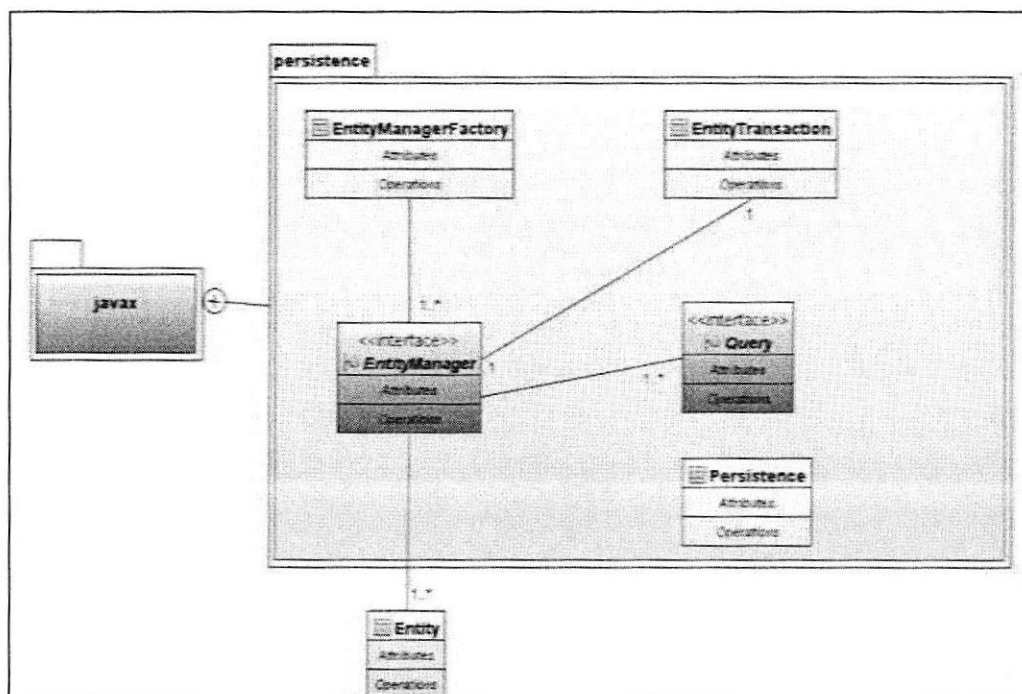


Figura 2.5: Elementos de la Arquitectura JPA

Fuente: Apache Open JPA. (2014). Java Persistence API Architecture. [Figura]. Recuperado de http://openjpa.apache.org/builds/1.2.3/apache-openjpa/docs/jpa_overview_arch.html.

La figura muestra que para cada unidad de persistencia hay un `EntityManagerFactory` y que muchos `Entity Managers` pueden ser creados para un solo `EntityManagerFactory`.

En tanto que muchos Entity Managers pueden apuntar al mismo contexto de persistencia. [7].

Algunas de las interfaces son sólo necesarias para su utilización fuera de un servidor de aplicaciones que soporte EJB 3, como es el caso del *EntityManagerFactory* que es ampliamente usado en desarrollo de aplicaciones de escritorio. En un servidor de aplicaciones, una instancia de *EntityManager* típicamente suele ser inyectada, haciendo así innecesario el uso de un *EntityManagerFactory*.

Por otra parte, las transacciones dentro de un servidor de aplicaciones se controlan mediante un mecanismo estándar de controles, por lo tanto la interfaz *EntityTransaction* también no es utilizada en este ambiente.

Tabla 1: Arquitectura JPA interfaces del paquete `javax.persistence`

Implementación/	Descripción
Persistence:	La clase <code>javax.persistence.Persistence</code> contiene métodos estáticos de ayuda para obtener una instancia de <i>EntityManagerFactory</i> de una forma

	independiente al vendedor de la implementación de JPA.
EntityManagerFactory:	La clase <i>javax.persistence.EntityManagerFactory</i> nos ayuda a crear objetos de <i>EntityManager</i> utilizando el patrón de diseño del Factory (fábrica).
EntityManager:	La clase <i>javax.persistence.EntityManager</i> es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones. Cada <i>EntityManager</i> puede realizar operaciones CRUD (Create, Read, Update, Delete) sobre un conjunto de objetos persistentes.
Entity:	La clase <i>javax.persistence.Entity</i> es una anotación Java que se coloca a nivel de clases Java serializables y que cada objeto de una de estas clases anotadas representa un registro de una base de datos.
EntityTransaction:	Cada instancia de <i>EntityManager</i> tiene una relación de uno a uno con una instancia de <i>javax.persistence.EntityTransaction</i> , permite operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.
Query:	La interface <i>javax.persistence.Query</i> está

	implementada por cada vendedor de JPA para encontrar objetos persistentes manejando cierto criterio de búsqueda. JPA estandariza el soporte para consultas utilizando Java Persistence Query Language (JPQL) y Structured Query Language (SQL). Podemos obtener una instancia de Query desde una instancia de un EntityManager.
--	---

Fuente: Oracle Java Documentation. (2014). Package javax.persistence. Recuperado de <http://docs.oracle.com/javase/6/api/javax/persistence/package-summary.html>.

2.5 Framework Java para aplicaciones web JSF (Java Server

Faces)

La tecnología JSF es un entorno Java estándar para diseñar interfaces de usuario para aplicaciones web. Se fundamenta en el patrón MVC que simplifica y facilita el desarrollo de las interfaces de usuario en aplicaciones Java EE, lo cual es quizás uno de los elementos más complejos de manejar cuando se trabaja con aplicaciones web, además, permite administrar de manera automática el manejo de las páginas y cuenta con un conjunto de componentes del lado del servidor, lo cual permite realizar validaciones.



CIB - ESPOL

En su esencia, JavaServer Faces es un framework estándar Java para la creación de interfaces de usuario para aplicaciones web. Lo más importante es que simplifica el desarrollo de la interfaz de usuario, que es a menudo una de las partes más difíciles y tediosas de desarrollo de aplicaciones Web. [8]

La finalidad principal de la tecnología JSF es el desarrollo de aplicativos web de manera similar a como se realizan las aplicaciones de escritorio utilizando Swing, Abstract Window Toolkit (AWT), Standard Widget Toolkit (SWT) o cualquier otra interfaz de programación de aplicaciones semejante.

JSF está diseñado para simplificar el desarrollo de interfaces de usuario para aplicaciones web Java en las siguientes formas:

- Proporciona un componente central, dirigido al desarrollo independiente del cliente para crear interfaces de usuario en un ambiente web, lo que aumenta el rendimiento en la etapa de desarrollo.
- Permite el manejo de la data a través de ajax y así evitar envíos completos de los datos de los componentes de la interfaz de usuario.

- Administra de manera automática el estado de la interfaz de usuario entre los diversos envíos de una forma simple, a través de su bean administrado.
- Provee un conjunto de librerías que mejoran la experiencia de desarrollo de los programadores.
- Especifica un conjunto de buenas prácticas que son aplicadas al desarrollo web.

Adicional a estas características, JSF proporciona otra ventaja importante. Toma las mejores propiedades del desarrollo de aplicaciones web adquiridas durante varios años de experiencia y las combina en una única, completa, y estándar API para la construcción de interfaces de usuario de aplicaciones Java en la web, también añade una capacidad importante para mejorar el rendimiento en el desarrollo de aplicaciones web Java sin que esto afecte el rendimiento de la aplicación.

2.5.1 Características de la Tecnología JSF

Las principales características de la tecnología JSF son:

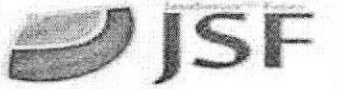

- Proporciona una API de componentes y una implementación de referencia para:

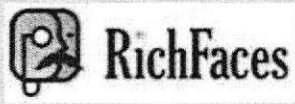
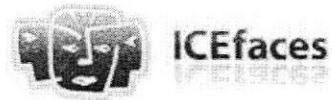




- Representar un conjunto de componentes de interfaz gráfica y administrar el estado del componente
 - Manipular eventos, validación de entrada incluido el apoyo para la validación del lado del cliente y conversión de datos
 - Definir un esquema de navegación entre páginas
 - Proporcionar extensibilidad para todas estas características.
- Dos bibliotecas de etiquetas personalizadas de JavaServer Pages (JSP) para expresar componentes de interfaz gráfica JSF en una página JSP.

2.5.2 Implementaciones JSF

Tabla 2: Implementaciones JavaServer Faces

Implementación	Empresa	Descripción
 Mojarra JSF Reference Implementation	Inicios: Sun Microsystem. Actual Oracle	Implementación de código abierto de JSF
 MyFaces	Apache Software	Implementación abierta de JSF, por

<p>Apache MyFaces</p>	<p>Foundation</p>	<p>medio del desarrollo de bibliotecas y componentes (Tomahawk, Trinidad, Tobago)</p>
 <p>RichFaces</p>	<p>Inicios: Exadel Actual: Jboss</p>	<p>Proporciona un entorno avanzado de componentes UI que integra capacidades AJAX.</p>
 <p>ICEfaces</p>	<p>ICEsoft Technologies</p>	<p>Entorno de desarrollo de código abierto para aplicaciones de internet enriquecidas (RIA). Extiende JSF para simplificar el desarrollo y ampliar el espectro de las capacidades RIA que pueden ser incluidas en cualquier aplicación web.</p>
 <p>PrimeFaces</p>	<p>Prime Technology</p>	<p>Permite que sea más fácil añadir componentes de interfaz gráfica en aplicaciones JSF. Es una librería madura y potente con más de</p>

		100 componentes de interfaz gráfica flexibles y fáciles de usar, integrado con Ajax, soporte HTML5 y más.
	OpenFaces Technologies	Es una librería de código abierto con diversos componentes JSF, un entorno Ajax y un entorno de validación de datos del lado del cliente.

Fuente: Wikipedia. (2014). JavaServer Faces. Recuperado de

http://es.wikipedia.org/wiki/JavaServer_Faces



CIB - ESPOL

2.5.3 El ciclo de vida del proceso de solicitudes JSF

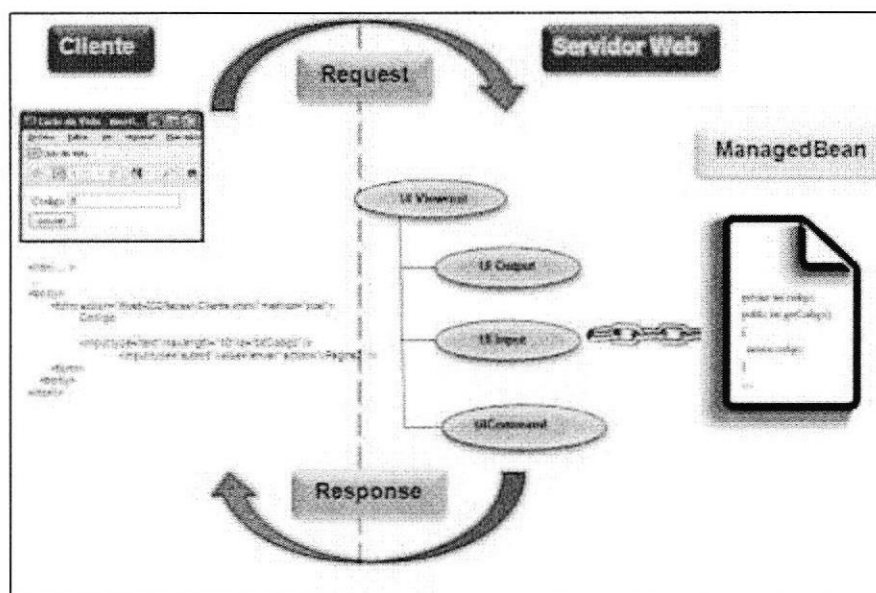


Figura 2.6: Modelo de Componentes del Servidor

Cuando una página XHTML con soporte JSF es solicitada o cuando el usuario invoca una acción en un componente de interfaz gráfica en una página XHTML con soporte JSF, es muy importante entender la secuencia exacta de eventos que ocurren en el servidor con la finalidad de atender la solicitud para visualizar o enviar una página JSF. Esta secuencia de eventos que se disparan durante las solicitudes a páginas JSF se conoce como el ciclo de vida del proceso de solicitudes JSF o simplemente como el ciclo de vida JSF.

Cuando se solicita una página JSF, por primera vez, se crea en tiempo de ejecución un árbol de componentes en la memoria. Entre las solicitudes, cuando no está pasando nada en la aplicación, el árbol de componentes a menudo se almacena en caché. Ante una solicitud posterior, el árbol es reconstituido rápidamente, y si valores de entrada del formulario son enviados en la solicitud, son procesados y las validaciones son ejecutadas. Tras la validación exitosa, los valores del modelo de lado del servidor de los campos de entrada son actualizados. Lo que sigue es procesamiento de eventos continuo, y los errores son reportados. Una vez que todo el procesamiento de eventos y actualizaciones del modelo (si es necesario) hayan finalizado, una respuesta completa finalmente se renderiza de vuelta al cliente.

El ciclo de vida del proceso de solicitudes JSF está compuesto de 6 fases:

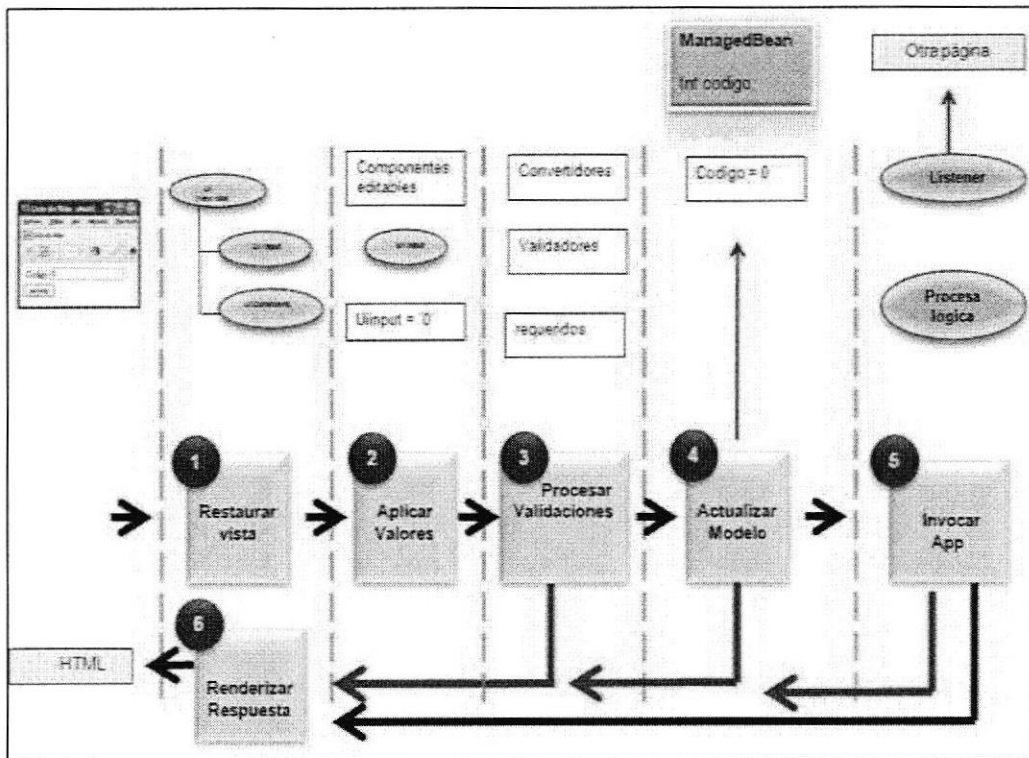


Figura 2.7: Ciclo de Vida de una página JavaServer Faces

Crear o restaurar la vista

En esta fase se restaura o crea el árbol de componentes del lado del servidor (vista) en memoria para representar la información de interfaz gráfica de un cliente.

Es tarea de la fase Restaurar la vista ya sea restaurar una vista existente de una operación anterior o crear una nueva Vista basada en una nueva solicitud. Si la solicitud es nueva, se crea

una nueva vista y se almacena en un objeto contenedor principal conocido como el FacesContext. El FacesContext sirve como almacenamiento de todos los datos pertinentes para la solicitud actual que se ejecuta a través del ciclo de vida de procesamiento de la solicitud.

Aplicar los valores de la solicitud

Durante esta fase se actualizan los componentes del lado del servidor con datos enviados desde el cliente.

Después de que la vista ha sido restaurada, la siguiente fase, conocida como la fase Aplicar los Valores de la Solicitud, realiza el trabajo de procesar los valores de solicitud entrantes o pares de nombre y valor de la información. Cada nodo componente de interfaz gráfica en la jerarquía View ahora es capaz de tomar los valores actualizados enviados por el cliente. En esta fase se obtienen los valores de la solicitud y se asignan a los objetos (beans) de la página.

Procesar validaciones

En esta fase se realizan las validaciones y conversiones de tipo de datos sobre los nuevos datos. Este proceso consiste en la validación del dato suministrado por el usuario en función de las reglas de validación definidas para el componente.

Una vez que la fase Aplicar los valores de la solicitud se ha completado, la fase Procesar validaciones se lleva a cabo. Aquí es donde ocurre la conversión y la validación de los datos de entrada. El runtime de JSF inicia esta fase llamando a un método maestro `processValidators()` (de la instancia `UIViewRoot`), que se propaga de forma recursiva hacia abajo del árbol de componentes llamando al método `processValidators()` de cada componente. Cuando se llama al método `processValidators()` de cada componente, serán invocados cualquier convertidor o validador asociado al componente.

Actualizar valores del modelo

Una vez que JSF determina que el dato es correcto, puede pasar por el árbol de componentes y actualizar los valores del modelo de



objetos del lado del servidor con los nuevos valores pasados en la solicitud.

Cuando los datos de entrada han pasado las validaciones y las conversiones, es el momento para que los datos sean promovidos y asignados al modelo de objetos enlazado al componente de interfaz gráfica de usuario. Se crea una clase Java, `UserBean`, que se registra como un bean administrado, y se enlaza sus propiedades a los diferentes componentes de interfaz gráfica de usuario en la página utilizando el lenguaje de expresiones de JSF. Es durante la fase Actualizar Valores del Modelo que el bean administrado o las propiedades del modelo de objetos se actualizan con los nuevos valores de los componentes de interfaz de usuario a las que estaban enlazados.

Invocar aplicación

En esta fase, JSF maneja cualquier acción y evento a nivel de aplicación solicitado por la página. Invoca cualquier lógica de aplicación (método) necesaria para cumplir con la solicitud y navegar a una página nueva si es necesario. Estos métodos

devuelven un String que le indica al gestor de navegación qué página tiene que devolverle al usuario.

Renderizar (desplegar) la vista

En esta última actividad se genera la página de respuesta que será enviada al navegador del usuario con todos sus elementos y valores actualizados. Almacena el estado actual de la vista para llamados futuros.

2.5.4 Java Beans, Beans Administrados (Managed Beans)

La especificación de JavaBeans, los define como unidades de software reusables que pueden ser manipulados en una herramienta de desarrollo de aplicaciones.

Los beans administrados son objetos Java que almacenan datos de las aplicaciones. Se refieren comúnmente como una clase java tradicional u objetos ligeros conocidos como POJO's (Plain Old Java Objects), que sigue ciertas convenciones de programación y se utilizan para



desacoplar por completo la vista desde el modelo de dominio de la aplicación.

Las convenciones requeridas de un bean administrado son:

- Debe tener un constructor sin argumentos.
- Sus propiedades deben ser accesibles mediante los métodos get y set que siguen una convención de nomenclatura estándar.
- Debe ser serializable.

En una aplicación JSF se usan beans administrados cuando se necesita enlazar clases Java con páginas Web; dicho de otra manera, se utilizan para aquellos datos que necesitan ser accedidos desde una página. Los beans son el medio para enlazar la GUI con la lógica aplicativa.

2.5.5 Navegación

La navegación se refiere a la forma en que se navegará entre las diferentes páginas de la aplicación a través de los eventos producidos por el usuario. Las aplicaciones JSF

utilizan estas reglas para controlar la navegación entre páginas, cada regla determina como ir de una página a las demás dentro de la aplicación. En patrón Modelo Vista Controlador el responsable de la navegación de la página es netamente el Controlador.

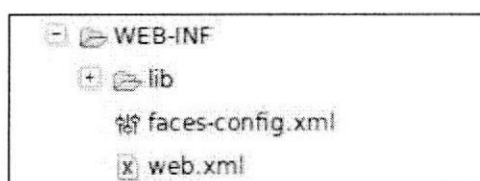


Figura 2.8: Archivo de configuración face-config.xml

Fuente: arquitecturajava. (2014). JSF Navegación Dinámica. [Figura]. Recuperado de <http://www.arquitecturajava.com/jsf-navegacion/>.

La navegación en las aplicaciones JSF está declarada en el archivo **faces-config.xml** que se encuentra bajo el directorio **WEB-INF** que serán valoradas a partir del evento generado por un componente (botón o link) en la página actual y así determinar la siguiente página o vista a mostrar. Es decir, cuando el usuario da clic en un botón, éste origina un **submit** de la forma, la aplicación debe evaluar los datos enviados y determinar que página a continuación desplegar. Para determinar la vista a mostrar,

la aplicación toma la respuesta generada por el atributo **action** del componente que inició el evento, y en base a ésta y a las reglas establecidas selecciona la página adecuada.

2.6 Framework Ajax: PrimeFaces

PrimeFaces es una suite de código abierto de componentes visuales que extiende a JSF, disponible a partir de la segunda versión de JSF. Creada y mantenida por la empresa Prime Technology. Es una de las tecnologías más frescas disponibles en la actualidad para el desarrollo de aplicaciones web con JavaServer Faces.

En pocas palabras, nunca ha sido tan fácil construir aplicaciones web poderosas, ricas y funcionales para escritorio y navegadores móviles como lo es hoy. Esto se debe al ingenio impresionante de PrimeFaces, y la excelente ingeniería que ha entrado en el diseño, implementación y evolución de esta increíble tecnología de código abierto. Es muy ligero, contiene apenas un archivo .jar que no necesita de ninguna configuración y no tiene dependencias externas.

Al igual que otras bibliotecas como RichFaces, ICEfaces, Apache MyFaces, etc., PrimeFaces incluye una amplia gama de componentes de interfaz gráfica con soporte Ajax tales como paneles de diseño, gráficos, botones, enlaces, tablas de datos, redes, calendarios, cuadros de diálogo modales, y más con soporte para temas, efectos, jQuery, arrastrar y soltar, y la mejorada API JSF2.

PrimeFaces proporciona un arsenal de componentes de interfaz gráfica con múltiples funciones con un alto grado de usabilidad, sofisticación, flexibilidad e interactividad. Estas características hacen de PrimeFaces una tecnología de gran alcance para los desarrolladores de JSF que ofrece ventajas significativas sobre los componentes JSF estándar.

Tiene soporte skinning y cuenta con más de 35 temas prediseñados. Posee una extensa documentación actualizada que la mantiene la comunidad.

Otra gran característica de PrimeFaces es el conjunto de herramientas de interfaz gráfica PrimeFaces Mobil utilizado para el desarrollo de aplicaciones web para dispositivos móviles. El diseño y la construcción de aplicaciones móviles que funcionan consistentemente a través de dispositivos es un desafío que enfrentan los desarrolladores de hoy.

Comparativa de implementaciones de JSF

Tabla 3: Comparativa de implementaciones JSF

Características	ICEfaces	RichFaces	Primefaces
Tecnología	JSF	JSF	JSF
Diversidad de componentes	70 componentes	alrededor de 70 componentes	117 componentes de código abierto
Soporte de Ajax	Se implementa de forma nativa en los componentes a través de la propiedad <code>partialSubmit</code> , lo que lo hace transparente para el desarrollador.	Para utilizarlo a parte de incluir los componentes de RichFaces hay que adicionar componentes no visuales de la librería Ajax4JSF.	Para activarlo se utiliza atributos determinados para disparar un método del servidor y para indicar los componentes gráficos a actualizar. Es

		No es tan transparente para el desarrollador porque hay que utilizar Ajax4JSF.	transparente para el desarrollador.
JavaScript	Si	Si	Si
Personalización de la interfaz de usuario	Integra el concepto de skins y dispone de 3 temas	Integra el concepto de skins y dispone de 12 temas	Integra el concepto de skins, empleando ThemeRoller y cuenta con 35 temas prediseñados
Soporte de servidores	Glassfish Apache Tomcat JBoss	Apache Tomcat 5.5 y 6.0 JBoss 4.2.x y 5.1 Glassfish 2.1.1 Weblogic Server 10 IBM WebSphere Jetty	Apache GlassFish Tomcat
Soporte de Navegadores	IE 6+ Firefox todas las versiones	IE parcialmente Firefox V 1.5+	IE 8+ V 14 Safari V5 V6

	Safari V 5.6 Opera V 9 Chrome V 25	Safari V 3+ Opera V 9.5+ Chrome V 1+	Opera V 12 Chrome V22
Componentes ricos	Si	No	Si
Documentación	Si	Si	Si

Fuente: Carrillo, E., Sosa, K. (2013). Estudio Comparativo de Librerías de Componentes para Desarrollo de Aplicaciones Web con Interfaces enriquecidas con JSF, aplicado al Sistema de Control Vehicular de la ESPOCH. Recuperado de <http://dspace.esPOCH.edu.ec/bitstream/123456789/2712/1/18T00531.pdf>.

2.7 Servidor Web Apache

El Servidor Apache HTTP es un servidor Web de tecnología Open Source sólido y para uso comercial desarrollado dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. El servidor Apache HTTP, más conocido como Apache, es el estándar en la entrega de servicios web y ha abierto el camino para la mayor expansión de las capacidades de la web. Apache inicialmente fue desarrollada únicamente para servidores Linux, sin embargo, luego se implementó para que funcione sobre otros sistemas operativos como por ejemplo Windows.

La primera versión del Apache estuvo disponible como una serie de parches, es por este motivo que se lo denominaba "patchy".

Recién en abril de 1995 lanzan la primera versión de Apache (vs. 0.6.2), meses después (diciembre del mismo año) sale a la venta la versión 1.0 resultando todo un éxito.

Ventajas

- Es un servidor Web universal por .que se puede ejecutar en un sinnúmero de sistemas operativos.
- Apache es una tecnología gratuita de código fuente abierta.
- Apache es un servidor de diseño modular y altamente configurable. Es fácil incrementar las capacidades del servidor Web Apache.
- Apache trabaja con gran cantidad de lenguajes de script, como por ejemplo Perl. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Tiene una alta configurabilidad en la creación y gestión de logs.
- Apache es multiplataforma.



2.8 Framework Spring

Es un entorno liviano que nos ayuda a desarrollar aplicaciones empresariales basadas en el lenguaje de programación Java, que se encarga de administrar los componentes de la aplicación, la ventaja es que no necesita extenderse o implementar alguna clase o interface de éste por lo tanto no es invasiva a la hora de desarrollar, gracias a esto permite integrar un POJO o un objeto Java para hacer cosas que antes sólo podían hacerse con EJBs. El beneficio de Spring no es sólo a la hora de desarrollar aplicaciones web o desarrollos del lado del servidor, sino que, cualquier aplicación Java puede aprovecharse del uso de este framework y si se lo usa de manera adecuada la aplicación quedará dividida en capas perfectamente delimitadas.

Spring se basa en tres principios (triángulo de Spring):

1. Inyección de dependencias
2. Programación Orientada a Aspectos
3. Abstracción de Servicios de Empresa

Inyección de Dependencias: Originalmente conocida como inversión de control, como se conoce, al diseñar una aplicación en

Java se dispone de muchos objetos que se relacionan entre sí, pero con este patrón de diseño, los objetos no crean o buscan sus dependencias (objetos con los cuales colabora) sino que éstas son dadas al objeto, logrando un bajo acoplamiento entre objetos de la aplicación. En resumen, se programa orientado a interfaces y se inyecta las implementaciones a través del contenedor.

Programación Orientada a Aspectos (AOP): Se lo ha planteado como un nuevo paradigma de programación que separa (encapsula uno o más conceptos transversales) las funcionalidades complementarias (por ejemplo la seguridad, el manejo de errores, optimización de memoria etc.) de la lógica del negocio.

La AOP busca modularizar estos servicios y aplicarlos de manera declarativa a los componentes que deban afectar. Al trabajar con AOP se centraliza el código que implementa estas funcionalidades y se configura Spring para que las ejecute donde se necesite.

Abstracción de Servicios de Empresariales: con este principio Spring se encarga aislar al programador de muchas tareas

haciéndolas transparentes, como por ejemplo el acceso a los datos, unificar excepciones, gestionar conexiones, etc., aumentando su productividad en el desarrollo de la aplicación.

Módulos de Spring

A continuación se muestra los módulos con los que cuenta Spring:



Figura 2.9: Principales Módulos de Spring

Fuente: Global Mentoring. (2014). Módulos de Spring Framework. [Figura]. Recuperado de <http://globalmentoring.com.mx/curso-spring-framework/>.

Las características principales del framework Spring son:

- Reduce significativamente la programación orientada a aspectos.
- Reduce el acceso a datos.

- Simplificación e integración con JEE
- Soporte para planificación de trabajos.
- Soporte para envío de correos electrónicos.
- Interacción con lenguajes dinámicos (como BeanShell, JRuby, y Groovy).
- Soporte para acceso a componentes remotos.
- Manejo de Transacciones.
- Posee su propio framework MVC.
- Posee su propio Web Flow.
- Manejo de excepciones simplificado.

2.9 Metodologías de Desarrollo Ágil

Hoy en día las Metodologías de Desarrollo de Software son muy diferentes incluso a aquellas con las que se desarrollaban años atrás y han sido creadas con la finalidad de proponer al desarrollador una guía al momento de crear un nuevo producto empleando una serie de técnicas y procedimientos.

