

TD-0051-04-01

003.7  
Cas  
S  
TD 0051-04-01

**INSTITUTO SUPERIOR POLITÉCNICO JOSÉ ANTONIO ECHEVERRÍA  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA DE INGENIERÍA INFORMÁTICA**



# **SISTEMA DE CONTROL DE INVERSIONES**

**INFORME TÉCNICO PARA OPTAR POR EL TÍTULO DE  
INGENIERO INFORMÁTICO**

Autor:  
Mario Antonio Castelló Polo

Tutor:  
Ing. Yadenis Piñeiro Pérez

Ciudad de La Habana, Cuba  
Julio del 2004

## Resumen

En los últimos años del pasado siglo y en los primeros del nuevo siglo la industria del software ha experimentado un crecimiento en todo el mercado mundial. Nuestro país con vistas a ubicarse entre los principales países productores de software ha creado la Universidad de las Ciencias Informáticas. Esta Universidad esta concebida como centro para el desarrollo y exportación de software.

En la actualidad esta Universidad se encuentra en un proceso de construcción. Durante este proceso constructivo en la UCI se invierten diariamente innumerables recursos tanto en la construcción como en tecnologías y otras esferas, recursos que deben ser controlados todos los días. Con este fin surge el Sistema de Control de Inversiones. Este sistema se encargará del procesamiento de toda la información relacionada con las inversiones en la UCI. Esta información permitirá generar reportes que ayuden a la toma de decisiones de los directivos así como mostrar el estado actual del presupuesto.

En estos momentos el proceso de control de las Inversiones en la UCI se realiza de forma manual, diariamente se originan un número elevado de datos que se almacenan. Estos datos son almacenados en tablas de Microsoft Office Excel y guardados por meses sin tener ningún sistema que ayude a su procesamiento, control y persistencia. Debido al elevado contenido de esta información es necesario utilizar grandes recursos tanto humanos como materiales para mantener su control. Además de la imposibilidad que esto brinda a la hora de obtener cualquier reporte muy necesario para la toma de decisiones de los directivos.

El presente trabajo tiene como objetivo realizar el análisis y diseño del sistema propuesto para dar una solución factible al problema existente. Para el desarrollo de este documento se utilizó la metodología RUP quedando dividido en varios capítulos el análisis y desarrollo del sistema; estos capítulos abarcan la fundamentación teórica, el estudio preliminar, análisis y diseño.

## Índice

Introducción	6
Capítulo I: Fundamentación teórica	9
1.1 Estudio de sistemas existentes	9
1.2 Herramientas a utilizar para el desarrollo del sistema	11
1.2.1 Plataforma .Net	11
1.2.2 XML	12
1.2.3 Servicios Web	13
1.2.4 El protocolo	15
1.2.5 SQL Server 2000	15
1.3 Metodología Utilizada	17
1.4 Lenguaje de Modelación	18
Capítulo II: Estudio Preliminar	21
Introducción	21
2.1 Objeto de Estudio	21
2.1.1 Problema	21
2.1.2 Objeto de Automatización	22
2.1.3 Software propuesto	23
2.2 Modelo del Negocio	25
2.3 Especificación de los requisitos del software	32
2.3.1 Requisitos funcionales	34
2.3.2 Requisitos no funcionales	38
2.4 Casos de uso de alto nivel	41
2.4.1 Actores	41
2.4.2 Lista de casos de uso	42
2.4.3 Casos de uso por ciclos	45
2.5 Diagrama de casos de uso	46
Capítulo III: Diseño del Sistema	47
Introducción	47
3.1 Especificación de casos de uso	47
3.2 Modelo de clases de análisis del Sistema	47

3.3 Diagramas de secuencia del Sistema-----	47
3.4 Diagramas de interacción-----	48
3.5 Diagramas de Clases Web-----	48
3.6 Diagrama de base de datos-----	48
3.7 Forma general y principios de la protección y seguridad-----	49
3.8 Forma general del tratamiento de errores-----	49
Conclusiones-----	50
Referencias bibliográficas-----	52
Bibliografía-----	53
Glosario de términos-----	54
Anexos.-----	56
Anexo 1: Especificación de los Casos de Uso-----	56
Anexo 2: Modelo conceptual del Sistema-----	66
Anexo 3: Modelo de clases de análisis del Sistema-----	67
Anexo 4: Diagramas de secuencia del Sistema-----	68
Anexo 5: Diagramas de Interacción-----	71
Anexo 6: Diagramas de Clases Web-----	79
Anexo 7: Descripción de las clases-----	84
Anexo 8: Diagrama de base de datos relacional-----	98
Anexo 9: Descripción de las tablas de la base de datos-----	99

## Introducción

En la actualidad las tecnologías de la información y las comunicaciones son casi imprescindibles en las actividades de la vida cotidiana. En las sociedades modernas están cambiando las formas de aprender y transmitir los conocimientos. En nuestro país se llevan a cabo un grupo de proyectos con vistas a lograr la informatización de la sociedad cubana. Con este objetivo surge la Universidad de las Ciencias Informáticas (UCI) una futura ciudad digital que será la vanguardia en todo este proceso.

En estos momentos en la UCI se invierten cada día innumerables recursos en sus diferentes áreas. Estos recursos deben ser controlados diariamente por el personal económico de la UCI. En la actualidad, todo el proceso se realiza de forma manual. Cada vez que llega una factura se registran sus datos y se almacena. Esta factura pertenece a una etapa y está dada por un determinado concepto. En todo este proceso se trabaja en moneda nacional y en divisas lo que hace el trabajo más complejo. En la mayoría de las facturas se incluyen las dos monedas. Solo en contados casos se factura la moneda nacional separada.

Cuando se define una etapa se definen también los conceptos, es decir cada etapa tiene un presupuesto asignado por cada concepto y uno general para la etapa que nunca debe sobrepasar la cantidad asignada. Si esto ocurre se solicitan aumentos y aprobados convenientemente se adicionan al presupuesto. También puede darse el caso de que una deuda sobrepase el presupuesto.

El estado de las deudas se despachan semanalmente. Se asigna una cantidad de momento que lo ubican en la cuenta. En base a esta cantidad se preparan los pagos y se cierran las deudas cuando el banco entrega los cheques o transferencias.

Cuando se le efectúa el pago a un suministrador se pueden pagar una o varias facturas. También se puede pagar parcialmente una factura. Sucede mucho que se asignan financiamientos en números cerrados para los suministradores y al final para cerrar se efectúa el pago parcial de una factura. Los pagos se efectúan en orden de vejez de las deudas, es decir se priorizan las facturas más viejas para

mantener la salud de las cuentas por cobrar de las empresas aunque también puede suceder que se solicite cerrar una factura en cuestión.

Todos los cheques se giran en una sola moneda y no se parecen ni en la numeración. Los cheques son muy diferentes, unos en divisas (USD) y otros en moneda nacional (CUP). Para esto se toma la numeración de cada uno de ellos y a qué suministrador se le giró. En varias ocasiones se pregunta por un determinado cheque y se da la numeración de este para efectuar la búsqueda y comprobar si este cheque fue emitido por la UCI.

En general, no se cuenta con un sistema que automatice el control del presupuesto, lo cual implica elevar en un alto por ciento la eficiencia del proceso. Con este trabajo se persiguen como objetivos llegar hasta la etapa de diseño de un sistema capaz de controlar el presupuesto de la UCI, eliminando en lo posible el volumen de información en papel y enfocando el problema de forma general, de manera que la solución sea aplicable a otros Programas de la Revolución que se llevan a cabo en todo el país.

A partir de un análisis del objetivo general se derivan los siguientes objetivos específicos:

- Facilitar el procesamiento y control de toda la información.
- Facilitar la gestión de la información necesaria para la toma de decisiones de los directivos.
- Proporcionar el método de almacenar el historial de las facturas, pagos y cheques efectuados.
- Mantener una constante actualización de la información referente al presupuesto.
- Mantener la persistencia de toda la información almacenada.

Para ello se realizó un estudio del proceso en cuestión a través de entrevistas a las personas implicadas, definiendo los procesos a automatizar, profundizar en la metodología de construcción de software a emplear (RUP), realizar el análisis del sistema y llevar a cabo el diseño del primer ciclo de desarrollo del sistema.

El presente documento se estructura en varios capítulos, que incluyen todo lo relacionado con el trabajo investigativo realizado, así como el análisis y diseño de la herramienta que se propone.

En el *Capítulo I: Fundamentación teórica*, se describe la metodología empleada en el análisis y diseño del sistema, las herramientas de modelación, la tecnología empleada en desarrollo del mismo así como un análisis de los sistemas existentes en la actualidad que dan solución a problemas similares.

En el *Capítulo II: Estudio preliminar*, se describe el objeto de estudio, problemas existentes, se especifican los detalles de la construcción de la herramienta, propuesta del sistema, requerimientos funcionales y no funcionales y los casos de uso del sistema.

En el *Capítulo III: Análisis y diseño del sistema*, se representa el análisis y parte del diseño del sistema que se propone, mostrándose la expansión de los casos de uso del sistema, los diagramas de secuencia del sistema, diagramas de colaboración y de clases, así como la descripción detallada de las clases. También se detalla la base de datos relacional con el diseño de la base de datos informativa, se plantean los detalles relacionados con el propósito del sistema que se propone, se describe el tratamiento de errores, los principios en que se basa la ayuda general y el sistema de seguridad y protección.

## **Capítulo I. Fundamentación Teórica.**

### **Introducción.**

Durante el proceso constructivo de la UCI se invierten diariamente innumerables recursos tanto en la construcción como en tecnologías y otras esferas, recursos que deben ser controlados exhaustivamente todos los días para lo cual se desea desarrollar un software que sea capaz de controlar, almacenar y brindar toda la información del estado actual del presupuesto.

Por su gran importancia el software debe permitir primeramente acceso solo al personal autorizado a trabajar con esta información definiendo no solo usuarios con acceso al sistema; sino también debe definir grupos de usuarios con determinados privilegios para cada uno de ellos. El sistema debe almacenar información necesaria sobre los proveedores con los que efectúa transacciones, así como permitirle a los usuarios del sistema definir la etapa en la que se está trabajando y el presupuesto con que se cuenta en esta etapa. Además el sistema es el encargado de crear un historial con todas las facturas efectuadas así como los cheques pagados a los diferentes proveedores por concepto de estas facturas.

El sistema debe brindar una alta confiabilidad y seguridad de la información que se está procesando por lo que debe permitir el acceso a personas autorizadas, así como garantizar en todo momento salvadas directas de la base de datos y demás información en otros formatos de almacenamiento como CDROM. También debe permitir la exportación de reportes a otros formatos como Excel, Word y otros que ayuden al trabajo con la información. además en todo momento debe permitir chequear la información reportada y permitir imprimirla.

### **1.1 Estudio de sistemas existentes.**

Analizando las tendencias y tecnologías más modernas de los productos de software existentes en el mercado que dan solución a problemas similares encontramos varios sistemas de los que daremos algunas características.

1. **Visual Conta:** Este es un sistema que fue desarrollado para funcionar como sistema contable para la empresa individual, o como la herramienta del contador para mantener la contabilidad de todos sus clientes, con múltiples



períodos en línea, definiendo el plan de cuentas, ingresando los asientos, y emitiendo libros y balances con gran facilidad.

Visual CONTA le permite controlar el libro diario y los movimientos por cuenta, emitir los balances. Permite mantener de una manera muy eficiente el plan de cuentas utilizando para esto un control del tipo arbolar, similar al que presenta Windows para administrar archivos.

Permite mantener los asientos contables de una manera muy sencilla. Brinda la posibilidad de realizar asientos desbalanceados (que obviamente no incluye en el balance general).

- 2. Visual Cuentas:** Conserva el registro de las operaciones, emite resúmenes de saldos y estados de cuenta, mantiene los archivos de clientes y proveedores ordenados, con búsquedas por varias claves.

Visual CUENTAS permite:

- Mantener los archivos de clientes y proveedores, así como las respectivas cuentas corrientes, en forma totalmente visual.
- La emisión de una minuta diaria, en la cual se detallan todos los movimientos efectuados en las cuentas corrientes en la fecha indicada.
- Emitir informes de movimientos y saldos, permitiendo conocer en todo momento la situación pasada, presente y futura de las cuentas.
- Visualizar informes en pantalla antes de ser enviados a la impresora.

Estos Software presentan altos precios en el Mercado Internacional por lo que se hace muy difícil la adquisición de alguno de ellos para su utilización. Además no reúnen todas las condiciones requeridas por los usuarios en el trabajo que realizan. Por lo que se necesita un software que reúna un grupo de características más específicas a los usuarios y de más fácil adquisición.

Nuestro Software agrupará características similares a estos Software pero además deberá incluir:

- Reporte de Pagos que se utiliza como guía para generar el reporte de pagos de la Etapa en la que se está trabajando.

- Reportes ordenados por diferentes criterios que sirven para la toma de decisión de los directivos.
- Garantizar en todo momento una copia de seguridad de la información para recuperación del sistema en caso de fallas.
- Permitir exportar los informes a otros formatos para una mejor comprensión de la información y facilitar el trabajo.
- Permitir crear grupos de usuarios definiéndole a cada uno sus privilegios en el sistema.
- Permitir ingresar una Clave de Acceso al sistema para evitar ingresos de terceros sin autorización.

## **1.2 Herramientas a utilizar para el desarrollo del sistema.**

Para el desarrollo del sistema se realizó un estudio de las posibles herramientas a utilizar en su construcción. Teniéndose en cuenta la tendencia actual y las novedades de cada una de ellas.

Estas son:

- Plataforma .Net.
- XML como lenguaje de los Servicios Web.
- Servicios Web para el acceso a datos.
- SQL Server 2000 como gestor de bases de datos.

### **1.2.1 Plataforma .Net.**

La plataforma .NET permite la creación y consumo de aplicaciones, procesos y sitios WEB basados en XML que comparten y combinan información y funcionalidad entre cada uno de ellos por diseño, en cualquier plataforma o dispositivo inteligente, y provee soluciones para las necesidades específicas de organizaciones e individuos.

La plataforma .NET incluye una familia de productos, construidos sobre estándares de Internet, que provee funcionalidad para los aspectos de construcción de software (herramientas), administración (servidores) y uso (servicios base y clientes inteligentes) y experiencias de servicios Web XML. .NET se convertirá en parte de

las aplicaciones, herramientas y servidores que ya se usan hoy en día – así como en productos que extienden los servicios Web XML a capacidades para todas las necesidades de negocios.

Para integración, los “Servicios Web XML” sobre .NET hacen posible que diferentes piezas de software trabajen en conjunto para:

- Unir aplicaciones. Convertir aplicaciones independientes a constelaciones de aplicaciones para usar datos reales.
- Intercambiar datos. Los datos de clientes residen en aplicaciones aisladas, impidiendo ofrecer nuevos servicios que exploten esos datos.

### **1.2.2 XML**

XML está transformado completamente a la creación y el uso de software. El Web revolucionó la comunicación entre usuarios y aplicaciones. XML está revolucionando la comunicación entre aplicaciones o, de forma más general, la comunicación entre equipos, pues ofrece un formato de datos universal que permite adaptar o transformar fácilmente la información.

Ya existen muchas herramientas de programación preparadas para trabajar con formatos basados en XML. Es una tendencia y se está consolidando. Es ya un formato universal, que lo están adoptando multitud de programadores y que puede abarcar cualquier caso de intercambio de información.

XML sirve para que muchos programas interpreten bien cualquier tipo de dato. No solo eso. XML sirve para que algunos programas hablen entre ellos sin intervención humana. La Computación Distribuida, Interoperabilidad, Monitorización, son situaciones en las que resulta imprescindible este tipo de comunicación. XML es también la solución en estos casos. Como veremos, los Servicios Web son un caso particular de “Computación Distribuida” y XML es su lenguaje de base.

Resumiendo los puntos más significativos del XML como lenguaje de desarrollo de aplicaciones:

- XML será probablemente la principal tecnología para ofrecer funcionalidad avanzada a las aplicaciones de Internet, Extranets e Intranets. En este

sentido permite ofrecer información actualizada y segura, con independencia del medio de conexión, ordenadores de sobremesa, PDA, teléfono WAP, etc.

- Se está convirtiendo en el formato estándar de intercambio de datos, permitiendo la integración de datos entre diversas aplicaciones y bases de datos.
- Reemplazará a HTML en aplicaciones Web donde se requieran grados elevados de reutilización, intercambio de datos, automatización e interacción con otras aplicaciones externas e internas, así como en la relación con otras personas y entidades.

Los puntos anteriores son debidos a las características y ventajas que tiene sobre lenguajes alternativos, entre las que se destacan:

- XML muestra el significado y las relaciones de la información contenida (en documentos y bases de datos), lo que permite gestionar y manejar datos, tanto estructurados como no estructurados.
- Es un formato para intercambiar y compartir información en bases de datos, documentos y aplicaciones.
- Es extensible y por lo tanto adaptable tanto a necesidades futuras, en términos generales como a necesidades específicas de una empresa o de una relación comercial en concreto.
- Utilización independiente del mecanismo de acceso a datos, transacciones e interacciones.
- Las aplicaciones confeccionadas con XML son aplicaciones Web nativas, fáciles de gestionar, mantener y actualizar. Sus componentes permiten un elevado grado de reutilización, por lo que *son más baratas* de desarrollar que las convencionales.

### **1.2.3 Servicios Web.**

Los servicios Web XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientes de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque

los servicios Web XML son independientes entre sí, pueden vincularse y formar grupos de colaboración para realizar tareas determinadas.

¿Para qué sirve un Servicio Web? La respuesta puede ser otra pregunta: ¿Para qué sirve en programación una rutina? Todos sabemos que una rutina es como una caja negra, que encierra cierto proceso o algoritmo, y que cumple una función clara. Muchas rutinas y un guión central componen un programa en lo que se llama "programación estructurada". Un Servicio Web viene a ser una rutina en Internet.

¿Por qué se llama "Servicio Web" y no "Rutina en Internet"? Los protocolos que soportan los Servicio Web se comunican normalmente por el puerto 80 y se basan en HTTP, métodos GET y PUT. Esto hace que podamos acceder a ellos al igual que lo hacemos en una página Web. La diferencia entre una página Web y un Servicio Web, es que la página la visita cualquier individuo interesado, mientras que el servicio sólo lo visitan programas que lo requieren.

De modo, que el conjunto de Servicio Web en Internet es una World Wide Web paralela, de carácter no humano, sino cibernético. Veamos que los ordenadores ya hablan solos a través de Internet.

Los Servicios Web se actualizan de forma transparente para el programador y para el encargado de mantenimiento de la aplicación. Además, mediante un Servicio Web puedes implementar a tu programa funciones imposibles de contemplar bajo el uso de rutinas de librerías, como por ejemplo, incorporar un buscador de páginas Web. Por otro lado, la carga de CPU que supone la ejecución de una rutina, desaparece al usar Servicios Web. La carga se reparte por Internet, sobre el servidor del Servicio Web. Esto es un comienzo de "Computación Distribuida".

Los Servicios Web no pretenden eliminar del mapa a las librerías o módulos de programación, que de toda la vida hemos usado. No pueden hacerlo, ya que no son una versión mejorada de éstas, sino una herramienta con distintas aplicaciones en determinados casos.

Los Servicios Web permiten a los usuarios usar aplicaciones que comparten datos con otros programas modulares. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar, como XML. El objetivo final es la creación de un directorio online de

Servicios Web, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad. La integración de aplicaciones hará posible obtener información demandada en tiempo real, acelerando el proceso de toma de decisiones.

#### **1.2.4 El protocolo.**

Hay un convenio generalizado que nos da a entender que los Servicios Web se invocan en Internet por medio de protocolos estándares basados en XML. Hoy día hay dos grandes tendencias: XML-RPC y SOAP, aunque hay que reconocer que la mayoría de los Servicios Web públicos e interesantes, están definidos bajo estándares SOAP. Un lenguaje de programación que trabaje con SOAP, como por ejemplo .NET de Microsoft.

SOAP(Protocolo de acceso a objetos simples) es un lenguaje de mensajería basado en XML, estandarizado por consorcio W3C, que especifica todas las reglas necesarias para ubicar los Servicio Web XML, integrados en aplicaciones y establecer la comunicación entre ellos. El SOAP es un protocolo ligero para el intercambio de la información en un ambiente descentralizado, distribuido. Consiste en tres porciones: un sobre que define un marco para describir cuál, está en un mensaje y cómo procesarlo, un sistema de las reglas de codificación para expresar casos de uso-definidos, y una convención para representar llamadas y respuestas alejadas del procedimiento. El SOAP está creado con idea de dar un soporte completo y minucioso de todo tipo de Servicio Web.

#### **1.2.5 SQL Server 2000.**

Una solución integral de base de datos y análisis, SQL Server 2000 ofrece el rendimiento, escalabilidad y confiabilidad que requieren el exigente Web y los entornos empresariales de línea-de-negocios. El nuevo soporte de XML y HTTP simplifica el acceso a datos y el intercambio, mientras que las poderosas capacidades de análisis mejoran el valor de los datos. Las características de disponibilidad mejoradas maximizan el tiempo de actividad, las funciones de la administración avanzadas automatizan las tareas rutinarias y las herramientas mejoradas de programación y los servicios aceleran el desarrollo.

SQL Server 2000 es el eje principal de la administración y análisis de datos de la siguiente generación de productos y servicios de Microsoft .NET.

### **Consulta, análisis y manipulación de datos en el Web.**

- Simplifica la integración de sistemas del back-end y la transferencia de datos por medio de firewalls que usan la completa funcionalidad de SQL Server 2000 XML y el soporte de otros estándares de Internet, como XPath, XSL y XSLT. Los desarrolladores de Web pueden acceder a los datos usando XML sin la programación de base de datos relacional, mientras que los administradores de bases de datos pueden manipular los datos fácilmente en formato XML usando Transact-SQL (T-SQL) y los procedimientos almacenados.
- Conecta las bases de datos de SQL Server 2000 y cubos de OLAP usando el Web flexiblemente sin la programación adicional. Usa la segura conectividad de bases de datos en HTTP para asegurar que se puedan consultar los datos a través de un URL y las interfaces de usuario intuitivas.
- Habilita para el Web e integra los sistemas de negocios existentes a través del penetrante soporte de XML.
- Aprovecha por completo sus recursos de hardware ejecutando aplicaciones aisladas múltiples en una sola computadora que use el soporte de instancias múltiples de SQL Server 2000.

