

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



**Título: Concepción y Desarrollo del Módulo de
Administración del Proyecto TeleBanca**



**TRABAJO DE DIPLOMA PARA OPTAR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS**

**AUTORES: JORGE ALBERTO HERNÁNDEZ CALLAO
EZEQUIEL MARTÍNEZ MENDOZA
YORDAN GAMBOA LAINA**

TUTOR: ING. YULIERKYS TORRES IGLESIAS

**AÑO DE LA REVOLUCIÓN ENERGÉTICA EN CUBA
CIUDAD DE LA HABANA
JULIO DE 2007**

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días de mes de _____ del _____

Nombre completo del primer autor

Nombre completo del segundo autor

Nombre completo del tercer autor

Nombre completo del tutor

Datos de contacto

Ing. Yuliesky Torres Iglesias

Graduado de Ingeniero Informático en el 2005 y profesor instructor con dos años de experiencia docente en la Universidad de las Ciencias Informáticas (UCI).

Actual jefe del Proyecto TeleBanca.

Opinión del usuario del trabajo de diploma

El Trabajo de Diploma titulado Concepción y Desarrollo de Módulo de Administración del Proyecto TeleBanca fue realizado en la Universidad de Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el presente trabajo le satisface

__ Totalmente

__ Parcialmente en un __%

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Representante de la entidad

Cargo

Firma

Cuño

Opinión del tutor del trabajo de diploma

Título: Concepción y Desarrollo del Módulo de Administración del Proyecto TeleBanca.

Autores: Jorge Alberto Hernández Callao, Ezequiel Martínez Mendoza, Yordan Gamboa Laina.

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan:

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingeniero en Ciencias Informáticas; y propongo que se les otorgue al Trabajo de Diploma la calificación de ____ puntos.

Firma

Fecha

Agradecimientos

La culminación de este trabajo, a pesar de todos nuestros esfuerzos, realmente no hubiesen sido posible sin la ayuda de un grupo de personas que nos apoyaron en la realización del mismo. Por lo tanto quisiéramos aprovechar esta sección para demostrarles a ellos nuestro agradecimiento infinito por toda la ayuda prestada.

Jorge A. Hernandez Callao.

En primer lugar agradecerles a mis padres por haberme siempre guiado por buen camino. A mi novia Virginia que siempre me apoyó, agradezco de la misma manera a todos mis amigos y muy en particular a Lizandra y Yannier que transitaron estos últimos años junto a mí, apoyándome en todo lo que estuviera a su alcance, de manera incondicional.

Ezequiel Martinez Mendoza

Agradecerles a mis padres, hermanos y a mi novia Siselis por estar siempre conmigo apoyándome y dándome las fuerzas necesarias para llegar a este soñado triunfo, a todas las personas que de una forma u otra con su amistad y apoyo han colaborado, también a aquellos que aunque no están presentes les hubiera gustado disfrutar como yo de este inolvidable momento de mi vida.

Yordan Gamboa Laina

Agradezco a mis padres por la confianza aún cuando siempre estoy lejos; a mis abuelos, hermanos y amigos que me inspiran a seguir adelante.

Agradecer a nuestro tutor Yulierki por su esfuerzo y apoyo en todo este camino, a nuestra instructora María por su cariño y amistad. Las gracias también a nuestro Comandante en Jefe Fidel Castro Ruz por darnos la oportunidad de estudiar para alcanzar los horizontes científicos y culturales que hemos alcanzado. Además a todos nuestros profesores que cumplieron con su objetivo de formar un profesional para el bien de la sociedad.

Dedicatoria

A mis padres Jorge Alberto y Maria Grisell que siempre confiaron en mi, a mis hermanos y abuelos, a todas las personas que de una forma u otra con su amistad y apoyo han colaborado para que se vieran realizados mis sueños y a aquellos que aunque no estén presente les hubiera gustado disfrutar como yo de este inolvidable momento de mi vida.

Jorge A. Hernandez Callao

A mi familia y en especial a mi madre Miraida Mendoza Púpo, a mi padre Ezequiel Martinez Valera, a mis queridos hermanos, Yanetsi Martínez Mendoza, Yetsidel Martínez Mendoza, Yeison Martínez Mendoza y a todas las personas que de una forma u otra con su amistad y apoyo han colaborado para que se viera realizado mi sueño de hacerme profesional.

Ezequiel Martinez Mendoza

A mis familiares y amigos.....

Yordan Gamboa Laina

Resumen

En la actualidad los Sistemas Informáticos juegan un papel importante en las tareas que realiza el hombre, facilitando el trabajo y mejorando su calidad de vida. En Cuba se está priorizando la informatización de los sectores que prestan servicios a la población, con el propósito de mejorar el pago de algunos servicios indispensables que se le brinda a la sociedad como son: el agua, electricidad, teléfono, y multas, entre otros. Dado que existe un incremento de cuentas de tarjeta red en los diferentes bancos de Cuba, surge la intención por parte de las entidades involucradas, de informatizar estos servicios a través de un centro de llamadas, el cual sería responsable de administrar los niveles de accesos de cada uno de los responsables de informar y ejecutar tareas, así como garantizar una fácil configuración, seguridad y administración del sistema responsable de efectuar el pago de los servicios.

Para facilitar la administración a los usuarios finales del sistema, se ha concebido e implementado el módulo de administración del proyecto TeleBanca; en este documento se describen las etapas de análisis, diseño, implementación y prueba del módulo antes mencionado, haciendo uso del Lenguaje Unificado de Modelación (UML: *Unified Modeling Language*) y su herramienta por excelencia el *Rational Rose*, que permitieron la concepción del módulo. Para la implementación se utiliza como lenguaje de programación el C# y como plataforma de desarrollo Microsoft. Net.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 LA ADMINISTRACIÓN DE SISTEMAS DE INFORMACIÓN	5
1.1.1 <i>¿Administrador de un Sistema Informático?</i>	8
1.2 PROCESO DE AUTENTICACIÓN EN SISTEMAS INFORMÁTICOS	9
1.2.1 <i>Proceso de autorización en Sistemas Informáticos</i>	9
1.3 USOS Y TÉCNICAS DE LA CRIPTOGRAFÍA	10
1.4 ALTERNATIVAS ACTUALES PARA LA REALIZACIÓN DE CONTROL DE ACCESOS A GRAN ESCALA	11
1.4.1 <i>Control de Acceso Basado en Roles (RBAC)</i>	12
1.5 ESTUDIOS DE MÓDULOS DE ADMINISTRACIÓN DE SISTEMAS EN LA UCI	14
1.5.1 <i>Módulo de Administración del sistema de la Aduana</i>	14
1.5.2 <i>Módulo de Administración del ERP en el sistema del MINFAR</i>	15
1.6 HERRAMIENTAS PARA EL DESARROLLO DEL SISTEMA	15
1.6.1 <i>Aplicaciones Web y Sitios Web</i>	15
1.6.2 <i>Estudios de Plataformas de Desarrollo</i>	17
1.6.2.1 <i>Tecnología ASP.NET</i>	19
1.6.2.2 <i>¿Por qué C# y no otro lenguaje de programación?</i>	20
1.6.2.3 <i>Visual Studio .NET como herramienta de desarrollo</i>	21
1.6.2.4 <i>Servicios Web (Web Services)</i>	22
1.6.3 <i>¿Por qué RoboHelp Office X5 como herramienta de desarrollo de la ayuda?</i>	23
1.6.4 <i>Metodologías de desarrollo</i>	24
1.6.5 <i>Lenguaje Unificado de Modelado (UML)</i>	26
1.6.6 <i>Herramienta Rational Rose Enterprise Edition</i>	26
1.7 CONCLUSIONES	27
CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA	28
2.1 OBJETO DE ESTUDIO	28
2.1.1 <i>Problema y situación problemática</i>	28
2.1.2 <i>Objeto de automatización</i>	29
2.1.3 <i>Propuesta de sistema</i>	29
2.1.4 <i>Especificación de los requisitos de software</i>	31
2.1.4.1 <i>Requisitos Funcionales</i>	31
2.1.4.2 <i>Requisitos No Funcionales</i>	33
2.1.5 <i>Definición de los casos de uso</i>	36
2.2 CONCLUSIONES	59
CAPITULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	60
3.1 ANÁLISIS.....	60
3.1.1 <i>Diagrama de Clases del Análisis</i>	60
3.2 DISEÑO.....	60
3.2.1 <i>Diagramas de interacción del sistema</i>	61
3.2.2 <i>Diagrama de clases del Diseño</i>	67
3.2.3 <i>Descripción de las clases</i>	70
3.2.4 <i>Concepción general de la ayuda del sistema del proyecto TeleBanca</i>	85
3.3 CONCLUSIONES	85
CAPITULO 4: IMPLEMENTACIÓN Y PRUEBA	86

4.1	IMPLEMENTACIÓN.....	86
4.1.1	<i>Diagrama de despliegue.....</i>	<i>86</i>
4.1.2	<i>Diagrama de componentes.....</i>	<i>87</i>
4.2	PRUEBA.....	89
4.3	CONCLUSIONES.....	103
	CONCLUSIONES.....	104
	BIBLIOGRAFÍA.....	106

Introducción

El adelanto de las tecnologías de la información y las comunicaciones (TIC) en Cuba y la indudable importancia que va tomando a nivel mundial la informática, junto con el desarrollo y la producción de software, hacen que Cuba trace sus líneas para insertarse en el mercado de software a nivel mundial, además del evidente provecho que trae el uso interno de sistemas informáticos para el desarrollo de nuestro país en general.

En la actualidad se ha promovido y priorizado con suma importancia la informatización de los sectores que prestan servicios a la población, con el fin de mejorar los mismo y la calidad de vida de la sociedad. El pago de los servicios elementales, como son: agua, electricidad, teléfono y multas, forman parte de los que a la población se le brinda. Dado que existe un notable aumento de cuentas de tarjeta red en los diferentes bancos de Cuba, surge la intención por parte de las entidades involucradas, de informatizar estos servicios a través de un centro de llamadas, que traería grandes ventajas, tanto a los clientes, como a las empresas, garantizándoles a los mismos un pago rápido y efectivo. Además haciendo uso de los sistemas telefónicos se lograría una relación más cercana a las inquietudes de los clientes, dándoles las respuestas que estos necesitan.

Para garantizar el funcionamiento del centro de llamadas, el mismo debe contar con un Sistema Informático que posea la capacidad de autenticar a los clientes cuando realicen la llamada para efectuar el pago, facilitar un pago rápido, seguro y eficiente, gestionar la información que los clientes solicitarán, además la administración que es el centro de atención del presente trabajo, que se encargaría de gestionar los niveles de accesos de cada uno de los responsables de informar y ejecutar tareas, así como garantizar una fácil configuración, seguridad y administración del sistema responsable de efectuar el pago de los servicios.

Por la necesidad de realizar un Sistema Informático, responsable de efectuar el pago de los servicio con un alto nivel de seguridad, un equipo del proyecto encargado de realizar el sistema (TeleBanca), de la facultad 4, de la Universidad

de las Ciencias Informáticas (UCI), ha seleccionado como línea de investigación, la Administración de Sistemas Informáticos.

Este trabajo surge como necesidad de resolver el siguiente **problema científico**:
¿Cómo facilitar la administración del sistema encargado de realizar el pago vía telefónica mediante un módulo de administración?

Con vista a la solución del problema antes planteado se define como **objeto de estudio** la administración de sistemas informáticos.

De lo antes expuesto se deriva el siguiente **objetivo general** del presente trabajo:

Obtener la implementación del módulo de administración del sistema TeleBanca.

Para lograr el objetivo general planteado se definen los siguientes **objetivos específicos**:

- Realizar una investigación sobre la administración de sistemas informáticos.
- Definir los procesos a implementar como parte de la administración del sistema TeleBanca.
- Realizar el Análisis y Diseño del módulo de administración del sistema.
- Realizar la Implementación y Prueba del módulo de administración del sistema TeleBanca.

Definiendo así el **campo de acción** como la administración del sistema encargado de realizar el pago vía Telefónica.

La investigación se sustenta en la siguiente **hipótesis** si se desarrolla un Módulo automatizado para la administración del sistema encargado de efectuar el pago vía telefónica, este permitirá una mayor seguridad y control de los niveles de acceso de cada uno de los responsables de efectuar tareas, mediante la gestión de usuarios y roles, además de facilitar la configuración y consulta de notificaciones del sistema.

Para llevar a cabo la investigación se proponen las siguientes **tareas de la investigación**:

- Realizar una investigación sobre la administración de sistemas informáticos.
- Realizar un estudio del arte sobre los principales procesos en la administración de sistemas.
- Realizar un estudio de las metodologías de desarrollo de software.
- Definir la metodología de desarrollo de software.
- Definir los requisitos que debe tener el módulo de administración del sistema TeleBanca.
- Realizar el análisis del módulo de administración.
- Realizar el diseño del módulo de administración
- Realizar la implementación del módulo de administración
- Elaborar los casos de prueba del módulo de administración.
- Aplicar los casos de prueba del módulo de administración.

Entre los **aportes prácticos esperados** con el desarrollo de este módulo se espera que los involucrados en este proceso cuenten con una herramienta de trabajo que sea capaz de configurar el sistema de una forma rápida y gestionar eficientemente los niveles de acceso para así lograr una mayor seguridad del mismo.

El presente trabajo está dividido en cuatro capítulos y varios anexos, que recogen todo lo abordado en la investigación y desarrollo del mismo, como el Análisis, Diseño, Implementación y Prueba del módulo de administración del Sistema Informático que propone el proyecto TeleBanca.

Capítulo 1: Fundamentación Teórica, abarca conceptos relacionados con la administración y seguridad de Sistemas Informáticos, así como los controles de acceso que pueden ser utilizados en los mismos. También incluye un estado del arte de los lenguajes a utilizar en la implementación del sistema, como son: ASP.NET y C#. La herramienta utilizada para diseñar e implementar: Visual

Studio 2005. Además del estudio de las metodologías y el lenguaje de modelación.

Capítulo 2: Características del Sistema, describe el objeto de estudio, el problema y situación problemática, objeto de automatización, la propuesta del sistema. Además la especificación de los requisitos funcionales y no funcionales, definiéndose los casos de uso del sistema.

Capítulo 3: Análisis y Diseño del Sistema, aborda los detalles relacionados con el análisis y diseño del módulo de administración, representándose un diagrama de clases del análisis, los diagramas de colaboración y los diagramas de clases del diseño, con sus respectivas descripciones.

Capítulo 4: Implementación y Prueba, se representa gráficamente un diagrama de despliegue y un diagrama de componentes. Además se describe el modelo de prueba correspondiente a los casos de uso del módulo de administración del sistema propuesto.

Capítulo 1: Fundamentación Teórica

Introducción

El desarrollo actual de la Informática y las Telecomunicaciones ha provocado una avalancha de ideas a través de todo el mundo, centradas principalmente en aprovechar las nuevas características tecnológicas, reducir los costos y brindar nuevos y mejores servicios.

Partiendo de las exigencias de seguridad y control de los nuevos sistemas para la gestión de la información que se maneja, se hace necesaria la creación de módulos capaces de permitir la administración y los controles de acceso, entre otras tareas de carácter administrativo.

1.1 La Administración de Sistemas de Información

¿Qué es la administración?

Antes que nada se definirá que es administración, sistema e información como:

"Administración es el proceso de planear, organizar, dirigir y controlar el uso de recursos para lograr objetivos". [CHIAVENATO, Colombia 1999]

Otra definición es la de Koontz "La administración es el proceso de diseñar y mantener un ambiente en el cual las personas, trabajando juntas en grupos, alcanzan con eficiencia metas seleccionadas". [KOONTZ y WEINRICHY, México 1998]

Entonces se puede concluir definiendo la administración como el proceso de organizar, planear, dirigir y controlar; actividades y recursos con el fin de lograr un objetivo.

¿Qué es un sistema?

Se empezará definiendo qué es un sistema; analizando varias definiciones y después se elaborará una tratándose de tomar de cada una de ellas lo más esencial de la idea que estas dan:

"El sistema es un conjunto de elementos organizados que se encuentran en interacción, que buscan alguna meta o metas comunes, operando para ello sobre datos o información sobre energía o materia u organismos en una referencia temporal para producir como salida información o energía u organismos". [MURDICK, México 1998]

"Un sistema es una reunión o conjunto de elementos relacionados". [GICH, México 1981]

"Un sistema es un agregado de personas, cosas, información etc., agrupados en conjunto de acuerdo con un objetivo". [DAVID]

Entonces se puede definir que un sistema es el conjunto de cosas, objetos, conceptos y elementos conectados unos con otros que interactúan para lograr un objetivo en común.

¿Qué es información?

La información son todos aquellos datos transformados o modificados que tienen valor para aquellos usuarios que hacen uso de ellos.

"Los datos están constituidos por los registros de los hechos, acontecimientos, transacciones, etc. Por el contrario, la información implica que los datos estén procesados de tal manera que resulten útiles o significativos para el receptor de los mismos, por lo que en cierto modo, los datos se pueden considerar la materia prima para obtener información"[CARRERA]. Llegamos a la conclusión que la información son datos procesados con un valor para aquel usuario que la necesita, aunque para otro sólo sean datos que después de procesarlos obtenga otra información.

¿Qué es un Sistema de Información?

Un Sistema de Información es: "Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan elaboran y distribuyen la información (o parte de ella)

necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) desempeñar su actividad de acuerdo a su estrategia de negocio". [IBIDEM]

Otra definición de Sistemas de Información es " procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas". [SENN, México 1995]

Por lo tanto se puede definir un Sistema de Información como un conjunto de subsistemas que incluyen hardware, software, medios de almacenamiento de datos ya sea primarios, secundarios y bases de datos relacionadas entre sí, con el fin de procesar entradas para realizar transformaciones a esas entradas y convertirlas en salidas de información importantes para una empresa u organización.

El objetivo de un Sistema de Información es ayudar al desempeño de las actividades que desarrolla la empresa, suministrando la información adecuada, con la calidad requerida, a la persona o departamento que lo solicita, en el momento y lugar especificados con el formato más útil para el receptor.

El Sistema de Información está al servicio de los objetivos de la empresa para lograr dichos objetivos, la empresa y sus individuos adoptan procedimientos y prácticas de trabajo que resultan más útiles y eficaces.

"Todo Sistema de Información basado en computadora cuenta con estos componentes:" [RODRÍGUEZ, 2002]



Figura 1.1. Sistema de Información basado en computadora

1.1.1 ¿Administrador de un Sistema Informático?

El significado preciso varía. En las organizaciones con un sistema muy grande y complicado, generalmente dividen al personal informático según su especialidad. En este caso un administrador de un Sistema Informático es la persona que tiene la responsabilidad de asegurar el correcto funcionamiento y mantenimiento del mismo, o algún aspecto de éste.

Responsabilidades que estos generalmente incluyen:

- Realizar copias de seguridad.
- Instalar y configurar el nuevo hardware y software.
- Agregar, borrar y modificar información de las cuentas de usuarios, restablecer contraseñas, etc.
- Responder consultas técnicas.
- Responsable de la seguridad.
- Responsable de la configuración del sistema.
- Resolución de problemas.

Se puede afirmar que la información como ya se ha mencionado es fundamental para las operaciones y actividades en las empresas, la información es necesario manejarla de una forma adecuada y a la vez rápida, así que es necesario de los Sistemas de Información y a su vez que estos estén bien administrados para así tener un mayor control y seguridad de los mismos, no basta con tener la información, hay que saber manejarla eficientemente.

Dentro de unos pocos años aquellas empresas u organismos que no cambien sus viejos sistemas por sistemas de información computacionales y hagan uso adecuado de la información, teniendo una apertura hacia nuevas ideas y uso de nuevas tecnologías, simplemente serán aplastadas por la tecnología y por su misma ignorancia al cambio.

1.2 Proceso de autenticación en Sistemas Informáticos

Es el proceso de verificar que alguien o algo es quien o lo que dice ser. En redes de equipos públicos y privados la autenticación se lleva a cabo comúnmente a través de contraseñas de inicio de sesión.

Todos los clientes son autenticados cuando se conectan a un servidor, después de eso el cliente es de confianza y a su vez cuando un cliente ejecuta alguna acción estas son chequeadas contra una política específica, donde cada uno de los controles y acciones son permitidos o denegados para ese usuario. Esta especificidad asume que el usuario está autenticado tanto tiempo como dure la conexión.

1.2.1 Proceso de autorización en Sistemas Informáticos

La autorización se realiza una vez que se ha autenticado al usuario. Este proceso consiste en determinar lo que el usuario puede y no puede hacer en una aplicación. Por ejemplo, las pantallas que pueden consultar, los campos que puede ver, si sólo puede leer estas pantallas o también puede escribir en ellas, etc.

Aunque la autorización se suele considerar como una función de seguridad, en realidad las decisiones que determinan qué usuarios tienen acceso a qué funcionalidades son decisiones de la Empresa. Por consiguiente, las reglas que implementan la autorización son reglas empresariales (políticas).

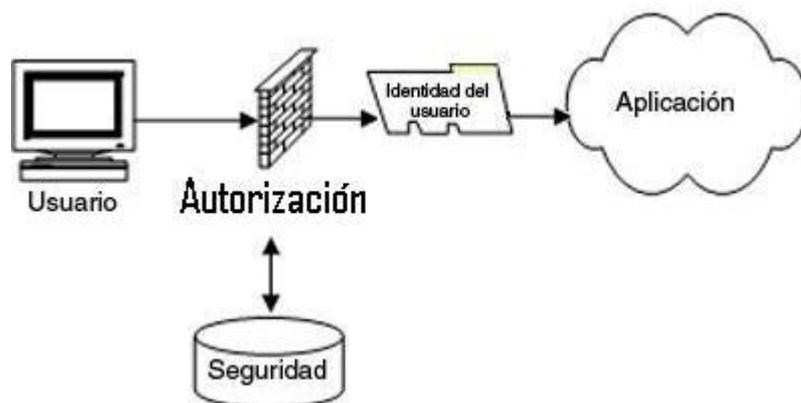


Figura 1.2. Proceso de autorización de usuarios.

1.3 Usos y técnicas de la criptografía

La criptografía es una tecnología importante para crear aplicaciones Web seguras. Esta ciencia existe desde hace muchos años, aunque mucho antes de que se hubieran inventado los equipos informáticos. Sin embargo con la llegada de estos últimos, la criptografía ha sido capaz de crear unos códigos prácticamente indescifrables.

