



# BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación



## “SISTEMA PARA VALIDACIONES DE ADEUDOS DE PERSONAL Y GASTOS DE VIAJE EN UNA EMPRESA”

**TESIS PROFESIONAL**  
QUE PARA OBTENER EL TÍTULO DE  
LICENCIADA EN CIENCIAS DE LA COMPUTACIÓN

**PRESENTA**

ERIKA CORTEZ RAMÍREZ

**ASESOR:**

MTRO. PEDRO BELLO LOPEZ

Puebla, Pue. Diciembre 2015

# *Agradecimientos*

## *A mi madre*

*Por el apoyo que me brinda día a día impulsándome a seguir desarrollándome tanto de manera profesional como personal*

## *A mi hermana*

*Por la paciencia durante los momentos difíciles y el apoyo que me brinda durante mi desarrollo profesional.*

## *A mi sobrina*

*Por darme la fuerza para alcanzar mis metas profesionales y personales.*

## *A mis amigos*

*Por darme un espacio en su vida y brindarme el apoyo en los momentos más difíciles tanto en mi desarrollo profesional como personal.*



## Contenido.

---

<b>Índice de imágenes</b> .....	iii
<b>Índice de tablas</b> .....	v
<b>Introducción</b> .....	vii
<b>Capítulo 1. Marco teórico</b> .....	1
1.1 Ingeniería de software.....	1
1.1.1 Modelos de Proceso. ....	2
1.2 Metodología de desarrollo de Software.....	3
1.3 Lenguaje Unificado de Modelado (UML).....	8
1.4 Sistemas de Bases de Datos .....	10
1.4.1 Diseño de bases de datos .....	11
1.4.2 Modelos de base de datos.....	12
1.4.3 Normalización .....	13
1.5 Herramientas de desarrollo.....	14
1.5.1 PHP (PHP Hypertext preprocessor).....	14
1.5.2 MySQL.....	15
1.5.3 Apache.....	15
1.5.4 WAMP.....	16
1.5.5 HTML (HyperText Markup Language) .....	17
1.5.6 CSS (Cascading Style Sheets) .....	17
<b>Capítulo 2. Análisis y Diseño del sistema</b> .....	19
2.1 Planteamiento del problema .....	19
2.2 Análisis de requerimientos .....	20
2.3 Especificación de requerimientos.....	20
2.3.1 Objetivos Generales .....	21
2.3.2 Objetivos específicos .....	22
2.3.3 Metas .....	22
2.3.4 Alcances .....	22
2.3.5 Limitaciones.....	23



2.4 Antecedentes .....	23
2.5 Metodologías, Técnicas y Herramientas .....	25
2.6 Modelo funcional .....	25
2.7 Modelo conceptual .....	31
2.7.1 Glosario de términos .....	31
2.7.2 Diccionario del modelo .....	31
2.8 Modelo de Diseño .....	32
2.8.1 Caso de uso Solicitar validación de adeudo .....	33
2.9 Diseño de Base de Datos .....	34
<b>Capítulo 3. Implementación del sistema.</b> .....	<b>42</b>
3.1 Interfaz de acceso para el Usuario .....	42
3.2 Funciones del sistema .....	46
3.2.1 Solicitar validación de adeudos. ....	46
3.2.2 Validar adeudos de bajas. ....	48
3.2.3 Solicitar gastos de viaje .....	49
3.2.4 Autorizar gastos de viaje .....	50
3.2.5 Comprobar gastos de viaje .....	50
3.2.6 Validar comprobaciones .....	51
3.2.5 Consultar comprobaciones .....	52
3.3 Conclusiones .....	53
3.3.1 trabajo a futuro .....	54
<b>Bibliografía</b> .....	<b>55</b>
<b>Anexos.</b> .....	<b>57</b>
Anexo 1. Documentación de casos de uso. ....	57
Anexo 2. Escenarios de casos de uso .....	61
Anexo 3. Diagramas de Secuencia, Colaboración y Clase Pantalla, Control y Entidad ..	65



## Índice de imágenes

---

Figura 1.1 Capas de la Ingeniería de Software .....	1
Figura 1.2 Proceso del software .....	2
Figura 1.3 Proceso Unificado .....	5
Figura 2.1 FlexiServer .....	24
Figura 2.2 NOM2001 .....	24
Figura 2.3 Diagrama de casos de uso.....	28
Figura 2.4 Diagrama de Clase Pantalla, Control y Entidad .....	33
Figura 2.5 Diagrama de Secuencia .....	33
Figura 2.6 Diagrama de Colaboración.....	34
Figura 2.7 Relaciones entre entidades.....	35
Figura 2.8 Diagrama Entidad - Relación .....	36
Figura 2.9 Diagrama de clases.....	40
Figura 3.1 Pantalla de inicio de sistema.....	43
Figura 3.2 Pantalla de Inicio de sesión.....	44
Figura 3.3 Menú para el personal de Recursos Humanos.	44
Figura 3.4 Menú para el personal de Jefes de área. ....	45
Figura 3.5 Menú para el personal de Finanzas .....	45
Figura 3.6 Menú para Empleados. ....	46
Figura 3.7 Consultar información de Empleado .....	47
Figura 3.8 Ejemplo de validación de adeudo.....	47
Figura 3.9 Formulario Validación de adeudos .....	48
Figura 3.10 Solicitud de gastos de viaje.....	49
Figura 3.11 Formulario de autorización de gastos de viaje .....	50
Figura 3.12 Formulario de comprobación de gastos de viaje.....	51
Figura 3.13 Consulta de empleado .....	51



Figura 3.14 Comprobaciones de empleado .....	52
Figura 3.15 Validación de comprobación .....	52
Figura anexo 3. 1 Diagrama de Clase Pantalla, Control y Entidad.....	65
Figura anexo 3. 2 Diagrama de Secuencia .....	65
Figura anexo 3. 3 Diagrama de Colaboración .....	66
Figura anexo 3. 4 Diagrama de Clase Pantalla, Control y Entidad.....	66
Figura anexo 3. 5 Diagrama de Secuencia .....	66
Figura anexo 3. 6 Diagrama de Colaboración .....	67
Figura anexo 3. 7 Diagrama de Clase Pantalla, Control y Entidad.....	67
Figura anexo 3. 8 Diagrama de Secuencia .....	68
Figura anexo 3. 9 Diagrama de Colaboración .....	68



## Índice de tablas

---

Tabla 2.1 Documentación de Caso de uso Solicitar validación de adeudo .....	29
Tabla 2.2 Documentación de Caso de uso Solicitar gastos de viaje .....	30
Tabla 2.3 Clase Empleado .....	31
Tabla 2.4 Clase Área.....	32
Tabla 2.5 Clase Adeudo .....	32
Tabla 2.6 Clase Gastos de viaje.....	32
Tabla 2.7 Tabla Adeudo normalizada.....	37
Tabla 2.8 Tabla Autoriza normalizada .....	37
Tabla 2.9 Tabla departamento normalizada .....	37
Tabla 2.10 Tabla Empleado normalizada .....	37
Tabla 2.11 Tabla Viáticos normalizada .....	37
Tabla 2.12 Tabla gasto normalizada .....	37
Tabla 2.13 Tabla Adeudo .....	38
Tabla 2.14 Tabla Autoriza .....	38
Tabla 2.15 Tabla Departamento.....	38
Tabla 2.16 Tabla Empleado .....	39
Tabla 2.17 Tabla Gasto .....	39
Tabla 2.18 Tabla Viáticos .....	39
Tabla anexo 1. 1 Ingresar al sistema .....	57
Tabla anexo 1. 2 solicitar validación de adeudo.....	58
Tabla anexo 1. 3 Validar información de adeudo .....	59
Tabla anexo 1. 4 Autorizar gastos de viaje .....	59
Tabla anexo 1. 5 Consultar adeudos.....	60
Tabla anexo 1. 6 Solicitar gastos de viaje .....	60
Tabla anexo 1. 7 Comprobar gastos de viaje .....	61



Tabla anexo 2. 1 Caso de uso Ingresar al sistema flujo alterno 1 .....	61
Tabla anexo 2. 2 Caso de uso Ingresar al sistema flujo alterno 2 .....	62
Tabla anexo 2. 3 Caso de uso Ingresar al sistema flujo alterno 3 .....	62
Tabla anexo 2. 4 Caso de uso Ingresar al sistema flujo alterno 4 .....	62
Tabla anexo 2. 5 Caso de uso Solicitar validación de adeudo .....	63
Tabla anexo 2. 6 Caso de uso Autorizar solicitud de viaje .....	63
Tabla anexo 2. 7 Caso de uso Verificar comprobación de viaje .....	64



## Introducción

---

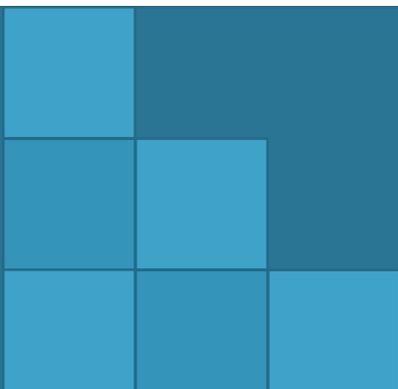
En un principio la web solo era una colección de páginas estáticas, documentos, etc., para su consulta o descarga. Con el paso del tiempo evoluciono a la creación de páginas dinámicas que permitieran definir mecanismos mediante los cuales se pudiera pasar información entre el servidor y ciertos programas externos, este método fue conocido como CGI (common gateway inteface). Además, de dar una total libertad para elegir el lenguaje de programación que se desea emplear. [1]

En la actualidad las aplicaciones web ofrecen grandes ventajas por ejemplo fácil acceso, ahorro de tiempo, compatibilidad entre sistemas, conexión con otras aplicaciones que no son necesariamente aplicaciones web, entre otras.

Basándose en las ventajas que nos brinda el desarrollo de aplicaciones web, se desarrolla el proyecto Sistema de validación de Adeudos y gastos de viaje la cual requiere una interfaz que permita interacción entre diferentes usuarios.

Para lograrlo se utilizó la metodología Proceso Unificado de Rational (RUP) y a través del Lenguaje Unificado de Modelado (UML) se logró interpretar y analizar los requerimientos del proyecto plasmándolos con ayuda del lenguaje PHP que se usó para la codificación, logrando con ello la comunicación con el gestor de base de datos MySQL.

En este documento podrá encontrar la información detallada del Análisis de requerimientos, Diseño del sistema, Implementación y Pruebas para el Sistema de Validación de adeudos y gastos de viaje. De igual manera se presentan las conclusiones y la visión que se tiene a futuro para el proyecto.



# **CAPITULO I**

## **Marco teórico**



## Capítulo 1. Marco teórico

### 1.1 Ingeniería de software

El software se ha hecho parte de nuestra vida diaria y, como consecuencia, el número de personas que se interesan en las características y funciones que brinda una aplicación específica es cada vez mayor. Cuando se construyen aplicaciones o sistemas para sus operaciones y controles cotidianos, implica desarrollar software de alta calidad que debe tener la facilidad de adaptarse a cambios y mejoras necesarias.

En otras palabras, la Ingeniería de Software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo, teniendo como principal objetivo el establecimiento y puesta en práctica de los principios y metodologías que nos lleven a un desarrollo eficiente de software en todas las etapas desde sus inicios hasta su implementación y mantenimiento. [1]

La ingeniería de software es una tecnología con varias capas como se muestra en la Figura 1.1 aquí podemos destacar la capa de proceso, que permite el desarrollo del software definiendo la estructura y el concepto para generar el producto, estableciendo los puntos clave.



Figura 1.1 Capas de la Ingeniería de Software



Podemos concluir que la ingeniería de software nos permite crear software de alta calidad, con apoyo de herramientas y metodologías que nos ayudan a entender las necesidades del cliente permitiéndonos definir la estructura del mismo para tener un producto de alta calidad.

### 1.1.1 Modelos de Proceso.

Los procesos de desarrollo de software juegan un papel muy importante durante la creación de un sistema de software, hasta su culminación donde intervienen todas las fases del sistema como los requerimientos, el desarrollo y la puesta en funcionamiento de cada una de las etapas e iteraciones.

Se puede definir como una colección de actividades, acciones y tareas que se realizan cuando se crea algún producto, cada actividad, acción o tarea que se encuentra dentro de una estructura o modelo que define su relación tanto con el proceso entre sí. En la Figura 1.2 Proceso del software podemos observar la estructura de un proceso de software.

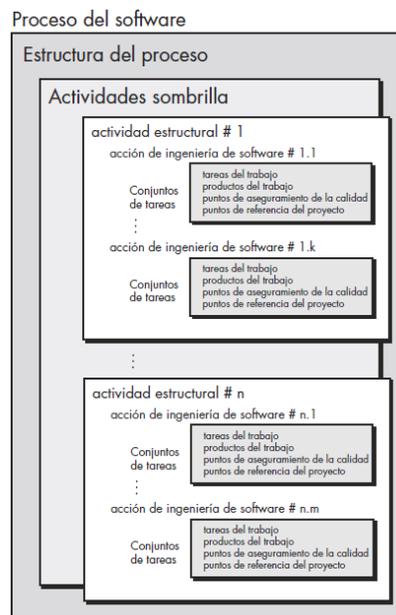


Figura 1.2 Proceso del software



Los flujos de proceso describen la forma de organización de las actividades, acciones y tareas para cada una de las cinco actividades. Los tipos de flujo de proceso son lineal, iterativo, evolutivo y paralelo. [1]

## 1.2 Metodología de desarrollo de Software

Las metodologías son una serie de procesos comunes que nos permiten las buenas prácticas para lograr los objetivos del proyecto, independientemente de cómo estos hayan sido diseñados.

Durante el desarrollo del software intervienen diferentes personas como el cliente quien realiza la solicitud del requerimiento, el analista de sistema quien se encarga de tomar los requerimientos al cliente y los hace llegar a los programadores encargados de la codificación y diseño del sistema.