En el desarrollo de software la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo. Una metodología consiste en un lenguaje de modelamiento y un proceso. El

lenguaje de modelamiento es la representación gráfica como el Lenguaje Unificado de Modelado (UML) y el proceso de desarrollo de software no es más que el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de información.

De antemano se conoce que el desarrollo de un software no es una tarea sencilla, prueba de aquello es que existen varias metodologías para la elaboración del software, dichas metodologías son:

- Las Metodologías Tradicionales o Pesadas: Se concentran en el control de procesos, estableciendo rigurosamente las actividades a realizar, herramientas y notaciones que se usarán; y
- Las Metodologías Ágiles o Ligeras: Que se enfocan en otorgar un mayor valor a la interacción del desarrollador del software con el cliente (ser humano) en completa colaboración, mostrándole versiones funcionales del producto en intervalos de tiempo cortos, para que éste pueda evaluar y sugerir cambios en el producto según se va desarrollando.





Figura 2.10: Metodología Tradicional vs. Metodología Ágil

Fuente: Global Mentoring. (2014). Módulos de Spring Framework. [Figura]. Recuperado de <http://globalmentoring.com.mx/curso-spring-framework/>.

Dicho de otra manera, la diferencia entre las metodologías tradicionales o clásicas versus las metodologías ágiles es que mientras la primera está guiada por una intensa planificación durante todo el proceso de desarrollo, donde se pone más énfasis en la etapa de análisis y diseño antes de la construcción del sistema y la segunda, que actualmente están teniendo gran aceptación, es un proceso mucho más ágil porque el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el

método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento)

2.9.1 Metodología ICONIX

El Proceso ICONIX es una metodología de desarrollo de software impulsado por casos de uso. Su enfoque principal es sobre cómo a partir de casos de uso obtener codificación confiable en el menor tiempo posible.

Aunque gran parte del Proceso ICONIX puede considerarse ágil, algunas partes del mismo están en marcado contraste con el pensamiento ágil reciente. En particular, los métodos ágiles a menudo indican que no hay que molestarse en mantener la documentación de diseño y código fuente sincronizados. La teoría es que una vez que el código ha sido escrito, no necesitamos más los diagramas. El Proceso de Iconix, por otro lado, sugiere exactamente lo contrario: una estrecha sincronización de la documentación de diseño con el código, es más rápido, más fácil de mantener, y más preciso (es decir, más cerca de los

requerimientos del cliente) el proyecto tendrá más versiones sucesivas.

Afortunadamente, el Proceso ICONIX también apunta a facilitar este proceso (de nuevo a diferencia de otros procesos ágiles). Para lograr esta estrecha sincronización entre diagramas y código, hay que reducir el número de diagramas que se tiene que elaborar (y por lo tanto mantener). También es importante conocer que diagramas van a ser importantes para iteraciones continuas y cuáles pueden ser descartados con seguridad. "Mínimo pero suficiente" análisis y el diseño son precisamente el núcleo del proceso de ICONIX.

ICONIX se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. Su principal objetivo es que a través de los casos de uso se obtenga el sistema final, aunque esto no restringe el uso de los otros diagramas de UML de los ya estipulados si se consideran convenientes. En pocas palabras, el proceso ICONIX describe cómo pasar

de casos de uso al código de forma fiable, en el menor número de pasos posibles. [9].

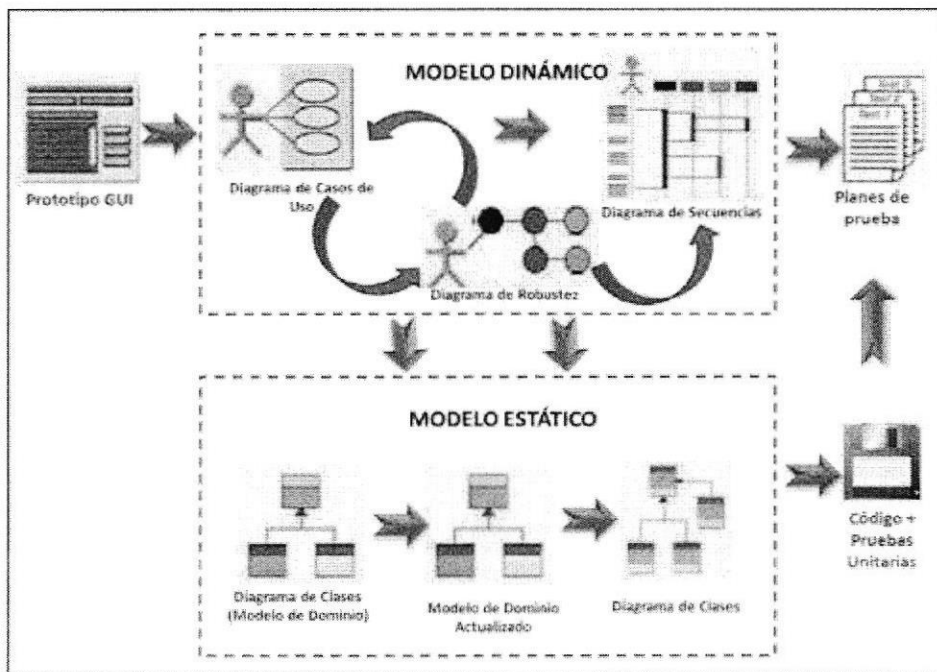


Figura 2.11: Metodología ICONIX

Fuente: Blog iconixinfo. (2014). ICONIX. [Figura]. Recuperado de <http://iconixinfo.blogspot.com/>.

El Proceso de Iconix se divide en flujos de trabajo dinámicos y estáticos, que son altamente iterativos: se puede ir a través de una iteración de todo el proceso de un lote pequeño de casos de uso hasta el código fuente y las pruebas unitarias. Por esta razón, el proceso de Iconix se

adapta bien a los proyectos ágiles, donde se necesita una rápida retroalimentación de factores tales como los requisitos, el diseño y las estimaciones.

2.9.2 Características Principales

ICONIX presenta las siguientes características principales:

- **Iterativo e Incremental**

Durante el desarrollo del modelo del dominio y la identificación de los casos de uso ocurren varias iteraciones. El modelo estático es incrementalmente refinado por los modelos estáticos.

El ciclo de vida incremental consiste en desarrollar por partes el producto de manera que puedas integrarlas funcionalmente. En cada ciclo de iteración se revisa y mejora el producto.

- **Trazabilidad**

Cada paso que se realiza está referenciado por algún requisito. Se define la trazabilidad como la capacidad de



seguir una relación entre los diferentes artefactos de software elaborados.

- **Dinámica del UML**

Ofrece un uso dinámico del UML porque utiliza algunos diagramas UML, sin exigir la utilización de todos, como en el caso de Rational Unified Process (RUP).

2.9.3 Tareas de ICONIX

La metodología está formada por las siguientes tareas:

1. Análisis de Requisitos
2. Análisis y Diseño Preliminar
3. Diseño Detallado
4. Implementación

1. Análisis de Requisitos

Se realiza:

Modelo de Casos de Uso: Identificar los casos de uso del sistema con los actores involucrados. Describir las

acciones o el comportamiento que un usuario realiza dentro del sistema.

Prototipo de Interfaz de Usuario: Crear un prototipo rápido de las interfaces del sistema en el que analistas y clientes deben estar de acuerdo, de forma que los clientes comprendan mejor el sistema. (Dinámico/ los usuarios se hacen participantes activos en el desarrollo).

Modelo de Dominio: Realizar un diagrama de clases de alto nivel para identificar objetos y cosas del mundo real que intervienen con el sistema. (Estático)

2. Análisis y Diseño Preliminar

Se realiza:

Descripción de los Casos de Uso: Describir los Casos de Uso a través de una ficha de caso de uso, como un flujo principal de acciones que puede contener flujos alternativos y flujos de excepción. Esta ficha está

formada por un nombre, una descripción, una precondición que se debe cumplir antes de iniciar y una post-condición que se debe cumplir al terminar.

Diagrama de Robustez: Ilustrar gráficamente las interacciones de los objetos participantes de un caso de uso. Este diagrama facilita el reconocimiento de objetos. Permite analizar el texto narrativo de cada caso de uso e identificar un conjunto inicial de objetos participantes de cada caso de uso.

Actualizar el modelo de Dominio: Actualizar el diagrama de clases que se definió en el modelo de Dominio con las nuevas clases, corregir las ambigüedades y agregar nuevos atributos que son descubiertos en los diagramas de robustez.

3. Diseño Detallado

Se realiza:



Diagramas de Secuencia: Especificar el comportamiento a través de un diagrama de secuencia realizando un diagrama para cada caso de uso, para mostrar en detalle cómo se va a implementar el caso de uso. Identificar los mensajes entre los diferentes objetos. Es necesario utilizar los diagramas de colaboración para representar la interacción entre los objetos.

Actualizar el modelo de dominio: Finalizar el modelo estático (diagrama de clases), con los detalles del diseño agregando las operaciones a las clases de dominio en el diagrama de clases.

4. Implementación

Se realiza:

Codificación y pruebas unitarias: Escribir y generar el código y las pruebas unitarias.

Integración y escenario de prueba: Realizar las pruebas de integración basada en los casos de uso, para probar tanto el flujo básico y los flujos alternos.

Revisión del código y actualización del modelo: Preparar la próxima ronda de trabajo de desarrollo.

CAPÍTULO 3

3 ANÁLISIS DE REQUISITOS

3.1 Descripción de la situación actual

Realizando una investigación exhaustiva a través de las entrevistas realizadas a los usuarios del sistema: Secretaria GTH, Analista GTH, Jefe GTH, Jefe de Planta; se detectó que la empresa lleva a cabo los procesos relacionados con la elaboración de turnos, control de marcaciones e ingreso de solicitudes de permisos del personal de forma manual con ayuda de algunos programas utilitarios (Excel, Word.), lo que ocasiona que la información no se la pueda obtener de forma rápida, precisa y segura.

A continuación se describen los procesos mencionados:

Elaboración de Turnos del Personal

La elaboración de turnos es un proceso que se realiza de forma mensual para el personal que cuenta con turno rotativo como es el caso del personal de producción que trabaja en la planta. El Jefe de planta dos días antes de finalizar el mes debe armar los turnos para el próximo mes, este trabajo lo realiza en una hoja de Excel donde debe armar los horarios para los grupos de la siguiente forma:

Se tienen tres horarios establecidos Matutino de 07H00 – 15H00, el Vespertino de 15H00 – 22H00 y el Nocturno de 22H00 – 07H00.

Un primer grupo se define la primera semana del mes el horario matutino, la 2da semana el horario vespertino, la 3era semana el horario nocturno y la 4ta semana el horario matutino. Para el siguiente mes la 1era semana le tocaría el horario vespertino y así sucesivamente. Se arman tres grupos de la forma expuesta y se asignan al personal que deben cumplirlo.



Lo que ocurre muy frecuentemente con el personal de producción son los cambios de turno repentinos, los cuales no son comunicados a tiempo al Departamento de Gestión de Talento Humano, lo que ocasiona que al realizar el cierre para el rol de pagos se realicen descuentos al personal por supuestas faltas e incumplimiento de la jornada de trabajo.

Tarjar marcaciones del Personal

La secretaria del Dpto. de Gestión de Talento Humano cada mes debe facilitar un informe de atrasos y faltas del personal, al Dpto. Financiero para que pueda generar el rol de pagos. Para realizar este informe la secretaria de Gestión de Talento Humano, se dedica durante dos días anteriores a tarjar las marcaciones realizadas por los empleados en el reloj biométrico.

El reloj biométrico genera un archivo texto el cual lo descarga la secretaria de forma diaria para no sobrecargar la capacidad del reloj biométrico. Cada línea dentro del archivo representa una marcación realizada por el empleado. Es decir que normalmente debería existir en el archivo por empleado el número de líneas acorde a su turno de trabajo. Algunos empleados como el caso del

personal administrativo realizan 4 marcaciones, por el lunch, y otros realizan sólo dos marcaciones. Pero esto se ve afectado para el caso del personal con turno rotativo, ya que sus marcaciones estarán repartidas entre dos archivos, o cuando el empleado realiza más marcaciones de las estipuladas.

Cada archivo es abierto en una hoja de Excel por la secretaria de Talento Humano, quien por empleado debe ir recopilando las marcaciones realizadas, y en el caso de que el empleado tenga más marcaciones de las que debería, debe escoger la que más se ajustan al turno del empleado. Las marcaciones seleccionadas son colocadas en un archivo de Excel. Una vez recopilada la información para todos los empleados se sumaliza los atrasos y las faltas. Se formatea el informe y se envía al Dpto. Financiero.

Registrar solicitudes de permisos del personal

Las solicitudes de permiso ocasional y de acción de personal, son realizadas por cada empleado. El empleado debe llenar el formulario respectivo (permiso ocasional o acción de personal) en conjunto con su jefe inmediato quien debe verificar y aprobar la solicitud. Posteriormente el empleado debe hacer llegar el

formulario con la aprobación de su jefe al Dpto. de Gestión Talento Humano para su validación y aprobación definitiva.

Los permisos ocasionales son aquellos permisos que solicita el personal de manera diaria y con una duración de 4 horas. Estos permisos pueden ser solicitados por asunto personal o por consulta médica.

Las acciones de personal son aquellos permisos o licencias que se solicitan por varios días, como es el caso de vacaciones, cursos, por maternidad, por paternidad, enfermedad, etc.

Lo que con frecuencia ocurre es que las solicitudes no se hacen llegar a tiempo al Dpto. de Gestión Talento Humano o se traspapelan, y esto ocasiona que se produzcan descuentos por supuestas inasistencia al puesto de trabajo.

3.2 Problemas y especificación de requerimientos

3.2.1 Especificación de Requerimientos

En esta parte se detallan los requisitos funcionales y no funcionales que deberán ser cubiertos por el sistema. Los requisitos que a continuación se muestran han sido declarados en lenguaje natural para que sean entendibles por cualquier usuario.

Se presentan a través de una lista y fueron obtenidos a partir del levantamiento de información realizado.

3.2.2 Requisitos Funcionales

La **tabla 4** que sigue a continuación presenta la lista de requisitos funcionales del sistema. Se detallan en cuatro columnas, donde la primera columna **ID** se refiere a un identificador donde las dos primeras letras indican que es un requisito funcional (**RF**) y a continuación un número secuencial, la segunda y tercera columna **requisito** y **descripción** definen el requisito funcional, y la última columna **usuario** se refiere a quien debe realizar el requisito.

Tabla 4: Requisitos Funcionales del Sistema

ID	Requisito	Descripción	Usuario
RF-001	Ingreso al sistema	Los usuarios para acceder al sistema, deben ingresar en la página principal de inicio un nombre de usuario y una contraseña, la cual será validada por el sistema, dando el acceso al sistema o enviando un mensaje para que ingrese nuevamente sus datos.	Usuario
RF-002	Perfiles del sistema	Los usuarios del sistema deben tener asociado un perfil, que será utilizado para activar las opciones a las que pueda acceder y utilizar en el sistema.	Administrador del sistema
RF-003	Ingresar turno fijo	El sistema debe permitir el registro de los turnos del personal administrativo con horario fijo, detallando los datos del turno tales como: el grupo del turno, el número de marcaciones (si se va a considerar lunch), los días del turno y las horas y minutos de cada marcación. El sistema asignará un número de turno automático y colocará el estado del turno en A "turno activo".	Secretaria GTH
RF-004	Ingresar Turno con frecuencia	El sistema debe permitir el registro de los turnos del personal de planta	Jefe Planta



	de marcación y descanso mensual	cuyo horario tiene una frecuencia de marcación y descanso (trabajan 24 horas y descansan 48 o 72 horas) por mes, detallando el grupo del turno, el número de marcaciones, la frecuencia de marcación en horas, la frecuencia de descanso en horas, la fecha de inicio del mes para la que se va a generar el turno, la hora y minutos de cada marcación. El sistema asignará un número de turno automático y genera el turno con los días a partir de la fecha de inicio y la frecuencia de marcación y descanso.	
RF-005	Ingresar Turno Variable	El sistema debe permitir el registro de los turnos del personal de mantenimiento cuyo horario es diferente cada día del mes dependiendo de la disponibilidad del personal para cubrir el turno, detallando el grupo del turno, el número de marcaciones, la fecha de inicio del mes para la que se va a generar el turno. El sistema generará el turno con los días a partir de la fecha de inicio hasta el último día del mes y se detalla la hora y minutos de cada marcación por fecha.	Jefe Planta
RF-006	Asignar Turno	El sistema debe facilitar que un turno sea normal, con frecuencia o variable que haya sido registrado	Secretaria GTH Jefe Planta

		previamente, sea asignado a uno o más empleados para posteriormente realizar la carga de marcaciones de los empleados asignados.	
RF-007	Subir archivo texto de marcaciones	El sistema debe permitir subir las marcaciones de los empleados generadas por el reloj biométrico en un archivo de texto, para posteriormente visualizarlas en los respectivos informes de control.	Secretaria GTH
RF-008	Ingresar Novedades del Personal	El sistema debe proporcionar el ingreso de la solicitud de aquellas novedades del personal que afectan a sus marcaciones, detallando los datos de la solicitud tales como: código y nombres del empleado, el tipo de novedad (marcaciones faltantes, permisos, vacaciones, capacitación, otros), la novedad, si es de un día o varios días, la marcación que se va a afectar (entrada, salida a lunch, retorno de lunch, salida), la fecha de inicio y fin, y una observación. El sistema asignará un número de solicitud automático y colocará el estado de la solicitud en / <i>"Ingresada"</i> .	Usuario
RF-009	Aprobar Novedades del Personal	El sistema debe permitir que una solicitud de novedades del personal que ha sido ingresada sea aprobada. Una vez aprobada la solicitud se	Jefe Planta

		colocará el estado de la misma en A "Aprobada"	
RF-010	Registrar empleado	El sistema debe permitir el registro de los datos de identificación de cada empleado de la empresa.	Secretaria GTH
RF-011	Actualizar empleado	El sistema debe permitir actualizar la información que se ha registrado del empleado.	Secretaria GTH
RF-012	Eliminar empleados	El sistema debe permitir eliminar (inactivar) los empleados registrados y que hayan dejado de laborar en la empresa.	Secretaria GTH
RF-013	Consultar empleado	El sistema debe permitir consultar la información que se ha registrado del empleado	Secretaria GTH
RF-014	Registrar departamento	El sistema debe permitir el registrar los departamentos de la empresa.	Secretaria GTH
RF-015	Actualizar departamento	El sistema debe permitir actualizar la información que se ha registrado del departamento.	Secretaria GTH
RF-016	Eliminar departamento	El sistema debe permitir eliminar (inactivar) los departamentos registrados que no se utilicen.	Secretaria GTH
RF-017	Consultar departamentos	El sistema debe permitir consultar los departamentos registrados.	Secretaria GTH
RF-018	Registrar cargo	El sistema debe permitir registrar los cargos que se asignan a los empleados de la empresa.	Secretaria GTH
RF-019	Actualizar cargo	El sistema debe permitir actualizar la información que se ha registrado del	Secretaria GTH

		cargo.	
RF-020	Eliminar cargo	El sistema debe permitir eliminar (inactivar) los cargos registrados que no se utilicen.	Secretaria GTH
RF-021	Consultar cargos	El sistema debe permitir consultar los cargos registrados.	Secretaria GTH
RF-022	Consultar e imprimir turno del empleado	El sistema debe permitir la consulta e impresión del turno activo del empleado, detallando la fecha de asignación del turno, el número de marcaciones, los días y las horas/minutos de marcación.	Secretaria GTH Jefe Planta Usuario
RF-023	Consultar e imprimir los turnos activos ingresados	El sistema debe permitir la consulta e impresión de todos los turnos activos registrados, detallando	Secretaria GTH Jefe Planta
RF-024	Consultar e imprimir los empleados asignados a un turno específico	El sistema debe proporcionar la consulta e impresión de los empleados que están asignados a un turno específico detallando código, cédula, nombres y departamento de cada empleado.	Secretaria GTH Jefe Planta
RF-025	Consultar e imprimir las marcaciones dentro de un rango de fechas del personal con turno fijo	El sistema debe facilitar la consulta e impresión de las marcaciones de los empleados con turno fijo dentro de un rango de fechas por departamento detallando la fecha, el turno y la marcación de esa fecha del empleado, el número de solicitud de novedad y la observación si la marcación fue justificada, el total de	Secretaria GTH Jefe Planta Jefe GTH

		horas trabajadas y el total de atrasos.	
RF-026	Consultar e imprimir las marcaciones diarias del personal	El sistema debe facilitar la consulta e impresión de las marcaciones de los empleados de una fecha específica, mostrando los empleados agrupados por departamento, detallando el código y nombres del empleado, el turno y la marcación y el total de horas trabajadas y el total de atrasos.	Secretaria GTH Jefe Planta
RF-027	Consultar e imprimir las marcaciones dentro de un rango de fechas del personal con turno variable	El sistema debe facilitar la consulta e impresión de las marcaciones de los empleados con turno variable dentro de un rango de fechas por departamento detallando la fecha, el turno y la marcación de esa fecha del empleado, el número de solicitud de novedad y la observación si la marcación fue justificada, el total de horas trabajadas y el total de atrasos.	Secretaria GTH Jefe Planta Jefe GTH
RF-028	Consultar e imprimir las marcaciones de un empleado específico dentro de un rango de fechas	El sistema debe facilitar la consulta e impresión de las marcaciones de un empleado en particular dentro de un rango de fechas detallando la fecha, el turno y la marcación de esa fecha del empleado, el número de solicitud de novedad y la observación si la marcación fue justificada, el total de horas trabajadas y el total de atrasos.	Secretaria GTH Jefe Planta Jefe GTH Usuario
RF-029	Consultar e imprimir las marcaciones	El sistema debe facilitar la consulta e impresión de las marcaciones descartadas de un empleado en	Secretaria GTH Jefe Planta



	descartadas de un empleado específico dentro de un rango de fechas	particular dentro de un rango de fechas detallando la fecha, la hora de marcación, una observación indicando el por qué se descartó la marcación y el archivo texto de la marcación.	Jefe GTH Usuario
RF-030	Consultar e imprimir las solicitudes de permisos (novedades) de un empleado específico dentro de un rango de fechas	El sistema debe facilitar la consulta e impresión de las solicitudes de permisos (ingresadas y/o aprobadas) de un empleado en particular dentro de un rango de fechas detallando el número de solicitud, el motivo del permiso, el número de días solicitados, la fecha de inicio, la fecha de finalización, una observación y la fecha de registro de la solicitud; agrupado por el tipo de motivo del permiso (marcaciones faltantes, permisos, vacaciones, capacitación, otros).	Secretaria GTH Jefe Planta Jefe GTH Usuario
RF-031	Consultar e imprimir las solicitudes de permisos (novedades) por departamento dentro de un rango de fechas	El sistema debe facilitar la consulta e impresión de las solicitudes de permisos (ingresadas y/o aprobadas) de los empleados agrupados por departamento, por empleado y por tipo de motivo del permiso (marcaciones faltantes, permisos, vacaciones, capacitación, otros) dentro de un rango de fechas detallando el número de solicitud, el	Secretaria GTH Jefe Planta Jefe GTH

		motivo del permiso, el número de días solicitados, la fecha de inicio, la fecha de finalización, una observación y la fecha de registro.	
--	--	--	--

3.2.3 Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe tener. La tabla 5 presenta la lista de la definición de los requerimientos no funcionales del Sistema, los cuales restringen o condicionan el desarrollo e implantación del sistema. Se detallan en dos columnas, donde la primera columna *ID* se refiere a un identificador donde las tres primeras letras indican que es un requisito no funcional (**RNF**) y a continuación un número secuencial y la segunda columna *descripción* define el requisito no funcional.

Tabla 5: Requisitos No Funcionales del Sistema

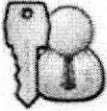

ID	Descripción
RNF-001	La interfaz del sistema debe ser implementada como una aplicación web sencilla, amigable para el usuario y de fácil entendimiento, uso y navegación.
RNF-002	El sistema será desarrollado con software libre, utilizando el


	lenguaje de programación JAVA y utilizará el estándar HTML para el diseño de las páginas web del sistema. Con esto se garantiza que el código HTML generado sea interpretado por cualquiera de los navegadores existentes en el mercado.
RNF-003	El sistema debe ser diseñado e implementado para que su mantenimiento sea fácil, de manera que permita ampliaciones, modificaciones y un mejoramiento progresivo de sus funcionalidades.
RNF-004	El sistema deberá estar disponible de forma continua, de manera que sus servicios puedan ser accedidos por los usuarios las 24 horas y los 7 días.
RNF-005	El sistema se ejecutará bajo la plataforma Linux Centos, aunque también puede ejecutarse bajo Windows.
RNF-006	Como Servidor Web se requiere instalar Servidor Apache Tomcat 7.0.34, y para el Servidor de Base de Datos se requiere instalar MySQL 5.5.
RNF-007	El sistema será diseñado bajo la arquitectura cliente/servidor de tres capas.
RNF-008	Los usuarios que se conectarán al sistema desde cualquier PC cliente deben tener instalado un navegador web, preferiblemente Firefox.
RNF-009	Los PC clientes requieren de al menos de un procesador Intel Celeron o superior y un mínimo de 2GB de memoria RAM.
RNF-010	El Servidor Web requiere de al menos de un procesador Intel Core i7 o superior, un mínimo de 4 GB de memoria RAM y 1 TB de espacio disponible en disco.


3.3 Sistema propuesto


3.3.1 Lista de Actores del Negocio


Tabla 6: Actores del Negocio

<p>ACT-01</p> 	<p>Administrador del Sistema</p>
<p>Rol</p>	<p>Este actor representa al personal del área de Sistemas responsable de la administración del Sistema Web de Marcaciones.</p>
<p>Funciones</p>	<p>Dar de alta o baja a un usuario. Dar de alta al usuario a las opciones del Sistema.</p>
<p>ACT-02</p> 	<p>Jefe GTH</p>
<p>Rol</p>	<p>Este actor representa la persona responsable de administrar y organizar de forma general el Departamento de Gestión de Talento Humano.</p>
<p>Funciones</p>	<p>Revisa informes de las asistencias, atrasos, horas extras, permisos del personal. Revisa estadísticas de las asistencias, atrasos, horas extras, permisos del personal.</p>

ACT-03 	Secretaria GTH
Rol	Este actor representa al personal secretarial que sirve de apoyo a las actividades propias de la oficina.
Funciones	Registra al personal nuevo. Ingresa turnos fijos del personal. Asigna turnos fijos al personal. Carga las marcaciones del reloj biométrico.

ACT-04 	Analista GTH
Rol	Este actor representa la persona responsable de analizar actividades administrativas con el Recurso Humano.
Funciones	Revisa y aprueba las solicitudes de novedades del personal por permisos, vacaciones, licencias, etc.

<p>ACT-05</p> 	<p>Jefe Mantenimiento</p>
<p>Rol</p>	<p>Este actor representa la persona responsable de administrar y organizar el área de Mantenimiento.</p>
<p>Funciones</p>	<p>Registra turnos rotativos (frecuencia y variable) de su personal.</p> <p>Asigna los turnos a su personal.</p>

<p>ACT-06</p> 	<p>Jefe Planta</p>
<p>Rol</p>	<p>Este actor representa la persona responsable de administrar y organizar el área de la Planta.</p>
<p>Funciones</p>	<p>Registra turnos rotativos (frecuencia y variable) de su personal.</p> <p>Asigna los turnos a su personal.</p>

3.3.2 Modelo de Casos de Uso

Tabla 7: Diagrama de Casos de Uso del Sistema de Control de Turnos y Solicitudes

Nombre Caso	Sistema de Control de Turnos, Asistencia y Solicitudes de		
Uso	Novedades del Personal		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actores	<i>Secretaria GTH, Jefe GTH, Analista GTH, Jefe Área, Usuario</i>		
Objetivo: Gestionar las diferentes tareas que se dan en el área de GTH en relación al control de asistencia del personal como: la asignación de los turnos, la carga de las marcaciones del reloj biométrico y las solicitudes del personal.			

Diagrama de Casos de Uso

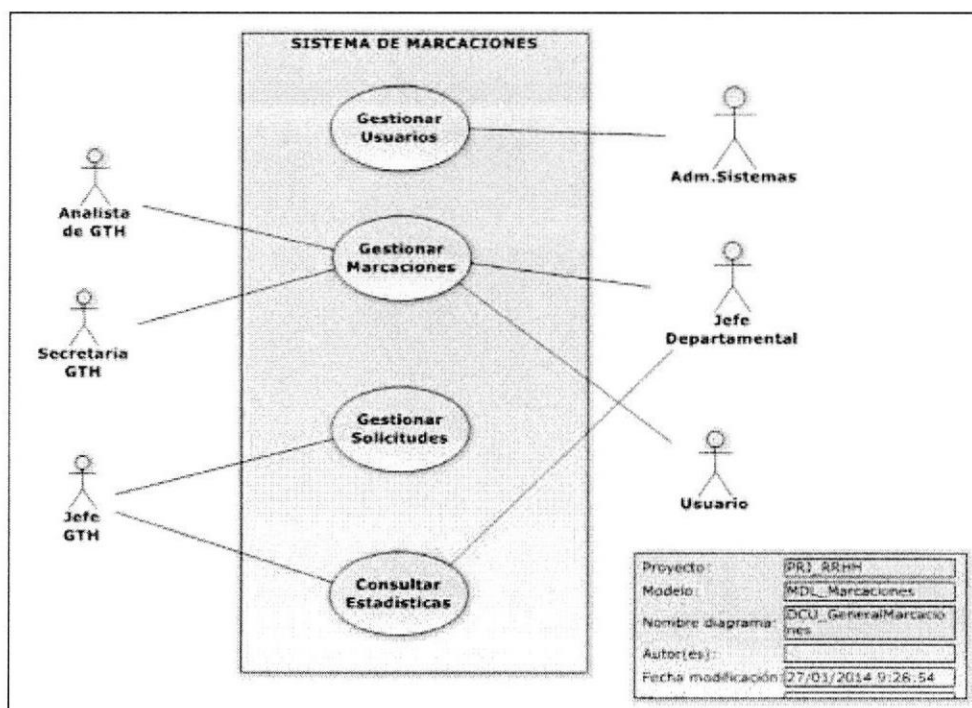


Tabla 8: Diagrama de Casos de Uso Gestión de Marcaciones

Nombre Caso Uso	Gestión de Marcaciones		
Creación	José Caicedo Salazar Julio Litardo Unda	Fecha	20/01/2014
Actores	<i>Secretaria GTH, Jefe GTH, Analista GTH, Jefe Área, Usuario</i>		
Objetivo:	Gestionar las diferentes tareas relacionadas con el control de las marcaciones del personal que pueden darse por diferentes motivos. Tareas relacionadas con la carga del archivo de marcaciones del reloj biométrico, la creación y asignación de los diferentes turnos del personal y la consulta de estadísticas de asistencia, atrasos y horas extras para la parte gerencial.		

