



Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

“Análisis y Diseño en UML de un Centro Comercial Virtual enfocado a Pequeñas y Medianas Empresas (PyMEs) Mexicanas”

TESIS

Que para obtener el Título de

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

Jesús Ocaña Zúñiga

Asesor

Dr. Mario Rossainz López

Puebla, Pue. Septiembre 2006

A DIOS

A mi papá Jesús y a mi mamá Amanda
A mi hermana Amanda Guadalupe

Resumen

En esta tesis se presenta una propuesta de aplicación Web a partir de la cual las pequeñas y medianas empresas mexicanas tengan la posibilidad de realizar comercio electrónico. Esta aplicación Web se describe como un Centro Comercial Virtual (CCV) en donde la idea principal es conglomerar a un número importante de este tipo de empresas con el propósito de que juntas puedan constituir una opción de compra para los usuarios potenciales de la Internet al mismo tiempo que cada empresa en particular se hace de los beneficios inherentes que el comercio electrónico proporciona a las empresas que lo incluyen como parte de su plan de negocios y, de esta forma, propiciar el crecimiento de la misma. En correspondencia, el objetivo principal de esta tesis es la realización del análisis y el diseño de dicha aplicación Web con el propósito de describir a cada una de las funcionalidades que el Centro Comercial Virtual debe cumplir para con cada uno de los diferentes usuarios que harán uso de él. Para tal efecto se utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) en el marco de la Ingeniería Web Basada en UML (UML Web Engineering, UWE). En síntesis, lo que en UWE se propone es la extensión en la semántica del UML puro hacia una nueva semántica que represente de manera adecuada a cada uno de los elementos asociados al desarrollo de aplicaciones Web. Aunque principalmente se utilizan las extensiones de UML descritas en UWE en este trabajo también se toman en consideración, aunque de manera minoritaria, otras propuestas de extensión del UML para el modelado de aplicaciones Web. Al final, como resultado del trabajo de análisis y diseño, se obtiene una colección de diagramas en UML extendido que se presentan a lo largo de todo en el documento y que, en su conjunto, constituyen el modelo que describe al Centro Comercial Virtual y que ha de utilizarse en un futuro para la construcción de un prototipo de la aplicación Web referida.

Por último cabe mencionar que, como parte del trabajo de investigación de esta tesis, se incluye al principio del documento el marco teórico en donde se exponen los fundamentos conceptuales de este trabajo. Así también se incluye, sobre el final, un anteproyecto de implementación en el que se presentan algunos lineamientos generales que se sugieren tomar en cuenta en el futuro proceso de implementación de la aplicación que se describe con el modelo.

Índice General

Introducción	5
Motivación	5
Objetivos Generales y Específicos	5
Organización del Documento	6
1 Marco Teórico	7
1.1 Comercio Electrónico	7
1.2 Las Pequeñas y Medianas Empresas Mexicanas	12
1.3 Las PyMEs Mexicanas y el Comercio Electrónico	15
1.4 Un Centro Comercial Virtual (CCV) para las PyMEs Mexicanas	18
1.5 Esquemas de Comercio Electrónico existentes orientados a la Pequeña y Mediana Empresa de México	19
1.6 El modelado de aplicaciones Web	22
1.7 Generalidades de las Aplicaciones Web	23
2 Introducción al análisis y diseño	26
2.1 La Ingeniería Web basada en UML	26
3 Especificación de Requerimientos	32
3.1 La asociación entre candidatos Vendedores y el CCV	32
3.2 Un primer acercamiento a la arquitectura propuesta	35
3.3 Diagramas de Casos de Uso	37
4 Modelo Lógico Conceptual	51
4.1 Modelo Lógico Conceptual: Diagrama de Clases	51
5 Modelo de Navegación	54
5.1 Estereotipos utilizados en el modelo de navegación	54
5.2 Semántica de convenciones utilizadas en el modelo de navegación	58
5.3 Semántica de estereotipos asociada con la Web	65
5.4 Modelo de Navegación de los subsistemas	70
6 Interacción Temporal	93
6.1 Semántica de convenciones utilizadas en los diagramas de secuencia	93
6.2 Secuencias del CCV	96

7 Escenarios Web	144
7.1 Los escenarios Web y la semántica asociada a los diagramas de estados	144
7.2 Escenarios Web para el CCV	146
8 Esquema de implementación	157
8.1 Semántica de elementos asociados al diagrama de componentes	157
8.2 Esquema de Implementación del CCV	157
9 Anteproyecto de Implementación	160
9.1 El equipo de desarrollo	160
9.2 Planificación preliminar de la implementación	161
9.3 Sugerencia de Lenguaje de programación, servidor de base de datos y servidor Web	164
9.4 Herramientas para la implementación de aplicaciones Web	166
9.5 Mecanismo de Prueba del Modelo	169
Conclusiones y Trabajo Futuro	187
Bibliografía	190

Introducción

Motivación

El advenimiento de Internet y su utilización en casi todas las áreas de desarrollo de las personas ha provocado una tendencia cada vez más marcada que consiste en la migración hacia la llamada e-sociedad. En el ámbito de los negocios, hoy en día la Red de Redes se ha consolidado como una herramienta poderosa para mejorar el ciclo productivo de las empresas, cualquiera que este sea su giro. Cada día, más y más empresas añaden a su estrategia de mercado la frase: punto com. Esta frase de dos palabras se ha convertido en un factor medular a tomar en cuenta por las empresas para adecuar su estrategia de mercado a un ambiente de intensa competencia que es el comercio electrónico.

Sin embargo, como cualquier innovación tecnológica de alto impacto, su aprovechamiento e integración al entorno regional varía en función de aspectos culturales, económicos, políticos y sociales de cada región o país.

En este contexto, la principal motivación de este trabajo radica en aportar conocimiento relacionado con el desarrollo de un esquema de comercio electrónico apropiado para México, el cual pueda, en un futuro, ser integrado con éxito en el sector productivo nacional y de esta manera favorecer el desarrollo. Para ello, se toma como dominio particular de este proyecto a las Pequeñas y Medianas Empresas (PyMEs) las cuales constituyen el más importante segmento de vida económica del País.

Objetivos Generales y Específicos

Objetivo General

Llevar a cabo el análisis y diseño de un Centro Comercial Virtual (CCV) como una aplicación Web en el lenguaje unificado de modelado (UML), que describa un esquema que permita a las pequeñas y medianas empresas (PyMEs) mexicanas incursionar en el mundo del comercio electrónico.

Objetivos Específicos

- Definir el marco de trabajo respecto al UML y el modelado de aplicaciones Web.
- Especificar las funcionalidades que debe cubrir el Centro Comercial Virtual a partir de la captura de requerimientos.
- Realizar el análisis y diseño del Centro Comercial Virtual, definiendo los elementos que integran la arquitectura, las relaciones entre ellos, el ambiente y las funcionalidades provistas. Todo esto en función de los requerimientos especificados.
- Sugerir lineamientos generales para la construcción a futuro del modelo de implementación por medio de un anteproyecto y un mecanismo de prueba de la aplicación Web modelada.

Organización del Documento

Este documento de tesis se encuentra organizado de la siguiente manera: En el capítulo 1 se establece el marco teórico adecuado al problema. En dicho capítulo se hace referencia a conceptos relacionados con el comercio electrónico, generalidades de las Pequeñas y Medianas empresas de México y la realidad actual del Comercio Electrónico. Posteriormente se comentan algunos esquemas de comercio electrónico utilizables por las PyMEs. Sobre final del capítulo 1 se hace referencia al modelado de aplicaciones Web, finalizando con algunas generalidades de las mismas. En el capítulo 2 se presenta una introducción al análisis y diseño donde se describe el marco de trabajo a tomarse en cuenta en el desarrollo del modelo. En el capítulo 3 se inicia con el análisis y diseño, primeramente con un 1er acercamiento a la arquitectura propuesta y posteriormente con la especificación de requerimientos por medio de diagramas de casos de uso. En el capítulo 4 se presenta el modelo lógico-conceptual como antecedente del modelo de navegación que se presenta en el capítulo 5. Al comienzo del capítulo 5 se describe la semántica de los estereotipos utilizados en el modelo de navegación. El capítulo 5 finaliza con la presentación de los diagramas de clases estereotipadas que constituyen el modelo de navegación. El capítulo 6 contiene los diagramas de secuencia que describen interacciones temporales entre elementos descritos en el modelo de navegación. En el capítulo 7 se presenta una colección de escenarios Web descritos mediante diagramas de estados. En el capítulo 8 se proporciona una vista general del esquema de implementación adecuado para el CCV mediante de un diagrama de componentes. En el capítulo 9 se presenta un anteproyecto de implementación que consiste de una serie de sugerencias relacionadas con la fase de implementación del modelo y de un caso de ejemplo particular como mecanismo de prueba del modelo. Al final del documento se presentan las conclusiones y el trabajo futuro.

Capítulo 1

Marco Teórico

1.1 Comercio Electrónico

En un tiempo notablemente corto, el Internet ha pasado de ser un recurso opcional de comunicación a ser un medio vital e imprescindible para la vida diaria de la sociedad, incluyendo los negocios. A la par de la Internet, la Web evoluciona a pasos agigantados, es por esto que la manera de dirigir y administrar negocios de manera exitosa debe expandir sus horizontes a fin de adaptarse y obtener los mayores beneficios de la Internet. Uno de los vehículos principales del cambio es el Comercio Electrónico, el cual permite a las empresas ampliar su campo de acción, dar mejor respuesta a las necesidades de los Clientes, y lo más importante, ofertar sus productos en un mercado global.

En este apartado se describen conceptos, definiciones y generalidades introductorias para el comercio electrónico que, en su conjunto, establecen un marco de referencia para el resto del documento.

1.1.1 Definición de Comercio Electrónico

Algunos estudiosos opinan que, en cierta forma, el comercio electrónico surgió aún antes de Internet, mediante transacciones comerciales por teléfono y fax, pero el desarrollo de la Web motivó que alcanzara mayor auge, por su masividad y rapidez de operación. La acepción más general que se puede hacer de comercio electrónico es la de “acercar el comprador al fabricante por medios electrónicos”, lo cual implica eliminación de intermediarios, reducción de costos y una filosofía diferente en la forma de comprar y vender, y lo que es más importante, de obtener información para esas gestiones. Las siguientes son algunas definiciones alternativas aceptadas para el término Comercio Electrónico.

Definición 1: El comercio electrónico se refiere a cualquier forma de transacción de negocios en la cual las partes interactúan electrónicamente en lugar de hacerlo mediante intercambios materiales o contacto físico directo [Fernández00].

Definición 2: Comercio electrónico es hacer negocio en línea [Loshin04].

Definición 3: El Comercio Electrónico es un concepto de negocio, de cambios de procesos en las organizaciones, de nuevas formas de abordar las relaciones comerciales entre las empresas y sus clientes que, evidentemente, se sustenta sobre una base tecnológica para llevar a cabo su cometido [ProcyCom03].

Definición 4: El comercio electrónico se refiere a cualquier actividad de negocio que se lleva a cabo utilizando un medio electrónico, frecuentemente la Web [Gangopadhyay02].

Definición 5: El Comercio Electrónico se puede definir como la aplicación de la tecnología para la automatización de las transacciones de negocios [Cohen00].

De lo anterior podemos decir que el comercio electrónico consiste en llevar a cabo cualquier actividad de negocio relacionada con la comercialización de productos o servicios a través de redes informáticas, aplicando para ello tecnologías de información dirigidas a realizar dichas transacciones comerciales.

1.1.2 Actores y Transacciones en Sistemas de Comercio Electrónico

Al igual que en el comercio presencial, en el comercio electrónico intervienen diversos actores que deben interactuar entre sí para llevar a cabo las transacciones de compra-venta. Los actores mínimos de un sistema de comercio electrónico:

- **Comprador:** Es todo aquel que adquiere un producto o servicio.
- **Vendedor:** Aquel que promociona y vende un producto o servicio.
- **Entidades y Servicios Financieros:** Validan y autorizan pagos y realizan los movimientos de dinero entre comprador y comerciante.

Cabe comentar que las entidades y servicios financieros pueden ser diversos aún en una misma transacción, por ejemplo: el banco del comprador, el banco del Vendedor, la pasarela entre bancos, marca de plásticos (débito o crédito), el broker para micropagos (corredor o agente que actúa como intermediario entre un comprador y un Vendedor, usualmente cobrando una comisión, por ejemplo Paypal), por citar algunos.

Todos estos actores en una transacción han de estar comunicados mediante la red (Internet). Dependiendo de la forma de pago, sistemas de seguridad y otros factores de logística pueden entrar en acción en la transacción otros actores como lo pueden ser Empresas de Paquetería y transporte para el envío de los productos.

Una transacción electrónica siempre se realiza con al menos 4 fases:

1. El comprador obtiene los datos del producto o servicio del vendedor
2. El comprador paga y se solicita la autorización de dicho pago.
3. Se confirma la autorización y se paga al vendedor (a veces el pago se realiza después de la cuarta fase)
4. Se entrega el producto o servicio mediante un transporte físico o por la red.

Un sistema de comercio electrónico debe de ser capaz de automatizar las 3 primeras fases, debe presentar, promocionar y facilitar la adquisición del bien, gestionar la autorización del pago y confirmado el pago debe iniciar el proceso de entrega del bien [Loshin04].

1.1.3 Tipos de Comercio electrónico

Con la evolución de las tecnologías de información y las crecientes necesidades de las organizaciones han surgido diferentes maneras de hacer comercio electrónico. A estas formas se le han llamado tipos de Comercio electrónico (algunos autores le llaman clasificación de Comercio electrónico) y se distinguen esencialmente por el tipo de actores (entiéndase usuarios) que participan en él. De entre los más importantes podemos mencionar los siguientes:

- **Negocio a Clientes (Business to Customers, B2C):** Es el comercio electrónico que realizan las empresas con los particulares. En esta categoría se incluyen todos aquellos negocios que utilizan Internet para comercializar cualquier tipo de bienes y/o servicios al público en general, desde libros, juguetes, equipos electrónicos, etc, de manera individual. Este tipo de transacción permite a las empresas mostrar su oferta en línea y mostrar sus productos a través de catálogos. En este tipo de comercio electrónico el segmento de clientes es toda la red. Los expertos aseguran que la gran ventaja de esta forma de venta en línea es que no se necesitan grandes cantidades de inventario físico sino solo rápidas soluciones de distribución.
- **Negocio a Negocio (Business to Business, B2B):** Es el comercio electrónico que realizan las empresas con otras empresas. Se refiere a las transacciones comerciales realizadas entre empresas exclusivamente, generalmente bajo un esquema de distribuidores y proveedores. Involucra a los procesos contractuales electrónicos en los que participan dos o más empresas, bien a través de las personas físicas que las representan o bien a través de sus propios sistemas informáticos. Esta rama es muy amplia y se desarrolla utilizando el Internet como plataforma múltiple y neutral de comunicación.
- **Clientes a Clientes (Customer to Customer, C2C):** Es el comercio electrónico que se realiza entre particulares. Se refiere a las transacciones comerciales

privadas entre personas individuales que pueden tener lugar mediante simplemente el intercambio de correos electrónicos, el uso de tecnologías P2P (Peer to Peer, Punto a Punto) o bien utilizando complejas aplicaciones Web especializadas como Ebay por citar alguna.

- **Negocio a Empleado (Business to Employee, B2E):** Se refiere a la relación comercial que se establece entre una empresa y sus propios empleados. Esta tendencia surgió debido a problemas de seguridad que hicieron que las empresas utilizaran las tecnologías ya no en Internet sino hacia adentro a través de un Intranet para tener mayor control y seguridad en sus transacciones.

Estos 4 esquemas son los que predominan en la red. Adicionalmente, se identifica otra más llamada Negocio a Gobierno (Business to Government, B2G), llamado también “gobierno electrónico”, que permite atender al ciudadano a través de la red para trámites aduanales, certificaciones, seguridad social, pagos, cobros y otros servicios oficiales [Potts01][Márquez04].

1.1.4 Escenario Global del Comercio Electrónico

En los albores del nuevo siglo, el Comercio Electrónico ha emergido a pasos agigantados en la última década y tiene lugar en todo el mundo, y aunque EUA, Japón y Europa son los líderes en este camino, el comercio electrónico es de alcance global, tanto en concepto como en realización. Su desarrollo ha experimentado un crecimiento sostenido a nivel mundial y se pronostica que seguirá esta tendencia y las transacciones por vía electrónica representarán una parte importante dentro del comercio mundial.

La mayor parte de las cifras y datos que colocan al comercio electrónico como factor estratégico en las economías de los países corresponden al mundo desarrollado, básicamente Estados Unidos y Europa, y aún entre ambos hay diferencias notables, pues si bien la penetración de Internet en la población (Porcentaje de la población que utiliza el Internet) en el primero es de aproximadamente de un 68%, en Europa va desde un 49% promedio en países de la Unión Europea hasta un 18 % promedio en el resto de Europa [IWStats05].

El atraso europeo en materia de Comercio Electrónico con respecto a los Estados Unidos tiene diversas causas: la falta de especialización en Comercio Electrónico, el índice de conexiones y de computadoras personales en hogares y empresas y, en general, la infraestructura insuficiente en Telecomunicaciones. Existen además barreras subjetivas, como la credibilidad en la seguridad del medio, dificultades en cambiar los hábitos de compra y consumo (que son variables de acuerdo a la cultura de cada país), la no adecuación de los productos existentes a los requisitos de la venta electrónica (la mayoría del mercado se concentra en música y libros) y las dificultades

en el manejo de Internet para el público no especializado, que constituye el grueso de los consumidores potenciales [Potts01].

En América Latina hay que multiplicar esas dificultades, al menos, por cinco, agregando una débil infraestructura de comunicaciones, falta de conectividad, menor porcentaje de computadoras personales por habitante, carencia de centros de distribución regionales, etc. Además habría que añadir las deficiencias en los sistemas educativos de los países, las políticas económicas y la estabilidad social. Contemplando este escenario tendremos una idea aproximada de lo que significa la brecha que existe entre Europa y los Estados Unidos de América y los países de América Latina en donde, además, la penetración de la Internet en la población es apenas en promedio del 13% [IWStats05]. Sin embargo, aún con todos esos obstáculos, las empresas y entidades estatales y privadas de la región pueden obtener beneficios del Comercio Electrónico como factor de tecnología de cambio, ya que de a poco las aplicaciones de comercio electrónico están experimentando una demanda creciente.

En los próximos años el porcentaje del mundo sin acceso a Internet o que no sepa aprovechar estas herramientas para el comercio estará desinformado y no podrá operar con los mismos criterios que quienes las utilicen. Para muchas naciones en desarrollo, al igual que sucedió con otras actividades informativas y electrónicas como la radio y la televisión, si no se afianzan en ellas a tiempo, la brecha con respecto al mundo desarrollado seguirá creciendo hasta un punto que llenarla será muy difícil, por no decir imposible, pues si las tecnologías anteriores tardaban un año o dos en cambiar de versión o formato, en Internet los cambios se producen en meses [Potts01].

1.1.5 Reflexiones sobre el Comercio Electrónico en México

Resulta evidente que la aparición de Internet ha provocado una serie de cambios fundamentales en el mundo de la información que provocan una multiplicidad de retos a solucionar, sobre todo en aspectos como el comercio. Debido al carácter aun novedoso de la comercialización electrónica en nuestro país, se encuentra en una fase de maduración por lo que se considera que el problema que genera la necesidad de la investigación en este ámbito recae en los esfuerzos para contrarrestar el insuficiente nivel de información en las empresas mexicanas acerca de la naturaleza, características y desarrollo del comercio electrónico como vía para alcanzar un crecimiento económico a mediano y largo plazo.

En México la tasa de penetración del Internet con respecto al total de la población es de aproximadamente 14%. Esto supone un mercado mexicano aproximado de 17 millones de usuarios de la Internet [AMIPCI05], un mercado potencial nada despreciable. Esta cobertura se logra con una base instalada de dispositivos de acceso a Internet que consta de 10.8 millones de computadoras personales, 46.1 millones de teléfonos móviles y 1.7 millones de PDAs, donde, obviamente el grueso del acceso a Internet se

lleva a cabo mediante las PCs. De esta base instalada de PCs el 42 % se encuentra en empresas y 58 % en hogares. De entre las que se encuentran instaladas en empresas el 61% tiene cuenta con una conexión a Internet. ¿Esto que representa? Definitivamente una infraestructura importante (si bien no comparable con la que cuentan países como Estados Unidos o Japón) que puede dar soporte a proyectos de comercio electrónicos exitosos. Y esto lo podemos ver en la cantidad de dinero que se movió en la red bajo la modalidad de comercio electrónico. Según la Asociación Mexicana de Internet (AMIPCI), en México en el 2004 las ventas totales del comercio electrónico al consumidor alcanzaron la cifra de 2,384 millones 687 mil pesos. Por si esto fuera poco en el primer trimestre del 2005, en comparación con el mismo período en 2004, se registró una tendencia de crecimiento del 81%. Sin embargo, estas cifras aun distan mucho de alcanzar su máximo potencial, las ventas por Internet, registraron un porcentaje promedio del 1.4% en el 2004, respecto a las ventas totales de los sectores considerados en este estudio. Esto quiere decir que aun existen mercados potenciales inexplorados para los que aun no se ha incorporado un esquema de comercio electrónico [AMIPCI04].

Las consideraciones anteriores mueven a reflexionar en dos direcciones: Primero, que México es un terreno fértil para hacer comercio electrónico, más sin embargo se deben evitar las aventuras inciertas y mal meditadas, y se deben anteponer soluciones que se traduzcan en beneficios económicos para quienes decidan invertir en ellas. Segundo ¿Cómo aplicar esta tecnología para ser más eficientes y competitivos en escala nacional e internacional? Quien no sea capaz de encontrar las respuestas, tampoco podrá obtener beneficios monetarios ni de ningún otro tipo en el mundo comercial del futuro cercano... ya prácticamente presente.

En conclusión, en México no puede abordarse la era del comercio global con una mentalidad artesanal y provinciana. Hay que difundir y fomentar en las empresas mexicanas una nueva visión del fenómeno comercial y productivo, generar mayor cultura informática, más información y conocimientos, porque para gestionar una corporación moderna con éxito ya no basta con los procedimientos tradicionales. Ese es el desafío y ya no es cosa del futuro, pues la revolución del comercio electrónico ha llegado para quedarse como uno de los aspectos claves del desarrollo.

1.2 Las Pequeñas y Medianas Empresas Mexicanas

En la última década, alrededor del mundo se ha podido observar un reconocimiento creciente al papel que desempeñan las Pequeñas y Medianas Empresas (PyMEs) en el crecimiento y desarrollo de las economías de toda nación. En México, con la llegada del gobierno del cambio en el año 2000, las PyMEs se han convertido en un foco de atención y de debate de gran importancia para la política económica del país; y no es

para menos, ya que las PyMEs son extremadamente importantes en la generación de empleos y riqueza lo que se traduce en bienestar.

El propósito de este apartado es el de introducir las definiciones, conceptos y datos referentes a las PyMEs de México, para que a partir de ellos se pueda resaltar su papel en el contexto económico nacional y la importancia de incorporar las tecnologías de información para promover su crecimiento y su competitividad.

1.2.1 ¿Qué es una PyME?

En ninguna definición se puede pretender recoger todos los elementos que determinan que una empresa sea “pequeña” o “mediana”, ni los que diferencian a las empresas, los sectores o los países en sus distintos niveles de desarrollo [Garza00]. La definición tradicional del concepto de PyMEs se ha basado en varios criterios, que son:

- El número de trabajadores que emplean
- El valor del capital invertido
- El volumen de producción o ventas
- El consumo de energía

Según la definición de la Organización de Cooperación y Desarrollo Económico (OCDE), se considera que los establecimientos que emplean hasta 19 empleados son “muy pequeños”, los que emplean hasta 99 se consideran “pequeñas”, los que emplean entre 100 y 499 personas se consideran “medianos” y los que emplean mas de “500” se consideran “grandes”, sin embargo, muchas de las empresas clasificadas en la categoría de medianas en algunos países de desarrollo se considerarían relativamente grandes, de modo que cabe prever que la definición varíe mucho con las condiciones de cada país [Garza00]. En la práctica, el número de personas empleadas constituye el criterio más común, ahora bien, cualesquiera que sean los criterios aplicados, no se puede evitar que la clasificación sea inexacta. En México la clasificación está basada en el número de personas empleadas y en el estrato de la empresa. En el diario oficial de la federación de fecha 13 de diciembre del 2002 se publicó la siguiente clasificación [DOFMX02]:

Estrato	Número de Trabajadores		
	Industria	Comercio	Servicios
Micro	De 0 a 10	De 0 a 10	De 0 a 10
Pequeña	De 11 a 50	De 11 a 30	De 11 a 50
Mediana	De 51 a 250	De 31 a 100	De 51 a 100
Grande	Más de 250	Más de 100	Más de 100

De acuerdo a la anterior, las pequeñas y medianas empresas en México son aquellas cuyo número de trabajadores va desde los 11 hasta los 250 empleados dependiendo de la actividad productiva que se trate. Este trabajo se enfoca particularmente a las PyMEs comerciales.

1.2.2 Características Generales de las PyMEs

De manera muy general todas las PyMEs comparten casi siempre las mismas características, por lo tanto, se podría decir, que estas son las características generales de las PyMEs [Méndez03]:

- El capital es proporcionado por una o dos personas que establecen una sociedad.
- Los propios dueños dirigen la marcha de la empresa; su administración es empírica.
- Su número de trabajadores empleados en el negocio crece y va de 10 hasta 250 personas.
- Se basan en el trabajo y no en el capital, aunque, en los últimos años utilizan más maquinaria y equipo.
- Dominan y abastecen un mercado más amplio, aunque no necesariamente tiene que ser local o regional, ya que muchas veces llegan a producir para el mercado nacional e incluso para el mercado internacional.
- Están en continuo proceso de crecimiento, la pequeña tiende a ser mediana y está, a su vez, aspira a ser grande.
- Obtienen algunas ventajas fiscales por parte del Estado que algunas veces las considera causantes menores dependiendo de sus ventas y utilidades.
- Su tamaño es pequeño o mediano en relación con las otras empresas que operan en el ramo.

Además de estas características cualitativas, habrá que añadir que en los últimos años las PyMEs gozan de especial atención por parte del Estado Mexicano, lo que se traduce en programas de financiamiento, de impulso a la competitividad y beneficios fiscales.

1.2.3 Importancia de las PyMEs en México

Las pequeñas y medianas empresas, PyMEs, tienen particular importancia para la economía nacional, no solo por sus aportaciones a la producción y distribución de bienes y servicios, si no también por la flexibilidad de adaptarse a los cambios tecnológicos y gran potencial de generación de empleos. Representan un excelente medio para impulsar el desarrollo económico y una mejor distribución de la riqueza. Hoy día, los gobiernos de países en desarrollo reconocen la importancia de las PyMEs por su contribución al crecimiento económico, a la generación de empleo, así como al desarrollo regional y local.

En México, la importancia de las PyMEs radica en la cantidad de establecimientos de este tipo que existen en nuestro país. Por comentar sólo algunas estadísticas: Las PyMEs representan actualmente 99% de la actividad productiva, generan 72% del empleo y aportan 54% del Producto Interno Bruto (PIB) nacional, el 98% de las empresas son PyME (incluidas las microempresas) y ocupan al 64% del personal de la planta productiva del país y, tan solo en lo que va del 2005, las PyMEs han generado más de 150 mil empleos. De acuerdo a todo esto, no es extraño que de enero a septiembre, la Secretaría de Economía, a través de la subsecretaría de las PyMEs, haya canalizado 70 millones en apoyo a emprendedores y que para el 2006 dicho órgano del gobierno solicite un presupuesto cercano a los 250 millones de pesos, a fin de apoyar a un número creciente de nuevos proyectos [González05]. Definitivamente las PyMEs son el motor del desarrollo del país.

Conociendo dichos antecedentes, las PyMEs son un nicho de mercado muy interesante y muy grande. El papel de las PyMEs en la economía de México ha sido un factor que ha generado incrementos en la producción, valor agregado, aportaciones fiscales, fortalecimiento del mercado interno, aumento de exportaciones, etc., por lo que han sido motivo de diseño de políticas encaminadas a promoverlas y apoyarlas para elevar su competitividad. También esta es una excelente razón y causa para que la comunidad académica dedique esfuerzos importantes para la aplicación de ciencia y tecnología orientada a este fundamental sector productivo de la nación a fin de catapultar su desarrollo y mejorar sus condiciones de operación.

1.3 Las PyMEs Mexicanas y el Comercio Electrónico

Podemos afirmar que las PyMEs mexicanas son técnicamente muy capaces para generar un producto o servicio de calidad y hacerlo llegar a los clientes, más sin embargo, en lo que se refiere al comercio electrónico el camino es, a lo sumo, incipiente. En este apartado se presentan algunos razonamientos y lineamientos acerca de la proyección del comercio electrónico en las PyMEs mexicanas.

1.3.1 ¿Por qué Comercio Electrónico para PyMEs Mexicanas?

Tal como se comentó en apartados anteriores, la realidad del comercio electrónico es totalmente diferente de un país a otro, aun dentro de un mismo continente o de una misma región. Asimismo la propia definición de pequeña y mediana empresa difiere aún de un país a otro debido, entre otras cosas, a las políticas económicas y sociales de cada país. Estos factores definitivamente rebasan la concepción de un esquema o modelo genérico global de comercio electrónico para las pequeñas y medianas empresas.

Por otra parte la tecnología para el desarrollo de sitios de Comercio Electrónico está a disposición de las grandes empresas que, por lo general, siguen modelos y esquemas estadounidenses; sin embargo para el caso de las pequeñas y medianas empresas de México, prácticamente no existen reportes en la literatura especializada sobre experiencias en el desarrollo de aplicaciones Web para dichas empresas. Las pequeñas y medianas empresas, como se ha comentado, tienen características muy diferentes a las de las grandes corporaciones, tanto en organización como en recursos, lo cual hace inviable la aplicación de los mismos modelos de comercio electrónico que utilizan las grandes empresas. Bajo estas premisas, definitivamente, la generación de esquemas y modelos adecuados para las PyMEs mexicanas debe ser uno de los primeros pasos para el desarrollo de sitios de comercio electrónico que garanticen calidad, que estén orientados a aplicaciones reales y cuya factibilidad sea total desde el punto de vista tecnológico y económico para el pequeño y mediano empresario.

1.3.2 La integración de la Internet y el comercio electrónico en las Empresas: ¿Dónde están las PyMEs Mexicanas?

La incorporación de la Internet en las empresas, como todo, tiene diferentes etapas, las cuales se recorren de manera ineludible. Según cita [Márquez04] existe un esquema básico de presencia en la red para las empresas y sus etapas son las siguientes:

1. **Presencia en Internet.** En esta etapa las empresas utilizan la Internet de manera análoga a colocar un anuncio en un periódico. El tener una página Web estática podría ser un buen ejemplo.
2. **Comercialización básica.** En esta etapa las empresas ya utilizan la Internet para realizar algunas actividades comerciales. En esta etapa la página Web de la empresa es mucho más elaborada, los contenidos se actualizan con cierta frecuencia y el correo electrónico es el medio estándar de comunicación con los visitantes para consultas, presupuestos, etc. La profundidad en los contenidos es mayor: historia de la empresa, ofertas de empleo, reseñas, etc.
3. **Marketing en la Internet.** La empresa realiza publicidad de su página Web en otras páginas, portales, etc. La página de la empresa en esta etapa es totalmente interactiva y permite realizar compras online mediante algún medio de pago electrónico. Al llegar a esta etapa, la empresa ya realiza B2C.
4. **Comercio entre empresas.** La empresa integra definitivamente las nuevas tecnologías de la información en su negocio. No solo vende a los clientes por medio de la red, sino que también negocia con proveedores, socios o mayoristas por medio de Internet. Esto es, el B2B.