Las fases que agrupan estos procesos y las más utilizadas son:

- **Análisis de especificaciones:** se define como la obtención de los requisitos de un producto software, obteniendo la información del cliente final.
- **Diseño:** en esta etapa se define el funcionamiento software de forma general.
- **Programación o Implementación:** se transfiere la información obtenida en las dos etapas anteriores a un lenguaje de programación.
- **Prueba:** consiste identificar del software y luego probarlo de forma integral, para así llegar al objetivo.
- **Mantenimiento:** consiste en aplicar métodos o técnicas para resolver los problemas utilizando las herramientas y procedimiento que son parte de la ingeniería del software con el propósito de la mejora de la calidad de los productos de software.



En base a lo anterior, podemos identificar las siguientes metodologías:

I. MODELOS DE PROCESO PRESCRIPTIVO

- **Modelo en cascada:** también llamado ciclo de vida clásico tiene un enfoque sistemático y secuencial, es el paradigma más antiguo sin embargo es muy útil en cuando el proyecto avanza en forma lineal.
- **Modelo iterativo e incremental:** es un modelo iterativo que a diferencia de los demás en cada iteración no entrega un producto terminado. Busca reducir el riesgo que surge entre las necesidades del usuario y el producto final.
- **Modelo evolutivo:** se caracterizan por la manera en la que permiten desarrollar versiones cada vez más completas del software.
  - **Prototipos:** se utiliza cuando el cliente define un conjunto de objetivos generales para el software pero que no identifican los requerimientos detallados, por lo que esta sería la mejor opción.
  - **Espiral:** es un modelo impulsado por el riesgo, se usa para guiar la ingeniería concurrente con participantes múltiples. Tiene dos características que lo distinguen primero un enfoque cíclico y un punto de referencia para identificar que la solución es factible.
- **Modelos concurrentes:** se puede representar en forma de esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas.

II. MODELO DE PROCESO ESPECIALIZADO

- **Desarrollo basado en componentes:** permite reutilizar piezas de código preelaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión.
- **Modelo de métodos formales:** se usa para referirse a cualquier actividad relacionada con representaciones matemáticas del software, incluyendo la



especificación formal de sistemas, análisis y demostración de la especificación, el desarrollo transformacional y la verificación de programas.

- **Desarrollo de software orientado a aspectos:** es un mecanismo que ayuda a resolver problemas de codificación en los requisitos, los cuales son un código disperso (scattered) y diseminado (tangled), que no se resuelven fácilmente usando el enfoque orientado a objetos. Se enfoca principalmente en la separación de intereses (separation of concerns) de un sistema para obtener una mejor modularización.

### III. EL PROCESO UNIFICADO

Es un proceso de software enfocado a casos de uso, se centra en la arquitectura, además de ser un método iterativo e incremental. Permite obtener los mejores rasgos y características de los modelos tradicionales del proceso de software manteniendo una comunicación con el cliente, describiendo su punto de vista a través de casos de uso. A continuación, podemos visualizar las diferentes etapas del proceso unificado en la Figura 1.3.

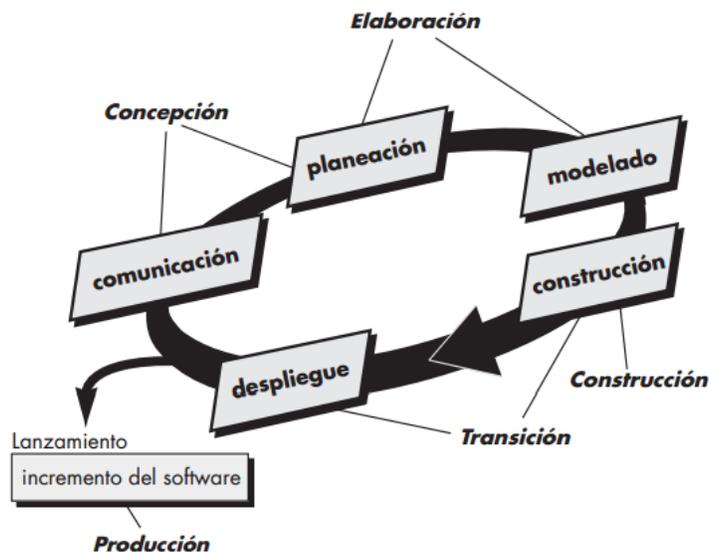


Figura 1.3 Proceso Unificado



Este enfoque utiliza el lenguaje unificado de modelado UML, y es una de las mejores y más utilizadas; para el análisis, implementación y documentación de sistemas orientados a objetos.

#### IV. MODELO DE PROCESO UNIFICADO RATIONAL (RUP)

El Proceso Unificado de Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software desarrollado por la empresa Rational Software de IBM.

Es un conjunto de metodologías adaptables al contexto y necesidades de cada organización, con documentación que apoya el ciclo de vida evolutivo incremental, además es un proceso de ingeniería de software que provee un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización. Su principal objetivo es asegurar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales dentro de un presupuesto y tiempo predecibles.

Las principales características de RUP son:

- **Dirigido por casos de uso:** es una técnica que se utiliza para la captura de requisitos por parte de los clientes/usuarios.
- **Centrado en la Arquitectura:** es la organización o estructura de todas las partes más relevantes del sistema, nos permite tener una visión común entre todos los involucrados en el proceso.
- **Interactivo e Incremental:** el equilibrio correcto entre los casos de uso y la arquitectura es muy parecido al equilibrio de la forma y la función en el desarrollo del producto lo cual se consigue con el tiempo.



La metodología RUP maneja seis principios clave:

- **Adaptación del proceso:** deberá adaptarse a las necesidades del cliente ya que es muy importante interactuar con él.
- **Balancear prioridades:** los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Gracias a este equilibrio se podrán corregir desacuerdos que puedan surgir en el futuro.
- **Colaboración entre equipos:** el desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes, resultados, etc.
- **Demostrar valor interactivamente:** los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.
- **Elevar el nivel de abstracción:** este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de trabajo (frameworks) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza que codificar para satisfacer de la mejor manera los requisitos y sin comenzar desde un principio pensando en reutilizar el código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas.
- **Enfocarse en la calidad:** el control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

RUP divide el proceso en cuatro fases el desarrollo del software, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Cada fase tiene definido un conjunto de objetivos y un punto de control específico.



1. **Fase de inicio:** es esta fase las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.
2. **Fase de elaboración:** en ella se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura.
3. **Fase de construcción:** se lleva a cabo la construcción del producto por medio de una serie de iteraciones. Para cada iteración se seleccionan algunos Casos de Uso, se refinan su análisis y diseño y se procede a su implementación y pruebas.
4. **Fase de transición:** en la fase de transición se pretende garantizar que se tiene un producto (versión beta) preparado para su entrega a la comunidad de usuarios. Como se puede observar en cada fase participan todas las disciplinas, pero dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

### 1.3 Lenguaje Unificado de Modelado (UML)

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

UML cuenta con varios tipos de diagramas como lo son los de estructura, comportamiento e interacción, los cuales muestran diferentes aspectos de las entidades representadas.



## Tipos de Diagramas de UML

### a) Estructura:

- De clases: Muestra elementos estáticos, tales como clases, tipos y sus contenidos y relaciones.
- De objetos: Presenta los objetos y sus relaciones en un punto del tiempo.
- De componentes: Representa los componentes que componen una aplicación, sistema o empresa.
- De estructura compuesta o estructura de composición: Representa la estructura de una clase, un componente o un caso de uso
- De paquetes: organiza los elementos de modelado en paquetes y las dependencias entre ellos, incluyendo importaciones y extensiones de paquetes.
- De despliegue físico: Muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos y la construcción interna puede ser representada por nodos o artefactos embebidos.

### b) Comportamiento:

- De casos de uso: Un diagrama que muestra las relaciones entre los actores y el sujeto (sistema), y los casos de uso.
- De actividades: Representa los procesos de negocios de alto nivel, incluidos el flujo de datos.
- De estado: Ilustra cómo un elemento se puede mover entre estados que clasifican su comportamiento, de acuerdo con disparadores de transiciones, guardias de restricciones y otros aspectos de los diagramas de Máquinas de Estados.

### c) Interacción:

- De secuencia: Representa una interacción, poniendo el foco en la secuencia de los mensajes que se intercambian, junto con sus correspondientes ocurrencias de eventos en las líneas de vida.



- De colaboración o de comunicaciones: es un diagrama que enfoca la interacción entre líneas de vida, donde es central la arquitectura de la estructura interna y cómo ella se corresponde con el pasaje de mensajes. La secuencia de los mensajes se da a través de un esquema de numerado de la secuencia.
- De tiempo: El uso más común es mostrar el cambio de estado de un objeto a lo largo del tiempo, en respuesta a los eventos o estímulos aceptados. Los eventos que se reciben se anotan, a medida que muestran cuándo se desea mostrar el evento que causa el cambio en la condición o en el estado.
- De interacción: Enfocan la revisión del flujo de control, donde los nodos son interacciones u ocurrencias de interacciones.

## 1.4 Sistemas de Bases de Datos

Una base de datos es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término base. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado [5].

La base de datos proporciona a los usuarios el acceso a información, ya sea para visualizarla, ingresarla o actualizarla, esto de acuerdo a los derechos de acceso otorgados de acuerdo a las funciones que realice el usuario.

Puede ser local, es decir que puede utilizarla sólo un usuario en un equipo, o puede ser distribuida, es decir que la información se almacena en equipos remotos y se puede acceder a ella a través de una red.



La principal ventaja de utilizar bases de datos es que múltiples usuarios pueden acceder a ellas al mismo tiempo.

### 1.4.1 Diseño de bases de datos

El diseño de bases de datos se simplifica cuando se utilizan modelos. Los modelos son abstracciones simplificadas de eventos y condiciones del mundo real.

Un modelo de base de datos es un conjunto de ideas lógicas utilizadas para representar la estructura de datos y las relaciones entre ellos dentro de la base de datos. Estos modelos se pueden agrupar en dos categorías: modelos conceptuales y modelos de ejecución.

- El **modelo conceptual** se enfoca en la naturaleza lógica de la representación de datos, incluyen el modelo Entidad-Relación (E-R) y el modelo orientado a objetos.
- Los **modelos de ejecución** hacen énfasis en como los datos están representados o en cómo se ejecutan las estructuras de datos, incluyen el modelo de datos jerárquico, el de base de datos de red, el modelo de base de datos relacional y el modelo de base de datos orientada a objetos

Los diseñadores de bases de datos utilizan un modelo de base de datos conceptual como base, el cual después lo utilizan para realizar el diseño de la base de datos con un modelo de base de datos de ejecución.



## 1.4.2 Modelos de base de datos

Existen cinco modelos de bases de datos, que se distinguen según cómo representan los datos almacenados:

1. **Modelo jerárquico:** Los datos se organizan jerárquicamente mediante un árbol invertido.
2. **Modelo de red:** Al igual que el modelo jerárquico, este modelo utiliza punteros hacia los datos almacenados. Sin embargo, no necesariamente utiliza una estructura de árbol invertido.
3. **Modelo relacional:** En su estructura los datos se almacenan en tablas de dos dimensiones (filas y columnas).
4. **Modelo entidad relación:** está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos.
  - **Entidad:** representa una “cosa” u "objeto" del mundo real con existencia independiente, es decir, se diferencia unívocamente de otro objeto o cosa, incluso siendo del mismo tipo, o una misma entidad.
  - **Atributos:** son las características o propiedades que definen o describen a una entidad.
  - **Relación:** describe cierta dependencia entre entidades o permite la asociación de las mismas.
5. **Modelo de orientación a objetos:** Los datos se almacenan como objetos, que son estructuras denominadas clases que muestran los datos que contienen y los campos son instancias de estas clases.



### 1.4.3 Normalización

El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Se dice que en un esquema de relación está en una determinada forma normal si se satisface un conjunto determinado de restricciones sobre los atributos. Mientras más restricciones existan, menor será el número de relaciones que las satisfagan. Así mientras más alta sea la forma en la que se encuentren los esquemas de relación, menores serán los problemas en el mantenimiento de la base de datos.

- **Primera Forma Normal (1FN):** Una relación está en la primera forma normal (o 1FN) si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos.
- **Segunda Forma Normal (2FN):** Una relación es, o pertenece a la segunda forma normal si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal.
- **Tercera Forma Normal (3FN):** Una relación es 3FN, si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no-clave. Nuevamente al analizar la descripción de dependencia funcional podemos apreciar que no existen atributos no clave, que sean dependientes de otro no clave en las relaciones.



## 1.5 Herramientas de desarrollo.

El desarrollo de software puede ser una actividad compleja y larga, por lo que las herramientas disponibles pueden reducir el estrés y aumentar el desempeño tanto de desarrolladores como de las aplicaciones resultantes.

En muchos tipos de aplicaciones y particularmente en las de interfaz web, podrían ser definidas como un juego de peticiones y respuestas. Un cliente requiere determinada acción (por ejemplo, a través de un enlace) y el servidor deberá, por medio de un procesamiento, resolver la demanda y devolver una respuesta. En general, podríamos decir que la aplicación cliente por excelencia es el navegador web, desde esta clase de programas accedemos a la interfaz del sistema para interactuar con él.

A continuación se mencionan algunas herramientas que nos permiten el desarrollo e implementación de Aplicaciones web.

### 1.5.1 PHP (PHP Hypertext preprocessor)

Es uno de los lenguajes de programación más utilizados en la actualidad para el desarrollo de sitios web [8].

Algunos de los factores por los que esta herramienta es muy utilizada son:

- Es de uso libre y gratuito: PHP, al igual que muchos otros lenguajes y entornos de programación, está amparado bajo el movimiento Open Source (código abierto), que permite a los programadores de aplicaciones poder sacar provecho de sus beneficios



de manera totalmente gratuita, sin la necesidad de pagar licenciamientos de uso ni actualizaciones. [2]

- Compatibilidad con diferentes Sistemas Operativos: MAC OS, Microsoft Windows, Unix, Unix / Linux, Unix / HP-UX, Unix / Mac OS X, Unix / OpenBSD y Unix / Solaris.
- Soporte a múltiples bases de datos: soportar DBase, Informix, Interbase/Firebird, Microsoft SQL Server, msql, MySQL, Oracle, Post-greSQL, SQLite y Sybase, entre otras.