### **Construir aplicaciones rápidamente y asegurar que se mantengan funcionando.**

- Entrega rápidamente aplicaciones de bases de datos robustas y escalables que usan las herramientas mejoradas de desarrollo de SQL Server 2000. El Analizador de Consultas de SQL ahora incluye un depurador de procedimientos almacenados. Los Data Transformation Services (DTS) extienden la capacidad para mover y transformar los datos desde cualquier fuente.

- Mejora la productividad con las mejoras de T-SQL. Las nuevas funciones definidas por el usuario permiten la reutilización de código y simplifican el desarrollo. Mantiene datos consistentes con la integridad referencial para controlar la propagación en operaciones de eliminación y actualización.

### **1.3 Metodología Utilizada.**

Durante el proceso de desarrollo de software empleamos la metodología RUP (Rational Unified Process, Proceso Unificado de Rational). RUP hace énfasis en la adopción de las mejores prácticas del desarrollo de software, como una manera de reducir los riesgos *inherentes en el desarrollo de una nueva aplicación de software*, de esta manera logramos resultados más predecibles unificando nuestro equipo con procesos comunes que mejoran la comunicación y crean un entendimiento de todas las tareas y responsabilidades.

RUP:

- Iterativo e Incremental.
- Dirigido por los Casos de Uso.
- Centrado en la Arquitectura.

Iterativo e Incremental:

Pequeños proyectos que incorporan incrementalmente nueva funcionalidad y cuyo desarrollo es una iteración.

- Obtiene un Sistema Robusto
- Reduce el Riesgo de tener un mal producto
- Reduce el Riesgo de no obtener el producto en el tiempo previsto
- Permite atacar problemas con requisitos incompletos.

Dirigido por los Casos de Uso:

- Servicios que un actor requiere del sistema y le proporcionan un resultado.
- Proporcionan los Requisitos Funcionales del Sistema.
- Describen toda la funcionalidad del Sistema.
- Cambios en Requisitos de un Caso de Uso fácil detectar las clases y componentes que afectan.



Centrado en la Arquitectura:

- Casos de Uso describen la *Funcionalidad* del Sistema
- Arquitectura define la *Forma* del Sistema
- Se describe mediante *Vistas* que incorporan el 5-10% de los casos de uso más relevantes.

#### **1.4 Lenguaje de Modelación.**

Para modelar el análisis y el diseño del software se utiliza el lenguaje UML que es el más utilizado mundialmente.

¿Por qué UML?

UML (Unified Modeling Language, Lenguaje Unificado de Modelación)

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos del sistema de un Software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

Además tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.

- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, interactivo e incremental.

**Rational Rose.** Es la herramienta Case desarrollada por los creadores de UML que cubren todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Nos permite una trazabilidad real entre modelo (análisis y diseño) y el código ejecutable.

**Rational Rose** domina el mercado de herramientas para el análisis, modelamiento, diseño y construcción orientada a objetos, tiene todas las características que los desarrolladores, analistas, y arquitectos exigen – soporte UML incomparable, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada.

La corporación Rational Rose ofrece el Proceso Unificado de Racional (RUP), que unifica las mejores prácticas de muchas disciplinas en un consistente y completo proceso del ciclo de vida, que permite al equipo de desarrollo disminuir los tiempos de liberación, además de hacer más predecible el software que ellos producen. Este proceso está basado en el Lenguaje Unificado de Modelación (UML – estándar de la industria) y únicamente integrado a herramientas líderes en el desarrollo de software de Rational, el Proceso Unificado de Racional apoya el equipo completo de desarrollo de software con guías detalladas e información crítica aplicable a la mayoría de las aplicaciones de la industria.

**Rose** es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo

utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

## **Capítulo II. Estudio Preliminar.**

### **Introducción**

En este capítulo se describen el objeto de estudio, los problemas existentes, el mecanismo que se pretende desarrollar, se hace una propuesta del sistema, se definen los requerimientos funcionales y no funcionales del mismo y los casos de uso del sistema.

### **2.1 Objeto de Estudio.**

De un análisis relativamente sencillo se puede definir en las universidades varias esferas de trabajo, una de estas esferas es la económica. El sistema a desarrollar se centra dentro del proceso económico de la UCI y en particular el presupuesto, por tanto resultó sumamente indispensable estudiar y analizar como se desarrolla el trabajo con el presupuesto y las inversiones.

El sistema que se propone no tiene relación con ningún otro sistema; además en la actualidad no existe ninguno que realice esta tarea en la UCI. Esta aplicación permitirá mantener un exhaustivo control sobre el estado actual del presupuesto y las inversiones.

Toda esta información manejada es de gran confidencialidad, se debe garantizar su seguridad y protegerla del mal uso, pérdida y alteración de la información, y velar por el cumplimiento de las normas de privacidad.

#### **2.1.1 Problema.**

En estos momentos el proceso de control de las Inversiones en la UCI se realiza de forma manual, diariamente se originan un número elevado de datos que se almacenan en papel provocando un manejo poco eficiente de tan alto volumen de información. Estos datos son almacenados en tablas de Microsoft Office Excel y guardados por meses sin tener ningún sistema que ayude a su procesamiento, control y persistencia; esto hace necesario que todo el volumen de información a papel tenga que almacenarse y conservarse durante tiempo indefinido para evitar pérdidas irreparables. Debido al elevado contenido de esta información es necesario utilizar grandes recursos tanto humanos como materiales para mantener

su control. Las búsquedas son cada vez más difíciles de realizar y más frecuentes en el trabajo por el alto volumen de información que se cuenta. Además de la imposibilidad que esto brinda a la hora de obtener cualquier reporte muy necesario para la toma de decisiones de los directivos. Muchas veces se le solicita verificar si un cheque fue emitido por la UCI para lo cual se requiere hacer una búsqueda que puede tardar desde cinco minutos hasta más de una hora.

### **2.1.2 Objeto de Automatización.**

En el proceso de control de Inversiones se desea automatizar todo lo referente al control y persistencia de la información originada en la UCI.

Primeramente se definirán los ministerios con los que se tendrá relación y las empresas que pertenecen a cada ministerio. Para cada empresa se tiene un contacto que es la persona encargada de controlar las acciones.

También se definen las etapas de trabajo, cada etapa tiene fecha de comienzo y fecha de terminación de la misma. Para esta etapa es necesario definir los conceptos de presupuesto que interactuarán.

Cada que vez que se emite una factura se registran sus datos y se almacena. Esta factura pertenece a una etapa y esta dada por un determinado concepto; también pertenece a una empresa en particular. En todo este proceso se trabaja en moneda nacional y en divisas lo que hace el trabajo más complejo. En la mayoría de las facturas se incluyen las dos monedas. Solo en contados casos se factura la moneda nacional por separado.

Las deudas se despachan semanalmente su estado. Se le asigna una cantidad determinada y en base a esta cantidad se preparan los pagos y se cierran las deudas cuando el banco entrega los cheques o transferencias. Las deudas son las facturas originadas por los diferentes conceptos de presupuesto que no se ha pagado.

Cuando se le efectúa el pago a un suministrador (empresa) se pueden pagar una o varias facturas o parcialmente una factura. Los pagos se efectúan en orden de vejez de las deudas, es decir, se priorizan las facturas más viejas para mantener la salud de las cuentas por cobrar de las empresas aunque; también puede cerrarse una

determinada factura en cuestión. Todos los cheques se giran en una sola moneda y son muy diferentes, unos en divisas (USD) y otros en moneda nacional (CUP).

Las principales entradas de este sistema son:

- Crear una nueva etapa.
- Adicionar conceptos de presupuesto para cada etapa.
- Registrar los ministerios.
- Registrar las empresas con las que se trabaja.
- Registrar las facturas.
- Registrar los contactos de las empresas.
- Registrar los usuarios del sistema.

### **2.1.3 Software propuesto.**

Con el objetivo de mejorar la calidad del trabajo y darle solución a los problemas mencionados se ha decidido desarrollar un sistema que se centra dentro de este proceso tan importante para la universidad. Se realizarán algunos cambios dentro de este proceso dando como resultado una propuesta del sistema que brindará amplias facilidades para almacenar, procesar, manipular y proteger la información. A continuación se mencionan algunos de los servicios que el software brindará:

**Gestionar conceptos de Presupuesto:** Permitirá crear nuevas etapas así como almacenar los datos de la etapa con la que se trabaja.

Permitirá adicionar para cada etapa los principales conceptos del presupuesto con los que se trabaja así como permitirle al usuario dividir este presupuesto padre en presupuestos más específicos.

**Gestionar Empresas:** Permitirá adicionar los datos de las empresas con las que se trabaja así como modificar sus datos en caso necesario.

Permitirá adicionarle al sistema los contactos de cada una de las empresas así como modificar sus datos personales en caso necesario.

**Gestionar Ministerios:** Permitirá adicionar los ministerios con los que se trabaja.

**Gestionar Facturas:** Permitirá insertar las facturas que se originan diariamente en las diferentes áreas por los diferentes conceptos.

**Gestionar autenticación de los usuarios:** Permitirá solo el acceso al sistema del personal autorizado y los usuarios deberán autenticarse antes de acceder a cualquier opción del sistema.

Permitirá cambiar la contraseña de inicio de sesión al sistema.

**Gestionar usuarios del sistema:** Permitirá crear nuevas cuentas de usuarios de acceso al sistema así como modificar sus datos personales en caso necesario.

Permitirá eliminar las cuentas de los usuarios que no deben tener acceso al sistema.

**Gestionar pagos:** Permitirá generar el reporte de pagos necesario para la toma de decisiones así como almacenar los pagos efectuados.

**Gestionar cheques:** Permitirá almacenar los cheques emitidos por la UCI así como realizar cualquier búsqueda referente a ellos.

**Gestionar la información necesaria para la toma de decisiones:** Permitirá reportar por fechas mostrando toda la información existente entre estas fechas.

Permitirá reportar por empresas mostrando toda la información relacionada con esta empresa.

Permitirá reportar por ministerios mostrando toda la información relacionada con este ministerio.

Permitirá reportar por etapa mostrando toda la información que se cuenta de esta.

**Administración:** Para el desarrollo de la aplicación es necesario tener almacenados los usuarios que podrán utilizar el sistema debido a que hay que garantizar un alto grado de seguridad, donde se garantizará que la información sea vista únicamente por quien tiene derecho a verla, por lo que se permitirá realizar actualizaciones de los usuarios definiéndose para cada uno el rol que juegan dentro del sistema.

**Autenticación:** El sistema permitirá la autenticación de los usuarios validando sus datos y según el rol o tipo de usuario se le permitirá realizar determinadas acciones. Por lo que el acceso al sistema será restringido, identificando con certeza a los diversos usuarios que pueden interactuar con él, permitiendo contemplar diversos mecanismos, que dependiendo del nivel operativo y responsabilidades de los usuarios le permitan garantizar la identidad de los mismos.

## 2.2 Modelo del Negocio.

### Reglas del Negocio:

- Solo tienen acceso a la información el personal autorizado.
- La información debe ser precisa.
- La información debe ser persistente.

### Descripción del negocio actual:

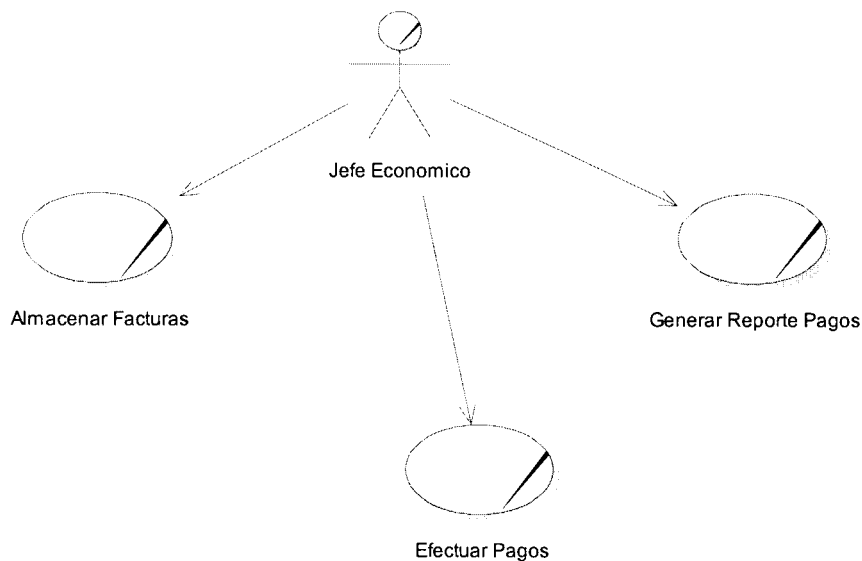
En la actualidad la UCI diariamente genera gastos por diferentes conceptos. Estos gastos son controlados por medio de facturas que responden a un concepto y a un determinado suministrador. Estas facturas se almacenan diariamente por el jefe de Economía y su asistente para mantener un control. Estas facturas pertenecen a una etapa en específico por tanto se define una etapa y un presupuesto para esta etapa. Cuando se le informa al jefe económico que tiene asignado cierta cantidad de dinero en el banco este le pide al asistente un reporte ordenado por suministradores de lo que le pagara hasta el momento. Este reporte se lleva al banco y este autoriza los cheques a pagar. Después de pagados se toma el número de cada cheque y se almacena para mantener su control; a quien fue pagado, la cantidad y la fecha. Los cheques se pagan en ambas monedas porque las facturas pueden ser en divisas (USD) o moneda nacional (CUP).

<b>Actores del negocio</b>	<b>Justificación</b>
Jefe Económico.	El Jefe Económico, es el encargado de controlar toda la información generada diariamente en la UCI por los diferentes conceptos tratados (Tecnología, Construcción, equipos y otros). También es el autorizado a tramitar los cheques para los diferentes suministradores a los que se les debe dinero por estos conceptos (uno). Es el encargado de decidir cuantos se le pagara a cada a suministrador hasta el momento.



Trabajadores del negocio	Justificación
Jefe de Área	Es el encargado de enviar las facturas generadas en el día al jefe económico.
Asistente	El Asistente, es el encargado de manipular toda la información generada diariamente por los diferentes conceptos. También es el encargado de hacer el reporte sugiriendo lo que se le debe pagar a cada suministrador hasta el momento. Además tiene la responsabilidad de efectuar los pagos y guardar la información de estos.

**Diagrama de casos de uso del negocio.**



**Caso de uso del negocio: Almacenar Facturas.**

**Actores del negocio:** Jefe Económico.

**Propósito:** Almacenar todas las facturas generadas por los diferentes conceptos en el día.

**Resumen:** El caso de uso se inicia cuando el jefe económico solicita las facturas del día generadas por los diferentes conceptos. Este recibe todas las facturas, o las facturas que se van generando en el transcurso del mismo. Las analiza y se las envía a su asistente que es el encargado de almacenarlas. El caso de uso finaliza cuando estas facturas han sido almacenadas.

**Referencias:**

**Curso normal de los eventos:**

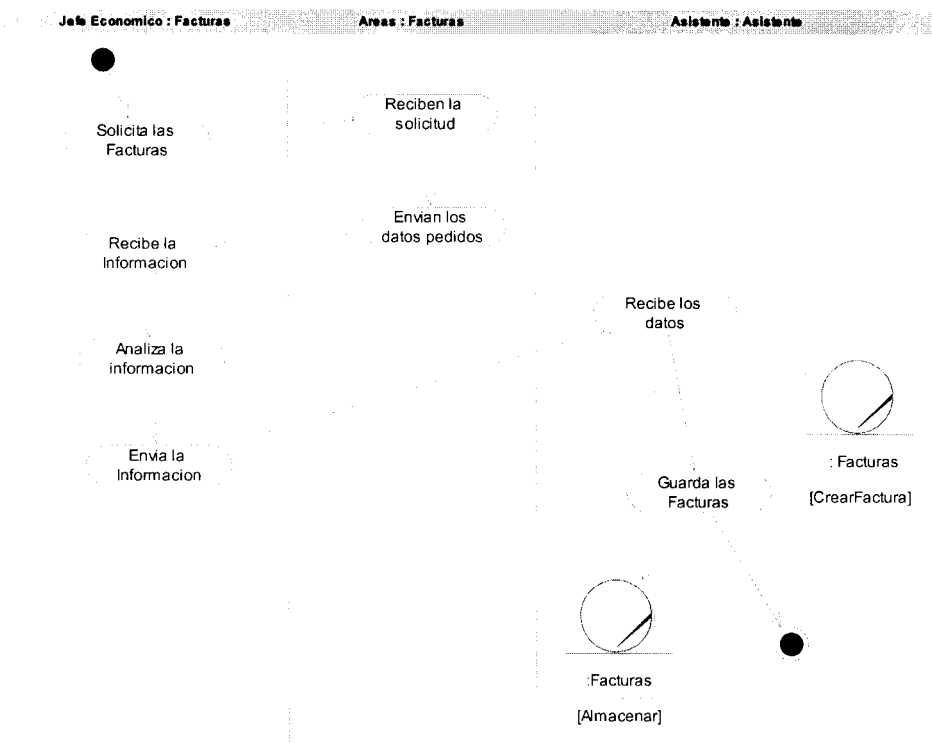
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El jefe económico solicita las facturas generadas ese día por los diferentes conceptos.	2. Las facturas se generan en las áreas por los diferentes conceptos.  3. Las facturas son enviadas al jefe económico.  4. El jefe económico analiza las facturas y las envía al asistente.  5. El asistente recibe la información enviada.  6. El asistente almacena todos los datos de las facturas recibidas ese día.

**Prioridad:** CUN Crítico

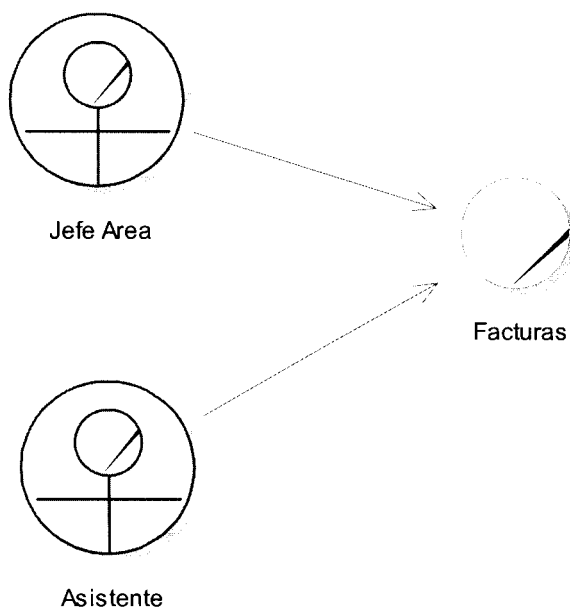
**Mejoras:** La automatización de este proceso permitirá agilizar el proceso de almacenamiento de la información así como una mayor seguridad y persistencia de esta información.

**Otras secciones:**

## Diagrama de actividad: Almacenar Facturas.



## DIAGRAMA DEL MODELO DE OBJETOS PARA EL CASO DE USO: Almacenar Facturas.

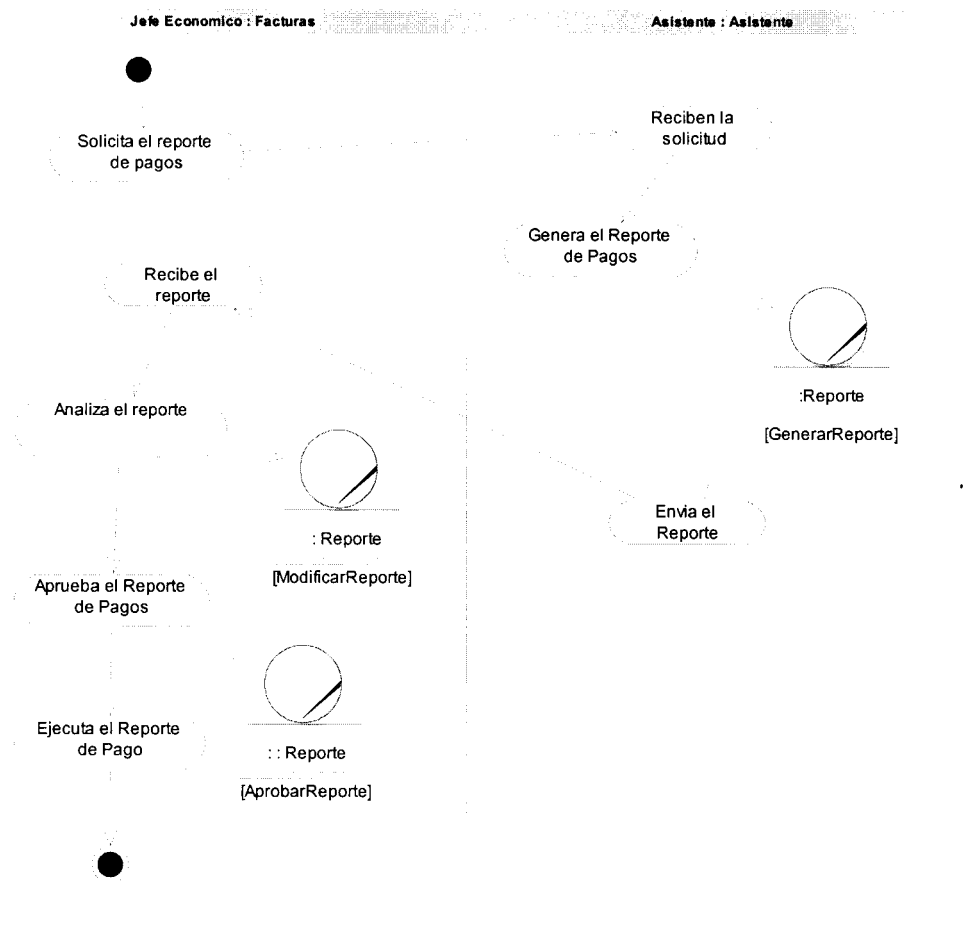


<b>Caso de uso del negocio: Generar Reporte Pagos.</b>	
<b>Actores del negocio:</b> Jefe Económico.	
<b>Propósito:</b> Aquí se genera la información necesaria para los cheques que se les pagaran a los diferentes proveedores.	
<b>Resumen:</b> El caso de uso se inicia cuando el jefe económico le solicita al asistente el reporte de pagos para generar los pagos que se efectuarán. El asiste genera un reporte con los posibles pagos a efectuar y lo envía al jefe económico. Este cuando recibe el reporte lo analiza y realiza los cambios pertinentes que piensa son necesarios y aprueba el reporte de pagos.	
<b>Referencias:</b>	
<b>Curso normal de los eventos:</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El jefe económico solicita el reporte de pagos al asistente.	2. El asistente analiza toda la información almacenada hasta el momento y genera un reporte de pagos.
	3. El asistente envía el reporte de pagos al jefe económico.
	4. El jefe económico recibe la información enviada.
5. El jefe económico analiza esta información y realiza todos los cambios pertinentes.	
6. El jefe económico aprueba el reporte de pagos.	
<b>Prioridad:</b> CUN Crítico	
<b>Mejoras:</b> La automatización de este proceso permitirá agilizar el proceso y le	

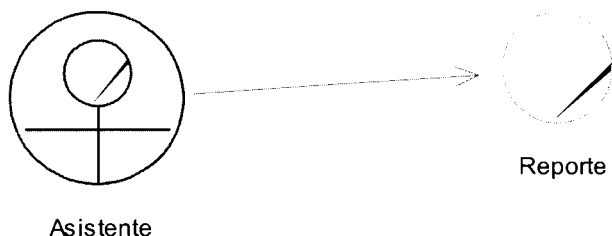
servirá de gran ayuda porque la información generada por el sistema será muy valiosa para la toma de decisiones. Además será una de las características más importantes que presentara el sistema.

