

UNIVERSIDAD DE ORIENTE  
NÚCLEO DE ANZOÁTEGUI  
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



PROYECTO DE TRABAJO DE GRADO

**“DESARROLLO DE UNA APLICACIÓN PARA LA AUTOMATIZACIÓN  
DE ALGUNAS ACTIVIDADES LLEVADAS A CABO EN EL ÁREA DE  
PERSONAL, PERTENECIENTE A LA DIVISIÓN DE ADMINISTRACIÓN  
DE LA GERENCIA GENERAL DE TRIBUTOS INTERNOS DE LA REGIÓN  
NOR-ORIENTAL”**

PRESENTADO POR:

Br. Daira del Carmen Hernandez Cabello.

Trabajo de Grado presentado ante la Universidad de Oriente  
como requisito parcial para optar al título de  
INGENIERO EN COMPUTACIÓN

Puerto La Cruz, Agosto de 2010

UNIVERSIDAD DE ORIENTE  
NÚCLEO DE ANZOÁTEGUI  
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



PROYECTO DE TRABAJO DE GRADO

**“DESARROLLO DE UNA APLICACIÓN PARA LA AUTOMATIZACIÓN  
DE ALGUNAS ACTIVIDADES LLEVADAS A CABO EN EL ÁREA DE  
PERSONAL, PERTENECIENTE A LA DIVISIÓN DE ADMINISTRACIÓN  
DE LA GERENCIA GENERAL DE TRIBUTOS INTERNOS DE LA REGIÓN  
NOR-ORIENTAL”**

Asesor:

---

Ing. Víctor Mujica  
Asesor Académico

Puerto La Cruz, Agosto de 2010

UNIVERSIDAD DE ORIENTE  
NÚCLEO DE ANZOÁTEGUI  
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



PROYECTO DE TRABAJO DE GRADO

**“DESARROLLO DE UNA APLICACIÓN PARA LA AUTOMATIZACIÓN  
DE ALGUNAS ACTIVIDADES LLEVADAS A CABO EN EL ÁREA DE  
PERSONAL, PERTENECIENTE A LA DIVISIÓN DE ADMINISTRACIÓN  
DE LA GERENCIA GENERAL DE TRIBUTOS INTERNOS DE LA REGIÓN  
NOR-ORIENTAL”**

Jurado Calificador:

---

Ing. Rhonald Rodriguez

Jurado Principal

---

Ing. Aquiles Torrealba

Jurado Principal

---

Ing. Víctor Mujica

Asesor Académico

Puerto La Cruz, Agosto de 2010



## RESOLUCIÓN

De acuerdo al artículo 41 del reglamento del trabajo de Grado:

“Los trabajos de grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quién deberá participarlo previamente al consejo universitario, para su autorización”.



## RESUMEN

El SENIAT es el Servicio Nacional Integrado de Administración Tributaria, este un organismo del estado central venezolano que se encarga de recolectar los impuestos. Dentro de su departamento de administración se encuentra el área de personal, el cual cuenta con un proceso manual, para llevar a cabo el seguimiento de ciertas actividades que allí se desarrollan, como son las asociadas a el registro de las solicitudes de vacaciones, permisos, reposos entre otras, ocasionado una serie de inconvenientes que originan la ineficiencia de las actividades involucradas. Por lo que surge la necesidad de implantar un software que permita optimizar la gestión desarrollada por el departamento, que permita el control de los procesos antes mencionados. Para lograrlo se utilizó en la elaboración del sistema SIP la metodología RUP y la herramienta UML, el lenguaje de programación de código libre PHP y el Sistema Manejador de Base de Datos MySQL. El sistema SIP, es una herramienta que lleva a cabo actividades relacionadas a la gestión de: Empleados, Pasantes, vacaciones, permisos y Reposos.



## DEDICATORIA

A mi dios todopoderoso, por darme valor de seguir hasta el final y hacer realidad este logro, que muchas veces vi lejano de obtener.

A mis padres, Ramon y Daisy que son y serán mi ejemplo a seguir, quienes en todo momento me brindaron el apoyo y el amor necesario para seguir adelante, mejores padres dios no pudo haberme dado, los AMO!.

A mis hermanos Ramon y Daisy, que siempre tuvieron palabras de aliento y cariño, este logro es de los tres, los QUIERO MUCHISIMO!. y a mi sobrinita bella Fabiana (mi clon), porque eres igual a mi, TE ADORO MI NIÑA!.

A dos personas que lamentablemente no se encuentran físicamente conmigo, MI ABUELA FIOR, abuelita linda desde el cielo me ayudaste a conseguir este objetivo.. y a EFRAIN FRANCESCHI, lastima que no pudiste estar conmigo,tanto que me preguntabas cuando era mi graduación, por eso se que estas feliz por mi!.

A mis tias y tios Mirian, Marilis, Marisol, Lourdes, Clarisa, Tony y Andres que siempre estuvieron pendiente de mi a lo largo de todos estos años..Lo conseguí, los QUIERO!.



## AGRADECIMIENTOS

Agradezco a mi señor DIOS, por no permitir que me rindiera, porque a pesar de todos los obstáculos, siempre entre la oscuridad puso un rayito de luz para guiarme.

A mis padres, por todo el sacrificio que dijeron para que pudiera terminar mi carrera, dios me permita recompensarlos con el mil por ciento de lo que ustedes me han dado, este titulo no es mio es gracias a ustedes.

A mis hermanos, por siempre estar dispuestos a ayudarme y sentir este logro como de ellos mismos..

A mis tias y tios, Mirian, Marilis, Marisol Tony y Andres por brindarme toda su ayuda a lo largo de esta carrera, pude cumplirles..ya soy INGENIERO!.

A la familia Campos Oscaño, mi mama Luisa,el Sr Alejandro, Grcia y Alejandrito, por hacerme parte de su familia, con ustedes condeguí un segundo hogar gracias de verdad.. y en especial a mi AMIGA Y HERMANA Kenia Campos, sin ti no hubiese podido obtener este logro mi amiga muchas gracias de corazón!.

A mis compañeras de estudio, Lennys y Edivanis, por todos los gratos momentos compartidos..y en especial mi AMIGO Jose Angel Sanchez,tantas cosas compartidas, risas y lagrimas y hasta el final me demostraste tu amisted.. michas gracias por ayudarme a ser tu colega..



## ÍNDICE DE CONTENIDO

	<b>Pág.</b>
<b>RESOLUCIÓN</b> .....	IV
<b>RESUMEN</b> .....	V
<b>DEDICATORIA</b> .....	VI
<b>AGRADECIMIENTOS</b> .....	VII
<b>ÍNDICE DE CONTENIDO</b> .....	VIII
<b>ÍNDICE DE FIGURAS</b> .....	XIV
<b>ÍNDICE DE TABLAS</b> .....	XVII
<b>CAPÍTULO I</b> .....	18
<b>PLANTEAMIENTO DEL PROBLEMA</b> .....	18
1.2.1 Objetivo General .....	23
1.2.2 Objetivos Específicos .....	23
<b>CAPITULO II</b> .....	24
<b>MARCO TEORICO</b> .....	24
2.1 Antecedentes .....	24
2.2 Bases Teóricas .....	25
2.2.1 Software .....	25
2.2.2 Ingeniería de Software .....	26
2.2.2.2 Fases de la Ingeniería de Software. ....	26
2.2.3 Software Libre. ....	28
2.2.3.1 Ventajas del Software Libre .....	30
2.2.3.2 Desventajas del software libre .....	31
2.2.4 Proceso Unificado para Desarrollo de Software (RUP) .....	32
2.2.4.1 Proceso Unificado Rational (RUP) .....	32
2.2.4.3 Fases en el ciclo de vida del Proceso Unificado de Desarrollo de Software. ....	35
2.2.4.4 Flujos del Proceso Unificado de Desarrollo de Software. ....	37



2.2.5 Lenguaje Unificado de Modelado (UML)..... 37

    2.2.5.1 Diagramas del Lenguaje Unificado de Modelado. .... 38

2.2.6 Programación orientada a objetos ..... 40

2.2.7 Programación web ..... 41

    2.2.7.1 Programación Web cliente/servidor: ..... 42

        2.2.7.1.1 HTML ..... 43

        2.2.7.1.2 Lenguaje PHP ..... 44

            a) Características de PHP: ..... 44

            b) Ventajas de PHP..... 45

        2.2.7.1.3 Apache. .... 45

    2.2.7.2 Aplicación Web. .... 46

        2.2.7.2.1 Interfaz de Usuario. .... 46

        2.2.7.2.2 Tipos de Interfaces de Usuario: ..... 47

2.2.8 Base de Datos ..... 47

    2.2.8.1 Modelo de Datos. .... 48

        2.2.8.1.1 Tipos de Modelado de Datos. .... 48

    2.2.8.2 Sistema de Gestión de Base de Datos..... 49

        2.2.8.2.1 Objetivos de la Base de Datos ..... 49

        2.2.8.2.2 Ventajas del Sistema de Gestión de Base de Datos..... 50

        2.2.8.2.3 Desventajas del Sistema de Gestión de Base de Datos..... 52

    2.2.8.3 Lenguajes de DBMS..... 54

    2.2.8.4 Lenguaje de consultas estructurado ..... 55

        2.2.8.4.1 SQL..... 55

2.2.9 MySQL ..... 56

    2.2.9.1 características de MySQL..... 56

**CAPITULO III..... 58**

**FASE DE INICIO ..... 58**

**PLANIFICACION DE LA FASE DE INICIO..... 58**

    3.1 Flujo de Trabajo: Requisitos. .... 59



3.1.1	Identificación de los requisitos funcionales.....	60
3.1.2	Modelo del Dominio del sistema.....	61
3.1.2	Clases del dominio .....	62
3.1.3	Riesgos del sistema.....	63
3.1.4	Modelos de casos de usos.....	64
3.1.4.1	Actores del Sistema .....	65
3.1.4.2	Casos de uso del sistema.....	66
3.1.4.3	Descripción de los casos de uso.....	67
3.1.4.4	Diagrama de casos de uso.....	68
3.1.4.5	Descripción detallada de los casos de uso del sistema .....	69
3.1.6	Requisitos No Funcionales. ....	75
3.1.7	Requisitos adicionales. ....	76
3.1.8	Requisitos de la plataforma hardware .....	77
3.1.9	Requisitos de plataforma software. ....	77
3.1.10	Diseño lógico de la interfaz de usuario .....	77
3.2	Flujo de Trabajo: Analisis. ....	78
3.2.1	Modelos de Análisis .....	78
3.2.2	Clases de Análisis.....	78
3.2.2.1	Descripción de las Clases Entidad del Sistema SIP.....	79
3.2.2.2	Descripción de las Clases Interfaz del Sistema SIP.....	80
3.2.2.3	Descripción de las Clases Control del Sistema SIP.....	81
3.2.3	Diagramas de Clases de Analisis.....	82
3.2.3.1	Clase de análisis procesar vacaciones.....	82
3.2.3.2	Clase de análisis administrar empleados. ....	83
3.2.3.3	Clase de análisis configurar clave.....	84
3.2.4	Diagramas de Colaboracion .....	85
3.2.4.1	Diagrama de colaboración Procesar Vacaciones .....	86
3.2.4.2	Diagrama de colaboración Administrar Empleados .....	88
3.2.4.3	Diagrama de colaboración Configurar Clave .....	89



3.2.5 Identificación de los Paquetes de Análisis .....	90
3.3 Flujo de Trabajo: Diseño .....	92
3.3.1 Identificación de Subsistemas .....	92
3.4 Evaluación de la Fase de Inicio .....	93
<b>CAPITULO IV .....</b>	<b>94</b>
<b>FASE DE ELABORACION.....</b>	<b>94</b>
<b>PLANIFICACION DE LA FASE DE ELABORACION.....</b>	<b>94</b>
4.1 Flujo de Trabajo: Requisitos. ....	95
4.2 Flujo de Trabajo: Análisis. ....	95
4.2.1 Análisis de la arquitectura .....	95
4.2.1.1 Identificación de los paquetes de análisis.....	96
4.3 Flujo de Trabajo: Diseño.....	97
4.3.1 Factores que inciden en el diseño de la Interfaz Hombre-Maquina .....	97
4.3.2 Definición de las Zonas de Comunicación entre Usuario y Programa.....	98
4.3.3 Diagrama de Clase de Diseño.....	100
4.3.4 Diagrama de Secuencia .....	101
4.3.4.1 Diagrama de secuencia del caso de uso Procesar Vacaciones.....	101
4.3.4.2 Diagrama de secuencia del caso de uso Administrar Empleados.....	103
4.3.4.3 Diagrama de secuencia del caso de uso Configurar Clave .....	104
4.3.5 Diseño de la Base de Datos .....	105
4.3.6 Diseño de la Arquitectura.....	108
4.3.6.1 Identificación de los subsistemas.....	108
4.3.6.2 Modelo de Despliegue .....	110
4.3.7 Prototipo de la interfaz de usuarios .....	111
4.3.7.1 Diseño Lógico.....	112
4.3.7.2 Diseño Físico .....	112
4.3.8 Esbozo de la Interfaz de Usuario .....	115



4.3.8.1	Diseño de Pantallas .....	115
4.3.8.1.1	Pantalla principal del sistema .....	115
4.3.8.1.2	Pantalla del menú administrador .....	116
4.3.8.1.3	Pantalla del menú empleado .....	117
4.3.8.1.4	Mensaje de datos.....	118
4.5	Evaluacion de la Fase de Elaboracion.....	120
<b>CAPITULO V</b>	.....	122
<b>FASE DE CONSTRUCCION</b>	.....	122
<b>PLANIFICACION DE LA FASE DE CONSTRUCCION</b>	.....	122
5.1	FLUJO DE TRABAJO: IMPLEMENTACION .....	123
5.1.1	Construcción de la Base De Datos .....	123
5.1.2	Modelo de Implementación.....	125
5.1.2.1	Subsistemas de Implementación.....	126
5.1.3	Implementación de la Interfaz de Usuario.....	127
5.1.4	Diagrama de Componentes.....	141
5.1.4.1	Implementación de los Componentes.....	142
5.1.4.1.1	Código Fuente de la Página Principal.....	142
5.1.4.1.2	Código Fuente de la Pagina de Seguridad .....	147
5.1.4.1.3	Código Fuente del Menú Principal .....	151
5.5	Flujo de Trabajo: Prueba .....	170
5.5.1	Pruebas por unidad .....	171
5.5.2	Pruebas de integración:.....	172
5.5.2.1	Pruebas de integración al sistema SIP .....	173
5.4	Evaluacion de la Fase de Construccion.....	178
<b>CAPITULO VI</b>	.....	179
<b>FASE DE TRANSICION</b>	.....	179
<b>PLANIFICACION DE LA FASE DE TRANSICION</b>	.....	179
6.1	Preparacion de la Version Beta .....	180
6.2	Instalacion de la Version Beta.....	180



## ÍNDICE DE CONTENIDO

---

6.3 Evaluacion de la Fase de Transicion .....	181
<b>CONCLUSIONES</b> .....	182
<b>RECOMENDACIONES</b> .....	185
<b>REFERENCIAS BIBLIOGRAFICAS</b> .....	186
<b>METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:</b> .....	189



---

## ÍNDICE DE FIGURAS

	<b>Pag.</b>
<b>Figura 2.1.</b> Casos de uso integran el trabajo. ....	33
<b>Figura 2.2.</b> Evolución de la arquitectura del sistema. ....	34
<b>Figura 2.3.</b> Proceso iterativo e incremental.....	34
<b>Figura 2.4</b> Componentes en la programación Cliente/Servidor.....	42
<b>Figura 3.1</b> Flujos de trabajo requeridos en la Fase de Inicio .....	59
<b>Figura 3.2</b> Modelo de Dominio.....	62
<b>Figura 3.3</b> Modelo General de Casos de Uso del Sistema de Personal SIP.....	68
<b>Figura 3.4</b> Caso de Uso Procesar Vacaciones.....	70
<b>Figura 3.5</b> Caso de Uso Procesar Reposos.....	71
<b>Figura 3.6</b> Caso de Uso Procesar Permisos .....	72
<b>Figura 3.7</b> Caso de Uso Administrar Empleados.....	73
<b>Figura 3.8</b> Caso de Uso Gestionar Pasantes.....	73
<b>Figura 3.9</b> Caso de Uso Configurar Clave .....	74
<b>Figura 3.10</b> Clase de Analisis Procesar Vacaciones .....	83
<b>Figura 3.11</b> Clase de Analisis Administrar Empleados .....	84
<b>Figura 3.12</b> Clase de Analisis Configurar Clave.....	84
<b>Figura 3.13</b> Diagrama de Colaboración Procesar Vacaciones .....	86
<b>Figura 3.14</b> Diagrama de Colaboración Administrar Empleados .....	88
<b>Figura 3.15</b> Diagrama de Colaboración Configurar Clave .....	89
<b>Figura 3.16</b> Paquete de Analisis Vacaciones .....	90
<b>Figura 3.17</b> Paquete de Analisis Permisos.....	91
<b>Figura 3.18</b> Paquete de Analisis Personal.....	91
<b>Figura 3.19</b> Paquete de Analisis Clave .....	92
<b>Figura 3.20</b> Identifiacion de Sbsistemas.....	92
<b>Figura 4.1</b> Flujos de trabajo requeridos en la Fase de Elaboración .....	95



<b>Figura 4.2</b> Diagrama de Paquetes de Análisis General del Sistema .....	96
<b>Figura 4.3</b> Zonas de Comunicación de la Interfaz de Usuario.....	99
<b>Figura 4.5</b> Diagrama de clase de diseño SIP.....	100
<b>Figura 4.5</b> Diagrama de secuencia del caso de uso Procesar Vacaciones.....	101
<b>Figura 4.6</b> Diagrama de secuencia del caso de uso Administrar Empleados.....	103
<b>Figura 4.7</b> Diagrama de secuencia del caso de uso Configurar Clave.....	104
<b>Figura 4.8</b> Modelo Relacional de la Base de Datos .....	107
<b>Figura 4.9</b> Dependencia entre Subsistemas distribuida en capas.....	109
<b>Figura 4.10</b> Diagrama de Despliegue del sistema SIP .....	111
<b>Figura 4.11</b> Diagrama de la Vista Principal del Software.....	113
<b>Figura 4.12</b> Diagrama de la Vista del Contenido de Unidades.....	114
<b>Figura 4.13</b> Diseño de la Interfaz Principal del Sistema.....	116
<b>Figura 4.14</b> Diseño de la Interfaz del Menú Administrador. ....	116
<b>Figura 4.15</b> Diseño de la Interfaz del Menú Empleado .....	117
<b>Figura 4.16</b> Mensaje de Error en la Pantalla Principal del Sistema.....	118
<b>Figura 4.17</b> Dependencias de Traza entre Componentes de Diseño.....	119
<b>Figura 4.18</b> Dependencias de Compilación entre los Componentes.....	119
<b>Figura 5.1</b> Flujos de trabajo requeridos en la Fase de Construcción .....	122
<b>Figura 5.2</b> Subsistemas de implementación a partir de Subsistemas de Diseño .....	126
<b>Figura 5.3</b> Diseño Básico de la Interfaz del Sistema .....	127
<b>Figura 5.4</b> Diseño Básico de la Interfaz del Agregar Usuarios.....	129
<b>Figura 5.5</b> Diseño Básico de la Interfaz del Agregar Usuarios.....	130
<b>Figura 5.6</b> Diseño Básico de la Interfaz del Consultar Usuarios .....	131
<b>Figura 5.7</b> Diseño de la Interfaz del Consultar Usuarios .....	131
<b>Figura 5.8</b> Diseño Básico de la Interfaz del Agregar Pasantes .....	132
<b>Figura 5.9</b> Diseño Básico de la Interfaz del Agregar Pasantes .....	133
<b>Figura 5.9 (a).</b> Diseño Básico de la Interfaz del Consultar Pasantes .....	134
<b>Figura 5.10</b> Diseño de la Interfaz del Consultar Pasantes.....	135



<b>Figura 5.11</b> Diseño de la Interfaz del Agregar Permisos .....	136
<b>Figura 5.12</b> Diseño de la Interfaz del Agregar Permisos .....	136
<b>Figura 5.13</b> Diseño de la Interfaz del Agregar Permisos .....	137
<b>Figura 5.14</b> Diseño de la Interfaz del Modificar Permisos .....	138
<b>Figura 5.15</b> Diseño de la Interfaz del Modificar Permisos .....	138
<b>Figura 5.16</b> Diseño de la Interfaz del Agregar Permisos .....	139
<b>Figura 5.17</b> Diseño de la Interfaz del Agregar Reposos .....	140
<b>Figura 5.18</b> Diseño de la Interfaz del Agregar Reposos .....	140
<b>Figura 5.19</b> Diseño de la Interfaz del Agregar Reposos .....	141
<b>Figura 5.20</b> Diagrama de Componentes del Sistema SIP .....	141
<b>Figura 5.21.</b> Respuesta a acceso exitoso para usuario con privilegio Administrador .....	174
<b>Figura 5.22.</b> Respuesta a acceso exitoso denegado al sistema.....	174
<b>Figura 5.23</b> Interfaz Agregar Usuario .....	175
<b>Figura 5.24.</b> Detalles Inerfaz Agregar Usuario .....	176
<b>Figura 5.25.</b> Inetrfaz Agregar Usuario Registrado.....	176
<b>Figura 5.26.</b> Interfaz Agregar Usuario con campos vacios.....	177
<b>Figura 5.27.</b> Mensaje de Usuario Registrado en la Bse de Dtos.....	178
<b>Figura 6.1</b> Flujos de trabajo requeridos en la Fase de Transición.....	179



## ÍNDICE DE TABLAS

<b>Tabla 3.1</b> Clases de Dominio .....	63
<b>Tabla 3.2</b> Actores del sistema .....	66
<b>Tabla 3.3</b> Descripción de los casos de uso del sistema .....	67
<b>Tabla 3.4</b> Requisitos No Funcionales.....	75
<b>Tabla 4.1</b> Base de Datos SIP (1/2) .....	105
<b>Tabla 4.1</b> Base de Datos SIP (2/2) .....	106
<b>Tabla 5.1</b> Estructura de las tablas de la Base de Datos de SIP (1/3).....	123
<b>Tabla 5.1</b> Estructura de las tablas de la Base de Datos de SIP (2/3).....	124
<b>Tabla 5.1</b> Estructura de las tablas de la Base de Datos de SIP (3/3).....	125
<b>Tabla 5.2</b> Clases de equivalencia para las pruebas de caja negra de registrar usuarios .....	171
<b>Tabla 5.3</b> Caso de prueba de caja negra para registrar usuarios .....	172



## **CAPÍTULO I**

### **PLANTEAMIENTO DEL PROBLEMA**

El Servicio Nacional Integrado de Administración Aduanera y Tributaria, SENIAT, nace en el año 1994, en conformidad al decreto N° 310 de fecha 10 de agosto de 1994 y el artículo 14 del decreto N° 363 de fecha 28 de septiembre del mismo año, con el propósito principal de dirigir y administrar el sistema de los tributos aduaneros e internos en concordancia con la política definida por el ejecutivo nacional y en especial participar y elaborar propuestas para definir la política tributaria y aduanera, su incidencia en el comercio exterior y dictar las directrices para su ejecución conforme a las leyes, reglamentos y demás providencias administrativas. Su estructura orgánica se encuentra organizada en dos niveles: central o normativo y operativo.

El nivel operativo lo conforman las gerencias regionales de tributos internos, los sectores, las unidades, las gerencias de aduanas principales y las aduanas subalternas. La gerencia regional de tributos internos de la región Nor-Oriental a su vez se encuentra integrada por varias divisiones:

- División de Asistencia al Contribuyente
- .División de Administración.
- División de Recaudación.
- División de Fiscalización.
- División de Sumario Administrativo.
- División Jurídico-Tributaria.
- División de Tramitaciones.
- División de Contribuyentes Especiales.



## CAPITULO I

### PLANTEAMIENTO DEL PROBLEMA

---

La División de Administración es la encargada de llevar a cabo muchas funciones dentro de las que se destacan:

- Dirigir, planificar, coordinar, supervisar, ejecutar, controlar y evaluar las actividades relacionadas con la gestión de la División, e impartir las instrucciones para la ejecución de las funciones correspondientes.
- Administrar, ejecutar y controlar el presupuesto de gastos asignados y llevar los registros contables de conformidad con las disposiciones administrativas y fiscales vigentes.
- Velar por adecuados niveles de inventarios, a efectos de garantizar el normal funcionamiento de la gerencia regional.
- Ejecutar los programas de administración del Sistema Profesional de Recursos Humanos en su jurisdicción, velar por el fiel cumplimiento de la normativa legal vigente en materia de recursos humanos e instruir los expedientes necesarios en los casos en que se amerite.
- Operar los sistemas de información que faciliten y permitan lograr los objetivos.
- Velar por el seguimiento de las normas y procedimientos de seguridad de los sistemas de información y protección de las instalaciones y equipos.

Todas estas múltiples ocupaciones son repartidas entre las distintas áreas que conforman la división de administración, y a pesar de que en la actualidad esta institución gubernamental cuenta con diferentes sistemas o aplicaciones que permiten llevar a cabo la mayoría de sus actividades de manera eficiente y efectiva, se ha visto en la necesidad de crear algún procedimiento que permita automatizar específicamente algunas actividades asociadas al área de personal perteneciente a la división antes mencionada.

El área de personal es la encargada del manejo de toda la información referente a los más de 250 empleados con los cuales consta la Gerencia de Tributos Internos



## CAPITULO I PLANTEAMIENTO DEL PROBLEMA

---

Región Nor\_Oriental dentro de los que se encuentran funcionarios fijos, obreros y contratados; además el área de personal lleva también la información vinculada a los pasantes de las diferentes divisiones que conformandicha gerencia. Dentro de esta área se realizan todas las operaciones asociadas a:

- Registro y Control de empleados.
- Solicitud de disfrute y suspensión de vacaciones
- Solicitud de permisos.
- Solicitud de reposos.
- Registro y control de pasantes.

Toda la información correspondiente a algún empleado de la institución se encuentra almacenada en un estante o archivo con más de 250 expedientes, incluyendo tanto empleados fijos, contratados y jubilados, por esta razón a la hora de querer acceder a algún dato específico de algún trabajador se hace bastante tediosa la búsqueda, debido a que es una persona la encargada de hacer este trabajo, además de otras obligaciones vinculadas a su área de trabajo.

Con respecto a las solicitudes de vacaciones, reposos o permisos se realizan de la misma forma actualmente, todos estos procesos son llevados a cabo de forma manual, de la siguiente manera: el funcionario le solicita a su superior inmediato le autorice el disfrute de sus vacaciones o bien sea le apruebe algún permiso o reposo u otra operación, de acceder el supervisor procede a llenar el formato conveniente, luego se dirige al área de personal (División de Administración) en donde de ser aceptado se incorporará dicho formato al expediente del empleado, y luego se archiva en el estante en donde se encuentran almacenados los expedientes de cada uno de los funcionarios.



## CAPITULO I PLANTEAMIENTO DEL PROBLEMA

---

Por último todo el control de los pasantes de la empresa, se encuentra en un libro, en donde al entrar algún pasante nuevo se registran todos sus datos, como por ejemplo: nombre, edad, institución de la que proviene, título a obtener, departamento donde efectuará sus pasantías, duración de las pasantías y nombre del supervisor a quien estará a cargo, lo que trae como consecuencia que al momento del término de las pasantías el pasante deba esperar un tiempo para retirar su constancia, ya que como todos los datos quedan almacenados en el libro no se cuenta con un método que permita por lo menos unos días antes, saber que el pasante ya ha culminado sus pasantías y así poder entregarle su constancia a tiempo, por otra parte si algún departamento solicitase un pasante de una área determinada, no se podría tener acceso rápido a dicha información por todo lo anteriormente expuesto.

Como resultado de no contar con un sistema automatizado que permita buscar, actualizar y manejar de forma rápida y eficiente la información de los funcionarios, a diario se presentan diversos inconvenientes e irregularidades a la hora de realizar alguna acción requerida por parte de algún empleado, ocasionando que el trabajo se torne engorroso, originando asimismo retardo en la respuesta hacia los funcionarios.

En tal sentido se ha planteado como solución: **Desarrollar una aplicación para la automatización de algunas actividades llevadas a cabo en el área de personal, perteneciente a la división de administración de la gerencia general de tributos internos de la región Nor-Oriental**, el cual debe satisfacer principalmente dos objetivos:

- En primer lugar garantizar la seguridad de la información evitando que cualquier persona pueda tener acceso a ella, e impidiendo así alguna alteración de la misma.
- En segundo lugar, evitar pérdida de tiempo al momento de una búsqueda, debido a que en un mismo estante es donde se encuentran



los expedientes de todos los funcionarios.

Para alcanzar los objetivos propuestos, el sistema se desarrollará usando la metodología del Proceso Unificado de Desarrollo de Software, la cual hace posible modelar de una forma clara y sencilla todos los procesos que se automatizarán, así como las relaciones que tendrán los usuarios con el sistema, cabe destacar que el proceso unificado utiliza el lenguaje unificado de modelado UML, para preparar todos los esquemas de un sistema de software y proporcionar así las soluciones apropiadas.

Por último es de importancia señalar que el sistema se llevará a cabo con la utilización de herramientas de software libre, debido a que el Seniat es una institución perteneciente a la administración pública, y por ende ha de cumplir con el Decreto presidencial N° 3390, de gaceta N° 38095 y fecha 28 de diciembre 2004, en el cual se establece que “La Administración Pública Nacional deberá emplear Prioritariamente Software Libre desarrollado con Estándares Abiertos, en todos sus Sistemas, proyectos y Servicios Informáticos”. Se usarán para la elaboración del sistema las herramientas de software libre que más se adapten a los requerimientos del proyecto, luego de haber valuado aquellas con las que se cuenta actualmente. Este proyecto se puede considerar pionero dentro de la gerencia regional de tributos internos región Nor-oriental, ya que constituirá el primero en realizarse aplicando técnicas y herramientas de software libre.



## **1.2 OBJETIVOS**

### **1.2.1 Objetivo General**

- Desarrollar una aplicación para la automatización de algunas actividades llevadas a cabo en el área de personal, perteneciente a la división de administración de la gerencia general de tributos internos de la región Nor-Oriental.

### **1.2.2 Objetivos Específicos**

- Analizar el procedimiento actual empleado para llevar a cabo las actividades desarrolladas dentro del área de personal.
- Determinar los requerimientos necesarios para la elaboración del sistema.
- Diseñar la estructura del sistema, representando los módulos que lo integrarán, así como las bases de datos.
- Realizar la codificación de los módulos anteriormente diseñados.
- Implementar la versión Beta, así como un modelo de prueba para corregir las posibles fallas que pueda presentar la aplicación.
- Elaborar el manual de usuario y mantenimiento, que brinde una visión clara y sencilla del sistema.



## CAPITULO II

### MARCO TEORICO

#### 2.1 Antecedentes

El software a desarrollar es el primero que se implementa dentro del área de personal de la división de administración del Seniat, ya que siempre todas las actividades relacionadas al mismo se han llevado de forma manual, más es en la Universidad de Oriente en donde se han venido desarrollando diversos sistemas para la automatización de toda clase de procesos llevados a cabo dentro de empresas o instituciones, como referencia se tomaron en cuentas los siguientes trabajos que han servido como requisito para optar al título de Ingeniero en Computación o Ingeniero de Sistemas:

- Gonzalez, M y Porras, K. (2006). *“Desarrollo de un software para la automatización de las actividades relacionadas con la nomina de empleados de una alcaldía, adscrita a la gobernación del estado Anzoátegui.* Trabajo de grado no publicado. Universidad de Oriente, Edo de Anzoátegui En este trabajo se elabora un sistema capaz de gestionar las distintas actividades administrativas de la alcaldía de Clarines, Edo Anzoategui, tale scomo salario, jubilación, bonos vacacionales, aguinaldos entre otras, empleando la metodología de Proceso Unificado de Desarrollo de Software.
- Araya J y Velásquez, A, (2006). *“Desarrollo de un sistema de información para el control de nómina en el departamento de recursos humanos de la alcaldía del municipio Arismendi estado Nueva Esparta”.* Trabajo de Grado no publicado. Universidad de Oriente, Edo Anzoátegui En este trabajo se diseño un sistema de información capaz de llevar a cabo actividades



desarrolladas en el departamento de recursos humanos una alcaldía tales como vacaciones, HCM, Seguro Social. El proyecto estuvo basado en el Proceso Unificado de desarrollo de software, la herramienta UML.

- Arcila A y Zacarías, E. (2006). *“Desarrollo de un software como soporte para la automatización de las actividades administrativas que se llevan a cabo en una institución educativa”*. Trabajo de Grado no publicado. Universidad de Oriente, Edo Anzoátegui. En este trabajo se desarrolló un software para la automatización de la información asociada al personal docente, obrero, administrativo y al alumnado de una institución educativa, utilizando el proceso unificado de desarrollo de software como metodología, PHP como herramienta de programación y MySQL como manejador de base de datos.

## 2.2 Bases Teóricas

### 2.2.1 Software

Procedimientos y reglas lógicas escritas en la forma de programas y aplicaciones, que definen el modo de operación de la computadora. Tienen carácter virtual (en contraposición con el hardware) y están almacenadas en los diferentes tipos de memoria de lectura/escritura. Comprende todo tipo de programas, utilidades, aplicaciones, sistemas operativos, drivers que hacen posible que el usuario pueda trabajar con la máquina.

Se pueden diferenciar dos grandes niveles a la hora de abordar que metodología emplearemos en la construcción de programas, estos son: el relativo a los pequeños programas, y el que se refiere al desarrollo de grandes sistemas de software, el primero normalmente lo realizan programadores individuales; y el segundo involucran a un gran equipo de programadores. Lógicamente, entre estos dos grandes



niveles se encuentra toda una gama de situaciones intermedias. La programación a gran escala necesita apoyarse en unos principios sólidos y firmes que faciliten el desarrollo de la actividad de programación. Estos principios los facilita la Ingeniería del Software.

### **2.2.2 Ingeniería de Software.**

La Ingeniería de software designa el conjunto de técnicas destinadas a la producción, más allá de la sola actividad de programación. Forman parte de esta disciplina las y el manejo de proyectos, entre otros campos, propios de la rama más genérica denominada.

La ingeniería de software tiene varios modelos o de desarrollo en los cuales se puede apoyar para la realización de software, de los cuales podemos destacar a éstos por ser los más utilizados y los más completos:

#### **2.2.2.1 Objetivos de la Ingeniería de software**

- Mejorar la calidad de los productos de software
- Aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

#### **2.2.2.2 Fases de la Ingeniería de Software.**

El desarrollo de toda aplicación informática, normalmente se hace cumpliendo un ciclo de desarrollo del software, el cual está dividido en las siguientes fases:



- **Análisis y Especificación de los Requerimientos:** Esta fase consiste en analizar, entender y registrar el problema que el patrocinante esta tratando de resolver. Los requerimientos son una descripción de las necesidades o deseos de un producto. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo, así tenemos que un requerimiento debe ser:
  1. *Necesario:* Si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o proceso.
  2. *Conciso:* Si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
  3. *Completo:* Si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
  4. *Consistente:* Si no es contradictorio con otro requerimiento.
  5. *No ambiguo:* Cuando tiene una sola interpretación.
  
- **Diseño:** Consiste en crear una solución que satisfaga las especificaciones definidas en la fase de análisis. En esta fase se crea la interfaz de usuario, estructura del software.
  
- **Codificación:** Esta fase corresponde con la implementación física de las bases de datos, de los programas (codificación), prueba de unidad o procedimientos separados y prueba de subsistemas o integración.
  
- **Prueba del Sistema:** Los grupos de procedimientos de unidades probadas como elementos separados en la fase previa son comprobados como sistema durante la integración.



- **Instalación:** Corresponde a la fase de puesta en marcha de la aplicación.
- **Mantenimiento del Sistema y Mejoras:** Esta fase se corresponde al mantenimiento y mejoras futuras que deben contemplarse a la hora de implantar una aplicación informática, una vez que el sistema está en funcionamiento.

### 2.2.3 Software Libre.

El Software Libre es una cuestión de libertad, no de precio. Para comprender este concepto, debemos pensar en la acepción de libre como en libertad de expresión y no como en “barra libre de cerveza”.

Con software libre nos referimos a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Nos referimos especialmente a cuatro clases de libertad para los usuarios de software:

- La libertad para ejecutar el programa sea cual sea nuestro propósito.

El ejercicio de esta libertad implica que lo podemos utilizar con cualquier fin, ya sea educativo, cultural, comercial, político, social, etc. Esta libertad deriva de que hay ciertas licencias que restringen el uso del software a un determinado propósito, o que prohíben su uso para determinadas actividades.

- La libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades\_ el acceso al código fuente es condición indispensable para esto.



Se puede *estudiar su funcionamiento* (al tener acceso al código fuente) lo que permite, entre otras cosas: descubrir funciones ocultas, averiguar como realiza determinada tarea, descubrir que otras posibilidades tiene, que es lo que le falta para hacer algo, etc. El *adaptar el programa a mis necesidades* implica que puedo suprimirle partes que no me interesan, agregarle partes que considero importantes, copiarle una parte que realiza una tarea y adicionarla a otro programa, etc.

- La libertad para redistribuir copias y ayudar así a tu vecino.