Como podemos ver el último eslabón en el proceso es el B2B por lo que es lógico que sea panacea a lograr por las empresas. Desde luego, antes de esa cuarta etapa, hay que pasar por las anteriores 3 de manera secuencial y/o paralela. En el caso específico de las PyMEs Mexicanas, el grueso de ellas no ha alcanzado siquiera la primera etapa, es decir, no cuentan aún con una página Web. Algunas de ellas alcanzan la etapa dos, y un número mínimo alcanzan la tres, esto quiere decir que pocas PyMEs en México realizan B2C. El llevar a las PyMEs a superar las dos primeras etapas no es difícil, porque los requerimientos son mínimos y de relativo fácil acceso, además que no representan un reto difícil de superar desde el punto de vista tecnológico. La etapa que representa un primer reto es la tercera, en virtud de que requiere mayores esfuerzos en todos aspectos, en el tecnológico, económico y en el humano, además es la tercera etapa la primera en la cual el proceso de negocio se ve afectado de manera importante. En México la cuarta etapa, el B2B, es aun muy incipiente y su desarrollo es lento debido, entre otros factores, a que son pocas las empresas (la mayoría de ellas empresas grandes) las que han cumplido el ciclo natural completo desde la primera hasta la cuarta etapa, condición deseable para realizar comercio electrónico entre empresas de manera exitosa y sustentable. Con lo que respecta a las PyMEs mexicanas, la escasa relación actual de éstas con la Internet y el comercio electrónico requiere de modelos y esquemas que permitan facilitar y acercar estos recursos a fin de que puedan comenzar a obtener sus beneficios e iniciar el camino largo que implica la relación siempre deseable entre el sector productivo y la tecnología.

De lo anterior podemos resumir que las empresas deben visualizar la Internet no como un fin, sino como un medio para lograr mejores resultados. Esta incorporación debe darse de manera gradual para reducir el riesgo de fracaso. En específico, para las PyMEs mexicanas la adopción de este recurso puede darse a partir de la simple publicación de una página Web estática o bien mediante el desarrollo de un esquema de comercio electrónico B2C. Sobre esta última opción se desarrolla el presente trabajo como una alternativa para facilitar el uso de la Internet y el comercio electrónico como parte importante en el desarrollo y crecimiento de la Pequeña y Mediana Empresa.

1.3.3 El Comercio Electrónico como valor agregado para las PyMEs

El comercio electrónico para las PyMEs en México, definitivamente puede constituir una ventaja competitiva. Éste puede ser el factor que las PyMEs requieran para crecer y ganar una ventaja competitiva sostenible respecto a aquellas otras PyMEs que ignoren y/o tengan por poca cosa el potencial del Comercio Electrónico. Además, si el esquema de Comercio Electrónico es el adecuado, no únicamente sus beneficios se reflejarán en una ventaja competitiva con respecto a otras empresas del mismo ramo, sino que además pueden obtener beneficios al interior de la propia PyME, como lo puede ser una mejor administración de todas las funciones básicas de la organización, incremento de la productividad, optimización de recursos, etc. En este momento es

oportuno comentar una observación: Tener presencia en Internet y hacer comercio electrónico a través de la red, a una PyME no le asegura el futuro ni representa una vacuna contra el fracaso, pero el hecho de no tener presencia en Internet y no aprovechar sus beneficios inherentes, lo aleja cada vez más de la competencia [Benitez05].

1.4 Un Centro Comercial Virtual para las PyMEs Mexicanas

Tal y como lo cita el título de esta tesis, el desarrollo de la misma consiste en la realización de un modelo de un Centro Comercial Virtual (CCV) en el que se describen los elementos de la arquitectura así como la interacción entre ellos. Este CCV es una propuesta a fin de ofrecer a las PyMEs de México una opción para realizar comercio electrónico y de esta forma aumentar las posibilidades de crecimiento.

Como se ha comentado con anterioridad, para cada región y aún, para cada país, la realidad actual del comercio electrónico difiere en gran medida, influenciada por factores como el desarrollo tecnológico, infraestructura en comunicaciones, nivel educativo, factores económicos, factores sociales, etc. La incorporación del comercio electrónico para cualquier empresa requiere de un proceso de evolución que, como se comentó anteriormente, puede ir desde incorporar el correo electrónico al proceso de negocio hasta el desarrollo de un esquema de comercio entre empresas. En este marco, el CCV en su concepción gira sobre dos objetivos con respecto a este proceso de evolución: el primero, permitir a las PyMEs mexicanas tener presencia en Internet y segundo, proveer a las mismas de una plataforma para la realización de comercio electrónico B2C.

Es necesario recalcar el foco del desarrollo de este modelo de CCV: las PyMEs Mexicanas. La idea de CCV se fundamenta en la agrupación de un número importante de PyMEs mexicanas a fin de que puedan cooperar entre ellas para usar y mantener una aplicación Web común a todas ellas, a partir de la cual puedan realizar transacciones del tipo B2C, es decir, venta directa de empresa a clientes como una estrategia para ampliar sus mercados y, por supuesto, como una estrategia de crecimiento. La colaboración en el uso y mantenimiento es una de las características importantes de la aplicación como solución, ya que es mediante este esquema que es la reducción de costos en la operación, mantenimiento y marketing de la aplicación web posible. Otra fortaleza a considerar es el hecho de que el CCV propuesto se integra únicamente por PyMEs que tienen una personalidad jurídica, es decir, que se encuentren formalmente establecidas; físicamente o bien que al menos tengan certidumbre fiscal, esto para ofrecer a los visitantes y potenciales clientes seguridad y confianza.

Por otra parte, el análisis y diseño del CCV se realiza tomando en cuenta lineamientos de ingeniería de software con respecto al modelado de aplicaciones Web. Esta es un área de conocimiento en la que se existen varias propuestas. La realización del análisis y diseño del CCV toma en cuenta algunas de estas propuestas, de las que se comentan algunas generalidades en los siguientes apartados.

De acuerdo a todo lo anterior se enuncia como hipótesis de trabajo lo siguiente: Que mediante la utilización de metodologías y técnicas de modelado de aplicaciones Web en UML es posible realizar el análisis y diseño de un Centro Comercial Virtual enfocado a PyMEs como una aplicación Web que integre un número importante de PyMEs mexicanas, a través de la cual estas puedan incorporar a su proceso de negocio un esquema de comercio electrónico de empresa a cliente (B2C) y de esta forma puedan incorporarse a la tendencia mundial a la vez de hacerse de una importante opción de crecimiento y de competitividad .

1.5 Esquemas de Comercio Electrónico existentes orientados a la Pequeña y Mediana Empresa de México

En este apartado se comentan esquemas de comercio electrónico existentes que constituyen opciones para que la pequeña y mediana empresa realice comercio electrónico.

1.5.1 Aplicación Web propia

Las empresas grandes de comercio presencial como Sanborns, Liverpool, entre otros, han ajustado sus estrategias de comercialización tomando en cuenta esta tendencia de crecimiento de la Internet, donde el objetivo es uno: ampliar el mercado de consumo y aumentar así las ganancias. Otras como Esmas.com y Todito.com operan únicamente sobre el comercio electrónico a través de Internet. La pequeña y mediana empresa mexicana puede también recurrir a este recurso. Sin embargo, la implementación, operación y mantenimiento de una aplicación de comercio electrónico (capaz de realizar transacciones de compra-venta de productos, es decir Marketing en el Internet) propia requiere de la conjunción de varios elementos: hospedaje Web, diseñador, desarrollador Web, publicidad, personal capacitado para operar el sitio, entre otros. Esto se traduce en una inversión si la estrategia utilizada para poner en marcha el sitio es la correcta, de no ser así, al contrario, el intento se traduce en un gasto y frustración para el pequeño y mediano empresario. Desafortunadamente, los casos de éxito son escasos, muchos errores en la elección y ejecución de las estrategias de implementación y operación suelen acompañar al intento de los empresarios pequeños y medianos. Además de estas dificultades, la aplicación Web de una PyME

tendría que competir en clara desventaja con las grandes empresas nacionales e internacionales que comercialicen los mismos productos. Sin embargo, esta opción de Aplicación Web particular (estrictamente hablando, sitio Web) es la adecuada cuando una PyME únicamente desea iniciarse en el comercio electrónico para tener presencia en Internet o realizar comercialización básica basada en contenidos, de acuerdo a lo expuesto en el apartado 1.3.2.

1.5.2 Comunidades: Mercadolibre

En los últimos años se ha gestado la idea de crear comunidades y asociaciones de usuarios que realicen operaciones de comercio electrónico bajo un mismo esquema. Ejemplo de este tipo de comunidades es el sitio mercadolibre.com.mx. Mercadolibre, subsidiaria para América latina de Ebay, es un sitio que opera elementalmente bajo un modelo de subastas (auctions), aunque también incluye la compra directa. Este sitio permite a los usuarios, personas o empresas (micro, medianas y raramente grandes), colocar sus productos en la red a fin de que puedan comercializarlos bajo las políticas y reglas que rigen al sitio. Para vender en MercadoLibre.com, los usuarios deben primero registrarse, hecho esto, un usuario registrado puede colocar productos a la venta y/o comprar productos dentro del sitio. El registro puede ser como un usuario individual o bien en la forma de e-shop, el caso de la segunda modalidad el vendedor requiere cumplir con una serie de requisitos extras, los cuales deben ser negociados con el webmaster del sitio. Un usuario puede elegir entre vender sus productos a precio fijo (compra inmediata) o en forma de subasta y utilizar descripciones genéricas para promocionar sus artículos. De esta manera los compradores tienen acceso a cientos de miles de productos independientemente de cual sea su ubicación. Los compradores pueden navegar en el sitio a través de una amplia variedad de categorías que agrupan los artículos ofertados como grandes catálogos temáticos. Además de comprar y vender, los usuarios interactúan entre sí a través de preguntas, comentarios y foros. Cada miembro de la comunidad de MercadoLibre.com tiene su propio pseudónimo y una reputación que se genera mediante un mecanismo de calificaciones otorgadas por miembros de la comunidad en base a la experiencia que hayan tenido con dicho usuario. Este pseudónimo y reputación son el principal activo para operar dentro de la comunidad ya que indica a los usuarios el grado de confianza que otros usuarios han tenido con cada miembro. Este mecanismo es fácilmente rebasado, ya que si un usuario acarrea una reputación negativa, basta con que el mismo usuario se registre de nuevo con un nombre de usuario distinto para borrar su negro pasado. Estos son las principales características de este sitio. Otros más, similares a este, operan bajo esquemas semejantes, pero es mercadolibre.com uno de los más conocidos y utilizados hasta el momento por la comunidad internauta de América latina y específicamente de México.

Basado en programación Web dinámica, en Mercadolibre.com.mx persisten muchas características de Ebay, sitio pionero en el ámbito del comercio electrónico C2C. Mercadolibre tiene una estructura modular, en donde se reconocen claramente

módulos especializados y una interfaz claramente definida. Algo importante de Mercadolibre es que es de acceso libre y masivo, por lo que su arquitectura esta diseñada para dar servicio a millones de usuarios. No obstante, aunque este tipo de propuesta es atractiva, para una PyME no representa una opción real para hacer comercio electrónico con características de eficiencia y de calidad por diversas razones. En primera instancia, existe un nivel de incertidumbre inherente a un sitio donde el registro es libre y totalmente abierto lo que, en un país como México donde aun existe gran temor a realizar transacciones vía Internet, constituye un factor de desconfianza, puesto que los compradores no tienen la total seguridad que todos los usuarios sean bien intencionados. En segunda instancia, el hecho que exista un número importante de usuarios se traduce en un mercado potencial amplio, mas sin embargo también aumenta la competencia entre vendedores, saturando en ocasiones la comercialización de un mismo producto lo que reduce la posibilidad de éxito para un vendedor. Por otra parte, el diseño de este sitio se basa en modelos y lineamientos estadounidenses, por lo que prácticamente su estructura corresponde a las necesidades de las empresas de dicho país.

El CCV retoma algunas de las deficiencias de Mercadolibre como áreas de mejora en el modelo que se propone. Primeramente, en lugar de permitir el libre acceso al público en general, el CCV propuesto en esta tesis únicamente se orienta a PyMEs formalmente constituidas de acuerdo al marco legal mexicano, esto con el propósito claro de proveer certidumbre y confianza a los potenciales compradores. Además, el hecho de acotar el CCV a un país en específico provee un respaldo institucional tanto para el comprador como para las PyMEs que lo integren, generando un ámbito de confianza. Por último, una aportación de este proyecto es la de fomentar la industria del software, mediante la presentación de un modelo que guíe la implementación del CCV que pueda servir como referencia de consulta a la comunidad mexicana dedicada al desarrollo de aplicaciones Web, buscando con esto abatir el rezago en cuanto al uso de tecnologías de la información Web en el País.

1.5.3 Directorios

En un intento por emular las secciones amarillas de los directorios telefónicos surgieron los sitios Directorios. Un sitio de este tipo consiste en un sitio Web en el cual se promocionan y publicitan, de manera clasificada y llamativa sitios particulares de empresas los cuales encuentran en el Sitio Directorio la posibilidad de dar a conocer su sitio. Esta opción es el complemento para cuando una PyME cuenta con un sitio Web particular. A diferencia del CCV propuesto en esta tesis, un sitio directorio no proporciona una plataforma común para la realización de transacciones de compra-venta de las PyMEs que agrupo, sino únicamente opera como un aparador donde se publicitan los sitios particulares, desarrollados de manera independiente, de cada PyME. Ejemplo de este tipo de sitios es mexicoweb.com.mx.

1.6 El modelado de aplicaciones Web

Hasta el día de hoy, lo más importante en el desarrollo de aplicaciones Web han sido las herramientas, pero muy poco se ha dicho y escrito sobre el proceso de desarrollo. La fácil creación de páginas HTML y en general de sitios Web, usando herramientas simples, ha hecho que el desarrollo de este tipo de aplicaciones se haga sin un trabajo serio de análisis y diseño. Cualquier sistema de complejidad no trivial, necesita ser analizado y modelado. Las aplicaciones Web, al igual que otras aplicaciones, necesitan métodos y técnicas formales de análisis y diseño.

Todas las metodologías propuestas para las aplicaciones Web desde mediados de los años noventa presentan su propia notación para casi todos sus diagramas. Una recopilación excelente de varios de estos métodos fue presentada en el primer taller internacional sobre tecnologías de software orientadas a Web [Schwabe01]. En ella se describen cada una de estas propuestas en base a un mismo caso de estudio. Jim Conallen, a finales de los noventa, presentó una extensión de UML basada en las técnicas de implementación actuales de ese tiempo. Esta extensión propuesta por Conallen es bastante útil como técnica de diseño para cuando la implementación de la aplicación que está siendo modelada se encuentra próxima a realizarse o se está realizando pero no cuando se trata de modelar una aplicación como parte de la planificación de un proyecto [Conallen02]. Estas metodologías, y otras muchas que se han propuesto, contribuyen con ideas importantes para el diseño de software orientado a Web. Entre estas aportaciones mencionamos a continuación algunas de las más significativas. El Modelo de Diseño Hipermedia Orientado a Objetos (Object-Oriented Hypermedia Design Model, OOHDM), es una metodología que tiene por objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia [Rossi00]. En este trabajo se destaca como una aportación importante el contexto conceptual que se presenta. Otro trabajo importante es el método Hipermedia Orientado a Objetos (Object-oriented Hypermedia, OOH) el cual proporciona un conjunto de nuevas vistas que extienden el UML para modelar la interfase de aplicaciones Web [Cachero00]. La aportación importante de este trabajo es la definición de los diferentes tipos de vínculos que existen en una aplicación Web. Otra propuesta interesante es el lenguaje de modelado Web (Web Modeling Language, WebML). El WebML es una notación visual para el diseño de aplicaciones Web complejas que usan datos dinámicos frecuentemente. El WebML provee especificaciones gráficas formales para un proceso de diseño completo que puede ser asistido por herramientas de diseño visuales [Ceri01]. De este trabajo podemos destacar la representación gráfica y en XML que realiza de conceptos asociados a las aplicaciones Web. La mayoría de las notaciones de estos métodos pueden ser fácilmente trasladados al UML de acuerdo a como se demostró por Nora Koch para el OOHDH en [Koch01]. De acuerdo a lo anterior, en el Modelado del Centro Comercial Virtual, se retoma un segundo trabajo de Koch en donde presenta lo que denomina como Ingeniería Web Basada en UML (UML-based Web Engineering, UWE) como una metodología alternativa de modelado de aplicaciones Web. No obstante, cabe aclarar que en el modelado del CCV que se

presenta en este documento toma en cuenta algunos otros elementos de otros trabajos importantes, como el de Conallen, por citar un ejemplo. Al respecto de UWE y su descripción se comenta en el próximo capítulo.

1.7 Generalidades de las Aplicaciones Web

Antes de comenzar con el análisis y diseño en forma, a continuación se presentan algunos términos y conceptos básicos asociados al dominio de las aplicaciones Web.

1.7.1 Arquitectura de las aplicaciones Web

La arquitectura de una aplicación Web tiene tres componentes principales: un servidor Web, una conexión de red, y uno o más clientes (browsers). El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP.

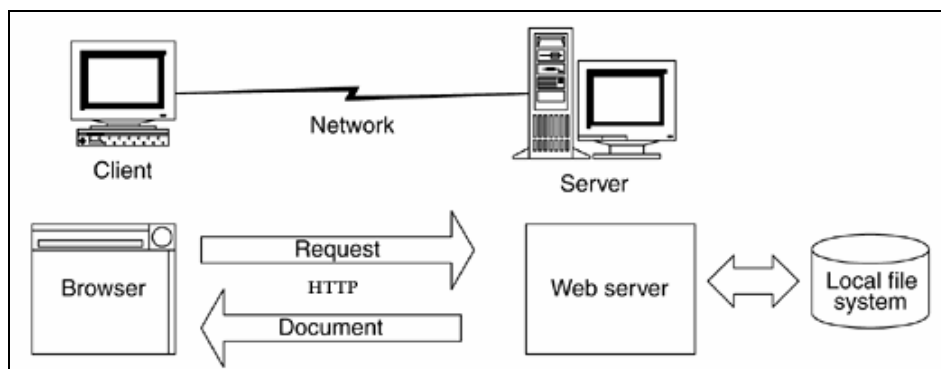


Figura 1.1: Arquitectura típica de las aplicaciones Web

La información mostrada en las páginas está típicamente almacenada en archivos. Sin embargo, muchas veces esta información está almacenada en una base de datos, y las páginas son creadas dinámicamente. Los sitios Web que usan este esquema, son llamados sitios dinámicos [Conallen02].

1.7.2 Aplicación Web vs Sitio Web

Una aplicación Web es un sitio Web donde la navegación a través del sitio, y la entrada de datos por parte de un usuario, afectan el estado de la lógica del negocio. En esencia, una aplicación Web implementa un sitio Web como entrada (front-end) a una aplicación que lleva a cabo operaciones asociadas a procesos integrales en un negocio u organización. Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web [Conallen02]. La lógica de negocio es una regla o un proceso que afecta el estado de negocio de un sistema.

De acuerdo a esta definición podemos decir que cualquier sitio Web de comercio electrónico, como lo es el CCV, es una aplicación Web en virtud de que en el momento en que un cliente realiza un pedido, existe una lógica de negocio definida para satisfacer la solicitud. Un sitio Web, como cualquier motor de búsqueda por ejemplo, no es una aplicación dado que la introducción de datos por parte del usuario no afecta ni requiere lógica de negocio alguna.

1.7.3 Páginas Web

Las páginas Web son el componente principal de una aplicación o de un sitio Web. Los navegadores Web piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web. En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página. Una vez que se entrega una página, la conexión entre el Browser y el servidor Web se rompe (a diferencia de otros esquemas tipo cliente/servidor). Es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el navegador en el servidor, no en el cliente [Guerrero03].

1.7.4 Formularios

La forma más común de capturar la información dada por el usuario en una aplicación Web, es a través de formularios. Un formulario (form) es una colección de campos de entrada: textbox, text area, checkbox, radio button group, button y selection list. Cuando un formulario es llenado, se envía a una página Web (que reside en el servidor Web) usando una operación submit solicitada por el usuario típicamente al hacer click en un botón también definido dentro del formulario [Guerrero03].

1.7.5 Sesiones

Una sesión representa a un uso cohesivo de una aplicación Web. Una sesión implica generalmente a muchas páginas Web ejecutables y muchas interacciones con la lógica de negocio en el servidor de la aplicación. Las sesiones son útiles cuando la realización de una funcionalidad (caso de uso) de la aplicación Web requiere la ejecución de un número determinado de páginas Web, y debido a esto, es útil no perder de vista el progreso de un cliente a través de todo el proceso realizado en la aplicación. Las sesiones en una aplicación Web se pueden implementar de diferentes maneras, sin embargo, la forma más común de implementarlas es por medio de una variable de sesión. Actualmente la mayoría de lenguajes de programación Web permiten la creación y gestión de variables de sesión [Guerrero03].

1.7.6 Objetos COM

La tecnología de objetos COM esta diseñada para la distribución de componentes de software dentro de páginas Web. La tecnología COM es una infraestructura en la cual los desarrolladores pueden construir componentes, en cualquier lenguaje, y distribuir y utilizar dichos componentes en la construcción de sistemas más grandes y complejos. La característica clave de COM es que los objetos desarrollados con esta tecnología pueden ser reutilizados por cualquier aplicación incluidas las aplicaciones Web. En las aplicaciones Web, los objetos COM son comúnmente usados con dos fines: proveer interfases de usuario y para implementar lógica de negocio. Los objetos COM, en general son módulos del programa que se descargan vía Web y se instalan en las máquinas cliente. Hecho esto, ellos pueden ser invocados y utilizados por el navegador Web y otros programas [Conallen02].

Capítulo 2

Introducción al Análisis y Diseño

En este capítulo se revisan algunas guías de referencia, términos y conceptos sobre los cuales se fundamentan los modelos descritos en los siguientes capítulos. Se comentan dos aspectos, la Ingeniería Web basada en UML y la Experiencia de Usuario. En función a estos tópicos se especifican los lineamientos sobre los cuales se realiza el análisis y diseño del CCV.

2.1 La Ingeniería Web basada en UML

El análisis y diseño del CCV toma como guía de referencia, principalmente, a la Ingeniería Web basada en UML (UWE, UML-based Web Engineering). La Ingeniería Web basada en UML, es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML [Koch02].

A diferencia de los sistemas de software convencionales, las aplicaciones Web tienen características especiales, que van desde el entorno en el que operan hasta los requerimientos de usuario. Para satisfacer estos aspectos especiales de las aplicaciones Web, UWE define vistas especiales representadas gráficamente por diagramas en UML, tales como el modelo de navegación y el modelo de presentación. Además, UWE permite que un diseñador Web pueda también hacer uso de otra técnica de modelado UML que agreguen otras vistas de la aplicación, en otras palabras, UWE no limita el número de vistas posibles de una aplicación. Las técnicas de modelado en UML abarcan la construcción de vistas estáticas y dinámicas de los sistemas de software: diagramas del objeto y de clase, diagramas de componentes, diagramas de casos de uso, diagramas de estado y de actividades, de secuencia y diagramas de la colaboración. Además de esta colección de diagramas, el UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se

obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML [Koch02].

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de tareas y la visualización de Escenarios Web para representar los aspectos dinámicos de la aplicación mediante la descripción de situaciones. Este trabajo, en virtud de que se trata de modelar un sistema Web en UML y no un proyecto meramente artístico de páginas Web, se centra en los aspectos significativamente arquitectónicos. Debido a esto se trabaja con la especificación de requerimientos, el modelo conceptual, el modelo navegacional, la visualización de Escenarios Web y la interacción temporal, todos ellos especificados en UWE [Koch02]. Sin embargo se incorporan a estos modelos de UWE artefactos, enfoques y elementos de modelado definidos en otros trabajos, como por ejemplo adiciones en el perfil UML a utilizar, la vista de Experiencia de Usuario, entre otros a los que se hará referencia oportunamente. Estas adiciones se proponen a fin de mejorar la semántica de los modelos de tal manera que el modelo completo de la aplicación sea manejable y fácil de entender.

2.1.1 El Perfil UML

La mayoría de los problemas del diseño Web pueden solucionarse si se utiliza la notación "pura" de UML y los diferentes tipos de diagramas UML, es decir, sin extensiones de cualquier tipo. Sin embargo, los aspectos especiales de un dominio se pueden satisfacer mediante mecanismos de extensión provistos por el mismo UML. La notación que se utiliza en este modelo es un híbrido de los trabajos previos [Koch02], [Conallen02] y [Baumeister02]. El perfil a utilizar incluye los estereotipos definidos para modelar los aspectos asociados a la navegación de la Aplicación Web que es objeto de estudio.

UML proporciona nueve tipos de diagramas y un conjunto de elementos para cada tipo de diagrama. La sintaxis y la semántica de estos elementos de modelado son definidas por el metamodelo y las reglas bien formadas de UML [BRJUML00]. La semántica prevista de los diagramas que sean relevantes para el desarrollo las aplicaciones Web será explicada en las secciones siguientes. En esta sección se presenta una breve introducción al mecanismo de extensión de UML. Los estereotipos representan un mecanismo incorporado de extensión de UML. Junto con los estereotipos, para enriquecer la extensión pueden también incorporarse mecanismos de extensión adicionales como lo son los valores Etiquetados y las acotaciones. De esta manera, UML puede ser considerado como una familia de lenguajes de modelado en lugar de un solo lenguaje si consideramos estas extensiones "ligeras" obtenidas mediante mecanismos de extensión (el término ligero significa que dicho perfil puede ser fácilmente soportado por herramientas y que los formatos de notación UML no se ven afectados de manera significativa) [Koch02].

2.1.1.1 Estereotipos

Un estereotipo de UML es un nuevo tipo de elemento de modelado definido dentro del modelo, basado en un tipo existente de elemento de modelado. Los estereotipos pueden ampliar la semántica pero no proporcionan acceso al metamodelo del lenguaje. Lo que se intenta es que una herramienta genérica de modelado, como lo puede ser un editor de modelos, trate a un elemento estereotipado como un elemento de modelado ordinario para la mayoría de los propósitos, a la vez que lo distingue para ciertas operaciones semánticas. La definición de un estereotipo también incluye un icono de la notación. El uso de este poderoso mecanismo tiene ventajas y riesgos. La ventaja es crear fácilmente lenguajes de modelado para aplicaciones de dominios específicos con elementos de modelado más expresivos y claramente definidos. El riesgo es que el excesivo uso de estereotipos puede hacer a un lenguaje, difícil de manejar y entender.

Algunos estereotipos solamente cambian la notación de un elemento de modelado. Los estereotipos de gran alcance, en cambio, agregan o redefinen restricciones semánticas en el elemento del metamodelo. Los estereotipos de UML se clasifican según su expresividad en estereotipos: decorativos, descriptivos, restrictivos y de redefinición [Koch02]. Los estereotipos decorativos cambian la sintaxis concreta y/o representación visual de un elemento de lenguaje. Los estereotipos descriptivos amplían la sintaxis de un elemento de lenguaje de tal manera que la información adicional puede ser expresada. Los estereotipos restrictivos amplían la sintaxis e imponen restricciones semánticas. Los estereotipos de redefinición, como su nombre lo indica, redefinen un elemento de lenguaje que modifica la semántica original del elemento del metamodelo. Los estereotipos utilizados en el perfil UML propuesto son del tipo de redefinición.

2.1.2 Actividades de modelado a realizar en el análisis y diseño del CCV

Tal como se mencionó anteriormente, las actividades de modelado de UWE sobre las cuales se fundamenta el análisis y diseño de una aplicación Web, como lo es el CCV, giran en torno a la especificación de requerimientos, el modelo conceptual y el modelo navegacional. Además de estos modelos base, este trabajo incorpora otros dos modelos descritos en UWE, estos son el modelo de interacción temporal y el modelo de visualización de Escenarios Web para representar los aspectos dinámicos de la aplicación. A continuación se describen cada una de estas actividades de modelado.

2.1.2.1 Los casos de uso para la especificación de requerimientos

Un modelo de casos de uso se puede utilizar para describir los requisitos funcionales en términos de los casos del uso. Un caso de uso en UML es una unidad coherente de la funcionalidad proporcionada por la aplicación que obra recíprocamente con unos o más actores de la aplicación. Describe una parte del comportamiento de la aplicación sin revelar la estructura interna. Los requisitos para una aplicación Web se pueden especificar con un modelo de casos de uso. Los diagramas de casos de uso en UML se construyen con tres principales elementos de modelado UML: casos de uso, actores y las relaciones entre estos elementos, como lo son las asociaciones entre un actor y un caso de uso y las dependencias "include" y "extends" entre casos de uso [Koch02].

2.1.2.2 Diagramas de Clases para el modelo lógico-conceptual

Un diagrama de clases en UML se utiliza para representar gráficamente un modelo conceptual como visión estática que demuestre una colección de los elementos estáticos del dominio. UWE apunta a construir un modelo conceptual de una aplicación Web, la cual procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, la presentación y de los aspectos de interacción. Estos aspectos se posponen a los pasos navegacionales y de presentación del diseño. [Koch02]

Los principales elementos de modelado usados en el modelo conceptual son: clases y asociaciones. Sin embargo, los diagramas de clases nos proporcionan una gran variedad de características adicionales que pueden ser utilizadas para mejorar la semántica de estos diagramas. Ejemplos de estas características son asociaciones y nombres de rol, la multiplicidad, diferentes formas de asociaciones soportadas por UML como la agregación, la herencia y la composición, las cuales son representadas gráficamente por la notación UML. Si el modelo conceptual consiste de muchas clases, es recomendable que sean agrupadas en paquetes. De acuerdo con los casos de uso que se definen en la especificación de requisitos se construye un modelo conceptual. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web, es decir, los objetos que son relevantes para la realización de una actividad o que son el resultado de una actividad.

2.1.2.3 Diagramas de Clases Estereotipado para el modelo de navegación

Modelar la navegación y la presentación no son preocupaciones exclusivas de las aplicaciones Web, pero con el advenimiento de la Web, la navegación y la presentación ha llegado a ser mucho muy importante en el desarrollo del software que opera en este dominio. Además, el modelar por separado los aspectos conceptuales de la navegación y de la presentación de las aplicaciones Web incrementa la independencia de los elementos del modelado y las posibilidades de reutilización. El modelo de navegación de una aplicación Web comprende la especificación de qué objetos pueden ser visitados mediante la navegación a través de la aplicación Web y las asociaciones entre ellos. Los modelos de la navegación son representados por los diagramas de clases estereotipadas [Koch02].

Los principales elementos de modelado son la clase estereotipada <<navigation class>> y la asociación estereotipada <<link>>. Estos representan a los nodos de página y a los vínculos en la terminología Web.

En este punto, este trabajo retoma el modelo de navegación definido por UWE y adiciona otros elementos de modelado como lo son las clases estereotipadas <<Index>> y <<Form>>, las asociaciones estereotipadas <<Submit>> y <<Redirect>>, entre otros que se comentan en una sección más adelante. Este modelo resalta como el más provechoso, pues con él se pueden representar elementos estáticos, a la vez que se pueden incorporar lineamientos semánticos de referencia para la funcionalidad dinámica.

2.1.2.4 Diagramas de Secuencia para modelar Interacción Temporal

Un diagrama de secuencia de UML demuestra la interacción de elementos dispuesta en orden temporal. Presenta los objetos que participan en la interacción y la secuencia de los mensajes enviados entre ellos. Conallen utiliza los diagramas de secuencia para la descripción de la realización del caso del uso, es decir, cómo se ponen en ejecución los casos de uso [Conallen02]. UWE propone el uso de los diagramas de secuencia iguales que otras metodologías hacen para representar los flujos de la presentación entre, por ejemplo, los aspectos dinámicos de la presentación, tal como las interacciones, las ventanas y los marcos. En este trabajo, en virtud que se da prioridad a los diagramas arquitectónicamente significativos por sobre los que se enfocan al diseño artístico de presentación, se presentan los diagramas de secuencia como descriptores de la realización de casos de uso.