“La criptografía (del griego Kryptos, "ocultar", y grafos, "escribir", literalmente "escritura oculta") es el arte o ciencia de cifrar y descifrar información utilizando técnicas matemáticas que hagan posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos”. [LUCENA LÓPEZ, Marzo 2004]

Con más precisión, cuando se habla de esta área de conocimiento como ciencia se debería hablar de criptología , que engloba tanto las técnicas de cifrado, la criptografía propiamente dicha, como sus técnicas complementarias: el criptoanálisis, que estudia los métodos que se utilizan para romper textos cifrados con objeto de recuperar la información original en ausencia de la clave.

Los algoritmos criptográficos se basan en un método matemático que se emplea para cifrar y descifrar un mensaje. Generalmente funciona empleando una o más claves (números o cadenas de caracteres) como parámetros del algoritmo, de modo que sean necesarias para recuperar el mensaje a partir de la versión cifrada, algunos de los algoritmos más representativos en la actualidad son:

a) Algoritmos simétricos de cifrado

Cifrados de Producto

El algoritmo DES (Estándar de cifrado de datos). Variantes

El algoritmo IDEA (algoritmo internacional del cifrado de datos)

Modos de Operación para Algoritmos Simétricos

Criptoanálisis de Algoritmos Simétricos

b) Algoritmos asimétricos de cifrado

El Algoritmo RSA (cifra asimétrica definida en la estándar de cifrado de datos)

Otros Algoritmos Asimétricos

Protocolo SSL (Protocolo de Seguridad en transacciones electrónicas).

Criptografía simplificada con Visual Estudio .NET

En la actualidad herramientas como el Visual Estudio .NET son un ejemplo concreto del buen uso de la criptografía, ya que esta contiene una serie de clases que envuelven sofisticados algoritmos y métodos de un sencillo manejo. Dichas clases se encuentran en el espacio de nombre System.Security.Cryptography.

Tabla 1.1. Algoritmos implementados por .NET Framework para la criptografía.

Algoritmos simétricos	Algoritmos asimétricos	Algoritmos hash
DES (Estándar de cifrado de datos)	DSA (Algoritmo de firma digital)	HMAC SHA1 (Código de autenticación de mensajes basado en hash que utiliza el algoritmo hash SHA1)
TripleDES (Estándar de cifrado de datos triple)	RSA	MAC Triple DES (Código de autenticación de mensajes mediante Triple DES)
Rijndael		MD5 (Algoritmo de Resumen del Mensaje 5).
RC2		SHA1, SHA256, SHA384, SHA512 (Algoritmo hash seguro mediante diversos tamaños de valor hash).

1.4 Alternativas actuales para la realización de Control de Accesos a gran escala

La tecnología es hoy por hoy algo que cambia continuamente, y ésta ayuda a los individuos a ser más productivos, y así pensar en organismos sociales que se unan para llegar a una meta en común y así lograr objetivos previamente establecidos, para lograr unificarse y llegar a formar empresas, que necesitan ser parte de esta evolución tecnológica.

Sin embargo, cada vez es más indispensable el uso de Control de Accesos para proteger cualquier tipo de información, ya sea personal o empresarial, ya que mucha de ésta es confidencial y debe de ser protegida, para así evitar que la información pueda ser robada o sufrir daños.

Por lo anterior, es importante que los administradores de sistemas tengan Listas de Control de Accesos, para así proteger la información que es requerida. Es importante tener en cuenta que cada usuario debe de tener una contraseña para acceder a dichos sistemas, la cual, le permitirá tener permisos para manejar información. Si un usuario es cambiado de puesto es seguro que a éste le sean asignados nuevos permisos, y por lo tanto nueva contraseña, para así lograr más seguridad.

Hoy en día se estudian diferentes tipos de Control de Acceso, como son:

MAC (Mandatory Access Control, Control de Acceso Obligatorio por sus siglas en ingles), DAC (Discretionary Access Control, Control de Acceso Discrecional por sus siglas en ingles) y RBAC (Role-based Access Control).

1.4.1 Control de Acceso Basado en Roles (RBAC)

El control de acceso basado en roles es una tecnología que llegó para satisfacer las principales necesidades en cuanto a Control de Accesos se refiere.

Al inicio, los sistemas manejaron dos tipos diferentes de Control de Accesos, uno llamado DAC en éste, el usuario es quien decide como proteger el sistema, mediante Controles de Acceso impuestos por el sistema, y el otro MAC, en este tipo de Control de Acceso, el sistema es quién protege los recursos.

Sin embargo, actualmente este tipo de problema ha disminuido, gracias a que la seguridad está siendo llevada a cabo por RBAC. La administración de la seguridad, consiste en que los roles deben asignarse adecuadamente a los diferentes tipos de personas, según sus capacidades y puestos de cada una de ellas. Básicamente, la gran evolución que se está teniendo con RBAC es porque funciona como una mezcla de DAC y MAC, puesto que contiene de cierta forma la flexibilidad para el Control de Accesos que tiene DAC y la rigidez que se tiene con MAC.

Una de las características principales que muestran el manejo de los sistemas de Control de Accesos es que el administrador del sistema es quien maneja los datos, para así satisfacer las necesidades de la empresa u organización, tal como lo hace RBAC

En RBAC, los permisos se encuentran asociados con los roles y los usuarios son miembros de los roles, de ésta manera se adquieren permisos a los roles, es por eso que la administración en el Control de Acceso es más sencilla. Los roles son creados en forma centralizada para las distintas funciones de trabajos en una organización, cada usuario tiene asignado uno o varios roles. Es por eso que cada usuario puede ser asignado de un rol a otro. Cada usuario tiene diferentes permisos, dependiendo del rol que esté llevando a cabo.

En la actualidad, muchos desarrolladores y usuarios de software han hablado de la seguridad que ofrece el Control de Acceso, muy especialmente de lo especificado en el RBAC, así mismo, cuenta con aplicaciones reales, que en la práctica han demostrado brindar solución a problemas existentes.

RBAC es reconocido como uno de los modelos que más han revolucionado en los últimos años, ya que su fundamental aporte es que puede llegar a funcionar con grandes sistemas, y al mismo tiempo brindarle seguridad.

Usuarios, roles y permisos en RBAC

Para que un sistema sea confiable se debe de tener en cuenta cuáles podrían ser los caminos para llegar a la confiabilidad, los niveles que existen en los usuarios y los permisos o privilegios que dichos usuarios puedan llegar a tener.

La seguridad es muy difícil de mantener, ya que muchas veces administrar sistemas se vuelve complicado y por lo tanto de alto coste, además de que no todas las aplicaciones funcionan, sin embargo es muy probable que el rendimiento final sea mejor.

Cuando en una organización se cambia a un usuario a otro puesto, éste tiene que cambiar de permisos y responsabilidades, esto es difícil y llega a ser de alto coste.

Estos problemas pueden ser evitados por RBAC debido a que son los roles del usuario los que hacen que se tenga acceso al sistema, en vez de que sea la identificación del usuario.

Por lo tanto, identificamos entonces que un usuario es la persona que está en una organización, cuya labor básicamente es cumplir una función, en este caso un rol, cada rol realiza la función que se tiene que desempeñar, y puede tener a uno o más usuarios asignados a su cargo.

Los permisos determinan los datos y aplicaciones que pueden ser accedidas, a cada rol se le asignan diferentes privilegios, los cuales le son asignados dependiendo de las capacidades que se tengan para ejecutar el rol.

1.5 Estudios de módulos de administración de sistemas en la UCI

Un módulo de administración es una herramienta o subsistema Informático que le brinda una serie de facilidades al personal administrativo para gestionar el mejor funcionamiento del sistema que se administra. A continuación se describe el funcionamiento de algunos módulos de administración de los proyectos de la facultad 4.

1.5.1 Módulo de Administración del sistema de la Aduana

El proyecto Aduana de la facultad 4 en la UCI cuenta con un Módulo de Administración.

Dentro de las funcionalidades principales de este módulo se encuentran: gestionar el nivel de acceso de cada uno de los responsables de ejecutar las tareas en el sistema y actualizar la Base de Datos (BD).

Para garantizar el nivel de acceso del sistema se ha organizado el mismo de la siguiente forma, éste cuenta con muchos usuarios y estos están asociados a muchos grupos de usuarios del sistema, el sistema está organizado por módulos, donde cada módulo cuenta con una serie de funcionalidades que están asociadas a cada uno de los grupos del sistema. En fin, para garantizar el nivel de acceso de un usuario, el sistema busca a que grupos del mismo éste está asociado y a su vez a qué funcionalidades de los módulos estos grupos están

vinculados, después de obtener esta lista de funcionalidades el sistema automáticamente las habilita para el uso del usuario.

Para un mejor funcionamiento de la Aduana de nuestro país, ésta cuenta con un sistema central que se alimenta de muchos subsistemas que están desplegados en diferentes partes del país, por razones de conectividad todos los sistemas no están conectado al sistema central, debido a esta razón dichos subsistema tienen que generar un fichero donde se recojan todas las acciones ejecutadas en el sistema hasta el momento, para poder actualizar el sistema central, esta es la principal razón por el cual el Módulo de Administración de la Aduana brinda el servicio de cargar un fichero para actualizar cualquier parte de la BD del Sistema Central.

1.5.2 Módulo de Administración del ERP en el sistema del MINFAR

El proyecto del MINFAR de la facultad 4 en la UCI está implementando un Sistema Informático para esta organización, dicho sistema cuenta con un Módulo de Administración encargado de gestionar la seguridad de cada una de las funcionalidades que este pone a disposición de los usuarios.

El nivel de acceso está organizado de la siguiente forma, el sistema cuenta con muchos usuarios que pueden estar asociados a uno o muchos roles y estos a su vez con un número determinado de funcionalidades del sistema. Para lograr la autenticación y configuración del sistema en dependencia del usuario que se autentique se realiza el siguiente proceso, el sistema identifica el usuario y los roles a los cuales este está vinculado para obtener la lista de funcionalidades y activarlas al usuario que se ha autenticado.

1.6 Herramientas para el desarrollo del sistema

Entre las herramientas utilizadas y algunas de las que gozan de mayor popularidad entre la comunidad de desarrolladores a nivel internacional se encuentran las que se explicarán a continuación.

1.6.1 Aplicaciones Web y Sitios Web

“Una aplicación Web es un sitio Web donde la navegación a través del sitio, y la entrada de datos por parte de un usuario, afectan el estado de la lógica del

negocio. En esencia, una aplicación Web usa un sitio Web como entrada (front-end) a una aplicación típica.

...Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web.” [CONALLEN, June 1999]

Se puede definir que las aplicaciones Web implementan lógica de negocios y su uso cambia el estado del negocio, esto hace que lo más importante de una aplicación Web es la lógica de negocios y no la lógica de presentación y trae como ventaja que no es necesaria una instalación o configuración en el lado del cliente.

La arquitectura de una aplicación Web de modo general posee tres componentes principales: un servidor Web, una conexión de red, y uno o más clientes (browsers).

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP.

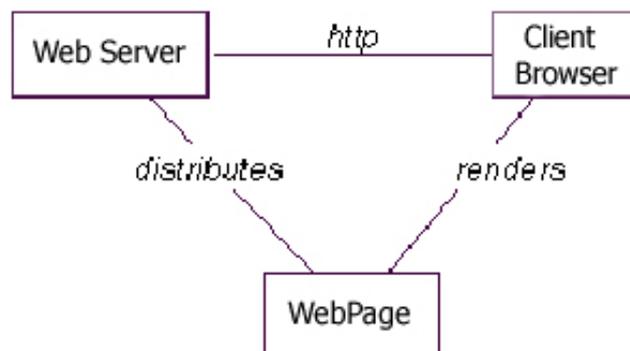


Figura 1.3. Arquitectura de una aplicación Web.

Las páginas Web son el componente principal de una aplicación o sitio Web. Los browsers piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web.

En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página.

Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe (a diferencia de otros esquemas tipo cliente/servidor). Es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente).

1.6.2 Estudios de Plataformas de Desarrollo

Plataforma Java

Java es una tecnología orientada al desarrollo de software con el cual podemos realizar cualquier tipo de programa. Hoy en día, la tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE. La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Con plataforma se refiere a la máquina virtual de Java (Java Virtual Machine).

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que un mismo programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras.

De hecho, su versatilidad y eficiencia, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en una opción tecnológica para su aplicación a redes, de manera que hoy en día, muchos dispositivos utilizan la tecnología Java. El lenguaje de programación Java ha sido totalmente mejorado, ampliado y probado por una comunidad de muchos desarrolladores de software.

Plataforma PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es una tecnología de desarrollo que funciona en toda máquina que sea capaz de compilar su código, entre ellas diversos sistemas operativos para PC y diversos Unix. Es relativamente multiplataforma, el código escrito en PHP en cualquier plataforma funciona exactamente igual en cualquier otra. Además es un lenguaje de Código

Abierto (Open Source) e interpretado, de alto nivel, potente, especialmente diseñado para desarrollos Web y el cual es generalmente mezclado con código HTML.

Como es software libre, implica menores costes que otras alternativas, a la vez que el tiempo entre el hallazgo de un fallo y su resolución es más corto. La meta de esta tecnología es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. Lo mejor de usar PHP es que es extremadamente simple, lo que lo hace productivo para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales.

Algunas de las características de las últimas versiones de PHP son el soporte mejorado para la programación orientada a objetos, nueva extensión para MySQL y manejo de errores y excepciones.

Plataforma .NET

La plataforma de desarrollo .NET es una nueva tecnología con énfasis en transparencia de redes, con independencia de plataforma, permitiendo un rápido desarrollo de aplicaciones, combinando la informática y las comunicaciones y ofreciendo las herramientas que se necesitan para transformar las aplicaciones Web.

.NET Utiliza los Servicios Web como un medio para poder interoperar a distintas tecnologías. Permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios. Les da a los desarrolladores las herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

La idea central que se implementa con la plataforma .NET es la del uso de servicios con el objetivo de simplificar el desarrollo de aplicaciones Web. Más concretamente software como servicio y de cómo construir, instalar, consumir, integrar o agregar estos servicios para que puedan ser accedidos mediante Internet. Esto es posible debido a que se tiene la infraestructura de comunicación global que es Internet cada vez más rápida y a un costo cada vez

menor y además, a la capacidad de los procesadores que continúa incrementándose año tras año.

La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a la funcionalidad que los Servicios Web proveen.

.NET está diseñada para que se puedan desarrollar componentes software utilizando casi cualquier lenguaje de programación, de forma que lo que se escriba en un lenguaje pueda utilizarse desde cualquier otro de la manera más transparente posible. Esto es, en vez de estar limitados a un único lenguaje de programación, permitir cualquier lenguaje de programación, siempre y cuando se adhiera a unas normas comunes establecidas para la plataforma .NET en su conjunto.

En el corazón de .NET se encuentra .NET Framework. Este contiene dos componentes principales: Common Language Runtime y la Biblioteca de Clases de Framework .NET, que incluye ADO.NET, ASP.NET y los formularios Windows Forms. Common Language Runtime es el fundamento de esta tecnología.

Por otra parte la Biblioteca de Clases es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario hasta aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

1.6.2.1 Tecnología ASP.NET

Es un conjunto de tecnologías de desarrollo de aplicaciones Web usadas para construir sitios Web domésticos y servicios XML. Esta novedosa herramienta forma parte de la plataforma .NET y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

ASP.NET rompe totalmente con el pensamiento de script que se tenía hasta el momento con su antepasado ASP. Una de las pocas características que se

mantiene de ASP a ASP.NET es la evolución de Visual Basic a Visual Basic .NET, por lo que el cambio se considera radical en su arquitectura.

También introduce el concepto del code-behind, por el que una misma página se compone de dos ficheros: el de la interfaz de usuario y el de código. Con ello se facilita la programación de aplicaciones en múltiples capas, lo que representa en definitiva la total separación entre lo que el usuario ve y lo que la base de datos tiene almacenado.

Dado que la Web no se lee secuencialmente sino que se compila, el primer resultado que llama la atención es el enorme incremento de velocidad de respuesta del servidor. Además, al compilarse, el incremento en seguridad y fortaleza es muy favorable.

1.6.2.2 ¿Por qué C# y no otro lenguaje de programación?

“C# es un nuevo lenguaje diseñado por Microsoft para la plataforma .NET por esta razón se suele decir que C# es el lenguaje nativo de .NET” [SECO, 2006]. Este nuevo lenguaje, posee la mayoría de las características de cualquier lenguaje de programación estándar, aunque su potencia reside en que es simple de utilizar y aprender, también reúne todo lo mejor de los lenguajes de programación existentes entre los que se destacan C++ y Java, junto a nuevas características que han sido añadidas debido a su demanda actual.

La razón fundamental de por qué se eligió C# en lugar de cualquier otro lenguaje es que C# se diseñó para la plataforma .NET y es capaz de utilizar al máximo nivel todo su potencial. La programación en este lenguaje se torna mucho más sencilla que en otros ya que por ejemplo C# carece de elementos heredados innecesarios de .NET. La sintaxis y estructuración de C# es de fácil entendimiento y similar a la del lenguaje C++, del que se tiene conocimiento de experiencias anteriores.

Además en C# se presenta lo que es denominado sistema unificado de tipos, en el que todos los tipos, incluso los primitivos, se derivan de un tipo objeto común, a la vez que permite el uso de optimizaciones para tipos primitivos y tipos sencillos. Contrariamente a la mayoría de lenguajes, C# no incluye una librería

específica, sino que utiliza la librería de clases de la plataforma .NET para todas sus necesidades, desde utilización de la consola hasta la programación multiproceso o el cifrado de seguridad.

Algunas de las características más representativas de C# se recogen a continuación de forma reducida:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- **Modernidad:** Incorpora en el propio lenguaje elementos que son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes hay que simular.
- **Orientación a objetos:** No admite ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos esto permite que se reduzcan problemas por conflictos de nombres y facilita la legibilidad del código.
- **Orientación a componentes:** La sintaxis de C# permite que se puedan definir **propiedades**, **eventos** y **atributos**; características que tienen que ser simuladas en otros lenguajes.
- **Seguridad de tipos:** Incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente.

Eficiencia: El código incluye numerosas restricciones para alcanzar su seguridad y no permite el uso de punteros.

1.6.2.3 Visual Studio .NET como herramienta de desarrollo

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes.

Dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web

ASP y servicios Web XML. Visual Studio.NET 2005 permite la creación de aplicaciones de consola, aplicaciones para Windows, archivos DLL, aplicaciones Web, XML Web Services y aplicaciones para dispositivos de bolsillo.

Ofrece algunas características exclusivas de alta productividad, tales como: diseñadores visuales de Web Forms y Windows Forms, esquemas XML y datos; un depurador de varios lenguajes que alterna sin problemas entre códigos escritos en lenguajes diferentes; una estrecha integración con .NET Framework; ayuda dinámica, que proporciona una ayuda contextual continua mientras se escribe; lista de tareas, que muestra los errores del compilador y las tareas pendientes y un explorador de servidores para obtener acceso visual a bases de datos, servicios de Windows, contadores de rendimiento y componentes de aplicaciones del lado del servidor.

1.6.2.4 Servicios Web (Web Services)

Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Una posible sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios.

“Los servicios Web, son por definición poco acoplados” [DANIEL RUBIOLLO, 2004]. Esto significa que puede cambiar la implementación en cualquier extremo de la conexión sin afectar el otro extremo. Técnicamente, esto se logra utilizando tecnología asíncrona basada en mensajes, con excelente desempeño y utilizando protocolos Web tales como HTTP, y lo más importante, XML para lograr un alcance universal.

Los Servicios Web permiten distribuir e integrar la lógica de aplicación a través de Internet. El concepto de distribuir lógica aplicación a través de la red no es nuevo. Lo que es nuevo, es el concepto de distribuir e integrar lógica de aplicación a través de Internet. Ya con unos pocos años de implantación los servicios Web serán quizás la tecnología futura de todas las aplicaciones.

1.6.3 ¿Por qué RoboHelp Office X5 como herramienta de desarrollo de la ayuda?

Es un software de elaboración de ayudas, favorito en la red a nivel mundial, para el desarrollo rápido y fácil de sistemas completos de ayuda. Es ideal en la creación de ayudas dirigidas a aplicaciones de software, aplicaciones multiplataforma, aplicaciones basadas en Web, Sitios Web, manuales en línea y documentación impresa. Dentro de sus principales ventajas se destacan las siguientes:

- **Rápido y Fácil de usar:** Este permite a cualquier persona crear Ayudas con funciones completas en poco tiempo.
- **Empleo de contenidos existentes:** Permite el uso de archivos FrameMaker, documentos de Word, documentos HTML u otros archivos de ayuda existentes, lo cual se traduce en un uso más rápido y completo de su información existente ya que RoboHelp los incorpora y respeta sus Índices, TOC y Glosarios de estos archivos y los incorpora a su proyecto.
- **Salida en múltiples formatos:** Crea Ayudas en todos los formatos de ayuda - WebHelp, Microsoft HTML Help, WinHelp, Oracle Help for Java and JavaHelp, desde un sólo conjunto de código fuente.
- **Soporta los editores más famosos:** Este desarrolla todos los formatos de Ayuda con una única edición flexible (cualquier editor HTML o Microsoft Word).
- **Máxima versatilidad:** RoboHelp crece con sus necesidades y le permite la transición a nuevos formatos de Ayuda con tecnología de un sólo archivo fuente. Maximiza su tiempo y esfuerzo invertidos creando nuevas posibilidades de asistencia a los usuarios desde un sólo proyecto de Ayuda.
- **Mejora el manejo:** Proporciona información completa y fácil de utilizar, mejorando el manejo y administración de sus aplicaciones.
- **Capacidades de internacionalización:** Dispone de 11 diferentes idiomas incluyendo: inglés, danés, holandés, francés, alemán, italiano, noruego, portugués, español, sueco, o portugués brasileño.

- **Incrementa su productividad con 15 herramientas extras:** Herramientas gratis para redimensionar gráficos, mejorar la forma de cargar archivos, búsqueda de errores, y más, dentro de su proyecto de Ayuda.

1.6.4 Metodologías de desarrollo

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte se tienen aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán.

Todo este desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

Dentro de los tipos de metodologías más importantes que se pueden usar se encuentran:

Rational Unified Process (RUP), Extreme Programming (XP) y Microsoft Solution Framework (MSF) entre otros.