**Otras secciones:**

### Diagrama de actividad: Generar Reporte de Pagos.



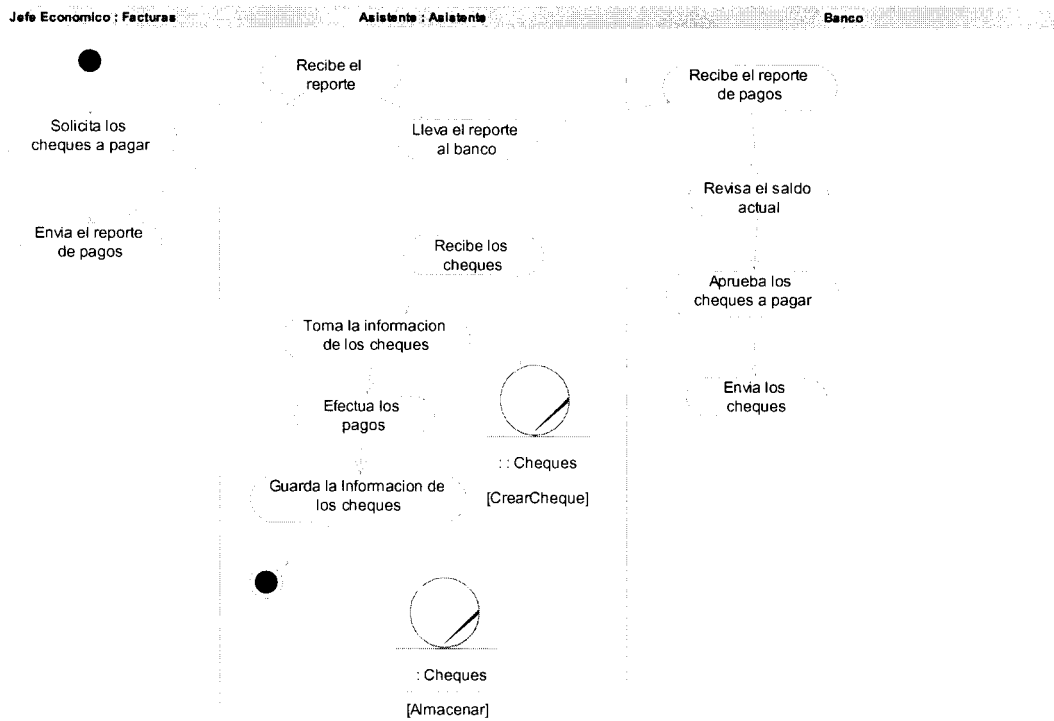
**DIAGRAMA DEL MODELO DE OBJETOS PARA EL CASO DE USO: Generar Reporte de Pagos.**



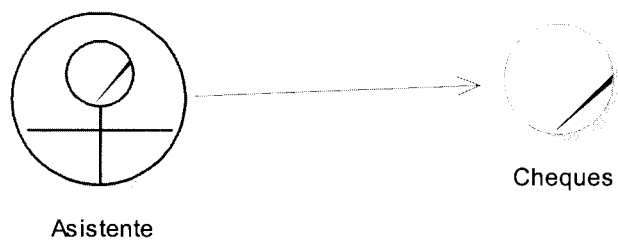
<b>Caso de uso del negocio: Efectuar Pagos.</b>	
<b>Actores del negocio:</b> Jefe Económico.	
<b>Propósito:</b> Se efectúa el pago de los cheques a los diferentes proveedores y se almacena la información de los mismos.	
<b>Resumen:</b> El caso de uso se inicia cuando el jefe económico solicita generar los cheques. El jefe económico lleva el reporte de pagos generado con la información de cada cheque a pagar. El banco revisa el saldo de la cuenta y autoriza los cheques. El asistente toma la información de los mismos y la almacena.	
<b>Referencias:</b>	
<b>Curso normal de los eventos:</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El jefe solicita autorizar los cheques a pagar.	2. El jefe envía el reporte de pagos al asistente que es el encargado de efectuar los pagos.
	3. El asistente lleva el reporte al banco para hacer efectivos los cheques.
	4. El banco verifica el saldo de la cuenta.

	<p>5. El banco autoriza los cheques.</p> <p>6. El asistente toma la información necesaria de cada uno de los cheques.</p> <p>7. El asistente efectúa el pago a los suministradores.</p> <p>8. El asistente almacena la información de los pagos efectuados.</p>
<p><b>Prioridad:</b> CUN Crítico</p>	
<p><b>Mejoras:</b> La automatización de este proceso permitirá agilizar el proceso de almacenamiento de la información así como una mayor seguridad y persistencia de esta información.</p>	
<p><b>Otras secciones:</b></p>	

## Diagrama de actividad: Efectuar Pagos.



## DIAGRAMA DEL MODELO DE OBJETOS PARA EL CASO DE USO: Efectuar Pagos.





## **2.3 Especificación de lo requisitos del software.**

### **2.3.1 Requisitos funcionales.**

1. Validar entrada de usuarios al sistema.
2. Crear un grupo de usuarios.
3. Administrar cuentas de usuario.
  1. Adicionar un usuario.
  2. Modificar un usuario.
  3. Eliminar un usuario.
4. Adicionar una empresa.
5. Adicionar un ministerio.
6. Adicionar un concepto (materiales de la construcción, tecnología, etc.).
7. Adicionar un presupuesto.
8. Adicionar una factura.
9. Generar solicitud de Pago.
10. Guardar detalles del Pago.
11. Guardar cheques.
12. Reportar Facturas.
13. Reportar Ministerios.
14. Reportar Empresas.
15. Reportar por fechas (Está relacionado con los anteriores).
16. Controles de etapas.
17. Reportar pendientes por ministerio.
18. Reportar pendientes por suministrador.
19. Realizar búsqueda por proveedor.
20. Realizar búsqueda por deuda.
21. Realizar búsqueda por pagos efectuados.

22. Realizar búsqueda por factura.
23. Realizar búsqueda por cheque.
24. Realizar búsqueda por importe.
25. Realizar búsqueda por fechas.
26. Exportar reportes.

### **Descripción de lo requerimientos.**

1. *Validar entrada de usuarios al sistema.*

El sistema debe ser capaz de permitirle al usuario abrir una sesión de trabajo y chequear si el usuario está autorizado para iniciar esta sesión.

2. *Crear un grupo de usuarios.*

El sistema debe permitirle al administrador crear un grupo de usuarios con las especificaciones de trabajo en el sistema.

3. *Administrar cuentas de usuario.*

El sistema debe permitirle al administrador administrar todas las cuentas de usuarios autorizados a trabajar en el sistema; crear un nuevo usuario definiendo el grupo al que pertenece, modificar un usuario o eliminar un usuario existente.

4. *Adicionar una empresa.*

El sistema debe permitirle al usuario agregar una empresa con sus datos correspondientes.

5. *Adicionar un Ministerio.*

El sistema debe permitirle al usuario agregar un ministerio con sus datos correspondientes.

6. *Adicionar un concepto.*

El sistema debe permitirle al usuario adicionar un ítem, materiales de la construcción, tecnología, etc.

7. *Adicionar un presupuesto.*

El sistema debe permitirle al usuario adicionar un presupuesto con sus datos correspondientes.

8. *Adicionar una factura.*

El sistema debe permitirle al usuario adicionar una factura con sus datos correspondientes.

9. *Generar solicitud de Pago.*

El sistema debe permitirles a los usuarios autorizados generar el informe con las solicitudes de pagos.

10. *Guardar detalles del Pago.*

El sistema debe permitirle al usuario guardar los detalles del pago con sus datos correspondientes.

11. *Guardar cheques.*

El sistema debe permitirle al usuario almacenar los cheques con sus datos correspondientes.

12. *Reportar Facturas.*

El sistema debe permitirle al usuario autorizado generar cualquier reporte relacionado con el estado existente de las facturas.

13. *Reportar Ministerios.*

El sistema debe permitirle al usuario autorizado generar cualquier reporte relacionado con los ministerios.

14. *Reportar Empresas.*

El sistema debe permitirle al usuario autorizado generar cualquier reporte relacionado con las empresas.

15. *Reportar por fechas.*

El sistema debe permitirle al usuario autorizado generar reportes por fechas, con la información necesaria del estado de todas las cuentas.

*16. Controles de etapas.*

El sistema debe permitir al usuario definir la etapa en la cual se esta trabajando.

*17. Reportar pendientes por ministerio.*

El sistema deberá permitir al usuario elaborar un reporte con los datos pendientes por un ministerio.

*18. Reportar pendientes por suministrador.*

El sistema deberá permitir al usuario elaborar un reporte con los datos pendientes por un suministrador.

*19. Realizar búsqueda por proveedor.*

El sistema deberá permitir al usuario autorizado realizar una búsqueda por nombre del proveedor o por organismo y listar los datos correspondientes a la misma.

*20. Realizar búsqueda por deuda.*

El sistema deberá permitir al usuario autorizado realizar una búsqueda por deudas de ambas monedas de forma general o especificando el organismo, el proveedor o el acápite de presupuesto y listar los datos correspondientes a la misma.

*21. Realizar búsqueda por pagos efectuados.*

El sistema deberá permitir al usuario autorizado realizar una búsqueda por pagos efectuados en ambas monedas de forma general o especificando el organismo, el proveedor, el acápite de presupuesto o el tipo de moneda y listar los datos correspondientes a la misma.

*22. Realizar búsqueda por factura.*

El sistema deberá permitir al usuario autorizado realizar una búsqueda por facturas especificando el número de la factura y el nombre del proveedor. El sistema deberá reportar si esta factura esta pagada o pendiente.

*23. Realizar búsqueda por cheque.*

El sistema deberá permitir al usuario autorizado realizar una búsqueda por cheque especificando el número de cheque y reportar si el cheque fue omitido y a nombre de quien (nombre del proveedor).

*24. Realizar búsqueda por importe.*

El sistema deberá permitir al usuario autorizado realizar una búsqueda de facturas por importes en ambas monedas y listar los datos correspondientes a la misma.

*25. Realizar búsqueda por fechas.*

El sistema deberá permitir al usuario autorizado realizar una búsqueda por fechas donde el sistema listara todas las facturas y cheques emitidos en un intervalo de tiempo determinado entrado por dicho usuario.

*26. Exportar reportes.*

El sistema deberá permitirle al usuario mostrar las salidas de los reportes en otros formatos como Excel, Word y en especial ficheros pdf.

### **2.3.2 Requisitos no funcionales.**

#### **Requisitos de funcionalidad**

- Tiempo de entrenamiento.

El sistema debe ser sometido a una fase de pruebas en las cuales los usuarios se familiaricen con este y a la vez se puedan detectar posibles errores del mismo.

## **Requisitos de confiabilidad**

1. Disponible todo el tiempo.
2. Tiempo de respuesta mínimo.
3. Preciso en la información.
4. Extensibilidad.
5. Mantenimiento.
6. Salvas de la información.

## **Descripción.**

1. Disponible todo el tiempo.

El sistema debe estar disponible todo el tiempo para trabajar de forma tal que usuario pueda entrar al sistema desde su cuarto y a cualquier hora del día (noche).

2. Tiempo de respuesta mínimo.

El sistema debe ser capaz de tener un tiempo mínimo de recuperación ante cualquier fallo en una de las operaciones.

3. Preciso en la información.

El sistema debe ser preciso en la información que le suministra al usuario para evitar cualquier tipo de error. Además la información manejada por el sistema será protegida contra corrupción y estados inconsistentes.

4. Extensibilidad.

Se debe garantizar la inserción de nuevos módulos sin negar lo realizado o afectar el buen funcionamiento.

5. Mantenimiento.

El sistema debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo.

6. Salvas de la información.

El sistema debe permitir hacer copias persistentes en otros dispositivos de toda la información.

### **Requisitos de performance**

1. Tiempo de respuesta rápido.
2. Alto rendimiento.
3. Varios usuarios conectados.

### **Descripción.**

1. Tiempo de respuesta rápido.

El sistema debe tener un tiempo de respuesta rápido ante cualquier solicitud del usuario.

2. Alto rendimiento.

EL sistema debe implementar varias transacciones por segundo para dar una respuesta rápida al usuario y evitar demoras innecesarias.

3. Varios usuarios conectados.

El sistema debe permitir que varios usuarios se conecten al mismo tiempo y dar respuestas consistentes a cada una de las solicitudes de estos.

### **Requisito de portabilidad**

El sistema deberá funcionar sobre plataforma Windows, que es la que más usada hasta el momento en la universidad y para la que se prevé implantar el software.

### **Requisitos de documentación online de usuarios y ayudas del sistema.**

1. Ayuda online.
2. Documentación del software.

## Descripción.

### 1. Ayuda online.

El sistema debe implementar una ayuda online de forma tal que el usuario que este en el sistema pueda visitarla en caso de cualquier duda, esta debe brindarle información de todas las actividades a realizar por el.

### 2. Documentación del software.

El software debe tener la documentación completa de todas las tareas y operaciones que realiza así como todo sobre su implementación.

## Interfaces

### 1. Interfaz de usuario.

El sistema debe tener una interfaz sencilla, intuitiva, amigable y mantener el formato en páginas similares. En general, fácil de usar.

### 2. Interfaz de comunicación.

El sistema debe permitirle al usuario conectarse desde otra maquina que no se encuentre en el local de trabajo.

## 2.4 Casos de uso de alto nivel.

En este epígrafe se abordarán los casos de uso de alto nivel, definiendo los actores que intervienen en los mismos así como los ciclos de ejecución de los mismos.

### 2.4.1 Actores.

Nombre del actor	Descripción
Supervisor	Es la persona encargada de visualizar reportes del sistema. No modifica información.
Administrador	Es la persona autorizada a realizar los pagos, trabajar con los presupuestos y hacer cualquier transacción del sistema.



	Además es el encargado de administrar las cuentas de usuarios.
Asistente	Entra datos básicos al sistema. Facturas, empresas, ministerios, cheques, no efectúa pagos ni transacciones.

#### 2.4.2 Lista de casos de uso.

Caso de uso	Gestionar autenticación de los usuarios
Actores	Administrador, Supervisor, Asistente.
Descripción: El Administrador, Supervisor o Asistente introducen su usuario y contraseña para poder acceder a las opciones que le sean permitidas en la aplicación, una vez que se verifiquen estos datos. El Administrador, Supervisor o Asistente puede solicitar cambiar su contraseña de acceso al sistema.	
Referencias	Requerimientos funcionales: 1.

Caso de uso	Gestionar usuarios del sistema.
Actores	Administrador
Descripción: El Administrador gestiona la existencia de los usuarios del sistema, creando nuevos usuarios, modificando la información de los que existen o eliminando aquellos que por diferentes motivos ya no deben tener acceso a la aplicación. El Administrador envía un correo a los diferentes usuarios con la contraseña asignada para poder acceder al sistema.	
Referencias	Requerimientos funcionales: 2, 3.

Caso de uso	Gestionar la información necesaria para la toma de decisiones
Actores	Supervisor, Administrador
Descripción: El Supervisor solicita la ejecución de reportes asociados con el control de cheques, las facturas, los pagos y el presupuesto para utilizarlos en la toma de decisiones.	
Referencias	Requerimientos funcionales: 19 al 25.

Caso de uso	Gestionar Conceptos de Presupuesto
Actores	Administrador, asistente.
Descripción: El administrador y el asistente introducen los datos de los conceptos con los cuales se trabajara en el presupuesto.	
Referencias	Requerimientos funcionales: 6, 7, 16.

Caso de uso	Gestionar empresas.
Actores	Administrador, asistente.
Descripción: El asistente es el responsable de introducir la información necesaria de las diferentes empresas con las que se trabaja. El administrador es el encargado de controlar la información que se cuenta de estas empresas o de los ministerios así como solicitarle al sistema cualquier reporte relacionado con las empresas o los ministerios.	
Referencias	Requerimientos funcionales: 4, 14, 15, 18.

Caso de uso	Gestionar ministerios.
-------------	------------------------

Actores	Administrador, asistente.
Descripción: El asistente es el responsable de introducir la información necesaria de los diferentes Ministerios con las que se trabaja. El administrador es el encargado de controlar la información que se cuenta de de los ministerios así como solicitarle al sistema cualquier reporte relacionado con los ministerios.	
Referencias	Requerimientos funcionales: 5, 13, 17.

Caso de uso	Gestionar facturas
Actores	Administrador, asistente.
Descripción: El asistente es el encargo de introducir la información requerida de las facturas al sistema. El administrador es el encargado de controlar la información que se cuenta de las facturas así como solicitarle al sistema cualquier reporte relacionado con una o varias facturas.	
Referencias	Requerimientos funcionales: 8, 12, 15.

Caso de uso	Gestionar cheques
Actores	Administrador, asistente.
Descripción: El asistente es el encargado de introducir la información requerida de los cheques efectuados al sistema. El administrador y es el encargado de solicitarle al sistema cualquier información de estos cheques así como controlar dicha información.	
Referencias	Requerimientos funcionales: 11, 23.

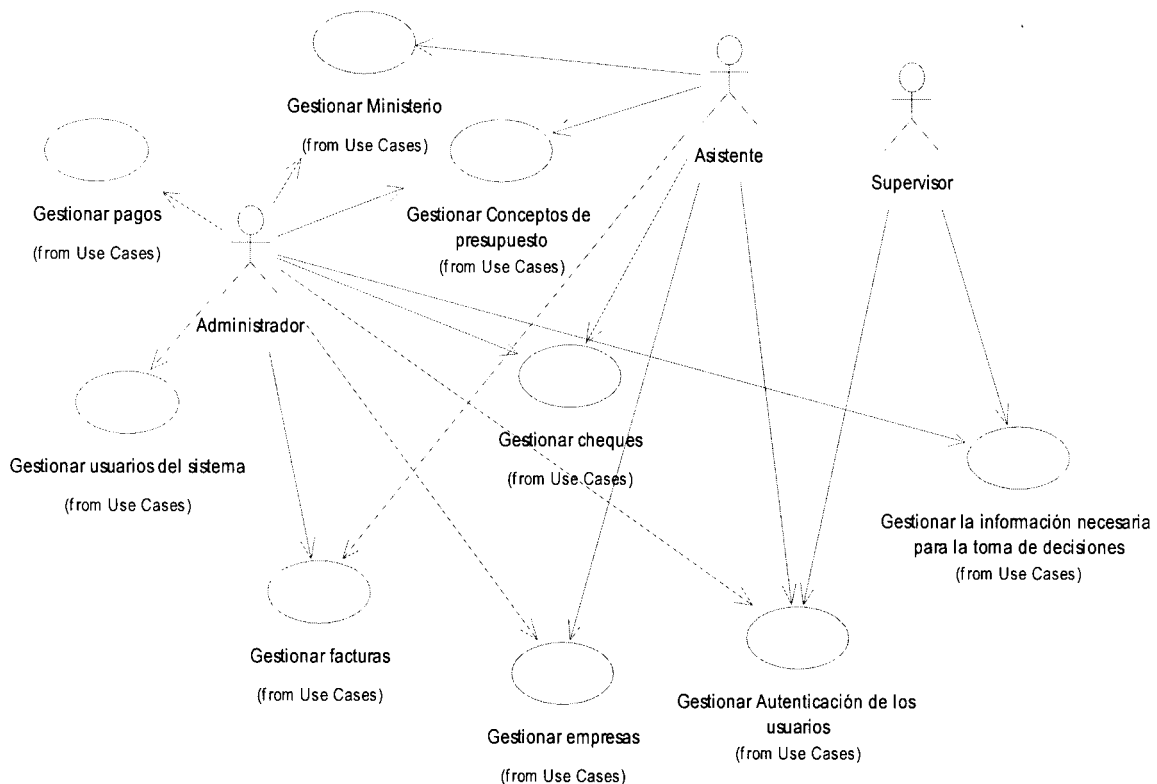
Caso de uso	Gestionar pagos.
Actores	Administrador, asistente.
Descripción: El administrador es el encargado de solicitarle al sistema que le genere la solicitud de pago así como almacenar esta información.	
Referencias	Requerimientos funcionales: 9, 10.