### 1.5.2 MySQL

MySQL es un sistema de administración de bases de datos relacionales (SGBDR) rápido, robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente-servidor [3]. Utiliza el lenguaje SQL (Structured Query Language) que es el estándar de consulta a bases de datos a nivel mundial.

Está incluido en muchas herramientas y es compatible con muchos lenguajes de programación, además es una herramienta OpenSource, en particular gracias a su compatibilidad con el servidor de páginas Web Apache y el lenguaje de páginas dinámicas PHP.

### 1.5.3 Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.



Apache tiene características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Es usado principalmente para enviar páginas Web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones Web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor Web [4].

### 1.5.4 WAMP

WAMP es un entorno de desarrollo web para el sistema Windows, usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo;
- Apache, como servidor web;
- MySQL, como gestor de bases de datos;
- PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un WAMP permite servir páginas HTML a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web [5].

Tiene como funcionalidades:

- Administrar sus servicios de Apache y mysql
- Cambiar en línea / fuera de línea (dar acceso a todo el mundo o solamente localhost)
- Administrar la configuración de los servidores
- Acceder a sus registros
- Acceder a sus archivos de configuración
- Crear alias



### 1.5.5 HTML (HyperText Markup Language)

Es un lenguaje de descripción de hipertexto compuesto por una serie de comandos, marcas o etiquetas, también denominadas “Tags” que permiten definir la estructura lógica de un documento web y establecer los atributos del mismo (color del texto, contenidos multimedia, hipervínculos, etc.) [6]. De igual manera permite crear páginas web y para ello utiliza unos comandos o etiquetas que indican o marcan que se debe mostrar y de qué forma.

Los comandos siempre van incluidos entre los signos <> e insertados en el propio texto que compone el contenido de la página. Especifican sus estructura (las distintas parte de la página) y formato. Además, permiten la inserción de contenidos especiales como imágenes, videos, sonidos, etc.

El software necesario para trabajar con HTML son dos tipos de aplicaciones los cuales son los navegadores y editores.

### 1.5.6 CSS (Cascading Style Sheets)

CSS es un lenguaje de hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>". [7]





## Capítulo 2. Análisis y Diseño del sistema

---

### 2.1 Planteamiento del problema

En la actualidad las empresas cuentan con procesos administrativos que aún no han sido automatizados, ya sea porque no existe ningún software adecuado a esa necesidad o el precio es muy elevado.

La tecnología tiene la facilidad simplificar esas tareas o procesos que realizamos cotidianamente, haciéndolas más eficientes y seguros, brindando la posibilidad de manejar información en tiempo real.

Analizando las necesidades de una empresa, se pretende automatizar el proceso de Validación de adeudos de un empleado que deja de laborar en la misma y el registro de solicitudes de viaje (viáticos). Se parte del hecho que actualmente se lleva a cabo de manera manual imprimiendo un formato con información necesaria para cada uno de los trámites, este documento debe ser revisado y autorizado por las jefaturas de los distintos departamentos de una la empresa por ejemplo Finanzas, Sistemas, etc.

Tomando en cuenta todo lo anterior, se tienen varias dificultades durante el proceso por ejemplo el tiempo que tarda en validarse cada solicitud y la exposición de los datos que tienen un origen es delicado es más vulnerable.

Se propone un sistema de manejo de incidencias, que permita la automatización de dos procesos validación de adeudos y autorización de gastos de viaje, con la finalidad de facilitar la interacción entre varios departamentos de una empresa, así como, llevar una mejor administración y control sobre esos procesos.



## 2.2 Análisis de requerimientos

Se requiere diseñar un sistema web que permita a las diversas áreas de una empresa la interacción entre ellas en línea que les permita realizar los procesos de autorización de gastos y validación de adeudos, para este último solo se tomaran dos áreas de validación Sistemas y Finanzas.

El sistema debe contar con formularios que permitan el registro de solicitudes de gastos de viaje y solicitudes de validación de adeudos, comprobaciones de gastos y autorizaciones de gastos de viaje, permitiendo almacenar la información en una base de datos.

Los formularios de cada uno de los proceso deberá estar habilitado de acuerdo al perfil de la persona que está ingresando. En la especificación de requerimiento se describe cada uno de los perfiles y las opciones a las que podrá tener acceso.

## 2.3 Especificación de requerimientos

Tomado la información de los requerimientos del sistema, podemos identificar cuáles son las necesidades del cliente para la implementación de la aplicación web:

- Debe contar con una base de datos relacional la cual almacenara toda la información.
- Se debe contar con los siguientes perfiles para cada usuario:
  - ‡ **Administrador de RH:** Persona del área de recursos humanos encargado de crear las validaciones de adeudos y verificarlas.
  - ‡ **Jefe de área:** Encargado de validar que el empleado que será dado de baja no tenga ningún adeudo a la empresa, también será el encargado de aprobar o rechazar las peticiones para gastos de viaje.



‡ **Empleado:** Generará las peticiones de gastos de viaje que se enviarán a aprobación con jefe de área

‡ **Encargado de finanzas:** Valida que los empleados comprueban correctamente los gastos que tienen durante un viaje por parte de la empresa.

- El usuario podrá generar validaciones de adeudos a través de un formulario
- El usuario podrá ingresar a una interfaz que le permita de acuerdo a su perfil seleccionar las opciones validación de adeudos, aprobación, comprobación y solicitud de gastos de viaje
- El usuario podrá realizar sus comprobaciones de gastos a través del sistema
- El usuario podrá autorizar los gastos de viaje de un empleado
- Los cambios en el sistema solo los podrá realizar el administrador
- Almacenar la información para cada uno de los departamentos responsables de verificar si existe un adeudo del empleado.
- En la interfaz de validación de adeudos podrá visualizar un listado con las validaciones pendientes.
- El sistema deberá contar con 4 perfiles de usuario los cuales podrán tener opciones módulos diferentes.

### 2.3.1 Objetivos Generales

Como objetivos generales se tiene la creación de una aplicación web que permita la automatización de procesos de incidencias de personal (validación de adeudos y solicitud de gastos de viaje), facilitando la interacción entre los diferentes miembros de una empresa, permitiendo una mejor administración de las solicitudes y brindarles una herramienta capaz de almacenar la información.



### 2.3.2 Objetivos específicos

- Crear una aplicación web que permita interactuar con una base de datos para almacenar y consultar información.
- Generar una base de datos relacional para almacenar toda la información.
- La aplicación deberá contar con diferentes módulos basados en los perfiles identificados.
- Cada módulo debe tener la capacidad de insertar, actualizar y consultar la información.
- La interfaz deberá ser amigable para el usuario.

### 2.3.3 Metas

- Hacer una aplicación web con una interfaz que sea de fácil adopción para los usuarios del sistema.
- Contar con una base de datos relacional capaz de almacenar la información necesaria para cada solicitud.
- Deberá contar con dos módulos Validación de adeudos y solicitud de gastos de viajes
- La aplicación deberá ser fácil de usar.
- El sistema deberá ser capaz de la generación de reportes.
- Deberá ser capaz de identificar el perfil del usuario para habilitar las opciones correspondientes a cada uno.

### 2.3.4 Alcances

- Las funciones del sistema estarán sujetas a la información que proporcione el usuario.
- Los usuarios tendrán que ingresar al sistema para revisar las aprobaciones y validaciones pendientes.



### 2.3.5 Limitaciones

- Este sistema debe ser alimentado con la información de otros sistemas ya que no se comunica con ningún sistema externo.
- Heredará las restricciones propias de un sistema desarrollado en PHP y MySQL.

## 2.4 Antecedentes

En la actualidad podemos encontrar diferentes aplicaciones que se utilizan para la administración de incidencias de personal por ejemplo *FlexiServer* [8], *NOM2001* [9], por mencionar algunas. Estas aplicaciones fueron desarrolladas para la administración del recurso humano de una empresa por ejemplo, manejo de nómina y registro de algunas incidencias.

Estas aplicaciones están desarrolladas para la administración de recursos humanos pero no toman en cuenta incidencias como las consideradas en este proyecto. Aunado a esto los costos por la adaptación de estos sistemas para incluir las opciones que no están incluidas son muy elevados, por otro lado existen empresas que no pueden contar con un sistema especializado para el manejo de las incidencias y llevan el control en papel.

Como ejemplo agregamos dos aplicaciones que manejan las incidencias del personal pero no cuentan con módulos para la solicitud de viáticos o validación de adeudos del personal cuando deja de laborar.



- **FlexiServer:** software para asistencia y productividad, en la Figura 2.1 se muestra algunas de las opciones que ofrece.

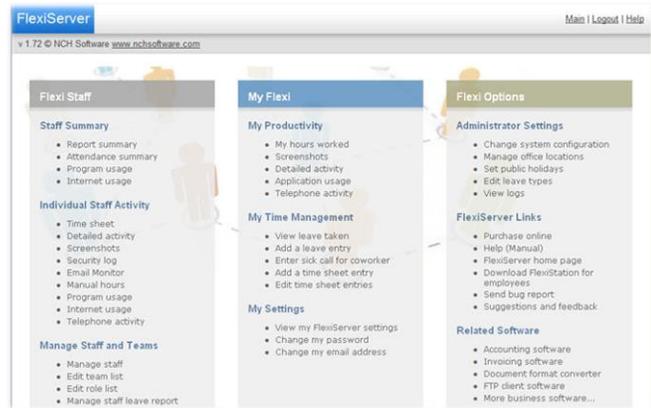


Figura 2.1 FlexiServer

- **NOM2001:** software para administración de Recursos Humanos, en la Figura 2.2 se muestra algunos de los módulos que contiene la herramienta

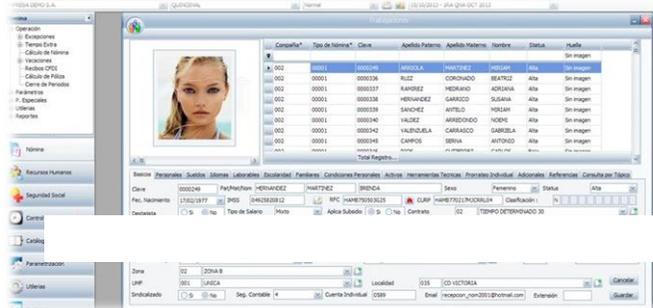


Figura 2.2 NOM2001

También podemos encontrar sistemas que se han desarrollado en la facultad destinados a la administración del personal como Sistema Para El Control Administrativo De Personal Del Instituto De Ciencias Sociales y Humanidades [10] y Sistema web de control de asistencias de personal administrativo y académico [11].

Con este proyecto se pretende atacar estos puntos, a través del desarrollando del sistema web para pequeñas o medianas empresas que permita mejorar este tipo de proceso que actualmente realizan en papel.



## 2.5 Metodologías, Técnicas y Herramientas

Para el análisis y diseño de la aplicación web se utilizó la teoría de ingeniería de software aplicando la metodología RUP, siguiendo las fases descritas en la misma.

- Fase de inicio (Capítulo II)
- Fase de elaboración (Capítulo II)
- Fase de Desarrollo (Capítulo III)
- Fase de Cierre (Capítulo III)

Con la ayuda de UML [12], se representaron algunos artefactos recomendados en las fases de la metodología (Diagrama de casos de uso, Diagrama de secuencia, Diagrama de clase).

La base de datos está basada en el modelo relacional, la cual permite definir la estructura e integridad de la Base.

Como herramientas para el desarrollo y la implementación de la aplicación se utilizó PHP como lenguaje de programación, MySQL como manejador de base de datos y WAMP como servidor local.

## 2.6 Modelo funcional

Para el desarrollo del sistema, se identificó el comportamiento a través del Diagrama de Casos de uso, permitiéndome tener una idea más clara de los requerimientos planteados en el análisis.



Identifiqué cuatro actores, los cuales fueron utilizados en el sistema como perfiles de usuario.

- ‡ **Administrador de RH:** es el encargado de hacer las solicitudes de validación de adeudos y verificar los que ya fueron validados.
- ‡ **Jefe de área:** se encarga de la autorización de gastos de viaje y solo para las áreas de Finanzas e IT realizaran validaciones de adeudos.
- ‡ **Empleado:** se encargara de enviar sus solicitudes de autorización de gastos de viaje y las comprobaciones de los mismos.
- ‡ **Encargado de finanzas:** el encargado de validar las comprobaciones de gastos de viaje y validará si existe algún adeudo de otro tipo por parte del empleado.

Con la información anterior se procedió a la identificación de los casos de uso, a continuación enlisto los que se identificaron y utilizaron para el desarrollo de la aplicación.

- ✂ **Solicitar gastos de viaje:** En este caso de uso interviene el actor Empleado, que de acuerdo a las especificaciones realiza solicitudes de gastos de viaje, que son enviadas a validación con su Jefe de departamento.
- ✂ **Autorizar solicitud de viaje:** En este caso de uso interviene el actor Jefe de área, después de ejecutar el caso de uso Solicitar gastos de viaje por el empleado el Jefe de área procederá a validar los montos solicitados por el empleado y procederá a Autorizar o rechazar los gastos.
- ✂ **Comprobar gastos de viaje:** En este caso de uso interviene el actor Empleado, que de acuerdo a las especificaciones después de pasar por la autorización del Jefe de área el empleado tendrá que comprobar los gastos generados durante su viaje lo que implica que tiene un adeudo con la empresa.
- ✂ **Verificar comprobaciones de viaje:** En este caso de uso interviene el actor Encargado de Finanzas, después de ejecutar el caso de uso Comprobar gastos de



viaje por el empleado, el encargado de finanzas procederá a validar los montos comprobados por el empleado y procederá a cotejarlos con la información entregada físicamente y procederá a validar o no dichas comprobaciones.

