

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



**DESARROLLO DE UN SOFTWARE PARA LA GESTIÓN Y CONTROL DE
LAS ACTIVIDADES ADMINISTRATIVAS DE UNA EMPRESA
CONSTRUCTORA UBICADA EN PUERTO LA CRUZ - ESTADO
ANZOÁTEGUI.**

Presentado por:

Lenys Marina Rengel Vásquez

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 2.010.

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



**DESARROLLO DE UN SOFTWARE PARA LA GESTIÓN Y CONTROL DE
LAS ACTIVIDADES ADMINISTRATIVAS DE UNA EMPRESA
CONSTRUCTORA UBICADA EN PUERTO LA CRUZ - ESTADO
ANZOÁTEGUI.**

Asesor:

Ing. Víctor Mujica
Asesor Académico

Puerto la Cruz, Agosto 2.010.

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



**DESARROLLO DE UN SOFTWARE PARA LA GESTIÓN Y CONTROL DE
LAS ACTIVIDADES ADMINISTRATIVAS DE UNA EMPRESA
CONSTRUCTORA UBICADA EN PUERTO LA CRUZ - ESTADO
ANZOÁTEGUI.**



Ing. Gabriela Veracierta
(Jurado)

Ing. Pedro Dorta
(Jurado)

Ing. Víctor Mujica
(Asesor Académico)

RESOLUCIÓN

De acuerdo con el artículo 41 del reglamento de Trabajo de Grado:

“Los Trabajos de Grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participará al Consejo Universitario, para su autorización”

DEDICATORIA

A Diosito y a mi Virgencita del Valle que siempre están presente en todo lo que hago, a ellos les debo la dicha de hacer mi sueño realidad.

A mi hermosa madre, por darme su apoyo incondicional sólo el que una verdadera madre sabe dar; tan preocupada y dedicada.

A mi querido padre porque siempre hizo hasta lo imposible por velar por mi bienestar y el de toda la familia.

A mis dos corazones Javier y Sebastian, los cuales fueron mi fuente de inspiración para la culminación de esta meta trazada.

A mi amado esposo por su amor, su compañía, su comprensión, por apoyarme siempre y ayudarme a lograr mis metas.

A mis hermanos (Mary, Dany, David, Denis, Cesar), por formar parte de mi hermosa familia, los quiero mucho!.

A mis sobrinos (Daniela, Angelo, Anderson, Paulina, Emir, Daniel A.) por llenar mi vida de alegría... los quiero muchísimo!

AGRADECIMIENTOS

Agradezco a Dios todopoderoso por haberme dado una familia tan maravillosa que me ayudara en tan duros momentos.

A mi Virgencita del Valle por nunca dejarme desamparada e iluminarme el camino correcto.

A mis padres por su apoyo y comprensión en todo momento, sin su presencia hubiese sido imposible lograr este objetivo; este triunfo es de ustedes.

A mi esposo por ser incondicional, por estar siempre a mi lado y ayudarme en todo lo que he necesitado.

A mis hermanos por ser tan incondicionales en mi vida; en especial a Denis por sus oportunas palabras de aliento y por no dejarme flaquear desde los inicios de mi carrera; a ti Cesar por siempre estar dispuesto a ayudarme y por tenerme presente para compartir tus chistes, los quiero mucho!.

A mis hijos, por ser todos unos hombrecitos y portarse bien para que su mami terminara ésta larga tarea; gracias Javi por entenderme y saber que todo lo hacia por ustedes, ya tu sabes papi como siempre me decías “mami cuando te gradúes vamos a tener mas dinero y podré comprar lo que quiera”, los amo mis amores.

Muy especialmente a María Luisa, porque sin su ayuda no estaría hoy redactando estas líneas; nunca encontraré un término que de la talla con todo lo que siento y quisiera expresarte en estas líneas, sólo puedo decirte que te estoy muy

agradecida por toda la ayuda prestada. Gracias por ser tan ESPECIAL. Dios te premiará por todo lo que has hecho por mí!

A mi gran amiguita Rene El Nemer por ayudarme a realizar mi sueño realidad, un día prometiste ayudarme y si allí estuviste, siempre tan incondicional, gracias por ofrecerme tu amistad y ayuda.

A mi comadre y amiga Naileth Olivier por siempre estar presente en los momentos en que mas te necesite!

A mi amigos Douglas y Daira por siempre brindarme apoyo y ayuda necesaria para salir adelante.

Agradezco a mis profesores, por haber compartido sus conocimientos.

A todos aquellos que de una u otra forma colaboraron para que se culminara con éxito esta dura tarea.

RESUMEN

La empresa Tecnología Construcción Soluciones Integrales, TECOM, C.A. realiza todas sus actividades administrativas de forma manual, trayendo como consecuencia retraso en el procesamiento de la información, por tal motivo se propuso el proyecto de Grado titulado: **DESARROLLO DE UN SOFTWARE PARA LA GESTIÓN Y CONTROL DE LAS ACTIVIDADES ADMINISTRATIVAS DE UNA EMPRESA CONSTRUCTORA UBICADA EN PUERTO LA CRUZ - ESTADO ANZOÁTEGUI**; el cual permitió la automatización de los procesos administrativos gestionados por la mencionada empresa, tales como: inserción de trabajadores, realización de recibos de pagos, etc. **SISTEMA ADMINISTRATIVO EMPRESAS TECOM (SADET)** fue desarrollado siguiendo la metodología del proceso Unificado de Desarrollo de Software, debido a su eficiencia en desarrollo de software aplicando las fases de inicio, elaboración, construcción y transición. Estas a su vez se llevan a cabo a través de una o varias iteraciones las cuales constan de cinco flujos de trabajo fundamentales como son: captura de requisitos, análisis, diseño, implementación y prueba. Se usaron Macromedia Firework 8 y Dreamweaver 8, para el diseño y construcción del sistema. La arquitectura se construyó de acuerdo al patrón cliente-servidor, razón por la cual se utilizó el lenguaje de programación **PHP**, el cual es el apropiado para construir sistemas con bases de datos usando el manejador MySQL.

CONTENIDO

RESOLUCIÓN	iv
DEDICATORIA	v
AGRADECIMIENTOS	vi
RESUMEN.....	viii
CONTENIDO	ix
LISTA DE FIGURAS	xiv
LISTA DE TABLAS	xvi
CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA	xvii
1.1. Planteamiento del Problema.....	18
1.2. Objetivos	22
1.2.1. Objetivo General	22
1.2.2. Objetivos Específicos.....	22
CAPÍTULO 2: MARCO TEÓRICO.....	24
2.2.8. Base de Datos	33
2.2.9. PHP	34
2.2.11. MySQL.....	34
CAPÍTULO 3: FASE DE INICIO.....	35
3.1. Fase de Inicio	36
3.2. Captura de requisitos.....	37
3.2.1. Requisitos Candidatos.....	37
3.2.2. Contexto del Sistema	38

3.2.2.1. Modelo de Dominio	38
3.2.3. Identificación de los riesgos.....	41
3.2.4. Requisitos Funcionales	44
3.2.4.1. Identificación de los Actores.....	45
3.2.4.2. Identificación de los casos de uso	45
3.2.4.3. Descripción del Diagrama del caso de uso de Contexto del Sistema Administrativo.	46
3.2.4.4. Caso de uso Entrar al Sistema.....	48
3.2.4.5. Caso de uso Gestionar Nómina.....	48
3.2.4.6. Caso de uso Gestionar Ingresos.	49
3.2.4.7. Caso de uso Gestionar Egresos.	49
3.2.4.8. Caso de uso Generar Reportes Impresos.....	50
3.2.4.9. Caso de uso Configurar Sistema.	50
3.2.5. Requisitos No Funcionales.....	51
3.3. Análisis.....	51
3.3.1. Diagrama de Clases de Análisis.....	52
3.3.1.1. Clases de Interfaz:	52
3.3.1.2. Clases de Control:	54
3.3.1.3. Clases de Entidad:	55
3.3.2. Diagramas de Clases de Análisis para los Casos de Uso del Sistema ..	56
3.3.2.1. Diagrama de Clases de Análisis: Entrar al Sistema	56
3.3.2.2. Diagrama de Clases de Análisis: Gestionar Nómina	56
3.3.2.3. Diagrama de Clases de Análisis: Gestionar Ingresos.....	57
3.3.2.4. Diagrama de Clases de Análisis: Gestionar Egresos.....	57
3.3.2.5. Diagrama de Clases de Análisis: Generar Reportes Impresos	58
3.3.2.6. Diagrama de Clases de Análisis: Configurar Sistema.....	59
3.4. Diagrama de Paquete de Análisis.....	59

3.5. DISEÑO	61
3.5.1. Diseño de la Arquitectura	61
3.4.1.1. Identificación de Subsistemas	62
3.6. Conclusión de la fase de inicio	62
CAPÍTULO 4: FASE DE ELABORACIÓN	63
4.1. Fase de Elaboración	64
4.2. Captura de Requisitos	65
4.2.1. Diagrama de Casos de Uso	65
4.2.1.1. Descripción del caso de uso extendido de Generar Nómina.....	67
4.2.1.2. Descripción del caso de uso extendido de Generar Reportes Impresos	
.....	67
4.3. Análisis.....	68
4.3.1. Análisis de los casos de uso	68
4.3.1.1. Diagrama de clases de análisis de la realización del caso de uso:	
Gestionar Nómina	68
4.3.1.2. Diagrama de clases de análisis de la realización del caso de uso:	
Generar Reportes Impresos.....	69
4.3.2. Interacción de las clases de análisis	70
4.3.2.1. Diagrama de colaboración de la realización del caso de uso: Gestionar	
Nómina.....	71
4.3.2.2. Diagrama de colaboración de la realización del caso de uso: Gestionar	
Reportes Impresos.....	72
4.4. Diseño	72
4.4.1. Diagrama de clases de diseño	73
4.4.2. Diagrama de Secuencia.....	61
4.4.2.1. Diagrama de secuencia del caso de uso: Gestionar Nómina.....	62

4.4.2.2. Diagrama de secuencia del caso de uso: Generar Reportes.....	64
4.4.3. Subsistemas de la Aplicación.....	66
4.4.4. Prototipo de la Interfaz de Usuario	68
4.4.5. Diseño de las bases de datos	69
4.4.5.1. Identificación de la Tablas	69
4.5. Resumen de la fase.....	71
CAPÍTULO 5: FASE DE CONSTRUCCIÓN.....	72
5.1. Fase de Construcción	73
5.2. Escogencia del Lenguaje de Programación.....	74
5.3. Implementación.....	76
5.3.1. Diagrama de Componentes de SADET.....	77
5.3.2. Implementación de la base de datos.....	77
5.4. Pruebas	80
5.4.1. Pruebas de unidad	81
5.4.2. Pruebas de integración	84
5.4.2.1. Prueba de Integración del componente Crear Beneficios	86
5.5. Codificación de los Componentes de SADET.....	90
5.6. Resumen de la fase.....	105
CAPÍTULO 6: FASE DE TRANSICIÓN.....	106
6.1. Fase de Transición.....	107
6.2. Preparación de la versión Beta.....	108
6.3. Instalación de la versión Beta	108
6.4. Reacción a los resultados de las pruebas.....	108
6.5. Resumen de la fase.....	108

CONCLUSIONES	10809
RECOMENDACIONES.....	108
BIBLIOGRAFÍA	10811

LISTA DE FIGURAS

FIGURAS	<i>Pág.</i>
 CAPÍTULO 3 : FASE DE INICIO	
Figura 3.1. : Diagrama de los flujos de trabajo fundamental en la fase de inicio	23
Figura 3.2. : Diagrama de Clases de Dominio	27
Figura 3.3. : Diagrama de caso de uso contextual	34
Figura 3.4. : Estereotipos de las clases de análisis	39
Figura 3.5. : Diagrama de Clases de Análisis: Entrar al Sistema	43
Figura 3.6. : Diagrama de Clases de Análisis: Gestionar Nómina	43
Figura 3.7. : Diagrama de Clases de Análisis: Gestionar Ingresos	44
Figura 3.8. : Diagrama de Clases de Análisis: Gestionar Egresos	44
Figura 3.9. : Diagrama de clase de Análisis: Reportes Impresos	45
Figura 3.10. : Diagrama de clase de Análisis: Configurar Sistema	45
Figura 3.11. : Representación de paquetes	46
Figura 3.12. : Diagrama de paquete de análisis del sistema SADET	47
Figura 3.13. : Subsistemas distribuidos en capas	49
 CAPÍTULO 4 : FASE DE ELABORACIÓN	
Figura 4.1. : Diagrama de los flujos de trabajo fundamental en la fase de elaboración	51
Figura 4.2. : Diagrama de caso de uso extendido del sistema	53
Figura 4.3. : Diagrama de caso de uso detallado: Gestionar Nómina	54
Figura 4.4. : Diagrama de clase de análisis Gestionar Nómina	55
Figura 4.5. : Diagrama de clase de análisis Reportes Impresos	56

Figura 4.6. : Diagrama de colaboración Gestionar Nómina	57
Figura 4.7. : Diagrama de colaboración Reportes Impresos	58
Figura 4.8. : Diagrama de clases de diseño del sistema SADET	60
Figura 4.9. : Diagrama de secuencia Gestionar Nómina	64
Figura 4.10. : Diagrama de secuencia Reportes Impresos	66
Figura 4.11. : Dependencia entre subsistemas distribuidos en capas	67
Figura 4.12. : Prototipo de la interfaz de usuario del sistema	68

CAPÍTULO 5 : FASE DE CONSTRUCCIÓN

Figura 5.1. : Diagrama de los flujos de trabajo fundamental en la fase de Construcción	74
Figura 5.2. : Diagrama de Componentes del modulo Nómina	79
Figura 5.3. : Definiciones de la tabla empleados de la base de datos	80
Figura 5.4. : Fases de Integración del modulo Nómina	86
Figura 5.5. : Componente Gestionar Empleados_Paso 1	88
Figura 5.6. : Componente Gestionar Empleados_Paso 2	89
Figura 5.7. : Componente Gestionar Empleados_Paso 3	90
Figura 5.8. : Componente Gestionar Empleados_Paso 4	91

CAPÍTULO 6 : FASE DE TRANSICIÓN

Figura 6.1. : Diagrama de los flujos de trabajo fundamental en la fase de Transición	109
---	-----

LISTA DE TABLAS

TABLAS	<i>Pág.</i>
 CAPÍTULO 3 : FASE DE INICIO	
Tabla 3.1 : Descripción de las clases de dominio del sistema	26
Tabla 3.2 : Descripción de los actores del sistema	32
Tabla 3.3 : Descripción de los casos de uso del sistema	32
Tabla 3.4 : Casos de uso: Entrar al Sistema	34
Tabla 3.5 : Casos de uso: Gestionar Nómina	35
Tabla 3.6 : Casos de uso: Gestionar Ingresos	35
Tabla 3.7 : Casos de uso: Gestionar Egresos	36
Tabla 3.8 : Casos de uso: Reportes Impresos	36
Tabla 3.9 : Casos de uso: Gestionar Sistema	37
Tabla 3.10 : Clases de Interfaz	40
Tabla 3.11 Clases de Control	41
Tabla 3.10 : Clases de Entidad	42
 CAPÍTULO 5 : FASE DE CONSTRUCCIÓN	
Tabla 5.1 : Clases de equivalencia para las pruebas de caja negra de Ingresar Empleado	83
Tabla 5.2 : Casos de prueba de caja negra para Ingresar Empleado	84

CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La empresa Tecnología Construcción Soluciones Integrales, TECOM, C.A. es una empresa totalmente venezolana que está a la vanguardia con la tecnología; fundada el 23 de junio del 2005; dedicada al montaje industrial en disciplinas de electricidad, instrumentación, suministro de equipos, consultoría, automatización de procesos y telecomunicaciones. La empresa está enfocada en prestar un servicio de máxima calidad a nivel nacional en las áreas de: montaje, pruebas, calibración, así como también, mantenimientos preventivos, correctivos en todo tipo de instalaciones o plantas de procesos; servicios profesionales y técnicos , incluye investigación, estudio, asesorías, proyectos, informes, planos, cálculos, inspección, pruebas de sistemas automatizados y proyectos multidisciplinarios de ingeniería, para ello, cuentan con un personal altamente calificado con excelente trayectoria en dichas áreas.

Actualmente, la empresa lleva a cabo una serie de actividades administrativas, el manejo de esta información se realiza de forma manual y para efectuar los cálculos referentes a las distintas operaciones que se deben realizar, se utiliza como herramienta la hoja de cálculo Microsoft Office Excel; una vez realizado este trabajo se generan los recibos de pago o los reportes impresos, los cuales se encuentra guardados en documentos de Microsoft Office Word; entre las actividades realizadas por la empresa se pueden mencionar:

- 👉 Relación de ingresos - egresos por contrato; son asentadas todas las transacciones que se realizan a diario en la empresa, esta información es modificada a medida que se genera un gasto, del cual se debe tomar en cuenta su nombre, fecha, monto, entre otros; así como también se deben registrar todos los ingresos de la empresa generados diariamente, mientras que lo relacionado con las facturas canceladas y por cancelar se encuentran archivadas.



- 👍 Relación de ingresos - egresos general, al final de mes realizan un balance general con todas las transacciones de los contratos manejados.
- 👍 Movimientos Bancarios, se lleva un control de la información asociada con: cheques emitidos, depósitos realizados y el capital inicial con que cuenta la empresa.
- 👍 Relación de gastos refleja todos los egresos producidos por la empresa desde la iniciación de cada contrato.
- 👍 Efectuar los cálculos referentes a las distintas operaciones que se deben realizar en la nómina de empleados de la empresa, la cual se encuentra dividida entre empleados fijos, y personal contratado; ellos difieren uno del otro por el cargo que desempeñan y el salario que reciben.
- 👍 Uno de los procesos más tediosos llevados a cabo es la emisión de los recibos de pago, el cual consiste en hacer el llenado de la información asociada con cada empleado, estos pueden ser de carácter semanal, quincenal o mensual y son realizados manualmente por la secretaria.

Debido al tipo de actividad a la que se dedican y la cantidad de empleados y contratos con que cuenta la empresa, se manejan grandes volúmenes de información, lo cual acarrea una serie de desventajas, entre las cuales se pueden mencionar:

- 👍 Descontento al personal que trabaja en el área, debido a que la secretaria tarda aproximadamente 2 días para transcribir todos los ingresos y egresos y para hacer los recibos de pago de cada empleado perteneciente a la empresa, lo que le produce agotamiento físico por el número de horas en el que permanece sentada e irritación de la vista.



- 👍 Inseguridad en el resguardo de la información es evidente ya que cualquier persona no autorizada puede tener acceso a ella y realizar modificaciones en algún dato.
- 👍 Ausencia de un historial de los empleados retirados de la empresa.
- 👍 Existe un desperdicio en el tiempo de búsqueda, porque cuando se desea obtener una información determinada, se requiere buscar en todo el archivo y realizar los cálculos correspondientes.

Es por esto que se expone la necesidad de encontrar un medio más eficiente que facilite la realización de actividades para mejorar y asegurar el uso de los recursos pertenecientes a la empresa y así obtener excelentes resultados en un tiempo significativamente reducido; debido a tales consecuencias, surge la necesidad de: **“Desarrollo De Un Software Para La Gestión Y Control De Las Actividades Administrativas De Una Empresa Constructora Ubicada En Puerto La Cruz - Estado Anzoátegui.”**

El propósito de este software será mejorar la eficiencia y aumentar la productividad del personal que labora en ella, al reducir el tiempo de espera por parte de los socios al momento de hacer una solicitud de un servicio hasta la obtención de la respuesta de dicha solicitud; esta automatización permitirá:

- 👍 Procesar todos los datos relacionados con los empleados de la empresa.
- 👍 Generar los recibos de pago de los empleados que laboran en la empresa, considerando el proceso de inserción de los beneficios y deducciones para determinar el salario de los empleados; además de las prestaciones



sociales acumuladas por los empleados y los bonos recibidos por los empleados (ayuda de ciudad, transporte, vacacional y/o navideño).

- 👍 Generar un historial de todos los empleados retirados de la empresa así como también los empleados jubilados.
- 👍 Registrar y consultar los datos requeridos a cada una de las relaciones de ingresos – egresos para así realizar los cálculos necesarios que permitan totalizar y guardar automáticamente toda esta información, creando un historial de las transacciones y así lograr un mejor control de los procesos administrativos.
- 👍 Generar reportes informativos de las diferentes operaciones realizadas por el sistema.
- 👍 Actualizar la data asociada a los empleados, contratos, cargos, emisores y destinatarios de los cheques.

Para el desarrollo de este software, se utilizará la metodología del Proceso Unificado de Desarrollo de Software, la cual se adapta a través de proyectos variados en tamaños y complejidad, basado en el uso de la tecnología orientada a objetos y también guía a los equipos del proyecto en cómo administrar el desarrollo iterativo de un modo controlado, mientras se balancean los requerimientos de la empresa y los riesgos del proyecto. Debido a la utilización de esta metodología, es necesario seleccionar un lenguaje de programación orientado a objetos, para elaborar el código del sistema y que se adapten a los requerimientos para el diseño del software. Para el desarrollo del mismo se utilizará PHP (Preprocesador de Hipertexto), el cual es un lenguaje de "código abierto" interpretado, basado en herramientas con licencia de software libre; se utiliza entre otras cosas para la programación de páginas web



activas y/o dinámicas, y se destaca por su capacidad de mezclarse con el código HTML. PHP está concentrado en la programación de scripts en el lado del servidor, lo que significa que el servidor ejecuta cada una de las órdenes y las envía al cliente en formato HTML. Para modelar los objetos se hará uso del lenguaje unificado de modelado (UML), el cual proporciona un modo estándar de visualizar, especificar, documentar, construir y comunicar los artefactos de un sistema basado en el software. Para el diseño de la base de datos se utilizará el Gestor de Bases de Datos MySQL, el cual es un sistema multihilo y multiusuarios; por medio del cual se aplicarán técnicas que permitan la integridad de la información, eviten la redundancia y anomalías de los datos.

1.2. OBJETIVOS

1.2.1. Objetivo General

Desarrollo De Un Software Para La Gestión Y Control De Las Actividades Administrativas De Una Empresa Constructora Ubicada En Puerto La Cruz - Estado Anzoátegui.

1.2.2. Objetivos Específicos

- ☝ Identificar los requerimientos del sistema tomando en cuenta las necesidades existentes en la administración de la empresa.

- ☝ Elaborar el diseño inicial de la arquitectura candidata del sistema, partiendo de los requisitos analizados.



Planteamiento del Problema

- 👍 Diseñar las interfaces del usuario, las bases de datos y la arquitectura del sistema.

- 👍 Codificar cada uno de los módulos que integran el proyecto.

- 👍 Realizar pruebas a cada uno de los módulos del software.

CAPÍTULO 2: MARCO TEÓRICO

2.1. ANTECEDENTES

En el Departamento de Computación y Sistemas de la Universidad de Oriente Núcleo Anzoátegui, se han realizado proyectos que facilitan el manejo administrativo o control de información de instituciones y empresas; las cuales utilizan técnicas y metodologías que pueden contribuir en esta investigación. A continuación se nombrará algunos de estos trabajos:

- 👍 (Ochoa, Y. y Madruga, R., 2006). **“Desarrollo de un software para la gestión y control de las actividades administrativas de la central telefónica de la Universidad de Oriente, Núcleo Anzoátegui”**, como requisito parcial para la obtención del título de Ingeniero en Computación en la Universidad de Oriente, Núcleo de Anzoátegui. El software fue desarrollado aplicando el Proceso Unificado de Desarrollo de Software usando UML para el modelado del sistema.
- 👍 (González, M. y Porras, K., 2006). **“Desarrollo de un software para la automatización de las actividades relacionadas con la nómina de empleados de una alcaldía, adscrita a la gobernación del estado Anzoátegui”**, como requisito parcial para la obtención del título de Ingeniero en Computación en la Universidad de Oriente, Núcleo de Anzoátegui. Para la elaboración del Proyecto se utilizó la metodología del Proceso Unificado de Desarrollo de Software.
- 👍 (Mujica, V. y García, M., 2005). **“Desarrollo de un software para la automatización de los procesos administrativos del Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo Anzoátegui”**, como requisito parcial para la obtención del título de Ingeniero en Computación en la Universidad de Oriente, Núcleo de Anzoátegui. El modelado del sistema se hizo mediante el Lenguaje Unificado de Modelado (UML) aplicando Proceso Unificado de desarrollo de software.