### 2.4.3 Casos de uso por ciclos.

Ciclo	Caso de Uso	Justificación de la selección
1	<ul style="list-style-type: none"> <li>• Gestionar conceptos de presupuestos.</li> <li>• Gestionar Empresas.</li> <li>• Gestionar Facturas.</li> <li>• Gestionar Ministerios.</li>   <li>• Gestionar autenticación de los usuarios.</li>   <li>• Gestionar usuarios del sistema.</li> </ul>	<p>Estos casos de uso influyen en la arquitectura básica del sistema, en ellos están presentes los procesos más complejos y que representan la funcionalidad principal del sistema.</p> <p>Este caso de uso es necesario puesto que solamente se les puede dar acceso a la aplicación a aquellos usuarios autorizados, para de este modo proteger la información que se maneja.</p> <p>Este caso de uso es necesario puesto que los usuarios para poder acceder al sistema necesitan una cuenta de usuario.</p>
2	<ul style="list-style-type: none"> <li>• Gestionar pagos.</li> </ul>	Este caso de uso es necesario porque le permite al administrador del sistema

	<ul style="list-style-type: none"> <li>• Gestionar la información necesaria para la toma de decisiones.</li> <li>• Gestionar cheques.</li> </ul>	<p>obtener el pago correspondiente así como la persistencia de este.</p> <p>Este caso de uso es necesario porque le permite al administrador realizar cualquier búsqueda acerca de una información que brinda el sistema, ya sea un cheque, una factura, un proveedor o una fecha determinada.</p> <p>Este caso de uso es necesario porque le permite al administrador almacenar los datos de los cheques emitidos así como realizar cualquier búsqueda.</p>
--	--	--

## 2.5 Diagrama de casos de uso.



## **Capítulo III. Diseño del Sistema.**

### **Introducción.**

En el presente capítulo se plantea el análisis del sistema, utilizando para su modelado el Lenguaje Unificado de Modelación (UML), que permite representar la expansión de los casos de uso y el modelo conceptual, también se presentarán los diagramas de secuencia y los diagramas de diseño Web del sistema.

### **3.1 Especificación de casos de uso.**

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema.

En el anexo 1 se muestran los casos de uso expandidos del primer ciclo de desarrollo.

### **3.2 Modelo conceptual del Sistema.**

El paso esencial de un análisis orientado a objetos es descomponer el problema en conceptos u objetos individuales. El modelo conceptual es la representación gráfica de los mismos.

En el anexo 2 se muestra modelo conceptual del sistema.

### **3.3 Modelo de clases de análisis del Sistema**

El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

En el anexo 3 se muestra diagrama de clases de análisis del sistema.

### **3.4 Diagramas de secuencia del Sistema.**

Los diagramas de secuencia de un sistema muestran gráficamente los eventos que fluyen de los actores al sistema. Su creación forma parte de la investigación para

conocer el sistema y se incluye dentro del modelo de análisis. Su creación depende de la formulación previa de los casos de uso.

En el anexo 4 se muestran los diagramas de secuencia del sistema.

### **3.5 Diagramas de interacción.**

La vista de interacción describe secuencias de intercambios de mensajes entre los roles que implementan el comportamiento de un sistema. Esta visión proporciona una vista integral del comportamiento del sistema, es decir, muestra el flujo de control a través de muchos objetos. La vista de interacción se exhibe en dos diagramas centrados en distintos aspectos pero complementarios: centrados en los objetos individuales y centrados en objetos cooperantes.

Los diagramas de secuencia muestran la secuencia de mensajes entre objetos durante un escenario concreto (anexo 5). Los diagramas de colaboración son útiles en la fase exploratoria para identificar objetos. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás.

### **3.6 Diagramas de Clases Web.**

Una vez terminados los diagramas de interacción para el ciclo actual de desarrollo de la aplicación, podemos identificar la especificación de las clases de software (y las interfaces) que participan en la solución y complementarlas con detalles de diseño, por ejemplo los métodos.

El diagrama de clases del sistema se puede ver en el anexo 6. Está dividido en varios diagramas para que sea más fácil su comprensión. También se puede ver una descripción de las clases en el anexo 7.

### **3.7 Diagrama de base de datos.**

En el Anexo 8 se muestra el diagrama de base de datos.

En el Anexo 9 se muestra la descripción de las tablas.

### **3.7 Forma general y principios de la protección y seguridad.**

El sistema analizado debe tener determinado nivel de seguridad ya que alguna modificación en la información existente puede crear pérdidas elevadas para la UCI. Para garantizar la seguridad de la información que se encuentra en la base de datos, se crean determinados permisos a los usuarios para el acceso a las funcionalidades del sistema.

En este caso será necesario la creación de nuevas cuentas de usuario para la autenticación en el sistema. Es necesario asignar a cada usuario autorizado los permisos o niveles de acceso. Serán empleados tres niveles: administrador, supervisor y usuario estándar.

- ✓ Usuario estándar: Necesita autenticarse y solo tendrá acceso a introducir la información general con la que se trabaja en la aplicación, esta información llega a través del jefe económico.
- ✓ Supervisor: Necesita autenticarse y solo podrá obtener informes del estado de la información del sistema a fin de realizar análisis de esta.
- ✓ Administrador: Necesita autenticarse y tendrá control total sobre el sistema, puede crear y eliminar cuentas de usuarios.

### **3.8 Forma general del tratamiento de errores.**

El sistema trata de minimizar al máximo los posibles errores que puedan existir. Para ello se decidió, a la hora de realizar operaciones de modificación o eliminación de elementos, mostrar los mismos en una lista desde donde el usuario pueda seleccionar los mismos. De esta manera siempre serán válidos los datos de entrada.

En el caso de los datos introducidos por usuarios del sistema, se hace una validación de los mismos utilizando las facilidades que brinda C# y ASP.Net para la validación de datos entrados por el usuario en los formularios.



## **Conclusiones.**

Una vez realizado el estudio de los diferentes procesos que se llevan a cabo en el control de las inversiones, se ha podido arribar a las siguientes conclusiones:

- Se planteó el desarrollo de un sitio Web que permita de forma fácil y rápida el acceso a la información de las personas autorizadas.
- Realizó un estudio sobre cómo se lleva a cabo este proceso en la actualidad, mediante el cual se pudo definir y especificar las funcionalidades del sistema propuesto.
- Para la implementación de la herramienta se propuso usar ASP.Net como lenguaje de programación y para la modelación en la etapa de Análisis y Diseño se utilizó UML.
- Se realizó la etapa de Análisis y Diseño para los 2 primeros ciclos de desarrollo de la herramienta.
- Se brinda la posibilidad de obtener información estadística que ayudan la toma de decisiones.
- Capacidad de Reportes.

Con el estudio realizado se ha confirmado la necesidad de la existencia de un software que automatice el proceso de control de inversiones. El sistema resultante logrará un procesamiento óptimo de la información y disponibilidad de la misma en todo momento para las personas autorizadas. Además será punto de partida para la incorporación de nuevas funcionalidades.

## **Recomendaciones.**

Se recomienda:

- ✓ Continuar con el cumplimiento de los casos de uso del segundo ciclo de desarrollo del sistema.
- ✓ Que se implemente la ayuda del sistema, para brindarle mayores facilidades a los usuarios.
- ✓ Que se desarrollen, ensamblen y pongan en funcionamiento los demás módulos del sistema.

## Referencias bibliográficas.

1. Microsoft Developer Network. MSDN Library para Visual Studio .NET 2003.
2. <http://www.iscmolina.com/Estado%20del%20Arte.html> (3/3/2004).
3. <http://www.estarinformado.com.ar/pag%20carta%20de%20lectores/tomas2.htm> (3/3/2004).
4. Ing. Alicia Enríquez Perdomo. Modelos de Negocio en Internet, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, 2003.
5. <http://www.forosdelweb.com/> (5/3/2004).
6. <http://samples.getdotnet.com/quickstart/aspplus/doc/secureservices.aspx> (5/3/2004).
7. <http://www.dcc.uchile.cl/~luquerre/cc61j> (3/3/2004).

## **Bibliografía.**

1. Microsoft Developer Network. MSDN Library para Visual Studio .NET 2003.
2. Anónimo. Conferencia de Ingeniería de Software (Curso 2003-2004). Modelamiento del negocio,"Download" Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, 2004.
3. Anónimo. Conferencia de Ingeniería de Software (Curso 2003-2004). Requerimientos,"Download" Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, 2004.
4. Anónimo. Conferencia de Ingeniería de Software (Curso 2003-2004).Flujo de Trabajo: Análisis,"Download" Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, 2004.
5. Anónimo. Conferencia de Ingeniería de Software (Curso 2003-2004). Diseño de Aplicaciones Web,"Download" Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, 2004.
6. Anónimo. Conferencia de Ingeniería de Software (Curso 2003-2004). Flujo de implementación,"Download" Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, 2004.
7. Anónimo. Conferencia de Ingeniería de Software (Curso 2003-2004). Flujo de Prueba,"Download" Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, 2004.
8. Ortín Ibáñez, María José. Modelando aplicaciones Web con UML, Parte I y II."Download" <http://scriba> .05/04/2004.

## **Glosario de términos.**

A.

ADO.NET: Conjunto de tecnologías de acceso a datos incluidas en las bibliotecas de clases de .NET Framework que proporcionan acceso a datos relacionales y a XML. ADO.NET se compone de clases que forman el DataSet (como tablas, filas, columnas, relaciones, etcétera), de proveedores de datos de .NET Framework y de definiciones de tipos personalizados (como SqlTypes para SQL Server).

ASP.NET: Conjunto de tecnologías de Microsoft .NET Framework para crear aplicaciones Web y servicios Web XML. Las páginas ASP.NET se ejecutan en el servidor y generan lenguaje de marcado (como HTML, WML o XML) que se envía a un explorador móvil o de escritorio. Las páginas ASP.NET utilizan un modelo de programación compilado y basado en eventos que mejora el rendimiento y permite la separación de la lógica de aplicación y de la interfaz de usuario. Las páginas ASP.NET y los archivos de servicios Web XML creados mediante ASP.NET contienen lógica del servidor (y no del cliente) escrita en Visual Basic .NET, C# .NET o en cualquier lenguaje compatible con .NET. Las aplicaciones Web y los servicios Web XML aprovechan las funciones de Common Language Runtime, como la seguridad de tipos, la herencia, la interoperabilidad entre lenguajes, el control de versiones y la seguridad integrada.

C.

C#: Nuevo lenguaje de programación diseñado para crear aplicaciones empresariales que se ejecutan en .NET Framework. C#, que es una evolución de C y C++, garantiza la seguridad de tipos y está orientado a objetos. Puesto que se compila como código administrado, aprovecha los servicios de Common Language Runtime, como interoperabilidad de lenguaje, mayor seguridad y recolección de elementos no utilizados.

F.

Firewalls: Con el gráfico término - firewalls o cortafuegos - se designa a una utilidad informática que se encarga de aislar redes o sistemas informáticos respecto de otros sistemas que se encuentran en la misma red. Constituyen una especie de

"barrera lógica" delante de nuestros sistemas que examina todos y cada uno de los paquetes de información que tratan de atravesarla.

H.

HTTP:(HyperText Transfer Protocol; Protocolo de transferencia de hipertexto): Es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida. Ha sido usado por el World-Wide Web desde 1990.

I.

Intranet: Red privada, desarrollada dentro de una compañía que utiliza el mismo software y provee de información similar que Internet, solo que es únicamente para el uso interno.

L.

Lenguaje de marcado extensible (XML, Extensible Markup Language): Subconjunto del Lenguaje de marcado generalizado estándar (SGML) optimizado para su uso a través del Web. XML proporciona un método uniforme para describir e intercambiar datos estructurados que es independiente de las aplicaciones o los proveedores.

S.

Servicios Web XML (XML Web services): Unidades de lógica de aplicaciones que proporcionan datos y servicios a otras aplicaciones. Las aplicaciones obtienen acceso a los servicios Web XML mediante protocolos Web estándar y formatos de datos como HTTP, XML y SOAP, con independencia de cómo se implementa cada servicio Web XML. Los servicios Web XML combinan los mejores aspectos del desarrollo basado en componentes y el Web, por lo que son una base fundamental del modelo de programación de Microsoft .NET.

SOAP: Protocolo simple basado en XML para intercambiar información estructurada y de tipos en el Web. El protocolo no contiene semántica de aplicación ni de transporte, por lo que resulta muy modular y extensible.

## **Anexos.**

### **Anexos 1: Especificación de los Casos de Uso.**

Caso de uso	Gestionar autenticación de los usuarios.
Actores	Administrador, Supervisor, Asistente.
Propósito	Realizar la autenticación de los usuarios del sistema y permitir que estos puedan cambiar su contraseña para acceder al mismo.
Resumen: El Supervisor, Asistente o Administrador introducen su usuario y contraseña para poder acceder a las opciones que le sean permitidas en la aplicación, una vez que se verifiquen estos datos. El Supervisor, Asistente o Administrador pueden solicitar cambiar su contraseña de acceso al sistema.	
Precondiciones:-	
Acción del actor	Respuesta del sistema
1) El usuario le solicita al sistema autenticarse.	2) El sistema muestra un formulario con lo datos requeridos para esta acción.
3) El usuario llena los datos del formulario: nombre y contraseña.	3) El sistema recepciona la información. 4) El sistema chequea si el usuario existe. 5) Los datos del usuario son correctos. 6) El sistema verifica el privilegio de este usuario. 7) El sistema aplica las restricciones necesarias en caso requeridas. 8) El sistema lo envía a la página principal del sitio.

**Poscondiciones:** El sistema le muestra al usuario la página principal del sitio en espera de la nueva solicitud del usuario.

**Flujo alternativo (5)**

6) El usuario no existe o sus datos son incorrectos.

7) El sistema le envía un mensaje de error al usuario indicándole que su nombre o contraseña son incorrectos.

8) El sistema le muestra el formulario inicial para que vuelva a introducir su nombre y contraseña.

**Poscondiciones:** El sistema le muestra al usuario la página de inicio de sesión al sitio.

**Flujo alternativo (9)**

9) El usuario le solicita al sistema cambiar su contraseña de inicio de sesión.

10) El sistema muestra un formulario con los datos requeridos para esta acción:

- Usuario.
- Contraseña.
- Nueva Contraseña

11) El usuario llena los datos del formulario.

12) El sistema recepciona la información.

13) El sistema chequea si el usuario existe.

14) Los datos del usuario son correctos.

15) El sistema almacena la información



	<p>entrada por el usuario.</p> <p>16) El sistema lo envía a la página principal del sitio.</p>
<p><b>Poscondiciones:</b> El sistema le muestra al usuario la página principal del sitio en espera de la nueva solicitud del usuario.</p>	

Caso de uso	Gestionar usuarios del sistema.
Actores	Administrador
Propósito	Administrar las cuentas de los usuarios que pueden acceder al sistema así como los grupos a los que pertenecen.
<p>Resumen: El Administrador gestiona la existencia de los usuarios del sistema, creando nuevos usuarios, modificando la información de los que existen o eliminando aquellos que por diferentes motivos ya no deben tener acceso a la aplicación. El Administrador envía un correo a los diferentes usuarios con la contraseña asignada para poder acceder al sistema.</p>	
<p>Precondiciones: El administrador debe estar autenticado con anterioridad.</p>	
Acción del actor	Respuesta del sistema
1) El administrador solicita administrar las cuentas de usuarios.	<p>2) El sistema muestra un formulario con los datos iniciales de la solicitud.</p> <ul style="list-style-type: none"> <li>• Crear un usuario.</li> <li>• Modificar un usuario.</li> <li>• Eliminar un usuario.</li> </ul>
3) El administrador selecciona crear un usuario.	<p>4) El sistema le muestra un formulario con los datos correspondientes a la entrada de datos de los usuarios.</p> <ul style="list-style-type: none"> <li>• Nombre.</li> </ul>

	<ul style="list-style-type: none"> <li>• Grupo.</li> <li>• Login.</li> <li>• Contraseña.</li> </ul>
5) El administrador llena los datos del formulario.	<p>6) El sistema recibe la información.</p> <p>7) El sistema verifica que los datos sean correctos.</p> <p>8) El sistema ejecuta la acción.</p> <p>9) El sistema le envía un mensaje con los resultados.</p>
<b>Poscondiciones:</b> El sistema le muestra al usuario la página de administración de cuentas de usuario.	
<b>Flujo alternativo (8)</b>	
	<p>8) Los datos entrados por el usuario son incorrectos.</p> <p>9) El sistema le muestra al usuario el formulario de crear un usuario con los errores en los datos.</p>
<b>Poscondiciones:</b> El sistema le muestra al usuario la página de crear las cuentas de usuarios.	
<b>Flujo alternativo (3)</b>	
3) El administrador selecciona modificar un usuario.	4) El sistema le muestra un formulario con las cuentas de usuarios del sistema.
5) Selecciona el usuario a modificar.	6) El sistema le muestra los datos de este usuario en el formulario.
7) Modifica los datos (Nombre, Login,	8) El sistema recibe los datos del

Grupo, Contraseña).	<p>usuario.</p> <p>9) El sistema modifica los datos del usuario seleccionado con los nuevos datos.</p> <p>10) El sistema lo envía a la página de administración de cuentas de usuarios.</p>
<b>Poscondiciones:</b> El sistema le muestra al usuario la página de administrar las cuentas de usuarios.	
<b>Flujo alternativo (3)</b>	
3) El administrador selecciona eliminar un usuario.	4) El sistema le muestra un formulario con las cuentas de usuarios del sistema.
5) Selecciona el usuario que desea eliminar.	<p>6) El sistema elimina el usuario seleccionado.</p> <p>7) El sistema lo envía a la página de administración de cuentas de usuarios.</p>
<b>Poscondiciones:</b> El sistema le muestra al usuario la página de administrar las cuentas de usuarios.	

Caso de uso	Gestionar Conceptos del Presupuesto.
Actores	Administrador, asistente.
Propósito	Crear y organizar los conceptos por los que se trabajara en la etapa. Mantener un control del estado del presupuesto así como insertar datos de este. También se define la etapa en cual se trabaja.

Resumen: El administrador tiene la responsabilidad de definir la etapa en la que se trabaja así como añadir la información requerida respecto al presupuesto. El asistente es el encargado de introducir los datos de los conceptos con los cuales se trabajará en la etapa.

Precondiciones: El usuario debe estar autenticado con anterioridad.

Acción del actor	Respuesta del sistema
1) El usuario solicita ver información del presupuesto.	2) El sistema muestra un formulario con el tipo de solicitud: <ul style="list-style-type: none"> <li>• Adicionar un presupuesto.</li> <li>• Crear Etapa.</li> </ul>
3) El usuario selecciona adicionar presupuesto.	4) El sistema muestra un formulario con los datos para adicionar un presupuesto. <ul style="list-style-type: none"> <li>• Etapa.</li> <li>• PresupuestoPadre.</li> <li>• Valor.</li> </ul>
5) El usuario llena los datos del formulario.	6) El sistema recibe la información. 7) El sistema almacena la información entrada por el usuario. 8) El sistema le muestra la página de adicionar un presupuesto.
<b>Poscondiciones:</b> El sistema le muestra al usuario la página de adicionar un presupuesto.	
<b>Flujo alternativo (3)</b>	
3) El usuario selecciona crear etapa.	4) El sistema muestra un formulario con los datos para crear una nueva etapa. <ul style="list-style-type: none"> <li>• Nombre de la Etapa.</li> <li>• Fecha de inicio.</li> <li>• Fecha de terminación.</li> </ul>

5) El usuario llena los datos del formulario.	6) El sistema recibe la información. 7) El sistema crea una nueva etapa de trabajo y almacena la información entrada por el usuario. 8) El sistema lo envía a la página principal del sitio.
<b>Poscondiciones:</b> El sistema le muestra al usuario la página principal del sitio.	

Caso de uso	Gestionar empresas.
Actores	Administrador, asistente.
Propósito	Adicionar la información de las empresas con las que se trabaja así como controlar la información de estas empresas y los ministerios.

Resumen: El asistente es el responsable de introducir la información necesaria de las diferentes empresas con las que se trabaja. El administrador y/o el supervisor son los encargados de controlar la información que se cuenta de estas empresas o de los ministerios.

Precondiciones: El usuario debe estar autenticado con anterioridad.

Acción del actor	Respuesta del sistema
1) El usuario solicita gestionar empresas.	2) El sistema muestra un formulario con los datos iniciales de la solicitud: <ul style="list-style-type: none"> <li>• Adicionar empresa.</li> <li>• Modificar empresa.</li> <li>• Adicionar contacto.</li> </ul>
3) El usuario selecciona adicionar empresa.	4) El sistema muestra un formulario con los datos a llenar por el usuario. <ul style="list-style-type: none"> <li>• Nombre de la empresa.</li> <li>• Ministerio.</li> </ul>

5) El usuario llena los datos del formulario.	6) El sistema recepciona la información. 7) El sistema almacena la información entrada por el usuario. 8) El sistema lo envía a la página principal del sitio.
<b>Poscondiciones:</b> El sistema le muestra al usuario la página principal del sitio.	
<b>Flujo alternativo (3)</b>	
3) El usuario selecciona modificar una empresa.	4) El sistema muestra un formulario con todas las empresas.
5) El usuario selecciona la empresa a modificar.	6) El sistema muestra un formulario con los datos de la empresa seleccionada a modificar.
7) El usuario modifica los datos de la empresa.	8) El sistema recepciona la información. 9) El sistema modifica la información existente en la base de datos con los datos entrados por el usuario. 10) El sistema lo envía a la página principal del sitio.
<b>Poscondiciones:</b> El sistema le muestra al usuario la página principal del sitio.	
<b>Flujo alternativo (3)</b>	
3) El usuario selecciona adicionar un contacto.	4) El sistema muestra un formulario con los datos a llenar por el usuario. <ul style="list-style-type: none"> <li>• Nombre del contacto.</li> <li>• Dirección.</li> <li>• Teléfono.</li> <li>• Fax.</li> <li>• E-Mail.</li> <li>• Empresa.</li> </ul>
5) El usuario llena los datos del formulario.	6) El sistema recepciona la información. 7) El sistema almacena la información entrada por el usuario.

	8) El sistema le muestra la página de adicionar un contacto.
<b>Poscondiciones:</b> El sistema le muestra al usuario la página de registrar un contacto.	

Caso de uso	Gestionar ministerios.
Actores	Administrador, asistente.
Propósito	Adicionar la información de los ministerios con los que se trabaja así como controlar la información de los ministerios.

**Resumen:** El asistente es el responsable de introducir la información necesaria de los diferentes ministerios con los que se trabaja. El administrador es el encargado de controlar la información que se cuenta de los ministerios.

