

Universidad de las Ciencias Informáticas
Facultad 4



Título: Concepción y desarrollo del
Módulo Servicio de Pago del proyecto Telebanca

Trabajo de Diploma para optar por el título de

Ingeniero en ciencias Informáticas

Autores: Lissett Díaz Mesa

Osniel Cainzo Vazquez

Yeslidier López Reinoso

Tutor: Lic. David Batard Lorenzo

Junio del 2007

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Lissett Díaz Mesa

Osniel Cainzo Vazquez

Yeslidier López Reinoso

Firma del autor

Firma del autor

Firma del autor

David Batard Lorenzo

Firma del tutor

DATOS DE CONTACTO DEL TUTOR

Lic. David Batard Lorenzo, Profesor Instructor de los Dptos. de IA e IGS.
Graduado de Ciencias de la Computación, en la UCLV y UH.
Actual Asesor de la Dirección de Investigaciones de la UCI.

AGRADECIMIENTOS

A todos los profesores que desde el primer año de nuestra carrera nos han brindado su apoyo y han sido comprensivos.

A nuestros amigos que aunque han variado de año en año hoy en esta simple oración se junta todos, a ellos gracias por estar con nosotros de una forma u otra, en las buenas y en las malas, en especial Ale, Robert, Erick, Alejandro, Misael, Vladimir, Dagoberto, Johan.

A Yuliesky, David, Yaimi, Alexander por las ideas que aportaron en la realización de este trabajo.

A los muchachos de cuarto año del proyecto que también nos ayudaron para terminar nuestro trabajo.

Por último y el más especial de los agradecimientos para nuestro comandante en jefe Fidel Castro Ruz, gracias por depositar todas su confianza en nosotros y aprovechar para desearle su pronta recuperación.

DEDICATORIA

Osniel

A toda mi familia que me ha apoyado mucho y siempre ha estado ahí cuando los he necesitado, especialmente:

A mi mami que lo que soy es gracias a ti y sin ti nada sería posible. A mi papa que siempre me diste tu apoyo y confiaste en mí. A papi y mima, ustedes son parte inseparable de mi corazón donde quiera que este y se que este sueño también es de ustedes. A mi hermanito querido, confió mucho en ti, es un camino largo y difícil pero se que tu también podrás lograrlo. A mima Amelia y Mingo que siempre me han dado su apoyo y me han ayudado mucho, a tía Coralia que siempre me ayudaste cuando lo necesité, a todos mis tíos y tías, a mis amigos, y a todos aquellos que han hecho posible que este sueño se haga realidad.

Yeslidier

A mami por ser mi ejemplo en todo momento, la luz que por tenue que sea no me abandona nunca, ella es la que me ha inspirado para ser quien soy y para llegar a donde estoy, la dueña de la casa, mi inspiración como persona y como todo. Mi mejor amiga, mi confidente, quien me enseñó a diferenciar lo bueno de lo malo. A ella dedico todo mi esfuerzo.

A mi hermano por estar ahí para ayudarme en cada uno de mis problemas, por ser alguien incondicional, por apoyarme en todas mis decisiones, por hacerme darme cuenta de mis malos actos en el momento adecuado, aunque no lo de tanto a conocer te quiero un montón hermano y pase lo que pase nunca te daré la espalda.

A mi otro hermano que también es incondicional que me aconseja y me defiende demasiado, siempre esta tirándole a todo el que se mete conmigo, te agradezco por ser especial y sabes que donde quiera que estés siempre podrás contar conmigo.

A toda mi familia que de una forma o de otra me han ayudado a llegar a donde estoy, me han apoyado espiritualmente y confiaron siempre en que sería un profesional, a Tata y René los quiero por ser tan especiales, a Mami Dulce y Papi Mingo lo mismo para ustedes.

Por último dedicar este trabajo a todos mis amigos que aunque no nos veamos mas el día de mañana nunca olvidaré en que parte de mi camino de una forma o de otra ustedes me dieron su mano y me ayudaron a pasar por los momentos difíciles de mi carrera.

Lissett

A mi mamá por tener siempre confianza en mí y guiarme en todos los caminos, siendo más que madre es una amiga apoyándome en todas mi desiciones estando o no de acuerdo con ellas. A mi papá, a pesar de no vivir conmigo siempre lo he sentido cerca, por su apoyo incondicional durante el transcurso de mis estudios. A mi hermano que lo quiero mucho. A mis tías Arismaida y Aridelia que me han ayudado mucho y han creído en mí. A mama Reinalda y abuela Eneida que estarán siempre en mi corazón. A mi tío Carlos que siempre esta bromeando conmigo. En general a toda mi familia que de una forma u otra a estado siempre presente en especial a mi tía Cuqui, mis primos Abdelin, Robertico, Reinier, Jandy y Ana Beatriz.

RESUMEN

La tendencia actual muestra que las entidades para ser más eficaces, eficientes y competitivas deben aplicar la informatización en todas las esferas. Contar con aplicaciones desarrolladas en computadoras posibilita un acceso rápido y fácil a la información lo que permite realizar una buena gestión. Es por ello que una de las principales tareas del Gobierno Cubano es desarrollar la Industria del Software, no solamente con el fin de insertarnos en el mercado de software a nivel mundial por su perspectiva económica, sino también por los beneficios que para el país aporta el desarrollo de sistemas informáticos para uso interno.

Con la idea de mejorar la gestión del cobro de los servicios que se prestan actualmente a la población el gobierno cubano decide desarrollar un sistema de banca telefónica, al cual se le dio por nombre "Telebanca", el mismo centra su actividad en la posibilidad de pago de servicios de tenedores de cuentas de ahorro asociados a tarjetas debito RED del Sistema Bancario a través de una operadora, la cual ejecutará las transacciones a partir de las indicaciones recibidas del cliente por vía telefónica. El presente trabajo está centrado en el modulo "Servicio de Pago" de dicho proyecto y parte de la necesidad gestionar dentro de la banca información de los servicios que se acogerán al método de pago vía telefónica, así como la gestión del pago de estos servicios, atención a reclamaciones y que controle información de los bancos que se asocien a la banca telefónica.

TABLA DE CONTENIDOS

AGRADECIMIENTOSI

DEDICATORIA II

RESUMEN.....IV

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 5

1.1. Introducción..... 5

1.2. Los Sistemas de banca telefónica..... 5

1.3. Tendencias y tecnologías actuales 8

1.4. Fundamentación de la metodología de desarrollo utilizada..... 9

1.5. Plataforma Microsoft.Net..... 12

1.5.1. Visual Studio .NET..... 13

1.5.2. El lenguaje C#..... 14

1.5.3. ASP. Net 15

1.6. Cristal Report para generar reportes 16

1.7. Visual SourceSafe 16

1.8. Conclusiones..... 17

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA..... 18

2.1. Introducción..... 18

2.2. Problema y Situación Problemática..... 18

2.3. Objeto de Automatización..... 19

2.4. Propuesta del sistema 19

2.5. Modelo del dominio 21

2.6. Especificación de los requisitos de software..... 22

2.6.1. Requerimientos Funcionales..... 22

2.6.2. Requerimientos no Funcionales..... 23

2.7. Diagrama de casos de uso 25

2.8. Definición de los casos de uso..... 28

2.8.1. Definición de los actores	28
2.8.2. Listado de casos de uso	28
2.9. Conclusiones	59
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	60
3.1. Introducción	60
3.2. Análisis del sistema	60
3.2.1. Diagrama de clases del análisis	60
3.3. Diseño del sistema	64
3.3.1. Diagramas de secuencias del diseño	64
3.3.2. Diagrama de clases del diseño	76
3.3.3. Descripción de las clases del diseño	78
3.4. Conclusiones	96
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	97
4.1. Introducción	97
4.2. Implementación	97
4.2.1. Diagrama de despliegue	97
4.2.2. Diagrama de componentes	98
4.3. Modelo de Prueba	99
4.3.1. Diseño de los casos de usos de prueba	99
4.4. Conclusiones	108
CONCLUSIONES	109
RECOMENDACIONES	110
BIBLIOGRAFÍA	111
GLOSARIO	113

INTRODUCCIÓN

La Ciencia de la Información es una disciplina joven, surgida ante la necesidad de organizar y controlar el enorme flujo de información científica y tecnológica que ha estado a disposición de la sociedad. Hoy los sistemas de información constituyen una de las características fundamentales de la modernidad, ocupando un espacio preponderante en la actividad de un número cada vez más importante de empresas, organismos e instituciones. La humanidad se encuentra en el inicio de una nueva época, en la que está perdiendo importancia el trabajo físico que requirió la revolución industrial surgida a finales del siglo XVIII, a la vez que adquiere mayor significado el trabajo mental que necesita esta nueva era de la información. Los grandes avances en la ciencia y la técnica logrados por la humanidad, en todas las ramas de la producción y los servicios, no se conciben sin la participación continuada y creciente de los recursos informáticos y computacionales, los cuales debido al acelerado desarrollo de nuevas tecnologías de la información han evolucionado, perfeccionado y aumentan cada vez más en potencialidad a favor del crecimiento de la calidad en los procesos empresariales y los servicios prestados a la población.

Cuba desde muy temprano se ha interesado por desarrollar las tecnologías de la información y las telecomunicaciones (TIC) para alcanzar mayor calidad, eficiencia y competitividad en el comportamiento social y las prácticas empresariales. Ha dedicado el uso de estos modernos medios a aquello que en una sociedad como la nuestra se considera importante y beneficioso para todos, como ha sido el desarrollo del programa “Universidad para todos” que llega hasta los lugares más intrincados del país, también están los “Joven Club de Computación y Electrónica” prestando importantes servicios a centros de salud, escuelas, programas sociales y culturales, entre otros. Por último la creación de la Universidad de las Ciencias Informáticas (UCI) que jugará un papel importante en el desarrollo de la Industria Cubana del Software, y en la realización de los proyectos asociados al programa cubano de informatización. Por lo que entre sus objetivos esta el uso masivo de las TIC, debido al creciente desafío que impone el desarrollo tecnológico.

El cobro de los servicios ofrecidos a la población e instituciones como agua, electricidad, teléfono, etcétera (etc.) es una gestión comercial que demanda personal calificado en el cobro de estos servicios. Existen muchas quejas de la población al ser este un proceso dilatado ya que tienen que trasladarse hacia las oficinas en horario laboral, sumando además el tiempo de espera en la cola para efectuar el pago. Por otra parte algunas empresas requieren contratar personas para llegar a cada uno de los hogares para

realizar el cobro, responder a quejas y reclamaciones. Sin contar el alto riesgo que corren los cobradores al tener que circular con gran cantidad de efectivo por las calles.

Con el objetivo de mejorar estos inconvenientes en las prestaciones de los servicios el gobierno cubano decide desarrollar un sistema de banca telefónica para el pago de servicios de tenedores de Cuentas de Ahorro asociados a Tarjetas Debito RED del Sistema Bancario, donde las transacciones se realizarán a través de una operadora por indicaciones del cliente vía telefónica. Al cual se le dio por nombre de "Telebanca". El mismo estará integrado por el modulo "Servicio de Pago", motor principal de la banca telefónica en el cual estará centrado nuestro trabajo.

Se partió de la necesidad de desarrollar un modulo capaz de responder a la población en cuanto al pago de los servicios y atención a las reclamaciones. Que controle información referente a las entidades asociadas a "Telebanca" y de los servicios que se prestan actualmente acogidos al método de pago a través de la banca telefónica.

En la actualidad son muchas las empresas que han optado por el desarrollo de sistemas de banca telefónica en la gestión del cobro de los servicios que prestan al cliente, como estrategia para ofrecer flexibilidad a estos para que puedan realizar todo tipo de operaciones en tiempo real y desde cualquier punto por distante que este sea. Este tipo de servicio nació orientado a un segmento de la población, profesionales con alta dedicación laboral que cuenta con poco tiempo y con el fin de alcanzar mejor calidad, operatividad y rapidez en la atención al cliente. En Cuba no se ha explorado aun en este campo, por lo que somos pioneros en el tema, pero en el mundo son muchas las entidades que operan con banca telefónica como son Caixa Galicia, entidad de Crédito sujeta a la supervisión del Banco de España, BICE y RIPLEY de Chile, Grupo Financiero G&T Continental en Guatemala, entre otros. La mayoría ofreciendo servicios de consulta de saldo, recepción de reclamos, transferencias, pago de servicios básicos, etc.

Dar solución al problema anteriormente expuesto es el objetivo que se persigue con el desarrollo de este trabajo por lo que el problema consiste en ¿Cómo desarrollar dentro de la banca telefónica un módulo que automatice la gestión del pago de servicios, atención a reclamaciones y gestione información que respecte a los bancos asociados a Telebanca y de los servicios dentro de la misma acogidos al método de pago vía telefónica?

El **objeto de estudio** del presente trabajo es la *gestión del pago de servicios* a través de los sistemas de banca telefónica.

Para responder al problema científico anteriormente planteado se trazó como **objetivo general**: Analizar, Diseñar e implementar el modulo Servicio de Pago de Telebanca para facilitar la gestión del pago de los servicios.

Del objeto de estudio derivamos que el **campo de acción** que abarca este trabajo, es la *gestión del pago de servicios* a través del sistema de banca telefónica "Telebanca" en Cuba.

Como **aportes prácticos** esperados en el desarrollo del modulo de Servicio de Pago tenemos que se logrará una respuesta rápida y eficiente al cliente en cuanto al pago de los servicios y atención a las reclamaciones.

Para dar cumplimiento al Objetivo general, se trazaron cuatro objetivos específicos, los que se enumeran a continuación:

- Analizar como se lleva a cabo la gestión del pago de servicios en Cuba.
- Analizar los diferentes sistemas de bancas telefónicas que existen actualmente en el resto del mundo.
- Analizar y diseñar una solución para la gestión del pago de servicios.
- Implementar el modulo Servicio de Pago para la gestión del pago de servicios a través del sistema de banca telefónica "Telebanca".

Para responder a los objetivos específicos llevaremos a cabo las siguientes **tareas**:

- Revisión bibliográfica sobre los procesos de pago y gestión de servicios a través de bancas telefónicas que existen actualmente en el mundo para evaluar, identificar y proponer una solución al problema que da paso a esta investigación.
- Investigar como se desarrolla la gestión del pago de servicios actualmente en Cuba.

- Modelar e implementar el modulo “Servicio de Pago” de la banca telefónica.
- Documentar las etapas de desarrollo del software basado en alguna de las metodologías de desarrollo que existen actualmente.

El presente trabajo tiene su contenido distribuido en 4 capítulos:

El Capitulo 1 El propósito de este capítulo es la formalización de todos los conceptos asociados al tema y que son necesarios para la comprensión de lo que se describe en el resto del trabajo. Se describen algunos sistemas de banca telefónica como alternativa para la gestión del pago de servicios que existen hoy en día y las ventajas de los mismos a favor del desarrollo de la calidad y rapidez en la atención al cliente. Trata la situación sobre las técnicas, tecnologías, metodologías y softwares que existen actualmente y utilizamos en el desarrollo de la propuesta.

El Capitulo 2 El propósito de este capítulo es la descripción del objeto de estudio del presente trabajo. Se describirá la propuesta del sistema como solución a los problemas que dan paso a esta investigación, así como la definición de los requisitos no funcionales como funcionales, de este ultimo se definirán los casos de usos, detallando aquellos de gran significado en el desarrollo de la solución.

El Capitulo 3 El propósito de este capítulo es realizar el análisis y diseño a la solución planteada en el capítulo 2 donde como artefactos resultantes de estas actividades se obtendrán, el diagrama de clases del análisis y diagrama de iteracion del diseño para aquellos casos de usos significativos para la arquitectura, además del diagrama de clases del diseño correspondientes a la lógica de presentación y negocio.

El Capitulo 4 El propósito de este capítulo es la elaboración del diagrama de despliegue y componentes, asi como la realización de los casos de usos de pruebas de aquellos casos de usos considerados significativos desde el punto de vista arquitectónico.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El surgimiento y la evolución incesante de nuevos dispositivos y tecnologías están transformando el mundo empresarial a un ritmo inimaginable. La necesidad de aplicaciones para la automatización de procesos empresariales marca una nueva tendencia. Estas aplicaciones aportan mayor eficacia y productividad, liberando a los trabajadores de muchas de las tareas cotidianas de una empresa.

En este primer capítulo trataremos sobre los sistemas de banca telefónica, como alternativa para las empresas en cobro de los servicios que brindan a sus clientes, en función de lograr beneficio social y garantía económica en sus prestaciones. Explicación sobre las metodologías, tecnologías y herramientas de desarrollo en las que nos apoyamos para resolver el problema.

1.2. Los Sistemas de banca telefónica

En la actualidad, la calidad de servicio ha constituido en un factor determinante para los clientes al momento de decidirse por un determinado producto o servicio. Uno de los aspectos de la calidad de estos servicios, que se puede cuantificar fácilmente, es el tiempo: nadie quiere pasar demasiado tiempo esperando ser atendido por la empresa que nos da un servicio. Es por esta razón que las tareas de dimensionamiento de recursos en actividades relacionadas con el servicio al cliente son importantes, pues no sólo es necesario cuantificar la mínima cantidad de recursos que se necesitan para atender a los clientes, sino también considerar un determinado nivel de servicio para ello. La evolución de la tecnología, sobretodo en el rubro de las telecomunicaciones, ha originado el desarrollo de la atención al cliente no presencial con la implantación de los sistemas de banca telefónica, potenciando al teléfono como principal herramienta de contacto con el cliente.

Las entidades financieras se han unido a la constante evolución de la tecnología; en banca telefónica, el teléfono es algo más que una herramienta, convirtiéndose en un intermediario que proporciona al cliente, en cualquier momento un sencillo y cómodo acceso a sus operaciones bancarias, y al banco un mayor acercamiento a sus clientes y aumento de su productividad. Esto no sería posible sin la integración de los últimos desarrollos informáticos.

La banca telefónica la encontramos dentro del sector de los call center, este no es más que una solución tecnológica que integra telefonía y computación orientado a atender grandes volúmenes de llamadas, encaminado hacia el logro de un objetivo principal, el de comunicar y hacer llegar a los clientes información de productos, atención de quejas, saldos de cuentas, promoción y ventas de productos o servicios, entre otros. Entidades que optan por esta vía de atención al cliente definen a la banca telefónica como un conjunto de servicios telefónicos para que desde la comodidad del hogar u oficina, sea posible realizar operaciones bancarias.

La banca telefónica ha sido creada por los bancos para satisfacer las necesidades de sus clientes y darles nuevas formas para facilitarles el uso de los servicios financieros con sólo marcar un número telefónico. Cada institución opera los servicios de banca por teléfono de manera particular. El cliente se comunica al número que el banco le ha proporcionado para emplear este sistema y, pulsando una clave personal, podrá acceder a los diferentes servicios de la banca por teléfono. Algunos bancos cuentan con la modalidad del sistema automático para los servicios de identificación y consultas mediante el reconocimiento de tonos (el cliente opera con el sistema usando el teclado del teléfono). Existe otra modalidad de la banca por teléfono donde se requiere de una operadora para que el trato sea más personal dependiendo del tipo de operaciones que se pretendan realizar. Los servicios que ofrece el sistema de banca por teléfono varían de acuerdo con cada banco, al tipo de producto (o servicios) que el cliente maneja dentro del banco y a las operaciones que necesite realizar.

En Cuba no existe ninguna entidad que brinde servicio de banca telefónica, pero en el mundo son muchos los países que cuentan con empresas que si disponen de estos e incluso aun mas desarrollados como es la banca a través internet, dentro de las entidades bancarias que hacen uso actualmente de la banca telefónica como servicio de atención al cliente está el Banco Bilbao Vizcaya Argentaria (BBVA), es uno de los grupos financieros más importantes de Europa que cuenta con 40 millones de clientes, casi 9.000 oficinas en todo el mundo y más de 100.000 empleados. Se reconoce como la entidad bancaria con mayor presencia en América Latina, donde dispone de una red de más de 4.000 oficinas.

El servicio de banca telefónica del BBVA comenzó su actividad en junio de 1994, con una plantilla de 50 empleados. Los principales segmentos de demanda atendidos eran y siguen siendo los clientes particulares. Ofrecen un servicio gratuito que permite al cliente, durante las 24 horas del día, todos los

días del año, entrar en el BBVA sin tener que desplazarse, ni soportar incómodas esperas. Mediante este sistema la entidad bancaria ofrece a todos sus clientes un servicio que les permite realizar, a través del teléfono, la consulta de cualquier tipo de información general o específica sobre los productos y servicios disponibles, así como realizar todas las operaciones más habituales como solicitar transferencias o domiciliaciones, vender o comprar valores, ejecutar aportaciones a fondos de inversión y planes de pensiones o contratar seguros, etc. La forma de acceder a este servicio es marcando el número de teléfono determinado por el banco para prestar el servicio de banca telefónica, al que responde un sistema automático que atiende la llamada. En función del tipo de petición, el sistema solicita las claves personales de acceso (los 12 últimos dígitos de alguna de las tarjetas BBVA y el número de identificación personal) y da respuesta a la petición o pasa la consulta al equipo de asesoramiento profesional con que cuenta la empresa.

Entre otras de las entidades que cuentan con banca telefónica se encuentran La Caixa de España, RIPLEY y BICE de Chile, Grupo Financiero G&T Continental en Guatemala. Las operaciones que se lleven a cabo a través de la banca telefónica serán en dependencia de los objetivos que persiga la misma con el desarrollo de esta, para lo cual se pone al cliente en contacto con un gestor telefónico que le ayudará y asesorará durante todo el proceso. Para que se le facilite el acceso a sus datos, es un requisito fundamental que el usuario se autentique previamente, para lo cual se le proporcionan una serie de contraseñas y claves de acceso que deberá revelar al operador telefónico que atiende su llamada. Estos servicios en general no añaden nada a lo que se ofrece en las propias oficinas del banco, más allá de la comodidad y flexibilidad de uso. De hecho, sólo suelen estar disponibles en horarios restringidos al menos cuando se trata de servicios atendidos por operadores humanos (los llamados gestores telefónicos). Para disfrutar el servicio es necesario firmar antes un contrato, tras lo cual se reciben las claves de acceso y a menudo también una tarjeta de coordenadas, utilizada para dotar de mayor seguridad al servicio. A pesar del impacto que ha tenido la banca por internet, la banca telefónica aún coexiste y lo hará por mucho tiempo debido a que muchas personas no disponen de ordenador o de conexión a Internet, mientras que un teléfono resulta ubicuo hoy en día.

1.3. Tendencias y tecnologías actuales

1.3.1. Información a través de la Web

La Web es un sistema interactivo, dinámico, distribuido y gráfico de hipertexto que funciona sobre Internet, preparado para el intercambio de información diseñado para el consumo humano. Consiste en ofrecer una interfaz simple y consistente para acceder a la inmensidad de los recursos inagotables de Internet. Es la forma más moderna de ofrecer información en forma de páginas electrónicas.

Dentro de sus características sobresale el texto remarcado, un método para referencias cruzadas instantáneas con gran variedad de datos como son: textos, sonidos, videos, imágenes, entre otros. Una vez conectado a internet es necesario contar con un programa capaz de leer los documentos de hipertextos para acceder a las páginas web y llevarte a otras siguiendo enlaces, al mismo se le conoce como "navegador", "browser", "visualizador" o "cliente", pero el diseño de la página debe mantener un equilibrio entre utilizar todas las capacidades y la posibilidad de ser leída por cualquier tipo de navegador. El protocolo de comunicación que se utiliza en la Web es el http (HiperText Transfer Protocol) y el formato que se utiliza para la transferencia es el HTML (HiperText Markup Lenguaje).

1.3.2. Aplicaciones Web

Los sitios Web han evolucionado notoriamente respecto a su complejidad, se pueden observar desde centros de información de solo lectura, hasta aplicaciones distribuidas basadas en Web. Las aplicaciones Web, son sitios especializados para realizar funciones más "inteligentes". Toman decisiones sobre que y cómo ofrecer un servicio al cliente. Tienen características similares a las páginas Web y su popularidad se debe a la habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes. Estas generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario.

Son diferentes de los documentos hypermedia tradicionales por tres aspectos principales como son: los usuarios no solo navegan sino también activan operaciones y transacciones, la estructura hypermedia puede evolucionar cuando las operaciones de la aplicación evolucionan y diferentes usuarios pueden tener diferente visibilidad de la información y diferentes capacidades para ejecutar operaciones. Existen numerosos lenguajes de programación utilizados para el desarrollo de Aplicaciones Web, entre los que

destacan: PHP ASP/ASP.NET Java, con sus tecnologías Java Servlets y JavaServer Pages (1)(JSP) Perl Ruby Python. Aunque ciertamente ASP no es un lenguaje de programación, sino una arquitectura de desarrollo web en la que se pueden usar por debajo distintos lenguajes como VB.Net o C#.Net para ASP.Net.

1.3.3. Servicios Web

Un servicio Web es un componente de aplicación que se puede invocar utilizando protocolos estándar de Internet tales como HTTP y XML. Por lo tanto, cualquier sistema compatible con estos protocolos puede admitir la implementación de servicios Web (1).

Los Servicios Web surgen de la necesidad de establecer un sistema de intercambio de datos normalizado entre aplicaciones. Son componentes de aplicaciones distribuidos que cumplen las normas para estar disponibles de forma externa. La función de servicios web soluciona el problema de integrar aplicaciones que se han desarrollado individualmente para que puedan ejecutarse en diferentes plataformas de hardware y software. Una ventaja importante de los servicios web es que son multiplataforma, independientes tanto de la arquitectura como del Sistema Operativo y el lenguaje de programación que se use, ya que se basan en XML como estándar para el intercambio de datos, por lo que a veces también se les llama servicios web XML. Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento. Al apoyarse en HTTP, los pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado. Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados. Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

1.4. Fundamentación de la metodología de desarrollo utilizada

Todo proyecto de software, independientemente de su envergadura, tiene dentro de sus objetivos entregar un producto de software de calidad, que el desarrollo alcanzado cumpla con todas las especificaciones establecidas dentro de los costos, tiempos y recursos planificados. Para esto, se hace necesario que el equipo del proyecto adopte un proceso de desarrollo de software, un marco de trabajo que defina las actividades necesarias para garantizar, técnica y administrativamente, que un Software pueda ser desarrollado de manera organizada, disciplinada y previsible. Por esto se decidió tomar el Proceso

Unificado de Rational (RUP) como metodología para el desarrollo de este proyecto acompañado de la herramienta Rational Rose y UML como lenguaje de modelado visual.

1.3.4. Proceso Unificado del Rational

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistema de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto (2).

El proceso unificado es un proceso de desarrollo de software que ayuda a mejorar la productividad del equipo de trabajo, definiendo claramente sus actividades, roles y responsabilidades, desde los jefes de proyectos a los analistas y desde los desarrolladores a los probadores. Esta basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para la construcción de todos los esquemas de un sistema de software. A parte de las características anteriormente mencionadas existen tres aspectos que resumen el proceso unificado y son:

Dirigido por casos de uso: Los casos de uso guían el proceso de desarrollo que avanza a través de una serie de flujos de trabajos que tienen como partida los casos de usos, los cuales reflejan lo que los futuros usuarios necesitan y es captado durante la modelación del negocio y representado a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo a partir de modelos que se obtienen como resultado de los diferentes flujos de trabajo, representando la realización de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura el cual representa a través de vistas en las que se incluyen los diagramas de UML.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones dividiendo el trabajo en pequeñas partes o mini proyectos, donde cada uno de ellos es una iteración que concluye en un incremento. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

1.3.5. Lenguaje UML

UML es un lenguaje de estandarización que permite modelar, documentar y desarrollar los elementos que forman parte de un sistema software orientado a objetos. Está compuesto por diversos elementos gráficos, los que combinados conforman los diagramas. Es la creación de Grady Booch, James Rumbaugh e Ivar Jacobson en sus esfuerzos para simplificar y consolidar el gran número de métodos de desarrollo orientado a objetos que habían surgido.

Hoy en día UML es una de las herramientas de gran importancia en el mundo del desarrollo de software y la naturaleza de su éxito radica principalmente en la naturalidad de su uso, pues se trata de un concepto nuevo ya que se emplean diagramas de distintos tipos para abordar el problema, de forma que la comprensión del problema en sí resulta mucho más "amigable" tanto para el desarrollador como para el cliente, pues éste también conocerá mucho mejor el producto que va a adquirir (3).

1.3.6. Herramienta Rational Rose

Las herramientas de modelado se vuelven cada día más importantes, a medida que los sistemas evolucionan y se vuelven más complejos, estas se tornan más significativas por los beneficios que proporcionan. El Rational Rose es una herramienta que cubre todo el ciclo de vida de un proyecto, desde la concepción y formalización del modelo, construcción de los componentes, transición a los usuarios hasta la certificación de las distintas fases y entregables. Nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

Rational Rose utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades. Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML. Proporciona mecanismos para realizar la denominada Ingeniería Inversa, a partir del código de un programa se puede obtener información de su diseño.

1.5. Plataforma Microsoft.Net

¿Entonces .NET es un lenguaje de programación?, No. Es mucho más que eso, Microsoft .NET es un conjunto de tecnologías de software, compuesto de varios lenguajes de programación que se ejecutan bajo el .NET Framework. Es además un entorno completamente orientado a objetos y que es capaz de ejecutarse bajo cualquier plataforma (4).