Extreme Programming (XP)

Es una metodología de desarrollo de software con éxito en la actualidad, utilizadas para proyectos de corto plazo, pequeño equipo y cuyo plazo de entrega es rápido. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores
- La simplicidad, al desarrollar y codificar los módulos del sistema
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

Rational Unified Process (RUP)

El RUP es un proceso de ingeniería de software que utiliza el paradigma de orientación a objetos para su descripción. Es un marco de proceso configurable para satisfacer necesidades específicas e implementa las mejores prácticas de desarrollo de software.

Esta metodología posee 3 características principales:

- **Dirigido por casos de uso:** Los casos de uso capturan requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario.
- **Centrado en una arquitectura:** Comprende los aspectos estáticos y dinámicos más importantes del sistema.
- **Iterativo e incremental:** El trabajo se divide en piezas pequeñas o mini proyectos; cada uno proveyendo un subproducto incremental.

Por tanto, el uso de esta metodología es la más adecuada, además su proceso iterativo e incremental posee grandes ventajas como son:

- Reducción de riesgos basado en la retroalimentación temprana.
- Pruebas continuas e iterativas promueven una mejor evaluación del estado del proyecto.
- Los patrocinadores reciben evidencia concreta del avance del proyecto.

- Se pueden acomodar mejor los cambios (requerimientos, tácticos y tecnológicos).

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Esta es la metodología más adaptable para proyectos de largo plazo.

1.6.5 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo.

El objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos). [RUMBAUGH *et al.*]

En resumen, UML resuelve de forma bastante satisfactoria un viejo problema del desarrollo de software como es su modelado gráfico.

1.6.6 Herramienta Rational Rose Enterprise Edition

Rational Rose Enterprise Edition es una herramienta de software para el modelado visual mediante UML de sistemas de software, permite especificar, analizar y diseñar el sistema antes de codificarlo. Mantiene la consistencia de los modelos del sistema software, además de un chequeo de la sintaxis UML.

Las herramientas Rational unifican el equipo de proyecto y esta forma parte del conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas (flujos fundamentales) del RUP e incluye un conjunto de herramientas de ingeniería inversa y generación de código que allanan el camino hasta el

producto final. En fin la plataforma Rational fundamentalmente mejora la manera que las organizaciones tienen de construir software para su mejor desarrollo.

1.7 Conclusiones

En el capítulo que se concluye se realizaron estudios sobre la administración de sistemas informáticos e investigaciones para definir las herramientas, metodologías y plataformas para el desarrollo del módulo que corresponde a la presente investigación.

Después de hacer un análisis de toda la información recopilada, sobre la Administración de Sistemas Informáticos y Controles de Acceso, además de investigar el comportamiento de los procesos de administración a nivel mundial y las experiencias de algunos de los proyectos de la facultad, se puede concluir que lo estudiado ayuda a determinar las responsabilidades que debe incluir la administración de sistemas, contando de esta manera con los elementos necesarios para realizar la propuesta que se presente a los usuarios, con el objetivo de que sea ajustada al sistema que se desea administrar. Dichos aspectos son tratados en el Capítulo 2.

Finalmente para el desarrollo del sistema se utilizarán las siguientes tecnologías por las características fundamentadas con anterioridad: la plataforma .NET de Microsoft (C# y ASP .NET), como metodología, lenguaje de modelado y herramienta de modelado, RUP, UML y Rational Rose respectivamente.

Capítulo 2: Características del Sistema

Introducción

A través de este capítulo se describe el objeto de estudio, se expone la solución propuesta con los requisitos funcionales y no funcionales que el Módulo de Administración debe cumplir, finalmente se describen los actores y los casos de uso. La documentación presentada se basa en el Proceso Unificado de Racional (RUP).

2.1 Objeto de estudio

En la actualidad, el manejo de la información es parte fundamental de cualquier empresa u organismo, para la realización de sus actividades de una forma rápida y eficiente. Con los adelantos tecnológicos en el área computacional, área de comunicaciones y tecnologías de información, se le ha dado suma importancia al uso de Sistemas de Información basados en computadora o Sistemas Informáticos, aprovechando los beneficios que estos les otorgan en el procesamiento de la información de una forma rápida y confiable.

Para lograr un mejor funcionamiento de cualquier Sistema Informático, es de suma importancia tener en cuenta la administración del software, ya que esta es la base esencial en la seguridad del mismo y a su vez le ofrece mayor confianza al usuario final. La administración de un Sistema Informático no es totalmente eficiente sino se realiza un estudio profundo de las tendencias actuales en esta rama, que garanticen una mejor selección de las características administrativas que se van a implementar en el software.

2.1.1 Problema y situación problemática

El Banco Central de Cuba y el proyecto TeleBanca que está ubicado en la Universidad de las Ciencias Informáticas (UCI), se ha dado a la tarea, realizar un Sistema Informático para garantizar el pago de algunos servicios por vía telefónica. Para garantizar el mejor funcionamiento de dicho Sistema Informático, el mismo debe tener la capacidad de gestionar los niveles de accesos de cada uno de los responsables de informar y ejecutar tareas, así como garantizar la configuración, salvadas y restauración de la Base de Datos y permitir consultar las

notificaciones de los problemas existentes a la hora de ejecutar una funcionalidad del sistema responsable de efectuar el pago de los servicios.

2.1.2 Objeto de automatización

Se desea automatizar todos los procesos que garanticen la administración del sistema en cuanto a gestión de roles, usuario, configuración, salva y restauración de datos, notificaciones de errores, o sea, todo lo que abarca la administración del Sistema Informático del proyecto TeleBanca, para que este posea una mayor seguridad de las funcionalidades que brinda y un mejor entorno de trabajo para la persona encargada de la administración del mismo.

Por lo tanto, teniendo en cuenta los argumentos planteados hasta este momento con relación a la administración en Sistemas de Información, se propone el presente módulo de administración como vía para lograr una mejor gestión del sistema encargado del pago vía telefónica.

2.1.3 Propuesta de sistema

Con el objetivo de lograr una mejor administración del Sistema Informático encargado de efectuar el pago vía telefónica del proyecto TeleBanca, se ha decidido desarrollar un módulo, que de alguna manera solucione, de la forma más eficiente, la administración de dicho sistema. Este Módulo debe brindar una forma cómoda para gestionar los niveles de acceso e información de los usuarios que utilizarán el sistema y una fácil configuración del mismo.

Dentro del estudio realizado se investigó acerca de la administración de sistemas y la seguridad de los mismos, de la cual se tomaron algunos puntos como guía, para seleccionar qué procesos o elementos no podían ser obviados en el Módulo de Administración del sistema TeleBanca.

Autenticación: Procedimiento de comprobación de la identidad de un usuario.

Autorización: Consiste en permitir el acceso personal a los elementos de un sistema a partir de la identidad de la persona.

Control del acceso: Después de la identificación y autenticación inicial, los controles de acceso les permiten a los usuarios acceder a las pantallas para

realizar ciertas tareas. En esencial estos controlan el acceso individual a las capacidades de los sistemas y le permite al administrador, personalizar y definir los derechos de los usuarios. Al utilizar los controles de acceso, el administrador puede definir quién tiene acceso para acceder a que pantalla del sistema y realizar determinadas tareas.

Criptografía: Técnica para codificar la información de forma que solo pueda ser interpretada por quienes tengan acceso a las claves de su codificación.

Copia de respaldo: Copia de los datos de un fichero automatizado en un soporte que posibilite su recuperación.

Configuración de sistema: Son los ficheros o datos que le permiten al sistema, realizar con éxito sus funcionalidades, por ejemplo la dirección del servidor de Base de Datos, servidor FTP, impresoras, entre otros datos.

Consulta de notificaciones: Notificación de los problemas existentes a la hora de ejecutar una funcionalidad del sistema.

Para el desarrollo de la administración en el sistema, es necesario tener almacenados los datos de los usuarios, incluyendo usuario y contraseña, datos por los cuales el sistema reconoce al usuario y le da acceso según el rol que este juegue dentro de la empresa, aunque el usuario luego tiene la posibilidad de modificar su contraseña.

También se registrarán todos los datos relacionados con la configuración del sistema, como son: Dirección del servidor de Base Datos, Dirección del servidor FTP, Usuario para conectarse al FTP, Contraseña para conectarse al FTP, Dirección de la salva de la BD, Hora de Salva, Hora de inicio de las peticiones, Hora de conciliaciones, Tiempo de inactividad.

Otros datos que serán almacenados son los referentes a la empresa donde estará instalado el Software, como son: Nombre, Dirección, Teléfono, Fax, Sitio Web, Logotipo, Nombre del director, Correo Electrónico, Descripción de los servicios que brinda, Organismo al que pertenece. Todos estos datos expresados anteriormente serán introducidos por el administrador.

2.1.4 Especificación de los requisitos de software

Para obtener las especificaciones de los requisitos que debe cumplir la aplicación a desarrollar, se utilizó como base el estudio realizado sobre las funcionalidades y procesos característicos de la administración de sistemas y se elaboró una propuesta que fue presentada a los usuarios.

Esta propuesta fue revisada, ajustada y enriquecida con los elementos propios del sistema que se desarrollará y que debe ser administrado, así como necesidades específicas de los usuarios finales.

De los talleres y encuentros realizados se definieron como requisitos funcionales y no funcionales los que en los siguientes epígrafes se especifican.

2.1.4.1 Requisitos Funcionales

1. Gestionar información de los Usuarios

*De los usuarios se registran los siguientes datos: nombre, usuario, contraseña, rol y Carné identidad. Se definen el nivel de acceso que poseen en dependencia de la función que realicen.

1.1 Insertar nuevo usuario. El sistema debe ser capaz de permitir adicionar un nuevo usuario.

1.2 Modificar Usuario. El sistema debe ser capaz de permitir modificar los datos de un usuario anteriormente creado.

1.3 Eliminar usuario. El sistema debe ser capaz de permitir eliminar un usuario anteriormente creado

Seguimiento: CU_Gestionar Usuario.

2. Realizar autenticación de usuario

El sistema debe ser capaz de permitir autenticar un usuario en todo momento.

*Mostrar mensaje de bienvenida y configurar el sistema en dependencia de las funcionalidades que pueda realizar, si el usuario y la contraseña son válidos y si no, mostrar mensaje de error.

Seguimiento: CU_Autenticar Usuario.

3. Definir rol de usuario

* De los roles se registran los siguientes datos: nombre, descripción del rol, lista de funcionalidades que tiene acceso.

- 3.1 Insertar nuevo rol. El sistema debe ser capaz de permitir adicionar un nuevo rol.
- 3.2 Modificar rol. El sistema debe ser capaz de permitir modificar los datos correspondientes a un rol anteriormente creado.
- 3.3 Eliminar rol. El sistema debe ser capaz de permitir eliminar un rol anteriormente creado.

Seguimiento: CU_Gestionar Roles.

4. Acceso a la Ayuda

El sistema debe ser capaz de permitir a todos los usuarios autenticados acceder a la ayuda del sistema.

Seguimiento: CU_Acceso Ayuda.

5. Actualizar información de la Banca Telefónica

El sistema debe ser capaz de permitir crear o modificar los datos correspondientes a la Banca Telefónica.

*Los datos son: nombre, dirección, Teléfono, Fax, Sitio Web, logotipo, nombre del director, ministerio, correo electrónico, breve descripción de los servicios que brinda.

Seguimiento: CU_Actualizar información BT.

6. Modificar clave de usuario

El sistema debe ser capaz de permitirles a los usuarios autenticados modificar su clave.

Seguimiento: CU_Modificar Clave.

7. Gestionar copias de respaldo

7.1 Salvar Datos. El sistema debe ser capaz de realizar en cualquier momento que se indique, salvadas de la base de datos y guardarlas en el lugar que estará indicado por la configuración del sistema.

7.2 Restaurar Datos.

El sistema debe ser capaz de permitir realizar en cualquier momento que sea indicado, una restauración de la base de datos a partir de una salva anteriormente hecha.

Seguimiento: CU_Salvar y Restaurar Datos.

8. Definir la configuración del Sistema

El sistema debe ser capaz de permitir modificar o crear la configuración del sistema.

*Los elementos de configuración son: dirección del servidor de base datos, dirección del servidor FTP, usuario y contraseña para conectarse al FTP, dirección de la salva de la base de datos, hora de salva, hora de inicio de las peticiones y las conciliaciones, tiempo de inactividad, para que expire una sección, dirección de la impresora de pines y dirección de la impresora de tarjetas.

Seguimiento: CU_Configuración.

9. Consultar las notificaciones

El sistema debe ser capaz de permitir mostrar las notificaciones que han sido almacenadas en el sistema.

9.1 Eliminar notificaciones. El sistema debe ser capaz de permitir eliminar cualquier notificación seleccionada.

Seguimiento: CU_Consultar Notificación.

2.1.4.2 Requisitos No Funcionales

2.1. Requerimientos de apariencia o interfaz externa

El módulo propuesto será usada por personas que tengan conocimiento de la informática, pero la interfaz debe ser amigable y fácil de usar, de manera que no sea una dificultad para los usuarios el uso de esta.

2.2. Requerimientos de Usabilidad

A los administradores finales de la aplicación se les debe dar un adiestramiento básico en el uso de la aplicación. Estas personas deben tener un nivel de acceso amplio en la aplicación para poder darle respuesta a cada incidente ocurrido.

2.3. Requerimientos de Rendimiento

Para un funcionamiento óptimo de la aplicación se deben seguir las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a sus páginas. La eficiencia del producto debe estar determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas en la Base de Datos.

La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el cliente requiere la respuesta a su petición.

2.4. Requerimientos de Soporte

Para garantizar el soporte de esta herramienta, se debe documentar la aplicación con un manual de ayuda para los usuarios, así como la posibilidad de emitir sus quejas y sugerencias a los desarrolladores de la herramienta, por correo o por teléfono, realizar mantenimiento al sistema y darle solución a cualquier problema que surja con la aplicación.

2.5. Requerimientos de Portabilidad

La herramienta propuesta debe poder usarse bajo cualquier sistema operativo Windows 2000 en adelante, y la Base de Datos bajo Windows Server 2000 en adelante. Los servidores Web y de Base de Datos deben estar en PC distintas. Para su implementación se debe usar como herramienta de desarrollo Visual Studio .NET y Gestor de Base de Datos SQL Server 2000 en adelante.

2.6. Requerimientos de Seguridad

Confiabilidad: La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación.

Integridad: La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

Disponibilidad: La aplicación debe estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para

lograr la seguridad no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

2.7. Requerimientos de Software

En la computadora de los usuarios sólo se requiere de navegador para Internet o Intranet, bajo cualquier sistema operativo Windows 2000 en adelante, y en la Base de Datos, Windows Server 2000 en adelante. Para su implementación se usará como herramienta de desarrollo Visual Studio .NET y Gestor de Base de Datos SQL Server 2000 en adelante.

2.8. Requerimientos de Hardware

En el cliente se requerirá de una máquina de 128 MB de RAM, dos servidores Web con 1 Giga de RAM como mínimo, y el servidor de Base de Datos con 2 Giga de RAM y 80 GB de capacidad del disco duro, todas las computadoras implicadas en los servidores, la administración como las de los usuarios, deben estar conectadas a una red y tener al menos 100 Mbps de velocidad.

2.9. Restricciones en el diseño y la implementación

Se debe realizar una herramienta que permita dar respuesta en el menor tiempo posible, garantizando la calidad del sistema y la conexión con el Web Services, la base de datos debe ser desarrollada en SQL Server. Para garantizar el desarrollo de la aplicación se utilizará como guía el Proceso Unificado de Desarrollo, obteniendo como documentación los distintos artefactos propuestos. Se utilizará para realizar los modelos del sistema UML (Unified Modelling Language) y como herramienta de apoyo a este Lenguaje de Modelación se utiliza Rational Rose. Para la gestión de configuración se utilizará Subversión y Tortoise.

- Políticos-culturales.
- Legales.
- Confiabilidad.
- Interfaz.
- Ayuda y documentación en línea.

2.1.5 Definición de los casos de uso

- Definición de los actores

Actores	Justificación
Usuario	Cualquier trabajador que interactúe con el sistema, puede ser Administrador de Información, Administrador de BT, Operadora de Autenticación, Operadora de Servicio de Pago.
Administrador	Es la persona que tiene la responsabilidad de asegurar el correcto funcionamiento y mantenimiento del sistema.

- Listado de casos de uso

CU_ Autenticar Usuario	Autenticar Usuario
Actor	Usuarios
Descripción	El usuario se identifica con su Usuario y Contraseña, si sus datos son correctos este podrá entrar al sistema, definiéndosele el nivel de acceso en dependencia del rol que este asociado al usuario.
Referencia	2

CU_ Gestionar Roles	Gestionar Roles
Actor	Administrador
Descripción	El administrador puede crear un rol introduciendo los datos pertinentes: nombre del rol, descripción del rol y nivel de acceso al cual estará asociado, además puede modificar los datos de un rol anteriormente creado, al igual que eliminarlo.
Referencia	3, 3.1, 3.2, 3.3

CU_Gestionar Usuario	Gestionar Usuario
Actor	Administrador
Descripción	El administrador puede crear un usuario introduciendo los datos pertinentes: nombre, CI, usuario, contraseña y define el rol al cual estará asociado, además puede modificar los datos de un usuario anteriormente creado, al igual que eliminarlo.
Referencia	1, 1.1, 1.2, 3,3.1, 3.2, 3.3

CU_ Modificar clave	Modificar clave
Actor	Usuarios
Descripción	El usuario introduce su contraseña actual, la nueva y la confirmación de la nueva, si estos datos son correctos se modifica la contraseña del usuario.
Referencia	6

CU_Actualizar información BT	Actualizar información BT
Actor	Administrador
Descripción	El administrador introduce los datos pertinentes y crea o modifica la información más relevante de la entidad donde esté instalada la aplicación.
Referencia	5

CU_Configuración	Configuración
Actor	Administrador
Descripción	El administrador introduce los datos pertinentes y crea o modifica los datos configurables del sistema.

Referencia	8
-------------------	---

CU_ Consultar notificación.	Consultar notificación
Actor	Administrador
Descripción	El administrador selecciona las notificaciones o mensajes de error que el determine que ya no le son necesario y los elimina.
Referencia	9

CU_Acceso Ayuda.	Acceso Ayuda.
Actor	Usuarios
Descripción	Los usuarios acceden a la ayuda del sistema.
Referencia	4

CU _ Salvar y Restaurar Datos	Salvar y Restaurar Datos
Actor	Administrador
Descripción	El administrador realiza la salva de la base de datos en el momento que este lo crea pertinente, además puede recuperarlos a partir de una salva anteriormente realizada.
Referencia	7, 7.1, 7.2

- Diagrama de casos de uso

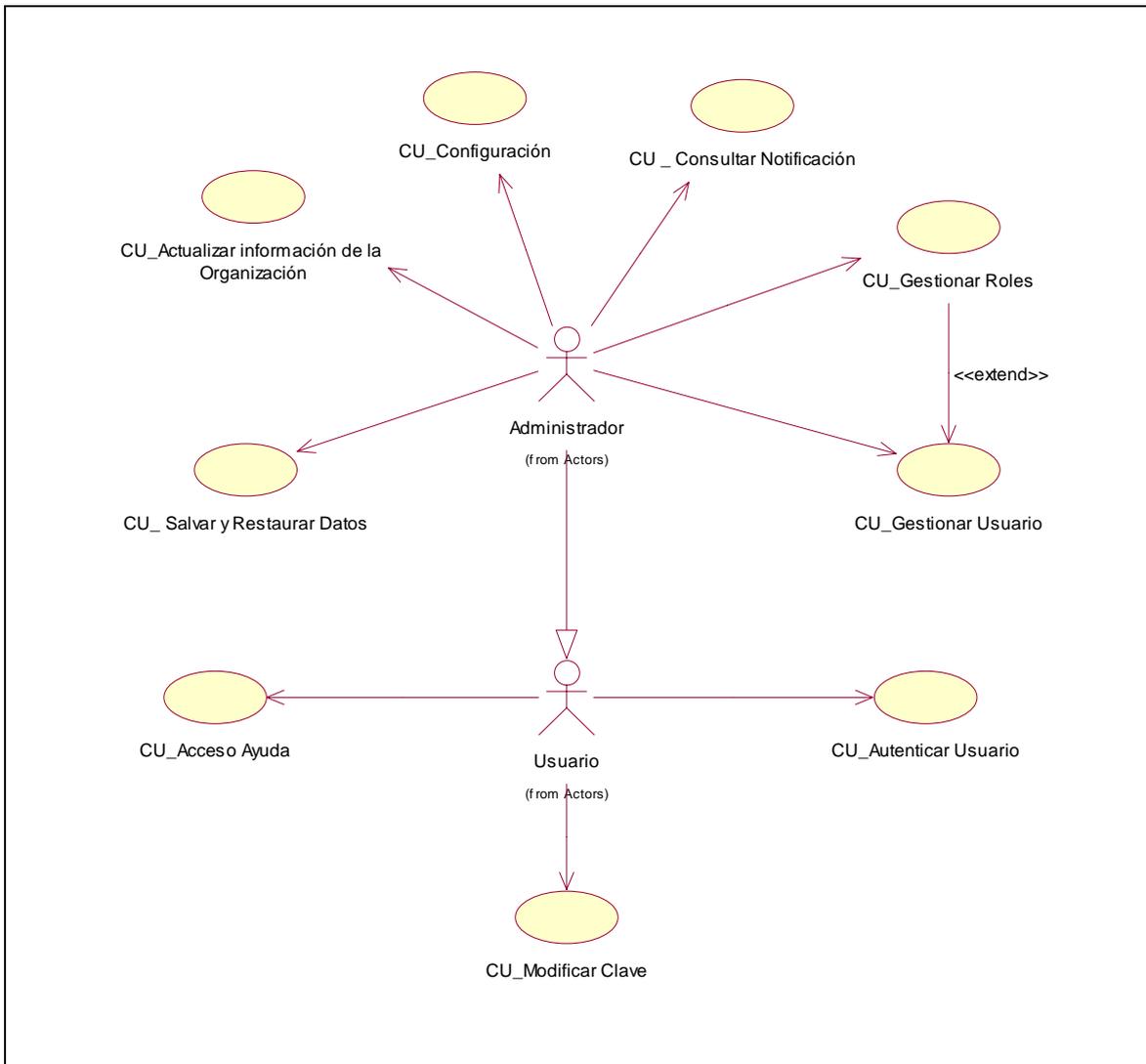


Figura 2.1. Diagrama de casos de uso del sistema

- Casos de uso por ciclo

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
	Autenticar Usuario		Permite identificar el usuario y decidir si este podrá entrar al sistema.
	Gestionar Roles		Le permite al administrador configurar los roles del sistema, definiéndole a que funcionalidades

1		Administración	este estará vinculado.
	Gestionar Usuario		Permite gestionar los datos de los usuarios del sistema y definir que rol este desempeñará.
	Modificar clave		Les brinda la posibilidad a todos los usuarios de modificar su contraseña cuando estos lo crean pertinente.
	Actualizar información BT		Permite que el administrador actualice los datos de la organización donde esté instalado el sistema.
	Configuración		Permite que el administrador actualice los datos configurables del sistema para su exitoso funcionamiento.
	Consultar notificación.		Permite que el administrador consulte los mensajes que ha generado el sistema y eliminarlos cuando este lo decida.
	Acceso Ayuda.		Les permite a todos los usuarios acceder a la ayuda del sistema.
	Salvar y Restaurar Datos		Permite que el administrador realice salvadas de la base de datos en el momento que este lo crea pertinente, además facilita la recuperación de los datos a partir de una salva anteriormente realizada.