**Precondiciones:** El usuario debe estar autenticado con anterioridad.

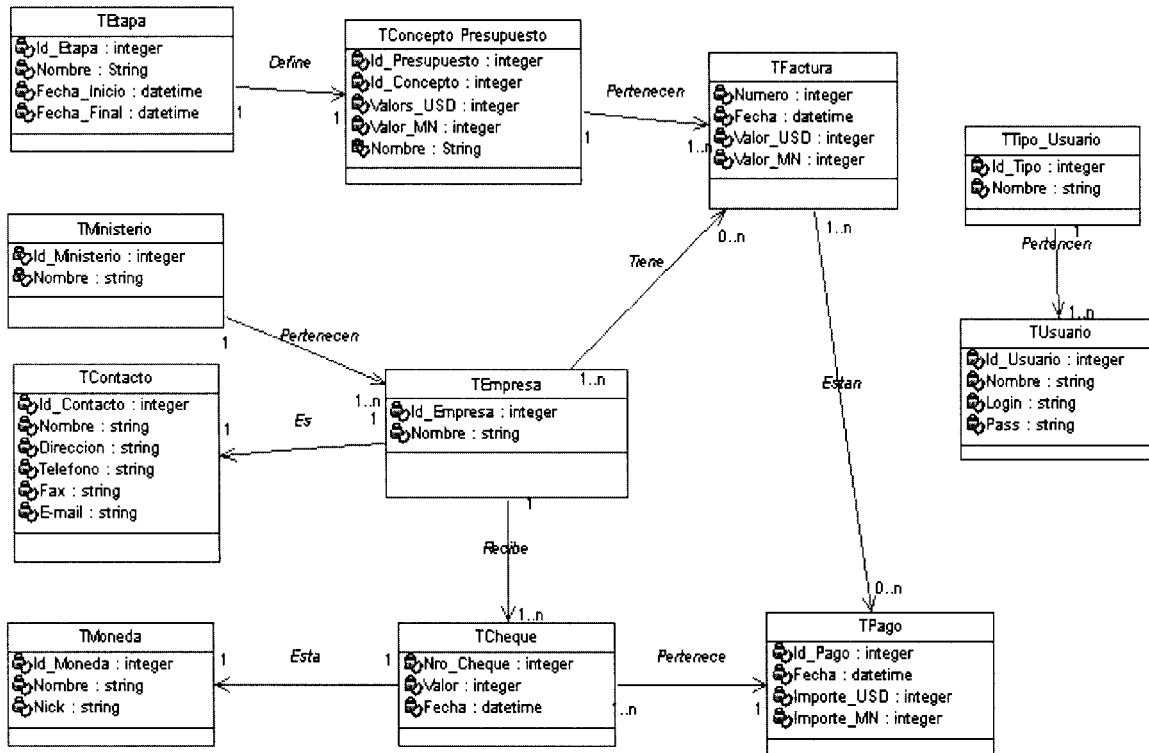
Acción del actor	Respuesta del sistema
1) El usuario solicita gestionar ministerios.	2) El sistema muestra un formulario con los datos necesarios: <ul style="list-style-type: none"> <li>• Nombre del ministerio.</li> </ul>
3) El usuario llena los datos del formulario.	4) El sistema recepciona la información. 5) El sistema almacena la información entrada por el usuario.  6) El sistema lo envía a la página principal del sitio.

**Poscondiciones:** El sistema le muestra al usuario la página principal del sitio.

Caso de uso	Gestionar facturas.	
Actores	Administrador, asistente.	
Propósito	Adicionar los datos de las facturas así como realizar cualquier reporte relacionado con las facturas.	
Resumen: El asistente es el encargado de introducir la información requerida de las facturas al sistema. El administrador y/o supervisor son los encargados de controlar la información que se cuenta de las facturas así como solicitarle al sistema cualquier reporte relacionado con una o varias facturas.		
Precondiciones: El usuario debe estar autenticado con anterioridad.		
	Acción del actor	Respuesta del sistema
	1) El usuario le solicita al sistema gestionar facturas.	2) El sistema muestra un formulario con los datos necesarios para adicionar una factura: <ul style="list-style-type: none"> <li>• Empresa.</li> <li>• Presupuesto.</li> <li>• Importe.</li> </ul>
	3) El usuario llena los datos del formulario.	4) El sistema recepciona la información. 5) El sistema almacena la información entrada por el usuario. 6) El sistema lo envía a la página principal del sitio.
<b>Poscondiciones:</b> El sistema le muestra al usuario la página principal del sitio.		



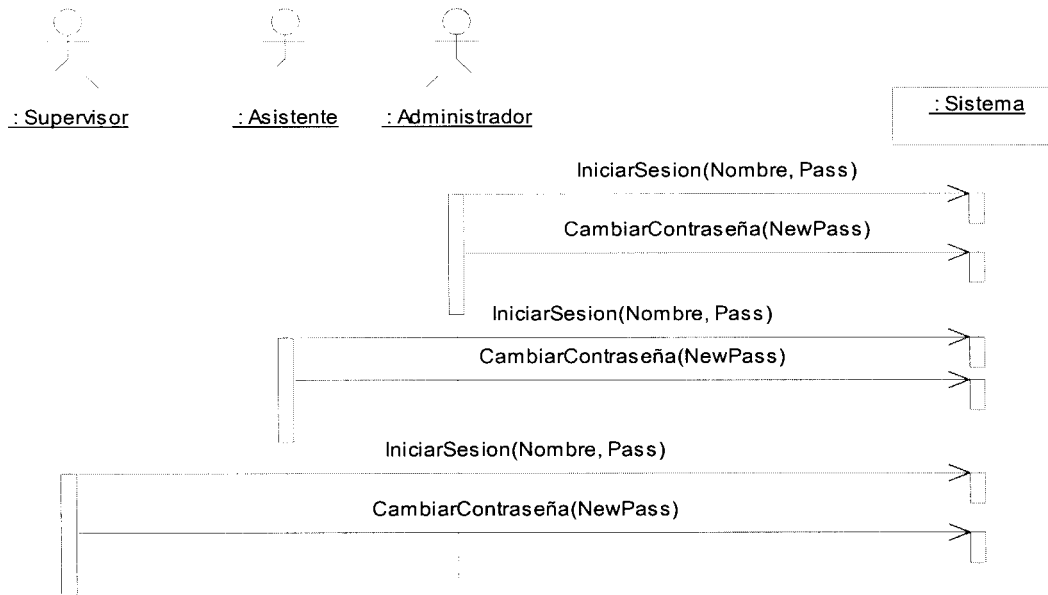
## Anexo 2: Modelo conceptual del Sistema.



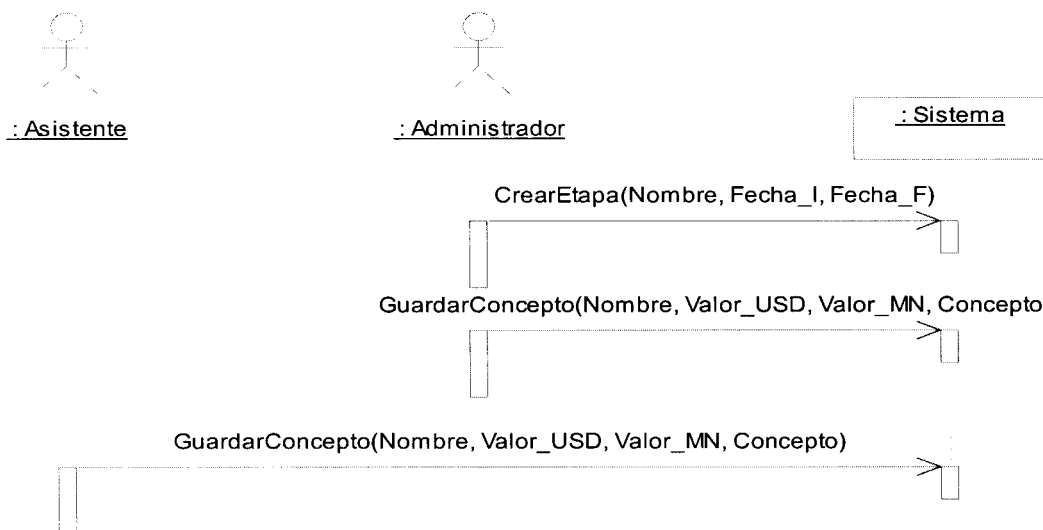


#### **Anexo 4: Diagramas de secuencia del Sistema.**

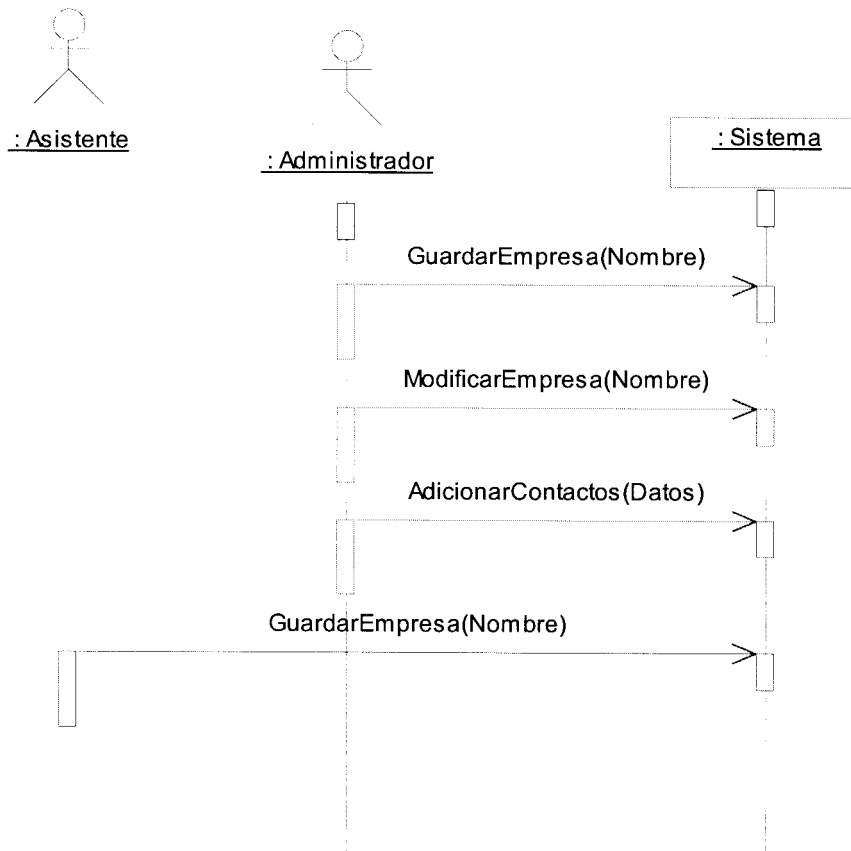
✓ Diagrama de secuencia del caso de uso “Gestionar Autenticación de los usuarios”.



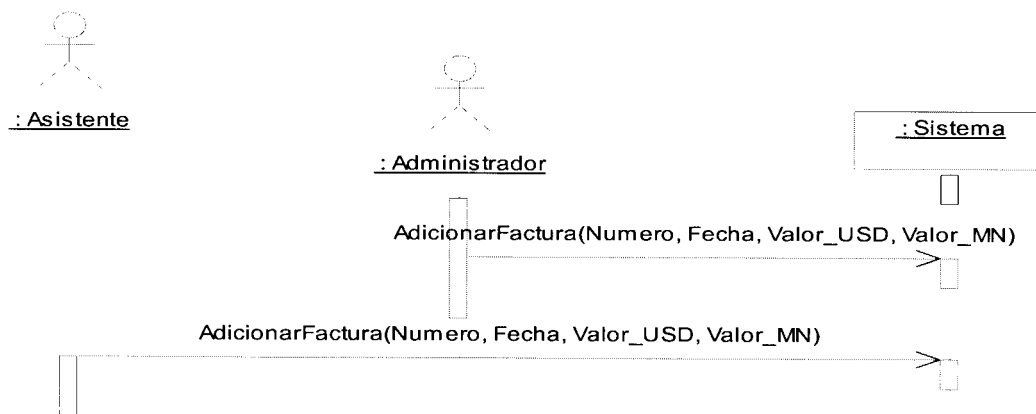
✓ Diagrama de secuencia del caso de uso “Gestionar Conceptos del presupuesto”.



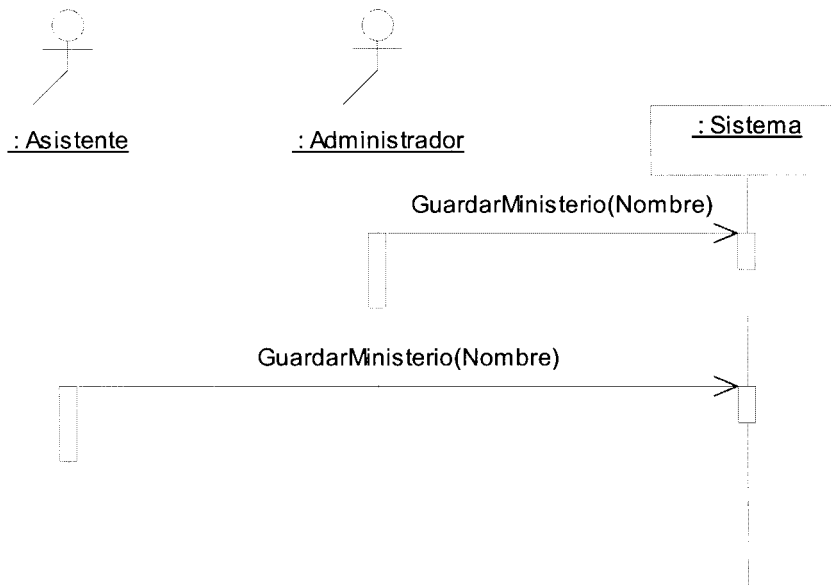
✓ Diagrama de secuencia del caso de uso "Gestionar Empresas".



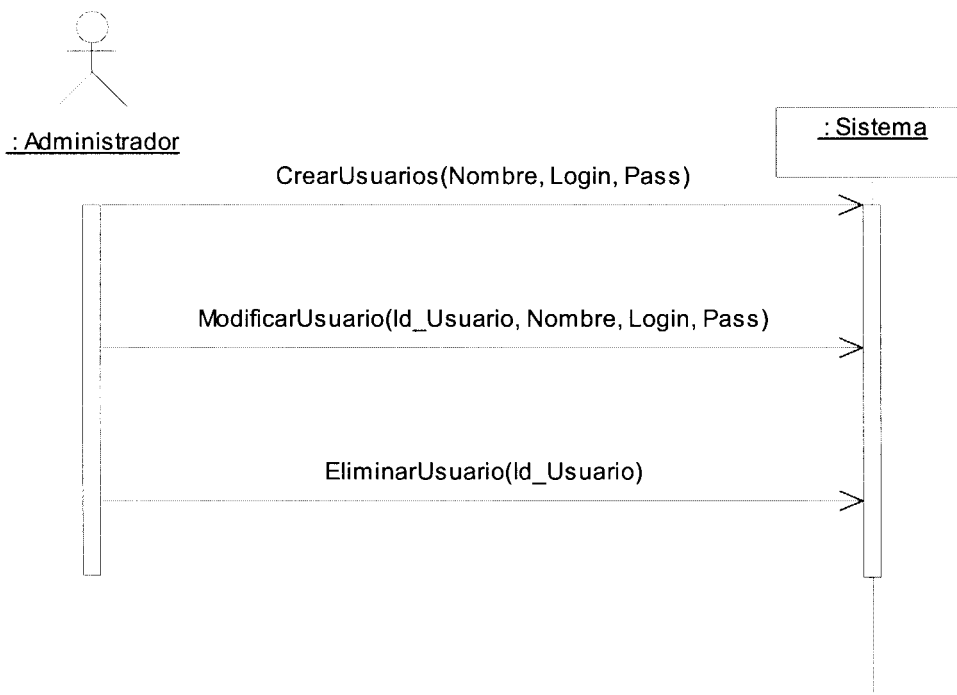
✓ Diagrama de secuencia del caso de uso "Gestionar Facturas".



✓ Diagrama de secuencia del caso de uso "Gestionar Ministerios".

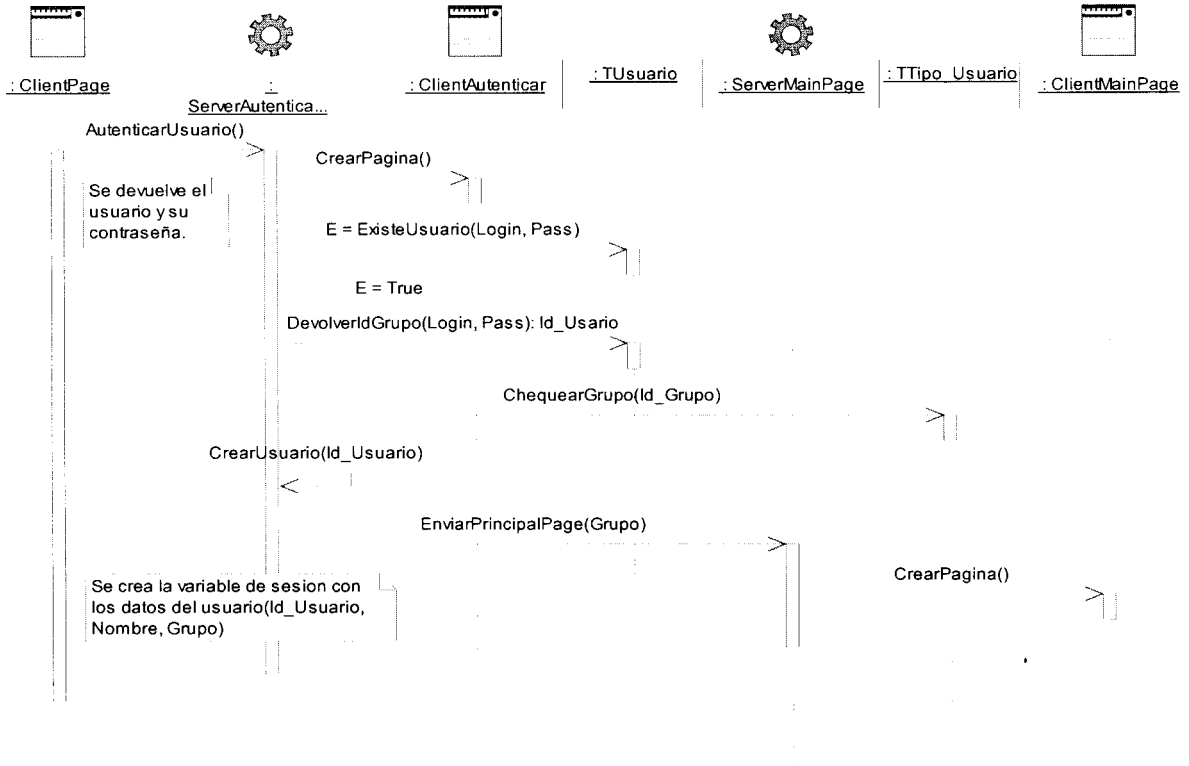


✓ Diagrama de secuencia del caso de uso "Gestionar Usuarios del sistema".

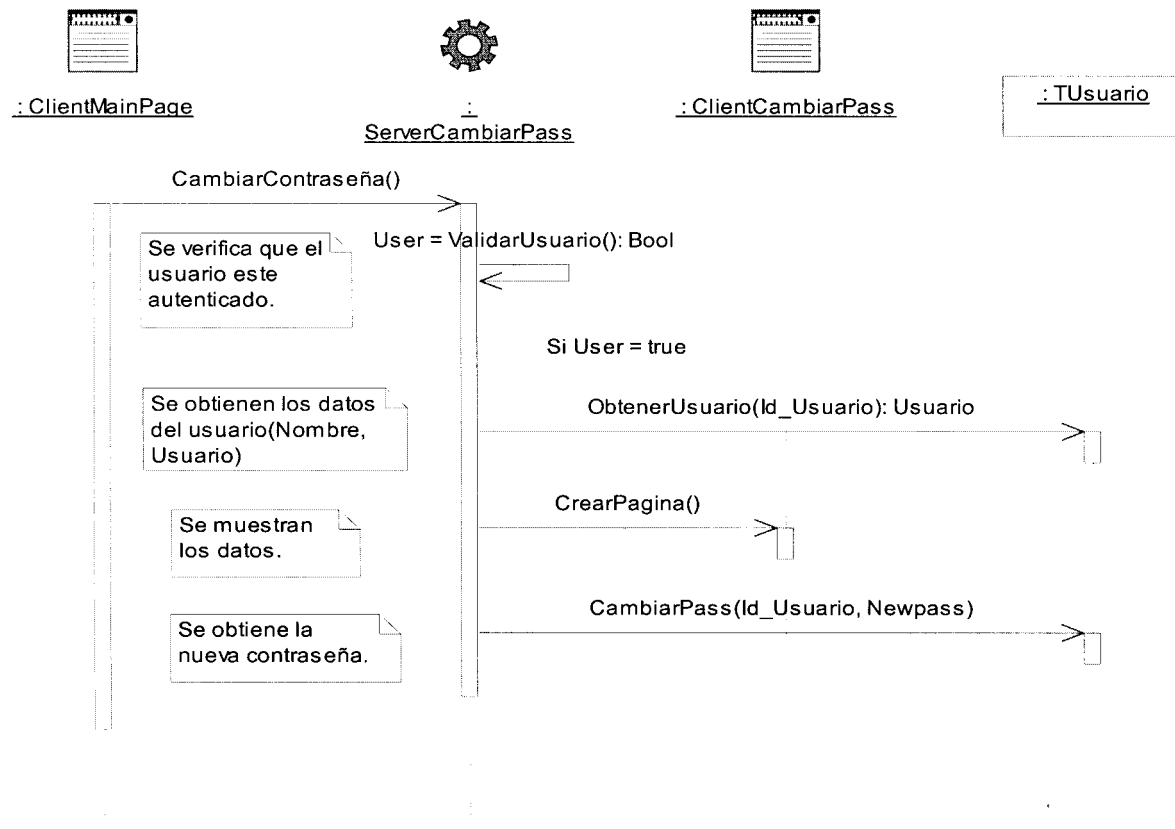


## Anexo 5: Diagramas de Interacción.

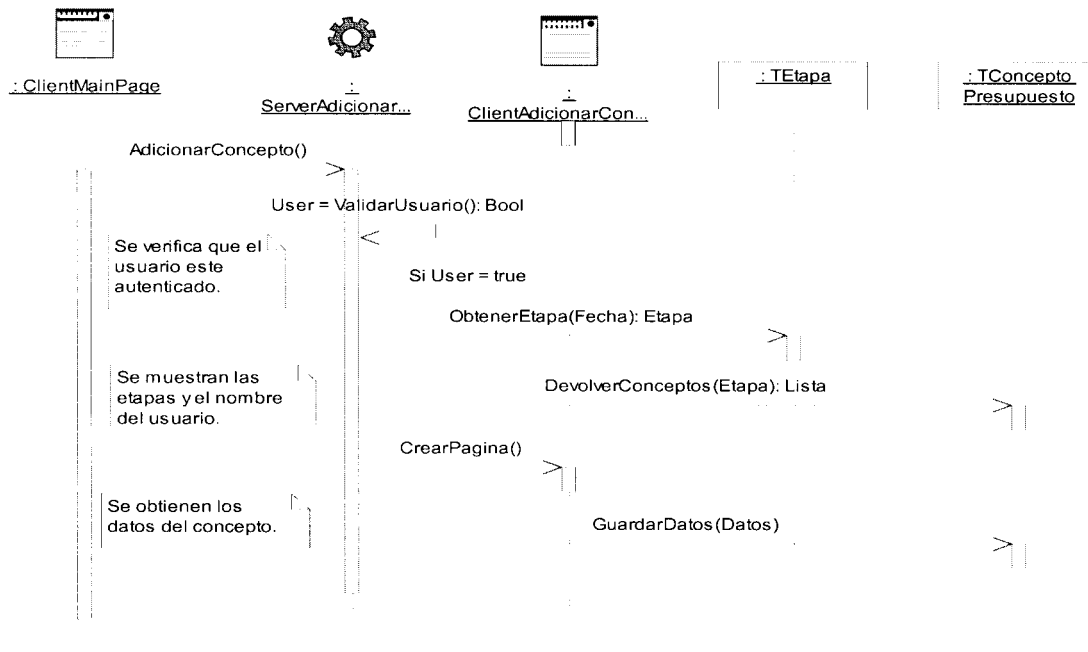
DI: AutenticarUsuario.