Conjuntamente con los diagramas de estado, los diagramas de secuencia proveen una representación funcional del modelo de navegación, en virtud de que se puede proveer una vista de estados y transiciones con los diagramas de estados y una vista centrada en el tiempo con los diagramas de secuencia.

2.1.2.5 Diagramas de Estados para modelar Escenarios Web

La investigación actual sobre esta área de conocimiento se enfoca, entre otras cosas, en mejorar el modelado y la visualización de los aspectos dinámicos de una aplicación Web. Un punto importante en este sentido, es descubrir qué tipos de diagramas UML son los mas apropiados para modelar este tipo de aspectos. UML provee diagramas de estados. Un diagrama de estados de UML denota una secuencia de los estados que un objeto puede adquirir durante su vida, junto con acciones responsivas, disparando eventos y las condiciones asociadas para indicar transiciones (el concepto de *máquina de estados* fue inventado por David Harel y es originalmente un gráfico de estados, de transiciones y de estados compuestos jerarquizados). UWE utiliza los diagramas de estados para visualizar escenarios de navegación. Estos diagramas permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. [Koch02]

2.1.2.6 Diagrama de componentes para la visualización del esquema de implementación

En este trabajo se utiliza este tipo de diagramas para documentar el esquema de implementación propuesto para la aplicación Web. En este caso, el diagrama de componentes representa la composición general del CCV y de los elementos que deben ser tomados en cuenta en la implementación [Koch02].

Capítulo 3

Especificación de Requerimientos

En este capítulo se presenta la especificación de requerimientos del Centro Comercial Virtual. En base a esta especificación se presenta el análisis y diseño de la arquitectura y funcionalidades del CCV en los capítulos posteriores. Los requerimientos de cada uno de los potenciales usuarios del CCV se describen mediante diagramas de Casos de Uso en UML.

Antes de pasar a la descripción de los requerimientos, en un primer apartado de este capítulo se presenta un diagrama de secuencia para describir el proceso de asociación entre un candidato a Vendedor y el CCV. Esta asociación es un prerequisite asumido por la arquitectura del sistema CCV. En el segundo apartado se presentan algunos diagramas de contexto de alto nivel que proporcionan un primer acercamiento a la arquitectura propuesta para el CCV. En el tercer punto de este capítulo se muestra la arquitectura en una notación UML. Por último, se presentan los requisitos de cada uno de los subsistemas que componen al CCV con base a los lineamientos de la arquitectura propuesta.

3.1 La asociación entre candidatos Vendedores y el CCV

Esta asociación es un prerequisite asumido por la arquitectura del CCV. Es decir, el registro no se considera dentro del dominio de la aplicación. Esto se debe a que el CCV se enfoca a comercios formalmente constituidos y/o establecidos, con personalidad jurídica y una identidad plenamente comprobable. La principal razón de esta condición prevista en este análisis es la de proveer a los potenciales clientes un nivel de confianza superior a cualquier otro esquema de comercialización electrónica en donde el registro del vendedor es libre.

El proceso que a continuación se describe es una propuesta orientada a reducir el registro al CCV de Vendedores fraudulentos o que operan de manera informal. El esquema de validación se realiza totalmente fuera de línea para minimizar el riesgo de la alteración o mal utilización de la información por terceros.

La figura 3.1 muestra un diagrama de secuencia que muestra el proceso de asociación entre un candidato a Vendedor y el CCV. Respecto a este diagrama de secuencia cabe señalar que las operaciones que describe la secuencia se asocian a quien las realiza y con quien las realiza, todos los actores son humanos y esta secuencia no se vincula de ninguna manera al modelado del CCV. El candidato descarga del sitio CCV la forma de registro y las instrucciones a seguir. A continuación, debe realizar un pago de derechos a nombre del CCV en una sucursal Bancaria (se sugiere un monto representativo) a fin de cubrir el costo administrativo de revisión de documentos. Al comprobante de pago y la forma de registro debidamente llenada, el candidato debe anexar: Copia de la Alta en Hacienda (forma R1), copia de la cédula fiscal, copia del comprobante de domicilio, copia de identificación oficial del solicitante, Acta constitutiva (sólo personas morales), copia de los 3 últimos estados de cuenta Bancaria a nombre del solicitante y/o apoderado (Importante: la cuenta que aparece en el estado de cuenta es la que será dada de alta en el CCV para ejecutar los pagos). Todos los documentos deben ser consistentes en los datos (nombres, dirección, etc.). Una vez reunidos los documentos, el candidato debe enviarlos por correo convencional, mensajería o por fax en un solo tanto. El centro de Atención CCV al recibir la solicitud procede a revisar los documentos y, en consecuencia, delibera si el candidato cumple o no con lo requerido. Si la resolución es favorable el centro de atención CCV procederá a tramitar con el Banco el nombre de usuario asociado a la nueva tienda para el acceso del componente de software de pago a través de Internet. El Banco devuelve las especificaciones de uso. Hecho esto el centro de Atención CCV instruye al Administrador de CCV el registro de la tienda en la aplicación. Este registro es llevado a cabo por el usuario Administrador de CCV por medio del subsistema de administración del CCV. Tras el registro de la tienda en el CCV se obtiene la documentación correspondiente a la nueva tienda (que incluye los nombres de usuario y una contraseñas asociados a la tienda), mismos que serán enviados por Correo Convencional o mensajería al vendedor conjuntamente con un contrato de uso y una guía de referencia para que el Vendedor pueda empezar a operar su tienda en el CCV. Si la resolución es desfavorable, se notifica al candidato exponiendo las razones por las cuales su registro no prosperó. El resultado de este procedimiento es el usuario Vendedor perfectamente validado y asociado al CCV al que hace referencia el resto del análisis y diseño.

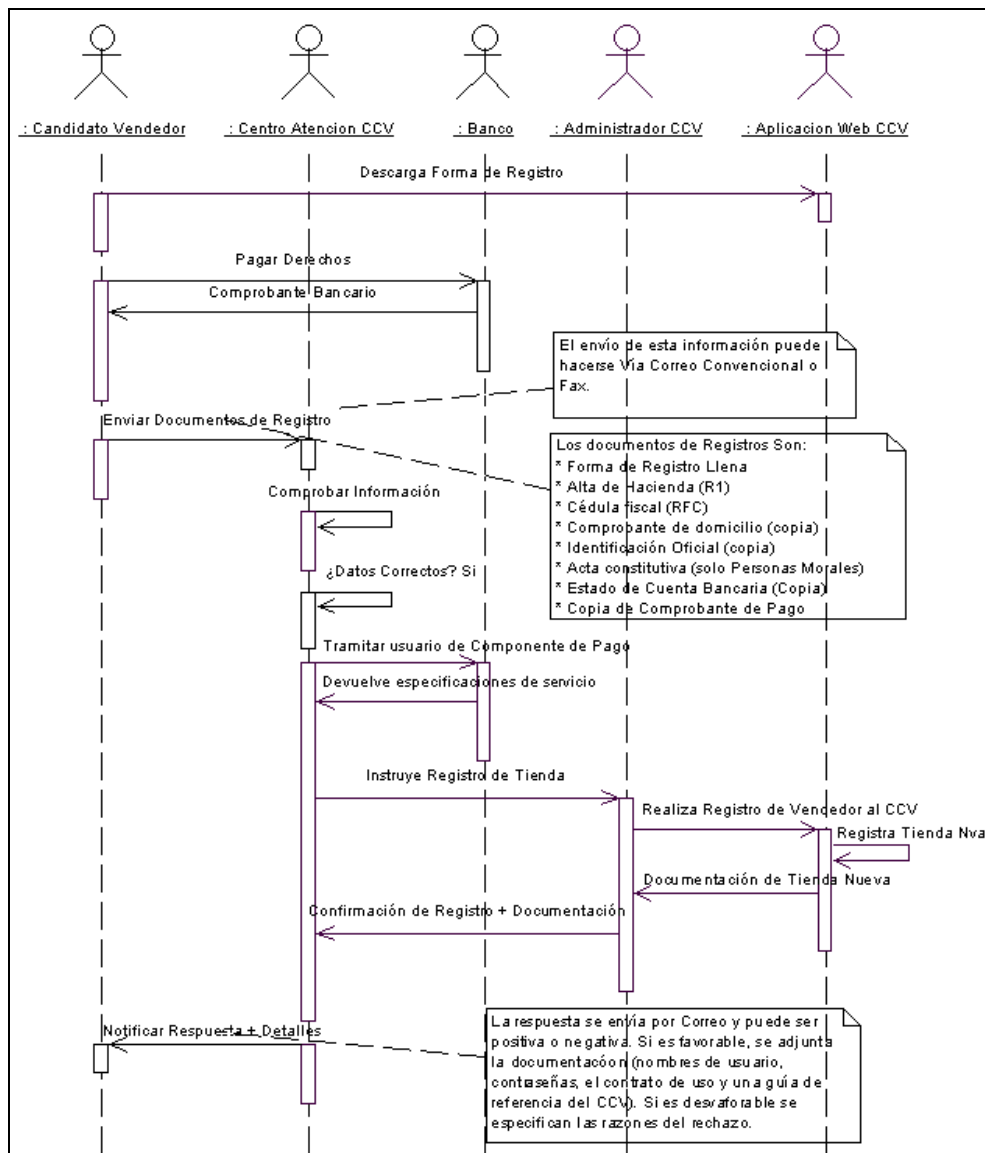


Figura 3.1: Asociación entre un candidato a Vendedor y el CCV

3.2 Un primer acercamiento a la arquitectura propuesta

En este apartado se presentan dos diagramas de contexto de alto nivel (Top-Level Context Diagram - TLCD). En la progresión del desarrollo de cualquier proyecto de software se necesita de una forma de ayudar tanto al desarrollador como al lector, a definir claramente el ámbito del mismo. Un diagrama de contexto responde a este propósito ya que constituye una vista de los elementos que participan en un sistema y el ambiente en el cual se van a desenvolver en un nivel de abstracción general. Específicamente, se presentan diagramas de contexto de alto nivel, a través de los cuales se intenta describir la arquitectura general de una aplicación, las entidades que participan en ella (actores), los límites alrededor de la misma y de las entidades que la forman, las interacciones hacia adentro y hacia fuera del sistema descrito, las fuentes de los datos que se han de procesar, los destinatarios de los datos producidos por el sistema y el ambiente sobre el cual opera la arquitectura descrita. En resumen, un TLCD constituye la vista más general de un Sistema [Clements02].

En la figura 3.2 se muestra un primer acercamiento a la arquitectura propuesta para el CCV. Este bosquejo general de la aplicación sirve como referencia para acotar los alcances de la misma y de esta forma abstraer de manera organizada los requerimientos que debe satisfacer la aplicación. La especificación de requerimientos mediante casos de uso que se presenta en el siguiente apartado y constituye el punto de partida del modelo de CCV desarrollado.

Por otra parte en la figura 3.3 se muestra el ambiente sobre el cual el CCV va a operar. Al ser una aplicación Web, se monta en un servidor Web, al cual todos los usuarios (Clientes, Vendedores y Administradores) acceden a sus respectivos subsistemas mediante un navegador. Las solicitudes http se atienden por el Servidor Web y el acceso a los datos se lleva a cabo mediante un servidor de Bases de Datos operando a la par. Cuando la solicitud http incluye el posteo de información bancaria en la ejecución del pago, el CCV envía la información a través del Socket de Seguridad provisto por la entidad bancaria. La respuesta de la terminal sobre el estado de la transacción a la interfase del CCV retorna por el mismo Socket. La respuesta es procesada por el CCV y el Cliente es notificado del resultado de su transacción. La abstracción del ambiente es importante para tomarlo en cuenta al momento de modelar una aplicación Web.

Como se pudo observar, los diagramas de contexto son útiles para representar preliminarmente a las entidades y las posibles relaciones que las conectan entre si. Los diagramas de contexto se utilizan a menudo para explicar sistemas a la gente, de manera sencilla, acerca de la arquitectura de un sistema. Mas sin embargo, para la descripción de la aplicación, es deseable utilizar una notación formal para describir

estos subsistemas. A continuación se presentan en UML los diagramas de casos de uso como herramienta para describir los requerimientos del Centro Comercial Virtual.

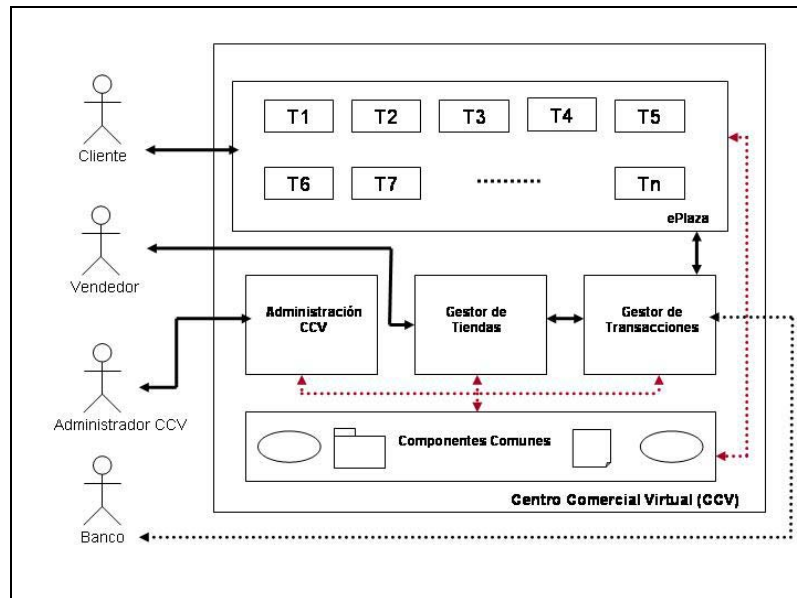


Figura 3.2: Acercamiento a la arquitectura del CCV

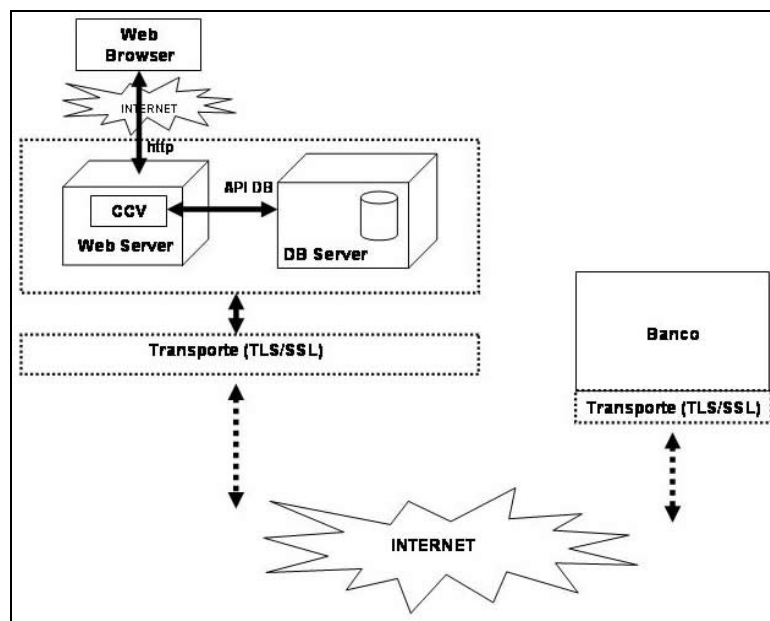


Figura 3.3: Esquema de implementación de la arquitectura propuesta del CCV

3.3 Diagramas de casos de uso

3.3.1 CCV general

En la figura 3.4 se presenta el diagrama de casos de uso general en el que se describen los requisitos genéricos que deben ser satisfechos por el Centro Comercial Virtual. Estos requisitos se representan con casos de uso y los actores que interactúan con ellos. El caso de uso *Visitar CCV* denota la principal funcionalidad a la que un usuario Cliente tiene acceso, que es la de tener la facultad de visitar el CCV y navegar en las tiendas asociadas. El CCV tiene que permitir a cada Vendedor la posibilidad de operar y administrar su tienda. Esta funcionalidad se describe con el caso de uso *Gestionar Tienda*. Así también, el CCV debe proveer la posibilidad de que un usuario con el rol de Administrador del CCV en su conjunto pueda realizar tareas propias de mantenimiento y administración de la aplicación en su conjunto. Esta funcionalidad se denota con el caso de uso *Administrar Centro Comercial Virtual*. Estos tres primeros requerimientos generales se definen como funcionalidades de experiencia de usuario, es decir, corresponden a operaciones y escenarios en los que participa de manera directa un usuario humano de la aplicación. El enfoque del modelo de CCV (y de los diagramas que lo constituyen) se centra fundamentalmente en estas funcionalidades para las cuales se describen requisitos específicos en los siguientes apartados.

No obstante, en este diagrama de casos de uso general se muestra también el caso de uso *Gestionar Transacciones de Pago*. Este caso de uso denota el requerimiento del CCV de realizar las transacciones de pago electrónico de manera segura. Este caso de uso provee una funcionalidad de propósito específico que es usada al interior del CCV al momento de que un Cliente realiza la compra de un producto de una tienda o bien, cuando un Vendedor realiza el pago de comisiones al CCV. Este caso de uso interactúa también con el usuario Entidad Bancaria que representa al prestador de servicio bancario por medio del cual se realiza la operación de pago. Esta funcionalidad, pese a que estrictamente hablando no constituye una funcionalidad de experiencia de usuario, se describe a detalle en el modelo debido a la importancia que esta funcionalidad tiene al interior del CCV al proveer de un mecanismo para la correcta realización de pagos de manera electrónica en el CCV.

El caso de uso *Componentes Comunes* describe un requerimiento no asociado con una funcionalidad, sino más bien, es un requerimiento arquitectónicamente significativo en una aplicación Web. Este caso de uso denota que el CCV debe incluir un repositorio de componentes comunes los cuales se encuentren disponibles en cualquier momento al interior del CCV para llevar a cabo las funcionalidades provistas a los usuarios humanos de la aplicación. Dicho de otra manera, este caso de uso describe la funcionalidad interna transparente a los usuarios humanos que trasciende únicamente cuando los desarrolladores realicen la implementación del CCV y elijan la manera de cómo satisfacer este tipo de funcionalidades. De acuerdo a este punto, en este modelo

enfocado en la experiencia de usuario, la funcionalidad interna pasa a segundo plano por lo que se modela únicamente lo estrictamente pertinente respecto a las funcionalidades de experiencia de usuario.

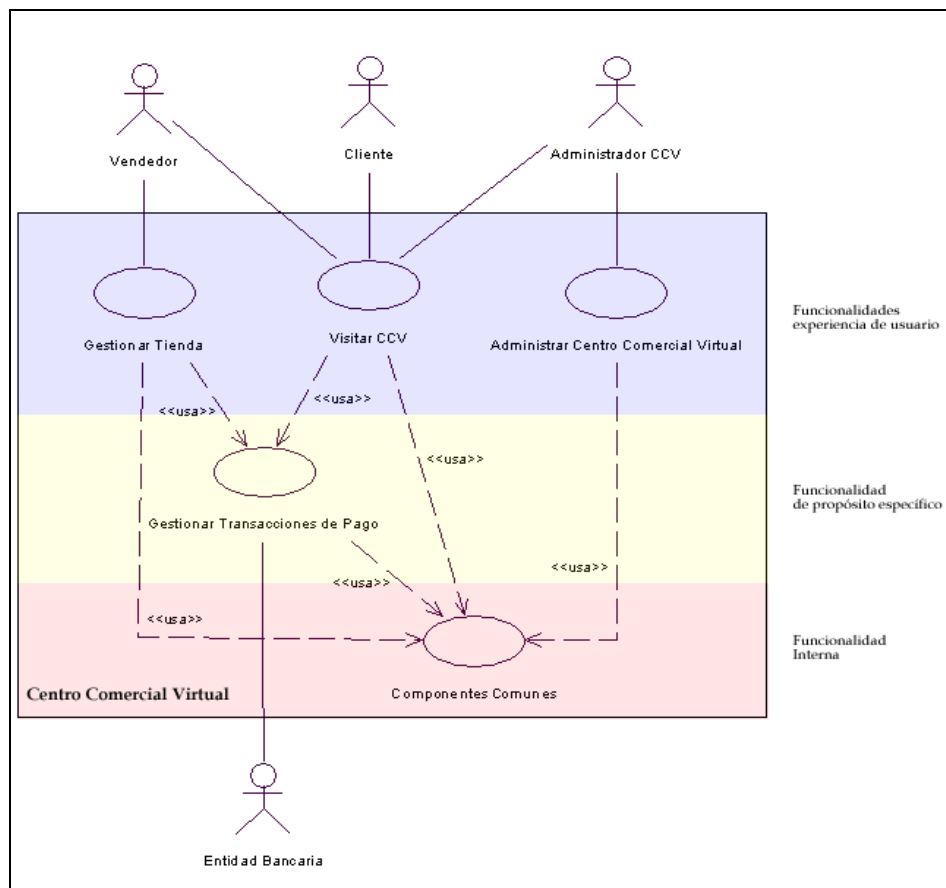


Figura 3.4: Caso de uso general para el CCV

3.3.2 Vista de Subsistemas

De acuerdo con los requerimientos generales descritos en el diagrama de casos de uso general y, con el propósito de acotar y referenciar claramente cada una de las funcionalidades del CCV, en la figura 3.5 se presenta en notación UML al CCV como un conjunto de 5 subsistemas: la ePlaza (eP), un Gestor de Transacciones (GTr), un Gestor de Tienda (GT), un subsistema para la Administración del CCV (AD) y un subsistema repositorio de Componentes Comunes (CC). Para ello, en este diagrama se puede observar el primero de los estereotipos de extensión que se utilizan para modelar el CCV que es el estereotipo `<<Subsystem>>`.

Subsystem: Este estereotipo se aplica a los paquetes de UML tradicionales y su semántica es análoga. A diferencia de un paquete nativo, El estereotipo <<Subsystem>> no denota únicamente la conglomeración de elementos que pueden ser utilizados por cualquier entidad en un diagrama de clases, sino que específicamente denota la agrupación de módulos, unidades, rutinas de un Sistema Global único en piezas claramente definidas y acotadas a implementar una funcionalidad o una familia de funcionalidades asociadas del Sistema en General [Conallen02].

Los subsistemas Gestor de tienda, ePlaza y de Administración CCV se encuentran en el front end del CCV, es decir, los actores humanos tienen interacción directa con estos subsistemas. Los subsistemas Gestor de Transacciones y de Componentes Comunes, son transparentes a los actores humanos y únicamente son utilizados por otros subsistemas. A continuación se describen brevemente y de manera general el propósito de cada uno de ellos.

- **La ePlaza.** En este subsistema reside la principal funcionalidad del CCV: la instauración de tiendas y las operaciones de compra-venta con los Clientes. Es decir, a través de la ePlaza se montan las “n” tiendas que van a hospedarse en la plataforma del CCV para realizar operaciones de compra-venta entre las tiendas y los Clientes. Los usuarios de este subsistema son los potenciales Clientes que navegarán a lo largo de la ePlaza buscando por un producto disponible en alguna de las “n” tiendas. Cuando en la ePlaza se genere la venta de un artículo, la ePlaza iniciará el proceso de ejecución de la transacción realizando una solicitud al Gestor de Transacciones para que sea éste quien gestione la realización del pago electrónico con la entidad Bancaria. Una vez realizado esto el Gestor de Transacciones notificará a la ePlaza sobre el resultado de la transacción. Este esquema permite que el esquema de pagos se asocie única y exclusivamente con un módulo dedicado a la parte más importante de cualquier proceso de negocios: El cobro. La asociación de la ePlaza con el subsistema de Componentes Comunes trasciende en la implementación cuando para proveer las funcionalidades de experiencia de usuario, el subsistema ePlaza requiera utilizar un componente definido en el subsistema de Componentes Comunes.
- **El Gestor de Transacciones.** Este subsistema es un módulo dedicado a la ejecución de la transacción bancaria entre el CCV y la entidad bancaria y al procesamiento de los datos resultantes de dicha tarea. El propósito en la arquitectura de este subsistema es la no intervención de ningún usuario humano en la manipulación malintencionada de dicha información. En este primer análisis el GTr mantiene comunicación con la eP para la atención de solicitudes de pago y para notificar en sentido inverso el resultado de los intentos de ejecución del pago. El GTr también mantiene comunicación con el GT para la atención de solicitudes de pago de comisiones. El GTr, en un ejemplo de cómo un subsistema utiliza el subsistema de Componentes

Comunes, se asocia a dicho subsistema ya que es ahí en donde se encuentra contenido la definición del componente de software a través del cual el GTTr realiza la transacción.

- **El Gestor de Tienda.** El propósito de este subsistema es el de proveer todas las funcionalidades necesarias para que los usuarios de tipo Vendedor puedan administrar su espacio dentro del CCV, esto es, administrar su Tienda. Para cada tienda existirá un espacio de administración de tienda en el gestor de tiendas. Dentro de las tareas a realizar a través de este subsistema se encuentra la configuración de la interfase de la tienda, la gestión de los catálogos, la gestión de pedidos, etc. La asociación del Gestor de Tienda con el subsistema de Componentes Comunes trasciende en la implementación cuando para proveer las funcionalidades de experiencia de usuario, el subsistema Gestor de Tienda requiera utilizar un componente definido en el subsistema de Componentes Comunes. Ejemplo de estos casos es la utilización en el Gestor de Tienda de componentes externos (selección de fecha, imágenes, etc.) necesarios para realizar las funcionalidades provistas al usuario Vendedor.
- **El subsistema Administración CCV.** La razón de ser de este subsistema es la de permitir al usuario Administrador del CCV realizar tareas generales para que la operación de la aplicación en general sea la adecuada. El subsistema es único para el CCV. Entre estas tareas se incluyen la administración del padrón de vendedores, la generación de informes, la gestión de usuarios, entre otras.
- **El subsistema de Componentes Comunes.** Este subsistema está orientado a servir como repositorio de componentes que den soporte a los 4 subsistemas restantes. Si bien, en lo futuro no se le dará especial énfasis, su inclusión dentro de la arquitectura se fundamenta en la idea que a partir de este esquema el sistema soporte el reuso, interoperabilidad y escalabilidad, características deseables en cualquier sistema de software.

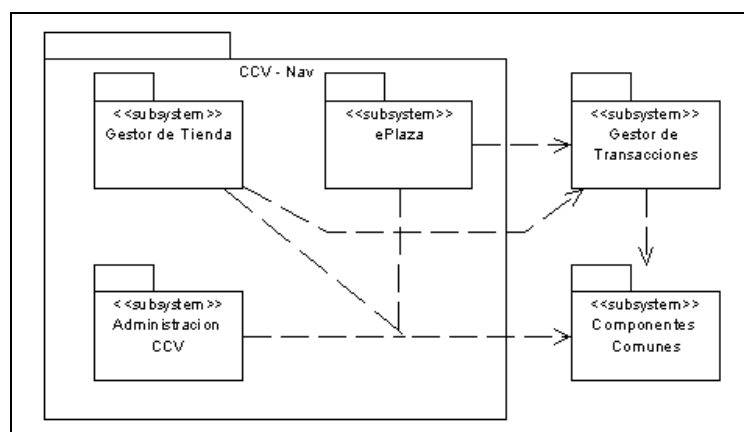


Figura 3.5: Vista de Subsistemas del CCV

3.3.3 Subsistema ePlaza

En la figura 3.6 se muestra el diagrama de casos de uso que realiza la abstracción de las principales funcionalidades del subsistema ePlaza y las relaciones que este subsistema establece con otros subsistemas para llevar a cabo la provisión de las diferentes funcionalidades al usuario. Un Cliente navega a través de la ePlaza buscando por algún producto de su interés. Esta funcionalidad se representa con el caso de uso *Navegar*. *Navegar* tiene dos puntos de extensión uno cuando el Cliente realiza una Búsqueda y otro más cuando el Cliente decide realizar la Compra de un producto. Cuando el punto de extensión para el caso de uso Comprar se cumple, es decir, cuando el usuario Cliente elige comprar un producto, el control pasa al caso de uso *Comprar*. Este caso de uso incluye otros casos de uso, el primero es el de *Autenticar Usuario*. Si el Cliente es nuevo, entonces debe registrarse, es por eso que el caso de uso *Autenticar Usuario* se extiende al caso de uso *Registrar Cliente* cuando el Cliente es nuevo. Sea Cliente nuevo o uno ya existente la autenticación se realiza mediante un nombre de usuario y una contraseña generada cuando el Cliente se registra. Además, se debe dar la posibilidad al usuario que en un futuro pueda modificar la información de envío y facturación en su ficha de registro. El caso de uso *Autenticar Usuario* también es utilizado para la autenticación de los usuarios Vendedor y los usuarios Administrador CCV. El caso de uso *Comprar* también incluye los casos de uso *Obtener Info Tarjeta* y *Realizar Pago*. La tarea del primero es la de obtener los datos de la Tarjeta de Crédito con la cual se intenta ejecutar el pago, esto es el número de tarjeta, las fechas de vencimiento, etc. El caso de uso *Realizar Pago* es el encargado de enviar la información bancaria al subsistema Gestor de Transacciones para ejecutar el pago. Después de enviar la información a través del Socket de seguridad el Gestor de Transacciones envía una respuesta a la ePlaza con el estado de la transacción. Esta respuesta se recibe en el caso de uso *Obtener Respuesta de Pago*, el cual notifica al Cliente el resultado de la transacción y si la transacción se realizó con éxito entonces se procede a *Ejecutar Venta* donde se realizan los procedimientos necesarios para que el pedido pagado pueda ser enviado por el Vendedor asociado a la Tienda correspondiente.

Como se comentó anteriormente, al nivel de experiencia de usuario, la asociación que aparece tanto en el diagrama de casos de uso general y en la vista de subsistemas entre el subsistema ePlaza y el subsistema de componentes comunes no aparece en este diagrama de casos de uso del subsistema ePlaza. El diagrama de casos de uso para la ePlaza describe funcionalidades que los usuarios requieren que cumpla el subsistema de tal manera que el nivel de abstracción es a nivel de usuario. La asociación entre el subsistema ePlaza y el subsistema de Componentes Comunes describe una relación al interior de la aplicación que, aunque es arquitectónicamente significativa, no es relevante para el usuario humano que constituye el elemento central en el modelado de aplicaciones Web desarrollado con la metodología UWE. No obstante esta relación entre el subsistema ePlaza y el subsistema de Componentes Comunes se describe en un caso de uso especial en un apartado subsiguiente.