- ✂ **Consultar adeudos:** En este caso de uso interviene el actor Encargado de finanzas, que de acuerdo a las especificaciones tendrá un apartado donde podrá consultar a los empleados que tenga algún adeudo con la empresa. Para el desarrollo de esta aplicación solo se podrán consultar los adeudos de gastos de viaje.
  
- ✂ **Solicitar validación de adeudo:** En este caso de uso interviene el actor Administrador RH (Recursos Humanos), que de acuerdo a las especificaciones realiza solicitudes validación de adeudos, que son enviadas a revisión con los jefes de departamento de IT y Finanzas. Para el desarrollo de esta aplicación solo se realizara para esas dos áreas ya que son las más importantes de una empresa.
  
- ✂ **Validar información de adeudo:** En este caso de uso interviene el actor Jefe de área (IT y Finanzas), después de ejecutar el caso de uso Solicitar validación de adeudo por el Administrador de RH el Jefe de área procederá a validar si el empleado tiene algún adeudo con la empresa que pertenezca a su área. Por ejemplo Laptop, gastos de viaje, etc.

Después de realizar una breve descripción de las funciones de los casos de uso y el actor o actores que intervienen en el mismo se procedió a la realización del diagrama de casos de uso que se muestra en la Figura 2.3.



Figura 2.3 Diagrama de casos de uso

Para cada uno de los casos de uso se generó la documentación de casos de uso, donde se describen las funciones y los requerimientos de cada uno.

En la Tabla 2.1 podemos visualizar la descripción, precondiciones, flujo de eventos y excepciones en el caso de uso Solicitar validación de adeudo, se colocó esta tabla ya que es la parte que inicia con el proceso de validación.



<b>No. de Caso de uso:</b>	<b>02</b>	<b>Nombre de Caso de Uso:</b>	<b>Solicitar validación de adeudo</b>
<b>Actores</b>	Administrador de RH		
<b>Descripción</b>	El caso se inicia un empleado dejará de laborar en la empresa, se requiere la validación de algunas áreas en específico para identificar que no deja ningún adeudo con la empresa		
<b>Precondiciones</b>	Ninguno		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El administrador de RH ingresa al sistema</li> <li>2. Selecciona la opción Solicitar validación de adeudos</li> <li>3. Llena el formulario con los datos del empleado a dar de baja fecha de baja y motivo de baja.</li> <li>4. El administrador de RH presiona el botón enviar a validación</li> <li>5. El sistema valida que se llenaron todos los campos requeridos</li> <li>6. El sistema crea los registros para las áreas que validaran</li> <li>7. El sistema guarda la información en la base de datos.</li> </ol>		
<b>Flujos alternos</b>	Ninguno		
<b>Excepciones</b>	<p>E1. La información del formulario es incorrecta o contiene caracteres extraños, el sistema envía mensaje de error</p> <p>E2. Los datos no pudieron almacenarse el sistema, manda un mensaje de error en el sistema</p>		

Tabla 2.1 Documentación de Caso de uso Solicitar validación de adeudo



En la Tabla 2.2 podemos visualizar la descripción, precondiciones, flujo de eventos y excepciones en el caso de uso Solicitar gastos de viaje.

No. de Caso de uso:	08	Nombre de Caso de Uso:	Solicitar gastos de viaje
<b>Actores</b>	Empleado		
<b>Descripción</b>	El caso inicia cuando el empleado desea solicitar autorización para que le asignen un presupuesto para gastos de viaje por parte de la empresa		
<b>Precondiciones</b>	Ninguna		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El empleado ingresa al sistema y selecciona la opción de Solicitud de gastos de viaje</li> <li>2. Llena todos los campos requeridos en el formulario como monto solicitado, departamento, fechas, etc.</li> <li>3. El empleado presión el botón enviar</li> <li>4. El sistema guarda la información en la base de datos</li> </ol>		
<b>Flujos alternos</b>	Ninguna		
<b>Excepciones</b>	<p>E1. Los datos colocados en el formato no corresponden a los del campo</p> <p>E2. Los datos no pudieron almacenarse el sistema, manda un mensaje de error en el sistema</p>		

**Tabla 2.2 Documentación de Caso de uso Solicitar gastos de viaje**

En el **Anexo 1** se muestra la documentación de los casos de uso identificados en la Figura 2.3 y en el **Anexo 2** se colocó la información sobre las excepciones.



## 2.7 Modelo conceptual

Para identificar los conceptos significativos del proyecto y tener una representación de las entidades y conceptos, se realizó el glosario de términos, diccionario de modelo y la realización de casos de usos los cuales se describen a continuación.

### 2.7.1 Glosario de términos

- **Validación de gastos:** acción que se realiza por parte de una empresa para validar los gastos realizados durante una salida por parte la empresa.
- **Validación de adeudos:** acción que se realiza por parte de una empresa para validar los adeudos con la empresa por ej. Equipo de cómputo, equipo telefónico, etc.
- **Adeudo:** deuda que adquiere el empleado con la empresa.
- **Comprobación de gastos:** Acción que se realiza un empleado al comprobar los gastos generados durante un viaje por parte de una empresa, esto puede ser a través de un ticket o factura.
- **Gastos de viaje:** Cantidad de dinero que se gasta durante un viaje por parte de la empresa.
- **Bajas de personal:** Acción que se realiza cuando un empleado termina su relación con la empresa.

### 2.7.2 Diccionario del modelo

En las Tablas 2.3, 2.4, 2.5 y 2.6 se define el contenido cada clase Empleado, Área, Adeudo y Gastos de viaje indicando los atributos que tendrá cada una

<b>Nombre:</b>	<b>Empleado</b>
<b>Definición de trabajo:</b>	Se define como la clase que contendrá los atributos de un empleado como el número de empleado, Nombre, RFC, Fecha de ingreso, etc.

Tabla 2.3 Clase Empleado



<b>Nombre:</b>	<b>Área</b>
<b>Definición de trabajo:</b>	Se define como él área la clase que llevará la información de los departamentos que tiene la empresa y contendrá la información de los jefes de área

Tabla 2.4 Clase Área

<b>Nombre:</b>	<b>Adeudo</b>
<b>Definición de trabajo:</b>	Contendrá información de los adeudos de un empleado con un departamento determinado, se podrá consultarlos y validarlos

Tabla 2.5 Clase Adeudo

<b>Nombre:</b>	<b>Gastos de viaje (viáticos)</b>
<b>Definición de trabajo:</b>	Persona que administra la aplicación y que realiza funciones distintas a un usuario normal, tendrá un método para autorizarlos

Tabla 2.6 Clase Gastos de viaje

## 2.8 Modelo de Diseño

La realización de un caso de uso nos permite describir en términos de colaboración de objetos a un caso de uso, especificando las clases que deben construirse para implementarlo.

Se puede representar usando diagramas de secuencia, colaboración y clase control y entidad.

A continuación se muestran los diagramas correspondientes al caso de uso Solicitar validación de adeudo.



## 2.8.1 Caso de uso Solicitar validación de adeudo

Para tener un mejor análisis del funcionamiento del sistema procedí a realizar los siguientes diagramas.

Diagrama Clase y control Figura 2.4, se utilizó para identificar las clases que se utilizaran para la interacción entre el usuario y el sistema del caso de uso Solicitar validación de adeudo.

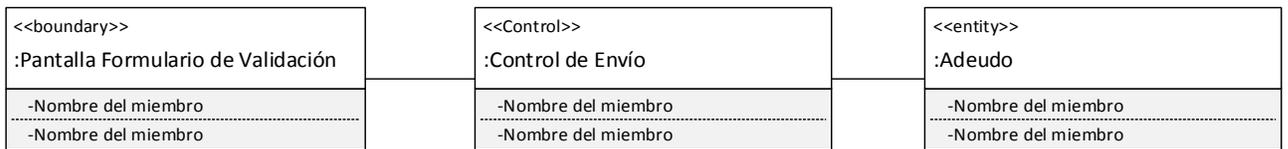


Figura 2.4 Diagrama de Clase Pantalla, Control y Entidad

Diagrama de secuencia y colaboración Figura 2.5 y 2.6 se muestra cual es la secuencia y de qué manera interactúan las clases del caso de uso Solicitar validación de adeudo

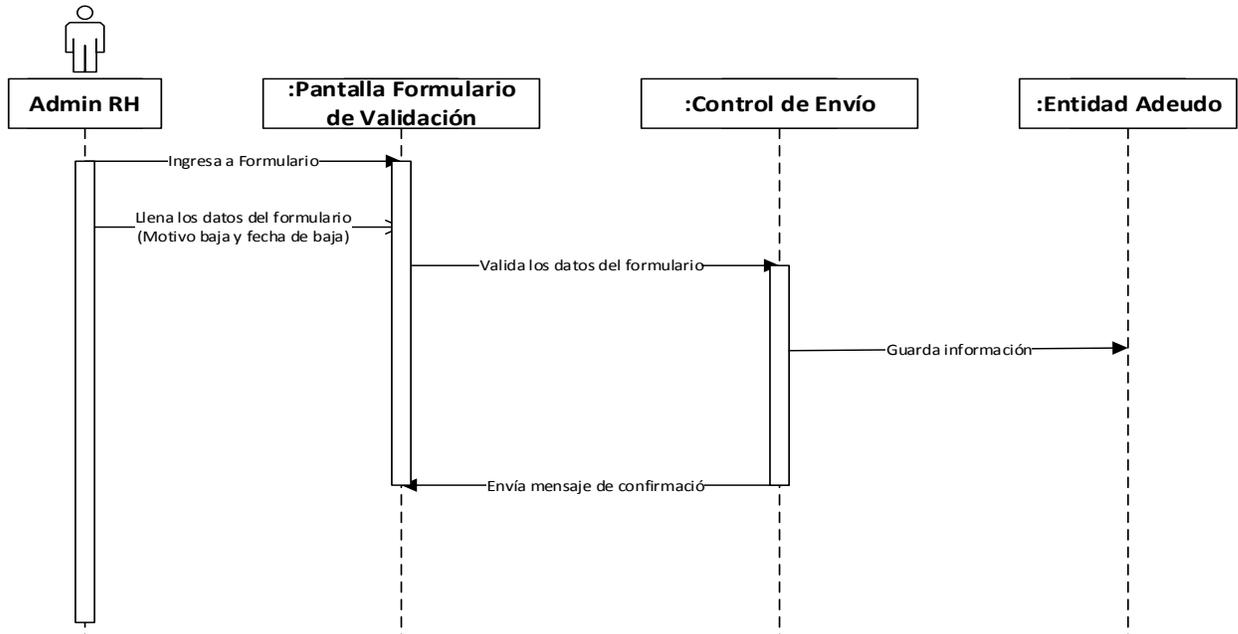


Figura 2.5 Diagrama de Secuencia

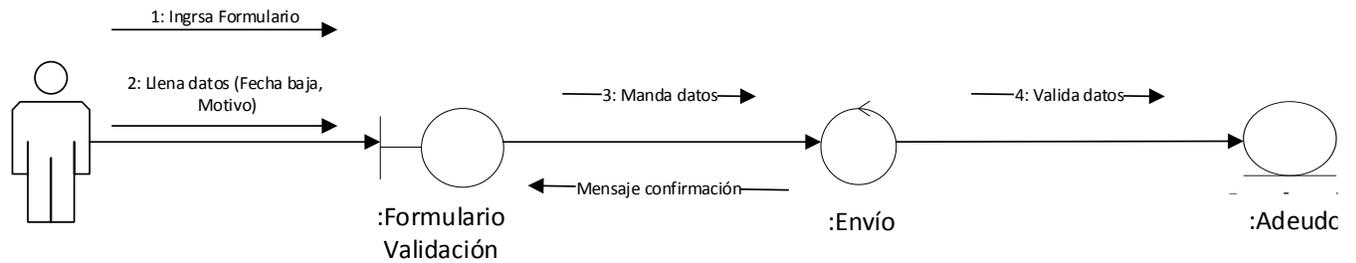


Figura 2.6 Diagrama de Colaboración

## 2.9 Diseño de Base de Datos

Basándose en el diseño conceptual se procedió a realizar la representación de los datos y las relaciones entre ellos. El modelo relacional proporciona una estructura de los datos que consiste en un conjunto de relaciones con objeto de representar la información que nos interesa del mundo real. [13].

Durante el proceso se identificaron las siguientes Entidades:

- Empleado
- Departamento
- Viáticos
- Gastos

A continuación se muestra en la Figura 2.7 de manera gráfica las relaciones identificadas entre cada una de las entidades.

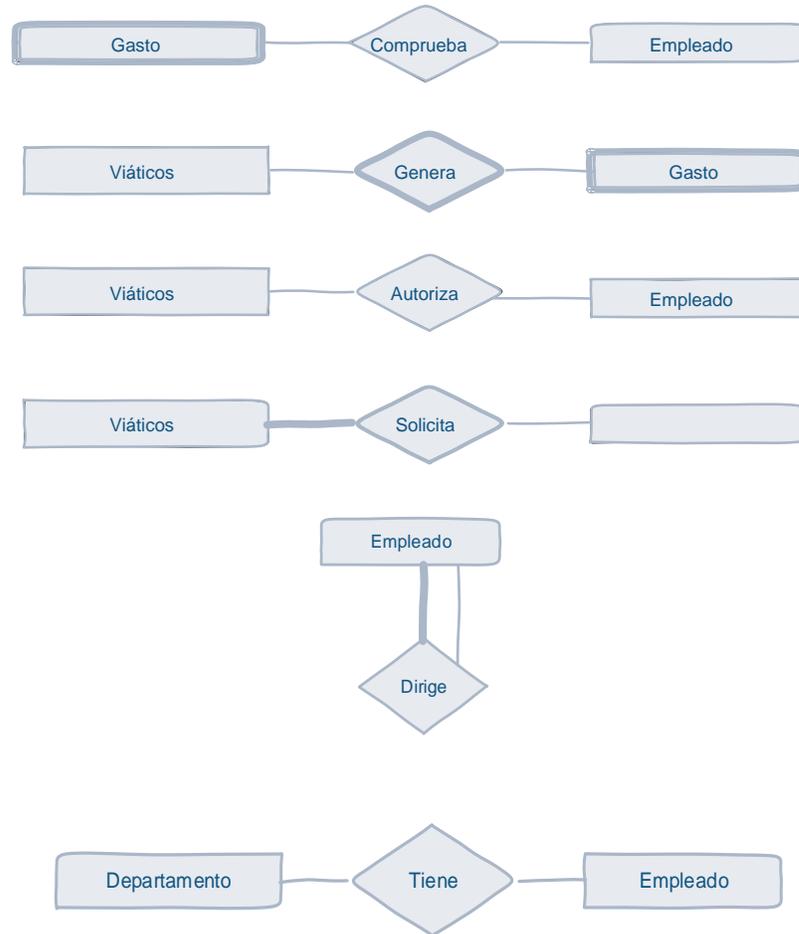


Figura 2.7 Relaciones entre entidades

Después de identificar las entidades y relaciones entre ellas, procedí a generar el **Diagrama Entidad – Relación** que se muestra en la Figura 2.8, el cual me permitió tener un esquema que describe más a detalle la base de datos.