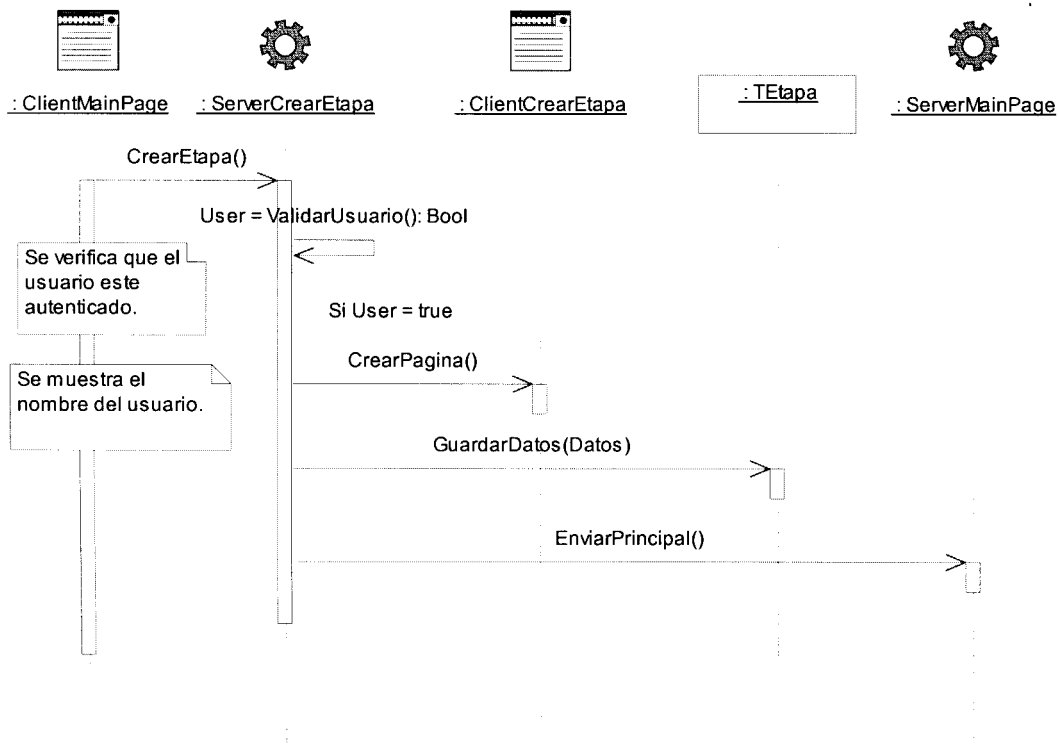
## DI: CambiarContraseña.



## DI: AdicionarConceptos.

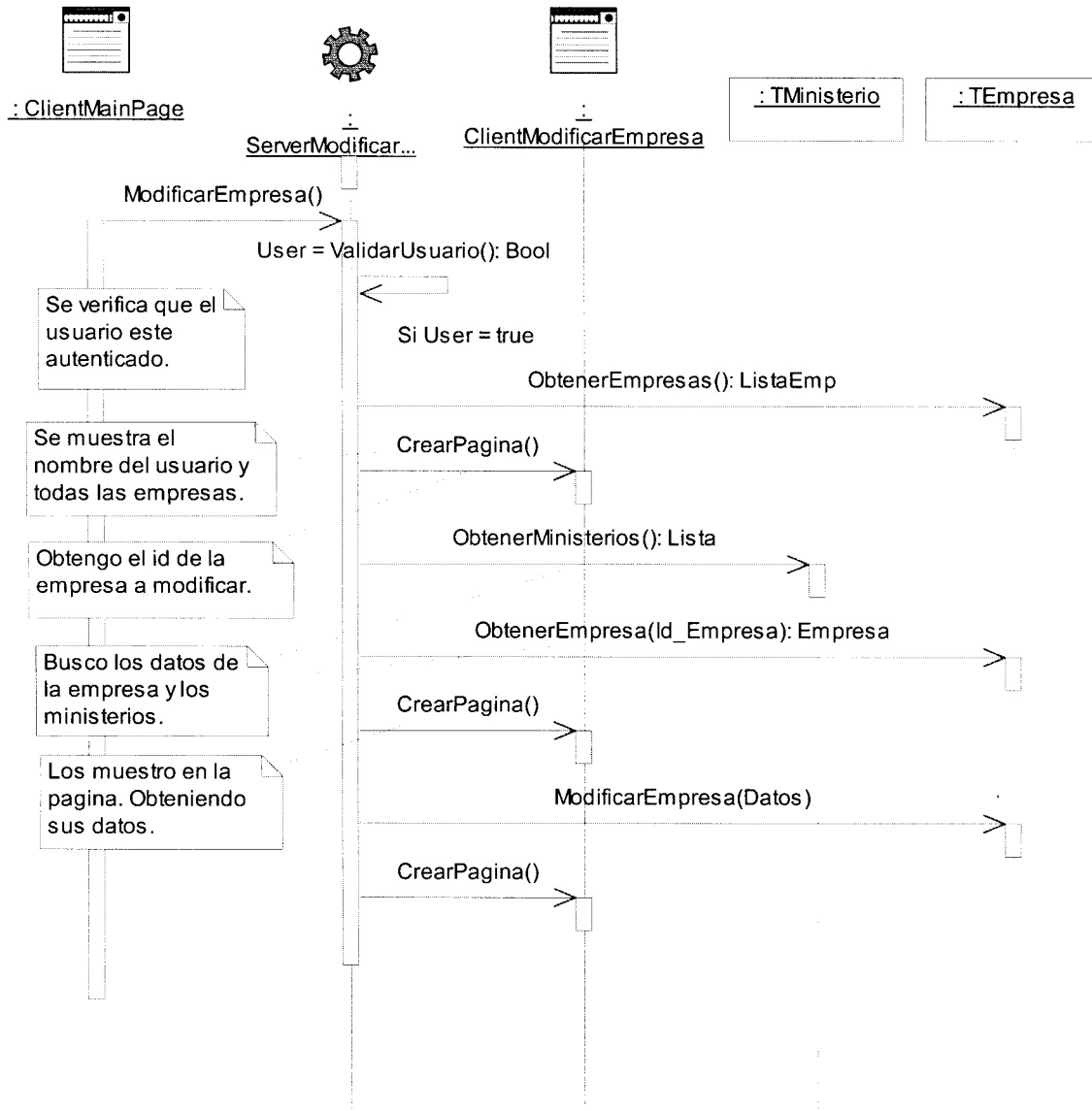


## DI: CrearEtapa.

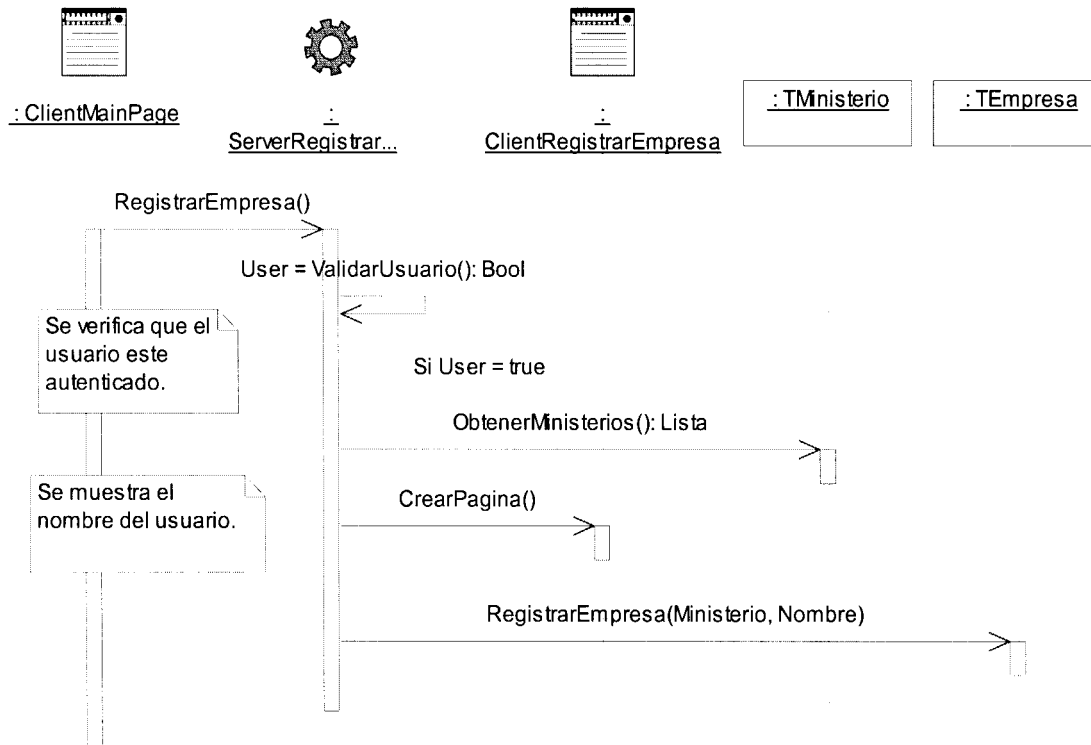




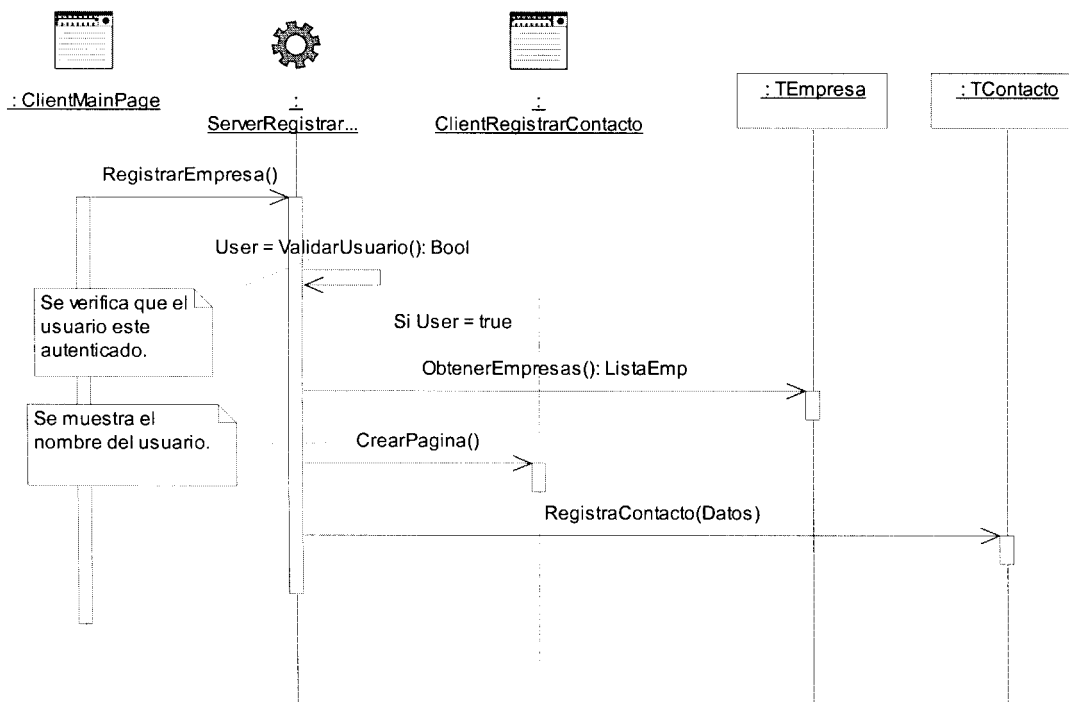
DI: ModificarEmpresa.



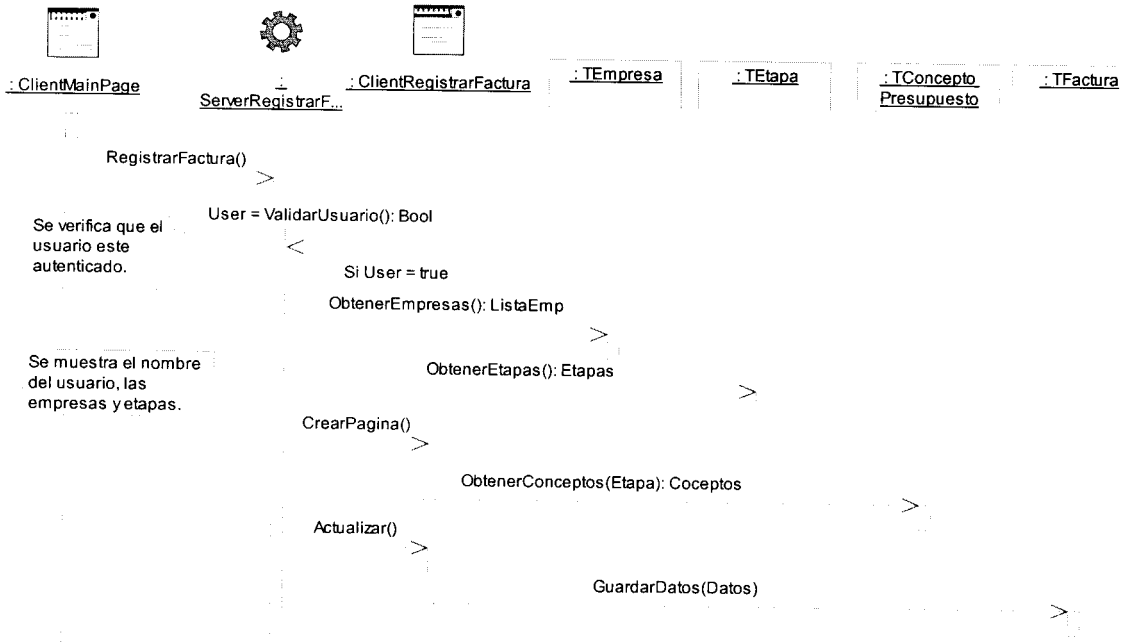
## DI: RegistrarEmpresa.



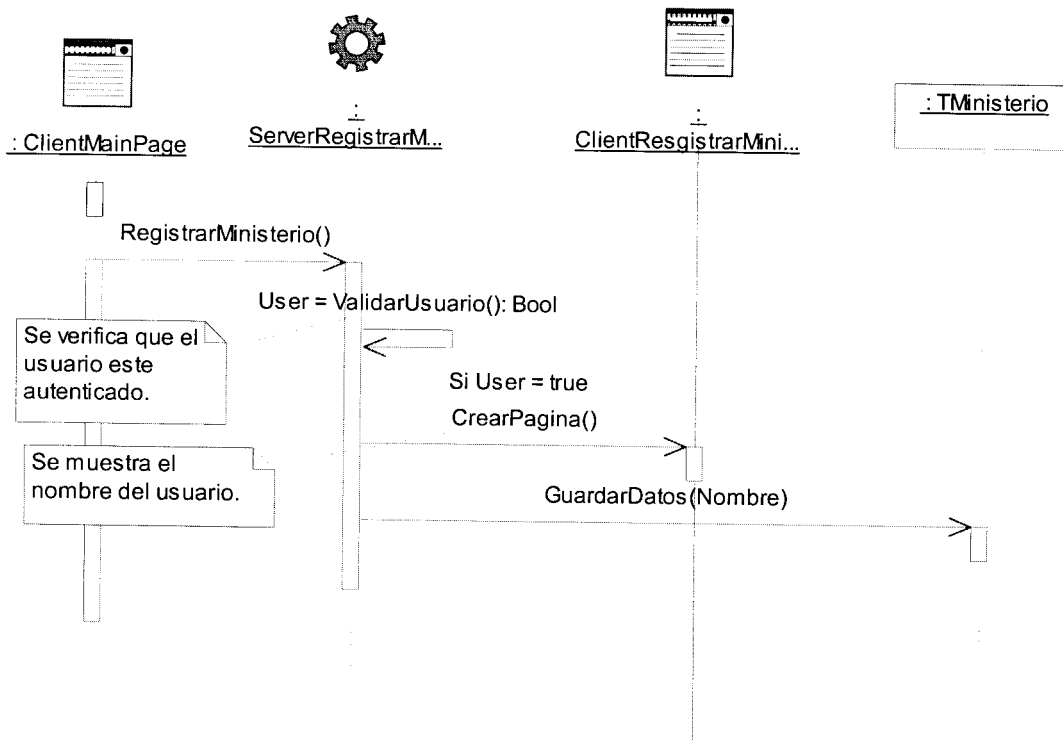
## DI: RegistrarContacto.



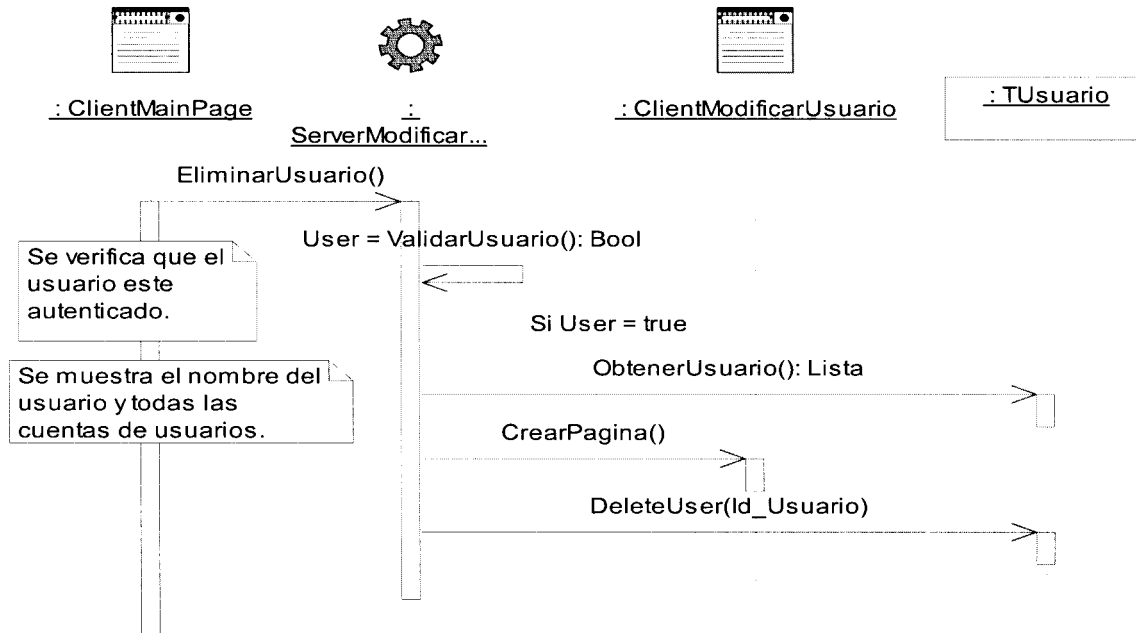
## DI: RegistrarFactura.



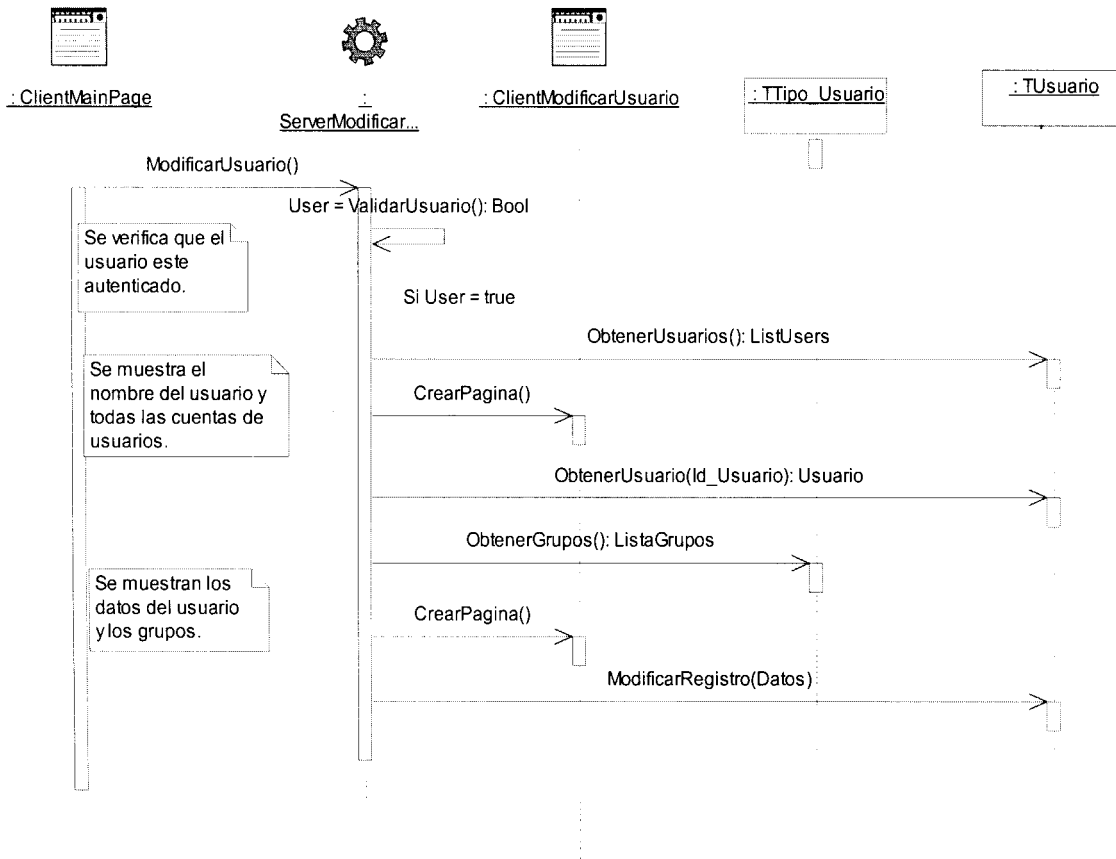
## DI: RegistrarMinisterio.