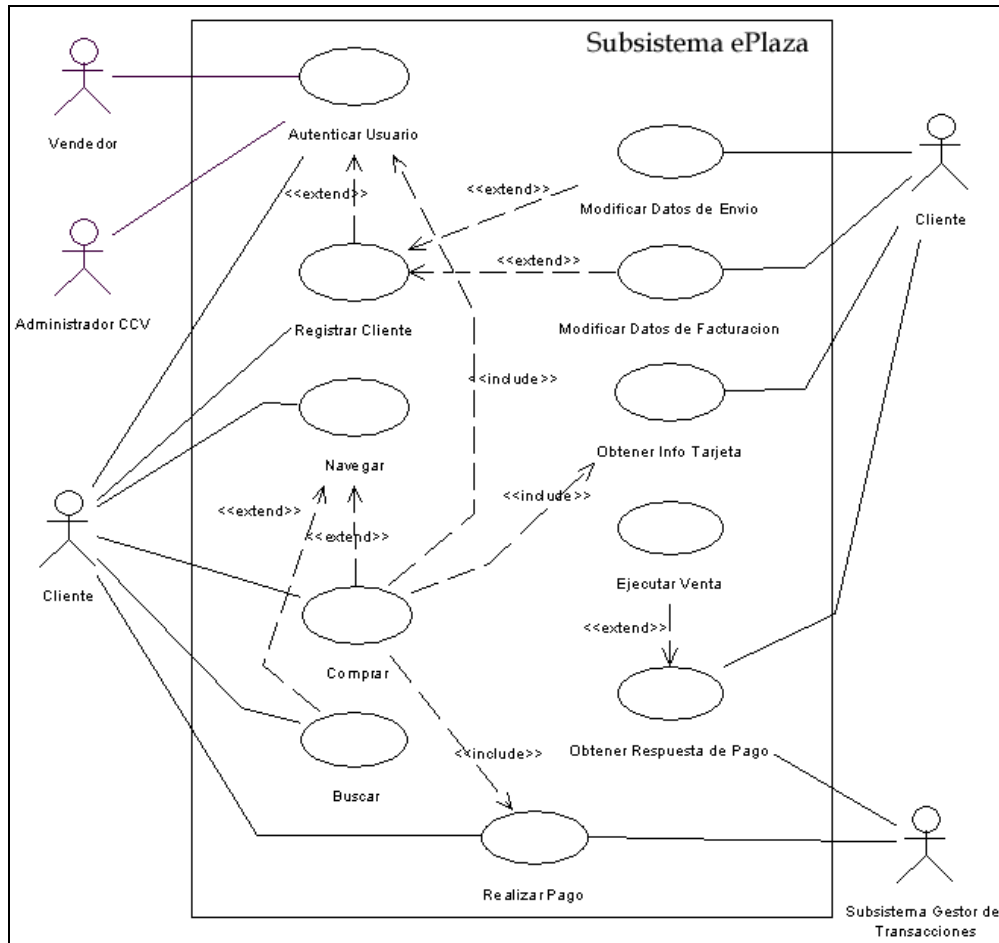


Figura 3.6: Caso de Uso ePlaza

3.3.4 Subsistema Gestor de Transacciones

En la figura 3.7 se muestra el diagrama de casos de uso para el Subsistema Gestor de Transacciones. Como se comentó anteriormente este subsistema provee una funcionalidad de propósito específico que es la de llevar a cabo la ejecución de Transacciones bancarias para los pagos que se realicen en la ePlaza, o bien, el pago de comisiones por parte de un Vendedor a través del Gestor de tienda, haciendo uso para ello de la interfase que se encuentra definida en el subsistema de Componentes Comunes. Estas asociaciones se representan con los actores Subsistema ePlaza y Subsistema CC. A diferencia del diagrama de casos de uso del subsistema ePlaza, en este diagrama de casos de uso del Gestor de transacciones se describe la asociación entre este subsistema y el de componentes comunes, ya que ambos proveen funcionalidad al interior de la aplicación. El caso de uso *Atender Solicitudes de Pago*,

lógicamente opera como una especie de monitor. El punto de extensión del mismo se cumple cuando una solicitud de pago llega, vía la ePlaza o el Gestor de Tienda, entonces el caso de uso se extiende al caso de uso *Enviar Solicitud de Pago*. Este caso de uso es el encargado de construir la cadena con la información bancaria que será “posteada” a través del Socket de seguridad. El caso de uso *Enviar Solicitud de Pago* se asocia al subsistema de Componentes Comunes de donde se obtiene el componente de software que se va a encargar de proveer el medio seguro para la realización de la transacción entre la aplicación y la entidad bancaria. La respuesta de la entidad bancaria hacia el Gestor de Transacciones, que se devuelve de igual manera por el medio seguro, se procesa en el caso de uso *Recibir Respuesta de Pago*. Este caso de uso incluye a los casos de uso *Registrar Respuesta* y *Notificar Respuesta*. El caso de uso *Registrar Respuesta* lleva a cabo las operaciones pertinentes para el almacenamiento de la información resultante de la transacción así como también información de control. Por último, el caso de uso *Notificar Respuesta*, envía al Subsistema ePlaza o Gestor de Tienda, según sea el caso, la información acerca del resultado de la transacción. Aunque este subsistema no es transparente a un usuario humano, se incluye en la vista de Casos de uso debido a la trascendencia que este tiene para la operación del CCV. Esta es una característica importante ya que se debe garantizar en todo momento que la información asociada con transacciones bancarias no pueda ser alterada o utilizada de manera malintencionada por una persona para fines distintos a la ejecución de una compra-venta o a la del pago de una comisión por parte de un usuario autorizado.

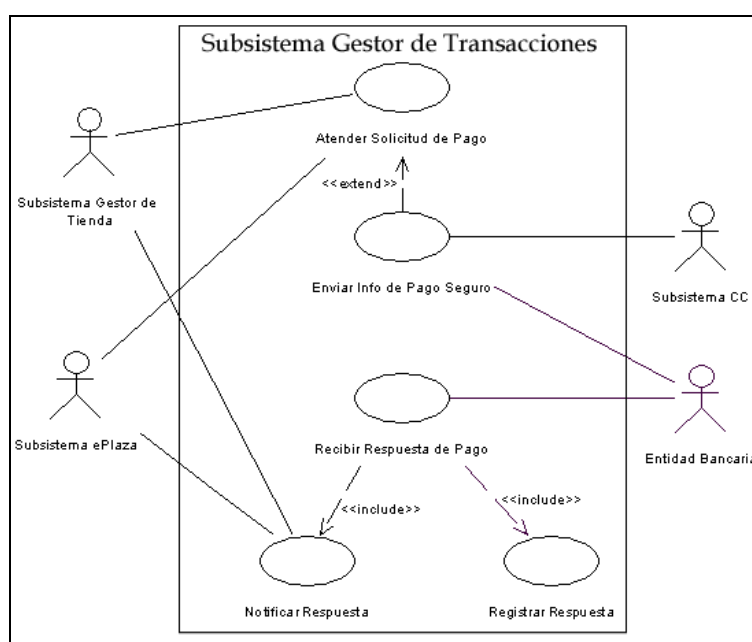


Figura 3.7: Caso de uso Gestor de Transacciones

3.3.5 Subsistema Gestor de Tienda

En la figura 3.8 se presenta el diagrama de casos de uso para el sistema Gestor de Tienda. El propósito del subsistema Gestor de Tienda es el de permitir a los usuarios Vendedores administrar su tienda en la ePlaza. Para cada tienda en la ePlaza existe una entrada en el Gestor de Tienda. Las funcionalidades que un Vendedor va a encontrar en este subsistema son las siguientes y se representan cada una con un caso de uso. Lo primero que debe contemplar el Gestor de tienda es el acceso restringido para Vendedores que tengan una tienda en el CCV. La autenticación de un usuario Vendedor, como ya se indicó anteriormente, se realiza mediante el subsistema ePlaza. Solamente cuando un usuario Vendedor se ha autenticado correctamente tiene acceso a las funcionalidades de este subsistema.

Ya en el subsistema, un Vendedor puede *Administrar Catálogos*, es decir, puede crear, modificar, borrar catálogos de productos, sus descripciones, imágenes, etc., todas aquellas operaciones relacionadas con el catálogo de productos que va a comercializarse dentro de la tienda del Vendedor. El caso de uso *Configurar Tienda* aglomera las operaciones que tienen que ver con las opciones de personalización de la tienda, los colores, los logotipos que han de aparecer, etc. El caso de uso *Administrar Pedidos* representa a las actividades relacionadas con el manejo de pedidos pagados para su seguimiento. A partir de este caso de uso el Vendedor puede verificar el detalle de cada pedido (que producto o productos fueron encargados en un pedido), la existencia de pedidos nuevos, el estatus de cada pedido procesado (pendiente de enviar, enviado, entregado, en proceso de entrega, etc.). El caso de uso *Administrar Servicios de Envío* da la posibilidad al Vendedor de poder dar de alta, eliminar, modificar, etc., servicios de envío de entre los cuales un Cliente va a poder escoger cuando realice una compra. El caso de uso *Enviar Mail* posibilita al Vendedor el envío de correos promocionales, notificaciones de novedades, entre otros a los Clientes registrados, así como también al Administrador CCV por cualquier cuestión relacionada con su tienda. El caso de uso *Consultar Informes de Tienda* permite al Vendedor generar diversos reportes relacionados con la tienda y con la actividad de la misma, esto con la finalidad de que pueda contar con información que le permita tomar decisiones respecto a su tienda y tomar acciones para mejorar sus ventas. Por último se incluye un caso de uso para que un Vendedor pueda realizar el pago de las comisiones generadas por las ventas que ha realizado mediante el CCV. El Vendedor paga con tarjeta y el procesamiento de la información es llevada a cabo por el Gestor de Transacciones de la misma manera que cuando un Cliente paga por un producto.

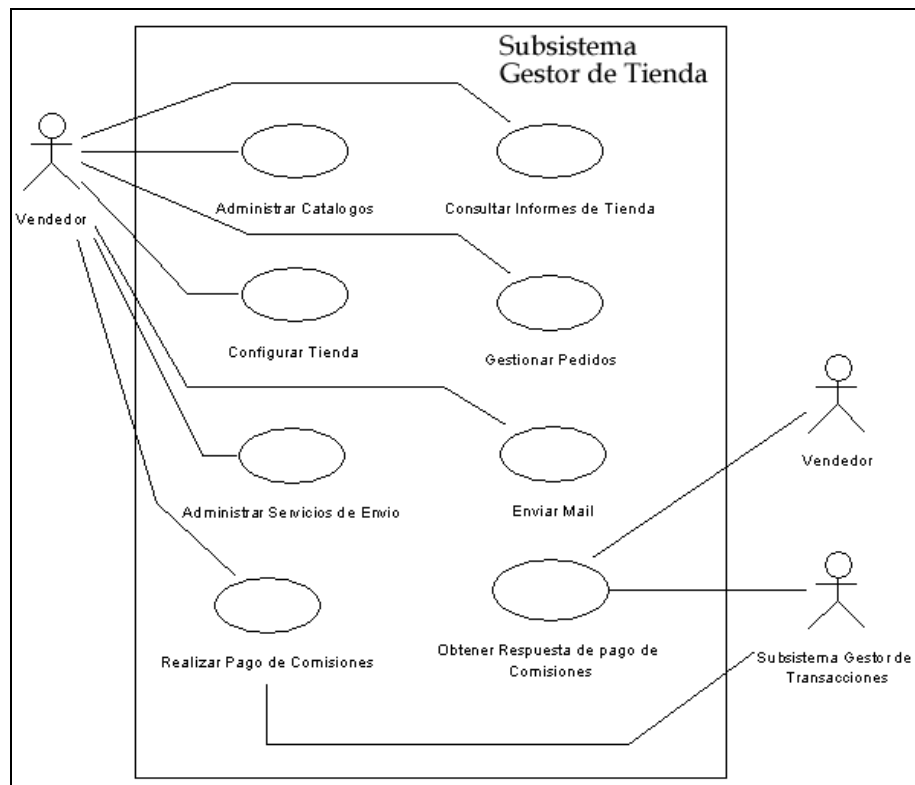


Figura 3.8: Caso de uso Gestor de Tienda

3.3.6 Subsistema de Administración CCV

En la figura 3.9 se muestra el diagrama de casos de uso para el subsistema Administración del CCV. Este subsistema permite al usuario Administrador realizar tareas diversas relacionadas con la administración del CCV en su conjunto. Lo primero que debe de contemplar en la concepción del subsistema de Administración CCV es la autenticación de un usuario Administrador. Esta autenticación, como ya se indicó anteriormente, se realiza mediante el subsistema ePlaza y el caso de uso *Autenticar Usuario*. Solo cuando un usuario Administrador se ha autenticado correctamente tiene acceso a las funcionalidades de este subsistema que son las que se presentan en este diagrama de casos de uso. Ya dentro del subsistema de Administración de CCV, el Administrador CCV podrá realizar diferentes tareas. El caso de uso *Configurar CCV* permite al Administrador cambiar diferentes opciones de configuración para el CCV en general, tal como, directorios, modos de operación, entre otras cosas. El caso de uso *Administrar Padrón de Tiendas* permite al usuario Administrador la gestión del padrón de tiendas, su registro en la ePlaza, la modificación de dicho registro y en un momento determinado, su eliminación. Este caso de uso maneja también la creación y concesión de permisos a los usuarios Vendedores. *Administrar cuentas y comisiones* es un caso de uso en el cual se contempla la gestión de las cuentas de los Vendedores asociados y los movimientos realizados sobre ellas. El caso de uso *Enviar Mail* posibilita al Administrador poder enviar correos a cualquier usuario registrado en el CCV, ya sea Cliente o Vendedor para cualquier propósito relacionado con la operación del CCV. Por último, el caso de uso *Consultar Informes de CCV* proporciona al usuario Administrador la posibilidad de hacerse de información cuantitativa acerca de la operación del CCV en su conjunto, por ejemplo, estadísticas de operaciones realizadas con éxito, reportes fechados, tasas de transferencia de datos, visitas al día, etc.

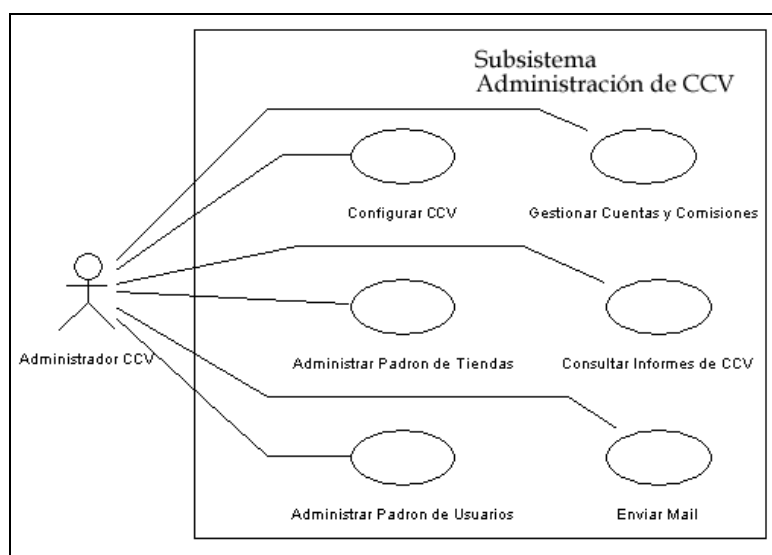


Figura 3.9: Caso de Uso Administración CCV

3.3.7 Subsistema Componentes Comunes

La figura 3.10 se presenta el diagrama de casos de uso correspondiente al subsistema de Componentes Comunes. Este subsistema no es transparente a usuarios humanos, los usuarios de este subsistema son los otros subsistemas del CCV. Este subsistema esta diseñado para proveer componentes de software independientes que puedan ser utilizados por todos los subsistemas del CCV. Entre los casos de uso definidos en el subsistema CC, se encuentran el de *Utilizar Interfase para Transacciones*. Este caso de uso denota la utilización que hace el subsistema Gestor de Transacciones del componente que provee la interfase para el envío/recepción de la información de las transacciones de dinero a través del Socket de seguridad entre el gestor de transacciones y la entidad bancaria. Este caso de uso se asocia exclusivamente con el subsistema Gestor de Transacciones. En la figura 3.10 aparece también el caso de uso *Utilizar Componente para la Generación de Reportes* que representa el uso que hacen los subsistemas Gestor de Tienda y el de Administración CCV del componente para la generación de reportes, ya que en estos subsistemas existen módulos de consulta de informes de tienda y de CCV, respectivamente. Los casos de uso *Utilizar Componente para la selección de Fecha* y *Utilizar Componente para la Exploración de Archivos* son dos casos de uso que describen la utilización por parte de los subsistemas asociados a ellos de componentes de software que realizan una tarea específica como lo son la Obtención de un String de tipo Fecha y la Exploración de Archivos. Estos casos de uso se proponen para ejemplificar el tipo de componentes que se espera se definan en este subsistema. Por último, se muestra un caso de uso *Utilizar otros Componentes Comunes para funcionalidad interna*. El propósito de este último caso de uso es el de dejar ver la posibilidad de incorporar 1 ó mas componentes del mismo tipo que los mostrados explícitamente. La inclusión de este subsistema en la arquitectura se fundamenta en la idea de que a partir de este esquema tipo repositorio, el sistema soporte el reuso, interoperabilidad y escalabilidad, características deseables en cualquier sistema de software.

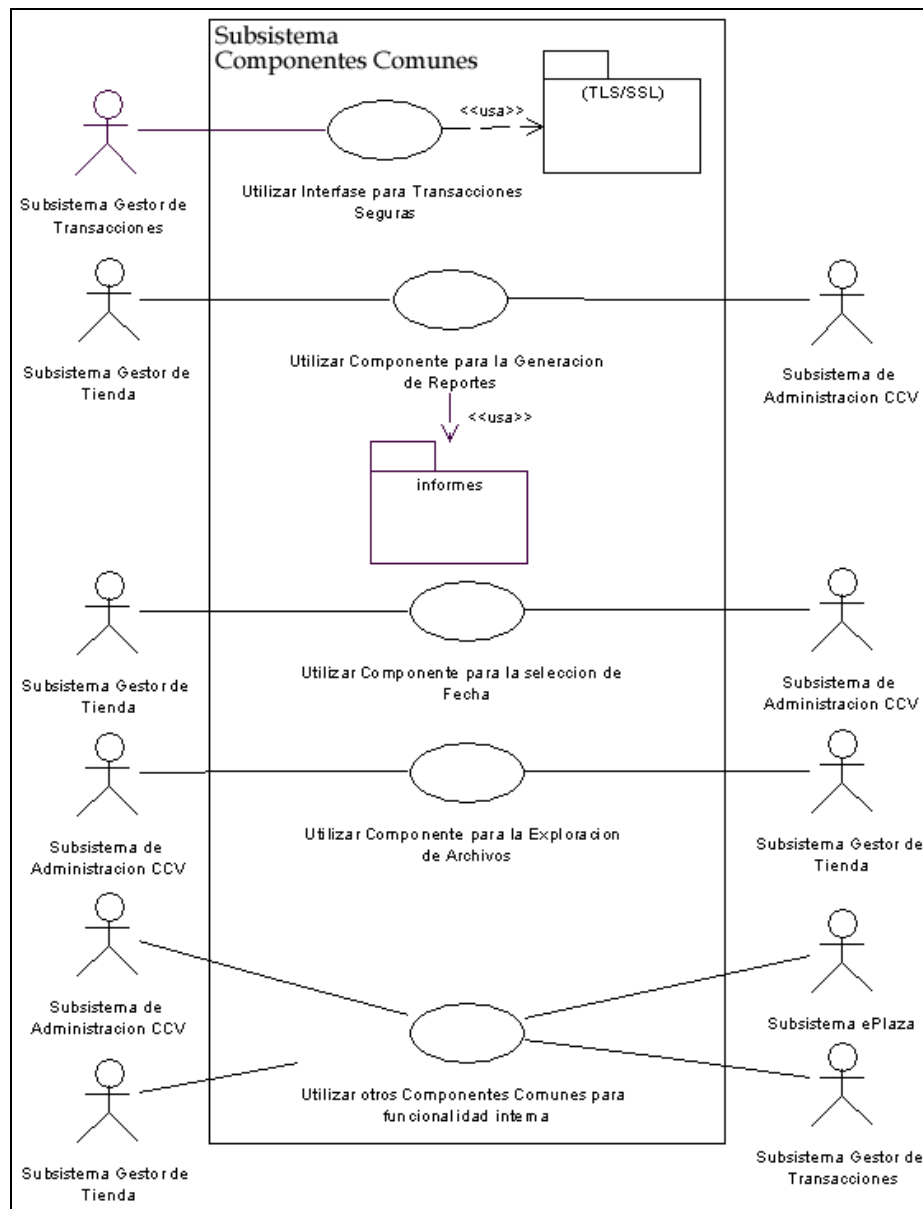


Figura 3.10: Caso de Uso Componentes Comunes

3.3.8 Acerca del Subsistema Componentes Comunes

Sobre este subsistema cabe resaltar algunos puntos importantes al respecto de su operación y su representación en este modelo. Como se comentó anteriormente la Ingeniería Web basada en UML toma como elemento básico en la construcción de los diagramas que describen una aplicación Web, la navegación. La navegación se puede describir como el paso de una entidad navegable hacia otra mediante un vínculo que es activado por un usuario humano. Es así que en la navegación es imprescindible la participación de un usuario humano. Es por eso que el desarrollo de este modelo de CCV se centra en lo que en la ingeniería de software se conoce como experiencia de usuario, es decir, en la participación que el usuario tiene para con la aplicación descartando las operaciones que no lo incluyen. Entre estas operaciones se encuentran todas aquellas que constituyen las funcionalidades internas que se realizan de manera transparente al usuario humano tales como: conectividad con las fuentes de datos, manejo de archivos y directorios, administración de memoria, entre otras muchas funcionalidades de este tipo. Entre este tipo de funcionalidades internas se encuentran los repositorios de componentes. El subsistema de Componentes Comunes al ser un repositorio no debería ser considerado en el análisis y diseño del CCV ya que no constituye un subsistema que interactúa con un usuario humano, quedando de esta manera, fuera del alcance de la experiencia de usuario. Esto es importante porque quiere decir que el subsistema de Componentes Comunes no es navegable y si la navegación es el foco sobre el cual se realiza el modelado de la aplicación entonces este subsistema no debe ser considerado en el modelo. Sin embargo en el modelado del CCV se incluye a este subsistema para describir lineamientos que deben ser tomados en cuenta en el desarrollo de la implementación. Por ello se especifica en el modelo de casos de uso su existencia y lo que debe permitir hacer al interior de la aplicación. Como se aprecia en la figura 3.10 y 3.12 los actores que hacen uso de este subsistema son otros subsistemas, ya que como se puede apreciar en la figura 3.11, el subsistema de Componentes Comunes se encuentra en un nivel de abstracción inferior a los otros cuatro. La razón de no descartarlo de este modelado es el significado arquitectónico que un repositorio de este tipo tiene en una Aplicación Web. En la práctica, cualquier aplicación Web (incluso sitios y páginas Web) incluyen dentro de su arquitectura operacional un repositorio de este tipo como buena práctica en la programación así como también para optimizar su funcionalidad en general.

Es por lo anteriormente expuesto que a los casos de uso que aparecen en la figura 3.10 se les da un seguimiento a través de la colección de diagramas que constituyen el modelado del CCV para ejemplificar la manera en que un componente contenido en el subsistema Componentes Comunes se utiliza en la provisión de una funcionalidad de experiencia de usuario. Por otra parte, en la figura 3.12, se presenta un diagrama de casos de uso que describe el tipo de tareas que debe satisfacer el subsistema de componentes comunes y, de esta forma, definir el tipo de componentes que deben ser incluidos en el subsistema. En lo subsiguiente a los casos de uso de la figura 3.12 no se les da un seguimiento dentro de este modelado.

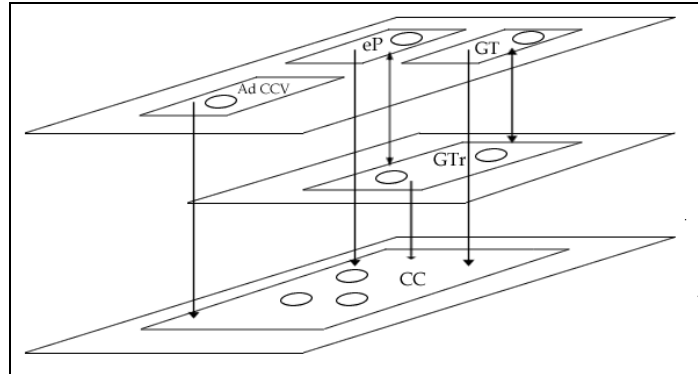


Figura 3.11: Niveles entre Subsistemas

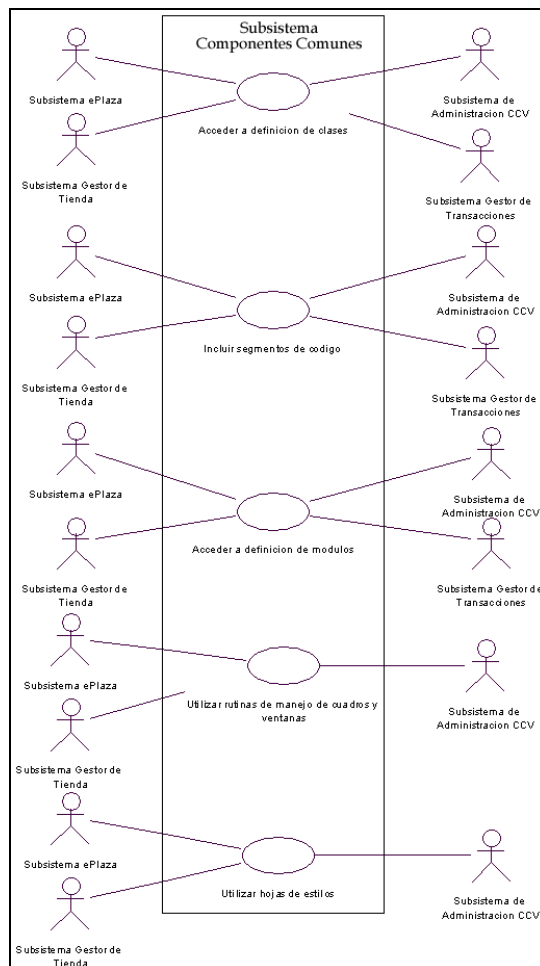


Figura 3.12: Caso de uso Componentes Comunes, especial

Capítulo 4

Modelo Lógico Conceptual

En este capítulo se presenta el modelo lógico conceptual del CCV. Como se comentó anteriormente este modelo describe una primera abstracción de los elementos de la arquitectura de la aplicación Web mediante un diagrama de clases simple de UML. Es en este modelo lógico conceptual donde se exponen asociaciones entre elementos de la arquitectura que se propone, haciendo énfasis en el flujo lógico entre elementos, en su semántica y evitando hacer referencia en la medida de lo posible de cuestiones relacionadas con la navegación y de los aspectos de interacción.

4.1 Modelo Lógico Conceptual: Diagrama de Clases

En la figura 4.2 se muestra un diagrama de clases simple de UML en donde se presentan los principales elementos del CCV y la asociación entre ellos. Esto es, un modelo lógico-conceptual, y como tal, su propósito es el de representar de manera general la estructura del CCV. A menudo un modelo lógico-conceptual también, aparte de establecer un marco de referencia que guié el diseño del modelo navegacional de una aplicación Web, por su semántica, es útil para guiar el diseño de la base de datos de la aplicación. Para el CCV se presenta un único diagrama de clases en el cual se muestran los elementos asociados a cada subsistema. Con la finalidad de poder mapear cada subsistema en este diagrama de clases se utiliza una disposición de colores específica para cada uno de ellos la cual se muestra en la figura 4.1.

Como se comentó anteriormente, en la figura 4.2 se muestra un diagrama de clases que muestra una visión general del CCV. En el se observan clases, multiplicidad y asociaciones de agregación y composición. La semántica del diagrama presenta estas clases como elementos involucrados en las actividades típicas que el usuario realizará con la aplicación Web, y las asociaciones entre ellos como relaciones de cooperación que, en algún momento, ha de llevarse a cabo para realizar una tarea. El subsistema de componentes comunes no se muestra representado en este modelo ya que su funcionalidad no se asocia directamente con los usuarios de la aplicación. En el diagrama se puede observar que el CCV esta compuesto por la ePlaza, el Gestor de

Tienda y el panel de Administración del CCV. El Gestor de Transacciones aparece en asociación con la forma de pago. A la ePlaza se asocian un numero N de tiendas. A cada tienda le corresponde un catálogo de productos definido en categorías que contienen un número determinado de productos. La ePlaza debe también incluir la posibilidad de realizar búsquedas de productos. Un producto puede ser agregado a un carrito de compras, el cual maneja un número determinado de cestas de pedidos. A cada pedido se asocia una forma de pago a través de la cual se envía la información pago hacia el Gestor de Transacciones. A este Gestor se asocia una bitácora y un libro de operaciones para llevar el registro de las transacciones realizadas. El resultado de las transacciones y la generación de las órdenes de pedido se incluyen en la ePlaza. También se incluye el Autenticador de Usuarios; este módulo es el encargado de realizar la autenticación de los diferentes tipos de usuario para permitir o no el acceso a los diferentes subsistemas.

En lo que respecta al subsistema de Administración de CCV se muestran las diferentes asociaciones con los módulos disponibles. De esta forma este subsistema incluye un módulo de configuración del CCV, una forma de envío de correos electrónicos, un módulo para la visualización de informes, un módulo para la gestión de comisiones y cuentas, un módulo para el manejo del padrón de tiendas y un módulo de gestión de usuarios. Como se observa en el diagrama una para cada tienda existe una cuenta, y para cada cuenta varios estados de cuenta, uno por cada período. Así también, cada tienda tiene varios usuarios asociados, todos los cuales son administrados por el padrón de usuarios. En la marca del color correspondiente al subsistema Gestor de Tienda se muestran las diferentes asociaciones con los módulos disponibles. En principio cabe acotar que existe el mismo número de instancias del Gestor de Tiendas que el número de tiendas asociadas al CCV, es decir, para cada tienda hay un gestor de Tienda. Este subsistema incluye un módulo de configuración de la Tienda, un gestor de pedidos, un gestor del catálogo de la tienda, un gestor de servicios de envío, un módulo para la visualización de informes, una forma para el pago de comisiones, y un módulo de envío de correos electrónicos. Por último, en este modelo lógico-conceptual no aparece el subsistema de Componentes Comunes ya que, como se comentó en el apartado 3.3.8, no se encuentra en el nivel de abstracción que denota este diagrama. En el siguiente capítulo se presenta el modelo de navegación para los diferentes subsistemas, tomando como referencia este modelo lógico-conceptual.

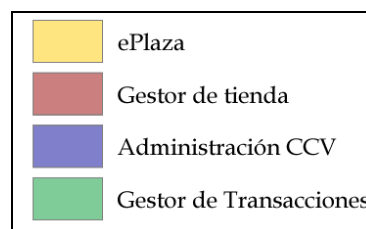


Figura 4.1: Asociaciones de Color del Modelo Lógico Conceptual

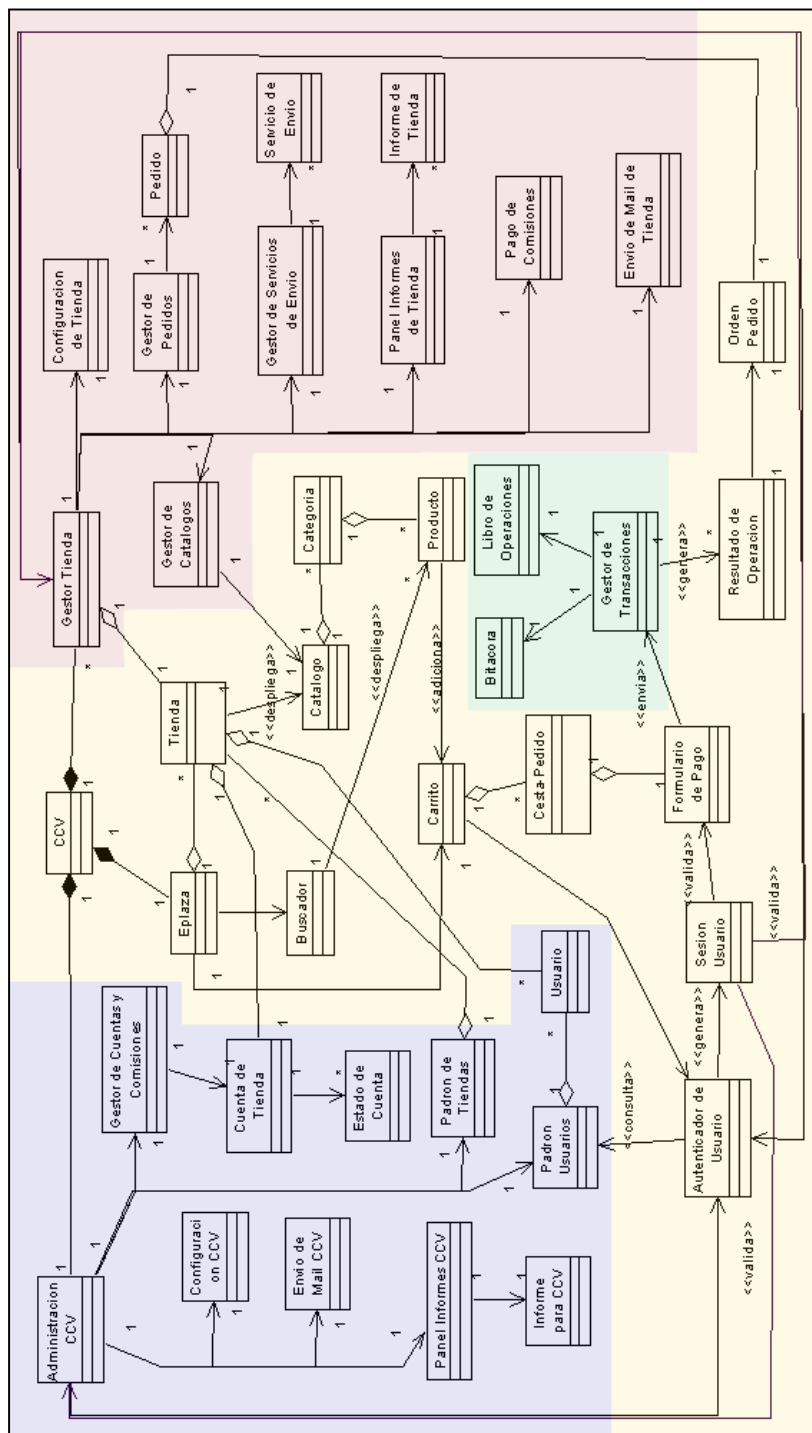


Figura 4.2: Modelo lógico-conceptual del CCV

Capítulo 5

Modelo de Navegación

En este capítulo se presenta la parte más elemental del análisis y diseño del CCV: el diseño del modelo de navegación.