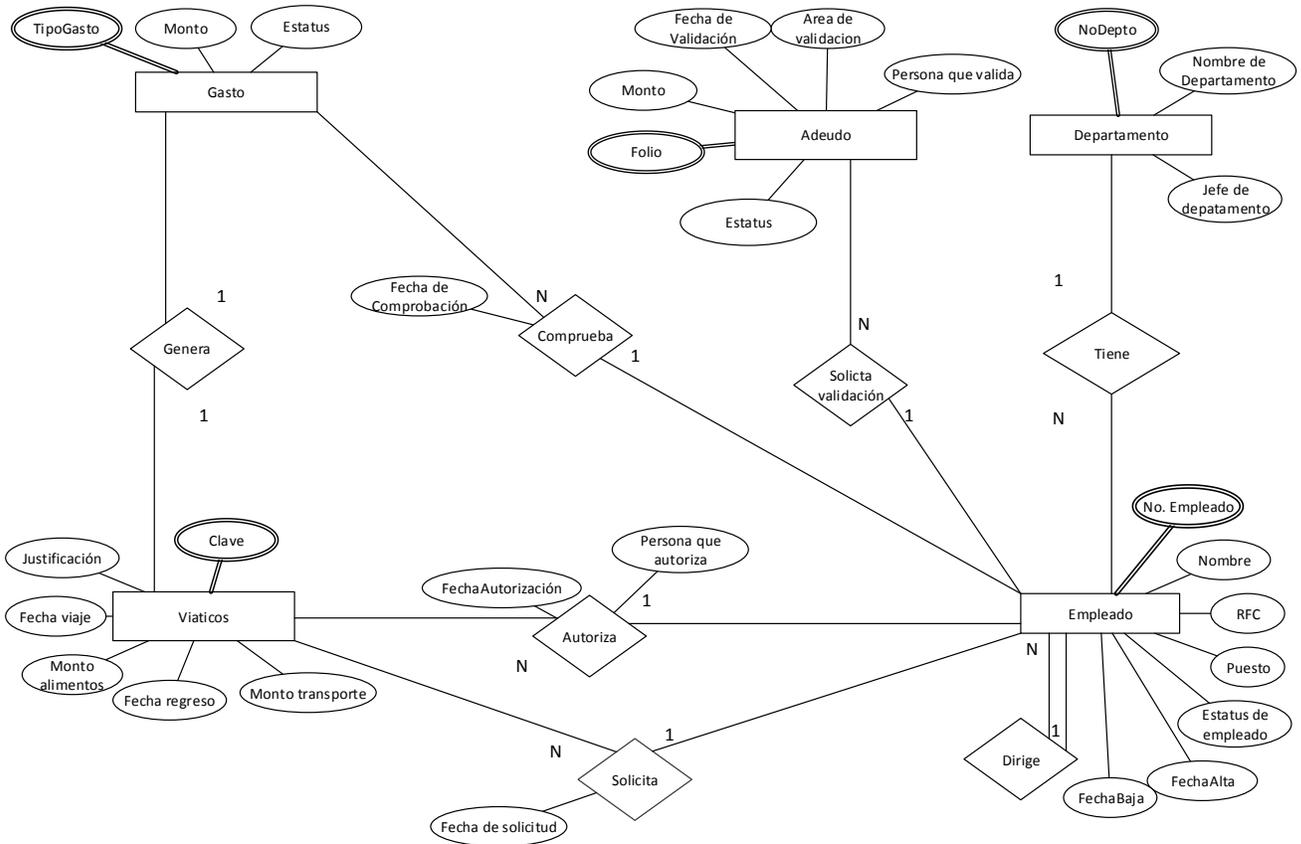


Figura 2.8 Diagrama Entidad - Relación

Una vez realizado el diagrama Entidad – Relación procedí a realizar el esquema lógico, convertí las relaciones en tablas. Se lograron identificar las siguientes tablas con sus atributos tomando las siguientes consideraciones:

- Eliminar las relaciones de muchos a muchos
- Eliminar las relaciones entre tres o más entidades
- Eliminar las relaciones recursivas
- Eliminar las relaciones con atributos
- Eliminar los atributos multievaluados
- Revisar las relaciones de uno a uno
- Eliminar las relaciones redundantes



Con la información obtenida con la normalización de la base de datos, se procedió a realizar el mapeo de las tablas colocando las llaves primarias y foráneas, así como los atributos de cada una cómo podemos visualizarlo en las tablas 2.7 a la 2.12.

<b>Adeudo</b>						
<u>Folio Adeudo</u>	Número de Empleado	Estatus	Monto	Fecha de validación	Área de validación	Jefe que valida
	FK					FK

Tabla 2.7 Tabla Adeudo normalizada

<b>Autoriza</b>		
<u>Folio de validación</u>	Fecha de Autorización	Persona que Autoriza
		FK

Tabla 2.8 Tabla Autoriza normalizada

<b>Departamento</b>	
<u>Número de departamento</u>	Nombre de departamento

Tabla 2.9 Tabla departamento normalizada

<b>Empleado</b>									
<u>Número de empleado</u>	Nombre del empleado	Puesto	RFC	Jefe	Fecha de alta	Fecha de baja	Motivo de baja	Número de departamento	Estatus
				FK				FK	

Tabla 2.10 Tabla Empleado normalizada

<b>Viáticos</b>											
<u>Folio</u>	Número de empleado	Justificación	Fecha de viaje	Fecha de regreso	Fecha de solicitud	Monto	Estatu s	Monto transporte	Monto Alimentos/ Hospedaje	Tipo de viaje	Departamento
	FK										

Tabla 2.11 Tabla Viáticos normalizada

<b>Gasto</b>					
Tipo de Gasto	Clave	Número de Empleado	Monto	Estatus	Fecha comprobación
	FK	FK			

Tabla 2.12 Tabla gasto normalizada



En base a lo anterior se muestra en las tablas 2.13 a la 2.18 los atributos que componen cada una de las tablas y el tipo de información que almacenará.

**Tabla 2.13 Tabla Adeudo**

Columna	Tipo
<b>Folio Adeudo</b>	int(7)
Número de Empleado	int(7)
Estatus	varchar(30)
Monto	int(11)
Fecha de validación	date
Área de validación	int(11)
Jefe que valida	int(11)

**Tabla 2.14 Tabla Autoriza**

Columna	Tipo
Folio de validación	int(11)
Fecha de Autorización	date
Persona que Autoriza	int(11)

**Tabla 2.15 Tabla Departamento**

Columna	Tipo
Número de departamento	int(5)
Nombre de departamento	varchar(50)



**Tabla 2.16 Tabla Empleado**

Columna	Tipo
Número de empleado	int(8)
Nombre del empleado	varchar(50)
Puesto	varchar(50)
RFC	varchar(15)
Jefe	int(7)
Fecha de alta	date
Fecha de baja	date
Motivo de baja	text
Número de departamento	int(5)
Estatus	text

**Tabla 2.17 Tabla Gasto**

Columna	Tipo
Tipo de Gasto	varchar(20)
Clave	int(11)
Número de Empleado	int(7)
Monto	float
Estatus	varchar(20)
Fecha comprobación	date

**Tabla 2.18 Tabla Viáticos**

Columna	Tipo
Folio	int(11)
Número de empleado	int(7)
Justificación	text
Fecha de viaje	date
Fecha de regreso	date
Fecha de solicitud	date
Monto	float
Estatus	varchar(20)
Monto transporte	float
Monto Alimentos/Hospedaje	float
Tipo de viaje	text
Departamento	text



Finalmente se diseñó el Diagrama de Clases de la Figura 2.9, el cual nos permite ver la relación que existe entre las diferentes tablas del sistema.

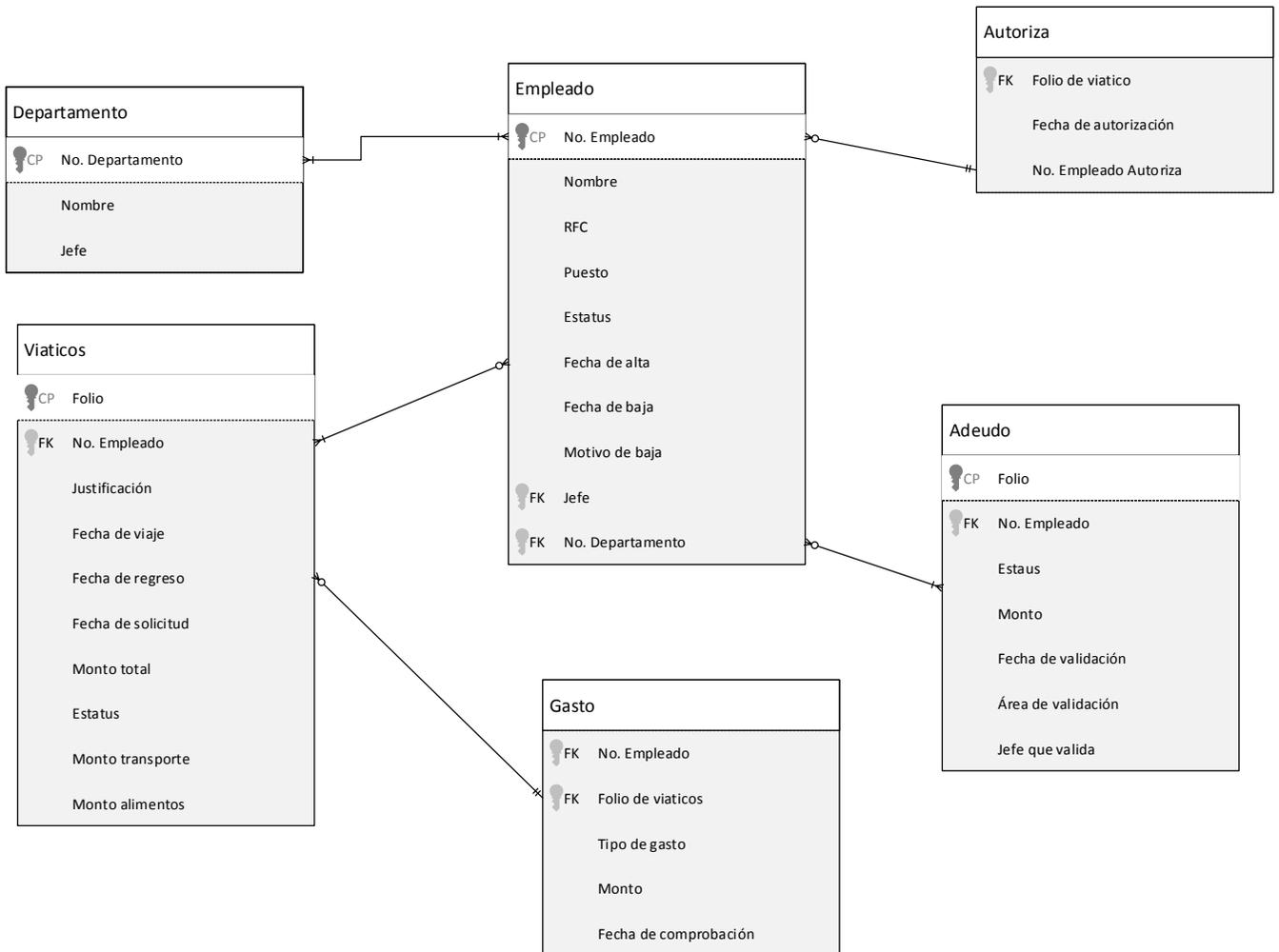
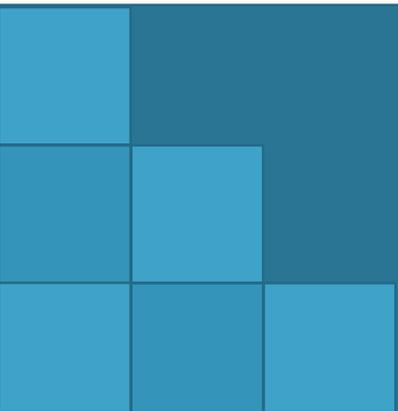


Figura 2.9 Diagrama de clases



# **CAPITULO III**

## **Implementación y pruebas del sistema**



## Capítulo 3. Implementación del sistema.

---

Finalmente, después del análisis y diseño del sistema, así como la implementación de la base de datos se procedió desarrollo del sistema haciendo uso de las herramientas mencionadas con anterioridad PHP como lenguaje de programación, como manejador de base de datos MySQL, Dreamweaver para el diseño de páginas web y como servidor se utilizó WAMP Server

### 3.1 Interfaz de acceso para el Usuario

Para el desarrollo de la aplicación Web, se pensó en una interfaz sencilla y amigable, con acceso solo interno ya que la información es solo para los empleados de la empresa.

En base al análisis que se realizó en los capítulos anteriores, se identificaron cuatro perfiles de usuario que tendrán diferentes opciones de acuerdo a las actividades que puede realizar.