- ☝ (Simoni, M., 2004). **“Desarrollo un sistema de información para la gestión de procesos administrativos que mejore el desempeño de las actividades del Departamento de Ciencias de la Unidad de Estudios Básicos de la Universidad de Oriente, Núcleo de Anzoátegui”**, como requisito parcial para la obtención del título de Ingeniero en Computación en la Universidad de Oriente, Núcleo de Anzoátegui. Los lineamientos de Proceso Unificado de Desarrollo de Software usando UML fueron los que se utilizaron en este trabajo para el modelado del sistema.

- ☝ (Otero, J., 2004). **“Desarrollo de una aplicación para el control administrativo de una organización de una organización farmacéutica”**. (2004). Requisito parcial para la obtención del título de Ingeniería en Computación en la Universidad de Oriente, Núcleo de Anzoátegui. Se utilizó como ayuda para la metodología de proceso unificado para el desarrollo de software.

2.2. MARCO TEÓRICO

2.2.1. Ingeniería de Software

La ingeniería del software es una rama de la ingeniería que aplica los principios de la ciencia de la Computación y la matemática para lograr soluciones costo-efectivas a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costo-efectivo.

2.2.2. Proceso Unificado de Desarrollo de Software

Es un proceso de desarrollo de software que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. Está basado en componentes software interconectados a través de interfaces bien definidas



y utiliza el Lenguaje Unificado de Modelado (UML) como parte vital del proceso para representar los esquemas del sistema software.

El Proceso Unificado es producto de la fusión de conocimientos y la experiencia de Jacobson I., Booch G. y Rumbaugh J.

👉 *Dirigido por Casos de Uso*

Los casos de uso representan la especificación funcional del sistema y estos son tomados como la entrada principal para el análisis, diseño, implementación y pruebas del sistema, lo que proporciona al usuario resultados importantes.

👉 *Centrado en La Arquitectura*

En la arquitectura se determinan las diferentes vistas del sistema en desarrollo incluyendo aspectos estáticos y dinámicos más significativos del mismo.

👉 *Iterativo e Incremental*

Una iteración es el resultado de la división del proyecto en mini-proyectos. Cada iteración se da en cada uno de los flujos de trabajo lo que conlleva a un incremento, que consiste en un crecimiento del producto en cuanto a funcionalidad.

2.2.3. Fases del Proceso Unificado [8]

El proceso Unificado es cíclico, cada ciclo está formado por cuatro fases: inicio, elaboración, construcción y transición.

👉 *Fase de Inicio*

En esta fase se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. Esencialmente esta fase responde a las siguientes preguntas:



- 1) ¿Cuáles son las principales funciones del sistema para sus usuarios más importantes?
- 2) ¿Cómo podría ser la arquitectura del sistema?
- 3) ¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?

La respuesta a la primera pregunta se encuentra en un modelo de casos de uso simplificado que contenga los casos de uso más críticos. Una vez obtenidos, la arquitectura es provisional, y consiste típicamente en un simple esbozo que muestra los subsistemas más importantes. En esta fase, se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración, y se estima el proyecto de manera aproximada.

👉 Fase de Elaboración

Durante esta fase se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. La relación entre la arquitectura del sistema y el propio sistema es primordial. Por tanto la arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representan al sistema entero. Esto implica que hay vistas arquitectónicas del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y modelo de despliegue. La vista del modelo de implementación incluye componentes para probar que la arquitectura es ejecutable. Durante esta fase del desarrollo, se realizan los casos de uso más críticos que se identificaron en la fase de inicio. El resultado de esta base es la línea base de la arquitectura.

👉 Fase de Construcción

En esta fase se crea el producto, la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. El grueso de los recursos requeridos se emplea durante esta fase del desarrollo. Sin embargo la



arquitectura del sistema es estable, aunque se pueden describir formas mejores de estructurar el sistema. Al final de esta fase, el producto contiene todos los casos de uso que la dirección y el cliente han acordado para el desarrollo de esta versión. Sin embargo, puede que no esté completamente libre de defectos. Muchos de estos defectos se descubrirán y solucionarán durante la fase de transición.

👉 Fase de Transición

Durante esta fase se cubre el período durante el cual el producto se convierte en versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el producto e informa de defectos y deficiencias. Esta fase conlleva actividades como la fabricación, formación del cliente y la corrección de los defectos que se encuentren tras la entrega.

2.2.4. Flujos de Trabajos Fundamentales en el Proceso Unificado

Son las distintas iteraciones que son llevadas a cabo en las fases del proceso. Los flujos fundamentales se dividen en: Requisitos, Análisis, Diseño, Implementación y Pruebas. Dichos flujos no ocurren una sola vez; sino que se repiten más bien en cada iteración, una y otra vez, como flujos de trabajos iterativos. Cada repetición, sin embargo, se diferencia en los detalles que se enfrentan o asuntos centrales de cada iteración.

👉 Requisitos

El esfuerzo principal en la fase de requisitos es desarrollar un modelo del sistema que se va a construir, y la utilización de los casos de uso en una forma adecuada de crear este modelo. Esto es debido a que los requisitos funcionales se estructuran de forma natural mediante casos de uso, y a que la mayoría de los otros requisitos no funcionales son específicos de un solo caso de uso, y pueden tratarse en el contexto de ese caso de uso.



Análisis

El esfuerzo principal en la fase de requisitos es desarrollar un modelo del sistema que se va a construir, y la utilización de los casos de uso en una forma adecuada de crear este modelo. Esto es debido a que los requisitos funcionales se estructuran de forma natural mediante casos de uso, y a que la mayoría de los otros requisitos no funcionales son específicos de un solo caso de uso, y pueden tratarse en el contexto de ese caso de uso.

Diseño

En el diseño se modela el sistema y se encuentra la forma para que soporte los requisitos que se suponen. El principal resultado del diseño es el modelo de diseño. Se esfuerza en conservar la estructura del sistema impuesta por el modelo de análisis, y que sirve como esquema para la implementación.

Implementación

El propósito fundamental de este flujo es esencial en la implementación del sistema en términos de componentes, es decir, código fuente, guiones, ficheros, etc.

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 y durante la fase de transición, para tratar defectos tardíos como los encontrados con la distribución beta del sistema.

Prueba

El propósito esencial de éste flujo es comprobar el resultado de la implementación mediante las pruebas de cada construcción, incluyendo tanto construcciones internas como intermedia, así como la versiones finales del sistema que van a ser entregadas a terceras partes.



2.2.5. Lenguaje Unificado de Modelado (UML)

Es un lenguaje de modelamiento para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema software. UML modela un sistema como una colección de objetos discretos que interactúan para desempeñar un trabajo que en última instancia beneficia a un usuario externo, además está relacionado con la captura, comunicación y nivelación de conocimientos.

2.2.6. Modelos

Un modelo es una simplificación de la realidad. En un modelo se han de incluir los elementos que tengan más relevancia y omitir los que no son interesantes para el nivel de abstracción que se ha elegido, además nos proporciona los planos de un sistema, desde los más generales, hasta los más detallados.

👉 Modelo de Casos de Uso

Permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos. Contiene actores, casos de uso y sus relaciones.

👉 Modelo de Análisis

Permite refinar los requisitos y razonar sobre los aspectos internos del sistema, incluyendo los recursos internos compartidos. En el modelo de análisis se tiene una estructura centrada en el mantenimiento en aspectos tales como la flexibilidad ante los cambios y la reutilización.

👉 Modelo de Diseño

Es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.



Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc.

Modelo de Despliegue

Define la arquitectura física del sistema por medio de nodos interconectados, se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

2.2.7. Diagramas

Un diagrama es la representación gráfica de un conjunto de elementos, usualmente representado como un grafo conectado de vértices (elementos) y arcos (relaciones).

Diagrama de Casos de Uso

Muestra un conjunto de casos de uso con sus actores implicados y sus relaciones desde un punto de vista estático, lo cual permite visualizar el comportamiento externo del programa.

Diagrama de Clase

Muestra una colección de elementos (estáticos) declarativos y; un conjunto de clases, interfaces y colaboraciones y las relaciones entre éstos.

Diagrama de Objeto

Es un diagrama que muestra un conjunto de objetos y sus relaciones en un momento determinado. También muestran el diseño o los procesos de un sistema desde un punto de vista estático.



👉 Diagrama de Colaboración

Es un diagrama de interacción que enfatiza la organización estructural de los objetos que envían y reciben mensajes. Se puede decir también que es “un diagrama que muestra las interacciones organizadas alrededor de instancias y de los enlaces entre ellas”.

👉 Diagrama de Secuencia

Muestran la secuencia de mensajes que se van lanzando los objetos implicados en una determinada operación del programa.

2.2.8. Base de Datos

Es un medio de almacenamiento orientado a satisfacer concurrentemente las necesidades de información de múltiples usuarios y a mantener un control centralizado de los datos como un recurso organizacional. Una base de datos debe coleccionar archivos relacionados, con la finalidad de permitir el manejo de la información de alguna compañía. Cada uno de dichos archivos pueden ser vistos como una colección de registros y cada registro está compuesto de una colección de campos. Cada uno de los campos de cada registro permite llevar información de alguna característica o atributo de alguna entidad del mundo real.

👉 Sistema de Administración de Base de Datos

Es un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra tanto a los usuarios como al administrador de la base de datos, los medios necesarios para describir, manipular y utilizar los datos almacenados, manteniendo la integridad, confidencialidad y seguridad. Mediante el suministro de una interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y consultas se simplifica y facilita el acceso a los datos.



2.2.9. PHP

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. Su nombre oficial es: Hypertext Processor (PHP) .Entre sus principales características se puede destacar su potencia, alto rendimiento y facilidad de aprendizaje. PHP es una eficaz herramienta de desarrollo para los programadores web, ya que permite la creación de sitios web dinámicos.

2.2.11. MySQL

Es un sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

CAPÍTULO 3: FASE DE INICIO

3.1. FASE DE INICIO

El proceso unificado de desarrollo de software se divide en cuatro fases (inicio, elaboración, construcción y transición), y a su vez cada fase se divide en iteraciones. Estas iteraciones pasan por una serie de flujos de trabajo fundamentales como son: requisito, análisis, diseño, implementación y prueba como se muestra en la figura 3.1.

En la fase de inicio se da comienzo al Proceso Unificado de Desarrollo de Software; es donde se realiza la especificación funcional del sistema, definiendo el ámbito en que se desenvuelve el sistema, lo que va a permitir delimitar el alcance y esbozar la arquitectura necesaria que le brinde soporte al sistema. Además se establecen las bases determinantes de la viabilidad, se establecen los requisitos, se identifican los posibles obstáculos que pongan en duda el término del proyecto, se hace un esquema del sistema para su comprensión mediante la identificación de sus funcionalidades; es decir, se establece un concepto general inicial del sistema.

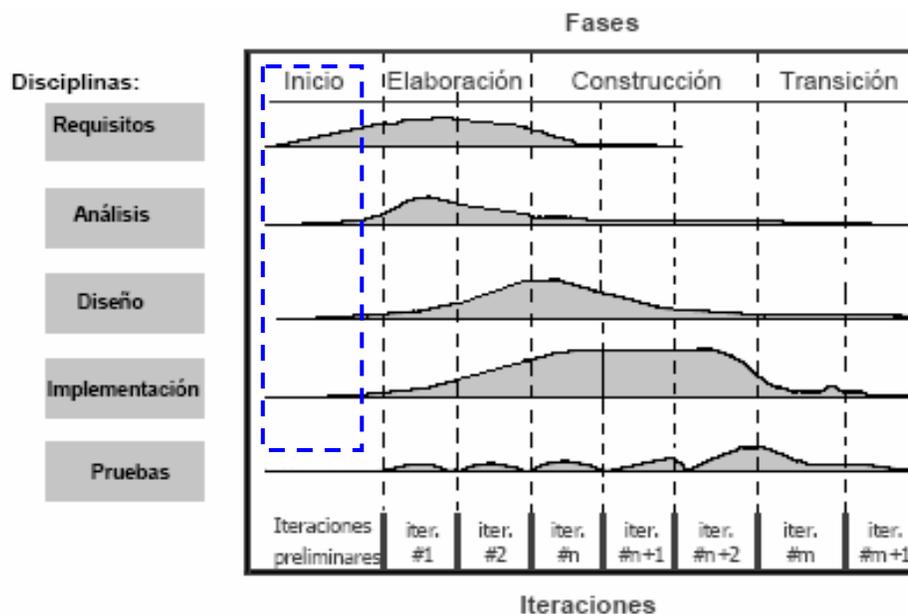


Figura 3.1. : Diagrama de los flujos de trabajo fundamental en la fase de inicio
(Jacobson I., Booch G, Rumbaugh J, 2000).N)



Como se puede apreciar en la figura 3.1, la mayor parte del trabajo de la fase de inicio se lleva a cabo en el primer flujo de trabajo de requisitos, seguido de los flujos de análisis y diseño, indicando así que en la etapa se identifican y asignan prioridades a los riesgos más importantes presentes en el proyecto, los cuales serán colocados en una lista para su control y posterior mitigación.

A continuación se muestran solo los primeros flujos de trabajo relacionados con el proyecto SADET que son: captura de requisitos y análisis debido a que esta fase no demanda trabajo para los flujos de diseño, implementación y pruebas.

3.2. CAPTURA DE REQUISITOS

En la captura de requisitos se desarrolla el modelo del sistema que se va a construir mediante la utilización de casos de uso puesto que de este modo los requisitos funcionales son estructurados de forma natural. Este flujo de trabajo incluye los siguientes pasos:

3.2.1. Requisitos Candidatos

Los requisitos candidatos son un conjunto de ideas que aparecen durante la vida de un sistema, por parte de los clientes, usuarios, analistas y desarrolladores. A continuación se especifica los principales requisitos candidatos del proyecto:

- 👍 Generar la nómina de los empleados que laboran en la empresa.
- 👍 Realizar el proceso de inserción de los beneficios y de las deducciones, para determinar el salario de los empleados.
- 👍 Generar las prestaciones sociales acumuladas por los empleados, que ya no forman parte de la nómina.



- 👍 Generar los bonos (ayuda de ciudad, transporte, vacacional y/o navideño) de los empleados.
- 👍 Generar los recibos de pago que son entregados a los empleados.
- 👍 Registro de todos los ingresos y egresos que realiza la empresa.
- 👍 Generar y almacenar cada uno de los reportes informativos, correspondientes a las actividades que se realizan en la empresa.

3.2.2. Contexto del Sistema

Para tener el firme conocimiento del entorno donde se desarrollará el software, se realizó un estudio del contexto, para observar de manera directa las actividades ligadas con el proceso administrativo que se da en la empresa. Luego se procedió a realizar entrevistas para consultar acerca de los procedimientos y funciones realizadas por la persona encargada de administrar el sistema; con la finalidad de conocer y establecer los requerimientos para el desarrollo del sistema propuesto.

3.2.2.1. Modelo de Dominio

El modelo de dominio captura los aspectos más importantes para comprender de una forma intuitiva el contexto del sistema y los representa como objetos o clases.

Dichos objetos del dominio, son todos aquellos eventos que existen y que ocurren dentro del sistema.

UML, cuenta con una notación en diagramas estructurales los cuales se usan en las vistas estáticas que explican gráficamente los modelos, esto se detalla a través del diagrama de clase de análisis. Al realizar el estudio del contexto de la empresa



constructora, se obtuvieron las clases de dominio que se describen a continuación en la tabla 3.1:

Tabla 3.1. : Descripción de las clases de Dominio del Sistema.

CLASES	DESCRIPCIÓN
Empleado	Representa información asociada con las personas que laboran en la empresa.
Nómina	Proceso que se encarga de realizar las operaciones administrativas relacionadas con la nómina de empleados.
Ingresos	Es un proceso que se encarga de ingresar todo lo relacionado con los Ingresos que entran al capital de la empresa.
Egresos	Es un proceso que se encarga de ingresar todo lo relacionado con los Egresos que salen del capital de la empresa.
Reportes Impresos	Documentos emitidos por el sistema con información requerida por el usuario.
Usuario	Persona encargada de manipular los procesos que realiza el sistema de administración.

La interacción entre las clases de dominio se ha estructurado en un diagrama que permite establecer las relaciones entre éstas. El diagrama de modelo de clase de dominio del proyecto en desarrollo que describe la *figura 3.2*, permite establecer las relaciones, asociaciones, composiciones y herencia que existen entre cada una de las clases

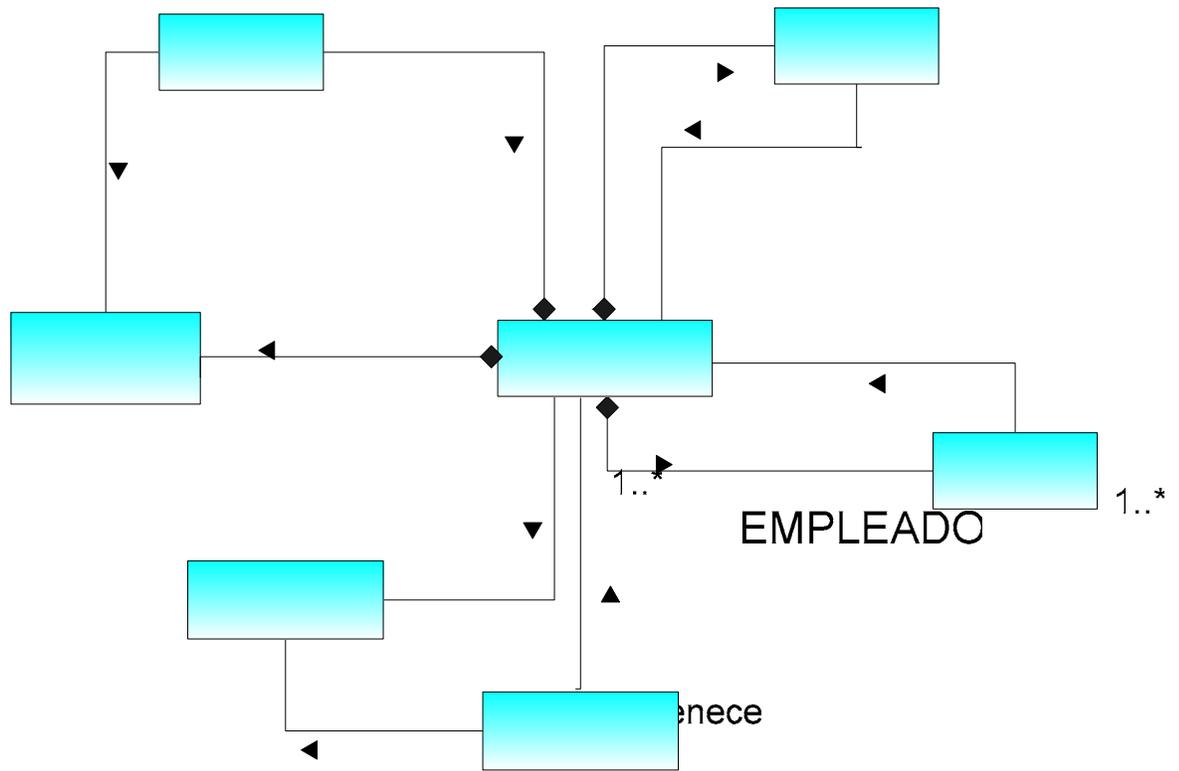


Figura 3.2. : Diagrama de Clases de Dominio del Sistema

Estas clases tienen como objetivo contribuir a la comprensión del problema que el sistema resuelve en relación a su contexto. Es de suma importancia identificar los elementos del diagrama para entender mejor el sistema. Los rectángulos representan clases, las más significativas son las de **Empleado**, **Nómina**, **Ingresos**, **Procesa** y **Egresos**, la clase Empleado registra a los empleados y la clase Nómina procesa las operaciones administrativas de los empleados; mientras que las clases Ingresos y Egresos manejan todo lo relacionado con el capital de la empresa. Las líneas representan asociaciones, el texto de una línea explica el rol de una asociación. La multiplicidad son los números y asteriscos al final de una línea, estos indican



cuantos objetos de la clase de un extremo pueden enlazarse a un objeto en el otro extremo. Por ejemplo, la línea que conecta las clases **Usuario** y **Reporte impreso** tiene una multiplicidad 1..*, dibujada en el extremo de la clase Reportes Impresos, significa que cada Usuario puede solicitar uno o más reportes impresos al sistema; en el caso de la entidad Empleado ésta posee un relación de composición con respecto a los procesos administrativos y una multiplicidad 1..*, dibujada en el extremo de la clase Empleado, lo que significa que pueden ser registrados uno o muchos empleados; Esta composición a la vez esta relacionada con la entidad Nómina con una multiplicidad 1..*, dibujada en el extremo de la clase Empleado, lo que significa que uno o muchos empleados pertenecen sólo a un tipo de nómina. En cuanto a las relaciones de Ingresos y Egresos ambas presentan unas relación de composición y una multiplicidad 1..*, dibujada en el extremo de ambas clases, debido a que se pueden procesar y registrar uno o muchos Ingresos y Egresos.

Hasta este momento se tiene un concepto inicial del sistema, una vez establecido su contexto. Lo anterior representa un alto nivel de abstracción, el cual irá descendiendo en la medida que se avance según lo demandado por el Proceso Unificado.

3.2.3. Identificación de los riesgos

Durante la fase de inicio se identifican los riesgos críticos y se va mitigando a lo largo del desarrollo del proyecto, dado que estos tienen además un impacto en la planificación, el costo e incluso en el éxito de un proyecto. Por lo que deben ser identificados los posibles riesgos en esta etapa inicial de desarrollo del software y de esa forma disminuir las posibles fallas durante el desarrollo del proyecto o su comportamiento.



Dada la influencia que tienen los riesgos en el desarrollo y planificación del software, la lista de riesgos debe contener la información necesaria para poder mitigarlo. En la lista de riesgos se incluye:

- 👍 **Descripción:** una breve descripción y se añaden a medida que se va aprendiendo.
- 👍 **Prioridad:** se le asigna una prioridad al riesgo, el riesgo se clasifica como: crítico, significativo o rutinario. Conforme se va desarrollando el proyecto es posible agregar más categorías.
- 👍 **Impacto:** se indica que partes del sistema se verán afectadas por el riesgo.
- 👍 **Responsabilidad:** se indica que individuo unidad de la organización es responsable de eliminar los riesgos.
- 👍 **Contingencia:** se indica lo que ha de hacerse en el caso de que el riesgo se materialice.

Entre los riesgos críticos que se pueden presentar en el desarrollo de SADET, se pueden mencionar los siguientes:

👍 ***No conseguir la arquitectura correcta***

- 🔗 **Descripción:** Es necesario conseguir una arquitectura de software que permita la evolución del sistema por sus diferentes fases y durante su tiempo de vida.
- 🔗 **Prioridad:** Crítico.
- 🔗 **Impacto:** Arquitectura del sistema.
- 🔗 **Responsabilidad:** Desarrollador del software.
- 🔗 **Contingencia:** Analizar en profundidad el sistema, estudiar y plantear un nuevo modelo si fuese necesario.



 **Fallas al acceder a la base de datos**

 **Descripción:** El riesgo presente en este proceso está relacionado con problemas de incompatibilidad con el manejador de bases de datos y las herramientas de programación a utilizar.

 **Prioridad:** Crítico.

 **Impacto:** Consultas a la base de datos.

 **Responsabilidad:** Desarrollador del software.

 **Contingencia:** Realizar pruebas de integración.

 **Configuración del sistema**

 **Descripción:** El riesgo presente en este proceso se relaciona con la inconsistencia de la base de datos.

 **Prioridad:** Crítico.

 **Impacto:** Todos los procesos presentes en el sistema.

 **Responsabilidad:** Desarrollador del software.

 **Contingencia:** Desarrollar un modulo en donde se pueda introducir, modificar o eliminar información de la base de datos manejadas por el sistema.

También existen riesgos secundarios, los cuales ocasionan un leve impacto en el desarrollo de todo proyecto, en el caso de SADET se pueden describir los siguientes:

 **Hardware**

 **Descripción:** No tener el hardware adecuado para dicha aplicación.

 **Prioridad:** Secundario

 **Impacto:** Implantación del sistema.

 **Responsable:** La empresa.



- 🔗 **Contingencia:** Reestructurar el hardware para aprovechar al máximo el sistema.