Se es libre de redistribuir el programa, ya sea gratis o con algún costo, ya sea por email, FTP o en CD, ya sea a una persona o a varias, ya sea a un vecino o a una persona que vive en otro país, etc.

- La libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad – el acceso al código fuente es condición indispensable para esto.

Se tiene la libertad de hacer mejor el programa, o sea que puedo: hacer menores los requerimientos de hardware para funcionar, que tenga mayores prestaciones, que ocupe menos espacio, que tenga menos errores, etc. El poder *liberar las mejoras al publico* quiere decir que si yo le realizo una mejora que permita un requerimiento menor de hardware, o que haga que ocupe menos espacio, soy libre de poder redistribuir ese programa mejorado, o simplemente proponer la mejora en un lugar publico (un foro de noticias, una lista de correo, un sitio Web, un FTP, un canal de Chat).

Software libre es cualquier programa cuyos usuarios gocen de estas libertades. De modo que deberías ser libre de redistribuir copias con o sin modificaciones, de forma gratuita o cobrando por su distribución, a cualquiera y en cualquier lugar. Gozar de esta libertad significa, entre otras cosas, no tener que pedir permiso ni pagar para ello.



### 2.2.3.1 Ventajas del Software Libre

#### 1. Escrutinio Público:

Al ser muchos las personas que tienen acceso al código fuente, eso lleva a un proceso de corrección de errores muy dinámico, no hace falta esperar que el proveedor del software saque una nueva versión.

#### 2. Independencia del proveedor:

- a. Al disponer del código fuente, cualquier persona puede continuar ofreciendo soporte, desarrollo u otro tipo de servicios para el software.
- b. No estamos supeditados a las condiciones del mercado de nuestro proveedor, es decir que si este se va del mercado porque no le conviene y discontinúa el soporte, nosotros podemos contratar a otra persona.

#### 3. Manejo de la Lengua:

- a. Traducción: cualquier persona capacitada puede traducir y adaptar un software libre a cualquier lengua.
- b. Corrección ortográfica y gramatical: una vez traducido el software libre puede presentar errores de este tipo, los cuales pueden ser subsanados con mayor rapidez por una persona capacitada.

#### 4. Mayor seguridad y privacidad:

- a. Los sistemas de almacenamiento y recuperación de la información son públicos. Cualquier persona puede ver y entender como se almacenan los datos en un determinado formato o sistema.
- b. Existe una mayor dificultad para introducir código malicioso como ser: espía (p/ej. capturador de teclas), de control remoto (p/ej. Troyano), de entrada al sistema (p/ej. puerta trasera), etc.



### 5. Garantía de continuidad:

El software libre puede seguir siendo usado aun después de que haya desaparecido la persona que lo elaboro, dado que cualquier técnico informático puede continuar desarrollándolo, mejorándolo o adaptándolo.

### 6. Ahorro en costos:

En cuanto a este tópico debemos distinguir cuatro grandes costos: de adquisición, de implantación (este a su vez se compone de costos de migración y de instalación), de soporte o mantenimiento, y de interoperabilidad. El software libre principalmente disminuye el costo de adquisición ya que al otorgar la libertad de distribuir copias la puedo ejercer con la compra de una sola licencia y no con tantas como computadoras posea (como sucede en la mayoría de los casos de software propietario). Cabe aclarar que también hay una disminución significativa en el costo de soporte, no ocurriendo lo mismo con los costos de implantación y de interoperatividad.

#### 2.2.3.2 Desventajas del software libre

Si observamos la situación actual, es decir la existencia mayoritaria de Software Propietario, tenemos:

1. **Dificultad en el intercambio de archivos:** esto se da mayormente en los documentos de texto (generalmente creados con Microsoft Word), ya que si los queremos abrir con un Software Libre (p/ ej. Open Office o LaTeX) nos da error o se pierden datos. Pero esta claro que si Microsoft Word creara sus documentos con un formato abierto (o público) esto no sucedería.
2. **Mayores costos de implantación e interoperabilidad:** dado que el software constituye "algo nuevo", ello supone afrontar un costo de aprendizaje, de instalación, de migración, de interoperabilidad, etc., cuya cuantía puede verse disminuida por: mayor facilidad en las instalaciones y/o en el uso, uso de



emuladores (p/ej. Si el usuario utiliza Microsoft Windows, la solución sería instalar alguna distribución de GNU/Linux y luego un emulador de Windows, como Wine, VMWare. Terminal X, Win4Lin). Vale aclarar que el costo de migración está referido al software, ya que en lo que hace a Hardware generalmente el Software Libre no posee mayores requerimientos que el Software Propietario.

#### **2.2.4 Proceso Unificado para Desarrollo de Software (RUP)**

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidos. Además, el Proceso Unificado utiliza el UML para expresar gráficamente todos los esquemas de un sistema.

El Proceso Unificado ha adoptado un enfoque que se caracteriza por:

- Interacción con el usuario desde un inicio.
- Mitigación de riesgos antes de que ocurran.
- Liberaciones frecuentes.
- Aseguramiento de la calidad.
- Involucramiento del equipo en todas las decisiones del proyecto.
- Anticiparse al cambio de requerimientos.

##### **2.2.4.1 Proceso Unificado Rational (RUP)**

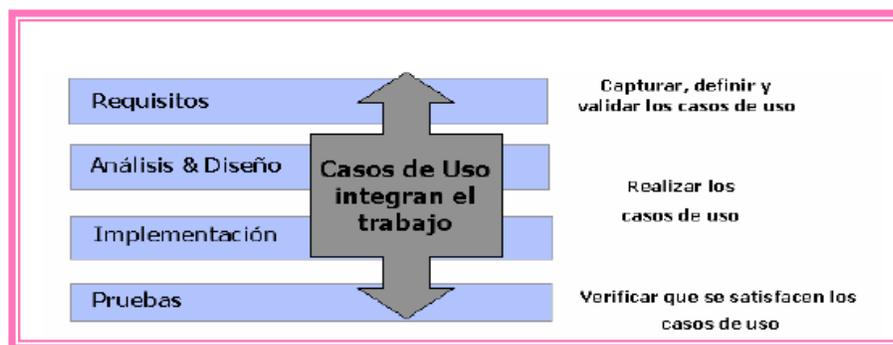
El RUP es un producto de Rational. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en

un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

#### 2.2.4.2 Características del Proceso Unificado de Desarrollo de Software

**Dirigido por casos de uso:** Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.

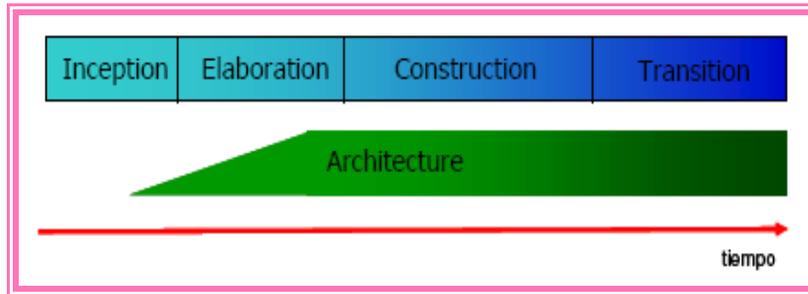


**Figura 2.1.** Casos de uso integran el trabajo.

**Fuente:** Booch, G., Rumbaugh, J. y Jacobson, I. (2000).

**Centrado en la arquitectura:** En la arquitectura de la construcción, antes de construir un edificio éste se contempla desde varios puntos de vista: estructura, conducciones eléctricas, fontanería, etc. Cada uno de estos aspectos está representado por un gráfico

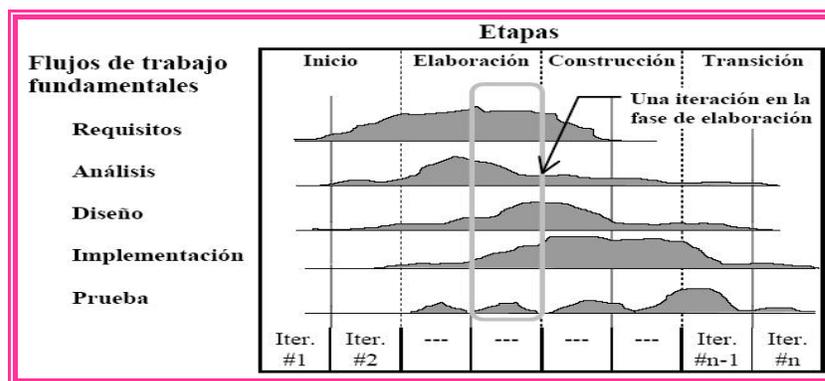
con su notación correspondiente. Siguiendo este ejemplo, el concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.



**Figura 2.2.** Evolución de la arquitectura del sistema.

**Fuente:** Booch, G., Rumbaugh, J. y Jacobson, I. (2000).

**Iterativo e incremental:** Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias fases. Actualmente se suele hablar de ciclos de vida en los que se realizan varios recorridos por todas las fases. Cada recorrido por las fases se denomina iteración en el proyecto en la que se realizan varios tipos de trabajo (denominados flujos). Además, cada iteración parte de la anterior incrementado o revisando la funcionalidad implementada. Se suele denominar proceso.



**Figura 2.3.** Proceso iterativo e incremental.

**Fuente:** Booch, G., Rumbaugh, J. y Jacobson, I. (2000).



### 2.2.4.3 Fases en el ciclo de vida del Proceso Unificado de Desarrollo de Software.

**Fase de inicio:** Su objetivo principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto. En esta fase se identifican y priorizan los riesgos más importantes.

Puede que todo el trabajo físico realizado en esta fase sea descartado. Lo único que normalmente sobrevive a la fase de inicio es el incremento del conocimiento en el sistema a desarrollar.

Los artefactos que típicamente sobreviven a esta fase son:

- Un enunciado de los mayores requerimientos planteados generalmente como casos de uso.
- Un boceto inicial de la arquitectura.
- Una descripción de los objetivos del proyecto.
- Una versión muy preliminar del plan del proyecto.
- Un modelo del negocio.

**Fase de elaboración:** Básicamente su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

El resultado de esta fase es la línea base de la arquitectura. En esta fase se construyen típicamente los siguientes artefactos:

- El cuerpo básico del software en la forma de un prototipo arquitectural.
- Casos de prueba



- La mayoría de los casos de uso (80%) que describen la funcionalidad del sistema.
- Un plan detallado para las siguientes iteraciones

**Fase de Construcción:** Tiene como función alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta. La línea base de la arquitectura crece hasta convertirse en el sistema completo.

Al final de esta fase, el producto contiene todos los casos de uso implementados, sin embargo puede que no este libre de defectos.

Los artefactos producidos durante esta fase son:

- El sistema software.
- Los casos de prueba.
- Los manuales de usuario.

**Fase de Transición:** Tiene como propósito final realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos. Las iteraciones en esta fase continúan agregando características al software. Sin embargo las características se agregan a un sistema que el usuario se encuentra utilizando activamente.

Los artefactos construidos en esta fase son los mismos que en la fase de construcción. El equipo se encuentra ocupado fundamentalmente en corregir y extender la funcionalidad del sistema desarrollado en la fase anterior.



#### 2.2.4.4 Flujos del Proceso Unificado de Desarrollo de Software.

1. **Requisitos:** Flujo de trabajo fundamental cuyo propósito esencial es orientado al desarrollado hacia el sistema correcto. Esto se lleva a cabo mediante la descripción de los requisitos del sistema de forma tal que se pueda llegar a un acuerdo entre el cliente (incluyendo los usuarios) y los desarrolladores del sistema, acerca de lo que el sistema debe hacer y lo que no.
2. **Análisis:** Flujos de trabajo fundamental cuyo propósito principal es analizar los requisitos descritos en la captura de requisitos, mediante su refinamiento y estructuración. El objetivo de esto es lograr una comprensión mas precisa de los requisitos, y obtener una descripción de los requisitos que sea fácil de mantener y que nos ayude a dar estructura al sistema en su conjunto incluyendo su arquitectura.
3. **Diseño:** Flujo de trabajo fundamental cuyo propósito principal es la de formular modelos que se centran en los requisitos no funcionales y el dominio de la solución y que prepara para la implementación y pruebas del sistema.
4. **Implementación:** Flujo de trabajo fundamental cuyo propósito esencial es implementar el sistema en términos de componentes, es decir código fuente guiones, ficheros binarios, ejecutables, et.
5. **Prueba:** Flujo de trabajo fundamental cuyo propósito esencial es comprobar el resultado de la implementación mediante las pruebas de cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema que van a ser entregadas a terceras personas.

#### 2.2.5 Lenguaje Unificado de Modelado (UML).

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los



modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

**Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

**Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.

UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos). UML es además un método formal de modelado.

#### **2.2.5.1 Diagramas del Lenguaje Unificado de Modelado.**

**Diagrama de Casos de Usos:** Representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. En la figura 3 se muestra un ejemplo de casos de uso, donde se muestran tres actores (los clientes, los taquilleros y los jefes de taquilla) y las operaciones que pueden realizar (sus roles).

**Diagrama de Clases:** Muestra un conjunto de clases, interfaces y sus relaciones. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos.



**Diagrama de Secuencia:** Muestra la interacción de los objetos que componen un sistema de forma temporal.

**Diagrama de Clases de Análisis:** Es utilizado por los desarrolladores de software para especificar los requerimientos funcionales, considerando una o varias clases, o subsistemas del sistema a desarrollar. Es importante notar que sólo se consideran clases que están en el dominio del problema.

**Diagrama de Componentes:** Representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Un módulo de software se puede representar como componente. Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución; otros en varias de éstas. Un componente de sólo compilación es aquel que es significativo únicamente en tiempo de compilación. Un componente ejecutable es un programa ejecutable. Un diagrama de componentes tiene sólo una versión con descriptores, no tiene versión con instancias. Para mostrar las instancias de los componentes se debe usar un diagrama de despliegue.

**Diagrama de Despliegue:** Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

**Diagrama de Colaboración:** Muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a



la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos (cuáles objetos son atributos, cuáles temporales, etc.) y ciclos en la ejecución. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

### 2.2.6 Programación orientada a objetos

La programación orientada a objetos se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

Ventajas de un lenguaje orientado a objetos:

- Fomenta la reutilización y extensión del código.
- Relacionar el sistema al mundo real.
- Facilita la creación de programas visuales.
- Construcción de prototipos.
- Agiliza el desarrollo de software.
- Facilita el mantenimiento del software.

*Objetos:* Es un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización. Un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados.

*Las Clases:* Las clases son declaraciones de objetos, son abstracciones de objetos, es decir que la definición de un objeto es la clase. Cuando programamos un objeto y definimos sus características y funcionalidades en realidad lo que estamos haciendo es programar una clase.



*Herencia:* Sirve para crear objetos que incorporen propiedades y métodos de otros objetos. Así podremos construir unos objetos a partir de otros sin tener que reescribirlo todo.

*Mensaje:* Un mensaje en un objeto es la acción de efectuar una llamada a un método.

*Encapsulamiento y Ocultación:* Cada objeto es una estructura compleja en cuyo interior hay datos y programas, todos ellos relacionados entre sí, como si estuvieran encerrados conjuntamente en una cápsula. Esta propiedad se denomina *encapsulamiento*.

Los objetos son inaccesibles, e impiden que otros objetos, los usuarios, o incluso los programadores conozcan cómo está distribuida la información o qué información hay disponible. Esta propiedad de los objetos se denomina *ocultación de la información*.

### **2.2.7 Programación web**

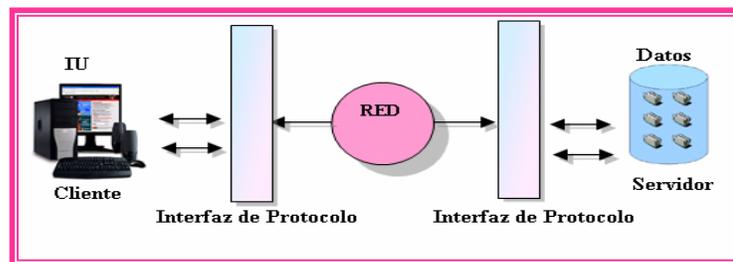
Para transmitir información se requiere un medio de comunicación. Uno de los modos de comunicación que usan los computadores es conectarse a Internet o a la WWW. La conversión Estándar que se sigue para esta comunicación es el Hyper Text Transfer Protocol (http Protocolo de Transmisión de Hipertexto) En http la información es enviada y recibida en forma de solicitudes y respuesta.

La información en Internet o Word Wide Web (WWW) esta guardada en forma de documentos, generalmente como páginas Web, estas páginas están almacenadas en una ubicación central denominada servidor para que múltiples

usuarios (clientes Web) acceden y comparten información, las ventanas de acceso a Internet se denomina navegador Web.

### 2.2.7.1 Programación Web cliente/servidor:

En la figura 2.1 explica los diversos componentes presentes en el modelo de programación Cliente/Servidor



**Figura 2.4** Componentes en la programación Cliente/Servidor

Fuente: Elaboracion Propia

**Cliente:** Actúa como representante del usuario enviando la petición al servidor, y como agente receptor recibiendo la respuesta del servidor, el cliente proporciona la interfaz para que los usuarios ingresen los valores solicitados, establece el enlace de comunicación con el servidor, recibe la respuesta del servidor y la presenta a los usuarios.

**Servidor:** Es un programa que responde la consulta del cliente. Toma los detalles solicitados por el usuario en la base de datos.

**Datos:** La transferencia de datos del servidor al cliente se denomina respuesta. Esta respuesta generalmente esta en forma de texto que representa algún lenguaje de marcado.



**Protocolo:** Es un conjunto de convenciones seguidas para una efectiva comunicación peer-to-peer entre el cliente y el servidor se logra a través de HTTP.

#### 2.2.7.1.1 HTML

El HTML (Lenguaje de Marcado de Hipertexto) fue desarrollado para que los documentos WWW incluyan texto, imágenes, tablas, hiperenlaces y archivos animados. Los documentos Web están disponibles en el servidor como archivos HTML.

Un Archivo HTML tendrá extensión .html o .htm dado que el navegador Web lee el archivo HTML por medio de etiquetas HTML específicas, la pagina Web mostrada como salida al usuario puede variar de un navegador a otro. Cualquier código HTML comprende dos componentes: Etiquetas y Atributos.

*Las Etiquetas:* Deciden la naturaleza del formato que se va a aplicar a los documentos HTML, las etiquetas tienen un conjunto de atributos posibles que deciden la extensión y el estilo del formato que se va a aplicar.

*Atributos:* Se utilizan para manejar requerimientos de los formatos El formato de texto puede incluir elementos como alineación, ancho, tamaño etc.

Beneficios de HTML:

- Se puede compartir gran cantidad de datos alrededor del mundo.
- El HTML es básicamente formato ASCII, por lo que la posibilidad de corrupción de datos por la red es mucho menor.

Dado que es un lenguaje de marcado (que tiene un conjunto de etiquetas), es fácil de desarrollar y simple para comprender.



### 2.2.7.1.2 Lenguaje PHP

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, ciclos (bucles), funciones. No es un lenguaje de marcado como podría ser HTML, XML o WML. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

#### a) Características de PHP:

1.- Se distingue de los lenguajes de Script del lado del cliente, porque el código es ejecutado en el servidor y solo la respuesta se envía al cliente

2.- El termino embebido quiere decir que se puede incorporar código PHP directamente dentro de código HTML utilizando las etiquetas especiales de inicio y fin `<?php...?>` ó `<?.?>`

3.- El hecho de que trabaje en plataformas cruzadas es que se puede utilizar en plataformas bajo distintos sistemas operativos (Unix, Linux, Windows (98, NT, 2000 y XP) OS/2) y servidores Web (Apache, Microsoft, Netscape, Xitame y otros).



4.- Por ser código abierto significa que su distribución es libre.

### **b) Ventajas de PHP**

- Utilizamos PHP cuando deseamos que un sitio Web posea dinamismo, ya que nos permite realizar operaciones para interactuar con los usuarios.
- PHP fue diseñado para trabajar en entorno Web, lo que permite que funcione de forma más rápida y eficiente con HTML.
- PHP fue pensado desde sus inicios como un lenguaje que se pudiera incrustar directamente en HTML para desarrollar sitios Web dinámicos y aplicaciones basadas en la Web, por lo que ofrece un mejor tiempo de respuesta, mayor seguridad y transparencias para el usuario final en entorno Web.
- Una gran ventaja es que PHP no está vinculado a ningún sistema operativo ni servidor Web.

#### **2.2.7.1.3 Apache.**

Un servidor HTTP (Web) con código libre de distribución introducido en 1995 por el grupo apache como una extensión y también una mejora del temprano HTTPd (versión 1.3) del centro nacional de aplicaciones para supercomputadoras. El servidor Apache es popular en los sistemas basados en UNIX, incluyendo Linux, y también corre sobre Windows y otros sistemas operativos como BeOS. A causa de estar basado en código existente con una serie de parches, fue conocido regularmente como “un servidor irregular (poco uniforme)” (patchy en inglés) lo cual llevó al nombre oficial de Apache.



### 2.2.7.2 Aplicación Web.

Una aplicación Web es una aplicación desarrollada utilizando tecnologías basadas en el entorno Web como HTML, XML, JavaScript, PHP, ASP etc. Una *aplicación Web* es un conjunto de páginas que interactúan unas con otras y con diversos recursos en un servidor Web, incluidas bases de datos. Esta interacción permite implementar características en su sitio. Adicionalmente podrá realizar consultas a bases de datos, registrar e ingresar información, solicitudes, pedidos y múltiples tipos de información en línea en tiempo real.

Las funcionalidades de los componentes de una aplicación Web son las siguientes:

*Navegador Web*: este es el componente del lado del cliente, es una aplicación usada para solicitar páginas Web, recuperarlas y mostrar a los usuarios la información solicitada ya procesada.

*Servidor Web*: Acepta las solicitudes por páginas Web hechas por el cliente y las entrega, puede aceptar solicitudes por scripts en el lado del servidor que generan páginas Web dinámicas, por ejemplo Scripts PHP

*Recursos Externos*: Las aplicaciones Web necesitan acceder a la información o solicitar servicios a más de una fuente. Un Sistema de Administración de Base de Datos Relacional (RDBMS) por ejemplo y cualquier otro recurso externo al servidor entran en esta categoría.

#### 2.2.7.2.1 Interfaz de Usuario.

Es un conjunto de elementos hardware y software de una computadora que presentan información al usuario y le permiten interactuar con la información y con la



computadora. El usuario dirige el funcionamiento de la máquina mediante instrucciones, denominadas genéricamente entradas.

Sus principales funciones son:

- Manipulación de archivos y directorios.
- Herramientas de desarrollo de aplicaciones.
- Comunicación con otros sistemas.
- Información de estado.
- Configuración de la propia interfaz y entorno.
- Intercambio de datos entre aplicaciones.
- Control de acceso.
- Sistema de ayuda interactivo.

#### **2.2.7.2.2 Tipos de Interfaces de Usuario:**

*Una interfaz de hardware:* Conjunto de dispositivos que permiten la interacción hombre-máquina, de modo que permiten ingresar, procesar y entregar los datos del ordenador: teclado, ratón y pantalla visualizadora.

*Una interfaz de software:* Destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario observa habitualmente en la pantalla.

#### **2.2.8 Base de Datos**

Un sistema de bases de datos es una colección de archivos interrelacionados y un conjunto de programas que permitan a los usuarios acceder y modificar estos archivos. Uno de los propósitos principales de un sistema de bases de datos es



proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

### 2.2.8.1 Modelo de Datos.

Es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos. Es formal pues los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores definidos en el sistema, independientemente de lo que estos objetos y operadores puedan significar.

#### 2.2.8.1.1 Tipos de Modelado de Datos.

**Modelo Entidad-Relación:** El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de otros objetos. Las entidades se describen en una base de datos mediante un conjunto de atributos. Una relación es una asociación entre varias entidades.

**Modelo Relacional:** En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único. El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada



tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos o atributos.

**Modelo Jerárquico:** Es un tipo de modelo de red con algunas restricciones. De nuevo los datos se representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos. Sin embargo, en el modelo jerárquico cada nodo puede tener un solo padre. Una base de datos jerárquica puede representarse mediante un árbol: los registros son los nodos, también denominados segmentos, y los arcos son los conjuntos. El SGBD jerárquico más importante es el sistema IMS.

**Modelo de Red:** En este modelo los datos se representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos, que son punteros en la implementación física. Los registros se organizan como un grafo: los registros son los nodos y los arcos son los conjuntos. El SGBD de red más popular es el sistema IDMS.

#### **2.2.8.2 Sistema de Gestión de Base de Datos.**

Los SGBD son paquetes de software muy complejo y sofisticado que deben proporcionar los servicios comentados en la sección anterior. No se puede generalizar sobre los elementos que componen un SGBD ya que varían mucho unos de otros. Sin embargo, es muy útil conocer sus componentes y cómo se relacionan cuando se trata de comprender lo que es un sistema de bases de datos.

##### **2.2.8.2.1 Objetivos de la Base de Datos**

En un ambiente multiusuario el SGBD ofrece a la empresa un control centralizado de su información. Los objetivos que se plantean estos sistemas están relacionados



con la intención de evitar los problemas que existían en los sistemas de información orientados a los procesos. Los principales objetivos son:

- Evitar la redundancia de los datos, eliminando así la inconsistencia de los mismos.
- Mejorar los mecanismos de seguridad de los datos y la privacidad. Podemos distinguir cuatro tipos de contextos para usar mecanismos de seguridad: seguridad contra accesos indebidos a los datos, seguridad contra accesos no autorizados a la BD, seguridad contra destrucción causada por el entorno (fuego, inundación, robo,...), seguridad contra fallos del propio sistema (fallos del hardware, del software,...).
- Asegurar la independencia de los programas y los datos, es decir, la posibilidad de modificar la estructura de la base de datos (esquema) sin necesidad de modificar los programas de las aplicaciones que manejan esos datos.
- Mantener la integridad de los datos realizando las validaciones necesarias cuando se realicen modificaciones en la base de datos.
- Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas imprevistas.
- 

#### **2.2.8.2.2 Ventajas del Sistema de Gestión de Base de Datos.**

- *Mejora en la integridad de datos.* La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.



- *Mejora en la seguridad.* La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros. Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones, de modo que un usuario puede estar autorizado a consultar ciertos datos pero no a actualizarlos, por ejemplo.
- *Mejora en la accesibilidad a los datos.* Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.
- *Mejora en la productividad.* El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación. El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel. Muchos SGBD también proporcionan un entorno de cuarta generación consistente en un conjunto de herramientas que simplifican, en gran medida, el desarrollo de las aplicaciones que acceden a la base de datos. Gracias a estas herramientas, el programador puede ofrecer una mayor productividad en un tiempo menor.
- *Mejora en el mantenimiento gracias a la independencia de datos.* En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan. Esto hace que los programas sean



dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados. Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- *Aumento de la concurrencia.* En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o, incluso, que se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.
- *Mejora en los servicios de copias de seguridad y de recuperación ante fallos.* Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

#### 2.2.8.2.3 Desventajas del Sistema de Gestión de Base de Datos.

- **Tamaño.** Al proveer todas las ventajas anteriormente nombradas, el Sistema de Manejo de Base de Datos (DBMS) requiere de mucho espacio en disco duro y



también requiere de mucha memoria principal (RAM) para poder correr adecuadamente.

- **Complejidad.** Debido a la cantidad de operaciones y a las capacidades del DBMS, se convierte en un producto complejo de entender. Esto requiere que los programadores y los analistas deben tomar cursos que los adiestren para poder comprender las capacidades y limitaciones del DBMS
- **Costo** - Los productos de Bases de Datos (Oracle, DB2, Etc.) son productos caros. Esto sin contar los adiestramientos del personal del centro de cómputos y de los usuarios. Ejemplo: Explicar el costo envuelto en implantar FRS en la Universidad.
- **Requerimientos adicionales de Equipo** - El adquirir un producto de Base de Datos, requiere a su vez adquirir equipo adicional para poder correr ese producto como por ejemplo, servidores, memoria, discos duros, etc. Si se pretende correr la Base de Datos con el mínimo de requerimientos, esta posiblemente se degrada y el "response time" al usuario se va a afectar grandemente.
- **En caso de falla, el impacto es mayor.** Si un componente de la Base de Datos sufre un desperfecto, se detiene las operaciones del producto por completo. En el caso de un ambiente no centralizado en Cobol, el impacto casi siempre afecta al departamento en donde ocurrió el problema únicamente.
- **Complejo el recuperar los datos.** En caso de un accidente que corrompa la Base de datos, el proceso de recuperación y de devolver a la Base de Datos su estado anterior al problema, es mucho mas complejo de ejecutar que en sistemas tradicionales.



### 2.2.8.3 Lenguajes de DBMS

En la estructura básica de un Sistema Manejador de Base de Datos se enuncian dos lenguajes que permiten trabajar sobre la base de datos. Estos lenguajes estándar son:

*DDL (Data Definition language) Lenguaje de Definición de Datos:* Por medio de este el DBMS identifica las descripciones de los elementos de los esquemas y almacena la descripción del esquema en el catálogo del DBMS. Es decir se especifica el esquema conceptual e interno (Base de datos Almacenada).

*SDL (Store Definition language) Lenguaje de definición de almacenamiento:* Es utilizado por el DBMS para especificar el esquema interno que corresponde a la Base de Datos Almacenada.

*VDL (View Definition language) Lenguaje de Definición de Vistas:* Es utilizado por el DBMS para especificar las vistas del usuario y sus correspondencias con el esquema conceptual.

En las Bases de Datos Relacionales, el SQL, representa una combinación de los anteriores.

*DML (Data Manipulation language) Lenguaje de Manipulación de Datos:* Permite la manipulación de las operaciones de Inserción, Eliminación y Modificación.



#### 2.2.8.4 Lenguaje de consultas estructurado

El lenguaje de consulta estructurado es un sublenguaje de base de datos utilizado para la consulta, actualización y administración de bases de datos relacionales, el estándar de facto para los productos de bases de datos.

##### 2.2.8.4.1 SQL

El SQL (Structured query language), lenguaje de consulta estructurado, es un lenguaje surgido de un proyecto de investigación de IBM para el acceso a bases de datos relacionales. Actualmente se ha convertido en un estándar de lenguaje de bases de datos, y la mayoría de los sistemas de bases de datos lo soportan, desde sistemas para ordenadores personales, hasta grandes ordenadores.

- **Características Generales.**

1. Una sentencia SQL es como una frase (escrita en inglés) con la que decimos lo que queremos obtener y de donde obtenerlo.
2. Todas las sentencias empiezan con un verbo (palabra reservada que indica la acción a realizar), seguido del resto de cláusulas, algunas obligatorias y otras opcionales que completan la frase. Todas las sentencias siguen una sintaxis para que se puedan ejecutar correctamente, para describir esa sintaxis utilizaremos un diagrama sintáctico como el que se muestra a continuación.
3. Las palabras que aparecen en mayúsculas son palabras reservadas se tienen que poner tal cual y no se pueden utilizar para otro fin, por ejemplo, en el diagrama de la figura tenemos las palabras reservadas SELECT, ALL, DISTINCT, FROM, WHERE.



4. Las palabras en minúsculas son variables que el usuario deberá sustituir por un dato concreto. En el diagrama tenemos nbcolumna, expresión-tabla y condición-de-búsqueda.
5. Una sentencia válida se construye siguiendo la línea a través del diagrama hasta el punto que marca el final. Las líneas se siguen de izquierda a derecha y de arriba abajo. Cuando se quiere alterar el orden normal se indica con una flecha.

### 2.2.9 MySQL

MySQL es un gestor de base de datos sencillo de usar y increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

#### 2.2.9.1 características de MySQL.

- Es un gestor de base de datos, una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional, una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- Es Open Source, el código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.



- Es una base de datos muy rápida, segura y fácil de usar, gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.
  
- Existe una gran cantidad de software que la usa.

## **CAPITULO III**

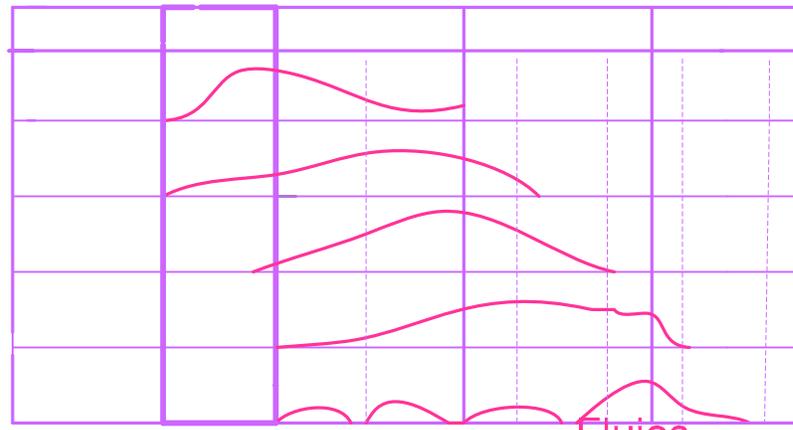
### **FASE DE INICIO**

#### **PLANIFICACION DE LA FASE DE INICIO**

En este capítulo se describe la fase de inicio para llevar a cabo el desarrollo el Sistema de Personal (SIP), siguiendo la metodología del proceso unificado de desarrollo de software. Durante esta fase se define el modelo del negocio, basándose en los casos de uso que definirán el entorno del sistema.

En el desarrollo de esta primera fase, se considero realizar una sola iteración, en la cual se dispondrá del uso de tres flujos de trabajo, requisitos, análisis y parte de diseño, esto gracias a que el proceso unificado de desarrollo de software se hace flexible y se adapta a las necesidades del sistema a desarrollar lo que le brinda al diseñador el poder escoger el numero de iteraciones, así como la manejo de los flujos de trabajo mas adecuados en cada fase, actividades como enumerar requisitos candidatos, entender el contexto del sistema, capturar requisitos funcionales entre otros.

En la captura de requisitos se recopilara información acerca del área de personal de la división de administración a través de entrevistas y consultas a los funcionarios pertenecientes al área, además se hará uso de la técnica de observación para obtener información acerca de cómo se llevan actualmente los procedimientos en dicha área. De toda la investigación realizada anteriormente, cuyo propósito esencial es determinar las condiciones y capacidades que el sistema debe cumplir, se determinan inicialmente los casos de uso del sistema que guiaran hacia un modelo inicial de su arquitectura.



Flujos Inicio Elaboración

**Figura 3.1** Flujos de trabajo requeridos en la Fase de Inicio

Fuente: Elaboración Propia

## Requisitos

El objetivo de la fase de inicio es la puesta en marcha del proyecto, en donde se describe una propuesta de arquitectura, se identifican los riesgos críticos y se construye un prototipo que muestre que se pueden solucionar los problemas del cliente y de los usuarios finales.

### 3.1 Flujo de Trabajo: Requisitos.

El objetivo principal de este flujo de trabajo es lograr total claridad sobre lo que el usuario desea y la forma en la cual se le va a presentar la solución que se esta buscando.

Mediante la descripción de los requisitos es que se hace posible guiar el desarrollo hacia el sistema correcto. Inicialmente se realiza un modelo de dominio simple, con el fin de comprender de una forma clara el contexto del sistema a desarrollar, luego se plantearan una serie de posibles riesgos que se puedan presentar y obstaculizar el buen funcionamiento de la aplicación. Posteriormente se procederá a realizar el modelo de caso de uso general del sistema, partiendo de la captura de los



requisitos funcionales antes descritos, identificando los actores que harán parte de así como comenzar con el diseño de la interfaz.

### **3.1.1 Identificación de los requisitos funcionales.**

Los requisitos candidatos son un conjunto de ideas donde se percibe un bosquejo general del software que se quiere desarrollar, estos se originan según las necesidades de cada usuario y aparecen durante la vida de un sistema.

A continuación se presenta una serie de requisitos con los cuales debe cumplir el sistema de personal a desarrollar:

- Control de las solicitudes de vacaciones de los funcionarios de la empresa, implantando formatos de búsqueda por división, cedula y numero de solicitud de las vacaciones .La búsqueda de Los diferentes tipos de vacación se realizara por fecha, estableciendo mes y año correspondiente.
- Control de las solicitudes de permisos por parte de los empleados de la empresa, estableciendo búsqueda por división, cedula, y numero de de los permisos pedidos por los funcionarios.
- Control de las solicitudes de reposos por parte de los empleados, con el fin de poder realizar búsquedas por división, cedula y numero de la solicitud, de los reposos pedidos por los trabajadores.
- Registro de los empleados de la empresa, así como de los pasantes de la misma, a fin de poder contar con expediente electrónico de cada uno de ellos, donde se ubiquen tanto sus datos personales, como los relacionados a la institución.



- Proporcionar una sección para generar consultas a través de formatos de búsqueda referentes a los empleados y pasantes de la institución, a través de su cedula, área de interés y división a la que pertenecen.
- Contar con una interfaz amigable, clara y sencilla para que los usuarios puedan interactuar con el sistema sin complicaciones.

### **3.1.2 Modelo del Dominio del sistema**

El modelo de dominio es un modelo conceptual, empleado para comprender de una forma intuitiva el contexto del sistema y representar conceptos concernientes al dominio de una forma clara. Estos conceptos surgen de una investigación del dominio del problema.

El lenguaje unificado para modelado (UML), permite expresar de una forma gráfica un sistema de modo que otro lo pueda entender y especificar cuales son las características de un sistema antes de su construcción, es por ello que a través de un diagrama de clase de UML se puedan expresar las entidades del mundo real, por medio de sus clases de dominio y las relaciones que existen entre ellas.