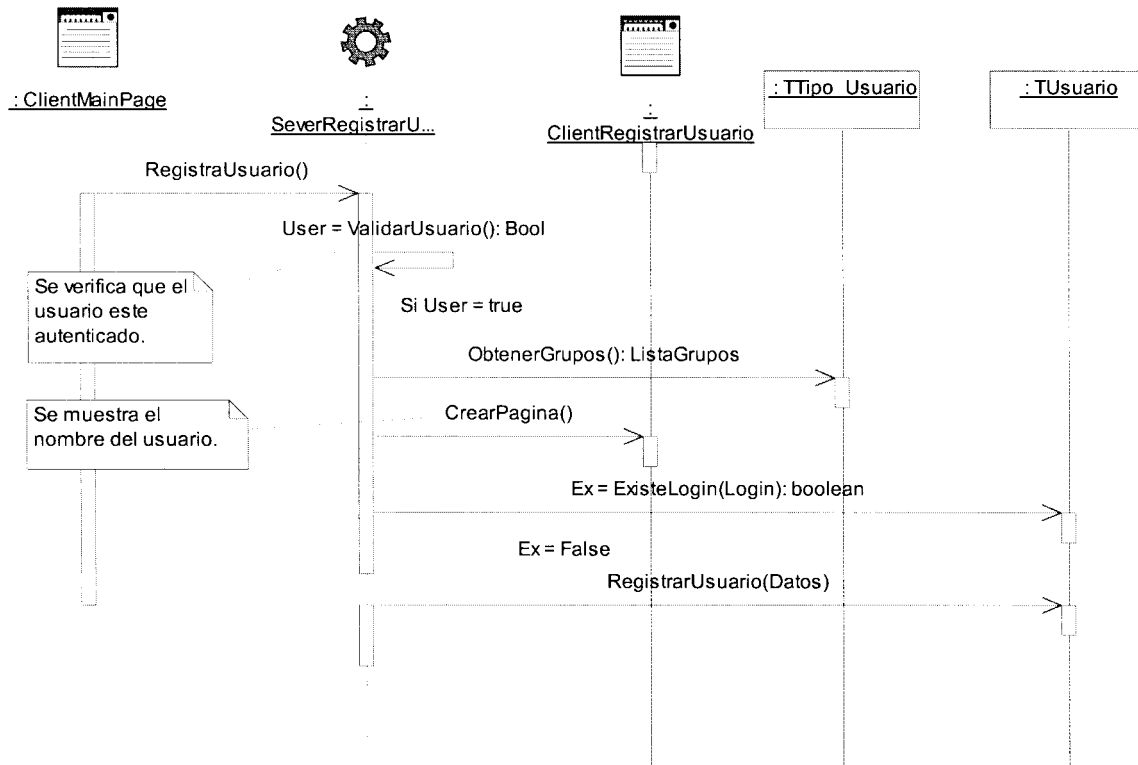
### DI: EliminarUsuario.



### DI: ModificarUsuario.

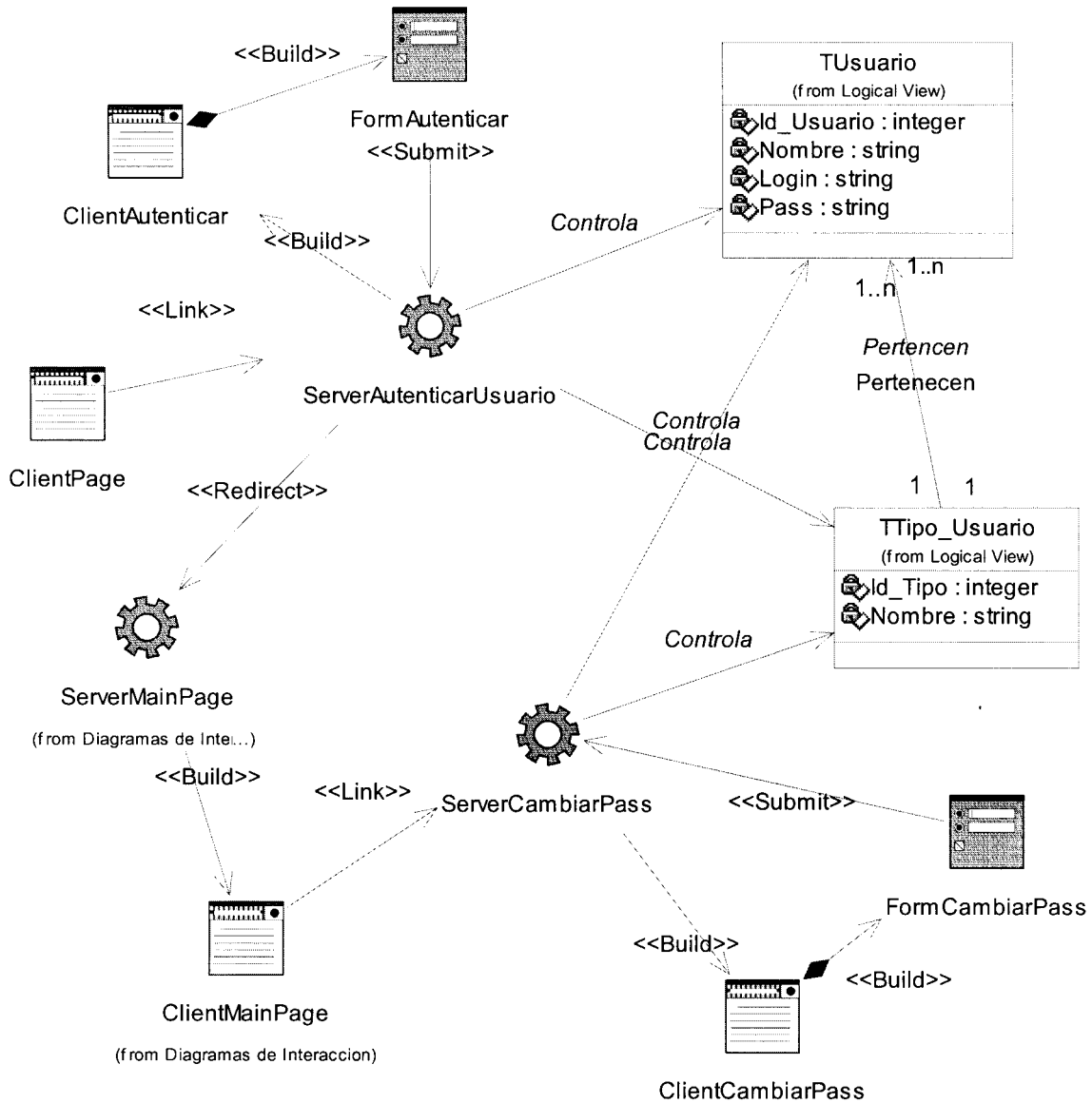


## DI: RegistrarUsuario.

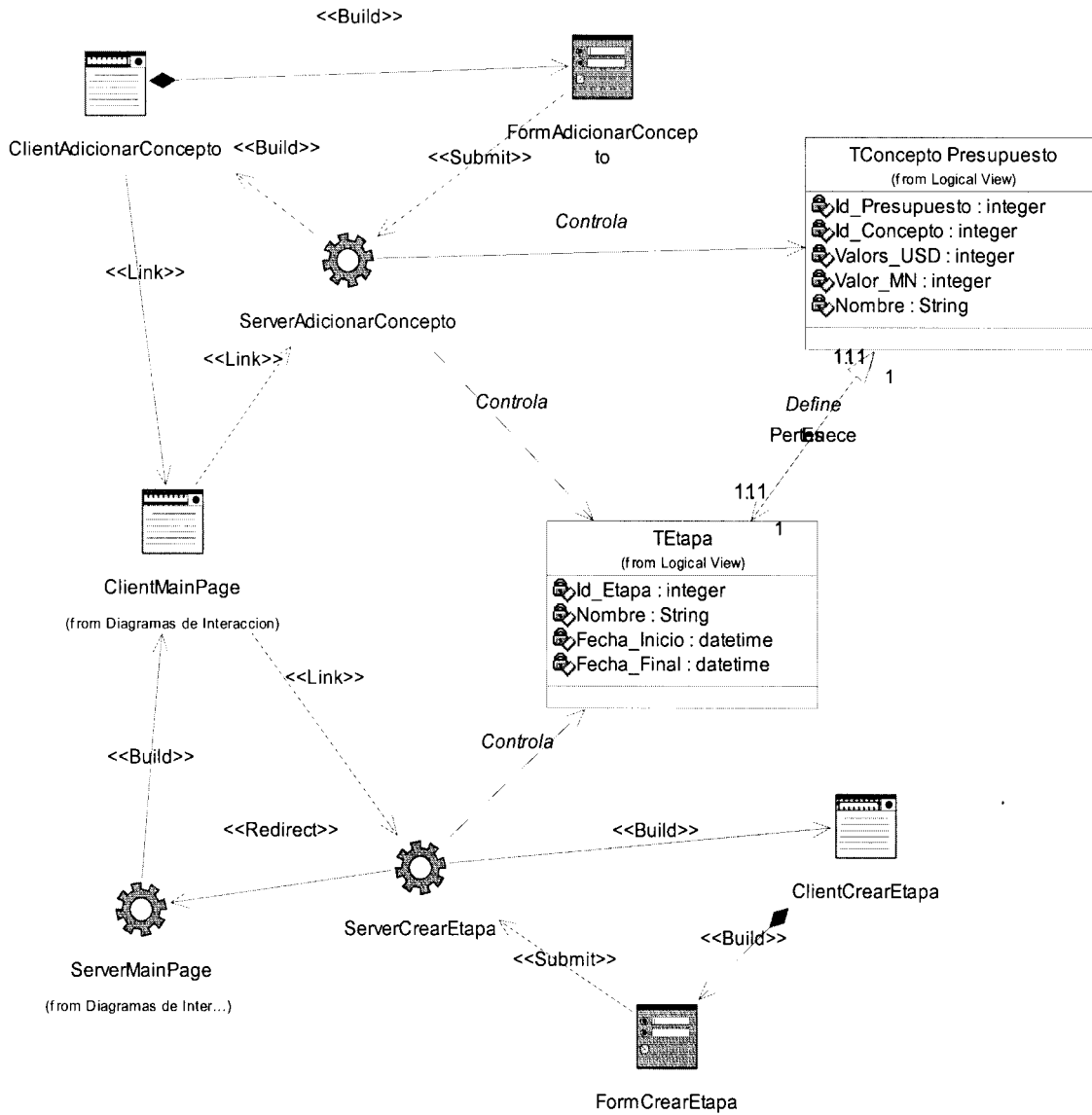


## Anexo 6: Diagramas de Clases Web.

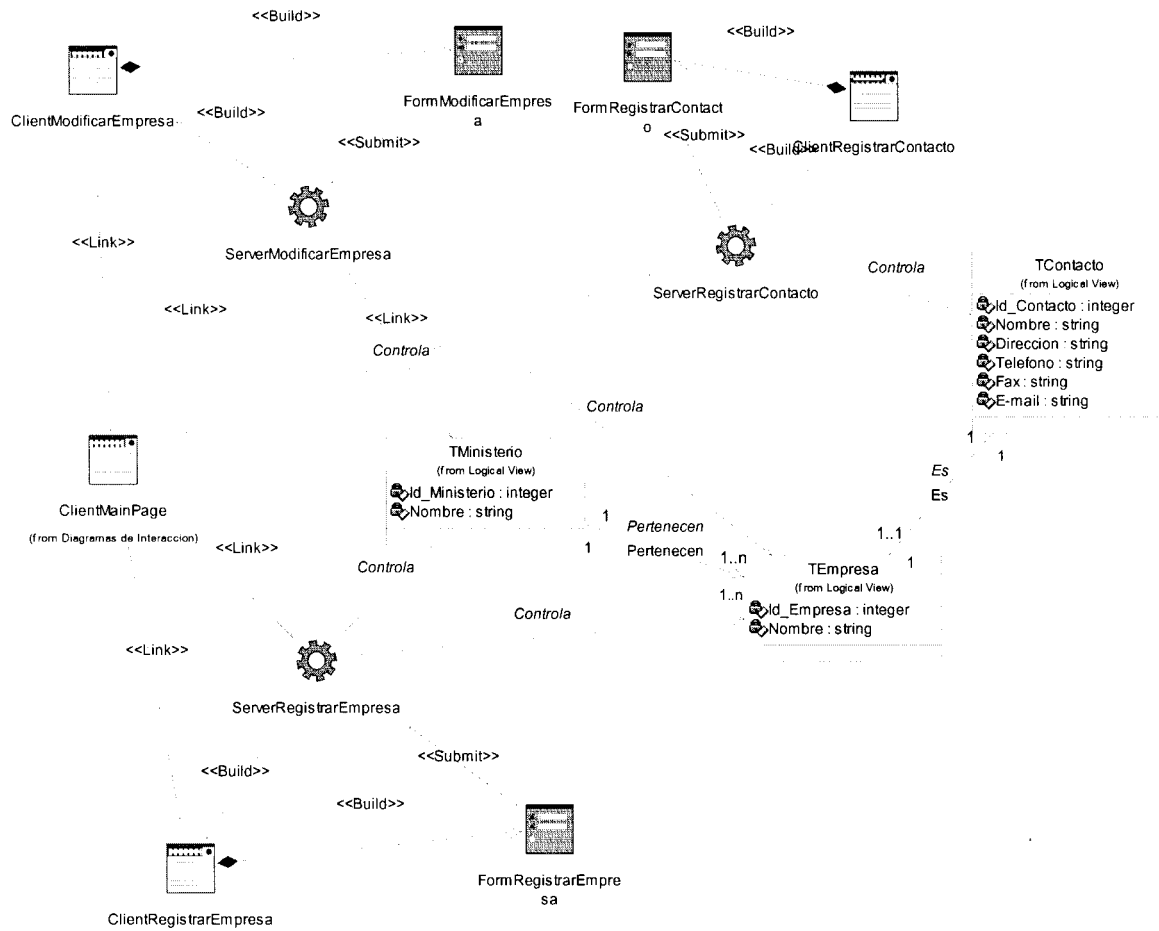
“Gestionar Autenticación de usuarios.”



“Gestionar Conceptos de Presupuesto”.

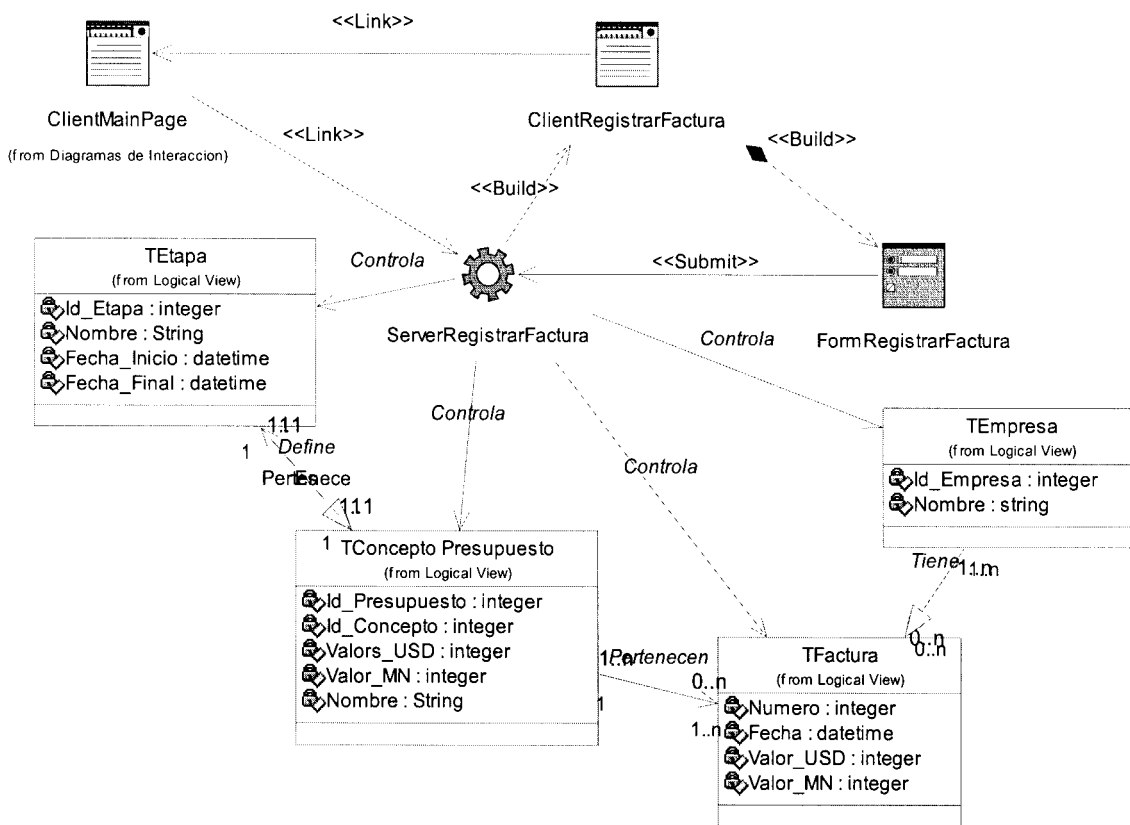


# "Gestionar Empresas".

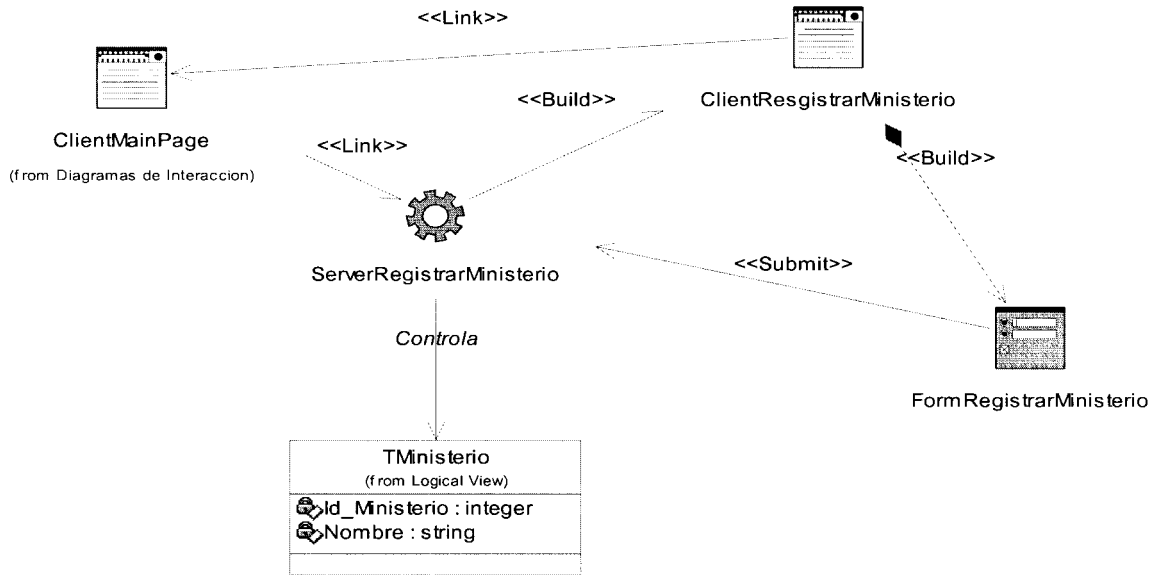




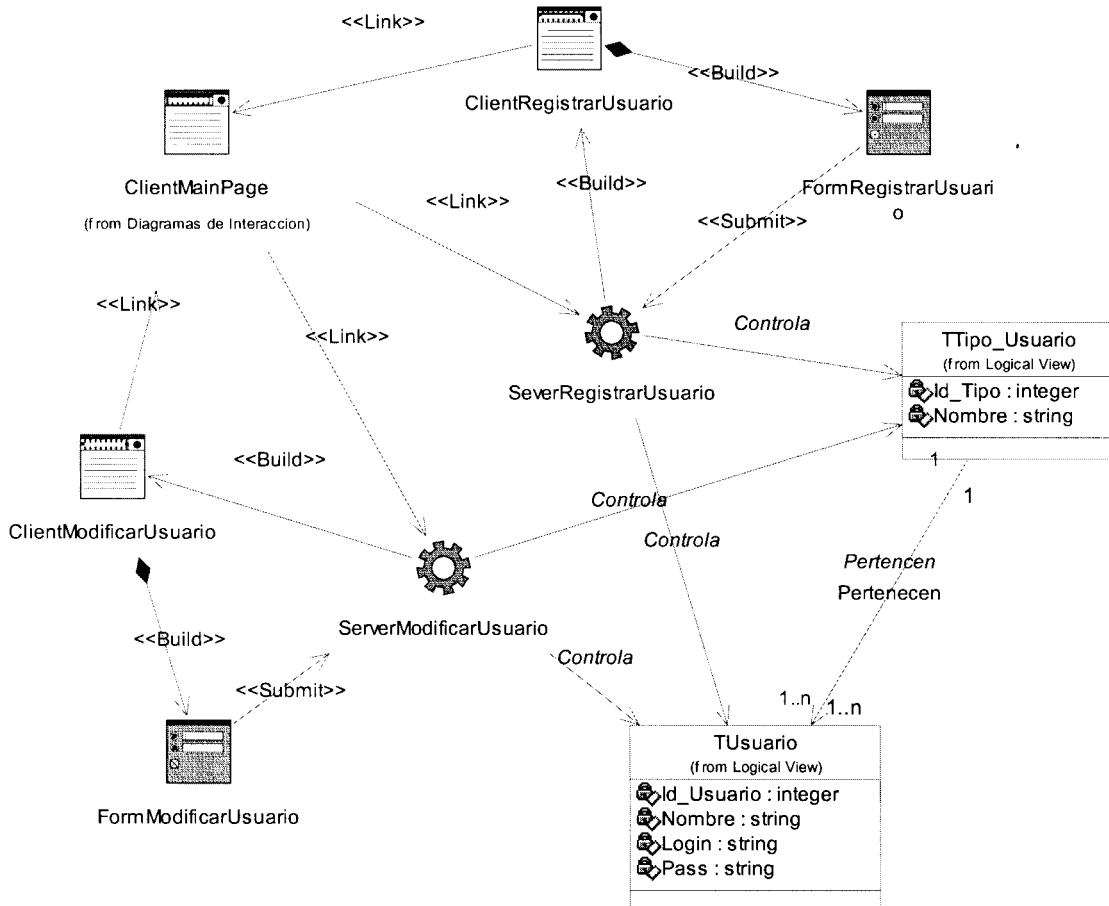
# "Gestionar Facturas".