Son muchas las plataformas de desarrollo que existen actualmente con sus características particulares que hacen que sean escogidas por equipos de desarrollo en dependencias de las necesidades de estos, para el presente trabajo se ha escogido la Plataforma Microsoft .NET por decisión de la arquitectura del proyecto, como la más apropiada por su robustez, escalabilidad, especial énfasis en el desarrollo rápido de aplicaciones, la independencia de lenguaje y la transparencia a través de redes.

Provee los cimientos para la nueva generación de software. 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 (5)

Está centrado en la idea de software como servicio, para construir, instalar, consumir, integrar o agregar estos servicios para que puedan ser accedidos a través Internet. El usuario de Internet puede con un explorador de Internet no solamente acceder a contenido como texto, imágenes o sonido, también puede

hacer uso de servicios Web, que no son mas que los bloques de construcción o componentes sobre los cuales se basa el modelo de computación distribuida en Internet. La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo y sobre cualquier sistema operativo. Los puntos fuertes de la plataforma son:

Independencia de lenguaje: Todos los lenguajes que conformen con los estándares .NET, sin importar cual, podrán ínter operar entre sí de forma totalmente transparente, las clases podrán ser heredadas entre unos lenguajes y otros, y se podrá disfrutar de polimorfismo entre lenguajes.

Librería de clases común: Más de 4000 clases, objetos y métodos incluidos en la plataforma .NET están disponibles para todos los lenguajes.

Multiplataforma: Cuando un programa es compilado, no es compilado en un archivo ejecutable sino en un lenguaje intermedio llamado "Lenguaje Intermedio" (IL) el cual podrá ser ejecutado por el CLR (Common Language Runtime) en la plataforma en que el CLR esté disponible. El CLR compilará estos archivos IL nuevamente en código de máquina en un proceso que se conoce como JIT (en ingles "just in time") el cual se ejecutará cuando se requiera. Este proceso producirá código de máquina bien eficiente que se reutilizará si es que hubiera código que se repitiera, haciendo que los programas sean ejecutados muy eficientemente.

Windows Forms, Web Forms, Web Services: La plataforma .NET incluye un conjunto de clases especial para datos y XML que son la base de 3 tecnologías claves: Servicios Web (Web Services), Web Forms, y Windows Forms los cuales son poderosas herramientas para la creación de aplicaciones tanto para la plataforma como para el Web.

Estandarización: Una de las razones del éxito de la plataforma .NET ha sido por el proceso de estandarización que Microsoft ha seguido. Microsoft, en lugar de reservarse todos los derechos sobre el lenguaje y la plataforma, ha publicado las especificaciones del lenguaje y de la plataforma, que han sido posteriormente revisadas y ratificadas por la Asociación Europea de Fabricantes de Computadoras (ECMA).

1.5.1. Visual Studio .NET

Visual Studio .NET es un conjunto completo de aplicaciones para la creación tanto de aplicaciones Windows, aplicaciones Web como servicios Web para trabajo en equipo, permitiendo utilizar las eficaces

herramientas de desarrollo basado en componentes y otras tecnologías para simplificar el diseño, desarrollo e implementación en equipo de soluciones para la empresa. Este entorno está creado para poder manejar proyectos que usen más de un lenguaje a la vez, teniendo en cuenta la característica multilinguaje de la plataforma. Incluye los siguientes lenguajes de programación como Visual Basic.NET, Visual C++. NET que permite tanto escribir código adaptado a la plataforma .NET como código C++ nativo, Visual C#.NET es un nuevo lenguaje de programación orientado a objetos con el que se ha desarrollado parte de la plataforma .NET y por último Visual J#.NET que no es más que la adaptación de Visual J++ para la plataforma .NET. 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.

Dentro de las principales características de Visual Studio se puede encontrar que facilita el diseño de formularios Windows a través de una superficie de diseño gráfico que permite al programador crear una interfaz de usuario. Tiene herramientas para el diseño de formularios Windows y plantillas de aplicaciones Windows. Tiene herramientas para el diseño de formularios Web y plantillas de aplicaciones Web. Posee herramientas para el desarrollo de servicios Web basado en XML. Para el acceso a distintos tipos de datos cuenta con clases y componentes visuales. Cuenta con herramientas de depuración de código que puede utilizarse en todos los lenguajes de soportados por Visual Studio.

1.5.2. El lenguaje C#

Dentro de los diferentes lenguajes que dispone Visual Studio. Net por la arquitectura del proyecto se escogió C# por ser un lenguaje de programación moderno e innovador y de conocimiento de los integrantes del equipo. Además por ser orientado a objetos, simple, seguro, de alto rendimiento y con especial énfasis en Internet y sus estándares como XML.

C#, el nuevo lenguaje presentado en .NET Framework, procede de C++. Sin embargo, C# es un lenguaje orientado a objetos (desde el principio), moderno y seguro (6). Ha sido diseñado por Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

C# amplía las capacidades de C, C++, Visual Basic (VB) y Java para proporcionar un completo entorno de desarrollo en el que crear aplicaciones. C# mezcla la potencia de C, las capacidades de orientación a objetos de C++ y la interfaz gráfica de Visual Basic [C#, 2002]. (7)

C# contiene las herramientas para definir nuevas clases, sus métodos y propiedades, al igual que la sencilla habilidad para implementar encapsulación, herencia y polimorfismo, que son los tres pilares de la programación orientada a objetos. C# tiene un nuevo estilo de documentación XML que se incorpora a lo largo de la aplicación, lo que simplifica la documentación en línea de clases y métodos. C# soporta también interfaces, una forma de estipular los servicios requeridos de una clase. C# también provee soporte para estructuras, un concepto el cual ha cambiado significativamente desde C++. Provee características de componentes orientados, como propiedades, eventos y construcciones declaradas (también llamados atributos). La programación orientada a componentes es soportada por el CLR. C# provee soporte para acceder directamente a la memoria usando el estilo de punteros de C++ y mucho más.

1.5.3. ASP. Net

ASP.NET, la siguiente versión de ASP, es una estructura de programación que se utiliza en la creación de aplicaciones Web de clases corporativas. El acceso a estas aplicaciones es global, lo que conlleva una eficaz gestión de la información. No obstante, se puede afirmar que ASP.NET es mucho más que una simple versión de ASP debido a sus numerosas ventajas (1).

Para el desarrollo de la aplicación web se utilizará como estructura de programación para la creación de aplicaciones Web ASP.Net, la cual esta integrada a Visual Studio. Net, ofrece numerosas ventajas entre ellas está, que utiliza el entorno común de ejecución (CLR), que ofrece la estructura .NET y se encarga de la ejecución del código. En un primer momento, el código se compila en el lenguaje MSIL. A continuación durante la ejecución, dicho lenguaje se compila en código nativo. Únicamente los fragmentos de código que lo necesiten se compilarán en código original. Estas opciones consiguen mejorar el rendimiento general de las aplicaciones ASP.NET.

La definición estándar de una aplicación ASP.NET la describe como una combinación de archivos, páginas, manejadores, módulos y código ejecutable que puede invocarse desde un directorio virtual (y, opcionalmente, sus subdirectorios) de un servidor web. En otras palabras, el directorio virtual es la estructura de agrupación básica que delimita una aplicación (8).

Desde las aplicaciones ASP.NET se puede acceder a la totalidad de la biblioteca de clases .NET puede utilizar el lenguaje que mejor describa el tipo de funcionalidad que desee implementar, ya que ASP.NET no depende de un lenguaje en concreto. Los parámetros de configuración de la aplicación se suelen almacenar en formato XML, un formato de texto jerárquico que resulta muy sencillo de leer y escribir. Con este formato se pueden aplicar nuevos parámetros a una aplicación sin necesidad de utilizar las herramientas locales de administración.

Las aplicaciones ASP.NET utilizan una serie de esquemas de autorización y autenticación que pueden modificarse en función de las necesidades de seguridad concretas que exija la aplicación. Además de todas estas opciones, la estructura ASP.NET facilita los procesos de cambio de una aplicación ASP a otra.

1.6. Cristal Report para generar reportes

Visual Studio .NET incluye Crystal Reports como un mecanismo "oficial" de creación de reportes (9). El cual se utilizará para la generación de reportes, es producto creado en su esencia orientado al usuario final, para pueda crear sus propios informes sin necesidad de asistencia de un desarrollador. Única herramienta para generar reportes que se puede utilizar con Visual Studio .NET existen otras entre las que podemos mencionar: Active Reports de Data Dynamics, SQL Reporting Services, Microsoft Offices Automation, entre otros.

Al igual que desde el ambiente de desarrollo de Visual Studio, también podemos crear nuestros reportes desde el ambiente de desarrollo de Crystal Reports y luego ejecutarlo desde nuestros proyectos, Los Reportes desarrollados bajo el ambiente de Crystal Reports pueden ser compilados a .EXE para poder ser ejecutados Stand-Alone.

1.7. Visual SourceSafe

Para el control de versiones escogimos Microsoft Visual SourceSafe es un sistema de control de versiones en el nivel de archivos, que permite a muchos tipos de organizaciones trabajar en distintas versiones de un proyecto al mismo tiempo. Esta funcionalidad es especialmente ventajosa en un entorno de desarrollo de software, donde se usa para mantener versiones de código paralelas. Sin embargo, el producto también se puede utilizar para mantener archivos en cualquier otro tipo de equipo. Admite el desarrollo

multiplataforma al permitir la edición y el uso compartido de los datos. Se ha diseñado para controlar los problemas de seguimiento y portabilidad que implica mantener una base de control de código fuente, como una base de código de software, en varios sistemas operativos. Para los desarrolladores, Visual SourceSafe aloja código reutilizable u orientado a objetos. Asimismo, facilita el seguimiento de las aplicaciones que utilizan módulos de código concretos.

Tiene una serie de características ventajosas como control de versiones y uso compartido de archivos. También puede ampliar su funcionalidad si se escribe un complemento que sea compatible con el entorno de desarrollo integrado (IDE, Integrated Development Environment) del programa de terceros que ejecutará el paquete de software. Admite el desarrollo paralelo y las técnicas de desarrollo multiplataforma. Se admiten las operaciones de combinación de archivos en dos o tres direcciones e incluye varios mecanismos para resolver los conflictos resultantes de la combinación. Se puede integrar con toda facilidad en Visual Studio. Proporciona una serie de herramientas de mantenimiento de bases de datos muy útiles que permiten que éstas funcionen de forma eficaz y segura. Admite el almacenamiento y la restauración mediante asistentes de uso sencillo, así como varias utilidades de mantenimiento basadas en la línea de comandos.

1.8. Conclusiones

En este capítulo se trataron los temas acerca de las bancas telefónicas para lograr un mejor entendimiento de los procesos que se desenvuelven dentro de la misma y en especial sobre la línea del BBVA en su estrategia de mejorar la atención al cliente, así como el fomento de las bases tecnológicas y herramientas de las cuales se harán uso para responder a cada uno de los objetivos trazados al inicio del documento.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

Existen muchas empresas en el mundo donde han perfeccionado la atención al cliente haciendo usos de herramientas y tecnologías desarrolladas para brindar un servicio al cliente tanto a través de internet como vía telefónica. Actualmente en Cuba no se ha desarrollado ningunos de los métodos avanzados que existen para gestionar la atención al cliente, por lo que esta es la primera banca telefónica que se crea con el fin de automatizar la gestión del pago de los servicios públicos que se prestan a la población. En este capítulo veremos el problema y la situación problemática, el modelo de dominio, la propuesta del sistema y la comparación de la propuesta con las ya existentes. Se definirán los requisitos de software tanto los no funcionales como los funcionales. De estos últimos se obtendrán los casos de usos del sistema, los cuales serán descritos para una mayor comprensión de los mismos.

2.2. Problema y Situación Problemática

El modulo Servicio de Pago es el encargado de gestionar las funcionalidades principales de la banca en cuanto a atención al cliente, bancos asociados como servicios se refiere. Como modulo principal de Telebanca surge con la idea de mejorar la atención al cliente y comodidad para la empresa aliviando los trámites y demoras que hoy afectan a la población para pagar estos servicios, la que tiene que acudir a las oficinas de cobros y sucursales bancarias con la consiguiente pérdida de tiempo y afectación de la jornada laboral. En Cuba algunas empresas como las que brindan servicios de agua y electricidad, contratan personal para llegar a cada uno de los hogares para realizar el cobro de estos, esto es una tarea agotadora ya que tiene que caminar grandes tramos, además el riesgo que corren al tener que circular con gran cantidad de efectivo por las calles. Por otra parte cuando pasan los cobradores hay hogares donde todos trabajan y pierden la oportunidad ya que el recorrido de estos es en tiempo laboral y en ese caso tendrían que trasladarse hasta la oficina para pagar la factura de los servicios. A esta última vía se une la empresa de servicio telefónico, pero a pesar de ser la forma de pago más utilizada para que la población pague los servicios resulta ser un proceso dilatado, ya que en ocasiones puede acumularse personal teniendo que esperar en cola, provocando quejas y malestar por parte de la población por la perdida de tiempo. Además no existe un proceso rápido para la atención a las reclamaciones de la población con respecto a los problemas que surgen con el proceso de pago.

2.3. Objeto de Automatización

Para este trabajo no se partió de ningún proceso que existiera anteriormente, se llevo a cabo con la idea de desarrollar dentro de la banca telefónica una solución que facilitara la atención al cliente a solo una llamada telefónica. No se pretende automatizar el proceso de pago de los servicios tal y como es actualmente por parte de las empresas que lo prestan, sino crear una nueva forma de ejecutar el pago pero de cualquier servicio que se acoja al método de pago vía telefónica. Un modulo capaz de controlar la información de estos servicios, los bancos asociados y centrado principalmente en la atención al cliente. Dentro de los sistemas automatizados que existen y estarán vinculados con el modulo se encuentra la base de datos del banco la cual contiene información acerca del estado de cuenta de un cliente para llevar a cabo acciones como consulta de saldo o pago de un servicio, que implica modificaciones en el saldo de la cuenta. Para ello el banco implementará un servicio web, el cual tendrá su utilidad también en el intercambio de información en cuanto a algunos procesos que se realicen dentro de la banca telefónica, como es el caso del envío de reportes, conocer y modificar el saldo de un cliente, para las operaciones de consulta de saldo como pago de un servicio. También existirá un servidor FTP del cual se obtendrá un fichero con la deuda de cada cliente respecto a un servicio.

2.4. Propuesta del sistema

Se gestionarán los servicios que se prestan actualmente y se acogen a método de pago a través de la banca telefónica, a los cuales se les definirá si se acogen al pago simple o complejo. En caso de acogerse al pago complejo se guardará la frecuencia y la fecha de la primera vez en que se debe descargar la información del servidor FTP. Dentro de la información a manejar de cada servicio está el nivel de autenticación (cantidad de coordenadas de matriz o pin que se le pedirá al cliente para efectuar el pago). Un cliente puede pagar varios servicios y a la vez poder pagar el servicio a otros asociados si los tiene contratados. Para cada servicio se determinarán los datos los cuales se clasificarán en relevante, de pago y auxiliares, estos datos no son más que información adicional además de la que se conoce sobre la identificación del asociado y el importe de servicio. Estos datos podrán ser configurables. Se actualizan a cada uno de los bancos asociados con la información de los servicios y datos.

Cada cliente se autenticará haciendo uso de información que se encuentra en la tarjeta que se le entrega una vez firmado el contrato con la banca telefónica, la cual tiene un número, matriz con coordenadas y pin. Para pagar un servicio una vez autenticado el cliente se mostrarán los servicios contratados por este. Se buscará el nivel de autenticación del servicio seleccionado, luego se mostrarán los datos del servicio relacionado con el cliente que son tomados durante la contratación de este. Se tomarán los datos de pago, que son aquellos que son indispensables para ejecutar la transacción y estos varían en dependencia de la empresa. Si el servicio seleccionado se acoge al método complejo es necesario buscar los datos de pago para ese servicio que entrega la empresa en el fichero que se coloca en el FTP. Si se realizan todas las operaciones de forma satisfactoria se guardan los datos, se le informa al cliente si esta fue ejecutada o no. Los datos a guardar para cada transacción son: operadora, cliente, fecha, hora, conversación y traza de la transacción para el Banco (13 dígitos). Se le entrega al banco el número del comprobante para el cajero automático (6 dígitos, 2 estarán ocupados por las letras BT de banca telefónica y los restantes 4 son los últimos 4 dígitos del identificador de la transacción).

A petición del cliente se puede consultar el saldo que tiene la Cuenta de Ahorro asociados a Tarjetas Débito RED que esté vinculada con la tarjeta de autenticación de la Banca Telefónica además de permitir que el cliente pueda deshabilitar la tarjeta de Banca Telefónica, los motivos en la mayoría de los casos es por pérdida o porque el cliente no pueda o no desee utilizar más esta o porque desee una nueva. El sistema gestionará las reclamaciones de los clientes respecto a transacciones anteriores que estos hayan efectuados, para el cual se le pedirá y verificará la traza de transacción que se le entrega una vez efectuada esta, si toda la operación es satisfactoria se guardarán los datos de la reclamación.

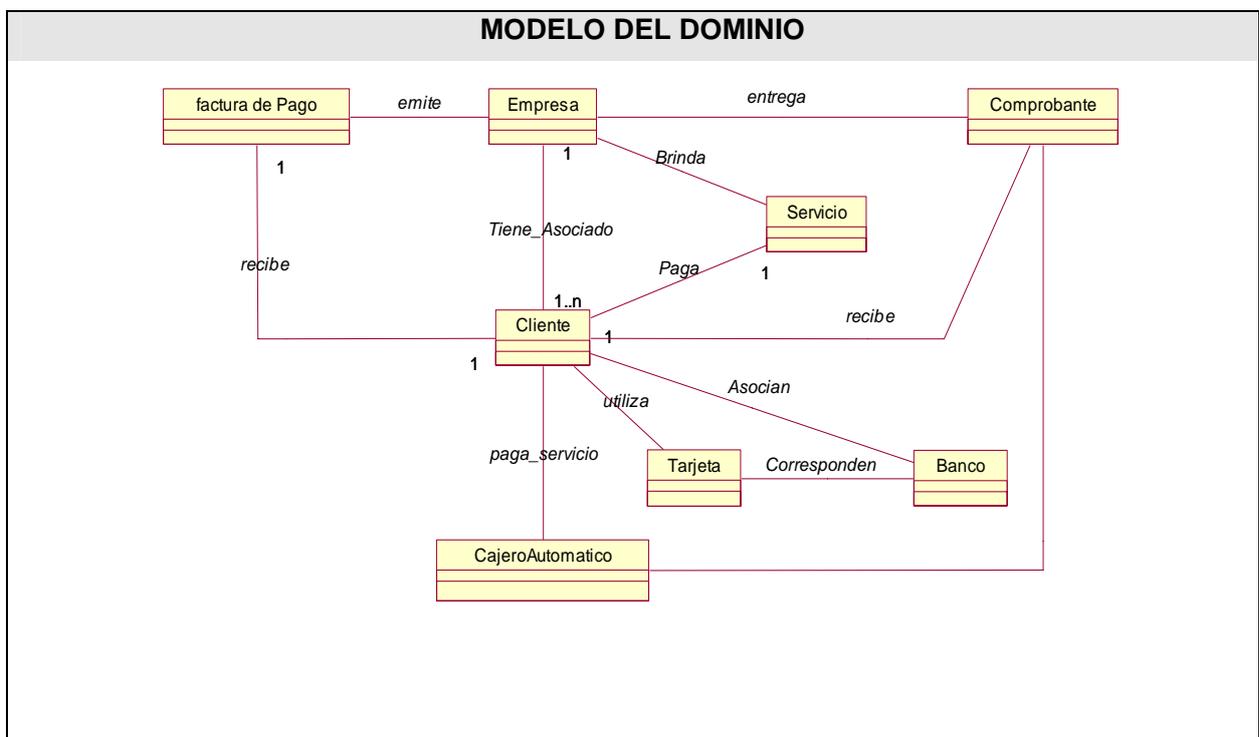
Se gestionará la información de los bancos asociados a la banca telefónica en la que los clientes tienen sus cuentas bancarias y hacia la cual deben dirigirse las solicitudes de esta. Se realizarán reportes de forma automática como auxiliar para enviar a cada uno de los bancos para notificar sobre cada una de las transacciones dentro de la banca y que esta relacionada con los clientes que pertenecen a cada uno de estos. Dentro de la información que se plasmarán en los reportes está además de las transacciones las reclamaciones, tarjetas deshabilitadas, solicitudes de nuevas tarjetas, entre otras.

Esta propuesta es similar a las soluciones que algunos bancos como BBVA, Caixa de España, RIPLEY y BICE de Chile disponen. A pesar de que no se pueda realizar un análisis crítico de dichas soluciones, ya

que información al respecto no se revela mucho, nada que vaya mas allá de la oferta de los servicios que prestan estas a través de la banca telefónica. Podemos identificar a simple vista la diferencia entre esas soluciones y la propuesta de este trabajo y esta radica en que su solución esta orientada a brindar a través de la banca telefónica servicios sobre los procesos inclinados mas hacia las operaciones bancarias como consulta de saldo, transferencias, etc. y no en si a enfocado la gestión del pago de servicios, objetivo en el cual esta centrado el presente trabajo.

2.5. Modelo del dominio

Se decidió desarrollar un modelo de dominio para captar los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, puesto que no existía un negocio actual que se pudiera modelar, ya que no se automatizará el proceso de gestión de un servicio determinado para una empresa, sino que se modelarán los elementos comunes que tienen las empresas en cuanto a la gestión del pago de servicios y adaptarlos a las condiciones de la banca telefónica.



2.6. Especificación de los requisitos de software

La correcta especificación de requisitos determina en gran medida el éxito de un proyecto de desarrollo de software. A continuación se verá la especificación de los requisitos funcionales como no funcionales correspondientes al módulo Servicio de Pago de la banca telefónica.

2.6.1. Requerimientos Funcionales

1. Autenticar a los clientes según el nivel de autenticación del servicio.
 - a) Verificar número de la tarjeta.
 - b) Verificar número PIN de la tarjeta con dos dígitos.
 - c) Verificar una o más coordenadas de la matriz de la tarjeta.

Seguimiento: CU_AutenticarCliente, CU_VerificarTarjeta.

2. Ejecutar el pago de un servicio.
 - a) Mostrar servicios contratados por el cliente.
 - b) Verificar nivel de autenticación del servicio.
 - c) Mostrar los datos del servicio.
 - d) Captar los datos de pago del servicio.
 - e) Buscar datos para el pago a la empresa.
 - f) Ejecutar la transacción.

Seguimiento: CU_Pagar Servicio.

3. Gestionar la Configuración de los Servicios.
 - a) Definir si el servicio se acoge al pago simple o complejo.
 - b) Determinar los datos de los Servicios.
 - c) Determinar el nivel de Autenticación de los Servicios.

Seguimiento: CU_Configurar Servicios.

4. Permitir insertar modificar y eliminar la información de los datos de un servicio.

Seguimiento: CU_Configurar Datos.

5. Permitir actualizar la información de los servicios y de los datos asociados.

Seguimiento: CU_Actualizar Información.

6. Consultar Saldo.

Seguimiento: CU_Consultar Saldo.

7. Poner en caliente las tarjetas de Banca Telefónica.

Seguimiento: CU_Tarjeta Caliente.

8. Iniciar Reclamaciones.

- a) Verificar la transacción a reclamar.
- b) Guardar los datos de la reclamación.

Seguimiento: CU_Iniciar Reclamación.

9. Descargar la información del Servicio Complejo.

Seguimiento CU_Información del Servicio Complejo.

10. Gestionar Banco.

- a) Insertar Banco.
- b) Modificar Banco.
- c) Eliminar Banco.
- d) Actualizar Sucursales.

Seguimiento: CU_Gestionar Banco, CU_Actualizar Sucursales.

11. Permitir realizar conciliaciones con los bancos.

Seguimiento: CU_Conciliaciones Automáticas, CU_Conciliaciones Auxiliares.

12. Permitir realizar e imprimir reportes de las transacciones.

Seguimiento: CU_ReporteTransacciones.

2.6.2. Requerimientos no Funcionales

13. Requerimientos de apariencia o interfaz externa.

La herramienta propuesta será usada por personas que no necesariamente tienen conocimientos profundos de informática, por lo que la interfaz debe ser amigable y fácil de usar, de manera que no sea una dificultad para los usuarios el uso de esta.

14. Requerimientos de Usabilidad

La herramienta será utilizada por operadores que no necesariamente tendrán experiencia en el uso de la computadora, a estos se les dará un adiestramiento básico en el uso de la computadora y fundamentalmente en el uso de la aplicación. Estas personas tendrán un nivel de acceso en la aplicación, en la cual realizarán solamente las acciones que tienen asignadas.

15. Requerimientos de Rendimiento

Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a las páginas. La eficiencia del producto estará 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. El tiempo de respuesta de la operadora al cliente debe estar alrededor de los 2.5 segundos.

16. Requerimientos de Soporte

Para garantizar el soporte de esta herramienta, se documentará 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.

17. Requerimientos de Portabilidad

La herramienta propuesta podrá ser usada bajo cualquier sistema operativo; el Servidor Web bajo 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 usará como herramienta de desarrollo Visual Studio .Net y Gestor de Base de Datos SQL Server 2003.

18. Requerimientos de Seguridad

La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación. Deberán estar encriptadas las matrices que se asociarán a las distintas tarjetas de

autenticación de los clientes. La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados inconsistentes. La aplicación deberá 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.

19. Requerimientos de Software

En la computadora de los usuarios sólo se requiere de navegador para Internet o Intranet, bajo cualquier sistema operativo; en el Servidor Web, 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 2003.

20. Requerimientos de Hardware

En el cliente se requiere 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 MB de velocidad.

21. 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.

2.7. Diagrama de casos de uso

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. Para mantener una estructura que sea fácil de comprender se decidió dividir cada uno de los casos de uso que dan soporte a un determinado proceso por paquetes. El paquete Gestión de Servicios, Datos y Bancos, recoge los procesos vinculados al control de la gestión de los bancos asociados a la

banca telefónica y los servicios que se acogen al pago a través de la misma. Atención a Clientes es el paquete que contiene los casos de usos que corresponden a procesos mediante los cuales intervienen los clientes como es el caso del pago de un servicio. Por último Reportes y Conciliaciones como su nombre lo indica acoge a aquellos procesos que se encargan de la generación de reportes de las transacciones que se realizan en la banca telefónica.