- **Casos de uso expandidos**

A continuación se muestra la descripción de los casos de uso más relevantes del sistema: Autenticar Usuario, Gestionar Usuario, Gestionar Rol. La descripción de los restantes casos de uso se encuentra en el Anexo 2.

a) Especificación del Caso de Uso: CU_ Autenticar Usuario

1. CU_ Autenticar Usuario

Este caso de uso tiene como objetivo permitirle a los Usuarios acceder al Sistema y utilizarlo según su nivel de acceso.

2. Flujos de eventos

a. Flujo básico

Caso de Uso:	CU_ Autenticar Usuario
Actores:	Usuario.
Resumen:	El caso de uso comienza cuando el usuario solicita entrar al Sistema, el mismo se identifica con su usuario y contraseña. Termina cuando el usuario entra al sistema o es denegado el acceso en caso de que no esté autorizado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso comienza cuando el usuario solicita acceder al sistema.	2. El sistema solicita usuario y contraseña correspondiente.
3. El usuario introduce los datos al sistema.	
4. El usuario solicita entrar al sistema.	5. El sistema verifica si el usuario y la contraseña están correctos.
	6. Se configura el sistema según el rol del usuario.
	7. el sistema muestra el mensaje "Bienvenido a Telebanca"
	8. Se termina el caso de uso.

b. Flujos alternativos

Acción 4 Si el usuario selecciona la opción cancelar.

Acción 5 F1: Si el sistema no pudo verificar el usuario y la contraseña porque no hay conexión con la base de datos.

Acción 5 F2: Si el usuario y la contraseña no son válidos.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 4	4.1 Se termina el caso de uso.
Acción 5 F1	5 F1.1 El Sistema muestra el mensaje "Error de conexión con la base de datos".
5 F1.2 El Administrador acepta el mensaje.	5 F1.3 Pasar a la acción 2.
Acción 5 F2	5 F2.1- El Sistema muestra el mensaje "Usuario o contraseña no válidos. Intente de nuevo".
5 F2.2 El Administrador acepta el mensaje.	5 F2.3 Pasar a la acción 2.

3. Requisitos

1.2 Realizar autenticación de Usuario.

4. Requisitos Adicionales del Caso de Uso

No tiene.

5. Precondiciones

Un usuario solicita entrar al sistema.

6. Poscondiciones

Se muestra un mensaje de bienvenida además de que puede acceder al sistema en caso de que el usuario y la contraseña sean correctos y un mensaje de error en caso de que sean incorrectas.

7. Puntos de extensión

No tiene.

b) Especificación del Caso de Uso: CU_Gestionar Roles

1. CU_ Gestionar Roles

Este caso de uso tiene como objetivo permitirle al administrador definir los roles del sistema a partir de sus funcionalidades.

2. Flujos de eventos

a. Flujo básico.

Caso de Uso:	CU_ Gestionar Roles
Actores:	Administrador.
Resumen:	El caso de uso comienza cuando el administrador selecciona la opción gestionar roles. A partir de ahí el administrador tiene la opción de adicionar, modificar o eliminar un rol. Finaliza el caso de uso cuando el Administrador termina la acción que ha seleccionado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el Administrador, selecciona la opción de gestionar rol.	2. El sistema muestra todos los roles existentes y las opciones adicionar, modificar y eliminar rol.
3. El Administrador selecciona una de las opciones: a) Solicita adicionar rol, Ver Sección Adicionar Rol. b) Solicita modificar el rol seleccionado, Ver Sección Modificar Rol.	

<p>c) Solicita eliminar el o los roles seleccionados, Ver Sección Eliminar Rol.</p>	
Sección Adicionar Rol	
	<p>4. El sistema muestra el formulario para definir un rol:</p> <p>a) Nombre del rol. b) Descripción. c) Lista de funcionalidades.</p>
<p>5. El Administrador introduce los datos del nuevo rol y selecciona las funcionalidades a asociar.</p>	
<p>6. El Administrador solicita guardar el nuevo rol.</p>	<p>7. El sistema valida los datos de entrada.</p>
	<p>8. El sistema valida que no exista otro rol con el nombre del rol a crear.</p>
	<p>9. El sistema valida que no exista otro rol con las mismas funcionalidades asignadas.</p>
	<p>10. El sistema valida que no tenga seleccionada a la vez las funcionalidades: Imprimir PIN e Imprimir Tarjeta.</p>
	<p>11. El sistema solicita confirmación de que desea crear el nuevo rol. “¿Desea guardar el nuevo rol?”</p>
<p>12. El Administrador acepta guardar los datos del nuevo</p>	<p>13. El sistema guarda los datos satisfactoriamente.</p>

rol.	
	14. El Sistema muestra un mensaje. "Se adicionó el rol con éxito".
15. El Administrador acepta el mensaje.	16. Se pasa a la Acción 2.
Sección Modificar Rol.	
	17. El sistema muestra el rol con sus datos: a) Nombre. b) Descripción. c) La lista de todas las funcionalidades existentes marcando las que tiene asociadas.
18. El Administrador modifica los datos del rol.	
19. El Administrador solicita guardar los cambios.	20. El sistema valida la entrada de datos
	21. El sistema valida que no exista otro rol con las mismas funcionalidades.
	22. El sistema solicita confirmación de que desea modificar el rol. "¿Desea guardar los datos?"
23. El Administrador acepta guardar los datos.	24. El sistema guarda los cambios del rol.
	25. El sistema muestra un mensaje. "Se modificó el rol con éxito".
26. El Administrador acepta el mensaje.	27. Se pasa a la Acción 2.

Sección Eliminar Rol	
	28. El sistema verifica que no exista ningún usuario con el rol seleccionado asignado.
	29. El sistema solicita la confirmación de que desea eliminar el rol seleccionado. “¿Desea eliminar el rol?”
30. El Administrador confirma que desea eliminar el rol seleccionado.	31. El sistema elimina el rol seleccionado.
	32. El sistema muestra un mensaje. “Se eliminó el rol con éxito”
33. El Administrador acepta el mensaje.	34. Se pasa a la Acción 2.

b. Flujos alternativos

Acción 2: El sistema no pudo mostrar los roles existentes porque la conexión con la Base de Datos está deshabilitada.

Acción 3: Si el administrador selecciona cancelar la acción de gestionar roles.

Acción 4: Si no se pueden cargar las funcionalidades porque la conexión con la Base de Datos está deshabilitada.

Acción 6: Si el administrador selecciona cancelar la acción de guardar el rol.

Acción 7: Si no son válidos los valores de entrada.

Acción 8: Si existe un rol con el nombre del rol a insertar.

Acción 9: Si las funcionalidades seleccionadas son las mismas de otro rol ya existente.

Acción 10: Si se seleccionaron a la vez las funcionalidades: imprimir PIN e imprimir matriz.

Acción 12: Si el administrador selecciona cancelar la solicitud de confirmación de insertar los datos del nuevo rol.

Acción 13: Si no se pueden guardar los datos del nuevo rol porque la conexión con la Base de Datos está deshabilitada.

Acción 19: Si el administrador selecciona cancelar la acción de modificar un rol existente.

Acción 20: Si no son válidos los valores de entrada.

Acción 22: Si existe otro rol con las mismas funcionalidades asignadas.

Acción 24: Si el administrador selecciona cancelar la solicitud de confirmación de guardar los datos modificados.

Acción 25: Si el sistema no puede guardar los cambios porque la conexión con la Base de Datos está deshabilitada.

Acción 29 F1: La conexión con la Base de Datos está deshabilitada.

Acción 29 F2: Si el rol está asignado a un usuario.

Acción 30: Si el administrador selecciona cancelar la solicitud de eliminar el rol.

Acción 31: Si el sistema no puede eliminar el rol porque la conexión con la Base de Datos está deshabilitada.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 2	2.1 El sistema muestra el mensaje. "El siguiente error ha ocurrido: " y se detalla el error.
2.2 El Administrador acepta el mensaje.	2.3 Se termina el caso de uso.
Acción 3	3.1 Se termina el caso de uso.
Acción 4	2.1 El sistema muestra el mensaje "El

	siguiente error ha ocurrido: " y se detalla el error.
4.2 El Administrador acepta el mensaje.	4.3 Se termina el caso de uso.
Acción 6	6.1 Se pasa a la Acción 2.
Acción 7	7.1 El sistema muestra el mensaje" Entrada de datos no válida".
	7.2 El Administrador acepta el mensaje.
7.3 El sistema muestra un asterisco al lado de cada entrada no válida.	7.4 Se pasa la Acción 4.
Acción 8	8.1 El sistema muestra el mensaje" El rol ya existe". Va a la Acción 4.
8.2 El Administrador acepta el mensaje.	8.3 Se pasa a la Acción 4.
Acción 9	9.1 El sistema muestra el mensaje. "No pueden existir dos roles con las mismas funcionalidades".
9.2 El Administrador acepta el mensaje.	9.3 Se pasa a la Acción 4.
Acción 10	10.1 El sistema muestra el Mensaje. "No se pueden seleccionar a la vez las funcionalidades: Imprimir PIN e Imprimir Tarjetas ".
10.2 El Administrador acepta el mensaje.	10.3 Se pasa la Acción 4.
Acción 12	12.1 Se pasa a la Acción 4.
Acción 13	13.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.

	13.2 El sistema muestra el mensaje. "El siguiente error ha ocurrido: " y se detalla el error.
13.3 El Administrador acepta el mensaje.	13.4 Se pasa a la Acción 4.
Acción 19	19.1 Se pasa a la Acción 2.
Acción 20	20.1 El Sistema muestra el mensaje "Entrada de datos no válida".
	20.2 El Administrador acepta el mensaje.
20.3 El sistema muestra un asterisco al lado de cada entrada no válida.	20.4 Se pasa a la Acción 17.
Acción 22	22.1 El Sistema muestra el mensaje. "No pueden existir dos roles con las mismas funcionalidades".
22.2 El Administrador acepta el mensaje.	22.3 Se pasa a la Acción 17.
Acción 24	24.1 Se pasa a la Acción 17.
Acción 25	25.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	25. 2 El Sistema muestra el mensaje. "El siguiente error ha ocurrido: " y se detalla el error.
25.3 El Administrador acepta el mensaje.	25.4 Va a la Acción 17.
Acción 29 F1	29 F1.1 El sistema muestra el mensaje "El siguiente error ha ocurrido: " y se detalla el error.
29 F1.2 El Administrador acepta el mensaje.	29 F1.3 Se pasa a la Acción 2.

Acción 29 F2	29 F2.1 El Sistema muestra el mensaje. "El rol no puede estar asignado a ningún usuario".
29 F2.2 El Administrador acepta el mensaje.	29 F2.3 Se pasa a la Acción 2.
Acción 30	30.1 Se pasa a la Acción 2.
Acción 31	31.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	31.2 El Sistema muestra el mensaje. "El siguiente error ha ocurrido: " y se detalla el error.
31.3 El Administrador acepta el mensaje.	31.4 Se pasa a la Acción 2.

3. Requisitos

3. Definir rol de usuario.

3.1 Insertar nuevo rol.

3.2 Modificar rol.

3.3 Eliminar rol.

4. Requisitos Adicionales del Caso de Uso

No tiene.

5. Precondiciones

Que se haya solicitado gestionar rol.

Para crear y modificar rol debe haber al menos uno creado.

6. Poscondiciones

El administrador ha realizado los cambios en dependencia de las opciones seleccionadas.

7. Puntos de extensión

No tiene.

c. Especificación del Caso de Uso: CU_Gestionar Usuario

1. CU_Gestionar Usuario

Este caso de uso tiene como objetivo permitirle al Administrador definir los usuarios del sistema y su nivel de acceso.

2. Flujos de eventos

a. Flujo básico.

Caso de uso:	CU_Gestionar Usuario
Actores:	Administrador
Resumen:	El caso de uso comienza cuando el Administrador selecciona la opción Gestionar Usuarios. A partir de ahí el administrador tiene la opción de Crear, Modificar, o Eliminar un usuario. Finaliza el caso de uso cuando el Administrador termina la acción que ha seleccionado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el Administrador del sistema, selecciona la opción de gestionar usuario.	2. El sistema muestra todos los usuarios existentes y muestra las opciones: crear, modificar y eliminar un usuario.
3. El Administrador del sistema selecciona una de las opciones relacionadas con la actualización de los usuarios: a) Solicita crear un usuario. Ver Sección Crear Usuario. b) Solicita modificar un usuario seleccionado. Ver Sección	

<p>Modificar Usuario.</p> <p>c) Solicita eliminar un usuario seleccionado. Ver Sección Eliminar Usuario.</p>	
Sección Crear Usuario	
	<p>4. El sistema muestra el formulario para introducir los datos:</p> <p>a) Nombre.</p> <p>b) Usuario</p> <p>c) Contraseña.</p> <p>d) Confirmar Contraseña</p> <p>e) Lista de roles.</p> <p>f) Carnet de Identidad.</p>
<p>5. El Administrador del sistema introduce los datos del nuevo usuario. Le define el rol.</p>	
<p>6. El Administrador solicita guardar el nuevo usuario.</p>	<p>7. El sistema valida la entrada de los datos.</p>
	<p>8. El sistema valida que el usuario no exista.</p>
	<p>9. El sistema solicita confirmación. "¿Está seguro que desea guardar el nuevo usuario?".</p>
<p>10. El administrador acepta guardar los datos.</p>	<p>11. El sistema guarda los datos satisfactoriamente.</p>
	<p>12. El Sistema muestra un mensaje. "Se creó el usuario con éxito"</p>
<p>13. El Administrador acepta el</p>	<p>14. Se pasa a la Acción 2.</p>

mensaje.	
Sección Modificar Usuario.	
	15. El sistema muestra los datos del usuario seleccionado. No se permite modificar el dato Usuario. El dato "Contraseña" es opcional cambiarlo.
16. El Administrador modifica los datos del usuario seleccionado.	
17. El Administrador solicita guardar las modificaciones.	18. El sistema valida la entrada de datos.
	19. El sistema solicita confirmación de que desea guardar las modificaciones. "¿Está seguro que desea guardar los cambios realizados?".
20. El administrador acepta guardar las modificaciones.	21. El sistema guarda las modificaciones satisfactoriamente.
	22. El Sistema muestra un mensaje. "Se modificó el usuario con éxito"
23. El Administrador acepta el mensaje.	24. Se pasa a la Acción 2.
Sección Eliminar Usuario.	
	25. El sistema verifica que no haya una sesión activa con el usuario a eliminar.
	26. El sistema solicita confirmación de que se quiere eliminar el

	usuario. “¿Está seguro que desea eliminar el usuario?”.
27. El administrador acepta eliminar el usuario.	28. El sistema elimina el usuario seleccionado poniéndolo en estado deshabilitado. Se termina el caso de uso.
	29. El Sistema muestra un mensaje. “Se eliminó el usuario con éxito”
30. El Administrador acepta el mensaje.	31. Se pasa a la Acción 2.

b. Flujos alternativos

Acción 2: Si el sistema no puede mostrar los usuarios porque la conexión con la Base de Datos está deshabilitada.

Acción 3: Si el administrador selecciona cancelar la acción de gestionar usuario.

Acción 4: Si el sistema no puede cargar el listado de roles porque la conexión con la Base de Datos está deshabilitada.

Acción 5: Si el administrador tiene que crear el rol o modificarlo para asignárselo al nuevo usuario.

Acción 6: Si el administrador selecciona cancelar la acción de insertar.

Acción 7: Si no son válidos los valores de entrada.

Acción 8 F1: Si el usuario ya existe.

Acción 8 F2: Si no se puede verificar que el usuario existe por estar la conexión con la base de datos deshabilitada.

Acción 10: Si el administrador selecciona cancelar la solicitud de confirmación de insertar los datos del nuevo usuario.

Acción 11: Si el sistema no puede guardar los datos porque la conexión con la Base de Datos está deshabilitada.

Acción 15: Si el sistema no puede cargar el listado de roles porque la conexión con la Base de Datos está deshabilitada.

Acción 16: Si el administrador al modificar el rol tiene que crearlo o modificarlo para asignárselo al usuario.

Acción 17: Si el administrador selecciona cancelar la acción de modificar el usuario seleccionado.

Acción 18: Si no son válidos los valores de entrada.

Acción 20: Si el administrador selecciona cancelar la solicitud de confirmación de guardar los datos modificados.

Acción 21: Si el sistema no puede guardar los datos porque la conexión con la Base de Datos está deshabilitada.

Acción 25: Si existe una sesión activa con el usuario a eliminar.

Acción 27: Si el administrador selecciona cancelar la confirmación de eliminar usuario.

Acción 28: Si no se puede eliminar el usuario porque la conexión con la Base de Datos está deshabilitada.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Acción 2	2.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	2.2 El Sistema muestra el mensaje. "El siguiente error ha ocurrido: " y se detalla el error.
2.3 El Administrador acepta el	2.4 Se termina el Caso de Uso.

mensaje.	
Acción 3	3.1 Se termina el Caso de Uso.
Acción 4	4.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	4.2 El sistema muestra el mensaje:"El siguiente error ha ocurrido: " y se detalla el error.
4.3 El Administrador acepta el mensaje.	4.4 Se pasa a la acción 2.
Acción 5	5.1 Ver CU _ Gestionar Roles. Va a la Acción 4.
Acción 6	6.1 Se pasa a la Acción 2.
Acción 7	7.1 El Sistema muestra el mensaje: "Entrada de datos no válida".
	7.2 El Administrador acepta el mensaje.
7.3 El Sistema muestra un asterisco al lado de cada entrada no válida.	7.4 Se pasa a la Acción 4.
Acción 8 F1.	8 F1.1 El sistema muestra el mensaje "Ya existe un usuario con ese nombre ó carnet de identidad"
8 F1.2 El administrador acepta el mensaje.	8 F1.3 Se pasa a la acción 4
Acción 8 F2.	8 F2.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.

	8 F2.2 El sistema muestra el mensaje "El siguiente error ha ocurrido: " y se detalla el error.
8 F2.3 El administrador acepta el mensaje.	8 F2.4 Se pasa a la acción 4
Acción 10	10.1 Se pasa a la Acción 4.
Acción 11	11.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	11.2 El Sistema muestra el mensaje:"El siguiente error ha ocurrido: " y se detalla el error.
11.3 El Administrador acepta el mensaje.	11.4 Se pasa a la Acción 4.
Acción 15	15.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	15.2 El Sistema muestra el mensaje. "El siguiente error ha ocurrido: " y se detalla el error.
15.3 El Administrador acepta el mensaje.	15.4 Se pasa a la Acción 2
Acción 16	16.1 Ver CU _ Gestionar Roles. Se pasa a la acción 15
Acción 17	17.1 Se pasa a la Acción 2
Acción 18	18.1 El Sistema muestra el mensaje "Entrada de datos no válida".

18.2 El Administrador acepta el mensaje.	18.3 El Sistema muestra un asterisco al lado de cada entrada no válida.
	18.4 Se pasa a la Acción 14
Acción 20	20.1 Se pasa a la Acción 14
Acción 21	21.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	21.2 El Sistema muestra el mensaje: "El siguiente error ha ocurrido: " y se detalla el error.
21.3 El Administrador acepta el mensaje.	21.4 Se pasa a la Acción 14
Acción 25	25.1 El sistema muestra el mensaje "No se puede eliminar un usuario en sección activa"
25.2 El administrador acepta el mensaje	25.3 Se pasa a la acción 2.
Acción 27	27.1 Se pasa a la Acción 2
Acción 28	28.1 El Sistema guarda la notificación del resultado de la acción en el fichero de notificaciones.
	28.2 El Sistema muestra el mensaje: "El siguiente error ha ocurrido: " y se detalla el error.
28.3 El Administrador acepta el mensaje.	28.4 Se pasa a la Acción 2

3. Requisitos

1. Gestionar información de los Usuarios.

1.1 Insertar nuevo Usuario.

1.2 Modificar Usuario.

1.3 Eliminar usuario.

4. Requisitos Adicionales del Caso de Uso

No tiene.

5. Precondiciones

Que se haya solicitado gestionar usuario.

Para crear y modificar usuario debe haber al menos uno creado.

6. Poscondiciones

El administrador ha realizado los cambios en dependencia de las opciones seleccionadas.

7. Puntos de extensión

CU _ Gestionar Roles. Extendido.

2.2 Conclusiones

En este capítulo se abordó todo lo relacionado con el objeto de estudio, un modelo de dominio que recoge los conceptos y eventos más comunes en el contexto del Negocio para lograr una mejor comprensión del mismo, también se realizó la captura de los requerimientos funcionales y no funcionales del sistema, el diagrama de casos de uso del sistema y la especificación detallada de los casos de uso.

Capítulo 3: Análisis y Diseño del Sistema

Introducción

En este capítulo se muestran los resultados obtenidos en el desarrollo del flujo de trabajo Análisis y Diseño del Sistema, utilizando para su modelado el Lenguaje Unificado de Modelado (UML), donde se elaboraran los diagramas de clases del Análisis y del Diseño, así como los diagramas de interacción que fueron necesarios utilizar para obtener una mayor claridad a la hora de elaborar la solución que se propone. En este capítulo también se realiza una breve descripción de las clases con sus respectivas responsabilidades. Además se da una breve descripción de la concepción general de la ayuda del sistema del proyecto TeleBanca.