El modelo de dominio del contexto actual del sistema está representado por el diagrama de clases de la figura 3.2, teniéndose 6 clases principales:

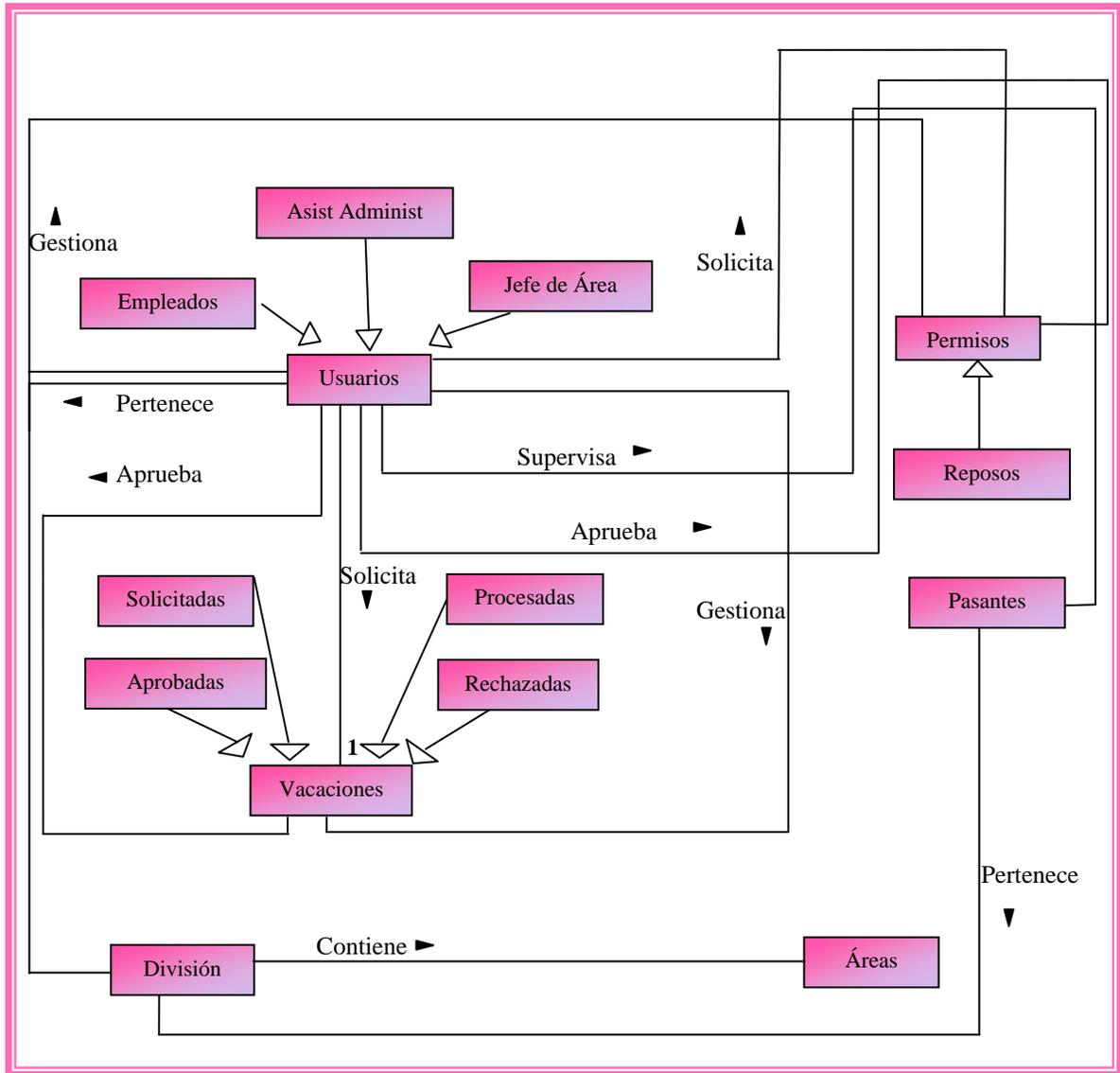


Figura 3.2 Modelo de Dominio

Fuente: Elaboracion Propia

### 3.1.2 Clases del dominio

En la Tabla se describen cada una de las clases de dominio que se obtuvieron del modelo anterior.



**Tabla 3.1 Clases de Dominio**

<b>Clases de Dominio</b>	<b>Descripción</b>
<b>Usuarios</b>	Representa los datos únicos asociados a los funcionarios de la institución. Los datos son heredados a los diferentes tipos de usuarios del sistema que son: El Jefe de Área, El Asistente Administrativo y El Empleado.
<b>Pasantes</b>	Representa los datos únicos asociados a los pasantes de la empresa.
<b>Vacaciones</b>	Contiene los datos de las solicitudes de vacaciones de todos los funcionarios, el registro de estas solicitudes es realizado por la asistente del área de personal. Cada empleado podrá consultar su historial de vacaciones.
<b>Permisos</b>	Esta clase se encuentran los datos correspondientes a las solicitudes de permisos que realizan los empleados, las solicitudes serán registradas por la asistente de personal. Cualquier trabajador podrá tener acceso a su historial de permisos.
<b>Reposos</b>	Esta clase hereda todos los atributos de la clase Permisos. Contiene toda la información concerniente a las solicitudes de permisos que son requeridas por los trabajadores, será encargado(a) de almacenar dicha información la asistente de personal. Todo funcionario tendrá disponibilidad de acceder a su historial de reposos.
<b>División</b>	Representa los datos únicos que representan las divisiones en la que esta constituida la Gerencia General de Tributos Internos de la Región Nor_Oriental
<b>Área</b>	Representa los datos únicos que representan las áreas que representan cada una de las divisiones en la que esta constituida la Gerencia General de Tributos Internos de la Región Nor_Oriental.

**Fuente: Elaboración Propia**

### **3.1.3 Riesgos del sistema.**

Los riesgos a tratar son todos aquellos que podrían provocar que el sistema fallara, por este motivo dentro de la fase de inicio se crea una lista de riesgos. Los riesgos que se perciben inciden significativamente en la forma en que se planifica el



desarrollo del nuevo sistema. A continuación se presentan una serie de riesgos que deben prevenirse para el correcto funcionamiento del sistema a desarrollar:

- *No Construir el sistema correcto* es un riesgo que debe mitigarse pronto hallando los requisitos apropiados. No se puede dejar ninguna funcionalidad por fuera, ya que es necesario saber que lo se construye puede evolucionar.
- *Que se presenten fallas a la hora de modificar el sistema de almacenamiento*, es decir que no se pueda acceder a la base de datos este riesgo es considerado como critico y debe ser mitigado en las primeras fases.
- *No obtener toda la accesibilidad deseada por parte de los usuarios del sistema a medida que recorre el software*, es decir que quiere explotar alguna zona y el sistema no se lo permita, este riesgo es considerado como significativo y debe mitigarse durante la fase de inicio y de elaboración.
- *No crear una arquitectura que sea lo suficientemente robusta*, es un riesgo considerado como critico y debe mitigarse durante las fases de inicio y de elaboración, hasta asegurarse de tener la arquitectura correcta y poder fijarla.

#### **3.1.4 Modelos de casos de usos.**

Un modelo de caso de uso es una técnica para capturar información de cómo un sistema trabaja, o cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objetos, es una técnica para captura de requisitos. Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario, además, permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.



Los casos de uso particionan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo. Están basados en el lenguaje natural, es decir, es accesible por los usuarios.

Los casos de uso bien estructurados denotan solo comportamientos esenciales del sistema o de un subsistema, y nunca deben ser excesivamente genéricos ni demasiado detallados, y su realización facilita la comprensión y el estudio del comportamiento del mismo.

Para obtener una primera versión del modelo de casos de uso hay que identificar los actores y los casos de uso del sistema de recursos humanos, a partir de del modelo de dominio descrito anteriormente.

#### **3.1.4.1 Actores del Sistema**

Un actor es un agente o ente externo que interactúa con el sistema ingresa, manipula o recibe información, pero es externo al sistema. Puede ser una persona o una aplicación que intercambia datos con el sistema.

El actor es un rol que un usuario juega con respecto al sistema, es importante destacar el uso de la palabra rol, pues con esto se especifica que un actor no necesariamente representa una persona en particular, sino más bien la labor que realiza frente al sistema.

A continuación se describen los actores correspondientes al software que se está desarrollando para automatizar las actividades relacionadas al área de personal de la Gerencia General de Tributos Internos:



Tabla 3.2 Actores del sistema

Actor	Descripcion
<b>Jefe de Área</b>	El usuario jefe de área es aquel que se encarga de darle o no la aprobación a los diferentes tipos de solicitudes presentadas por los empleados. Este usuario tendrá acceso a todo el sistema, ya que también podrá realizar cualquier tipo de consultas correspondientes a los funcionarios
<b>Asistente Administrativo</b>	El usuario asistente administrativo se encargara de ingresar todos los registros correspondientes a un nuevo empleado o pasante a la empresa, así como modificar o eliminar algún dato referente a un trabajador o pasante ya existente. El asistente administrativo se encargara además de ingresar, modificar o eliminar las solicitudes de vacaciones, permisos o reposos emitidas por los trabajadores y podrá efectuar consultas acerca de las mismas.
<b>Empleado</b>	El usuario empleado esta representado por todos los trabajadores de la empresa que utilizaran el sistema para acceder a cualquier información correspondiente a su persona, y no así a la de algún compañero.
<b>SGBD</b>	Se considera un actor debido a que interactúa directamente con el sistema que se está desarrollando. El sistema gestor de base de datos es el que ejecuta las diferentes transacciones invocadas por los usuarios en forma transparente, es decir, su participación esta implícita en cada una de las tareas solicitadas por los usuarios administrador, asistente administrativo y empleado.

Fuente: Elaboracion Propia

Con la identificación de los actores del sistema de personal, se ha determinado el entorno externo al sistema.

#### 3.1.4.2 Casos de uso del sistema

El diagrama de Casos de Uso representa la forma de cómo un actor opera con el sistema en desarrollo, además de la forma, tipo y orden en cómo los elementos



interactúan, además de mostrar los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno.

Los diagramas de casos de uso se describen utilizando el lenguaje del cliente, un lenguaje natural, sin embargo, con la utilización solamente del lenguaje natural, se pierde poder expresivo, y en la captura de los requerimientos pueden quedar sin tratar, o quedar solo vagamente descritos, muchos detalles que pueden ser precisados con notaciones mas formales, como diagramas de actividad o interacción.

### 3.1.4.3 Descripción de los casos de uso

Luego de tener claros los requerimientos y requisitos del sistema a desarrollar, se identificaron seis casos de uso principales para el sistema. A continuación se describen brevemente los casos de uso del sistema SIP.

**Tabla 3.3** Descripción de los casos de uso del sistema

<b>Caso de Uso</b>	<b>Descripcion</b>
<b>Procesar Vacaciones</b>	Engloba todas las actividades relacionadas a las solicitudes de vacaciones emitidas por los trabajadores del Seniat.
<b>Procesar Permisos</b>	Este caso de uso el encargado de procesar todo los datos referentes a las solicitudes de permisos realizadas por los empleados de la institución.
<b>Procesar Reposos</b>	Trata toda la información correspondiente a las solicitudes de reposos hechas por los empleados del Seniat.
<b>Administrar Empleados</b>	Permitirá registrar los datos personales de los empleados del Seniat, así como realizar consultas acerca de los mismos.
<b>Gestionar Pasantes</b>	Permitirá registrar los datos personales de los pasantes del Seniat, así como realizar consultas acerca de los mismos.
<b>Configurar Clave</b>	Permitirá que cualquier usuario del sistema SIP, actualizar los datos de su cuenta de usuario.

**Fuente:** Elaboracion Propia

### 3.1.4.4 Diagrama de casos de uso

Una vez finalizada la identificación y descripción de los actores y los casos de uso que participan en el Sistema de Personal SIP, se hace necesario la elaboración del diagrama de caso de uso general del sistema, con el fin de de explicar como se encuentran relacionados entre si los casos de uso y los actores, utilizando para ello el lenguaje unificado de modelado.

#### Modelo de Caso de Uso General del Sistema SIP

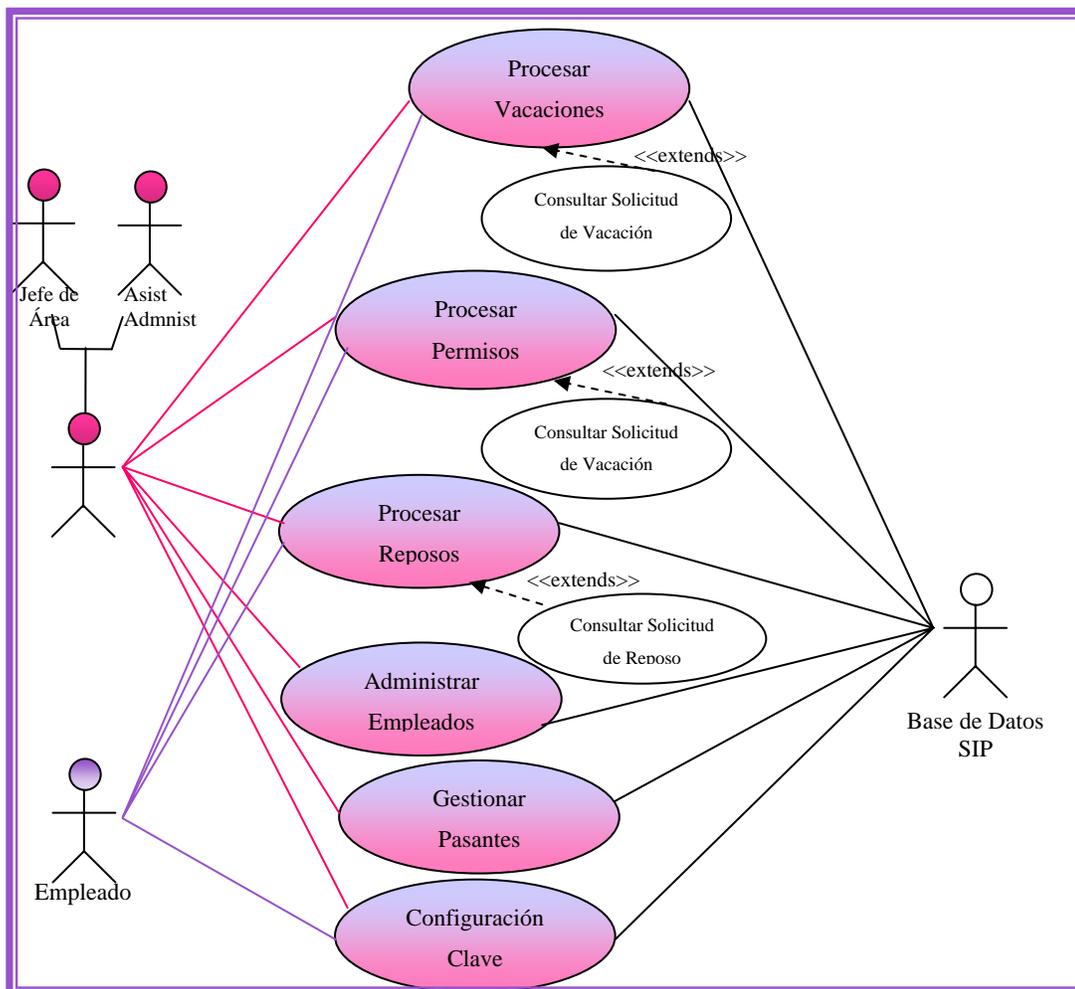


Figura 3.3 Modelo General de Casos de Uso del Sistema de Personal SIP

Fuente 3.4 Elaboracion Propia



En la figura 3.3 que se muestra a continuación se describe el diagrama de caso de uso General del Sistema de Personal SIP, el cual está formado por un conjunto de casos de uso que son activados por los actores del sistema como son los el Jefe de Área, el Asistente Administrativo y los Empleados.

El actor usuario que representa tanto al actor jefe de área, al actor asistente administrativo y al actor empleado, utiliza el caso de uso activar sesión, creando cada vez que se activa una nueva sesión, luego de identificarse correctamente.

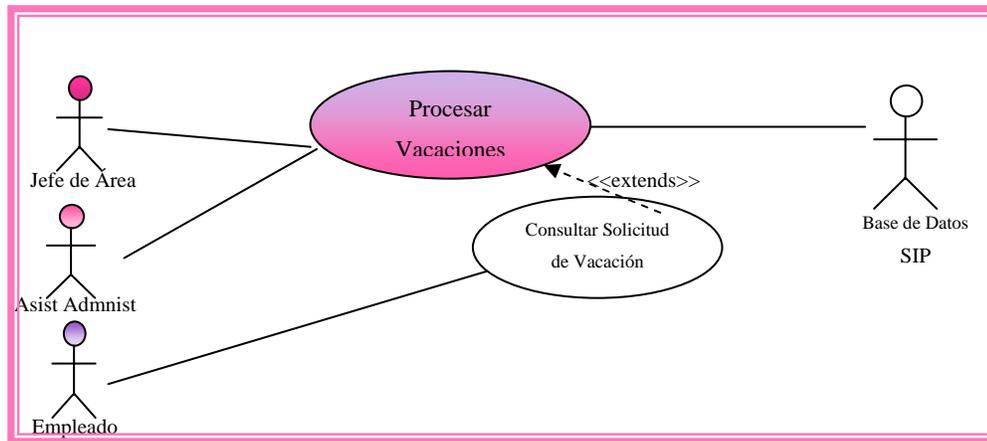
Los Actores Jefe de Área y Asistente Administrativo comparten la autorización para utilizar los casos de uso Procesar Vacaciones, Procesar Permisos, Procesar Reposos, Administrar Datos Personal

El Actor Empleado puede participar en los casos de uso Procesar vacaciones, Procesar Permisos y Procesar Reposos.

#### **3.1.4.5 Descripción detallada de los casos de uso del sistema**

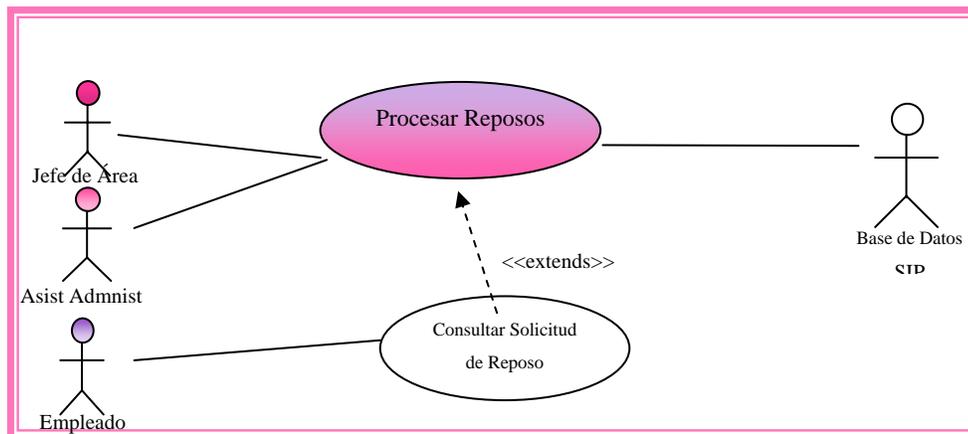
El Proceso de identificación de los casos de uso se basa en los actores, en los eventos y a partir del modelo del dominio. Así se identifican los procesos que los actores inician o en los que participan, los eventos externos a los que el sistema ha de responder t se relacionan los eventos con los actores y con los casos de uso.

A continuación se describirá de forma detallada cada uno de los casos de uso del sistema de recursos humanos, con el cual se automatizaran los procesos llevados a cabo en el área de personal de la división de administración de la gerencia general de tributos internos de la región Nor-oriental.

**Caso de uso: *Procesar Vacaciones*****Figura 3.4** Caso de Uso Procesar Vacaciones

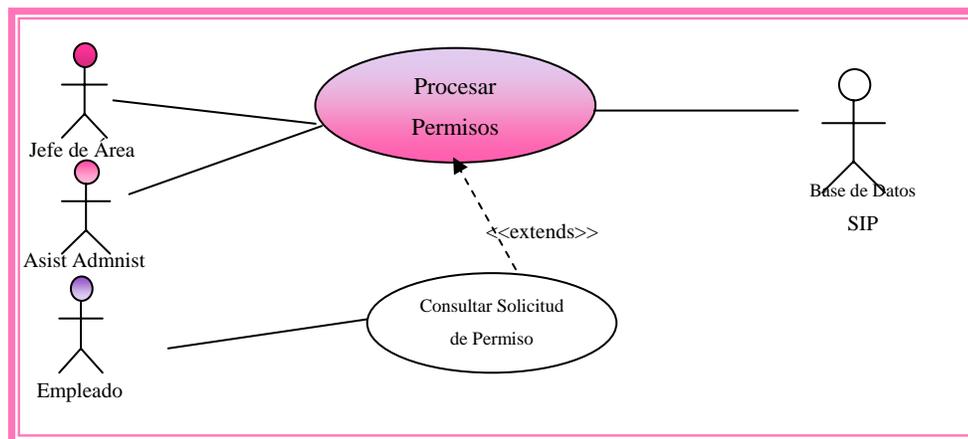
Fuente: Elaboración propia

**Descripción:** Los Actores Jefe de Área, Asistente Administrativo y Empleado después de haberse identificado, utilizan este caso de uso para realizar consultas o actualizar la información correspondiente. En este caso de uso el usuario asistente administrativo realiza operaciones tales como: consultar, ingresar, modificar o eliminar las solicitudes de vacaciones de los empleados de la empresa, el usuario empleado solo puede consultar su historial de vacaciones y el usuario jefe de área tiene acceso a este caso de uso para aprobar o desaprobar cualquier solicitud de vacaciones emitida por algún empleado, así como realizar cualquier tipo de consulta acerca de de los mismos. Las consultas pueden ser acerca de las vacaciones solicitadas, procesadas, disfrutadas y pendientes de los trabajadores. Después que el caso de uso comience, los usuarios implicados visualizan la presentación principal eligiendo hacia donde desean desplazarse. El SGBD ejecuta todas las transacciones invocadas.

**Caso de uso: *Procesar Reposos*****Figura 3.5** Caso de Uso Procesar Reposos**Fuente:** Elaboracion propia

**Descripción:** Los Actores Jefe de Area, Asistente Administrativo y Empleado participan en este caso de uso previamente identificados, al activarse este caso de uso por el asistente administrativo este puede efectuar operaciones como: ingresar, modificar, eliminar o consultar las solicitudes de reposos llevadas por los empleados, las consultas pueden ser en particular, es decir, de un empleado específico bien sea por la cedula del empleado o por el número de solicitud de reposo, o pueden ser general, que se refiere a empleados pertenecientes a una división determinada de la empresa, el jefe de área puede hacer cualquier tipo de consulta de las antes mencionada, y los empleados solo podrán hacer consultas particulares; es decir, su correspondiente historial de reposos. Después que el caso de uso comience, los actores involucrados visualizan la presentación principal eligiendo hacia donde desean desplazarse. El SGBD ejecuta todas las transacciones invocadas.

### Caso de uso: *Procesar Permisos*



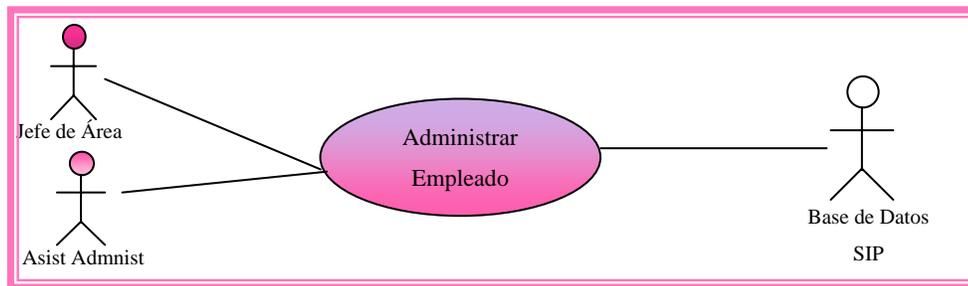
**Figura 3.6** Caso de Uso *Procesar Permisos*

**Fuente:** Elaboracion propia

**Descripción:** En este caso de uso pueden intervenir el jefe de área, el asistente administrativo o por cualquier empleado, luego de ser identificados por el sistema. El asistente administrativo usa este caso de uso para manipular los datos relacionados a las solicitudes de permisos efectuadas por los trabajadores, ejecutando operaciones de diversos tipos como: ingresar, modificar, eliminar y consultar la información correspondiente cuando así se necesitase, las consultas pueden ser en particular, es decir, de un empleado específico ya sea por medio de la cedula o por el número de solicitud de permiso o puede ser general, que se refiere a empleados pertenecientes a una división determinada de la empresa, el jefe de área utilizara este caso de uso para dar la autorización o no a las solicitudes de permisos formuladas por los funcionarios, a su vez este podrá ejecutar cualquier tipo de consulta referentes a los mismos, el actor empleado tendrá disponibilidad de consultar únicamente su historial de permisos. Después que el caso de uso comience, los usuarios participantes visualizan

la presentación principal eligiendo hacia donde desean desplazarse. El SGBD ejecuta todas las transacciones invocadas.

### Caso de uso: *Administrar Empleados*

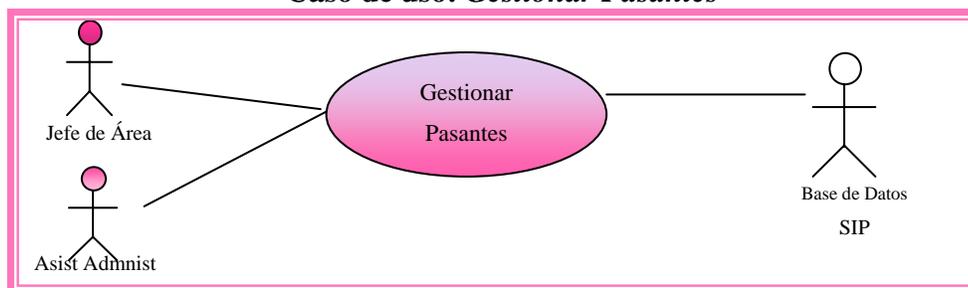


**Figura 3.7** Caso de Uso Administrar Empleados

Fuente: Elaboracion propia

**Descripción:** Los Actores Asistente Administrativo y Jefe de Area pueden activar este caso de uso para manejar los datos de todos los empleados pertenecientes a la gerencia regional de tributos internos de la región Nor-Oriental, bien sea para ingresar, modificar eliminar y/o consultar un registro ya existente. Las consultas de los empleados pueden ser por cedula, cargo y por la división a la que pertenecen. Luego que el caso de uso comience, el jefe de área y el asistente administrativo visualizan la presentación principal eligiendo hacia donde desean desplazarse. El SGBD ejecuta todas las transacciones invocadas.

### Caso de uso: *Gestionar Pasantes*

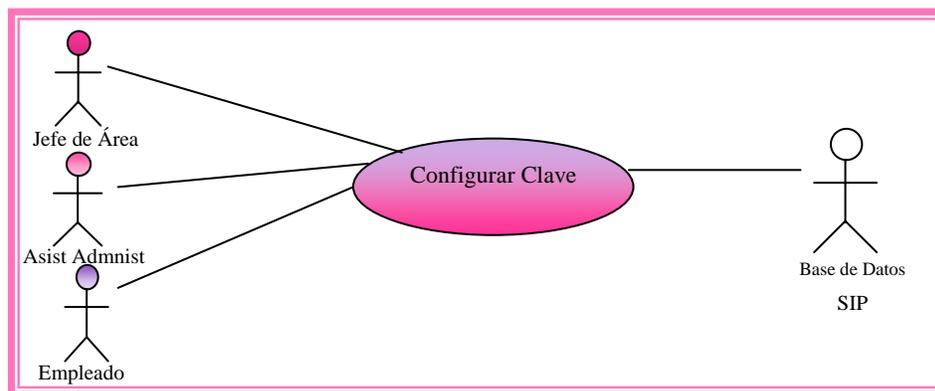


**Figura 3.8** Caso de Uso Gestionar Pasantes

Fuente: Elaboracion propia

**Descripción:** Los actores Asistente Administrativo y Jefe de Area pueden activar este caso de uso para manejar los datos de todos los pasantes pertenecientes a la gerencia regional de tributos internos de la región Nor-Oriental, bien sea para ingresar, modificar eliminar y/o consultar un registro ya existente. Las consultas de los empleados pueden ser por cedula, de acuerdo a su área específica de trabajo establecida por la institución y por la división a la que pertenecen. Luego que el caso de uso comience, el jefe de área y el asistente administrativo visualizan la presentación principal eligiendo hacia donde desean desplazarse. El SGBD ejecuta todas las transacciones invocadas.

#### *Caso de uso: Configurar Clave*



**Figura 3.9** Caso de Uso Configurar Clave

**Fuente:** Elaboracion propia

**Descripción:** Este caso de uso puede ser activado por los actores Jefe de Área, Asistente Administrativo y Empleado de la institución. Este caso de uso tendrá la función de cambiar la clave a cada uno de los usuarios del sistema de personal SIP, para conservar la privacidad de sus datos.



### 3.1.6 Requisitos No Funcionales.

Los requisitos definen qué deberá hacer el sistema y las restricciones acerca de cómo debe operar y como debe ser desarrollado éste, por tanto, los requisitos funcionales definen los servicios que el sistema debería proporcionar mientras que los requisitos no funcionales restringen el sistema a desarrollar o el mismo proceso de desarrollo, es decir; Representan restricciones de los servicios o funciones ofrecidas por el sistema que surgen de las necesidades del usuario, entre las que se encuentran: fiabilidad, tiempo de respuesta, capacidad de almacenamiento, entre otros.

Los Requisitos no funcionales aplicados al sistema Sip, se encuentran descritos en la siguiente tabla.

**Tabla 3.4** Requisitos No Funcionales

<b>Requisito No Funcional</b>	<b>Descripción</b>
<b>Entorno de Desarrollo</b>	El sistema debe implementarse con tecnología Web (HTML y Scripts PHP), con el uso de software libre, un servidor de aplicación que soporte el manejador de base de datos MySQL sobre un sistema operativo WindowsXP.
<b>Portabilidad</b>	El Software debe ser portable y funcionar en una plataforma, donde se encuentre instalado el servidor de apache con acceso a Internet y sistema manejador de base de datos mysql.
<b>Usabilidad</b>	El Software debe realizar todos los registros de los procesos de manera eficiente, además de poseer una interfaz clara y sencilla para la mayor comprensión del sistema.
<b>Reutilización o Extensibilidad</b>	El Software debe estar diseñado para soportar cambios de actualizaciones a largo plazo, que permita adicionar nuevas funciones requeridas en el departam.ento de personal.
<b>Accesibilidad</b>	Se debe proporcionar un nivel de acceso para los usuarios que permita establecer cada uno de los módulos según el rol que éste representa.

Fuente: Elaboracion propia



### **3.1.7 Requisitos adicionales.**

Existen requisitos no funcionales que no se pueden asociar a ningún caso de uso concretamente, sin embargo cada uno de estos requisitos pueden tener o no impacto en varios casos de uso. A estos requisitos se les conoce fundamentalmente como requisitos adicionales.

A continuación se muestra un conjunto de requisitos adicionales que fueron capturados para el sistema, los cuales son restrictivos para el desarrollo ya que se cuentan con ellos desde el momento que se inicia el proyecto, estos garantizarían el buen comportamiento del sistema en su posterior implementación. A saber:

#### **Disponibilidad.**

El sistema debe estar disponible para cualquier usuario registrado, cuando así fuese necesario.

#### **Facilidad de Aprendizaje.**

El sistema debe contar con una interfaz sencilla y amigable, que conlleve a un aprendizaje rápido de su funcionamiento, de tal manera que el usuario sea capaz de manejarla con facilidad y eficiencia. Los equipos en donde se utilizara el sistema deben garantizar la correcta ejecución de su contenido.

#### **Seguridad.**

El sistema debe tener bien definido hacia que módulo(s) o interfaces de la aplicación tiene acceso cada usuario, respetando las restricciones con las que cuentan los usuarios que intervienen en el sistema.



### **3.1.8 Requisitos de la plataforma hardware**

- Un procesador Pentium o equivalente a 2.0 GHZ o superior
- Se recomienda 1 GB de memoria RAM y 512 MB como mínimo.
- Monitor a color capaz de mostrar una resolución de 800x600 píxeles.
- Unidad de D-ROM.
- Disco duro 80 GB.

### **3.1.9 Requisitos de plataforma software.**

- Sistema operativo: Windows 9x/NT, XP, 2000
- Software para Internet: Microsoft Internet Explorer versión 6.02 en adelante.
- Microsoft Office 2007: Word, Power Point, Excel, Access.
- Php 5.04, Mysql, Apache http Server.
- Macromedia Dreameawer 8.

### **3.1.10 Diseño lógico de la interfaz de usuario**

Hasta los momentos se ha desarrollado un modelo de casos de uso que especifica que usuarios hay y para que necesitan el sistema, ahora es necesario diseñar la interfaz de usuario que permita a cada actor llevar a cabo los casos de uso de manera eficiente.

Cuando los actores interactúan con el sistema, utilizan y manipulan elementos de interfaces de usuario que representan atributos (de los casos de uso). Estos elementos se identifican actor por actor, recorriendo todos los casos de uso a los que el actor puede acceder, e identificando los elementos apropiados de la interfaz de usuario para cada caso de uso.



### 3.2 Flujo de Trabajo: Analisis.

El objetivo general del flujo de trabajo de análisis es analizar los requisitos, refinarlos y estructurarlos en un modelo de objetos que sirva como primera impresión del modelo de diseño.

#### 3.2.1 Modelos de Análisis

El modelo de análisis es un modelo de objetos conceptual que analiza los requerimientos mediante su refinamiento y su estructuración, nos permite razonar sobre los aspectos internos del sistema incluidos sus recursos compartidos internos.

Este modelo ofrece mayor poder expresivo y una mayor formalización, como la que expresan los diagramas de interacción que se utilizan para describir los aspectos dinámicos del sistema. El modelo de análisis puede considerarse como una primera aproximación del modelo de diseño. Dentro del modelo de análisis, los casos de uso se describen mediante clases de análisis y sus objetos. Esto se representa mediante diagramas de colaboraciones.

#### 3.2.2 Clases de Análisis

Una clase de análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Las clases de análisis ayudan a construir un sistema robusto y a encontrar activos reutilizables. Estas siempre encajan en cada uno de los tres estereotipos básicos que se muestran a continuación:

- **Clases de Entidad.** Se modelar información que posee una vida larga y que es a menudo persistente. Las clases de Entidad modelan la información y el



comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto o un suceso utiliza para del mundo real.

- **Clases de Interfaz.** Se utiliza para modelar la interacción entre el sistema y sus actores (es decir, usuarios y sistemas externos). Esta interacción a menudo implica recibir (y presentar) información y peticiones de (y hacia) los usuarios y los sistemas externos. Representan ventanas, formularios, paneles, interfaces de comunicación, etc.
- **Clases de Control.** Representan coordinación, secuencia, transacciones y control de los objetos y se usan para encapsular el control relativo de un caso de uso en concreto. Las clases de control también se utilizan para representar derivaciones y cálculos complejos, como la lógica del negocio, que no pueden asociarse con ninguna información concreta de larga duración, almacenada por el sistema(es decir, una clase de entidad concreta).

### 3.2.2.1 Descripción de las Clases Entidad del Sistema SIP

Las clases de entidad modelan información persistente y en ocasiones pueden tener un comportamiento complejo y métodos relativos a la información que representa. Para el sistema SIP se determinaron las siguientes clases de entidad:

- **Entidad Usuarios:** Representa los datos con los que cada uno de los empleados cuenta para tener acceso al sistema.
- **Entidad Vacaciones:** Guarda los datos y métodos relacionados con las solicitudes de vacaciones que hayan sido pedidas por parte de los empleados.



- *Entidad Permisos:* Guarda los datos de los relacionadas a cada una de las solicitudes de permisos concedidas a los trabajadores de la empresa. que tendrán acceso al sistema
- *Entidad Reposos:* Representa toda la información que va a estar contenida en todas las solicitudes de reposos hechas por todo el personal de la institución.
- *Entidad Personal:* Representa los datos personales de todas aquellas personas que laboran dentro del Seniat.
- *Entidad Pasantes:* Guarda los datos referentes a cada uno de los pasantes q están cumpliendo con su pasantia dentro de la empresa.

### 3.2.2.2 Descripción de las Clases Interfaz del Sistema SIP

Las clases de fronteras o clases de interfaz comúnmente llamadas, se relacionan con entradas y salidas. Cada pantalla de entrada o salida se modelan por su propia clase de frontera, es decir, la clase de frontera incorpora todos los elementos de datos y las diversas operaciones de un módulo específico.

- *Interfaz Empleado:* Permite la comunicación entre cualquier empleado de la empresa con el sistema, este podrá visualizar cualquier tipo de solicitud, bien sea de vacaciones, permiso o reposo correspondiente a su persona.
- *Interfaz Administrador:* Mediante esta interfaz se establece la comunicación el administrador del sistema (Jefe de Area y Asistente Administrativo) y el sistema, quienes podrán acceder a todos los modulos de la aplicación y así realizar cualquiera de las actividades correspondientes a cada modulo.