“Gestionar Ministerios”.



“Gestionar Usuarios”.



## Anexo 7: Descripción de las clases.

### Clases Controladoras:

<b>Nombre:</b> ServerAutenticarUsuario	
<b>Tipo de clase :</b> Controladora	
Nombre:	AutenticarUsuario(Usuario, Contraseña)
Descripción:	Tiene como objetivo permitirles a los usuarios autenticarse en el sistema.
Nombre:	CrearUsuario(Id_Usuario, Nombre, Grupo)
Descripción:	Tiene como objetivo crear un usuario con los datos del usuario que se autenticó en el sistema.

<b>Nombre:</b> ServerCambiarPass	
<b>Tipo de clase :</b> Controladora	
Nombre:	CambiarPassword(Id_Usuario, NewPass)
Descripción:	Tiene como objetivo permitirle al usuario autenticado cambiar su contraseña para acceder al sistema.
Nombre:	MostrarUsuario(Nombre, Login)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado y el login para cambiar su contraseña.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre:</b> ServerBuscarCheque	
<b>Tipo de clase :</b> Controladora	
Nombre:	BuscarCheque(Numeracion)
Descripción:	Tiene como objetivo permitirle al usuario autenticado buscar los datos de un determinado cheque y comprobar si realmente este cheque fue emitido por la UCI.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.

Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: SeverRegistrarCheque</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	RegistrarCheque(Numeración, Moneda, Valor, Fecha)
Descripción:	Tiene como objetivo permitirle al usuario autenticado almacenar los datos de los cheques emitidos por la UCI.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: ServerAdicionarConcepto</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	AdicionarConcepto (Etapa, Concepto, Nombre , Valor)
Descripción:	Tiene como objetivo permitirle al usuario autenticado almacenar los datos de los conceptos con los que se trabajara en la etapa.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: ServerCrearEtapa</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	CrearEtapa(Nombre, Fecha_Inicio, Fecha_Fin)

Descripción:	Tiene como objetivo permitirle al usuario autenticado crear la nueva etapa en la que se trabajara así como almacenar los datos de esta.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

**Nombre: ServerRegistrarEmpresa**

**Tipo de clase :**Controladora

Nombre:	RegistrarEmpresa(Ministerio, Nombre)
Descripción:	Tiene como objetivo permitirle al usuario autenticado almacenar los datos de las empresas con las que se trabaja.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

**Nombre: ServerModificarEmpresa**

**Tipo de clase :**Controladora

Nombre:	ModificarEmpresa(Ministerio, Nombre)
Descripción:	Tiene como objetivo permitirle al usuario autenticado modificar los datos de las empresas con las que se trabaja.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	MostrarEmpresa(ListaEmpresas)
Descripción:	Tiene como objetivo mostrarle al usuario autenticado todas las empresas almacenadas en el sistema.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)

Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.
--------------	--

<b>Nombre: ServerRegistrarContacto</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	RegistrarContacto(Nombre, Direccion, Telefono, Fax, E-Mail)
Descripción:	Tiene como objetivo permitirle al usuario autenticado almacenar los datos de los contacto de cada empresa.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: ServerRegistrarFactura</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	RegistrarFacturas(Concepto, Empresa, Fecha, Valor)
Descripción:	Tiene como objetivo permitirle al usuario autenticado almacenar las facturas que se van generando en la UCI por los diferentes conceptos.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: ServerRegistrarMinisterio</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	RegistrarMinisterios(Nombre)
Descripción:	Tiene como objetivo permitirle al usuario autenticado almacenar los

	datos de los ministerios con lo que se trabaja.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: SeverRegistrarUsuario</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	RegistrarUsuario(Nombre, Grupo, Login, Pass, E-Mail)
Descripción:	Tiene como objetivo permitirle al administrador crear nuevas cuentas de usuarios en el sistema así como almacenar los datos de estos.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: ServerModificarUsuario</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	ModificarUsuario(Nombre, Grupo, Login, E-Mail)
Descripción:	Tiene como objetivo permitirle al administrador modificar los datos de los usuarios del sistema.
Nombre:	EliminarUsuario(Id_Usuario)
Descripción:	Tiene como objetivo permitirle al administrador eliminar cualquier cuenta de usuario del sistema.
Nombre:	MostrarListaUsuarios(ListaUsuarios)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea todas las cuentas de usuarios del sistema.
Nombre:	MostrarDatos(Nombre, Grupo, Login, E-Mail)
Descripción:	Tiene como objetivo mostrar en la página cliente los datos del usuario que se van a modificar.

Nombre:	MostrarGrupos(Listagrupos)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página cliente que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

<b>Nombre: ServerMainPage</b>	
<b>Tipo de clase :</b> Controladora	
Nombre:	MostrarUsuario(Nombre)
Descripción:	Tiene como objetivo mostrar en la página principal (cliente) que se crea el nombre del usuario que esta autenticado.
Nombre:	ValidarUsuario(Id_Usuario, Grupo)
Descripción:	Tiene como objetivo permitirle al usuario poder acceder la página verificando si este realmente esta autenticado y si tiene permisos para acceder a esta opción.

### Clases Interfaces:

<b>Nombre:</b> ClientAutenticar	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para la autenticación de los usuarios al sistema.

<b>Nombre:</b> ClientCambiarPass	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para cambiar la contraseña de acceso al sistema.



<b>Nombre:</b> ClientBuscarCheque	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para realizar la búsqueda de un cheque determinado.

<b>Nombre:</b> ClientRegistrarCheque	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para realizar la entrada de datos de los cheques emitidos.

<b>Nombre:</b> ClientAdicionarConcepto	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para realizar la entrada de datos de los conceptos con los que se trabajará.

<b>Nombre:</b> ClientCrearEtapa	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para crear la nueva etapa y realizar la entrada de datos de la misma.

<b>Nombre:</b> ClientRegistrarEmpresa	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para realizar la entrada de datos de las empresas con las que se trabajará.

<b>Nombre:</b> ClientModificarEmpresa	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para modificar los datos de las empresas con las que se trabaja en el sistema.

<b>Nombre:</b> ClientRegistrarContacto	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para realizar la entrada de datos de los contactos de cada una de las empresas con las que se trabaja.

<b>Nombre:</b> ClientRegistrarFactura	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para realizar la entrada de datos de las facturas que se originan por los diferentes conceptos.

<b>Nombre:</b> ClientResgistrarMinisterio	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para realizar la entrada de datos de los ministerios.

<b>Nombre:</b> ClientRegistrarUsuario	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para crear nuevas cuentas de usuarios así como realizar la entrada de datos de estos.

<b>Nombre:</b> ClientModificarUsuario	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene el formulario para modificar los datos de cualquier usuario registrado o eliminar cualquier cuenta de estos en el sistema.

<b>Nombre:</b> ClientMainPage	
<b>Tipo de clase :</b> Interfaces	
Nombre:	Crear_Pagina()
Descripción:	Página que contiene todas las opciones permitidas por el usuario.

### Clases Entidades:

<b>Nombre:</b> TCheque	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Nro_Cheque	String
Valor	Real
Fecha	DateTime
<b>Para cada responsabilidad:</b>	
Nombre:	GuardarDatos(Numeración, Valor, Fecha)
Descripción:	Tiene como objetivo almacenar los datos de los cheques en el sistema.
Nombre:	Existe_Cheque(Numeracion): Bool
Descripción:	Tiene como objetivo realizar una búsqueda en el sistema y determinar si un determinado cheque existe.
Nombre:	ObtenerDatos(Numeracion): DatosCheque
Descripción:	Tiene como objetivo devolver todos los datos de un cheque determinado.

<b>Nombre:</b> TConcepto_Presupuesto	
<b>Tipo de clase:</b> Entidad	

<b>Atributo</b>	<b>Tipo</b>
Id_Presupuesto	Integer
Concepto_Padre	Integer
Valor_USD	Real
Valor_MN	Real
Nombre	String
<b>Para cada responsabilidad:</b>	
Nombre:	DevolverConceptos(Etapa): Lista
Descripción:	Tiene como objetivo devolver un listado con los datos de todos los conceptos.
Nombre:	GuardarDatos(Datos)
Descripción:	Tiene como objetivo almacenar los datos de los nuevos conceptos de presupuesto con los que se trabajará.
Nombre:	ObtenerConcepto(Id_Concepto): Concepto
Descripción:	Tiene como objetivo devolver todos los datos de un concepto en específico.

<b>Nombre:</b> TContacto	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Id_Contacto	Integer
Nombre	String
Direccion	String
Telefono	String
Fax	String
E-Mail	String
<b>Para cada responsabilidad:</b>	
Nombre:	RegistrarContacto(Nombre, Direccion, Telefono, Fax, E-Mail)
Descripción:	Tiene como objetivo almacenar los datos de los contactos de las empresas.

<b>Nombre:</b> TEmpresa	
<b>Tipo de clase:</b> Entidad	

<b>Atributo</b>		<b>Tipo</b>
Id_Empresa		Integer
Nombre		String
<b>Para cada responsabilidad:</b>		
Nombre:	GuardarDatos(Nombre)	
Descripción:	Tiene como objetivo almacenar los datos de las empresas con las que se trabajará.	
Nombre:	ObtenerEmpresas(): ListaEmp	
Descripción:	Tiene como objetivo devolver un listado con los datos de todas las empresas.	
Nombre:	ObtenerEmpresa(Id_Empresa): Empresa	
Descripción:	Tiene como objetivo devolver todos los datos de una empresa determinada.	
Nombre:	ModificarEmpresa(Id_Empresa, Nombre)	
Descripción:	Tiene como objetivo modificar los datos de una empresa determinada.	

<b>Nombre:</b> TEtapa		
<b>Tipo de clase:</b> Entidad		
<b>Atributo</b>		<b>Tipo</b>
Id_Etapa		Integer
Nombre		String
Fecha_Inicio		DateTime
Fecha_Fin		DateTime
<b>Para cada responsabilidad:</b>		
Nombre:	GuardarDatos(Datos)	
Descripción:	Tiene como objetivo almacenar todos los datos de la etapa en el sistema.	
Nombre:	ObtenerEtapa(Fecha): Etapa	
Descripción:	Tiene como objetivo devolver todos los datos de una etapa en cuestión.	
Nombre:	ObtenerEtapas(): Etapas	
Descripción:	Tiene como objetivo devolver un listado con todos los datos de las todas las etapas.	
Nombre:	ObtenerEtapa(Id_Etapa): Etapa	
Descripción:	Tiene como objetivo devolver los datos de una determinada	

	etapa.
--	--------

<b>Nombre:</b> TFactura	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Numero	Integer
Fecha	DateTime
Valor_USD	Real
Valor_MN	Real
<b>Para cada responsabilidad:</b>	
Nombre:	GuardarDatos(Fecha, Valor_USD, Valor_MN)
Descripción:	Tiene como objetivo almacenar todos los datos de las facturas generadas por los diferentes conceptos.
Nombre:	ObtenerFacturas(): Lista
Descripción:	Tiene como objetivo devolver un listado con todos los datos de todas las Facturas.
Nombre:	BuscarFacturas(FechaI, FechaF): Facturas
Descripción:	Tiene como objetivo devolver un listado con todos los datos de las facturas que están entre esas fechas.

<b>Nombre:</b> TMinisterio	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Id_Ministerio	Integer
Nombre	String
<b>Para cada responsabilidad:</b>	
Nombre:	GuardarMinisterio(Nombre)
Descripción:	Tiene como objetivo almacenar todos los datos de los ministerios.
Nombre:	ObtenerMinisterios(): Lista
Descripción:	Tiene como objetivo devolver un listado con todos los datos de los ministerios.
Nombre:	ObtenerMinisterio(Id_Ministerio): Ministerio
Descripción:	Tiene como objetivo devolver todos los datos de un ministerio en específico.

<b>Nombre:</b> TMoneda	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Id_Moneda	Integer
Nombre	String
<b>Para cada responsabilidad:</b>	
Nombre:	GuardarMonedas(Nombre)
Descripción:	Tiene como objetivo almacenar todos los datos de las monedas con las que se trabajará.
Nombre:	ObtenerMonedas(): Monedas
Descripción:	Tiene como objetivo devolver un listado los datos de las monedas con las que se trabaja.

<b>Nombre:</b> TPago	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Id_Pago	Integer
Fecha	DateTime
Valor_USD	Real
Valor_MN	Real
<b>Para cada responsabilidad:</b>	
Nombre:	GuardarDatos(Datos)
Descripción:	Tiene como objetivo almacenar todos los datos de los pagos efectuados.
Nombre:	ObtenerPagos(): Lista
Descripción:	Tiene como objetivo devolver un listado con los datos de pagos efectuados.
Nombre:	ObtenerPago(Id_Pago)
Descripción:	Tiene como objetivo devolver los datos de un pago en específico.

**Nombre:** TTipo\_Usuario

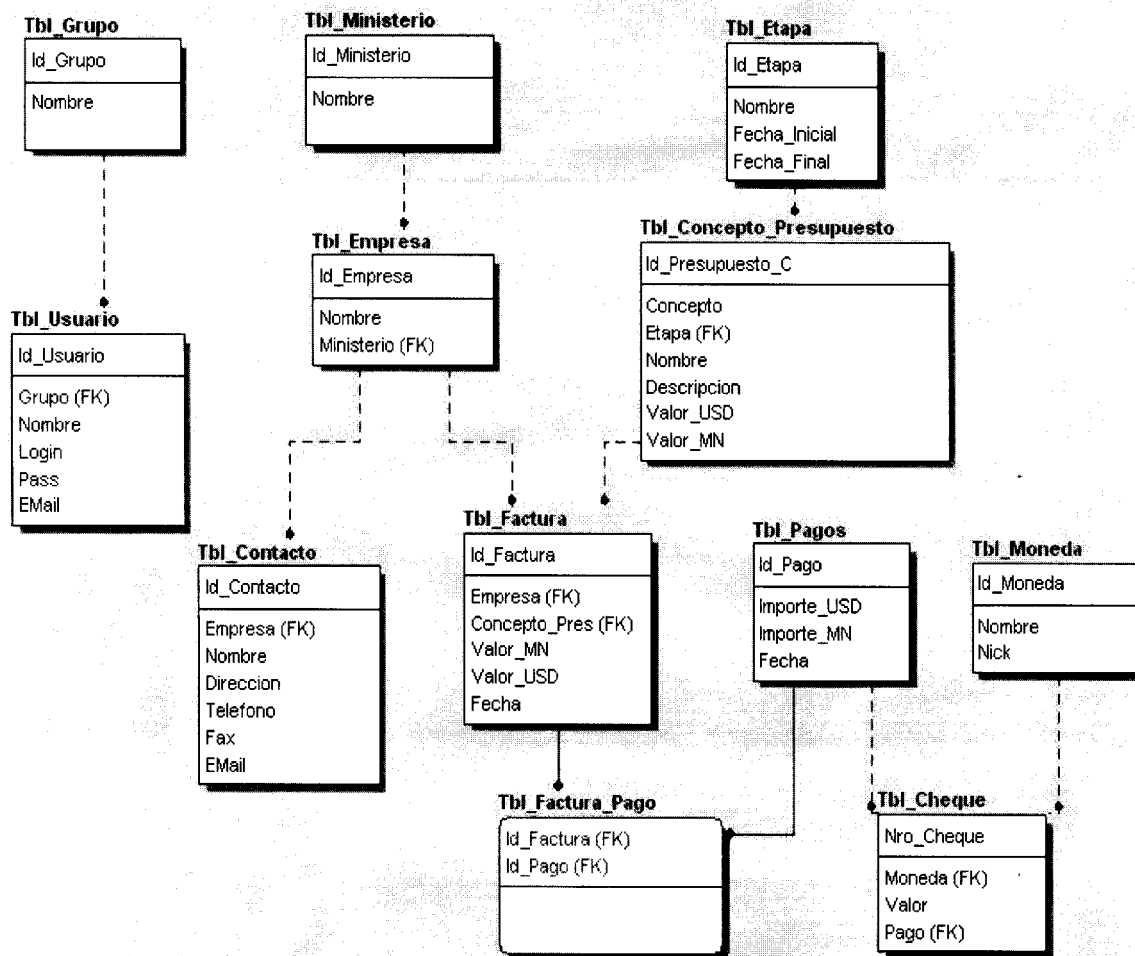
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Id_Tipo	Integer
Nombre	String
<b>Para cada responsabilidad:</b>	
Nombre:	ObtenerGrupos(): ListaGrupos
Descripción:	Tiene como objetivo devolver un listado con todos los tipos de usuarios que existen en el sistema.
Nombre:	ChequearGrupo(Id_Grupo): Bool
Descripción:	Tiene como objetivo chequear si existe un determinado grupo.

<b>Nombre:</b> TUsuario	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
Id_Usuario	Integer
Nombre	String
Login	String
Password	String
<b>Para cada responsabilidad:</b>	
Nombre:	DeleteUser(Id_Usuario)
Descripción:	Tiene como objetivo eliminar una cuenta de usuario del sistema.
Nombre:	ExisteUsuario(Login, Pass): Bool
Descripción:	Tiene como objetivo chequear si existe un usuario con estos datos.
Nombre:	ObtenerUsuarios(): ListUsers
Descripción:	Tiene como objetivo devolver todos los usuarios del sistema con sus datos.
Nombre:	ModificarRegistro(Id_Usuario, Nombre, Login)
Descripción:	Tiene como objetivo modificar los datos de un usuario en específico.
Nombre:	ObtenerUsuario(Id_Usuario): Usuario
Descripción:	Tiene como objetivo devolver todos los datos de un usuario en específico.
Nombre:	ExisteLogin(Login): bool
Descripción:	Tiene como objetivo chequear si ya existe un usuario con ese



	mismo login.
Nombre:	RegistrarUsuario(Nombre, Login, Password)
Descripción:	Tiene como objetivo almacenar los datos de un usuario en el sistema.
Nombre:	CambiarPass(Id_Usuario, Password, Newpass)
Descripción:	Tiene como objetivo cambiar la contraseña de un usuario en específico.

**Anexo 8 : Diagrama de base de datos relacional.**



**Anexo 9: Descripción de las tablas de la base de datos.**

Nombre: Tbl_Etapa.		
Atributo	Tipo	Descripción
Id_Etapa	Numeric	Identificador de la etapa.
Nombre	Varchar	Nombre de la etapa.
Fecha_Inicial	Datetime	Fecha de inicio de la etapa.
Fecha_Final	Datetime	Fecha de terminación de la etapa.

Nombre: Tbl_Concepto_Presupuesto.		
Atributo	Tipo	Descripción

Id_Presupuesto_C	Numeric	Identificador del concepto de presupuesto.
Concepto	Numeric	Concepto padre al que pertenece.
Etapa	Numeric	Etapa a la que pertenece.
Nombre	Varchar	Nombre del concepto.
Descripción	Varchar	Descripción del concepto.
Valor_USD	Money	Valor en divisas del concepto.
Valor_MN	Money	Valor en moneda nacional del concepto.

<b>Nombre: Tbl_Ministerio.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Ministerio	Numeric	Identificador del ministerio.
Nombre	varchar	Nombre de ministerio.

<b>Nombre: Tbl_Empresa.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Empresa	Numeric	Identificador de la empresa.
Ministerio	Numeric	Ministerio al que pertenece.
Nombre	Varchar	Nombre de la empresa.

<b>Nombre: Tbl_Contacto.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Conctacto	Numeric	Identificador del concepto de presupuesto.
Empresa	Numeric	Empresa a la que pertenece.
Nombre	Varchar	Nombre del contacto.
Direccion	Varchar	Dirección del contacto.
Telefono	Varchar	Teléfono del contacto.
Fax	Varchar	Fax del contacto.
EMail	Varchar	Dirección de correo electrónico del contacto.

<b>Nombre: Tbl_Factura.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

Id_Factura	Numeric	Identificador de la factura.
Empresa	Numeric	Empresa a la que pertenece.
Concepto_pres	Numeric	Concepto al que pertenece.
Valor_USD	Money	Valor en divisas.
Valo_MN	Money	Valor en moneda nacional.
Fecha	Datetime	Fecha en la que se origina.

<b>Nombre: Tbl_Pago.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Pago	Numeric	Identificador del pago.
Importe_usd	Money	Valor en divisas.
Importe_mn	Money	Valor en moneda nacional.
Fecha	Datetime	Fecha en la que se efectúa el pago.

<b>Nombre: Tbl_Moneda.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Moneda	Numeric	Identificador de la moneda.
Nombre	Varchar	Nombre de la moneda.
Nick	Varchar	Siglas de la moneda.

<b>Nombre: Tbl_Cheque.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Cheque	Numeric	Identificador del cheque.
Moneda	Numeric	Moneda a la que pertenece el cheque.
Pago	Numeric	Pago al que pertenece el cheque.
Valor	Money	Valor del cheque.

<b>Nombre: Tbl_Grupo.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Grupo	Numeric	Identificador del grupo.
Nombre	Varchar	Nombre del grupo.

<b>Nombre: Tbl_Usuario.</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Usuario	Numeric	Identificador del usuario.
Grupo	Numeric	Grupo al que pertenece el usuario.
Nombre	Varchar	Nombre del usuario.
Login	Varchar	Nombre de inicio de sesión del usuario.
Pass	Varchar	Contraseña del usuario.
Email	Varchar	Dirección de correo electrónico del usuario.