En un primer apartado se describe la semántica de los elementos de modelado que se utilizan en los diagramas de clases estereotipadas. Posteriormente se listan otras consideraciones con relación a convenciones y notaciones que se adoptan en el modelo de navegación. Finalmente, se presenta el diagrama de clases estereotipado para cada subsistema: la ePlaza, el Gestor de Transacciones, el gestor de Tienda, la Administración de CCV y el de Componentes Comunes.

5.1 Estereotipos utilizados en el modelo de navegación

Con base en la extensibilidad de UML utilizando estereotipos, a continuación se listan y describen estereotipos aplicados a clases y a relaciones utilizados en los diagramas. De la colección de estereotipos que se presentan a continuación, la mayoría de ellos están definidos por UWE, más sin embargo se han retomado estereotipos de otros trabajos relacionados con técnicas de modelado de aplicaciones Web en UML, esto con la finalidad de presentar una opción más cercana a las necesidades de los analistas y diseñadores de Aplicaciones Web.

5.1.1 Navigation Class

El estereotipo <<Navigation Class>> es aplicable a una clase simple en un diagrama de clases estándar de UML. Este estereotipo representa una clase conceptual cuyas instancias son visitadas o alcanzadas en una actividad o proceso propiciada por el usuario durante la navegación por la aplicación Web. Una <<navigation class>> en la convención del dominio de la aplicación, es decir la Web, puede ser asociada con una página Web la cual combina contenido estático y dinámico que se presenta al usuario o que sirven como pasarelas de procesamiento de código de un proceso requerido en la navegación por la aplicación Web [Koch02]. En un modelo de navegación es el

elemento más usual y más importante. Su representación gráfica se muestra a continuación.

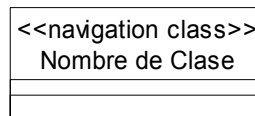


Figura 5.1: Estereotipo Navigation Class

5.1.2 Form

El estereotipo <<Form>> es aplicable a una clase simple en un diagrama de clases estándar de UML. La semántica de este estereotipo tiene que ver con un contenedor de controles (cuadros de texto, botones, cuadros de listas, entre otros) para que un usuario pueda introducir datos a la aplicación, por decirlo de otra manera, este estereotipo representa a los formularios que aparecen en las páginas Web. Este estereotipo aparece siempre como un compuesto o agregado a una <<navigation class>> ya que un formulario aparece siempre embebido en una página Web, y no puede existir de manera independiente [Koch02], [Conallen02] y [Baumeister02]. Su representación gráfica se muestra a continuación.

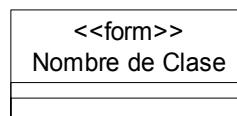


Figura 5.2: Estereotipo Form

5.1.3 Index

El estereotipo <<Index>> es aplicable a una clase simple en un diagrama de clases estándar de UML. La semántica de este estereotipo es un elemento compuesto que contiene un número arbitrario de elementos índices a manera de listado. Un <<index>> en términos de programación Web, comúnmente va a ser generado por una consulta, y el resultado de esta es el que se muestra como un listado de registros. Cada elemento índice es un elemento el cual va a tener atributos estáticos, y al mismo tiempo puede tener uno o más vínculos a una <<navigational class>> [Koch02] y [Baumeister02]. Su representación gráfica se presenta a continuación.

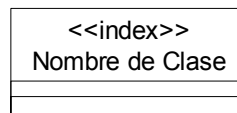


Figura 5.3: Estereotipo Index

5.1.4 Component

El estereotipo <<Component>> es aplicable a una clase simple en un diagrama de clases estándar de UML. Este estereotipo representa a una componente de software ya existente el cual es utilizado en forma de un componente que se incrusta en una o varias <<navigation class>> de la aplicación Web. El estereotipo <<component>> es genérico y no distingue a los componentes a los que representa por la tecnología con las que fueron desarrollados, su plataforma nativa, etc. Su representación gráfica se muestra a continuación.

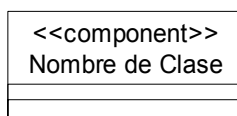


Figura 5.4: Estereotipo Component

5.1.5 File

El estereotipo <<File>> es aplicable a una clase simple en un diagrama de clases estándar de UML. Este estereotipo representa a un archivo cualquiera que la aplicación requiera para su lectura y/o escritura. Su representación gráfica se muestra a continuación.

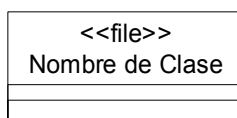


Figura 5.5: Estereotipo File

5.1.6 Object COM

El estereotipo <<Object COM>> es aplicable a una clase simple en un diagrama de clases estándar de UML. Este estereotipo representa a un objeto COM. El COM (Component Object Model) es un estándar binario, independiente del lenguaje, que especifica una interfaz de programación de objetos. La interfaz es la forma en que un programa se comunica con un objeto para hacer operaciones determinadas. En términos más prácticos, un objeto COM puede tomar la forma de un componente de servidor, que es un archivo DLL que, registrado en el servidor Web, proporciona uno o varios objetos para ser utilizados desde cualquier aplicación Web.

El estereotipo <<objeto COM>> se utiliza en el modelo navegacional una sola vez para representar a un Objeto COM que realiza el procesamiento de transacciones de dinero por concepto de compra-venta de productos o por pago de comisiones. Se define este estereotipo de manera particular ya que este proceso es sumamente relevante en la aplicación. Su representación gráfica se muestra a continuación.

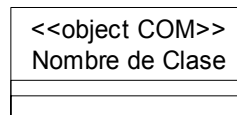


Figura 5.6: Estereotipo Object COM

5.1.7 Link

El estereotipo <<Link>> es aplicable a una asociación simple unidireccional en un diagrama de clases estándar de UML. Una vez definidas las <<navigation class>> se necesita una relación que haga posible la navegación entre ellas. El estereotipo <<Link>> representa a esa relación [Conallen02]. Este estereotipo tiene una clara asociación semántica con los hipervínculos utilizados para ligar páginas Web y por lo tanto su utilización debe acotarse al mismo contexto, condiciones y limitaciones de los hipervínculos en una página web.

5.1.8 Submit

El estereotipo <<Submit>> es aplicable a una asociación simple unidireccional en un diagrama de clases estándar de UML. Este estereotipo representa una relación direccional entre un <<form>> y una <<navigational class>>. Similar a una relación <<Link>>, sin embargo, cuando se define una relación <<submit>>, los atributos de <<form>> son enviados hacia una “navigation class” donde serán recuperados para la operar con ellos [Conallen02].

5.1.9 Redirect

El estereotipo <<Redirect>> es aplicable a una asociación simple unidireccional en un diagrama de clases estándar de UML. Similar a una relación <<Link>>, sin embargo, <<Redirect>> esta asociado con un comando que indica el paso directo de una <<navigation class>> a otra sin que un usuario lo solicite, es decir, el comando de redireccionamiento es disparado al interior de la aplicación Web [Conallen02].

5.2 Semántica de convenciones utilizadas en el modelo de navegación

En esta sección se comentan algunas otras convenciones que se utilizan en el modelo de navegación y que son importantes para la mejor comprensión de los diagramas.

5.2.1 Semántica de atributos ReqF, ReqQS y Req

El modelo de navegación, como se ha comentado en apartados anteriores, es una representación de los diferentes elementos de la aplicación los cuales pueden ser visitados por un usuario en algún momento. En una aplicación Web la representación de dichos elementos puede asociarse claramente con elementos del dominio de la aplicación, es decir, páginas y vínculos. Pues bien, además de esa asociación natural entre el modelo de navegación y el dominio de la aplicación, en el modelo de navegación del CCV se incorporan algunas otras asociaciones mas específicas, como por ejemplo la requisición de variables de entorno Web como atributos de las <<navigation class>>.

Como en todos los lenguajes de programación, en la programación Web se necesita hacer uso de variables a partir de las cuales se pueda dar la creación de contenidos dinámicos y la personalización de los sitios. En este contexto, existen muchas formas de utilizarlas, dependiendo del lenguaje de programación que se utilice. Sin embargo, sin importar el lenguaje, existen formas comunes de hacer viajar esas variables de una página a otra. Dos de estas formas son los formularios y otras las cadenas de consulta URL.

Los formularios se utilizan para obtener información directa del usuario, para ello, en las páginas Web se definen dichos formularios con controles asociados los cuales son enviados de una pagina otra en un solo paquete etiquetado con el nombre del formulario que los contiene. En la página destino debe existir un mecanismo para recuperar los valores para cada variable asociada a un control del formulario. A este mecanismo se le conoce como Request Form. El Request Form, dependiendo del lenguaje en el cual se implemente, tiene una sintaxis diferente, pero la semántica es la misma: Recuperar la colección de variables asociadas a un formulario. Una vez que en una página se recupera un formulario, las variables persisten hasta que la página sea cerrada. En el modelo de navegación se utiliza la estructura ReqF[Nombre_del_Formulario.Nombre_Variable] donde Nombre_del_Formulario es el nombre del formulario sobre el que esta operando ReqF y Nombre_variable, es el nombre de la variable que se esta recuperando. ReqF se asigna a un atributo de una <<navigation class>> denotando con ello que el valor de la operación de recuperación es el valor que adquiere dicho atributo. Cuando Nombre_Variable aparece como [...] significa que lo que se recupera es la colección completa de variables asociadas al

formulario que esta siendo recuperado, y por lo consiguiente, el atributo al cual es asignado el ReqF también será una colección.

Las cadenas de consulta por URL son un mecanismo que se utiliza para transportar variables en forma de parámetros de una pagina a otra. Esto se realiza mediante la incrustación de variables en la cadena URL definida en un hipervínculo HTML. Cuando en una pagina existe un vínculo agregararticulo.asp?IdProducto=123 significa que el vínculo esta direccionado a la página agregararticulo.asp y que se le esta pasando la variable IdProducto con valor 123. Cuando la página agregararticulo.asp sea cargada a partir de este vínculo, la variable IdProducto sera tambien creada con el valor 123, mas sin embargo, la variable debe recuperarse de manera explicita. Esta recuperación se hace comúnmente con el mecanismo Request Query String. En el modelo de navegación se utiliza la estructura ReqQS[Nombre_Variable] donde Nombre_variable, es el nombre de la variable que se esta recuperando de la cadena URL. ReqQS se asigna a un atributo de una <<navigation class>> denotando con ello que el valor de la operación de recuperación es el valor que adquiere dicho atributo. Por último, Req denota un mecanismo abierto de recuperación de información, cuya implementación no tiene asociación alguna con algún método ordinario, sino que se realiza mediante algún procedimiento definido por el desarrollador.

5.2.2 Semántica de atributos Ses

En el dominio de la programación Web es recurrente utilizar variables de sesión administradas por el servidor Web. Estas variables persisten a través de toda una aplicación a partir de cuando son creadas por un tiempo determinado y pueden utilizarse como instrumento de validación de acceso a las páginas. Las variables de sesión se crean en algún momento en la aplicación y su recuperación puede realizarse en cualquier página visitada posterior a su creación. La recuperación de estas variables se realizan con la estructura Ses[Nombre_Variable] y el valor recuperado con esta estructura se asigna a un atributo de una <<navigation class>>.

5.2.3 Semántica de los tipos asociados a otros atributos

En el modelo de navegación, los atributos tipo ReqF, ReqQS y Ses proporcionan al modelo de navegación la posibilidad de representar el paso de información que debe persistir de una clase estereotipada a otra, a fin de proveer la generación dinámica de la información a ser presentada en la <<navigation class>>. Sin embargo, en el modelo se listan otros atributos a los cuales se les asocia un tipo específico. Estos atributos representan información que se despliega en la clase estereotipada que lo contiene. A continuación se presenta un listado de estos otros tipos utilizados en el modelo de navegación.

Tipo	Semántica en el dominio	Clase Estereotipada asociada
URL	Un atributo del tipo URL esta asociado con un direccionamiento de este tipo. Un atributo de este tipo en una <<navigation class>> indica que la clase de navegación que lo contiene esta disponible a partir de un direccionamiento directo del usuario mediante la introducción de la dirección URL en un navegador Web (Microsoft Explorer, Netscape, Mozilla, etc.).	<<navigation class>>
Text	Representa una secuencia de caracteres, con o sin formato, que se despliega en una <<navigation class>>. Un atributo tipo text representa ese texto presentado al usuario que indistintamente puede ser estático o generado de manera dinámica por alguna operación o proceso (consulta a una BdD, script, rutina, etc.) incluido en la clase estereotipada que lo contiene.	<<navigation class>>, <<index>>
Image	Representa a un objeto multimedia del tipo imagen desplegada en la clase estereotipada que lo contiene. Un atributo de este tipo indica que una imagen es incluida en la <<navigational class>> cuando esta es navegada por el usuario.	<<navigation class>>, <<index>>
Num	Representa a un número entero, con o sin formato, que se despliega en una <<navigation class>>. Un atributo tipo num representa una cantidad que se despliega en la <<navigational class>> para información al usuario. Indistintamente un atributo num puede ser estático o generado de manera dinámica por alguna operación o proceso (consulta a una BdD, script, rutina, etc.) incluido en la clase estereotipada que lo contiene.	<<navigation class>>, <<index>>
Currency	Representa a una cifra monetaria, con formato, que se despliega en una <<navigation class>>. Un atributo tipo currency denota información asociada con cantidades (enteras y fraccionarias) de dinero que se despliega en una <<navigational class>> para que sea presentada	<<navigation class>>, <<index>>

	al usuario. Aunque al igual que los anteriores el valor del atributo puede ser estático, comúnmente el valor para un atributo currency se obtiene de manera dinámica producto de una operación (consulta a una BdD, script, rutina, etc.) incluida en la clase estereotipada que lo contiene, ya que es bastante usual realizar cálculos con cantidades de dinero.	
Date	Representa a una secuencia de caracteres con formato de fecha que se despliega en una <<navigation class>>. Un atributo tipo date representa una fecha que se despliega en la <<navigation class>> para información al usuario. Indistintamente un atributo date puede ser estático o generado de manera dinámica producto de una operación o proceso (consulta a una BdD, script, rutina, etc.) incluido en la clase estereotipada que lo contiene.	<<navigation class>>
Input	Un atributo del tipo input representa un campo de entrada en forma de cuadro de texto, a partir de cual un usuario puede introducir información a la aplicación. El valor del atributo es la cadena que el usuario haya introducido en cuadro de texto. En el modelado se asocia este tipo de atributo con clases estereotipadas <<form>> indicando que está contenido en un formulario.	<<form>>
Inputpass	Un atributo del tipo inputpass representa un campo de entrada en forma de cuadro de texto con máscara de caracteres a partir de cual un usuario puede introducir información enmascarada a la aplicación. La información comúnmente vinculada con este tipo de atributos son contraseñas. El valor del atributo es la cadena que el usuario haya introducido en cuadro de texto enmascarado. En el modelado se asocia este tipo de atributo con clases estereotipadas <<form>> indicando que está contenido en un formulario.	<<form>>
Radio	Un atributo del tipo radio representa un campo de entrada en forma de un grupo de opciones	<<form>>

	de radio a partir del cual un usuario puede introducir información a la aplicación al seleccionar una de las opciones del grupo. El valor del atributo es el valor asociado a la opción seleccionada por el usuario. En el modelado se asocia este tipo de atributo con clases estereotipadas <<form>> indicando que está contenido en un formulario.	
List	Un atributo del tipo list representa un campo de entrada en forma de cuadro de lista a partir del cual un usuario puede introducir información a la aplicación al seleccionar un de los elementos de la lista. El valor del atributo es el valor asociado al elemento de la lista seleccionado por el usuario. En el modelado se asocia este tipo de atributo con clases estereotipadas <<form>> indicando que está contenido en un formulario.	<<form>>
Hidden	Un atributo del tipo hidden representa a un campo oculto cuyo propósito es el que sea utilizado como depósito de algún dato que no se obtiene mediante la introducción del usuario, pero que se requiere que sea incluido en el formulario que lo contiene juntamente con todos los otros campos incluidos en el formulario. El valor del atributo tipo hidden es asignado directamente por código y a menudo el valor es una copia de alguna variable. En el modelado se asocia este tipo de atributo con clases estereotipadas <<form>> indicando que está contenido en un formulario.	<<form>>
Button	Un atributo del tipo button representa a un elemento clickeable por el usuario, el cual tiene asociada una acción. Aunque esta acción puede programarse para diversos fines, la acción predeterminada es la de submit (enviar, remitir). Para efecto del modelo de navegación que se presenta en este trabajo, la acción submit es la que se vincula con un atributo del tipo button. Un atributo de tipo button tiene un valor booleano de 1 cuando se mantiene pulsado por un usuario, y de 0 cuando no esta siendo pulsado. Cuando el valor de un atributo	<<form>>

	button es 1 la acción asociada al botón se dispara y el formulario completo es enviado. En el modelo de navegación la acción submit asociada a los atributos del tipo button se describe de manera explícita mediante la definición de una operación en el <<form>> para denotar de una manera mas el envío de la información que incluye el formulario, pero al final de cuentas es la misma acción submit con una nomenclatura homogénea a todos los formularios que busca hacer mas claro el modelo. Por último, en el modelado, siempre se asocia este tipo de atributo con clases estereotipadas <<form>> indicando que esta contenido en un formulario.	
--	--	--

5.2.4 Semántica de los prefijos utilizados en la nomenclatura de las operaciones

Las operaciones que aparecen en una clase estereotipada, al igual que todos los demás elementos de modelado, tienen una asociación directa con el dominio de la aplicación. En esta sección se presentan algunos prefijos que se utilizan en la nomenclatura de las operaciones de las clases estereotipadas que aparecen en el modelo de navegación. La utilización de prefijos provee una idea general de los procesos que se realizan en los métodos especificados en las clases estereotipadas. La manera en que se utilizan es la siguiente: PrefijoComplemento, por ejemplo, el prefijo Cargar_ se utiliza en el modelo como CargarInterfaceCCV. A continuación se presentan algunos de los más representativos.

Prefijo	Semántica en el dominio
IrA_	Este prefijo denota la inclusión en la clase estereotipada que lo contiene de un elemento descriptor (texto, imagen, etc.) que realice la función de describir y disparar la ejecución del vínculo simple desde la clase estereotipada que lo define hacia alguna otra clase estereotipada mediante una asociación estereotipada <<link>>.
CargarLista_	Una operación con este prefijo denota un proceso (generalmente que involucra una consulta sobre la base de datos) que resulta en la generación de listado de registros el cual se presenta en el modelo de navegación en la forma de la clase estereotipada <<index>>.
Cargar_	Denota un proceso cuyas operaciones incluyen invariablemente una consulta sobre la base de datos de la aplicación y el despliegue de los resultados. Además, estos resultados pueden, dentro del

	mismo proceso, asociarse con atributos de la misma clase estereotipada. CargarDatos_ es un prefijo complementario de Cargar_ y la única diferencia al respecto es que se realiza la consulta sobre un registro completo de la base de datos como puede ser un pedido, un producto, etc. y su despliegue comúnmente se realiza sobre un <<form>> rellenándolo con los resultados de la consulta para que el usuario los visualice y, si es el caso, tenga la posibilidad de modificarlos.
Seleccionar_	Este prefijo se utiliza, principalmente, en las clases estereotipadas <<index>> y describe un vínculo simple individualizado específico a cada elemento del listado <<index>>. La individualización se logra con la parametrización del vínculo simple mediante la inclusión de un identificador del elemento del listado en el hipervínculo.
Inicializar_	Este prefijo denota que la operación a realizar es un proceso para establecer el estado inicial de una clase estereotipada, comúnmente una <<navigation class>>. Esta inicialización puede incluir la inicialización de atributos de la clase estereotipada respectiva.
RedireccionarA	En el modelo se utiliza la operación RedireccionarA no como un prefijo de operación sino como una operación tal cual. Esta operación denota un redireccionamiento automático que en algún momento se realiza por alguna operación o proceso de la clase estereotipada que lo contiene. El redireccionamiento se realiza a alguna de las clases asociadas con el estereotipo <<link>>.
Registrar_	Este prefijo denota que la operación correspondiente realiza la inserción de una tupla nueva a una base de datos. Los datos del registro generalmente se obtienen de los datos enviados desde una clase <<form>>.
Actualizar_	Este prefijo denota que la operación correspondiente realiza la actualización de un registro existente en la base de datos de la aplicación previa modificación del mismo por parte del usuario mediante una clase <<form>>.
Eliminar_	Este prefijo denota que la operación correspondiente a realizar es la de eliminación de un registro existente en la base de datos.
EnviarDatos_	Este prefijo se utiliza en el modelo de navegación de manera particular para las clases estereotipadas <<form>> y la operación correspondiente es un descriptor del posteo (envío) de la información contenida en la clase <<form>> hacia otra clase

	estereotipada (comúnmente con <code><<navigation class>></code>) mediante una asociación <code><<submit>></code> .
Generar_	Este prefijo se utiliza en operaciones que realizan un proceso que, en base a operaciones definidas, generan un resultado el cual se utiliza para un propósito específico en la clase estereotipada que lo contiene.

5.3 Semántica de estereotipos asociada con la Web

5.3.1 Acerca del estereotipo `<<navigation class>>`

Este estereotipo se asocia, en la terminología Web, con una página Web. En la práctica, una página Web se constituye por un archivo hospedado en un servidor, el cual es interpretado por un navegador Web cuando una computadora cliente lo solicita a dicho servidor a través del Internet. Debido a este mecanismo, un archivo de Página Web es un archivo de código, por lo mismo puede tener dos vistas distintas, la vista de contenido real en la que se puede ver el código de la página y la vista que se genera cuando el código que contiene el archivo es interpretado por los navegadores.

Por otra parte, el código de cada página puede generar dos tipos de contenido, el estático y el dinámico. Para cada tipo de contenido existe un gran número de lenguajes para que su generación. El lenguaje universal para el contenido estático es el HTML (Hypertext Meta Language), mientras que para el contenido dinámico los más utilizados son los lenguajes como PHP (Hypertext Pre-Procesor), ASP (Active Server Pages) y JSP (Java Server Pages), entre muchos otros. En la figura 5.7 se muestra una `<<navigation class>>`.



Figura 5.7: Semántica en la implementación de <<navigation class>>

5.3.2 Acerca del estereotipo <<index>>

Este estereotipo es un elemento compuesto que contiene un número arbitrario elementos índices a manera de listado. La siguiente figura intenta clarificar el propósito de clarificar la semántica de este estereotipo.

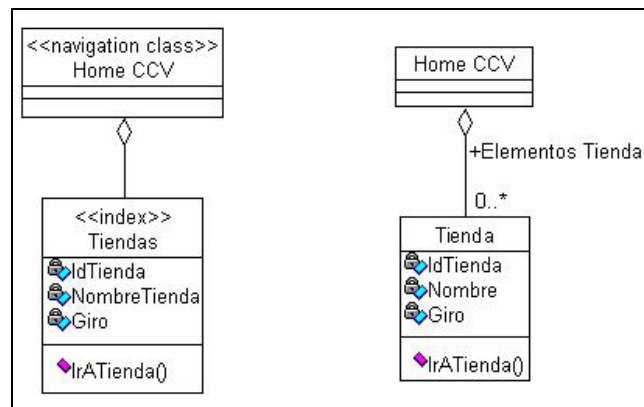


Figura 5.8: Equivalencia de <<index>> en UML puro

En la figura 5.8 se muestra en el lado derecho como se utiliza el estereotipo <<index>> en el modelo de navegación con clases estereotipadas. Sobre el lado izquierdo se

muestra el diagrama equivalente en UML puro. Como se observa, lo que se logra al utilizar el estereotipo <<index>>, además de eliminar la multiplicidad y la especificación de roles, es la aproximación del elemento de modelado en términos más cercanos a la implementación. Un ejemplo de la implementación de las mismas clases mostradas arriba se muestra en la siguiente figura.

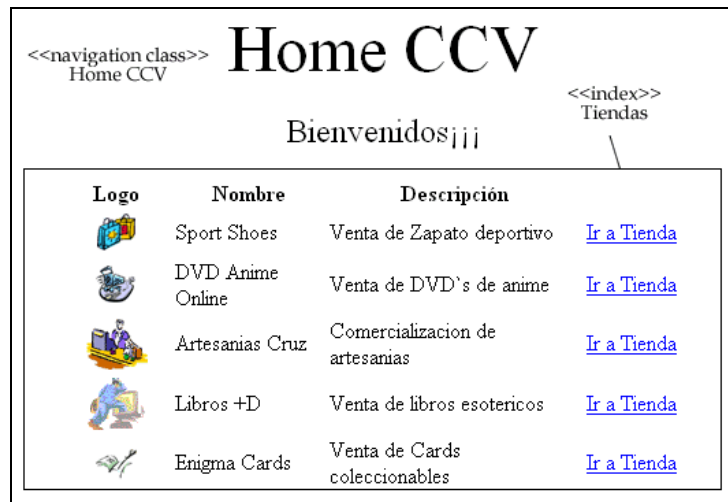


Figura 5.9: Semántica en la implementación de <<Index>>

5.3.3 Acerca del estereotipo <<form>>

Este estereotipo es un elemento de modelado que representa a un contenedor de controles para que un usuario pueda introducir datos a la aplicación Web. En la figura 5.10 se muestra un ejemplo de la implementación del estereotipo <<form>> y de los atributos asociados a éste. En el ejemplo se observa la relación entre el estereotipo <<form>> y el estereotipo <<navigation class>>, se observa que un <<form>> se encuentra embebido dentro de una <<navigation class>> y esta es la única forma en que puede ser utilizado. El <<form>> al final de cuentas representa a un formulario en una página Web. Dentro del formulario se encuentran contenidos diversos controles que en un <<form>> del modelado son representados como atributos de tipo especial como lo son input, inputpass, radio, list, hidden y button. En el ejemplo aparecen algunos controles y el tipo de atributo con el cual son representados en una clase estereotipada con <<form>>.



Figura 5.10: Semántica en la implementación de <<Form>>

5.3.4 Acerca del estereotipo <<Component>>

El estereotipo <<Component>>, en el dominio Web, representa la incrustación de un elemento de software independiente en la aplicación Web. Este componente de software se desarrolla y existe de manera independiente a la aplicación Web, la cual únicamente lo incorpora para realizar alguna tarea específica.

En la figura 5.11 se muestra un ejemplo de la inclusión de un <<component>> en la aplicación. En este ejemplo el <<component>> que se muestra es el Selector Fecha, que se utiliza en el registro de un nuevo producto en el Gestor de Tienda, el cual es un componente ActiveX diseñado para facilitar al usuario la introducción de una fecha en un formato específico.

CCV : Tienda Sport Shoes

Registro Nuevo Producto

<<navigation class>>
Registro Producto

<<form>>
Datos Producto

<<component>>
Selector Fecha

Estado: ☒ Disponible ☐ Agotado

Fecha de Alta:

Fecha Publicación:

Fabricante:

Nombre Producto:

Precio Neto: \$

I.V.A: \$

Tasa Comisión: %

Precio: \$

Existencia:

Imagen:

Examinar

Junio 2006

d	l	m	m	j	v	s
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

Hoy es: Viernes, 9 de Jun

Cancelar

Figura 5.11: Semántica en la implementación de <<component>>

5.4 Modelos de Navegación de los subsistemas

5.4.1 Subsistema ePlaza

En la figura 5.12 se muestra el diagrama de clases estereotipado correspondiente al modelo de navegación del subsistema ePlaza. Como se comentó anteriormente este subsistema es la principal interfase mediante la cual se van a realizar las operaciones de compra de productos en las diferentes tiendas asociadas al CCV. Las principales funcionalidades que este subsistema provee al CCV son: la navegación a través de las tiendas y de sus catálogos, la búsqueda de productos, la autenticación de los usuarios, el registro de Clientes nuevos, la gestión del carrito de compras, y la interfase para la realización del pago de los pedidos.

La ePlaza tiene una <<navigation class>> principal que es *Home CCV*. A partir de esta <<navigation class>> principal, todas las demás que se describen en el subsistema son navegables en algún momento en la realización de una tarea. A *Home CCV* no le precede ninguna <<navigation class>>, puesto que a esta se accede mediante la introducción directa de la dirección Web en la barra de URL del navegador Web de la computadora. El mismo caso aplica para la <<navigation class>> *Login Admon* que es la interfase para la autenticación de usuarios Administradores para acceder al subsistema de Administración de CCV.

La navegación en la ePlaza a través de las tiendas asociadas a este y su catálogo de productos, se provee con las <<navigation class>> *Home CCV*, *Home Tienda*, *Producto* y los <<index>> *Tiendas*, *Catálogo* y *Categoría*. El listado de tiendas asociadas se encuentra agregado en *Home CCV*, a partir de este índice de tiendas, el Cliente tiene la posibilidad de acceder a una tienda en específico. *Home Tienda* es la <<navigation class>> en la cual los datos específicos de cada tienda se cargan conforme a la configuración especificada por el administrador de la tienda. Así también, en *Home Tienda* se carga el catálogo correspondiente a la tienda, el cual se organiza y presenta al Cliente mediante los <<index>> *Catálogo* y *Categorías*. *Catálogo* es un listado de las categorías en la que se clasifican los productos, mientras que *Categoría* esta compuesta por el listado de los productos que contiene la categoría. El <<index>> *Categoría* se asocia con la <<navigation class>> *Producto*. En *Producto* se cargan los detalles para cada producto del catálogo, como lo son el precio, especificaciones, etc. Desde la <<navigation class>> *Producto* se puede navegar hacia la <<navigation class>> *Carrito* mediante la adición del producto al pedido o bien, navegar hacia *Home Tienda* para explorar alguna otra sección del catálogo de la tienda actual navegar hacia *Home CCV* para acceder a otra tienda.

La búsqueda de productos se provee con el <<form>> *Buscar*, la <<navigation class>> *Búsqueda en CCV*, el <<index>> *Productos Encontrados* y la <<navigation class>> *Producto*. El <<form>> *Buscar* es incluido tanto en *Home CCV* como en *Home Tienda*, ya que la búsqueda debe tomar en cuenta ambos ámbitos. *Buscar* es un formulario a partir del cual se obtienen del Cliente los parámetros de búsqueda. La búsqueda en las fuentes de datos del CCV se realiza en la <<navigation class>> *Búsqueda en CCV* donde residen los procedimientos correspondientes y a la cual se agrega el <<index>> *Productos Encontrados* para el despliegue de los resultantes del proceso de escrutinio. Para la presentación de los detalles de los productos encontrados se utiliza la <<navigation class>> *Producto*.