### 3.2.2.3 Descripción de las Clases Control del Sistema SIP

En la clase de control se modelan todos aquellos cálculos no triviales, como cálculos y algoritmos complejos. Para llevar a cabo un proceso específico es necesario el uso de una clase de control. En SIP se han identificado las siguientes clases de control:

- *Gestor Procesar Vacaciones:* Controla todos los procesos relativos a las solicitudes de vacaciones, así como registrar, modificar, eliminar o consultar información. Este gestor es activado a través de la interfaz Administrador.
- *Gestor Consultar Vacaciones:* Lleva a cabo el control de todos los procesos concernientes a las consultas de vacaciones. Este gestor se activa en la interfaz Empleado.
- *Gestor Procesar Permisos:* Controla la gestión de datos referentes a las solicitudes de Permisos, así como registrar, modificar, eliminar o consultar información. Este gestor es activado a través de la interfaz Administrador.
- *Gestor Consultar Permisos:* Lleva a cabo el control de todos los procesos concernientes a las consultas de Permisos. Este gestor se activa en la interfaz Empleado.
- *Gestor Procesar Reposos:* Gestiona el manejo de toda la información relacionada a las solicitudes de Reposos, como registrar, modificar, eliminar y consultar. A través de la interfaz Administrador se activa este gestor.
- *Gestor Consultar Reposos:* Se encarga de controlar todos los procesos concernientes a las consultas de Reposos. Este gestor se activa en la interfaz Empleado.
- *Gestor Administrar Empleado:* Controla la gestión de datos referente a los empleados. Por medio de este gestor se generan las consultas que permiten ingresar, modificar o eliminar información. En la interfaz Administrador se activa este gestor.



- **Gestor Gestionar Pasantes:** Lleva a cabo el control de la información vinculada a los pasantes. Por medio de este gestor se generan las consultas que permiten ingresar, modificar o eliminar información. En la interfaz Administrador se activa este gestor.
- **Gestor Configurar Clave:** Este gestor se encarga de llevar a cabo los procesos asociados al cambio de la clave personal de cada usuario del sistema, para que así puedan acceder al mismo y mantener la seguridad de sus datos su cuenta. Este gestor se activa tanto en la interfaz Administrador como en la interfaz Empleado.

La aplicación de estos conceptos da origen a la elaboración de los diagramas de clase de análisis del sistema SIP, explicados a continuación:

### **3.2.3 Diagramas de Clases de Analisis**

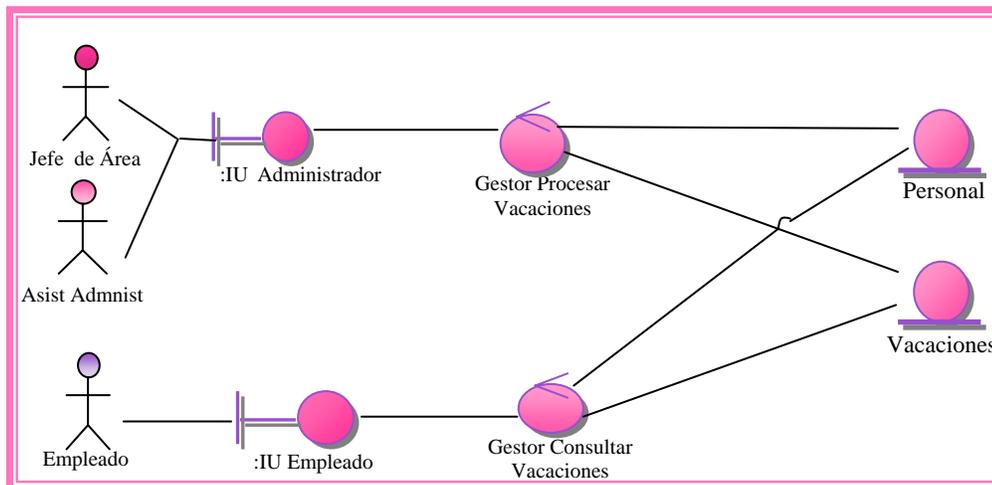
Las clases de análisis proporcionan mayor información sobre las relaciones y colaboraciones entre cada clase. Luego de la descripción de cada una de las clases que actúan en el sistema SIP se realiza el diagrama de clase de análisis para los casos de usos del sistema.

#### **3.2.3.1 Clase de análisis procesar vacaciones.**

Clase(s) de interfaz: **IU\_Administrador**

Clase(s) de Control: **Gestor Procesar Vacaciones, Gestor Consultar Vacaciones.**

Clase(s) de Entidad: **Vacaciones, Personal.**



**Figura 3.10** Clase de Analisis Procesar Vacaciones

Fuente: Elaboracion propia

Los actores Asistente Administrativo, y Jefe de Area previamente identificados inician su interacción con el caso de uso invocando la clase de interfaz *IU- Administrador*, luego la clase de control *Gestor Procesar Vacaciones* se encarga de darle respuesta a las solicitudes hechas por los usuarios a través de la manipulación de los datos registrados en las clases de entidad *Personal* y *Vacaciones*.

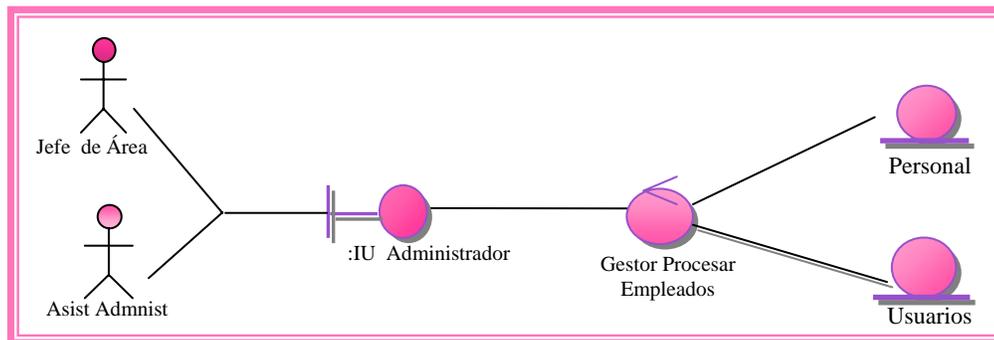
El actor Empleado identificados inician su interacción con el caso de uso invocando la clase de interfaz *IU- Empleado*, luego la clase de control *Gestor Consultar Vacaciones* realiza las validaciones de datos, se comunica con las entidades *Personal*, *Vacaciones* y efectúa las consultas respectivas.

### 3.2.3.2 Clase de análisis administrar empleados.

Clase(s) de interfaz: **IU\_Administrador**

Clase(s) de Control: **Gestor Administrar Empleados.**

Clase(s) de Entidad: **Usuarios Personal.**



**Figura 3.11** Clase de Analisis Administrar Empleados

Fuente: Elaboracion propia

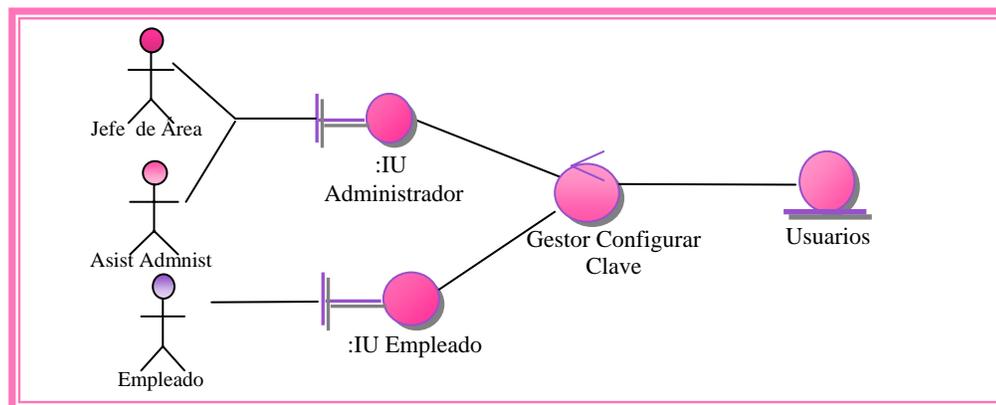
Cada uno de los actores participantes debe ser identificado con anticipación. La activación de este caso de uso se puede dar a través de interfaz: *IU\_Administrador* seguidamente la clase de control *Gestor de Procesar Empleados* es la responsable de coordinar las tareas emitidas por los usuarios, y que se efectúan sobre la información contenida en las clases de entidad *Usuarios* y *Personal*.

### 3.2.3.3 Clase de análisis configurar clave

Clase(s) de interfaz: *IU\_Administrador*, *IU\_Empleado*

Clase(s) de Control: *Gestor Configurar Clave*

Clase(s) de Entidad: *Permisos*, *Personal*.



**Figura 3.12** Clase de Analisis Configurar Clave

Fuente: Elaboracion propia



Este caso de uso comienza a ejecutarse después que los usuarios empleado, asistente administrativo y jefe de área han sido chequeados por el sistema, para así poder iniciar la interacción con cualquiera de las clases que intervienen en este modulo: *IU-Asistente administrador* o *IU\_Empleado*, ya después de haber sido activado, la clase de control *Gestor de Permisos* se encarga de dirigir las acciones que requieran obtener datos contenidos en las clases de entidad *Permisos* y *Personal*.

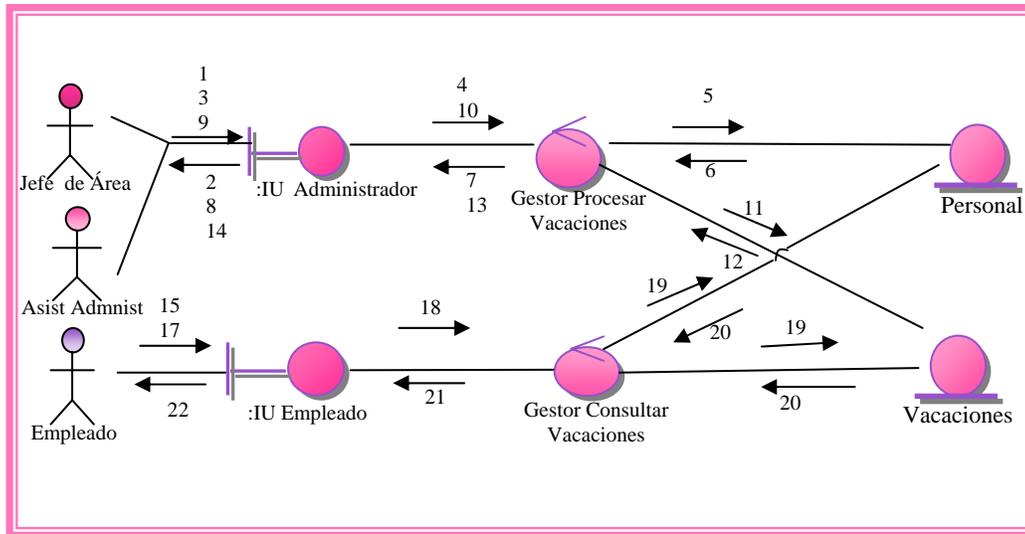
### **3.2.4 Diagramas de Colaboracion**

El Diagrama de Colaboración es la descripción de una colección de objetos que interactúan para implementar un cierto comportamiento dentro de un contexto, describiendo una sociedad de objetos cooperantes unidos para realizar cierto propósito. El diagrama de colaboración representa colaboraciones que enfatizan la organización estructural de los objetos que pasan mensajes entre ellos mismos, los diagramas de colaboración incluyen una ruta y diversos números de secuencia.

En estos diagramas mostramos las relaciones entre los objetos (instancia de una clase que participa como una interacción), no la secuencia en el tiempo en que se producen los mensajes.

Los diagramas de colaboracion cubren la vista dinámica del sistema, resaltando la organización estructural de los objetos que envían y reciben mensajes. A continuación se presenta el diagrama de colaboración correspondiente a los casos Procesar Vacaciones, Administrar Empleados y Configurar Clave, todo partiendo de la interaccion entre objetos que se puede observar en los diagramas de clases de análisis de los casos de uso mencionados anteriormente

### 3.2.4.1 Diagrama de colaboración Procesar Vacaciones



**Figura 3.13** Diagrama de Colaboración Procesar Vacaciones

**Fuente:** Elaboracion Propia

**LEYENDA:**

- 1.El Usuario Jefe de Área o Asistente Administrativo, selecciona la opción vacaciones y selecciona una de las opciones disponibles.
- 2.El sistema responde mostrando la interfaz solicitada.
- 3.El usuario Jefe de Área o Asistente Administrativo ingresa el numero de la cedula del empleado que solicita la vacación.
- 4.La interfaz IU Administrador le envía un mensaje al Gestor Procesar Vacaciones solicitando la ejecución de la operación.
- 5.El Gestor Procesar Vacaciones establece comunicación con la clase de entidad Personal.
- 6.La entidad Personal devuelve los datos correspondientes a dicha cedula.
- 7.Proporciona los datos correspondientes al empleado.
- 8.Despliega los datos correspondientes al empleado.



9. El usuario Jefe de Área o Asistente Administrativo, introduce los datos correspondientes a la operación seleccionada.
10. La interfaz IU Administrador le envía un mensaje al Gestor Procesar Vacaciones solicitando la ejecución de la operación.
11. El Gestor Procesar Vacaciones establece comunicación con la clase de entidad Vacaciones
12. La entidad Vacaciones responde a la operación y es restituida al Gestor Procesar Vacaciones.
13. El Gestor Procesar Vacaciones envía la respuesta solicitada por el usuario.
14. Se muestran los resultados a través de la IU Administrador.
15. El Usuario Empleado, selecciona la opción Consultar Vacaciones.
16. El sistema responde mostrando la interfaz solicitada.
17. El usuario Empleado ingresa su número de cedula, el número de vacación o el tipo de vacación.
18. La interfaz IU Empleado le envía un mensaje al Gestor Consultar Vacaciones solicitando la ejecución de la operación.
19. El Gestor Consultar Vacaciones establece comunicación con las clases de entidad Vacaciones y Personal.
20. Las entidades Vacaciones y Personal devuelven los datos correspondientes a dicha cedula, número de vacación o tipo de vacación.
21. Proporciona los datos correspondientes del empleado de acuerdo al criterio de búsqueda.
22. Despliega los datos correspondientes al empleado.

### 3.2.4.2 Diagrama de colaboración Administrar Empleados

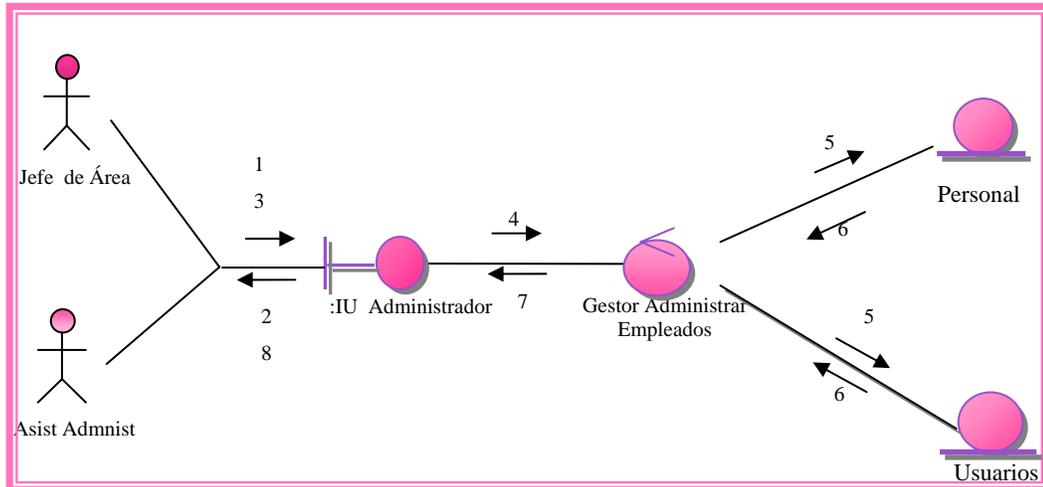


Figura 3.14 Diagrama de Colaboración Administrar Empleados

Fuente: Elaboración Propia

#### LEYENDA

- 1.El Usuario Jefe de Área o Asistente Administrativo, selecciona la opción Usuarios y selecciona y la tarea a realizar.
- 2.El sistema responde mostrando la interfaz solicitada.
- 3.El usuario Jefe de Área o Asistente Administrativo ingresa los datos de entrada de la tarea a realizar
- 4.La interfaz IU Administrador le envía un mensaje al Gestor Administrar Empleados solicitando la realización de la tarea.
- 5.El Gestor Administrar Vacaciones establece comunicación con la clase de entidad Personal y Usuarios.
- 6.Las entidades Personal y Vacaciones devuelven la respuesta al Gestor Administrar Empleados
- 7.El Gestor Administrar Empleados remite la respuesta pedida por el usuario.
- 8.Se despliega los resultados a través de la IU\_Administrador.

### 3.2.4.3 Diagrama de colaboración Configurar Clave

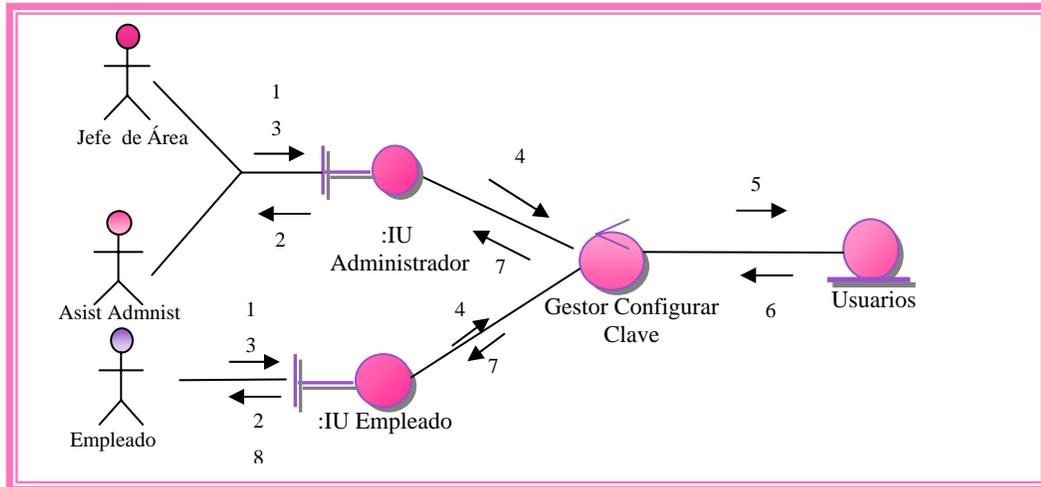


Figura 3.15 Diagrama de Colaboración Configurar Clave

Fuente: Elaboracion Propia

#### LEYENDA

- 1.El Usuario Jefe de Área, Asistente Administrativo o empleado selecciona la opción cambiar clave.
- 2.El sistema responde mostrando la interfaz solicitada.
- 3.El usuario Jefe de Área , Asistente Administrativo o empleado ingresa los datos de entrada.
- 4.La interfaz IU Administrador o la interfaz IU Empleado, le envía un mensaje al Gestor Configurar Clave solicitando la ejecución de la tarea.
- 5.El Gestor Configurar Clave establece comunicación con la clase de entidad Usuarios.
- 6.La respuesta de la tarea es devuelta al Gestor Configurar Clave.
- 7.El Gestor Configura Clave envía la respuesta requerida por el usuario.
8. Se muestran los resultados a través de la IU Administrador o la interfaz IU Empleado

### 3.2.5 Identificación de los Paquetes de Análisis

Los paquetes de análisis proporcionan un medio para organizar y manejar la complejidad de modelos grandes, convirtiéndolas en piezas más pequeñas y manejables.

A partir del análisis de los casos de uso anteriormente especificados, se puede determinar una arquitectura candidata, a través de la identificación de los paquetes de análisis. El sistema se divide en paquetes, y estos a su vez están integrados por los casos de uso que forman parte del sistema.

El sistema de personal estará dividido en cuatro paquetes de análisis: Vacaciones, Permiso, Persona y Clave, en donde se encuentran distribuidos los casos de uso del sistema SIP. A continuación se identifican los paquetes de análisis a partir de los casos de uso.

- Paquete de Análisis Procesar Vacaciones: Contiene el caso de uso Procesar Vacaciones, el cual permite llevar a cabo el control de las solicitudes de vacaciones que son pedidas por los trabajadores de la institución.

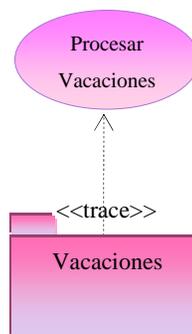
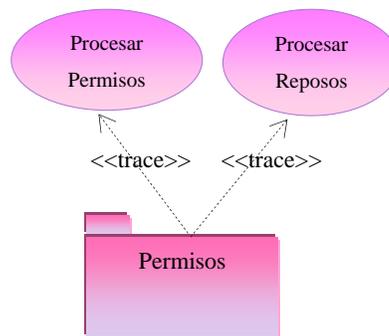


Figura 3.16 Paquete de Análisis Vacaciones

Fuente: Elaboración Propia

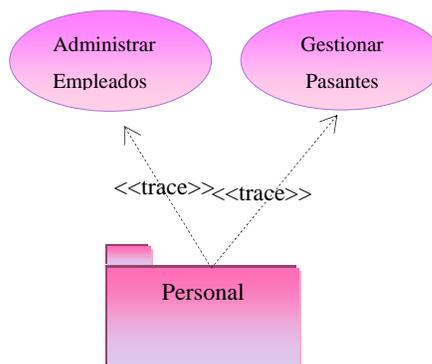
- Paquete de Permisos: Dentro de este paquete de análisis, se encuentran los casos de uso Procesar Reposos, y Procesar Permisos, esto debido a que un reposos se establece como un tipo de permiso para el empleado.



**Figura 3.17** Paquete de Analisis Permisos

**Fuente:** Elaboracion Propia

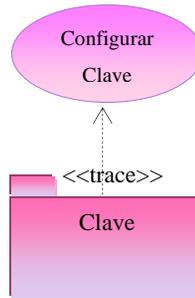
- Paquete de Análisis Personal: El paquete de análisis Personal esta constituido por los casos de uso Administrar Empleados y Gestionar Pasantes, y contiene todos los procesos relacionados a los registros y consultas de las personas que laboran dentro de la empresa.



**Figura 3.18** Paquete de Analisis Personal

**Fuente:** Elaboracion Propia

- Paquete de Análisis Clave: El paquete de análisis Clave esta constituido por el caso de uso Configurar Clave, y contiene todos los procesos relacionados para la actualización de la cuentas de los diferentes usuarios del sistema.



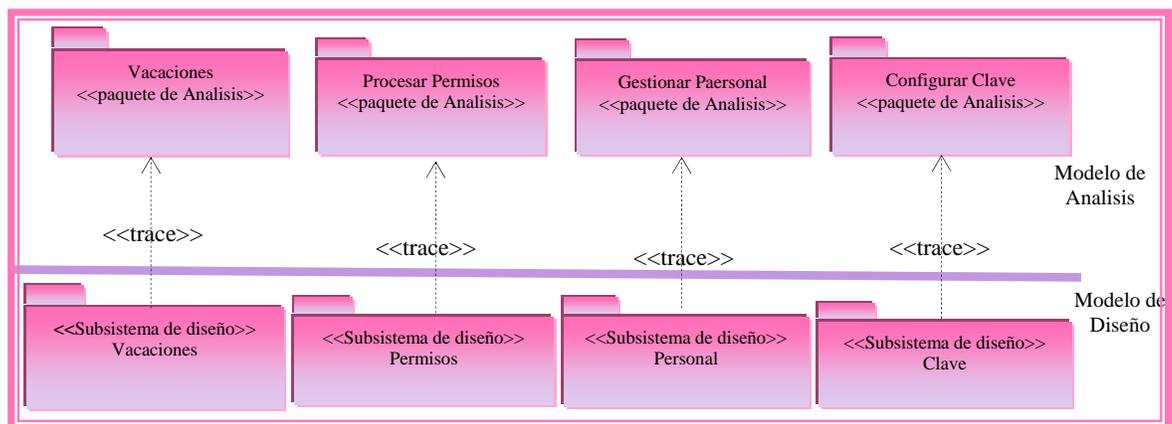
**Figura 3.19** Paquete de Analisis Clave

**Fuente:** Elaboracion Propia

### 3.3 Flujo de Trabajo: Diseño

#### 3.3.1 Identificacion de Subsistemas

Un subsistema es simplemente una parte de un sistema, que se utiliza para descomponer el sistema complejo en partes casi independientes, es decir, los subsistemas constituyen un medio para organizar el modelo de diseño en piezas más manejables. Para la identificación de los subsistemas, se utilizan los paquetes de análisis obtenidos en el flujo de trabajo análisis, tanto como sea posible. La figura 3.20 muestra los subsistemas identificados a partir de los paquetes existentes.



**Figura 3.20** Identifiacion de Sbsistemas

**Fuente:** Elaboracion Propia



### **3.4 Evaluacion de la Fase de Inicio**

Esta fase comienza con la descripción de los requerimientos necesarios, para el desarrollo del sistema de personal (SIP). A través del uso de técnicas como la observación y la entrevista entre otras, se logro recopilar toda la información asociada a la forma cómo se llevan a cabo todos los procesos relacionados al área de personal y que harán parte del sistema a desarrollar, lo que condujo en primer termino a establecer el modelo de dominio del sistema, ya que para el desarrollo del proyecto se sigue la metodología del proceso unificado de desarrollo de software.

Se definieron los requisitos funcionales y no funcionales, se trazaron los posibles riesgos que pueda presentar el sistema, como un medio de prevención, para así poder corregirlos antes del inicio del proyecto. Seguidamente se plantearon e ilustraron los casos de uso del sistema de personal, así como los diferentes actores que intervendrán en el mismo, definiendo el rol que cada uno desempeñará.

Con el desarrollo en análisis de los casos de uso, a través de los diagramas de clase de análisis y de colaboración, se mostró de manera dinámica el planteamiento y las funciones que debe cumplir el sistema y por medio de la definición de los paquetes de análisis se pudo una obtener de forma clara y sencilla los elementos del sistema que se encuentran relacionados entre si y a su vez sirvieron de punto de referencia para identificar los subsistemas que formara al sistema en desarrollo.

## **CAPITULO IV**

### **FASE DE ELABORACION**

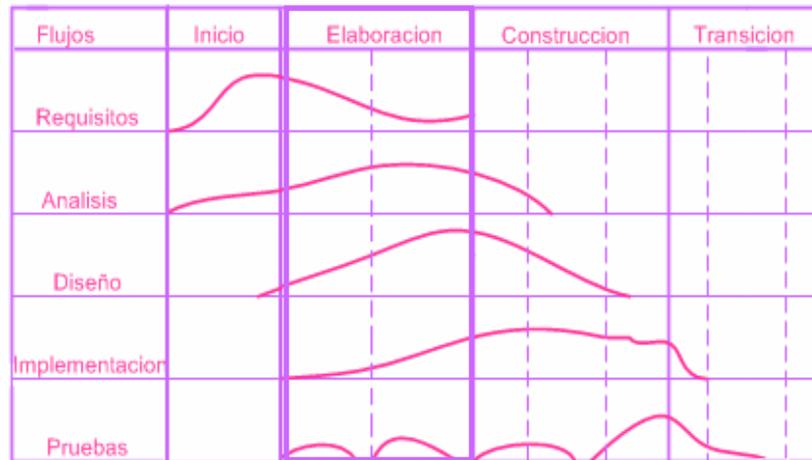
#### **PLANIFICACION DE LA FASE DE ELABORACION**

El principal objetivo de la fase de elaboración es formular la arquitectura básica del sistema, tomando como punto de partida toda la información recopilada durante la fase anterior, es decir, los riesgos, requisitos y contexto del sistema SIP, todo ello descrito por medio de los casos de uso y diagramas de análisis. De esta manera se dará fiel cumplimiento con los flujos de trabajo característicos de esta fase.

Esta etapa de elaboración proporciona la definición y validación de una arquitectura estable y sólida para el sistema. Se hace un refinamiento de la visión del sistema estableciéndose una sólida comprensión de los casos de uso que definen las decisiones arquitectónicas y de planificación, lo que nos servirá de apoyo durante las siguientes fases y para detallar el plan de trabajo de la primera iteración de la fase de construcción.

Es en la fase de Elaboración donde se establece un primer enfoque de la interfaz de usuario y del diseño de la base de datos.

Para el logro eficiente de esta fase, es necesario cumplir con cada uno de los flujos de trabajos: requisitos, análisis, diseño e implementación, que se muestra en la figura 4.1



**Figura 4.1** Flujos de trabajo requeridos en la Fase de Elaboración

Fuente: Elaboración Propia

La planificación de la fase de elaboración es necesaria a lo largo de todo el ciclo de desarrollo, ya que se identifican y aminoran los riesgos, se realizan los casos de uso adicionales (si es el caso) y llevando a cabo la evaluación y ejecución de cada iteración.

#### **4.1 Flujo de Trabajo: Requisitos.**

En este flujo de trabajo no se lograron identificar nuevos actores, casos de uso ni requisitos del sistema que puedan ser añadidos a los establecidos anteriormente en la fase de inicio.

#### **4.2 Flujo de Trabajo: Analisis.**

##### **4.2.1 Análisis de la arquitectura**

La idea fundamental para una arquitectura estable se basa en abstraer los principales casos de uso en función de sus necesidades, por tanto se debe determinar

los mecanismos que permitan fundamentar la implementación de los casos de uso establecidos, para que puedan ser ubicados arquitectónicamente en el contexto general del sistema.

#### 4.2.1.1 Identificación de los paquetes de análisis.

Los paquetes de análisis proporcionan un medio para organizar los artefactos del modelo de análisis en piezas manejables. Los paquetes de análisis deberían crearse basándonos en los requisitos funcionales y en dominio del problema.

A Continuación se puede observar la identificación de los paquetes de análisis, descritos en la fase de inicio a partir del modelo general de Caso de Uso del sistema, de manera que se note reflejado todo el sistema.

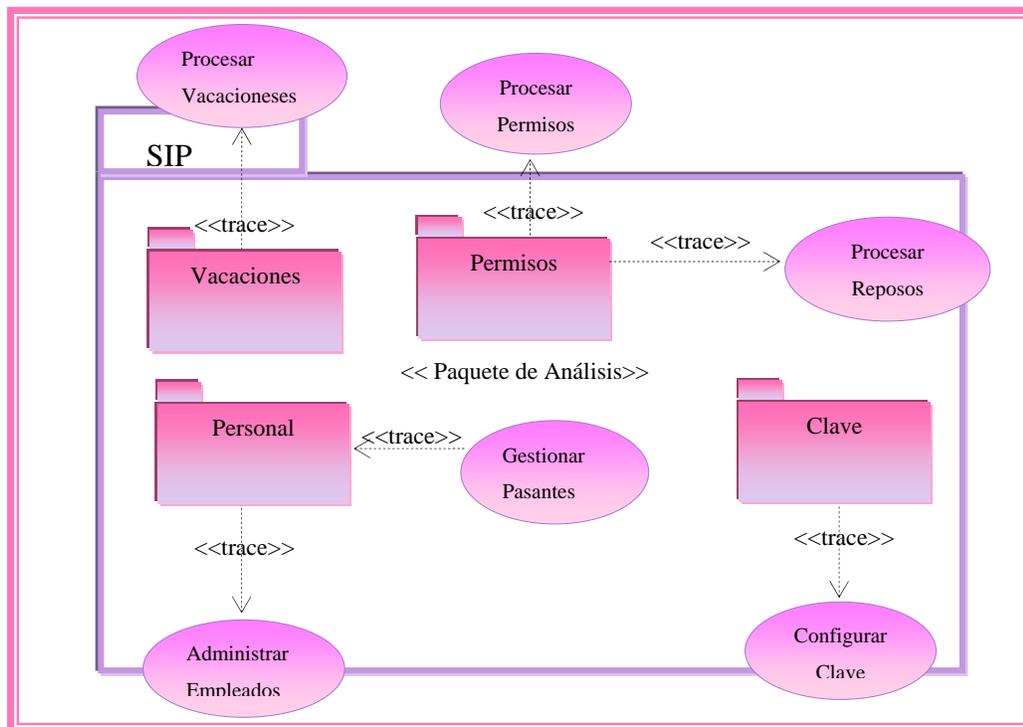


Figura 4.2 Diagrama de Paquetes de Análisis General del Sistema

Fuente: Elaboración Propia



### **4.3 Flujo de Trabajo: Diseño**

Luego de haber desarrollado los flujos de requisitos y análisis, se tienen las herramientas suficientes para proceder con el flujo de diseño, que tiene como objetivo fundamental en la fase de elaboración la obtención de la vista de la arquitectura del Modelo de Diseño, que es un modelo de objeto, que describe la realización física de los casos de usos obteniendo de esta manera las bases para una arquitectura sólida y estable capaz de soportar los requisitos del sistema SIP, creando de esta manera un plano de modelo de implementación.

Es aquí donde se realizarán la reestructuración de los subsistemas, identificados previamente en la fase de inicio, así se desarrollarán las clases de diseño y diagramas de secuencia relacionados con cada caso de uso, se diseñarán las bases de datos, sentando las bases necesarias para proseguir con la última fase de este proyecto y, por consiguiente, con la codificación de la aplicación final.

#### **4.3.1 Factores que inciden en el diseño de la Interfaz Hombre-Maquina**

El diseño de una interfaz usuario-programa requiere considerar las relaciones hombre-maquina (ergonomía) a nivel de “equipo” (Hardware), de “Programación” (Software) y de “Comunicación”. Cada una de ellas tiene un papel importante, pero usualmente no todas son consideradas explícitamente en los procesos de diseño de aplicaciones del computador.

- **Ergonomía a Nivel de Equipos**

Se aplica definitivamente a la selección de dispositivo de E/S y al diseño de sus formas de uso, de manera que el suministro y la obtención de información del computador se vea facilitada (al menos no obstaculizada) por tales mecanismos. Se debe considerar para esto, las alternativas que permitan atender las características de



la población objeto y las formas de adecuar los dispositivos de E/S a estas, en aras de poner a su disposición medios de comunicación que sean fáciles de usar.

- **Ergonomía a Nivel de Programación**

Se aplica a la especificación de los elementos del procesador dinámico y de las tareas de soporte. Las actividades de los procesadores de entrada y salida que maneja el procesador dinámico se especifican mediante la descripción funcional del sistema, es decir, de las funciones que este realiza para cada tipo de usuario.

Por otra parte, la relación entre las actividades de E/S y las del procesador de soporte se definen mediante la estructura lógica del sistema.

- **Ergonomía a Nivel de Comunicación**

Se aplica al diseño de ambientes de intercomunicaciones que haciendo el mejor uso de los equipos y de la programación, favorezcan la máxima efectividad en este proceso. Algunas de las cualidades de comunicación importantes en la interfaces, son las siguientes: consistencia a lo todo lo largo de la aplicación, simplicidad y adecuación de los usuarios.

#### **4.3.2 Definición de las Zonas de Comunicación entre Usuario y Programa**

El diseño de la comunicación se centra en definir las zonas que irán asociadas a los dispositivos de E/S seleccionados, que harán posible que el usuario y el software se entiendan.

- **Zonas de Trabajo.**

Son aquellas donde el usuario tiene a disposición lo que le sirve para aprender: teorías, ejemplos y ejercicios, en ella se lleva a cabo las operaciones que este quiere efectuar sobre el objeto de estudio.



- Zonas de Control del Programa.

En estas es posible alterar el flujo y el ritmo de ejecución del programa. El control de flujo de ejecución suele estar asociado con la posible activación de las secciones del software en desarrollo a partir de los menús de trabajo desplegados, asimismo se relaciona con las posibilidades de abandono y reinicio del programa. El control del ritmo tiene que ver con la posibilidad que debe tener el usuario de decidir cuando sigue la acción, a través del uso de botones, iconos y menús para seleccionar los temas. En la figura esta zona aparece como BOTONES y ENCADEZADO Y LOGO.

- Zonas de Contexto para la Acción.

A través de estas el usuario puede saber en que unidad o modulo del sistema se encuentra, a que accesorios puede recurrir, como navegar por la aplicación, como escoger una opción, dar una respuesta. En ellas se especifica el nombre del sistema, nombre de la unidad de estudio, capítulo y tema que se desea explorar. En la figura esta zona aparece como OPCIONES DE MENU

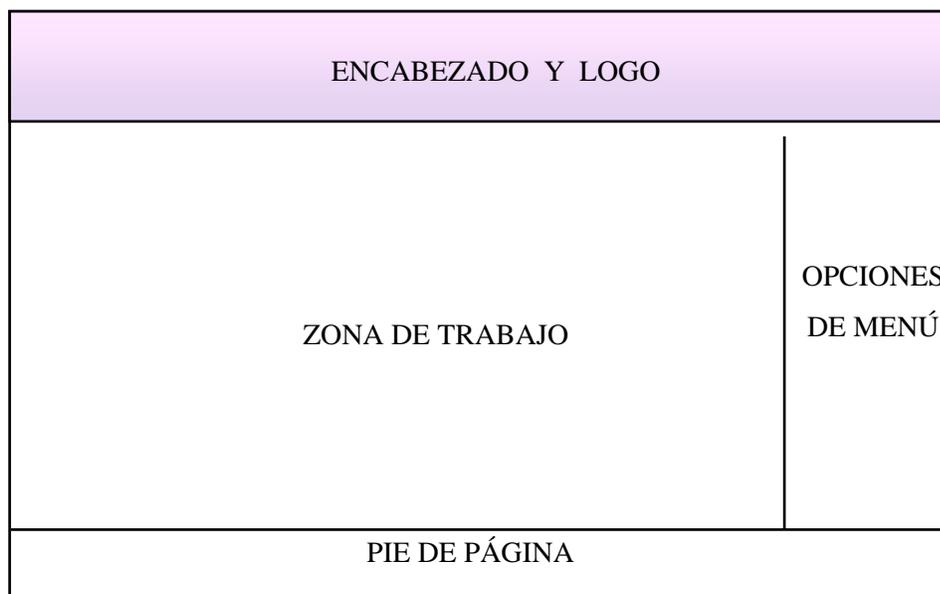


Figura 4.3 Zonas de Comunicación de la Interfaz de Usuario

Fuente: Elaboración Propia

### 4.3.3 Diagrama de Clase de Diseño

Con el diagrama de clase de diseño se visualizan las clases que componen el sistema y sus relaciones entre si. Permitiendo mostrar lo que el sistema puede hacer, además de cómo puede ser construido

Este diagrama se desarrolla a partir de información obtenida con los casos de uso del sistema, los objetos encontrados en la fase de análisis son modelados en términos de las clases a la que instancian, y las interacciones entre objetos se presentan como relaciones entre las clases instanciadas.