DIAGRAMA DE CASOS DE USO – GESTION DE SERVICIOS, DATOS Y BANCOS

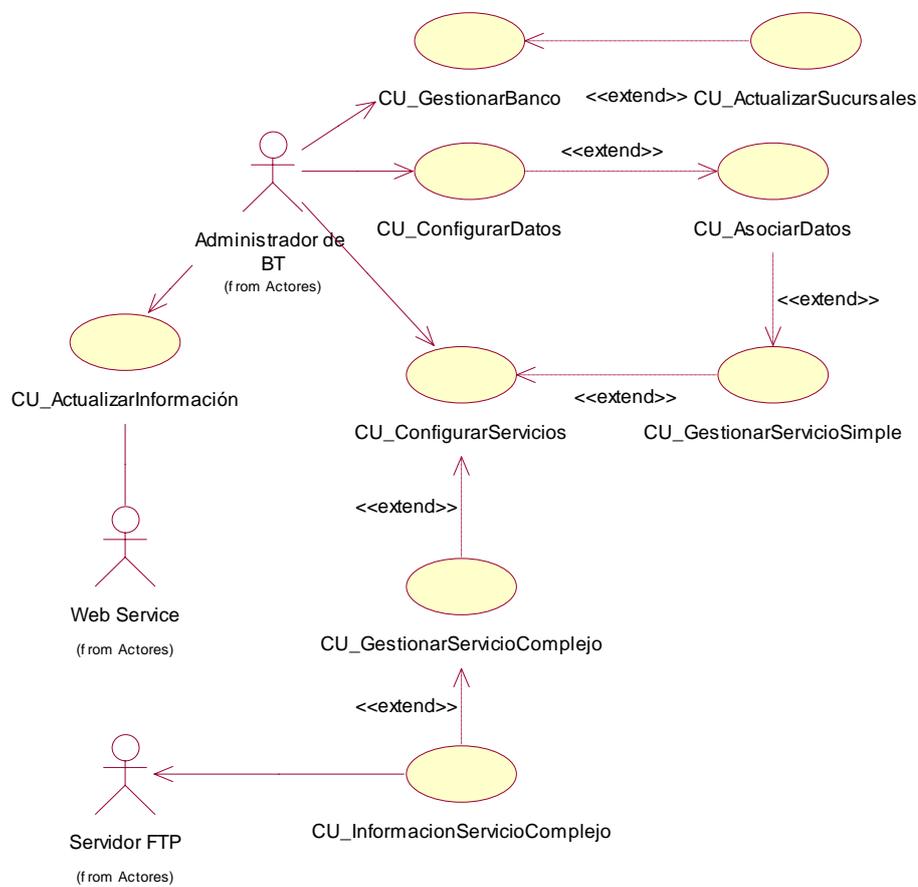


DIAGRAMA DE CASOS DE USO – ATENCION A CLIENTES

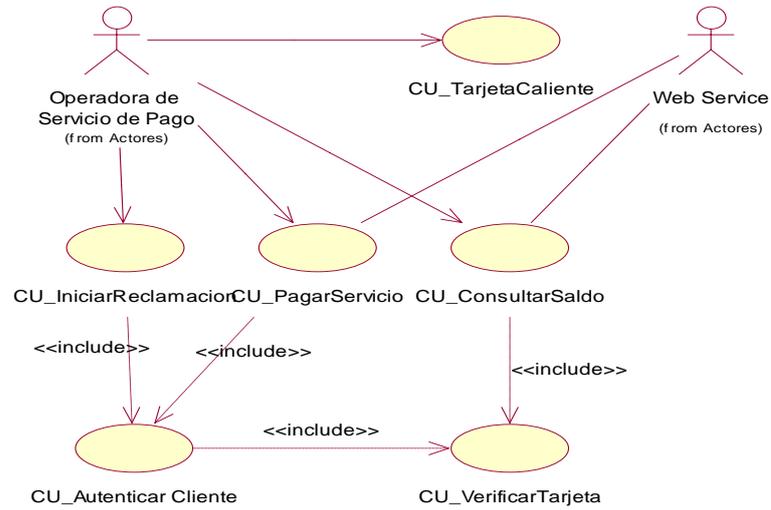
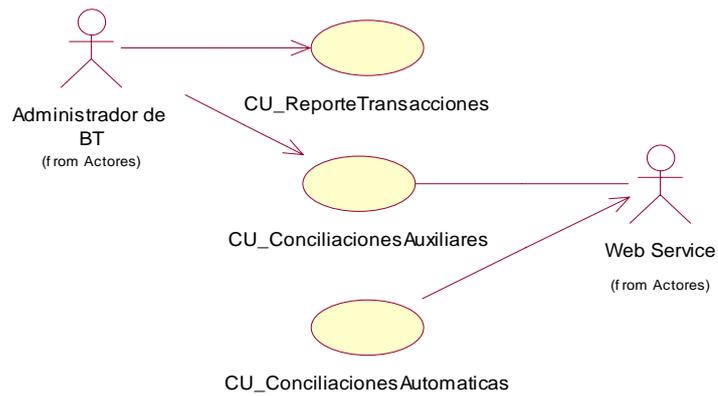


DIAGRAMA DE CASOS DE USO – REPORTES Y CONCILIACIONES



2.8. Definición de los casos de uso

2.8.1. Definición de los actores

Actor	Descripción
Operadora de Servicio de Pago	Persona encargada de interactuar y ejecutar las siguientes actividades a solicitud del cliente: pagar un servicio, consultar saldo, iniciar reclamación y poner en caliente la tarjeta.
Administrador de BT	Persona encargada de administrar los servicios que brinda la banca telefónica y los bancos asociados y de enviar las conciliaciones a los Bancos.
Servidor FTP	Una máquina donde se almacenan los ficheros que contiene la información de los asociados a las empresas que se acogen al pago complejo.
Web Service	Aplicación a través de la cual interactuamos con la Base de Datos del Banco.

2.8.2. Listado de casos de uso

A continuación se verán las especificaciones de aquellos casos de uso que se determinaron desde el punto de vista arquitectónico como significativos.

22. CU_AutenticarCliente

Caso de uso	CU_AutenticarCliente.
Actores:	Operadora de Servicio de Pago
Resumen:	El caso de uso se inicia cuando la Operadora de Servicio de Pago desea autenticar a un cliente para que este pueda acceder a los servicios de la banca telefónica. Para dicho proceso se le pedirá información relacionada con la tarjeta que se le entrega una vez sean clientes de la banca telefónica.
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. La Operadora de Servicio de Pago solicita identificar un nuevo cliente.	2. El sistema solicita el número de la tarjeta de identificación del cliente para la Banca Telefónica.
3. La Operadora de Servicio de Pago introduce el número de la tarjeta.	
4. La Operadora de Servicio de Pago selecciona aceptar.	5. El sistema valida la entrada de datos.
	6. El sistema verifica el número de la tarjeta. Ver CU_VerificarTarjeta.
	7. El sistema muestra el nombre y apellidos del cliente asociado a la tarjeta.el nombre y los apellidos del cliente autenticado.
	8. El sistema solicita introducir dos dígitos del PIN.
9. La Operadora de Servicio de Pago Introduce dos dígitos del PIN.	
10. La Operadora de Servicio de Pago selecciona aceptar.	11. El sistema valida la entrada de los datos.
	12. El sistema verifica que los dos dígitos del PIN asociado a la tarjeta estén correctos.El sistema verifica los dos dígitos del El sistema verifica los dos dígitos.
	13. El sistema solicita una coordenada de la matriz.
14. La Operadora de Servicio de Pago Introduce la coordenada.	
15. La Operadora de Servicio de Pago selecciona aceptar.	16. El sistema valida la entrada de los datos.
	17. El sistema descripta la matriz asociada a la tarjeta. Esta operación se hace solo al primer

	intento.
	18. El sistema verifica que la coordenada pertenezca a la matriz asociada a la tarjeta.
	19. Finaliza el caso de uso.
Flujos Alterativos	
Acción del Actor	Respuesta del Sistema
Acción 4: Si la Operadora de Servicio de Pago selecciona cancelar cuando va a introducir el número de la tarjeta.	4.1 Finalizar el caso de uso.
Acción 5: Si el número de la tarjeta entrada no es válido.	5.1 El sistema muestra un asterisco en los campos con errores.
	5.2 El sistema muestra el mensaje "Entrada de datos no válida".
5.3 La Operadora de Servicio de Pago acepta el mensaje.	5.4 Pasa a la acción 2.
Acción 6 Si el número de la tarjeta no pertenece a la Banca Telefónica.	6.1 El sistema muestra mensaje "El número de la tarjeta es erróneo o no se encuentra registrada en nuestra Banca Telefónica". El sistema verifica que no es el tercer intento de entrada del número de la tarjeta. Pasa a la acción 2. En caso contrario muestra el mensaje "Ha realizado el numero máximo de intentos " y finaliza el caso de uso.
Acción 10: Si la Operadora de Servicio de Pago selecciona cancelar cuando va a introducir los dos dígitos del PIN.	10.1 Finalizar el caso de uso.
Acción 11: Si los dos dígitos del PIN entrados no son validos.	11.1 El sistema muestra un asterisco en los campos con errores.

	11.2 El sistema muestra el mensaje "Entrada de datos no válida".
11.3 La Operadora de Servicio de Pago acepta el mensaje.	11.4 Pasa a la acción 8.
Acción 12: Si los dígitos del PIN no son los correctos.	12.1 El sistema muestra el mensaje "Los dígitos del PIN son erróneos".
12.2 La Operadora de Servicio de Pago acepta el mensaje.	12.3 El sistema verifica que no sea el tercer intento de entrada de pin y pasa a la acción 8. En caso contrario muestra el mensaje "Ha realizado el número máximo de intentos" y finaliza el caso de uso
Acción 15: Si la Operadora de Servicio de Pago selecciona cancelar cuando va a introducir las coordenadas.	15.1 Finalizar el caso de uso.
Acción 16: Si la coordenada entrada no es valida.	16.1 El sistema muestra un asterisco en el campo con error.
	16.2 El sistema muestra el mensaje "Entrada de datos no válida".
16.3 La Operadora de Servicio de Pago acepta el mensaje.	16.4 Pasa a la acción 13.
Acción 18: Si la coordenada es incorrecta.	18.1 El sistema muestra el mensaje "La coordenada es errónea".
18.2 La Operadora de Servicio de Pago acepta el mensaje.	18.3 El sistema verifica que no sea el tercer intento de entrada de coordenada y pasa a la acción 13. En caso contrario muestra el mensaje "Ha realizado el número máximo de intentos" y finaliza el caso de uso

23. CU_VerificarTarjeta

Caso de uso	CU_VerificarTarjeta.
Actores:	...
Resumen:	El caso de uso se inicia cuando el sistema verifica el estado en que se encuentra la tarjeta de un cliente dado dentro de la banca telefónica. Dentro de los estados están deshabilitada, pedida, creada o no válida.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1. El sistema busca el estado en que se encuentra la tarjeta dentro de la banca telefónica.
	2. El sistema devuelve que la tarjeta se encuentra en estado habilitada.
	3. Finaliza el caso de uso.
Flujos Alterativos	
Acción 2 F1 Si no existe conexión con la base de datos.	2 F1.1 El sistema muestra el mensaje "Error de conexión con la base de datos". Pasar a la acción 3.
Acción 2 F2 Si la tarjeta no se encuentra en la banca telefónica.	2 F2.1El sistema muestra el mensaje " El número de la tarjeta no se encuentra registrada en la Banca Telefónica ". Pasar a la acción 3
Acción 2 F3 Si la tarjeta se encuentra en estado deshabilitada.	2 F3.1El sistema muestra el mensaje " La tarjeta esta deshabilitada ". Pasar a la acción 3
Acción 2 F4 Si la tarjeta se encuentra en estado pedida.	2 F4.1El sistema muestra el mensaje " La tarjeta esta pedida ". Pasar a la acción 3
Acción 2 F5 Si la tarjeta se encuentra en estado creada.	2 F5.1El sistema muestra el mensaje " La tarjeta esta pedida ". Pasar a la acción 3

24. CU_PagarServicio

Caso de uso	CU_PagarServicio.
Actores:	Operadora de Servicio de Pago, Web Services
Resumen:	El caso de uso se inicia cuando la Operadora de Servicio de Pago desea acceder al pago de servicio, cada servicio tiene un nivel de autenticación definido, el pago puede ser simple o complejo en dependencia de la configuración. Un cliente puede pagar uno o más servicios a varios asociados si los tiene contratados.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. La Operadora de Servicio de Pago solicita identificar un nuevo cliente.	2. Ver CU_AutenticarCliente
	3. Una vez autenticado el sistema solicita los servicios asociados al número de la tarjeta al Web Services.
4. El Web Services envía los servicios contratados por ese cliente y los datos relevantes .	5. El sistema muestra el o los servicios contratados por el cliente.
6. La Operadora de Servicio de Pago selecciona un servicio.	
7. La Operadora de Servicio pago solicita aceptar.	8. El sistema verifica el nivel de autenticación del servicio.
	9. El Sistema verifica el tipo de servicio según el codificador del servicio: a) Si simple. Ver sección Simple. b) Si complejo. Ver sección Complejo.
	10. El sistema envía los datos de la transacción y los datos de pago al Web Services.
11. El Web Services ejecuta la transacción.	
12. El Web Services envía la traza	13. El sistema guarda los datos de la

correspondiente y el codificador de error en cero (identificador de la transacción para el banco, 13 dígitos).	transacción, los datos de pago y los datos de la traza de la acción: Fecha, Identificación del usuario y descripción="Pago de Servicio"+ fecha + datos del pago
	14. El sistema muestra el mensaje: "La transacción ha sido realizada satisfactoriamente" y la traza "BT"+los últimos 4 id de la transacción.
15. La Operadora de Servicio de Pago informa los resultados de la transacción y la traza al cliente.	
16. La Operadora de Servicio de Pago selecciona aceptar.	17. Pasar a la Acción 9.
Sección Simple	
	18. El sistema muestra los datos relevantes asociados correspondiente al servicio seleccionado.
19. La Operadora de Servicio de Pago selecciona el asociado a pagar por solicitud del cliente.	
20. La Operadora de Servicio de Pago acepta.	21. El Sistema muestra un formulario con los datos de pago definidos por el servicio y la opción de "Actualizar".
22. La Operadora de Servicio de Pago selecciona los datos uno a uno.	
23. La Operadora de Servicio de Pago introduce el valor a introducir por cada dato de pago.	
24. La Operadora de Servicio de Pago presiona el botón actualizar.	25. El sistema verifica el dato entrado.

26. La Operadora de Servicio de Pago selecciona aceptar.	27. El sistema verifica que se actualizaron todos los datos de pago. Pasar a la acción 26.
Sección Complejo	
	28. El Sistema busca dado el CI el asociado al servicio.
	29. El sistema muestra los datos del servicio relacionado con el cliente.
30. La Operadora de Servicio de Pago selecciona el asociado a pagar por solicitud del cliente.	31. El sistema muestra un formulario con los datos de pago y las opciones Sí y No.
32. La Operadora de Servicio de Pago selecciona que si va a pagar.	33. Pasar a la acción 26.
Flujos Alterativos	
Acción 3: Si no se puede solicitar los servicios y datos relevantes del cliente porque no hay conexión con el Web Services.	3.1 El sistema muestra el mensaje "Error de conexión con el Web Services".
3.2 La Operadora de Servicio de Pago acepta le mensaje.	3.3 Finalizar el caso de uso.
Acción 7: Si la Operadora de Servicio de Pago selecciona cancelar cuando va seleccionar el servicio a pagar.	7.1 Finalizar el caso de uso.
Acción 8 F1: Si el servicio requiere un nivel de autenticación superior al entrado.	8 F1.1 El sistema solicita los datos adicionales que necesita el servicio y pasa a la acción 9.
Acción 8 F2: Si no se puede verificar el nivel de autenticación que requiere el servicio porque no hay conexión con la base de datos.	8 F2.1 El sistema muestra el mensaje "Error de conexión con la base de datos".
8 F2.2 La Operadora de Servicio de Pago acepta el mensaje.	8 F2.3 Finalizar el caso de uso.

Acción 9: Si no se puede verificar el tipo de servicio porque no hay conexión con la base de datos.	9.1 El sistema muestra el mensaje "Error de conexión con la base de datos".
9.2 La operadora de servicio de pago acepta el mensaje.	9.3 Finaliza el caso de uso.
Acción 10: Si no se puede enviar los datos de la transacción porque no hay conexión con el Web Service.	10.1 El sistema muestra el mensaje "Error de conexión con el Web Services".
10.2 La Operadora de Servicio de Pago acepta el mensaje.	10.2 Finalizar el caso de uso.
Acción 12: Si el codificador de error enviado por el Web Services es distinto de cero.	12.1 El sistema muestra el mensaje con la descripción asociada al error.
	12.2 El sistema guarda los datos de la transacción y en la traza guarda "E" los tres dígitos del error.
Acción 12.1: Si el Sistema no puede tener la descripción del error porque no hay conexión con la base de datos.	12.1.1 El sistema muestra el mensaje "Error en la transacción". Finalizar el caso de uso.
Acción 12.2: Si el Sistema no puede guardar los datos de la transacción porque no hay conexión con la base de datos.	12.2.1 El sistema guarda los datos de la transacción en el fichero de emergencias. Pasa a la acción 14.
Acción 13: Si el sistema no pudo guardar la transacción porque no hay conexión con la base de datos.	13.1 El sistema guarda los datos de la transacción en el fichero de emergencia.
Acción 20: Si la Operadora de Servicio de Pago selecciona cancelar cuando va a seleccionar el asociado a pagar.	21.1 Pasa a la acción 5.
Acción 21: Si el sistema no pudo obtener los datos para el pago simple porque no hay conexión con la	21.1 El sistema muestra el mensaje "Error de conexión con la base de datos".

base de datos.	
	21.2 Termina el caso de uso.
Acción 25 F1 Si el dato de pago es erróneo.	25 F1.1 El sistema muestra un asterisco en los campos con errores.
	25 F1.2 El sistema muestra el mensaje " Entrada de datos no válida".
25 F1.3 La Operadora de Servicio de Pago acepta el mensaje.	25 F1.4 Pasa a la acción 21.
Acción 25 F2 Si no se ha seleccionado el dato de pago.	25 F2.1 El sistema muestra el mensaje " Debe seleccionar un dato "
25 F2.2 La operadora de Servicio de Pago acepta el mensaje.	25 F2.3 El sistema pasa a la acción 21.
Acción 26: Si la Operadora de Servicio de Pago selecciona cancelar cuando va a introducir los datos de pago.	26.1 Pasar a la acción 21.
Acción 28: Si el Sistema no puede captar los datos que entrega la empresa porque no hay conexión con la base de datos.	28.1 Termina el caso de uso
Acción 29 Si no hay pagos relacionados con el cliente.	29.1 El sistema muestra el mensaje "No hay pagos relacionados al cliente ". Pasa a la acción 28.
Acción 32: Si la Operadora de Servicio de Pago selecciona cancelar cuando va a seleccionar el asociado a pagar.	32.1 Pasa a la acción 21.
Acción 34: Si la Operadora de Servicio de Pago selecciona que no va a pagar.	34.1 Pasa a la acción 46.

25. CU_ConfigurarDatos

Caso de Uso:	CU_ConfigurarDatos.	
Actores:	Administrador de BT.	
Resumen:	El caso de uso se inicia cuando el Administrador de BT solicita la configuración de los Datos. Permite insertar, modificar o eliminar Datos. La información a configurar para cada Dato es: nombre, el tipo (que se refiere a char, string, etc.), y el tamaño.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Administrador de BT solicita la configuración de Datos.	2. El Sistema muestra un listado con los Datos que no han sido asociados a los servicios y las opciones para configurar los Datos: insertar, modificar y eliminar Datos.	
3. El Administrador de BT selecciona: a) Opción de insertar nuevos Datos, Ver Sección Insertar Datos. b) Opción de modificar un Dato. Ver Sección Modificar Datos. c) Opción de eliminar un Dato. Ver sección Eliminar Datos.		
Sección Insertar Datos		
	4. El Sistema muestra un formulario para insertar la información referente a un Dato: la cual es: a) Nombre del Dato (obligatorio). b) Tipo de Dato (obligatorio). c) Tamaño del Dato (obligatorio).	
5. El Administrador de BT inserta la información del nuevo Dato.		

6. El Administrador de BT solicita guardar la información.	7. El Sistema valida las entradas correspondientes.
	8. El Sistema valida que no exista ningún Dato con el mismo nombre.
	9. El Sistema solicita confirmación de guardar la información. “¿Confirma que desea guardar la información?”
10. El Administrador de BT confirma guardar la información.	11. El Sistema guarda la información satisfactoriamente.
	12. El Sistema guarda los datos de la traza de la acción: Fecha, Identificación del usuario y descripción=“Configuración de los Datos + ID del dato”
	13. El Sistema muestra el mensaje “La información ha sido guardada satisfactoriamente”.
14. La Operadora de Servicio de Pago acepta el mensaje.	15. El Sistema pasa a la acción 2.
Sección Modificar Datos.	
	16. El Sistema muestra la información del Dato seleccionado (nombre, tipo de Dato, y tamaño). a) Nombre del Dato. b) Tipo c) Tamaño del Dato.
17. El Administrador de BT modifica la información deseada referente al Dato.	
18. El Administrador de BT solicita guardar los cambios.	19. El Sistema valida las entradas correspondientes.
	20. El Sistema valida que no exista ningún Dato

	con el mismo nombre.
	21. El Sistema solicita confirmación de guardar la información. “¿Confirma que desea guardar la información?”
22. El Administrador de BT confirma guardar la información.	23. El Sistema guarda la información satisfactoriamente.
	24. El Sistema guarda los datos de la traza de la acción: Fecha, Identificación del usuario y descripción=“Configuración de los Datos + ID del dato”
	25. El Sistema muestra el mensaje “La información ha sido guardada satisfactoriamente”.
26. La Operadora de Servicio de Pago acepta el mensaje.	27. El Sistema pasa a la acción 2.
Sección Eliminar Datos.	
	28. El Sistema solicita confirmación de eliminar el Dato seleccionado. “¿Confirma que desea eliminar el Dato seleccionado?”
29. El Administrador de BT confirma eliminar el Dato.	30. El Sistema elimina el Dato satisfactoriamente.
	31. El Sistema guarda los datos de la traza de la acción: Fecha, Identificación del usuario y descripción=“Configuración de los Datos + ID del dato”
	32. El Sistema muestra el mensaje “La información ha sido eliminada satisfactoriamente”.
33. La Operadora de Servicio de Pago acepta el	34. El Sistema pasa a la acción 2.

mensaje.	
Flujos Alternativos	
Acción 2 F1: Si no hay Datos.	2 F1.1 Pasar a la Acción 4
Acción 2 F2: Si no se puede cargar la lista de Datos porque no hay conexión con la base de datos.	2 F2.1 El Sistema muestra el mensaje. "Error de conexión con la base de Datos".
2 F2.2 El Administrador de BT acepta el mensaje.	2 F2.3 Finalizar el caso de uso.
Acción 3: Si el Administrador de BT escoge la opción cancelar de configurar Datos.	3.1 Finalizar el caso de uso.
Acción 6: Si el Administrador de BT escoge la opción cancelar la opción de insertar datos.	6.1. Pasar a la Acción 2.
Acción 7: Si no son válidos los valores de entrada.	7.1 El Sistema muestra el mensaje "Entrada de datos no válida".
7.2 El Administrador de BT acepta el mensaje.	7.3 Pasar a la acción 4.
Acción 8 F1: Si no se puede verificar si existe el Dato porque no hay conexión con la base de datos.	8 F1.1 El Sistema muestra el mensaje. "Error de conexión con la base de Datos".
8 F1.2 El Administrador de BT acepta el mensaje.	8 F1.3 Pasar a la acción 4.
Acción 8 F2: Si existe un Dato con el mismo nombre.	8 F2.1 El Sistema muestra un mensaje. " Existe otro Dato con las mismas características".
8 F2.2 El Administrador de BT acepta el mensaje.	8 F2.3 Pasar a la acción 4.
Acción 10: Si el Administrador de BT escoge cancelar en la opción de confirmar guardar la información.	10.1 Pasar a la acción 4.
Acción 11: Si no se puede guardar el nuevo Dato porque no hay conexión con la base de datos.	11.1 El Sistema muestra el mensaje. "Error de conexión con la base de Datos".

11.2 El Administrador de BT acepta el mensaje.	11.3 Pasar a la acción 4.
Acción 16: Si no se pueden cargar los datos del Dato seleccionado porque no hay conexión con la base de datos.	16.1 Muestra el mensaje. "Error de conexión con la base de Datos".
16.2 El Administrador de BT acepta el mensaje.	16.3 Pasar a la acción 2.
Acción 18: Si el Administrador de BT escoge cancelar en la opción de guardar los cambios.	18.1 Se pasa a la acción 2.
Acción 19: Si no son válidos los valores de entrada.	19.1 El Sistema muestra el mensaje "Entrada de datos no válida".
19.2 El Administrador de BT acepta el mensaje.	19.3 Pasar a la acción 16.
Acción 20 F1: Si no se puede verificar si existe otro Dato con las mismas características porque no hay conexión con la base de datos.	20 F1.1 El Sistema muestra el mensaje. "Error de conexión con la base de datos".
20 F1.2 El Administrador de BT acepta el mensaje.	20 F1.3 Pasar a la acción 16.
Acción 20 F2: Si existe un Dato con el mismo nombre.	20 F2.1 El Sistema muestra un mensaje. " Existe otro Dato con las mismas características".
20.2 El Administrador de BT acepta el mensaje.	20.3 Pasar a la acción 16.
Acción 22: Si el Administrador de BT escoge cancelar en la opción de confirmar guardar la información.	22.1 Pasar a la acción 16.
Acción 23: Si no se puede actualizar la información de los datos porque no hay conexión con la base de datos.	23.1 Muestra el mensaje. "Error de conexión con la base de Datos".
23.2 El Administrador de BT acepta el mensaje. Se pasa a la acción 15:	23.3 Pasar a la acción 16.
Acción 29: Si el Administrador de BT cancela la	29.1 Pasar a la acción 2.

opción de confirmar de la eliminación del dato.	
Acción 30: Si no se puede eliminar el dato porque no hay conexión con la base de Datos.	30.1 Muestra el mensaje. "Error de conexión con la base de Datos".
30.2 El Administrador de BT acepta el mensaje.	30.3 Pasar a la acción 2.

26. CU_AsociarDatos

Caso de Uso:	CU_AsociarDatos
Actores:	Administrador de BT.
Resumen:	El caso de uso se inicia cuando el Administrador de BT una vez configurado un servicio simple desee asociarle a este los datos existentes.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
Sección Asociar Datos.	
	1. El Sistema muestra un formulario con <ul style="list-style-type: none"> a) La lista de datos existentes b) La lista de datos asociados c) Las opciones de agregar y quitar datos d) La opción de configurar los datos. Ver CU_ConfigurarDatos.
2. El Administrador de BT puede quitar y agregar al servicio los datos que necesite, clasificándolos en: <ul style="list-style-type: none"> a) Relevante. b) Pago. c) Importe 	
3. El Administrador de BT solicita aceptar.	4. El sistema guarda la información de los datos asociados en memoria.

	5. Finaliza el caso de uso.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
Acción 1 F1 Si no existen datos para asociar.	1 F1.1 Pasar a Configurar datos. Ver CU_ Configurar Datos.
Acción 1 F2 Si el Sistema no puede mostrar los datos asociados porque no hay conexión con la base de datos.	1 F2.1 El Sistema muestra el mensaje "Error de conexión con la base de datos."
1 F2.2 El Administrador de BT acepta el mensaje.	1 F2.3 Finaliza el caso de uso.
Acción 3 Si el Administrador de BT cancela los cambios en la asociación de datos.	3.1 Finaliza el caso de uso.

27. CU_ConfigurarServicios

Caso de Uso:	CU_ ConfigurarServicios.
Actores:	Administrador de BT.
Resumen:	El caso de uso se inicia cuando el Administrador de BT solicita insertar, modificar o eliminar un Servicios. El tipo de servicio será simple o complejo. En dependencia de este varían las acciones a realizar.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador de BT solicita la configuración de servicios.	2. El Sistema muestra la lista con todos los servicios disponibles, las opciones para configurar los servicios: insertar, modificar y eliminar.
3. El Administrador de BT selecciona servicio simple: d) Solicita insertar un nuevo servicio. Ver CU_GestionarServicioSimple. Sección Insertar	4. El Sistema guarda los datos de la traza de la acción: Fecha, Identificación del usuario y descripción="Configuración de los Servicios + ID del Servicio"

<p>Servicios.</p> <p>e) Selecciona un servicio a modificar. Ver CU_GestionarServicioSimple. Sección Modificar Servicios.</p> <p>f) Selecciona un servicio a eliminar. Ver Sección Eliminar Servicios.</p>	
<p>Sección Eliminar Servicios.</p>	
<p>Flujo Normal de Eventos</p>	
	<p>5. El Sistema solicita confirmación de eliminar el servicio seleccionado. “¿Confirma que desea eliminar los servicios?”</p>
<p>6. El Administrador de BT confirma eliminar los servicios</p>	<p>7. El Sistema elimina los servicios satisfactoriamente.</p>
	<p>8. El Sistema guarda los datos de la traza de la acción: Fecha, Identificación del usuario y descripción=“Configuración de los Datos + ID del dato”</p>
	<p>9. El sistema muestra el mensaje “El servicio fue eliminado satisfactoriamente”</p>
<p>Flujos Alternativos</p>	
<p>Acción 3 F1 Si el administrador selecciona la opción cancelar.</p>	<p>3 F1.1 Finaliza el caso de uso.</p>
<p>Acción 3 F2 Si el administrador selecciona un servicio complejo.</p>	<p>3 F2.1 Si administrador de BT selecciona:</p> <p>a) Solicita insertar un nuevo servicio. Ver CU_GestionarServicioComplejo. Sección Insertar Servicios.</p> <p>b) Selecciona un servicio a modificar. Ver CU_GestionarServicioComplejo. Sección Modificar Servicios.</p>

Acción 6: Si el Administrador de BT escoge cancelar en la opción de confirmar eliminar el servicio.	6.1 Pasar a la Acción 2.
Acción 7: Si no se puede eliminar correctamente porque no hay conexión con la base de datos.	7.1 El Sistema muestra el mensaje "Error de conexión con la base de datos."
7.2 El Administrador de BT acepta el mensaje.	7.3 Pasar a la Acción 2.

28. CU_GestionarServicioSimple

Caso de Uso:	CU_GestionarServicioSimple
Actores:	Administrador de BT.
Resumen:	El caso de uso se inicia cuando el Administrador de BT solicita la Configuración de Servicios. Permite insertar un nuevo servicio al que se le define el nivel de autenticación, así como su tipo, el nombre, el estado y los datos asociados. El nivel de autenticación de los servicios será: CI del cliente, PIN, Número de la Tarjeta y/o Coordenada de la Matriz.
Flujo Normal de Eventos	
Sección Insertar Servicio Simple.	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. El Sistema muestra un formulario para insertar los datos del nuevo servicio, los cuales son: <ol style="list-style-type: none"> d) Nombre del servicio. e) Id del servicio. f) Estado del servicio activo o no activo. g) Niveles de autenticación <ol style="list-style-type: none"> 1. Si o no se autentica por número de la tarjeta. 2. Si o no se autentica por CI. 3. Si o no se autentica por PIN.