3.1 Análisis

Durante este flujo de trabajo, se analizan los requisitos funcionales que se describen en el capítulo anterior, refinándolos y estructurándolos con el objetivo de conseguir una comprensión más precisa y una descripción más detallada de los casos de uso, que ayudan a la estructuración del sistema. Además los resultados de esta etapa se utilizan para comprender como debería ser diseñado e implementado el sistema ya que esbozan como llevar a cabo las funcionalidades dentro del mismo y sirve como una primera aproximación al diseño.

3.1.1 Diagrama de Clases del Análisis

Durante el análisis es importante coordinar todas las clases que están presentes en los diferentes casos de uso, por esta razón se construye el diagrama de clases del Análisis, que muestra las clases participantes y las relaciones entre ellas. Ver Anexo 1.

3.2 Diseño

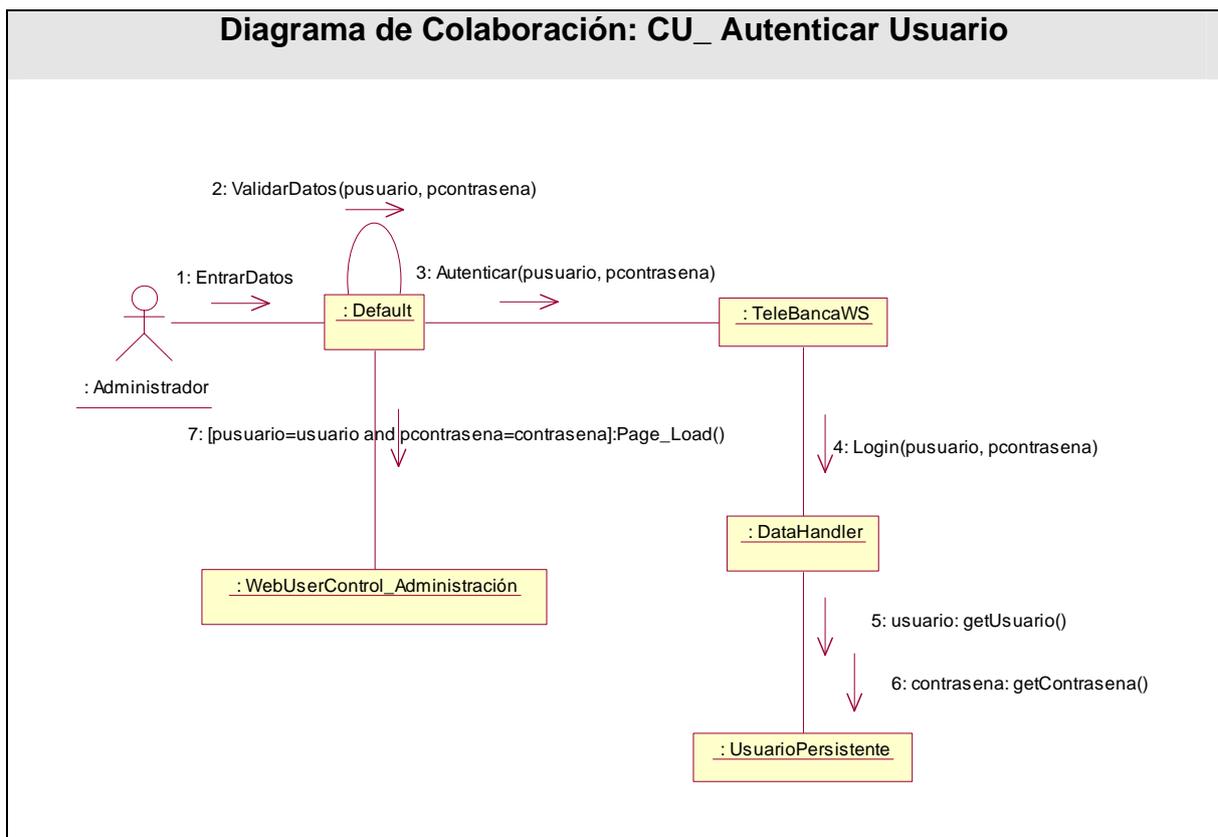
Durante este flujo de trabajo se modela el sistema incluyendo su arquitectura, para que soporte todos los requisitos ya sean funcionales o no funcionales. Una entrada fundamental de esta etapa es el resultado del análisis. La esencia de este flujo es la elaboración de los diagramas de interacción y el diagrama de clases del diseño.

3.2.1 Diagramas de interacción del sistema

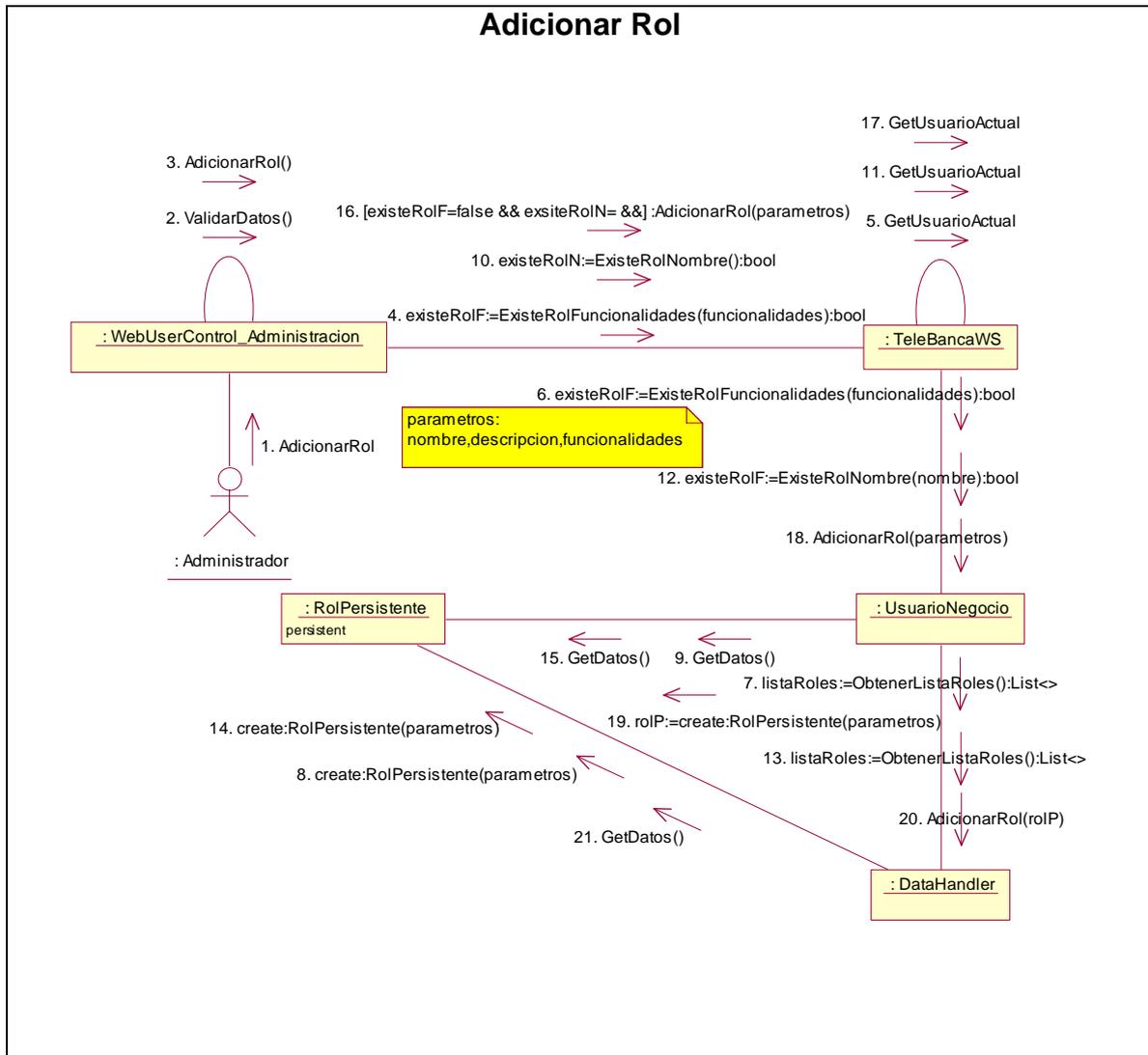
Los diagramas de interacción del sistema muestran gráficamente como lo objetos se comunican entre ellos a través de mensajes para realizar las tareas y con el fin de cumplir con los requerimientos, los mismos se pueden expresar en diagramas de secuencia o en diagramas de colaboración. Los primeros detallan las secuencias de interacciones de forma ordenada en el tiempo, los segundos son los que se utilizan en el presente trabajo, que describen las interacciones entre los objetos en un formato de grafo, siendo esto el centro de la atención en esta etapa.

A continuación se muestra la representación gráfica de los diagramas de colaboración más relevantes: Autenticar Usuario, Gestionar Usuario, Gestionar Roles. El resto de los diagramas se encuentran en el Anexo 3.

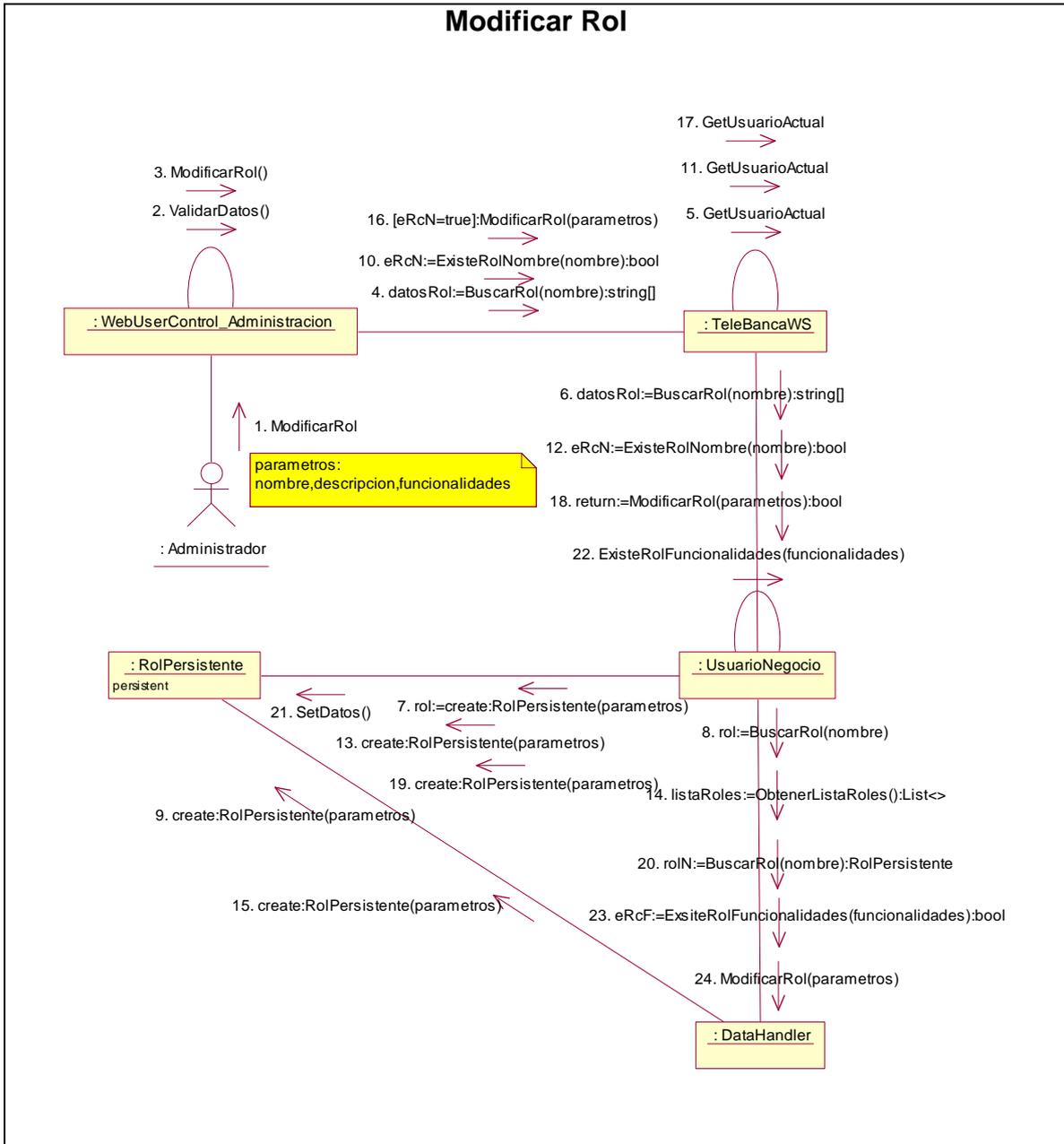
Diagramas de Colaboración CU_Autenticar Usuario



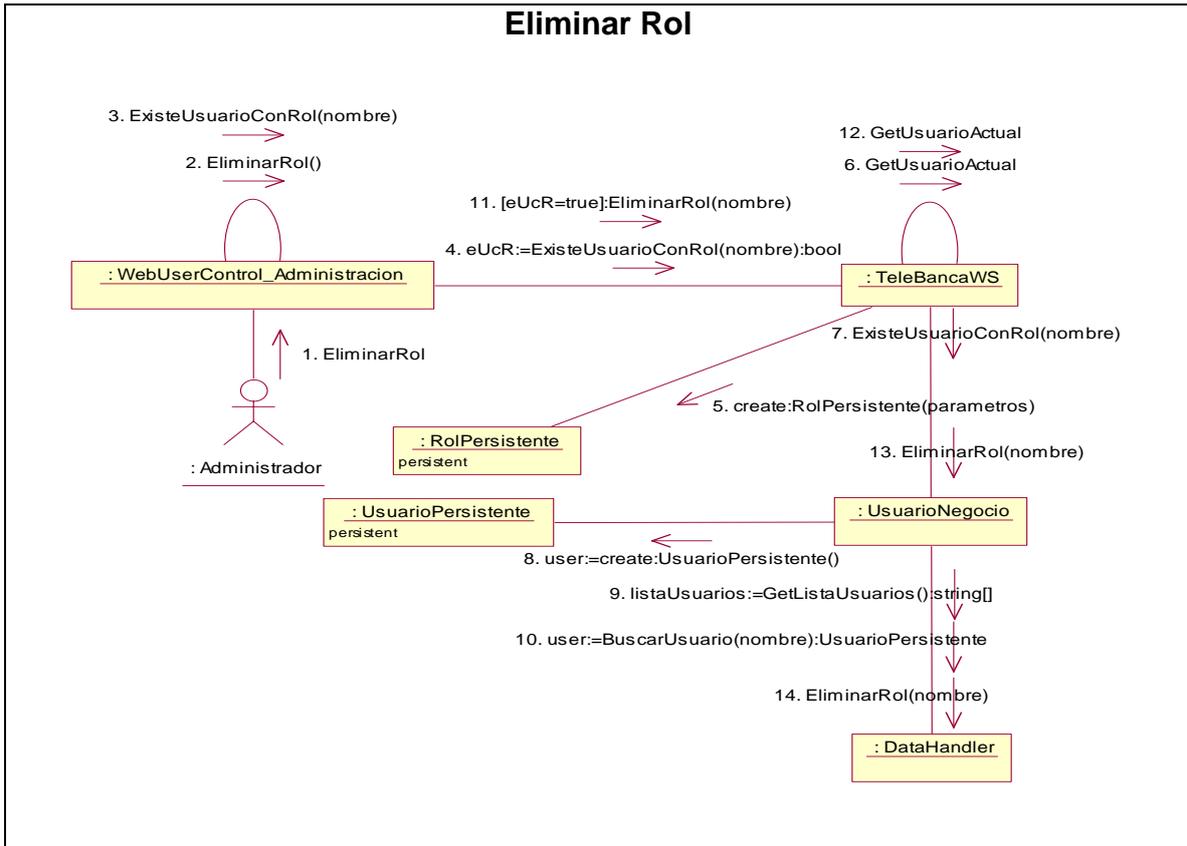
Diagramas de Colaboración CU_Gestionar Roles