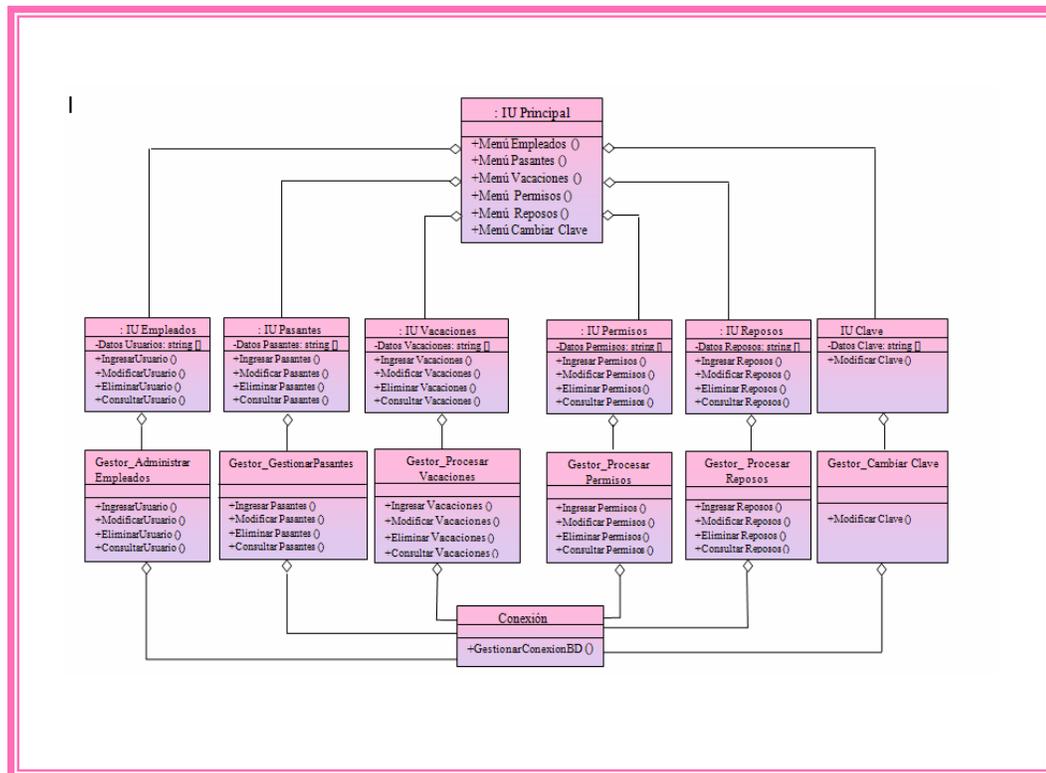


Figura 4.5 Diagrama de clase de diseño SIP

Fuente: Elaboracion Propia

### 4.3.4 Diagrama de Secuencia

Con estos diagramas se complementan los diagramas de clase de diseño, permitiendo apreciar las interacciones de los actores y la transmisión de mensajes entre las clases relacionadas a este proceso. Aquí se representan de forma jerárquica el orden en el cual se activan los objetos mediante los mensajes.

#### 4.3.4.1 Diagrama de secuencia del caso de uso Procesar Vacaciones

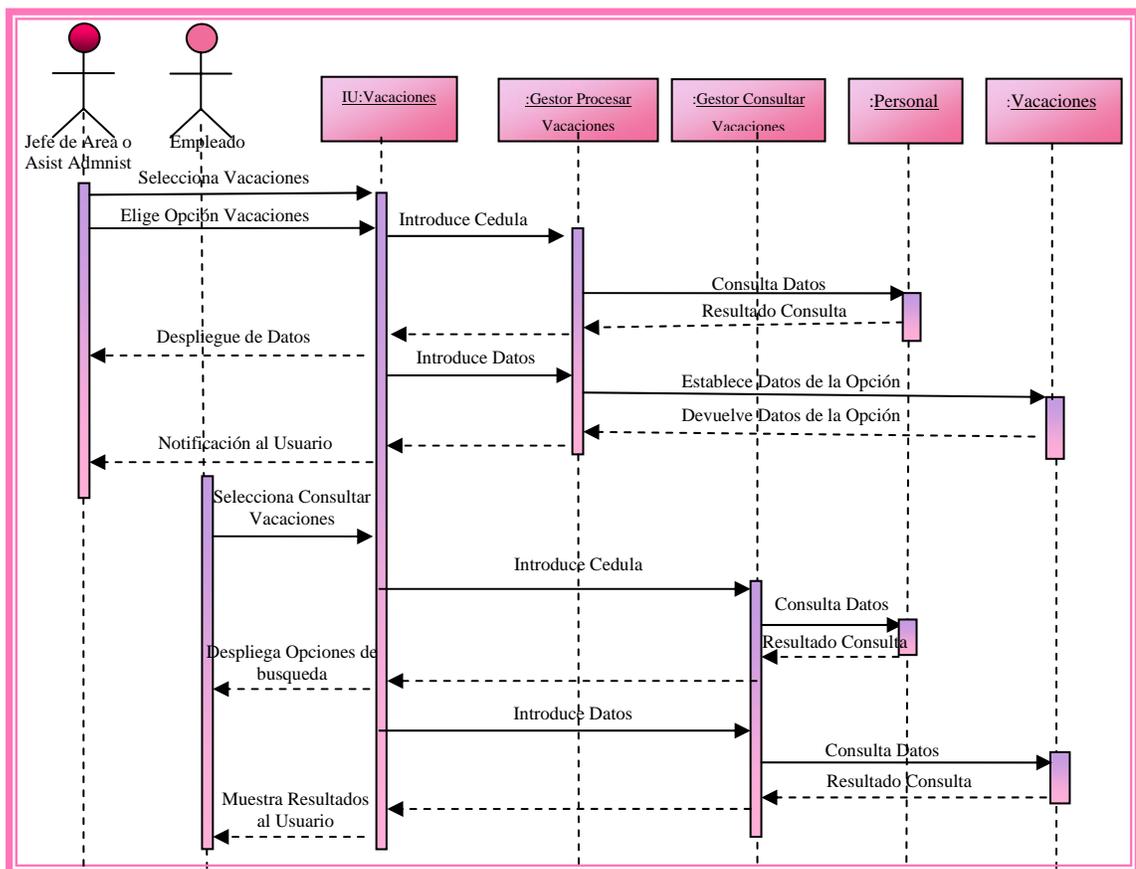


Figura 4.5 Diagrama de secuencia del caso de uso Procesar Vacaciones

Fuente: Elaboracion Propia



En el diagrama de la Figura 4.5, se muestran las opciones disponibles para el caso de uso Procesar Vacaciones y cómo interactúa el usuario y el sistema para ejecutar estas acciones. Además se pueden observar las acciones que puede ejecutar cada uno de los actores.

Para iniciar la ejecución del caso de uso el usuario activa la opción Vacaciones, esta acción genera la activación de la interfaz IU Vacaciones en la cual se presentan las diferentes operaciones que se pueden realizar.

En caso de activar la opción Ingresar, modificar o eliminar, o consultar el usuario a través de la IU Administrador, debe ingresar la cedula de la persona solicitante de la vacacion, para verificar que se encuentre en la base de datos del sistema. Esta interfaz se encarga de enviar los datos al Gestor Procesar vacacion, quien es el encargado de procesar dicha información y de comunicarse para la verificación de información con la entidad Personal. Una vez obtenida la información, el Gestor Procesar Vacacion se encarga de mostrar al usuario la IU Vacacion correspondiente para que introduzca los datos. Seguidamente el Gestor Procesar Vacacion se comunica con la entidad Vacacion para almacenar dicha data. Por último el Gestor Procesar Vacacion muestra la interfaz notificando al usuario el resultado de la operación realizada.

En el caso de solicitar la opción Consultar Vacacion a través de la IU Empleado, el usuario deberá ingresar la cedula de la persona solicitante. La interfaz IU Vacacion lleva estos datos al Gestor Consultar Vacacion, quien se comunica con la entidad Personal y autentica la información. Posterior a la autenticación el Gestor Consultar Vacacion, se ocupa de mostrar al usuario la IU Vacacion con las opciones de consultar para que introduzca los datos de la consulta. Luego el Gestor Consultar Vacacion establece comunicación con la entidad Vacacion y ejecuta la consulta

correspondiente a la entidad Vacacion, por último se emite una notificación al usuario sobre el resultado de la acción.

#### 4.3.4.2 Diagrama de secuencia del caso de uso Administrar Empleados

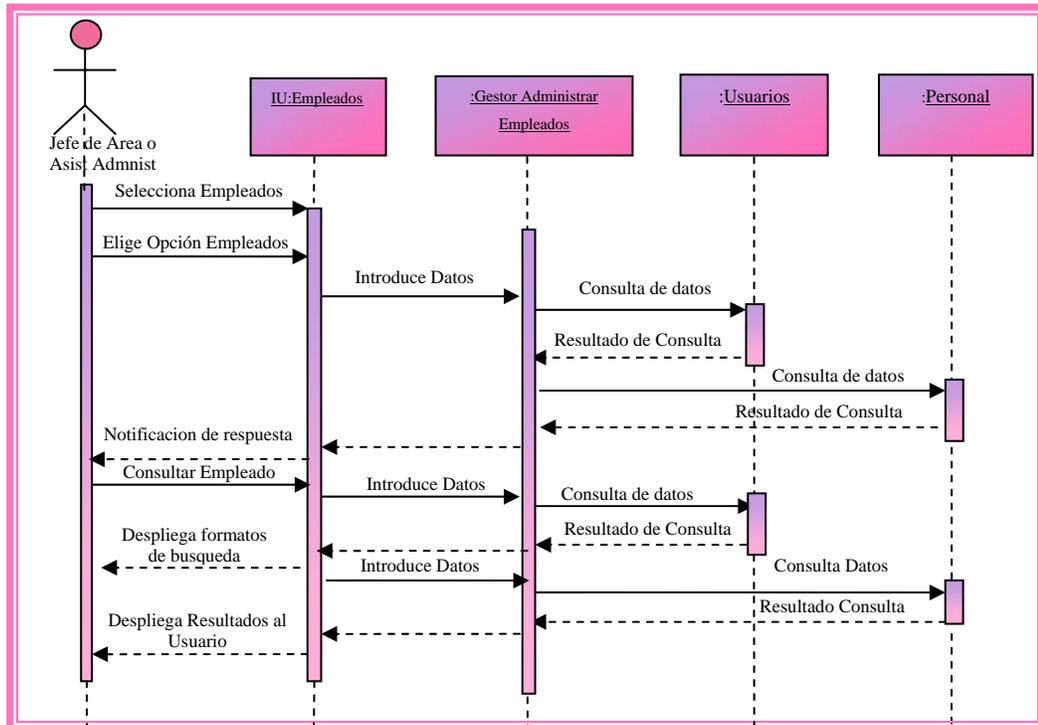


Figura 4.6 Diagrama de secuencia del caso de uso Administrar Empleados

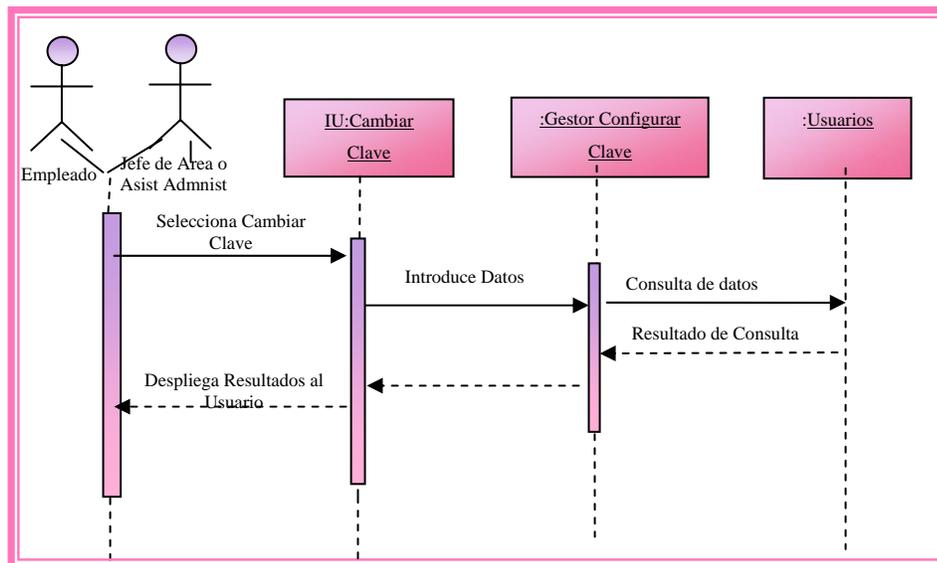
Fuente: Elaboracion Propia

En caso de activar la opción Ingresar, modificar o eliminar, el usuario a través de la IU Administrador, se ingresan los datos correspondientes al empleado sobre el cual se realiza la operación. Esta interfaz se encarga de enviar los datos al Gestor Administrar Empleado, quien es el encargado de procesar dicha información y de comunicarse con las entidades Personal y Usuario, luego el Gestor Procesar Vacacion se encarga de mostrar al usuario los resultados de la operación mediante la IU Empleados.



En el caso de solicitar la opción Consultar Vacacion a través de la IU Administrador, se introducen los datos correspondiente de la consulta del empleado. La interfaz IU Empleado lleva estos datos al Gestor Consultar Empleado, quien se comunica con las entidades Personal y Usuario y certifica la información. Posterior a la certificación el Gestor Consultar Empleado, se ocupa de mostrar al usuario la IU Empleado con las opciones de consultar para que introduzca los datos de la consulta. Luego el Gestor Consultar Vacacion establece comunicación con la entidad Personal y efectua la consulta correspondiente a la entidad Personal, por último se emite una notificación al usuario sobre el resultado de la acción.

#### 4.3.4.3 Diagrama de secuencia del caso de uso Configurar Clave



**Figura 4.7** Diagrama de secuencia del caso de uso Configurar Clave

**Fuente:** Elaboracion Propia

Al activarse el la IU Cambiar Clave por cualquier usuario del sistema, se introducen los datos, para que a través de la IU Configurar Clave se trasmita esta



información al Gestor Configurar Clave. Luego el Gestor Cconfigurar Clave se comunica con la entidad Usuarios para enviarle los datos, y finalmente se muestra los resultados obtenidos de dicha operación.

#### 4.3.5 Diseño de la Base de Datos

Es a través del diseño de una correcta base de datos que se logra evitar la redundancia de los datos y la inconsistencia de los mismos. Para la elaboración de la base de datos que usara el sistema de persona SIP, se utilizara el modelo relacional, el cual organiza la información en tablas que representan los datos y las relaciones entre ellos y debido a su sencillez la información puede ser fácilmente distribuida y compactada, y permite la posibilidad de acceder a dos o más tablas de manera simultánea y el hecho de poder identificar a cada una de ellas por medio de su o sus columnas claves

Las tablas que se describen a continuación, son las que integran la base de datos del sistema de personal (SIP):

**Tabla 4.1** Base de Datos SIP (1/2)

<b>Tabla</b>	<b>Descripcion</b>
<b>Usuarios</b>	La tabla Usuarios almacena la información correspondiente a todos los usuarios que tienen acceso al sistema, independientemente del tipo de usuario, se almacenaran datos como el login, el password y su identificación correspondiente.
<b>Personal</b>	Esta tabla contiene todos los datos personales comunes a los empleados y pasantes que laboran en la gerencia general de tributos internos de la región Nor.-oriental, es decir, nombres, apellidos, cedula de identidad, edo civil, fecha de ingreso, cargo, teléfono, dirección, entre otros
<b>Pasantes</b>	En la tabla Pasantes se recopila toda la información referente a los datos de cada uno de los pasantes de la institución, esta representada por campos como:, numero de pasante, fecha de inicio, fecha de finalización, área de estudio, supervisor entre otros.



**Tabla 4.1 Base de Datos SIP (2/2)**

<b>Tabla</b>	<b>Descripcion</b>
<b>Vacaciones</b>	Dentro de la tabla Vacaciones se encuentra almacenada todos los datos concernientes a las solicitudes de vacaciones pedidas por los funcionarios de la gerencia regional de tributos internos de la región Nor.-oriental, dentro de los campos que contiene están: cedula, fecha de ingreso, inicio de vacaciones, fin de vacaciones, año de vacaciones.
<b>Permisos</b>	En esta tabla se recogen todos los datos correspondientes a las solicitudes de permisos requeridas por los trabajadores, campos como: cedula, fecha de ingreso, inicio del permiso, fin del permiso, causa del permiso, son los que la representan. En esta tabla también se guardan todas las solicitudes de los reposos, debido a que un reposo se establece como una modalidad de un permiso y poseen los mismos atributos.
<b>División</b>	La tabla división constituye la manera en que se encuentra particionada la gerencia general de tributos internos, región Nor.-oriental de acuerdo a las actividades que allí se desempeñan, contiene los campos: id-división, nombre dela división.
<b>Area</b>	En esta tabla se almacena todas las áreas que comprenden cada una de las divisiones que conforman la estructura administrativa de la gerencia general de tributos internos región Nor.-Oriental, contiene los campos: id_área, nombre del área
<b>Area_Interes</b>	Esta tabla representa las diferentes áreas de estudio a la que puede pertenecer un pasante, permite agruparlos según sea su especialización, sus campos son: id_Area, nombre Área.

Fuente: Elaboracion Propia

El sistema de personal SIP está compuesto por 6 tablas principales y 2 tablas auxiliares, para un total de 9, las cuales constituyen la estructura necesaria para desarrollar y cumplir con los requisitos establecidos y definidos por el sistema.

En la figura 4.8 el modelo de la base de datos SIP, con las tablas que los representan y sus respectivos campos o atributos que las conforman.

En la figura 4.9 muestra las relaciones entre las tablas representado por el modelo relacional, en éstas se indican todos los atributos de cada tabla así como su clave principal y claves foráneas.



CAPITULO III  
FASE DE INICIO

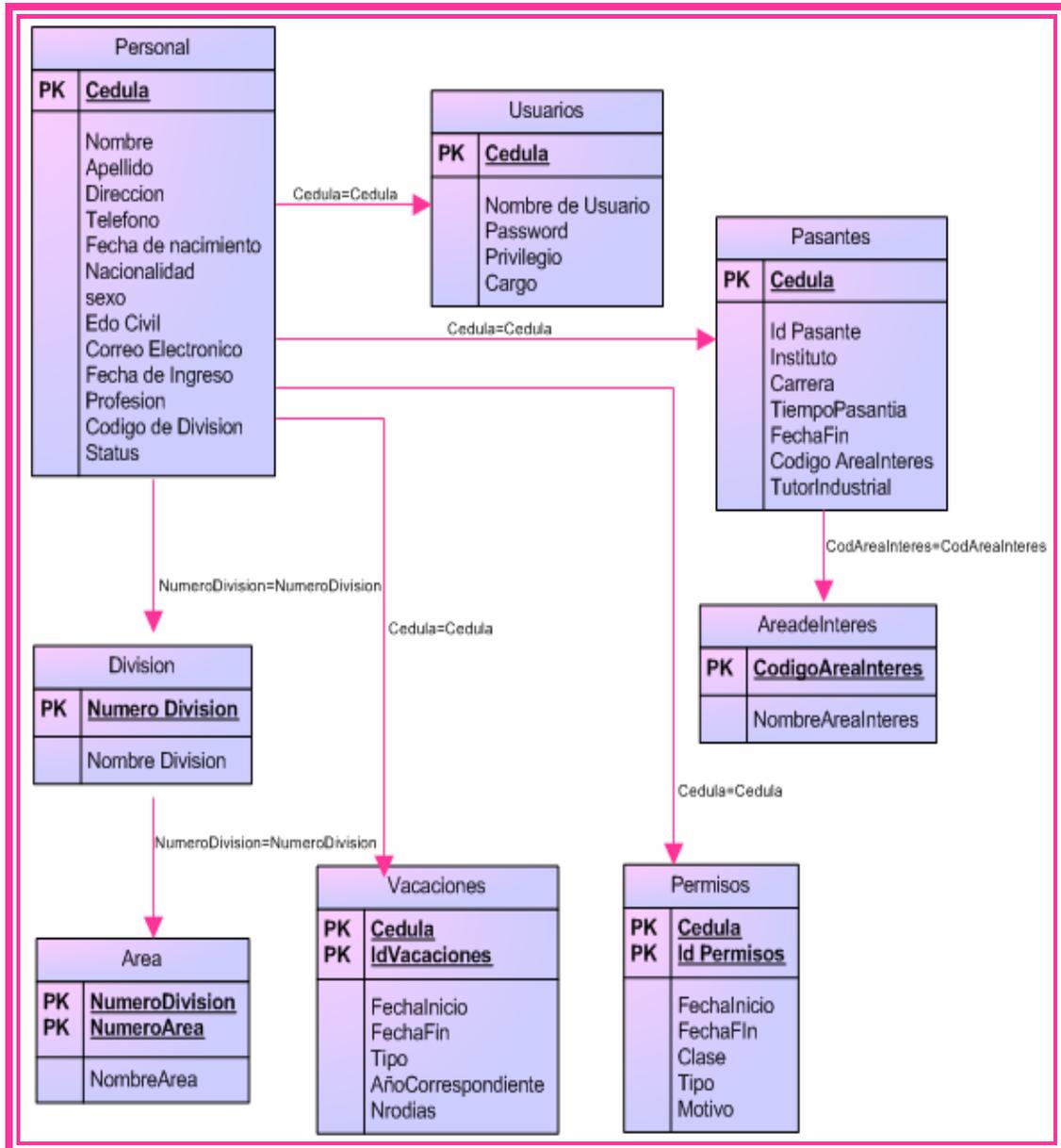


Figura 4.8 Modelo Relacional de la Base de Datos

Fuente: Elaboracion Propia



### **4.3.6 Diseño de la Arquitectura**

El diseño de la arquitectura brinda una clara perspectiva del sistema completo, describiendo los elementos que son mas significativos, arquitectónicamente hablando, incluye algunos de los subsistemas, dependencias, interfaces, colaboraciones, nodos y clases activas, que describen los cimientos del sistema, que son necesarios como base para comprenderlo y desarrollarlo.

La arquitectura esta condicionada por los casos de uso que soportan el sistema, los cuales ayudan a cubrir todas las funcionalidades significativas del mismo.

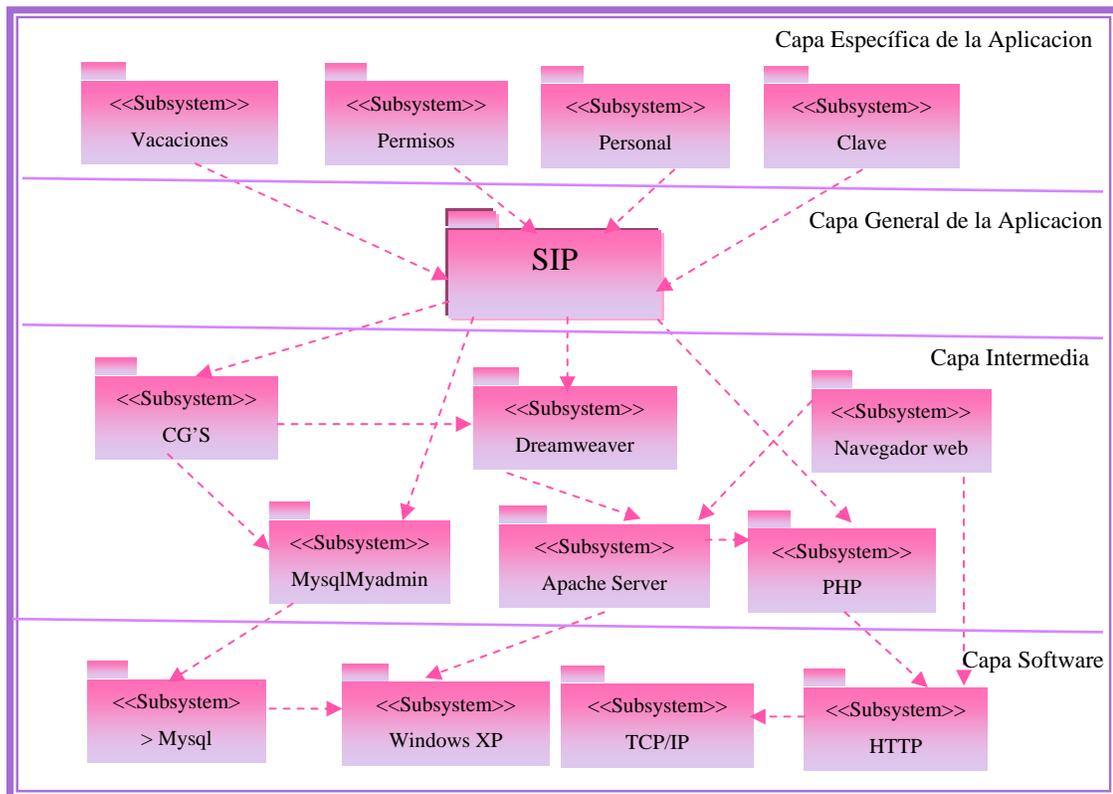
Para la organización del diseño de la arquitectura existen diferentes patrones, que se aplican para recoger soluciones estándares a problemas de la arquitectura que ocurren frecuentemente. Un patrón se define como una solución a un problema de diseño que aparece con frecuencia. Entre estos, se encuentra el patrón Layers o capas, que es aplicable a muchos tipos de sistemas, y es este el patrón que se ha seleccionado para describir la arquitectura del software. Este patrón define como organizar el modelo de capas, como se muestra en la figura.

#### **4.3.6.1 Identificación de los subsistemas**

Con la construcción de subsistemas, podemos organizar el modelo de diseño en piezas más pequeñas y más fáciles de manejar, con el fin de obtener un sistema menos complejo.

En el flujo de trabajo requisitos y de análisis, se profundizó la comprensión del sistema y sus procesos, adicionando un nuevo caso de uso, lo que genera una reformatión en la identificación de los subsistemas establecidos en la fase de inicio,

los cuales se establecieron tomando como referencia los paquetes de análisis obtenidos en dicha fase.



**Figura 4.9** Dependencia entre Subsistemas distribuida en capas

**Fuente:** Elaboracion Propia

El sistema es desarrollado utilizando el sistema operativo Microsoft® Windows XP Professional Service Pack 2, utilizaremos el lenguaje interpretado PHP 5.3 (Hipertext Php Preprocessor) ya que está especialmente diseñado para desarrollos Web y puede ser embebido en paginas HTML lo que nos permitirá crear paginas dinámicas de manera rápida y sencilla, además por ser código abierto, puede trasladarse de un sistema a otro con muy pocos o ningún cambio.



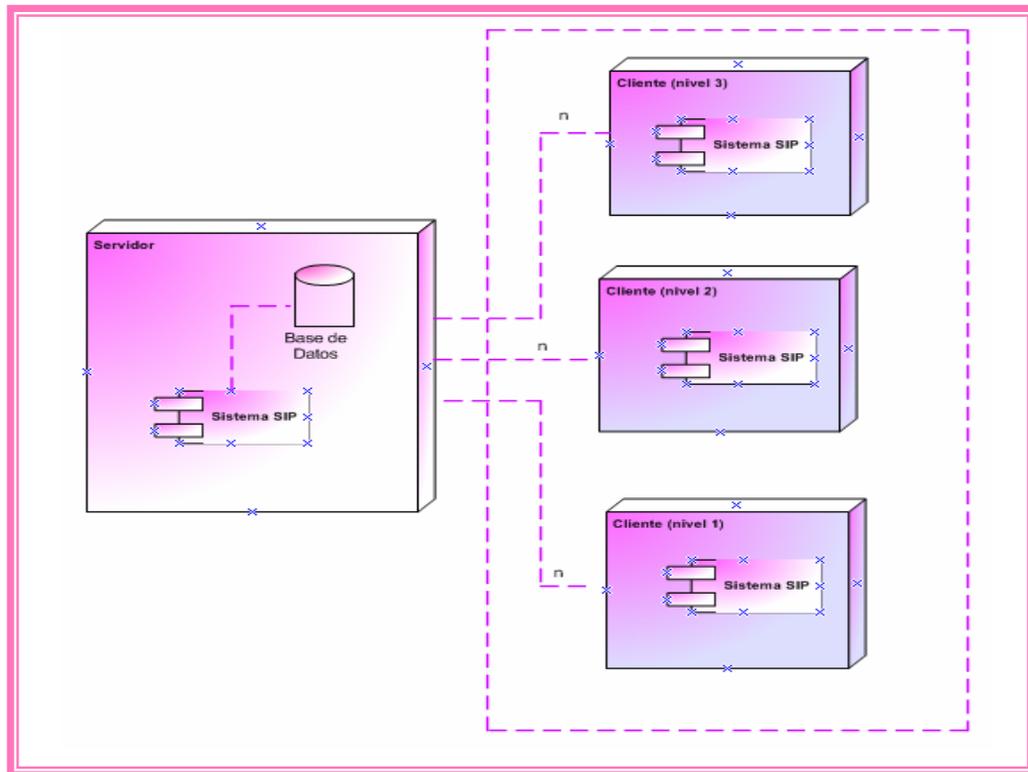
PHP es un lenguaje de Scripts del lado del servidor por lo que utilizaremos Apache Server 2.2. PHP ofrece soporte a base de datos MySQL que es la base de datos utilizada en el Sistema SIP; siendo esta una aplicación Web donde se podrán hacer muchas consultas a la base de datos. MySQL es un sistema manejador de base de datos que puede resolver rápidamente las peticiones hechas, adaptándose a la velocidad que se exige a las aplicaciones Web. El entorno de Programación a implementar es DreamWeaver, que nos permitirá desarrollar un sitio Web de forma más eficiente. En la figura 4.9 se muestra la arquitectura del sistema SIP.

Los subsistemas de aplicación, son los que conforman las capas específica y general de la aplicación, es decir, son los subsistemas que se ubica en las dos capas superiores de la arquitectura en capas. El software del sistema y la capa intermedia constituyen las bases del software que se está desarrollando, pues en ellas descansa toda la funcionalidad del mismo, por esta razón resulta de gran importancia la escogencia de los productos software que se utilizaran, haciendo una selección cuidadosa y verificando que estos encajen en la arquitectura y permitan la implementación del sistema de una manera confiable y económica.

#### **4.3.6.2 Modelo de Despliegue**

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de computo

Es necesario establecer el tipo de configuración de red con la que contara el sistema, para la construcción del modelo de despliegue. El sistema de personal se llevara a cabo bajo el funcionamiento del modelo cliente-servidor, por lo que contara con un nodo servidor y varios nodos cliente



**Figura 4.10** Diagrama de Despliegue del sistema SIP

**Fuente:** Elaboracion Propia

En la figura 4.19 se tiene el diagrama de despliegue para el software, en el que se ilustra que el sistema se ejecute en entorno cliente-servidor, específicamente sobre un nodo servidor en donde se alojan un conjunto de páginas Web que representan el sistema como tal y cierto número de nodos cliente a través de los cuales los usuarios finales tienen acceso al sistema para su exploración y se representan con interfaces distintas de acuerdo al usuario.

#### 4.3.7 Prototipo de la interfaz de usuarios

La interfaz de usuario permite la interacción entre el sistema y los usuarios, pues es a través de ella se realiza la comunicación entre ambos. Para ello se utilizan los aportes del diseño Web con el fin de crear una interfaz donde sea aplicable.



El diseño de las distintas interfaces se realiza tomando en cuenta la idea de que al usuario se le permita navegar libremente a través de las distintas secciones que comprende al software. La información contenida en las pantallas se muestra de manera tal que el usuario siempre tenga el control sobre cómo dirigirse y que evento activar, pudiendo familiarizarse rápidamente con las herramientas a su disposición tales como: botones e indicadores de enlaces.

#### **4.3.7.1 Diseño Lógico**

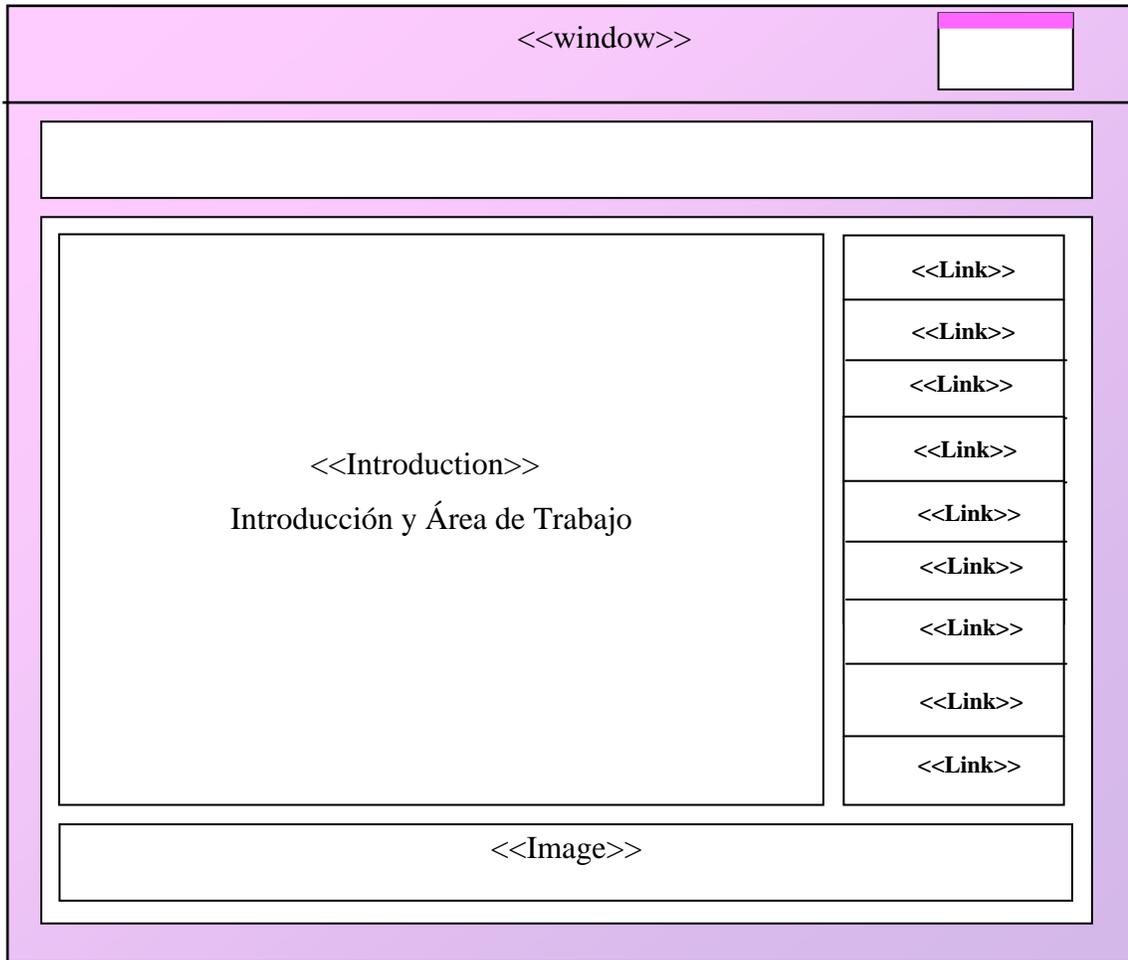
El usuario requiere acceder y manipular el contenido de la información de las clases de entidad que están relacionadas a la mayoría de los casos de uso; por tal motivo es de suma importancia contar con interfaces que permita realizar diferentes procesos como el de ingresar, consultar, modificar y eliminar data correspondiente a dichas entidades para producir un comportamiento esperado.

#### **4.3.7.2 Diseño Físico**

En esta etapa se conocen las principales necesidades de la interfaz de usuario, ya que anteriormente se han establecido los requerimientos y descrito los casos de uso involucrados en el sistema. Por lo tanto, se han diseñado los prototipos de interfaces, usando el diseño de pantallas.

Con base en la interfaz se insertan los elementos que le dan funcionalidad. En las figuras 4.11 se observa la inclusión de dichos elementos en la forma de estereotipos que representan métodos o atributos de las clases de diseño, o bien clases completas.