	<p>4. Si se autentica por las coordenadas de la matriz el número de coordenadas.</p> <p>h) La opción de Asociar los Datos Existentes. Ver CU_AsociarDatos.</p>
2. El Administrador de BT llena el formulario del nuevo servicio.	
3. El Administrador de BT solicita insertar el nuevo servicio simple.	4. El Sistema valida las entradas correspondientes.
	5. El Sistema valida que el servicio a insertar sea único en la base de datos.
	6. El Sistema solicita confirmación de guardar la información. “¿Confirma que desea guardar la información?”
7. El Administrador de BT confirma guardar la información.	8. El Sistema guarda la información satisfactoriamente.
	9. El Sistema muestra el mensaje “La información ha sido guardada satisfactoriamente”.
10. El Administrador de BT acepta el mensaje.	11. El Sistema pasa a la acción 2.
Sección Modificar Servicio Simple.	
Flujo Normal de Eventos	
	<p>12. El Sistema muestra el formulario con la información para modificar los datos del servicio simple escogido, los cuales son:</p> <p>a) Estado del servicio (activo o no activo).</p> <p>b) Niveles de autenticación:</p> <ol style="list-style-type: none"> 1. Si o no se autentica por número de la tarjeta. 2. Si se autentica o no por CI. 3. Si o no se autentica por PIN. 4. Si se autentica por las coordenadas de la matriz,

	<p>el número de coordenadas.</p> <p>c) La opción de Asociar los Datos Existentes. Ver CU_AsociarDatos.</p>
13. El Administrador de BT modifica los datos del servicio.	
14. El Administrador de BT solicita modificar la información.	15. El Sistema valida las entradas correspondientes.
	16. El Sistema solicita confirmación de guardar la información. "¿Confirma que desea guardar la información?"
17. El Administrador de BT confirma guardar la información.	18. El Sistema guarda la información satisfactoriamente.
	19. El Sistema muestra el mensaje "La información ha sido guardada satisfactoriamente".
20. El Administrador de BT acepta el mensaje.	21. Finaliza el caso de uso.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
Acción 3: Si el Administrador de BT escoge cancelar la opción de insertar el nuevo servicio simple.	3.1 Pasar a la Acción 2.
Acción 4: Si la información entrada para el servicio nuevo están incorrectas.	4.1 El Sistema muestra el mensaje "Entrada de datos no válida".
4.2 El Administrador de BT acepta el mensaje.	4.3 Pasar a la Acción 1.
Acción 5 F1: Si el servicio a insertar ya existe.	5 F1.1 El Sistema muestra el mensaje "El servicio a insertar ya existe."
5 F1.2 El Administrador de BT acepta el mensaje.	5 F1.3 Pasar a la Acción 1.
Acción 5 F2: Si no se puede verificar si el servicio	5 F2.1 El Sistema muestra el mensaje "Error de

a insertar ya existe porque no hay conexión con la base de datos.	conexión con la base de datos.”
5 F2.2 El Administrador de BT acepta el mensaje.	5 F2.3 Pasar a la Acción 1.
Acción 7: Si el Administrador de BT cancela la confirmación guardar el nuevo servicio.	7.1 Pasar a la Acción 1.
Acción 8: Si el servicio no puede ser guardado porque no hay conexión con la base de datos.	8.1 El Sistema muestra el mensaje “Error de conexión con la base de datos.”
8.2 El Administrador de BT acepta el mensaje.	8.3 Pasar a la Acción 1.
Acción 12 Si no se pueden mostrar las informaciones del servicio seleccionado porque no hay conexión con la base de datos.	12.1 El Sistema muestra el mensaje “Error de conexión con la base de datos.”
12.2 El Administrador de BT acepta el mensaje.	12.3 Finaliza el caso de uso.
Acción 14: Si el Administrador de BT cancela modificar la información del servicio.	14.1 Finaliza el caso de uso.
Acción 15: Si no son válidos los valores de entrada.	15.1 El Sistema muestra el mensaje “Entrada de datos no válida.”
15.2 El Administrador de BT acepta el mensaje.	15.3 Pasar a la Acción 11.
Acción 17: Si el Administrador de BT cancela la confirmación guardar las modificaciones en la configuración del Servicio.	17.1 Pasar a la Acción 12.
Acción 18: Si el servicio no puede ser guardado correctamente porque no hay conexión con la base de datos.	18.1 El Sistema muestra el mensaje “Error de conexión con la base de datos.”
18.2 El Administrador de BT acepta el mensaje.	18.3 Pasar a la Acción 12.

29. CU_GestionarServicioComplejo

Caso de Uso:	CU_GestionarServicioComplejo.
Actores:	Administrador de BT.
Resumen:	El caso de uso se inicia cuando el Administrador de BT solicita la configurar un servicio complejo. Permite insertar un nuevo servicio al que se le define el nivel de autenticación, así como su tipo, el nombre y el estado. El nivel de autenticación de los servicios será: CI del cliente, PIN, Número de la Tarjeta y/o Coordinada de la Matriz. Tendrá asociado una frecuencia y la fecha de próxima actualización de los ficheros en el servidor FTP.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
Sección Insertar Servicio Complejo.	
	<ol style="list-style-type: none"> 1. El Sistema muestra un formulario para insertar los datos del nuevo servicio, los cuales son: <ol style="list-style-type: none"> a) Nombre del servicio. b) Id del servicio. c) Estado (activo y no activo). d) Niveles de autenticación: <ol style="list-style-type: none"> 1. Si o no se autentica por número de la tarjeta. 2. Si o no se autentica por CI. 3. Si o no se autentica por PIN. 4. Si se autentica por las coordenadas de la matriz, el número de coordenadas. d) Frecuencia de descarga. e) Si es servicio con asociados. f) Fecha de la última descarga de FTP. g) La opción de descargar ahora. Ver

	CU_InformacionServicioComplejo.
2. El Administrador de BT llena el formulario del nuevo servicio.	
3. El Administrador de BT solicita insertar la información.	4. El Sistema valida las entradas correspondientes.
	5. El Sistema valida que el servicio a insertar sea único en la base de datos.
	6. El Sistema solicita confirmación de guardar la información. “¿Confirma que desea guardar la información?”
7. El Administrador de BT confirma guardar la información.	8. El Sistema guarda la información satisfactoriamente y los Datos Asociados, los cuales son: a) Relevante (Descriptor, Id del asociado) b) Pago (Importe, Informativo)
	9. El Sistema muestra el mensaje “La información ha sido guardada satisfactoriamente”.
10. El Administrador de BT acepta el mensaje.	11. Finaliza el caso de uso.
Sección Modificar Servicio Complejo.	
	12. El Sistema muestra un formulario para modificar los datos del nuevo servicio, los cuales son: a) Niveles de autenticación: 1. Si o no se autentica por número de la tarjeta. 2. Si o no se autentica por CI. 3. Si o no se autentica por PIN.

<p>13. El Administrador de BT modifica las informaciones que necesite del servicio.</p>	<p>4. Si se autentica por las coordenadas de la matriz el número de coordenadas.</p> <p>b) Estado (activa, no activa).</p> <p>c) Frecuencia.</p> <p>d) Fecha de la última descarga.</p>
<p>14. El Administrador de BT solicita guardar la información.</p>	<p>15. El Sistema valida las entradas correspondientes.</p>
	<p>16. El Sistema solicita confirmación de guardar la información. “¿Confirma que desea guardar la información?”</p>
<p>17. El Administrador de BT confirma guardar la información.</p>	<p>18. El Sistema guarda la información satisfactoriamente.</p>
	<p>19. El Sistema muestra el mensaje “La información ha sido guardada satisfactoriamente”.</p>
<p>20. El Administrador de BT acepta el mensaje.</p>	<p>21. Finaliza el caso de uso.</p>
<p>Flujos Alternativos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>Acción 3: Si el Administrador de BT cancela insertar la información del servicio complejo.</p>	<p>3.1 Finaliza el caso de uso.</p>
<p>Acción 4: Si la información a insertar está incorrecta.</p>	<p>4.1 El Sistema muestra el mensaje “Entrada de datos no válida”.</p>
<p>4.2 El Administrador de BT acepta el mensaje.</p>	<p>4.3 Pasar a la Acción 1.</p>

Acción 5 F1: Si el servicio a insertar ya existe.	5 F1.1 El Sistema muestra el mensaje “El servicio a insertar ya existe.”
5 F1.2 El Administrador de BT acepta el mensaje.	5 F1.3 Pasar a la Acción 31.
Acción 5 F2: Si no se puede verificar si el servicio a insertar ya existe porque no hay conexión con la base de datos.	5 F2.1 El Sistema muestra el mensaje “Error de conexión con la base de datos.”
5 F2.2 El Administrador de BT acepta el mensaje.	5 F2.3 Pasar a la Acción 1.
Acción 7: Si el Administrador de BT cancela la confirmación guardar el nuevo servicio.	7.1 Pasar a la Acción 1.
Acción 8: Si el servicio no puede ser guardado correctamente porque no hay conexión con la base de datos.	8.1 El Sistema muestra el mensaje “Error de conexión con la base de datos.”
8.2 El Administrador de BT acepta el mensaje.	8.3 Pasar a la Acción 1.
Acción 12: Si no se pueden mostrar la información del servicio porque no hay conexión con la base de datos está deshabilitada.	12.1 El Sistema muestra el mensaje “Error de conexión con la base de datos.”
12.2 El Administrador de BT acepta el mensaje.	12.3 Finaliza el caso de uso.

Acción 14: Si el Administrador de BT cancela hacer las modificaciones al servicio.	14.1 Pasar a la Acción 2.
Acción 15: Si la información modificada esta incorrecta.	15.1 El Sistema muestra el mensaje “Entrada de datos no válida”.
15.2 El Administrador de BT acepta el mensaje.	15.3 Se pasa a la Acción 12.
Acción 17: Si el Administrador de BT cancela la confirmación de guardar los cambios realizados al servicio.	17.1 Pasar a la Acción 12.
Acción 18: Si el servicio modificado no puede ser guardado correctamente porque no hay conexión con la base de datos.	18.1 El Sistema muestra el mensaje “Error de conexión con la base de datos.”
18.2 El Administrador de BT acepta el mensaje.	18.3 Finaliza el caso de uso.

30. CU_InformacionServicioComplejo

Actores:	Servidor FTP, Administrador de BT
Resumen:	El caso de uso se inicia a una hora previamente acordada el Sistema procede a descargar el fichero con los datos de los servicios complejos que les corresponda. La descarga puede hacerse manual a petición del administrador de BT
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el administrador de BT selecciona descargar el fichero o cuando es la hora de inicio de peticiones. a. Descarga manual por Administrador de	

<p>BT. Ver sección "Tratar fichero". b. Descarga automática. Pasar a la acción 2</p>	
	<p>2. A una hora previamente escogida para comenzar las peticiones el Sistema verifica en la configuración de los servicios si hay ficheros de servicios complejos atrasados o para descargar ese día. Se procede a actualizar los datos de cada servicio complejo y para ello se ejecutan las acciones descritas en la Sección Tratar Fichero.</p>
Sección Tratar Fichero	
	<p>3. El Sistema se conecta al FTP.</p>
	<p>4. El Sistema le solicita al FTP el listado de los ficheros que posee.</p>
<p>5. El FTP entrega el listado solicitado.</p>	<p>6. El Sistema verifica que este el fichero a descargar.</p>
	<p>7. El Sistema solicita el fichero del servicio complejo.</p>
<p>8. El FTP entrega el fichero solicitado.</p>	<p>9. El Sistema actualiza todos los datos del servicio descargándolos a la tabla correspondiente verifica siempre que el formato de los ficheros sea admisible.</p>
	<p>10. El Sistema guarda en las notificaciones el mensaje " La descarga del fichero del servicio X fue satisfactoria ".</p>

	11. El Sistema a partir de la frecuencia calcula la fecha de la próxima descarga.
	12. El Sistema actualiza en la configuración del servicio complejo la fecha de la próxima descarga.
Flujos Alternativos	
Acción del Actor	Respuesta del Sistema
Acción 2 F1: Si no hay servicios complejos con fecha de descarga para ese día o atrasadas.	F1 2.1 Termina el Caso de uso
Acción 2 F2: Si el sistema no puede verificar en la configuración de los servicios si hay ficheros de servicios complejos atrasados o para descargar ese día porque no existe conexión con la base de datos.	F2 2.1 El Sistema guarda en el fichero de emergencia el mensaje "No pudo verificar si hay ficheros de servicios complejos atrasados o para descargar porque no existe conexión con la base de datos".
	F2 2.2 Termina el caso de uso.
Acción 3: Si el Sistema no puede conectarse con el Servidor FTP.	3.1 El Sistema espera 20 minutos.
	3.2 El Sistema incrementa el contador.
	3.3 El Sistema verifica si el contador llegó a 3.
	3.4 Pasar a la Acción 3.
Acción 3.3: Si el contador llegó a 3.	3.3.1 El Sistema guarda en las notificaciones el mensaje "El Sistema no puede conectarse con el Servidor FTP".
Acción 3.3.1: Si el Sistema no puede guardar en las notificaciones el mensaje "El Sistema no puede conectarse con el Servidor FTP", porque no existe conexión con la base de datos.	3.3.1.1 El Sistema guarda en el fichero de emergencia el mensaje "El Sistema no puede conectarse con el Servidor FTP".

Acción 4: Si el Sistema no puede solicitar el listado al FTP porque no hay conexión con el Servidor FTP.	4.1 El Sistema espera 15 minutos.
	4.2 El Sistema incrementa el contador.
	4.3 El Sistema verifica si el contador llegó a 3.
	4.4 Pasar a la Acción 4.
Acción 4.3: Si el contador llegó a 3.	4.3.1 El Sistema guarda en las notificaciones el mensaje “El Sistema no puede conectarse con el Servidor FTP”.
Acción 4.3.1: Si el Sistema no puede guardar en las notificaciones el mensaje “El Sistema no puede conectarse con el Servidor FTP” porque no existe conexión con la base de datos.	4.3.1.1 El Sistema guarda en el fichero de emergencia el mensaje “El Sistema no puede conectarse con el Servidor FTP” .
Acción 6: Si no está el fichero en el FTP.	6.1 El Sistema guarda en las notificaciones el mensaje “No se encuentra en el FTP el fichero del servicio X”.
Acción 6.1: Si el Sistema no puede guardar en las notificaciones el mensaje “No se encuentra en el FTP el fichero del servicio X” porque no existe conexión con la base de datos.	6.1.1 El Sistema guarda en el fichero de emergencia el mensaje “No se encuentra en el FTP el fichero del servicio X”.
Acción 7: Si el Sistema no puede solicitar el fichero al FTP porque no hay conexión con el Servidor FTP.	7.1 El Sistema espera 15 minutos.
	7.2 El Sistema incrementa el contador.
	7.3 El Sistema verifica si el contador llegó a 3.
	7.4 Pasar a la Acción 7.
Acción 7.3: Si el contador llegó a 3.	7.3.1 El Sistema guarda en las notificaciones el mensaje “El Sistema no puede conectarse con el

	Servidor FTP”.
Acción 7.3.1: Si el Sistema no puede guardar en las notificaciones el mensaje “El Sistema no puede conectarse con el Servidor FTP” porque no existe conexión con la base de datos.	7.3.1.1 El Sistema guarda en el fichero de emergencia el mensaje “El Sistema no puede conectarse con el Servidor FTP”.
Acción 9 F1: Si el fichero no está en el formato requerido.	9 F1.1 El Sistema guarda en las notificaciones el mensaje “El fichero del servicio X no está en el formato requerido”.
Acción 9 F1.1: Si el Sistema no puede guardar en las notificaciones el mensaje “El fichero del servicio X no está en el formato requerido” porque no existe conexión con la base de datos.	9 F1.1.1 El Sistema guarda en el fichero de emergencia el mensaje “No se encuentra en el FTP el fichero del servicio X”.
Acción 9 F2: Si el sistema no puede descargar el fichero porque no existe conexión con la base de datos.	9 F2.1 El Sistema guarda en el fichero de emergencia el mensaje “No se descargó la información de los ficheros por error de conexión con la base de datos”.
Acción 9 F3: Si el fichero está en blanco.	9 F3. 1 El sistema elimina los elementos almacenados en la base de datos referente al servicio.
	9 F3.2 El sistema guarda en las notificaciones el mensaje: “El fichero del Servicio X está en blanco.”.
Acción 9 F3.1: Si el Sistema no puede eliminar la información del servicio porque no existe conexión con la Base de Datos.	9 F3.1.1 El sistema guarda en el fichero de emergencia el mensaje “El fichero del Servicio X esta en blanco y no se puedo eliminar la información referente al servicio.”.
Acción 9 F3.2: Si el Sistema no puede guardar en las notificaciones el mensaje “El fichero del	9 F3.2.1 El sistema guarda en las notificaciones el mensaje “El fichero del Servicio X está en blanco.”.

Servicio X está en blanco.” porque no existe conexión con la Base de Datos.	
Acción 10: Si el Sistema no puede guardar en las notificaciones el mensaje “ La descarga del fichero del servicio X fue satisfactoria” porque no existe conexión con la base de datos.	10.1 El Sistema guarda en las notificaciones el mensaje “La descarga del fichero del servicio X fue satisfactoria ”.
Acción 12: Si el sistema no puede actualizar la configuración del servicio complejo porque no existe conexión con la base de datos.	12.1 El Sistema guarda en las notificaciones el mensaje “No se pudo actualizar la fecha de la próxima descarga porque no hay conexión con la base de datos ”.

2.9. Conclusiones

En este capítulo se describieron las particularidades del sistema, delimitadas en requisitos funcionales y no funcionales que responderán a las características del producto final. Se delimitaron cada uno de los procesos en casos de usos y se describieron aquellos que se consideraron de gran importancia desde el punto de vista arquitectónico.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

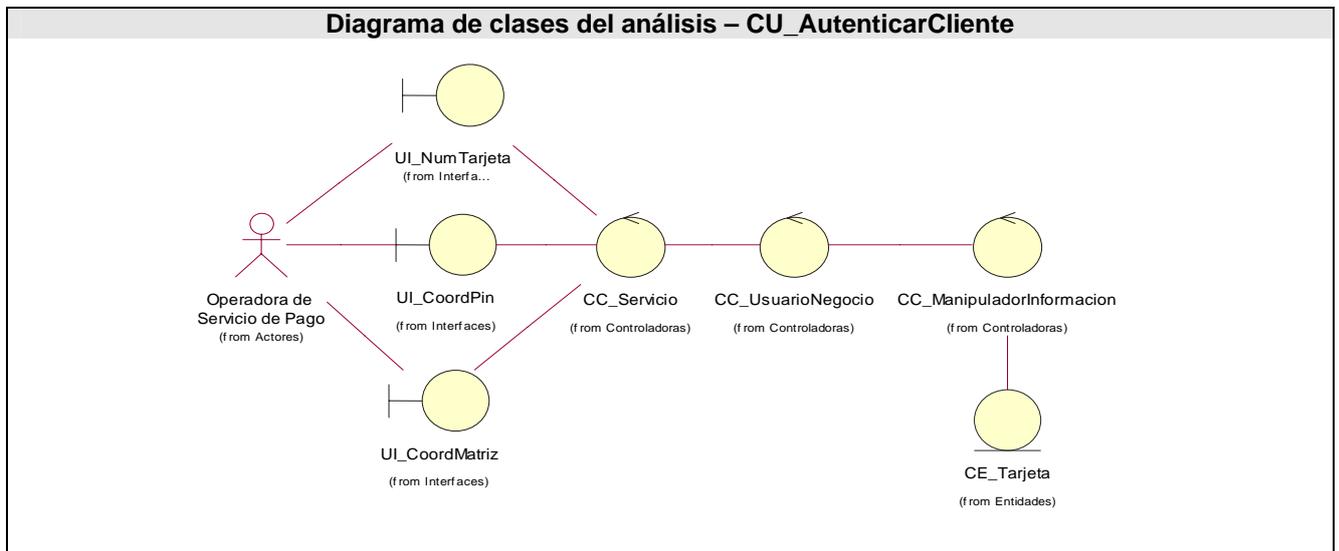
3.1. Introducción

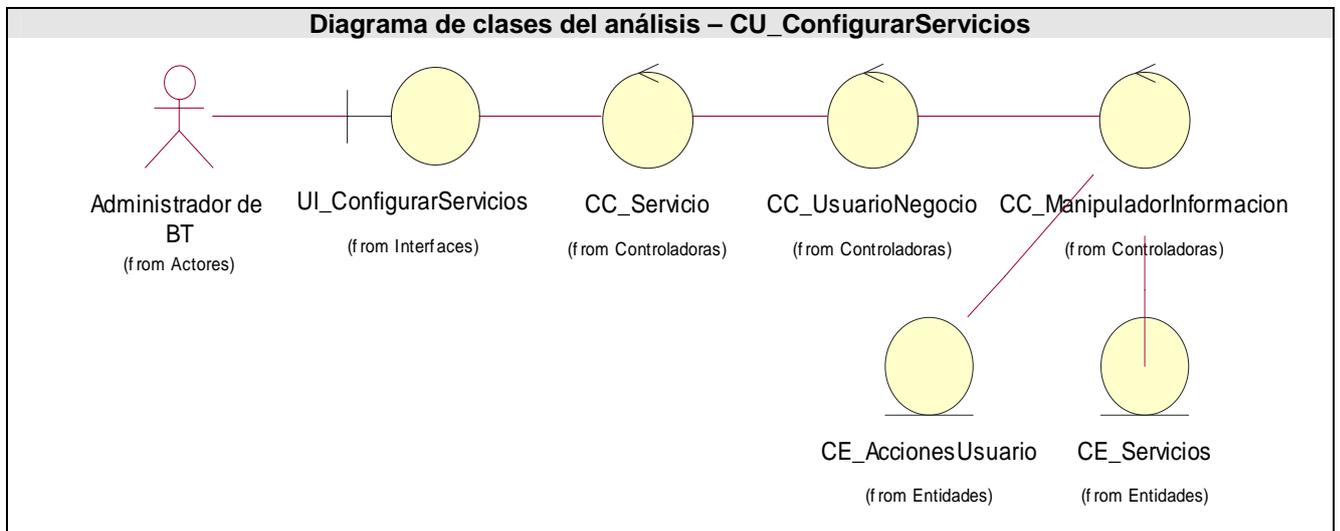
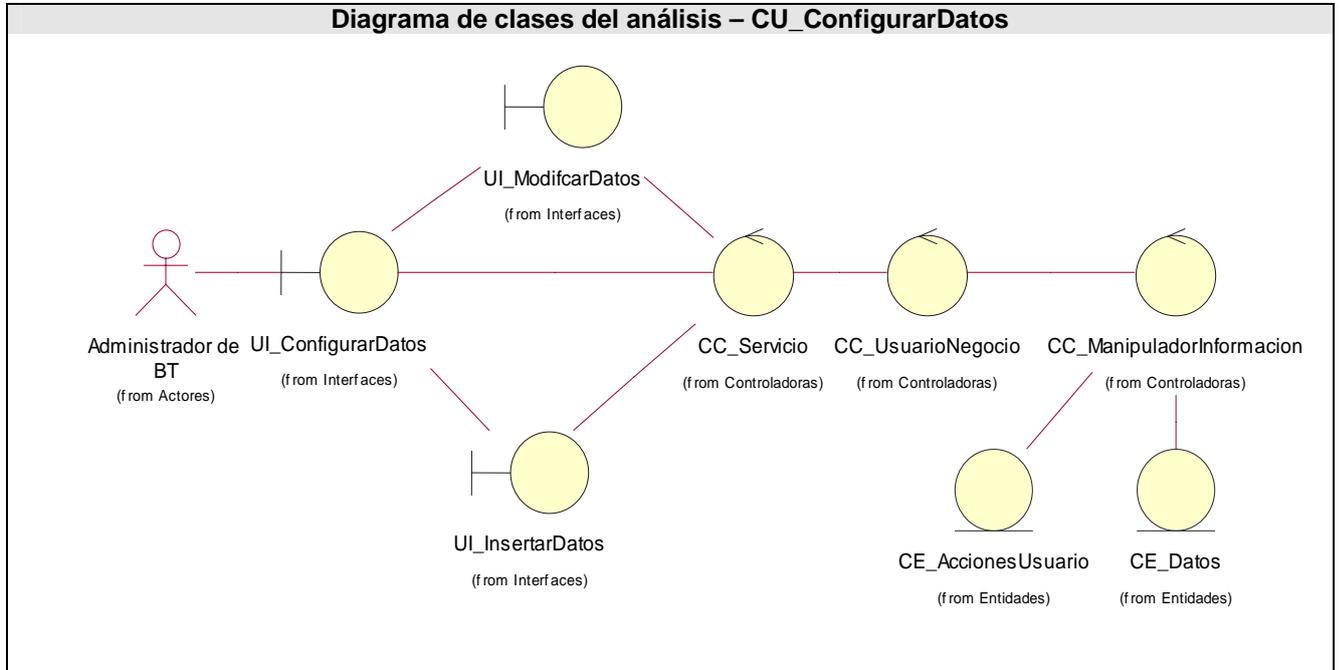
En este capítulo se traducirán los requisitos a una especificación que describe cómo implementar el sistema. Se realizará el análisis para obtener una visión del sistema que se preocupa en ver que se hace desde el punto de vista de los requisitos funcionales, obteniendo un modelo de clases del análisis por cada caso de uso significativo. Por otro lado se realizará el diseño para refinar los modelos obtenidos del análisis partiendo también de los requisitos no funcionales y ver como cumple el sistema sus objetivos, representado en diagrama de clases del diseño y diagrama de iteración respectivamente.