Modificar Rol

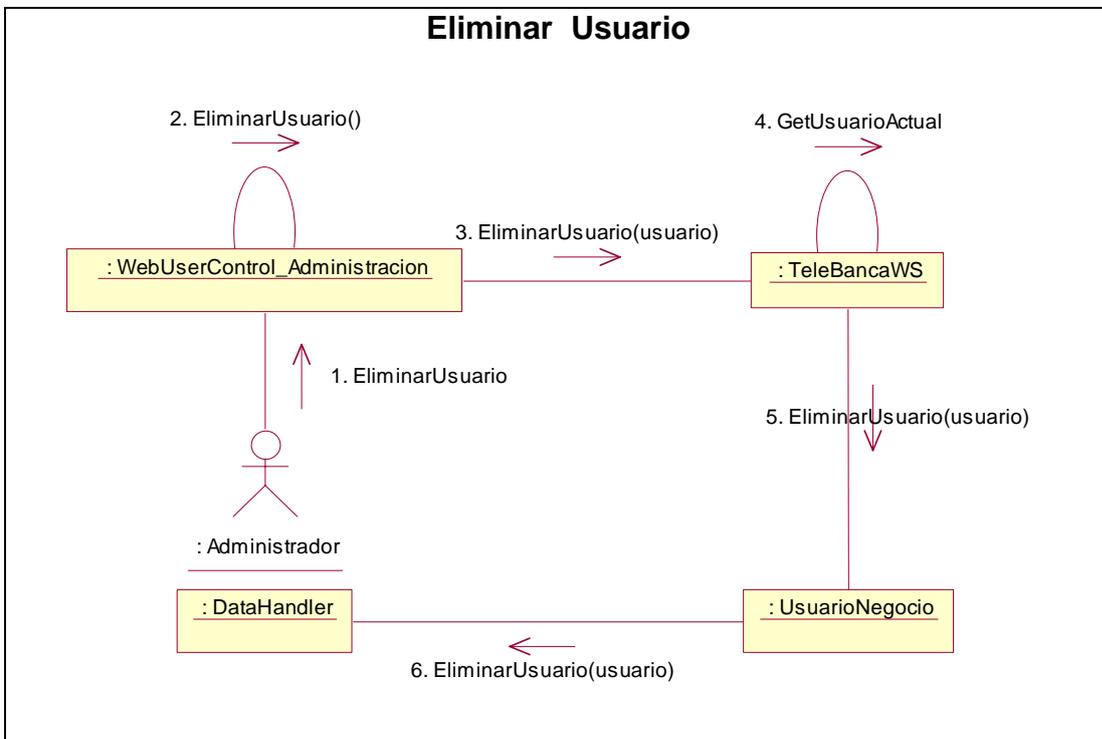


Eliminar Rol

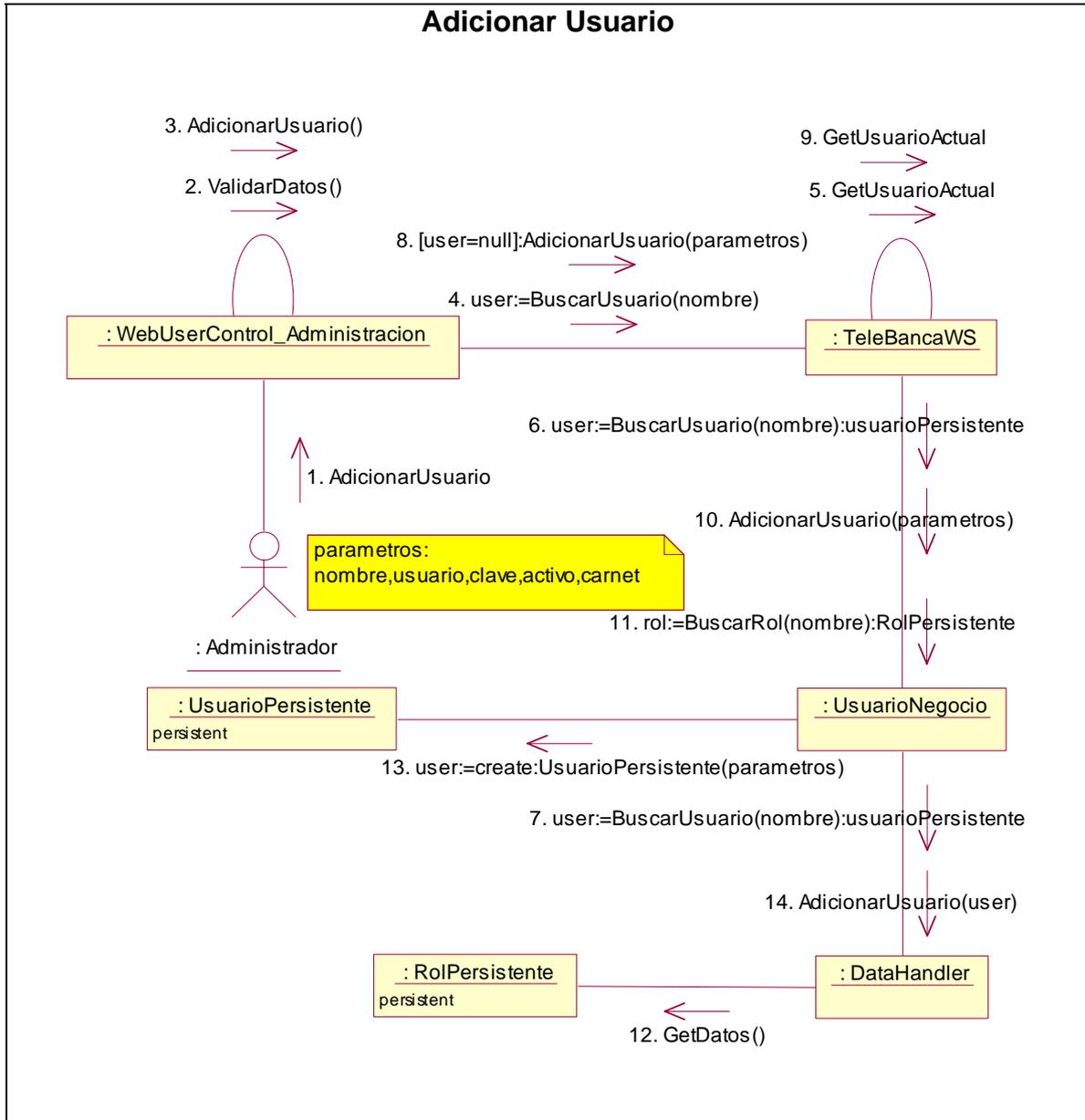


Diagramas de Colaboración CU_Gestionar Usuarios

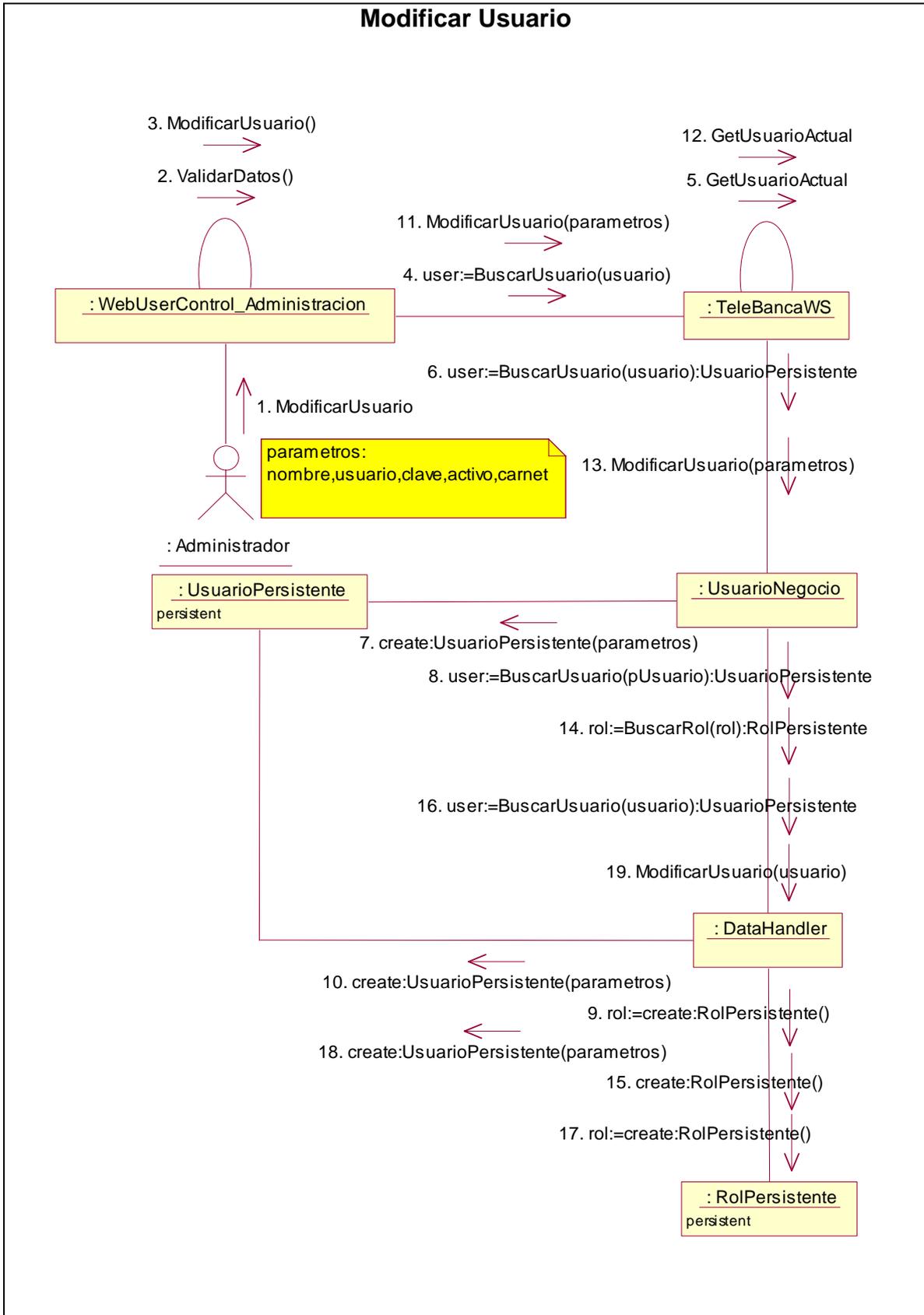
Eliminar Usuario



Adicionar Usuario



Modificar Usuario



3.2.2 Diagrama de clases del Diseño

Este diagrama concluye la parte de asignación de responsabilidades obtenidas con los diagramas de interacción. UML es un lenguaje a utilizar para modelar aplicaciones Web, ya que este posee extensiones para este tipo de aplicación como son: estereotipos, valores etiquetados y restricciones, que permiten modelar aplicaciones con este tipo de arquitectura.

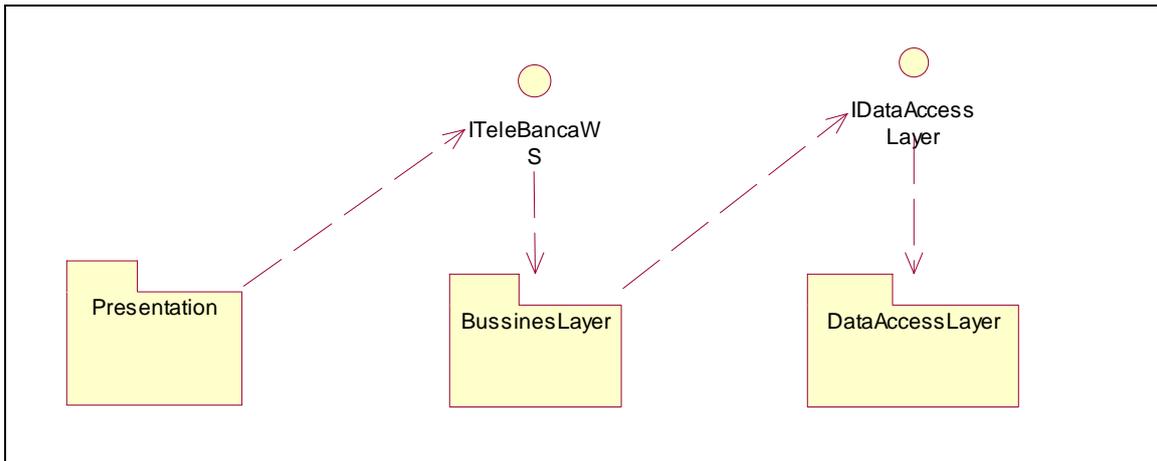


Figura 3.1. Diagrama de clases general del Módulo de Administración

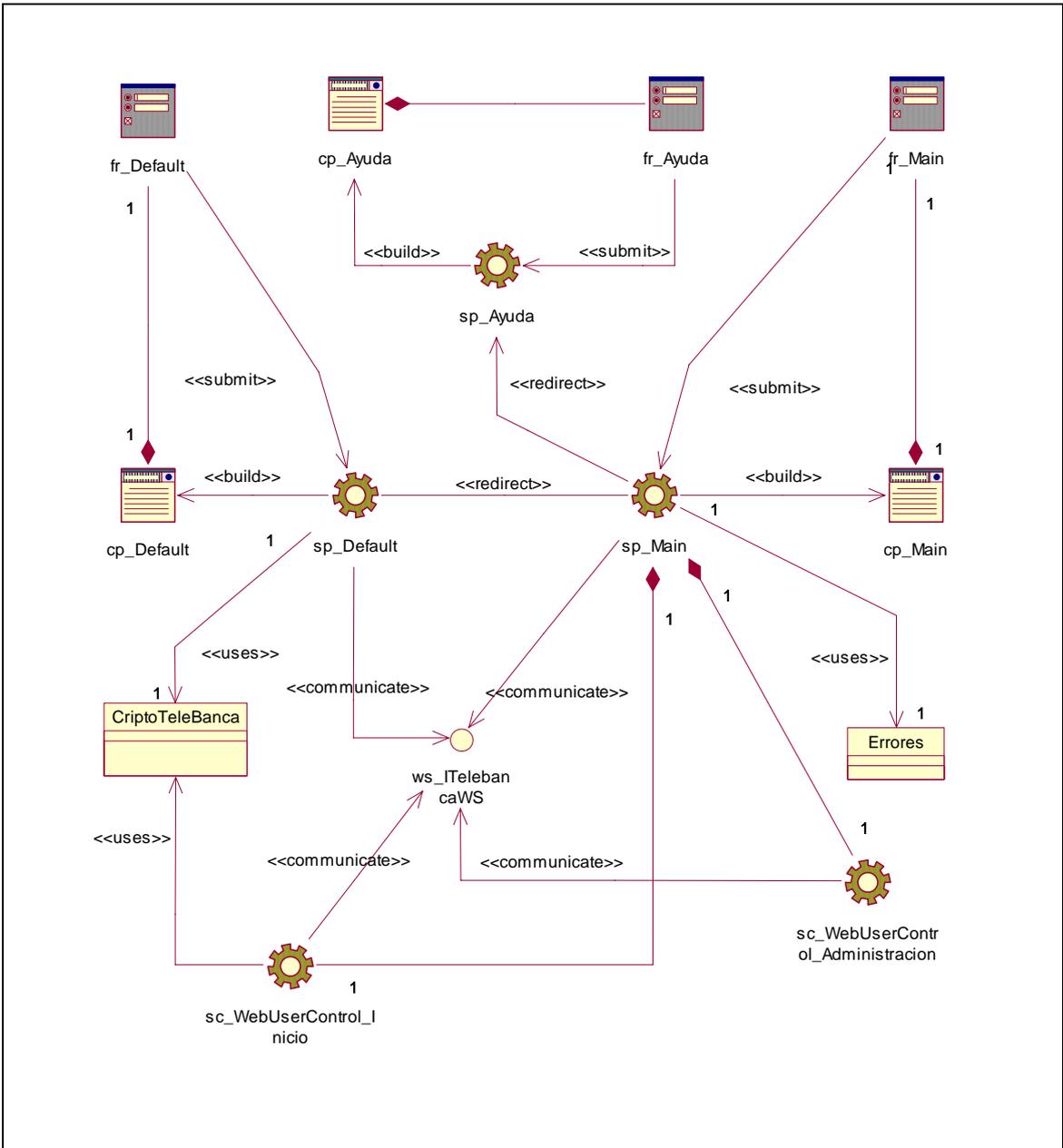


Figura 3.2. Diagrama de clases del paquete de presentación

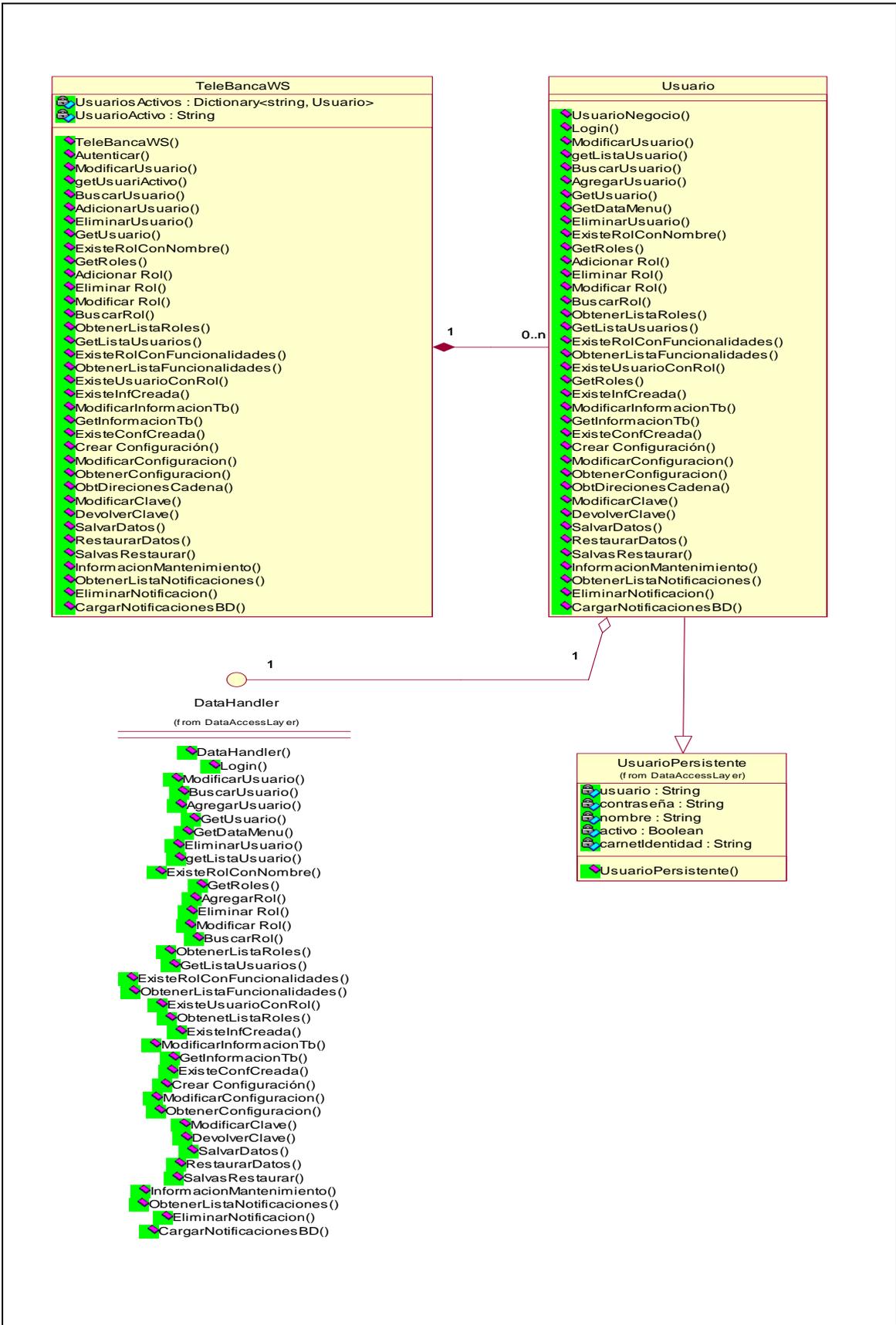


Figura 3.3. Diagrama de clases del paquete de Negocio

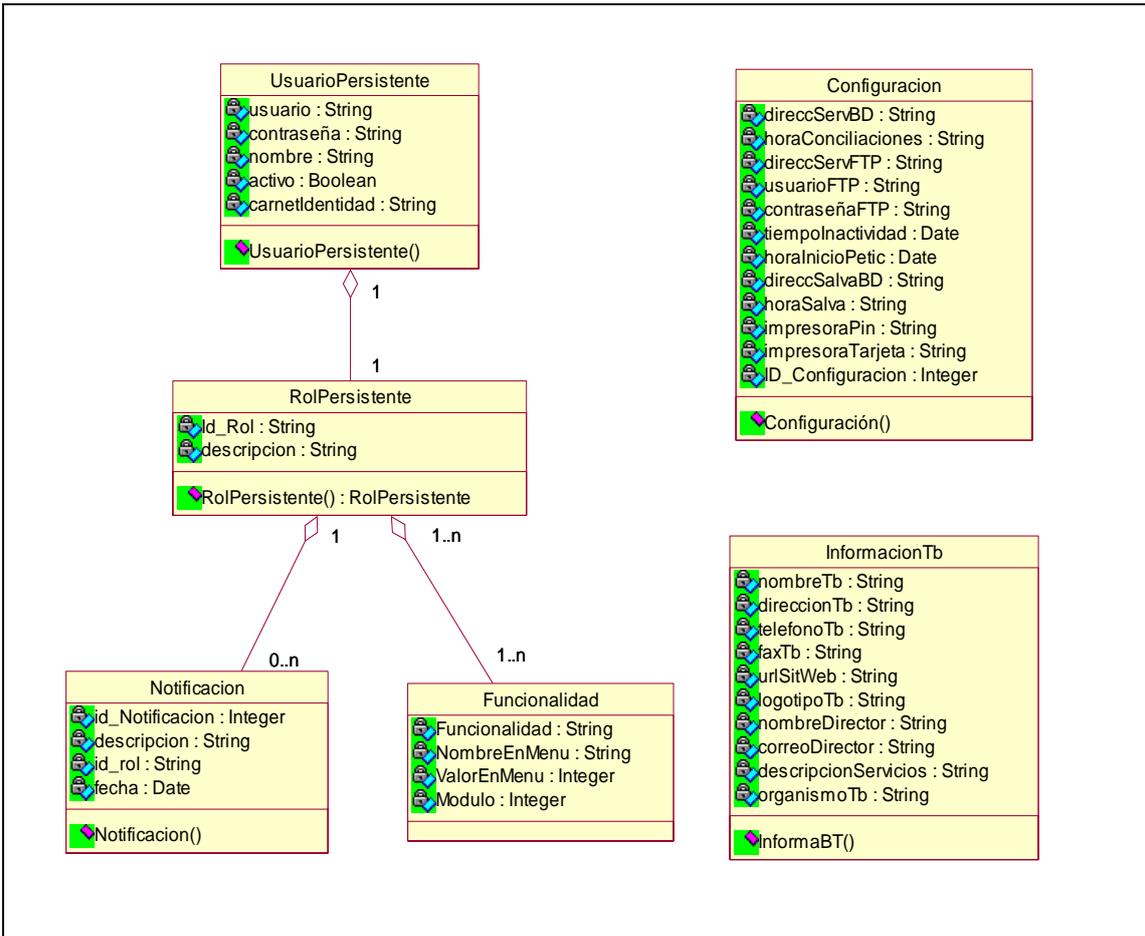


Figura 3.4. Diagrama de clases del paquete de Acceso a Datos

3.2.3 Descripción de las clases

Nombre: TeleBancaWS	
Tipo de clase: Controladora	
Atributo	Tipo
UsuariosActivos	Dictionary<string, Usuario>
UsuarioActivo	string
Responsabilidades:	
Nombre:	Autenticar(string pusuario, string pcontrasena)

Descripción:	Se encarga de invocar al método Login de la clase DataHandler
Nombre:	ModificarUsuario(string nombre, string clave, string rol, string usuario, bool activo, string carnetIdentidad)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	GetUsuarioActual()
Descripción:	Se encarga de obtener el usuario actual.
Nombre:	EliminarUsuario(string usuario)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase usuario si el usuario que se desea eliminar no está activo
Nombre:	BuscarUsuario(string usuario)
Descripción:	Se encarga de obtener los datos de un usuario determinado invocando al método de mismo nombre en la clase Usuario
Nombre:	AdicionarUsuario(string nombre, string clave, string rol, string usuario, bool activo, string carnetIdentidad)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	GetUsuario(string usuario, string clave)
Descripción:	Se encarga de obtener los datos de un determinado usuario
Nombre:	GetListaUsuarios()
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	ExisteRolConNombre(string nombre)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	ExisteRolConFuncionalidades(string funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario

Nombre:	ExisteRol(string nombre,string descripcion,string funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	GetRoles()
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	AdicionarRol(string nombre, string descripcion, string[] funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	EliminarRol(string idRol)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	ModificarRol(string nombre, string descripción, string[] funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	BuscarRol(string idRol)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtenerListaRoles()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtenerListaFuncionalidades()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ExisteUsuarioConRol(string idRol)
Descripción:	Se encarga de verificar si existe un usuario con un rol determinado

Nombre:	ExistelnfCreada()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ModificarInformacionTb(string nombreTb, string direccionTb, string telefonoTb, string faxTb, string urlSitioWeb, string logoTipo, string nombreDirector, string correoElectronico, string organismoTb, string descripcionServicios)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	GetInformacionTb()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ExisteConfCreada()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ModificarConfiguracion(string direccionServidorBd, DateTime horaConciliaciones, string direccionServidorFtp, string UsuarioFtp, string ContraseñaFtp, DateTime TiempoInactividad, string direccionSalvaBd, DateTime horaInicioPeticones, DateTime horaSalva, string imprPin, string imprTarj)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtenerConfiguracion()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtDireccionesCadena(string aCadena)
Descripción:	Dada una cadena te devuelve de la misma todas las subcadenas que están entre (;)
Nombre:	ModificarClave(string usuario, string nuevacontrasenna)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario

Nombre:	DevolverClave(string usuario)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	SalvarDatos()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	RestaurarDatos(string dir)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	SalvasRestaurar()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	InformacionMantenimiento(string tipo)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtenerListaNotificaciones()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	EliminarNotificacion(int id_Notificacion)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	CargarNotificacionesBD()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario

Nombre: Usuario	
Tipo de clase: Controladora	
Atributo	Tipo
Handler	DataAccessLayer.DataHandler

operaciones	COperacionAutomatica
Responsabilidades:	
Nombre:	ModificarUsuario(string nombre, string clave, string rol, string usuario, bool activo, string carnetIdentidad)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	getUsuariActivo()
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	EliminarUsuario(string usuario)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase usuario si el usuario que se desea eliminar no está activo
Nombre:	BuscarUsuario(string usuario)
Descripción:	Se encarga de obtener los datos de un usuario determinado invocando al método de mismo nombre en la clase Usuario
Nombre:	AdicionarUsuario(string nombre, string clave,string rol,string usuario,bool activo, string carnetIdentidad)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	GetUsuario(string usuario,string clave)
Descripción:	Se encarga de obtener los datos de un determinado usuario
Nombre:	GetListaUsuarios()
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	ExisteRolConNombre(string nonmbre)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	ExisteRolConFuncionalidades(string[] funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase

	Usuario
Nombre:	ExisteRol(string nombre,string descripcion,string[] funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	GetRoles()
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	AdicionarRol(string nombre, string descripcion, string []funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	EliminarRol(string idRol)
Descripción:	Se encarga de invocar al método de mismo nombre en la clase Usuario
Nombre:	ModificarRol(string nombre, string descripción, string[] funcionalidades)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	BuscarRol(string idRol)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtenerListaRoles()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtenerListaFuncionalidades()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ExisteUsuarioConRol(string idRol)
Descripción:	Se encarga de verificar si existe un usuario con un rol

	determinado
Nombre:	ExisteInfCreada()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase DataHandler.
Nombre:	ModificarInformacionTb(string nombreTb, string direccionTb, string telefonoTb, string faxTb, string urlSitioWeb, string logoTipo, string nombreDirector, string correoElectronico, string organismoTb, string descripcionServicios)
Descripción:	Crea un objeto de tipo InformacionTb a partir de los datos, pasados por parámetros e invoca al método de mismo nombre de la clase DataHandler.
Nombre:	GetInformacionTb()
Descripción:	Se encarga de crear y retornar un arreglo con todos los atributos de InformacionTb.
Nombre:	ExisteConfCreada()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase DataHandler.
Nombre:	ModificarConfiguracion(string direccionServidorBd, DateTime horaConciliaciones, string direccionServidorFtp, string UsuarioFtp, string ContraseñaFtp, DateTime TiempoInactividad, string direccionSalvaBd, DateTime horaInicioPeticones, DateTime horaSalva, string imprPin, string imprTarj)
Descripción:	Crea un objeto configuración a partir de los datos, pasados por parámetros e invoca al método de mismo nombre de la clase DataHandler.
Nombre:	ObtenerConfiguracion()
Descripción:	Se encarga de crear y retornar un arreglo con todos los atributos de la configuración.
Nombre:	ModificarClave(string usuario, string nuevacontrasenna)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	DevolverClave(string usuario)

Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	SalvarDatos()
Descripción:	Se encarga de obtener la dirección donde se van a almacenar las salvas de la tabla configuración, he invoca al método HacerBackup de la clase DataHandler pasándole como parámetro dicha dirección.
Nombre:	RestaurarDatos(string dir)
Descripción:	Se encarga de invocar al método RestaurarBaseDato de la clase DataHandler.
Nombre:	SalvasRestaurar()
Descripción:	Se encarga de devolverte un arreglo con todas las salvas que se encuentra en la dirección de salvas de la configuración del sistema.
Nombre:	InformacionMantenimiento(string tipo)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase DataHandler
Nombre:	ObtenerListaNotificaciones()
Descripción:	Retorna una matriz con todos los datos de las notificaciones que ha generado el sistema.
Nombre:	EliminarNotificacion(int id_Notificacion)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase DataHandler
Nombre:	CargarNotificacionesBD()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase DataHandler.