👍 *Aplicación*

- 🔗 **Descripción:** Incapacidad en el manejo de la aplicación.
- 🔗 **Prioridad:** Secundario.
- 🔗 **Impacto:** Manipulación del sistema.
- 🔗 **Responsable:** La empresa y el desarrollador del software.
- 🔗 **Contingencia:** Realizar planes de entrenamiento, así como un buen manual de usuario.

👍 *Plataforma*

- 🔗 **Descripción:** Incapacidad en integración a plataforma.
- 🔗 **Prioridad:** Secundario
- 🔗 **Impacto:** Implantación del sistema.
- 🔗 **Responsable:** La empresa.
- 🔗 **Contingencia:** Reestructuración del sistema operativo del equipo.

3.2.4. Requisitos Funcionales

La captura de requisitos requiere el conocimiento profundo de las necesidades del usuario, para ello se debe comprender el contexto del sistema, realizar entrevistas a los interesados, para así recopilar los aspectos que cada usuario quiere que el sistema haga, los cuales serán llevados por el analista a casos de uso, que representarán los diferentes modos en que el usuario puede utilizar el sistema.



Los desarrolladores de software y los clientes deben llegar a un acuerdo sobre las necesidades y condiciones que debe cumplir el sistema, para esto se utiliza el modelo de caso de uso.

3.2.4.1. Identificación de los Actores

Los actores representan a los usuarios, sistemas o hardware externo que interactúan con el sistema. Pueden utilizar funcionalidades suministradas por el sistema, pero también pueden dar funcionalidad al sistema, por lo tanto pueden obtener o ingresar información al sistema. En la tabla 3.2 se muestran los actores principales del sistema, identificados durante esta primera fase.

Tabla 3.2. : Descripción de los actores del Sistema SADET.

ACTOR	DESCRIPCIÓN
Usuario	Analista encargado de llevar el control de los procesos que realiza el sistema de administración.
Administrador del sistema	Representa la persona encargada de la configuración, mantenimiento y actualización del sistema.
BD-SADDET	Representa al sistema encargado de manipular todos los registros que se necesitan en el desarrollo del proyecto. En este se encuentra toda la data utilizada para el control de los procesos, esta información está organizada en tablas.

3.2.4.2. Identificación de los casos de uso

Los casos de uso representan el comportamiento de una parte del sistema desde el punto de vista del usuario. El usuario requiere de varios casos de uso diferentes en el



sistema, por tanto, utilizará de forma diferente a estos casos de uso. Luego para capturar los casos de uso correctos es necesario comprender el contexto del sistema, para así poder aproximarse a las reales necesidades del usuario que le permitan trabajar cómodamente.

A continuación se presenta en la tabla 3.3 descripción de los caso de uso del Sistema de Administración, donde se establecen sus funcionalidades.

Tabla 3.3. : Descripción de los casos de uso del Sistema SADET

CASO DE USO	DESCRIPCIÓN
Entrar al Sistema	Este caso de uso permite a los actores ingresar al sistema verificando su existencia.
Gestionar Nomina	Este caso de uso se encarga de efectuar las operaciones administrativas necesarias para la realización de la nómina; tomando en cuenta el tipo de nómina a la que pertenece cada empleado.
Gestionar Ingresos	Control de todos los ingresos de dinero al capital de la empresa.
Gestionar Egresos	Control de todos los gastos generados por la empresa.
Generar Reportes Impresos	Este caso de uso se refiere a los procesos involucrados en la elaboración y almacenamiento de los reportes impresos.
Configurar Sistema	Procesamiento del mantenimiento y la configuración del sistema.

3.2.4.3. Descripción del Diagrama del caso de uso de Contexto del Sistema Administrativo.

El Diagrama de casos de uso, permite visualizar la relación y el uso que tienen los actores con cada uno de los casos de uso del sistema.



En el siguiente Diagrama (figura 3.3), se muestra un actor llamado BD-SADDET que va a representar la base de datos del sistema e interactúa con todos los casos de uso del diagrama; esto es debido a que es él quien registra toda la información manejada por cada una de las interacciones de los procesos. Del mismo modo, el actor Usuario tiene como función gestionar las operaciones administrativas dentro de la empresa, encargándose de los casos de uso: Gestionar Nómina, Gestionar Ingresos, Gestionar Egresos, Generar Reportes Impresos. Y para finalizar, el actor Desarrollador del Sistema manipula todos los casos de uso y en especial se encarga de Configurar el Sistema.

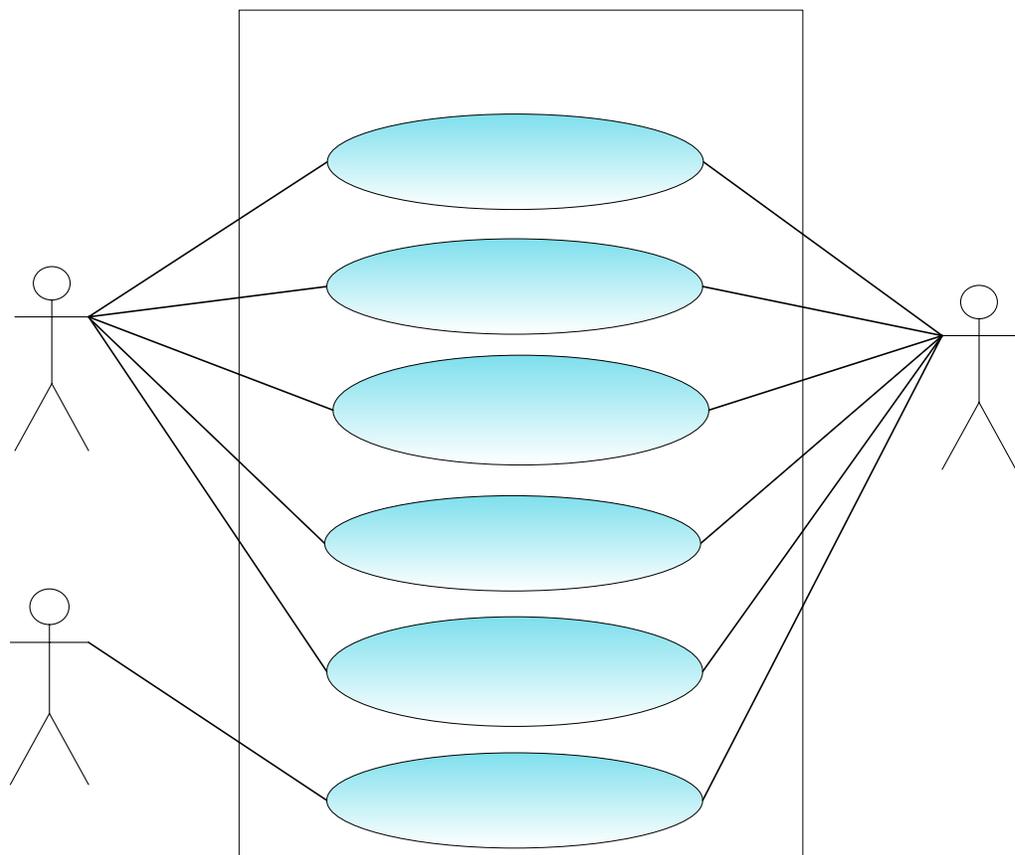


Figura 3.3. : Diagrama de Caso de Uso Contextual



3.2.4.4. Caso de uso Entrar al Sistema.

Tabla 3.4. Caso de Uso: Entrar al Sistema.

Caso de Uso: Entrar al Sistema.

Actores: Usuario y BD-SADDET.

Pre-condición: Ninguna.

Flujo de sucesos:

- 1.- El usuario invoca al caso de uso entrar al sistema.
- 2.- El usuario ingresa su nombre de usuario y su contraseña.
- 3.- El sistema verifica los datos y dependiendo del grupo en el que se encuentre el sistema le proporciona los permisos de acceso.
- 4.- Finaliza el caso de uso.

Caminos alternativos: El usuario puede salir o cancelar el ingreso al sistema.

3.2.4.5. Caso de uso Gestionar Nómina.

Tabla 3.5. Caso de Uso: Gestionar Nómina.

Caso de Uso: Gestionar Nómina.

Actores: Usuario, BD-SADDET.

Pre-condición: El usuario debe haberse identificado en el sistema.

Flujo de sucesos:

- 1.- El usuario invoca cualquiera de los casos de uso siguientes: ingresar empleado semanal, ingresar empleado quincenal, ingresar beneficios, ingresar deducciones, ingresar cargo, ingresar contrato.
- 2.- El sistema carga el formulario de solicitud.
- 3.- El usuario debe ingresar los datos que requiere el formulario.
- 4.- Si el usuario está de acuerdo con los datos ingresados se guardarán en la BD_SADDET.
- 5.- Si el usuario no está de acuerdo se vuelven a ingresar los datos.
- 6.- Finaliza el caso de uso.

Caminos alternativos: El usuario puede seleccionar cualquier ítem del menú principal.



3.2.4.6. Caso de uso Gestionar Ingresos.

Tabla 3.6. Caso de Uso: Gestionar Ingresos.

Caso de Uso: Gestionar Ingresos.

Actores: Usuario, BD-SADDET.

Pre-condición: El usuario debe haberse identificado en el sistema.

Flujo de sucesos:

- 1.- El usuario invoca cualquiera de los casos de uso siguientes: ingresar tipo de ingreso o ingresar emisor.
- 2.- El sistema carga el formulario de solicitud.
- 3.- El usuario debe ingresar los datos que requiere el formulario.
- 4.- Si el usuario está de acuerdo con los datos ingresados se guardarán en la BD_SADDET.
- 5.- Si el usuario no está de acuerdo se vuelven a ingresar los datos.
- 6.- Finaliza el caso de uso.

Caminos alternativos: El usuario puede seleccionar cualquier ítem del menú principal.

3.2.4.7. Caso de uso Gestionar Egresos.

Tabla 3.7. Caso de Uso: Gestionar Egresos.

Caso de Uso: Gestionar Egresos.

Actores: Usuario, BD-SADDET.

Pre-condición: El usuario debe haberse identificado en el sistema.

Flujo de sucesos:

- 1.- usuario invoca cualquiera de los casos de uso siguientes: ingresar tipo de gasto, ingresar detalle de gasto, etc.
- 2.- El sistema carga el formulario de solicitud.
- 3.- El usuario debe ingresar los datos que requiere el formulario.
- 4.- Si el usuario está de acuerdo con los datos ingresados se guardarán en la BD_SADDET.
- 5.- Si el usuario no está de acuerdo se vuelven a ingresar los datos.

Caminos alternativos: El usuario puede seleccionar cualquier ítem del menú principal.



3.2.4.8. Caso de uso Generar Reportes Impresos.

Tabla 3.8. Caso de Uso: Reportes Impresos.

Caso de Uso: Generar Reportes Impresos.

Actores: Usuario.

Pre-condición: El usuario debe haberse identificado en el sistema.

Flujo de sucesos:

- 1.- El usuario invoca el caso de uso generar reportes.
- 2.- El sistema genera el reporte.
- 3.- El usuario a través de la para visualizar selecciona la opción de guardar o de imprimir.
- 4.- Se guarda o se imprime el documento.
- 5.- Finaliza el caso de uso.

Caminos alternativos: El usuario puede seleccionar cualquier ítem del menú principal.

3.2.4.9. Caso de uso Configurar Sistema.

Tabla 3.9. Caso de Uso: Configurar Sistema.

Caso de Uso: Configurar Sistema.

Actores: Administrador del Sistema y BD-SADET.

Pre-condición: El usuario debe haberse identificado en el sistema.

Flujo de sucesos:

- 1.- El administrador del sistema invoca el caso de uso: CONFIGURAR EL SISTEMA.
- 2.- El administrador del sistema selecciona el recurso a manipular, entre ellos se encuentran: los empleados, el tipo de nómina, el salario, los beneficios, las deducciones, los tipos de gastos, el emisor, los contratos, el cargo, etc.
- 3.- El sistema muestra la interfaz correspondiente, y por medio de ella el administrador del sistema ingresará nuevos datos, modificará o eliminará información existente.
- 4.- El sistema actualiza los cambios.
- 5.- Finaliza el caso de uso.

Caminos alternativos: El usuario puede seleccionar cualquier ítem del menú principal.



3.2.5. Requisitos No Funcionales

Son requisitos adicionales, los cuales no pueden asociarse a ningún caso de uso específico o determinado, sin embargo cada requisito de estos tiene impacto sobre varios casos de uso o en ninguno. Estos requisitos responden esencialmente a limitaciones arquitectónicas y de rendimiento, además especifican propiedades del sistema, como restricciones de implementación, mantenibilidad, extensibilidad o fiabilidad. A continuación se presentan los requisitos no funcionales que complementan el entorno, contexto y adaptabilidad del sistema:

- 👍 El diseño de la interfaz del usuario debe poseer una estructura amigable, para que resulte de fácil entendimiento para los usuarios.
- 👍 El sistema debe ser codificado en PHP, que soporte los parámetros de un software con plataforma .Net y un manejador de base de datos MySQL.
- 👍 El software diseñado debe permitir el mantenimiento, sin ninguna complicación y la incorporación de nuevas funcionalidades en su estructura.
- 👍 Para la manipulación de los datos se deben realizar restricciones hacia los diferentes usuarios del sistema.

3.3. ANÁLISIS

En el flujo de análisis se analizan y refinan los requisitos del sistema, utilizando un lenguaje más formal, a diferencia del empleado en el caso de uso. Este lenguaje se basa en un modelo de objeto conceptual, llamado *modelo de análisis*, donde además se deducen los aspectos internos del sistema y se realiza una estructura de los requisitos manipulados. Así dentro del modelo de análisis, los casos de uso se describen mediante clases de análisis y sus objetos.



3.3.1. Diagrama de Clases de Análisis

Las clases de análisis permiten identificar las acciones de cada caso de uso, además de indicar acciones o parte de acciones comunes o compartidas por varios casos de uso. La clase de análisis se centra en el tratamiento de los requisitos funcionales, esto hasta llegar a las actividades de diseño e implementación de fases subsiguientes a medida que se analizan más y más los casos de uso.

El diagrama de Clases de Análisis describe la forma de llevarse a cabo los casos de uso de este sistema; donde el usuario a través de la clase de interfaz puede comenzar el proceso de interacción, continua el trabajo con una clase de control que coordina la forma de llevar a cabo el caso de uso en cuestión y ordena a la clase de entidad actualizar los datos.

Para realizar el modelo de Análisis, se emplean una serie de estereotipos, los cuales contribuyen a la creación de un modelo de objeto y una arquitectura robusta.

Existen tres estereotipos estándares en UML, los cuales son utilizados por los desarrolladores para distinguir el ámbito de las diferentes clases. Cada estereotipo tiene un símbolo que lo representa, como se muestra en la figura 3.4.



Figura 3.4 Estereotipos de las clases de análisis

3.3.1.1. Clases de Interfaz:

Estas se utilizan para modelar interacciones entre el sistema y sus actores, es decir, usuarios y sistemas externos. Esta interacción a menudo implica recibir información y



peticiones de los usuarios y los sistemas externos. Las clases de interfaz modelan las partes del sistema que dependen de sus actores, lo cual implica que clasifican y reúnen los límites del sistema. Por tanto, un cambio en una interfaz de usuario o una interfaz de comunicación queda normalmente aislado en una o más clases de interfaz. Las clases de interfaz representan a menudo abstracciones de ventanas, formularios, paneles, interfaces de comunicación, interfaces de impresoras, sensores y terminales. En la tabla 3.10 se muestran las clases de interfaz del sistema.

Tabla 3.10 Clases de Interfaz del Sistema SADET

Clases	Definición
IU Entrar al Sistema	Esta clase permite a los actores acceder al sistema.
IU Gestionar Nómina	Esta clase permite al usuario interactuar con el sistema, ingresando la información relacionada con la nómina: ingresar empleados, ingresar beneficios, ingresar deducciones, etc.
IU Gestionar Ingresos	Esta clase permite al usuario interactuar con el sistema, ingresando la información relacionada con los ingresos al capital de la empresa.
IU Gestionar Egresos	Esta clase permite al usuario interactuar con el sistema, ingresando la información relacionada con los gastos efectuados por la empresa.
IU Generar Reportes Impresos	Esta clase permite a los usuarios y al administrador interactuar con el sistema, generando un reporte digital o impreso
IU Configurar el Sistema	Permite al usuario realizar las respectivas configuraciones del sistema como son: creación, actualización, modificación y eliminación de la información.



3.3.1.2. Clases de Control:

Representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Los aspectos dinámicos del sistema se modelan con las clases de control, debido a que ellas manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos, es decir, objetos de interfaz y de entidad. En la tabla 3.11 se muestran las clases de control del sistema.

Tabla 3.11 Clases de Control del Sistema SADET

Clases	Definición
Gestor Entrar al Sistema	Esta clase permite comprobar que el usuario esta activo y tiene acceso al sistema.
Gestor Gestionar Nómina	Coordina el proceso que gestiona las operaciones relacionadas con la nómina de empleados.
Gestor Gestionar Ingresos	Coordina el proceso que gestiona las operaciones relacionadas con los ingresos al capital de la empresa.
Gestor Gestionar Egresos	Coordina el proceso que gestiona las operaciones relacionadas con los gastos efectuados por la empresa.
Gestor Generar Reportes Impresos	Coordina el proceso que permite elaborar y almacenar el recibo que es entregado a los empleados por el pago de: la nómina, las prestaciones, la relación Ingresos / Egresos.
Gestor Configurar el Sistema	Es el encargado de controlar todas las operaciones de configuración del sistema, así como también la administración de los diferentes recursos que éste posee.



3.3.1.3. Clases de Entidad:

Representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Los aspectos dinámicos del sistema se modelan con las clases de control, debido a que ellas manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos, es decir, objetos de interfaz y de entidad. En la tabla 3.12 se muestran las clases de entidad del sistema.

Tabla 3.12 Clases de Entidad del Sistema SADET

Clases	Definición
Acceso al sistema	Esta clase permite generar el acceso correspondiente a cada tipo de usuario.
Empleado	Representa a los distintos trabajadores que laboran en la empresa.
Nómina	Representa el tipo de nómina a la que pertenecen los empleados.
Reportes	Representa el informe sobre todas las operaciones administrativas que se realizan en la empresa.
Ingresos	Representa los tipos de ingresos que entran a la empresa.
Egresos	Representa los tipos de gastos que entran a la empresa.
Beneficios	Representa los beneficios que reciben los empleados que laboran en la empresa.
Deducciones	Representa las deducciones que se descuentan a los empleados de la empresa.
Control	Esta clase permite la eliminación de los registros de las tablas de la base de datos.



3.3.2. Diagramas de Clases de Análisis para los Casos de Uso del Sistema

A continuación se describen los diagramas de clases de análisis para cada una de los casos de uso del sistema:

3.3.2.1. Diagrama de Clases de Análisis: Entrar al Sistema

Se identificó la clase de interfaz Entrar al Sistema, el usuario interactúa con esta clase para solicitar el acceso al sistema mediante su usuario que es el identificador y su contraseña, todos los usuarios tienen acceso a esta clase. Mediante la clase de control Gestor Entrar al Sistema, el actor podrá entrar a la página principal obteniendo así acceso al sistema con sus respectivos permisos (ver figura 3.5).

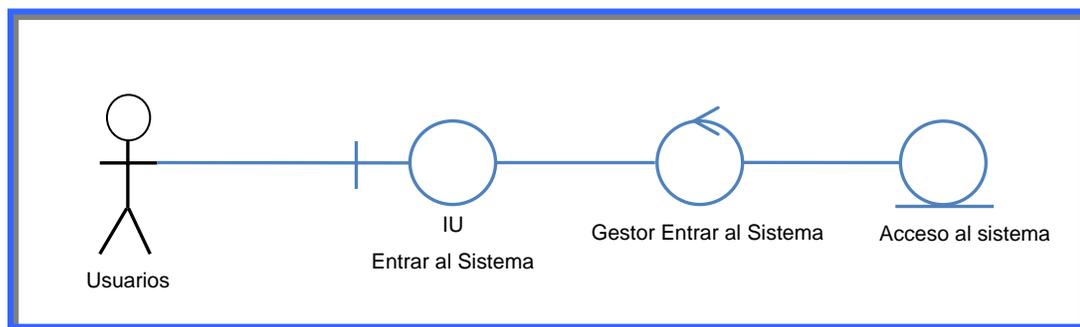


Figura 3.5 Diagrama de Clases de Análisis: Entrar al Sistema

3.3.2.2. Diagrama de Clases de Análisis: Gestionar Nómina

Se identificó la clase de interfaz Gestionar Nómina: ingresar empleados, ingresar beneficios, ingresar deducciones, etc., el usuario interactúa con esta clase cargando la información relacionada con la nómina. Toda la data ingresada es verificada por la clase de control Gestor Gestionar Nómina, para corregir de esta manera cualquier error por parte de los usuarios y mantener el formato de los datos, finalmente los datos son guardados en la BD-SADDET esto gracias a la clase de entidad Nómina (ver figura 3.6).

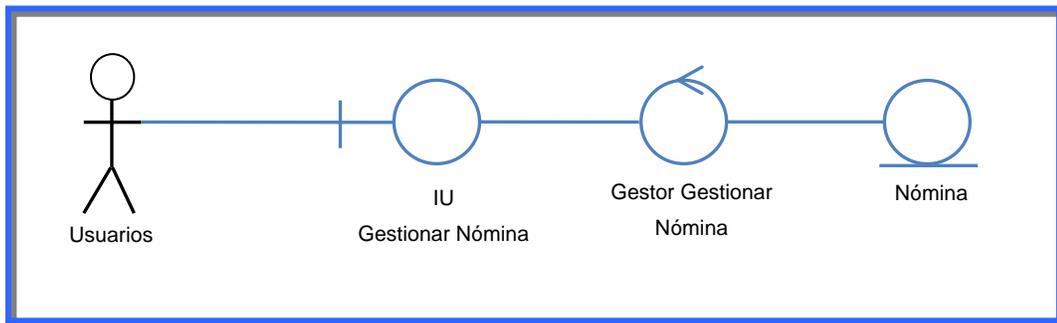


Figura 3.6 Diagrama de Clases de Análisis: Gestionar Nómina

3.3.2.3. Diagrama de Clases de Análisis: Gestionar Ingresos

Se identificó la clase de interfaz Gestionar Ingresos: ingresar tipo de ingreso, ingresar emisor, etc., el usuario interactúa con esta clase cargando la información relacionada con los ingresos. Toda la data ingresada es verificada por la clase de control Gestor Gestionar Ingresos, para corregir de esta manera cualquier error por parte de los usuarios y mantener el formato de los datos, finalmente los datos son guardados en la BD-SADDET esto gracias a la clase de entidad Ingresos (ver figura 3.7).

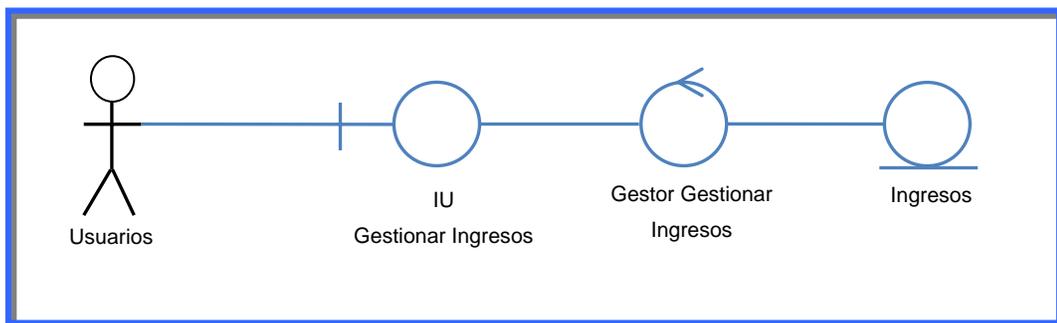


Figura 3.7 Diagrama de Clases de Análisis: Gestionar Ingresos

3.3.2.4. Diagrama de Clases de Análisis: Gestionar Egresos

Se identificó la clase de interfaz Gestionar Egresos: ingresar tipo de gasto, ingresar detalle gasto, etc., el usuario interactúa con esta clase cargando la información relacionada con los egresos. Toda la data ingresada es verificada por la clase de



control Gestor Gestionar Egresos, para corregir de esta manera cualquier error por parte de los usuarios y mantener el formato de los datos, finalmente los datos son guardados en la BD-SADET esto gracias a la clase de entidad Egresos (ver figura 3.8).