Diagrama de Casos de Uso

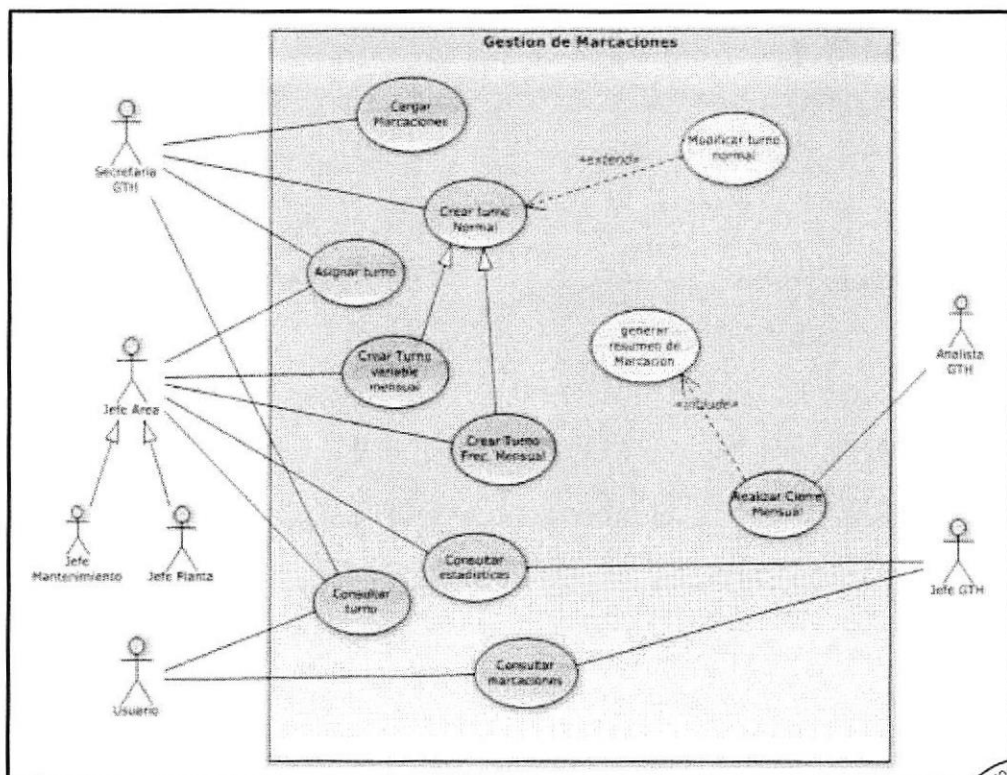


Tabla 9: Diagrama de Casos de Uso Gestión de Solicitudes

Nombre Caso Uso	Gestión de Solicitudes de Acción de Personal		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actores	<i>Secretaria GTH, Analista GTH</i>		
Objetivo:	Gestionar las diferentes tareas que se dan en el área de GTH en relación al seguimiento de las solicitudes de acción de personal. Tareas relacionadas con el ingreso, seguimiento y aprobación de las solicitudes y como estas afectan a las marcaciones del personal.		

Diagrama de Casos de Uso

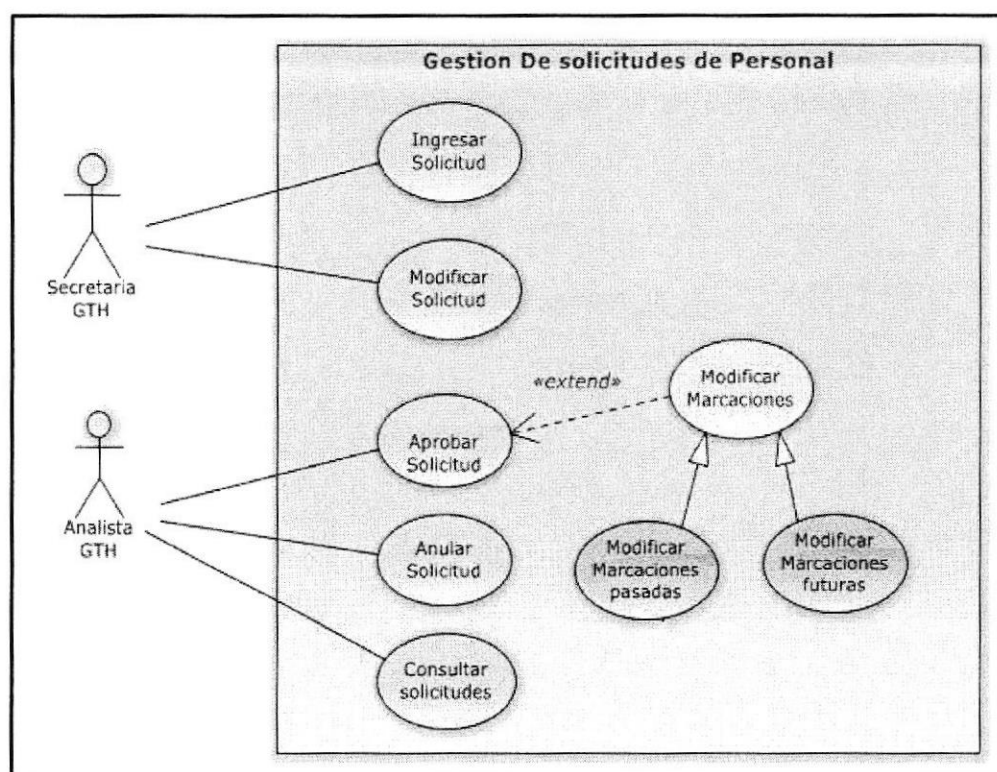
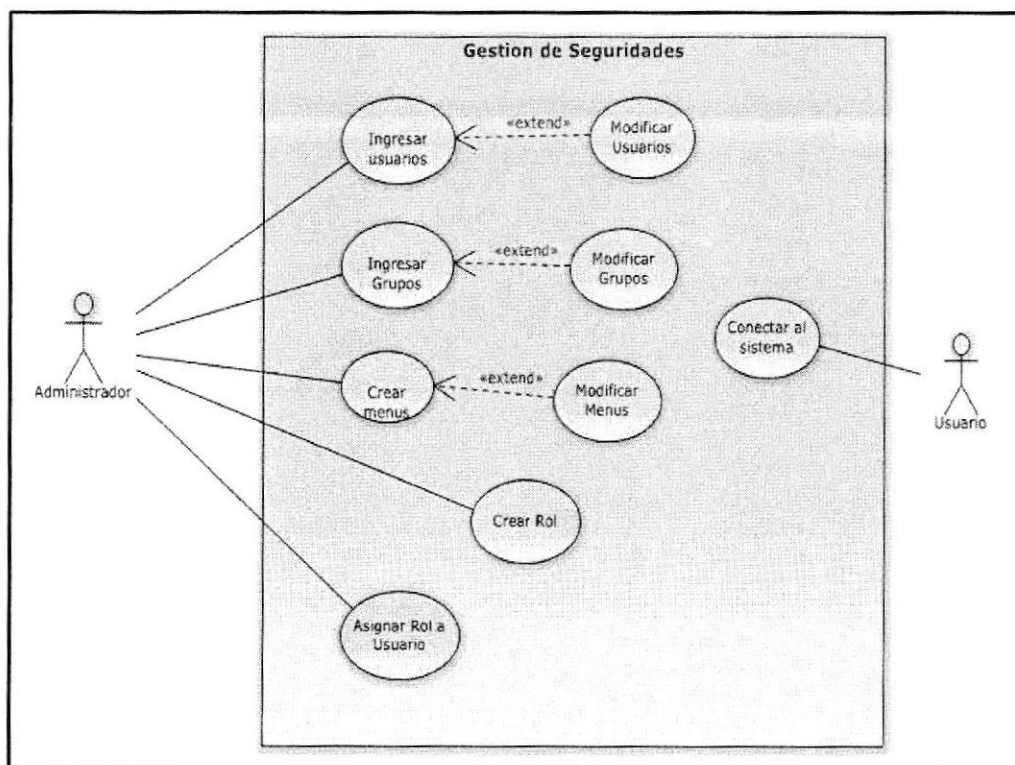


Tabla 10: Diagrama de Casos de Uso Gestión de Seguridades

Nombre Caso Uso	Gestión de Seguridades		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actores	<i>Secretaria GTH, Jefe GTH, Analista GTH, Jefe Área, Usuario</i>		
Objetivo:	Gestionar las diferentes tareas relacionadas en cuanto a la seguridad del sistema como: creación de usuarios, creación de grupos, la creación de menús y la definición de roles a los usuarios para acceder al sistema.		

Diagrama de Casos de Uso



3.3.3 Prototipo de interfaz de usuario

Tabla 11: Interfaz de usuario crear turno normal

Nombre Caso de Uso		Crear turno normal	
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		

Administración de turnos

Código: Descripción:

Grupo: Número de Marcaciones:

Tempo de Gracia: Lunch: Tipo Turno:

Marcación Entrada Salida									
Sel.	Dia	Entrada Empresa		Salida Lunch		Retorno Lunch		Salida Empresa	
		Hora	Minutos	Hora	Minutos	Hora	Minutos	Hora	Minutos
<input type="checkbox"/>	DOMINGO	0	0	0	0	0	0	0	0
<input checked="" type="checkbox"/>	LUNES	8	0	12	30	13	30	17	0
<input checked="" type="checkbox"/>	MARTES	8	0	12	30	13	30	17	0
<input checked="" type="checkbox"/>	MIERCOLES	8	0	12	30	13	30	17	0
<input checked="" type="checkbox"/>	JUEVES	8	0	12	30	13	30	17	0
<input checked="" type="checkbox"/>	VIERNES	8	0	12	30	13	30	17	0
<input type="checkbox"/>	SABADO	0	0	0	0	0	0	0	0

Tabla 12: Interfaz de usuario crear turno frecuencia mensual

Nombre Caso de Uso		Crear turno frecuencia mensual	
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		

Generación de turno Mensual

Código: Descripción:

Grupo: Mercaderías:

Tiempo de Gracia: Lunch:

Mercadería: Descanso: Fecha de Inicio:

Seteo de Entrada Salida

Entrada Empresa		Salida Lunch		Retorno Lunch		Control Boche	
Hora	Minutos	Hora	Minutos	Hora	Minutos	Hora	Minutos
7	30	12	0	12	30	23	30

Horario Generado

Fecha	Entrada Empresa		Salida Lunch		Retorno Lunch		Control Boche	
	Hora	Minutos	Hora	Minutos	Hora	Minutos	Hora	Minutos
02-01-2014	7	30	12	0	12	30	23	30
03-01-2014	7	30	0	0	0	0	0	0
06-01-2014	7	30	12	0	12	30	23	30
07-01-2014	7	30	0	0	0	0	0	0
10-01-2014	7	30	12	0	12	30	23	30
11-01-2014	7	30	0	0	0	0	0	0
14-01-2014	7	30	12	0	12	30	23	30
15-01-2014	7	30	0	0	0	0	0	0
18-01-2014	7	30	12	0	12	30	23	30
19-01-2014	7	30	0	0	0	0	0	0
22-01-2014	7	30	12	0	12	30	23	30

Tabla 13: Interfaz de usuario crear turno variable mensual

Nombre Caso de Uso		Crear turno variable mensual	
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		

Generacion de turno Mensual variable

Código: Descripción:

Grupo: Número de Marcaciones:

Tempo de Gracia: Lunch:

Fecha de inicio:

Horario Generado

Sel	Fecha	Entrada Empresa		Control Noche	
		Hora	Minutos	Hora	Minutos
<input checked="" type="checkbox"/>	06-01-2014	<input type="text" value="7"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="30"/>
<input checked="" type="checkbox"/>	07-01-2014	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="15"/>	<input type="text" value="30"/>
<input checked="" type="checkbox"/>	08-01-2014	<input type="text" value="15"/>	<input type="text" value="0"/>	<input type="text" value="2"/>	<input type="text" value="40"/>
<input type="checkbox"/>	09-01-2014	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="checkbox"/>	10-01-2014	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
<input checked="" type="checkbox"/>	11-01-2014	<input type="text" value="7"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="30"/>
<input checked="" type="checkbox"/>	12-01-2014	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="15"/>	<input type="text" value="30"/>

Tabla 14: Interfaz de usuario asignar turno

Nombre Caso de Uso		Asignar turno	
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		

Asignación de Turnos

Código:

Empleados a seleccionar

	Código	Cedula	Nombres
<input checked="" type="checkbox"/>	1615	0509040744	MENDOZA CEDEÑO TERESA YOOONDA
<input checked="" type="checkbox"/>	1625	0903350896	HIDALGO HOLGUIN RUTH JACQUELINE
<input type="checkbox"/>	1630	0907924668	LOPEZ PALMA CECILIA ELIZABETH
<input checked="" type="checkbox"/>	1635	0910774520	DAMIAN ESTRADA MARIA AZUCENA
<input checked="" type="checkbox"/>	1640	1705200969	ZAMORA VELEZ AGUSTIN SALLUSTIANO
<input type="checkbox"/>	1645	0909413959	PEÑAFIEL MURILLO CECILIA DE LOURDES
<input checked="" type="checkbox"/>	1650	0909404818	GUINAPALLO NARANJO NELLY GRETA
<input type="checkbox"/>	1655	0909695059	ESPINOZA ESPINOZA SANDRA DE FATIMA
<input type="checkbox"/>	1660	0911428260	POLIT RIVAS MONICA DEL ROCIO
<input type="checkbox"/>	1665	0602433401	OCAÑA SALAS ROSA DE LOURDES

Tabla 15: Interfaz de usuario carga de marcaciones

Nombre Caso de Uso	Carga de marcaciones		
Creación	José Caicedo Salazar Julio Litardo Unda	Fecha	20/01/2014
<div data-bbox="268 672 1262 1356"> <p>Carga de archivo de Marcaciones</p> <p>Seleccionar Oficina: <input type="button" value="v"/></p> <p>Seleccionar archivo: <input type="button" value="+ Subir archivo:"/> <input type="text"/></p> <p>Resultados de la carga</p> <p>Archivo:</p> <p>Numero de Registros Procesados:</p> <p>Numero de Empleados Procesados:</p> <p>Numero de marcaciones descartadas:</p> <p><input type="button" value="Limpiar"/> <input type="button" value="Ver Reporte"/> <input type="button" value="Salir"/></p> </div>			

CAPÍTULO 4

4 ANÁLISIS Y DISEÑO PRELIMINAR

4.1 Descripción de los Casos de Uso

De acuerdo a la metodología Iconix, como parte integrante del Análisis y Diseño Preliminar, se debe realizar la descripción de los casos de uso identificados en el Análisis de Requisitos.

Para realizar dicha descripción se ha elaborado un formato de tabla, en donde consta el nombre del caso de uso, autores, fecha de creación, se especifica el actor principal, los actores secundarios, la precondición que se debe cumplir antes de iniciar el caso de uso, la postcondición que se debe cumplir al finalizar el caso de uso, el propósito, una descripción, el escenario primario (flujo normal) y los escenarios alternos (flujos alternativos) si los

hubiere de cada caso de uso. A continuación se presenta la descripción de cada caso de uso del sistema.

4.1.1 Descripción de Caso de Uso Crear Turno Normal

Tabla 16: Descripción Caso de Uso Crear Turno Normal

Nombre Caso Uso	Crear Turno Normal		
Creación	José Caicedo Salazar Julio Litardo Unda	Fecha	20/01/2014
Actor Principal			
Secretaria GTH			
Actores Secundarios			
Precondiciones			
La Secretaria GTH debe haber iniciado sesión en el sistema. Deben estar ingresados en el sistema los grupos de turnos y los días de la semana.			
Postcondiciones			
El nuevo registro del turno normal (fijo) queda almacenado en el sistema.			
Propósito/Función			
Registrar el turno normal (fijo) para ser asignado al personal.			
Descripción/Resumen			
La Secretaria GTH ingresa el turno normal (fijo) que puede o no tener horario de lunch. Al finalizar la transacción se registra el turno normal.			

Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando la Secretaria GTH ingresa la descripción del turno		
2	La Secretaria GTH selecciona el grupo del turno	3	Despliega la descripción del grupo del turno
4	La Secretaria GTH escoge el número de marcaciones a considerar para el turno: a. Si escoge 2 marcaciones, vea sección Generar turno para 2 marcaciones. b. Si escoge 4 marcaciones, vea sección Generar turno para 4 marcaciones.		
5	La Secretaria GTH selecciona los días del turno y registra las horas y minutos de entrada y salida de la empresa por cada día seleccionado. (Salida y retorno de lunch si es el caso).		
6	La Secretaria GTH da click en el botón Guardar	7	Muestra mensaje que los datos fueron guardados satisfactoriamente y registra el nuevo turno



Escenarios Alternos (Flujo alternativo)

Escenario Alternativo: Generar Turno para 2 marcaciones

Paso	Acción de los actores	Paso	Respuesta del sistema
4.a.1	La Secretaria GTH escoge 2 en número de marcaciones	4.a.2	Muestra la letra "N" en lunch, porque es un turno sin horario de lunch.

Escenario Alternativo: Generar turno para 4 marcaciones

Paso	Acción de los actores	Paso	Respuesta del sistema
4.b.1	La Secretaria GTH escoge 4 en número de marcaciones	4.b.2	Muestra la letra "S" en lunch, porque es un turno con horario de lunch. Se muestran las columnas de horas/minutos para la salida y retorno de lunch.

4.1.2 Descripción de Caso de Uso Modificar Turno Normal

Tabla 17: Descripción Caso de Uso Modificar Turno Normal

Nombre Caso Uso	Modificar Turno Normal		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actor Principal			
Secretaria GTH			
Actores Secundarios			
Precondiciones			
Secretaria GTH debe haber iniciado sesión en el sistema. El turno debe estar ingresado en el sistema y no debe estar asignado al personal.			
Postcondiciones			
Se actualiza el turno.			
Propósito/Función			
Actualiza el turno si no ha sido asignado al personal.			
Descripción/Resumen			
La Secretaria GTH ingresa o busca el turno al que va a cambiar los datos. Al finalizar la transacción se actualiza el turno.			

Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando la Secretaria GTH ingresa el código del turno o busca el turno a modificar	2	Presenta los datos del turno seleccionado
3	La Secretaria GTH cambia los datos de los días del turno y las horas y minutos de entrada, salida y retorno del lunch, y salida.		
4	La Secretaria GTH da click en el botón Guardar	5	Muestra mensaje que los datos fueron guardados satisfactoriamente y actualiza el turno.

4.1.3 Descripción de Caso de Uso Crear Turno Frecuencia Mensual

Tabla 18: Descripción Caso de Uso Crear Turno Frecuencia Mensual

Nombre Caso Uso	Crear Turno Frecuencia Mensual		
Creación	José Caicedo Salazar Julio Litardo Unda	Fecha	20/01/2014
Actor Principal			
Jefe de Área			
Actores Secundarios			
Precondiciones			
El Jefe de Área debe haber iniciado sesión en el sistema. Deben estar ingresados en el sistema los grupos de turnos.			
Postcondiciones			
El nuevo registro del turno con frecuencia de marcación y descanso mensual queda almacenado en el sistema.			
Propósito/Función			
Registrar el turno con frecuencia mensual para ser asignado al personal.			
Descripción/Resumen			
El Jefe de Área ingresa el turno con frecuencia mensual que puede o no tener horario de lunch. Al finalizar la transacción se registra el turno con frecuencia mensual.			

Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando el Jefe de Área ingresa la descripción del turno		
2	El Jefe de Área selecciona el grupo del turno	3	Despliega la descripción del grupo del turno
4	El Jefe de Área escoge el número de marcaciones a considerar para el turno: a. Si escoge 2 marcaciones, vea sección Generar turno para 2 marcaciones. b. Si escoge 4 marcaciones, vea sección Generar turno para 4 marcaciones		
5	El Jefe de Área selecciona la frecuencia de marcación (24 – 48 – 72 horas)	6	Muestra la frecuencia de marcación seleccionada.
7	El Jefe de Área selecciona la frecuencia de descanso (24 – 48 – 72 horas)	8	Muestra la frecuencia de descanso seleccionada.
9	El Jefe de Área da click en el campo fecha	10	Muestra un calendario con los días del mes actual.

Paso	Acción de los actores	Paso	Respuesta del sistema
11	El Jefe de Área selecciona la fecha de inicio, dentro del mes, a partir del cual se va a generar el turno con frecuencia mensual.		
12	El Jefe de Área selecciona las horas y minutos de entrada de la empresa y control de noche. (salida y retorno de lunch si es el caso)		
13	El Jefe de Área da click en el botón Generar	14	Genera y presenta los días del turno de acuerdo con la frecuencia de marcación y descanso mensual.
15	El Jefe de Área da click en el botón Guardar	16	Muestra mensaje que los datos fueron guardados satisfactoriamente y registra el nuevo turno con frecuencia de marcación y descanso mensual

Escenarios Alternos (Flujo alternativo)

Escenario Alterno: Generar Turno para 2 marcaciones

Paso	Acción de los actores	Paso	Respuesta del sistema
4.a.1	El Jefe de Área escoge 2 en número de marcaciones para el turno	4.a.2	Muestra la letra "N" en lunch, porque es un turno sin horario de lunch.

Escenario Alternativo: Generar turno para 4 marcaciones

Paso	Acción de los actores	Paso	Respuesta del sistema
4.b.1	El Jefe de Área escoge 4 en número de marcaciones para el turno	4.b.2	Muestra la letra "S" en lunch, porque es un turno con horario de lunch. Se muestran las columnas de horas/minutos para la salida y retorno de lunch.



4.1.4 Descripción de Caso de Uso Crear Turno Rotativo Mensual

Tabla 19: Descripción Caso de Uso Crear Turno Rotativo Mensual

Nombre Caso Uso	Crear Turno Rotativo Mensual		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actor Principal			
Jefe de Área			
Actores Secundarios			
Precondiciones			
El Jefe de Área debe haber iniciado sesión en el sistema. Deben estar ingresados en el sistema los grupos de turnos.			
Postcondiciones			
El nuevo registro del turno rotativo mensual queda almacenado en el sistema.			
Propósito/Función			
Registrar el turno rotativo mensual para ser asignado al personal.			
Descripción/Resumen			
El Jefe de Área ingresa el turno rotativo mensual que puede o no tener horario de lunch. Al finalizar la transacción se registra el turno.			

Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando el Jefe de Área ingresa la descripción del turno		
2	El Jefe de Área selecciona el grupo del turno	3	Despliega la descripción del grupo del turno
4	El Jefe de Área escoge el número de marcaciones a considerar para el turno: a. Si escoge 2 marcaciones, vea sección Generar turno para 2 marcaciones. b. Si escoge 4 marcaciones, vea sección Generar turno para 4 marcaciones		
5	El Jefe de Área da click en el campo fecha	6	Muestra un calendario con los días del mes actual.
7	El Jefe de Área selecciona la fecha de inicio, dentro del mes, a partir del cual se va a generar el turno rotativo mensual.		
8	El Jefe de Área da click en el botón Generar	9	Genera y presenta el turno rotativo mensual, desde la fecha de inicio seleccionada hasta el último día del mes de la fecha de inicio.

Paso	Acción de los actores	Paso	Respuesta del sistema
10	El Jefe de Área selecciona las fechas de marcación y las horas y minutos de entrada de la empresa y control de noche, (salida y retorno de lunch si es el caso) por cada fecha.		
11	El Jefe de Área da click en el botón Guardar	12	Muestra mensaje que los datos fueron guardados satisfactoriamente y registra el nuevo turno rotativo mensual

Escenarios Alternos (Flujo alternativo)

Escenario Alternativo: Generar Turno para 2 marcaciones

Paso	Acción de los actores	Paso	Respuesta del sistema
4.a.1	El Jefe de Área escoge generar el turno para 2 marcaciones	4.a.2	Muestra la letra "N" en lunch, porque es un turno sin horario de lunch.

Escenario Alternativo: Generar turno para 4 marcaciones

Paso	Acción de los actores	Paso	Respuesta del sistema
4.b.1	El Jefe de Área escoge generar el turno para 4 marcaciones	4.b.2	Muestra la letra "S" en lunch, porque es un turno con horario de lunch.

4.1.5 Descripción de Caso de Uso Asignar Turno

Tabla 20: Descripción Caso de Uso Asignar Turno

Nombre Caso Uso	Asignar Turno		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actor Principal			
Secretaria GTH, Jefe de Área			
Actores Secundarios			
Precondiciones			
La Secretaria GTH o el Jefe de Área deben haber iniciado sesión en el sistema. El turno que se va a asignar debe estar ingresado en el sistema y activo.			
Postcondiciones			
Se registra la asignación del turno al empleado o a los empleados.			
Propósito/Función			
Asigna el turno activo al empleado o a los empleados.			
Descripción/Resumen			
La Secretaria GTH o el Jefe de Área buscan y asignan el turno a cada empleado. Al finalizar la transacción se registra el turno al empleado o a los empleados.			

Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando la Secretaria GTH o el Jefe Área da click en el icono de búsqueda para buscar el turno a asignar	2	Presenta todos los turnos activos ingresados en el sistema
3	La Secretaria GTH/Jefe Área selecciona el turno a asignar	4	Presenta el número y la descripción del turno seleccionado
5	La Secretaria GTH o el Jefe de Área selecciona el empleado o los empleados para asignar el turno.		
6	La Secretaria GTH o el Jefe de Área da click en el botón Guardar	7	Muestra mensaje que el registro fue grabado satisfactoriamente y registra el turno signado al(los) empleado(s)

4.1.6 Descripción de Caso de Uso Cargar Marcaciones

Tabla 21: Descripción Caso de Uso Cargar Marcaciones

Nombre Caso Uso	Cargar Marcaciones		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actor Principal			
Secretaria GTH			
Actores Secundarios			
Precondiciones			
La Secretaria GTH debe haber iniciado sesión en el sistema. El archivo texto de las marcaciones debe haber sido descargado del reloj biométrico.			
Postcondiciones			
Se registran las marcaciones de los empleados.			
Propósito/Función			
Registrar las marcaciones de los empleados del archivo texto generado por el reloj biométrico.			
Descripción/Resumen			
La Secretaria GTH selecciona el archivo texto con las marcaciones de los empleados. Al finalizar la transacción se registran las marcaciones de los empleados.			

Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando la Secretaria GTH selecciona la oficina o sucursal de la cual se va a cargar las marcaciones	2	Presenta la descripción de la oficina o sucursal
3	La Secretaria GTH selecciona el archivo texto con las marcaciones de los empleados a ser cargadas.	4	Carga el archivo texto y los nuevos registros de las marcaciones quedan almacenados en el sistema
		5	Muestra los resultados de la carga del archivo texto: número de registros procesados, número de empleados procesados y número de marcaciones descartadas.

4.1.7 Descripción de Caso de Uso Ingresar Solicitud

Tabla 22: Descripción Caso de Uso Ingresar Solicitud

Nombre Caso Uso	Ingresar Solicitud		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actor Principal			
Secretaria GTH			
Actores Secundarios			
Precondiciones			
La Secretaria GTH debe haber iniciado sesión en el sistema. Debe haber sido registrado el empleado al cual se le va a ingresar la solicitud. Deben estar registrados los tipos de novedad y las novedades. El empleado debe tener asignado un turno activo. Debe haber sido cargado el archivo texto de marcaciones para el caso en que se va a afectar marcaciones pasadas.			
Postcondiciones			
Se registra la solicitud de novedad del empleado.			
Propósito/Función			
Registrar la solicitud de novedad del empleado que puede ser por permisos, vacaciones, capacitación u otros.			
Descripción/Resumen			
La Secretaria GTH ingresa la solicitud de novedad especificando el tipo y motivo de la misma (permiso, vacaciones, capacitación u otros), la duración en días, la fecha de inicio y fin y una observación. Al finalizar la transacción se registra la solicitud, pero no afecta marcaciones hasta no ser aprobada.			



Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando la Secretaria GTH da click en el ícono de búsqueda para buscar el empleado al que se va a ingresar la solicitud.	2	Presenta la lista de todos los empleados activos ingresados en el sistema
3	La Secretaria GTH selecciona el empleado	4	Presenta los nombres del empleado y el turno activo del empleado
5	La Secretaria GTH selecciona el tipo de novedad.	6	Presenta la descripción del tipo de novedad seleccionada
7	La Secretaria GTH selecciona la novedad.	8	Presenta la descripción de la novedad seleccionada
9	La Secretaria GTH selecciona la duración en días de la novedad (un día o varios días). a. Si escoge un día, vea sección Generar novedad para 1 día. b. Si escoge varios días, vea sección Generar novedad para varios días.		