- **Administrador de RH**
- **Jefe de área**
- **Empleado**
- **Encargado de finanzas**

Para identificar cada perfil de usuario se utilizó la información de la sesión del usuario, ingresando al sistema con su número de empleado y contraseña asignada, esta información se almaceno en una tabla diferente por cuestiones de seguridad.



El usuario ingresará al sitio <http://localhost/ControlRH> en la página principal como se muestra en la Figura 3.1, podrá encontrar la información de la empresa como la identidad y los servicios que brinda, y la opción de Login que le dará el acceso a los diferentes formularios.



Figura 3.1 Pantalla de inicio de sistema

Al dar clic en Login los enviará a un formulario como se muestra en la Figura 3.2 el cual les solicitará un usuario y un password. El empleado deberá ingresar su número de empleado proporcionado por el área de recursos humanos y la contraseña que asignó en su momento, al dar clic en el botón **Ingresar** se verifica el tipo de usuario registrado en la base de datos y se redirecciona al menú que le corresponde de acuerdo a su perfil.



HOME NUESTRA IDENTIDAD NUESTROS SERVICIOS

**Ingresa tus datos**

Usuario  
3400147

Password  
.....

Ingresar

Figura 3.2 Pantalla de Inicio de sesión

En el caso de que el empleado tenga el perfil de Administrador de Recursos humanos lo enviará al menú que se muestra en la Figura 3.3.

- Solicitar validación de adeudos.
- Solicitar gastos de viaje
- Comprobar gastos de viaje

HOME LOGOUT

Módulo de Recursos Humanos

Módulo para empleados

>>Solicitar validación de adeudo

>>Solicitar gastos de viaje

>>Comprobar gastos de viaje

Figura 3.3 Menú para el personal de Recursos Humanos.



Si el empleado que ingreso tiene el perfil de jefe de área le habilitará las opciones que se muestran en la Figura 3.4.

- Solicitar gastos de viaje
- Comprobar gastos de viaje
- Autorizar gastos de viaje
- Validar adeudos de bajas



Figura 3.4 Menú para el personal de Jefes de área.

Para el perfil de Encargado de Finanzas le habilitará las opciones que se muestran en la Figura 3.5.

- Solicitar gastos de viaje
- Comprobar gastos de viaje
- Validar comprobaciones
- Consultar comprobaciones

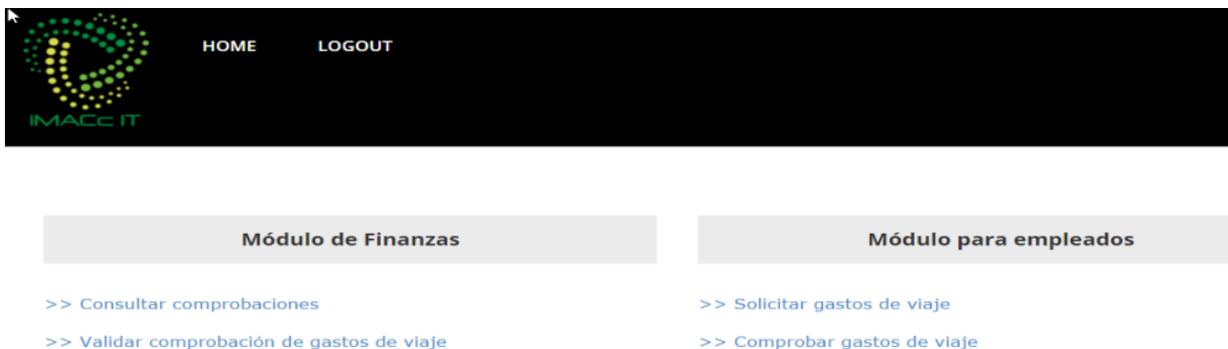


Figura 3.5 Menú para el personal de Finanzas



Finalmente el perfil de Empleado solo tendrá habilitadas las opciones de la Figura 3.6. Estas opciones también aparecen en los perfiles anteriores que todos tiene el rol de empleado en algún momento.

- Solicitar gastos de viaje
- Comprobar gastos de viaje



Figura 3.6 Menú para Empleados.

## 3.2 Funciones del sistema

A continuación se describe cada una de las funciones colocadas en los menús de cada perfil.

### 3.2.1 Solicitar validación de adeudos.

Después de que el usuario ingresa al sistema y se identifica el perfil de Recursos Humanos, el usuario podrá solicitar la validación de adeudos de un empleado con los jefes de los departamentos de finanzas e IT. Para esto llenará un formulario de validación de adeudo, para esto deberá ingresar primero el número del empleado que desee validar como se muestra en la figura 3.7.



Figura 3.7 Consultar información de Empleado

El sistema tomará la información enviada y consultará la tabla de empleado para mostrar su información, así el encargado de finanzas solo colocará la información del motivo y fecha de baja que se almacenará en la misma tabla. En la figura 3.8 se muestra un ejemplo de la información que se muestra en el formulario.

**Ingrese la información para validar los adeudos del empleado:**

ID:	3405068
Nombre:	PATRICIA AGUILAR DE ANGEL
Puesto:	AUXILIAR
Fecha de alta:	13/11/2006
Jefe Inmediato:	REBECA AVILA MERAZ
Departamento:	Finanzas
Fecha de baja:	<input type="text" value="dd/mm/aaaa"/>
Motivo de baja:	<input type="text"/>

Figura 3.8 Ejemplo de validación de adeudo



Al dar clic en Enviar, almacenará la información como se mencionó en la tabla empleado y a su vez creará un nuevo registro en la tabla adeudo para que pueda ser validado por los jefes de departamento correspondiente.

### 3.2.2 Validar adeudos de bajas.

El usuario ingresa al sistema y se identifica el perfil de Jefe de área y se valida que pertenezca al departamento de TI o Finanzas, ya que solo ellos podrán realizar la validación de adeudos de un empleado.

A continuación el usuario ingresa a la opción de Validar adeudos de bajas y se muestra el listado de las validaciones que tiene que realizar como en la figura 3.9 donde solo tendrá que ingresar el folio de la validación y el monto del adeudo.

<b>Folio:</b>	8
<b>No. Empleado:</b>	3404552
<b>Nombre de empleado:</b>	IVAN AGUILA ARTECHE
<b>Folio:</b>	11
<b>No. Empleado:</b>	3405386
<b>Nombre de empleado:</b>	ANGELICA AVELINO BADILLO
<b>Folio:</b>	13
<b>No. Empleado:</b>	3405068
<b>Nombre de empleado:</b>	PATRICIA AGUILAR DE ANGEL
<b>Folio:</b>	15
<b>No. Empleado:</b>	3404552
<b>Nombre de empleado:</b>	IVAN AGUILA ARTECHE
<b>Folio:</b>	17
<b>No. Empleado:</b>	1
<b>Nombre de empleado:</b>	Juana Perez Perez
<b>Coloque el folio a validar</b>	<input type="text"/>
<b>Monto del adeudo:</b>	<input type="text"/>
	<input type="button" value="Validar"/>

Figura 3.9 Formulario Validación de adeudos

Cabe mencionar que el monto lo determina el jefe de departamento en base a la información obtenida por otros medios, ya que el sistema no interactúa con otras aplicaciones.



Al dar clic en Validar actualizará la información en la tabla de adeudos indicando los montos que se deberá descontar al empleado.

### 3.2.3 Solicitar gastos de viaje

De acuerdo a lo identificado en la etapa de análisis y diseño del sistema, a través del formulario de solicitud de gastos de viaje se enviará a autorización con su jefe de departamento indicando las fechas en que viajará, tipo de viaje, y montos para transporte, alimentación y hospedaje.

Para este formulario no se requiere tener un perfil específico ya que todos pueden solicitar gastos de viaje. Tienen que ingresar a la opción de solicitud de gastos de viaje y colocar la información requerida como se muestra en la figura 3.10 y al dar clic en enviar almacenará la información en la tabla de viáticos asignando un folio automático con el que su Jefe de departamento podrá autorizar.

No. Empleado	<input type="text"/>		
Fecha de salida:	<input type="text" value="dd/mm/aaaa"/>	Fecha de regreso:	<input type="text" value="dd/mm/aaaa"/>
Motivo de viaje:	<input type="text"/>		
Hospedaje y/o alimentos:	<input type="text"/>	Transporte:	<input type="text"/>
Viaja en:	<input type="radio"/> Nacional	<input type="radio"/> Extranjero	
<input type="button" value="Enviar"/>			

Figura 3.10 Solicitud de gastos de viaje



### 3.2.4 Autorizar gastos de viaje

Al ingresar a la opción de autorización de gastos de viaje, le mostrará las autorizaciones pendientes de su departamento con la información básica de cada solicitud. El jefe de departamento solo tendrá que colocar el folio de la solicitud para autorizarla.

Se colocó de esta manera para hacer lo menos complicado las autorizaciones de estas solicitudes y a su vez mostrar la información pendiente sin necesidad de ingresar a otro formulario de consulta. En la figura 3.11 podemos observar un ejemplo de Autorización de gastos.

Folio	Nombre de empleado	Fecha de salida	Fecha de regreso	Tipo de viaje	Moto total
17	MARIA GUADALUPE AYALA DEOLARTE	2015-11-30	2015-12-04	Nacional	7500
18	MARIA GUADALUPE AYALA DEOLARTE	2015-11-30	2015-12-04	Nacional	7500
19	JUAN CARLOS AVILA SANCHEZ	2015-10-02	2015-10-05	Nacional	400
20	JUAN CARLOS AVILA SANCHEZ	2015-10-02	2015-10-05	Nacional	400

Coloque el folio a validar

Figura 3.11 Formulario de autorización de gastos de viaje

### 3.2.5 Comprobar gastos de viaje

Cuando el empleado regresa de su viaje, deberá registrar los montos generados durante su viaje como gastos de hospedaje, alimentos, transporte u otros que se llegaran a dar.



Para eso el usuario deberá ingresar a la opción de comprobación de gastos de viaje, en el formulario deberá agregar la información solicitada en el formulario que se muestra en la figura 3.12

		FOLIO:	<input type="text"/>
No. Empleado	<input type="text"/>	Monto total a comprobar:	<input type="text"/>
Monto de Hospedaje:	<input type="text"/>	Monto de Transporte:	<input type="text"/>
Monto de Alimentos:	<input type="text"/>	Otros gastos:	<input type="text"/>
<input type="button" value="Enviar"/>			

Figura 3.12 Formulario de comprobación de gastos de viaje

### 3.2.6 Validar comprobaciones

El encargado de finanzas tendrá que validar las comprobaciones del usuario y cotejarlas con los documentos que entregue físicamente, para eso deberá ingresar a la opción de Validación de comprobaciones.

Para ver el listado de gasto del empleado deberá colocar el número de empleado y dar clic en buscar, como lo podemos visualizar en la figura 3.13.

**Escriba el número de empleado:**

No. Empleado:

Figura 3.13 Consulta de empleado



A continuación el sistema consultará el listado de gastos que comprobó el usuario para proceder a su validación. En la figura 3.14 podemos ver un ejemplo de este caso.



Folio	Fecha de salida	Fecha de regreso	Tipo de viaje	Moto total
6	2015-10-26	2015-10-29	Nacional	6000

Figura 3.14 Comprobaciones de empleado

En encargado de finanzas solo tendrá que colocar el folio y dar clic en validar como en la figura 3.15 para indicar que la comprobación se realizó de manera correcta.



Coloque el folio a validar

Validar

Figura 3.15 Validación de comprobación

### 3.2.5 Consultar comprobaciones

El empleado de finanzas deberá ingresar a la opción Consultar comprobaciones y lo enviará a una página que le solicitará el número de empleado y dar clic en buscar, como lo podemos visualizar en la figura 3.13.



### 3.3 Conclusiones.

Puedo concluir que la ingeniería de software es una herramienta que nos permite el desarrollo eficaz de sistemas como este proyecto y garantizar la calidad del mismo. Por lo anterior podemos decir que el proyecto cumple con los requerimientos establecidos en el Capítulo 2 siguiendo las diferentes etapas de la Metodología RUP.

Este proyecto le permitirá a las medianas empresas reducir en gran cantidad de tiempo los procesos que realizaba manualmente en el manejo de validaciones de adeudos de personal y solicitudes de gastos de viaje.

A su vez, nos abre la oportunidad de unirse con otros sistemas, que nos permita darle a la empresa una herramienta más completa. Esto se podrá lograr gracias a las herramientas utilizadas durante la implementación ya que son de uso libre.

Finalmente podemos mencionar que se concluye este proyecto dejándolo como base para la generación de un sistema más completo. Permittiéndonos el agregar más procesos que actualmente se realizan en papel, logrando así un beneficio para la empresa tanto en tipo como en dinero.



### 3.3.1 trabajo a futuro

En este proyecto como trabajo a futuro se pueden realizar diferentes configuraciones que nos permitirían tener una herramienta más completa, por ejemplo se pretende la implementación de notificaciones vía correo electrónico lo cual permitiría mejorar el tiempo de atención a las solicitudes ya que se tendría un aviso en tiempo real al ingresar la solicitud.

Lo mencionado anteriormente solo es una de las opciones que se tiene, se piensa también en la conexión del sistema con otras bases de datos que permitan la interacción entre diversos sistemas que nos permitan obtener la información alojada en ellas y a su vez la actualización de la misma.

La seguridad también forma una parte muy importante dentro del proyecto, por lo que se pretende el manejo de encriptación de información permitiéndonos una mayor seguridad en los datos que se manejan.