Nombre: DataHandler	
Tipo de clase: Controladora	
Atributo	Tipo
cString	string
MyCurrent	HttpContext
Responsabilidades:	
Nombre:	Login(string usuario, string pass)
Descripción:	Dado el usuario y la contraseña verifica si existe un usuario con esa contraseña en el sistema
Nombre:	BuscarUsuario(string pUsuario)
Descripción:	Se encarga de devolver los datos de un usuario determinado
Nombre:	EliminarUsuario(string pUsuario)
Descripción:	Se encarga de eliminar un usuario determinado
Nombre:	AgregarUsuario(UsuarioPersistente usuarioP)
Descripción:	Se encarga de adicionar un usuario determinado
Nombre:	ModificarUsuario(UsuarioPersistente usuarioP)
Descripción:	Se encarga de modificar un usuario determinado
Nombre:	GetListaUsuarios()
Descripción:	Se encarga de obtener la lista de todos los usuarios
Nombre:	ExisteRol(string rolP)
Descripción:	Se encarga de verificar si existe un rol con un nombre determinado
Nombre:	ExisteUsuarioConRol(string rolP)
Descripción:	Se encarga de verificar si existe un usuario con un rol determinado
Nombre:	EliminarRol(string rolP)
Descripción:	Se encarga de eliminar un rol determinado

Nombre:	AgregarRol(RolPersistente rolP)
Descripción:	Se encarga de adicionar un rol determinado
Nombre:	ModificarRol(RolPersistente rolP)
Descripción:	Se encarga de modificar un rol determinado
Nombre:	BuscarRol(string nombreRol)
Descripción:	Se encarga de obtener y devolver los datos de un rol determinado
Nombre:	ObtenerListaRoles()
Descripción:	Se encarga de obtener la lista de todos los roles
Nombre:	ObtenerListaFuncionalidades()
Descripción:	Se encarga de obtener la lista de todas las funcionalidades
Nombre:	ExisteInfCreada()
Descripción:	Verifica si ya esta creada la información de TeleBanca en la BD
Nombre:	ModificarInformacionTb(InformacionTb informacionTb)
Descripción:	Dado un objeto de tipo InformacionTB modifica la información TB contenida en la BD
Nombre:	GetInformacionTb()
Descripción:	Crea y retorna un objeto de tipo informaciónTB a partir de la información que hay en la BD.
Nombre:	ExisteConfCreada()
Descripción:	Verifica si ya esta creada la configuracion de TeleBanca en la BD
Nombre:	ModificarConfiguracion(Configuracion Cconf)
Descripción:	Dado un objeto de tipo Configuracion modifica la Configuración existente en la BD
Nombre:	GetConfiguracion()
Descripción:	Devuelve los datos de la configuración existente en la BD.
Nombre:	ModificarClave(string usuario, string nuevacontrasenna)

Descripción:	Dado el usuario y la nueva contraseña busca a dicho usuario en la base de datos y modifica su clave.
Nombre:	DevolverClave(string usuario)
Descripción:	Retorna la clave de dicho usuario existente en la BD.
Nombre:	SalvarDatos()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	RestaurarDatos(string dir)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	SalvasRestaurar()
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	InformacionMantenimiento(string tipo)
Descripción:	Se encarga de invocar al método de mismo nombre de la clase Usuario
Nombre:	ObtenerListaNotificaciones()
Descripción:	Devuelve una lista con todos los datos de las notificaciones existentes en la BD.
Nombre:	EliminarNotificacion(int idNotificacion)
Descripción:	Elimina todos los datos de la notificación en la BD que coincide con el ID de notificación pasado por parámetros.
Nombre:	CargarNotificacionesBD()
Descripción:	Carga todas las notificaciones del fichero de notificaciones para la BD y elimina dichas notificaciones del fichero.

Nombre: Configuracion	
Tipo de clase : Entidad	
Atributo	Tipo
direccServBD	string
horaConciliaciones	DateTime
direccServFTP	string
usuarioFTP	string
contraseñaFTP	string
tiempolnactividad	DateTime
horaInicioPetic	DateTime
direccSalvaBD	string
horaSalva	DateTime
impresoraPin	string
impresoraTarjeta	string
Responsabilidades	
Nombre:	Configuracion()
Descripción:	Constructor por defecto de la clase
Nombre:	Configuracion(string direccServBD, DateTime horaConciliaciones, string direccServFTP, string usuarioFTP, string contraseñaFTP, DateTime tiempolnactividad, string direccSalvaBD, DateTime horaInicioPetic, DateTime horaSalva, string imprPin, string imprTarj)
Descripción:	Constructor de la clase.

Nombre: InformacionTb	
Tipo de clase : Entidad	
Atributo	Tipo
nombreTb	string
direccionTb	string
telefonoTb	string
faxTb	string
urlSitWeb	string
logotipoTb	string
nombreDirector	string
correoDirector	string
descripcionServicios	string
organismoTb	string
Responsabilidades	
Nombre:	InformacionTb(string NombreTb, string DireccionTb, string TelefonoTb,string FaxTb,string UrlSitioWeb,string LogotipoTb,string NombreDirector, string CorreoElectronico,string DescripcionServicios,string OrganismoTb)
Descripción:	Constructor de la clase.

Nombre: Notificacion	
Tipo de clase : Entidad	
Atributo	Tipo
id_Notificacion	int
descripcion	string
id_rol	string

fecha	DateTime
Responsabilidades	
Nombre:	Notificacion(string descripcion, string id_rol)
Descripción:	Constructor de la clase

Nombre: RolPersistente	
Tipo de clase : Entidad	
Atributo	Tipo
funcionalidades	List<string>
nombreMenu	List<string>
valorMenu	List<int>
modulo	List<int>
nombre	string
descripcion	string
Responsabilidades	
Nombre:	RolPersistente()
Descripción:	Se encarga de crear un objeto de tipo RolPersistente

Nombre: UsuarioPersistente	
Tipo de clase : Entidad	
Atributo	Tipo
usuario	string
nombre	string
contrasena	string
activo	bool

carnetIdentidad	string
rol	RolPersistente
Responsabilidades	
Nombre:	UsuarioPersistente(string Usuario, string Contraseña, string Nombre, RolPersistente Rol, bool Activo, string CarnetIdentidad)
Descripción:	Se encarga de crear un objeto del tipo UsuarioPersistente

3.2.4 Concepción general de la ayuda del sistema del proyecto TeleBanca

Como el nombre del tema lo indica la ayuda que se propone no es sólo del Módulo de Administración, sino del sistema del proyecto TeleBanca, en la misma se explicará de una forma detallada y sencilla el uso de cada una de las funcionalidades que dicho sistema brindará. Una entrada esencial para la construcción de la Ayuda son los correspondientes manuales de usuario de cada una de las partes del sistema, ya que estos explican a los usuarios finales cómo funciona el sistema con todos sus detalles. Se escogió como herramienta de desarrollo de la ayuda a RoboHelp Office X5, por las grandes ventajas e integración con aplicaciones Web que esta herramienta ofrece, las mismas fueron explicadas en el Capítulo 1.

3.3 Conclusiones

Se finaliza la etapa de Análisis del Sistema, habiéndose estructurado el diagrama de clases del Análisis. También se concluye así con la etapa de Diseño del Sistema, graficándose los diagramas de interacción mediante diagramas de colaboración, de modo tal que se desarrolló un diagrama de interacción para cada una de las operaciones del sistema. Con los resultados obtenidos, se creó el diagrama de diseño Web del sistema, la descripción de las clases. Con todos estos elementos, se tiene una idea más precisa sobre los elementos constructivos del sistema que se propone.

Capítulo 4: Implementación y Prueba

Introducción

Este capítulo es el último del presente trabajo, el cual muestra la construcción de los artefactos esenciales de la fase de Implementación y Prueba. De la fase de Implementación se muestra el diagrama de despliegue y el diagrama de componente del Módulo de Administración, construyéndose también de la fase de Prueba el modelo de prueba con sus respectivos casos de prueba.

4.1 Implementación

Durante la Implementación se empieza con el resultado del Diseño y se implementa el sistema en términos de componentes. Afortunadamente, la mayor parte de la arquitectura del sistema es capturada durante el Diseño, siendo el propósito principal de la Implementación el desarrollar la arquitectura y el sistema como un todo.

4.1.1 Diagrama de despliegue

El diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación, que muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

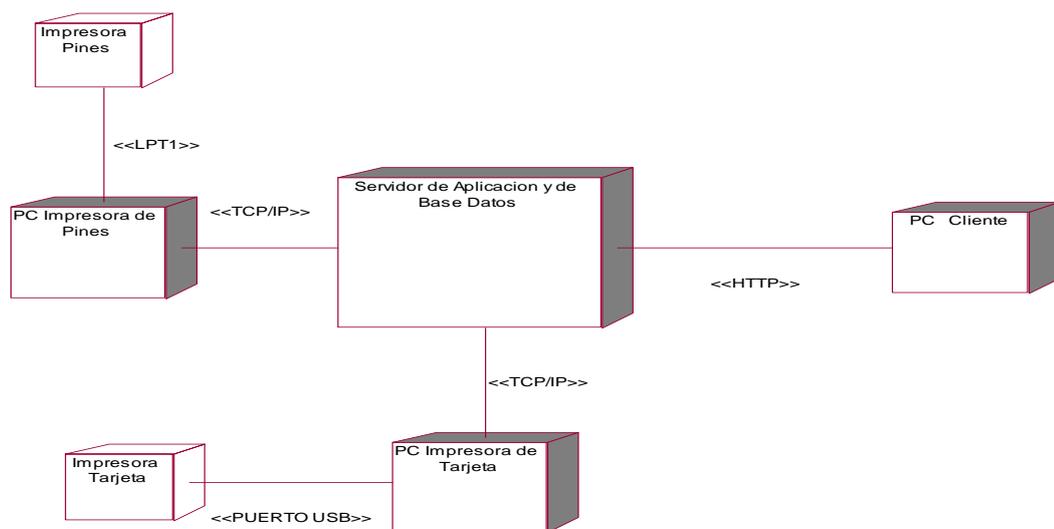


Figura 4.1. Diagrama de despliegue

4.1.2 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc.

Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente. Estos cubren la vista de la implementación estática y se relacionan con los diagramas de clases ya que en un componente puede existir una o más clases, interfaces o colaboraciones. Un componente es una parte física y reemplazable de un sistema que conforma un conjunto de interfaces y proporciona la implementación de dicho conjunto.

A continuación se mostrará una representación general de las relaciones que existen entre los componentes agrupados en paquetes de presentación y negocio, que representan capas del Módulo de Administración de TeleBanca.

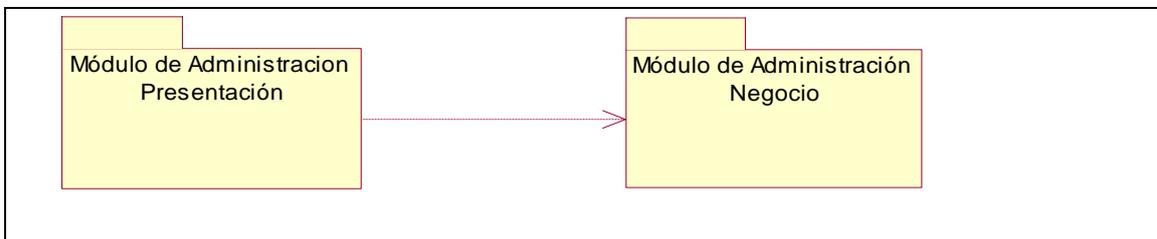


Figura 4.2. Diagrama general de componentes del Módulo de Administración

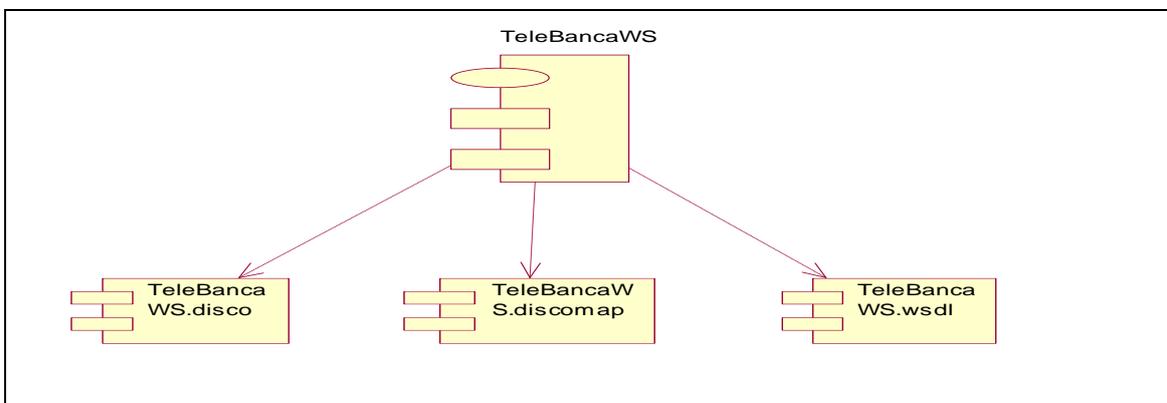


Figura 4.3. Representación del paquete TeleBancaWS y la relación con sus componentes

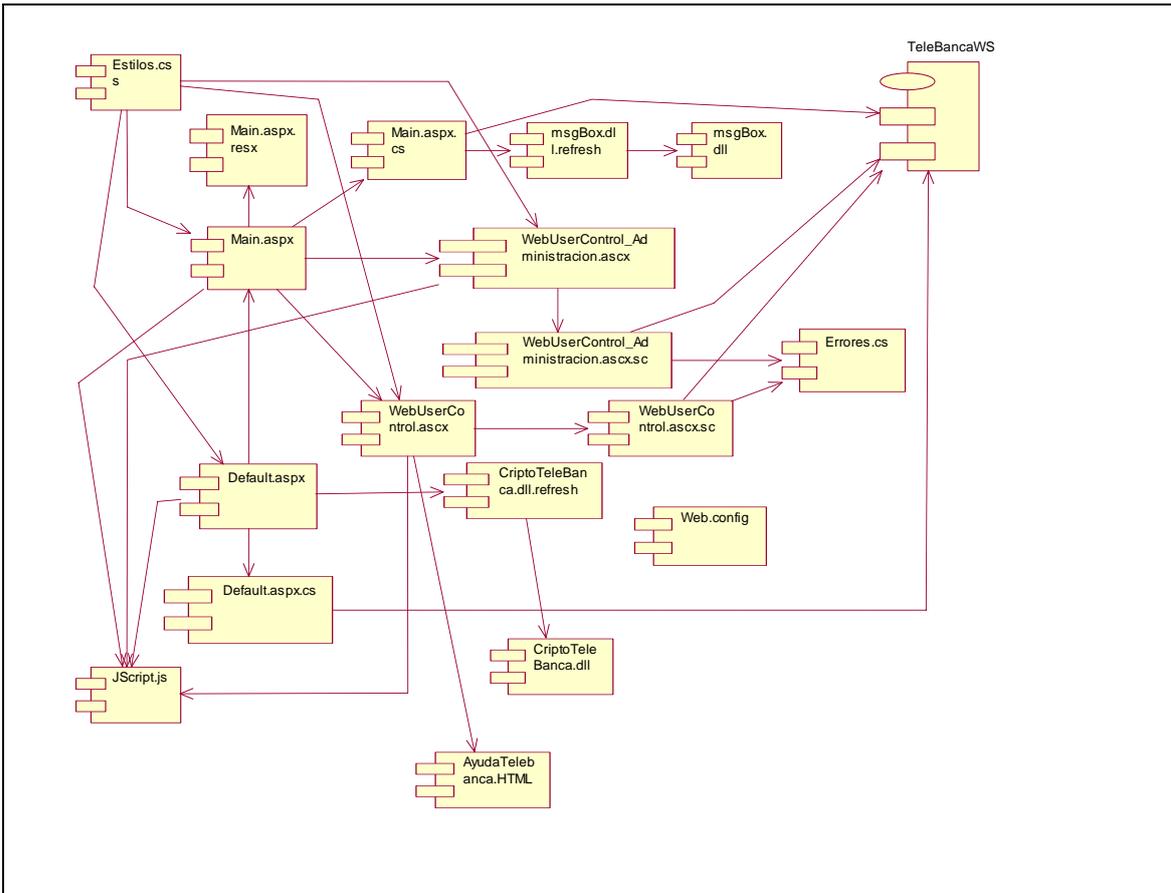


Figura 4.4. Representación de las relaciones entre componentes del paquete de presentación.

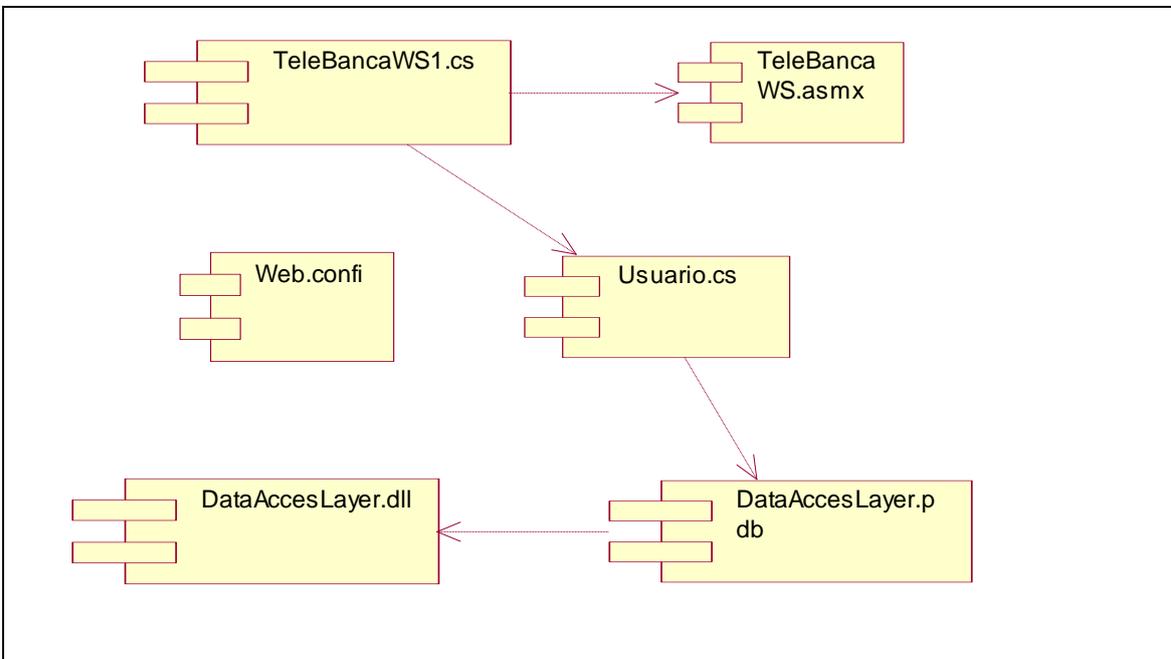


Figura 4.5. Representación de las relaciones entre componentes del paquete de negocio.

4.2 Prueba

Durante el flujo de trabajo Prueba se verifica el resultado de la implementación, probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros.[JACOBSON et al. , 2000]

4.2.1 Modelo de prueba

El objetivo central de la fase de pruebas es ejecutar el software para encontrar los defectos en el mismo, es por esta razón que se plantea que las pruebas tienen éxito si descubren por lo menos un defecto y las mismas fracasan si hay defectos que no son encontrados.

Una entrada esencial en el diseño de casos de prueba es las especificación de los requerimientos de software, que deben tener una explicación clara, precisa y completa del problema, para que facilite el análisis de los errores y la generación de casos de prueba, los cuales deben ser escritos tanto para condiciones de entradas inválidas o inesperadas, como para condiciones válidas y esperadas, como se muestran en los siguientes casos de prueba del presente trabajo, es decir del modulo de administración del proyecto TeleBanca.

CU_ Autenticar Usuario

Nombre del caso de uso: CU_ Autenticar Usuario		
Entrada	Resultados	Condiciones
El usuario introduce sus datos y solicita autenticarse. Usuario: "administrador" Contraseña: "123"	Muestra la pagina principal de la aplicación.	Un usuario solicita entrar al Sistema.