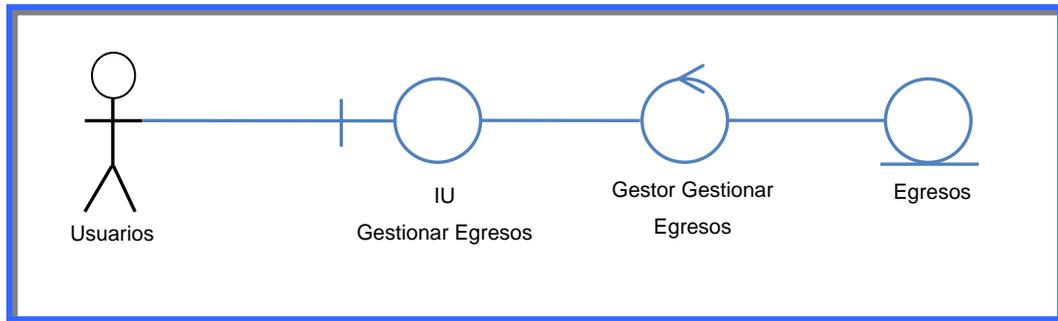


Figura 3.8 Diagrama de Clases de Análisis: Gestionar Egresos

3.3.2.5. Diagrama de Clases de Análisis: Generar Reportes Impresos

Se identifico la clase de interfaz Reportes Impresos, todos los actores tienen acceso a esta clase. Mediante la clase de control Gestor Reportes Impresos permite generar un reporte en formato digital o impreso a través de la clase de entidad Reportes, (ver figura 3.9).

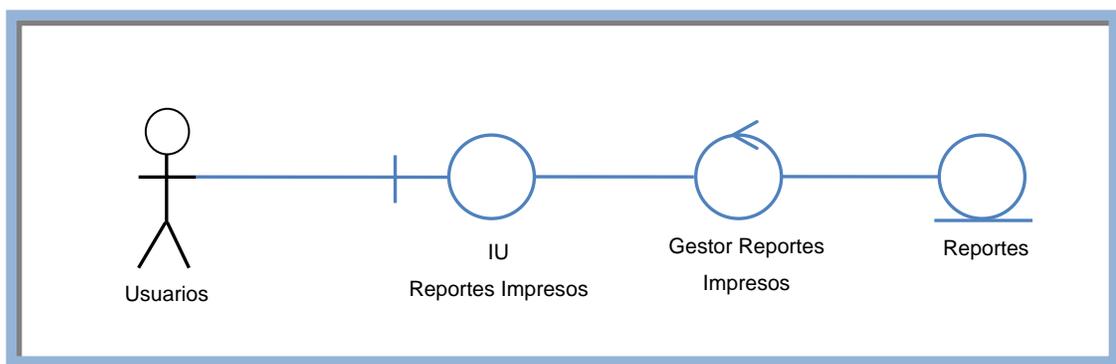


Figura 3.9 Diagrama de Clases de Análisis: Generar Reportes Impresos



3.3.2.6. Diagrama de Clases de Análisis: Configurar Sistema

Se identifico la clase de interfaz Configurar Sistema, sólo el administrador del sistema tiene acceso a esta clase, la cual permite eliminar la información de las tablas de la base de datos realizando antes una copia de seguridad de la misma. Tenemos la clase de Control que se encarga de generar los permisos a los distintos tipos de usuarios, (ver figura 3.10).

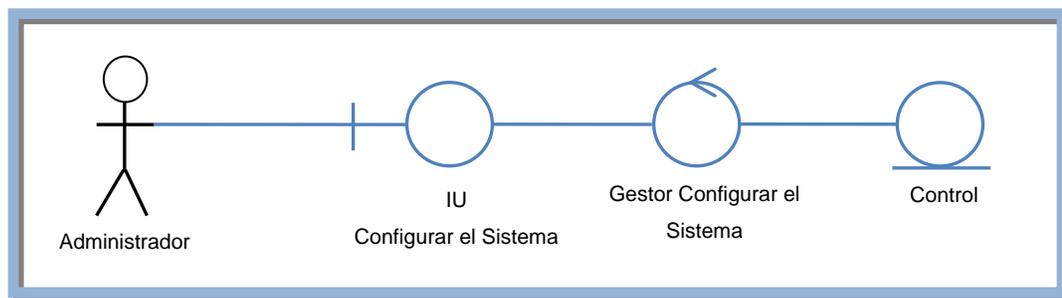


Figura 3.10 Diagrama de Clases de Análisis: Configurar Sistema

3.4. DIAGRAMA DE PAQUETE DE ANÁLISIS

El paquete de análisis representa un medio de organizar el modelo de análisis en piezas más pequeñas, basándose en los requisitos funcionales y en el dominio del problema.

Un paquete define un espacio de nombre ver figura. 3.11, de modo que dos elementos diferentes, contenidos en dos paquetes diferentes, pueden tener el mismo nombre.

Un paquete puede contener otros paquetes, sin límite de nivel de anidamiento. Un nivel dado puede contener una mezcla de paquetes y de otros elementos de modelado, de la misma manera que un directorio puede contener directorios y archivos.



Un paquete puede constar de clases de análisis, de realizaciones de casos de uso, y de otros paquetes del análisis. Los paquetes de análisis poseen una traza con subsistemas en el diseño, donde una traza es una correspondencia entre un paquete del análisis y los subsistemas de diseño correspondientes.

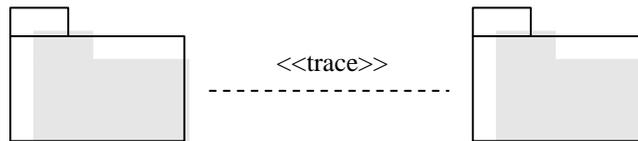


Figura 3.11 Representación de Paquete

El diagrama de paquete de análisis general del sistema SADET, ver figura 3.18, muestra los paquetes que encapsulan los diferentes casos de usos que fueron definidos al realizar el análisis del sistema, donde se puede observar el paquete general de nombre SADET, y en su interior se ubican los subsistemas principales como lo son: Entrar al Sistema, Gestionar Nómina, Gestionar Ingresos, Gestionar Egresos, Generar Reportes Impresos y Configurar Sistema; estos paquetes se asocian a sus funcionalidades mediante *trace* que parten de un paquete a otro paquete. Esto con la finalidad de esbozar dichos casos de usos en piezas más pequeñas que puedan ser fáciles de manejar.

Paquete

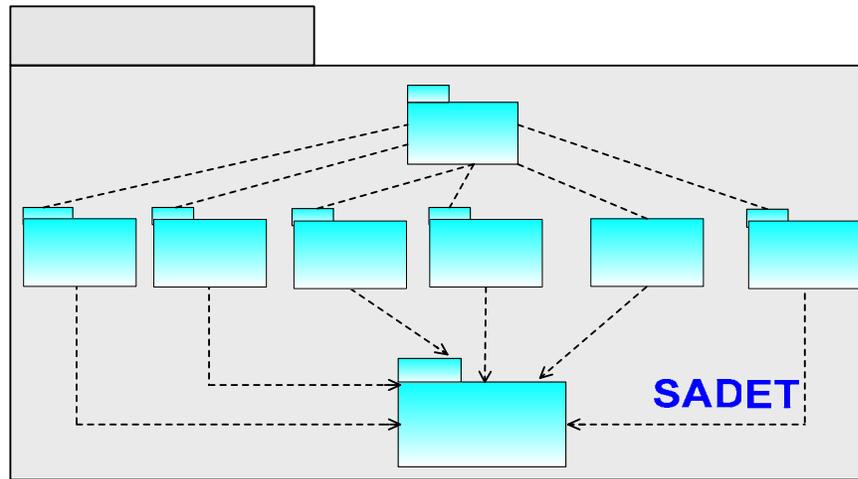


Figura 3.12 Diagrama de Paquete de Análisis de SADET

3.5. DISEÑO

Este flujo de trabajo permitirá el diseño de un esquema general de la arquitectura candidata, en el cual el sistema que se está desarrollando será distribuido sobre varios puntos de conexión (nodos), los cuales son elementos de hardware y sobre ellos pueden ejecutarse los elementos de software (modelo de despliegue), utilizando para esto la arquitectura cliente/ servidor, que se refiere a la existencia de dos niveles, uno donde se encuentran los servidores del sistema (los que contienen las bases de datos) y el otro son los niveles para los usuarios, encontrándose ambos conectados entre sí.

Entrar a Sistema

Gestionar Nómina

Gestionar Ingresos

3.5.1. Diseño de la Arquitectura

El diseño de la arquitectura en esta etapa proporcionará una estructura general del funcionamiento del sistema.



3.4.1.1. Identificación de Subsistemas

Para ejecutar el sistema se utilizará la plataforma Microsoft Windows, ya que ésta proporciona las herramientas necesarias para que el proyecto se lleve a cabo de una manera sencilla y segura, logrando así la eficiencia del mismo. De igual forma el sistema manejador de bases de datos MYSQL será empleado para administrar la base de datos del sistema, y también se realizará la respectiva conexión de dicha base de datos con la aplicación a través de la tecnología ADO.

3.6. CONCLUSIÓN DE LA FASE DE INICIO

Los objetivos principales de la fase de inicio son: establecer un análisis del sistema propuesto, describir el contexto del sistema, capturar los requerimientos funcionales, identificar los riesgos críticos que pondrían en peligro el desarrollo del proyecto y proponer una arquitectura candidata factible. Dichos objetivos se cumplieron de una forma satisfactoria, ya que se obtuvo una primera versión del modelo del negocio que describe el contexto del sistema, una lista inicial de riesgos y un esbozo de los modelos que representan una primera versión del modelo de casos de uso y el modelo de análisis, los cuales describen una arquitectura candidata factible. En esta fase se obtuvo una buena comprensión del proyecto y la factibilidad de culminarlo. Los resultados alcanzados en esta fase se refinarán en la fase de elaboración.

CAPÍTULO 4: FASE DE ELABORACIÓN

4.1. FASE DE ELABORACIÓN

La etapa de elaboración es muy importante en el desarrollo de este proyecto, ya que nos permitirá establecer la Arquitectura sólida del software, la cual guiará el trabajo durante la fase de construcción y transición, así como también en las posteriores generaciones del proyecto.

El objetivo principal en esta fase, es formular la *línea base de la arquitectura*, para así disponer de unos conocimientos sólidos cumpliendo con los flujos de trabajo de requisito, análisis, diseño e implementación, tal y como se muestran en la **figura 4.1**.

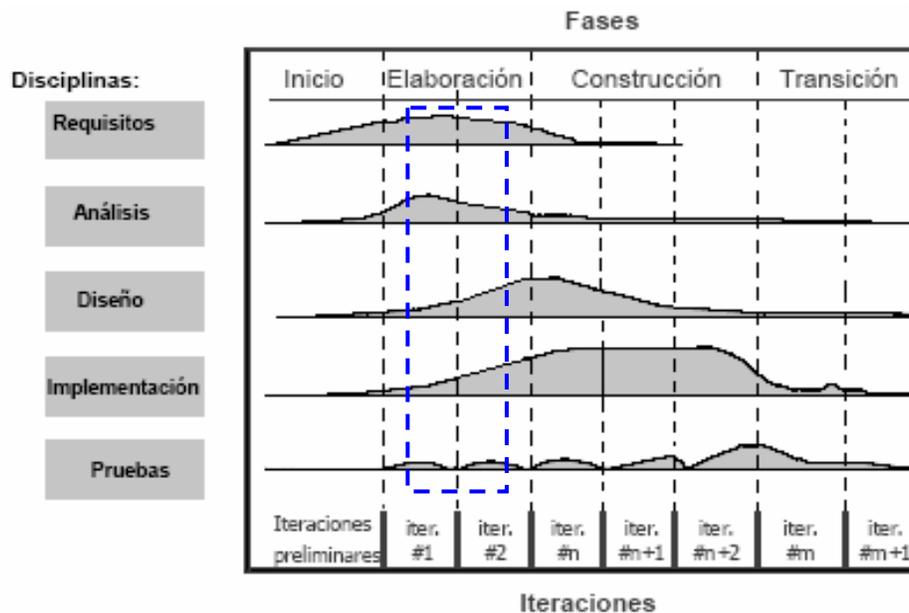


Figura 4.1. : Diagrama de los flujos de trabajo fundamental en la fase de elaboración

El fundamento de la fase de elaboración es desarrollar el flujo de trabajo del diseño, sustentado en la captura de requisitos, en el análisis y diseño preliminar; se evalúan y depuran los requisitos de la fase anterior considerando la creación de nuevos casos de uso, los cuales serán sujetos al análisis y al diseño.



Se considera desarrollar el flujo de trabajo de implementación y prueba en la fase de Construcción, ya que los mismos no tienen relevancia con el objetivo principal de la etapa de elaboración.

En esta etapa se estima la creación de la interfaz de usuario y la elaboración de un buen diseño de las bases de datos, ya que el proyecto manipula bases de datos robustas que manejan gran cantidad de información.

4.2. CAPTURA DE REQUISITOS

Para la fase de elaboración en la recolección de requisitos, se identifican casos de uso y actores adicionales a los identificados en la fase de inicio. Se describen solamente los nuevos casos de uso, para luego analizar sólo parte de aquellos que se describieron.

4.2.1. Diagrama de Casos de Uso

En el primer flujo de trabajo, se captura la mayoría de los requisitos, realizando una descripción detallada de los diagramas de casos de uso señalados en la fase de inicio.

A continuación se describen los nuevos diagramas de casos de uso, vinculados con el proyecto de automatización del flujo de información asociado a la administración de una empresa (*SADET*); todo esto se realiza con la finalidad de especificar en profundidad el comportamiento de cada uno de los procesos relacionados con este sistema.

El diagrama de caso de uso realizado en la fase anterior; se ha ampliado con la adición de nuevos casos de uso, como se puede observar en la figura 4.2.

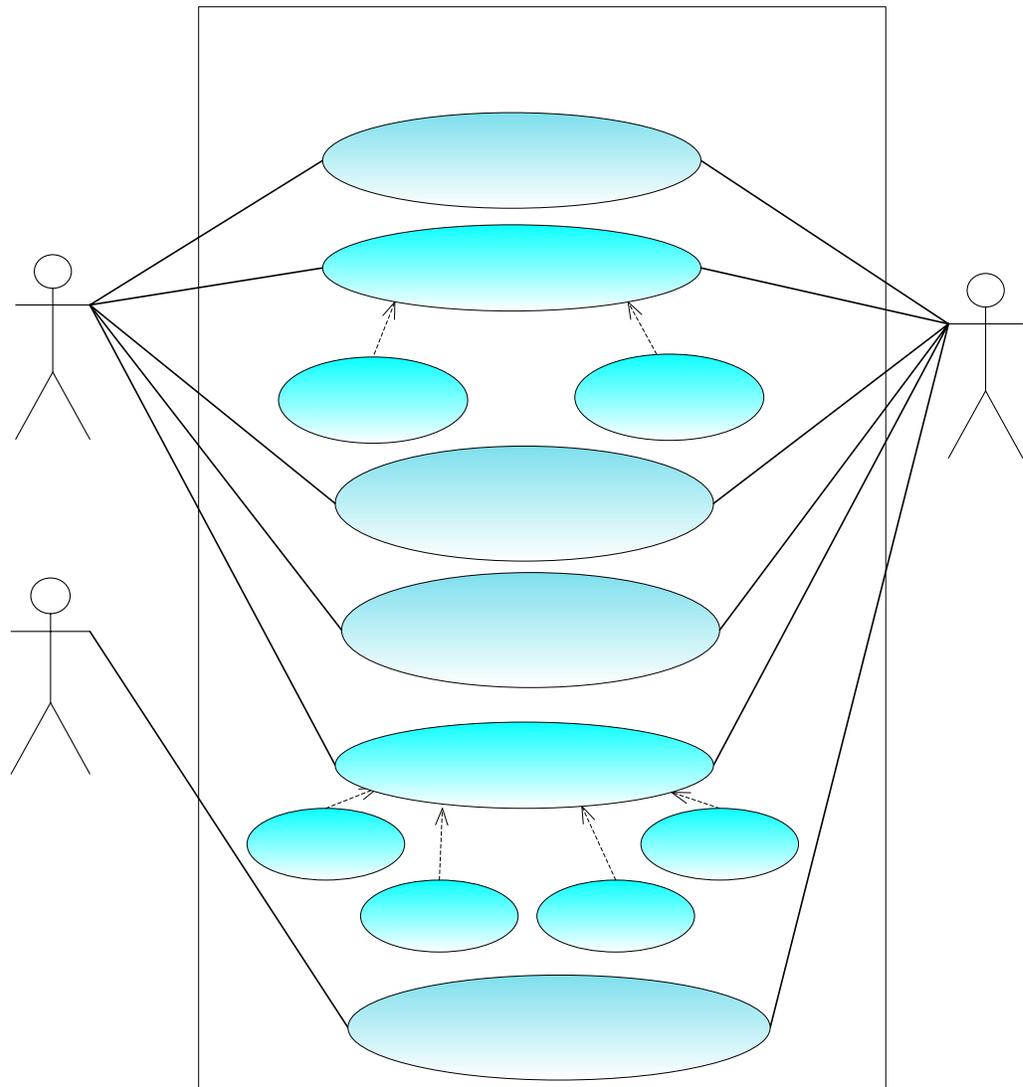


Figura 4.2. : Diagrama de Caso de Uso Extendido

Usuario

Como se puede apreciar en la figura 4.2., los actores mantienen la misma relación con los procesos vinculados al sistema, con la diferencia que ahora se adicionan dos modelos de caso de uso ligeramente modificados, porque el inicial no cubría todas las expectativas.

Generar N
Quince



4.2.1.1. Descripción del caso de uso extendido de Generar Nómina

Para el diagrama que se muestra en la figura 4.3, se representan los casos de uso extendidos de Gestionar Nómina, con el objeto de poder apreciar los cambios existentes relacionados en este capítulo. Los casos de uso extendido Gestionar Nómina Semanal Y Quincenal permitirán llevar el control de las distintas tipos de nóminas existentes en la empresa.

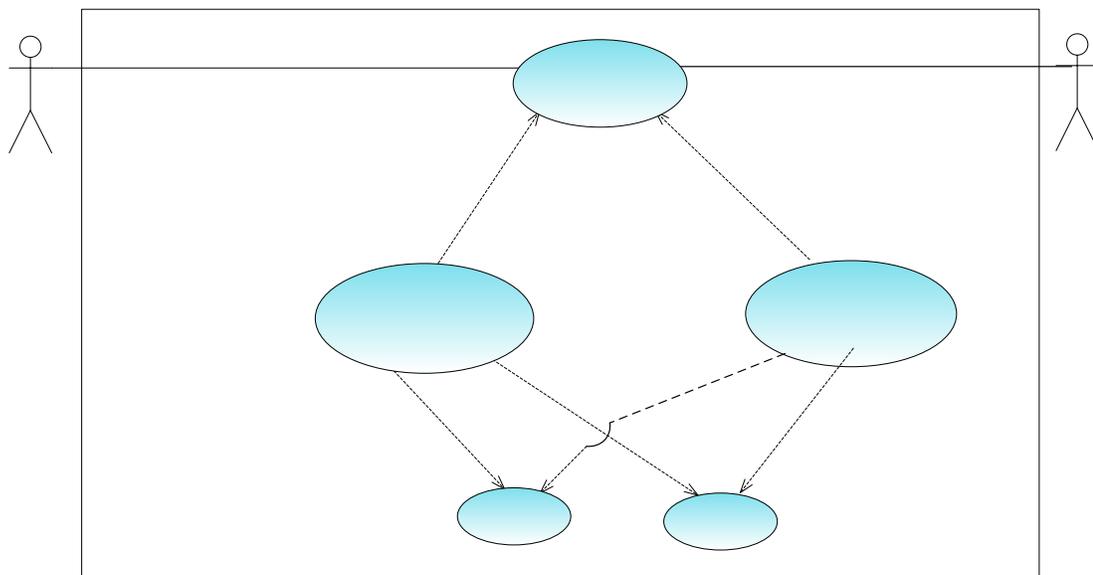


Figura 4.3. : Diagrama de Caso de Uso Detallado Gestionar Nómina

4.2.1.2. Descripción del caso de uso extendido de Generar Reportes Impresos

Para el diagrama que se muestra en la figura 4.4, se representan los casos de uso extendidos de Generar Reportes Impresos, en el cual se detallan los diferentes reportes que serán generados por el sistema SADET.

Usuario

<<extenc:

GESTIONAR
NÓMINA
SEMANAL

<<

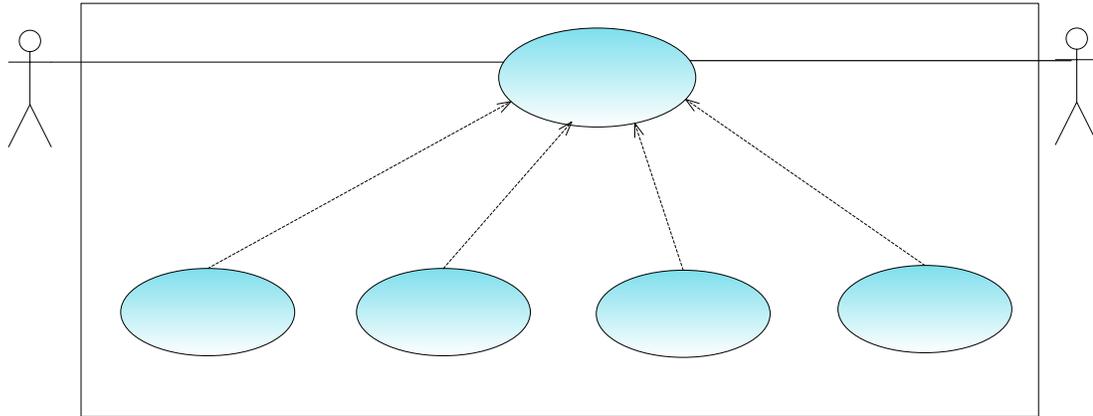


Figura 4.4. : Diagrama de Caso de Uso Detallado Generar Reportes Impresos

<<extenc>>

4.3. ANÁLISIS Usuario

El desarrollo de este flujo de trabajo para la fase de elaboración, comienza con la realización el modelo de análisis, sobre lo esbozado en la fase de inicio. Se analizarán los casos de uso que son significante desde el punto de vista de la arquitectura del sistema.

RECIBO
SEMANTAL

RECIBO
QUINCENAL

4.3.1. Análisis de los casos de uso

A continuación se muestran los diagramas de clase de análisis relacionados con los casos de uso recientemente elaborados:

4.3.1.1. Diagrama de clases de análisis de la realización del caso de uso: Gestionar Nómina

Estas clases de análisis agrupan las principales actividades relacionadas con la realización del ingreso de los empleados, el cargo, los beneficios y las deducciones como se muestra en la figura 4.4. El usuario como actor principal, puede activar el



caso de uso Gestionar Nómina, la nueva clase relacionada con los tipos de nómina de la siguiente manera:

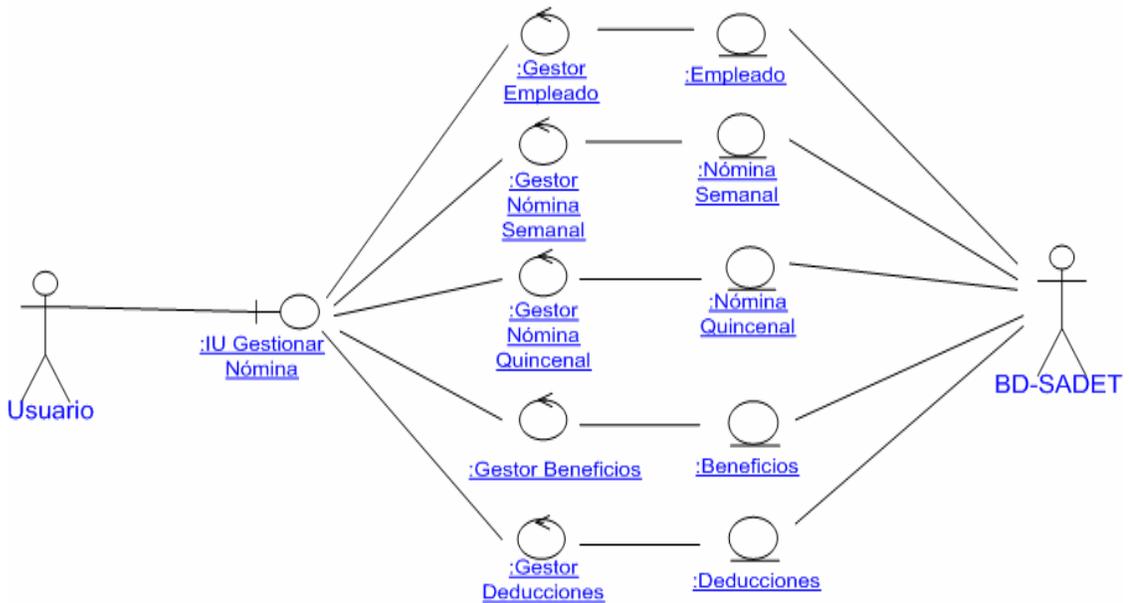


Figura 4.5. : Diagrama de Clase de Análisis del Caso de Uso Detallado Gestionar Nómina

4.3.1.2. Diagrama de clases de análisis de la realización del caso de uso: Generar Reportes Impresos

En la creación del caso de uso Gestionar Reportes se realizó el diagrama de clase de análisis como se muestra en la figura 4.5, el cual se encarga de habilitar los gestores de los reportes del sistema, donde se realizan diferentes listados dependiendo de la solicitud que realice el respectivo actor del caso de uso correspondiente.