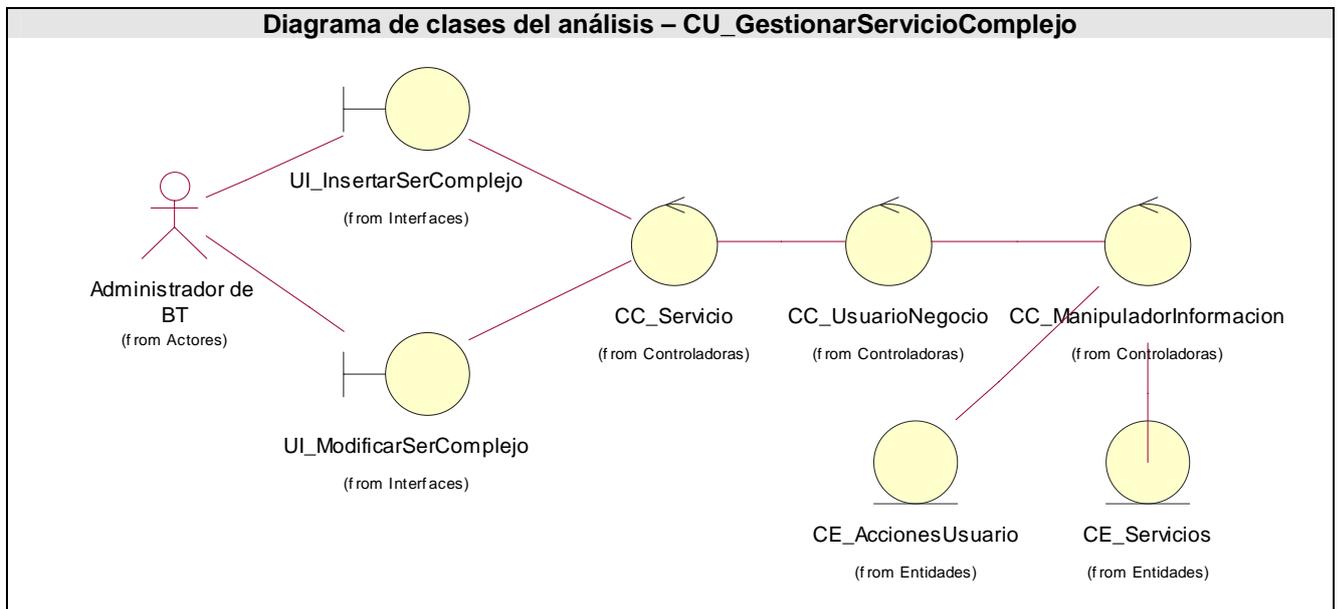
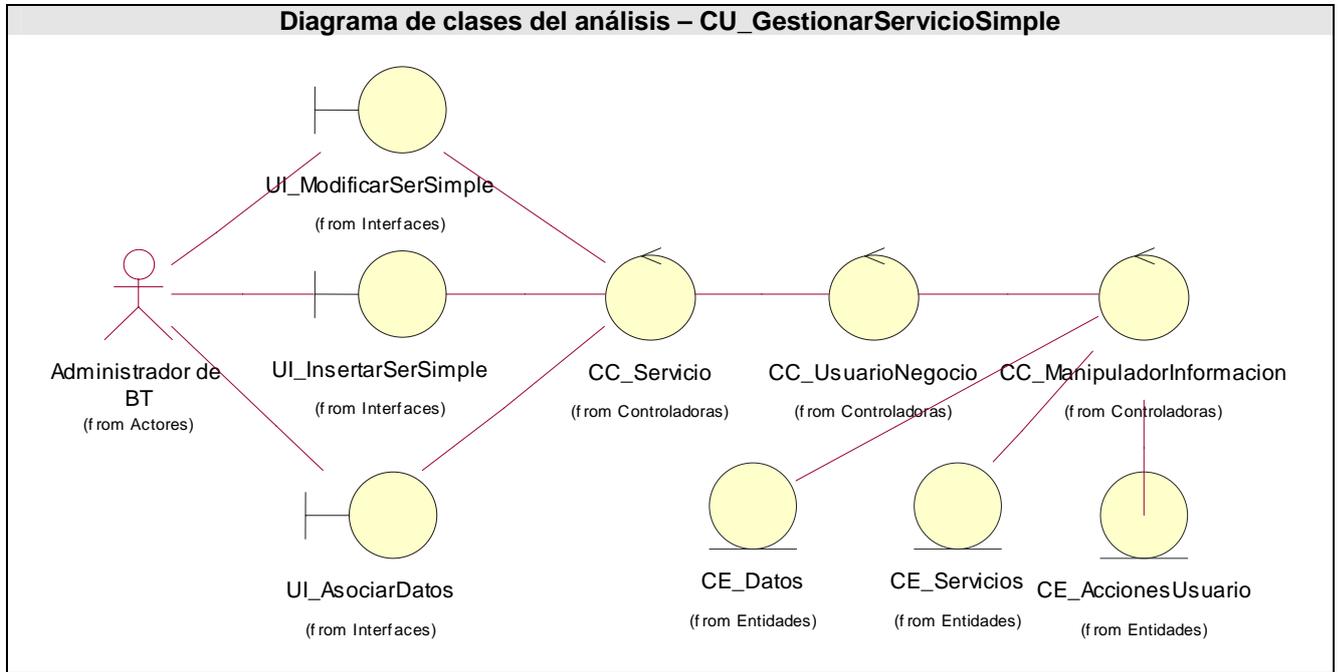
3.2. Análisis del sistema

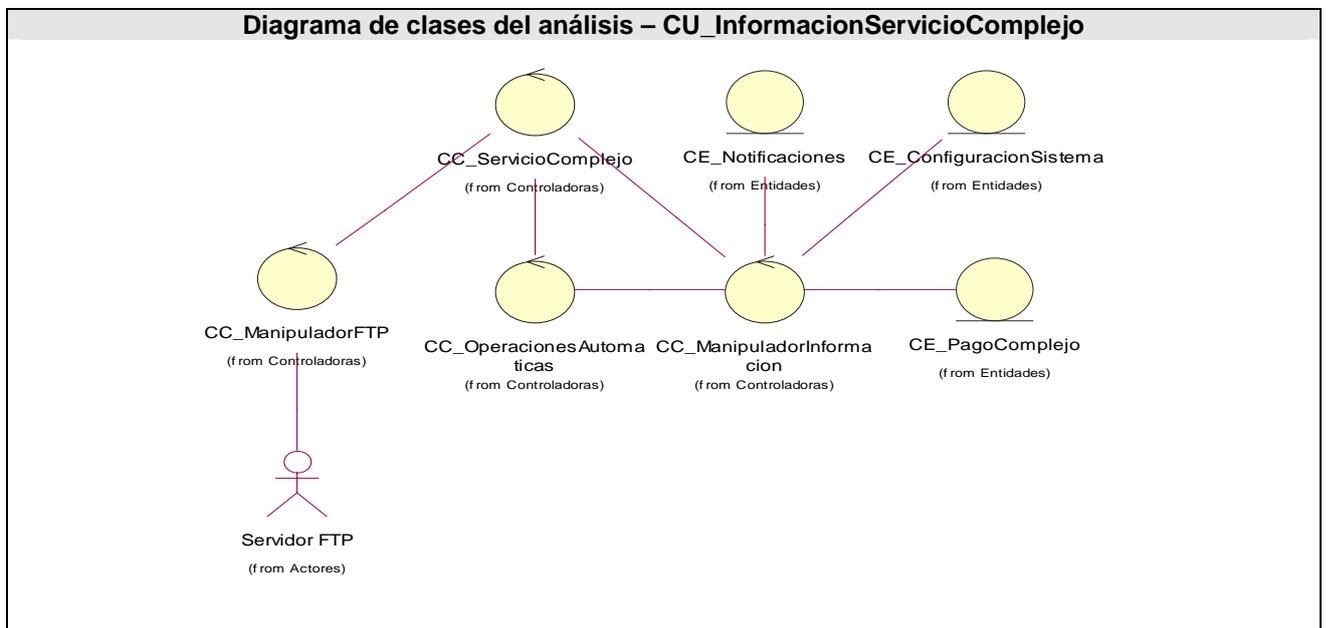
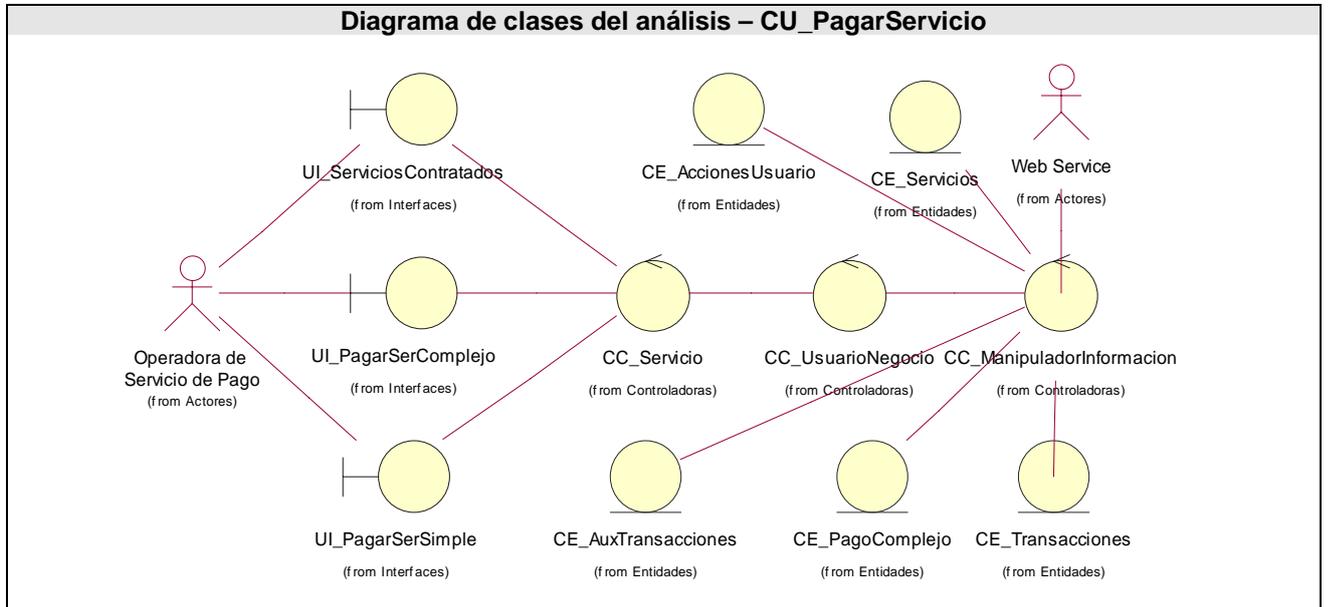
En la construcción de los diagramas de clases del análisis se tienen que identificar las clases que describirán la realización de los casos de uso y las relaciones entre ellas. En el análisis encontramos tres tipos de clases las cuales se clasifican en: entidad, son las que modelan información que posee larga vida y que es a menudo persistente, interfaz, modelan la interacción entre el sistemas y sus actores, por último la clase controladora, que coordina la realización de los casos de usos. A continuación se verán los diagramas de clases del análisis definido para cada caso de uso.

3.2.1. Diagrama de clases del análisis





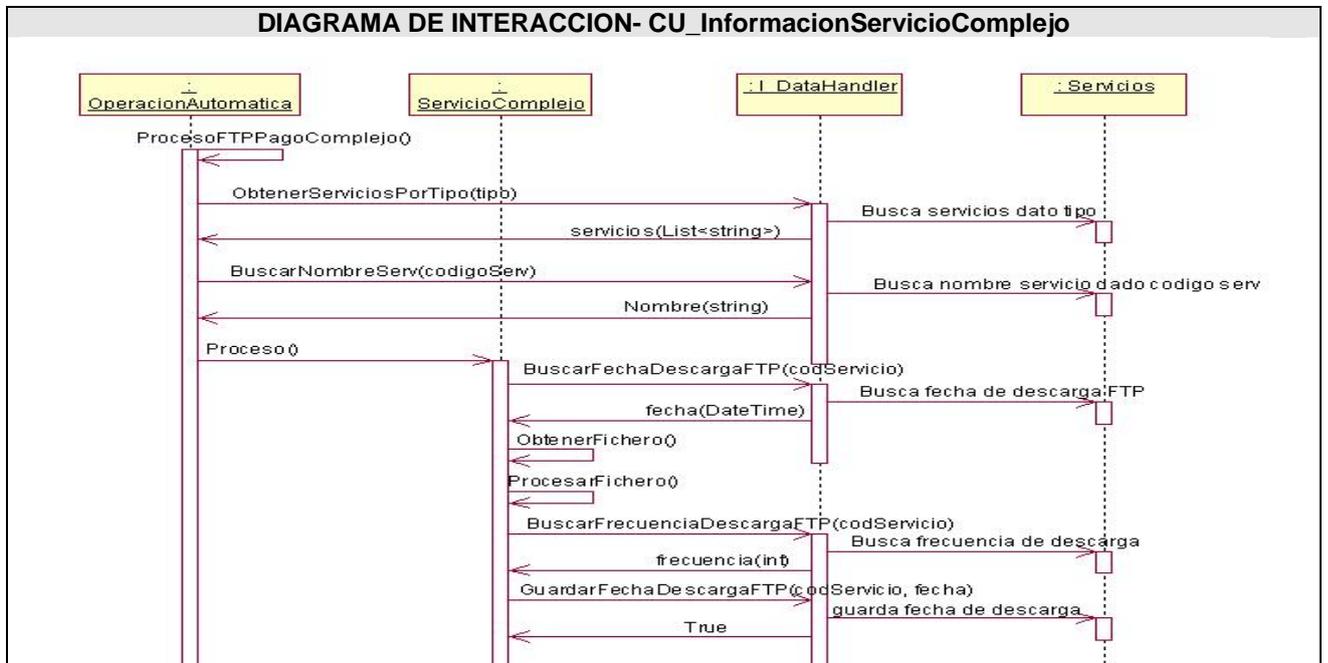


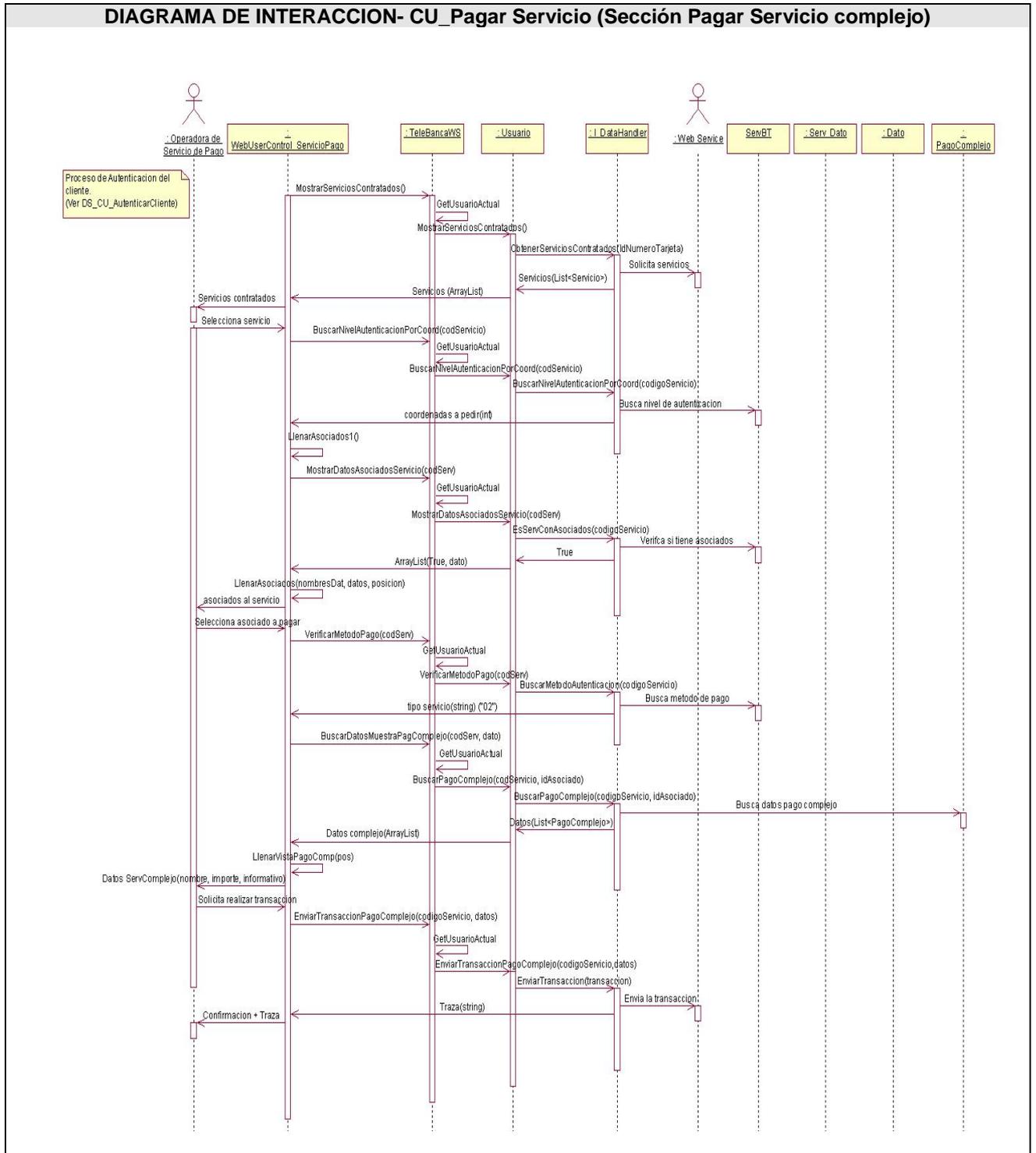


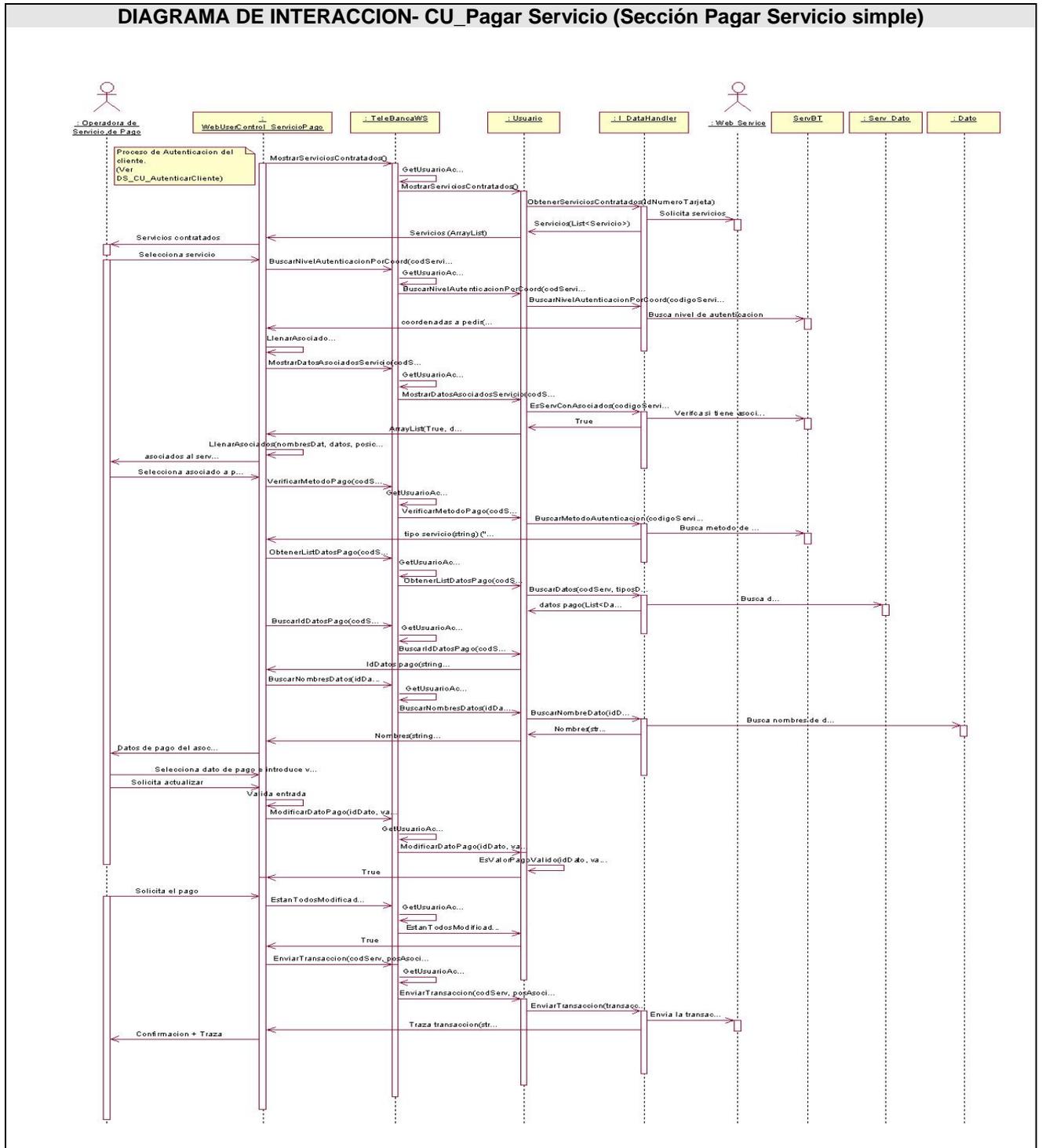
3.3. Diseño del sistema

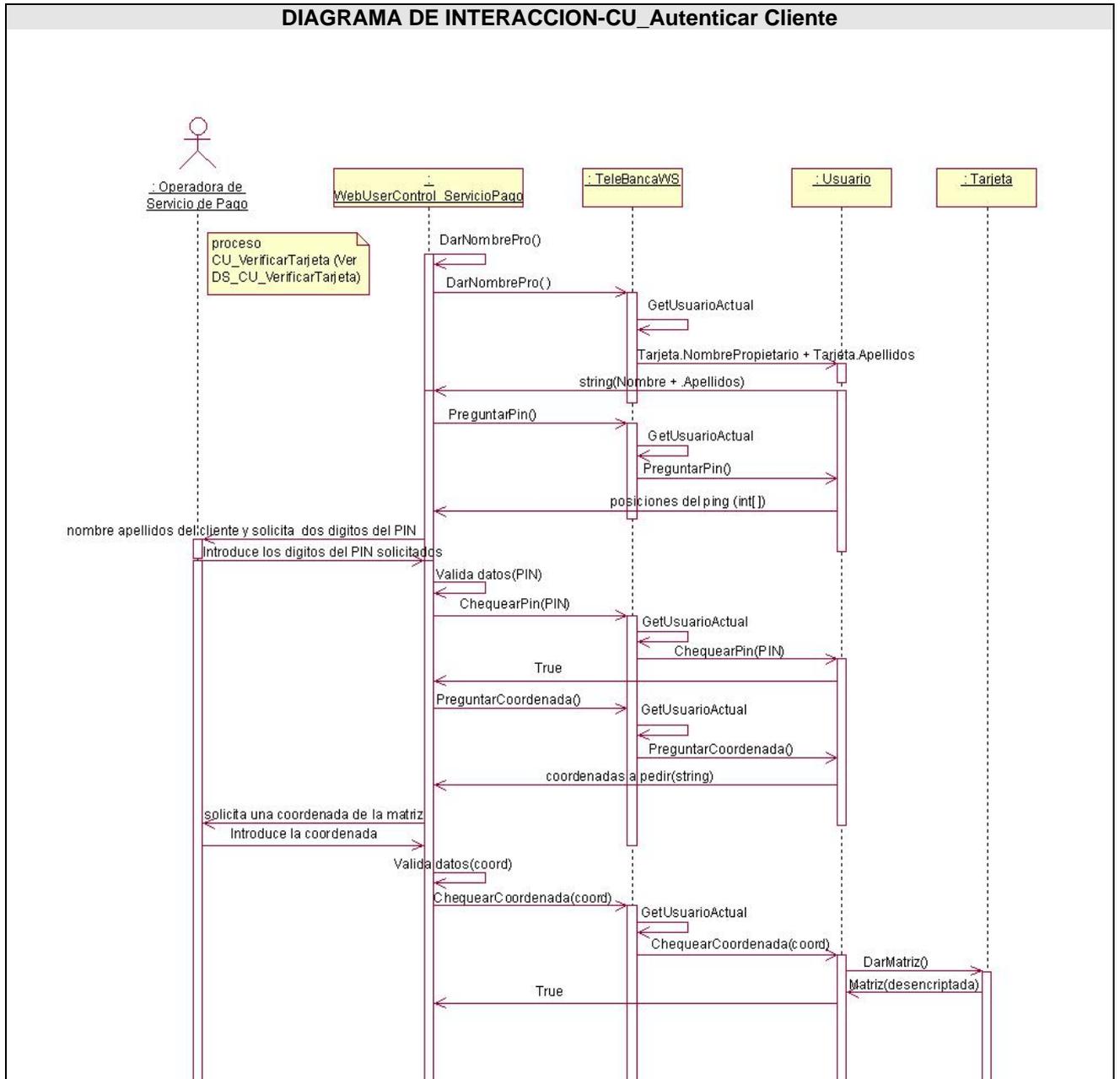
En el diseño se tiene el propósito de formular los modelos para preparar la entrada a las actividades de implementación y pruebas del sistema, preparando un plano para los artefactos que se crean durante cada uno de estos flujos de trabajo. En el diseño modelamos el sistema y encontramos su forma para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. A continuación se encuentran las representaciones de los diagramas de clases del diseño y diagramas de secuencias correspondientes a cada uno de los casos de usos.

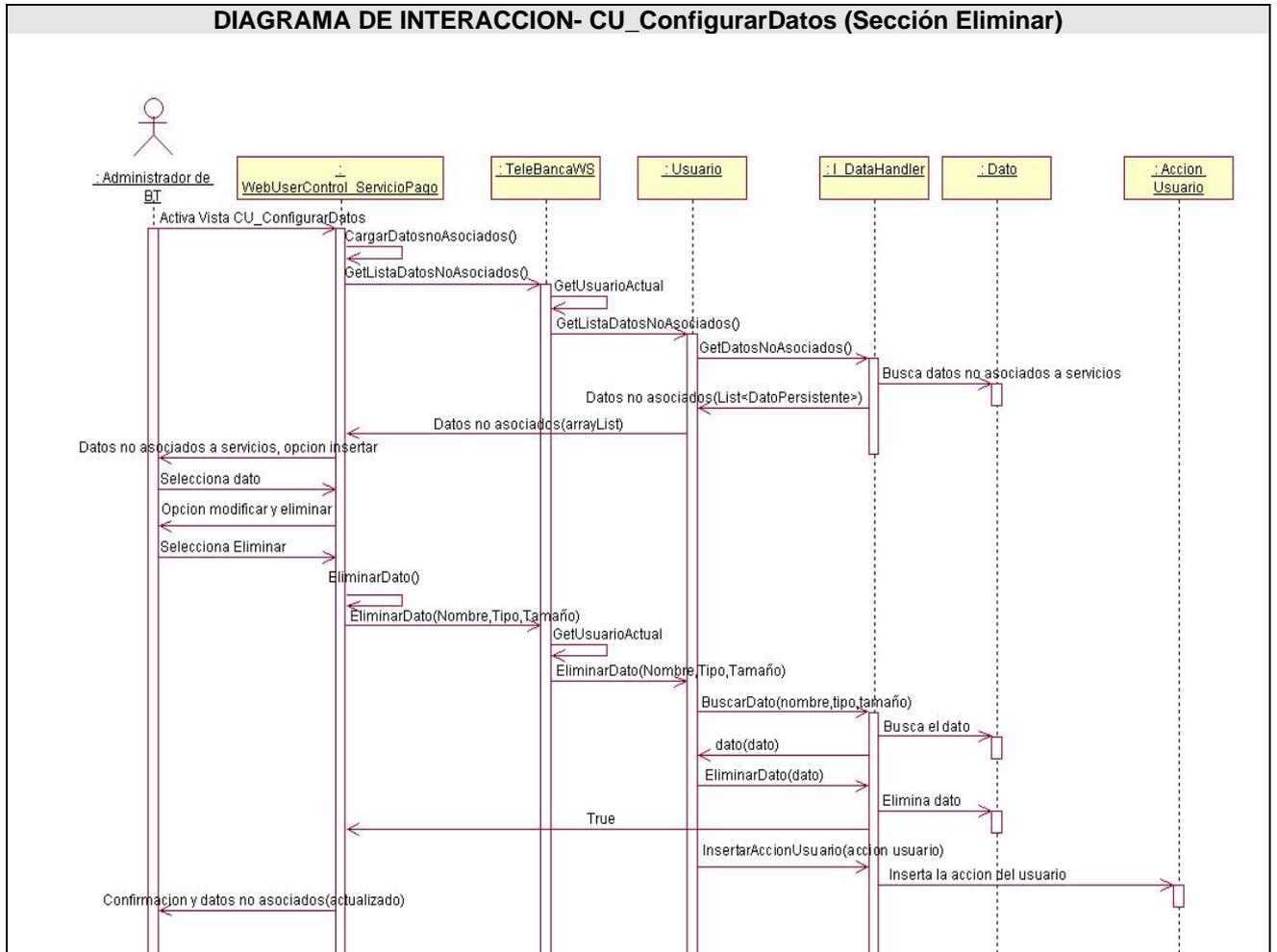
3.3.1. Diagramas de secuencias del diseño