Paso	Acción de los actores	Paso	Respuesta del sistema
10	La Secretaria GTH selecciona a que marcaciones se va a afectar con la novedad, entrada y salida de la empresa (salida y retorno de lunch si es el caso).		
11	La Secretaria GTH ingresa una observación de la solicitud.		
12	La Secretaria GTH da click en el botón Guardar	13	Muestra mensaje que los datos fueron guardados satisfactoriamente y registra la solicitud de la novedad.

Escenarios Alternos (Flujo alternativo)

Escenario Alterno: Generar novedad para 1 día

Paso	Acción de los actores	Paso	Respuesta del sistema
9.a.1	La Secretaria GTH escoge 1 día en días de novedad	9.a.2	Muestra sólo el campo fecha de inicio
9.a.3	La Secretaria GTH da click en el campo fecha inicio	9.a.4	Muestra un calendario con los días del mes actual.
9.a.5	La Secretaria GTH selecciona la fecha de inicio de la solicitud		

Escenario Alternativo: Generar novedad para varios días

Paso	Acción de los actores	Paso	Respuesta del sistema
9.b.1	La Secretaria GTH escoge varios días en días de novedad	9.b.2	Muestra los campos fecha de inicio y fecha final.
9.b.3	La Secretaria GTH da click en el campo fecha inicio	9.b.4	Muestra un calendario con los días del mes actual.
9.b.5	La Secretaria GTH selecciona la fecha de inicio de la solicitud		
9.b.6	La Secretaria GTH da click en el campo fecha final	9.b.7	Muestra un calendario con los días del mes actual.
9.b.8	La Secretaria GTH selecciona la fecha final de la solicitud	9.b.9	Valida la fecha final contra la fecha inicio

4.1.8 Descripción de Caso de Uso Aprobar Solicitud

Tabla 23: Descripción Caso de Uso Aprobar Solicitud

Nombre Caso Uso	Aprobar Solicitud		
Creación	José Caicedo Salazar	Fecha	20/01/2014
	Julio Litardo Unda		
Actor Principal			
Analista GTH			
Actores Secundarios			
Precondiciones			
El Analista GTH debe haber iniciado sesión en el sistema. La solicitud que se va a aprobar debe estar ingresada en el sistema y con estado de ingresada "I".			

Postcondiciones
Se registra con estado de aprobada "A" la solicitud de novedad del empleado.
Propósito/Función
Registrar el estado de aprobada de la solicitud de novedad del empleado.
Descripción/Resumen
El Analista GTH busca y aprueba la solicitud de novedad. Al finalizar la transacción se registra el estado de aprobada de la solicitud, y se afecta a las marcaciones.

Escenario primario (Flujo normal)

Curso normal de los eventos

Paso	Acción de los actores	Paso	Respuesta del sistema
1	El caso de uso inicia cuando el Analista GTH da click en el ícono de búsqueda para buscar la solicitud a aprobar.	2	Presenta todos las solicitudes con el estado de ingresadas en el sistema ordenadas por número de solicitud.
3	El Analista GTH selecciona la solicitud.	4	Presenta la información de la solicitud seleccionada.
5	El Analista GTH da click en el botón Aprobar.	6	Muestra mensaje que los datos fueron guardados satisfactoriamente y registra la solicitud con el estado de aprobada.

4.2 Diagrama de Robustez

En la metodología Iconix se utiliza el diagrama de robustez para mostrar de forma gráfica las interacciones de los objetos que interviene en un caso de uso. A través del análisis de la descripción de cada caso de uso, el diagrama de robustez permite identificar un conjunto inicial de objetos participantes de cada caso de uso. A continuación se presenta los diagramas de robustez correspondientes a los principales casos de uso identificados del sistema.

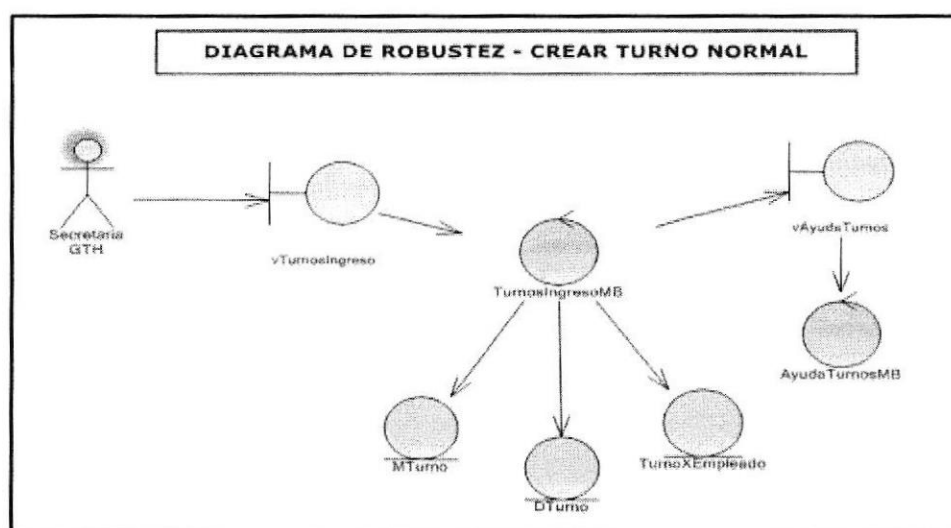


Figura 4.1: Diagrama de Robustez Crear Turno Normal



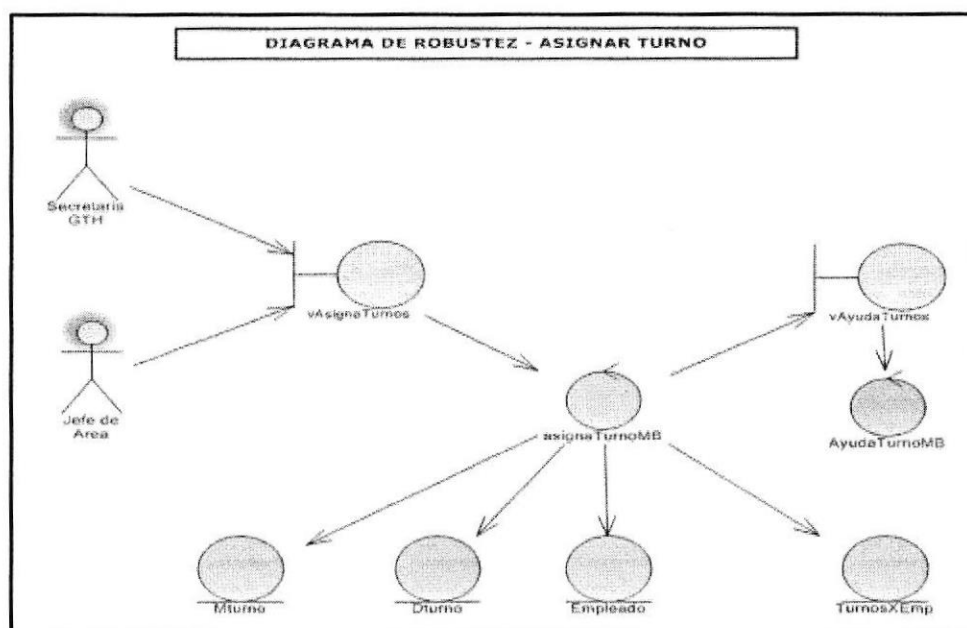


Figura 4.2: Diagrama de Robustez Asignar Turno

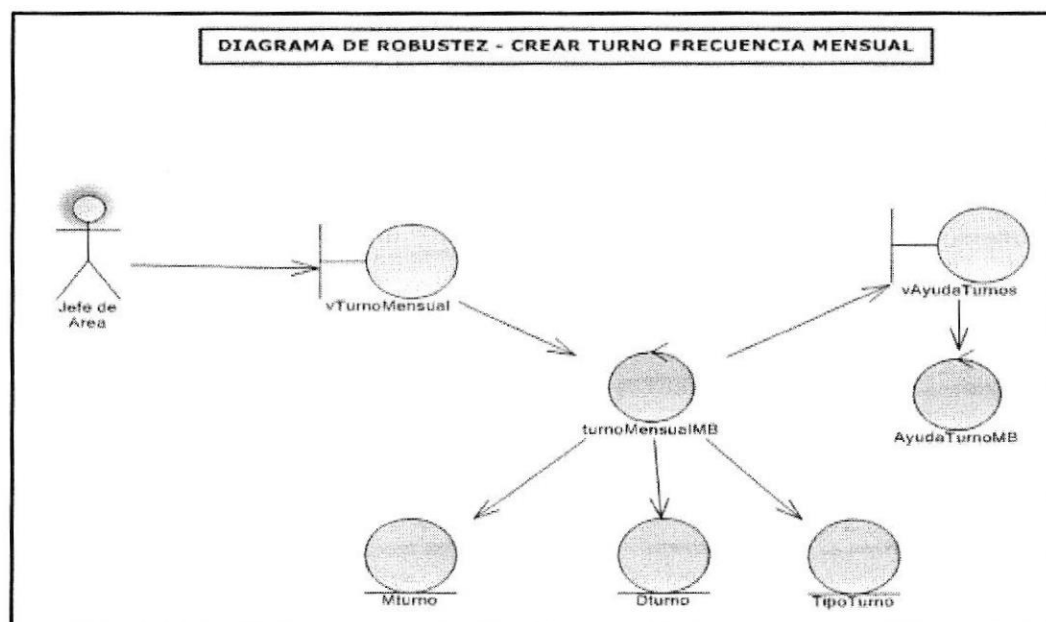


Figura 4.3: Diagrama de Robustez Crear Turno Frecuencia Mensual

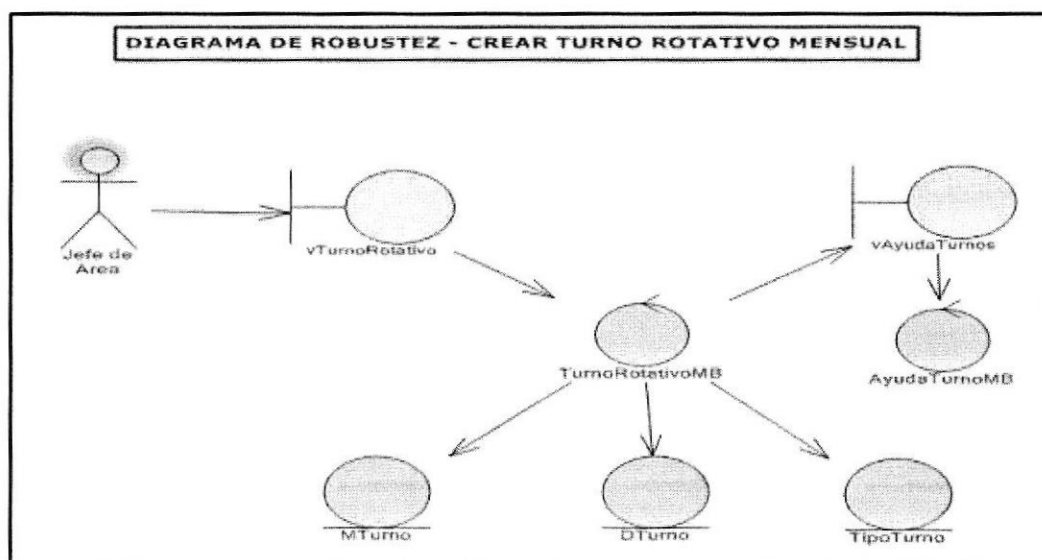


Figura 4.4: Diagrama de Robustez Crear Turno Rotativo Mensual

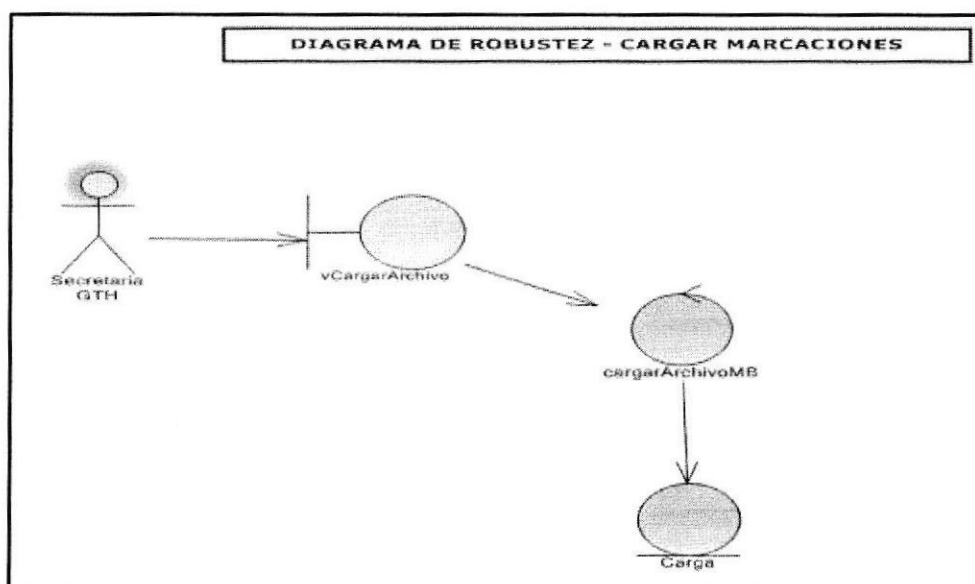


Figura 4.5: Diagrama de Robustez Cargar Marcaciones



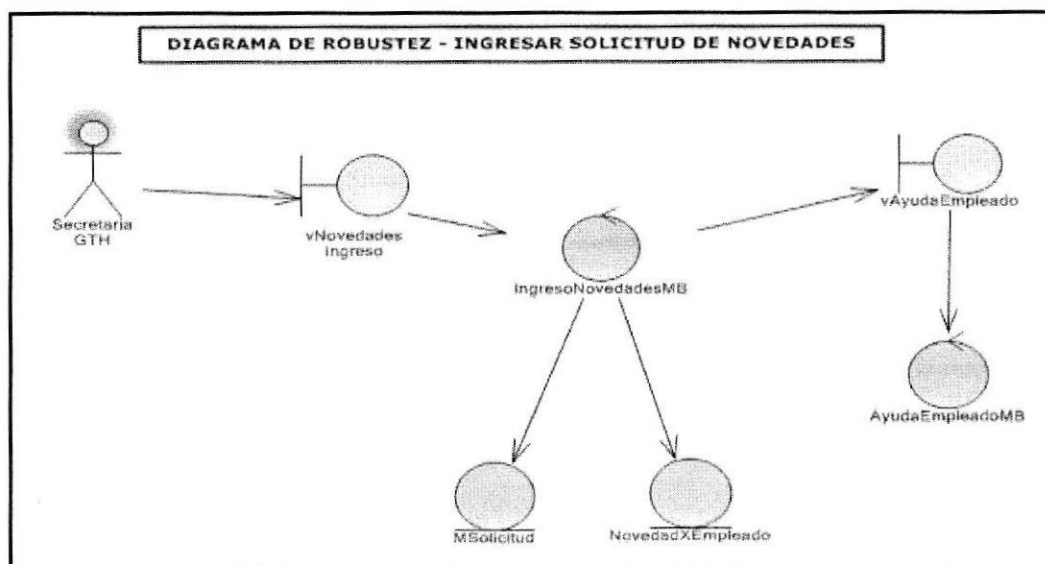


Figura 4.6: Diagrama de Robustez Ingresar Solicitud

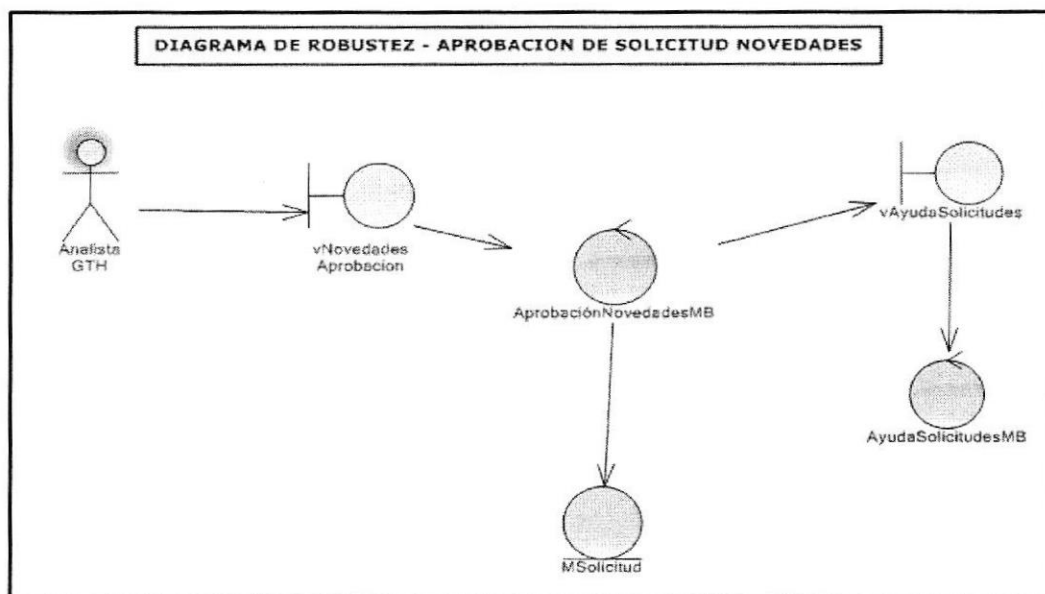


Figura 4.7: Diagrama de Robustez Aprobar Solicitud

4.3 Diagramas de Clases (Modelo Conceptual)

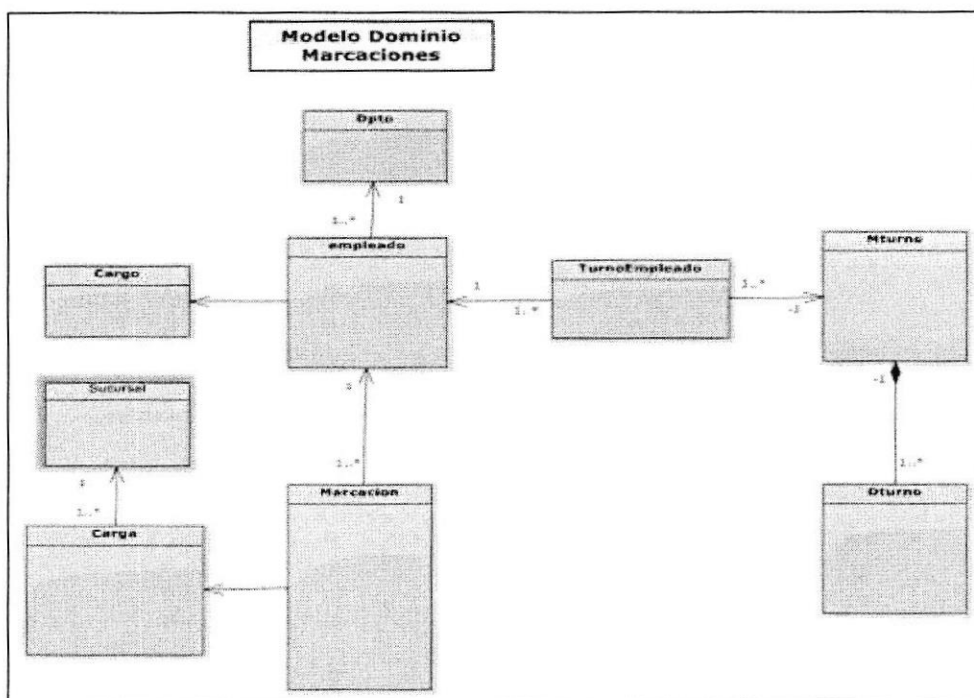


Figura 4.8: Modelo de Dominio Gestión de Marcaciones

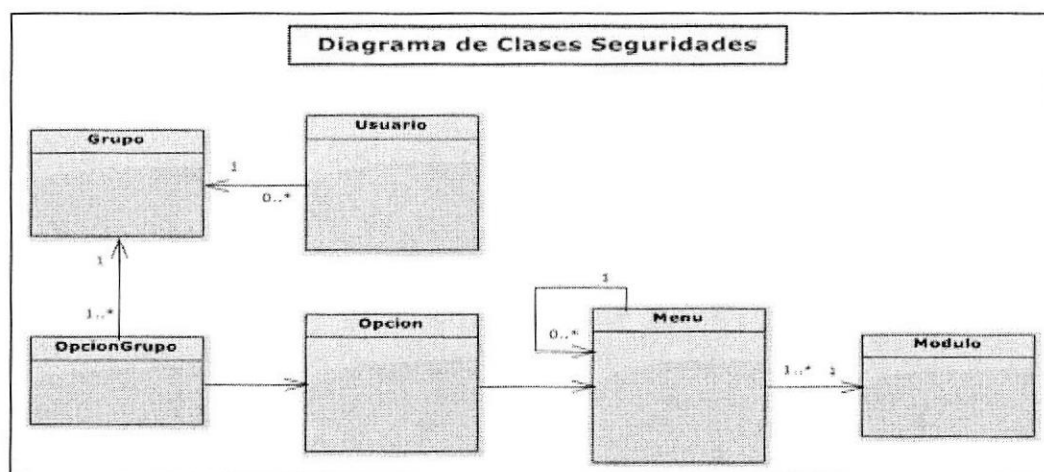


Figura 4.9: Modelo de Dominio Gestión de Seguridades

CAPÍTULO 5

5 DISEÑO DETALLADO

5.1 Entorno de software

La arquitectura tecnológica empleada para el desarrollo del **Sistema de Información Web Transaccional de control de turnos, asistencia y solicitudes de novedades de personal**, es la arquitectura de 3 capas.

La arquitectura de tres capas, es la arquitectura común de la mayoría de los sistemas de información web, que abarca una interfaz para el usuario y el almacenamiento de datos persistentes. Esta arquitectura plantea tres capas verticales: Presentación (Interfaz de usuario), Lógica de Aplicaciones (tareas y reglas que

rigen el proceso de negocio) y Almacenamiento (medio de almacenamiento persistente).

La tecnología que se usa para desarrollar e implementar el sistema es:

Tabla 24: Tecnología de Implementación

Tecnología	Descripción
Servidor Web	Apache Tomcat 7.0.34
Base de Datos	MySQL Server 5.5
Administrador de Base de Datos	Navicat 8 Lite para MySQL
Driver JDBC para MySQL	mysql-connector-java-5.1.18-bin.jar
Sistema Operativo	Linux Centos 6.4
IDE de Desarrollo	Netbeans 7.3.1
Máquina Virtual de Java	JVM 1.7.0
ORM	Java Persistence API con implementación de Hibernate
Diseñador de Reportes	iReport-5.1.0
Librería para compilar reportes	Jasper Report
Lenguaje de Programación Orientado a Objetos	Java
Contenido Web Dinámico	XHTML
Lenguaje de programación para la validación de formularios	JavaScript
Hojas de estilo	CSS