La gestión del carrito de compras y los pedidos se realiza con la <<navigation class>> *Carrito*, el <<index>> *Listado Productos* y el <<form>> *Actualización Línea*. En *Carrito* se definen las operaciones de adición de un producto a un pedido, la modificación de cantidad del producto en el pedido y la eliminación del producto en un pedido. La adición de un producto se da de uno a la vez mediante una solicitud enviada desde la <<navigation class>> *Producto* hacia la <<navigation class>> *Carrito*. En *Carrito* además aparece asociado mediante agregación el <<index>> *Listado Productos*, en el cual se enumeran todos los productos adicionados al momento en el pedido. A cada elemento del listado de productos adicionados en el pedido se asocia también por composición un <<form>> *Actualización de Línea*. El propósito de este formulario es el de servir de interfase para que el Cliente modifique la cantidad requerida de un producto previamente adicionado, o bien marcarlo para que sea eliminado del pedido. Los datos de este <<form>> se envían a la misma <<navigation class>> *Carrito* para que se hagan efectivos los cambios realizados por el Cliente. A partir de *Carrito*, el Cliente puede elegir entre continuar de compras en la Tienda actual y navegar hacia *Home Tienda*, o bien, inicializar el proceso de pago y navegar a *Login Cliente*. En este momento, es importante acotar algunos aspectos del carrito de compras. Primeramente cabe señalar que para cada Cliente se asocia un pedido inicial temporal sobre el cual se realiza la administración del carrito. Así también, otra consideración importante es que cada carrito está estrechamente relacionado a una Tienda, por lo que si un Cliente en su navegación llega a cambiar de Tienda un nuevo carrito (y por lo consiguiente, un nuevo pedido inicial temporal) es inicializado para la gestión del pedido para esa nueva Tienda visitada.

Cuando un Cliente desea iniciar el proceso de pago, pueden darse dos situaciones: que sea la primera vez que accede al CCV a comprar o bien que ya se encuentre registrado. La funcionalidad del registro del Cliente nuevo se provee con la <<navigation class>> *Registro Cliente* y el <<form>> *Datos Cliente*. *Registro Cliente* es navegable a partir de la <<navigation class>> *Login Cliente* (con la que inicia el proceso de pago) y en ella se encuentra definida la operación de registro de Cliente nuevo. A *Registro Cliente* se asocia por composición el <<form>> *Datos Cliente* que es el formulario a través del cual el Cliente introduce su información de acceso, datos generales, datos de envío y de facturación. *Registro Cliente* recibe esta información, la valida y realiza el registro del Cliente. Cuando el registro del nuevo Cliente es realizado, el proceso de pago

continúa redireccionando la navegación hacia la <<navigation class>> *Resumen Pago*. Cabe mencionar que *Registro Cliente* también es navegable a partir de las <<navigation class>> *Resumen Pago* y *Pago de Productos* debido a que en *Registro Cliente* se encuentran definidas las operaciones de actualización de datos de envío y actualización de datos de facturación.

En el caso de que el Cliente ya se encuentre registrado, entonces, para iniciar el proceso de pago, debe introducir sus datos de registro, que son su correo electrónico y su contraseña. Esta funcionalidad se provee mediante el <<form>> *Datos Registro Cliente* asociado a la <<navigation class>> *Login Cliente*. Este par de datos se envía hacia la <<navigation class>> *Autenticación Usuario* en donde se realiza la verificación de los datos del registro del Cliente. Si los datos de registro son correctos entonces el proceso se reanuda en la <<navigation class>> *Resumen Pago*. La funcionalidad de *Resumen Pago* es la de presentar al Cliente un resumen del pedido sobre el cual se va a proceder a realizar la transacción y la información de registro del Cliente; además desde esta <<navigation class>> el Cliente tiene la posibilidad de requerir la modificación de la dirección de envío del pedido, funcionalidad que, como se mencionó anteriormente, se encuentra provista por la <<navigation class>> *Registro Cliente*. El proceso de pago se completa mediante la <<navigation class>> *Pago de Productos* a la cual está asociada por composición el <<form>> *Info Pago Prod*. Desde *Pago de Productos* el Cliente tiene la posibilidad de requerir la modificación de los datos de facturación, funcionalidad que, al igual que la modificación de los datos de envío, se encuentra provista por la <<navigation class>> *Registro Cliente*. La última fase para la requisición del pago del pedido se realiza mediante el <<form>> *Info Pago*. A través de este, el Cliente introduce la información de la tarjeta bancaria sobre la cual se debe realizar el cargo del producto. Esta información es la que se envía hacia la terminal bancaria para la ejecución de la operación mediante la gestión de la transacción por parte del subsistema Gestor de Transacciones descrita en el modelo. De acuerdo al resultado de la operación, éxito o fracaso, la respuesta obtenida del Gestor de Transacciones se presenta en la <<navigation class>> *Confirm Pago Prod* o en la <<navigation class>> *Error Pago Prod*, según sea el caso de éxito o fracaso. Una consideración referente al proceso de pago es el caso especial cuando un Cliente ya registrado olvida sus datos de registro. Para esta situación se incluyen en el modelo la <<navigation class>> *Recordar Pass* y el <<form>> *Dato Mail a Recordar* como instrumentos para enviar vía correo electrónico los datos de registro del Cliente.

Por último, la autenticación de los tres diferentes tipos de usuarios del CCV se realiza mediante las <<navigation class>> *Autenticación Usuario*, *Login Cliente*, *Login Vendedor*, *Login Administrador* y los <<form>> *Datos Registro Cliente*, *Datos Registro Vendedor* y *Datos Registro Administrador*. Las interfases para la autenticación de cada tipo de usuario se proveen mediante la <<navigation class>> correspondiente a cada tipo de usuario con su respectivo <<form>> asociado por composición, como se observa en el modelo. Los <<form>> se utilizan para que por medio de ellos los usuarios introduzcan sus datos de registro. El procedimiento de autenticación para cualquier tipo de usuario se realiza en la <<navigation class>> *Autenticación Usuario*. Esta

<<navigation class>> es de suma importancia para todo el CCV ya que en ella encuentran definidas las operaciones para verificar, por medio de la comprobación de los datos de registro, si un usuario ya sea Cliente, Vendedor o Administrador de CCV se encuentra inscrito en el padrón de usuarios del CCV. *Autenticación Usuario* verifica que los datos de registro que son enviados desde las interfases correspondientes al logeo de Clientes, Vendedores y Administradores, correspondan con los datos que se encuentran resguardados en la base de datos del CCV. Si la verificación es correcta, entonces *Autenticación Usuario* reanuda la navegación hacia la <<navigation class>> que corresponda de acuerdo al tipo de usuario que se autenticó, de lo contrario la navegación se reanuda solicitando de nueva cuenta los datos de registro del usuario.

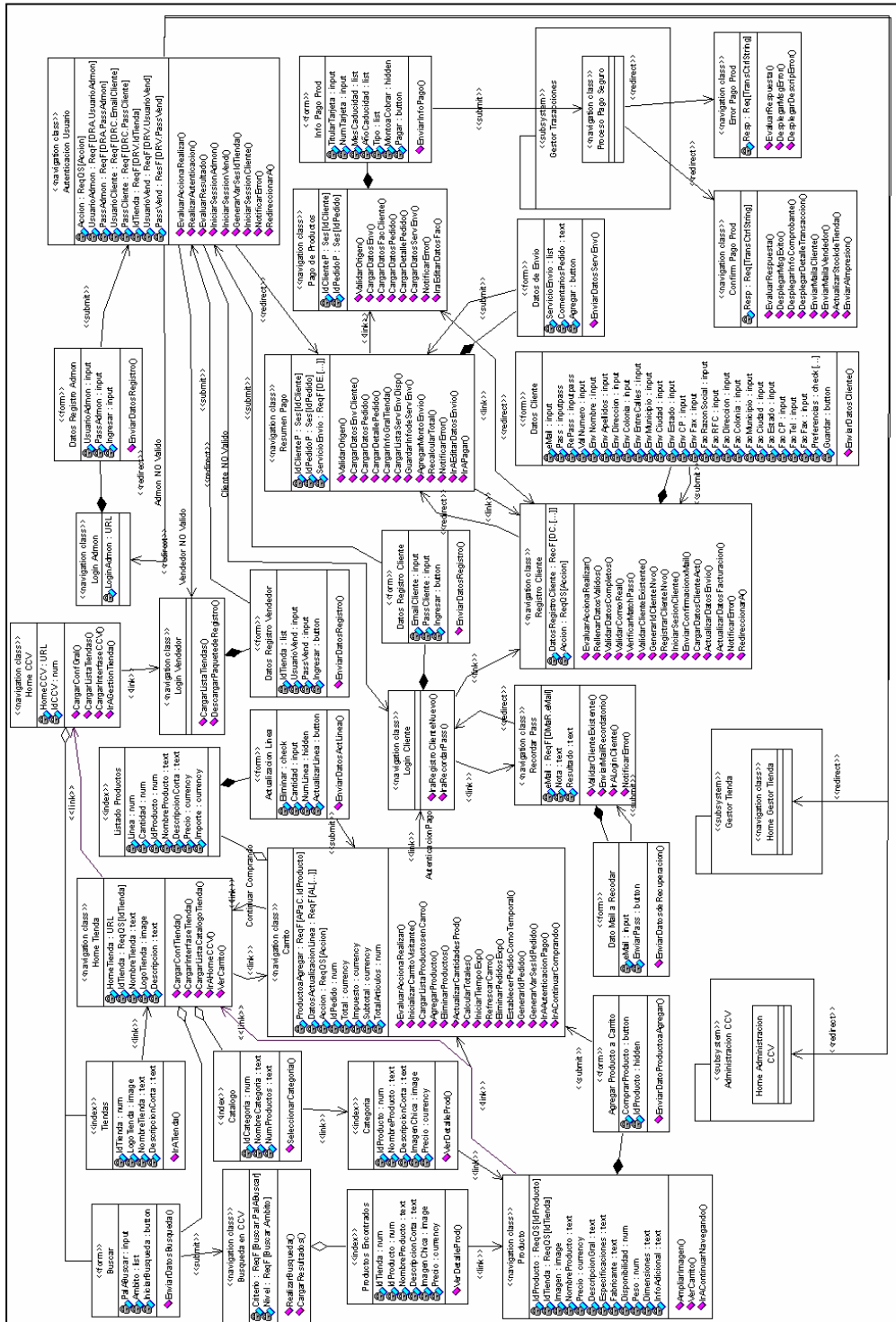


Figura 5.12: Modelo de navegación ePlaza

5.4.2 Subsistema Gestor de Transacciones

El modelo de navegación del subsistema Gestor de Transacciones se muestra en la figura 5.13. El subsistema consta de la <<navigation class>> *Proceso Pago Seguro* en la que se definen los métodos y operaciones necesarias para el procesamiento de la información bancaria que los Clientes introducen al CCV a fin de realizar el pago de los productos seleccionados en una Tienda específica. *Proceso Pago Seguro* realiza este procesamiento en asociación con el <<object COM>> *e24Tranpipe*. El procesamiento de la información bancaria es un activo que se debe proteger mediante mecanismos de seguridad. A lo largo de todo el modelo de navegación que comprende los cinco subsistemas, el único elemento del modelado que se asocia a un término de implementación es el <<object COM>> *e24TranPipe*. El *e24TranPipe* es un componente de software de propósito específico que provee a las aplicaciones comerciales en Internet de conectividad segura con una entidad bancaria para la realización de transacciones de dinero de probada efectividad en sitios de comercio electrónico que operan actualmente en la red. Con la inclusión mediante agregación de este objeto en la <<navigation class>> *Proceso Pago Seguro* se delega a este la responsabilidad de la comunicación segura con la entidad bancaria en el envío de la información de la tarjeta del Cliente. *Proceso Pago Seguro*, entonces, realiza tareas bien definidas: recibir e identificar las solicitudes de pago de productos desde el subsistema ePlaza y las solicitudes de pago de comisiones desde el subsistema Gestor de Tienda, validar el origen de dichas solicitudes de pago, acondicionar el entorno del *e24TranPipe*, instruir la ejecución de la transacción mediante el *e24TranPipe* y procesar la respuesta obtenida, de éxito o fracaso, que resulte del intento de pago. El procesamiento de la respuesta incluye el registro de la operación realizada en una base de datos, la reanudación del flujo de la navegación y la notificación al usuario correspondiente del resultado de la transacción.

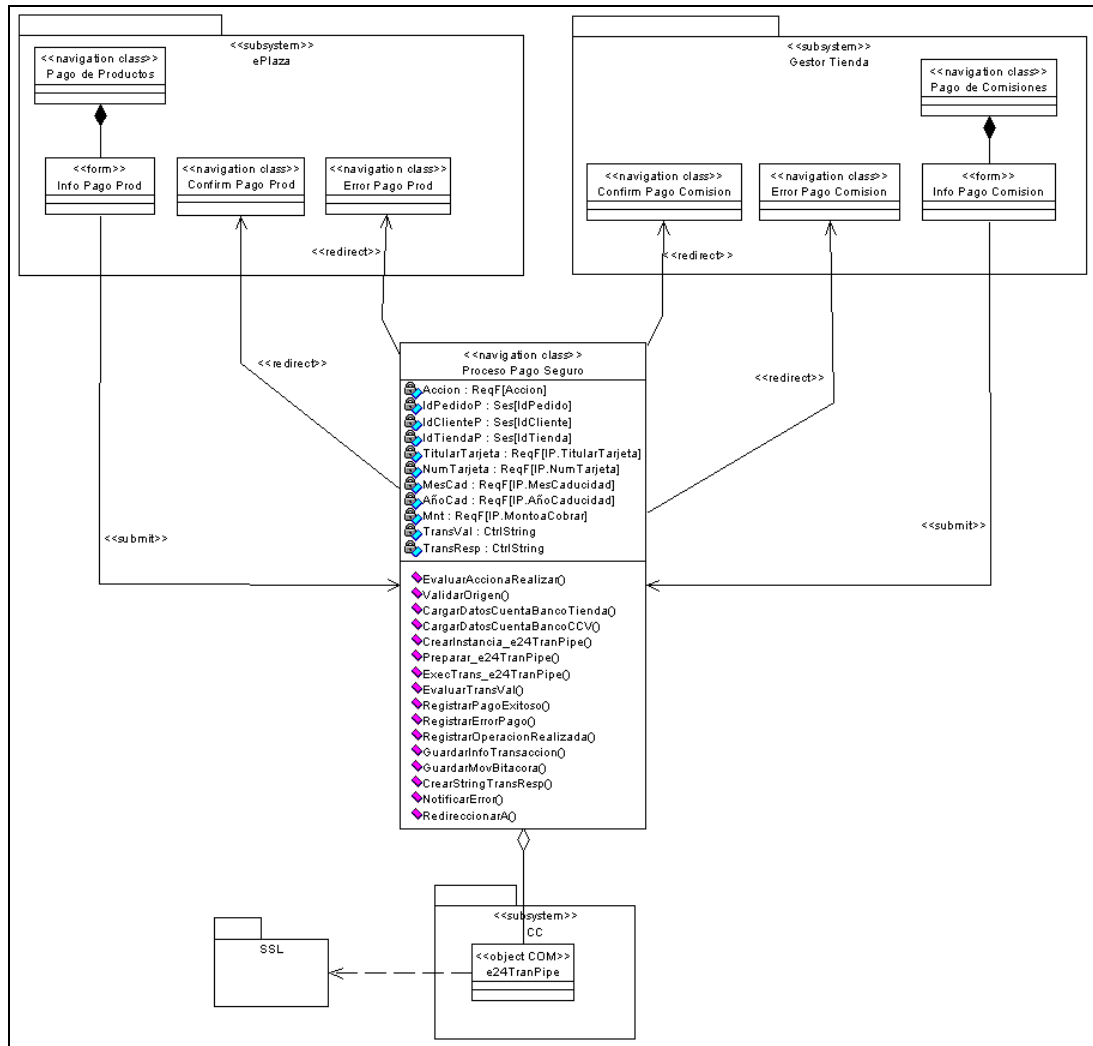


Figura 5.13: Modelo de navegación Gestor de Transacciones

5.4.3 Subsistema Gestor de Tienda

En la figura 5.14 se muestra el modelo de navegación para el subsistema Gestor de Tienda. Una vez que un usuario Vendedor se ha autenticado correctamente en el CCV, accede a la <<navigation class>> principal del Gestor de Tienda que es *Home Tienda*, a partir de la cual se puede navegar hacia los diferentes módulos que este contiene. Las principales funcionalidades de este subsistema son: la modificación de las opciones de configuración de la tienda, el manejo del catálogo de productos, la gestión de pedidos, el manejo de servicios de envíos disponibles en la tienda, el envío de correos electrónicos a direcciones conocidas por el CCV, la visualización de

informes de la tienda y el pago de comisiones. Estas funcionalidades se proveen mediante clases estereotipadas y asociaciones que a continuación se describen.

La modificación de opciones de configuración de la tienda se provee mediante la <<navigation class>> *Modificar Conf Tienda* y el <<form>> *Datos Conf Tienda*. Entre ambos elementos posibilitan al Vendedor a que especifique opciones asociadas con la personalización de la visualización de la Tienda dentro del CCV, tal como lo son: la combinación de colores del CCV, logotipos, textos de contenido y la estructura de la tienda. En *Datos Conf Tienda* el Vendedor selecciona e introduce los valores de la configuración y *Modificar Conf Tienda* retoma esa información para procesarla y registrarla en la fuente de datos del CCV para que los cambios realizados surtan efecto. En la especificación de estos datos, al <<form>> *Datos Conf Tienda* se asocia el <<component>> *Explorador Archivo* que forma parte del subsistema CC.

La gestión del catálogo de productos es la funcionalidad esencial de este subsistema. Esta opción permite al Vendedor organizar la parte más importante de su Tienda: sus productos. En la consecución de este propósito participan las <<navigation class>> *Gestión Catálogos*, *Registro Producto*, *Registro Categoría* y *Conf Stock*, los <<index>> *Catálogo* y *Categoría* y los <<form>> *Datos Producto*, *MoverA*, *CopiarA*, *Datos Categoría* y *Datos Conf Stock*. A partir de *Gestión Catálogo* el Vendedor puede navegar hacia la <<navigation class>> *Registro Producto* para agregar un producto nuevo o bien, navegar hacia la <<navigation class>> *Registro Categoría* para agregar una categoría nueva al catálogo. *Gestión Catálogo* tiene asociado el <<index>> *Catálogo*, el cual es un listado de todas las categorías actuales en la tienda. En cada categoría de este listado se encuentran opciones para solicitar la eliminación de la categoría, la edición de la misma o la visualización de los productos que contiene a través del <<index>> *Categoría* en donde se listan todos los productos que contiene la categoría correspondiente. Para cada producto del listado existen opciones para solicitar la eliminación del producto, editarlo, moverlo de categoría o duplicarlo en dos o más categorías. En la <<navigation class>> *Registro Producto* se encuentran definidas las operaciones para registrar un producto nuevo, modificarlo, eliminarlo, o bien, duplicarlo en varias categorías o moverlo a otra categoría. El Vendedor introduce los datos requeridos para llevar a cabo estas tareas a través del <<form>> *Datos Producto* para el registro de producto nuevo y para la modificación de un producto existente, y a través de los <<form>> *MoverA* y *CopiarA* para las tareas de movimiento de producto y copia de producto, respectivamente. En la <<navigation class>> *Registro Categoría* se encuentran definidas las operaciones para dar de alta a una nueva categoría en el catálogo, modificar el nombre de una existente o bien, eliminarla. En el manejo del catálogo se sugiere la preexistencia de una súper categoría general a la cual se asocien inicialmente todos los productos, y la cual no puede ser eliminada por los Vendedores. Como complemento a la Gestión del catálogo de productos se provee la <<navigation class>> *Conf Stock* y el <<form>> *Datos Conf Stock*, los cuales permiten al Vendedor configurar opciones para el manejo que se le dará al stock de productos al interior de la tienda.

La gestión de los pedidos que los Clientes han realizado a la tienda es otra funcionalidad que satisface el subsistema Gestor de Tienda. En la consecución de este propósito participan las <<navigation class>> *Gestión Pedidos*, *Detalle Pedido* y *Ordenes EFE*, los <<index>> *Pedidos de Tienda* y *Movimientos Pedidos*, y el <<form>> *Movimiento Pedido*. En la <<navigation class>> *Gestión de Pedidos* se presentan la información más relevante para el Vendedor: cuantos pedidos han sido realizados y exitosamente cobrados. Esto significa la obligación del Vendedor de despachar dicho pedido hacia el Cliente. A *Gestión de Pedidos* se asocia el <<index>> *Pedidos de Tienda*, en el cual pueden desplegarse todos los pedidos hechos a la tienda o únicamente los que quedan pendientes de enviar. Para cada uno de los pedidos que aparecen en el listado se asocia la <<navigation class>> *Detalle Pedido*, en donde se desglosa el pedido, es decir, los datos del Cliente, los artículos que integran el pedido, los montos, fecha de realización, etc. Para el manejo de los diferentes estados en los que puede estar un pedido (pendiente, atendido, enviado, entregado, etc.) se provee un mecanismo de registro de movimientos que el Vendedor actualiza mediante el <<form>> *Nuevo Movimiento*. Todos los movimientos realizados se despliegan a través del <<index>> *Movimientos Pedido*. Por último, en la <<navigation class>> *Ordenes EFE* se despliegan de manera organizada para impresión, la orden de embarque, la requisición de la factura correspondiente y la etiqueta con los datos de envío del Cliente.

Un punto importante a tomar en cuenta en la logística de operación del negocio es la de proporcionar a los potenciales Clientes la confianza que los productos adquiridos serán entregado oportunamente. Un servicio de envío se define en el subsistema como la especificación a manera de registro de un convenio establecido entre la Tienda y una oficina de paquetería; dicha especificación describe el servicio de envío en función de los límites en volumen del paquete, dimensiones, peso y el costo del mismo. Es por eso que el Gestor de Tienda incluye dentro de las funcionalidades disponibles al Vendedor, la posibilidad de organizar diferentes servicios de envíos apropiados para el embarque de sus productos. Es así que esta funcionalidad se proporciona mediante las <<navigation class>> *Gestión Servicios Envío*, *Registro Servicios Envío*, el <<index>> *Servicios Envío Disponibles* y el <<form>> *Datos Servicio Envío*. A *Gestión Servicio Envío* se asocia el <<index>> *Servicio Envío Disponibles*, el cual es el listado de los servicios de envío que actualmente se encuentran dados de alta para la tienda, los cuales pueden ser editados o bien, eliminados. En la <<navigation class>> *Registro Servicios Envío* se definen las operaciones para el registro de los servicios de envío nuevos, la edición de los existentes y de eliminación. A esta <<navigation class>> se asocia el <<form>> *Datos Servicios Envío* a través del cual el Vendedor introduce la información del servicio de envío para que sea procesada de acuerdo a la operación, de alta o modificación, que se esté realizando.

Para dar la posibilidad al Vendedor administrador de la Tienda de cuantificar la actividad de la tienda en CCV, el subsistema Gestor de Tienda incluye un gestor de Informes. Esta funcionalidad se provee mediante las <<navigation class>> *Informes para Tienda*, *Informe Tienda*, el <<index>> *Opciones de Informe* y el <<component>> *Generador Reportes*, este último descrito en el subsistema de Componentes Comunes.

En *Informes para tienda* se presentan las familias de informes que se encuentran disponibles. Cada familia contiene un grupo de opciones de visualización, el cual se presenta mediante el <<index>> *Opciones de informe*. Cuando el Vendedor ha seleccionado un informe en específico este se presenta en la <<navigation class>> *Informe Tienda* por medio del <<component>> asociado *Generador Reportes*. Este *Generador Reportes* es un componente de software especializado en generación de informes, el cual mediante la especificación de parámetros que se envían, en este caso desde *Informe Tienda*, se realiza la consulta en las fuentes de información y se genera un resultado y se presenta en un formato claro. De esta manera, a partir de la <<navigation class>> *Informe Tienda* y el generador de reportes se presenta información relevante al Vendedor para la toma de decisiones.

Ahora bien, en el CCV se establece la condición en la que cada Tienda debe cumplir con el pago de los derechos de operación de su espacio en el CCV. El mecanismo propuesto es el de comisiones sobre transacciones de compra venta de productos, en función a un porcentaje establecido. Pues bien, el cálculo de este monto se realiza por adición a la cuenta de la Tienda dicho porcentaje aplicado a cada operación de compra venta realizada por un Cliente en dicha tienda. El mecanismo establece una fecha de corte determinada a cada tienda por la administración del CCV. Antes de que dicha fecha llegue, el Vendedor debe cubrir el monto de las comisiones que generaron sus ventas por medio del CCV. El CCV mediante el subsistema Gestor de Tienda pone al alcance del Vendedor una manera fácil y rápida de realizar este pago, mediante la inclusión de un módulo para la realización del pago con tarjeta bancaria. Esta funcionalidad se implementa mediante la <<navigation class>> *Pago de Comisiones*, *Confirm Pago Comisión*, *Error Pago Comisión* y el <<form>> *Info Pago Comisión*. Desde *Pago de Comisiones* y el <<form>> asociado *Info Pago Comisión* el Vendedor introduce la información de la tarjeta bancaria con la cual se va a sufragar el monto de las comisiones del período. Esta información es la que se envía hacia la terminal bancaria para la ejecución de la operación mediante la gestión de la transacción por parte del subsistema Gestor de Transacciones descrita en el modelo. De acuerdo al resultado de la operación, éxito o fracaso, la respuesta obtenida del Gestor de Transacciones se presenta en la <<navigation class>> *Confirm Pago Comisión* o en la <<navigation class>> *Error Pago Comisión*, según sea el caso de éxito o fracaso.

Por último, el subsistema provee un canal de comunicación entre cada tienda y los Clientes a través de correo electrónico. Este canal, el Vendedor puede aprovecharlo para enviar correos de manera masiva a los Clientes registrados en el CCV, entre ellos, a aquellos que ya realizaron una compra en su tienda. El contenido de los correos puede ir desde boletines, ofertas, novedades, invitaciones o mensajes. Esta funcionalidad se provee mediante la <<navigation class>> *Enviar MailA* y el <<form>> *Datos Mail*. El Vendedor introduce los campos y contenido de correo a enviar mediante el <<form>> *Datos Mail*. En *Enviar MailA* se encuentran definidos las operaciones para el procesamiento de esta información y para el envío efectivo del correo hacia el o los destinatarios especificados.

5.4.4 Subsistema de Administración CCV

En la figura 5.15 se muestra el modelo de navegación para el subsistema Administración CCV. Una vez que un usuario Administrador se ha autenticado correctamente en el CCV accede a la <<navigation class>> *Administración CCV* que es la principal del subsistema de Administración CCV, a partir de la cual se puede navegar hacia los diferentes módulos que este contiene. Las principales funcionalidades de este subsistema son: la modificación de las opciones de configuración para todo el CCV en general, la configuración de parámetros para el manejo de las comisiones, el manejo de las cuentas asociadas a cada tienda, la gestión del padrón de tiendas, la administración de usuarios, la visualización de informes y el envío de correos electrónicos a direcciones conocidas por el CCV.

Estas funcionalidades se proveen mediante clases estereotipadas y asociaciones que a continuación se describen. La posibilidad de modificar la configuración del CCV se provee mediante la <<navigation class>> *Modificar Conf CCV* y los <<form>> *Datos Conf Servidor* y *Datos Conf BDD*. Estos elementos de modelado permiten al usuario Administrador de CVV modificar la configuración del CCV en dos aspectos: la especificación de parámetros asociados al servidor en donde se encuentre hospedado el sitio, y la especificación de parámetros asociados a la administración de la base de datos que se utiliza. Ambos tópicos son esenciales para toda la operación del CCV. En *Datos Conf Servidor* el Administrador introduce los datos de configuración del servidor, por ejemplo, el nombre del directorio raíz, del directorio de comandos, el dominio de las cookies, entre otros parámetros. Por otra parte, mediante el <<form>> *Datos Conf BDD* el Administrador introduce los datos de configuración de base de datos, como por ejemplo, el usuario principal y el password. Ambas configuraciones se guardan en los <<file>> *CCVConf* y *BdDConf*, los cuales se describen en el subsistema de Componentes Comunes.

A partir de la <<navigation class>> *Gestión de Comisiones y Cuentas*, el Administrador tiene acceso a la configuración de las comisiones y al manejo de las cuentas asociadas a cada tienda. La configuración de las comisiones se lleva a cabo mediante la <<navigation class>> *Conf Comisiones* y el <<form>> *Datos Conf Comisiones*. Mediante estos elementos el Administrador especifica si las operaciones que se realicen en el CCV debe causar una comisión, y si es así, el porcentaje de la misma, el criterio para cobrarlas y el periodo de corte para su cobro. Por otra parte, la gestión de las cuentas asociadas a cada tienda se realiza mediante las <<navigation class>> *Gestión Cuentas Tienda*, *Estado de Cuenta Tienda* y *Status Tienda*, los <<index>> *Cuentas Tienda* y *Comisiones por Pedido*, y los <<form>> *Datos Periodo a Visualizar* y *Datos Status*. A *Gestión Cuentas Tienda* se asocia el <<index>> *Cuentas Tienda*, el cual es un listado de todas las cuentas de las tiendas del CCV, esto es, una cuenta por Tienda. Para cada tienda existen varios estados de cuenta, uno por cada periodo que ha transcurrido desde que la tienda existe en el CCV. A cada uno de ellos, el Administrador puede acceder mediante la selección de un periodo específico. Dicha selección se provee en el

<<form>> *Datos Periodo a Visualizar*, el cual se asocia a cada elemento del <<index>> *Cuentas Tienda*. La <<navigation class>> *Estado de Cuenta Tienda* despliega el listado de las comisiones que sean generados a partir de la actividad de la tienda en el CCV. Este listado se representa por el <<index>> *Comisiones por Pedido*. El estado de cuenta incluye montos totales, fechas, y un resumen de la actividad de la tienda en el CCV. Cuando sea necesario, el Administrador debe tener la posibilidad de modificar el estado de una tienda dentro del CCV. Es decir, dentro del CCV una tienda puede estar habilitada, inhabilitada, condicionada, entre otros estados más. Así también, el CCV plantea la asociación de dos tipos de estados para cada tienda, el estado general y el estado financiero. Ambos describen la condición de cada tienda en el CCV. Esta funcionalidad de modificar el estado de cada tienda, se provee mediante la <<navigation class>> *Status Tienda* y el <<form>> *Datos Status Tienda*.

El Administrador del CCV, es el encargado de organizar y gestionar el registro de cada tienda aceptada para integrar el CCV. Para ello debe existir en el subsistema de Administración de CCV la funcionalidad para que el Administrador pueda llevar a cabo estas tareas. En efecto, en el subsistema se provee esta funcionalidad mediante las <<navigation class>> *Manejo Padrón de Tiendas*, *Registro Tienda*, *DocsNueva Tienda* y *DocsNueva2 Tienda*, el <<index>> *Tiendas* y los <<form>> *Datos Tienda* y *Datos Bancos Tienda*. Desde la <<navigation class>> *Manejo Padrón de Tiendas* el Administrador puede navegar hacia *Registro Tienda* para registrar una nueva tienda al CCV. A *Manejo Padrón de Tiendas* se asocia el <<index>> *Tiendas*, el cual es un listado de todas las tiendas actualmente asociadas al CCV. A partir de este listado se puede navegar hacia *Registro Tienda* para realizar la Edición o Eliminación de cada tienda. En *Registro Tienda* se encuentran descritas las operaciones para registrar una nueva tienda (incluyendo la vinculación con los logotipos predeterminados, y la creación de usuarios Vendedores), modificar el registro de una existente y, eventualmente, eliminar el registro de una tienda. Los <<form>> *Datos tienda* y *Datos Banco Tienda* son el medio a través de los cuales el usuario Administrador introduce la información general y la información bancaria respectivamente, información que se procesa en *Registro Tienda* para la ejecución de la operación de registro o modificación. Desde *Registro Tienda* se puede navegar hacia *DocsNueva Tienda* y *DocsNueva2 Tienda*, en las cuales se despliegan las formas y formatos asociados con el registro de tiendas nuevas, y los cuales, una vez impresos, se envían al Vendedor administrador de la nueva tienda asociada al CCV como parte del paquete de notificación que el CCV envía al aceptar su inscripción al CCV.