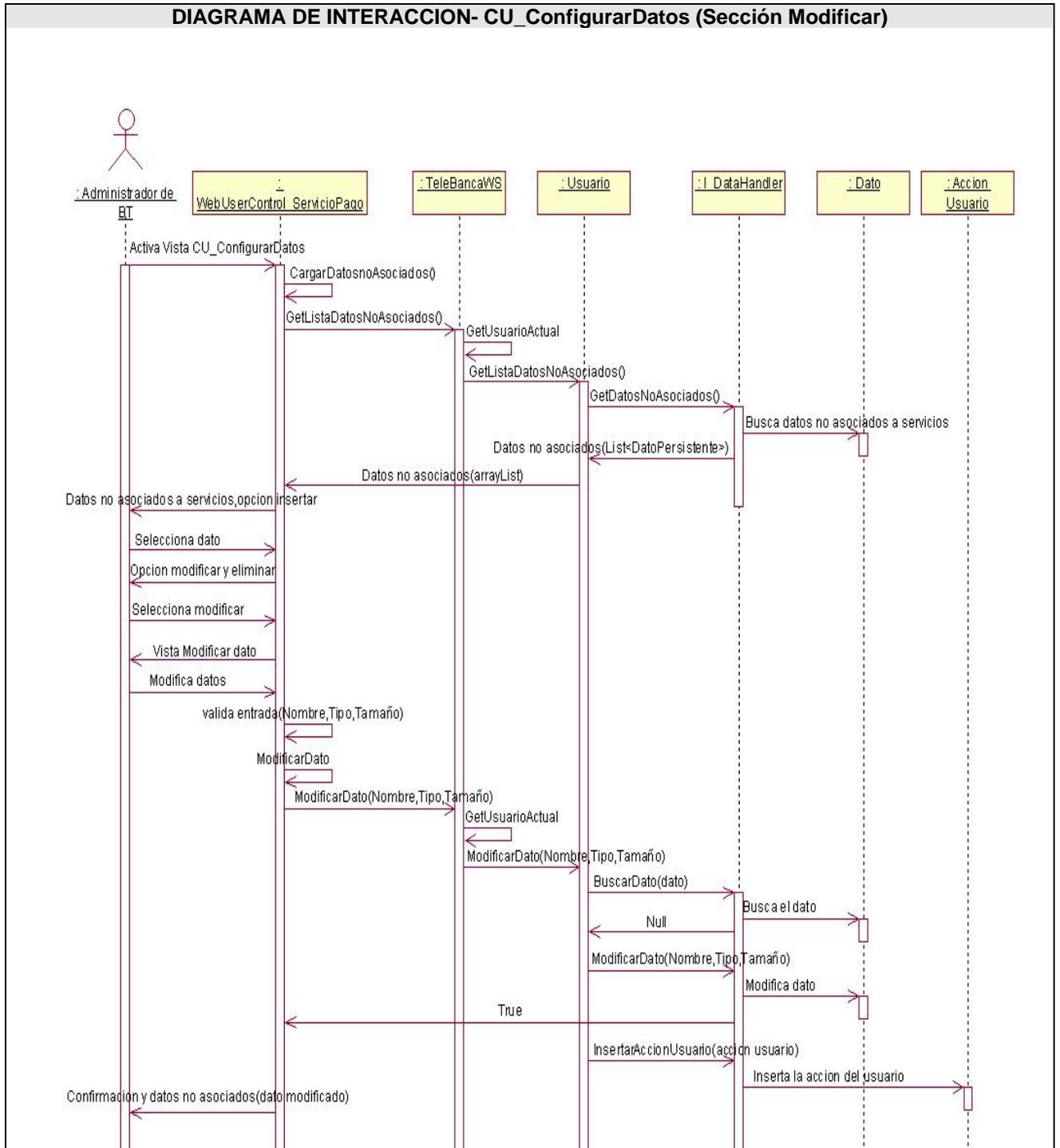


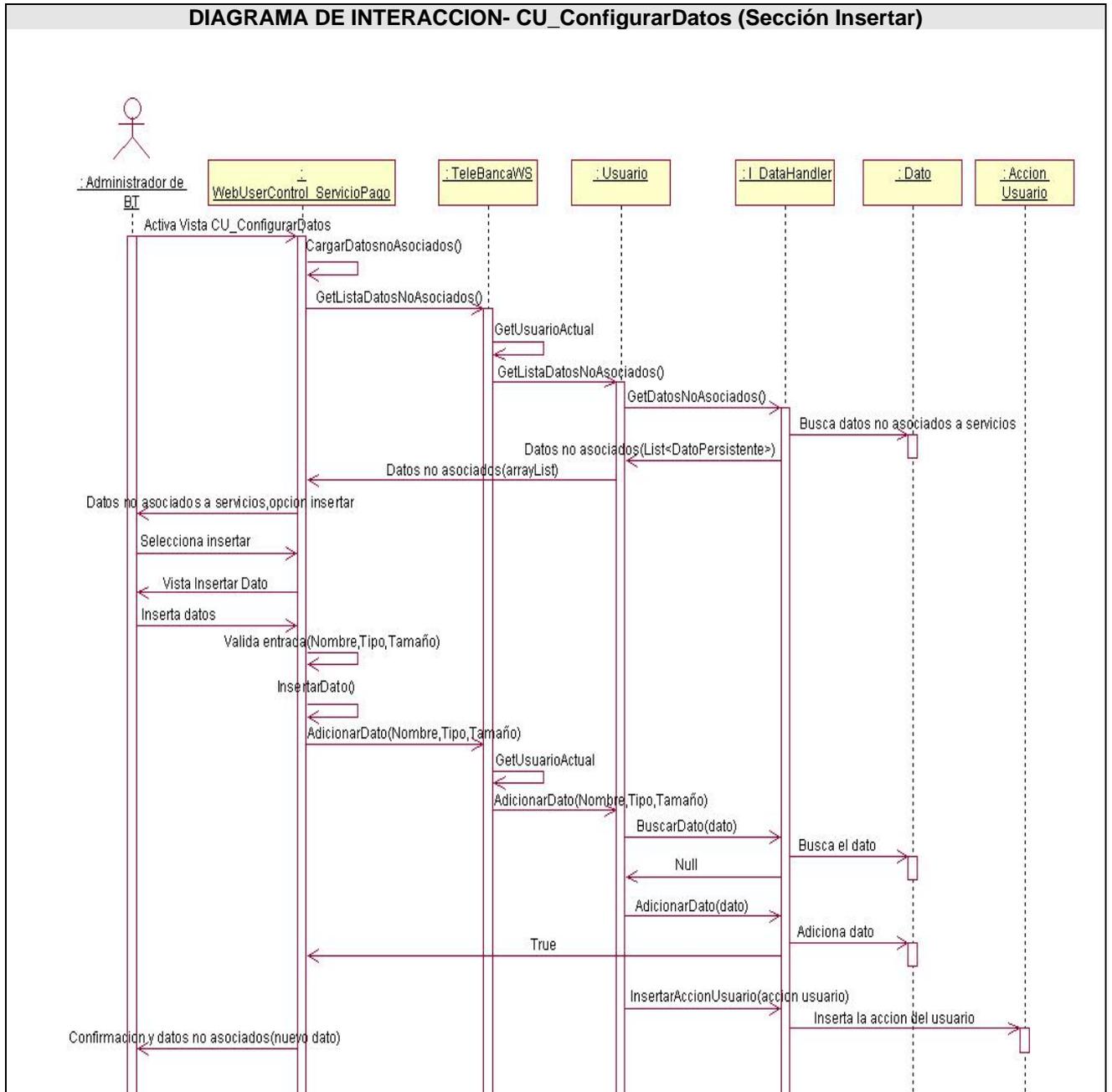












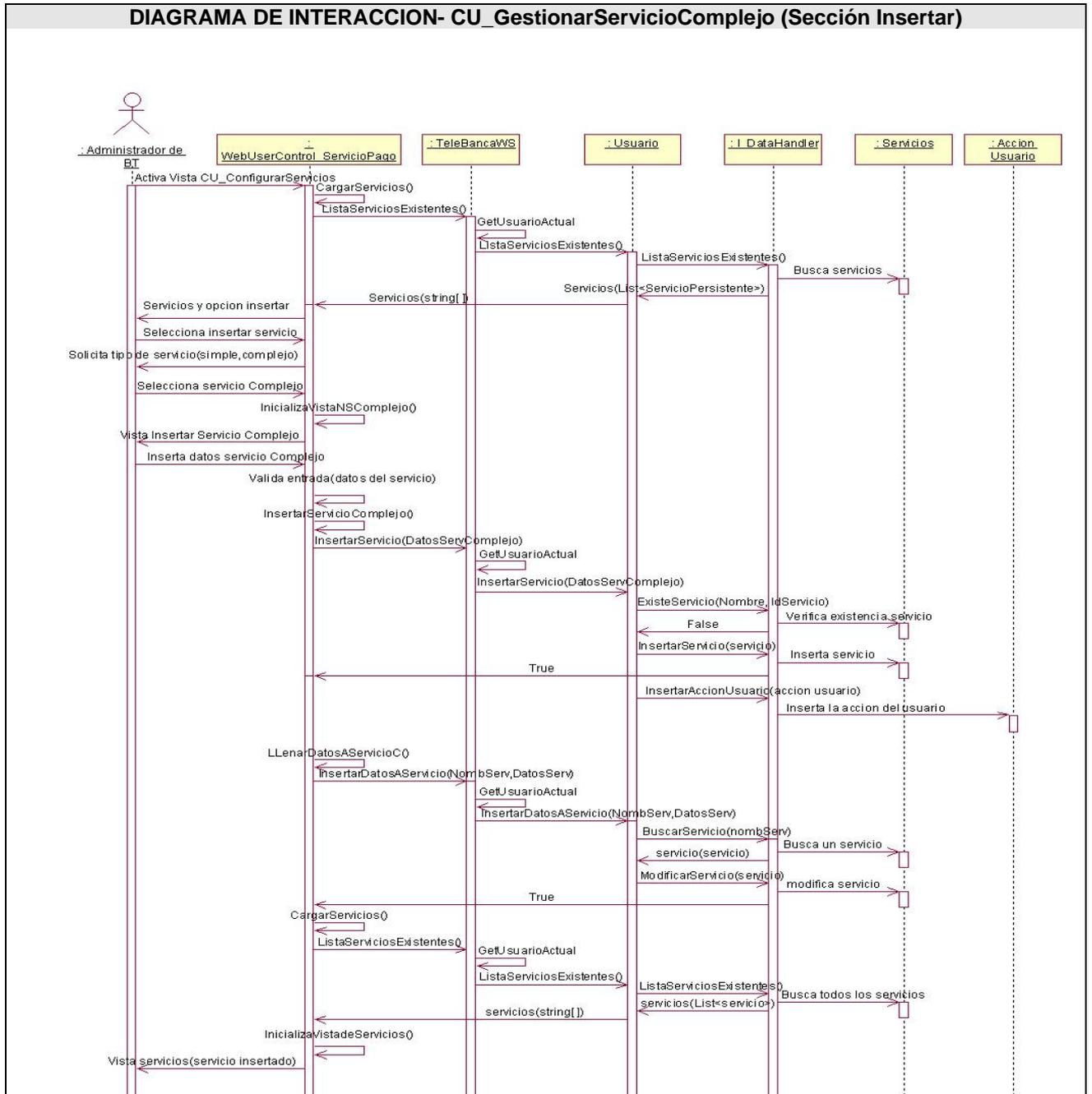
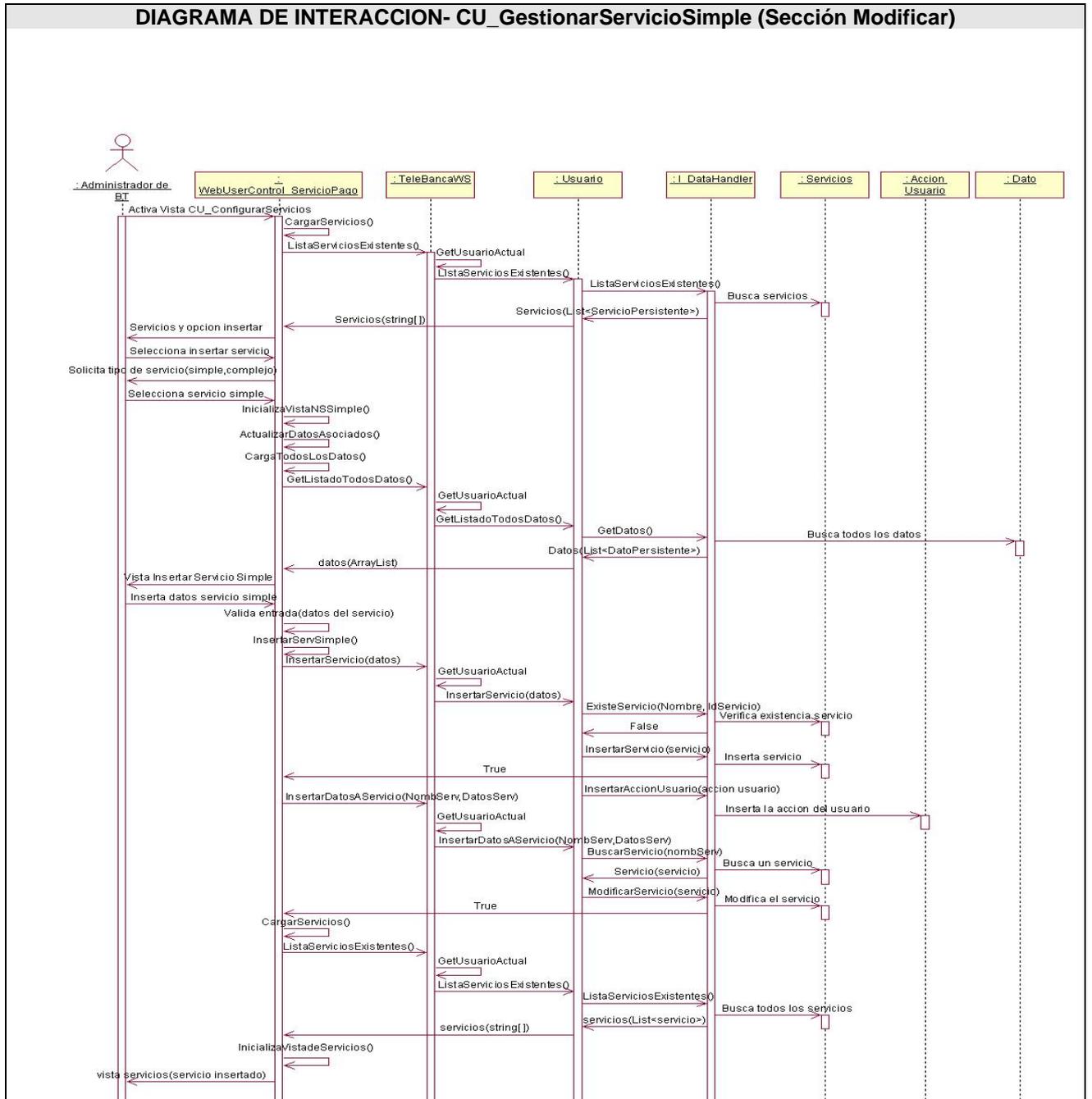


DIAGRAMA DE INTERACCION- CU_GestionarServicioSimple (Sección Modificar)



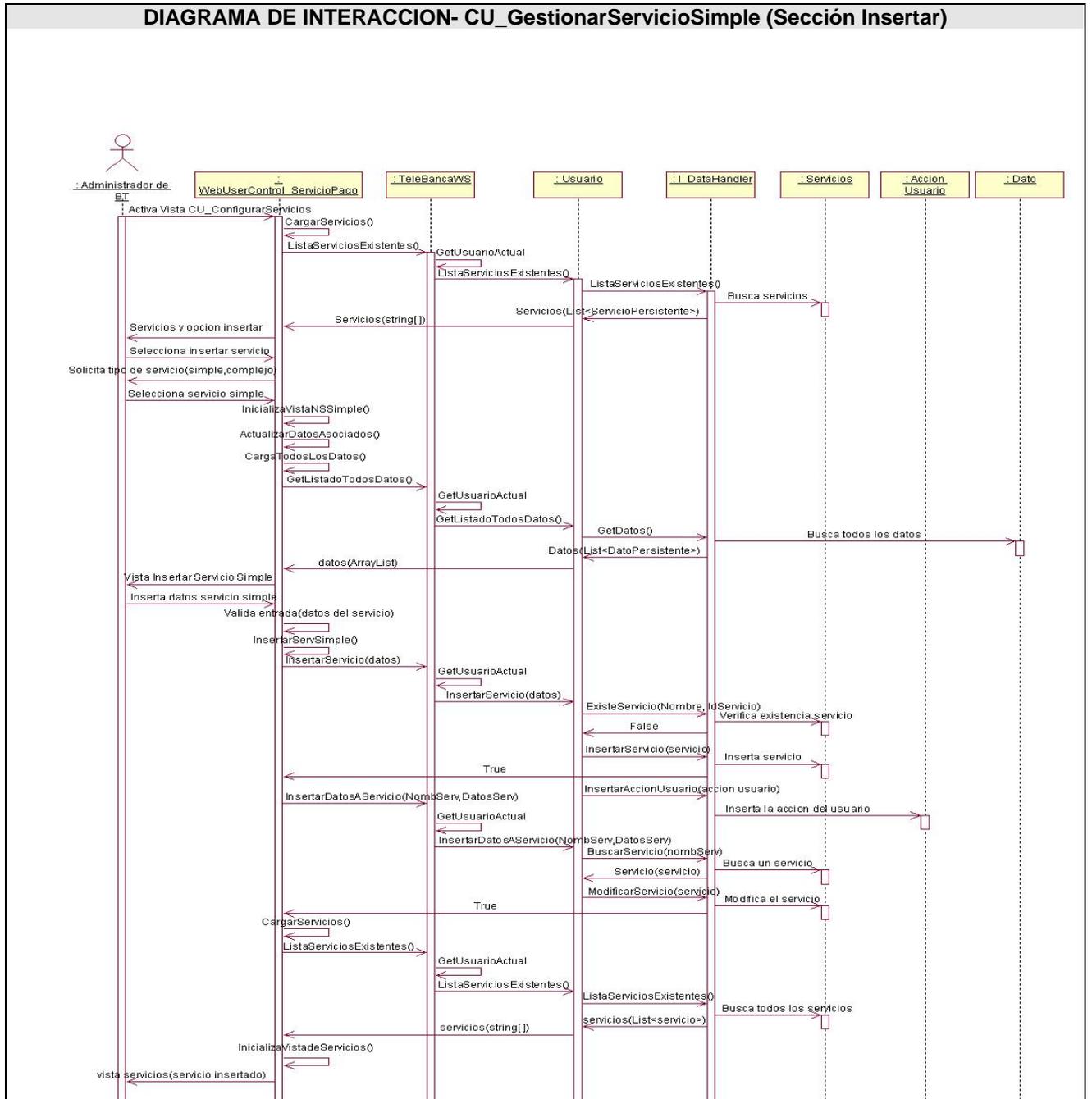
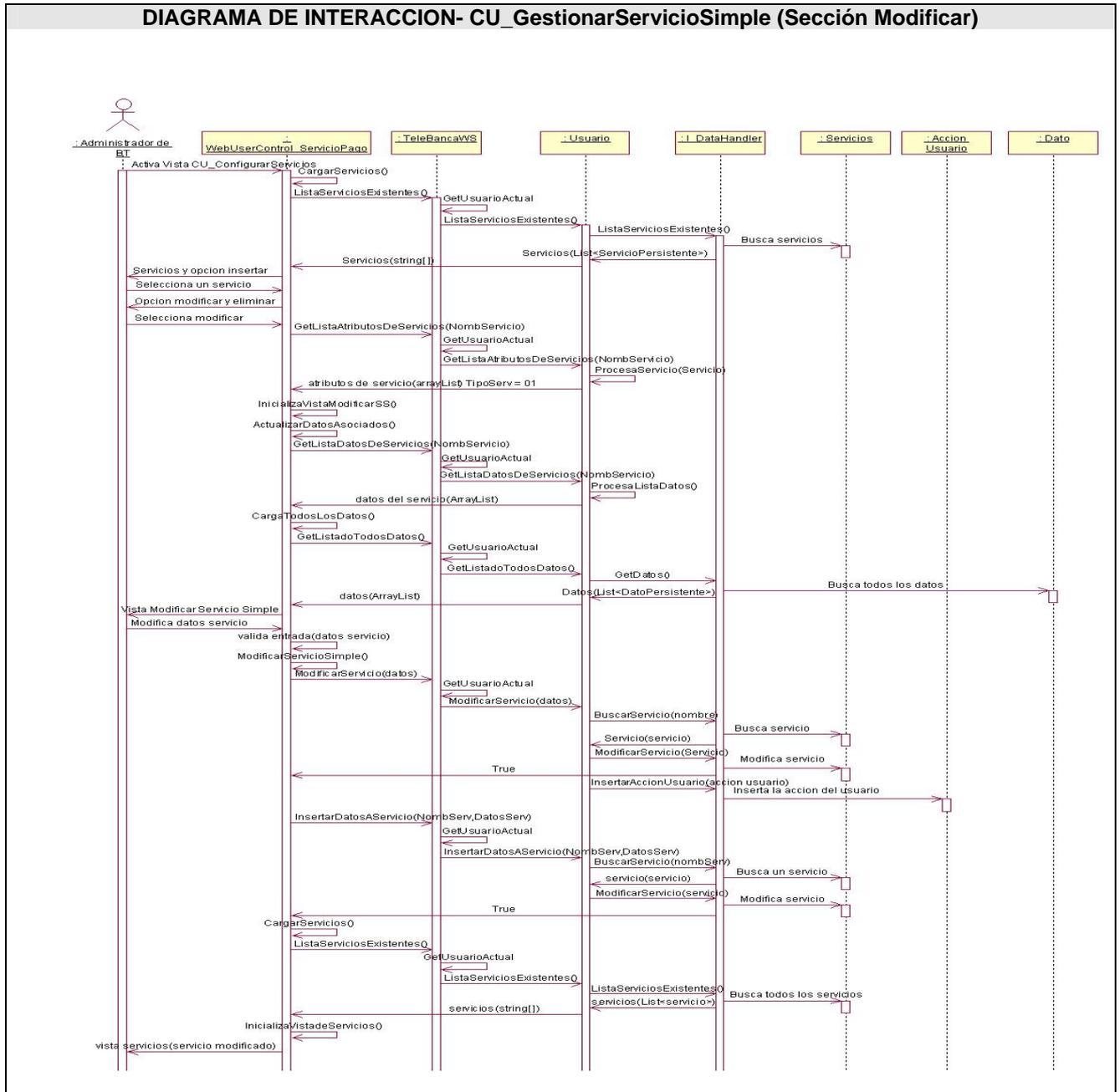
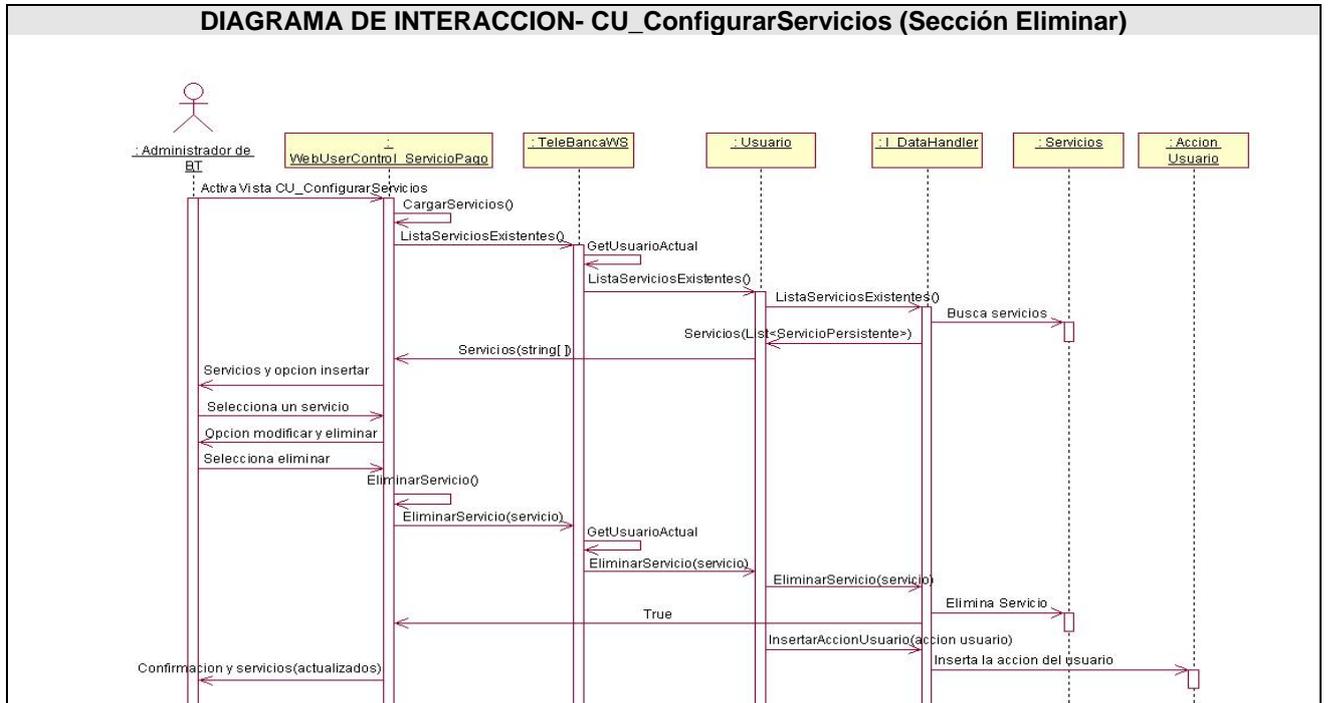
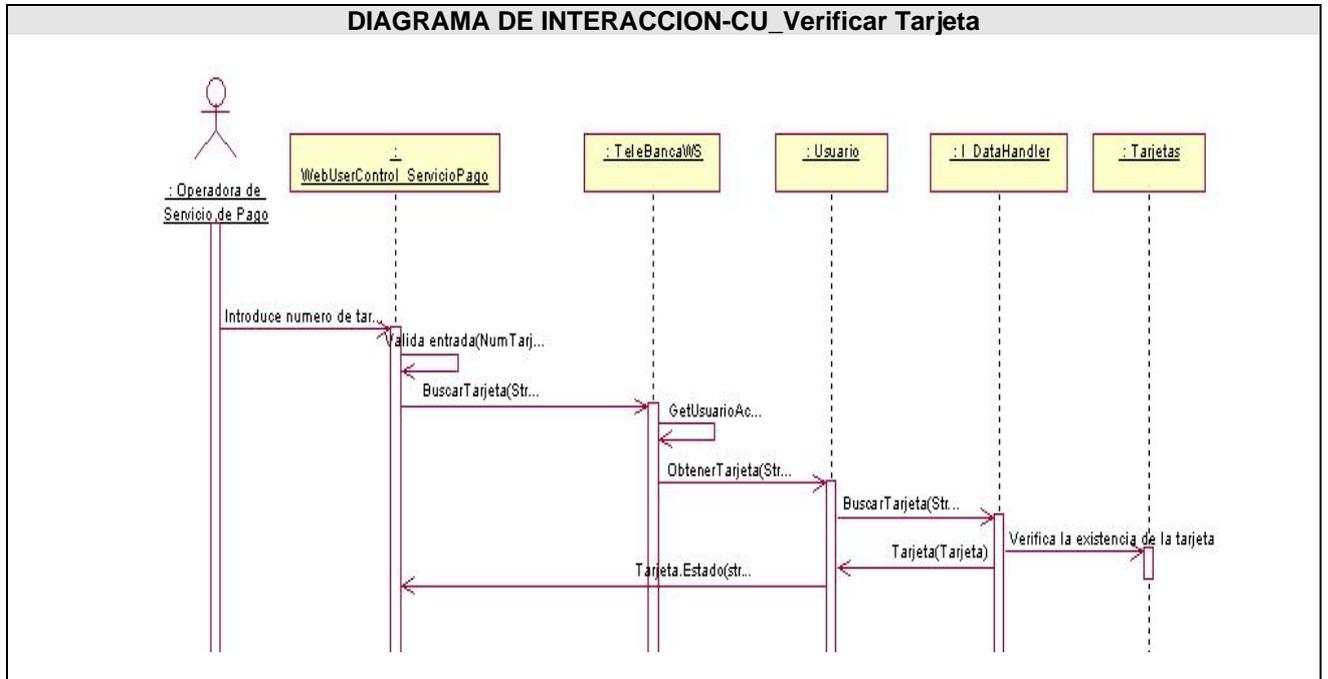


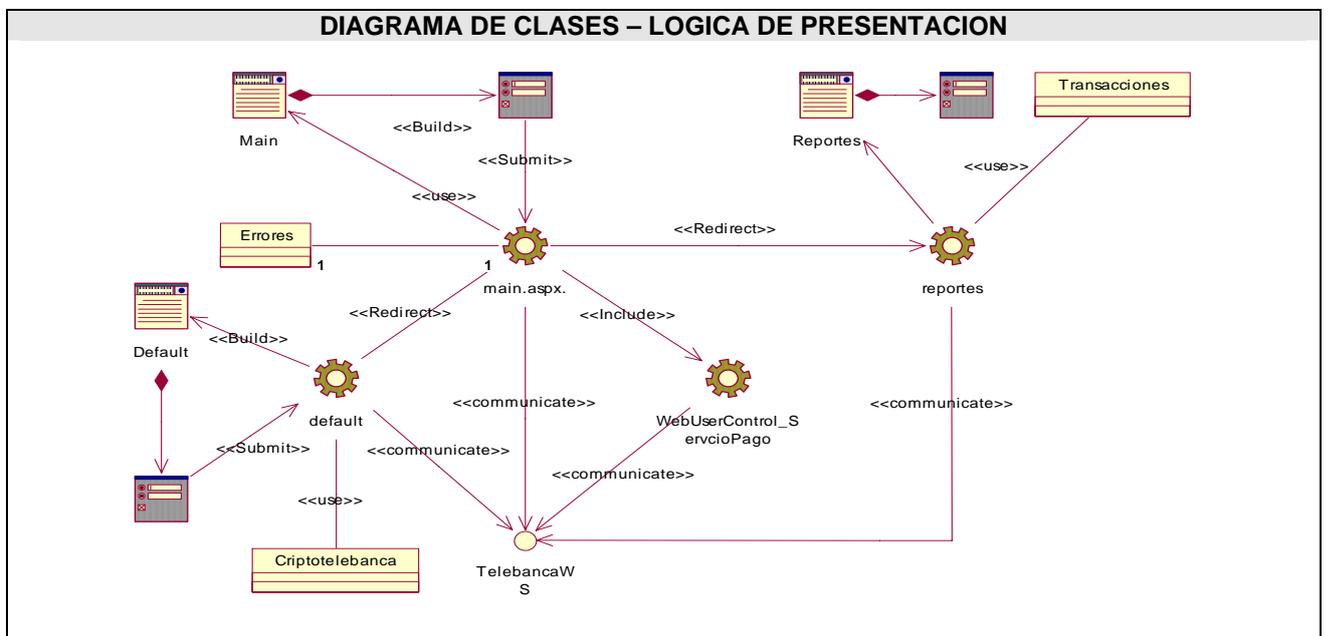
DIAGRAMA DE INTERACCION- CU_GestionarServicioSimple (Sección Modificar)