5.1 Diagramas de secuencias

Tabla 25: Diagrama de Secuencias Crear Turnos

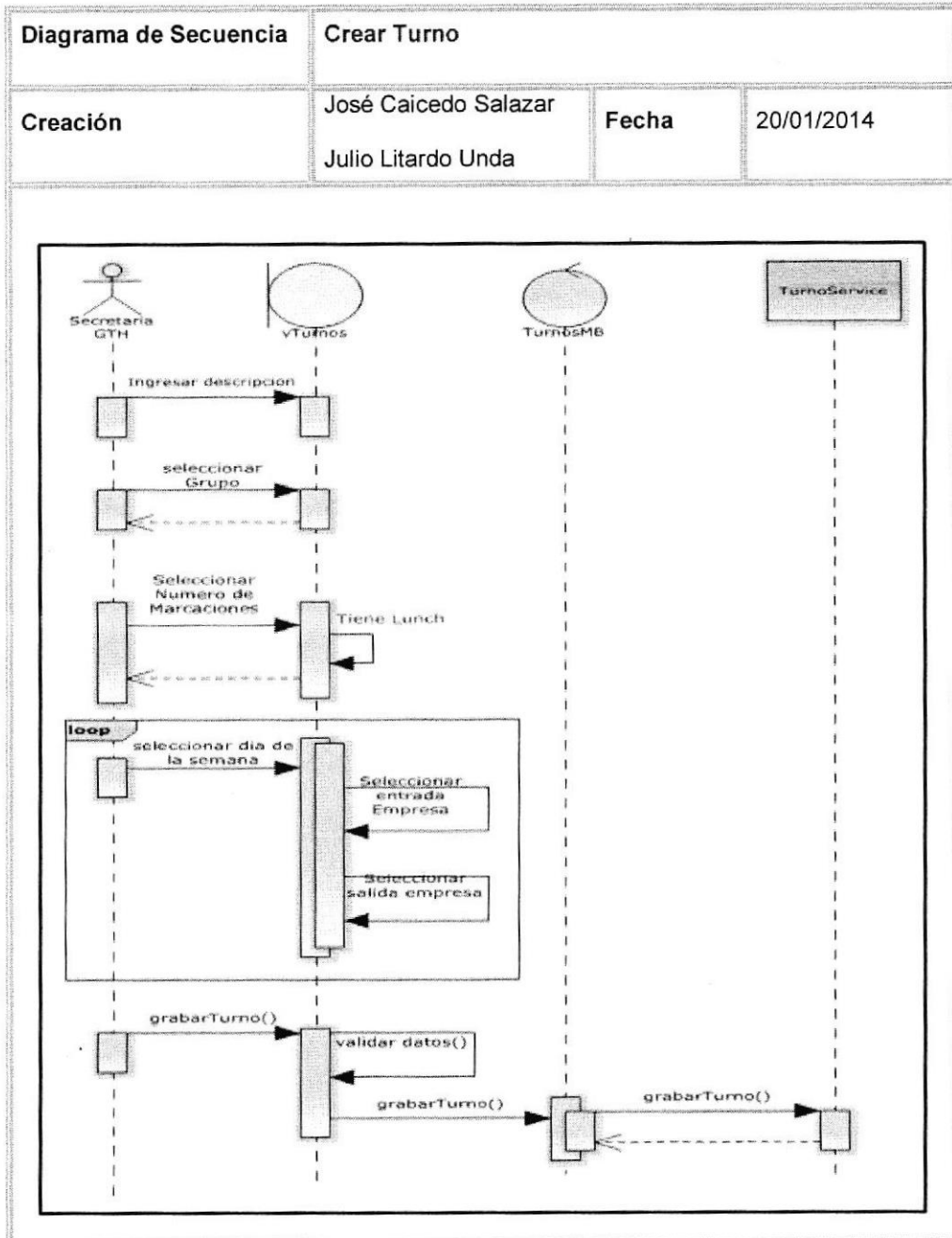


Tabla 26: Diagrama de Secuencias Carga de Marcaciones

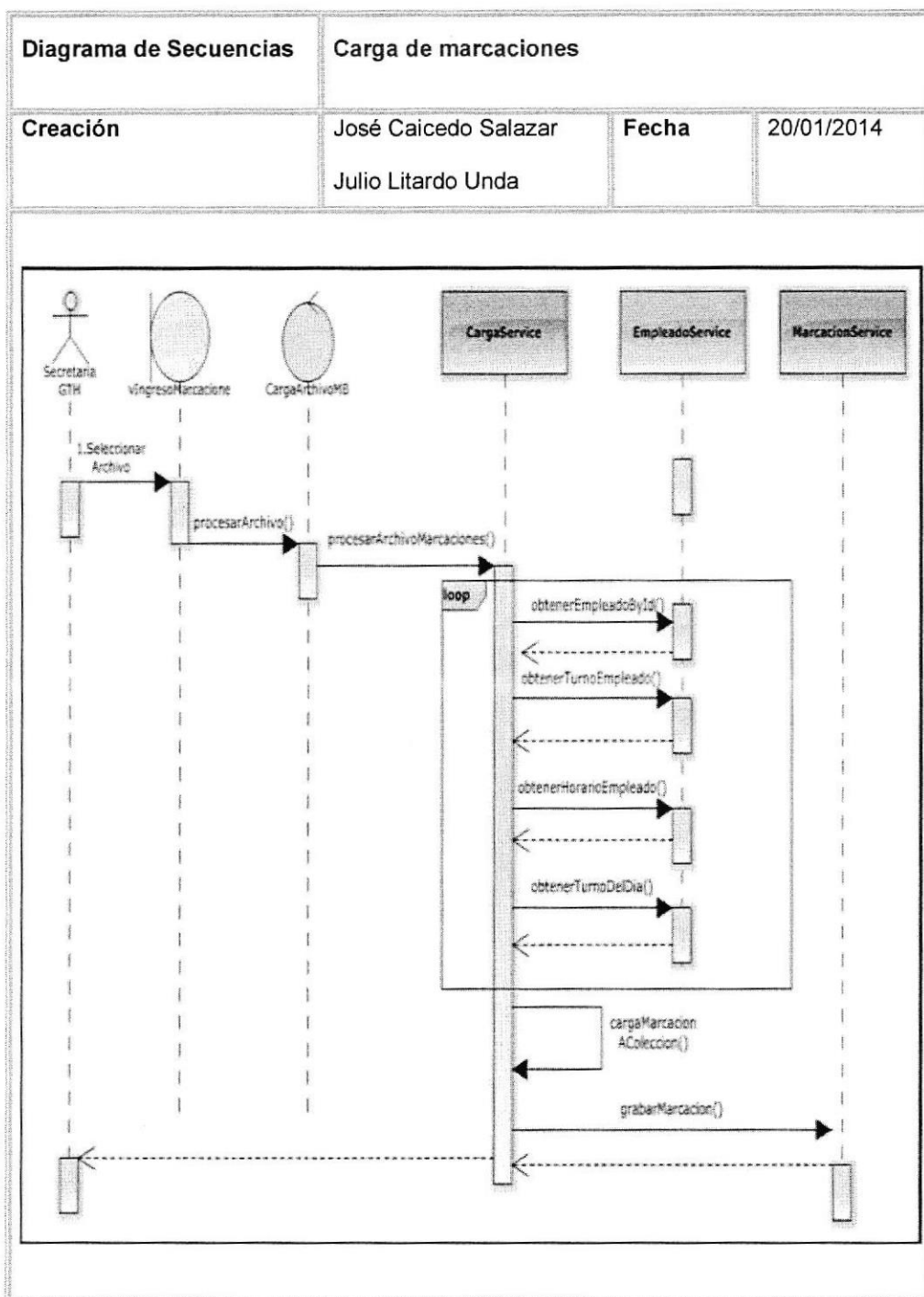


Tabla 27: Diagrama de Secuencias Asignar Turno

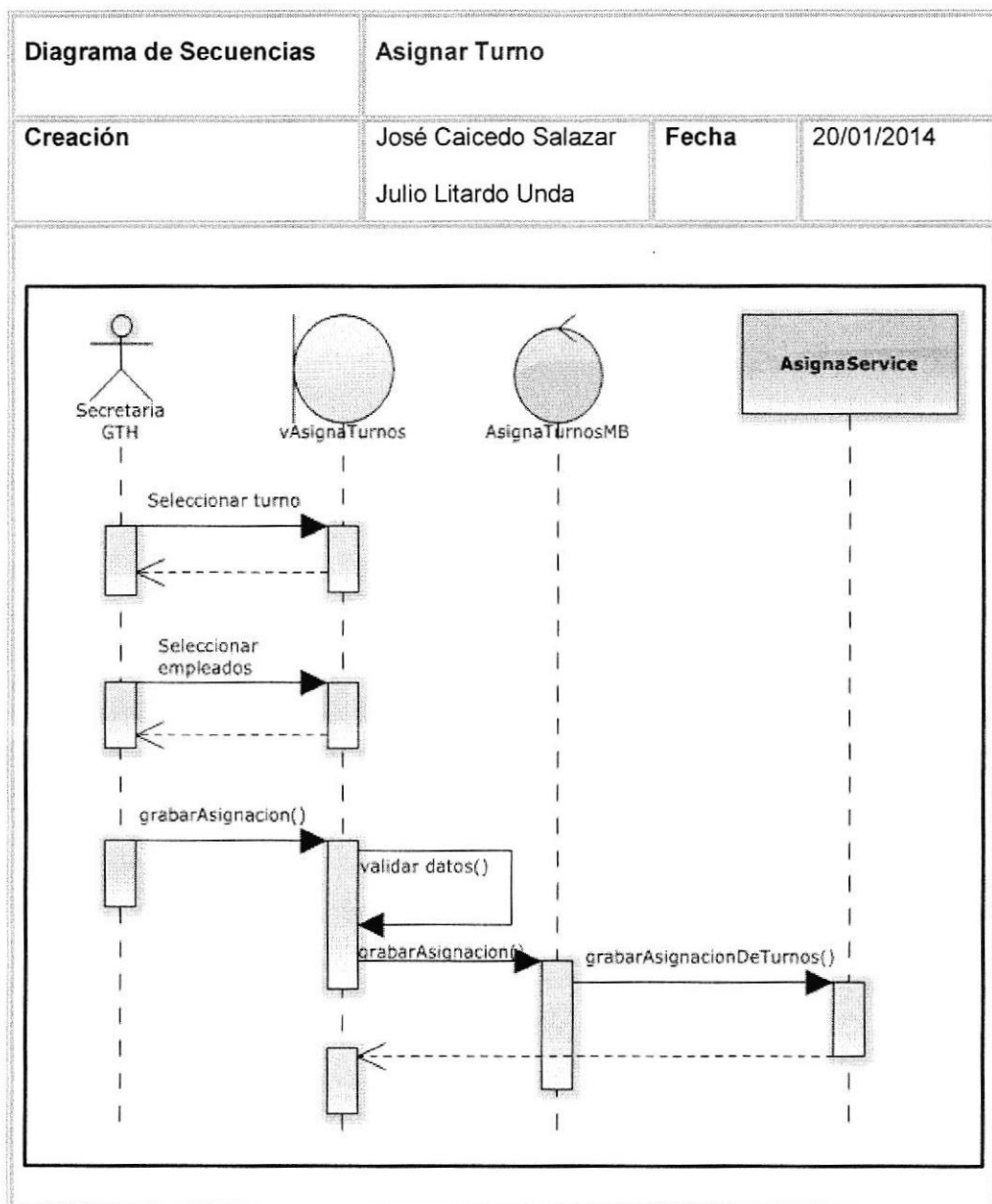
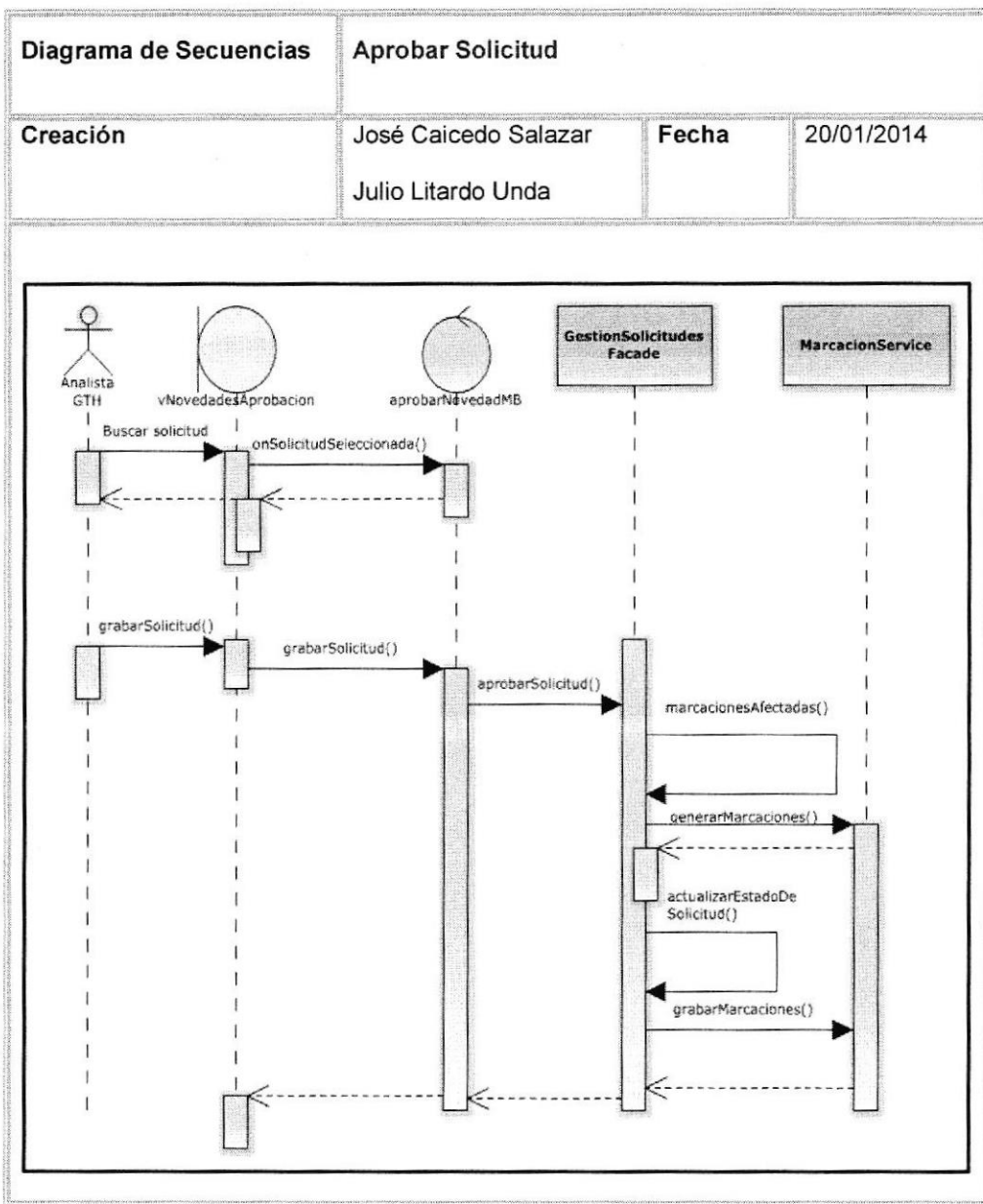


Tabla 28: Diagrama de Secuencias Aprobar Solicitud



5.2 Diagramas de Clases (Diseño)

Tabla 29: Diagrama de Clases Gestión de Marcaciones

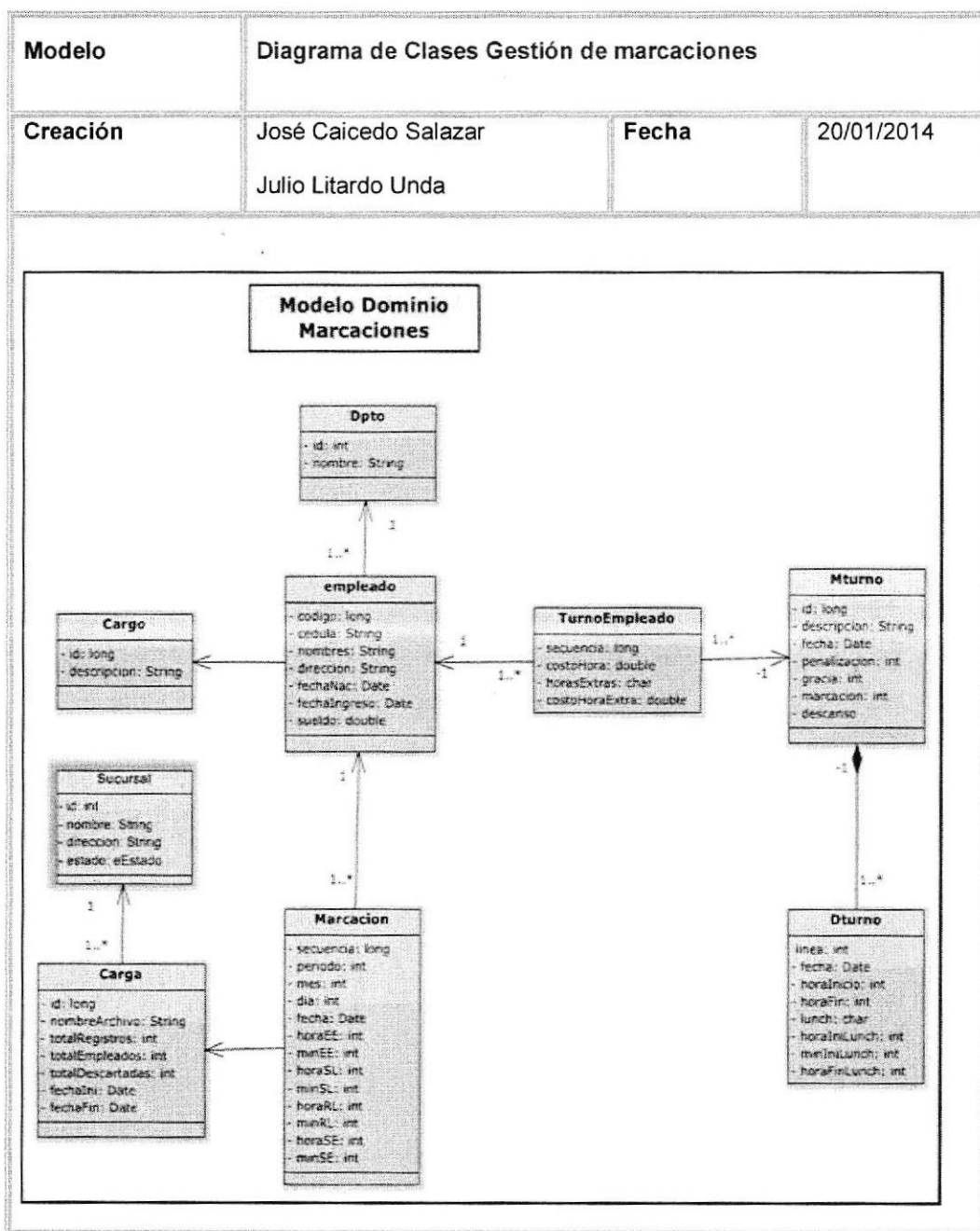


Tabla 30: Diagrama de Clases Gestión de Solicitudes

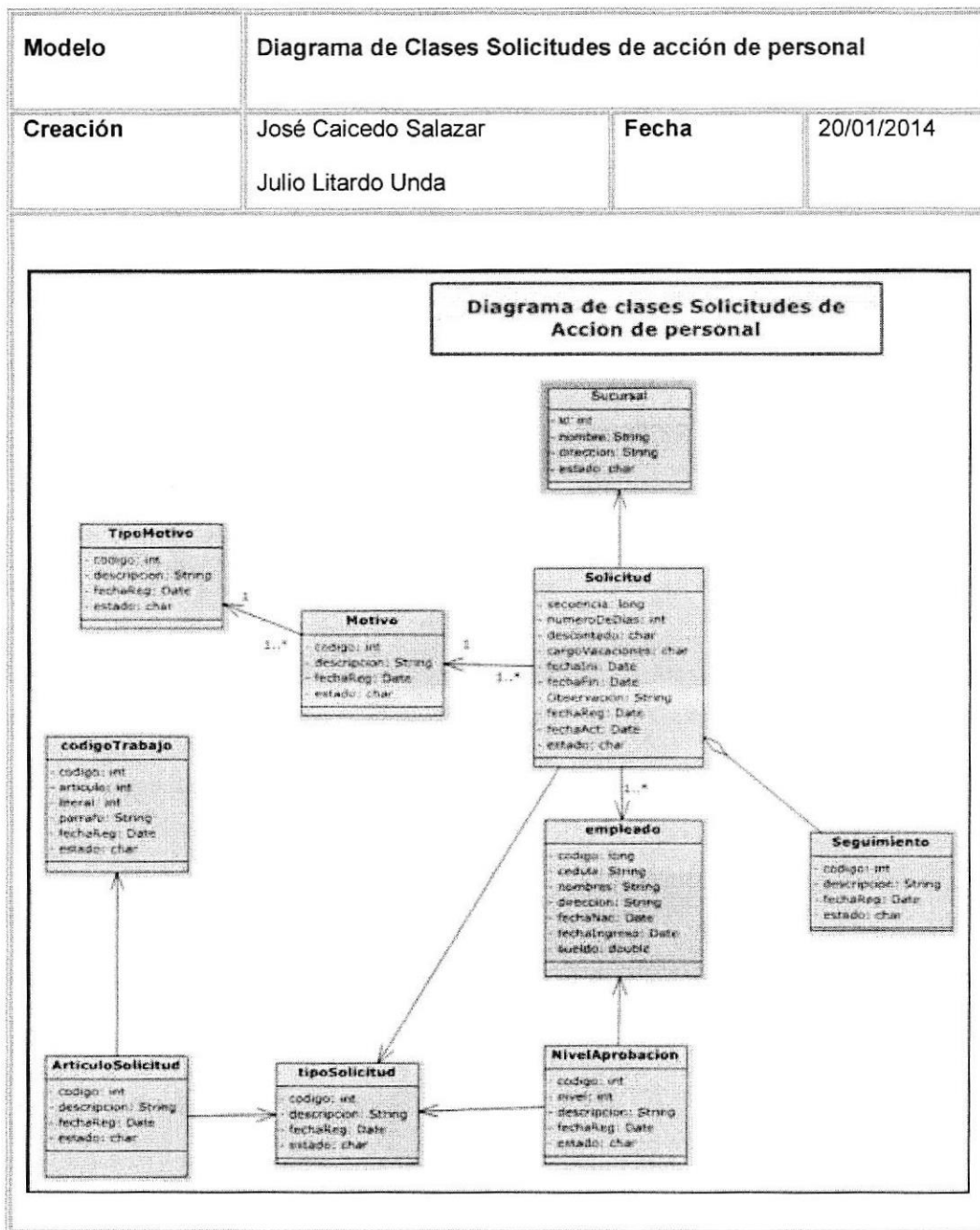
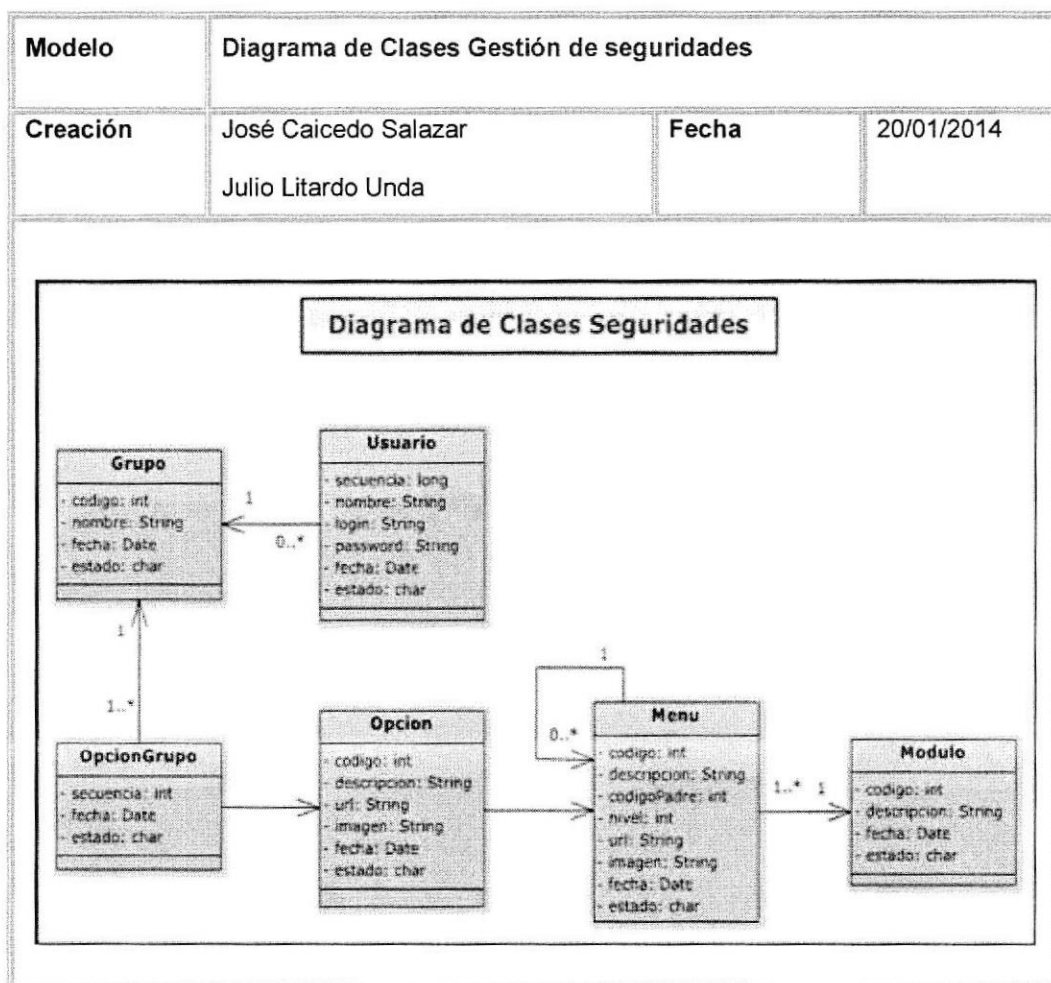


Tabla 31: Diagrama de Clases Gestión de Seguridades



5.3 Esquema de base de datos

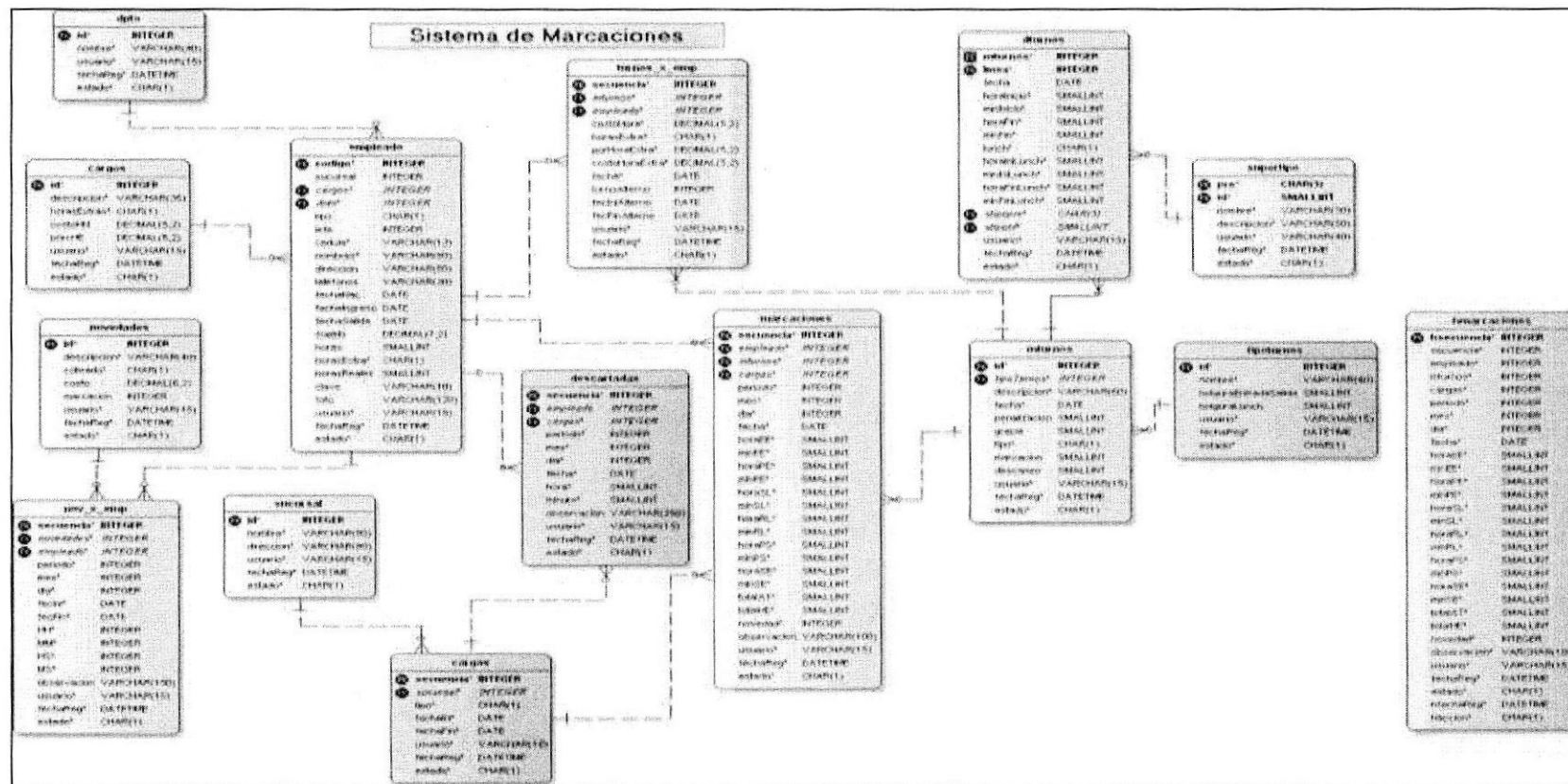


Figura 5.1: Diagrama Entidad – Relación

5.4 Diccionario de datos

Lista de Tablas

Tabla 32: Lista de Tablas

Tablas	
Nombre	Descripción
dpto	Guarda información de los departamentos de la empresa
cargos	Guarda información de la clasificación de cargos existentes en la empresa
empleado	Guarda información de los empleados de la empresa
sucursal	Guarda información de las sucursales de la empresa
cargas	Guarda información de los archivos textos de las marcaciones cuando son cargadas al sistema
marcaciones	Guarda información de las marcaciones de entrada y salida de los empleados
descartadas	Guarda información de las marcaciones que fueron descartadas al subir el archivo texto de marcaciones
mtornos	Tabla cabecera de turnos que guarda información general de los turnos establecidos para los empleados
dturnos	Guarda el detalle de los turnos de trabajo con hora de inicio y fin de la jornada de los empleados.
turnos_x_emp	Guarda los turnos asignados por empleado
tipoTurnos	Guarda la clasificación de turnos de la empresa
superTipo	Tabla general para clasificar los diferentes tipos de



	clasificaciones manejados en el sistema como: días de la semana, colores, etc.
hmarcaciones	Guarda un histórico de las marcaciones que son actualizadas por el proceso de novedades
motivos	Guarda información de la subclasificación de los tipos de motivos para solicitudes de novedades
tiposMotivos	Guarda información de la clasificación principal de los diferentes tipos de motivos para la solicitudes de novedades (permisos ocasionales y acción de personal)
msolicitudes	Tabla maestra que guarda información de las solicitudes de permisos ocasionales y de acciones del personal
grupos	Guarda los grupos bajo los cuales están agrupados los usuarios que van a acceder al sistema de acuerdo a su perfil
usuarios	Guarda información de los usuarios que pueden acceder al sistema
menu	Guarda la descripción del menú principal del sistema
modulos	Guarda la descripción de los módulos que conforman el sistema
opciones	Guarda la descripción de las opciones de cada menú del sistema
opciones_x_grupo	Guarda la relación entre opciones – grupo para habilitar las opciones del sistema al usuario de acuerdo a su perfil

El detalle de las tablas con los respectivos campos se encuentra especificado en el Anexo 1.

5.5 Seguridad

Para el desarrollo del sistema se ha considerado la seguridad en los accesos, como una de las principales características del sistema. Cada usuario debe digitar su contraseña para el ingreso al sistema, para su respectiva autenticación de la identidad y del perfil bajo el cual podrá visualizar las opciones que puede operar en el sistema de acuerdo a los permisos establecidos que tenga el usuario.

Al tratarse de un sistema web se debe considerar la seguridad en: el cliente, en el servidor, en la aplicación y en la comunicación. A continuación se mencionan ciertos aspectos a considerar para la seguridad del sistema:

- Acceso a la aplicación será mediante: Nombre de usuario, password y rol utilizando autenticación basada en la aplicación.

- Una vez autenticado, al usuario se le asigna un identificador de sesión. Este identificador acompañará invisiblemente a cada petición del usuario, con lo cual se garantizará que la petición proviene de un usuario previamente autenticado.
- Toda la documentación estará ubicada en un único repositorio centralizado pudiendo ser gestionada por los Administradores del sistema.
- Al estar toda la documentación en un único repositorio centralizado es posible hacer regularmente copias de seguridad de toda la documentación de todos los usuarios.
- Se deberá establecer las medidas de seguridad (cortafuegos) necesarias para preservar la integridad de la red interna.
- Revisar periódicamente los ficheros de log (access_log y error_log en Apache) para detectar posibles ataques al servidor.
- Restringir el acceso remoto al servidor de base de datos y almacenar los datos sensibles de forma encriptada como el caso de las contraseñas.

CAPÍTULO 6

6 IMPLEMENTACIÓN

Se especifican los estándares de codificación utilizados para la construcción del sistema, el tipo de codificación que fue necesaria usar para asegurar una adecuada manipulación y gestión del código fuente del sistema en cada uno de sus respectivos módulos.

Adicional se presentan los diagramas de paquetes, componentes y despliegue para contemplar la estructura del sistema y los componentes a utilizar para su implementación.

6.1 Estándares de codificación

El manejo de estándares de codificación resulta muy importante para el desarrollo de la aplicación de software porque facilitan el

entendimiento del código, lo que permite su reutilización y futuro mantenimiento.

“El correcto desarrollo de una aplicación de software profesional, requiere mantener una estandarización tanto en las estructuras como en los tipos de datos a lo largo del programa”. Esto se realiza con el objetivo de lograr una rápida ubicación y comprensión de los desarrolladores del módulo que están construyendo.

Con la finalidad de facilitar el mantenimiento de la aplicación de software y la adecuada organización y limpieza de su código fuente se ha utilizado las nomenclaturas Pascal Casing y Camel Casing para los nombres de los elementos.

Pascal Casing: La nomenclatura puede estar compuesta por tantas palabras como sean necesarias. La primera letra de cada una de las palabras irá siempre en mayúsculas.



Camel Casing: La nomenclatura puede estar compuesta por tantas palabras como sean necesarias. La primera palabra inicia con minúscula y las siguientes palabras inician con mayúscula.