El actor Usuario emplea una forma para activar el caso de uso Reportes Impresos, como se puede describir a continuación:

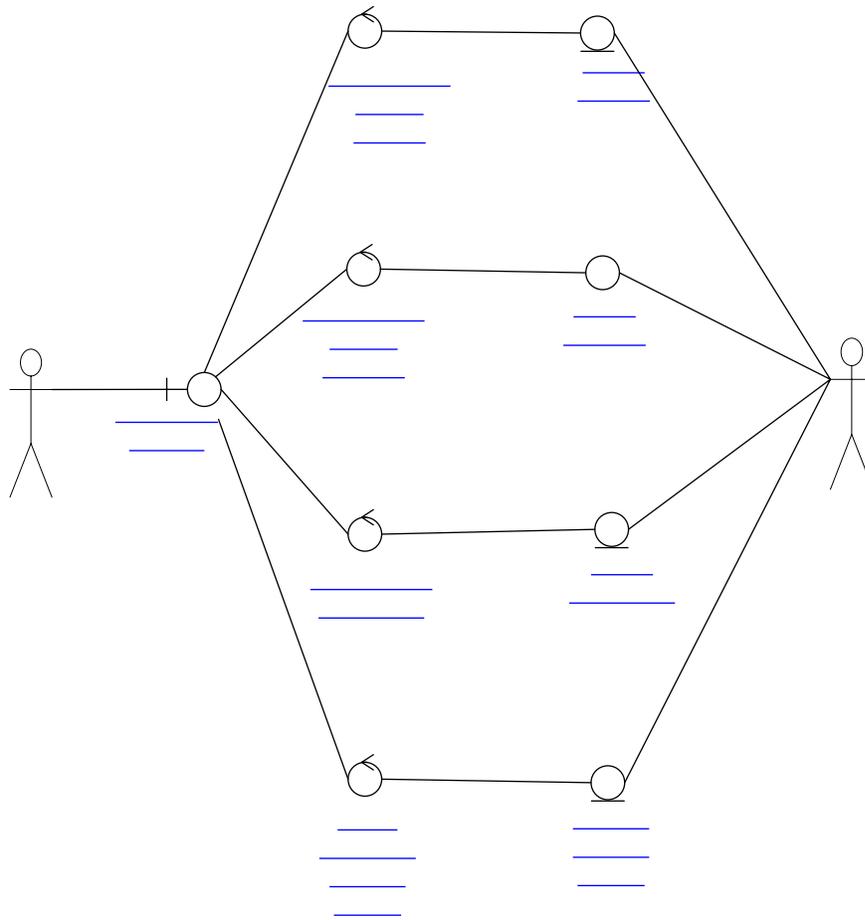


Figura 4.6. : Diagrama de Clase de Análisis del Caso de Uso Detallado Generar Reportes Impresos

4.3.2. Interacción de las clases de análisis

En este punto tenemos un esbozo de las clases necesarias para realizar el caso de uso, ahora se debe describir la interacción de sus correspondientes objetos del análisis. Esto se realiza mediante diagramas de colaboración que contienen las instancias de actores participantes, los objetos del análisis y sus enlaces.

Los diagramas de colaboración expuestos a continuación, representan instancias de las clases definidas en el segmento de análisis. Estos diagramas también

:Gestor Re
de Pag
Seman

:Gestor Recib
de Pago
Quincenal

:IU Reportes
Impresos

Usuario

:Gestor Recib
Prestacione



muestran las iteraciones que existen entre los objetos del análisis a través de los mensajes, estos mensajes pueden provenir de los actores o entre objetos.

4.3.2.1. Diagrama de colaboración de la realización del caso de uso: Gestionar Nómina

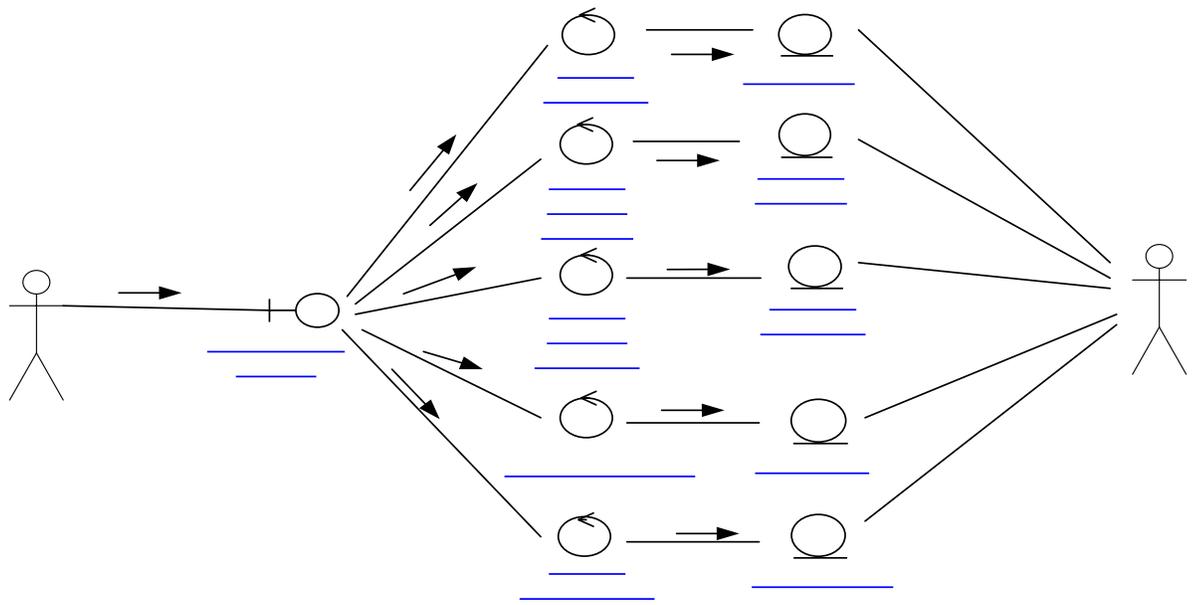


Figura 4.7. : Diagrama de Colaboración del Caso de Uso Detallado Gestionar Nómina

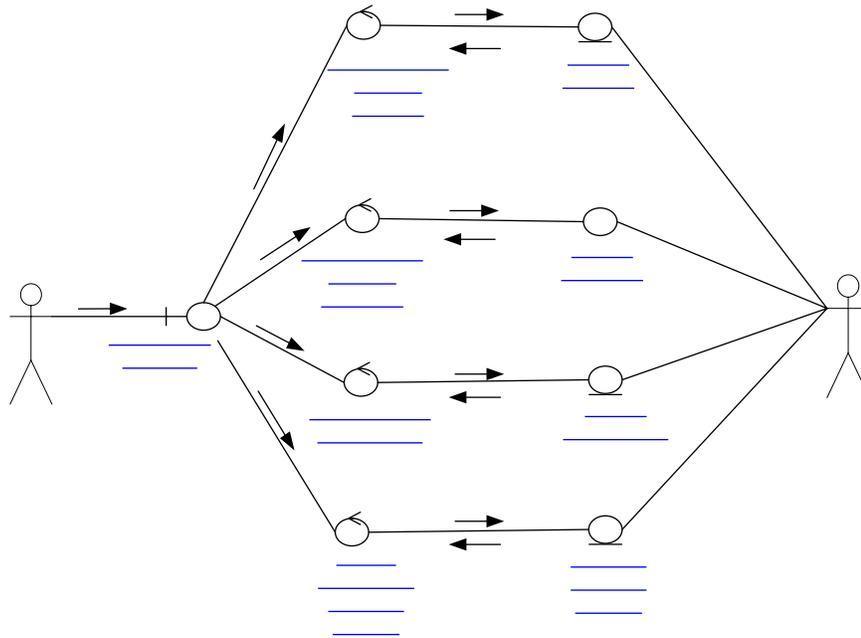
2:

1:

:IU Gestionar
Nómina



4.3.2.2. Diagrama de colaboración de la realización del caso de uso: Gestionar Reportes Impresos



3:
:Gestor Recibo de Pago Semanal

2:

Figura 4.8. : Diagrama de Colaboración del Caso de Uso Detallado Generar Reportes Impresos

4.4. DISEÑO

1:

:Gestor Recibo de Pago Quincenal

El papel de este flujo de trabajo en la fase de elaboración contribuye a la creación de una arquitectura estable y sólida, gracias a la realización del diseño e implementación de los casos de uso estructuralmente más significativos del sistema.

:IU Reportes Impresos

Usuario

8:
9:
11: :Gestor Recibo Prestaciones

Se aplicará el modelo de diseño para describir la realización física de los casos de uso desarrollados en la sección previa a esta fase.

:Gestor Relación de Ingresos-



4.4.1. Diagrama de clases de diseño

El diagrama de clases de diseño sirve para visualizar las clases que componen el sistema y cómo se relacionan entre sí. Permite mostrar lo que el sistema puede hacer, además de cómo puede ser construido.

Para la realización del diagrama de clases de diseño, se tomaron como base las clases de análisis, además de agregar nuevas clases que no se habían considerado. Las clases de diseño se construyen de manera similar a las usadas en la implementación del software y el lenguaje empleado para su representación será PHP, por ser el lenguaje con el cual se codificará el sistema.

La interacciones de las clases de diseño se lleva a cabo mediante la aplicación del diagrama de clases y de interacción para expresar la comunicaciones entre objeto del diseño. Cuando se tiene el esquema de la relación de las clases se procede a elaborar los diagramas de secuencia los cuales contienen las instancias de los actores, los objetos del diseño y las transmisiones de mensajes entre ellos.

El diagrama de clase de diseño mostrado a continuación contienen clases relacionadas entre si por medio de líneas. Estas relaciones pueden ser de agregación, compuestas o simples en estos casos se mostrará la relación simple y por Agregación. Además contienen también la información de multiplicidad que precisa el número de instancias que participan en la relación, la cual en su mayoría es de uno a uno (1 a 1).

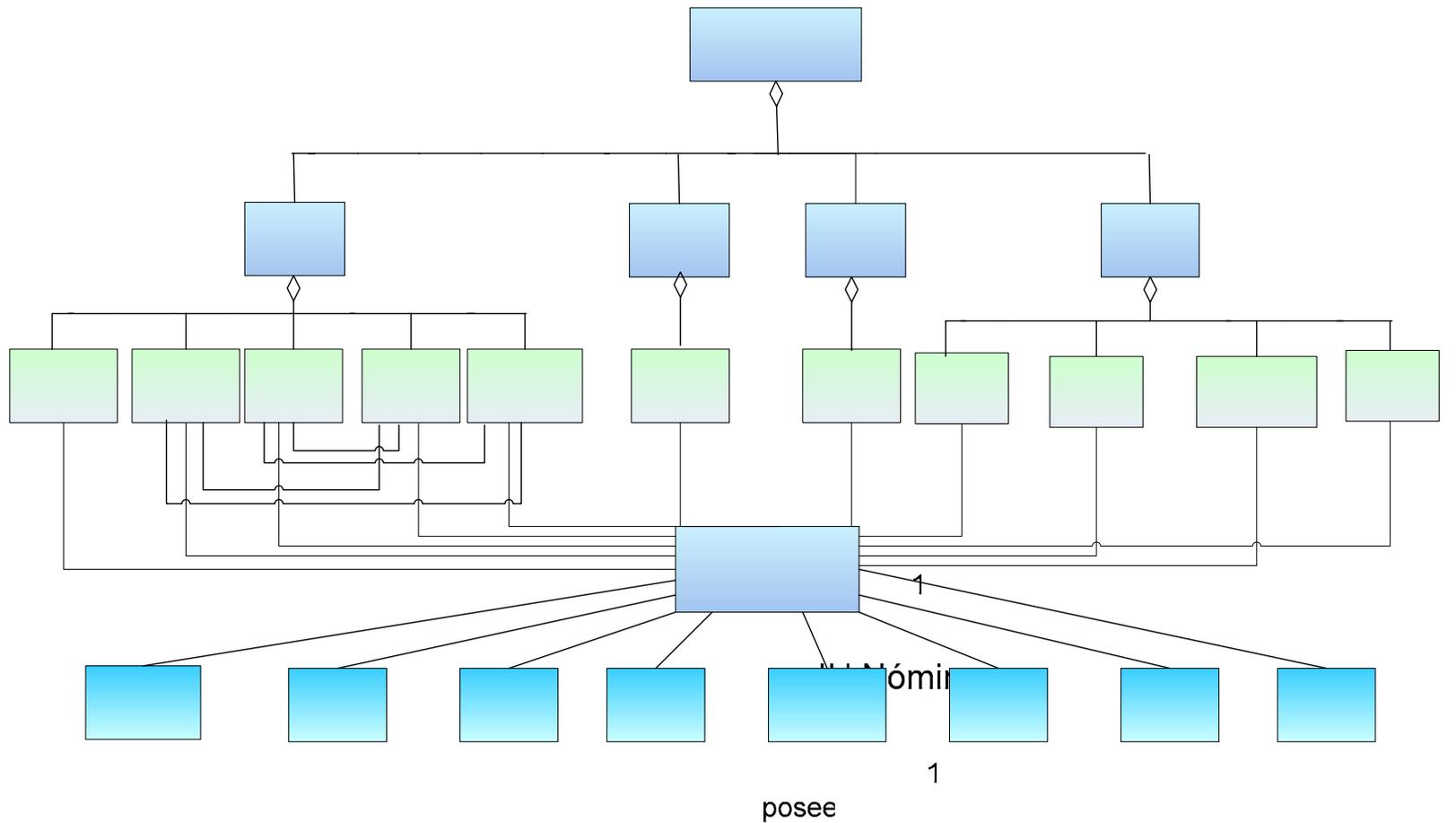


Figura 4.9. : Diagrama de Clase de Diseño del Sistema SADET

Gestionar
Empleado

Gestionar
Nómina
Quincenal

Gestionar
Nómina
Semanal

Gestionar
Beneficios

Gestionar
Deducciones

1 1
utiliza



El diagrama de clase de diseño de la figura 4.9, expone la interacción de las clases que realizan al sistema. La aplicación está conformada conceptualmente por clases activas encargadas de controlar las tareas específicas de los procesos. Estas clases deberán solicitarle a la base de datos procesar las operaciones necesarias para suministrar y almacenar la información correspondiente a través de la interfaz de comunicación IS. Manejador Base de Datos.

4.4.2. Diagrama de Secuencia

Los diagramas de Clase de diseño se complementan con los diagramas de secuencia, en el cual se aprecia las interacciones de los actores y la transmisión de mensajes entre las clases vinculadas a este proceso.

Para crear un diagrama de secuencia se debe comenzar por el principio del flujo del caso de uso, y después seguir este flujo paso a paso, decidiendo qué objetos de diseño y qué interacciones de instancias de actores son necesarias para cada paso. Es importante comprender el orden cronológico de las transferencias de mensajes entre objetos.

Los diagramas de secuencia son diagramas dinámicos que representan de forma jerárquica el orden en el cual se activan los objetos mediante los mensajes, a diferencia de los diagramas de colaboración que representan el orden de activación de los objetos pero a través de números, también se puede observar en estos diagramas el flujo de control del regreso cuando se activa un objeto mediante la instancia de una clase, así como también la destrucción de los objetos cuando no son utilizados. Además es importante mencionar que estos diagramas permiten representar los periodos de actividad de los objetos. Un periodo de actividad corresponde al tiempo durante el cual un objeto ejecuta una acción.



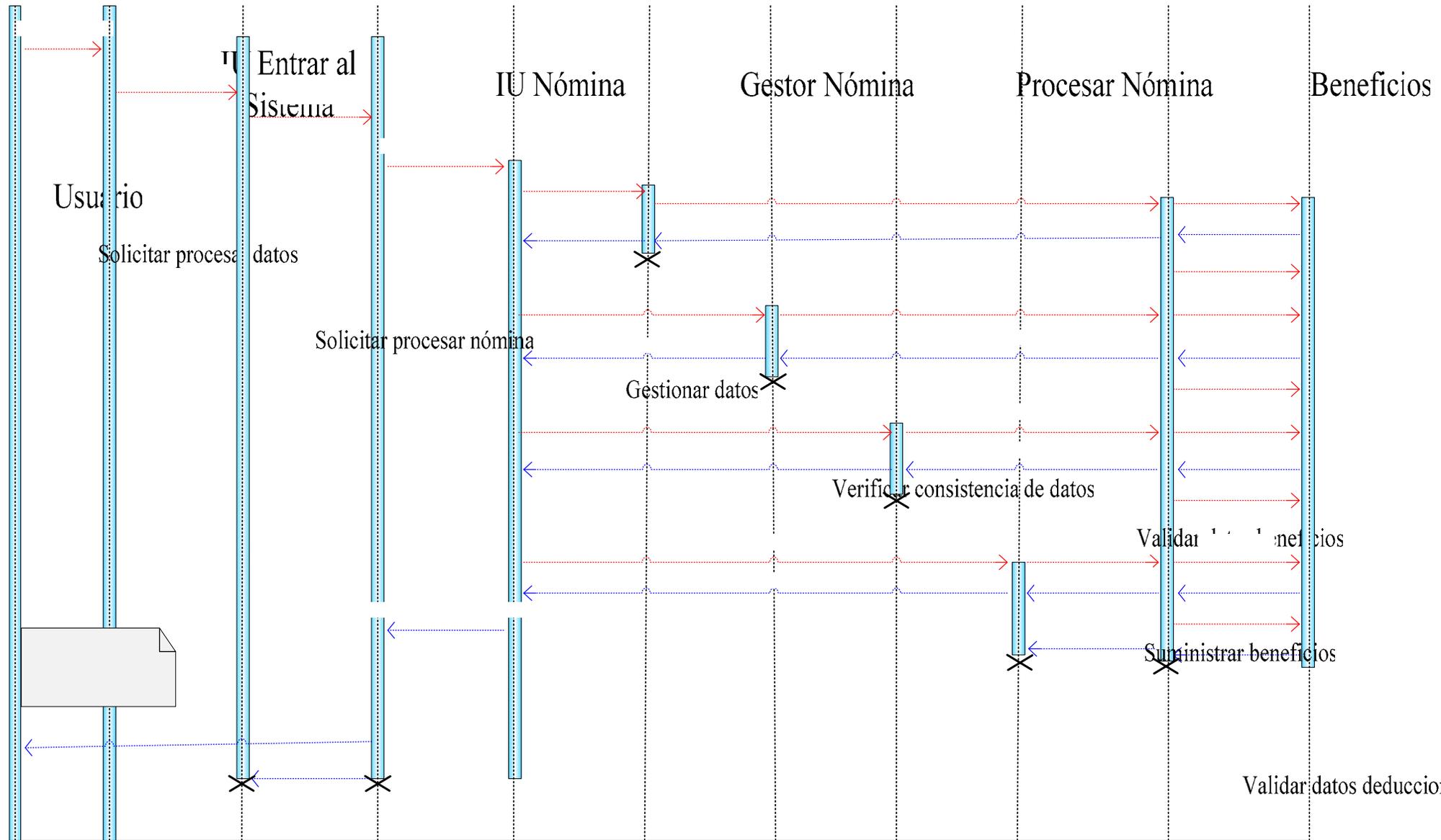
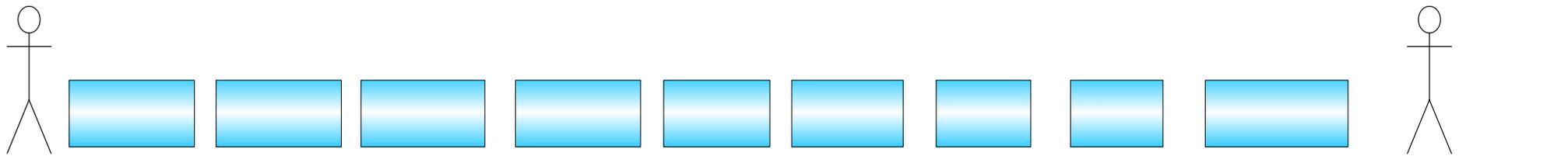
A continuación se representan los diagramas de secuencia de los casos de uso más críticos del sistema. Las líneas de color azul mostradas en las figuras indican un recorrido de ejemplo para la explicación de las iteraciones.

4.4.2.1. Diagrama de secuencia del caso de uso: Gestionar Nómina

El diagrama de secuencia de la figura 4.9 muestra las instancias de las clases y actores y el paso de mensajes entre dichos objetos de manera jerárquica.

A continuación se explica cada uno de los mensajes que se relacionan con los objetos cuando el actor *Usuario* desea generar la nómina de un empleado, los cuales entran en funcionamiento cuando:

- 👍 El Usuario solicita procesar la nómina en la IU. Entrar al Sistema.
- 👍 La interfaz activa al objeto Gestor Nómina, que gestiona los datos involucrados para generar la nómina, dicho gestor se encarga de validar la información, en los objetos Beneficios, Deducciones, Nómina y Empleados; a su vez ellos solicitan a la clase Interfaz de comunicación con la Base de Datos procesar la operación necesaria para verificar los datos.
- 👍 El objeto IS. Manejador Base de Datos se encarga de solicitarle al actor BD SADET que procese la solicitud realizada, para luego administrar dicha información para habilitar los datos a los objetos Beneficios, Deducciones, Nómina y Empleados; para luego ordenarla y mostrarla a través de la IU. Nómina. una vez habilitada esta información se procede a la destrucción de los objetos empleados en el proceso. Finalmente se procede a destruir los objetos utilizados.