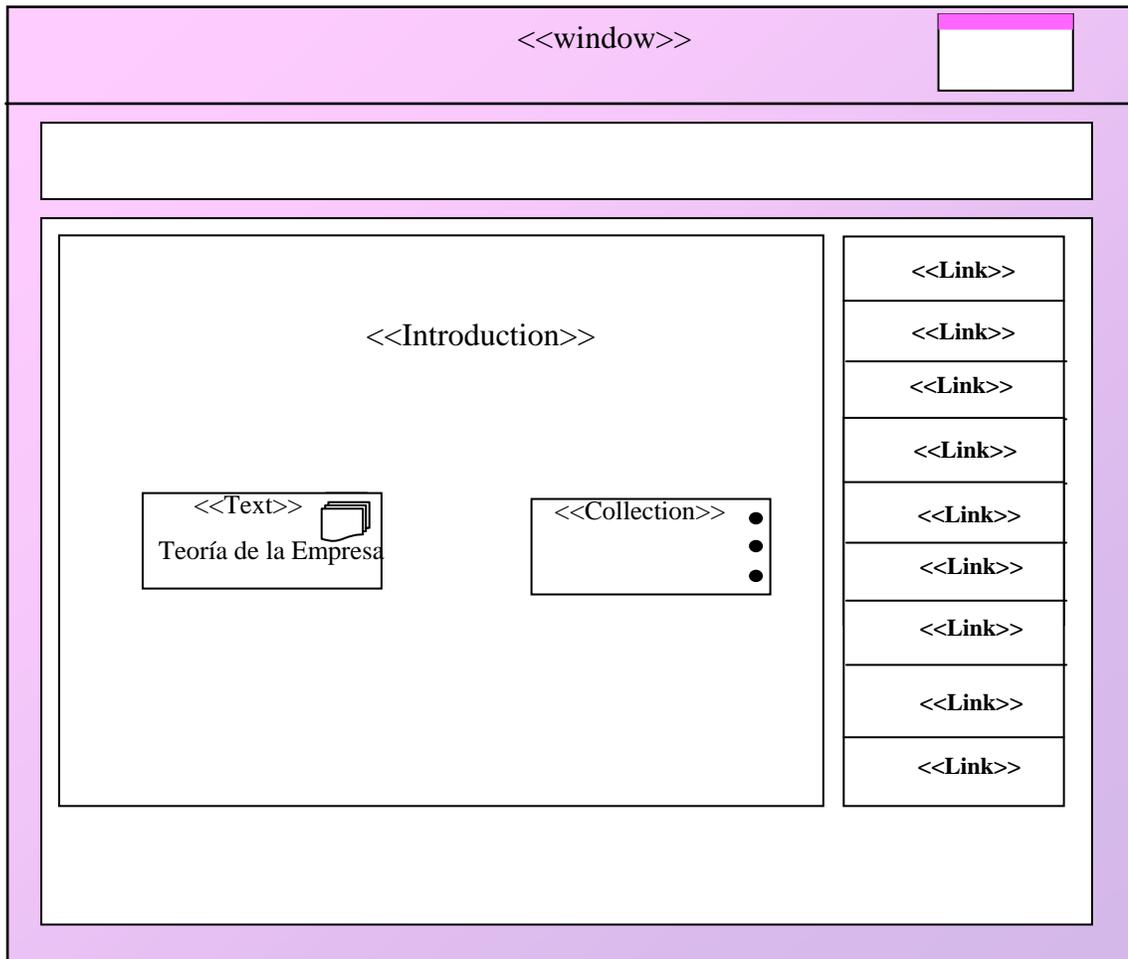
Aquí se presenta una página una colección de anclas; éstas son enlaces a páginas que corresponden al contenido de las unidades.



**Figura 4.11** Diagrama de la Vista Principal del Software

**Fuente:** Elaboracion Propia

En la figura 4.12 se tiene el diagrama de la vista del contenido de las unidades. El ambiente de la aplicación está circunscrito a un objeto de tipo ventana («window») que contiene todas las instancias de las clases que intervienen en cada instante de la ejecución del programa. El objeto de ventana esta estrechamente ligada al sistema operativo del cliente.



**Figura 4.12** Diagrama de la Vista del Contenido de Unidades.

**Fuente:** Elaboracion Propia

Los enlaces ubicados en la parte derecha constituyen el menú desplegable que permite el acceso a las unidades y páginas de la aplicación. Este menú está conformado por enlaces “Link”.

Los objetos de texto de las clases <<Image>>, aparecen al activar la página Principal del sistema SIP.



### **4.3.8 Esbozo de la Interfaz de Usuario**

En esta sección se aborda todo lo referente al diseño de algunas de las pantallas del sistema propuesto. Se diseñaron las principales pantallas del sistema, la interfaz de validación de usuario, la interfaz principal del sistema para los diferentes usuarios y las interfaces de los distintos módulos que brindan las funcionalidades al usuario ya sea administrador o consultor, que sirven a su vez de pantallas de datos. El diseño de una interfaz limpia y sencilla sin sacrificar funcionalidades, garantiza el rápido aprendizaje de la herramienta de software a la vez que minimiza las instrucciones acerca del uso del mismo. El presente sistema es basado en tecnología Web, por lo cual el navegador en el cliente visualizará las distintas pantallas del sistema.

#### **4.3.8.1 Diseño de Pantallas**

La interfaz es el medio de comunicación entre el usuario y el sistema, por medio de esta el usuario introduce datos y recibe resultados que el software presenta de una manera inteligible.

##### **4.3.8.1.1 Pantalla principal del sistema**

La figura 4.13 muestra el diseño de la pantalla principal del sistema. Esta representa la pantalla inicial en donde se ofrecen los privilegios o los usuarios. El usuario necesita introducir nombre de usuario y una clave de acceso que le facilitara el asistente de informática y dependiendo de estos valores la persona puede entrar al menú principal como administrador o como consultor de la aplicación



Figura 4.13 Diseño de la Interfaz Principal del Sistema.

Fuente: Elaboracion Propia

#### 4.3.8.1.2 Pantalla del menú administrador

La figura 4.14 muestra el diseño del menú principal del sistema al que únicamente podrá entrar aquellos usuarios con privilegios de administrar. En esta se ofrecen las funcionalidades a las que el usuario podrá acceder.

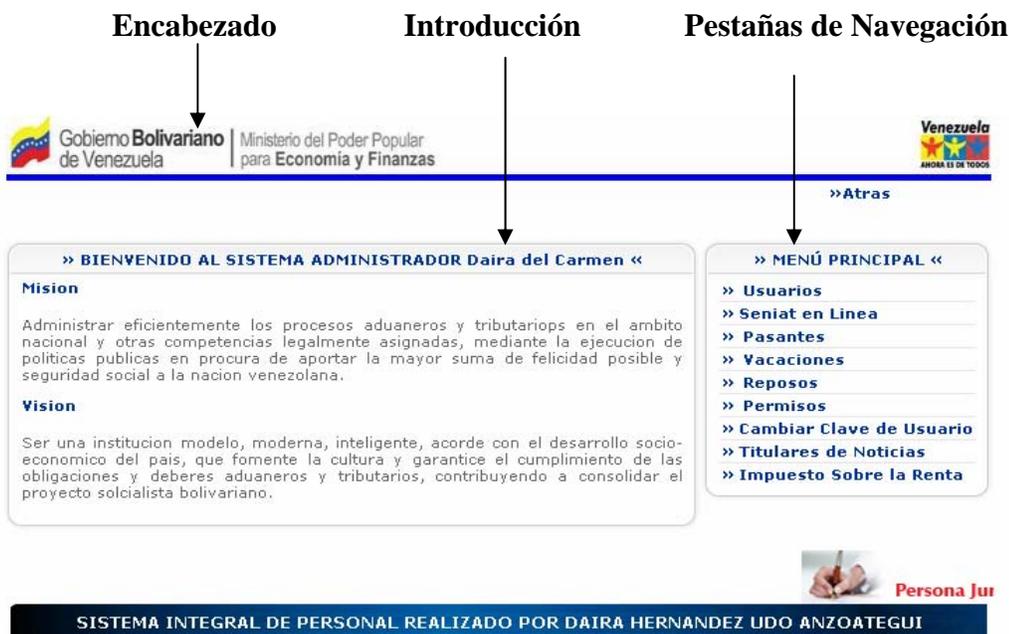


Figura 4.14 Diseño de la Interfaz del Menú Administrador.

Fuente: Elaboracion Propia



La figura anterior muestra la pantalla principal del Menú Administrador, en ella se detallan el encabezado del sistema, así como el sistema de navegación basado en pestañas desplegadas que facilita el acceso hacia los distintos módulos, además en el centro de la pantalla, se ofrecen una breve descripción del departamento de la visión y misión del Seniat.

#### 4.3.8.1.3 Pantalla del menú empleado

La figura 4.15 muestra el diseño del menú principal del sistema al que únicamente tendrán acceso los empleados para consultar acerca de su data personal correspondiente, es decir que no actualizarán información dentro del mismo. En esta se ofrecen las funcionalidades a las que el usuario podrá acceder.

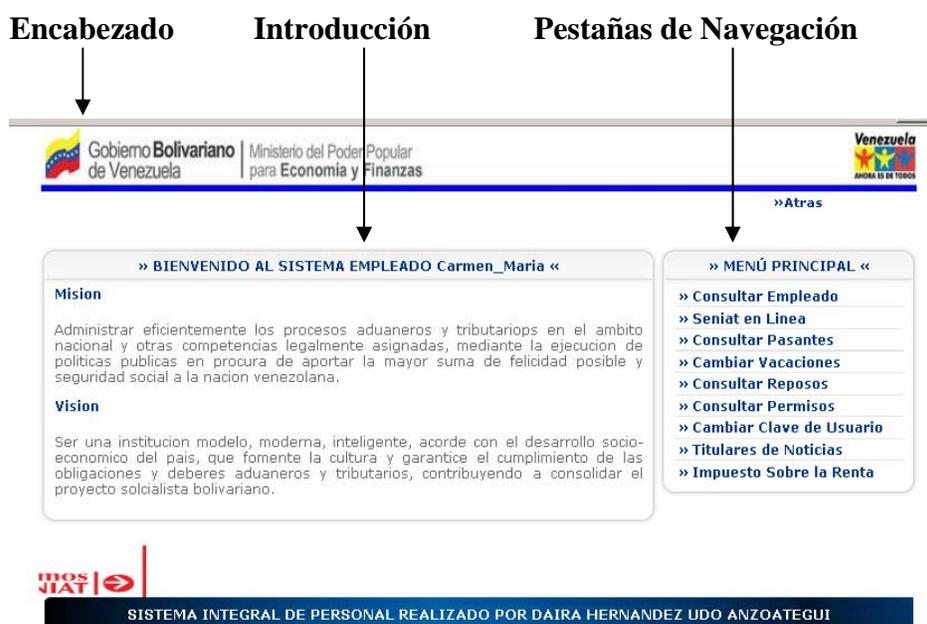


Figura 4.15 Diseño de la Interfaz del Menú Empleado

Fuente: Elaboracion Propia



#### 4.3.8.1.4 Mensaje de datos

Los mensajes de error forman parte del diseño de una interfaz de usuario, los mensajes en el sistema propuesto se muestran en una pantalla diferente y de acuerdo a la funcionalidad del mismo, permitiendo al usuario poder ver el tipo de error cometido. En la figura 4.16 se muestran errores de introducción de datos para ingresar al Menú Principal.



**Figura 4.16** Mensaje de Error en la Pantalla Principal del Sistema.

**Fuente:** Elaboracion Propia

La implementación es el centro durante las interacciones de construcción, aunque también se lleva a cabo trabajo de implementación durante la fase de elaboración, para crear la línea base ejecutable de la arquitectura.

Ya que el modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación, es natural mantener el modelo de implementación a lo largo de todo el ciclo del software.

En la figura 4.17 podemos observar las dependencias de trazas entre los componentes de implementación y clases de diseño.

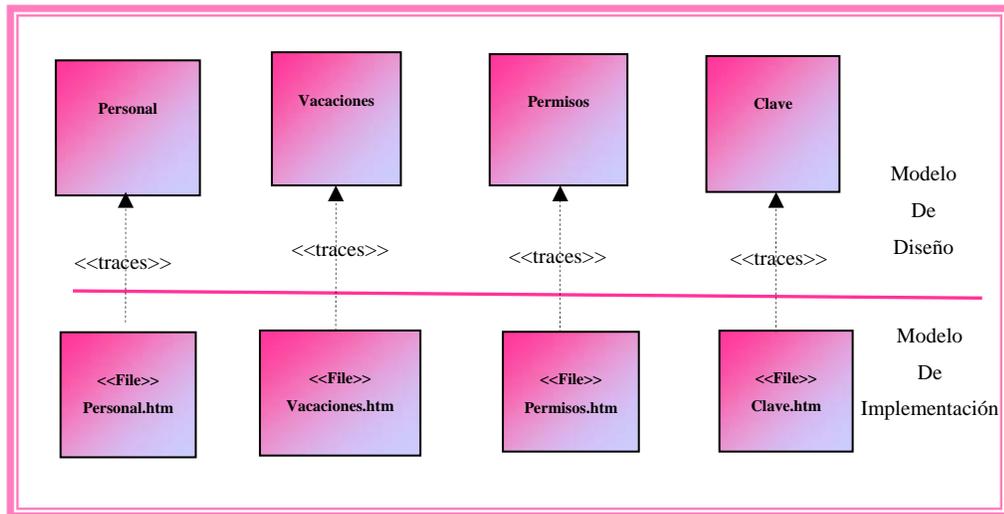


Figura 4.17 Dependencias de Trazas entre Componentes de Diseño

Fuente: Elaboracion Propia

En la figura 4.18 se muestra podemos observar las dependencias de compilación entre los componentes.

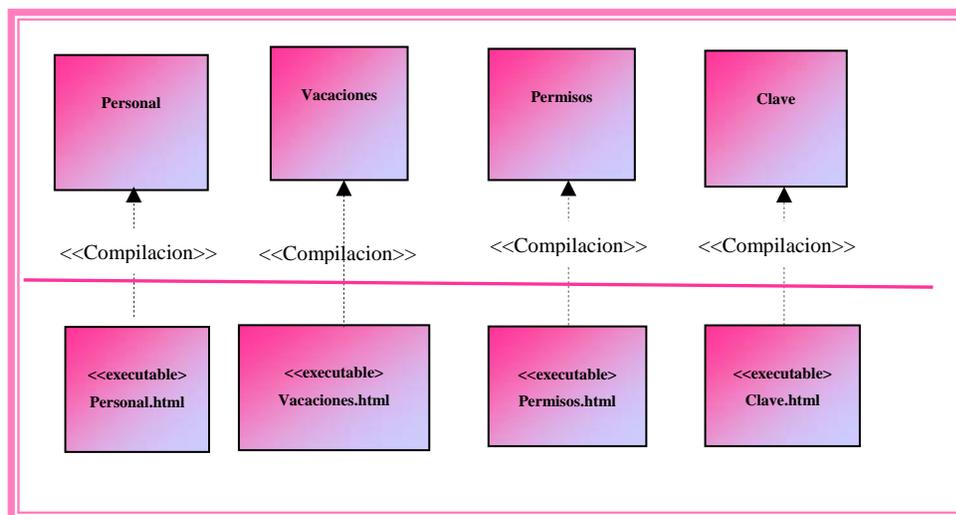


Figura 4.18 Dependencias de Compilación entre los Componentes.

Fuente: Elaboracion Propia



#### **4.5 Evaluacion de la Fase de Elaboracion**

Como producto principal de la fase de elaboración se ha obtenido la arquitectura base del sistema, es decir, el soporte principal del mismo. Todo esto quedó sintetizado por medio del modelo de casos de uso, análisis y diseño.

No se determinaron nuevos casos de uso, lo que permitió dar por establecido que los requisitos totales del sistema fueron los definidos en la fase de inicio.

En el flujo de análisis se determinó el diagrama de paquetes general, de manera que se note reflejado todo el sistema de personal SIP a través de él.

A continuación con el flujo de diseño, se procedió al estudio de las clases de diseño, por medio del diagrama de clases de diseño y diagramas de secuencia. Se obtiene la configuración básica de la red, en forma de nodos claramente identificables y de fácil definición, así como la base de datos. Igualmente se ha ampliado el esquema de la arquitectura del sistema, mostrando con más detalle los subsistemas involucrados en el proyecto SIP, desde la capa específica y general, pasando por la capa intermedia, hasta la capa de software.

En último lugar se empezó a trabajar en la aplicación en sí, creando una vista previa de la interfaz de usuario que tendrá la misma. Todo permitió acumular información necesaria para el plan de construcción y se obtuvo suficiente información para hacer realizable el proyecto, permitiendo tener una perspectiva clara para la fase de construcción.



### CAPITULO III

#### FASE DE INICIO

---

Con la realización de la fase de transición se buscara determinar las posibles fallas y defectos que pudiera presentar el producto inicial, con el propósito de corregir algunas deficiencias que no hayan sido cubiertas en las fases anteriores, y a su vez, implementar algunas mejoras. Se realizará el adiestramiento de los usuarios para la operación del sistema

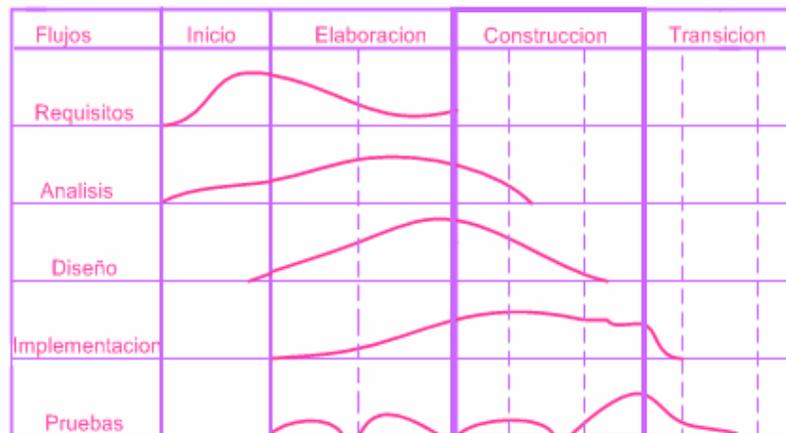
# CAPITULO V

## FASE DE CONSTRUCCION

### PLANIFICACION DE LA FASE DE CONSTRUCCION

Las fases anteriores han reducido los riesgos críticos y significativos, que pueden ser gestionados durante el plan de construcción, es por ello que en esta fase se integran todos los subsistemas, se realizan las pruebas a los componentes para verificar que la construcción se ajuste a la arquitectura base obtenida y se monitorean los riesgos de tal forma que se reflejen los riesgos reales del sistema.

En esta etapa se codifica el sistema, creando un software con capacidad operativa que pueda ser puesto en marcha lo más prontamente posible, acorde con la arquitectura planteada y con la flexibilidad suficiente para admitir cualquier tipo de modificación.



**Figura 5.1** Flujos de trabajo requeridos en la Fase de Construcción

**Fuente:** Elaboracion Propia

En el capítulo anterior se desarrollaron los flujos de trabajo Requisitos, Análisis y Diseño del proyecto SIP y parcialmente el flujo de trabajo Implementación. En éste se ampliará este último flujo de trabajo, además se



desarrollará el flujo de Pruebas, de acuerdo a lo mostrado en la figura 5.1. Para lograr los objetivos trazados, se realizará la construcción de las páginas Web para crear un software acorde a la arquitectura, satisfaciendo todos los requerimientos planteados a lo largo del desarrollo del proyecto.

## 5.1 FLUJO DE TRABAJO: IMPLEMENTACION

Este flujo de trabajo implementa clases, subsistemas y los componentes arquitectónicamente significativos como los son: código fuente, Scripts y ficheros, trabajando principalmente a partir del modelo de diseño, llevando a cabo la codificación de los componentes e implementación de la interfaces de usuario.

### 5.1.1 Construcción de la Base De Datos

Para la construcción de la base de datos se eligió el servidor MySQL, como se dijo con anterioridad, el cual es un software libre que puede resolver rápidamente peticiones hechas, adaptándose a la velocidad que exige SIP como aplicación Web, cumple con los estándares del ANSI SQL, es de fácil uso, su disponibilidad es de manera gratuita y es compatible tanto con el sistema operativo Windows como con LINUX. En dicha sección se mostró la estructura básica y las relaciones entre las tablas pertenecientes a SIP, las cuales se conservan en este capítulo.

**Tabla 5.1** Estructura de las tablas de la Base de Datos de SIP (1/3)

Tabla	Script
Usuarios	<pre>CREATE TABLE IF NOT EXISTS `Usuarios` (   `usuario` varchar(30) NOT NULL,   `cedula` varchar(15) NOT NULL,   `clave` varchar(20) NOT NULL,   `administrar` varchar(20) NOT NULL,   `cargo` varchar(25) NOT NULL,    PRIMARY KEY (`usuario`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1;</pre>

**Funete:** Elaboración Propia.



**Tabla 5.1 Estructura de las tablas de la Base de Datos de SIP (2/3)**

Tabla	Descripcion
<b>Personal</b>	<pre>CREATE TABLE IF NOT EXISTS `Personal` (   `cedula` varchar(15) NOT NULL,   `nombre` varchar(30) NOT NULL,   `apellido` varchar(30) NOT NULL,   `direccion` varchar(60) NOT NULL,   `telefono` varchar(25) NOT NULL,   `fechadenac` varchar(15) NOT NULL,   `nacionalidad` char(12) NOT NULL,   `sexo` char(12) NOT NULL,   `edocivil` varchar(15) NOT NULL,   `correoelect` varchar(40) NOT NULL,   `fechaingreso` varchar(15) NOT NULL,   `profesion` varchar(30) NOT NULL,   `coddivision` int(2) NOT NULL,   `estatus` varchar(25) NOT NULL,   PRIMARY KEY (`cedula`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1;</pre>
<b>Pasantes</b>	<pre>CREATE TABLE IF NOT EXISTS `Pasantes` (   `cedula` varchar(15) NOT NULL,   `nropasante` int(5) NOT NULL,   `instituto` varchar(30) NOT NULL,   `carrera` varchar(40) NOT NULL,   `tiempopasantia` varchar(10) NOT NULL,   `fechafin` varchar(15) NOT NULL,   `codareadeinteres` int(2) NOT NULL,   `tutor` varchar(30) NOT NULL,   PRIMARY KEY (`cedula`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1;</pre>
<b>Vacaciones</b>	<pre>CREATE TABLE IF NOT EXISTS `Vacaciones` (   `cedula` varchar(15) NOT NULL,   `idvacaciones` int(10) NOT NULL AUTO_INCREMENT,   `coddivision` int(2) NOT NULL,   `fechaInicio` varchar(15) NOT NULL,   `fechafin` varchar(15) NOT NULL,   `ndias` varchar(11) NOT NULL,   `anoscorrespondiente` varchar(11) NOT NULL,   `tipo` varchar(12) NOT NULL,   PRIMARY KEY (`cedula`,`idvacaciones`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;</pre>
<b>Permisos</b>	<pre>CREATE TABLE IF NOT EXISTS `Permisos` (   `cedula` varchar(15) NOT NULL,   `idpermisos` int(12) NOT NULL AUTO_INCREMENT,   `fechainicio` varchar(15) NOT NULL,   `fechafinal` varchar(15) NOT NULL,   `motivo` varchar(80) NOT NULL,   `coddivision` int(2) NOT NULL,   `clase` varchar(10) NOT NULL,   `tipo` varchar(25) NOT NULL,   PRIMARY KEY (`cedula`,`idpermisos`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;</pre>



**Tabla 5.1 Estructura de las tablas de la Base de Datos de SIP (3/3)**

Tabla	Descripcion
<b>División</b>	CREATE TABLE IF NOT EXISTS `Division` ( `coddivision` int(2) NOT NULL, `nombredivision` varchar(30) NOT NULL, PRIMARY KEY (`coddivision`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
<b>Área</b>	CREATE TABLE IF NOT EXISTS `AreaDivision` ( `idarea` int(2) NOT NULL, `coddivision` int(2) NOT NULL, `area` varchar(30) NOT NULL, PRIMARY KEY (`idarea`,`coddivision`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
<b>ÁreadeInetres</b>	CREATE TABLE IF NOT EXISTS `AreadeInteres` ( `codareadeinteres` int(2) NOT NULL, `nombreatreadeinteres` varchar(30) NOT NULL, PRIMARY KEY (`codareadeinteres`) ) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Fuente: Elaboracion Propia

### 5.1.2 Modelo de Implementación

El modelo de implementación describe como los elementos del modelo de diseño se implementan en términos de componentes, como se organizan estos de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y como dependen los componentes unos de otros.

Para implementar el software se utilizó un entorno cliente-servidor en el que se aplicaron tecnologías Web, básicamente páginas de cliente y páginas de servidor.

En el lado del cliente se usan páginas de HTML dinámicas (DHTML), las cuales muestran los contenidos de las unidades, e interactúan con aplicaciones.

En el lado del servidor se empleó el lenguaje de servidor PHP para la realización de las modificaciones y actualizaciones de las páginas que lo ameriten y para el

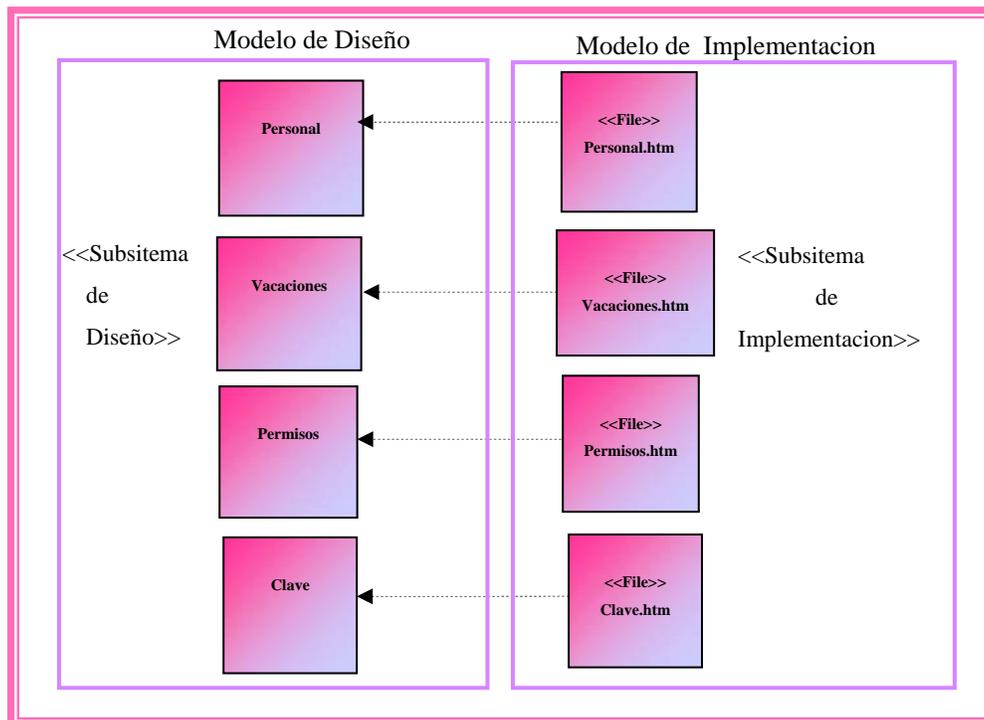


mantenimiento de la base de datos Usuarios, Pasantes, Vacaciones, Permisos, entre otras.

### 5.1.2.1 Subsistemas de Implementación

Los subsistemas de implementación proporcionan una forma de organizar los artefactos del modelo de implementación en trozos más manejables. Un subsistema puede implementar y así proporcionar las interfaces que representan la funcionalidad que exportan en forma de operaciones.

Los subsistemas de implementación están muy relacionados con los subsistemas de diseño en el modelo de diseño. De hecho, los subsistemas de implementación deberían seguir la traza uno a uno de sus subsistemas de diseño correspondientes.



**Figura 5.2** Subsistemas de implementación a partir de Subsistemas de Diseño

Fuente: Elaboración Propia



En la figura 5.2 se tienen los modelos de diseño y de implementación con sus respectivos subsistemas y trazas.

### 5.1.3 Implementación de la Interfaz de Usuario

En esta fase se diseñaron las interfaces restantes del software, estas se diseñaron pensando en la facilidad de uso lo cual reduce los tiempos de manipulación de la herramienta y búsqueda de la información, se hace énfasis en un diseño sencillo y homogéneo que permita al usuario acceder a los controles y enlaces de manera intuitiva.

Las figuras siguientes presentan las interfaces de los diferentes módulos del software.

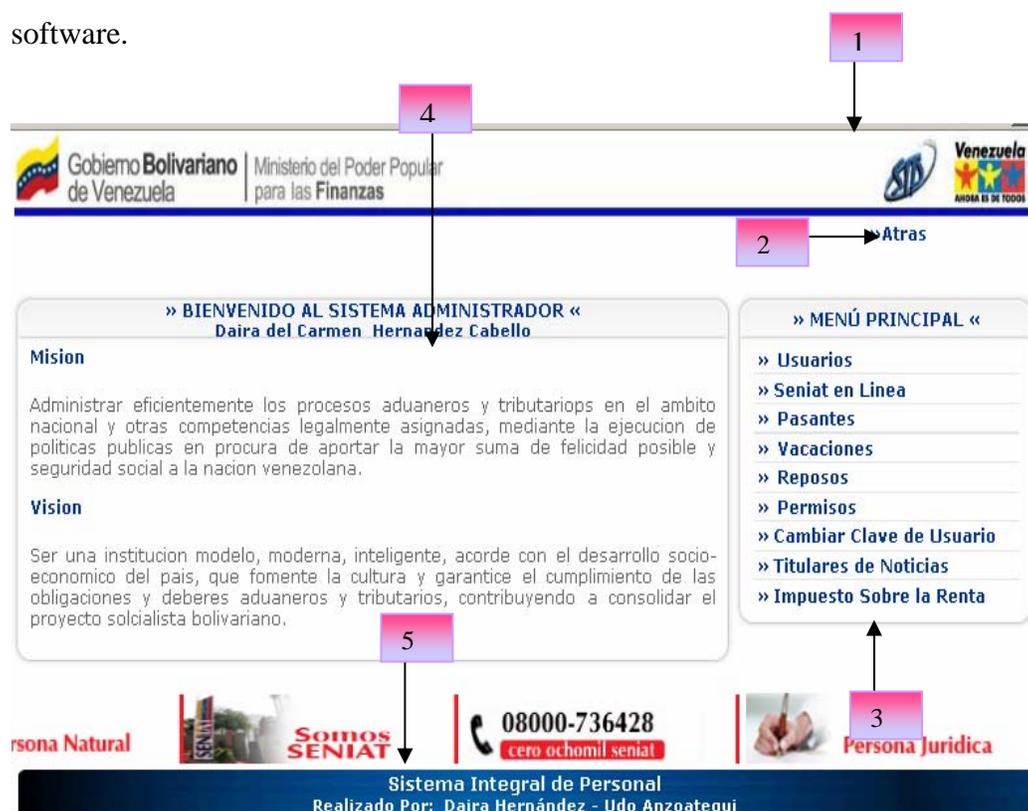


Figura 5.3 Diseño Básico de la Interfaz del Sistema

Fuente: Elaboracion Propia



En la figura 5.3 se muestran los elementos básicos que forman parte de las diferentes interfaces de la aplicación, estos se describen a continuación:

1. Encabezado de la página y logo de identificación de SIP
2. Enlace a la página principal de la aplicación.
3. Menú Principal del Sistema.
- 4.- de Introducción, se ofrece una breve descripción del departamento de planta interna.
- 5.- Pie de Página.

▪ **Interfaz de Agregar Usuarios:**

En esta interfaz el administrador de la aplicación ingresa los datos de los empleados, tanto personales como administrativos. El encargado de ingresar estos datos será el Asistente Administrativo o el Jefe del Área de Personal

En la figura 5.4 se muestran los elementos básicos que conforman la interfaz de agregar usuarios:

1. Encabezado de la página y logo de identificación de SIP
2. Enlace a la página principal de la aplicación.
3. Área de Trabajo, en esta se ofrece los campos que el administrador debe llenar para introducir un nuevo usuario.



## CAPITULO V FASE DE CONSTRUCCIÓN

1

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela

>> Menú 2

**Datos Personales de los Usuarios**

Nombre Completo	Carmen Maria
Apellido completo	Perez Gonzalez
cedula	4000000
Direccion	Calle el Milagro, Sector PLC
Telefono	0416332100
Fecha de Nacimiento	11/05/1985
Nacionalidad	Venezolana
Sexo	Femenina
Edo civil	Casada
Correo Electronico	CarmenPerez@hotmail.com
Fecha de Ingreso	07/10/2003
Profesion	Licenciado en Administracion
Cargo	Jefe de Personal
Division	Administracion
Area de Division	RRHH
Nombre de Usuario	CarmenPerez14
Privilegio de administracion	no
Password	*****
Estatus	activo

3

Limpiar Enviar

INTRODUCE LOS DATOS DEL EMPLEADO Y PRESIONE ACEPTAR

**Figura 5.4** Diseño Básico de la Interfaz del Agregar Usuarios

**Fuente:** Elaboracion Propia

En la figura 5.5 se muestra una interfaz en donde se muestra el mensaje de que el usuario fue ingresado satisfactoriamente dentro del sistema, luego se muestra menú principal al oprimir aceptar.



## CAPITULO V

### FASE DE CONSTRUCCIÓN

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela  
ANHO 11 DE 10001

»Menú

**El usuario se Inserto Correctamente en la Base de Datos**

Nombre Completo	Carmen_Maria
Apellido Completo	Perez_Gonzalez
Cedula de Identidad	4000000
Dirección	Calle_el_Milagro_Sector_PLC
Telefono	0416332100
Fecha de Nacimiento	11/05/1985
Nacionalidad	Venezolana
Sexo	Femenino
Estado Civil	Casada
Correo Electronico	CarmenPerez@hotmail.com
Fecha Ingreso	07/10/2003
Profesión	Licenciada_en_Administracion
Cargo	Jefe_de_Personal
Division	Administracion
Area	RRHH

Aceptar

**Figura 5.5** Diseño Básico de la Interfaz del Agregar Usuarios

**Fuente:** Elaboracion Propia

#### ▪ Interfaz de Consultar Usuarios:

En esta interfaz el administrador de la aplicación (que para el sistema de personal SIP, puede ser el Jefe de Área o el Asistente Administrativo) escoge el campo a través del cual desea hacer la consulta, por ejemplo un usuario se puede consultar por la cedula, su cargo o la división a la que pertenecen. El resultado sería el mostrado en la figura 5.7, en donde aparecen los datos personales del usuario.

1. Encabezado de la página y logo de identificación de SIP.
2. Enlace a la página principal de la aplicación.
3. Área de Trabajo, en esta se ofrece el campo que el administrador debe seleccionar para consultar al usuario.



## CAPITULO V FASE DE CONSTRUCCIÓN

1

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela  
AYUDA EN TODO

»Menú 2

Cedula  Cargo  Division

Aceptar 3

INTRODUCE LA CEDULA, CARGO O LA DIVISIÓN DEL EMPLEADO Y PRESIONE ACEPTAR

Figura 5.6 Diseño Básico de la Interfaz del Consultar Usuarios

Fuente: Elaboracion Propia

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela  
AYUDA EN TODO

» Menú

Cedula  Cargo  Division

Aceptar

**Datos Personales de los Usuarios**

Nombre Completo	Carmen_Maria
Apellido Completo	Perez_Gonzalez
Cedula de Identidad	4000000
Dirección	Calle_el_Milagro_Sector_PLD
Telefono	0416332100
Fecha de Nacimiento	11/05/1985
Nacionalidad	Venezolana
Sexo	Femenino
Estado Civil	Casada
Correo Electronico	CarmenPerez@hotmail.com
Fecha Ingreso	07/10/2003
Profesión	Licenciada_en_Administracion
Cargo	Jefe_de_Personal
Division	Administracion
Area	RRHH

Figura 5.7 Diseño de la Interfaz del Consultar Usuarios

Fuente: Elaboracion Propia



## CAPITULO V FASE DE CONSTRUCCIÓN

### ▪ Interfaz de Agregar Pasantes:

En esta interfaz el administrador de la aplicación ingresa los datos de los Pasantes, tanto personales como administrativos. El encargado de ingresar estos datos será el Asistente Administrativo o el Jefe del Área de Personal

En la figura 5.4 se muestran los elementos básicos que conforman la interfaz de agregar Pasantes:

The screenshot shows a web application interface for adding apprentices. The header includes the logo of the 'Gobierno Bolivariano de Venezuela' and the 'Ministerio del Poder Popular para las Finanzas'. The main form is titled 'Datos Personales de los Pasantes' and contains the following fields:

Nombre Completo	Maria Celeste
Apellido completo	Cabrera Arca
cedula	14333888
Direccion	Calle Monagas casa 21. Tierra Adentro. PLC
Telefono	04165814140
Fecha de Nacimiento	07/01/1981
Nacionalidad	Venezolana
Sexo	Femenino
Edo civil	Soltera
Correo Electronico	Maria_cabrera@hotmail.com
Fecha de Ingreso	12/01/2009
Profesion	Estudiante
Division	Administracion
Area de Division	informatica
Estatus	Activo
Numero de Pasante	0010
Universidad/Instituto	Universidadde Oriente
Carrera	Ingenieria de Sistemas
Tiempo Duracion de pasantias	6 meses
Fecha Finalizacion Pasantias	07/07/2009
Area de Interes	Informatica
Tutor	Luisa Campos

At the bottom of the form are two buttons: 'Limpiar' and 'Enviar'.