Tabla 33: Estándares de Codificación

Elemento	Descripción	Ejemplo
Clase o Interfaz	Nomenclatura Pascal Casing Nombre debe ser un sustantivo Nombre simple y descriptivo Si es compuesto cada palabra inicia con mayúscula	<code>public class Empleado</code> <code>public interface IAsignaService</code>
Métodos	Nomenclatura Camel Casing Nombre debe ser un verbo Si el nombre es simple se escribe en minúscula Si es compuesto, la primera palabra inicia con minúscula y las siguientes palabras inician con mayúscula	<code>public void</code> <code>insertarTurnoPorEmpleado();</code>
Variables	Nomenclatura Camel Casing Nombre corto y significativo Si el nombre es simple se escribe en minúscula Si es compuesto, la primera palabra inicia con minúscula y las siguientes palabras inician con mayúscula No deben iniciar con "_", ni "\$"	<code>private String ruta;</code> <code>private String nombreArchivo;</code>
Constantes	Nomenclatura Húngara	<code>private static final String</code>

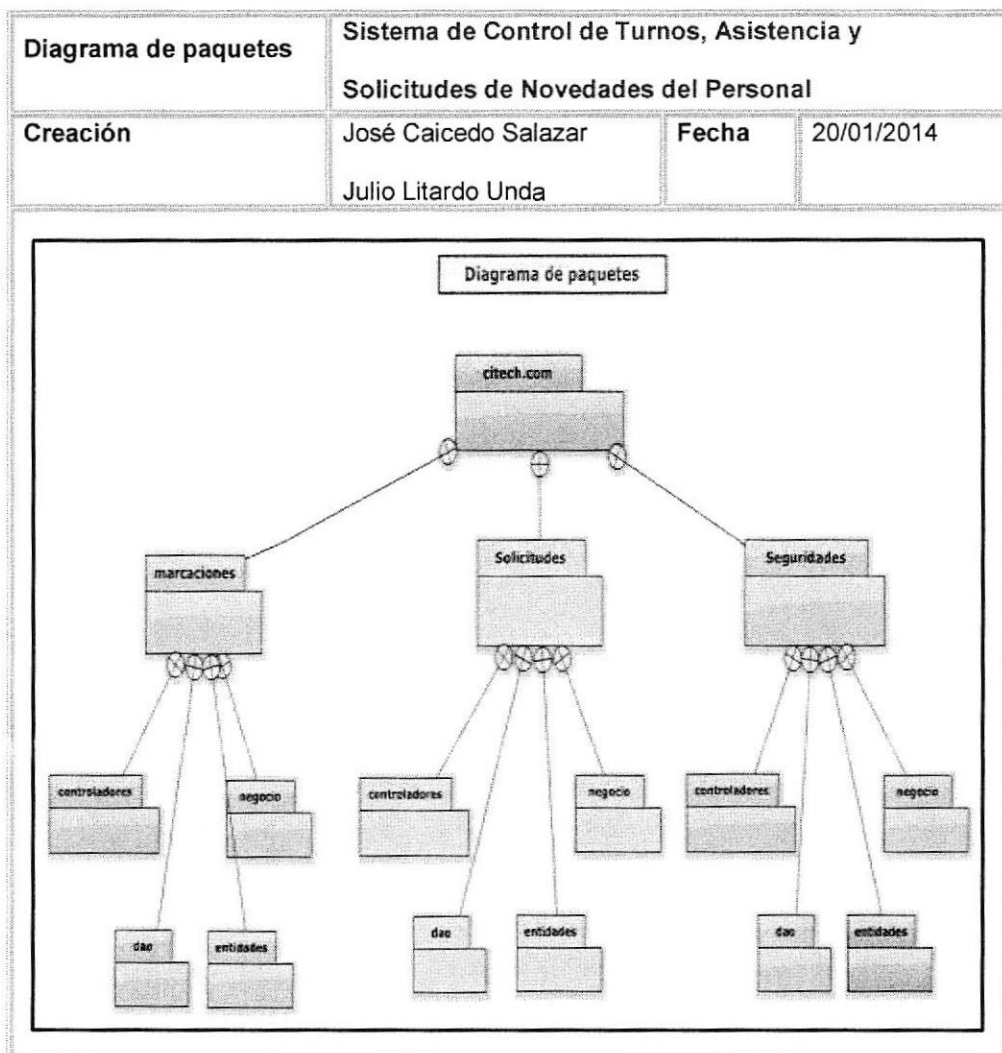
	Se escriben en mayúscula Si es compuesta las palabras que la componen se separan con " " - nombre corto y con significado	DELIMITADOR = ",";
--	--	--------------------

Además es muy útil el empleo de comentarios que expliquen el objetivo del código y esclarezcan el significado de aquellos aspectos del código fuente que puedan resultar dudosos. Empleo de comentarios en todas las declaraciones de clases y funciones más complejas.

6.2 Diagrama de paquetes

El diagrama de paquetes muestra la descomposición jerárquica del sistema en unidades lógicas y sus dependencias. A continuación se presenta el diagrama de paquetes del sistema:

Tabla 34: Diagrama de Paquetes



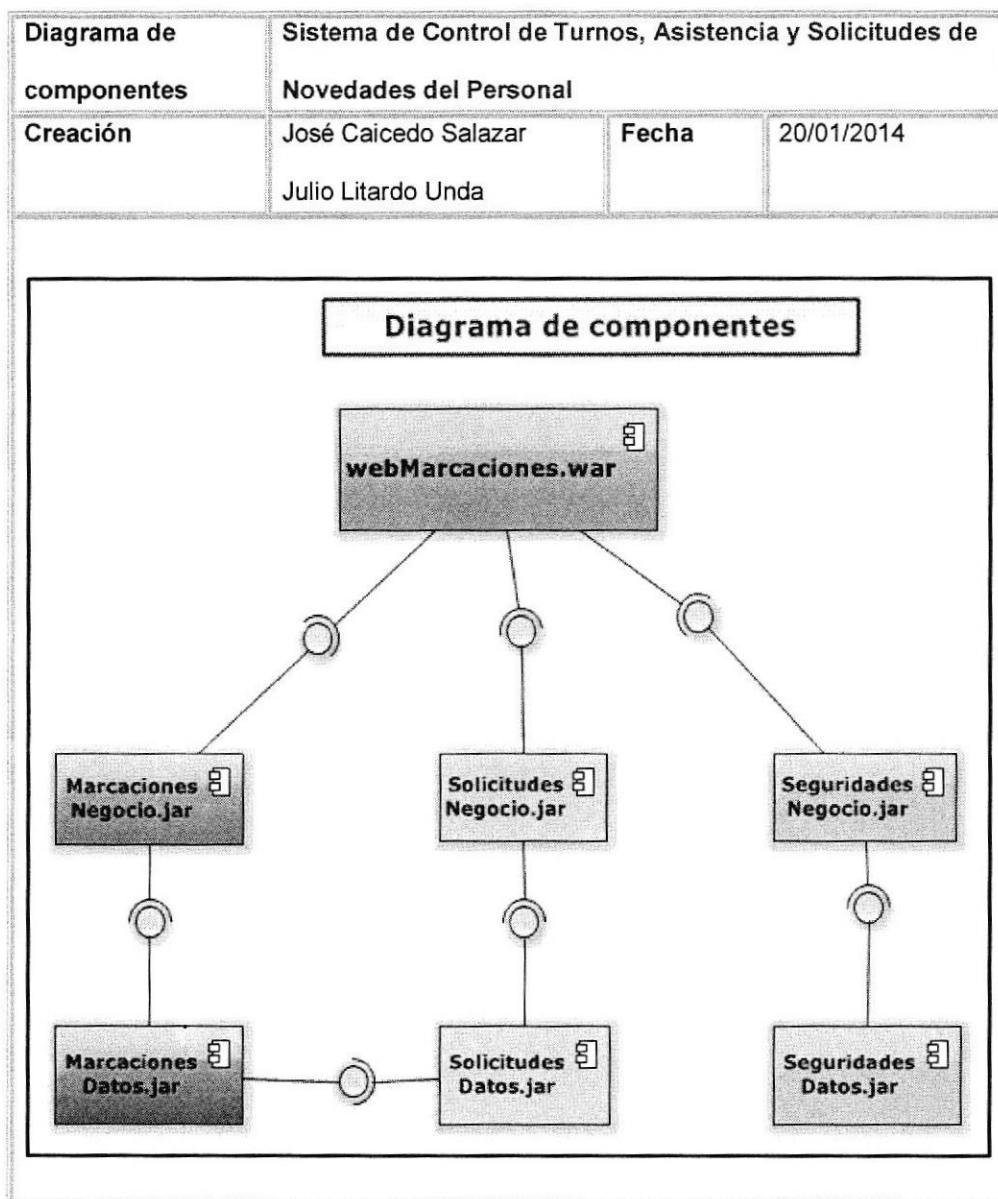
6.3 Diagrama de componentes

Se ha considerado conveniente incorporar el diagrama de componentes porque muestra el sistema dividido en componentes y las relaciones entre estos componentes.

Estos componentes representan los elementos de software necesario para la elaboración de un sistema como archivos simples, librerías, paquetes, ejecutables, etc. Para el caso del sistema de Control de Turnos, Asistencia y Solicitudes de Novedades del Personal el diagrama de componentes muestra las librerías .jar que lo componen.

A continuación se presenta el diagrama de componentes del sistema:

Tabla 35: Diagrama de Componentes



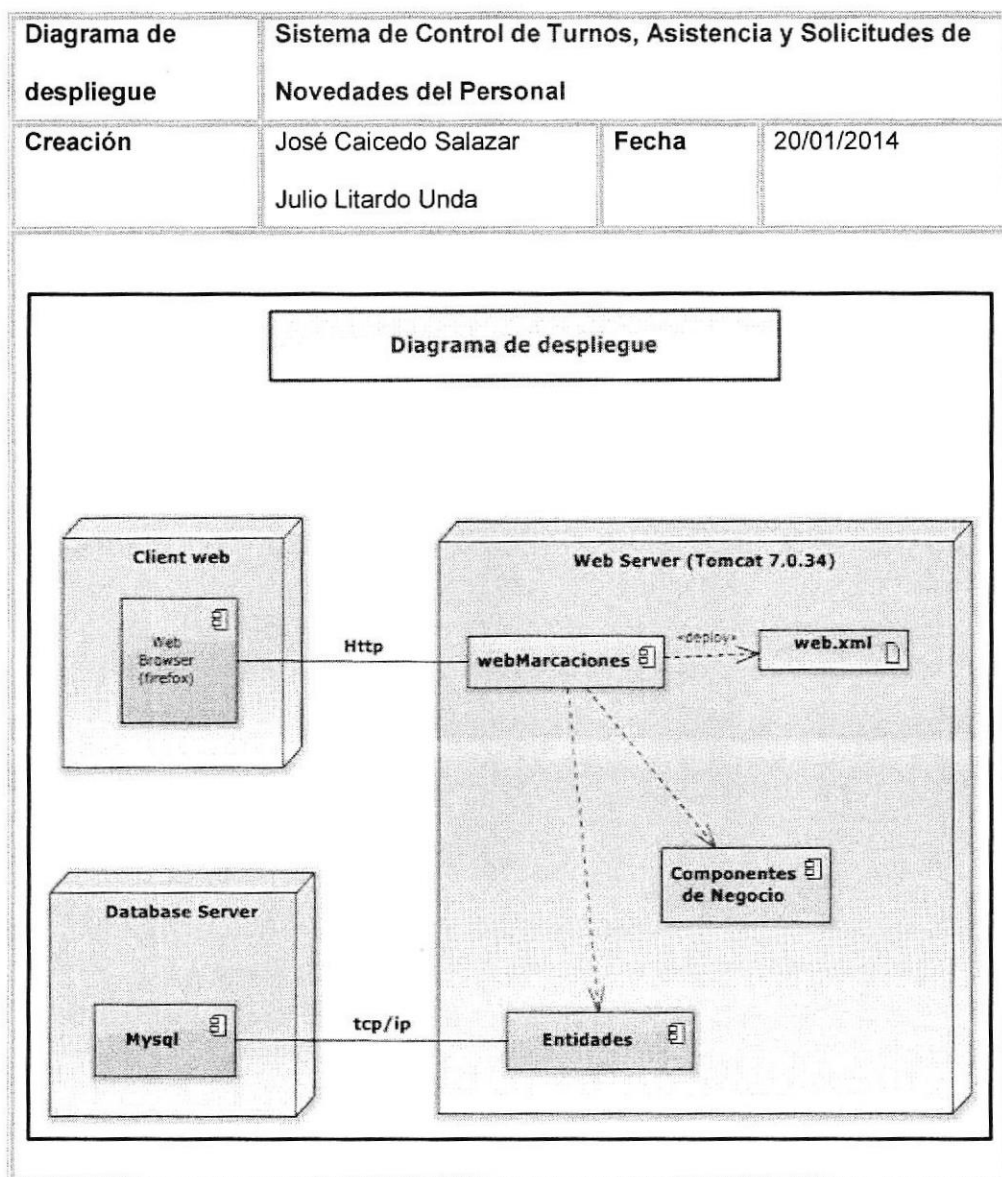
6.4 Diagrama de despliegue

En los diagramas UML contemplados por la metodología Iconix no se encuentra el diagrama de despliegue, pero Iconix no restringe el uso de otros diagramas de UML. Por eso se ha considerado necesario realizar el diagrama de despliegue para modelar el hardware utilizado en la implementación del sistema y las relaciones entre sus componentes.

El diagrama de despliegue muestra la configuración de los elementos de hardware (nodos) y la disposición física de los artefactos software en esos nodos. El diagrama de despliegue junto con el diagrama de componentes es muy útil porque proporciona la vista de implementación del sistema.

A continuación se presenta el diagrama de despliegue del sistema:

Tabla 36: Diagrama de Despliegue



6.5 Pruebas y Análisis de Resultados

6.5.1 Pruebas Unitarias

Para realizar las pruebas unitarias se empleó la librería JUnit4 dentro del IDE Netbeans. El soporte integrado del IDE para la librería de pruebas unitarias JUnit permite escribir y ejecutar pruebas unitarias de clases Java así como conjuntos de pruebas de una forma rápida, fácil y sencilla.

Con la ayuda de la librería JUnit se generaron las pruebas unitarias para las clases entidades, las clases DAO de la capa de acceso a datos y las clases de la capa de negocio. Las clases de la capa de datos y de negocio interactúan con las clases entidades.

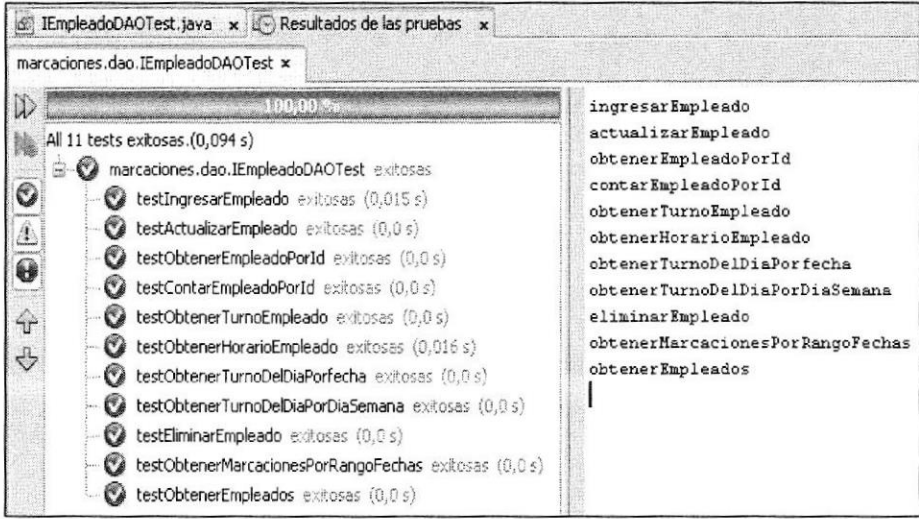
A continuación se presenta el resultado de la prueba unitaria de la clase entidad Empleado.java que se generó con la librería JUnit4. Los resultados fueron satisfactorios de cada uno de los métodos de la clase.

Tabla 37: Prueba unitaria clase entidad

ID Prueba		PU-001	
Tipo Clase	Entidad	Clase	Empleado.java
RESULTADO			

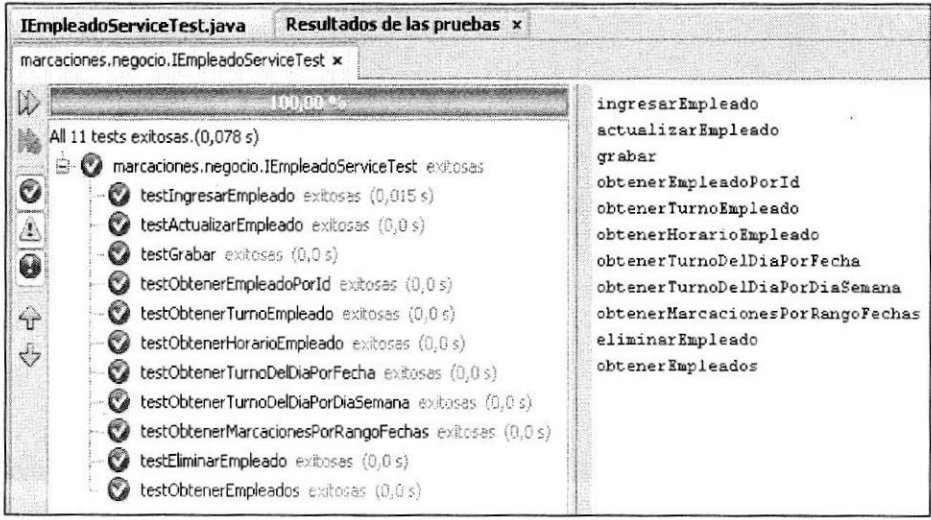
A continuación se presenta el resultado de la prueba unitaria de la clase DAO IEmpleadoDAO.java que se generó con la librería JUnit4. Los resultados fueron satisfactorios de cada uno de los métodos de la clase.

Tabla 38: Prueba unitaria clase capa de datos

ID Prueba		PU-002	
Tipo Clase	Datos	Clase	IEmpleadoDAO.java
RESULTADO			
 <p>The screenshot shows a test runner window with the following content:</p> <ul style="list-style-type: none"> marcaciones.dao.IEmpleadoDAOTest x 100,00 % All 11 tests exitosos. (0,094 s) marcaciones.dao.IEmpleadoDAOTest exitosos testIngresarEmpleado exitosos (0,015 s) testActualizarEmpleado exitosos (0,0 s) testObtenerEmpleadoPorId exitosos (0,0 s) testContarEmpleadoPorId exitosos (0,0 s) testObtenerTurnoEmpleado exitosos (0,0 s) testObtenerHorarioEmpleado exitosos (0,016 s) testObtenerTurnoDelDiaPorFecha exitosos (0,0 s) testObtenerTurnoDelDiaPorDiaSemana exitosos (0,0 s) testEliminarEmpleado exitosos (0,0 s) testObtenerMarcacionesPorRangoFechas exitosos (0,0 s) testObtenerEmpleados exitosos (0,0 s) <p>On the right side of the screenshot, the following methods are listed:</p> <pre> ingresarEmpleado actualizarEmpleado obtenerEmpleadoPorId contarEmpleadoPorId obtenerTurnoEmpleado obtenerHorarioEmpleado obtenerTurnoDelDiaPorFecha obtenerTurnoDelDiaPorDiaSemana eliminarEmpleado obtenerMarcacionesPorRangoFechas obtenerEmpleados </pre>			

A continuación se presenta el resultado de la prueba unitaria de la clase de negocio IEmpleadoService.java que se generó con la librería JUnit4. Los resultados fueron satisfactorios de cada uno de los métodos de la clase.

Tabla 39: Prueba unitaria clase capa de negocio

ID Prueba	PU-003		
Tipo Clase	Negocio	Clase	IEmpleadoService.java
RESULTADO			
 <p>The screenshot shows a test runner window titled "IEmpleadoServiceTest.java" and "Resultados de las pruebas x". The main area displays "All 11 tests exitosos.(0,078 s)". A tree view on the left lists the following tests, all marked as successful (exitosos) with their execution times:</p> <ul style="list-style-type: none"> testIngresarEmpleado exitosos (0,015 s) testActualizarEmpleado exitosos (0,0 s) testGrabar exitosos (0,0 s) testObtenerEmpleadoPorId exitosos (0,0 s) testObtenerTurnoEmpleado exitosos (0,0 s) testObtenerHorarioEmpleado exitosos (0,0 s) testObtenerTurnoDelDiaPorFecha exitosos (0,0 s) testObtenerTurnoDelDiaPorDiaSemana exitosos (0,0 s) testObtenerMarcacionesPorRangoFechas exitosos (0,0 s) testEliminarEmpleado exitosos (0,0 s) testObtenerEmpleados exitosos (0,0 s) <p>On the right side of the window, a list of methods is displayed:</p> <ul style="list-style-type: none"> ingresarEmpleado actualizarEmpleado grabar obtenerEmpleadoPorId obtenerTurnoEmpleado obtenerHorarioEmpleado obtenerTurnoDelDiaPorFecha obtenerTurnoDelDiaPorDiaSemana obtenerMarcacionesPorRangoFechas eliminarEmpleado obtenerEmpleados 			

6.5.2 Pruebas basadas en casos de uso

En las pruebas a partir de casos de uso, se ha utilizado las descripciones de los casos de uso, ya que estas descripciones nos indican la interacción esperada entre el usuario y el sistema para realizar una determinada acción o tarea.

Tabla 40: Prueba PR_RF-001

Módulo	Gestión de Marcaciones	
ID Prueba	PR_RF-001	
Fecha Definición de Prueba	02-12-2014	
ID Caso de uso	CU-001	
Caso de uso	Crear Turno Normal	
ALCANCE		
Definir la configuración de la transacción "CU-001" "Crear Turno Normal", para el proceso de pruebas para la verificación de los campos en el ingreso de los turnos fijos o normales.		
PREREQUISITOS		
DATOS GENERALES DE PRUEBA		
Tipo de Error: "Los campos necesarios no deben estar vacíos (campos indicados con *)"		
Tipo de Error: "Debe ingresar descripción del turno"		
Tipo de Error: "Debe seleccionar un grupo de turno"		
Tipo de Error: "Debe seleccionar al menos un día de marcación"		
Tipo de Error: "Horas de marcación no deben ser cero"		
CASOS DE PRUEBA		
Ord.	Descripción	Resultado
1	Ingresar la descripción del turno	
2	Seleccionar el grupo del turno	
3	Seleccionar el número de marcaciones	
4	Seleccionar los días de marcación del turno	
5	Seleccionar las horas y minutos de las marcaciones	
6	Grabar los datos	Mensaje que indica que los datos fueron guardados



CIB - ESPOL

RESULTADOS OBTENIDOS	
Fecha	Resultado
04-12-2014	Mensaje satisfactorio

Tabla 41: Prueba PR_RF-004

Módulo	Gestión de Marcaciones	
ID Prueba	PR_RF-004	
Fecha Definición de Prueba	02-12-2014	
ID Caso de uso	CU-004	
Caso de uso	Asignar Turno	
ALCANCE		
Definir la configuración de la transacción "CU-004" "Asignar Turno", para el proceso de pruebas para la asignación de turno al personal.		
PREREQUISITOS		
Ejecutar la prueba PR_RF-001		
DATOS GENERALES DE PRUEBA		
Tipo de Error: "Debe seleccionar un turno"		
Tipo de Error: "Debe seleccionar al menos un empleado"		
CASOS DE PRUEBA		
Ord.	Descripción	Resultado
1	Seleccionar el ícono de búsqueda para mostrar toda la lista de turnos ingresados	Muestra la lista de todos los turnos ingresados
2	Seleccionar el turno para asignar	
3	Seleccionar el(los) empleado(s)	
4	Grabar los datos	Turno asignado de forma correcta
RESULTADOS OBTENIDOS		
Fecha	Resultado	
04-12-2014	Turno asignado exitosamente	

Tabla 42: Prueba PR_RF-005

Módulo	Gestión de Marcaciones	
ID Prueba	PR_RF-005	
Fecha Definición de Prueba	02-12-2014	
ID Caso de uso	CU-005	
Caso de uso	Cargar marcaciones	
ALCANCE		
Definir la configuración de la transacción "CU-005" "Cargar marcaciones", para el proceso de pruebas de la carga del archivo texto de marcaciones generado por el reloj biométrico.		
PREREQUISITOS		
Ejecutar la prueba PR_RF-001, PR_RF-004		
DATOS GENERALES DE PRUEBA		
Tipo de Error: "Debe seleccionar un archivo"		
CASOS DE PRUEBA		
Ord.	Descripción	Resultado
1	Se escoge la opción Subir archivo	Se despliega una pantalla para seleccionar el archivo texto de las marcaciones
2	Seleccionar el archivo texto a subir	
3	Cargar archivo	Mensaje que indica que el archivo se subió satisfactoriamente
RESULTADOS OBTENIDOS		
	Fecha	Resultado
	04-12-2014	Mensaje satisfactorio

Tabla 43: Prueba PR_RF-006

Módulo	Gestión de Solicitudes del Personal	
ID Prueba	PR_RF-006	
Fecha Definición de Prueba	02-12-2014	
ID Caso de uso	CU-006	
Caso de uso	Ingresar solicitud de novedad	
ALCANCE		
Definir la configuración de la transacción "CU-006" "Ingresar solicitud de novedad", para el proceso de pruebas de verificar el ingreso de las solicitudes de novedades y los posibles errores.		
PREREQUISITOS		
Ejecutar la prueba PR_RF-001, PR_RF-004, PR_RF-005		
DATOS GENERALES DE PRUEBA		
Tipo de Error: "Los campos necesarios no deben estar vacíos (campos indicados con *)"		
Tipo de Error: "Debe seleccionar un empleado"		
Tipo de Error: "Debe seleccionar un tipo"		
Tipo de Error: "Debe seleccionar un motivo"		
Tipo de Error: "Debe ingresar una observación"		
Tipo de Error: "Debe seleccionar los días de la novedad"		
Tipo de Error: "Fecha final debe ser mayor a fecha inicio"		
CASOS DE PRUEBA		
Ord.	Descripción	Resultado
1	Seleccionar el ícono de búsqueda para mostrar toda la lista de empleados activos ingresados	Muestra la lista de todos los empleados activos ingresados
2	Seleccionar el empleado	Muestra el turno activo del empleado



3	Seleccionar duración en días de la novedad (un sólo día o varios días)	
4	Seleccionar la marcación a afectar	
5	Ingresar la fecha de inicio	
6	Ingresar la fecha final si es requerida	Validar los campos de fecha final contra fecha de inicio
7	Ingresar la observación	
8	Grabar los datos	Novedad almacenada satisfactoriamente
RESULTADOS OBTENIDOS		
Fecha		Resultado
04-12-2014		Mensaje satisfactorio

Tabla 44: Prueba PR_RF-007

Módulo	Gestión de Solicitudes del Personal
ID Prueba	PR_RF-007
Fecha Definición de Prueba	02-12-2014
ID Caso de uso	CU-007
Caso de uso	Aprobar solicitud de novedad
ALCANCE	
Definir la configuración de la transacción "CU-007" "Aprobar solicitud de novedad", para el proceso de pruebas de verificar la aprobación de las solicitudes de novedades y los posibles errores.	
PREREQUISITOS	
Ejecutar la prueba PR_RF-006	
DATOS GENERALES DE PRUEBA	
Tipo de Error: "Debe seleccionar una solicitud"	
CASOS DE PRUEBA	

Ord.	Descripción	Resultado
1	Seleccionar el ícono de búsqueda para mostrar toda la lista de solicitudes ingresadas	Muestra la lista de todos las solicitudes ingresadas
2	Seleccionar la solicitud	Muestra la información de la solicitud
3	Aprobar la solicitud	Solicitud aprobada satisfactoriamente
RESULTADOS OBTENIDOS		
Fecha		Resultado
04-12-2014		Mensaje satisfactorio



CIB - ESPOL

CAPÍTULO 7

7 INSTALACIÓN

En este capítulo se especifican los elementos y requerimientos propios para el proceso de instalación e implementación del sistema una vez que ha concluido su construcción. Se explican entre otras cosas los componentes de hardware y software requeridos que asegurarán el funcionamiento óptimo del sistema, así como el proceso de capacitación de los usuarios para lograr la explotación de las ventajas del sistema.

7.1 Requerimientos de Instalación

A continuación se determina la infraestructura tecnológica y la capacidad técnica que se requiere para la implementación del sistema.

7.1.1 Requerimiento de Hardware

No es necesario que se adquiriera un equipo servidor ya que el sistema puede funcionar sobre una computadora de escritorio. Se requiere de un computador donde se instalará el servidor web, el servidor de base de datos y el sistema web con las siguientes características:

Tabla 45: Requerimiento de Hardware Servidor Web y Servidor Base de Datos

HARDWARE SERVIDOR WEB Y SERVIDOR BASE DE DATOS	
EQUIPO	CARACTERISTICAS
1 computadora de escritorio propietaria o ensamblada	Procesador: i Core 7 Disco Duro: 1 Tb Memoria: 4 Gb de RAM. Monitor: 17" VGA Mouse: De dos botones, Teclado: Estándar de 101/102 teclas. DVD-ROM

Los equipos que utilicen los clientes se requiere que cuenten con las siguientes características:

Tabla 46: Requerimiento de Hardware Computadoras Clientes

HARDWARE COMPUTADORAS CLIENTES	
EQUIPO	CARACTERISTICAS
computadora de escritorio propietaria o ensamblada	Procesador: Celeron o superior Disco Duro: 40 Gb Memoria: 2 Gb de RAM Monitor: 17" VGA Mouse: De dos botones, Teclado: Estándar de 101/102 teclas. DVD-ROM

La empresa ya cuenta con equipos disponibles con las características indicadas y hasta superiores, para que los usuarios se puedan conectar al sistema.

7.1.2 Requerimiento de Software

La computadora donde se instalará el servidor Web, el servidor de base de datos y el sistema se requiere que cuente con el siguiente software:

Tabla 47: Requerimiento de Software Servidor Web y Servidor Base de Datos

SOFTWARE SERVIDOR WEB Y BASE DE DATOS	
SOFTWARE	CARACTERISTICA
Sistema Operativo	Linux Centos 6.4
Servidor Web	Apache Tomcat 7.0.34
Servidor Base de Datos	MySQL 5.5

Las computadoras clientes operan bajo la plataforma Windows 7. Para el uso general de los equipos en la utilización del sistema se requiere que cuenten con el siguiente software:

Tabla 48: Requerimiento de Software Computadoras Clientes

SOFTWARE COMPUTADORAS CLIENTES	
CARACTERISTICA	DESCRIPCION
Sistema Operativo	Windows XP SP 3 o superior
Navegador Web	Mozilla Firefox última versión

7.2 Capacitación

La capacitación en los procesos de instalación, configuración y operación constituye uno de los pasos finales que se realiza después del desarrollo de una aplicación de software. La

capacitación en el sistema se realizó en base a un cronograma con los usuarios en las distintas áreas de la empresa.