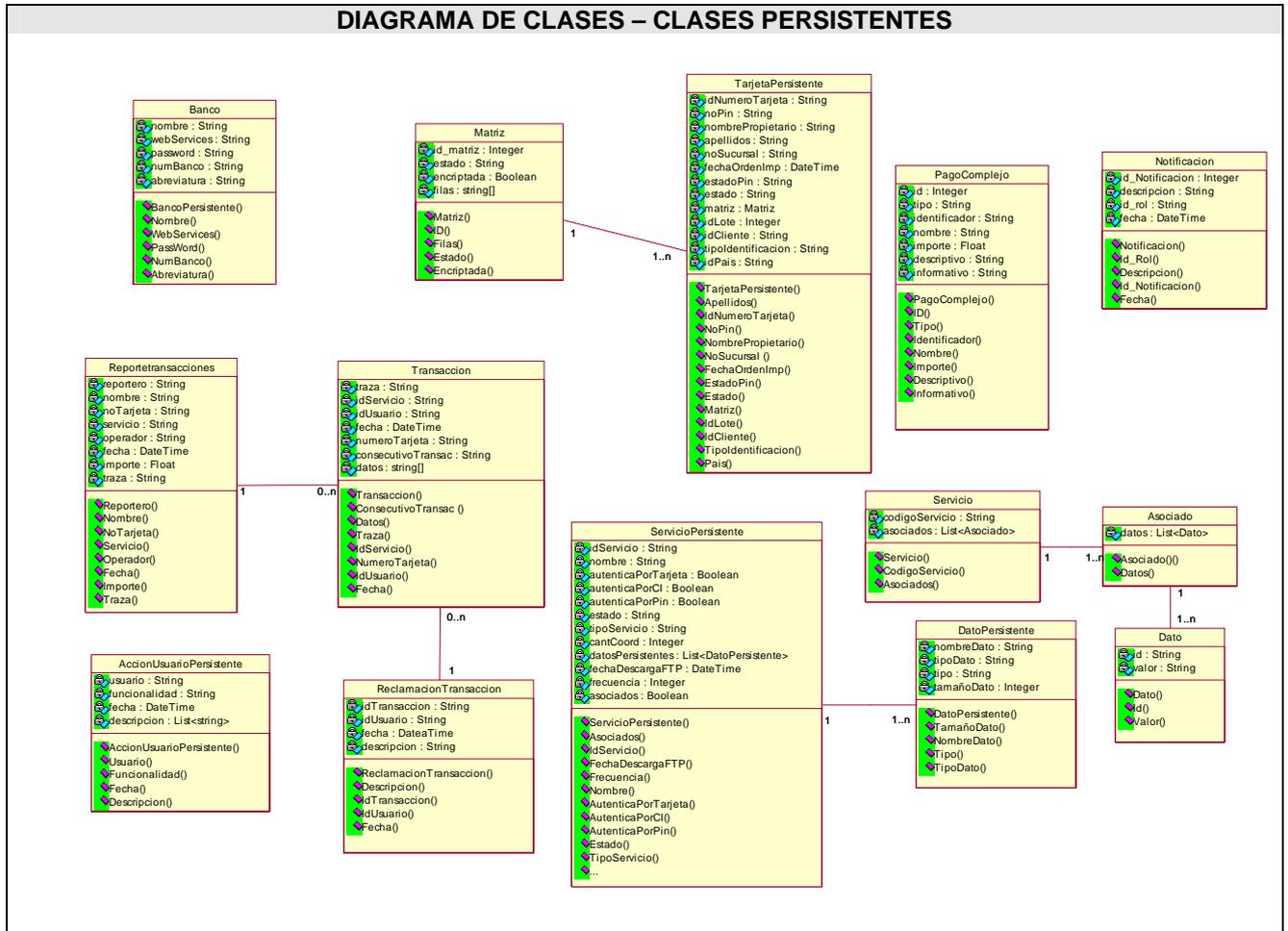






3.3.2. Diagrama de clases del diseño





3.3.3. Descripción de las clases del diseño

Nombre: COperacionAutomatica	
Tipo de clase: Controladora	
Atributo	Tipo
configuracion	Configuracion
Handler	DataHandler
A	Timer
B	Timer
Bloqueo	Object

Bloqueo1	Object
Bloqueo2	Object
empesoConciliaciones	bool
empesoPeticones	bool
EstaEnConciliacionesTrans	bool
Para cada responsabilidad:	
Nombre:	COperacionAutomatica()
Descripción:	Método constructor de la clase.
Nombre:	Actualizar()
Descripción:	Método que realiza actualizaciones con un intervalo de tiempo definido.
Nombre:	ActualizarTrans()
Descripción:	Método que realiza las transacciones con un intervalo de tiempo determinado.
Nombre:	EjecutarEspera(Object Info)
Descripción:	Método que define el intervalo de tiempo de una espera determinada.
Nombre:	EjecutarTransacciones(Object Info)
Descripción:	Método que realiza las transacciones de un banco determinado.
Nombre:	SolicitudesDeBajaDeTarjeta()
Descripción:	Método que se usa para obtener el número de solicitudes de dar baja a tarjetas.
Nombre:	EjecutarSolicitudesDeBajaDeTarjeta()
Descripción:	Método que ejecuta las bajas a las diferentes solicitudes de dar baja.
Nombre:	ObtenerNuevaMatriz()
Descripción:	Método que sirve para obtener una nueva matriz disponible y controla su estado.
Nombre:	GestionarAutenticaciones()
Descripción:	Método que nos permite crear nuevas autenticaciones o sea asignarle nuevas tarjetas al cliente.
Nombre:	EnviarConciliaciones()
Descripción:	Método que comprueba el estado de una tarjeta y guarda las mismas en arreglo agrupadas por dicho estado.
Nombre:	ProcesoFTPPagoComplejo()

Descripción:	Método que realiza las descargas del FTP en caso de que haya alguna.
Nombre:	ProcesoFTPPagoComplejo (string IdServicio)
Descripción:	Método de prueba.
Nombre:	ConciliacionesAutomaticasRecTransac ()
Descripción:	Método que obtiene todas las transacciones y las reclamaciones de un listado de banco determinado.
Nombre:	ConciliacionesAutomaticasTransacciones (BancoPersistente B)
Descripción:	Método que guarda en la base de datos una transacción realizada en un momento determinado.
Nombre:	ConciliacionesAutomaticasReclamaciones (BancoPersistente B1)
Descripción:	Método que guarda en la base de datos una reclamación realizada en un momento determinado.

Nombre: Manipulador FTP	
Tipo de clase: Controladora	
Atributo	Tipo
remoteHost	string
remotePath	string
remoteUser	string
remotePass	string
remotePort	int
bytes	int
clientSocket	Socket
retValue	int
logged	Boolean
reply	string
mes	string
BLOCK_SIZE	int
buffer	Byte

ASCII	Encoding
Para cada responsabilidad:	
Nombre:	Boolean isLogined()
Descripción:	Método que verifica si existe alguien logueado.
Nombre:	ManipuladorFTP()
Descripción:	Método constructor de la clase.
Nombre:	DireccionServidor
Descripción:	Método que permite obtener o modificar la dirección de un servidor.
Nombre:	DirectorioRemoto
Descripción:	Método que permite obtener o modificar un directorio remoto.
Nombre:	UsuarioFTP
Descripción:	Método que me permite obtener o modificar un usuario.
Nombre:	ClaveUsuarioFTP
Descripción:	Método que permite obtener o modificar una contraseña del FTP.
Nombre:	PuertoRemoto
Descripción:	Método que permite obtener o modificar un puerto remoto.
Nombre:	DarListadoFicheros(string mask)
Descripción:	Método que da el listado de ficheros que hay en el directorio en ese momento.
Nombre:	DarTamañoFichero(string fileName)
Descripción:	Método que devuelve el tamaño de un fichero.
Nombre:	Conectarse()
Descripción:	Método que realiza la conexión al servidor remoto.
Nombre:	CambiarAModoBinario(Boolean mode)
Descripción:	Método que cambia un valor a modo binario o a modo ASCII.
Nombre:	Descargar1(string remFileName)
Descripción:	
Nombre:	Descargar2(string remFileName, Boolean resume)
Descripción:	
Nombre:	Descargar3(string remFileName, string locFileName)

Descripción:	
Nombre:	Descargar4 (string remFileName, string locFileName, Boolean resume)
Descripción:	Método que realiza las descargas del servidor.
Nombre:	SubirAlServidor (string fileName)
Descripción:	Método que se usa para subir un archivo al servidor.
Nombre:	SubirAlServidor (string fileName, Boolean resume)
Descripción:	Método que se usa para continuar con la subida de un fichero al servidor en caso de que se haya interrumpido la conexión.
Nombre:	BorrarFicheroRemoto (string fileName)
Descripción:	Método que permite borrar un fichero del directorio FTP.
Nombre:	RenombrarFicheroRemoto (string oldFileName, string newFileName)
Descripción:	Método que se usa para renombrar un fichero que este en el directorio FTP.
Nombre:	HacerDirectorioRemoto (string dirName)
Descripción:	Método que permite crear un nuevo directorio remoto.
Nombre:	BorrarDirectorioRemoto (string dirName)
Descripción:	Método que se usa para borrar un directorio de FTP.
Nombre:	CambiarDeDirectorioRemoto (string dirName)
Descripción:	Método para cambiar de un directorio remoto a otro.
Nombre:	Retroceder ()
Descripción:	Método para volver a la carpeta anterior en que se estaba trabajando en el servidor.
Nombre:	Avanzar (string carpeta)
Descripción:	Método para avanzar a la carpeta indicada dentro del servidor.
Nombre:	CerrarConeccion ()
Descripción:	Método que posibilita cerrar la conexión con el servidor FTP.
Nombre:	LeerRespuesta ()
Descripción:	Método que lee la respuesta de datos que haya en el servidor.
Nombre:	Limpiar ()
Descripción:	Método para limpiar campos después de desloguear un usuario del servidor.

Nombre:	LeerLinea()
Descripción:	Método que lee línea por línea la respuesta de datos que haya en el servidor.
Nombre:	MandarComando(String command)
Descripción:	Método que manda los comandos del servidor.
Nombre:	obtenerRemotePath()
Descripción:	Método para obtener un directorio remoto.
Nombre:	CrearSocketParaDatos()
Descripción:	Método para crear una nueva conexión de datos con el servidor.

Nombre: ServicioComplejo	
Tipo de clase: Controladora	
Atributo	Tipo
fecha	DateTime
frecuencia	int
handler	DataHandler
directorioTemporal	string
directorioRemoto	string
codServicio	string
manipuladorFTP	ManipuladorFTP
fichero	string
dirFicheroReporte	string
idNotificacion	int
existeFichero	bool
cantEsperaFTP	int
Para cada responsabilidad:	
Nombre:	ServicioComplejo (DataHandler handler, string codServicio, string directorioTemporal, string directorioRemoto)
Descripción:	Método constructor de la clase.
Nombre:	ServicioComplejo (DateTime fecha,DataHandler handler, string codServicio,

	string directorioTemporal, string directorioRemoto)
Descripción:	Método constructor de la clase.
Nombre:	DirFicheroReporte
Descripción:	Método que me da la posibilidad de crear, modificar u obtener un fichero de reporte.
Nombre:	Handler
Descripción:	Método que me da la posibilidad de crear, modificar u obtener un handler.
Nombre:	ManipuladorFTP
Descripción:	Método que me da la posibilidad de crear, modificar u obtener un fichero de ManipuladorFTP.
Nombre:	Fecha
Descripción:	Método que me da la posibilidad de crear, modificar u obtener una fecha.
Nombre:	DirectorioTemporal
Descripción:	Método que me da la posibilidad de crear, modificar u obtener un directorio temporal
Nombre:	DirectorioRemoto
Descripción:	Método que me da la posibilidad de crear, modificar u obtener un directorio remoto.
Nombre:	CodServicio
Descripción:	Método que me da la posibilidad de crear, modificar u obtener un código de servicio.
Nombre:	Frecuencia
Descripción:	Método que me da la posibilidad de crear, modificar u obtener una frecuencia.
Nombre:	Proceso()
Descripción:	Método que actualiza tras una descarga todos los datos de fecha y demás de la misma.
Nombre:	DescompactarFichero()
Descripción:	Método que nos da la posibilidad de descompactar un fichero para en caso de que cuando se haga el recibo venga alguno en esas condiciones.

Nombre:	ObtenerFichero()
Descripción:	Método que gestiona la conexión con el servidor FTP, controla y valida los ficheros.
Nombre:	ValidarLista (string[] lista)
Descripción:	Método que comprueba la autenticidad de los datos de una lista.
Nombre:	NombreValido (string nombre)
Descripción:	Método que comprueba que el nombre de un fichero sea válido.
Nombre:	FechaValida (string nombre)
Descripción:	Método que comprueba que el formato de la fecha sea el indicado.
Nombre:	ProcesarFichero ()
Descripción:	Método que actualiza la base de datos con los nuevos ficheros descargados de FTP.
Nombre:	EjecutarFuncionFTP (string metodo, object[] parametros, string mensajeError, int tiempoEspera)
Descripción:	Método que controla el tiempo de conexión al FTP y el tiempo de cambiar de directorio remoto.
Nombre:	EscribirFichero (string mensaje)
Descripción:	Método que escribe en un fichero un mensaje determinado.

Nombre: Tarjeta	
Tipo de clase: Controladora	
Atributo	Tipo
idNumeroTarjeta	string
noPin	string
nombrePropietario	string
apellidos	string
noSucursal	string
fechaOrdenImp	DateTime

estadoPin	string
estado	string
matriz	Matriz
idLote	int
idCliente	string
tipoidentificacion	string
idPais	string
Para cada responsabilidad:	
Nombre:	DarMatriz()
Descripción:	Método que me descripta una matriz.

Nombre: TeleBancaWS1	
Tipo de clase: Controladora	
Atributo	Tipo
UsuariosActivos	Dictionary<string, Usuario>
UsuarioActivo	string
Para cada responsabilidad:	
Nombre:	ActualizarSucursales()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a hacer la actualización de las sucursales.
Nombre:	BuscarTarjeta (string numTarjeta)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a hacer una búsqueda de tarjeta.
Nombre:	BuscarTarjetaPorCI (string CI)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a hacer una búsqueda de tarjeta por CI.
Nombre:	ConsultarSaldo()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a consultar el saldo.

Nombre:	DarNombrePro()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a devolver el nombre de un propietario según su tarjeta.
Nombre:	CambiarEstadoTarjeta (string numeroTarjeta, string estado)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo cambiar el estado de una tarjeta o no.
Nombre:	PreguntarPin()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a generar un pin.
Nombre:	PreguntarCoordenada()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a generar una coordenada.
Nombre:	ChequearPin (int[] ping)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si un pin generado es correcto o no.
Nombre:	ChequearCoordenada (string[] coordenadas)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si una coordenada generada es correcta o no.
Nombre:	BuscarNivelAutenticacionPorCoord (string codigoServicio)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a buscar la cantidad de coordenadas mas que hay que pedir.
Nombre:	BuscarNivelAutenticacionPorCI (string codigoServicio)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si hay que pedir CI o no.
Nombre:	VerificarMetodoPago (string codigoServicio)
Descripción:	Método que partiendo del usuario que esta autenticado, dice el modo de pago de un servicio ya sea simple o complejo.
Nombre:	EnviarTransaccion (string codigoServicio, int posAsociado)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a enviar una transacción.

Nombre:	EnviarTransaccionPagoComplejo (string codigoServicio,string[] datos)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a enviar una transacción para un pago complejo.
Nombre:	MostrarTransaccionAReclamar (string traza, int[] fechaTrans)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar la transacción a reclamar.
Nombre:	ReclamarTransaccion (string idTransaccion, string descripcion)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo reclamar una transacción o no.
Nombre:	MostrarServiciosContratados ()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar los servicios contratados.
Nombre:	MostrarIdAsociados (string codigoServicio)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar el ID de todos los asociados a un servicio.
Nombre:	MostrarDatosAsociadosServicio (string codigoServicio)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar todo los datos asociados a un servicio.
Nombre:	BuscarNombresDatos (string[] idDato)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a buscar el nombre de todos los datos de un servicio.
Nombre:	ModificarDatoPago (string idDato,string valor)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se puede modificar los datos de un pago o no.
Nombre:	MostrarDatoPago (string idDato)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar un dato de un pago determinado.
Nombre:	EstanTodosModificados ()
Descripción:	Método que partiendo del usuario que esta autenticado, dice si los datos de un

	pago están modificados o no.
Nombre:	ObtenerListDatosPago (string codServ)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar un listado con los datos de un pago dado un servicio determinado.
Nombre:	BuscarIdDatosPago (string codServ)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a buscar los ID de los datos de un pago dado un servicio.
Nombre:	BuscarDatosMuestraPagComplejo (string codServicio, string idAsociado)
Descripción:	Método que partiendo del usuario que esta autenticado, método que manda a buscar los datos de un pago complejo dado un servicio determinado.
Nombre:	AdicionarBanco (string nombre, string webServices, string password, string numBanco, string abreviatura)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se puede adicionar un banco o no.
Nombre:	BuscarDatosBanco (string numBanco)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a buscar los datos de un banco determinado.
Nombre:	ObtenerListaBanco ()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a crear un listado con todos los bancos.
Nombre:	ModificarBanco (string nombre, string webServices, string password, string numBanco, string abreviatura)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se puede modificar un banco o no.
Nombre:	EliminarBanco (string numBanco)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se puede eliminar un banco o no.
Nombre:	getLitadoTransacciones (DateTime ini, DateTime fin)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a devolver un

	listado con todas las transacciones.
Nombre:	InsertarServicio (string idser, string nombre, bool autenticaPorTarjeta, bool autenticaPorCI, bool autenticaPorPin, string estado, string tipoServicio, int cantCoord, DateTime fechaDescargaFTP, int frecuencia, bool asociados)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo insertar un servicio o no.
Nombre:	InsertarDatosAServicio (string nombServ, object[] datos)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo insertar datos a un servicio o no.
Nombre:	ModificarServicio (string idser, string nombre, bool autenticaPorTarjeta, bool autenticaPorCI, bool autenticaPorPin, string estado, string tipoServicio, int cantCoord, DateTime fechaDescargaFTP, int frecuencia, bool asociados)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se puede modificar un servicio o no.
Nombre:	EliminarServicio (string nombServ)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo eliminar un servicio o no.
Nombre:	ListaServiciosExistentes ()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar todo los servicios existentes para una interfaz.
Nombre:	GetListaAtributosDeServicios (string nombre)
Descripción:	Método que partiendo del usuario que esta autenticado, mana a mostrar todos los atributos sencillos de un servicio.
Nombre:	GetListaDatosDeServicios (string nombSer)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar los datos de un servicio determinado.
Nombre:	AdicionarDato (string nombreDato, string tipoDato, string tipo, int tamañoDato)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo adicionar un dato a un servicio o no.

Nombre:	ModificarDato (string nombreDato, string tipoDato, string tipo, int tamañoDato, string nombreAnt)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo modificar un dato de un servicio o no.
Nombre:	EliminarDato (string nombreDato, string tipoDato, string tipo, int tamañoDato)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se pudo eliminar un dato de un servicio o no.
Nombre:	GetListadoTodosDatos ()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a devolver un listado de todos los datos de la BD.
Nombre:	GetListaDatosNoAsociados ()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a devolver un listado con los datos no asociados a ningún servicio.
Nombre:	ActualizarServiciosDatos (object[] listBancos)
Descripción:	Método que partiendo del usuario que esta autenticado, manda a actualizar todos los servicios y sus datos para un listado de bancos determinado.
Nombre:	ListadoDeBancosId ()
Descripción:	Método que partiendo del usuario que esta autenticado, manda a mostrar un listado del ID de todos los bancos.
Nombre:	ConciliacionesAuxTransaccion (DateTime fecha,string NumBanco)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se puede realizar una transacción o no.
Nombre:	ConciliacionesAuxReclamacion (DateTime fecha, string NumBanco)
Descripción:	Método que partiendo del usuario que esta autenticado, dice si se puede realizar una reclamación o no.

Nombre: Usuario	
Tipo de clase: Controladora	
Atributo	Tipo

tarjeta	Tarjeta
Servicios	List<Servicio>
listaServ	List<ServicioPersistente>
datosPago	List<Dato>
idCliente	string
Para cada responsabilidad:	
Nombre:	Tarjeta
Descripción:	Método que me devuelve y también me permite modificar una tarjeta.
Nombre:	ActualizarSucursales()
Descripción:	Método que actualiza los datos respecto a los servicios y datos.
Nombre:	BuscarTarjetaPorCI(string CI)
Descripción:	Método que nos devuelve el nombre de la persona y su número de tarjeta, con solo entrar su número de CI.
Nombre:	ObtenerTarjeta(string numeroTarjeta)
Descripción:	Método que nos devuelve el estado de una tarjeta entrando el número de la misma.
Nombre:	ConsultarSaldo()
Descripción:	Método que devuelve el saldo de una tarjeta.
Nombre:	CambiarEstadoTarjeta(string numeroTarjeta, string estado)
Descripción:	Método que nos permite modificar el estado de una tarjeta pasando el número de la misma y su estado actual.
Nombre:	PreguntarPin()
Descripción:	Método que me genera las coordenadas del pin a pedir.
Nombre:	PreguntarCoordenada()
Descripción:	Método que genera las coordenadas que se van a pedir.
Nombre:	ChequearPin(int[] ping)
Descripción:	Método que chequea que las coordenadas del pin sean válidas.
Nombre:	ChequearCoordenada(string[] coordenadas)
Descripción:	Método que valida que las coordenadas que fueron generadas y entradas por teclado sean correctas.

Nombre:	ObtenerListDatosPago (string codServ)
Descripción:	Método que genera un listado de los datos de pago para un servicio determinado.
Nombre:	MostrarServiciosContratados ()
Descripción:	Método que muestra todos los servicios contratados de una tarjeta dado un cliente determinado.
Nombre:	QuitarServiciosDesactivados ()
Descripción:	Método que se encarga de remover los servicios desactivados de la lista de servicios de una tarjeta.
Nombre:	EstanTodosModificados ()
Descripción:	Método que verifica si se han modificado todos los datos de un pago.
Nombre:	BuscarIdDatosPago (string codServ)
Descripción:	Método que devuelve los identificadores de los datos de pago de un servicio.
Nombre:	MostrarIdAsociados (string codigoServicio)
Descripción:	Método que muestra el nombre de las personas que están asociados a un servicio determinado.
Nombre:	MostrarDatosAsociadosServicio (string codigoServicio)
Descripción:	Método que muestra todos los datos asociados a un servicio determinado.
Nombre:	BuscarNombresDatos (string[] idDato)
Descripción:	Método que devuelve el nombre de un dato.
Nombre:	MostrarTransaccionAReclamar (string traza, int[] fechaTrans)
Descripción:	Método que devuelve todos los datos de una transacción
Nombre:	ReclamarTransaccion (string idTransaccion, string idUsuario, string descripcion)
Descripción:	Método que me dice si una transacción ha sido reclamada o no.
Nombre:	BuscarNivelAutenticacionPorCoord (string codigoServicio)
Descripción:	Método que me devuelve la cantidad de coordenadas de un servicio según el nivel de autenticación.
Nombre:	BuscarNivelAutenticacionPorCI (string codigoServicio)
Descripción:	Método que devuelve si el servicio que se entra requiere autenticación mediante el CI.
Nombre:	VerificarMetodoPago (string codigoServicio)

Descripción:	Método que devuelve el nivel de complejidad(simple o complejo) de un servicio determinado.
Nombre:	BuscarPagoComplejo (string codigoServicio, string idCliente)
Descripción:	Método que devuelve el importe de un cliente dado un servicio determinado, además de su nombre y otros datos.
Nombre:	IdDatoToString (string id)
Descripción:	Método para controlar la longitud del identificador de un dato.
Nombre:	buscarEnDatosPagosPorID (string id)
Descripción:	Método que devuelve el valor de un dato dado su identificador.
Nombre:	EnviarTransaccion (string codigoServicio,int posAsociado)
Descripción:	Método que envía los datos de una transacción para un servicio simple.
Nombre:	EnviarTransaccionPagoComplejo (string codigoServicio, string[] datos)
Descripción:	Método que envía los datos de una transacción para un servicio complejo.
Nombre:	EsValorPagoValido (string idDato, string valor)
Descripción:	Método que comprueba si el valor de un dato es válido o no.
Nombre:	ModificarDatoPago (string idDato, string valor)
Descripción:	Método que nos dice si el valor de un dato ha sido modificado o no.
Nombre:	MostrarDatoPago (string idDato)
Descripción:	Método encargado de mostrar el valor de un dato dado su identificador.
Nombre:	AdicionarBanco (string nombre, string webServices, string passWord, string numBanco, string abreviatura)
Descripción:	Método que adiciona un banco.
Nombre:	BuscarDatosBanco (string numBanco)
Descripción:	Método que devuelve los datos de un banco.
Nombre:	ObtenerListaBanco ()
Descripción:	Método que devuelve un listado con todos los bancos.
Nombre:	ModificarBanco (string nombre, string webServices, string passWord, string numBanco, string abreviatura)
Descripción:	Método a través del cual se modifican los datos de un banco.

Nombre:	EliminarBanco (string numBanco)
Descripción:	Método que me permite eliminar un banco de la lista.
Nombre:	getLitadoTransacciones (DateTime ini, DateTime fin)
Descripción:	Método que me devuelve un listado de transacciones dado un intervalo de tiempo.
Nombre:	InsertarServicio (string idServ, string nombre, bool autenticaPorTarjeta, bool autenticaPorCI, bool autenticaPorPin, string estado, string tipoServicio, int cantCoord, DateTime fechaDescargaFTP, int frecuencia, bool asociados)
Descripción:	Método que me devuelve si un servicio ha sido insertado correctamente o no.
Nombre:	InsertarDatosAServicio (string nombServ, object[] datos)
Descripción:	Método que me dice si se pudieron insertar nuevos datos a un servicio o no.
Nombre:	ModificarServicio (string idserv, string nombre, bool autenticaPorTarjeta, bool autenticaPorCI, bool autenticaPorPin, string estado, string tipoServicio, int cantCoord, DateTime fechaDescargaFTP, int frecuencia, bool asociados)
Descripción:	Método que me dice si se pudo modificar un servicio o no.
Nombre:	EliminarServicio (string nombServ)
Descripción:	Método que me dice si se pudo eliminar correctamente un servicio o no.
Nombre:	ListaServiciosExistentes ()
Descripción:	Método que me muestra un listado con todos existentes.
Nombre:	GetListaAtributosDeServicios (string nombre)
Descripción:	Método que me devuelve la lista de atributos de un servicio dado su nombre.
Nombre:	ProcesaServicio (ServicioPersistente serv)
Descripción:	Método que devuelve un arreglo con los atributos simples de un servicio.
Nombre:	GetListaDatosDeServicios (string nombSer)
Descripción:	Método que devuelve una lista de los datos que tiene un servicio determinado.
Nombre:	AdicionarDato (string nombreDato, string tipoDato, string tipo, int tamañoDato)
Descripción:	Método que me dice si un dato fue adicionado correctamente o no.
Nombre:	ModificarDato (string nombreDato, string tipoDato, string tipo, int tamañoDato, string nombreAnt)
Descripción:	Método que me dice si un dato fue modificado o no.

Nombre:	EliminarDato (string nombreDato, string tipoDato, string tipo, int tamañoDato)
Descripción:	Método que me dice si un dato fue eliminado o no.
Nombre:	ObtenerPosServicio (string nomServ)
Descripción:	Método que me devuelve la posición de un servicio en un listado de servicios.
Nombre:	GetListadoTodosDatos ()
Descripción:	Método que me devuelve un listado de todos los datos.
Nombre:	ProcesaListaDatos (List<DatoPersistente> listDatos)
Descripción:	Método que se utiliza para obtener un listado de datos específicos de un servicio.
Nombre:	GetListaDatosNoAsociados ()
Descripción:	Método que me devuelve un arreglo de los datos no asociados a un servicio determinado.
Nombre:	ActualizarServiciosDatos (object[] listBancos)
Descripción:	Método que actualiza todos los servicios y datos de un banco determinado.
Nombre:	ListadoDeBancosId ()
Descripción:	Método que devuelve una arreglo con el nombre y el numero del banco.
Nombre:	ConciliacionesAuxiliaresTransaccion (DateTime fecha,string NumBanco)
Descripción:	Método que manda todas las transacciones en una fecha determinada.
Nombre:	ConciliacionesAuxiliaresReclamaciones (DateTime fecha, string NumBanco)
Descripción:	Método que envía todas las reclamaciones para una fecha determinada.

3.4. Conclusiones

En este capítulo se defieron los diagramas de clases del análisis y diseño para aquellos procesos que responden a funcionalidades de gran importancia para el sistema, para cada uno de estos se elaboraron los diagramas de secuencia para representar el comportamiento entre las clases del diseño, así como la descripción cada una de estas.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

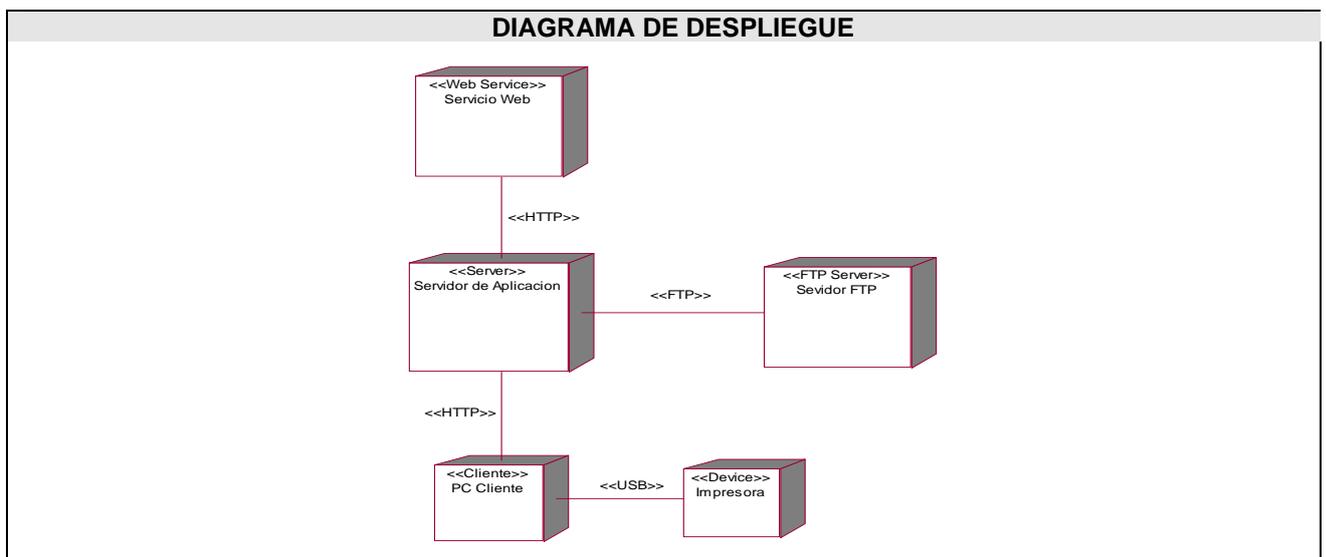
4.1. Introducción

En este capítulo se verán los modelos correspondientes a los flujos de trabajo de implementación y pruebas. Se comenzará con implementación partiendo de los resultados obtenidos en el diseño, para dar entrada a las etapas de pruebas, en esta última se realizará el diseño de los casos de pruebas para aquellos casos de usos que se determinaron de gran importancia para el proyecto correspondientes al modulo Servicio de Pago.