Nombre del caso de uso: CU_ Autenticar Usuario		
Entrada	Resultados	Condiciones
<p>El usuario introduce sus datos y solicita autenticarse.</p> <p>Usuario: "administrador"</p> <p>Contraseña: "abc".</p>	<p>Muestra un cartel: "Usuario o contraseña incorrecta".</p>	<p>Un usuario solicita entrar al Sistema.</p>

CU_ Gestionar Roles

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
<p>El administrador introduce los datos para crear un nuevo rol y presiona el botón "Adicionar". Los datos de entrada son: Nombre del Rol: "operadora de información", Descripción del Rol: "operadora de información" y Funcionalidades="Gestionar roles, Gestionar Usuario, Salvar y restaurar datos".</p>	<p>Se genera el siguiente mensaje:" ¿Desea guardar el nuevo rol?".</p> <p>Si se acepta el mensaje se adicionará un nuevo rol y se genera el siguiente mensaje: "Se adicionó el rol con éxito" y se reinician los valores de los campos relacionados.</p>	<p>Debe haber al menos un usuario creado.</p> <p>Debe haber al menos un rol creado.</p> <p>El administrador debe estar autenticado.</p>

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
<p>El administrador presiona el botón "Modificar". Los datos de entrada son: Descripción</p>	<p>Se genera el siguiente mensaje:" ¿Desea guardar los datos?".</p>	<p>Debe haber al menos un usuario creado.</p>

del Rol: “operadora de información” y Funcionalidades=”Gestionar roles, Gestionar Usuario, Salvar y restaurar datos, Consultar Notificaciones”.	Si se acepta el mensaje se modifica el rol deseado y se genera el siguiente mensaje: “Se modificó el rol con éxito”.	Debe haber al menos un rol creado. El administrador debe estar autenticado.
---	--	--

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
El administrador presiona el botón “Eliminar”. El dato de entrada es la selección del rol a eliminar.	Se genera el siguiente mensaje:” ¿Desea eliminar el rol?”. Si se acepta el mensaje se elimina el rol deseado y se genera el siguiente mensaje: “Se eliminó el rol con éxito”.	Debe haber al menos un usuario creado. Debe haber más de un rol creado. El administrador debe estar autenticado.

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
El administrador introduce los datos para crear un nuevo rol y presiona el botón “Adicionar”. Los datos de entrada son: Nombre del Rol: “operadora de información”, Descripción del Rol: “operadora de información” y Funcionalidades=”Gestionar roles, Gestionar Usuario, Salvar y restaurar datos”.	Se genera el siguiente mensaje: "El siguiente error ha ocurrido: " y a continuación de detalla la causa del error.	El administrador debe estar autenticado.

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
El administrador introduce los datos para crear un nuevo rol y presiona el botón "Adicionar". Los datos de entrada son: Nombre del Rol: "", Descripción del Rol: "operadora de información" y Funcionalidades=" Gestionar Usuario, Salvar y restaurar datos, Consultar Saldo".	Se genera el siguiente mensaje: "Entrada de datos no válida" .	El administrador debe estar autenticado.

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
El administrador introduce los datos para crear un nuevo rol y presiona el botón "Adicionar". Los datos de entrada son: Nombre del Rol: "administrador", Descripción del Rol: "administrador" y Funcionalidades=" Gestionar Usuario, Salvar y restaurar datos".	Se genera el siguiente mensaje: "El rol ya existe".	Debe haber al menos un usuario creado. Debe haber al menos un rol creado. El administrador debe estar autenticado.

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
El administrador introduce los datos para crear un nuevo rol y presiona el botón "Adicionar". Los datos de entrada son:	Se genera el siguiente mensaje: "No se pueden seleccionar a la vez las funcionalidades: Imprimir Pines e Imprimir Tarjetas" y	El administrador debe estar autenticado. Debe haber al

Nombre del Rol: "operadora de información", Descripción del Rol: "operadora de información" y Funcionalidades="Imprimir Pines, Imprimir Tarjetas, Gestionar Notificaciones".	se deseleccionan ambas opciones.	menos un rol creado.
--	----------------------------------	----------------------

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
El administrador introduce los datos para crear un nuevo rol y presiona el botón "Adicionar". Los datos de entrada son: Nombre del Rol: "usuario", Descripción del Rol: usuario común" y Funcionalidades=" Gestionar roles, Gestionar Usuario, Salvar y restaurar datos".	Se genera el siguiente mensaje: "No pueden existir dos roles con las mismas funcionalidades".	Debe haber al menos un usuario creado. Debe haber al menos un rol creado. El administrador debe estar autenticado.

Nombre del caso de uso: CU_ Gestionar Roles		
Entrada	Resultados	Condiciones
El administrador presiona el botón "Eliminar". El dato de entrada es la selección del rol a eliminar.	Se genera el siguiente mensaje: "El rol no puede estar asignado a ningún usuario".	Debe haber al menos un rol creado. El administrador debe estar autenticado. El rol a eliminar debe estar asignado a algún usuario en ese momento autenticado.

CU_Gestionar Usuario

Nombre del caso de uso: CU_Gestionar Usuario		
Entrada	Resultados	Condiciones
<p>El administrador introduce los datos para crear un nuevo usuario y presiona el botón "Adicionar". Los datos de entrada son: Nombre:"Antonio Perez", Usuario:"antonio", Contraseña:"aaa", Confirmar Contraseña="aaa", Rol="administrador" y Carnet de identidad:"82101122487".</p>	<p>Se genera el siguiente mensaje:" ¿Desea guardar el nuevo usuario?".</p> <p>Si se acepta el mensaje se adicionará un nuevo usuario y se genera el siguiente mensaje: "Se adicionó el usuario con éxito" y se reinician los valores de los campos relacionados.</p>	<p>Debe haber al menos un usuario creado.</p> <p>Debe haber al menos un rol creado.</p> <p>El administrador debe estar autenticado.</p>

Nombre del caso de uso: CU_Gestionar Usuario		
Entrada	Resultados	Condiciones
<p>El administrador presiona el botón "Modificar". Los datos de entrada son: Nombre:"Antonio Perez", Contraseña:"a", Confirmar Contraseña:"a", Rol:"usuario" y Carnet de identidad:" 82101122487".</p>	<p>Se genera el siguiente mensaje:" ¿Está seguro que desea guardar los cambios realizados?".</p> <p>Si se acepta el mensaje se modifica el usuario deseado y se genera el siguiente mensaje: "Se modificó el usuario con éxito" y se reinician los valores de los campos relacionados</p>	<p>Debe haber al menos un usuario creado.</p> <p>Debe haber al menos un rol creado.</p> <p>El administrador debe estar autenticado.</p>
Nombre del caso de uso: CU_Gestionar Usuario		
Entrada	Resultados	Condiciones
El administrador	Se genera el siguiente	El administrador

presiona el botón "Eliminar". El dato de entrada es la selección del usuario a eliminar.	mensaje: "¿Está seguro que desea eliminar el usuario?". Si se acepta el mensaje se elimina el usuario deseado y se genera el siguiente mensaje: "Se eliminó el usuario con éxito"	debe estar autenticado. Debe haber más de un usuario creado.
--	--	---

Nombre del caso de uso: CU_Gestionar Usuario		
Entrada	Resultados	Condiciones
El administrador presiona el botón "Modificar". Los datos de entrada son: Nombre:"Antonio Perez Perez", Contraseña:"abc", Confirmar Contraseña:"abc", Rol:"usuario" y Carnet de identidad:"82101122487".	Se genera el siguiente mensaje: "El siguiente error ha ocurrido: " y a continuación de detalla la causa del error.	El administrador debe estar autenticado.

Nombre del caso de uso: CU_Gestionar Usuario		
Entrada	Resultados	Condiciones
El administrador presiona el botón "Modificar". Los datos de entrada son: Nombre:"Antonio Perez", Contraseña:"", Confirmar Contraseña:"", Rol:"usuario" y Carnet de identidad:"7101122487".	Se genera el siguiente mensaje: "Entrada de datos no válida".	El administrador debe estar autenticado.

Nombre del caso de uso: CU_Gestionar Usuario		
Entrada	Resultados	Condiciones
El administrador presiona el botón "Eliminar". El dato de entrada es la selección del usuario a eliminar.	Se genera el siguiente mensaje: "No se puede eliminar un usuario en sección activa".	Debe haber al menos un usuario creado. El usuario a eliminar debe estar en ese momento en sección activa. El administrador debe estar autenticado.

CU_ Configuración

Nombre del caso de uso: CU_ Configuración		
Entrada	Resultados	Condiciones
El administrador llena o modifica los campos correspondientes a la configuración del sistema y solicita salvar los datos. Hora de Conciliaciones: "05.00.00". Tiempo de Inactividad: "02.06.06". Dirección Servidor BD:"10.7.22.220" Hora inicio Peticiones WS:"03.00.00" Direc Salvas de la Base Datos: "10.7.22.220"	Se mostró el siguiente mensaje: "¿Esta seguro que desea modificar la Configuración de la Banca?" Si lo aceptas se muestra el mensaje. "Se ha modificado la configuración de la Banca con éxito.	El Administrador este autenticado. El Administrador solicite definir la configuración del Sistema.

Hora de Salva :“05..00.00”		
Dirección Servidor FTP:“10.7.22.159”		
Usuario FTP:“administrador”		
Contraseña FTP:“pepe”		
Impresora de Pines: “10.8.10.10””		
Impresora de Tarjetas:“10.8.10.20”		

Nombre del caso de uso: CU_ Configuración		
Entrada	Resultados	Condiciones
<p>El administrador llena o modifica los campos correspondientes a la configuración del sistema y solicita salvar los datos.</p> <p>Hora de Conciliaciones: “05.00.00”.</p> <p>Tiempo de Inactividad: “02.06.06”.</p> <p>Dirección Servidor BD:“10.7.22.220”</p> <p>Hora inicio Peticiones WS:“03.00.00”</p> <p>Direc Salvas de la Base Datos: “dirección”</p> <p>Hora de Salva :“05..00.00”</p> <p>Dirección Servidor FTP:“Dirección FTP”</p> <p>Usuario FTP:“administrador”</p> <p>Contraseña FTP:“pepe”</p> <p>Impresora de Pines:“10.8.10.10””</p> <p>Impresora de Tarjetas:”Local”</p>	<p>Se mostró el siguiente mensaje.</p> <p>“Entrada de datos no valida”</p> <p>Además se mostró un asterisco de color rojo en los campos donde la entrada no fue valida, Direc Salvas de la Base Datos y Dirección Servidor FTP.</p>	<p>El Administrador este autenticado.</p> <p>El Administrador solicite definir la configuración del Sistema.</p>

CU_ Actualizar información BT

Nombre del caso de uso: CU_ Actualizar información BT		
Entrada	Resultados	Condiciones
<p>El administrador llena o modifica los campos correspondientes a la Organización donde se encuentra el Sistema y solicita salvar los datos.</p> <p>Nombre: "TeleBanca"</p> <p>Dirección: "Playa la CH"</p> <p>Teléfono: "8352142"</p> <p>Fax: "02145"</p> <p>Sitio Web: "http://TeleBanca"</p> <p>Logotipo: \\logo</p> <p>Nombre del director:"Ezequiel"</p> <p>Correo electrónico: "martinezm@estudiante.uci.cu"</p> <p>Ministerio: "UCI"</p> <p>Descripción de los servicios: "banca"</p>	<p>Se mostró el mensaje.</p> <p>"¿Esta seguro que desea modificar la Información de la Banca?".</p> <p>Si lo aceptas se muestra el mensaje.</p> <p>"Se ha modificado la información de la Banca con éxito".</p>	<p>El Administrador este autenticado.</p> <p>El Administrador accede a la opción de actualizar información de la Banca Telefónica.</p>

Nombre del caso de uso: CU_ Actualizar información BT		
Entrada	Resultados	Condiciones
<p>El administrador llena o modifica los campos correspondientes a la Organización donde se encuentra el Sistema y</p>	<p>Se mostró el siguiente mensaje.</p> <p>"Entrada de datos no</p>	<p>El Administrador este autenticado.</p> <p>El Administrador accede a la opción de</p>

<p>solicita salvar los datos.</p> <p>Nombre: "TeleBanca"</p> <p>Dirección: "Playa la CH"</p> <p>Teléfono: "8352142"</p> <p>Fax: "02145"</p> <p>Sitio Web: "TeleBanca"</p> <p>Logotipo: \\logo</p> <p>Nombre del director:"Ezequiel"</p> <p>Correo electrónico: "martinezm@yahoo.es"</p> <p>Ministerio:"UCI"</p> <p>Descripción de los servicios: "banca"</p>	<p>valida"</p> <p>Además se mostró un asterisco de color rojo en los campos donde la entrada no fue valida Sitio Web y Correo electrónico.</p>	<p>actualizar información de la Banca Telefónica.</p>
--	--	---

CU_ Consultar notificación.

Nombre del caso de uso: CU_ Consultar notificación		
Entrada	Resultados	Condiciones
<p>El Administrador selecciona los mensajes que ya no son de su interés y solicita eliminarlos.</p>	<p>Muestra una lista de mensajes sin las que fueron seleccionadas.</p>	<p>El Administrador este autenticado.</p> <p>El Administrador solicite consultar las notificaciones.</p> <p>Que el Administrador seleccione notificaciones que desea eliminar.</p>

CU_ Modificar clave.

Nombre del caso de uso: CU_ Modificar clave		
Entrada	Resultados	Condiciones
El usuario llena los campos correspondientes y solicita modificar su contraseña. Contraseña anterior: "123". Contraseña Nueva: "pepe". Confirmar Contraseña Nueva: "pepe".	Se le muestra un mensaje al usuario. "¿Esta seguro que desea cambiar su contraseña?" Después que se confirme esta acción se muestra el siguiente mensaje. "Su contraseña ha sido cambiada satisfactoriamente".	El usuario este autenticado.

Nombre del caso de uso: CU_ Modificar clave		
Entrada	Resultados	Condiciones
El usuario llena los campos correspondientes y solicita modificar su contraseña. Contraseña anterior: "123456". Contraseña Nueva: "pepe". Confirmar Contraseña Nueva: "pepe".	Se le muestra un mensaje al usuario. "Contraseña incorrecta".	El usuario este autenticado.

Nombre del caso de uso: CU_ Modificar clave		
Entrada	Resultados	Condiciones
<p>El usuario llena los campos correspondientes y solicita modificar su contraseña.</p> <p>Contraseña anterior: "123".</p> <p>Contraseña Nueva: "pepe".</p> <p>Confirmar Contraseña Nueva: "pepito".</p>	<p>Se le muestra un mensaje al usuario.</p> <p>"La nueva contraseña y su confirmación no coinciden".</p>	<p>El usuario este autenticado.</p>

CU_ Salvar y Restaurar Datos.

Nombre del caso de uso: CU_ Salvar y Restaurar Datos		
Entrada	Resultados	Condiciones
<p>El administrador selecciona la opción realizar salva y la ejecuta.</p>	<p>Se le muestra un mensaje al Administrador.</p> <p>"¿Está seguro que desea salvar los datos?".</p> <p>Después que se confirme esta acción se muestra el siguiente mensaje.</p> <p>"Se realizo la salva con éxito"</p>	<p>El Administrador este autenticado.</p>

Nombre del caso de uso: CU_ Salvar y Restaurar Datos		
Entrada	Resultados	Condiciones
El administrador selecciona la opción realizar salva y la ejecuta.	<p>Se le muestra un mensaje al Administrador.</p> <p>“¿Está seguro que desea salvar los datos?”.</p> <p>Después que se confirme esta acción se muestra el siguiente mensaje.</p> <p>“Error de conexión con la base de datos.”</p>	El Administrador este autenticado.

Nombre del caso de uso: CU_ Salvar y Restaurar Datos		
Entrada	Resultados	Condiciones
El administrador selecciona la opción restaurar datos y selecciona por cuál de las salvas realizadas, desea hacer la restauración de la base de datos, ejecuta dicha acción.	<p>Se le muestra un mensaje al Administrador.</p> <p>“¿Está seguro que desea Restaurar los datos?”.</p> <p>Después que se confirme esta acción se muestra el siguiente mensaje.</p> <p>“Se realizo la restauración con éxito”</p>	El Administrador este autenticado.

Nombre del caso de uso: CU_ Salvar y Restaurar Datos		
Entrada	Resultados	Condiciones
El administrador selecciona la opción restaurar datos y selecciona por cuál de las salvadas realizadas, desea hacer la restauración de la base de datos, ejecuta dicha acción.	Se le muestra un mensaje al Administrador. “¿Está seguro que desea Restaurar los datos?”. Después que se confirme esta acción se muestra el siguiente mensaje. “Error de conexión con la base de datos.”	El Administrador este autenticado.

4.3 Conclusiones

Este capítulo es el ultimo de presente trabajo, el cual mostró los artefactos principales del los flujos de trabajo Implementación y Prueba. En la etapa de Implementación se construyó el diagrama de despliegue y el diagrama de componente de cada una de las capas del Módulo de Administración, que proporcionaron la vista física del módulo, modelando su estructura de Implementación y su organización en componentes. De la fase de Prueba se construyó el modelo de prueba con sus respectivos casos de prueba, los cuales permitieron chequear cada una de las funcionalidades del sistema.

Conclusiones

Después de haber realizando una investigación sobre la administración de sistemas informáticos, definir los procesos de administración, realizar el Análisis, Diseño, Implementación y Prueba del módulo de administración del sistema TeleBanca, se considera que se ha cumplido con éxito el objetivo general del proyecto: Obtener la implementación del módulo de administración del sistema TeleBanca, con el cual se logra:

Gestionar los Roles y Usuarios del sistema.

- Alcanzar una personalización del entorno de trabajo de acuerdo al rol que desempeña cada usuario en el Sistema.
- Facilitar la actualización de los datos configurables del sistema para garantizar una mejor operabilidad del mismo.
- Permitir consultar y eliminar las notificaciones que genera el sistema.
- Realizar copias de seguridad que ayudan a evitar la pérdida accidental de datos, en un soporte que posibilite su recuperación.
- Obtener un acceso sencillo a la ayuda del sistema.

Que garantizan realizar con gran facilidad las principales tareas de Administración del sistema encargado de gestionar el pago de los servicios por vía telefónica.

Recomendaciones

Con el objetivo de mejorar el desempeño del Módulo de Administración que se propone en el presente trabajo, se recomienda lo siguiente:

- Continuar con la investigación para garantizar nuevas y buenas mejoras en futuras versiones del Módulo de Administración y del proyecto TeleBanca en general.
- En próximas versiones incluir la gestión de las trazas, con el fin de dejar constancias de cada operación que se realice en el sistema.
- Estudiar en profundidad las herramientas de software libre, para el desarrollo de futuras versiones.

Bibliografía

CARRERA, R. Apuntes de la materia de sistemas integrales de información. 2002.

CHIAVENATO, I. Introducción a la Teoría General de la Administración. Colombia 1999.

CONALLEN, J. Modeling Web Application Architectures with UML. June 1999.

Cambio de aplicaciones a .NET Disponible en:

http://www.microsoft.com/spanish/msdn/articulos/archivo/130902/voices/dotnet_MovingJavaApps.asp.

Guía del desarrollador de .NET Framework Información general acerca de .NET Framework Disponible en:

<http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpovrIntroductionToNETFrameworkSDK.asp>.

.NET FRAMEWORK Disponible en:

<http://www.intec.edu.do/~rmfs/componentes/herramientas.html>.

¿Qué es ASP.NET? Disponible en:

<http://es.gotdotnet.com/quickstart/aspplus/doc/whatisaspx.aspx>.

Tecnología .NET Disponible en:

<http://www.microsoft.com/spain/technet/productos/net/microsoftnet.mspix>.

DANIEL RUBIOLO, J. D. M., EDWARD JEZIERSKI, ALEX MACKMAN
Microsoft®.NET Explained. 2004.

DAVID, C. ob.cit.

FERRAILOLO, D. F.; SANDHU, R., et al. Proposed NIST Standard for Role-Based Access Control. Disponible en: <http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf>.

GICH, J. Teoría General de sistemas. México. 1981.

IBIDEM. nº

JACOBSON, I.; BOOCH, G., et al. El Proceso Unificado de Desarrollo de Software Editado por: Educación, P. Pearson Educación. S.A., Madrid, 2000, ISBN 84-7829-036-2.

KOONTZ, H. y WEINRICHY, H. Elementos de la Administración. México 1998.

LARMAN, C. UML y Patrones.

- Introducción al análisis y diseño orientado a objetos. Editado por: Hall, P.
Prentice Hall Hispanoamericana, S.A., 1999, ISBN 970-17-0261-1.
- LHOTKA, R. Autenticación y autorización (Aventuras en .NET) Disponible en:
<http://www.microsoft.com/spanish/msdn/articulos/archivo/020104/voices/vbnet10282003.asp>.
- LUCENA LÓPEZ, M. J. Criptografía y Seguridad en Computadores Tercera Edición. Marzo de 2004.
- MURDICK, R. Sistemas de Información administrativa. México. 1998.
- NARANJO, H. M. I. y ARGEMÍ, J. A. M. RBAC: Alternativa actual para la realización de Control de Accesos.
- PRESSMAN, R. S. Ingeniería del Software: un enfoque práctico Editado por: Varela, E. F. Español ED. 2005. Vol. 1.
- RODRÍGUEZ, A. Sistemas para la toma de decisiones. 2002.
- RUMBAUGH, J.; JACOBSON, I., et al. El Lenguaje Unificado de Modelado. Manual de Referencia. Editado por: Wesley, E.
- SANCHEZ, M. A. M. Metodologías De Desarrollo De Software. Junio 7 del 2004.
- SANDHU, R. S.; COYNEK, E. J., et al. Role-Based Access Control Models. Febrero de 1996, vol. 29.
- SECO, J. A. G. Tema 2: Introducción a C#. Publicado el: abril del 2007 de 2006, última actualización: abril del 2007. Disponible en:
<http://www.programacion.com/tutorial/csharp/3/>.
- SENN, J. Análisis y diseño de sistemas de información. México. 1995.
- SHERIFF, P. D. Criptografía con Microsoft .NET (Seguridad de .NET Framework) PDSA.com, Disponible en:
<http://www.microsoft.com/spanish/msdn/articulos/archivo/101203/voices/cryptosimplified.asp>.
- SUN. Conozca más sobre la tecnología Java Sun Microsystems, Disponible en:
<http://www.java.com/es/about/>.
- TOMAS, J. G. y LANCHARRO, E. A. Principales tipos de códigos. 1993.
- WILLIAM, S. Cryptography and Network Security. Principles and Practice. 1999.