En este proceso de capacitación se revisaron en detalle las diferentes opciones del sistema, usando algunos casos de prueba establecidos y algunos ejemplos integrales para lograr un mejor entendimiento y de esta manera despejar cualquier duda acerca del sistema.

A continuación se detalla la temática cubierta durante el aprendizaje del sistema:

- Presentación general de los módulos del sistema.
- Instalación y configuración de los parámetros
- Descripción y operación de los módulos.
- Análisis de los casos de prueba y ejemplos
- Operaciones de ingresos, eliminación y modificación de la información.
- Manejo de las ayudas y del manual de usuario.
- RespalDOS y Reportes.

Una vez culminado con este proceso, los usuarios directos e indirectos del sistema pudieron notar su gran utilidad para el desarrollo de las diferentes actividades que ellos realizan en sus diferentes áreas.

7.3 Mantenimiento

El mantenimiento del sistema es una de las actividades más relevantes que se debe realizar después de la implementación del sistema. Esta actividad demuestra el cumplimiento de un mito en el desarrollo de software que dice: "Con la implementación del software el trabajo termina", mientras que la realidad demuestra que "el mantenimiento requiere mayor esfuerzo y tiempo".

Para realizar operaciones efectivas de mantenimiento en el sistema, se ha acordado con la Gerencia General de la empresa una garantía de servicio de mantenimiento del sistema por un tiempo de 6 meses. Durante este período se brindará soporte técnico remoto (vía telefónica con una asistencia a cualquier requerimiento) y presencial (se realizará dos veces por mes y en horario establecido de mutuo acuerdo).

Durante el tiempo del soporte técnico presencial en que los usuarios utilizan el sistema, se procederá a documentar todas las novedades en cuanto a la funcionalidad, veracidad de los procesos y tiempos de respuesta. Posteriormente se ejecutará un procedimiento de depuración del código y rectificación de los errores del sistema para volver a probarlo nuevamente.

De igual forma, es menester ir registrando las peticiones de los usuarios para analizarlas, documentarlas por escrito y de ser factibles entrar en el proceso de desarrollo de estas nuevas peticiones y modificaciones.

CAPITULO 8

8 ESTUDIO DE FACTIBILIDAD

Se revisan aspectos que permiten la viabilidad o factibilidad del desarrollo del sistema. Se realiza la estimación de costo del sistema y se plantean los beneficios tangibles e intangibles que ofrecerá el sistema.

8.1 Costos del Sistema

Solamente se consideran los costos del desarrollo del sistema, no se toman en cuenta los costos de equipos (hardware) para la implementación del mismo debido a que la empresa cuenta con los equipos que le permite satisfacer los requerimientos de hardware y software.



CIB - ESPOL

Costos del Desarrollo del Sistema

Se divide en: Costos del personal de desarrollo, costos de equipo para el desarrollo, y otros costos.

- **Costos del personal de desarrollo.** Lo representa el valor por honorarios profesionales para el personal que se va a encargar del desarrollo del sistema.

Tabla 49: Costos del personal de desarrollo

Descripción	Costo (\$)
Análisis, diseño, programación e implementación (\$ 600.00 mes * 2 personas * 4 meses)	\$ 4.800,00

- **Costos del equipo para desarrollar.** Lo constituyen los costos incurridos en el uso del equipo necesario para desarrollar el sistema en un total de 4 meses.

Tabla 50: Costos del equipo para desarrollar

Descripción	Costo (\$)
Computador (\$ 1.00 hora * 8horas/día * 6 días/semana * 4 semanas/mes * 4 meses)	\$ 760,00
Impresora (Pruebas y Reportes) (\$ 0.10 hoja * 100 hojas máximo)	\$ 10,00
TOTAL	\$ 770,00

- **Costos varios.** Está formado por los costos que incluyen materiales de oficina y algunos imprevistos.

Tabla 51: Costos Varios

Descripción	Costo (\$)
Materiales de Oficina	\$ 160,00
Imprevistos	\$ 80,00
TOTAL	\$ 240,00

A continuación se presenta una tabla con el resumen de los costos del desarrollo del sistema.

Tabla 52: Resumen de Costos del Desarrollo del Sistema

COSTOS	TOTAL (\$)
Costos del personal de desarrollo	\$ 4.800,00
Costos del equipo para desarrollar	\$ 770,00
Costos Varios	\$ 240,00
TOTAL COSTOS DEL DESARROLLO DEL SISTEMA	\$ 5.810,00

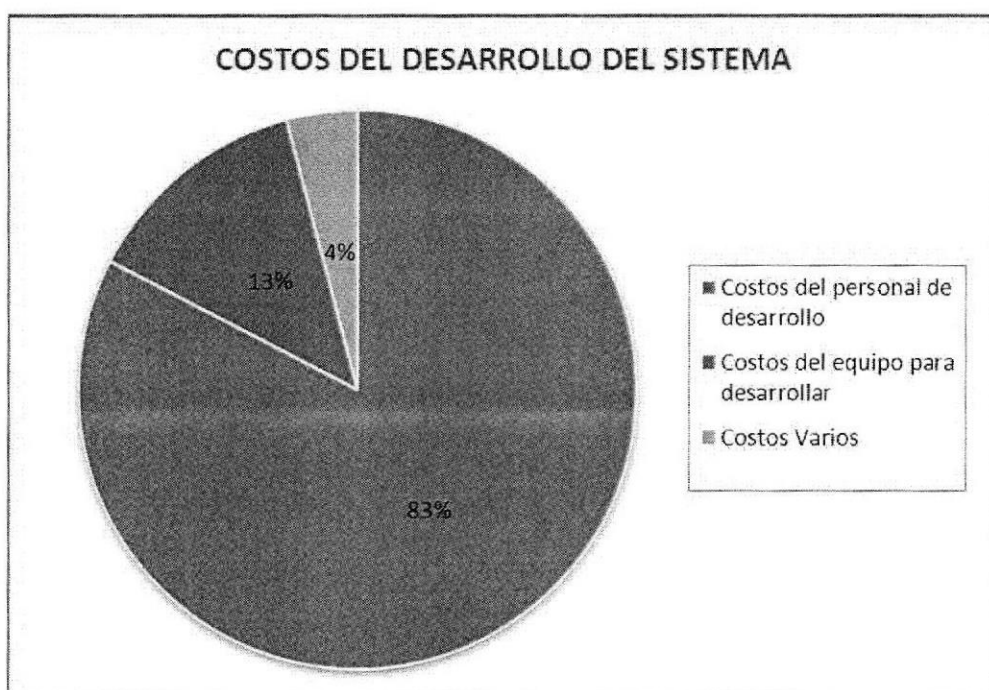


Figura 8.1: Resumen de Costos del Desarrollo del Sistema

8.2 Beneficios tangibles e intangibles

Uno de los principales beneficios obtenido con la implementación del sistema es que mejora la forma en que actualmente se lleva a cabo el proceso de control de asistencia y de turnos del personal y

por ende favorece la labor de la empresa. Es decir, mejora el proceso en varios aspectos, reportando beneficios tanto tangibles como intangibles.

8.2.1 Beneficios tangibles

Son aquellos beneficios que proporcionan ventajas económicas y de tiempo cuantificables. El sistema brinda los siguientes beneficios tangibles:

1. Disminución notable del tiempo empleado para elaborar los turnos del personal, sobre todo los turnos rotativos.
2. Ahorro notable de papelería y del material empleado en procesar y archivar las solicitudes de permisos ocasionales del personal. Aproximadamente se emiten 8 permisos ocasionales diarios, que al mes representan 160 permisos. Es decir en el mes se consumían aproximadamente un block y medio, y cada block tiene un costo de \$ 2,80 lo que representa un total de ahorro

de \$ 4,20. Con el sistema estas solicitudes son ingresadas y aprobadas, sin ser necesaria su impresión.

3. Reducción significativa del tiempo empleado por el personal del Departamento de Gestión de Talento Humano para determinar el total de atrasos, faltas y horas extras de cada empleado de manera mensual.
4. Apoya el trabajo al departamento Financiero ya que contará con información real sobre las faltas, atrasos y horas trabajadas del personal. Y de esta forma podrá emitir los roles de pago con valores reales de acuerdo al trabajo realizado.
5. Simplifica las labores del departamento de Gestión de Talento Humano ya que podrá obtener reportes y resúmenes sobre el cumplimiento del personal en su horario de trabajo bajo diferentes parámetros como son: rangos de fecha, departamento, tipo de empleado, para así tomar las decisiones adecuadas.

6. Permitirá que el Departamento de Gestión de Talento Humano realice los respectivos memos de llamado de atención al personal por los atrasos, de manera rápida y ágil.
7. Acortar de manera significativa el tiempo de ingreso y aprobación para una solicitud de permiso ocasional, vacaciones, etc. El jefe departamental visualizará las solicitudes pendientes de aprobación ordenadas de forma descendente por fecha de ingreso, que fueron ingresadas por el personal a su cargo.
8. Ahorro de gastos en adquisición de software propietario con altos costos de licencia, ya que el sistema para su implementación emplea software Open Source para el servidor web y el servidor de base de datos, es decir cero costos por licenciamiento.

Beneficios del ahorro de tiempo

En la siguiente tabla, se compara los tiempos efectuados en una actividad de forma manual y el sistema de información, en la cual podemos apreciar que el sistema de información frente al sistema manual, ofrece significativamente un ahorro de tiempo en cada una de las actividades mencionadas. Las comparaciones se han hecho considerando como base que el personal tiene una jornada de 8 horas diarias.



Tabla 53: Beneficios de tiempo

#	Tareas Frecuentes	Sistema		Beneficio	Ahorro en tiempo (min.)
		Manual	Automática		
		Tiempo (min)	Tiempo (min)		
1	Obtener información personal de un empleado	15	1	Diariamente se busca aproximadamente información de 5 empleados, lo que implica utilizar 75 min. de forma manual mientras que con el sistema será 5 min.	14
2	Registrar un nuevo empleado	20	5	Registrar la información de un nuevo empleado toma 20 min de forma manual, mientras que con el sistema se toma 5 min	15
3	Tarjar las marcaciones del personal del mes	960	6	Tarjar las marcaciones de manera manual toma 960 min, es decir dos días de trabajo, mientras que con el sistema será de 6 min. Se podrá utilizar el tiempo en otras actividades	954
4	Obtener un informe mensual de	300	3	Información rápida y segura en el	297

	atrasos y horas extras del personal			momento necesario	
5	Elaborar el turno del empleado	25	5	En el día se podrá elaborar más de 19 turnos con el sistema	20
6	Asignar turnos al personal	20	2	Se podrá asignar más de 24 turnos diariamente con el sistema	18
7	Elaborar un informe mensual de las solicitudes de permisos ocasionales y de acciones de personal realizadas por un empleado	300	3	Información rápida y segura en el momento necesario	297
8	Realizar una solicitud de permiso ocasional y de acción de personal	30	3	En el día se podrán ingresar más de 16 solicitudes con el sistema	27
9	Aprobar la solicitud de permiso ocasional y de acción de personal	20	2	Diariamente se podrán aprobar más de 24 solicitudes con el sistema	18

Beneficios económicos

El objetivo principal de la empresa con la implementación del Sistema es obtener un significativo ahorro de tiempo, el mismo que se expresará en términos económicos. Para poder apreciar el beneficio económico que se obtiene con el sistema, se ha tomado en cuenta lo siguiente:

1. El sueldo neto del Asistente del Dpto. de Talento Humano es de \$ 700,00 mensuales en una jornada de 8 horas diarias de trabajo.
2. El costo del Asistente del Dpto. de Talento Humano es de \$ 0,058 por minuto ($\$ 700,00/\text{mes} * 1 \text{ mes}/5 \text{ semanas} * 1 \text{ semana}/5 \text{ días} * 1 \text{ día}/480 \text{ minutos}$).
3. El personal labora 300 días en el año ($12 \text{ meses/año} * 5 \text{ semanas/mes} * 5 \text{ días/semana}$).

Estos puntos se emplearán como base para hallar el beneficio obtenido en el desarrollo de algunas actividades

en un lapso de un año. En la siguiente tabla se muestran los beneficios económicos que se obtienen en cada actividad realizada de la Tabla 53.

Tabla 54: Beneficios Económicos por Actividad

# Actividad	Tiempo por actividad		Costo Anual por Actividad (\$)		Beneficio Anual (\$) I - II
	Manual A	Sistema Información B	Manual A * (300) * (0,058) I	Sistema Información B * (300) * (0,058) II	
1	15	1	261	17,4	243,6
2	20	5	348	87	261
3	960	6	16704	104,4	16599,6
4	300	3	5220	52,2	5167,8
5	25	5	435	87	348
6	20	2	348	34,8	313,2
7	300	3	5220	52,2	5167,8
8	30	3	522	52,2	469,8
9	20	2	348	34,8	313,2
Total	1690	30	29406,00	522,00	28884,00

8.2.2 Beneficios Intangibles

Son aquellos que reportan beneficios organizativos, de funcionamiento o eficiencia. El sistema propuesto brinda los siguientes beneficios intangibles:

1. Se cuenta con una información organizada, que permitirá al personal de Gestión de Talento Humano contar con información oportuna en forma más rápida y segura.
2. Garantizar seguridad a la información a través de los usuarios.
3. Reducción de pérdidas de información y facilidad para realizar cambios.
4. Incremento de la eficiencia y precisión en el proceso de elaboración de turnos del personal.

5. Agilizar el proceso de elaboración de los turnos incrementando la eficiencia y precisión del personal.

6. Mejora el funcionamiento y la organización de los procesos del Departamento de Gestión de Talento Humano.

7. Incremento en la satisfacción de los empleados al facilitar tareas tediosas y complejas como la de consultar atrasos y horas extras.

8.3 Análisis Económico

Consiste en evaluar los costos estimados y los beneficios obtenidos, con el fin de determinar si la rentabilidad de la implementación del sistema.

El mayor beneficio obtenido será cuando el sistema entre a funcionar, ya que el objetivo principal es el ahorro de tiempo.

Para determinar si el sistema es rentable, se considera que el mismo tiene 5 años de vida útil. En la siguiente tabla se muestra los costos proyectados para 5 años de vida útil.

Tabla 55: Costos Manual vs. Sistema Información en proyección de 5 años

Años \ Costos	0	1	2	3	4	5	Total (CM)
Manual (CM):							
Actividades		29406,00	29406,00	29406,00	29406,00	29406,00	\$ 147030,00

Sistema Información (CSI):							
Costos							
Desarrollo del Sistema	5810,00						5810,00
Mantenimiento Actividades		522,00	522,00	522,00	522,00	522,00	2610,00
						Total (CSI)	\$ 8420,00

$$VA = CM - CSI$$

Donde:

VA: Valor del Ahorro

CM: Costo Manual

CSI: Costo del Sistema de Información

$$VA = 147030,00 - 8420,00$$

$$VA = \$ 138610$$



Dado que $VA > 0$, se concluye que el Sistema de Información es rentable para el tiempo de vida útil de 5 años.

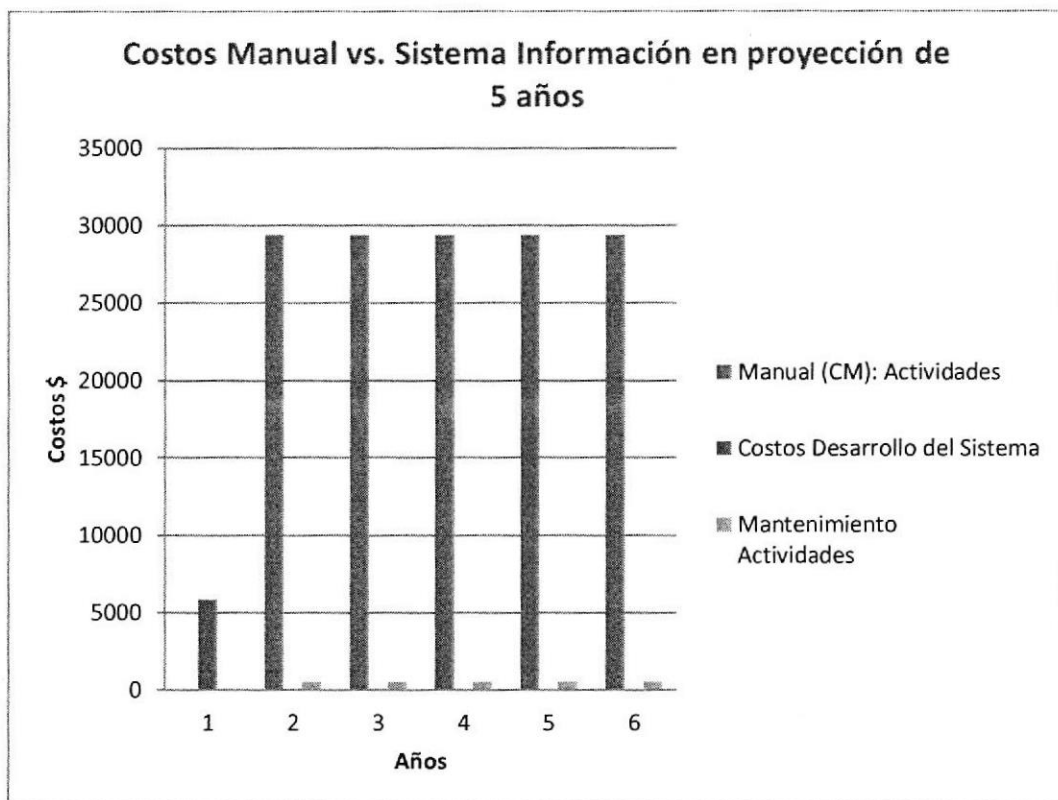


Figura 8.2: Costos Manual vs. Sistema Información

CONCLUSIONES

1. Se desarrolló el sistema, utilizando los fundamentos que propone la metodología ICONIX para el análisis y diseño del sistema; permitiendo realizar un correcto modelado ya que esta metodología se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. Su principal objetivo es que utiliza sólo ciertos diagramas UML que son los más importantes dentro de cada una de las fases del desarrollo, permitiendo que su uso pueda dar la trazabilidad y seguimiento al proyecto.
2. La arquitectura por capas y la integración de cada una de las capas a través de interfaces permite que el sistema sea flexible a cualquier cambio de tecnología que se realice en cualquiera de estas, así como

los componentes de negocio puedan ser reutilizados, en cualquier otro sistema o pueda ser utilizado por herramientas de integración de componentes.

3. Como se pudo apreciar en el análisis de costo beneficio, el sistema es factible ya que al amortizar el costo de desarrollo a 5 años podemos observar el beneficio económico es tangible con la utilización del sistema.

RECOMENDACIONES

1. El sistema podría integrarse a otros sistemas que maneja la empresa y que han sido desarrollados con tecnología open source, pueden integrarse a través del portal liferay, esto está en estudio para que pueda ser realizado más adelante.

GLOSARIO

AJAX: Es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

API: Conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Representa una interfaz de comunicación entre componentes software.

ANOTACIÓN JAVA: Forma de añadir metadatos al código fuente Java que están disponibles para la aplicación en tiempo de ejecución.



CIB - ESPOL

ARQUITECTURA: Representación abstracta de los componentes de un sistema y su comportamiento

BEAN: Es un componente software que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno. Un Bean puede representar desde un botón, un grid de resultados, un panel contenedor o un simple campo de texto, hasta otras soluciones mucho más complejas como conexiones a bases de datos, etc.

CLIENTE: Aplicación que permite a un usuario obtener un servicio de un servidor localizado en la red.

CLIENTE/SERVIDOR: describe la relación entre dos programas informáticos en la que un programa, el cliente, realiza una solicitud de servicio a otro programa, el servidor, que satisface esta solicitud. Normalmente, varios programas de cliente comparten los servicios de un programa de servidor común. Un navegador Web es un programa de cliente que solicita servicios (el envío de páginas Web o archivos) a un servidor Web.

COMPONENTE: Es una clase abstracta que representa todo lo que tiene una posición, un tamaño, puede ser pintado en pantalla y puede recibir eventos.

CONTENEDORES AJAX. *AjaxContainer* es una interfaz que describe un área de una página JSF que debe ser decodificado en una petición Ajax.

DAO: componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

ENTITYMANAGER: Interfaz que define los métodos que son usados para interactuar con el contexto de persistencia.

ETIQUETA: también conocida como *TAG*. Es una marca con clase que delimita una región en los lenguajes basados en XML.

FRAMEWORK DE PERSISTENCIA: Componente de software encargado de traducir entre objetos y registros (de la base de datos relacional). Es decir es el encargado de que el programa y la base de datos se "entiendan".

HIBERNATE: Herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

JAVA: Lenguaje de Programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

JDBC: Java DataBase Connectivity es el API de Java que define como una aplicación cliente accederá a una base de datos, independientemente del motor de base de datos al que accedamos.

JPA: Framework de Persistencia.

JSR (Java Specification Request): Son documentos formales que describen las especificaciones y tecnologías propuestas para que sean añadidas a la plataforma Java.

KIT RENDERIZADOR: define como se mapean las clases de los componentes a las etiquetas de componentes apropiadas para un cliente particular.

LIBRERÍA DE COMPONENTES: Componentes agrupados con o sin criterio pero lo ideal es que se dediquen a funciones específicas

METADATOS: Datos que describen otros datos.

MVC: Patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

ORM: Técnica que realiza la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

PATRÓN DE DISEÑO: Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema

satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

PERSISTENCIA: Capacidad de almacenar y recuperar el estado de los objetos, de forma que sobrevivan a los procesos que los manipulan.

POJO: Simple clase Java que tiene métodos get y set para cada uno de los atributos.

PATRÓN DE DISEÑO: Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

PROTOCOLO DE TRANSFERENCIA: Un protocolo de transferencia de datos es un formato estandarizado de transmisión de datos entre dos dispositivos. El tipo de protocolo usado puede determinar variables como el método de comprobación de errores, el método de compresión de datos y el reconocimiento de fin de archivo.

RED: Grupo de computadoras y otros dispositivos conectados que pueden compartir archivos y recursos, como por ejemplo una impresora.

SERIALIZACIÓN: Secuencia de bytes escrita en un fichero en disco.

SERVIDOR WEB: Es el programa que, utilizando el protocolo de comunicaciones HTTP, es capaz de recibir peticiones de información de un programa cliente (navegador), recuperar la información solicitada y enviarla al programa cliente para su visualización por el usuario.

SISTEMA DINÁMICO: Un sistema dinámico es un sistema complejo que presenta un cambio o evolución de su estado en un tiempo, el comportamiento en dicho estado se puede caracterizar determinando los límites del sistema, los elementos y sus relaciones; de esta forma se puede elaborar modelos que buscan representar la estructura del mismo sistema

SQL: (STRUCTURED LANGUAGE QUERY. LENGUAJE ESTRUCTURADO DE CONSULTAS). Lenguaje utilizado para manipular bases de datos relacionales. Es uno de los lenguajes más utilizados de la informática.

VALIDACIÓN: proceso de revisión al que se somete un programa informático para comprobar que cumple con sus especificaciones. Dicho proceso, que suele tener lugar al final de la etapa de desarrollo, se realiza principalmente con la intención de confirmar que el software esté en condiciones de desarrollar las tareas que el **usuario** que lo adquiere planea llevar a cabo.

BIBLIOGRAFÍA

- [1] Peláez, J., Arquitectura basada en capas, <http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-basada-en-capas/>, fecha de consulta noviembre 2014
- [2] Pérez, L., Introducción a Arquitectura en Capas, <http://www.practicadesoftware.com.ar/2010/12/introduccion-a-arquitectura-en-capas/#.VNJW1iwpo2k>, fecha de consulta noviembre 2014
- [3] González, Y. D., & Romero, Y. F., Patrón Modelo-Vista-Controlador, *Revista Telem@ Tica*, 2012
- [4] King, G., & Bauer, C., Hibernate in Action: Practical Object/Relational Mapping, Manning, October 2004
- [5] Hibernate.org, Hibernate ORM. Retrieved <http://hibernate.org/orm/>, fecha de consulta noviembre 2014
- [6] Christian, B., & Gavin, K., *Java Persistence with Hibernate*, Manning, 2006.



[7] Keith, M., & Schincariol, M., *Pro EJB 3: Java Persistence API*, Apress, 2006.

[8] Burns, E., Schalk, C., & Griffin, N., *JavaServer Faces 2.0*, McGraw-Hill, 2010.

[9] Rosenberg, D., Stephens, M., & Collins-Cope, M., *Agile development with ICONIX process*, Apress, 2005.

ANEXOS

Anexo 1: Descripción de campos de las tablas

mar_cargas:		Registra información de la carga de los archivos de marcaciones que son descargados del reloj biométrico				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	secuencia	Secuencia de la tabla	int(11)	PK		No
2	sucursal	Código de la sucursal	int(11)	FK	mar_sucursal	No
3	tipo	Tipo : A = Activo I = Inactivo	char(1)			No
4	reloj	Número del reloj	smallint(6)			Si
5	nombreArchivo	Nombre del archivo del	varchar(30)			Si
6	totalRegistros	Número de registros del archivo	smallint(6)			Si
7	totalEmpleados	Cantidad de empleados que realizaron las marcaciones	smallint(6)			Si
8	totalDescartadas	Número de registros descartados	smallint(6)			Si
9	observacion	Observaciones que existan en relación a la carga del registro	varchar(100)			Si
10	fechaIni	Fecha inicial de la carga	date			No
11	fechaFin	Fecha final de carga	date			No
12	Usuario	Usuario que realizó la	varchar(15)			No
13	fechaReg	Fecha en que se realizó la carga	datetime			No



14	estado	Estado: A = Activo I = Inactivo	char(1)			No
----	--------	---------------------------------	---------	--	--	----

mar_cargos:		Registra los diferentes cargos que existen en la empresa				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	Id	Identificador del cargo.	int(11)	PK		No
2	descripcion	Nombre del cargo.	varchar(35)			No
3	horasExtras	Cantidad de horas extras.	char(1)			No
4	costoHN	Valor de hora laborable.	decimal(5,2)			Si
5	porCHE	Porcentaje de horas extras.	decimal(5,2)			Si
6	usuario	Usuario que realizó la acción.	varchar(15)			No
7	fechaReg	Fecha que se realizó la acción.	datetime			No
8	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_descartadas:		Contiene las marcaciones descartadas y sus observaciones				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	secuencia	Secuencia de la tabla	int(11)	PK		No
2	empleado	Código del empleado	int(11)			Si
3	cargas	Número de carga	int(11)	FK	mar_cargas	No
4	periodo	Año de la carga	int(11)			No
5	mes	Mes de la carga	int(11)			No
6	día	Día de la carga	int(11)			No
7	fecha	Fecha completa de la carga	date			No
8	hora	Hora de marcación del	smallint(6)			No
9	minuto	Minuto de marcación del empleado	smallint(6)			No
10	observacion	Novedad de la marcación	varchar(250)			Si
11	usuario	Usuario que realizó la acción	varchar(15)			No
12	fechaReg	Fecha que se realizó la acción	datetime			No
13	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_dpto:		Registra los diferentes departamentos de la empresa				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	Id	Secuencia de la tabla.	int(11)	PK		No
2	nombre	Nombre del departamento.	varchar(40)			No
3	usuario	Usuario que realizó la acción.	varchar(15)			No
4	fechaReg	Fecha que se realizó la acción.	datetime			No
5	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_dturnos:		Registra el detalle de los turnos de trabajo con hora de inicio y fin de la jornada de los empleados				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	mturnos	Secuencia del turno	int(11)	PK, FK	mar_mturnos	No
2	linea	Día de la semana.	int(11)	PK		No
3	fecha	Fecha de inicio de los turnos rotativos	date			Si
4	horalnicio	Hora de inicio de labores	smallint(6)			No
5	minInicio	Minuto de inicio de labores	smallint(6)			No
6	horaFin	Hora de fin de labores	smallint(6)			No
7	minFin	Minuto de fin de labores	smallint(6)			No
8	lunch	Si tiene o no lunch	char(1)			No
9	horalniLunch	Hora de inicio de lunch	smallint(6)			No
10	minIniLunch	Minuto de inicio de lunch	smallint(6)			No
11	horaFinLunch	Hora de fin de lunch	smallint(6)			No
12	minFinLunch	Minuto de fin de lunch	smallint(6)			No
13	stipopre	Prefijo del tipo	char(3)	FK	gen_supertip	No
14	stipoid	Id del tipo	smallint(6)	FK	gen_supertip	No
15	usuario	Usuario que realizó la acción	varchar(15)			No
16	fechaReg	Fecha que se realizó la acción	datetime			No
17	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_ empleado:		Almacena información de los empleados				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código del empleado.	int(11)	PK		No
2	sucursal	Código de la sucursal.	int(11)			Si
3	cargos	Código del cargo	int(11)	FK	mar_cargos	No
4	dpto	Código del departamento	int(11)	FK	mar_dpto	No
5	tipo	Tipo: A = Administrativo	char(1)			Si
6	jefe	Código del Jefe del empleado.	int(11)			Si
7	cedula	Cédula del empleado.	varchar(13)			No
8	nombres	Nombres y apellidos del empleado.	varchar(80)			No
9	direccion	Dirección de domicilio del empleado.	varchar(80)			Si
10	telefonos	Convencional /celular.	varchar(30)			Si
11	fechaNac	Fecha de nacimiento del empleado.	date			Si
12	fechaIngreso	Fecha de ingreso a la empresa.	date			Si
13	fechaSalida	Fecha de salida de la empresa.	date			Si
14	sueldo	Salario del empleado.	decimal(7,2)			Si
15	horas	Horas que labora el empleado.	smallint(6)			Si
16	horasExtra	Horas adicionales que labora el empleado.	char(1)			No