Validar datos deducción

Suministrar deducción

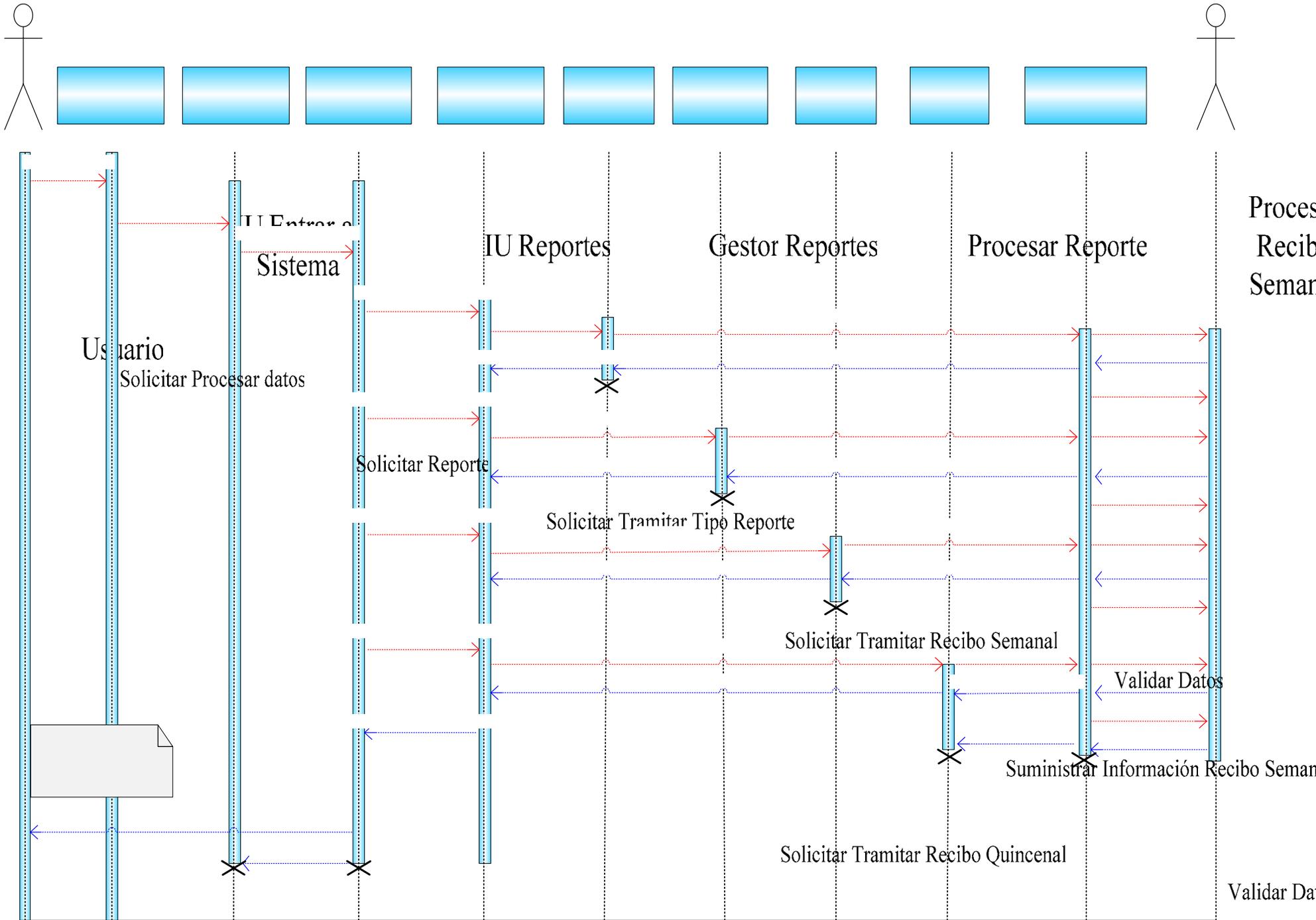


4.4.2.2. Diagrama de secuencia del caso de uso: Generar Reportes

Las instancias de las clases y actores y el paso de mensajes entre dichos objetos de manera jerárquica se muestran a continuación.

Cuando el usuario desea procesar un reporte de un Recibo de Pago Quincenal en la figura 4.10 entra en funcionamiento cuando:

- ☞ El usuario solicita procesar datos para el reporte en la IU. Entrar al Sistema.
- ☞ El actor activa el objeto IU. Reporte donde se solicita tramitar los datos para procesar el reporte de Recibo de Pago Quincenal de un empleado.
- ☞ La interfaz activa al objeto Procesar Reporte para gestionar los datos suministrados, validando.
- ☞ A través del mensaje solicitar Reporte de Recibo de Pago Quincenal el objeto Procesar Reporte activa el objeto IS Manejador Base de Datos, donde se procesa la operación necesaria para verificar los datos.
- ☞ El objeto IS: Manejador Base de Datos solicita al actor BD-SADET procesar la solicitud realizada, el cual administra el proceso para habilitar los datos del Recibo de Pago Quincenal de un empleado y visualizarlos a través de la IU Reporte.
- ☞ Finalmente se procede a destruir los objetos utilizados.





4.4.3. Subsistemas de la Aplicación

En el diseño de la arquitectura, se muestra la capa específica y la capa general de la aplicación, estas están compuestas de los paquetes de análisis realizados en la fase anterior. En la capa intermedia y la capa de software del sistema se muestran los subsistemas relacionados con software prefabricado, que ejecute y proporcione soporte a las funcionalidades requeridas por el sistema, tales como sistemas operativos, manejadores de bases de datos y componentes reutilizables.

La capa específica de la aplicación está conformada por los diferentes paquetes de análisis que constituyen el sistema SADET y la capa general de la aplicación se encuentra conformada por el paquete de datos del sistema, el cual contiene todas las clases de entidad relacionada con el sistema.

La capa intermedia está integrada por PHP, que junto al manejador de bases de datos MySQL permite generar páginas WEB de contenido dinámico. Las peticiones a dichas páginas son procesadas por el servidor WEB, pudiendo concebir de esta forma la ejecución del sistema SADET.

En la capa de software del sistema, el subsistema TCP/IP es el método de varias capas mediante el cual los datos se envían por la red que conecta al usuario con el servidor. El sistema operativo es el software encargado de coordinar, administrar y gestionar todos los recursos del sistema, para así dar soporte a las actividades realizadas por los demás subsistemas, el sistema manejador de base de datos MySQL, es el encargado de manejar o atender de forma transparente las solicitudes realizadas sobre la base de datos. En la figura 4.12 se muestra la Vista Lógica de las Capas de la Arquitectura del Sistema.

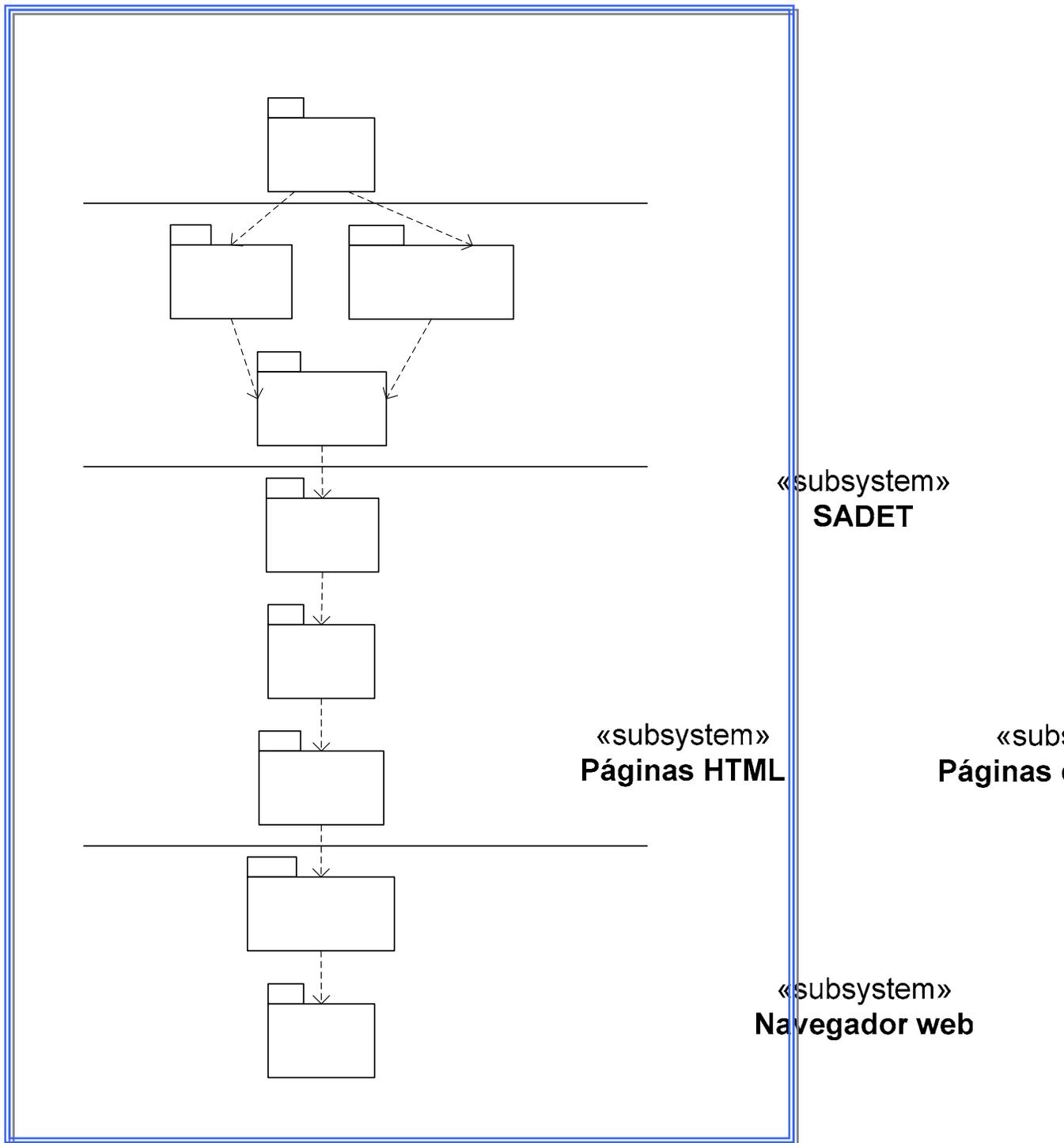


Figura 4.12.: Vista Lógica de las Capas de la Arquitectura del Sistema SADET



4.4.4. Prototipo de la Interfaz de Usuario

El prototipo de la interfaz de usuario que se muestra en la figura 4.12, ayuda a comprender y especificar las interacciones entre el personal administrativo de la empresa TECOM con el proyecto SADET.

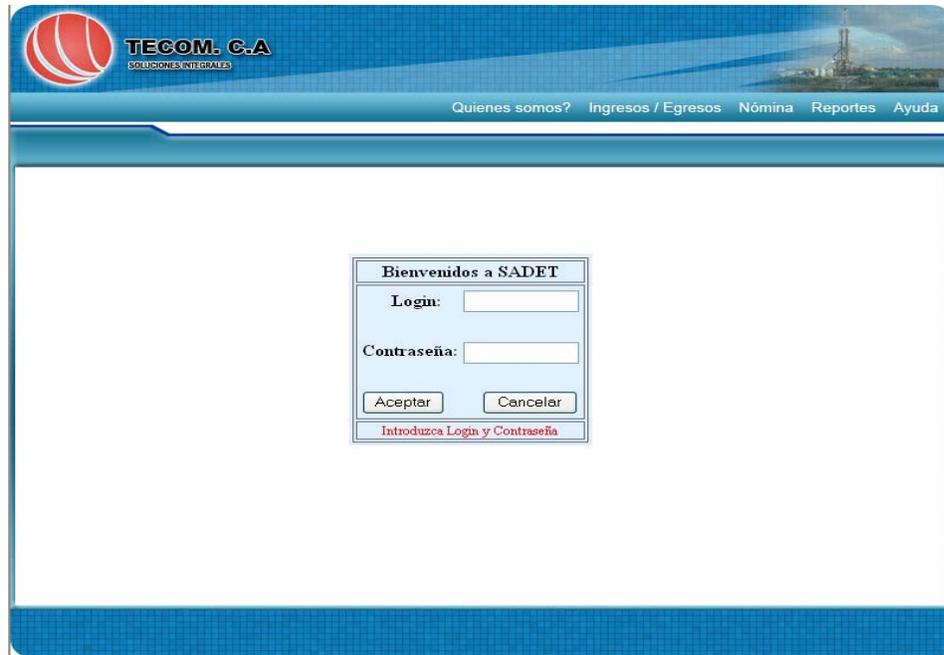


Figura 4.12: Prototipo de la interfaz de usuario del sistema.

El modelo definido como interfaz de usuario ilustra como los usuarios pueden utilizar el sistema para activar los casos de uso. La mayoría de los casos de uso analizados contienen una o más clases de entidad asociadas, que representan objetos de la empresa manejados por el sistema. El usuario necesita acceder a la información contenida en los mismos y manipularlas. Para ello, la interfaz, debe contar con elementos que permitan realizar tareas de agregación, búsqueda, actualización y eliminación de los datos correspondientes al objeto en cuestión. Por tal motivo y con la intención de estandarizar las distintas interfaces requeridas se ha diseñado una interfaz modelo que generalice las funciones que deben poseer todas.



4.4.5. Diseño de las bases de datos

Basándose en las clases de análisis estudiadas en el flujo de análisis, se identificaron los componentes que va a tener la Base de Datos que se va a diseñar.

Igualmente, el diseño de la misma se basará en el modelo relacional, el cual representa la base de datos como una colección de relaciones y las tablas serán normalizadas y llevadas a la Forma Normal de Boyce Codd (FNBC), la cual proporciona las mejores herramientas para que la base de datos sea más eficiente.

4.4.5.1. Identificación de la Tablas

En esta sección se procederá a realizar una análisis minucioso para lograr la identificación adecuada de las tablas que van a formar la Base de Datos del Sistema.

Las tablas identificadas son las siguientes:

Tabla Empleados: Esta tabla guarda la información relacionada con los datos de los empleados que laboran en la empresa.

empleados (idempleado: int(4), cedula: varchar(12) nombre: varchar (30), apellido: varchar (25), Dirección: varchar (50), tlf: varchar(11), nacimiento: varchar(12), fecha_ingreso: double, fecha_egreso: double, idcargo: int(2), salario: double, salario_basico: double, descanso legal: double, idnomina: int(2), idcontrato: int(2)).

Tabla Contrato: En esta tabla se almacenan los nombres de los distintos contratos que se ejecutan en la empresa.

contrato (idcontrato: int(2), nom_contrato: varchar(30))

Tabla Cargo: En esta tabla se almacenan los nombres de los distintos cargos existentes en la empresa.

cargo (idcargo: int(2), nom_cargo: varchar(30))



Tabla Nomina: En esta tabla se almacenan los nombres de los tipos de nóminas existentes en la empresa, es decir, la nómina fija y contratada.

nomina (idnomina: int(2), nombre: varchar(15))

Tabla Beneficios: En esta tabla se almacena los datos referentes a los beneficios de cada uno de los empleados de la empresa.

beneficios (idbeneficio: int(2), idempleado: int(4), ayuda_ciudad: double, transporte: double, vacacional: double, navideno: double, sobretiempo: int(2), feriados: int(2), bono nocturno: int(2)).

Tabla Deducciones: En esta tabla se almacena los datos referentes a las deducciones de cada uno de los empleados de la empresa.

deducciones (iddeducción: int(1), lph: double, paro forzoso: double, ivss: double, cajade ahorro: double, fecha: double).

Tabla Detalle Gastos: En esta tabla se almacena los datos referentes a los gastos generados por la empresa.

detalle_gasto (iddetallegasto: int(2), idgasto: int(4), fecha: double, cheque: double, monto: double, destinatario: varchar(100), detalles: text).

Tabla gastos: En esta tabla se almacena los datos referentes a los tipos de gastos generados por la empresa, por ejemplo: nómina, artículos de oficina, etc.

gasto (idgasto: int(2), nombre: varchar(30)).

Tabla Ingresos: En esta tabla se almacena los datos referentes a los ingresos de dinero al capital de la empresa.

ingresos (idingreso: int(2), idemisor: varchar(15), fecha: double, facture: varchar(10), monto: double).



Tabla Emisor: En esta tabla se almacena los datos referentes a la persona que emite el ingreso a la empresa.

emisor (idemisor: int(2), nombre: varchar(30)).

4.5. RESUMEN DE LA FASE

En la fase de elaboración se identificaron y refinaron nuevos requisitos, ampliando así los ya encontrados en la fase de inicio, dando como producto final la obtención de la línea base de la arquitectura del sistema **SADET**, la cual queda expresada en las vistas arquitectónicas del modelo de caso de uso, del modelo de análisis y del modelo de diseño.

Se requirió de una iteración debido a que fue posible identificar todos los requisitos, visualizados ya desde la fase de inicio y mediante un análisis y diseño realizado considerando los riesgos previamente identificados.

El modelo de diseño implementado en esta fase abarcó la creación de las clases de diseño, para la formulación del diagrama de clases y los diagramas de secuencia.

Se realizó el diseño de la base de datos del sistema, denominada SADET, guiado por el modelo relacional y las formas normales

CAPÍTULO 5: FASE DE CONSTRUCCIÓN

5.1. FASE DE CONSTRUCCIÓN

La construcción esta orientada a la implementación y pruebas al sistema. Se detallan los casos de uso y escenarios restantes y se modifica la descripción de la arquitectura si es necesario, con el fin de terminar los modelos de análisis, diseño e implementación. Además se realiza la integración y prueba de los subsistemas y la integración y prueba del sistema completo.

La etapa de **construcción** es fundamental porque permite implementar los componentes ya diseñados, además de la puesta en marcha del proyecto creando un producto software con capacidad operativa inicial, mejor conocida como “versión Beta”, esperando que en un futuro sea utilizada y actualizada para generar nuevas versiones del proyecto **SADET**.

En esta fase se culmina el estudio del sistema con la implementación total del mismo y la ejecución de las primeras pruebas haciendo énfasis en el flujo de trabajo fundamental de implementación y prueba tal y como se muestran en la *figura 5.1*.

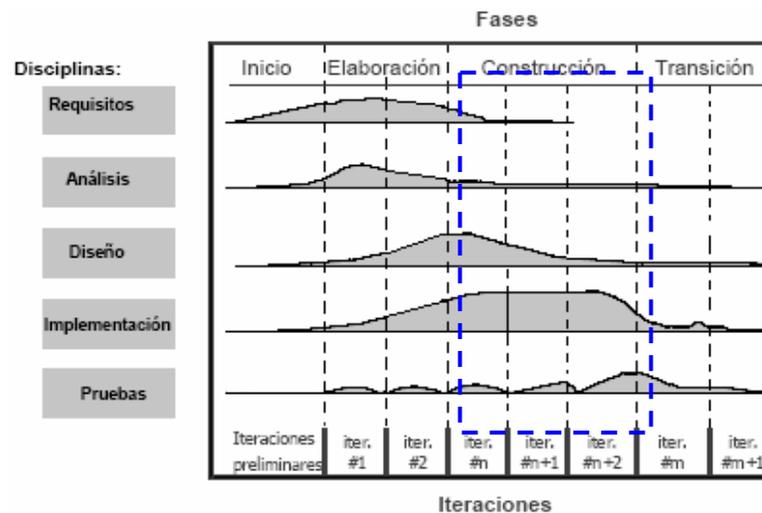


Figura 5.1.: Diagrama de los flujos de trabajo fundamental en la fase de Construcción



El fundamento de la fase de construcción es desarrollar el flujo de trabajo de implementación y prueba. En la implementación se sustenta la codificación de cada componente, los cuales formarán parte del sistema ejecutable, y en las pruebas se valida la calidad del software como mecanismo que posee el Ingeniero de software.

En esta etapa se estima codificar la interfaz de usuario, para fundamentar la base de la comunicación entre ellas. Por la cantidad de datos que maneja este software se debe realizar las pruebas de unidad e integración, la primera para verificar la entrada y salida de datos a través de la prueba de la caja negra y la segunda para evaluar la unión entre los componentes.

No se consideró la necesidad de identificar nuevos requisitos durante esta fase y se trabajará directamente en la implementación y las pruebas en base al análisis y diseño de la fase de elaboración.

5.2. ESCOGENCIA DEL LENGUAJE DE PROGRAMACIÓN

Para el desarrollo del software se debe emplear un lenguaje que aplique un enfoque orientado a objeto, ya que la programación orientada a objeto promueve una mejor comprensión de los requisitos, diseños más limpios y sistemas con mejor mantenimiento. Es relativamente sencillo implementar un diseño orientado a objeto con un lenguaje orientado a objeto, por cuanto las estructuras del lenguaje son similares a las estructuras del diseño.

SADET es un proyecto que será implementado en **PHP**, el cual es un lenguaje de servidor que se escoge para la codificación de las páginas del servidor que cambian el estado de la base de datos mediante consultas a MySQL. PHP es ideal para la creación de sitios Web ya que es un potente y robusto lenguaje de programación embebido en documentos HTML, dispone de librerías de conexión con la mayoría de sistemas de gestión de bases de datos para el almacenamiento de



información permanente en el servidor. Además proporciona soporte a múltiples protocolos de comunicaciones en Internet (HTTP, IMAP, FTP, LDAP, SNMP, etc.).

La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.

Las aplicaciones realizadas en el lenguaje PHP, requieren el uso de un servidor para realizar la ejecución de la misma, por ese motivo se utilizará el servidor Apache, para ejecutar la aplicación **SADET**.

El servidor Apache, es un servidor http de código abierto para plataformas UNIX (BSD, GNU/Linux), Windows y otras, que implementa el protocolo http/1.1(RFC 2616) y la noción de sitio virtual.

Para la construcción de las páginas HTML que contendrán el contenido de **SADET** en su versión remota, se usa **Macromedia Dreamweaver 8**, debido a que es una herramienta robusta para la creación de aplicaciones y sitios Web. Ofrece una combinación muy útil de herramientas de diseño visual, funciones de desarrollo de aplicaciones y soporte para la edición de código. Todo ello permite a los desarrolladores y diseñadores de diferentes niveles de conocimiento crear sitios y aplicaciones atractivos basados en estándares de forma rápida. Dreamweaver incluye numerosas herramientas y funciones relacionadas con la codificación, ayudando a crear aplicaciones Web dinámicas basadas en bases de datos, empleando tecnologías de servidor como PHP, ColdFusion, ASP, ASP.NET, JSP, etc.

La gestión de las páginas del servidor y las consultas a la base de datos de **TECOM**, cuyo diseño se presenta más adelante, serán manejadas usando el DBMS **MySQL**. Las características principales de este gestor de base de datos se basan en



aprovechar la potencia de sistemas multiprocesador, gracias a su implementación multihilo; en el soporte de gran cantidad de tipos de datos para las columnas; en la disposición de API's en gran cantidad de lenguajes (C, C++,Java,PHP,etc.); en el soporte de hasta 32 índices por tabla y en la gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

MySQL es un sistema de gestión de bases de datos relacional (SQL), licenciado bajo la GPL de la GNU. Es la base de datos de código fuente más usada del mundo.

Luego de haber justificado la escogencia de las herramientas de desarrollo, se realizarán los flujos de trabajos de Implementación y prueba.

5.3. IMPLEMENTACIÓN

La implementación se inicia a partir del resultado obtenido del análisis y diseño de la arquitectura del sistema, para luego implementar este modelo (implementación) en términos de componentes, es decir; ficheros de código fuente, ejecutables entre otros. En este punto se realiza la identificación de los subsistemas de implementación que serán necesarios para realizar la implementación del software.

El modelo de implementación contiene la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma y su organización en componentes. Esta vista proporciona la oportunidad de establecer correspondencias entre los subsistemas de implementación.

- 👍 Este flujo de trabajo para la fase de construcción es decisivo para finalizar el desarrollo de SADET, llevándose a cabo la construcción de todos los componentes que conforman la aplicación es por esto que el trabajo realizado será fundamentalmente la codificación de los componentes del modelo de implementación, la integración y pruebas de dichos componentes, obteniendo de esta forma una versión aceptable de SADET.



5.3.1. Diagrama de Componentes de SADET

El diagrama de componentes permite describir los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos software, incluyendo componentes de código fuente, componentes del código binario y componentes ejecutables. Un módulo de software se puede representar como un componente. También puede usarse para mostrar interfaces y las dependencias de llamada entre componentes, usando flechas con líneas discontinuas desde los componentes a las interfaces de otros componentes.

Es importante mencionar que el sistema SADET es muy robusto, por lo que contiene muchos componentes; por tal razón a continuación se muestra sólo el diagrama de componentes del módulo Nómina.

En la figura 5.2 se puede observar como el componente Procesar Nómina tiene relación con los componentes Crear Beneficios, Crear Deducciones e Ingresar Empleados los cuales son necesarios para el funcionamiento del componente Modelo Procesar Nómina.

5.3.2. Implementación de la base de datos

Se crea la base de datos TECOM según lo establecido. En las figuras se pueden ver las definiciones de algunas de las tablas que componen la base de datos mencionada, además se incluyen los campos pertenecientes a dichas tablas.