**Figura 5.8** Diseño Básico de la Interfaz del Agregar Pasantes

**Fuente:** Elaboracion Propia



## CAPITULO V

### FASE DE CONSTRUCCIÓN

- 1.- Encabezado de la página y logo de identificación de SIP
- 2.- Enlace a la página principal de la aplicación.
- 3.- Área de Trabajo, en esta se ofrece los campos que el administrador debe llenar para introducir un nuevo Pasante.

En la figura 5.9 se muestra una interfaz en donde se muestra el mensaje de que el pasante fue ingresado satisfactoriamente dentro del sistema, luego se muestra menú principal al oprimir aceptar.

El Pasante se Inserto Correctamente en la Base de Datos

Nombre Completo	Maria_Celeste
Apellido Completo	Cabrera_Arcia
Cedula de Identidad	14333888
Dirección	Calle_Monagas_casa_21_Tierra_Adentro_PLC
Telefono	04165814140
Fecha de Nacimiento	07/01/1981
Nacionalidad	Venezolana
Sexo	Femenino
Estado Civil	Soltera
Correo Electronico	Maria_cabrera@hotmail.com
Fecha Ingreso	12/12/2009
Profesión	Estudiante
Division	Administracion
Area	informatica

» Menú

Aceptar

**Figura 5.9** Diseño Básico de la Interfaz del Agregar Pasantes

**Fuente:** Elaboracion Propia



▪ **Interfaz de Consultar Pasantes:**

En esta interfaz el administrador de la aplicación (que para el sistema de personal SIP, puede ser el Jefe de Área o el Asistente Administrativo) escoge el campo a través del cual desea hacer la consulta, por ejemplo un usuario se puede consultar por la cedula, su área de interés(área de estudio) o la división a la que pertenecen. El resultado sería el mostrado en la figura 5.10, en donde aparecen los datos personales del Pasante.

- 1.- Encabezado de la página y logo de identificación de SIP.
- 2.- Enlace a la página principal de la aplicación.
- 3.- Área de Trabajo, en esta se ofrece el campo que el administrador debe seleccionar para consultar al Pasante.

**Figura 5.9 (a).** Diseño Básico de la Interfaz del Consultar Pasantes

**Fuente:** Elaboracion Propia



## CAPITULO V

### FASE DE CONSTRUCCIÓN

**Figura 5.10** Diseño de la Interfaz del Consultar Pasantes

**Fuente:** Elaboracion Propia

#### ▪ Interfaz de Agregar Permisos:

En esta interfaz el administrador de la aplicación ingresa los datos de las solicitudes de Permisos. Para agregar un permiso, primero el administrador del sistema introduce la cédula del empleado que requiere la solicitud. En la figura 5.11 se muestran los elementos básicos que conforman la interfaz de agregar Permisos.

- 1.- Encabezado de la página y logo de identificación de SIP.
- 2.- Enlace a la página principal de la aplicación.
- 3.- Área de Trabajo, en esta se ofrece el campo que el administrador debe seleccionar para buscar al empleado que requiere la solicitud de Permiso.



## CAPITULO V FASE DE CONSTRUCCIÓN

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela AHORA ES DE TODOS

>>Menú

Cedula 4000000

Aceptar

INTRODUCE ALGUN DATO DEL USUARIO Y PRESIONE ACEPTAR

Figura 5.11 Diseño de la Interfaz del Agregar Permisos

Fuente: Elaboracion Propia

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela AHORA ES DE TODOS

>>Menú

**Datos del Usuarios**

Nombre Completo	Carmen_Maria
Apellido completo	Perez_Gonzalez
cedula	4000000
Fecha de Ingreso	07/10/2003
Division	Administracion
Area de Division	RRHH

**Solicitud de Permisos**

Fecha de Inicio	07/12/2010
Fecha de Finalizacion	07/16/2010
Motivo	Curso de Especializacion
Clase de Solicitud	Permiso

Aceptar

Figura 5.12 Diseño de la Interfaz del Agregar Permisos

Fuente: Elaboracion Propia



El resultado de la solicitud de Permiso procesada correctamente se muestra en la figura 5.13, en donde aparecen algunos datos personales del usuario.

**Gobierno Bolivariano**  
de Venezuela

Ministerio del Poder Popular  
para las Finanzas

Venezuela  
AHORA ES DE TODOS

[» Menú](#)

**Datos del Basicos del Empleado**

<b>Nombre Completo</b>	<input type="text" value="Carmen_Maria"/>
<b>Apellido Completo</b>	<input type="text" value="Perez_Gonzalez"/>
<b>Cedula de Identidad</b>	<input type="text" value="4000000"/>
<b>Fecha Ingreso</b>	<input type="text" value="07/10/2003"/>
<b>Division</b>	<input type="text" value="Administracion"/>
<b>Area</b>	<input type="text" value="RRHH"/>
<b>Numero de Permiso</b>	<input type="text" value="21"/>
<b>Fecha de Inicio</b>	<input type="text" value="07/12/2010"/>
<b>Fecha de Finalizacion</b>	<input type="text" value="07/16/2010"/>
<b>Motivo</b>	<input type="text" value="Curso_de_Especializacion"/>
<b>Clase de Solicitud</b>	<input type="text" value="Permiso"/>
<b>Estatus de Solicitud</b>	<input type="text" value="Pendiente"/>

**Figura 5.13** Diseño de la Interfaz del Agregar Permisos

**Fuente:** Elaboracion Propia

- **Interfaz de Modificar Permisos:**

En esta interfaz el administrador de la aplicación modifica los datos de las solicitudes de Permisos. Para Aprobar un permiso, primero el administrador del sistema introduce la cédula del empleado que tiene la solicitud de permiso pendiente, para luego cambiar el estado de pendiente a aprobado. En la figura 5.15 se muestran los elementos básicos que conforman la interfaz de agregar Permisos.

- 1.- Encabezado de la página y logo de identificación de SIP.
- 2.- Enlace a la página principal de la aplicación.



## CAPITULO V

### FASE DE CONSTRUCCIÓN

3.- Área de Trabajo, en esta se ofrece el campo que el administrador debe seleccionar para buscar al empleado que requiere la solicitud de Permiso.

1

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela

» Menú

2

3

Numero de Solicitud 21

Aceptar

LA SOLICITUD NO SE ENCUENTRA REGISTRADA EN LA BASE DE DATOS

**Figura 5.14** Diseño de la Interfaz del Modificar Permisos

**Fuente:** Elaboracion Propia

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para las Finanzas

Venezuela

» Menú

**Datos del Permiso**

Nombre Completo	Carmen_Maria
Apellido completo	Perez_Gonzalez
Cedula de Identidad	4000000
Cargo	Jefe_de_Personal
Division	Administracion
Area	RRHH
Numero de Permiso	21
Fecha de Inicial	07/12/2010
Fecha de Finalizacion	07/16/2010
Motivo	Curso_de_Especializacion
Clase de Solicitud	Permiso
Estatus de Solicitud	Aprobado

Modificar

**Figura 5.15** Diseño de la Interfaz del Modificar Permisos

**Fuente:** Elaboracion Propia



El resultado de la aprobación de la solicitud de Permiso procesada correctamente se muestra en la figura 5.16, en donde aparecen algunos datos personales del usuario.

Datos Personales de los Usuarios	
Nombre Completo	Carmen_Maria
Apellido Completo	Perez_Gonzalez
Cedula de Identidad	4000000
Cargo	Jefe_de_Personal
Division	Administracion
Area	RRHH
Numero de Permiso	21
Fecha de Inicio	12/07/2010
Fecha de Finalizacion	07/16/2010
Motivo	Curso_de_Especializacion
Clase de Solicitud	Permiso
Estatus de Solicitud	Aprobado

**Figura 5.16** Diseño de la Interfaz del Agregar Permisos

**Fuente:** Elaboracion Propia

- **Interfaz de Agregar Reposos:**

En esta interfaz el administrador de la aplicación ingresa los datos de las solicitudes de Reposos. Para agregar un Reposo, primero el administrador del sistema introduce la cedula del empleado que requiere la solicitud. En la figura 5.18 se muestran los elementos básicos que conforman la interfaz de agregar Permisos.

- 1.- Encabezado de la página y logo de identificación de SIP.
- 2.- Enlace a la página principal de la aplicación.
- 3.- Área de Trabajo, en esta se ofrece el campo que el administrador debe introducir para buscar al empleado que requiere la solicitud de Reposo.



## CAPITULO V FASE DE CONSTRUCCIÓN

**Figura 5.17** Diseño de la Interfaz del Agregar Reposos

**Fuente:** Elaboracion Propia

El resultado de la aprobación de la solicitud de Permiso procesada correctamente se muestra en la figura 5.18 en donde aparecen algunos datos personales del usuario.

Datos del Usuarios	
Nombre Completo	Carmen_Maria
Apellido completo	Perez_Gonzalez
cedula	4000000
Fecha de Ingreso	07/10/2003
Division	Administracion
Area de Division	RRHH

Solicitud de Reposos	
Fecha de Inicio	07/12/2010
Fecha de Finalizacion	07/16/2010
Motivo	Curso de Especializacion
Clase de Solicitud	Reposos

**Figura 5.18** Diseño de la Interfaz del Agregar Reposos

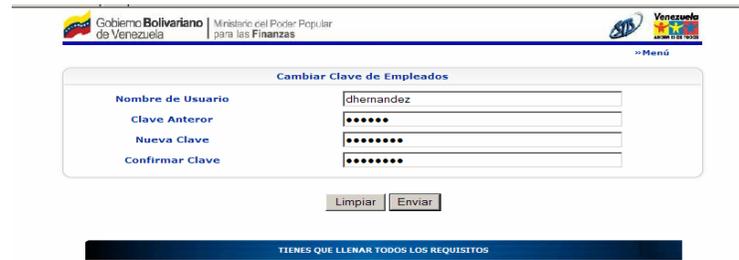
**Fuente:** Elaboracion Propia

### ▪ Interfaz de Cambiar Clave:

En esta interfaz el cualquier usuario de la aplicación podrá cambiar su clave de acceso al programa sin ninguna restricción y en cualquier momento. En la figura 5.19 se muestra la interfaz que se utiliza para cambiar la clave de acceso a la aplicación:



**CAPITULO V**  
**FASE DE CONSTRUCCIÓN**

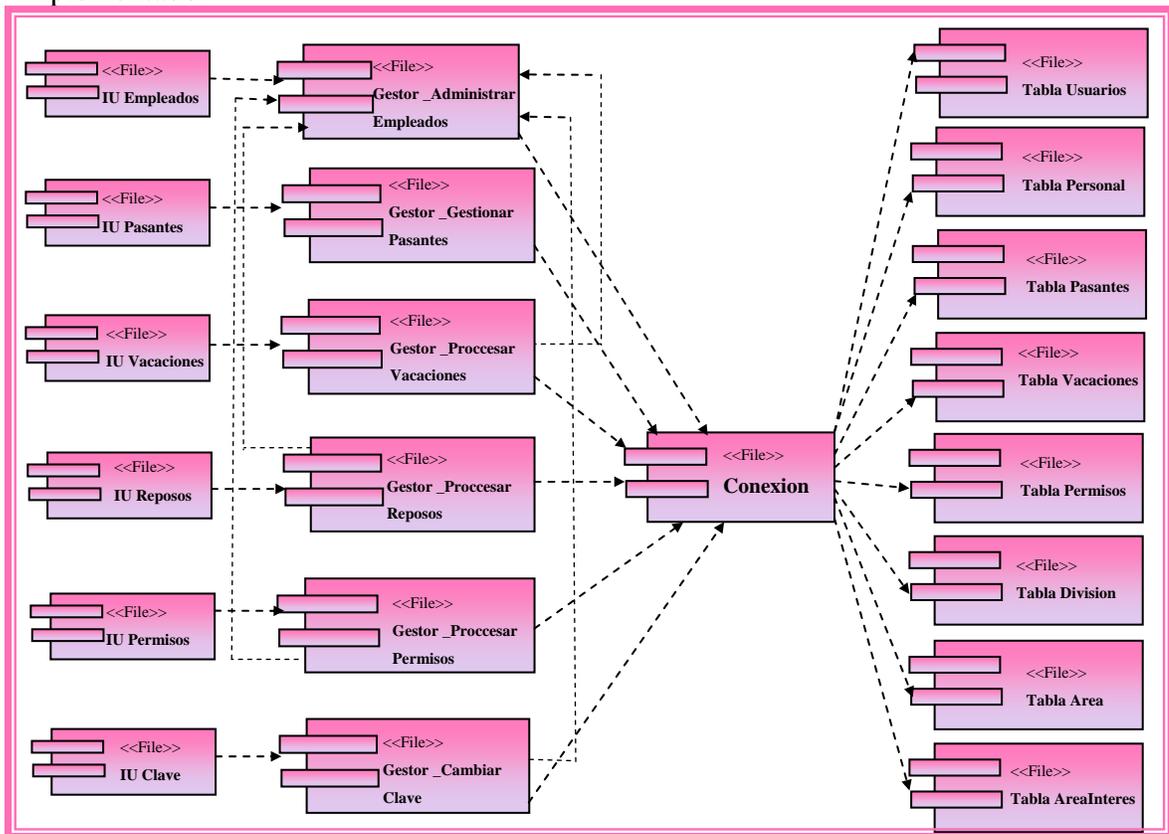


**Figura 5.19** Diseño de la Interfaz del Agregar Reposos

**Fuente:** Elaboracion Propia

**5.1.4 Diagrama de Componentes**

Un diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Se usa para modelar la estructura del software que será implementado.



**Figura 5.20** Diagrama de Componentes del Sistema SIP

**Fuente:** Elaboracion Propia



En este punto se desarrolla el diagrama de componentes total del sistema SIP, indicando las dependencias entre cada uno de los componentes como se puede observar en la figura 5.20.

#### 5.1.4.1 Implementación de los Componentes

Ahora veamos los diferentes códigos fuentes utilizados en el desarrollo del software. Las líneas que contienen el siguiente símbolo delante “//”, indican un comentario. El código muestra la implementación de funciones pertenecientes a los componentes del sistema. Se toman como muestra la interfaz principal del sistema, el menú administrador del mismo y el caso de uso Administrar Usuarios.

##### 5.1.4.1.1 Código Fuente de la Página Principal

```
<?
// Inicio de la sesión, esta utiliza código php
<?
// Inicio de la sesión, esta utiliza código php
if ( $_SESSION['texto1']!="" && $_SESSION['texto2']!="")
{
// Inicializa de la sesión.
    session_start();
    // Destruye todas las variables de la sesión
    session_unset();
    // Finalmente, destruye la sesión;
    session_destroy();
    header('Location: index.php');
}
?>
// Cuerpo del programa
```



```

<html>
<head>
<title>Sistema Integral de Personal Seniat</title>

<link href="images/pdvsastyle.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--
body {
    background-color: #FFFFFF;
}

.Estilo1 {font-size: 11px; font-weight: bold; text-transform: uppercase; text-
decoration: none; font-family: Verdana;}
-->
</style>
</head>

<body>
<form action="PaginadeSeguridad.php" method="post">
<table width="100%" height="70%" border="0" cellpadding="0" cellspacing="0">
<tr>
<td align="center" valign="middle"><table width="100%" border="0"
cellspacing="0" cellpadding="0">

<tr>
<td><div align="center">
</div>

</td

```



**CAPITULO V**  
**FASE DE CONSTRUCCIÓN**

---

```

</tr>
<tr>
  <td><table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td><div align="center">
        <table width="706" height="0%" border="0" cellpadding="0"
cellspacing="0">
          <tr>
            <td width="13" height="19%" align="left"
background="images/vc_esq_izq_sup.gif">&nbsp;
              </td>
            <td width="100%" align="center" valign="middle"
background="images/v_back_sup.jpg"><table width="100%" border="0"
cellspacing="0" cellpadding="0">
              <tr>
                <td height="18"><div align="center"><span class="ventana_tit">
</span>
                  <P class=link_azul_bold><span class="ventana_tit"><font
color="#0099FF">&raquo;</font></span><span class="Estilo1"><font
color="#0099FF">SISTEMA INTEGRAL DE PERSONAL
SENIAT</font></span><span class="ventana_tit"><font
color="#0099FF">&laquo;</font></span></P>
                    <span class="ventana_tit"></span></div>
                  </td>
                </tr>
              </table></td>
            </tr>
          </table></td>
        </tr>
      </td>
    </tr>
  </table></td>
  </tr>
</table></td>

```



**CAPITULO V**  
**FASE DE CONSTRUCCIÓN**

---

```

        <td width="13" height="19%" align="right"
background="images/vc_esq_der_sup.gif">
                </td>
        </tr>
        <tr>
                <td width="13" height="74%" background="images/v_back_izq.gif">
</td>
                <td><table width="100%" border="0" cellspacing="0"
cellpadding="0">
                        <tr>
                                <td align="center"><table width="50%" border="0"
cellspacing="0" cellpadding="4">
                                        <tr>
                                                <td colspan="2"></td>
                                        </tr>
                                        <tr>
                                                <td width="39%" align="right"><div align="right"
class="ventana_tit">USUARIO:</div></td>
                                                <td width="61%" align="left"><input type="text"
name="texto1">
                                                        <br>
                                                        </td>
                                                </tr>
                                        <tr>
                                                <td colspan="2"></td>
                                                </tr>
                                        <tr>
                                                <td colspan="2">
                                                        </td>
                                                </tr>
                                        </table>
                                </td>
                        </tr>
                </table>
        </tr>

```



```
<td align="right"><div align="right"
class="ventana_tit">Contrase&ntilde;a:</div></td>
<td align="left"><input type="password"
name="texto2"></td>
</tr>
<tr>
<td colspan="2"><div align="center" class="link_azul_bold">
<hr align="center" width="400" size="1" noshade>
<input type="submit" name="A1" value="INGRESAR">
<input type="reset" name="A2" value="BORRAR">
</div></td>
</tr>
</table>
<div align="center"></div></td>
</tr>
</table></td>
<td width="13" background="images/v_back_der.gif"></td>
</tr>
<tr>
<td width="13" height="7%" align="left"
background="images/v_esq_izq_inf.gif"> </td>
<td width="501" height="7%"
background="images/v_back_inf.gif"></td>
```



```

                <td          width="13"          height="7%"          align="right"
background="images/v_esq_der_inf.gif"> </td>
            </tr>
        </table>
    </td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</body>
</html>

```

#### 5.1.4.1.2 Código Fuente de la Pagina de Seguridad

Este código se utiliza para validar la entrada de los usuarios de la aplicación, ya sean administradores o consultores, es exclusivamente php.

**// Inicio de la sesión, esta utiliza código php**

```

<?php

    $Aceptar1=$_POST['texto1'];

    $Aceptar2=$_POST['texto2'];

```



```
trim($Aceptar1);
//echo $Aceptar2;
trim($Aceptar2);
//echo $Aceptar2;
if ($Aceptar1!="" && $Aceptar2!="")
    {
        $db=mysql_connect("localhost","root");

        if (!$db)
            {
                header("Location: ValidarPagina.php");
                exit();
            }

        mysql_select_db("sip",$db);

        $consulta = ("select * from usuarios where clave='$Aceptar2' &&
usuario='$Aceptar1'");
        $resultado = mysql_query($consulta);
        //echo $Aceptar2;
        echo mysql_error ( $db ) . "" ;
        $num_resultados = mysql_num_rows($resultado);
        if($num_resultados==0)
            header("Location: ValidarPagina1.php");
```



```
if($num_resultados>0)
{
    while($row = mysql_fetch_array($resultado))
    {
        $cedula = $row["cedula"];
        $aux=$row['administrar'];
    }

    $consulta = ("select * from personal where
cedula='$cedula'");

    $resultado = mysql_query($consulta);
    //echo $Aceptar2;
    echo mysql_error ( $db ) . "" ;
    $num_resultados = mysql_num_rows($resultado);

    if($num_resultados>0)
    {
        while($row = mysql_fetch_array($resultado))
        {
            $nombre=$row['nombre'];
            $apellido=$row['apellido'];

            session_start();
            $_SESSION['texto1']=$Aceptar1; //usuario
            $_SESSION['texto2']=$Aceptar2;// clave o
Password

            $_SESSION['A']=$aux;
            $_SESSION['aux1']=$nombre;
```



```
        $_SESSION['aux2']=$apellido;
    }

    if($aux=="si" ||$aux=="Si" || $aux=="SI" ||
$aux=="Si_Administra")
    {
        $_SESSION['A']="Si";
        ?><META                HTTP-EQUIV=Refresh
CONTENT=0;URL=Administrador.php?>
        <?PHP
            exit();
        }
        else
        {
            ?><META                HTTP-EQUIV=Refresh
CONTENT=0;URL=Usuarios.php?&usuario>
            <?PHP
                exit();
            }
        }
    }

    else
    //echo "nada";
        header("Location: ValidarPagina.php");

?>
```



#### 5.1.4.1.3 Código Fuente del Menú Principal

Este código se utiliza como menú principal de la aplicación y esta restringido para aquellas personas que no sean administradores del sistema.

**// Inicio de la sesión, esta utiliza código php**

```
<?PHP
session_start();
if ( $_SESSION['texto1']=="" && $_SESSION['texto2']=="" )
    header('Location: index.htm');

$usuario=$_SESSION['texto1'];
//echo $usuario;
$clave=$_SESSION['texto2'];
//echo $clave;
$nombre=$_SESSION['aux1'];
$apellido=$_SESSION['aux2'];
?>

<head>
<script          type="text/javascript"          language="JavaScript"
src="_pgtres/stm31.js"></script>
<title>Sistema Integral de Personal</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```



```

<style type="text/css">
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
</style>
<link href="images/pdvsastyle.css" rel="stylesheet" type="text/css">
<style type="text/css">
.style3 {color: #FF0000}
</style>

<script          type="text/javascript"          language="JavaScript"
src="_pgtres/stm31.js"></script>
<script language="javascript">

```

**// Las siguientes Funciones permiten utilizar un menú desplegable que se abre de derecha a izquierda y se activa con la colocación del puntero en la zona deseada, al abrir el menú se puede dirigir a un conjunto de paginas que conforman el menú principal del sistema//**

```

function FP_swapImg() { //v1.0
var doc=document,args=arguments,elm,n; doc.$imgSwaps=new Array(); for(n=2;
n<args.length;
n+=2) { elm=FP_getObjectByID(args[n]); if(elm) {
doc.$imgSwaps[doc.$imgSwaps.length]=elm;
elm.$src=elm.src; elm.src=args[n+1]; } }

```



```
}

```

```
function FP_preloadImgs() { //v1.0
  var d=document,a=arguments; if(!d.FP_imgs) d.FP_imgs=new Array();
  for(var i=0; i<a.length; i++) { d.FP_imgs[i]=new Image; d.FP_imgs[i].src=a[i]; }
}

function FP_getObjectByID(id,o) { //v1.0
  var c,el,els,f,m,n; if(!o)o=document; if(o.getElementById) el=o.getElementById(id);
  else if(o.layers) c=o.layers; else if(o.all) el=o.all[id]; if(el) return el;
  if(o.id==id || o.name==id) return o; if(o.childNodes) c=o.childNodes; if(c)
  for(n=0; n<c.length; n++) { el=FP_getObjectByID(id,c[n]); if(el) return el; }
  f=o.forms; if(f) for(n=0; n<f.length; n++) { els=f[n].elements;
  for(m=0; m<els.length; m++){ el=FP_getObjectByID(id,els[m]); if(el) return el; } }
  return null;
}

function FP_swapImgRestore() { //v1.0
  var doc=document,i; if(doc.$imgSwaps) { for(i=0;i<doc.$imgSwaps.length;i++) {
  var elm=doc.$imgSwaps[i]; if(elm) { elm.src=elm.$src; elm.$src=null; } }
  doc.$imgSwaps=null; }
}
</script>
</head>

```

```
// Inicio del Menú Usuario en el cual se detallan las paginas a las cuales se tiene
//acceso, entre ellas están agregar usuarios, consultar usuarios, modificar
usuarios y eliminar usuarios//

```



```
<script id="Sothink Widgets:_Marco_Legal.DynamicMenu.pgt"
type="text/javascript" language="javascript">

beginSTM("yfatchr","static","0","0","left","false","false","310","1000","0","250","",""
_pgtres/blank.gif");
beginSTMB("auto","0","0","vertically","_pgtres/blank.gif","0","0","0","0","#ccccff",
_pgtres/menulegal.gif","tiled","#000000","0","none","0","Normal","50","0","0","0",
"0","0","0","0","#000000","false","#000000","#000000","#000000","none");
appendSTMI("false","Usuarios","left","middle","","","-1","-
1","0","normal","#ffffff","#ffffff","","1","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","-1","-
1","0","","","_self","Verdana","8pt","#003399","bold","normal","none","Verdana","8
pt","#999999","bold","normal","none","0","solid","#d6d6ce","#d6d6ce","#d6d6ce",
"#d6d6ce","#d6d6ce","#d6d6ce","#d6d6ce","#d6d6ce","tiled","tiled");
beginSTMB("down",-236,"-
36","vertically","_pgtres/blank.gif","0","0","0","3","transparent","_pgtres/menulegal.
gif","tiled","#000000","0","solid","0","Normal","50","0","0","0","0","0","0","#7f
7f7f","false","#000000","#000000","#000000","none");
appendSTMI("false","&nbsp;Propiedades&nbsp;de&nbsp;Usuarios","center","middl
e","","","130","19","0","normal","transparent","transparent","","0","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","0","0","0","","","_self","Verdana","8pt",
"#003399","bold","normal","none","Verdana","8pt","#003399","bold","normal","none",
"0","none","#999900","#999900","#999900","#999900","#999900","#999900","#99
9900","#999900","","","tiled","tiled");
appendSTMI("false","&nbsp;&nbsp;","center","middle","","","130","19","0","normal",
"transparent","transparent","","0","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","0","0","0","","","_self","Arial","4pt",
"#000000","bold","normal","none","Arial","4pt","#000000","bold","normal","none","0",
```





```
#999900", "#999900", "#999900", "#999900", "#999900", "Eliminar_Usuarios.php", "", "  
", "tiled", "tiled");
```

```
endSTMB();
```

```
endSTMB();
```

```
endSTM();
```

**// Inicio del Menú Pasantes en el cual se detallan las paginas a las cuales se tiene acceso, entre ellas están agregar Pasantes, consultar Pasantes, modificar Pasantes y eliminar Pasantes//**

```
<script id="Sothink Widgets:_Marco_Legal.DynamicMenu.pgt"  
type="text/javascript" language="javascript">
```

```
beginSTM("yfatchr", "static", "0", "0", "left", "false", "false", "310", "1000", "0", "250", "", "  
_pgtres/blank.gif");
```

```
beginSTMB("auto", "0", "0", "vertically", "_pgtres/blank.gif", "0", "0", "0", "0", "#ccccff", "  
_pgtres/menulegal.gif", "tiled", "#000000", "0", "none", "0", "Normal", "50", "0", "0", "0", "  
"0", "0", "0", "0", "#000000", "false", "#000000", "#000000", "#000000", "none");
```

```
appendSTMI("false", "Pasantes", "left", "middle", "", "", "-1", "-
```

```
1", "0", "normal", "#ffffff", "#ffffff", "", "1", "-1", "-
```

```
1", "_pgtres/blank.gif", "_pgtres/blank.gif", "-1", "-
```

```
1", "0", "", "", "_self", "Verdana", "8pt", "#003399", "bold", "normal", "none", "Verdana", "8  
pt", "#999999", "bold", "normal", "none", "0", "solid", "#d6d6ce", "#d6d6ce", "#d6d6ce", "  
#d6d6ce", "#d6d6ce", "#d6d6ce", "#d6d6ce", "#d6d6ce", "", "", "", "tiled", "tiled");
```

```
beginSTMB("down", "-236", "-
```

```
36", "vertically", "_pgtres/blank.gif", "0", "0", "0", "3", "transparent", "_pgtres/menulegal.  
gif", "tiled", "#000000", "0", "solid", "0", "Normal", "50", "0", "0", "0", "0", "0", "0", "#7f  
7f7f", "false", "#000000", "#000000", "#000000", "none");
```



CAPITULO V  
FASE DE CONSTRUCCIÓN

---

```
appendSTMI("false","&nbsp;Propiedades&nbsp;de&nbsp;Pasantes","center","middle",
"","","130","19","0","normal","transparent","transparent","","0","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","0","0","0","","","_self","Verdana","8pt","#
003399","bold","normal","none","Verdana","8pt","#003399","bold","normal","none"
,"0","none","#999900","#999900","#999900","#999900","#999900","#999900","#99
9900","#999900","","","","tiled","tiled");
appendSTMI("false","&nbsp;&nbsp;","center","middle","","","130","19","0","normal",
"transparent","transparent","","0","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","0","0","0","","","_self","Arial","4pt","#00
0000","bold","normal","none","Arial","4pt","#000000","bold","normal","none","0","
none","#999900","#999900","#999900","#999900","#999900","#999900","#999900"
,"#999900","","","","tiled","tiled");
appendSTMI("false",">&nbsp;&nbsp;Consultar&nbsp;Pasantes<","left","middle","","",
"","","130","19","0","normal","transparent","transparent","","0","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","0","0","0","","../sip/Pasantes/Consultar_Pa
santes.php","_self","Verdana","8pt","#003399","bold","normal","none","Verdana","8
pt","#999999","bold","normal","none","0","none","#999900","#999900","#999900","#
999900","#999900","#999900","#999900","#999900","Consultar_Pasantes.php","","",
"","tiled","tiled");
appendSTMI("false",">&nbsp;Agregar&nbsp;Pasantes<","left","middle","","","130",
"19","0","normal","transparent","transparent","","0","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","0","0","0","","../sip/Pasantes/Agregar_Pas
antes.php","_self","Verdana","8pt","#003399","bold","normal","none","Verdana","8p
t","#999999","bold","normal","none","0","none","#999900","#999900","#999900","#
999900","#999900","#999900","#999900","#999900","Agregar_Pasantes.php","","",
"tiled","tiled");
appendSTMI("false",">&nbsp;Modificar&nbsp;Pasantes<","left","middle","","","130"
,"19","0","normal","transparent","transparent","","0","-1","-
1","_pgtres/blank.gif","_pgtres/blank.gif","0","0","0","","../sip/Pasantes/Modificar_P
```



```

asantes.php", "_self", "Verdana", "8pt", "#003399", "bold", "normal", "none", "Verdana", "
8pt", "#999999", "bold", "normal", "none", "0", "none", "#999900", "#999900", "#999900"
, "#999900", "#999900", "#999900", "#999900", "#999900", "Modificar_Pasantes.php", "
", "", "tiled", "tiled");
appendSTMI("false", ">>&nbsp;Eliminar&nbsp;Usuarios<<", "left", "middle", "", "", "130"
, "19", "0", "normal", "transparent", "transparent", "", "0", "-1", "-
1", "_pgtres/blank.gif", "_pgtres/blank.gif", "0", "0", "0", "", "../sip/Pasantes/Eliminar_Pas
antes.php", "_self", "Verdana", "8pt", "#003399", "bold", "normal", "none", "Verdana", "8p
t", "#999999", "bold", "normal", "none", "0", "none", "#999900", "#999900", "#999900", "#
999900", "#999900", "#999900", "#999900", "#999900", "Eliminar_Pasantes.php", "", "",
"tiled", "tiled");

endSTMB();
endSTMB();
endSTM();

```

**// Función que permite visualizar un calendario que toma la hora del sistema operativo y permite a su vez seleccionar una fecha específica//**

```

<script language="JavaScript" src="pupdate.js"></script>
<script language="JavaScript">
if (document.all) {
    document.writeln("<div id=\"PopUpCalendar\" style=\"position:absolute; left:0px;
top:0px; z-index:7; width:200px; height:77px; overflow: visible; visibility: hidden;
background-color: #FFFFFF; border: 1px none #000000\"
onMouseOver=\"if(ppcTI){ clearTimeout(ppcTI);ppcTI=false;}\"
onMouseOut=\"ppcTI=setTimeout('hideCalendar()',500)\">");

```



```

document.writeln("<div id=\"monthSelector\" style=\"position:absolute; left:0px;
top:0px; z-index:9; width:181px; height:27px; overflow: visible;
visibility:inherit\">");}
else if (document.layers) {
document.writeln("<layer id=\"PopUpCalendar\" pagex=\"0\" pagey=\"0\"
width=\"200\" height=\"200\" z-index=\"100\" visibility=\"hide\"
bgcolor=\"#FFFFFF\"
onMouseOver=\"if(ppcTI){clearTimeout(ppcTI);ppcTI=false;}\"
onMouseOut=\"ppcTI=setTimeout('hideCalendar()',500)\">");
document.writeln("<layer id=\"monthSelector\" left=\"0\" top=\"0\" width=\"181\"
height=\"27\" z-index=\"9\" visibility=\"inherit\">");}
else {
document.writeln("<p><font color=\"#FF0000\"><b>Error ! The current browser is
either too old or too modern (usind DOM document structure).</b></font></p>");}
</script>
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resized
if (init==true) with (navigator) {if
((appName=="Netscape")&&(parseInt(appVersion)==4)) {
document.MM_pgW=innerWidth; document.MM_pgH=innerHeight;
onresize=MM_reloadPage; }}
else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH)
location.reload();
}
MM_reloadPage(true);
//-->
</script>

```



**CAPITULO V**  
**FASE DE CONSTRUCCIÓN**

---

```
<noscript><p><font color="#FF0000"><b>JavaScript is not activated
!</b></font></p></noscript>
<table border="1" cellspacing="1" cellpadding="2" width="200"
bordercolorlight="#000000" bordercolordark="#000000" vspace="0"
hspace="0"><form name="ppcMonthList"><tr><td align="center"
bgcolor="#CCCCCC"><a href="javascript:moveMonth('Back')"
onMouseOver="window.status=' ';return true;"><font face="Arial, Helvetica, sans-
serif" size="2" color="#000000"><b></b></font></a><font face="MS Sans Serif,
sans-serif" size="1">
<select name="sItem" onMouseOut="if(ppcIE){ window.event.cancelBubble =
true;}" onChange="switchMonth(this.options[this.selectedIndex].value)" style="font-
family: 'MS Sans Serif', sans-serif; font-size: 9pt"><option value="0" selected>2000
• January</option><option value="1">2000 • February</option><option
value="2">2000 • March</option><option value="3">2000 • April</option><option
value="4">2000 • May</option><option value="5">2000 • June</option><option
value="6">2000 • July</option><option value="7">2000 • August</option><option
value="8">2000 • September</option><option value="9">2000 •
October</option><option value="10">2000 • November</option><option
value="11">2000 • December</option><option value="0">2001 •
January</option></select></font><a href="javascript:moveMonth('Forward')"
onMouseOver="window.status=' ';return true;"><font face="Arial, Helvetica, sans-
serif" size="2" color="#000000"><b> </b></font></a></td></tr></form></table>
<table border="1" cellspacing="1" cellpadding="2" bordercolorlight="#000000"
bordercolordark="#000000" width="200" vspace="0" hspace="0"><tr align="center"
bgcolor="#CCCCCC"><td width="20" bgcolor="#FFFFCC"><b><font face="MS
Sans Serif, sans-serif" size="1">Su</font></b></td><td width="20"><b><font
face="MS Sans Serif, sans-serif" size="1">Mo</font></b></td><td
width="20"><b><font face="MS Sans Serif, sans-serif"
size="1">Tu</font></b></td><td width="20"><b><font face="MS Sans Serif, sans-
```



**CAPITULO V**  
**FASE DE CONSTRUCCIÓN**

```

serif" size="1">We</font></b></td><td width="20"><b><font face="MS Sans Serif,
sans-serif" size="1">Th</font></b></td><td width="20"><b><font face="MS Sans
Serif,      sans-serif"      size="1">Fr</font></b></td><td      width="20"
bgcolor="#FFFFCC"><b><font      face="MS      Sans      Serif,      sans-serif"
size="1">Sa</font></b></td></tr></table>

```

```
<script language="JavaScript">
```

```
if (document.all) {
```

```
    document.writeln("</div>");
```

```
    document.writeln("<div id=\"monthDays\" style=\"position:absolute; left:0px;
top:52px; z-index:8; width:200px; height:17px; overflow: visible; visibility:inherit;
background-color: #FFFFFF; border: 1px none #000000\"> </div></div>");}
```

```
else if (document.layers) {
```

```
    document.writeln("</layer>");
```

```
    document.writeln("<layer id=\"monthDays\" left=\"0\" top=\"52\" width=\"200\"
height=\"17\" z-index=\"8\" bgcolor=\"#FFFFFF\" visibility=\"inherit\">
</layer></layer>");}
```

```
else { /*NOP*/ }
```

```
</script>
```

**//Estas funciones la utiliza el calendario que se describe anteriormente y sin estas el calendario lo funciona //**

```
var ppcDF = "m/d/Y";
```

```
var ppcMN = new
Array("January", "February", "March", "April", "May", "June", "July", "August", "Septem
ber", "October", "November", "December");
```

```
var ppcWN = new
Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"
);
```