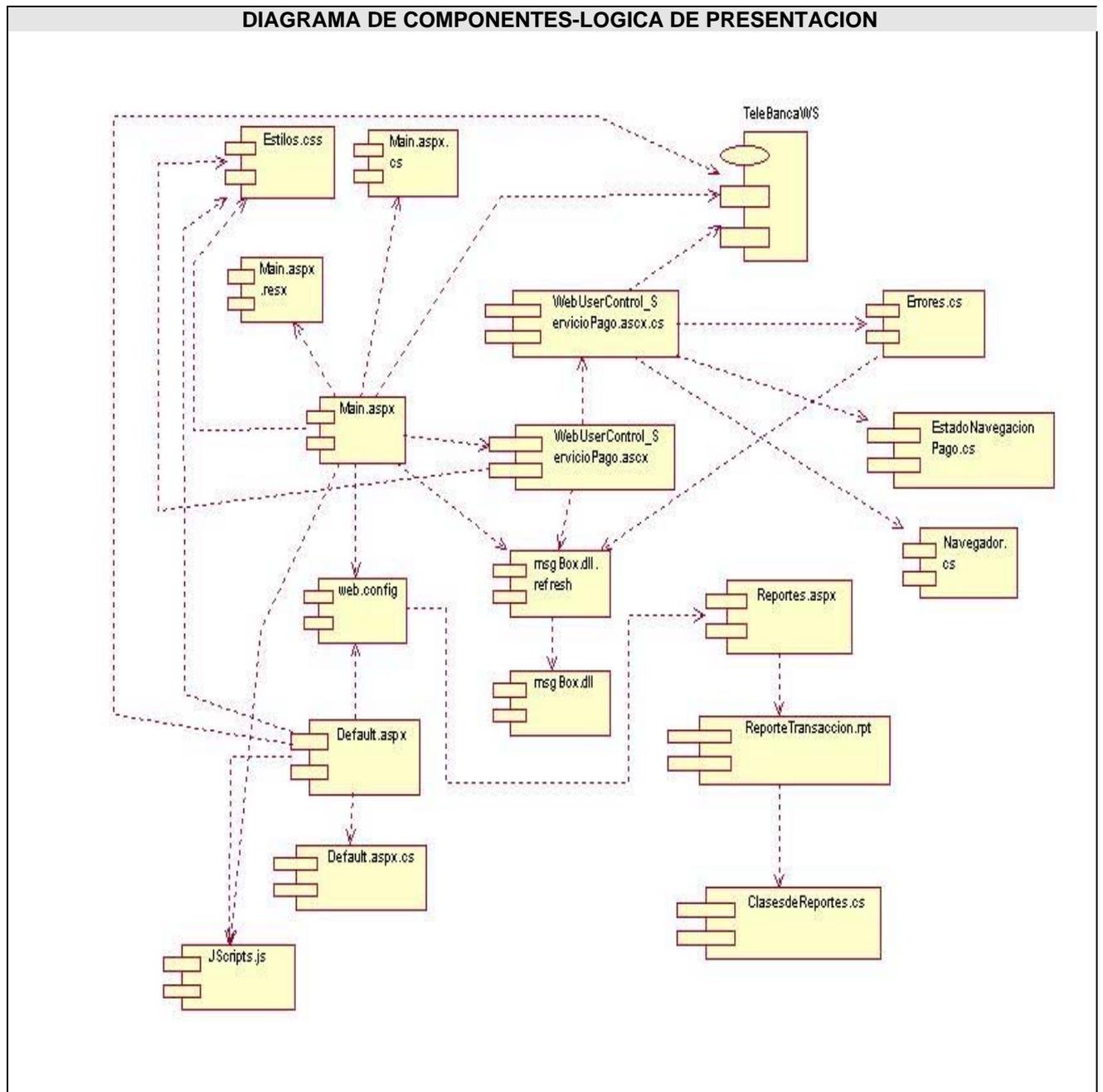
4.2. Implementación

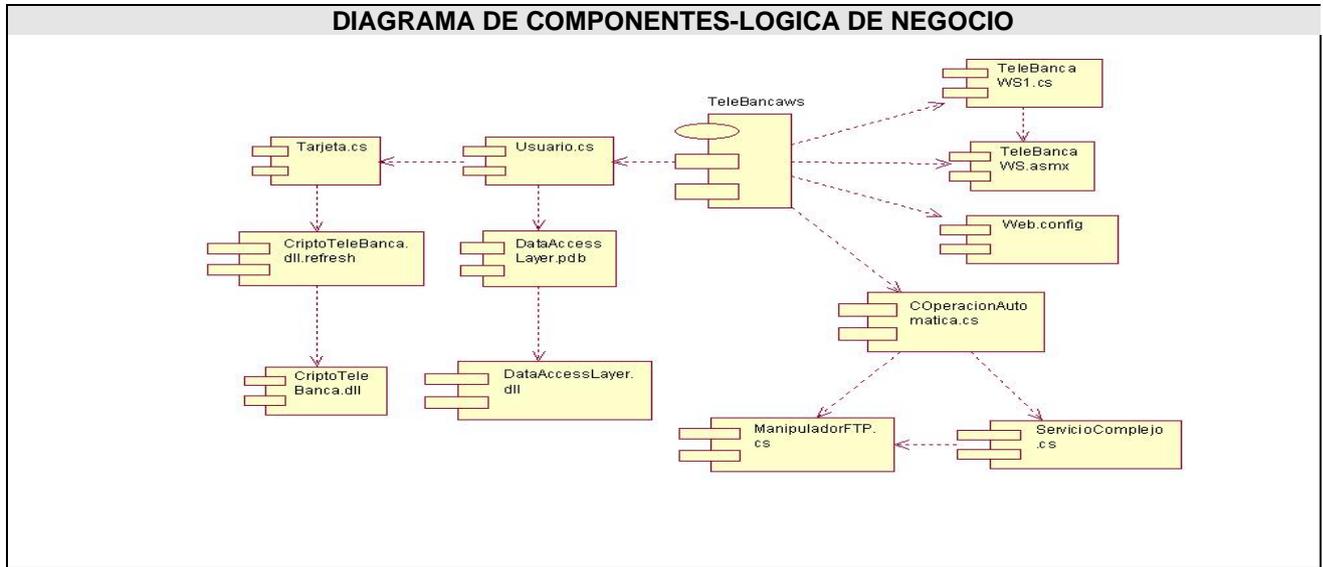
El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Los diagramas de despliegue y componentes, son artefactos que conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará a aplicación. A continuación se representa el diagrama de despliegue y diagrama de componentes correspondiente a la lógica de presentación y negocio del modulo Servicio de pago de Telebanca.

4.2.1. Diagrama de despliegue



4.2.2. Diagrama de componentes





4.3. Modelo de Prueba

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son comunes. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. Mas adelante se encuentran los diseños de los casos de pruebas para los casos de usos significativos desde el punto de vista arquitectónico.

4.3.1. Diseño de los casos de usos de prueba

Nombre del caso de uso: Configurar Datos (Insertar Datos)

Entrada	Resultados	Condiciones
El Administrador de la BT introduce fecha (Nombre del dato).	El sistema solícita confirmación de guardar la información. “¿Confirma que desea guardar la información?”	El Administrador de la BT debe haber seleccionado la opción Insertar y que exista conexión con la base de datos.
El administrador	El Sistema muestra el mensaje “Entrada	El Administrador de la BT debe haber

de la BT deja el campo en blanco.	de datos no válida". El Sistema muestra el formulario para insertar la información referente a un Dato.	seleccionado la opción Insertar y que exista conexión con la base de datos.
El Administrador de la BT introduce un string (Tipo).	El sistema solicita confirmación de guardar la información. "¿Confirma que desea guardar la información?"	El Administrador de la BT debe haber seleccionado la opción Insertar y que exista conexión con la base de datos.
El Administrador de la BT introduce un número (Tamaño del Dato).	El sistema solicita confirmación de guardar la información. "¿Confirma que desea guardar la información?"	El Administrador de la BT debe haber seleccionado la opción Insertar y que exista conexión con la base de datos.
El Administrador de la BT introduce fddgdftgr.	El Sistema muestra el mensaje "Entrada de datos no válida". El Sistema muestra el formulario para insertar la información referente a un Dato.	El Administrador de la BT debe haber seleccionado la opción Insertar y que exista conexión con la base de datos.

Nombre del caso de uso: Configurar Datos (Modificar Dato)

Entrada	Resultados	Condiciones
El Administrador de la BT introduce fecha (Nombre del Dato)	El sistema solicita confirmación de guardar la información. "¿Confirma que desea guardar la información?"	El Administrador de la BT de haber seleccionado la opción de Modificar un Dato y que exista conexión con la base de datos.
El administrador de la BT introduce un nombre de dato repetido.	El Sistema muestra un mensaje. " Existe otro Dato con las mismas características". El Sistema muestra la información del Dato seleccionado.	El Administrador de la BT de haber seleccionado la opción de Modificar un Dato y que exista conexión con la base de datos.
El Administrador de la BT deja el	El Sistema muestra el mensaje "Entrada de datos no válida". El Sistema muestra	El Administrador de la BT de haber seleccionado la opción de Modificar un

campo en blanco.	la información del Dato seleccionado.	Dato y que exista conexión con la base de datos.
El Administrador de la BT introduce un int(Tipo).	El sistema solicita confirmación de guardar la información. “¿Confirma que desea guardar la información?”	El Administrador de la BT de haber seleccionado la opción de Modificar un Dato y que exista conexión con la base de datos.
El Administrador de la BT introduce un número (Tamaño del Dato).	El sistema solicita confirmación de guardar la información. “¿Confirma que desea guardar la información?”	El Administrador de la BT de haber seleccionado la opción de Modificar un Dato y que exista conexión con la base de datos.
El Administrador de la BT introduce fddgdfrgr.	El Sistema muestra el mensaje “Entrada de datos no válida”. El Sistema muestra la información del Dato seleccionado.	El Administrador de la BT de haber seleccionado la opción de Modificar un Dato y que exista conexión con la base de datos.

Nombre del caso de uso: Configurar Servicios (Insertar Servicio Simple)

Entrada	Resultados	Condiciones
El Administrador de la BT introduce un nombre(Nombre del Servicio)	El sistema muestra un mensaje: “¿Confirma que desea guardar la información?”	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Simple además debe existir conexión con la base de datos.
El administrador de la BT deja el campo en blanco.	El Sistema muestra el mensaje “Error en la entrada de datos”. El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Simple además debe existir conexión con la base de datos.
El Administrador	El Sistema muestra el mensaje “El	El Administrador de la BT debe haber

de la BT introduce el nombre Etecsa.	servicio a insertar ya existe". El Sistema muestra un formulario para insertar los datos del nuevo servicio.	seleccionado la opción Insertar y ahí selecciona la opción Simple además debe existir conexión con la base de datos.
El Administrador de la BT introduce tr.	El Sistema muestra el mensaje "Error en la entrada de datos". El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Simple además debe existir conexión con la base de datos.
El Administrador de la BT introduce un número (# Coordenada de matriz)	El sistema muestra un mensaje: "¿Confirma que desea guardar la información?"	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Simple además debe existir conexión con la base de datos.

Nombre del caso de uso: Configurar Servicios (Modificar Servicio Simple)

Entrada	Resultados	Condiciones
El Administrador de la BT introduce un nombre (Nombre del Servicio)	El sistema muestra un mensaje: "¿Confirma que desea guardar la información?"	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.
El administrador de la BT introduce un nombre como tr.	El Sistema muestra el mensaje "Entrada de datos no válida". El Sistema muestra el formulario con la información para modificar los datos del servicio simple escogido.	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.
El Administrador de la BT deja el campo en blanco.	El Sistema muestra el mensaje "Entrada de datos no válida". El Sistema muestra el formulario con la información para modificar los datos del servicio simple	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.

	escogido.	
El Administrador de la BT introduce el nombre Etecsa.	El Sistema muestra el mensaje “El servicio a insertar ya existe”. El Sistema muestra el formulario con la información para modificar los datos del servicio simple escogido.	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.
El Administrador de la BT introduce un número (# Coordenada de matriz)	El sistema muestra un mensaje: “¿Confirma que desea guardar la información?”	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.

Nombre del caso de uso: Configurar Servicios (Insertar Servicio Complejo)

Entrada	Resultados	Condiciones
El Administrador de la BT introduce un nombre (Nombre del Servicio)	El sistema muestra un mensaje: “¿Confirma que desea guardar la información?”	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Complejo además debe existir conexión con la base de datos.
El administrador de la BT deja el campo en blanco.	El Sistema muestra el mensaje “Error en la entrada de datos”. El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Complejo además debe existir conexión con la base de datos.
El Administrador de la BT introduce el nombre Eléctrica.	El Sistema muestra el mensaje “El servicio a insertar ya existe”. El Sistema muestra un formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Complejo además debe existir conexión con la base de datos.

El Administrador de la BT introduce tr.	El Sistema muestra el mensaje "Error en la entrada de datos". El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Complejo además debe existir conexión con la base de datos.
El Administrador de la BT introduce un número (# Coordenada de matriz)	El sistema muestra un mensaje: "¿Confirma que desea guardar la información?"	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Complejo además debe existir conexión con la base de datos.
El Administrador de la BT introduce un número (Frecuencia)	El Sistema muestra el mensaje "Error en la entrada de datos". El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado la opción Insertar y ahí selecciona la opción Complejo además debe existir conexión con la base de datos.

Nombre del caso de uso: Configurar Servicios (Modificar Servicio Complejo)

Entrada	Resultados	Condiciones
El Administrador de la BT introduce un nombre (Nombre del Servicio)	El sistema muestra un mensaje: "¿Confirma que desea guardar la información?"	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.
El administrador de la BT deja el campo en blanco.	El Sistema muestra el mensaje "Error en la entrada de datos". El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.

El Administrador de la BT introduce el nombre Eléctrica.	El Sistema muestra el mensaje “El servicio a insertar ya existe”. El Sistema muestra un formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.
El Administrador de la BT introduce tr.	El Sistema muestra el mensaje “Error en la entrada de datos”. El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.
El Administrador de la BT introduce un número (# Coordenada de matriz)	El sistema muestra un mensaje: “¿Confirma que desea guardar la información?”	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.
El Administrador de la BT introduce un número (Frecuencia)	El Sistema muestra el mensaje “Error en la entrada de datos”. El Sistema muestra el formulario para insertar los datos del nuevo servicio.	El Administrador de la BT debe haber seleccionado un servicio y la opción Modificar además debe existir conexión con la base de datos.

Nombre del caso de uso: Pagar Servicio

Entrada	Resultados	Condiciones
La Operadora de Servicio de Pago introduce el número 0141355450(# de	El Sistema muestra el nombre y apellido del cliente correspondiente a la tarjeta y dos campos para comprobar dos dígitos específicos del PIN.	Un cliente debe haber solicitado a la Operadora de Servicio de Pago pagar un servicio y que exista conexión con la base de datos.

tarjeta).		
La Operadora de Servicio de Pago deja el campo en blanco.	El Sistema muestra un asterisco al lado de cada entrada no válida, el Sistema muestra el mensaje "Entrada de datos no válida ", el Sistema solicita el número de la tarjeta de identificación del cliente para la Banca Telefónica.	Un cliente debe haber solicitado a la Operadora de Servicio de Pago pagar un servicio y que exista conexión con la base de datos.
La operadora de Servicio de Pago introduce 8594943302.	El Sistema muestra el mensaje "El número de la tarjeta no se encuentra registrada en la Banca Telefónica", el Sistema solicita el número de la tarjeta de identificación del cliente para la Banca Telefónica.	Un cliente debe haber solicitado a la Operadora de Servicio de Pago pagar un servicio y que exista conexión con la base de datos.
La Operadora de Servicio de Pago introduce 5 y 8 (dígitos dentro del PIN).	El sistema solicita una coordenada específica de la matriz.	La Operadora de Servicio de Pago debe haber entrado un número de tarjeta valido y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago introduce un 9 y 4.	El Sistema muestra el mensaje "Los Dígitos del PIN son erróneos ", el Sistema solicita introducir dos dígitos del PIN.	La Operadora de Servicio de Pago debe haber entrado un número de tarjeta valido y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago introduce un 8.	El Sistema muestra un asterisco al lado de cada entrada no válida, el Sistema muestra el mensaje "Entrada de datos no válida ", el Sistema solicita introducir dos dígitos del PIN.	La Operadora de Servicio de Pago debe haber entrado un número de tarjeta valido y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago	El Sistema muestra un asterisco al lado de cada entrada no válida, el Sistema	La Operadora de Servicio de Pago debe haber entrado un número de tarjeta valido

deja los campos en blanco.	muestra el mensaje “Entrada de datos no válida “, el Sistema solicita introducir dos dígitos del PIN.	y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago introduce 67 (# Coordenada de la matriz)	El Sistema solicita la identificación de la transacción y la fecha en que se realizó la misma.	La Operadora de Servicio de Pago debe haber entrado los dígitos del PIN correctos y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago introduce 47.	El Sistema muestra un el mensaje “La Coordenada es errónea “, El Sistema solicita una coordenada de la matriz.	La Operadora de Servicio de Pago debe haber entrado los dígitos del PIN correctos y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago deja el campo en blanco.	El Sistema muestra un asterisco al lado de la entrada no válida, el Sistema muestra el mensaje “Entrada de datos no válida “, El Sistema solicita una coordenada de la matriz.	La Operadora de Servicio de Pago debe haber entrado los dígitos del PIN correctos y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago introduce 50 (Datos de Pago)	El sistema verifica que se actualizaron todos los datos de pago, el sistema envía los datos de la transacción y los datos de pago al Web Services.	La Operadora de Servicio de Pago debe haber entrado el valor correcto que pertenecía a las coordenadas de la matriz y presionado el botón Aceptar, además que exista conexión con la base de datos.
La Operadora de Servicio de Pago introduce 5.	El sistema muestra un asterisco en los campos con errores, el sistema muestra el mensaje” Entrada de datos no válida”, el Sistema muestra un formulario con los datos de pago definidos por el servicio y la opción de “Actualizar”.	La Operadora de Servicio de Pago debe haber entrado el valor correcto que pertenecía a las coordenadas de la matriz y presionado el botón Aceptar, además que exista conexión con la base de datos.

La Operadora de Servicio de Pago deja el campo en blanco.	El sistema muestra un asterisco en los campos con errores, el sistema muestra el mensaje "Entrada de datos no válida", el Sistema muestra un formulario con los datos de pago definidos por el servicio y la opción de "Actualizar".	La Operadora de Servicio de Pago debe haber entrado el valor correcto que pertenecía a las coordenadas de la matriz y presionado el botón Aceptar, además que exista conexión con la base de datos.
---	--	---

4.4. Conclusiones

En este capítulo se vieron los artefactos que se generan durante las actividades de los flujos de trabajo de implementación y pruebas, obteniendo como resultados el diagrama de componente, modelo de despliegue y el diseño de los casos de usos de pruebas respectivo a los procesos importantes desde el punto de vista de la arquitectura del proyecto Telebanca.

CONCLUSIONES

El proceso de creación del presente trabajo ha atravesado diferentes etapas dando cumplimiento a los objetivos del mismo. Este proceso constó de una etapa inicial de investigación, en la cual se buscó información acerca de los métodos de pago de los servicios existentes en Cuba y en el mundo, además de una breve investigación de las bases tecnológicas para resolver el problema planteado. Posteriormente se pasó a una etapa de análisis y diseño en la cual se obtuvieron todos los artefactos necesarios para llegar a la solución esperada. En la siguiente fase se implementaron las capas de presentación y negocio correspondientes al módulo servicio de pago. Ya una vez terminada la implementación se dio inicio a la realización de las pruebas en vista de verificar la calidad y cumplimiento de cada una de las funcionalidades vinculadas a los procesos comprendidos dentro del módulo Servicio de Pago. Con la culminación y puesta en funcionamiento del presente trabajo se espera proceso rápido, cómodo y seguro de la gestión del pago de los servicios que se prestan actualmente a la población y que se acogen al método de pago a través de la banca telefónica.

RECOMENDACIONES

Hacer público el servicio que brinda Telebanca mediante el noticiero, radio u otros medios de difusión masiva para que todo el que desee disfrutar del mismo lo pueda hacer, también en cada uno de los centros de trabajo que tenga que ver con las empresas que se han acogido al servicio de Telebanca, se le puede dar información a la población de cómo es el funcionamiento de dicho servicio y las ventajas que tiene el mismo para que estén bien informados.

Desarrollo de una aplicación Web que permita a los clientes o las empresas asociadas realizar las operaciones de forma personal en la Web y no a través de una Operadora, de esta forma le permitiría a los usuarios que tienen la posibilidad de conexión ya sea en su trabajo o desde su casa realizar el proceso desde allí, sin restringir el servicio de Operadora para los demás clientes que no tengan dicha oportunidad.

Continuar trabajando en la mejora de los resultados de la aplicación, para de esta forma perfeccionar aun más los servicios que presta la misma.

Por último se recomienda la puesta en funcionamiento para todas las provincias del país la mayoría de los servicios que están asociados a Telebanca, ya que actualmente solo se brinda el servicio de pagar el teléfono que es con la Empresa Etecsa.

BIBLIOGRAFÍA

1. **Parihar, Mridula.** *La biblia de ASP. NET.* Madrid : ANAYA MULTIMEDIA, S.A, 2002. ISBN.
2. **Jacobson, G.Booch, J Rumbaugh.** *El Proceso Unificado de Desarrollo de Software.* Madrid : ISBN, 2000.
3. **Schmuller, Joseph.** *Aprendiendo UML en 24 HR.* México : PEARSON EDUCATION, 2000.
4. **Sánchez, Pedro Herrarte.** Introducción a .NET. *djk.* [En línea] 21 de 02 de 2006. [Citado el: 20 de marzo de 2007.] http://www.devjoker.com/asp/ver_contenidos.aspx?co_contenido=25&as_categoria=1.
5. **C., Benjamín González.** Introducción al entorno de desarrollo de Microsoft .NET. *DesarrolloWeb.com.* [En línea] [Citado el: 15 de abril de 2007.] <http://www.desarrolloweb.com/articulos/1680.php>.
6. **Jeff Ferguson, Brian Patterson,.** *La biblia de.* Madrid : ANAYA, 2003. ISBN.
7. **GEETANJALI ARORA, BALASUBRAMANIAM AIASWAMY, NITIN PANDEY.** *C#.* Madrid : ANAYA MULTIMEDIA, 2002. ISBN.
8. **McDonald, Matthew.** *ASP.NET. Manual de referencia.* (Madrid : McGRAW-HILLIINTERAMERICANA DE ESPANA, S.A.U., 2002. ISBN.
9. **Sant'Anna, Mauro.** Reportes en .NET con Crystal Reports. *MSDN.* [En línea] [Citado el: 10 de marzo de 2007.] <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art11.asp>.

ARCHER, T. A *FONDO C#.* McGRAW-HILLIINTERAMERICANA DE ESPmA, S. A. U., 2001. p. 84-481-3246-7

ARGENTARIA, B. B. V. *Línea BBVA: Servicios avanzados de banca telefónica.* Disponible en: http://www.cervantes.es/seg_nivel/lect_ens/oesi/Estudios%20de%20caso/bbva.PDF.

CABELLO, L. M. C. *Modelo de dimensionamiento de un Call Center basado en simulación de sistemas,* 2006. [1]. Disponible en: <http://www.pucp.edu.pe/facultad/ingenieria/seccion/industrial/docs/Modelo%20de%20dimensionamiento%20de%20un%20call%20center%20basado%20en%20la%20simulacion%20de%20sistemas.pdf>.

COMUNICACIONES, U. E. *CALL CENTER.* Disponible en: [http://www.union-network.org/uniindep.nsf/4f88980a9911f5eec1256db2003d8895/8c922a8af61a70a8c1256dc600479468/\\$](http://www.union-network.org/uniindep.nsf/4f88980a9911f5eec1256db2003d8895/8c922a8af61a70a8c1256dc600479468/$)

[FILE/JORNADA%20CALL%20CENTER.pdf](#)

CONDUSEF. *Banca por Teléfono*, 2004. [2007]. Disponible en:
http://www.condufef.gob.mx/Seccion_Fija/ftc/ftc_20.pdf

CORPORATION, M. *Introducción a Visual Studio* 2007. [Disponible en: [http://msdn2.microsoft.com/es-es/library/6x6bk1f4\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/6x6bk1f4(VS.80).aspx)

GÓMEZ, M. *Despachar con el Banco de un 'Telefonazo'*, 1998. [117]. Disponible en:
<http://www.elmundo.es/sudinero/noticias/act-117-08.html>

GONZÁLEZ, R. M. *Marketing en el Siglo XXI*: 27.

MARAÑÓN, G. Á. *Banca telefónica y cajeros automáticos*, 2000. [Disponible en:
<http://www.iec.csic.es/cryptonicon/comercio/primerospasos.html>

MARTÍN, J. *Entrevista a José Luis Goytre*, 2003. [Disponible en: http://www.icemd.com/area-entrada/experto/entrev_anteriores.asp?id=69

MIKE YOUNG, C. W. Y. *Deploying Solutions with .NET Enterprise Servers*.

NÚÑES, M. *Call Center: Una efectiva ruta para llegar a sus clientes*. *Call News*, Entel Call Center, 2004. 13.

PLASENCIA, F. H. *Los call centers, una herramienta de CRM en apoyo al e-commerce* 2004. [Disponible en: <http://www.masterdisseny.com/master-net/claves/00026.php3>

SECO, J. A. G. *El lenguaje de programación C#* 2001. p.

STURM, J. *Desarrollo de soluciones XML*: 367.

TEIXIDÓ, C. *Falencias en el desarrollo de los Call Center*, 2004. [Disponible en:
http://www.plusonline.cl/art_falencias.htm

WENDY BOGGS, M. B. *UML with Rational Rose* 2002, 2002: 701.

GLOSARIO

Cuentas de Ahorro asociados a Tarjetas Débito RED: Cuentas de Ahorro abiertas por el Cliente en bancos emisores de Tarjetas Débito RED.

Red de Cajeros Automáticos: Cajeros Automáticos conectados a un Centro de Control a través de una red de transmisión de datos.

Cliente: Persona que utiliza el servicio brindado por las Empresas (En este caso es el que accederá al pago del mismo a través de la Banca Telefónica).

Centro de Llamadas: Lugar único donde se recibirán todas las llamadas telefónicas, las operadoras ejecutarán las transacciones a partir de las indicaciones recibidas por el cliente.

Servicios de Banca Telefónica: Servicios que pueden ser pagados a través de la modalidad de Banca Telefónica (Agua, Luz, Teléfono, etc.).

Transacciones: Operaciones a realizar asociadas a una indicación recibida por el cliente y que pueden involucrar a varios entes indistintamente (Cliente, Empresa, Sucursal Electrónica, Sabic, etc.). Cada tipo de transacción debe ser identificada de manera única.

Tarjeta: Tarjeta necesaria para poder acceder el cliente a los servicios de Banca Telefónica. Contribuye a la seguridad del sistema.

Traza de las transacciones: Número que identifica de forma única en el día una transacción.

Saldo: Transacción que posibilita consultar el saldo contable de la cuenta de ahorro del cliente.

Pago de servicio: Transacción que posibilita realizar el pago de servicio solicitado teniendo como efecto final el correspondiente movimiento contable en la cuenta de ahorro del cliente.

Pago simple: método de pago de un servicio, el cual se realiza sin necesidad de conocer la deuda del cliente respecto a este.

Pago complejo: método de pago de un servicio, el cual se realiza para el cual es necesario conocer la deuda del cliente respecto al mismo

Matrices: Cuadro compuesto con números colocados en filas y columnas. La matriz se imprime en las tarjetas de autenticación del servicio de Banca Telefónica.

PIN: Numero de Identificación Personal (clave) asociado a una tarjeta.