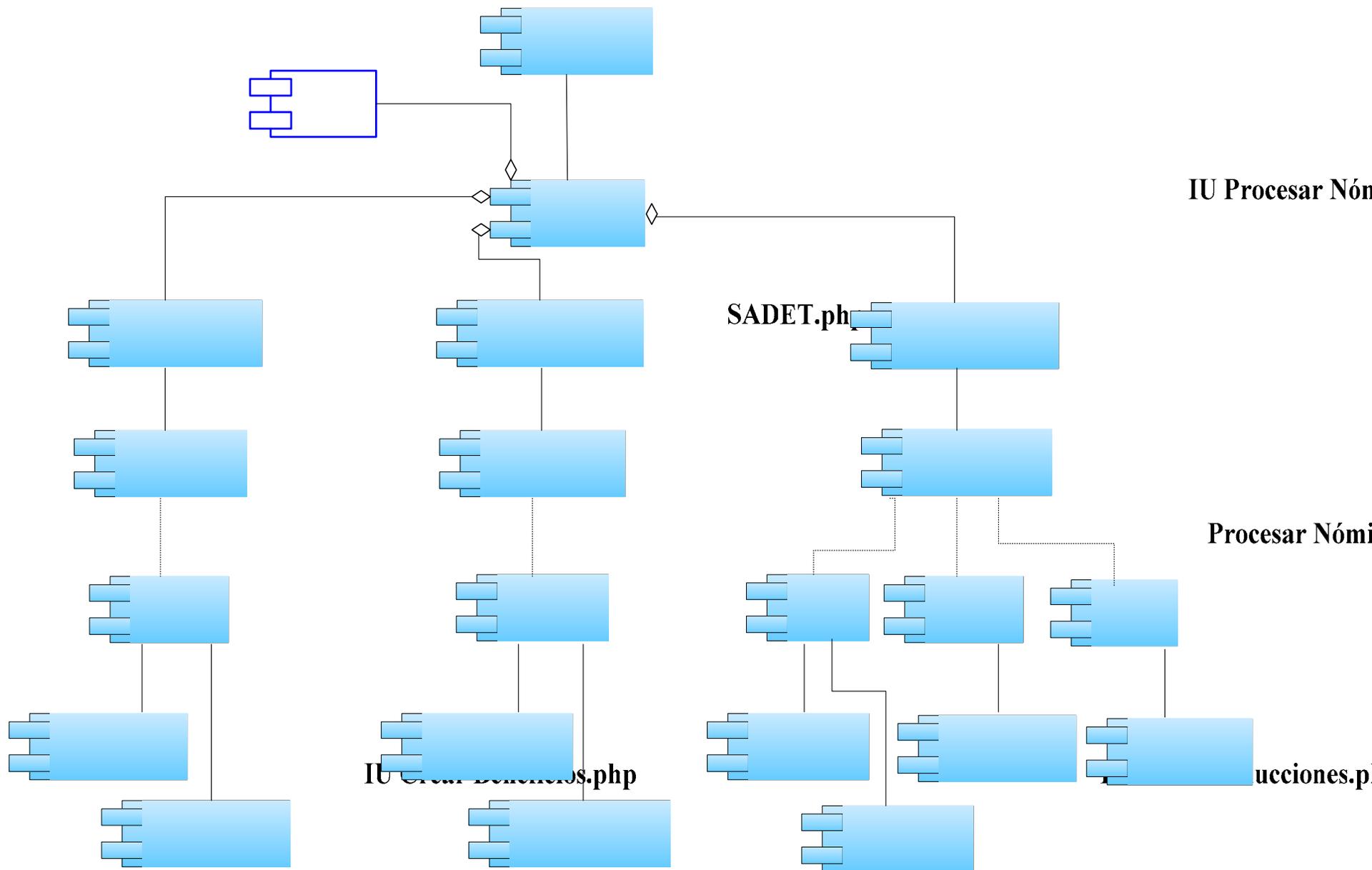


Figura 5.2. : Diagrama de componente del modulo Nómina.



```
CREATE TABLE `empleados` (  
  `idempleado` int(4) NOT NULL auto_increment,  
  `cedula` varchar(12) collate latin1_spanish_ci NOT NULL default "",  
  `nombre` varchar(30) collate latin1_spanish_ci NOT NULL default "",  
  `apellido` varchar(25) collate latin1_spanish_ci NOT NULL default "",  
  `direccion` varchar(50) collate latin1_spanish_ci default NULL,  
  `tlf` varchar(11) collate latin1_spanish_ci default NULL,  
  `nacimiento` varchar(12) collate latin1_spanish_ci default NULL,  
  `fecha_ingreso` double NOT NULL default '0',  
  `fecha_egreso` double default NULL,  
  `idcargo` int(2) NOT NULL default '0',  
  `salario` double NOT NULL default '0',  
  `salario_basico` double NOT NULL default '0',  
  `descanso_legal` double NOT NULL default '0',  
  `idnomina` int(2) NOT NULL default '0',  
  `idcontrato` int(2) NOT NULL default '0',  
  PRIMARY KEY (`idempleado`,`cedula`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci  
AUTO_INCREMENT=5;
```

Figura 5.3. Definiciones de la tabla empleados de la base de datos TECOM

```
CREATE TABLE `beneficios` (  
  `idbeneficio` int(2) NOT NULL auto_increment,  
  `idempleado` int(4) NOT NULL default '0',  
  `ayuda_ciudad` double default NULL,  
  `transporte` double default NULL,  
  `vacacional` double default NULL,  
  `navideno` double default NULL,  
  `sobretiempo` int(2) default '0',  
  `feriados` int(2) default '0',  
  `bono nocturno` int(2) default '0',  
  PRIMARY KEY (`idbeneficio`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci  
AUTO_INCREMENT=5;
```

Figura 5.4. Definiciones de la tabla beneficios de la base de datos TECOM



```
CREATE TABLE `ingresos` (  
  `idingreso` int(4) NOT NULL auto_increment,  
  `idemisor` varchar(15) collate latin1_spanish_ci NOT NULL default "",  
  `fecha` double NOT NULL default '0',  
  `factura` varchar(10) collate latin1_spanish_ci NOT NULL default "",  
  `monto` double NOT NULL default '0',  
  PRIMARY KEY (`idingreso`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci  
AUTO_INCREMENT=6 ;
```

Figura 5.5. Definiciones de la tabla ingresos de la base de datos TECOM

```
CREATE TABLE `deducciones` (  
  `iddeducion` int(1) NOT NULL auto_increment,  
  `lph` double NOT NULL default '0',  
  `paroforzoso` double NOT NULL default '0',  
  `ivss` double NOT NULL default '0',  
  `cajadeahorro` double NOT NULL default '0',  
  `fecha` double NOT NULL default '0',  
  PRIMARY KEY (`iddeducion`)  
)ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci  
AUTO_INCREMENT=2 ;
```

Figura 5.6. Definiciones de la tabla deducciones de la base de datos TECOM

5.4. PRUEBAS

Las pruebas realizadas al sistema SADET son efectuadas con el propósito de garantizar su calidad. Dichas Pruebas representan una revisión final de las especificaciones del diseño y codificación, con la finalidad de descubrir posibles defectos o errores que pudiese contener la aplicación durante su ejecución.

Las pruebas constituyen el proceso de ejecutar y evaluar el sistema para verificar si éste satisface las especificaciones requeridas.

Los componentes que presentan defectos deberán ser corregidos, regresándolos al diseño o implementación, y probándolo nuevamente.



Las pruebas que se aplicaron al sistema SADET para evaluar su funcionamiento y efectividad, son las pruebas de unidad y las de integración. A continuación se describe en que consisten ambas pruebas en el entorno de la aplicación.

5.4.1. Pruebas de unidad

Las pruebas por unidad se aplicarán mediante la prueba de la caja negra en el componente *Procesar Nómina*. Para realizar este tipo de pruebas se identifican un conjunto de valores que pueden ser introducidos por un actor y se expresaron como clases de equivalencia para poder abarcar la totalidad de las ocurrencias de un evento de inserción de datos.

A continuación, son representadas las clases de equivalencia del componente IU Procesar Nómina, dicho componente es el encargado de la inclusión de los datos de la nómina de empleados de la empresa.

En la tabla 5.1 se observan las clases de equivalencia según los datos que se almacenan en la tabla Empleados al momento de añadir un nuevo empleado.

Tabla 5.1. Clases de equivalencia para las pruebas de caja negra de *Ingresar Empleado*. 1/2

Nº	Dato	Clase de Equivalencia	Válido	Inválido
1	Nombre	Caracteres numéricos		X
2	Nombre	Caracteres alfabéticos	X	
3	Nombre	Caracteres alfanuméricos		X
4	Nombre	Longitud de caracteres<1		X
5	Nombre	3<=Longitud de caracteres<=40	X	
6	Nombre	Longitud de caracteres>40		X
7	Nombre	Dato Nulo		X
8	Apellido	Caracteres numéricos		X
9	Apellido	Caracteres alfabéticos	X	



10	Apellido	Caracteres alfanuméricos		X
11	Apellido	Longitud de caracteres<1		X
12	Apellido	3<=Longitud de caracteres<=40	X	
13	Apellido	Longitud de caracteres>40		X
14	Apellido	Dato Nulo		X
15	Cédula	Caracteres numéricos	X	
16	Cédula	Caracteres alfabéticos		X
17	Cédula	Caracteres alfanuméricos		X
18	Cédula	Longitud de caracteres<6		X
19	Cédula	6<=Longitud de caracteres<=10	X	
20	Cédula	Longitud de caracteres>10		X
21	Cédula	Dato Nulo		X
22	Dirección	Caracteres numéricos		X
23	Dirección	Caracteres alfabéticos		X
24	Dirección	Caracteres alfanuméricos	X	
25	Dirección	Longitud de caracteres<10		X
26	Dirección	10<=Longitud de caracteres<=100	X	
27	Dirección	Longitud de caracteres>100		X
28	Dirección	Dato Nulo	X	

Se realizaron pruebas de caja negra para los módulos previamente construidos, con la finalidad de garantizar que la aplicación maneje la información de manera fidedigna.

Para comprobar que la información de los empleados ingresados se ajusta a los requerimientos y restricciones de la aplicación, se plantean casos de prueba de la caja negra que permiten evaluar las salidas que genera la aplicación según las clases de equivalencia previamente definidas. En la tabla 5.2 se muestran los casos de prueba



en el ingreso de nuevos empleados a la tabla Empleados. Si se cumple todas las clases involucradas con dicho dato la salida será válida sino será inválida.

Tabla 5.2. Casos de prueba de caja negra para Ingresar Empleados. 1/2

Nº	Dato	Casos de Prueba	Salida	Clases cubiertas
1	Nombre	`6235´	Inválida	1
2	Nombre	`Maria´	Válida	2,5
3	Nombre	`Jose15´	Inválida	3
4	Nombre	`a´	Inválida	2,4
5	Nombre	`pia´	Válida	2,5
6	Nombre	`Julio Alberto Rodrigo Hector Pedro Juan´	Inválida	2,6
7	Nombre	Null	Inválida	7
8	Apellido	`1980´	Inválida	8
9	Apellido	`Martinez´	Válida	9,12
10	Apellido	`Pérez50´	Inválida	10
11	Apellido	`S´	Inválida	9,11
12	Apellido	`um´	Inválida	9,12
13	Apellido	`Díaz García´	Válida	9,12
14	Apellido	Dato Nulo	Inválida	14
15	Cédula	`6532´	Inválida	15,18
16	Cédula	`JoseA´	Inválida	16
17	Cédula	`15789387´	Válida	15,19
18	Cédula	`1508198014854893´	Inválida	15,20
19	Cédula	` evaluacion325´	Inválida	17
20	Cédula	`Null´	Inválida	21
21	Cédula	`14854893´	Válida	15,19
22	Dirección	`38351´	Inválida	22
23	Dirección	` hola´	Inválida	23
24	Dirección	`teo2´	Inválida	25
25	Dirección	`c/Democracia N° 16´	Válida	24,26
26	Dirección	`alejandra12583´	Inválida	25



27	Dirección	Null	Válida	28
28	Dirección	'C/Tucupita N° 6'	Válida	17,19

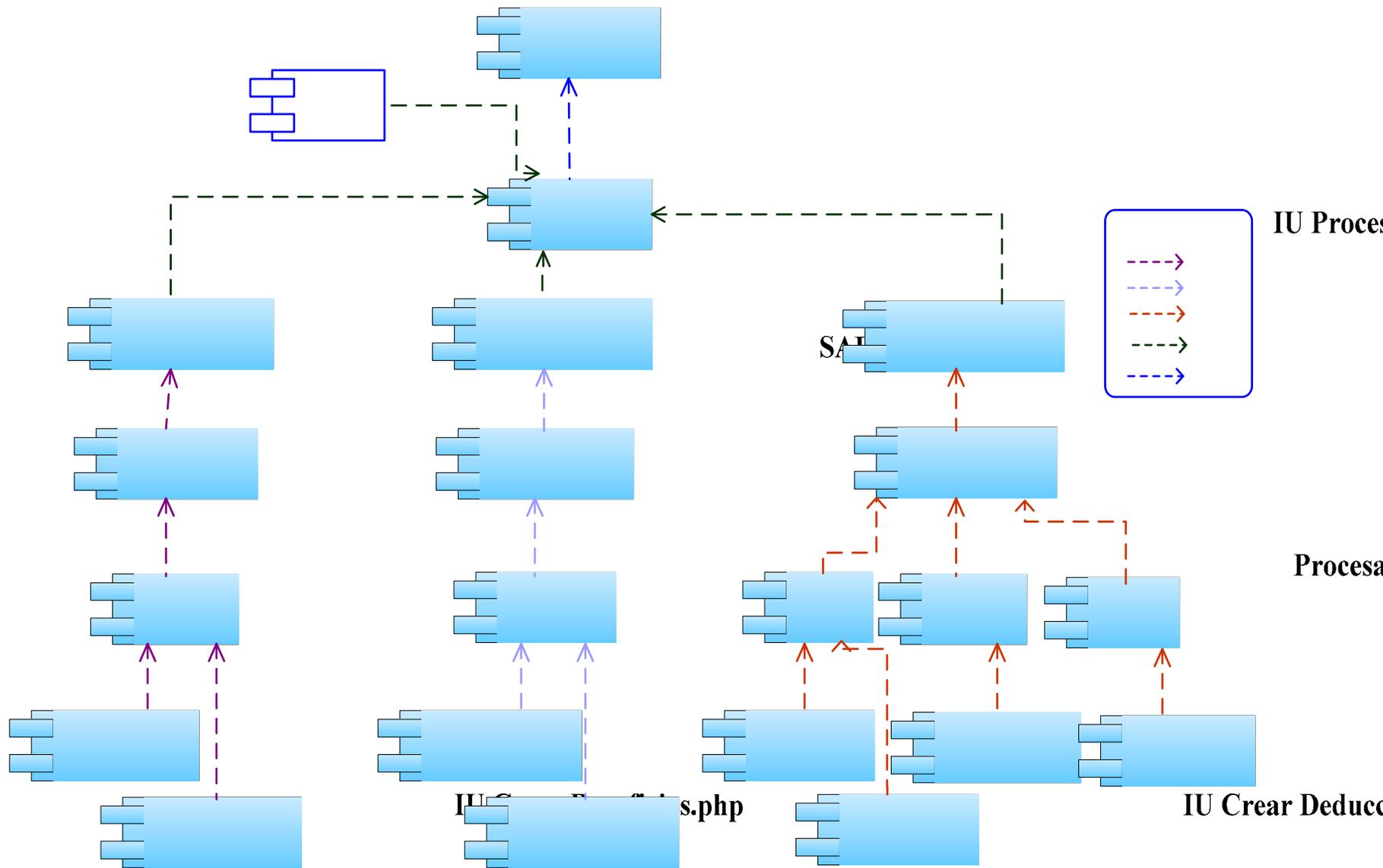
5.4.2. Pruebas de integración

Luego de verificar la calidad de cada hoja del diagrama de componentes se procede a comprobar la eficiencia de un conjunto de componentes integrados que completan la funcionalidad del Sistema Administrativo SADET, se realizaron pruebas de integración para verificar que los mismos dieran en conjunto el resultado esperado, y de no ser así, corregir las fallas detectadas antes de la entrega a los usuarios.

Los casos de prueba que se utilizaron para validar el funcionamiento de SADET, pueden observarse a través de las fases de integración en el modulo Ingresar Nomina verificado con la integración de los subsistemas Crear beneficios, Crear Deducciones e Ingresar Empleado presentadas bajo la figura 5.4.

Mediante las fases de integración (fig. 5.4) se proporciona una forma de organizar los componentes del modelo de implementación en piezas manejables. Cada Prueba de integración se representa por fase y se denota con un color distinto, donde cada componente del diagrama que está relacionado con otro por un mismo color fue sometido a pruebas, para luego realizar la integración.

La integración del sistema fue realizada de manera incremental, es decir; se fueron incorporando componentes del sistema a la construcción de la aplicación, de acuerdo a la planificación de integración jerárquica, en pequeños segmentos de programas, previamente probados en unidad, logrando la detección y corrección de errores.



CrearBeneficios.php

CrearDeducion



En la figura 5.4 se muestran diferentes trazas y objetos resaltados de colores, los cuales se relacionan para efectuar las pruebas de integración correspondiente.

5.4.2.1. Prueba de Integración del componente Crear Beneficios

Para determinar el funcionamiento correcto de la implementación de los componentes asociados con la realización de los beneficios de un empleado, correspondiente a la primera fase de integración según la figura 5.4., se realizaron una serie de pruebas.

Estas pruebas están basadas en la verificación del funcionamiento correcto de los campos de texto, los botones, listas de selección, etc., cuando el usuario accede a ellos.

Para este caso de procesar beneficios el componente se denomina Crear Beneficios, en el cual se probará su funcionamiento con los demás componentes involucrado en este caso. Esto se considera una acertada prueba de integración por el contenido de varios componentes de diversas índoles.

A continuación se muestran una serie de figuras mediante las cuales se evidencia gráficamente las pruebas de integración realizadas a los componentes involucrados en el ingreso de un empleado.

Procedimiento de Prueba:

1. Se selecciona Nómina en el menú, ubicado en la parte superior derecha de la interfaz principal de SADET. Luego se presiona Empleados y a su vez se selecciona el tipo de empleado a insertar (semanal o quincenal), para activar los procesos y las tablas asociados a éste. Seguido a esto se muestra la interfaz Gestionar Empleados.

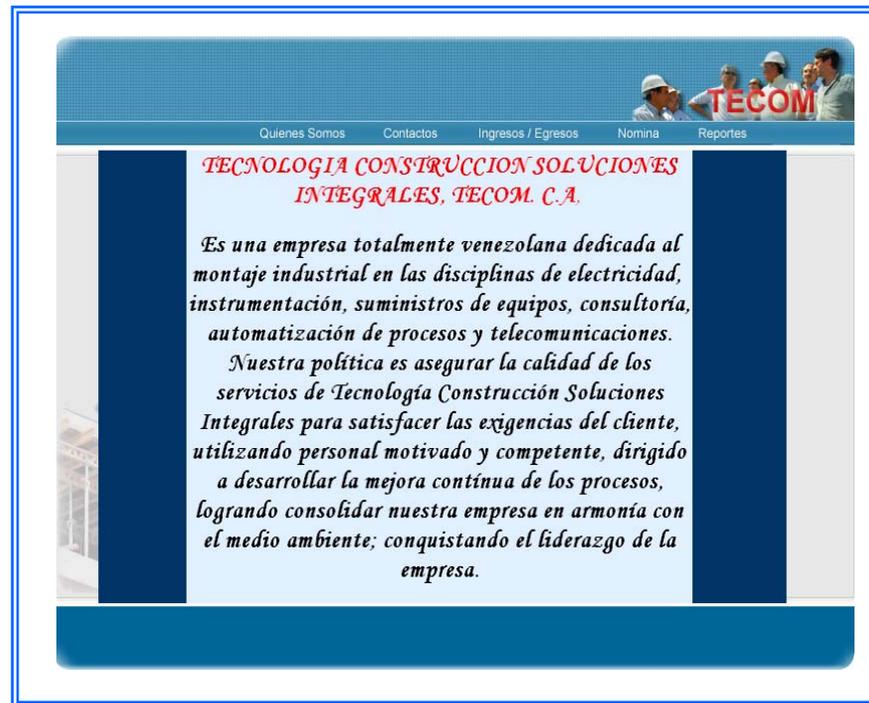


Figura 5.5 Componente Gestionar Empleados_Paso 1.

2. Se procede a ingresar los datos del empleado y luego presione el botón INGRESAR como se muestra en la figura 5.6.

TECOM. C.A.
SOLUCIONES INTEGRALES

Quienes somos? Ingresos / Egresos Nómina Reportes Ayuda

Gestionar Empleados

Datos Personales

Cédula:

Nombre:

Apellido:

Dirección:

Teléfono: Fecha Nac.:

Datos Nómina Quincenal

Nómina:

Fecha Ingreso:

Salario: Bs. Fecha Retiro:

Cargo:

Contrato:

Figura 5.6 Componente Gestionar Empleados _Paso 2.



3. La interfaz Gestionar_Empleado carga los datos del empleado y los muestra al usuario. (figura 5.7). En caso de que quiera ingresar los beneficios para ese empleado debe presionar el botón ASIGNAR BENEFICIOS.

TECOM. C.A.
SOLUCIONES INTEGRALES

Quienes somos? Ingresos / Egresos Nómina Reportes Ayuda

Gestionar Empleados

Datos Personales

Cédula: 14316266

Nombre: LENYS

Apellido: RENGEL

Dirección: POZUELOS

Teléfono: 04168875230 Fecha Nac.: 18/02/1980 ...

Datos Nómina Quincenal

Nomina: Fija

Salario: 2000 Bs.

Cargo: GERENTE GENERAL

Contrato: TRABAJOS DE REDES VOZ

Figura 5.7 Componente Gestionar Empleados _Paso 3.

4. El usuario ingresa los beneficios que el empleado devenga y presiona el botón insertar, luego el sistema muestra los datos ingresados. (figura 5.7).



TECOM. C.A.
SOLUCIONES INTEGRALES

Quienes somos? Ingresos / Egresos Nómina Reportes Ayuda

Gestionar Beneficios

Datos del Empleado

Código: 2 Cédula: 14316266

Nombre: LENYS

Apellido: RENGEL

Beneficios

Transporte: 0 Bs. Ayuda de Ciudad: 600000 Bs.

Navideño: 120000 Bs.

Sobre Tiempos & Feriados

Domingo: 0 Bs.

Bono Nocturno: 4 Horas.

Ingresar Limpiar

Microsoft Internet Explorer

REGISTRO INGRESADO

Aceptar

Figura 5.8 Componente Gestionar Empleados_Paso 4.

Se puede observar a través de estos pasos que los componentes involucrados funcionaron adecuadamente por ejemplo cuando se ingresaron los datos del empleado y posteriormente se verificó su existencia en la base de datos para luego hacer el ingreso de sus beneficios.

De esta manera se obtuvo que al integrar los módulos del sistema, tal es el caso de la interfaz de Nómina los enlaces de las distintas actividades presentes en la primera se conectaron satisfactoriamente a su respectivo contenido en la segunda interfaz. Igualmente, se probó que cada botón en la zona de control de SADET daba como resultado la acción esperada. Al final de las pruebas se concluyó que SADET se



encuentra operativo y que no se necesita otra iteración para entrar en la fase de transición, en donde se pondrá a prueba con los usuarios finales.

5.5. CODIFICACIÓN DE LOS COMPONENTES DE SADET

- 👍 Es importante realizar la construcción del software incrementalmente en pasos manejables, de forma que cada paso dé lugar a pequeños problemas de integración o prueba. El resultado de cada paso es una versión ejecutable del sistema.
- 👍 A continuación se detallará la codificación de algunos de los procedimientos de los componentes especificados, los cuales se consideran más relevantes desde el punto de vista de la arquitectura del sistema.