## Bibliografía

---

- [1] C. Mateu, «Desarrollo de aplicaciones web,» 2004.
- [2] R. S. Pressman, Ingeniería del Software, Mc Graw Hill.
- [3] «PHP,» [En línea]. Available: <http://php.net/>.
- [4] T. Boronczyk, PHP y MySQL, Anaya Multimedia, 2009.
- [5] «Apache Server Project,» [En línea]. Available: [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html).
- [6] «WampServer,» [En línea]. Available: <http://www.wampserver.com/en/>.
- [7] B. Kennedy y C. Musciano, HTML la guía completa, Mc Graw Hill & O'reilly, 1999.
- [8] «Libros web,» [En línea]. Available: [http://librosweb.es/libro/css/capitulo\\_1.html](http://librosweb.es/libro/css/capitulo_1.html).
- [9] «FlexiServer, software para asistencia y productividad,» [En línea]. Available: <http://www.nchsoftware.com/flexi/es/index.html>.
- [10] «NOM2001,» [En línea]. Available: <http://www.nom2001.com.mx/interiores/productos.php>.
- [11] I. V. Alarcón, «Sistema Para El Control Administrativo De Personal Del Instituto De Ciencias Sociales Y Humanidades,» 2013.
- [12] E. A. Galvez, «Sistema web de control de asistencias de personal administrativo y academico,» 2014.
- [13] J. Arlow, Programación UML 2, Anaya Multimedia, 2006.
- [14] L. A. C. S. D. C. C. M. G. G. C. M. E. y. O. P. M. Rafael Camps Paré, «Software Libre,» *Base de Datos*, 2005.
- [15] C. E. c. Arriagada y A. C. a. Chamorro Malagueño, «Introducción al modelamiento de bases de datos y SQL básico para Bibliotecarios. (Spanish).,» 2008.
- [16] A. Silberschatz y H. F. Korth, Fundamentos de Bases de Datos, Mc Graw Hill, 2002.
- [17] C. Larman, UML y patrones : una introducción al análisis y diseño orientado a objetos y al proceso unificado, Pearson Educación, 2003.



- [18] S. R. Schach, Analisis Y Diseño Orientado A Objetos Con Uml Y El Proceso Unificado, Mc-Graw Hill, 2005.
- [19] A. Gutierrez, A. Alcocer, J. Bobadilla y S. Alonso, Html Dinamico, Asp Y Javascript Atraves De Ejemplos, Alfaomega Ra-Ma, 2000.



## Anexos.

A continuación se agrega la información adicional que complementa la información de los casos de uso del sistema de validación de adeudos.

### Anexo 1. Documentación de casos de uso.

En las tablas anexo 1.1 a la 1.7, se muestra la información de los casos de uso identificados en el Capítulo 2, indicando la descripción Precondiciones, actores, Flujo de eventos, Flujos alternos y las excepciones de cada uno.

No. de Caso de uso:	01	Nombre de Caso de Uso:	Ingresar al sistema
<b>Actores</b>	Administrador de RH, Jefe de área, Empleado y Encargado de Finanzas		
<b>Descripción</b>	El caso inicia cuando un usuario ingresar al sistema		
<b>Precondiciones</b>	Debe estar dado de alta en el sistema		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa al sistema</li> <li>2. Coloca su usuario y contraseña proporcionada</li> <li>3. Envía la información para validación</li> <li>4. El sistema verifica que el usuario este dado de alta</li> <li>5. El sistema identifica el perfil del usuario</li> <li>6. El sistema muestra las funcionalidades que le corresponden al usuario</li> </ol>		
<b>Flujos alternos</b>	<ol style="list-style-type: none"> <li>1. El sistema identifica el perfil de Administrador de RH</li> <li>2. El sistema identifica el perfil del usuario es de Jefe de área</li> <li>3. El sistema identifica el perfil de Encargado de finanzas</li> <li>4. El sistema identifica el perfil de Empleado</li> </ol>		
<b>Excepciones</b>	<p>E1. El usuario no es valido</p> <p>E2. El usuario no recuerda la contraseña</p> <p>E3. El usuario ingreso caracteres no válidos</p>		

Tabla anexo 1. 1 Ingresar al sistema



No. de Caso de uso:	02	Nombre de Caso de Uso:	Solicitar validación de adeudo
<b>Actores</b>	Administrador de RH		
<b>Descripción</b>	El caso se inicia un empleado dejará de laborar en la empresa, se requiere la validación de algunas áreas en específico para identificar que no deja ningún adeudo con la empresa		
<b>Precondiciones</b>	Ninguno		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El administrador de RH ingresa al sistema</li> <li>2. Selecciona la opción Solicitar validación de adeudos</li> <li>3. Llena el formulario con los datos del empleado a dar de baja fecha de baja y motivo de baja.</li> <li>4. El administrador de RH presiona el botón enviar a validación</li> <li>5. El sistema valida que se llenaron todos los campos requeridos</li> <li>6. El sistema crea los registros para las áreas que validaran</li> <li>7. El sistema guarda la información en la base de datos.</li> </ol>		
<b>Flujos alternos</b>	Ninguno		
<b>Excepciones</b>	<p>E1. La información del formulario es incorrecta o contiene caracteres extraños, el sistema envía mensaje de error</p> <p>E2. Los datos no pudieron almacenarse el sistema, manda un mensaje de error en el sistema</p>		

Tabla anexo 1. 2 solicitar validación de adeudo

No. de Caso de uso:	03	Nombre de Caso de Uso:	Validar información de adeudo
<b>Actores</b>	Jefe de área		
<b>Descripción</b>	El caso se inicia un empleado dejará de laborar en la empresa, se requiere la validación de que no deja ningún adeudo con la empresa con un departamento en específico		
<b>Precondiciones</b>	Se debe ejecutar el CU Solicitar validación de adeudo		



<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El Jefe de área ingresa al sistema en la opción Validar Adeudos</li> <li>2. El sistema despliega el listado de empleados a validar</li> <li>3. El Jefe de área llena la información del monto y concepto del adeudo</li> <li>4. El jefe de área presiona el botón enviar</li> <li>5. El sistema valida que se llenaron todos los campos requeridos</li> <li>6. El sistema actualiza el estatus de la validación de Pendiente a Validada</li> <li>7. El sistema guarda la información en la base de datos</li> </ol>
<b>Flujos alternos</b>	Ninguno
<b>Excepciones</b>	E1. Los datos no pudieron almacenarse el sistema, manda un mensaje de error en el sistema

**Tabla anexo 1. 3 Validar información de adeudo**

No. de Caso de uso:	04	Nombre de Caso de Uso:	Autorizar gastos de viaje
<b>Actores</b>	Jefe de área		
<b>Descripción</b>	El caso se inicia cuando un empleado debe realizar un viaje por parte de la empresa, debe solicitar la aprobación de un presupuesto para sus gastos		
<b>Precondiciones</b>	Se ejecuta el CU Solicitar gastos de viaje		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El Jefe de área ingresa al sistema en la opción Autorizar gastos de viaje</li> <li>2. El Jefe de área identifica al empleado que está realizando la solicitud</li> <li>3. El Jefe de área cambia el estatus de la solicitud a Autorizada</li> <li>4. El jefe de área presiona el botón guardar</li> <li>5. El sistema guarda la información en la base de datos</li> </ol>		
<b>Flujos alternos</b>	1. El jefe de área no autoriza la solicitud		
<b>Excepciones</b>	E1. Los datos no pudieron almacenarse el sistema, manda un mensaje de error en el sistema		

**Tabla anexo 1. 4 Autorizar gastos de viaje**



<b>No. De Caso de uso:</b>	<b>05</b>	<b>Nombre de Caso de Uso:</b>	<b>Consultar adeudos</b>
<b>Actores</b>	Encargado de finanzas		
<b>Descripción</b>	El caso se inicia cuando el jefe de área o encargado de finanzas desea revisar que un empleado no tiene adeudos de viajes anteriores.		
<b>Precondiciones</b>	Ninguna		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de Consultar adeudos</li> <li>2. El usuario debe colocar el número de empleado a validar</li> <li>3. El sistema consulta la información en la base de datos, en el campo de adeudos</li> <li>4. El sistema regresa el monto de adeudo, si es cero solo envía un mensaje de no adeudo.</li> <li>5. El usuario cierra la ventana de consulta</li> </ol>		
<b>Flujos alternos</b>	Ninguno		
<b>Excepciones</b>	Ninguna		

Tabla anexo 1. 5 Consultar adeudos.

<b>No. de Caso de uso:</b>	<b>07</b>	<b>Nombre de Caso de Uso:</b>	<b>Solicitar gastos de viaje</b>
<b>Actores</b>	Empleado		
<b>Descripción</b>	El caso inicia cuando el empleado desea solicitar autorización para que le asignen un presupuesto para gastos de viaje por parte de la empresa		
<b>Precondiciones</b>	Ninguna		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El empleado ingresa al sistema y selecciona la opción de Solicitud de gastos de viaje</li> <li>2. Llena todos los campos requeridos en el formulario como monto solicitado, departamento, fechas, etc.</li> <li>3. El empleado presión el botón enviar</li> <li>4. El sistema guarda la información en la base de datos</li> </ol>		
<b>Flujos alternos</b>	Ninguna		
<b>Excepciones</b>	<p>E1. Los datos colocados en el formato no corresponden a los del campo</p> <p>E2. Los datos no pudieron almacenarse el sistema, manda un mensaje de error en el sistema</p>		

Tabla anexo 1. 6 Solicitar gastos de viaje



<b>No. de Caso de uso:</b>	<b>08</b>	<b>Nombre de Caso de Uso:</b>	<b>Comprobar gastos de viaje</b>
<b>Actores</b>	Empleado		
<b>Descripción</b>	El caso inicia cuando el empleado desea comprobar sus de gastos de viaje, valida que la información colocada el formulario corresponda a la que entrego físicamente.		
<b>Precondiciones</b>	Ejecutar el CU Autorizar gastos de viaje		
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El empleado ingresa al sistema en la opción comprobación de gastos</li> <li>2. Ingresa los montos de la comprobación de gastos de viaje</li> <li>3. Presiona el botón enviar</li> <li>4. El sistema guarda la información en la base de datos</li> </ol>		
<b>Flujos alternos</b>	Ninguno		
<b>Excepciones</b>	<p>E1. Los datos colocados en el formato no corresponden a los del campo</p> <p>E2. Los datos no pudieron almacenarse el sistema, manda un mensaje de error en el sistema</p>		

Tabla anexo 1. 7 Comprobar gastos de viaje

## Anexo 2. Escenarios de casos de uso

Para complementar la información de los casos de uso de las tablas anexo 1.1 a la 1.7, se muestra agrego la información de los flujos alternos se identificaron en cada uno de ellos, como lo podrá verificar en las tablas anexo 2.1 a la 2.7.

**Nombre del escenario:** Perfil de Administrador de RH

Actores:	Administrador de RH
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. Elisa ingresa a la aplicación web</li> <li>2. Coloca el usuario número de empleado y la contraseña</li> <li>3. El sistema identifica que es el usuario número de empleado</li> <li>4. El sistema habilita las opciones de Validación de adeudos, Solicitar gastos de viaje y comprobación de gastos de viaje</li> </ol>

Tabla anexo 2. 1 Caso de uso Ingresar al sistema flujo alternativo 1



**Nombre del escenario: Perfil de Jefe de área**

Actores:	Jefe de área
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. Elsa ingresa a la aplicación web</li> <li>2. Coloca su número de empleado y la contraseña</li> <li>3. El sistema identifica al empleado y verifica el puesto que tiene asignado</li> <li>4. El sistema identifica que es Jefe de área y habilita las opciones de Validación de adeudo, autorización de gastos de viaje, solicitar gastos de viaje y comprobar gastos de viaje</li> </ol>

**Tabla anexo 2. 2 Caso de uso Ingresar al sistema flujo alternativo 2**

**Nombre del escenario: Perfil de Empleado**

Actores:	Empleado
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. Paola ingresa a la aplicación web</li> <li>2. Coloca su número de empleado y la contraseña</li> <li>3. El sistema identifica al empleado y verifica el puesto que tiene asignado</li> <li>4. El sistema notifica habilita las opciones de comprobación de gastos de viaje y solicitud de gastos de viaje</li> </ol>

**Tabla anexo 2. 3 Caso de uso Ingresar al sistema flujo alternativo 3**

**Nombre del escenario: Perfil de Encargado de finanzas**

Actores:	Encargado de Finanzas
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. Mirna ingresa a la aplicación web</li> <li>2. Coloca el usuario Finanzas y la contraseña</li> <li>3. El sistema identifica que es el usuario Finanzas</li> <li>4. El sistema habilita las opciones de Verificar comprobaciones de viaje, consulta de adeudos, comprobación de gastos de viaje y solicitud de gastos de viaje</li> </ol>

**Tabla anexo 2. 4 Caso de uso Ingresar al sistema flujo alternativo 4**



**Nombre del escenario:** Solicitar validación de adeudo

Actores:	Administrador de RH
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. Juan ingresa al sistema en la opción de Baja</li> <li>2. Selecciona la opción solicitar validar adeudos</li> <li>3. Llena el formulario con los datos del empleado fecha de baja y motivo de baja</li> <li>4. Presiona el botón enviar a validación</li> <li>5. El sistema valida que se llenaron todos los campos requeridos</li> <li>6. El sistema crea los registros para las áreas Sistemas y Finanzas</li> <li>7. El sistema guarda la información en la base de datos</li> </ol>

**Tabla anexo 2. 5 Caso de uso Solicitar validación de adeudo**

**Nombre del escenario:** El jefe de área no autoriza la solicitud

Actores:	Jefe de área
Flujo de eventos:	<ol style="list-style-type: none"> <li>1. Juan ingresa al sistema en la opción Autorizar gastos de viaje</li> <li>2. Identifica al empleado que está realizando la solicitud</li> <li>3. Consulta los adeudos del empleado</li> <li>4. Modifica el estatus de la solicitud a Autorizada</li> <li>5. Presiona el botón guardar</li> <li>6. El sistema guarda la información en la base de datos</li> </ol>

**Tabla anexo 2. 6 Caso de uso Autorizar solicitud de viaje**



**Nombre del escenario:** El usuario no reporto las mismas cantidades

<i>Actores:</i>	Encargado de finanzas
<i>Flujo de eventos:</i>	<ol style="list-style-type: none"> <li>1. Juan ingresa al sistema y selecciona la opción de Verificar comprobaciones</li> <li>2. En el formulario coloca el número de empleado por ej. 17962</li> <li>3. Verifica los montos de los documentos entregados físicamente y se percata que no correspondan a los registrados</li> <li>4. Coloca el monto de la comprobación correcto</li> <li>5. Juan presiona el botón enviar</li> <li>6. El sistema actualiza el monto autorizado restando la cantidad comprobada</li> <li>7. El sistema guarda la información en la base de datos</li> </ol>

**Tabla anexo 2. 7 Caso de uso Verificar comprobación de viaje**



## Anexo 3. Diagramas de Secuencia, Colaboración y Clase Pantalla, Control y Entidad

Para cada caso de uso se generó un Diagrama Secuencia, Colaboración y Clase pantalla, control y entidad.