```

var ppcER = new Array(4);
ppcER[0] = "Required DHTML functions are not supported in this browser.";
ppcER[1] = "Target form field is not assigned or not accessible.";
ppcER[2] = "Sorry, the chosen date is not acceptable. Please read instructions on the
page.";
ppcER[3] = "Unknown error ocured while executing this script.";
var ppcUC = false;
var ppcUX = 4;
var ppcUY = 4;

var ppcIE=(navigator.appName == "Microsoft Internet Explorer");
var ppcNN=((navigator.appName == "Netscape")&&(document.layers));
var ppcTT="<table width=\"200\" cellspacing=\"1\" cellpadding=\"2\" border=\"1\"
bordercolorlight=\"#000000\" bordercolordark=\"#000000\">\n";
var ppcCD=ppcTT;var ppcFT="<font face=\"MS Sans Serif, sans-serif\" size=\"1\"
color=\"#000000\">";var ppcFC=true;
var ppcTI=false;var ppcSV=null;var ppcRL=null;var ppcXC=null;var ppcYC=null;
var ppcML=new Array(31,28,31,30,31,30,31,31,30,31,30,31);
var
                                                                    ppcWE=new
Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"
);
var ppcNow=new Date();var ppcPtr=new Date();
if (ppcNN) {
    window.captureEvents(Event.RESIZE);
    window.onresize = restoreLayers;
    document.captureEvents(Event.MOUSEDOWN|Event.MOUSEUP);
    document.onmousedown = recordXY;
    document.onmouseup = confirmXY;}
function restoreLayers(e) {

```



```

if (ppcNN) {
  with (window.document) {
    open("text/html");
    write("<html><head><title>Restoring the layer structure...</title></head>");
    write("<body bgcolor=#FFFFFF onLoad=history.go(-1)>");
    write("</body></html>");
    close();}}
function recordXY(e) {
  if (ppcNN) {
    ppcXC = e.x;
    ppcYC = e.y;
    document.routeEvent(e);}
function confirmXY(e) {
  if (ppcNN) {
    ppcXC = (ppcXC == e.x) ? e.x : null;
    ppcYC = (ppcYC == e.y) ? e.y : null;
    document.routeEvent(e);}
function getCalendarFor(target,rules) {
  ppcSV = target;
  ppcRL = rules;
  if (ppcFC) {setCalendar();ppcFC = false;}
  if ((ppcSV != null)&&(ppcSV)) {
    if (ppcIE) {
      var obj = document.all['PopUpCalendar'];
      obj.style.left = document.body.scrollLeft+event.clientX;
      obj.style.top = document.body.scrollTop+event.clientY;
      obj.style.visibility = "visible";}
    else if (ppcNN) {
      var obj = document.layers['PopUpCalendar'];

```



```

obj.left = ppcXC
obj.top = ppcYC
obj.visibility = "show";}
else {showError(ppcER[0]);}
else {showError(ppcER[1]);}
function switchMonth(param) {
var tmp = param.split("|");
setCalendar(tmp[0],tmp[1]);}
function moveMonth(dir) {
var obj = null;
var limit = false;
var tmp,dptrYear,dptrMonth;
if (ppcIE) {obj = document.ppcMonthList.sItem;}
else if (ppcNN) {obj =
document.layers['PopUpCalendar'].document.layers['monthSelector'].document.ppcMonthList.sItem;}
else {showError(ppcER[0]);}
if (obj != null) {
if ((dir.toLowerCase() == "back")&&(obj.selectedIndex > 0)) {obj.selectedIndex--;}
else if ((dir.toLowerCase() == "forward")&&(obj.selectedIndex < 12))
{obj.selectedIndex++;}
else {limit = true;}}
if (!limit) {
tmp = obj.options[obj.selectedIndex].value.split("|");
dptrYear = tmp[0];
dptrMonth = tmp[1];
setCalendar(dptrYear,dptrMonth);}
else {
if (ppcIE) {

```



```
obj.style.backgroundColor = "#FF0000";
window.setTimeout("document.ppcMonthList.sItem.style.backgroundColor =
'FFFFFF'",50);}}
function selectDate(param) {
var arr = param.split("|");
var year = arr[0];
var month = arr[1];
var date = arr[2];
var ptr = parseInt(date);
ppcPtr.setDate(ptr);
if ((ppcSV != null)&&(ppcSV)) {
if (validDate(date)) {ppcSV.value = dateFormat(year,month,date);hideCalendar();}
else {showError(ppcER[2]);if (ppcTI) {clearTimeout(ppcTI);ppcTI = false;}} }
else {
showError(ppcER[1]);
hideCalendar();} }
function setCalendar(year,month) {
if (year == null) {year = getFullYear(ppcNow);}
if (month == null) {month = ppcNow.getMonth();setSelectList(year,month);}
if (month == 1) {ppcML[1] = (isLeap(year)) ? 29 : 28;}
ppcPtr.setYear(year);
ppcPtr.setMonth(month);
ppcPtr.setDate(1);
updateContent();}
function updateContent() {
generateContent();
if (ppcIE) {document.all['monthDays'].innerHTML = ppcCD;}
else if (ppcNN) {
```



```
with (document.layers['PopUpCalendar'].document.layers['monthDays'].document)
{
  open("text/html");
  write("<html>\n<head>\n<title>DynDoc</title>\n</head>\n<body
bgcolor=\"#FFFFFF\">\n");
  write(ppcCD);
  write("</body>\n</html>");
  close();}
else {showError(ppcER[0]);}
ppcCD = ppcTT;}
function generateContent() {
  var year = getFullYear(ppcPtr);
  var month = ppcPtr.getMonth();
  var date = 1;
  var day = ppcPtr.getDay();
  var len = ppcML[month];
  var bgr,cnt,tmp = "";
  var j,i = 0;
  for (j = 0; j < 7; ++j) {
    if (date > len) {break;}
    for (i = 0; i < 7; ++i) {
      bgr = ((i == 0)||i == 6) ? "#FFFCC" : "#FFFFFF";
      if (((j == 0)&&(i < day))||(date > len)) {tmp += makeCell(bgr,year,month,0);}
      else {tmp += makeCell(bgr,year,month,date);++date;}
    }
    ppcCD += "<tr align=\"center\">\n" + tmp + "</tr>\n";tmp = "";
  }
  ppcCD += "</table>\n";
}
function makeCell(bgr,year,month,date) {
  var param = "\""+year+"|"+month+"|"+date+"\"";
  var td1 = "<td width=\"20\" bgcolor=\""+bgr+"\" ";
```



```
var td2 = (ppcIE) ? "</font></span></td>\n" : "</font></a></td>\n";
var evt = "onMouseOver=\"this.style.backgroundColor=#FF0000\"
onMouseOut=\"this.style.backgroundColor=\"+bgr+\"\"
onMouseUp=\"selectDate(\"+param+)\";
var ext = "<span Style=\"cursor: hand\">";
var lck = "<span Style=\"cursor: default\">";
var lnk = "<a href=\"javascript:selectDate(\"+param+)\";
onMouseOver=\"window.status='';return true;\">";
var cellValue = (date != 0) ? date+" " : "&nbsp;";
if ((ppcNow.getDate() == date)&&(ppcNow.getMonth() ==
month)&&(getFullYear(ppcNow) == year)) {
    cellValue = "<b>"+cellValue+"</b>";
}
var cellCode = "";
if (date == 0) {
    if (ppcIE) { cellCode = td1+"Style=\"cursor:
default\">"+lck+ppcFT+cellValue+td2;}
    else { cellCode = td1+">"+ppcFT+cellValue+td2;}}
else {
    if (ppcIE) { cellCode = td1+evt+"Style=\"cursor:
hand\">"+ext+ppcFT+cellValue+td2;}
    else {
        if (date < 10) { cellValue = "&nbsp;"+ cellValue + "&nbsp;";}
        cellCode = td1+">"+lnk+ppcFT+cellValue+td2;}}
return cellCode;}
function setSelectList(year,month) {
var i = 0;
var obj = null;
if (ppcIE) {obj = document.ppcMonthList.sItem;}
```



```

else          if          (ppcNN)          {obj          =
document.layers['PopUpCalendar'].document.layers['monthSelector'].document.ppcM
onthList.sItem;}
else {/* NOP */}
while (i < 13) {
    obj.options[i].value = year + "|" + month;
    obj.options[i].text = year + " • " + ppcMN[month];
    i++;
    month++;
    if (month == 12) {year++;month = 0;}}
function hideCalendar() {
    if (ppcIE) { document.all['PopUpCalendar'].style.visibility = "hidden";}
    else    if    (ppcNN)    { document.layers['PopUpCalendar'].visibility    =
"hide";window.status = " ";}
    else {/* NOP */}
    ppcTI = false;
    setCalendar();
    ppcSV = null;
    if (ppcIE) { var obj = document.ppcMonthList.sItem;}
    else          if          (ppcNN)          { var          obj          =
document.layers['PopUpCalendar'].document.layers['monthSelector'].document.ppcM
onthList.sItem;}
    else {/* NOP */}
    obj.selectedIndex = 0;}
function showError(message) {
    window.alert("[ PopUp Calendar ]\n\n" + message);}
function isLeap(year) {
    if ((year%400==0)||((year%4==0)&&(year%100!=0))) {return true;}
    else {return false;}}

```



```

function getFullYear(obj) {
  if (ppcNN) {return obj.getYear() + 1900;}
  else {return obj.getYear();} }
function validDate(date) {
  var reply = true;
  if (ppcRL == null) { /* NOP */ }
  else {
    var arr = ppcRL.split(":");
    var mode = arr[0];
    var arg = arr[1];
    var key = arr[2].charAt(0).toLowerCase();
    if (key != "d") {
      var day = ppcPtr.getDay();
      var orn = isEvenOrOdd(date);
      reply = (mode == "[^]") ? !((day == arg)&&((orn == key)||((key == "a")))) : ((day ==
arg)&&((orn == key)||((key == "a")))); }
    else {reply = (mode == "[^]") ? (date != arg) : (date == arg);} }
  return reply;}
function isEvenOrOdd(date) {
  if (date - 21 > 0) {return "e";}
  else if (date - 14 > 0) {return "o";}
  else if (date - 7 > 0) {return "e";}
  else {return "o";} }
function dateFormat(year,month,date) {
  if (ppcDF == null) {ppcDF = "m/d/Y";}
  var day = ppcPtr.getDay();
  var crt = "";
  var str = "";
  var chars = ppcDF.length;

```



```

for (var i = 0; i < chars; ++i) {
  crt = ppcDF.charAt(i);
  switch (crt) {
    case "M": str += ppcMN[month]; break;
    case "m": str += (month<9) ? ("0"+(++month)) : ++month; break;
    case "Y": str += year; break;
    case "y": str += year.substring(2); break;
    case "d": str += ((ppcDF.indexOf("m")!=-1)&&(date<10)) ? ("0"+date) : date;
  break;
    case "W": str += ppcWN[day]; break;
    default: str += crt;}}
return unescape(str);}

```

### 5.5 Flujo de Trabajo: Prueba

Las pruebas aplicadas al sistema representan una revisión final de las especificaciones, del diseño y de la codificación, con la finalidad de descubrir los posibles errores que se puedan presentar. Existen diferentes tipos y todas con los mismos objetivos de tratar de descubrir errores. Las pruebas se aplican principalmente sobre los componentes ejecutables del modelo de implementación. En esta fase se realizaran las pruebas de unidad y las pruebas de integración.

Las pruebas se realizan para verificar los resultados de la implementación probando cada construcción. Las pruebas que se realizan son pruebas de caja negra y pruebas de integración. Las pruebas de caja negra verifican el funcionamiento de los métodos que se llevan a cabo a medida que se ejecuta la aplicación. Las pruebas de integración son para corroborar que los distintos componentes del sistema tienen una buena comunicación y los mensajes que se envían unos a otros se procesan correctamente y de esta forma verificar el funcionamiento integral del software.



### 5.5.1 Pruebas por unidad

Las pruebas por unidad evalúan los componentes establecidos como unidades individuales, verificando la interacción interna entre los ellos. Estas pruebas se aplicarán a través de las pruebas de “caja negra”, a los componentes que integran el subsistema.

Para ello se identifican un conjunto de valores que pueden introducirse por un actor y se expresaron como clases para poder abarcar la totalidad de las ocurrencias de un evento de inserción de datos.

El caso de uso administrar empleados proporciona entradas que deben ser ingresadas por el usuario a las cuales se le aplicarán las pruebas de “caja negra”, para el cual se han identificado las clases de equivalencia que muestra la tabla

**Tabla 5.2** Clases de equivalencia para las pruebas de caja negra de registrar usuarios

Nro	Dato	Clase de Equivalencia	Valido	No Valido
1	cedula	Carácter numérico	x	
2	cedula	Carácter alfabético		x
3	cedula	$7 \leq \text{Longitud de caracteres} \leq 8$	x	
4	cedula	Longitud de caracteres $> 8$		x
5	cedula	Dato nulo		x
6	Nombre Completo	Carácter numérico		x
7	Nombre Completo	Carácter alfabético	x	
8	Nombre Completo	Longitud de caracteres = 1		x
9	Nombre Completo	Longitud de caracteres $> 1$	x	
10	Nombre Completo	Dato nulo		x

Fuente: Elaboración Propia



A partir de las clases de equivalencia expuestas anteriormente, se elaboran los casos de prueba que se presentan en la tabla 5.8:

**Tabla 5.3** Caso de prueba de caja negra para registrar usuarios

Nro	Dato	Caso de Prueba	Salida	Clases Cubiertas
1	cedula	'14580059'	Valida	1,5
2	cedula	'Maria'	No valida	2,6
3	cedula	'15036320'	Valida	1,5
4	cedula	'145098765'	No valida	6
5	cedula	'NULL'	No valida	7
6	Nombre Completo	'12345'	No valida	8,11
7	Nombre Completo	'Daira'	Valida	10
8	Nombre Completo	'L'	No valida	11
9	Nombre Completo	'LI'	Valida	12
10	Nombre Completo	'NULL'	No valida	14

**Fuente:** Elaboracion Propia

Estos casos de pruebas fueron aplicadas a todos los casos de uso que involucran entrada de datos.

### 5.5.2 Pruebas de integración:

El propósito de las pruebas de integración es verificar que los componentes que están siendo probados puedan trabajar conjuntamente.

Las pruebas de integración fueron aplicadas a SIP de manera incremental, incorporando paso a paso componentes previamente probados en unidad, relacionados según la función que desempeñan, con lo que se logró la detección y



corrección de errores, permitiendo ajustar al sistema a los requisitos especificados para obtener la respuesta correspondiente a cada caso.

A continuación se describirán las pruebas de integración para ingresar al sistema de personal.

### **5.5.2.1 Pruebas de integración al sistema SIP**

**Entrada:** login y clave del usuario

**Resultado:** El usuario es autenticado e ingresa al sistema

**Condiciones:** El usuario debe estar registrado en la base de datos del sistema.

**Procedimiento de prueba:** El usuario debe ingresar el login y la clave en los espacios correspondientes, luego presionar el botón aceptar, el sistema realiza la verificación de los datos, si el usuario se encuentra registrado y activo, el sistema muestra la interfaz correspondiente al privilegio que tenga sobre el mismo, si ocurre un error, se muestra un mensaje de acceso denegado y se remite a la pagina inicial para que éste pueda realizar nuevos intentos.

En la figura 5.12 se observa gráficamente la respuesta producida por el sistema, cuando un usuario (en este caso con cargo hemoterapista) accede exitosamente al sistema y en la figura 5.13 muestra cuando un usuario no se encuentra registrado o no esta activo en la base de datos.



## CAPITULO V FASE DE CONSTRUCCIÓN

» SISTEMA INTEGRAL DE PERSONAL SENIAT «

USUARIO:

CONTRASEÑA:

---

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para Economía y Finanzas

Venezuela AHORA ES DE TODOS

» Atras

» BIENVENIDO AL SISTEMA ADMINISTRADOR Daira\_del\_Carmen «

**Mision**

Administrar eficientemente los procesos aduaneros y tributariops en el ambito nacional y otras competencias legalmente asignadas, mediante la ejecucion de politicas publicas en procura de aportar la mayor suma de felicidad posible y seguridad social a la nacion venezolana.

**Vision**

Ser una institucion modelo, moderna, inteligente, acorde con el desarrollo socio-economico del pais, que fomente la cultura y garantice el cumplimiento de las obligaciones y deberes aduaneros y tributarios, contribuyendo a consolidar el proyecto socialista bolivariano.

» MENÚ PRINCIPAL «

- » Usuarios
- » Seniat en Línea
- » Pasantes
- » Vacaciones
- » Reposos
- » Permisos
- » Cambiar Clave de Usuario
- » Titulares de Noticias
- » Impuesto Sobre la Renta

SISTEMA INTEGRAL DE PERSONAL REALIZADO POR DAIRA HERNANDEZ UDO ANZOATEGUI

**Figura 5.21.** Respuesta a acceso exitoso para usuario con privilegio Administrador

**Fuente:** Elaboración propia.

» ERROR AL TRATAR DE CONECTARSE CON LA BASE DE DATOS «

»COMPRUEBE SU NOMBRE DE USUARIO Y SU CONTRASEÑA, PRESIONE AQUI PARA INTERTARLO NUEVAMENTE«

**Figura 5.22.** Respuesta a acceso exitoso denegado al sistema

**Fuente:** Elaboración propia.



### 5.5.2.2 Pruebas de integración al Módulo Administrar Empleados

**Entrada:** Datos personales y administrativos de un empleado.

**Resultado:** Registro de un nuevo empleado.

**Condiciones:** El empleado no debe estar en la base de datos.

**Procedimiento de prueba:**

**Caso 1:**

1.- El Asistente Administrativo presiona la opción Usuarios del menú de la página principal (Administrador) y luego de escoger la opción Ingresar Usuario del menú desplegable, se muestra el siguiente formulario:

The screenshot shows a web interface for adding a user. At the top, there are logos for the Government of Venezuela and the Ministry of Popular Power for Economy and Finance, along with the SIP Venezuela logo. Below the logos is a navigation menu with a '»Menú' link. The main form is titled 'Datos Personales de los Usuarios' and contains the following fields:

- Nombre Completo
- Apellido completo
- cedula
- Direccion
- Telefono
- Fecha de Nacimiento
- Nacionalidad
- Sexo
- Edo civil
- Correo Electronico
- Fecha de Ingreso
- Profesion
- Cargo
- Division
- Area de Division
- Nombre de Usuario
- Privilegio de administracion
- Password

At the bottom of the form are two buttons: 'Enviar' and 'Borrar'.

**Figura 5.23** Interfaz Agregar Usuario

**Fuente:** Elaboración propia.



## CAPITULO V

### FASE DE CONSTRUCCIÓN

2.- El Asistente Administrativo llena todos los campos y presiona el botón enviar, como se muestra en la figura.

Datos Personales de los Usuarios	
Nombre Completo	Carlos
Apellido completo	Martinez
cedula	13400000
Direccion	Pueblo Nuevo,PLC
Telefono	0234555333
Fecha de Nacimiento	08/30/1987
Nacionalidad	Venezolano
Sexo	Masculino
Edo civil	Divorciado(a)
Correo Electronico	CarlosMartinez@hotmail.com
Fecha de Ingreso	02/04/2008
Profesion	Abogado
Cargo	Fiscal
Division	Fiscalizacion
Area de Division	tributos
Nombre de Usuario	cmartinez
Privilegio de administracion	no
Password	*****

Enviar Borrar

INTRODUCE LOS DATOS DEL EMPLEADO Y PRESIONE ACEPTAR

Figura 5.24. Detalles Inerfaz Agregar Usuario

Fuente: Elaboración propia.

2.- Luego se mostrara una interfaz donde se indica que se ingreso correctamente el usuario en la base de datos, como se indica en la figura .

El usuario se Inserto Correctamente en la Base de Datos

Nombre Completo	Carlos
Apellido Completo	Martinez
Cedula de Identidad	13400000
Dirección	Pueblo_Nuevo,PLC
Telefono	0234555333
Fecha de Nacimiento	08/30/1987
Nacionalidad	Venezolano
Sexo	Masculino
Estado Civil	Divorciado(a)
Correo Electronico	CarlosMartinez@hotmail.com
Fecha Ingreso	02/04/2008
Profesión	Abogado
Cargo	Fiscal
Division	Fiscalizacion
Area	tributos

Aceptar

Figura 5.25. Inetrfaz Agregar Usuario Registrado

Fuente: Elaboración propia.



## CAPITULO V

### FASE DE CONSTRUCCIÓN

#### Caso 2:

Luego de que el usuario realiza el paso 1 e intenta registrar la transfusión dejando uno o ambos campos vacíos, el sistema se quedara en la misma interfaz (figura )hasta que haya completado todos los campos.

Datos Personales de los Usuarios	
Nombre Completo	Alejandra
Apellido completo	Oscario
cedula	15320000
Direccion	Pueblo Nuevo. # 50 .PLC
Telefono	0416578900
Fecha de Nacimiento	08/30/1981
Nacionalidad	Venezolano
Sexo	Femenino
Edo civil	Soltero(a)
Correo Electronico	
Fecha de Ingreso	06/14/2010
Profesion	Ing en Computacion
Cargo	Asistente Informatica
Division	Sumario Administrativo
Area de Division	Informatica
Nombre de Usuario	aoscaño
Privilegio de administracion	si
Password	••••

**Figura 5.26.** Interfaz Agregar Usuario con campos vacios

**Fuente:** Elaboración propia.

#### Caso 3:

Luego de que el usuario realiza el paso 1 e intenta registrar la transfusión y por equivocación ingresa un empleado que ya se encuentre en la base de datos, el sistema mostrara una inetrfaz, donde arrojara un mensaje indicando el usuario se encuentra registrado, asi como s muestra en la figura.



**Figura 5.27.** Mensaje de Usuario Registrado en la Bse de Dtos.

**Fuente:** Elaboración propia.

#### **5.4 Evaluacion de la Fase de Construccion.**

Durante la implementación de esta fase, se realizó la codificación efectiva de las páginas del sistema, la base de datos SIP y los diferentes componentes que conforman el software, se aplicaron pruebas tanto de integridad como por unidad al software las cuales permitieron mejorar la calidad de la aplicación, ya que al corregir los errores presentados queda garantizada la futura estabilidad operativa del mismo

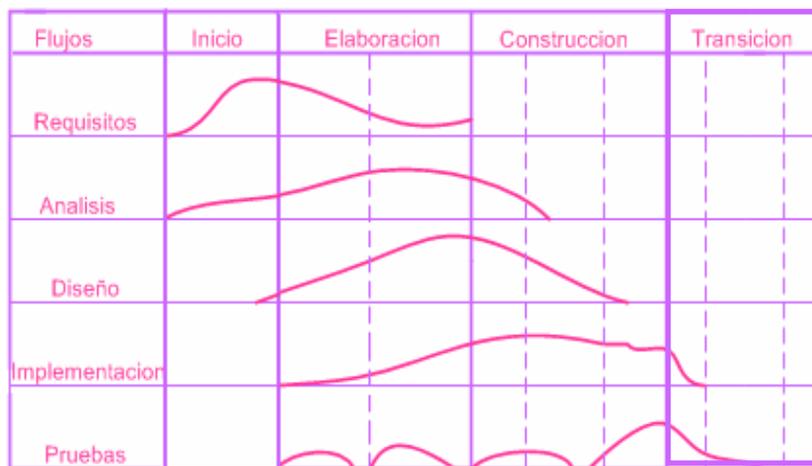
# CAPITULO VI

## FASE DE TRANSICION

### PLANIFICACION DE LA FASE DE TRANSICION

Esta ultima fase esta dirigida a que el software este disponible para los usuarios finales. Su objetivo principal se basa en la realizaci3n de parte de la implementaci3n y las pruebas beta.

Estas pruebas permiten validar el sistema corrigiendo peque1os de errores que permitan mejorar el funcionamiento, rendimiento y usabilidad de la aplicaci3n.



**Figura 6.1** Flujos de trabajo requeridos en la Fase de Transici3n

**Fuente:** Elaboraci3n Propia

En el desarrollo de esta fase, se debe instalar la aplicaci3n, entrenar a los usuarios finales y ajustar la aplicaci3n seg1n la validaci3n del usuario.



### **6.1 Preparacion de la Version Beta**

Toda la documentación necesaria para el buen uso del sistema, es desarrollada de una manera clara y ordenada, especificando cada uno de los procedimientos que debe seguir el usuario a la hora de utilizar el sistema con el propósito de proporcionar una línea de ayuda, asistencia, y corrección de los defectos que se encuentran tras la entrega.

### **6.2 Instalacion de la Version Beta**

Para la puesta en práctica del sistema de personal SIP se requiere la instalación de:

- **Sistema Manejador de Base de Datos:** se instalan los paquetes del SMBD MySQL 4.0 .
- **Servidor de Paginas Web:** se instala el servidor de paginas Web http Apache 2.0 con las opciones predeterminadas.
- **Lenguaje de Programación de Aplicaciones Web:** Se instala el software de PHP 5.0, seleccionando los módulos adicionales, php-gd, php-snmp, php-curl, php-apache2, php-mysql4, php-perlbase, y los seleccionados por defecto.

Los paquetes anteriores constituyen el sistema base sobre el cual se debe instalar el software. Los usuarios realizaran las revisiones programadas, las cuales tienen como objetivo verificar la marcha real del sistema y en el caso de desvíos ó errores, tomar las acciones correctivas correspondientes para el mejoramiento del producto final, lo cual determina la calidad y efectividad funcional del sistema



### **6.3 Evaluacion de la Fase de Transicion**

Se ha cubierto el ciclo completo de desarrollo (requisitos, análisis, diseño, codificación y pruebas), dando como resultado final una versión ejecutable del software, con una funcionalidad limitada

Se corrigieron los problemas y se realizo la entrega de un manual de usuario para el personal del área de personal del la división de administración.

Con esta fase se da por terminado el proyecto SIP, ya que se logra obtener un software con capacidad operativa confiable.



## CONCLUSIONES

- El proceso de seguimiento y soporte de las actividades asociadas al área de personal, perteneciente a la división de administración de la Gerencia General de Tributos Internos de la Región Nor\_Oriental, ocasiona retrasos a la hora de resolver las solicitudes de vacaciones, permiso, o reposos de los empleados, sobre tiempo y vulnerabilidad de la información manejada; todo esto atribuido al método manual que utilizan para desarrollarlo y a la falta de organización en la elaboración y manipulación de los documentos técnicos involucrados.
  
- Para la especificación de los requerimientos observamos los diferentes focos problemáticos que presentaba el área de personal de la Gerencia General de Tributos Internos de la Región Nor\_Oriental, relacionados a las solicitudes administrativas, se evaluó la documentación existente, la estructura organizativa, se realizaron las entrevistas a las personas que laboran en él, obteniéndose los requisitos que luego de ser analizados proporcionaron los modelos para el diseño de datos, arquitecturas e interfaces.
  
- El uso de la metodología del Proceso Unificado de Desarrollo de Software a través de sus varias fases de desarrollo: Inicio, Elaboración, Construcción, Transición, generando en cada una de ellas los componentes necesarios para el ciclo de vida óptimo del sistema. permitió cubrir con éxito todos los pasos para la realización del software.
  
- El lenguaje Unificado de Modelado (UML) nos proporciona diagramas con los que se obtuvo diferentes vistas del sistema. Como son: el diagrama de casos de uso que nos permitió plasmar la interacción entre los usuarios y el sistema, el diagrama de clases y colaboración nos permitió definir las abstracciones iniciales para la



## CONCLUSIONES

---

aplicación y con el diagrama de secuencia pudimos visualizar el tipo de relación existente entre los objetos y la cronología de los envíos de mensaje.

- La Interfaz de usuario fue diseñada para evitar errores en el llenado de los distintos documentos que permiten sustentar las actividades asociadas al área de personal.
- La base de datos se diseñó a través del modelo relacional para evitar la redundancia de datos y garantizar la integridad de los mismos, y se seleccionó el sistema manejador de Bases de datos MySQL versión 5.1 ya que se pueden realizar muchas consultas, además de soportar varios motores de almacenamiento y es licenciado bajo la GPL de GNU mas utilizado en software libre.
- Se utilizó PHP como lenguaje de programación, para la crearon páginas dinámicas, ya que funciona de forma más rápida y eficiente con HTML ofreciendo mayor tiempo de respuesta y seguridad para los usuarios.
- El Sistema desarrollado proporciona un ambiente seguro para el acceso a la información que maneja, esto a través de los módulos que lo conforman, provee de la asignación a los roles de usuarios y su respectiva autenticación cada vez que entren al sistema.
- Con la realización de las pruebas se validó la calidad del software, representando una revisión final de las especificaciones del diseño, de la codificación y la integridad de los datos. Se plantearon soluciones a los posibles problemas a través de estrategias de mitigación y corrección de errores, manejando así los riesgos en el proceso de desarrollo del sistema.



## CONCLUSIONES

---

- La primera versión del sistema fue evaluada por los usuarios finales y las sugerencias fueron ejecutadas para lograr a cabalidad el buen funcionamiento del sistema.
- Se elaboró un manual de usuario que brinda soporte a los usuarios para uso del sistema.
- Las consultas a los diferentes módulos brindan información resumida y detallada de todas las operaciones que se realizan y son un instrumento de suma importancia para la consolidación de la información del área de personal de la Gerencia General de Tributos Internos de la Región Nor\_Oriental.



## RECOMENDACIONES

- Instaurar la aplicación Web propuesta siguiendo los métodos y los requerimientos especificados en el proyecto.
- Se recomienda la implementación de casos de uso que no habían sido considerados durante el proyecto, como por ejemplo, Generar Reportes de solicitudes.
- Desarrollar un módulo de impresión de las solicitudes que permita al administrador del sistema tener un respaldo de toda la información del sistema.
- Establecer o definir claramente las responsabilidades y deberes de los usuarios del sistema a través de los niveles establecidos.
- Realizar mantenimiento periódico al sistema de manera de actualizar la información de las bases de datos y realizar respaldos de estas para prevenir su pérdida
- Finalmente, se recomienda el adecuado entrenamiento a los usuarios del sistema.
- Es importante hacer un seguimiento continuo del sistema a fin de introducir las mejoras que se requieran



## REFERENCIAS BIBLIOGRAFICAS

GONZALEZ M y Porras, K. **“Desarrollo de un software para la automatización de las actividades relacionadas con la nomina de empleados de una alcaldía, adscrita a la gobernación del estado Anzoátegui.** Trabajo de grado. Universidad de Oriente, Edo de Anzoátegui (2006).

ARAYA, J y Velásquez, A, **“Desarrollo de un sistema de información para el control de nómina en el departamento de recursos humanos de la alcaldía del municipio Arismendi estado Nueva Esparta”.** Trabajo de Grado. Universidad de Oriente, Edo Anzoátegui (2006).

ARCILA A y Zacarías, E. **“Desarrollo de un software como soporte para la automatización de las actividades administrativas que se llevan a cabo en una institución educativa”.** Trabajo de Grado. Universidad de Oriente, Edo Anzoátegui. (2006).

Núñez, K. **“Desarrollo de un software para manejar los servicios académicos del Departamento de Mecánica”.** Trabajo de Grado. Universidad de Oriente, Edo Anzoátegui, Edo Anzoátegui(2005).

SIMONI M. **“Desarrollo de un Sistema de Información para la Automatización de los procesos Realizados en el Departamento de Ciencias de la Universidad de Oriente, Núcleo de Anzoátegui”.** Trabajo de Grado. Universidad de Oriente, Edo. Anzoátegui (2.004).

JACOBSON, I. y BOOCH, G. **“El proceso Unificado de Desarrollo de Software”.** Pearson Educación, S.A. Madrid (2000).



## REFERENCIAS BIBLIOGRÁFICAS

---

ZAVALA, R. **“Ingeniería de Software”**, Disponible en: [www.angelfire.com/scifi/jzavalap/apuntes/IngSoftware.html](http://www.angelfire.com/scifi/jzavalap/apuntes/IngSoftware.html) (2000)

REY, A. LEAL, M y GONZÁLEZ C. **“Qué es UML?”**. Disponible en: [www.creangel.com/uml/home.html](http://www.creangel.com/uml/home.html)(2.000)

AUTOR DESCONOCIDO **“Ingeniería de Software”** volumen 1. IBM Learning Services. Venezuela (2005)

NÚÑEZ, L. **“Programación Orientada a Objeto”**.Universidad Tecnológica de Santiago. Disponible en:

<http://www.monografias.com/trabajos4/basesdatos/basesdatos.shtml>  
(1.997)

ALVAREZ, M. **¿Qué es la Programación Orientada a Objetos?** Disponible en: <http://www.desarrolloweb.com/articulos/499.php> (2006)

WALES, J. y SANGER L., Enciclopedia Libre, **“PHP”**. Disponible en: <http://es.wikipedia.org/wiki/PHP> (2006)

AUTOR DESCONOCIDO **“Lenguaje de Marcado de Hipertexto:HTML”** volumen 1. IBM Learning Services. Venezuela (2005)

AUTOR DESCONOCIDO **“PHP Hypertext Preprocessor”** volumen 1. IBM Learning Services. Venezuela (2006).

DATE C., **“Introducción a los sistemas de Bases De Datos”**. Vol 1. Editorial Addison Wesley (1.990)



## REFERENCIAS BIBLIOGRÁFICAS

---

SOMMERVILLE I., **“Ingeniería de Software”**. Segunda Edición. Editorial Addison-Wesley Iberoamericana. Buenos Aires, (1988).

WALES, J. y SANGER L., Enciclopedia Libre, **“Bases de Datos”**

MORENO G., **“Ingeniería de Software UML”**, Disponible en: <http://www.monografias.com/trabajos5/insof/insof.shtml>, (2005).

## METADATOS PARA TRABAJOS DE GRADO, TESIS Y

### ASCENSO:

TÍTULO	DESARROLLO DE UNA APLICACIÓN PARA LA AUTOMATIZACION DE ALGUNAS ACTIVIDADES LLEVADAS A CABO EN EL AREA DE PERSONAL, PERTENECIENTE A LA DIVISION DE ADMINISTRACION DE LA GERENCIA GENERAL DE TRIBUTOS INTERNOS DE LA REGION NOR-ORIENTAL
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
HERNÁNDEZ C., DAIRA DEL C.	CVLAC: 14.580.059 E MAIL: dhernandez281@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

DESARROLLO, SISTEMA, SOFTWARE, AUTOMATIZACIÓN, DIVISION, AREA, VACACIONES, PERMISOS, EMPLEADOS, PASANTES.

## **METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**

<b>ÁREA</b>	<b>SUBÁREA</b>
INGENIERÍA Y CIENCIAS APLICADAS	INGENIERÍA EN COMPUTACIÓN

### RESUMEN (ABSTRACT):

El SENIAT es el Servicio Nacional Integrado de Administración Tributaria, este un organismo del estado central venezolano que se encarga de recolectar los impuestos. Dentro de su departamento de administración se encuentra el área de personal, el cual cuenta con un proceso manual, para llevar a cabo el seguimiento de ciertas actividades que allí se desarrollan, como son las asociadas a el registro de las solicitudes de vacaciones, permisos, reposos entre otras, ocasionado una serie de inconvenientes que originan la ineficiencia de las actividades involucradas. Por lo que surge la necesidad de implantar un software que permita optimizar la gestión desarrollada por el departamento, que permita el control de los procesos antes mencionados. Para lograrlo se utilizó en la elaboración del sistema SIP la metodología RUP y la herramienta UML, el lenguaje de programación de código libre PHP y el Sistema Manejador de Base de Datos MySQL. El sistema SIP, es una herramienta que lleva a cabo actividades relacionadas a la gestión de: Empleados, Pasantes, vacaciones, permisos y Reposos.

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
MUJICA, VICTOR.	ROL	CA	AS: X	TU	JU
	CVLAC:	14.054.907			
	E_MAIL	VmujicaUDO@hotmail.com			
	E_MAIL				
TORREALBA, AQUILES	ROL	CA	AS	TU	JU: <del>X</del>
	CVLAC:				
	E_MAIL				
	E_MAIL				
RODRIGUEZ, RHONALD.	ROL	CA	AS	TU	JU: <del>X</del>
	CVLAC:				
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2010	08	13
AÑO	MES	DÍA

LENGUAJE. SPA

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESSIP.DOC	APLICACIÓN/MSWORD

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F  
G H I J K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p  
q r s t u v w x y z. 0 1 2 3 4 5 6 7 8 9.

**ALCANCE**

ESPACIAL: AREA DE PERSONAL DE LA DIVISION DE  
ADMINISTRACION DE LA GERENCIA GENERAL DE TRIBUTOS  
INTERNOS DE LA REGION NOR-ORIENTAL(OPCIONAL)

TEMPORAL: \_\_\_\_\_  
(OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:  
INGENIERO EN COMPUTACIÓN

NIVEL ASOCIADO CON EL TRABAJO:  
PRE-GRADO

**ÁREA DE ESTUDIO:**

DEPARAMENTO DE COMPUTACIÓN Y SISTEMAS

**INSTITUCIÓN:**

UNIVERSIDAD DE ORIENTE, NÚCLEO DE ANZOÁTEGUI.

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**

DERECHOS

*“Los Trabajos de Grado son Propiedad exclusiva de la Universidad y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participará en Consejo Universitario”*

**HERNÁNDEZ C., DAIRA DEL C.**

**AUTOR 1**

**AUTOR 2**

**AUTOR 3**

**MUJICA, VICTOR.**

**TUTOR**

**TORREALBA, AQUILES**

**JURADO 1**

**RODRIGUEZ, RHONALD**

**JURADO 2**

**BASTARDO, JOSÉ**

**POR LA SUBCOMISION DE TESIS**