Código fuente para la conexión general con la base de datos SADET

```
<?php  
function Conectarse()  
{  
    $host = "localhost"; //servidor de p  
    $bd = "TECOM";  
    $login = "root";  
    $password = "";  
    $link = mysql_connect($host,$login,$password);  
    mysql_select_db($bd,$link);  
    return $link;  
}  
?>
```



Código fuente para el Componente Modelo Gestionar Empleado

```
<?php
require_once("../Connections/tecom.php");
?>
<html>
<head>
<title>...::: TECOM, C.A. :::...</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">@import url(..\jscalendar-1[1].0/jscalendar-
1.0/calendar-blue.css);</style>
<style type="text/css">
-->
</style>
<script type="text/javascript" src="..\jscalendar-1[1].0/jscalendar-
1.0/calendar.js" ></script>
<script type="text/javascript" src="..\jscalendar-1[1].0/jscalendar-
1.0/lang/calendar-es.js"></script>
<script type="text/javascript" src="..\jscalendar-1[1].0/jscalendar-
1.0/calendar-setup.js"></script>
<script type="text/JavaScript">
<!--
<CENTER>
<!--Crear div que contendra el contenido de la pantalla--
>
<div id="contenedor" style="width:792px;
height:451px; background-color:#C2E1ED; overflow:auto;">
<form name="empleados"
action="..\php/procesareempleadosemanal.php" method="post">
<fieldset>
```



```
Personales</legend>
align="center">
colspan="7" align="left" valign="middle" scope="row">
align="left">&nbsp;  </div>
width="3"></th>
width="40"></th>
type="hidden" name="codigo" value="<?php if (isset($_GET['codigo'])) {echo
$_GET['codigo'];} ?>" />
width="3"></th>
width="69"></th>
width="14"></th>
width="24"></th>
height="26" valign="top" scope="row">
align="right" class="Estilo2">C&eacute;dula:</div>
valign="top">
name="cedula" type="text" size="15" value="<?php if (isset($_GET['ced'])) {echo
$_GET['ced'];} ?>" onKeyPress="return validarn(event)" />
</label>
<label></label>
```

```
<legend class="Estilo9">Datos
<table width="440" border="0"
<tr>
<th height="21"
<div
</th>
<th
<th
<th width="35">
<input
</th>
<th
<th
<th
<th
</tr>
<tr>
<th width="71"
<div
</th>
<td colspan="2"
<input
<label>
```



```

</td>
<td colspan="7"
valign="top">
<input
name="boton" type="submit" id="boton" value="Buscar &gt;&gt;" />
</td>
<td>&nbsp;&nbsp;&nbsp;</td>
<td
colspan="2"><!--DWLayoutEmptyCell-->&nbsp;&nbsp;&nbsp;</td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<th height="24"
valign="top" scope="row">
<div
align="right" class="Estilo2">Nombre:</div>
</th>
<td colspan="10"
valign="top">
<label>
<input name="nombre" type="text" size="40" value="<?php if
(isset($_GET['nombre'])) {echo $_GET['nombre'];}?" onKeyPress="return
validar(event)"/>
</label>
</td>
</tr>
<tr>
<th height="24"
valign="top" scope="row">
<div
align="right" class="Estilo2">Apellido:</div>
</th>
<td colspan="10"
valign="top">
<input
name="apellido" type="text" size="40" value="<?php if
(isset($_GET['apellido'])) {echo $_GET['apellido'];}?" onkeypress="return
validar(event)"/>
<label></label>
</td>
</tr>
</tr>
```



```

        <tr>
            <th height="24"
                valign="top">
                <div
                    align="right" class="Estilo2"><strong>Direcci&oacute;n:</strong></div>
            </th>
            <td colspan="13"
                valign="top">
                <label>
                    <input name="direccion" type="text" size="58" value="<?php if
                    (isset($_GET['dir']))){echo $_GET['dir'];}?>" />
                </label>
            </td>
        </tr>
        <tr>
            <th height="28"
                valign="top">
                <div
                    align="right" class="Estilo2"><strong>Tel&eacute;fono:</strong></div>
            </th>
            <td colspan="3"
                valign="top" >
                <input
                    name="tlf" type="text" size="11" maxlength="11" value="<?php if
                    (isset($_GET['tlf']))){echo $_GET['tlf'];}?>" onKeyPress="return validarn(event)"/>
            </td>
            <td colspan="5"
                valign="top" >
                <div
                    align="right" class="Estilo2"><strong>Fecha Nac.: </strong></div>
            </td>
            <td colspan="5"
                valign="top" >
                <input
                    name="nacimiento" type="text" id="Nacimiento" value="<?php if
                    (isset($_GET['nacimiento']))){echo $_GET['nacimiento'];}?>" size="10"
                    maxlength="12" onKeyPress="return validarn(event)" readonly="readonly" />
                <button
                    id="trigger">...</button>
            </td>
        </tr>
        <script
            type="text/javascript">Calendar.setup({
```



```
inputField : "Nacimiento", // ID of the input field

ifFormat : "%d/%m/%Y", // the date format

button : "trigger" // ID of the button

});
</script>
</td>
</tr>
</table>
</fieldset>
<fieldset>
<legend class="Estilo9">Datos
N&oacute;mina Semanal</legend>
<table width="440" border="0"
align="center">
<tr>
<th height="5"
colspan="7"></th>
</tr>
<tr>
<th height="31"
valign="top" scope="row">
align="right" class="Estilo2">Cargo:</div>
</th>
<th colspan="5"
valign="top">
align="left">
<select name="cargo">
<?php
mysql_select_db($database_tecom,$tecom);
$sql = "SELECT * FROM cargo ";
$resultado=mysql_query($sql,$tecom) or die ("Invalid query");
while ($fila = mysql_fetch_assoc($resultado)){
```



```
        if (isset($_GET['cargo']) &&
($_GET['cargo']==$fila['idcargo'])){
            echo "<option selected='selected' value="; echo
$fila['idcargo'].''";echo ">";
            echo $fila["nom_cargo"];
            echo "</option>";
        }else{
            echo "<option value=";echo $fila['idcargo'].''"; echo
">";
            echo $fila["nom_cargo"];
            echo "</option>";
        }
    }
?>
</select>
</div>
</th>
<th colspan="5"
valign="top">
align="right" class="Estilo2"><strong>Salario Basico:</strong></div>
</th>
<td valign="top">
<input
name="salario" type="text" size="10" maxlength="15" value="<?php if
(isset($_GET['basico']))){echo $_GET['basico'];}?>" onkeypress="return
validarn(event)"/>
</td>
valign="top"><strong class="Estilo2">Bs.</strong></td>
</tr>
```




```

        </tr>
        <tr>
            <th height="40"
                colspan="2" valign="top">
                <div
                    align="right" class="Estilo2">
                    <div
                        align="right" class="Estilo2"><strong>Fecha Ingreso:</strong></div>
                    </div>
                </th>
            <th colspan="4"
                align="left" valign="top" >
                <input
                    name="ingreso" type="text" id="Ingreso" value="<?php if
                    (isset($_GET['ing']))){echo date("d/m/Y",$_GET['ing']);}?>" size="10"
                    maxlength="12" onkeypress="return validarn(event)" readonly="readonly"/>
                <button
                    id="trigger1">...</button>
                <script
                    type="text/javascript">Calendar.setup({
                        inputField : "Ingreso", // ID of the input field
                        ifFormat : "%d/%m/%Y", // the date format
                        button : "trigger1" // ID of the button
                    });
                </script>
            </th>
            <th colspan="5"
                valign="top" >
                <div
                    align="right" class="Estilo2">Fecha Retiro:</div>
            </th>
            <td colspan="3"
                rowspan="2" valign="top" >
                <div
                    align="left">
                    <input name="egreso" type="text" id="A&ntilde;o Retiro" size="10"
                    maxlength="12" value="<?php if (isset($_GET['egr']) && ($_GET['egr']!="0"
                    )){echo date("d/m/Y",$_GET['egr']);}?>" readonly="readonly" />

```



```
<button id="trigger2">...</button>

<script type="text/javascript">Calendar.setup({
inputField : "Año Retiro", // ID of the input field
ifFormat : "%d/%m/%Y", // the date format
button : "trigger2" // ID of the button
});

</script>

</div>
</td>
</tr>
<tr>
<th height="21"
valign="top"><!--DWLayoutEmptyCell-->&nbsp;</th>
<th colspan="4"
align="left" valign="top" ><!--DWLayoutEmptyCell-->&nbsp;</th>
<th width="3"
></th>
<th width="10"
></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
<tr>
<th height="36">
<div
align="right" class="Estilo2">Contrato:</div>
</th>
<th colspan="2" >
<div
align="left">
<select name="contrato">
<?php
```



```
mysql_select_db($database_tecom,$tecom);

$sql = "SELECT * FROM contrato ";

$resultado=mysql_query($sql,$tecom) or die ("Invalid query");

while ($fila = mysql_fetch_assoc($resultado)){

    if (isset($_GET['contrato']) &&
($_GET['contrato']==$fila['idcontrato'])){

        echo "<option selected='selected' value="; echo
$fila['idcontrato'].''";echo ">";

        echo $fila["nom_contrato"];

        echo "</option>";

    }else{

        echo "<option value=";echo $fila['idcontrato'].''"; echo
">";

        echo $fila["nom_contrato"];

        echo "</option>";

    }

}

?>

</select>

width="3">&nbsp;</th>

width="3">&nbsp;</th>

</div>

</th>

<th

<th

<th>&nbsp;</th>
```



```

                                <th colspan="7"
valign="top">
                                <table
border="0" cellpadding="0" cellspacing="0" width="150">
                                <!-- fwtable
fwsrc="beneficios.png" fwbase="beneficios.gif" fwstyle="Dreamweaver" fwdocid =
"863853108" fwnested="0" -->
                                <tr>
                                <td></td>
                                <td></td>
                                </tr>
                                <tr>
                                <td>
                                <a href="gestionarbeneicios.php?ced=<?php if
(isset($_GET['ced'])){echo $_GET['ced'];}?>&nombre=<?php if
(isset($_GET['nombre'])){echo $_GET['nombre'];}?>&apellido=<?php if
(isset($_GET['apellido'])){echo $_GET['apellido'];}?>&codigo=<?php if
(isset($_GET['codigo'])){echo $_GET['codigo'];}?>"
onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('beneficios_r1_c1','../exportados/beneficios_r1_c
1_f2.gif',1);">
                                <?php if ( (isset($_GET["enable"])) &&
($_GET["enable"]=="1")){echo "<img name=\"beneficios_r1_c1\"
src=\"exportados/beneficios_r1_c1.gif\" width=\"150\" height=\"24\" border=\"0\"
id=\"beneficios_r1_c1\" alt=\"\" />";}?>
                                </a>
                                </td>
                                <td></td>
                                </tr>
                                <tr>

```



```
<td></td>
```

```
<td></td>
```

```
</tr>
```

```
</table>
```

```
</th>
```

```
<td>&nbsp;&nbsp;&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td
```

```
height="21">&nbsp;&nbsp;&nbsp;</td>
```

```
<td
```

```
width="13">&nbsp;&nbsp;&nbsp;</td>
```

```
<td
```

```
width="92">&nbsp;&nbsp;&nbsp;</td>
```

```
</tr>
```

```
</table>
```

```
</fieldset>
```

```
<table width="440" height="25"
```

```
border="0" align="center" cellpadding="2">
```

```
<tr>
```

```
<td width="14"
```

```
height="28"><input type="hidden" name="MM_insert" value="empleados" /></td>
```

```
<td width="64"
```

```
valign="top">
```

```
<label>
```

```
<input
```

```
name="boton" type="submit" id="boton"
```

```
onClick="MM_validateForm('cedula','RisNum','nombre','R','apellido','R','direcci
```

```
on','R','tlf','RisNum','D&iacute;a Nacimiento','NinRange1:31','A&ntilde;o
```

```
Nacimiento','NinRange1900:9999','salario','RisNum','D&iacute;a
```

```
Ingreso','RinRange1:31','Mes Ingreso','RinRange1:12','A&ntilde;o
```

```
Ingreso','RinRange1990:9999','Dia Retiro','NinRange1:31','Mes
```

```
Retiro','NinRange1:12','A&ntilde;o Retiro','NinRange1990:9999');return
```

```
document.MM_returnValue" value="Ingresar" <?php if ( (isset($_GET["enable"]))
```

```
&& ($_GET["enable"]=="1"))){echo "disabled";}?> />
```

```
</label>
```

```
</td>
```



```
width="20">&nbsp;</td>
valign="top">
name="boton" type="submit" id="boton" value="Modificar" />
width="27">&nbsp;</td>
valign="top">
name="boton" type="submit" id="boton" value="Limpiar" />
width="14">&nbsp;</td>
valign="top">
name="boton" type="submit" id="boton" value="Eliminar" />
width="44">&nbsp;</td>
</tr>
</table>
</form>
<?php
if (isset($_GET["error"])){
    if($_GET["error"]=="1"){
        echo "<script
language='javascript'>";
        echo "alert('ESTA
CEDULA YA FUE ASIGNADA A UN EMPLEADO)";
        echo "</script>";
    }
    if ($_GET["error"]=="2"){
```



```
language='javascript">";
ESTA ASIGNADA A NINGUN EMPLEADO)";
language='javascript">";
INGRESADO)";
language='javascript">";
MODIFICADO)";
echo "<script
echo "alert('CEDULA NO
echo "</script>";
}
}else if (isset($_GET["aviso"])){
if ($_GET["aviso"]=="1"){
echo "<script
echo "alert('REGISTRO
echo "</script>";
}
if ($_GET["aviso"]=="2"){
echo "<script
echo "alert('REGISTRO
echo "</script>";
}
}
?>
</div><!--Cierre del Div Contenedor-->
</CENTER>
</body>
</html>
```



5.6. RESUMEN DE LA FASE

Durante la fase de construcción se codificaron, probaron e integraron todos los subsistemas de SADET, mediante una serie de construcciones que conllevaron a la capacidad operativa total del sistema.

También se consiguió diseñar e implementar la arquitectura del sistema dando como resultado un software en su primera versión de prueba que ofrece cubrir las expectativas del usuario para el que fue desarrollado.

Las pruebas son importantes ya que es el único mecanismo que posee los Ingenieros de Software para validar la calidad del software, es por ello que se aplicaron tanto pruebas de unidad como de integración.

La aplicación de la prueba de la caja negra permitió identificar y corregir errores asociados a los datos de entrada.

La prueba de integración no es menos importantes ya que mediante estas se valida la integridad de varios componentes que fueron enlazados, permitiendo así probar si la comunicación entre dichos componentes se llevo a cabo eficientemente.

CAPÍTULO 6: FASE DE TRANSICIÓN

6.1. FASE DE TRANSICIÓN

La etapa de transición permite probar la calidad del software sobre la plataforma del entorno de operación. Es donde los usuarios empiezan a interactuar con el sistema y el desarrollador del software comienza a corregir defectos encontrados.

El objetivo principal en esta fase, es preparar la versión Beta del software e instalarlo en el entorno de trabajo para lo cual fue creado, cumpliendo el flujo de trabajo de pruebas como se puede observar en la figura 6.1:

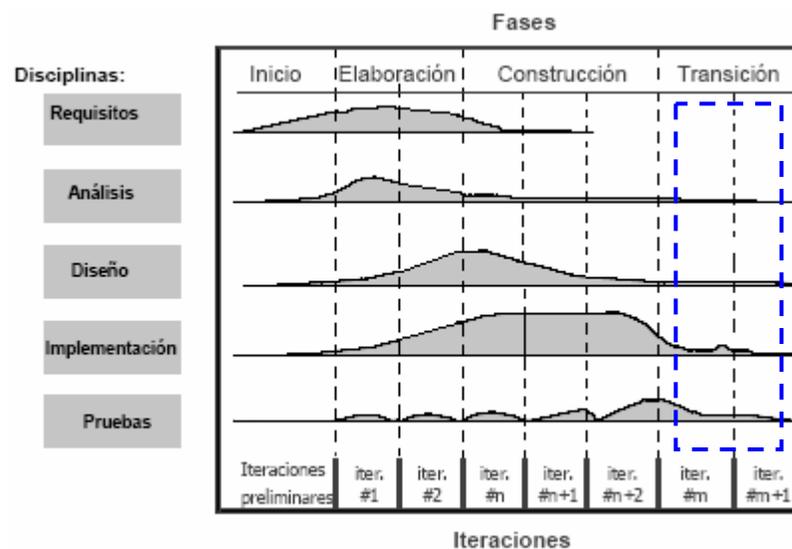


Figura 6.1. Diagrama de los flujos de trabajo fundamental en la fase de Transición

En la fase de inicio se establecieron requisitos que fueron analizados y diseñados en la fase de elaboración, para ser codificados en la etapa de construcción y finalmente implantado en la fase de transición. En esta fase se ha culminado el estudio del sistema y éste se encuentra operativo fundamentándose en el flujo de trabajo de pruebas con la finalidad de ajustarlo a los parámetros del entorno de trabajo de la empresa.



6.2. PREPARACIÓN DE LA VERSIÓN BETA

Se preparan documentos acerca del procedimiento de la aplicación, los cuales estarán dirigidos al personal administrativo de la empresa.

Se seleccionan los usuarios Beta, quienes poseen alta experiencia sobre el funcionamiento de los procesos que fueron automatizados.

6.3. INSTALACIÓN DE LA VERSIÓN BETA

La instalación se desarrolló en el computador de la administración de la empresa donde funcionará el proyecto. Se instalaron todas las herramientas de soporte para la aplicación como el servidor de base de datos MySQL y PHP.

Se realizó una presentación con las funciones principales del software con la finalidad de que el usuario tuviera las nociones necesarias para manejar sin ninguna dificultad la aplicación.

6.4. REACCIÓN A LOS RESULTADOS DE LAS PRUEBAS

Se encontraron algunas fallas de codificación que fueron corregidas, no se encontró errores de mayor importancia que comprometieran la arquitectura del sistema desarrollado, por lo que no hubo necesidad de desarrollar ningún flujo de trabajo adicional.

6.5. RESUMEN DE LA FASE

En esta fase se instaló la versión Beta del software y se efectuaron las pruebas de aceptación, con las cuales fueron localizadas fallas menores en algunos componentes. Estas fallas fueron corregidas sin afectar la arquitectura del sistema. De esta manera concluye la fase final del proceso de desarrollo de software con una versión completa y corregida del producto.

CONCLUSIONES

1. El sistema SADET representa una herramienta automatizada, integral y confiable que facilita la gestión de las actividades administrativas desempeñadas dentro de la empresa.
2. Se completo exitosamente la integración, pruebas y documentación del funcionamiento del sistema, obteniendo un software altamente funcional.
3. Las interfaces del sistema fueron diseñadas bajo la concepción de que el usuario se sienta cómodo, interactuando con el sistema de forma rápida y sencilla.
4. En la construcción de la base de datos SADET, fueron empleadas las técnicas del Modelo Relacional de Base de Datos, permitiendo un efectivo diseño lógico de las tablas necesarias para el correcto funcionamiento del software.
5. En la fase de construcción se concretaron la mayoría de las funcionalidades a nivel de aplicación, que permitieron realizar las operaciones diseñadas y analizadas en las fases anteriores.
6. Este sistema cumplió con los objetivos esperados, ya que la implementación de SADET permite agilizar las actividades administrativas relacionadas con los empleados de la empresa, mediante la automatización del flujo de información asociado a los procesos descritos en este trabajo de investigación, resaltando por su importancia la Gestión de las Nóminas.

RECOMENDACIONES

1. Realizar la configuración inicial completa del sistema para su correcto funcionamiento, obteniendo así los mejores beneficios del software construido.
2. Realizar planes de entrenamiento a los usuarios que faciliten la manipulación de la aplicación.
3. Mejorar la base de datos existente a fin de que las consultas realizadas a la página se hagan en menor tiempo y de manera optimizada.

BIBLIOGRAFÍAS

OCHOA, Y. y MADRUGA, R. (2006) **“Desarrollo de un software para la gestión y control de las actividades administrativas de la central telefónica e la Universidad de Oriente, Núcleo Anzoátegui”**. Trabajo de grado, Ingeniería en Computación, Universidad de Oriente. Anzoátegui, Venezuela.

PORRAS, K. y GONZÁLEZ, M. (2006) **“Desarrollo de un software para la automatización de las actividades relacionadas con la nómina de empleados de una alcaldía, adscrita a la gobernación del estado Anzoátegui”**. Trabajo de grado, Ingeniería en Computación, Universidad de Oriente. Anzoátegui, Venezuela.

GARCÍA, M. y MUJICA V. (2005) **“Desarrollo de un software para la automatización de los procesos administrativos del Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo Anzoátegui”**. Trabajo de grado, Ingeniería en Computación, Universidad de Oriente. Anzoátegui, Venezuela.

SIMONI, M. (2004) **“Desarrollo un sistema de información para la gestión de procesos administrativos que mejore el desempeño de las actividades del Departamento de Ciencias de la Unidad de Estudios Básicos de la Universidad de Oriente, Núcleo de Anzoátegui”**. Trabajo de grado, Ingeniería en Computación, Universidad de Oriente. Anzoátegui, Venezuela.

OTERO, J. (2004) **“Desarrollo de una aplicación para el control administrativo de una organización de una organización farmacéutica”**. Trabajo de grado, Ingeniería en Computación, Universidad de Oriente. Anzoátegui, Venezuela.

Cota A. (1994) "**Ingeniería de Software**". Soluciones Avanzadas. Julio de 1994. pp. 5-13.

Jacobson, I. *et. al.* (1992). "**Object-Oriented Software Engineering; A Use Case Driven Aproach**". ACM Press. Adison-Wesley Publishing. Co. U.S.A. 524 p. Ilus. pp. 465-493.

Jacobson I., Booch G. y Rumbaugh J. (2000). "**El Proceso Unificado de Desarrollo de Software**". Primera Edición. Pearson Educación. Madrid.

http://www.lafacu.com/apuntes/informatica/apuntes_de_programacion/default.htm.

"**Conceptos de Bases de Datos**" (2001).

http://www.lafacu.com/apuntes/informatica/base_datos/default.htm.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	“Desarrollar un software para la gestión y control de las actividades administrativas de una empresa constructora ubicada en Puerto La Cruz - Estado Anzoátegui.”
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
RENGEL V. LENYS M.	CVLAC: 14.316.266 E MAIL: rengelvasquez@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:**Software Administrativo****Empleados****Nómina****Flujo de información**

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
INGENIERÍA	INGENIERÍA EN COMPUTACIÓN

RESUMEN (ABSTRACT):

La empresa Tecnología Construcción Soluciones Integrales, TECOM, C.A. realiza todas sus actividades administrativas de forma manual, trayendo como consecuencia retraso en el procesamiento de la información, por tal motivo se propuso el proyecto de Grado titulado: **DESARROLLO DE UN SOFTWARE PARA LA GESTIÓN Y CONTROL DE LAS ACTIVIDADES ADMINISTRATIVAS DE UNA EMPRESA CONSTRUCTORA UBICADA EN PUERTO LA CRUZ - ESTADO ANZOÁTEGUI**; el cual permitió la automatización de los procesos administrativos gestionados por la mencionada empresa, tales como: inserción de trabajadores, realización de recibos de pagos, etc. **SISTEMA ADMINISTRATIVO EMPRESAS TECOM (SADET)** fue desarrollado siguiendo la metodología del proceso Unificado de Desarrollo de Software, debido a su eficiencia en desarrollo de software aplicando las fases de inicio, elaboración, construcción y transición. Estas a su vez se llevan a cabo a través de una o varias iteraciones las cuales constan de cinco flujos de trabajo fundamentales como son: captura de requisitos, análisis, diseño, implementación y prueba. Se usaron Macromedia Firework 8 y Dreamweaver 8, para el diseño y construcción del sistema. La arquitectura se construyó de acuerdo al patrón cliente-servidor, razón por la cual se utilizó el lenguaje de programación **PHP**, el cual es el apropiado para construir sistemas con bases de datos usando el manejador **MySQL**.

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

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
	ROL	CA	AS	TU	JU
VERACIERTA GABRIELA	CVLAC:	14.616.683			
	E_MAIL	gveracierta@hotmail.com			
	E_MAIL				
	E_MAIL				
DORTA PEDRO	ROL	CA	AS	TU	JU
	CVLAC:	14.054.907			
	E_MAIL	victormujica@yahoo.com			
	E_MAIL				
MUJICA VICTOR	ROL	CA	AS <input checked="" type="checkbox"/>	TU	JU
	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

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: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

NIVEL ASOCIADO CON EL TRABAJO:

ÁREA DE ESTUDIO:

INSTITUCIÓN:

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

“Los Trabajos de Grado son de exclusiva propiedad de la Universidad de Oriente y solo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participará al Consejo Universitario para su aprobación.”

AUTOR

AUTOR

AUTOR

RENGEL V. LENYS M.

TUTOR

JURADO

JURADO

MUJICA VÍCTOR

VERACIERTA GABRIELA

DORTA PEDRO

POR LA SUBCOMISION DE TESIS

BASTARDO JOSÉ