El Caso de uso Solicitar Validación de adeudo, podemos visualizar los diagramas en las Figuras anexo 3.1 Diagrama de clase, pantalla, control y entidad, 3.2 Diagrama de secuencia y 3.3 Diagrama de colaboración

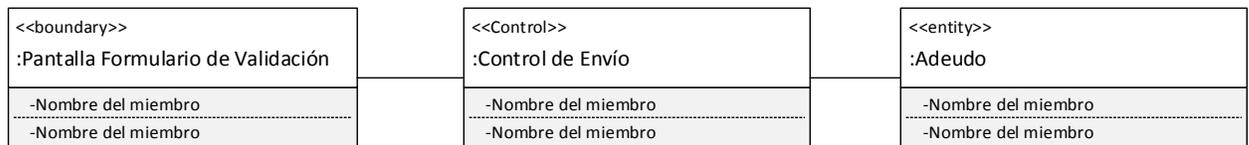


Figura anexo 3. 1 Diagrama de Clase Pantalla, Control y Entidad

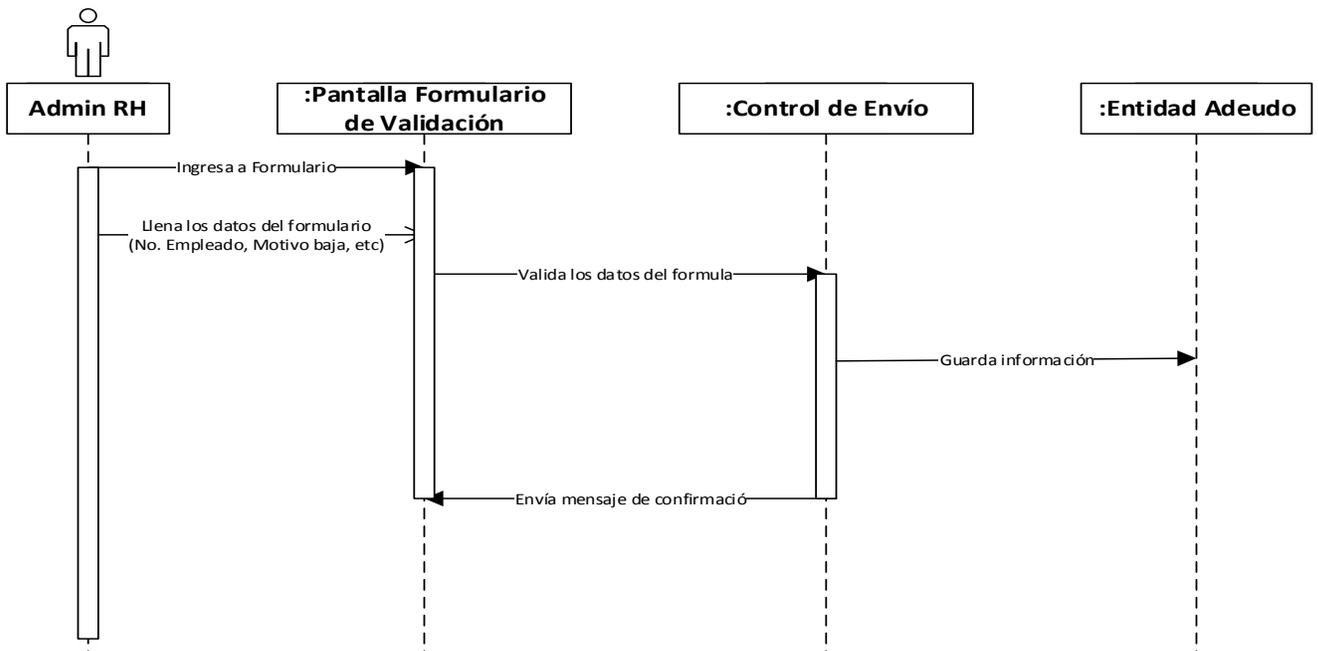


Figura anexo 3. 2 Diagrama de Secuencia

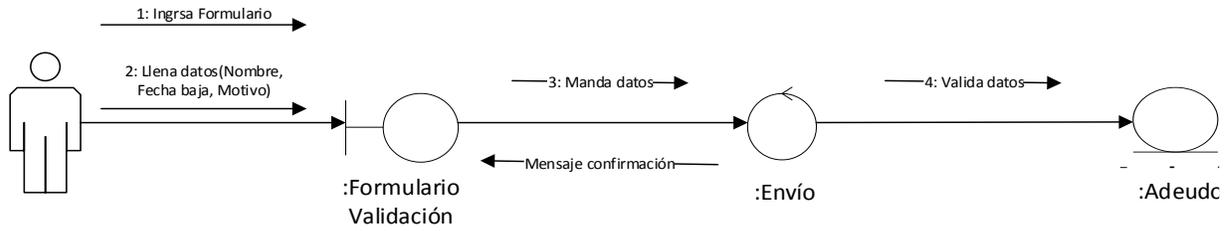


Figura anexo 3. 3 Diagrama de Colaboración

El Caso de uso Consultar Adeudos, podemos visualizar los diagramas en las Figuras anexo 3.4 Diagrama de clase, pantalla, control y entidad, 3.5 Diagrama de secuencia y 3.6 Diagrama de colaboración

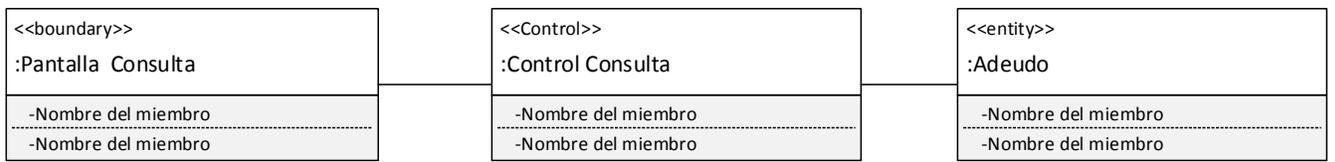


Figura anexo 3. 4 Diagrama de Clase Pantalla, Control y Entidad

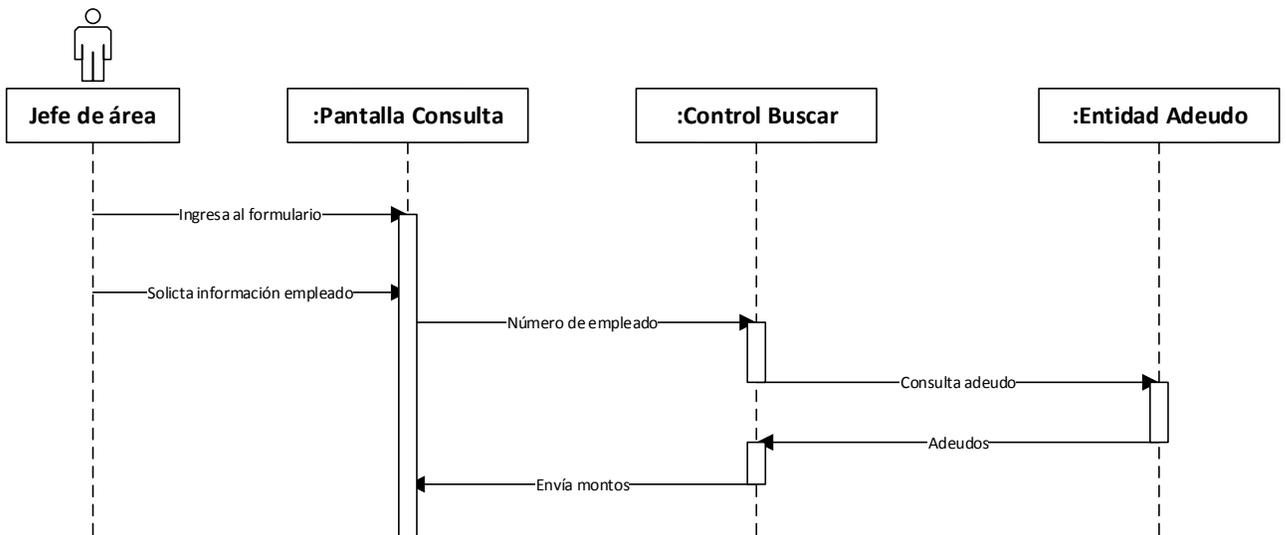


Figura anexo 3. 5 Diagrama de Secuencia

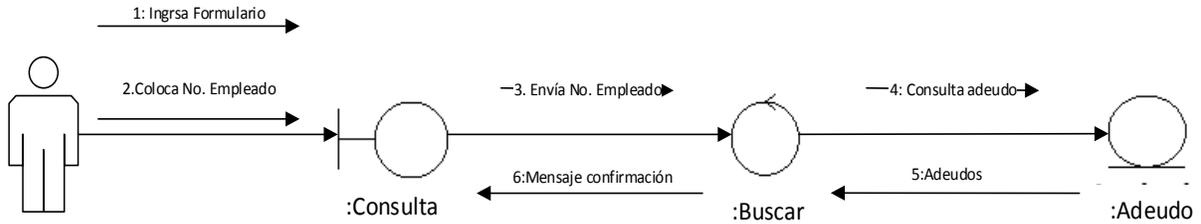


Figura anexo 3. 6 Diagrama de Colaboración

El Caso de uso Validar información de adeudo, podemos visualizar los diagramas en las Figuras anexo 3.4 Diagrama de clase, pantalla, control y entidad, 3.5 Diagrama de secuencia y 3.6 Diagrama de colaboración

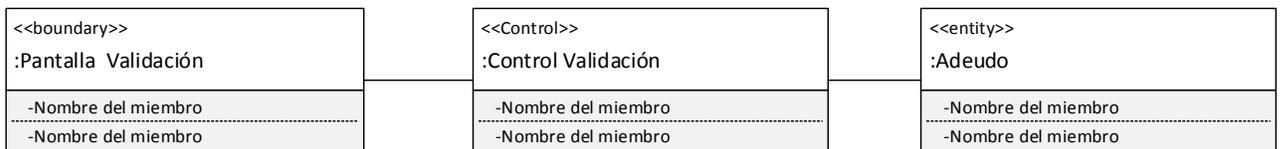


Figura anexo 3. 7 Diagrama de Clase Pantalla, Control y Entidad

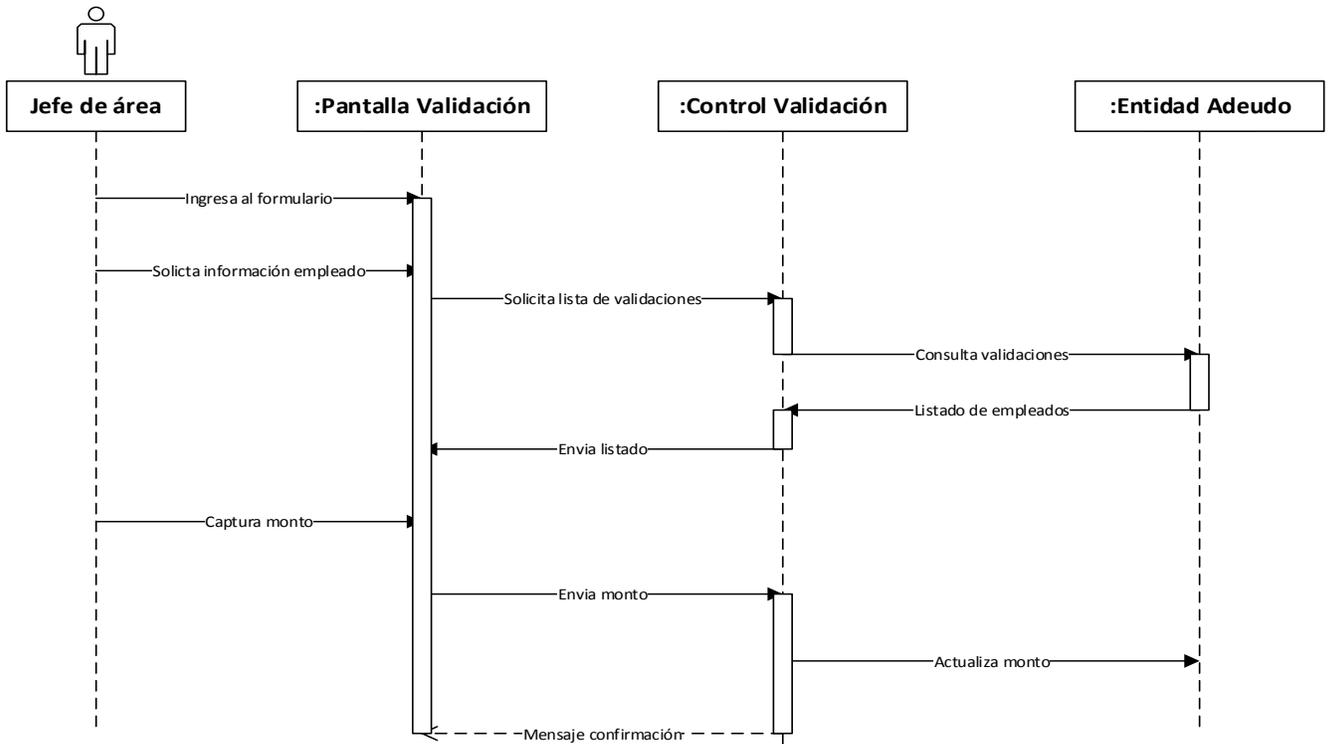


Figura anexo 3. 8 Diagrama de Secuencia



Figura anexo 3. 9 Diagrama de Colaboración