Otro punto importante a considerar es la gestión de los usuarios. Aunque en el registro de una tienda nueva se incluye un procedimiento para la creación automática de usuarios Vendedores asociados a la tienda, es necesario, además, que el Administrador del CCV tenga la posibilidad de administrar el padrón de usuarios para crear nuevos usuarios Vendedores, usuarios Administradores, eliminar usuarios existentes, otorgar mayor nivel de permisos, etc. Esta funcionalidad está cubierta por el subsistema Administración CCV mediante las <<navigation class>> *Gestión de Usuarios* y *Registro Usuario*, los <<index>> *Listado Tiendas*, *Usuarios de Tienda* y *Usuarios Admon*, y el

<<form>> *Datos Usuario*. A *Gestión de Usuarios* se asocia los <<index>> *Listado Tiendas* y *Usuarios Admon*. A partir del primero de ellos el Administrador selecciona una tienda específica del listado para desplegar en el <<index>> *Usuarios Admon* el listado de usuarios vinculados a la tienda seleccionada. El <<index>> *Usuarios Admon* es un listado de todos los usuarios Administradores que existen para la administración del CCV. En la <<navigation class>> *Registro Usuario* se describen las operaciones para el registro de usuarios nuevos, la modificación de identidades de los usuarios existentes y la eliminación de usuarios. El usuario Administrador introduce la información de los usuarios a través del <<form>> *Datos Usuario*, el cual envía esta información a *Registro Usuario* para que se realicen las operaciones correspondientes.

Al igual que en el subsistema Gestor de Tienda, en el subsistema Administración CCV se provee al Administrador la posibilidad de visualizar informes que reflejen la actividad del CCV en su conjunto. Esta funcionalidad se provee mediante las <<navigation class>> *Informes para CCV*, *Informe Admon*, el <<index>> *Opciones de Informe* y el <<component>> *Generador Reportes*, este último descrito en el subsistema de Componentes Comunes. En *Informes para CCV* se presentan las familias de informes que se encuentran disponibles. Cada familia contiene un grupo de opciones de visualización, el cual se presenta mediante el <<index>> *Opciones de Informe*. Cuando el Administrador selecciona un informe en específico este se presenta en la <<navigation class>> *Informe CCV* por medio del <<component>> asociado *Generador Reportes*. La especificación de parámetros que se envían a este generador se realiza en la <<navigation class>> *Informe CCV*, el generador realiza la consulta en las fuentes de información y se genera un resultado y se presenta en un formato claro. De esta manera, a partir de la <<navigation class>> *Informe CCV* y el *Generador Reportes* se presenta información relevante al Administrador para *cuantificar* las actividades del CCV.

Por último, el subsistema provee un canal de comunicación entre el CCV y los Vendedores y entre el CCV y los Clientes registrados en el CCV a través de correo electrónico. La administración del CCV, a través de esa funcionalidad, puede enviar correos de manera masiva, o uno a uno, a Vendedores y/o Clientes registrados en el CCV. El contenido del correo puede ir desde notificación de eventos a Vendedores, hasta la promoción del sitio con los Clientes. Esta funcionalidad se provee mediante la <<navigation class>> *Enviar MailA* y el <<form>> *Datos Mail*. El Administrador introduce los campos y contenido de correo a enviar mediante el <<form>> *Datos Mail*. En *Enviar MailA* se encuentran definidas las operaciones para el procesamiento de esta información y para el envío efectivo del correo hacia el o los destinatarios especificados.

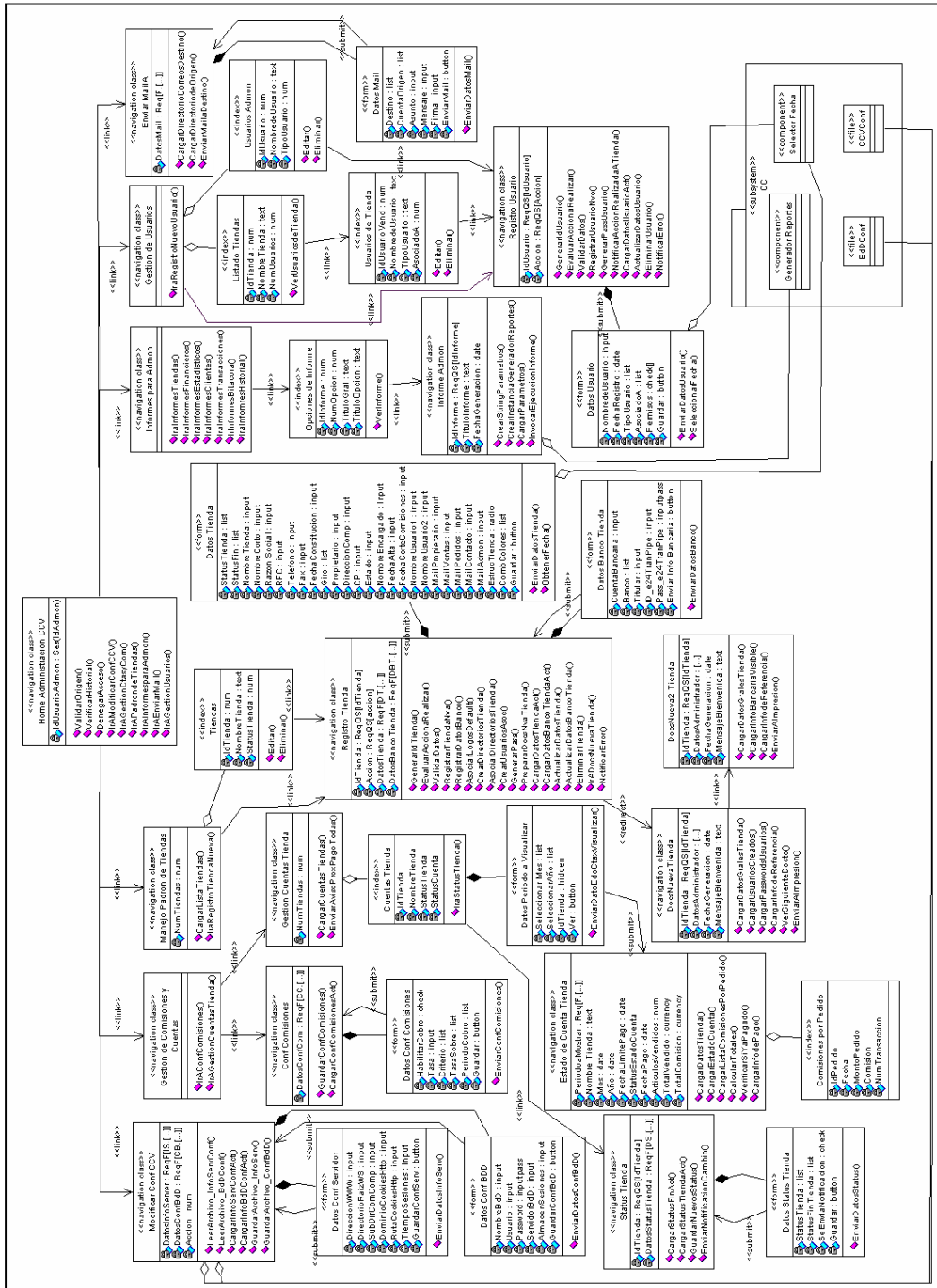


Figura 5.15: Modelo de navegación Administración CCV

5.4.5 Subsistema Componentes Comunes

Los componentes comunes a todo el CCV se agrupan en el subsistema del mismo nombre. Aunque estos elementos no son navegables directamente por si solos, sino únicamente cuando son requeridos en la realización de operaciones y procedimientos que se describen en otros subsistemas del CCV, tienen un significado arquitectónico que corresponde a los requerimientos de tareas solicitadas por los usuarios. El subsistema de Componentes Comunes es un repositorio de elementos que dan soporte a los distintos subsistemas, por lo cual, ningún elemento guarda asociación con otro dentro del mismo subsistema, tal como se observa en la figura 5.16. A continuación se describen los elementos que aparecen en el subsistema.

En el desarrollo de aplicaciones Web es común, en la práctica, la utilización de elementos de software desarrollados para una tarea específica. Este es el caso de los elementos <<component>> *Selector Fecha*, *Explorador Archivos* y *Generador Reportes*. Este tipo de elementos, como se comentó en apartados anteriores, representan a componentes de software que se incluyen dentro de una aplicación Web para realizar una tarea específica. En este caso, los componentes que se incluyen dentro de este subsistema son representativos de una gran colección de componentes que se encuentran a disposición de los desarrolladores de aplicaciones Web que pueden ser utilizados en la implementación del CCV. Sin embargo, en virtud de que detalles de implementación se han discriminado en lo posible en el desarrollo de este modelo, se han incluido únicamente tres de este tipo de componentes.

El <<component>> *Selector Fecha* es un componente de software que permite a los usuarios introducir fácilmente una fecha en un campo de un <<form>> mediante la selección de la misma a través de una ventana en la que se despliega un calendario dispuesto de una manera fácilmente entendible. El componente obtiene la selección del usuario y la entrega al <<form>> en un formato previamente definido. Este componente es muy útil ya que con él se logra homologar el formato de la fecha que el usuario introduce. El <<component>> *Explorador Archivos* es un componente de software que permite a los usuarios la exploración y selección de un archivo mediante una ventana de interfase familiar al usuario. Una vez que el usuario selecciona el archivo el cual la aplicación debe procesar, el explorador retoma la selección y entrega a la aplicación el nombre del archivo seleccionado en un formato claro y listo para que sea utilizado en los procedimientos y operaciones que lo requieran.

El <<component>> *Generador Reportes* es un elemento de software de uso común en el desarrollo de sistemas. En el modelo, el <<component>> *Generador Reportes* representa a un componente de software provisto por una herramienta generadora de reportes para la visualización y generación de informes en aplicaciones Web. Entre estas herramientas podemos encontrar a JasperReports, Crystal Reports, Agata Reports, entre muchas otras existentes en el mercado. Estas herramientas permiten al desarrollador de aplicaciones la posibilidad de diseñar informes completos sobre

fuentes de datos, los cuales se almacenan en archivos y posteriormente se invocan desde la aplicación mediante un componente provisto por la herramienta. Este componente es el que se representa por medio del <<component>> *Generador Reportes*, el cual es agregado en una <<navigation class>>, en donde se procesan una serie de parámetros que son los que el Generador Reportes utiliza para la vinculación con el archivo del informe correspondiente ubicado en el directorio de Archivos de Informes y la presentación del mismo. En este subsistema aparece el <<object COM>> *e24Tranpipe* que se incluye como un componente de software a través de cual se realizarán las operaciones transaccionales entre el CCV y la entidad bancaria. En el siguiente apartado se comenta al respecto de este componente y de la seguridad que su uso provee a la aplicación.

Por último, en el subsistema de Componentes Comunes se observan cinco paquetes. La inclusión de estos paquetes en el subsistema Componentes Comunes es una propuesta de organización y clasificación de todos los elementos que posiblemente se requieran en la implementación. Estos paquetes representan a familias de elementos que se utilizan frecuentemente en el diseño y programación de aplicaciones Web. En *Includes* se conglomeran elementos tales como archivos y librerías los cuales pueden ser incluidos (insertados) en el código de la aplicación. Por otra parte, *Classes* se propone como un repositorio de clases de objetos que puedan ser utilizadas en cualquier subsistema. *Styles* representa a un repositorio de hojas de estilo CSS, las cuales son comunes y muy importantes en la presentación y formato de contenidos en las aplicaciones Web. *Boxes* denota a un conjunto de bloques de código que se utilizan para el trazado de cuadros en el diseño visual de la aplicación, utilizados frecuentemente en la disposición de tablas y etiquetas. Por último, *Modules* se describe como un repositorio de módulos de código reutilizable e invocable en toda la aplicación.

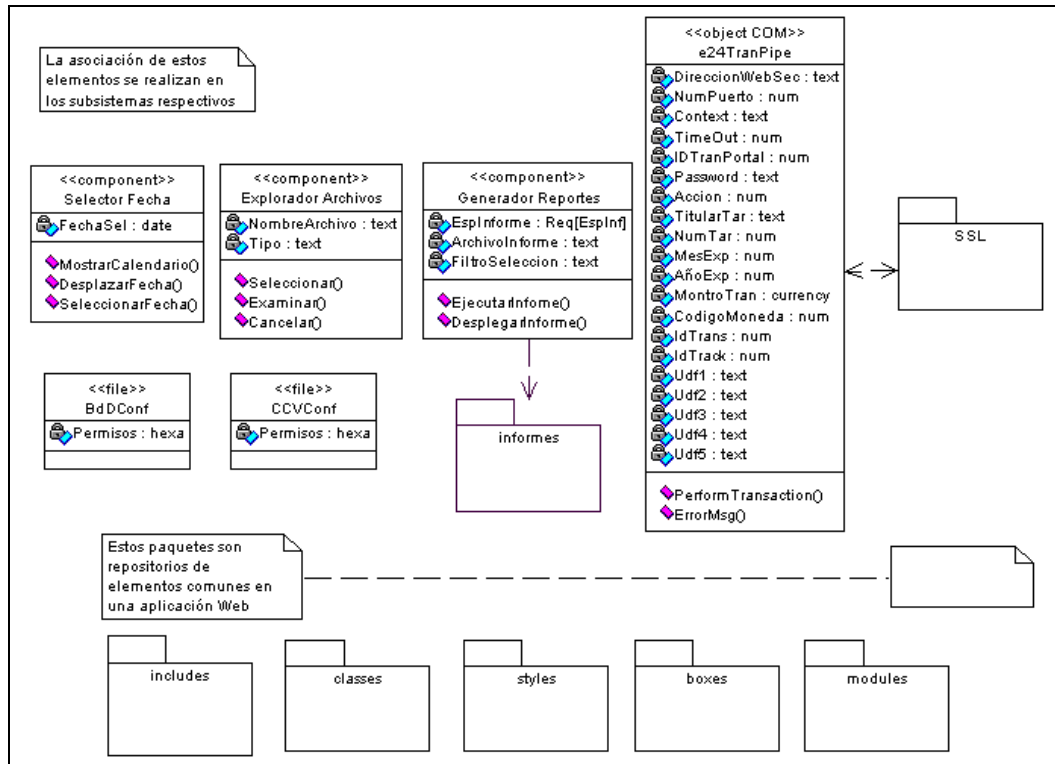


Figura 5.16: Modelo de navegación Componentes Comunes

5.4.6 Acerca del <<object COM>> e24Tranpipe y la seguridad del CCV

Este componente de software modelado con el estereotipo <<object com>>, es el componente a través de cual se modela fundamentalmente un aspecto sumamente importante en cualquier aplicación de comercio electrónico que es la seguridad disponible en el CCV. En síntesis, el <<object com>> *e24TranPipe* es un componente de software que constituye una interfase a través de la cual se llevan a cabo las transacciones de dinero entre el CCV y la entidad bancaria. Específicamente este componente es utilizado por el subsistema Gestor de Transacciones, tal y como se muestra en la figura 5.13, para llevar a cabo las operaciones asociadas con las transacciones de dinero entre el CCV y la entidad bancaria. Este componente de software es provisto por la propia entidad bancaria (Banco) luego de establecer un contrato de uso entre la propia entidad bancaria y el interesado que en este caso es el CCV (tiendas que lo conforman) y el equipo de desarrollo de la aplicación Web. El uso del *e23TranPipe* requiere de un identificador de usuario con una contraseña asociada y una cuenta de banco sobre la cual se llevarán a cabo las transacciones. En el prerrequisito del modelo descrito en la Fig. 3.1, cuando un candidato a Vendedor

solicita su registro al CCV envía una copia del estado de la cuenta de banco. Esta información, en el caso en que el registro del Vendedor al CCV sea aprobado, es la que el equipo de administración del CCV utiliza para gestionar el identificador de usuario y la contraseña correspondiente a dicha tienda en específico para poder realizar operaciones a través del *e24TranPipe*. De esta manera, para cada tienda existirá un identificador de uso (con su correspondiente contraseña) de *e24TranPipe* y una cuenta asociada sobre la cual se llevarán a cabo las transacciones. Como cualquier otra tienda, el CCV también necesita un conjunto de estos mismos elementos, en virtud que se provee la funcionalidad de realizar el pago de comisiones a través de la aplicación Web.

El *e24TranPipe* es un objeto interfase desarrollado bajo la especificación TranPortal. El sistema de captura de transacciones TranPortal provee un método de integración de procesamiento de transacciones de tarjetas bancarias en una variedad de productos de software, sitios Web, software de servidor y software de Cliente a través de Internet. El sistema de captura de transacciones es un conjunto de protocolos de comunicación, formatos de transmisión de transacciones, formatos de respuesta de transacciones y mensajes de error que pueden ser usados para desarrollar una interfase u objeto que provee al desarrollador todo un conjunto de herramientas simples para comunicar con servidores de procesamiento TranPortal. El objeto *e24TranPipe* esta desarrollado bajo esta especificación y provee una interfase de alto nivel para uso en aplicaciones Web de comercio electrónico, como lo es el CCV aquí propuesto.

Las especificaciones del protocolo de comunicación utilizado por el *e24TranPipe* son las siguientes:

- Protocolo: http
- Puerto: 443 Verisign 3.0 SSL Certificate
- Objetivo (acción):
<https://www.vposhsbc.com/servlet/TranPortalHTTPServlet>
- Método: Post
- Tipo-Contenido: application/www-form-urlencoded or application/x-www-form-urlencoded
- Formato de Transmisión de Datos: Respuesta de una cadena de texto, delimitada por dos puntos...string:string:string:
- Nivel de encriptación: SSL Versión 3.0

El sistema de captura de transacciones requiere que el desarrollador se comunique con una gama de variables a través de Internet por medio de posteo http hacia un servidor TranPortal, como por ejemplo:

<https://www.vposhsbc.com/servlet/TranPortalHTTPServlet>. Esta dirección es la dirección del servidor TranPortal de HSBC. Toda la información posteada a este servidor debe de estar en la forma de una cadena URL codificada de pares de valores. A continuación se presenta un ejemplo de una transacción estándar:

id=TranPortalID&password=password&action=1&card=4444333322221111&expYear=2005&expMonth=12&CVV2=123&member=Joe Smith &addr=50 Main St.&zip=12345&amt=10.00&trackid=unique tracking id&udf1=User Defined Field &udf2=User Defined Field&udf3=User Defined Field&udf4=User Defined Field&udf5=User Defined Field

En la transacción de ejemplo que se muestra arriba se pueden identificar las variables que son utilizadas en la transmisión de una transacción. Estas variables son las que aparecen en el <<object COM>> *e24TranPipe* como atributos. A continuación se describen estas variables de transmisión de transacción.

- **Id:** Número de identificación TranPortal. El ID TranPortal ID es emitido por la entidad bancaria proveedora del servicio y es utilizado para identificar el comercio y a la terminal para procesamiento de transacciones.
- **Password:** Es el Password TranPortal. Este password es emitido por la entidad bancaria para autenticar el comercio y terminal. Tan pronto se emite el posteo http, la información es cifrada y este password es asegurado y ocultado. Con esto, se asegura que ningún otro comercio haga uso del mismo identificador para realizar operaciones fraudulentas.
- **Acción:** Especifica la acción a realizar mediante el TranPortal. Las siguientes acciones son acciones válidas:
 1. Venta
 2. Crédito
 3. Cancelación de Venta
 4. Autorización
 5. Captura
 6. Cancelación de Crédito
 7. Cancelación de Captura
 8. Cancelación de autorización para valores mayores o igual a 9
- **Card:** Número de tarjeta de crédito.
- **Expyear:** Fecha Año de Expiración.
- **Expmonth:** Fecha Mes Expiración
- **Expday:** Fecha Día de Expiración.
- **Member:** Nombre completo del tarjeta habiente.
- **Amt:** Monto de la transacción.
- **Currencycode:** El código de moneda de la transacción.
- **Transid:** Es un identificador de Transacción generado al interior del componente que se usa para identificar la transacción original con el propósito de darle seguimiento a la transacción y opcionalmente, realizar movimientos posteriores sobre ella como cancelaciones y transacciones de capturas.
- **Trackid:** Es una variable para que la aplicación Web que haga uso del *e24TranPipe* la utilice como un identificador de rastreo único que le permita darle seguimiento a la transacción.

- **Udf1, udf2, udf3, udf4 y udf5:** Los Campos definidos por el usuario (User Defined Field, UDF) pueden ser usados para pasar y salvar cualquier dato adicional de la transacción que el desarrollador desee archivar en la transacción y estar disponible como un criterio de búsqueda o rastreo de la transacción.

Cada transacción procesada a través del *e24TranPipe* devuelve una respuesta que se recibe como un string simple de texto. Los valores de retorno de la transacción procesada son separados (delimitados) por caracteres ":". Es responsabilidad del desarrollador de interpretar esta cadena de caracteres al incluir el *e24TranPipe*. El siguiente es un ejemplo de una respuesta obtenida como resultado del procesamiento de una transacción realizada con el *e24TranPipe*:

Result:Auth:Ref:AVR:Date:TransId:TrackId:UDF1:UDF2:UDF3:UDF4:UDF5

En el CCV la interpretación de esta cadena se delega al subsistema Gestor de Transacciones. A continuación se describen las variables de la respuesta.

- **Result:** En esta variable se devuelve el código del resultado de la Transacción. Este resultado es el que se evalúa para determinar si la transacción fue efectuada satisfactoriamente o si ha ocurrido un error. Los valores de esta variable se citan a continuación.
 - CAPTURED: La tarjeta fue capturada propiamente.
 - APPROVED: La transacción fue aprobada.
 - VOID: La transacción fue propiamente cancelada.
 - NOT CAPTURED: La tarjeta no fue capturada.
 - NOT APPROVED: La transacción no fue aprobada.
 - NOT VOIDED: La transacción no fue cancelada.
 - DENIED BY RISK: Transacción denegada por considerarse de riesgo.
 - FAILED AVS: La transacción no pasó la verificación de dirección.
 - HOST TIMEOUT: El sistema de autorización no respondió dentro del límite de tiempo.
- **Auth:** El número de autorización resultado de la transacción.
- **Ref:** El número de referencia resultado de la transacción. Este número ó serie de letras son usadas para propósitos referenciales para algunas instituciones adquirentes y deben de ser propiamente guardadas.
- **AVR** - AVR es una letra simple que provee información acerca de la información del tarjetahabiente. La letra que se devuelve indica que tanto concuerda el número de tarjeta con el nombre del tarjetahabiente y la fecha de expiración.
- **Date:** Es la fecha de la transacción en el formato del sistema de autorización.

- **TransId:** Es el identificador único de la transacción emitido por el Commerce Gateway de la entidad bancaria.
- **TrackId:** Es el mismo identificador de rastreo que se envió en la solicitud de la transacción. Se utiliza en la respuesta para verificar que la transacción se recibió correctamente.
- **Udf1, udf2, udf3, udf4 y udf5:** Estos campos definidos por el usuario (User Defined Field, UDF), al igual que en el envío, pueden ser usados para pasar y salvar cualquier dato adicional de la transacción que el desarrollador desee archivar en la transacción y estar disponible como un criterio de búsqueda o de rastreo de la transacción.

Sí durante la transmisión de la información de la transacción ocurre un error, entonces el formato de respuesta contendrá una cadena indicando que un error ocurrió. Todas las respuestas de error inician con el identificador de ERROR. Por lo tanto es importante para el desarrollador revisar la cadena del mensaje de respuesta y determinar si un error ha ocurrido.

Fuente: HSBC.

Con la inclusión en el modelo del CCV del componente *e24TranPipe* se proveen servicios de seguridad suficientes para una aplicación de comercio electrónico como lo es el CCV. Los servicios de seguridad definidos por el modelo de referencia OSI (Open System Interconnection) en el documento ITU-T X.800 son: Autenticación, Control de Acceso, Confidencialidad, Integridad y No rechazo. La definición de cada uno de estos servicios se encuentra ampliamente documentada en dicho documento oficial así como también en la bibliografía especializada. En la práctica, para proporcionar estos servicios de seguridad es necesario proveerlos a varios niveles mediante mecanismos de seguridad. El objeto *e24TranPipe* implementa diferentes mecanismos de seguridad a partir de los cuales se proveen varios de estos servicios. El primero de ellos es el servicio de **autenticación** ya que, en la comunicación entre el componente incrustado en la aplicación CCV y la entidad bancaria prestadora del servicio, se utiliza un identificador único y una contraseña que sirven para autenticar de manera única al comercio (en este caso la tienda registrada en el CCV) y al componente *e24TranPipe* asociado con la cuenta correspondiente. Este es un mecanismo de autenticación simple y, como se comentó anteriormente, el identificador y la contraseña son provistos por la entidad bancaria. El **control de acceso** se da normalmente en un contexto de autenticación previa y de lo que se trata es el permitir que un usuario pueda acceder a servicios o realizar distintas actividades conforme a su identidad. En el *e24TranPipe* el servicio de control de acceso se provee a partir de la autenticación del componente asociado al CCV con la entidad bancaria. Una transacción es enviada a la entidad bancaria y dentro de la transacción se incluye el identificador y la contraseña de componente asociado al CCV. Cuando la entidad bancaria recibe la solicitud de transacción verifica en sus fuentes de datos que tanto el identificador y la contraseña autentiquen a un usuario válido del servicio de pago a través de Internet. Si el solicitante es identificado como válido entonces procede a continuar con la ejecución de la transacción, de lo contrario, la función de control de acceso de la entidad bancaria

rechazará el intento, al tiempo que puede informar del incidente, con el propósito de generar una alarma y/o registrarlo. El servicio de **confidencialidad** proporciona protección contra la revelación deliberada o accidental de los datos en una comunicación. En el *e24TranPipe* el servicio de confidencialidad es soportado por el estándar SSL/TLS y el algoritmo de cifrado que utiliza dicho estándar que es el RC4. La cadena que describe a la transacción se postea de manera oculta hacia la entidad bancaria mediante el puerto 443 (es el que utiliza SSL Versión 3.0) del Cliente hacia la entidad bancaria. El servicio de **integridad de datos** consiste en lograr que los datos de las transacciones no sean alterados en la trayectoria. En este sentido, al igual que el servicio de confidencialidad, en el *e24TranPipe* el servicio de integridad es soportado por el estándar SSL/TLS y el algoritmo asimétrico RSA incluido en dicho estándar. Por último, el servicio de **No rechazo** consiste en garantizar que el emisor de un mensaje no podrá posteriormente negar haberlo enviado, mientras que el receptor no podrá negar haberlo recibido. Esta protección se efectúa por medio de una colección de evidencias irrefutables que permitirán la resolución de cualquier disputa. Las firmas digitales constituyen el mecanismo más empleado para este fin. En este caso, en el *e24TranPipe* el servicio de no rechazo se provee en dos niveles. El primero de ellos es provisto en la transmisión de los datos por el propio algoritmo RSA que utiliza el estándar SSL/TLS. En un segundo nivel, el *e24TranPipe* implementa el no rechazo incluyendo en los mensajes las variables de comprobación TransID, TranckID, auth y ref; las dos últimas solo en la respuesta del banco hacia el CCV. Estas variables constituyen elementos de comprobación de la transacción tanto de la solicitud del CCV mediante el *e24TranPipe* así como de la respuesta devuelta por parte del banco por el mismo conducto.

En resumen, y de acuerdo a lo expuesto, se puede decir que el esquema de seguridad provisto en el CCV con la utilización del <<object com>> *e24TranPipe* proporciona un nivel de seguridad apropiado para una aplicación Web de comercio electrónico al cumplir satisfactoriamente los servicios de seguridad descritos en el OSI.

Por otra parte cabe acotar que el <<object COM>> *e24TranPipe* provee un medio seguro en la comunicación entre la aplicación y la entidad bancaria, no obstante al interior del CCV se deben tomar en cuenta otras medidas de seguridad para evitar usos malintencionados. Entre las medidas que toma en cuenta el modelo de CCV es la validación de <<navigation class>> origen, es decir, la verificación de que la procedencia de los datos que han de manipularse en una <<navigation class>> destino sea la correcta y no producto de un intento malintencionado. Además esta verificación se complementa con la utilización de variables de sesión para asegurar que la solicitud de un pago sea precedido por una correcta autenticación de un usuario válido del CCV. Ejemplo de la inclusión de estos mecanismos en el modelo se observan en la figura 5.12, que describe el modelo de navegación para el subsistema ePlaza, con la operación ValidarOrigen () y el atributo tipo Ses[IdCliente] IdClienteP.

Capítulo 6

Interacción Temporal

En este capítulo se presentan las interacciones en orden temporal entre las diferentes <<navigation class>> y demás clases estereotipadas que aparecen en el modelo navegacional. En un primer apartado se describe la semántica de los elementos de modelado que se utilizan en los diagramas de secuencia. Posteriormente se listan los diagramas de secuencia correspondientes a cada subsistema.

6.1 Semántica de convenciones utilizadas en los diagramas de secuencia

En esta sección se comentan algunas otras convenciones que se utilizan en los diagramas de secuencia y que son importantes para la mejor comprensión de los diagramas.

6.1.1 La estructura //

Esta estructura se utiliza en los diagramas de secuencia para etiquetar mensajes entre las clases estereotipadas descritas en el modelo de navegación. Los mensajes etiquetados con // se asocian a un hecho o conjunto de hechos los cuales se llevan a cabo para que el control pase de la clase estereotipada emisora del mensaje hacia la clase estereotipada receptora del mensaje.

El contenido del mensaje // no tiene una sintaxis definida ni estructurada. El mensaje es descriptivo de los hechos que desencadenan la transferencia del control en el tiempo entre las clases estereotipadas involucradas. En los diagramas de secuencia que se presentan en este apartado, esta descripción se realiza en lenguaje natural y denotando con claridad actividades o hechos que se llevan a cabo en la aplicación. Esta construcción es utilizada también en [Conallen02].

6.1.2 La semántica asociada a las etiquetas de Comentario

En los diagramas de secuencia que se presentan en este apartado se incorpora un uso distinto al habitual de las etiquetas de comentario. Este nuevo uso se propone como una opción para representar cursos alternativos de flujo dentro de un diagrama de secuencia. Como sabemos, los diagramas de secuencia UML representan la realización de operaciones en función del tiempo mediante el paso de mensajes entre los elementos de diseño. La secuencia completa de un diagrama se compone por dos tipos de flujos, el básico y el alternativo [Rosenberg01]. El básico se compone por todos aquellos mensajes y operaciones que forman parte de la secuencia y que se realizan invariablemente dentro de ella, sin depender de condiciones o circunstancias indicadas en los mensajes. Por otra parte, el flujo alternativo se compone de los mensajes y operaciones que se realizan de manera condicionada, es decir, que su realización depende de condiciones y circunstancias que resultan de mensajes (comúnmente de respuestas) o bien de resultados de operaciones realizadas con anterioridad dentro de la secuencia, y son excluyentes entre ellos. Para representar estos flujos, UML propone varios mecanismos, entre estos, la realización de un diagrama de secuencia por cada flujo alternativo dentro de la secuencia. En este sentido, UML propone también la inclusión de todos los flujos alternativos dentro del mismo diagrama de secuencia e indicar la separación de cada flujo alternativo mediante un espacio en blanco en la línea de tiempo del objeto del cual dependen dichos flujos alternativos.

Sin embargo, para el modelado de el CCV ambas opciones resultan poco convenientes por el tamaño de la aplicación que se esta modelando y, consecuentemente, por el tamaño de las secuencias propuestas. Debido a esto, en este modelado se incluye en estas secuencias una estructura basada en las etiquetas de comentarios disponibles en la mayoría de las herramientas de modelado UML. Estas etiquetas de comentario se utilizan para crear una estructura para identificar a cada flujo alternativo. Esta estructura basada en estas etiquetas se muestra en la figura 6.1.