17	horasReales	Horas exactas de labores.	smallint(6)			Si
18	clave	Clave del empleado.	varchar(10)			Si
19	foto	Foto del empleado.	varchar(120)			Si
20	usuario	Usuario que realizó la acción	varchar(15)			No
21	fechaReg	Fecha que se realizó la acción	datetime			No
22	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_marcaciones:		Registra las marcaciones de entrada y salida de los empleados				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	secuencia	Secuencia de la marcación	int(11)	PK		No
2	empleado	Código del empleado	int(11)	FK	mar_empleado	No
3	mtornos	Código del turno	int(11)	FK	mar_mtornos	No
4	cargas	Número de la carga	int(11)	FK	mar_cargas	Si
5	periodo	Año del periodo de marcación	int(11)			No
6	mes	Mes de marcación	int(11)			No
7	día	Día de marcación	int(11)			No
8	fecha	Fecha de marcación	date			No
9	horaEE	Hora de entrada empresa	smallint(6)			No
10	minEE	Minuto de entrada	smallint(6)			No
11	horaPE	Hora de entrada del permiso	smallint(6)			No
12	minPE	Minuto de entrada del permiso	smallint(6)			No
13	horaSL	Hora de salida de lunch	smallint(6)			No
14	minSL	Minuto de salida de lunch	smallint(6)			No
15	horaRL	Hora de retorno de lunch	smallint(6)			No
16	minRL	Minuto de retorno de lunch	smallint(6)			No
17	horaPS	Hora de salida del permiso	smallint(6)			No
18	minPS	Minuto de salida del permiso	smallint(6)			No
19	horaSE	Hora salida empresa	smallint(6)			No
20	minSE	Minuto de salida empresa	smallint(6)			No

21	totalAT	Total de atrasos	smallint(6)			No
22	totalHE	Total de horas extras	smallint(6)			No
23	novedad	Número de la novedad	int(11)			No
24	observación	Observación de las marcaciones	varchar(100)			Si
25	usuario	Usuario que realizó la acción	varchar(15)			No
26	fechaReg	Fecha que se realizó la acción	datetime			No
27	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_mturnos:		Registra los diferentes turnos de la empresa				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	id	Identificador del turno	int(11)	PK		No
2	tipoturnos	Tipo de grupo de turnos	int(11)	FK	tipoturnos	No
3	descripcion	Nombre del turno	varchar(60)			No
4	fecha	Fecha de creación del turno	date			No
5	penalizacion	Número de marcaciones	smallint(6)			Si
6	gracia	Tiempo adicional	smallint(6)			Si
7	tipo	N = Normal, F = Frecuencia, V = Variable	char(1)			No
8	marcacion	Horas laborables	smallint(6)			Si
9	descanso	Horas de descanso	smallint(6)			Si
10	usuario	Usuario que realizó la acción	varchar(15)			No
11	fechaReg	Fecha que se realizó la acción	datetime			No
12	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_nov_x_emp:		Registra las novedades que se presenten de la marcaciones de los empleados				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	secuencia	Secuencia de novedades	int(11)	PK		No
2	novedades	Código de la novedad	int(11)	FK	mar_novedad	No
3	empleado	Código del empleado	int(11)	FK	mar_emplead	No
4	periodo	El código de novedades	int(11)			No
5	mes	El mes de la novedad	int(11)			No
6	día	El día de la novedad	int(11)			No
7	fecIni	Fecha inicial de la novedad	date			No
8	fecFin	Fecha final de la novedad	date			No
9	HH	Hora inicio	int(11)			No
10	MM	Minuto inicio	int(11)			No
11	HS	Hora fin	int(11)			No
12	MS	Minuto fin	int(11)			No
13	observacion	Observación de la novedad	varchar(150)			Si
14	usuario	Usuario que realizó la acción	varchar(15)			No
15	fechaReg	Fecha que se realizó la acción	datetime			No
16	estado	Estado: A = Activo I = Inactivo	char(1)			No



mar_novedades:		Registra los diferentes tipo de novedades como: faltas, atrasos, permisos, etc.				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	Id	Identificador de la novedad	int(11)	PK		No
2	descripcion	Descripción de la novedad.	varchar(40)			No
3	cobrado	Si se cobra la novedad N/S	char(1)			No
4	costo	Costo por la novedad	decimal(6,2)			Si
5	marcacion	Número de la marcación.	int(11)			Si
6	usuario	Usuario que realizó la acción.	varchar(15)			No
7	fechaReg	Fecha que se realizó la acción.	datetime			No
8	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_sucursal:		Registra los datos principales de cada una de las sucursales de la empresa				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	id	Identificador de la sucursal	int(11)	PK		No
2	nombre	Nombre de la sucursal.	varchar(80)			No
3	direccion	Dirección de la sucursal.	varchar(80)			No
4	usuario	Usuario que realizó la acción.	varchar(15)			No
5	fechaReg	Fecha que se realizó la acción.	datetime			No
6	estado	Estado: A = Activo I = Inactivo	char(1)			No

mar_turnos_x_emp:		Registra la asignación de turnos a los empleados				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	secuencia	Secuencia de la tabla.	int(11)	PK		No
2	mtornos	Código del turno	int(11)	FK	mar_mtornos	No
3	empleado	Código del empleado	int(11)	FK	mar_empleado	No
4	costoHora	Costo hora	decimal(5,2)			Si
5	horasExtra	Si considera horas extra S/N	char(1)			Si
6	porHoraExtra	Porcentaje de horas extras	decimal(5,2)			Si
7	costoHoraExtra	Costo hora extra	decimal(5,2)			Si
8	fecha	Fecha de asignación del	date			Si
9	turnoAlterno	Turno alterno	int(11)			Si
10	fecIniAlterno	Fecha de inicio del turno	date			Si
11	fecFinAlterno	Fecha de fin del turno	date			Si
12	usuario	Usuario que realizó la acción	varchar(15)			No
13	fechaReg	Fecha que se realizó la acción	datetime			No
14	estado	Estado: A = Activo I = Inactivo	char(1)			No

seg_grupos:		Registra los diferentes grupos de usuario del sistema				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código del grupo	int(11)	PK		No
2	descripcion	Nombre del grupo.	varchar(40)			No
3	fecha	Fecha/hora que se realizó la acción	datetime			No
4	estado	Estado: A = Activo I = Inactivo	char(1)			No

seg_menu:		Registra los menús del sistema				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código del menú	smallint(6)	PK		No
2	seg_modulos	Código del módulo	smallint(6)	FK	seg_modulos	No
3	codigoPadre	Código del menú padre	smallint(6)	FK	seg_menu	Si
4	nivel	Nivel del menú	smallint(6)			Si
5	descripcion	Nombre del menú.	varchar(40)			No
6	url	Ruta de la página web	varchar(80)			Si
7	imagen	Imagen del menú	varchar(100)			Si
8	fecha	Fecha/hora que se realizó la acción	datetime			No
9	estado	Estado: A = Activo I = Inactivo	char(1)			No

seg_modulos:		Registra los módulos del sistema				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código del módulo	smallint(6)	PK		No
2	descripcion	Nombre del módulo.	varchar(40)			No
3	imagen	Imagen del módulo	varchar(100)			No
4	fecha	Fecha/hora que se realizó la acción	datetime			No
5	estado	Estado: A = Activo I = Inactivo	char(1)			No

seg_opciones:		Registra las opciones del sistema				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código de la opción	smallint(6)	PK		No
2	seg_menu	código del menú	smallint(6)	FK	seg_menu	No
3	descripción	Nombre de la opción	varchar(40)			No
4	nombreMenu	Ruta de la página web	varchar(40)			No
6	url		varchar(60)			No
7	imagen	Imagen del modulo	varchar(100)			Si
8	fecha	Fecha/hora que se realizó la acción	datetime			No
9	estado	Estado: A = Activo I = Inactivo	char(1)			No

seg_opciones_x_gru		Registra las opciones asignadas a los grupos				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	secuencia	Secuencia de la tabla.	int(11)	PK		No
2	seg_grupo	Código del grupo	smallint(6)	FK	seg_grupos	No
3	seg_opciones	Código de la opción	smallint(6)	FK	seg_opciones	No
4	fecha	Fecha/hora que se realizó la acción	datetime			No
5	estado	Estado: A = Activo I = Inactivo	char(1)			No

seg_usuarios:		Registra los usuarios que pueden ingresar al sistema				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	secuencia	Secuencia de la tabla	int(11)	PK		No
2	seg_grupos	Código de grupo	smallint(6)	FK	seg_grupos	No
3	login	Nombre se usuario.	varchar(10)			No
4	nombre	Nombre completo de usuario o departamento.	varchar(40)			No
5	password	Clave de usuario.	varchar(500)			No
6	jefe	Secuencia de usuario jefe	int(11)			Si
7	fecha	Fecha/hora que se realizó la acción	datetime			No
8	estado	Estado: A = Activo I = Inactivo	char(1)			No



tipotumos:		Registra los diferentes tipos de grupos de turnos.				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	id	Identificador del grupo	int(11)	PK		No
2	nombre	Nombre del grupo de turno	varchar(40)			No
3	holguraEntradaS	Holgura para entrada/salida	smallint(6)			Si
4	holguraLunch	Holgura para lunch	smallint(6)			Si
5	usuario	Usuario que realizó la acción	varchar(15)			No
6	fechaReg	Fecha que se realizó la acción	datetime			No
7	estado	Estado: A = Activo I = Inactivo	char(1)			No

sol_tiposmotivos:		Registra los diferentes tipos de motivos para la solicitudes de novedades				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código del tipo e motivo	int(11)	PK		No
2	descripcion	Descripción del motivo	varchar(40)			No
3	usuario	Usuario que realizó la acción	varchar(15)			No
4	fechaReg	Fecha que se realizó la acción	datetime			No
5	estado	Estado: A = Activo I = Inactivo	char(1)			No

sol_motivos:		Registra los diferentes subtipos de motivos de las novedades				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código del motivo	int(11)	PK		No
2	tiposmotivos	Código del tipo de motivo	int(11)			No
3	descripcion	Descripción del motivo	varchar(40)			No
4	cobrado	Si es cobrado S/N	char(1)			Si
5	cargoVacacione	Si es con cargo a	char(1)			Si
6	numeroMarcacio	Número de la marcación	smallint(6)			Si
7	usuario	Usuario que realizó la acción	varchar(15)			No
8	fechaReg	Fecha que se realizó la acción	datetime			No
9	estado	Estado: A = Activo I = Inactivo	char(1)			No

sol_msolicitudes:		Almacena las solicitudes de novedad de los empleados				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	codigo	Código de la solicitud	int(11)	PK	mar_mturnos	No
2	sucursal	Código de la sucursal.	int(11)	FK	mar_sucursal	Si
3	motivos	Código del motivo	int(11)	FK	sol_motivos	No
4	empleado	Código del empleado	int(11)	FK	mar_empleado	No
5	numeroDias	Número de días de la	smallint(6)			No
6	descuento	Descuento por la novedad	decimal(6,2)			No
7	fechaIni	Fecha de inicio de la novedad	date			No
8	horaIni	Hora de inicio de la novedad	smallint(6)			Si
9	minutosIni	Minutos de inicio de la novedad	smallint(6)			Si
10	fechaFin	Fecha de fin de la novedad	date			Si
11	horaFin	Hora de fin de la novedad	smallint(6)			Si
12	minutosFin	Minutos de fin de la novedad	smallint(6)			Si
13	MEE	Afecta marcación de entrada	char(1)			No
14	MSL	Afecta marcación salida de lunch	char(1)			No
15	MRL	Afecta marcación regreso de lunch	char(1)			No
16	MSE	Afecta marcación de salida	char(1)			No
17	observacion	Observaciones de la solicitud	varchar(150)			No



18	usuario	Usuario que realizó la acción	varchar(15)			No
19	fechaReg	Fecha que se realizó la acción	datetime			No
20	estado	Estado: A = Activo I = Inactivo	char(1)			No

gen_supertipo:		Registra los diferentes tipos de clasificaciones utilizados				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	pre	Prefijo del súper tipo	char(3)	PK		No
2	id	Identificador del súper tipo	smallint(6)	PK		No
3	nombre	Nombre general del súper tipo	varchar(30)			No
4	descripcion	Nombre asignado a cada súper tipo	varchar(50)			No
5	usuario	Usuario que realizó la acción	varchar(15)			No
6	fechaReg	Fecha que se realizó la acción	datetime			No
7	estado	Estado: A = Activo I = Inactivo	char(1)			No

hmarcaciones:		Tabla histórica de la tabla mar_marcaciones				
No.	Columna	Descripción	Tipo de dato	Clave	Referencia	Nulo
1	hsecuencia	Secuencia de la tabla	int(11)	PK		No
2	secuencia	Secuencia de la marcación	int(11)			No
3	empleado	Código del empleado	int(11)			No
4	mtornos	Código del turno	int(11)			No
5	cargas	Número de la carga	int(11)			Si
6	periodo	Año del período de marcación	int(11)			No
7	mes	Mes de marcación	int(11)			No
8	día	Día de marcación	int(11)			No
9	fecha	Fecha de marcación	date			No
10	horaEE	Hora de entrada empresa	smallint(6)			No
11	minEE	Minuto de entrada empresa	smallint(6)			No
12	horaPE	Hora de entrada del permiso	smallint(6)			No
13	minPE	Minuto de entrada del permiso	smallint(6)			No
14	horaSL	Hora de salida de lunch	smallint(6)			No
15	minSL	Minuto de salida de lunch	smallint(6)			No
16	horaRL	Hora de retorno de lunch	smallint(6)			No
17	minRL	Minuto de retorno de lunch	smallint(6)			No
18	horaPS	Hora de salida del permiso	smallint(6)			No
19	minPS	Minuto de salida del permiso	smallint(6)			No
20	horaSE	Hora de salida empresa	smallint(6)			No
21	minSE	Minuto de salida empresa	smallint(6)			No

22	totalAT	Total de atrasos	smallint(6)			No
23	totalHE	Total de horas extras	smallint(6)			No
24	novedad	Número de la novedad	int(11)			No
25	observación	Observación de las marcaciones	varchar(100)			Si
26	usuario	Usuario que realizó la acción	varchar(15)			No
27	fechaReg	Fecha que se realizó la acción	datetime			No
28	estado	Estado: A = Activo I = Inactivo	char(1)			No
29	hfechaReg	Fecha que se registra el histórico	datetime			No
30	haccion	Acción porque se registra A = Actualización I = Inserción	char(1)			No

Anexo 2: Entrevistas realizadas a los actores del sistema

FORMULARIO DE ENTREVISTAS A USUARIOS	
FEU 2014HPNG-GYE #0001	

Cliente:	Hiunsa S.A.	Departamento:	Gestión de Talento Humano
Proyecto:	Sistema de Control de Marcaciones		
Fecha:	2014/05/19	Duración:	16H30 – 17H30 (1 hora)
Entrevistador:	Ing. José Caicedo Salazar		
Entrevista a:	Ruth Vanegas	Cargo:	Secretaria

DESARROLLO DE ENTREVISTA

Entrevista realizada a la Sra. Ruth Vanegas, Secretaria de Talento Humano del Hiunsa S.A.

1 ¿Cuáles son las funciones que usted desempeña dentro de la institución?

Yo tengo el cargo de secretaria del Dpto. de Gestión de Talento Humano y me encargo de llevar los registros del personal en cuanto a su hoja de vida, elaboro los diferentes memos que se dirigen al personal, reviso los horarios de los diferentes departamentos, recepto los permisos de salida y acción del personal para que sean aprobados por el Jefe de Talento Humano, elaboro la orden de cuerpo para realizar el control de asistencia del personal y su cumplimiento de horario.

2 ¿Cuentan actualmente con algún tipo de reloj biométrico para el control de asistencia del personal?

Actualmente no. Anteriormente se tenía dos relojes biométricos de huella dactilar, uno en la fábrica y otro en el Edificio Administrativo; pero se dejaron de utilizarlos hace dos años, por cuánto presentaba problemas al tomar la marcación, ya que en ocasiones no tomaba la marcación del personal sobre todo de la fábrica.

3 ¿Cómo realizan actualmente el control de asistencia del personal?

En la garita de entrada se encuentra una hoja donde el personal coloca sus nombres, el departamento, su hora de entrada/salida y su firma. Adicional se realiza por

semana la Orden del Cuerpo para revisar que el personal de cada departamento haya asistido durante el día.

4 ¿Cuentan con informes para revisar la horas de entrada/salida del personal y los atrasos?

No se tiene ningún informe. Cuando se pide un informe de las faltas de un empleado en ese momento se elabora, ya que se debe revisar en las hojas de entrada/salida que firman los empleados en la garita y esto toma tiempo.

5 ¿Se realiza algún descuento al personal por atrasos o faltas?

No se realiza ningún tipo de descuento por los atrasos, ya que estos no se pueden determinar con exactitud por no contar con informes elaborados. En cuanto a las falta se realiza el llamado de atención por memos, cuando se verifican en las hojas de entrada/salida del personal.

6 ¿Cuántos tipos de turnos manejan para el personal?

En la fábrica se tienen turnos rotativos, en el área administrativa son horarios de lunes a viernes, 8 horas diarias (8H00 a 17H00).

Los turnos rotativos son aquellos que tienen horarios diferentes cada día del mes y en general se trata que cada persona trabaje sus 8 horas diarias. Los turnos:

Un día trabaja en la mañana (07H00 – 15H00)

Un día trabaja en la tarde (15H00 – 22H00)

Un día trabaja en la noche (22H00 – 07H00)

Descansa dos días, y así se repite nuevamente.

7 ¿Quién se encarga de elaborar los turnos del personal?

Los elabora cada Jefe de Área en una hoja de Excel de manera mensual.

8 ¿Quién se encarga de revisar y aprobar los turnos?

Cuando los Jefes de Área han elaborado los turnos del personal a su cargo, los imprimen y los envían al Dpto. de Talento Humano para que sean sellados.

9 ¿Qué revisión realiza la Unidad de Talento Humano con los turnos que le llegan de cada Área?

Talento Humano verifica los turnos ya que en ellos viene especificado el personal que está de vacaciones, que tiene licencia por enfermedad, por maternidad, etc.

FORMULARIO DE ENTREVISTAS A USUARIOS

FEU 2014HPNG-GYE #0002

Cliente:	Hiunsa S.A.	Departamento:	Gestión de Talento Humano
Proyecto:	Sistema de Control de Marcaciones		
Fecha:	2014/05/20	Duración:	10H00 – 11H00 (1 hora)
Entrevistador:	Ing. Julio Litardo		
Entrevista a:	Lidia Moreno	Cargo:	Analista

DESARROLLO DE ENTREVISTA

Entrevista realizada a la Lcda. Lidia Moreno, Analista de Talento Humano del Hiunsa S.A.

1 ¿Cuáles son las funciones que usted desempeña dentro de la institución?

Mi cargo es de Analista del Dpto. de Gestión de Talento Humano y mi función principal es la de llevar los registros de los pasivos laborales del personal activo, elaborar y controlar la nómina, tramitar las diferentes solvencias frente al IESS y al Ministerio de Relaciones laborales, descuentos a personal por faltas, etc.

2 ¿Considera usted que a usted ayudaría el tener relojes biométricos para que cada empleado registre su asistencia diariamente?

Obviamente que si, como usted sabrá en la actualidad se realiza un control manual de asistencia lo cual evidentemente no nos brinda la seguridad que cada empleado

llegue y salga a la hora que dicho registro indica, pero mucho más importante sería que toda la información que receipte dicho dispositivo pueda ser descargada mínimo a una hoja de Excel directamente aunque lo ideal sería que alimente directamente al sistema de Nómina de Recursos Humanos y así realizar los descuentos automáticamente.

- 3 ¿Qué tiempo se demora en procesar toda la información de entradas y salidas?**
Algunas veces hemos tenido la necesidad de utilizar hasta 2 recursos para poder procesar todos los registros que se dan durante el mes, en algunos casos la letra del empleado no es muy clara y en otras la firma no es siempre la misma, en muchas ocasiones hemos tenido que consultar al Jefe de planta para que nos certifique a quién le pertenece cada registro.
- 4 ¿Cómo maneja actualmente el tema de los horarios rotativos?**
En la actualidad el Jefe de Planta es el encargado de administrar los turnos de cada empleado de dicha área, él nos envía los turnos de cada uno y nosotros nos encargamos de ubicarlos en los registros manuales.
- 5 Se realiza algún descuento al personal por atrasos o faltas**
No se realiza ningún tipo de descuento por los atrasos, ya que estos no se pueden determinar con exactitud por no contar con informes elaborados. En cuanto a las faltas se realiza el llamado de atención por memos, cuando se verifican en las hojas de entrada/salida del personal.
- 6 ¿Cuántos tipos de turnos manejan para el personal?**
En la fábrica se tienen turnos rotativos, en el área administrativa son horarios de lunes a viernes, 8 horas diarias (8H00 a 17H00).
Los turnos rotativos son aquellos que tienen horarios diferentes cada día del mes y en general se trata que cada persona trabaje sus 8 horas diarias. Los turnos:
Un día trabaja en la mañana (07H00 – 15H00)
Un día trabaja en la tarde (15H00 – 22H00)
Un día trabaja en la noche (22H00 – 07H00)

Descansa dos días, y así se repite nuevamente.

7 ¿Quién se encarga de elaborar los turnos del personal?

Los elabora cada Jefe de Área en una hoja de Excel de manera mensual.

8 ¿Quién se encarga de revisar y aprobar los turnos?

Cuando los Jefes de Área han elaborado los turnos del personal a su cargo, los imprimen y los envían al Dpto. de Talento Humano para que sean sellados.

9 ¿Qué revisión realiza la Unidad de Talento Humano con los turnos que le llegan de cada Área?

Talento Humano verifica los turnos ya que en ellos viene especificado el personal que está de vacaciones, que tiene licencia por enfermedad, por maternidad, etc.; y el personal policial que está franco.

FORMULARIO DE ENTREVISTAS A USUARIOS

FEU 2014HPNG-GYE #0003

Cliente:	Hiunsa S.A.	Departamento:	Gestión de Talento Humano
Proyecto:	Sistema de Control de Marcaciones		
Fecha:	2014/05/21	Duración:	10H00 – 11H00 (1 hora)
Entrevistador:	Ing. José Caicedo		
Entrevista a:	Carlos Perazo Santos.	Cargo:	Jefe

DESARROLLO DE ENTREVISTA

Entrevista realizada al Ing. Carlos Perazo, Jefe de Talento Humano del Hiunsa S.A.

1 ¿Cuáles son las funciones que usted desempeña dentro de la institución?

Mi cargo es Jefe del departamento de Gestión de Talento Humano y una de mis funciones principales es la de supervisar y revisar el proceso de nómina a fin de

garantizar el depósito oportuno de los empleados de la empresa, también el desarrollar iniciativas para la solución de problemas y toma de decisiones.

2 ¿Considera usted que a usted ayudaría el tener relojes biométricos para que cada empleado registre su asistencia diariamente?

Definitivamente que si nos ayudaría y mucho, sólo en procesar la asistencia algunas veces utilizamos más de 1 recurso, tengo entendido que en el pasado se usaron relojes biométricos para que cada empleado se registre utilizando su huella dactilar pero algunas veces no registraban la entrada y/o salida del empleado. Me parece importante indicarles que dichos relojes deberían volcar la información de entradas y salidas automáticamente al sistema de nómina, además que me ayuden con información estadísticas de atrasos y faltas por empleado y departamentos.

3 ¿Cómo llevan actualmente los registros de entradas y salidas?

Actualmente se maneja de manera manual, cada empleado se registra escribiendo sus nombres completos, su número de cédula y su firma, los problemas que esto nos acarrea son:

- Letra ilegible al momento de escribir sus nombres
- No escriben correctamente. sus números de cédulas.
- Rúbricas en lugar de firmas

En incontables ocasiones debemos consultar al Jefe de planta por a qué le pertenece cuál registro, esto definitivamente es un cuello de botella en nuestras actividades diarias.

4 ¿Cómo maneja actualmente el tema de los horarios rotativos?

El Jefe de Planta es el encargado de administrar los turnos de cada empleado de dicha área, él nos envía los turnos de cada uno y nosotros nos encargamos de ubicarlos en los registros manuales.

5 ¿Cómo hace para procesar los descuentos por atrasos y/o faltas?

Como política del departamento de Gestión de Talento Humano en el caso de faltas injustificadas al lugar de trabajo se realiza un llamado de atención escrito y se notifica al respectivo Jefe para que él esté en conocimiento, por el tema de atrasos no

se realizan descuentos ya que como política de nuestro departamento le damos a cada empleado una ventana de 5 minutos de atraso.

6 ¿Quién se encarga de elaborar los turnos del personal?

Cada Jefe de Área elabora los turnos del personal a su cargo en una hoja de Excel de manera mensual.

7 ¿Quién se encarga de revisar y aprobar los turnos?

Cuando los Jefes de Área han elaborado los turnos del personal a su cargo, los imprimen y los envían al Dpto. de Talento Humano para que sean sellados.

8 ¿Qué revisión realiza la Unidad de Gestión de Talento Humano con los turnos que le llegan de cada área?

Talento Humano verifica los turnos ya que en ellos viene especificado el personal que está de vacaciones, que tiene licencia por enfermedad, por maternidad, etc.; y el personal policial que está franco.

9 ¿Qué tipo de estadística le servirían a usted para poder tomar adecuadas decisiones?

En mi experiencia necesito que con la información que el biométrico capture pueda tener los siguientes informes estadísticos:

- Empleados y departamento para en el cuál labora que llega constantemente tarde (todo empleado que excede los 5 minutos permitidos).
- Empleados que no marcan su salida de la oficina o de la planta.
- Faltas de empleados.
- Indicador de ausentismo laboral
- Indicador de atrasos

FORMULARIO DE ENTREVISTAS A USUARIOS

FEU 2014HPNG-GYE #0004

Cliente:	Hiunsa S.A.	Departamento:	Planta
Proyecto:	Sistema de Control de Marcaciones		
Fecha:	2014/05/21	Duración:	10H00 – 11H00 (1 hora)
Entrevistador:	Ing. Julio Litardo		
Entrevista a:	Manuel Gando Lavayen	Cargo:	Jefe de Planta

DESARROLLO DE ENTREVISTA

Entrevista realizada al Ing. Manuel Gando, Jefe de Planta de Hiunsa S.A.

1 ¿Cuáles son las funciones que usted desempeña dentro de la institución?

Mi cargo es el de Jefe de Planta, y las funciones que desempeño en esta compañía son muy diversas, por ejemplo, me encargo de:

- Superviso la línea de producción.
- Realizo la atención a proveedores.
- Cumplir y hacer cumplir el plan de trabajo establecido.
- Implantar y ejecutar las políticas de calidad, medio ambiente y prevención de riesgos laborales.
- Asignar los horarios de cada uno de mis subalternos.
- Etc.

2 ¿Cómo diseña los horarios rotativos de sus subalternos?

Yo los he distribuido de acuerdo a 3 turnos, cada grupo de operarios deben trabajar su turno de 8 horas, ya que la carga de trabajo ha sido diseñada para que cumplan dicho horario, los turnos asignados son:

- Un turno en la mañana (07H00 a 15H00)
- Un turno vespertino (15H00 a 22H00)
- Un turno nocturno (22H00 a 07H00)

Siempre trato de dar a cada empleado 2 días seguidos de descanso sobre todo a los que cubren el turno nocturno, los horarios los elaboro en una hoja de Excel, luego los envío a Gestión de Recursos Humanos para que ellos tengan conocimientos de dichos horarios.

3 ¿Cómo se registran actualmente los empleados de la planta?

Cada empleado debe registrarse al entrar en la garita escribiendo claramente sus nombres completos, número de cédula y firma, mensualmente recibo quejas por parte del personal de Gestión de Recursos Humanos ya que muchos empleados no escriben claramente sus 2 nombres y 2 apellidos o registran sus números de teléfonos móviles o hacen un garabato en lugar de firmas.

4 ¿Cómo notifica actualmente las faltas de su personal al departamento de Gestión de Talento Humano?

Diariamente cada supervisor constata que todos los operarios estén en sus respectivos puestos de trabajo si falta alguno me lo reportan a mí directamente para estar en conocimiento y esperar el respectivo justificativo del empleado. Debo indicar que algunas veces por el sin número de responsabilidades que tengo en la planta día a día no reporto dicha ausencia al departamento de Gestión de Talento Humano.

5 ¿Qué hace el departamento de Gestión de Talento Humano cuándo detecta una ausencia no justificada por usted?

Cuando esto sucede me piden los respectivos soportes (justificativos) por el cual el o los empleados se ausentaron de su lugar de trabajo durante el mes en el que ellos están procesando la nómina.

6 ¿Cree usted que un reloj biométrico ayudaría en el proceso de registro de hora de entrada y salida de su personal?

Ayudaría y mucho, se minimizaría las quejas que yo recibo desde el departamento de Gestión de Talento Humano, ya que pienso que el empleado sólo registraría su entrada por medio de algún código y/o huella dactilar. Además pienso que con eso sabrían a ciencia cierta quienes llegan tarde y quienes faltan constantemente.