En dicha figura podemos ver la estructura propuesta, donde se pueden identificar claramente las secuencias alternativas en base a la evaluación de un mensaje. Bajo esta misma lógica se pueden incluir N flujos alternativos y excluyentes. Normalmente, después de la conclusión de cada flujo alternativo la secuencia finaliza, sin embargo, la secuencia puede proseguir después de la conclusión de un flujo alternativo, para lo cual se sugiere incluir la anotación correspondiente.

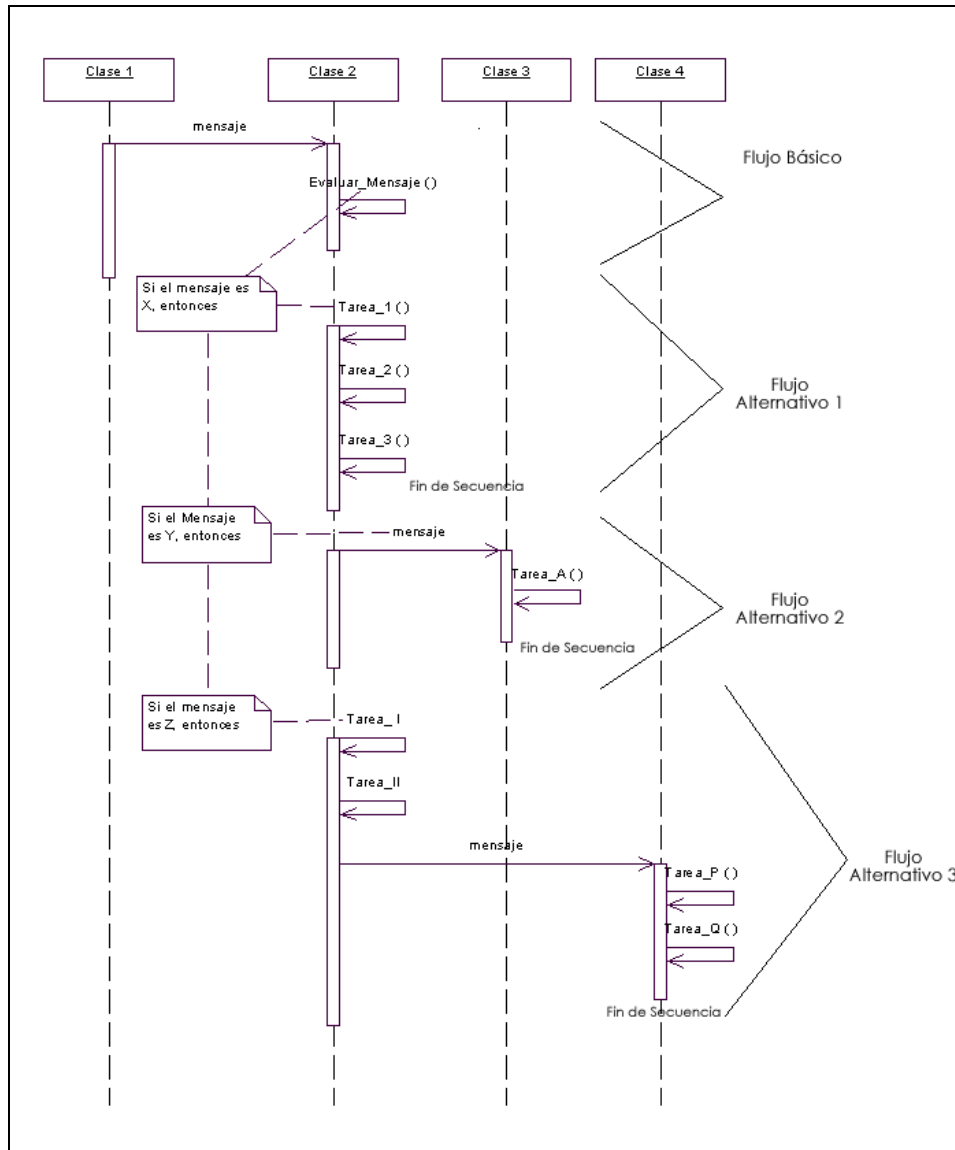


Figura 6.1: Semántica asociada a las etiquetas de comentario

6.2 Secuencias del CCV

6.2.1 Secuencia de ePlaza: Adición de un Producto al Carrito

Esta secuencia describe el procedimiento para la adición de un producto al carrito de compras. Esta secuencia comienza con el acceso por parte del Cliente al CCV mediante la introducción de la dirección URL de *Home CCV*, la <<navigation class>> inicial. *Home CCV* carga la Configuración general mediante la operación *CargarConfGral()*, carga la interfase del CCV con la operación *CargarInterfaceCCV()* y carga el Listado de Tiendas con la operación *CargarListaTiendas()*. El Cliente entonces selecciona visitar una tienda específica del listado de tiendas e ingresa a *Home Tienda*. Esta <<navigation class>> *Home Tienda* carga la configuración de la tienda, carga la interfase de la tienda y carga el catálogo correspondiente a la tienda mediante las operaciones *CargarConfTienda()*, *CargarInterfaseTienda()* y *CargarListaCatálogoTienda()*, respectivamente. A continuación el Cliente selecciona una de las categorías del catálogo y al hacer esto, se despliega el listado de los productos que contiene dicha categoría. El Cliente ahora selecciona ver el detalle de un producto de dicho listado, el cual se muestra en la <<navigation class>> *Producto*. En *Producto* se despliega la descripción del producto, especificaciones, precio, etc. El Cliente ahora selecciona agregar el producto al carrito, esto mediante la operación *EnviarDatoProductoaAgregar()* definida en el <<form>> *Agregar Producto a Carro*. Cuando esto se realiza los datos del producto se envían a la <navigation class>> *Carrito*. Cuando es el primer producto que se está añadiendo al carrito de un Cliente en específico se deben realizar las operaciones de *InicializarCarritoVisitante()*, *GenerarIDPedido()*, *GenerarVarSesIdPedido()* e *IniciarTiempoExp()*. Con estas operaciones se inicializa el carrito visitante, se genera un identificador del pedido, se genera una variable de sesión con el identificador del pedido, y se inicializa el tiempo de caducidad de las variables generadas. El identificador del pedido y su depósito en una variable de sesión se utiliza como mecanismo de control para asociar a cada pedido con el Cliente correspondiente en un periodo de tiempo específico. Esta serie de operaciones únicamente se realizan cuando un Cliente agrega su primer producto al carrito. Indistintamente el procedimiento de adición de producto al carrito continúa con la operación *AgregarProducto()*, *CargarListaProductosenCarro()* y *CalcularTotales()*. Con estas operaciones se realiza la adición física del producto al carrito, se actualiza la lista de los productos que ya se encuentran en el carrito y se calculan los nuevos totales. En este momento el producto ya se encuentra en el carrito del Cliente. Por último, el Cliente puede elegir entre continuar comprando regresando al *Home Tienda*, o bien elegir iniciar el proceso de pago del pedido con la Autenticación del Cliente.

Figura 6.2: Secuencia Adición de Producto a Carrito

6.2.2 Secuencia de ePlaza: Búsqueda de Productos

La secuencia para la búsqueda de un producto dentro del CCV se muestra en siguiente figura. La secuencia comienza cuando el Cliente ingresa al CCV. A Home CCV se asocia el <<form>> *Buscar*. El Cliente introduce los parámetros de la búsqueda y solicita dicha búsqueda mediante la operación del <<form>> *EnviarDatosBusqueda()*. Los parámetros de la búsqueda son enviados a la <<navigation class>> *Búsqueda en CCV* en donde se realiza la búsqueda en las fuentes de datos mediante la operación *RealizarBúsqueda()* para posteriormente cargar los resultados en el <<index>> *Productos Encontrados*. La secuencia concluye con la selección por parte del Cliente de uno de los productos de este listado de productos encontrados para visualizar los detalles del mismo mediante en la <<navigation class>> *Producto*.

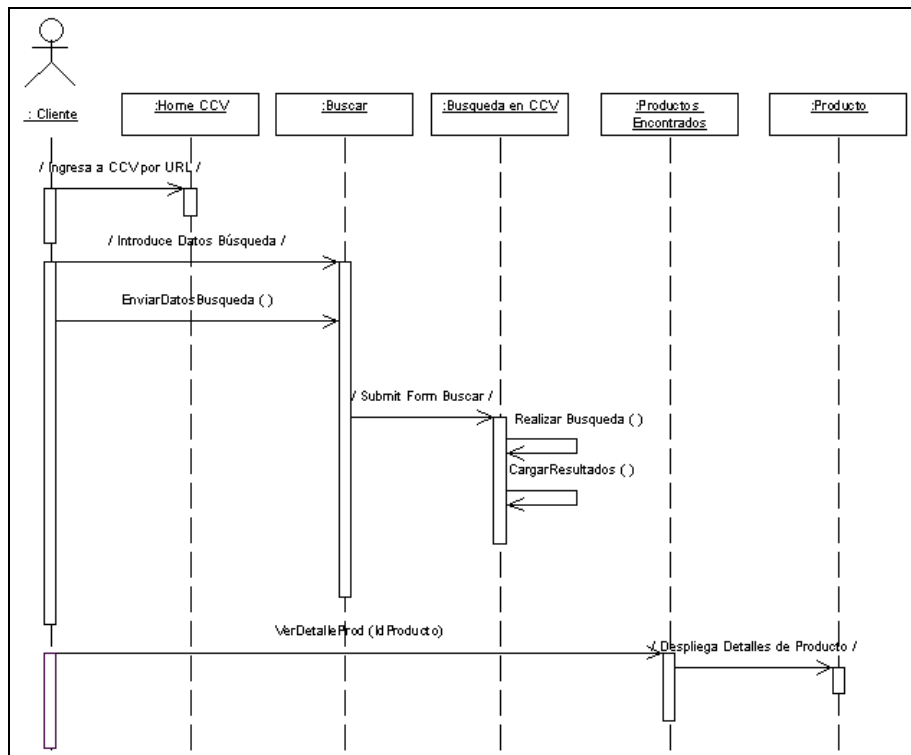


Figura 6.3: Secuencia Búsqueda de Productos

6.2.3 Secuencia de ePlaza: Actualización de Artículo en Carrito

En la secuencia de la figura 6.4 se muestra el flujo de las operaciones que se realizan para la gestión de los productos que ya se encuentran agregados al carrito de compras de un Cliente. La secuencia inicia cuando un Cliente ingresa a la <<navigation class>> *Carrito*. El ingreso puede darse después de haber agregado un producto al carrito o bien, desde la <<navigation class>> *Home Tienda*. Ya estando en *Carrito* se carga la lista de productos ya incluidos en el carro y se realiza el cálculo de los totales del pedido en general. A continuación el Cliente introduce los datos de actualización de cada línea correspondiente a cada producto adicionado al carrito. Estos datos corresponden a la cantidad requerida de cada producto o bien, marcar el producto para que sea eliminado del pedido. Por medio del <<form>> *Actualizacion Linea* estos datos son enviados a la misma <<navigation class>> *Carrito*, la cual dependiendo de los datos introducidos por el Cliente, actualiza las cantidades requeridas de cada producto y elimina aquellos que fueron marcados con ese propósito. Estas tareas las realizan las operaciones *ActualizarCantidadesProd()* y *EliminarProductos()*. A continuación que se realizan estas operaciones se refresca el carrito mediante la operación *RefrescarCarro()* para desplegar los cambios realizados. La secuencia concluye con la elección del Cliente de continuar comprando regresando al *Home Tienda*, o bien eligiendo iniciar el proceso de pago del pedido con la Autenticación del Cliente.

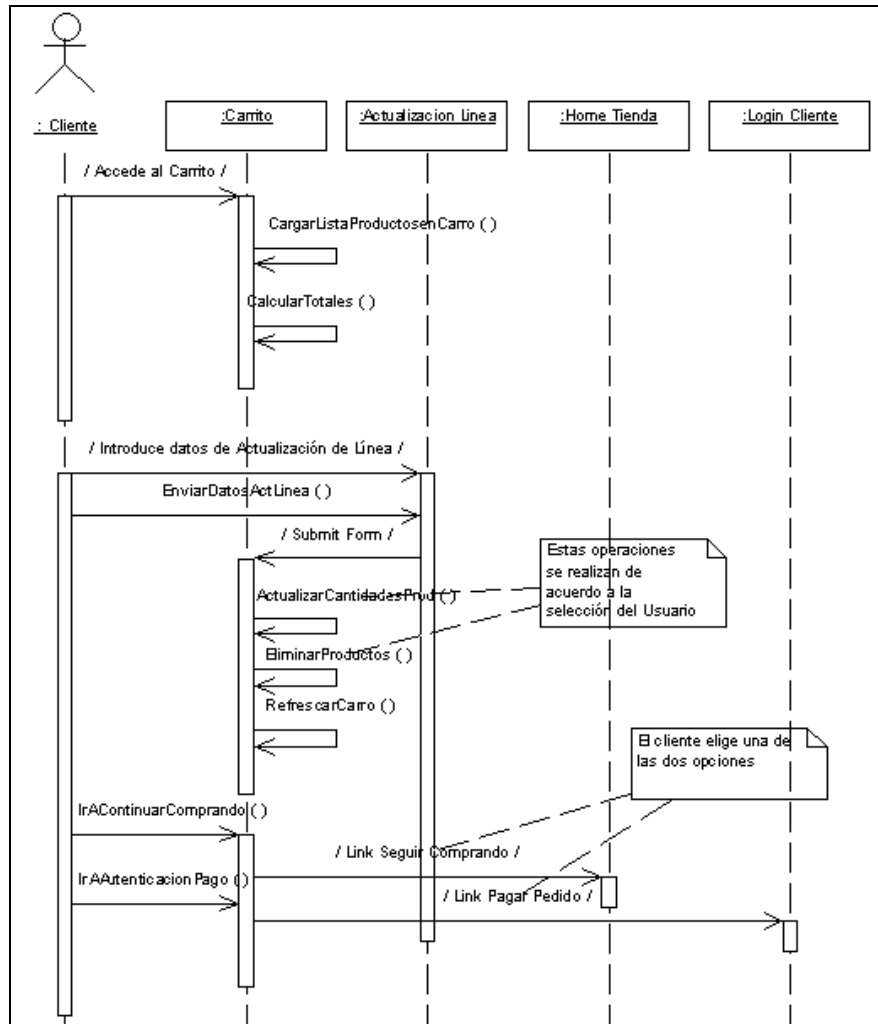


Figura 6.4: Secuencia Actualización Artículo Carrito

6.2.4 Secuencia de ePlaza: Autenticación de un Cliente nuevo

Esta secuencia, que se muestra en la figura 6.5, describe el proceso de autenticación de un Cliente nuevo. Esta autenticación incluye el registro de dicho Cliente. La secuencia inicia cuando el Cliente selecciona pasar a pagar el pedido realizado desde la <<navigation class>> *Carrito*. Ya estando el control en la <<navigation class>> *Login Cliente*, el Cliente ya que no cuenta con un nombre de usuario y un password, selecciona ir a registrarse como Cliente nuevo en la <<navigation class>> *Registro Cliente*. Cuando esta <<navigation class>> se carga, lo primero que realiza es la evaluación de la tarea que va a realizarse en esa clase que, en este caso, es el registro de un nuevo Cliente. La secuencia continua con la introducción por parte del Cliente de todos y cada uno de los datos de registro requeridos por la aplicación. Esta introducción se realiza sobre el <<form>> *Datos Cliente*. Una vez que haya completado los campos requeridos, el Cliente indica al <<form>> el envío de esta información mediante la operación *EnviarDatosCliente()*. Los datos son enviados a la misma <<navigation class>> *Registro Cliente* en donde se verifica la validez de los datos proporcionados por el Cliente mediante las operaciones *ValidarClienteExistente()*, *ValidarDatosCompleto()*, *ValidarCorreoReal()* y *VerificarMatchPass()*, con las cuales se valida que el Cliente no se encuentre previamente registrado, que el Cliente haya proporcionado todos los datos requerido, que el correo proporcionado por el Cliente sea válido y que el password proporcionado haya sido introducido correctamente en la forma de registro, respectivamente. En este punto se tienen dos flujos alternativos dependiendo de los resultados que arrojen dichas validaciones. Si en alguna de estas validaciones no es satisfactoria, entonces la <<navigation class>> *Registro Cliente*, debe rellenar los campos de los datos que si pasaron la comprobación y notificar al Cliente que ocurrió un error. Si todas las validaciones fueron satisfactorias, entonces *Registro Cliente* debe realizar las operaciones *GenerarIdClienteNvo()*, *RegistrarClienteNvo()*, *EnviarConfirmacionxMail()* e *IniciarSesionCliente()*. Mediante la operación *GenerarIdClienteNvo()* se genera el identificador del Cliente a registrar. *RegistrarClienteNvo()* realiza el registro físico de la información del nuevo Cliente en las fuentes de datos de la aplicación. *EnviarConfirmacionxMail()* es una operación para por la cual la aplicación notifica al Cliente vía correo electrónico que su registro al CCV fue satisfactorio. El inicio de sesión del Cliente recién registrado se realiza mediante la operación *InicioSesionCliente()*. Por último, la <<navigation class>> *Registro Cliente* redirecciona la navegación hacia la <<navigation class>> *Resumen Pago* mediante la operación *RedireccionarA()* para continuar con el proceso de pago.

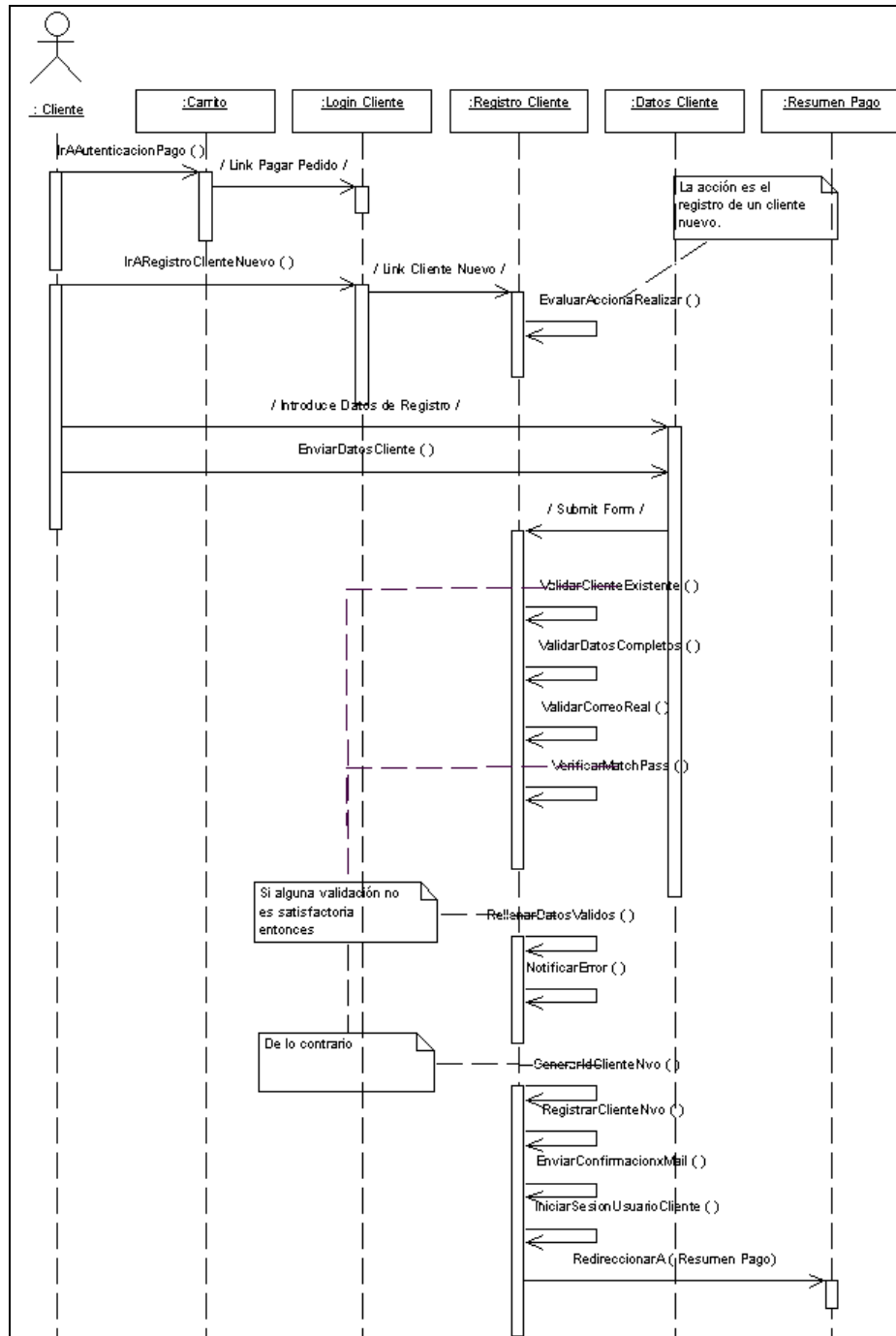


Figura 6.5: Secuencia Autenticación de un Cliente nuevo

6.2.5 Secuencia de ePlaza: Autenticación de un Cliente existente

Esta secuencia, que se muestra en la figura 6.6 describe, el proceso de autenticación de un Cliente previamente registrado. La secuencia inicia cuando el Cliente selecciona pasar a pagar el pedido realizado desde la <<navigation class>> *Carrito*. Ya estando el control en la <<navigation class>> *Login Cliente*, el Cliente introduce sus datos de registro, es decir su nombre de usuario que es su correo electrónico y su contraseña. El Cliente solicita el envío de esta información mediante la operación *EnviarDatosRegistro()* definida en el <<form>> *Datos Registro Cliente*. La información se envía a la <<navigation class>> *Autenticación Usuario*, en donde se evalúa la acción a realizar mediante la operación *EvaluarAccionaRealizar()*. En esta secuencia la operación a realizar es la autenticación de un usuario Cliente. Hecho esto, la <<navigation class>> *Autenticación Usuario* realiza la autenticación del usuario utilizando los datos proporcionados mediante la operación *RealizarAutenticacion()*, en donde se realiza una consulta a la fuente de datos de la aplicación. Esta operación devuelve una respuesta que se evalúa en la operación *EvaluarRespuesta()*. En base a esta evaluación la secuencia puede tomar dos distintos flujos alternativos. Si la respuesta no es favorable, es decir, que los datos de registro proporcionados por el Cliente no encuentran una correspondencia en las fuentes de datos entonces se debe notificar un error, finalizando de esta manera la secuencia. Por el contrario, la respuesta es favorable, entonces *Autenticación Usuario* realiza la operación *IniciarSesionCliente()* con la que se inicializan las variables de sesión con los datos de registro del Cliente, finalizando la secuencia con el redireccionamiento la navegación hacia la <<navigation class>> *Resumen Pago* mediante la operación *RedireccionarA()* para continuar con el proceso de pago.

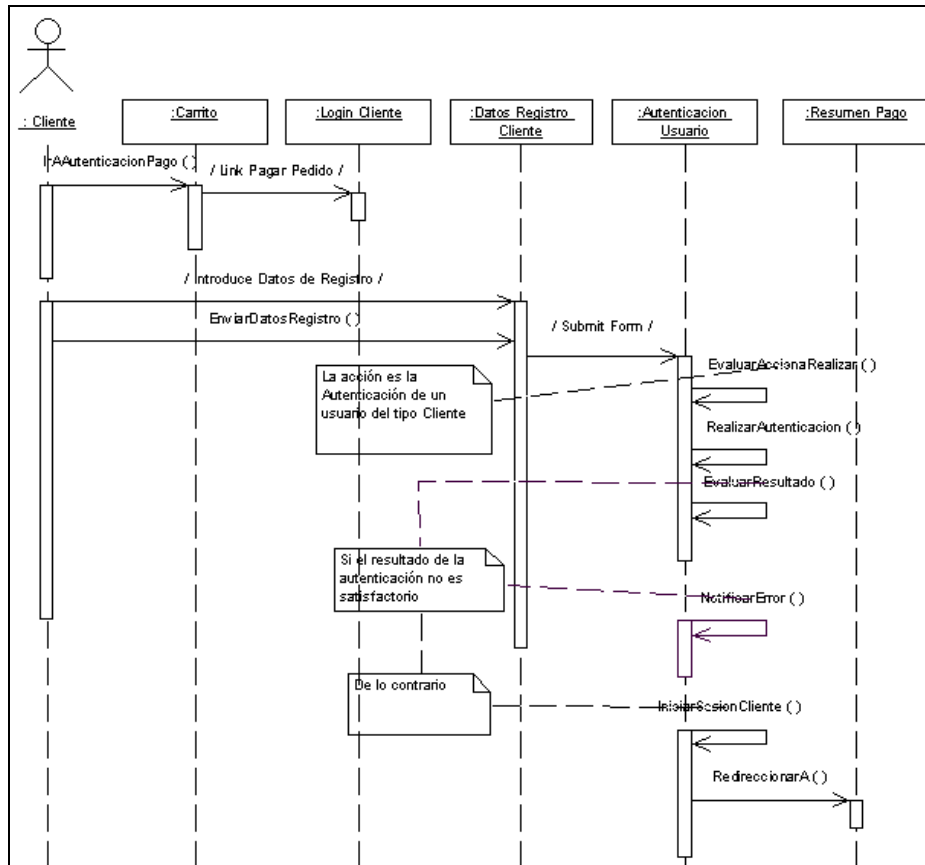


Figura 6.6: Secuencia Autenticación de un Cliente existente

6.2.6 Secuencia de Proceso de Pago: Resumen de pago, cambio en la dirección de envío

En la figura 6.7 se muestra la secuencia para la primera etapa del proceso de pago. En esta etapa se muestra al Cliente el resumen del pedido realizado y se requiere al Cliente la selección de un servicio de envío, además, se le proporciona la posibilidad de cambiar la dirección de envío. La secuencia inicia una vez que el Cliente se ha autenticado y accede a la <<navigation class>> *Resumen pago*. Esta <<navigation class>> valida la <<navigation class>> de procedencia mediante la operación *ValidarOrigen()*, esto como mecanismo de seguridad. Si el origen no es válido, entonces se trata de un intento de pago malintencionado, por lo que se notifica la ocurrencia del error, y la secuencia finaliza. Si el origen es válido, entonces se continúa con el proceso cargando los datos del Cliente, los datos del pedido, el detalle del pedido, los datos de la tienda y se carga la lista de servicios de envío disponibles. La secuencia continua con la selección por parte del Cliente de un servicio de envío adecuado para el pedido realizado en el <<form>> *Datos de Envío*. Los datos son enviados a la <<navigation class>> *Resumen Pago* para que el servicio de envío sea asociado al pedido y se agregue al total el costo del servicio seleccionado. La secuencia continua con la edición opcional de los datos de envío del Cliente. El Cliente accede a la <<navigation class>> *Registro Cliente* desde *Resumen Pago* indicando que la operación a realizar es la de modificación de los datos de envío del Cliente. Cuando el control llega a *Registro cliente*, se evalúa la acción y de acuerdo a esta, se cargan los datos actuales del Cliente para que puedan ser modificados. El Cliente introduce la nueva información a través del <<form>> *Datos Cliente*. Esta información se reenvía a la <<navigation class>> *Registro Cliente* en donde se actualiza el registro del Cliente con los nuevos datos de envío. Hecho esto, *Registro Cliente* redirecciona el control de navegación a la <<navigation class>> *Resumen Pago* para continuar con el proceso de pago. Una vez de vuelta en *Resumen Pago*, el Cliente debe continuar con el procedimiento navegando hacia la <<navigation class>> *Pago de Productos*.

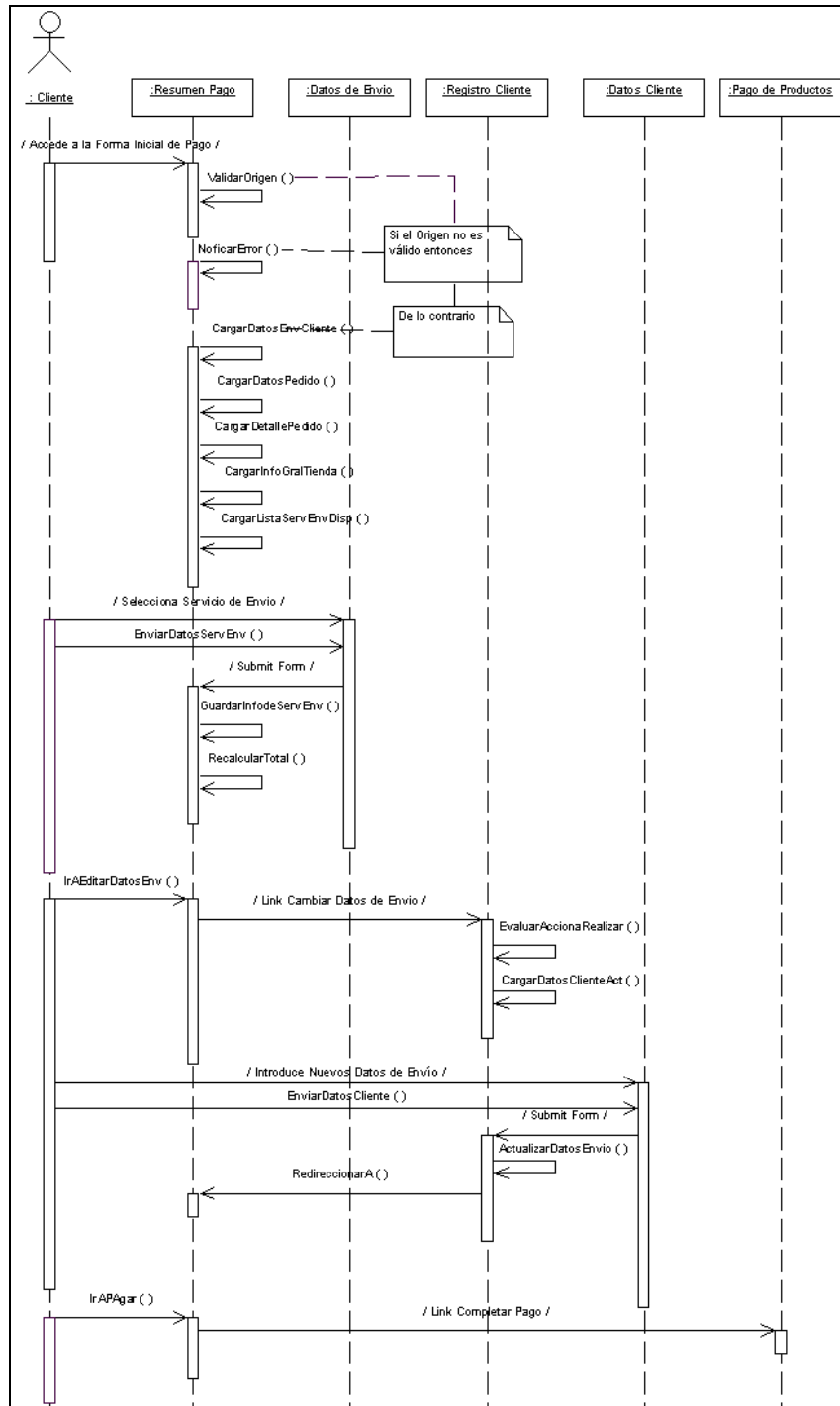


Figura 6.7: Secuencia Resumen de pago, cambio en la dirección de envío

6.2.7 Secuencia de Pago: Cambios en datos de facturación y captura de datos de la tarjeta

En la figura 6.8 se muestra la secuencia para la segunda etapa del proceso de pago. En esta etapa se muestra al Cliente el resumen del pedido realizado y se le proporciona al Cliente la posibilidad de cambiar los datos de facturación. La secuencia inicia una vez que el Cliente navega y accede a la <<navigation class>> *Pago de Productos*. Esta <<navigation class>> valida la <<navigation class>> de procedencia mediante la operación *ValidarOrigen()*, esto como mecanismo de seguridad. Si el origen no es válido, entonces se trata de un intento de pago malintencionado, por lo que se notifica la ocurrencia del error, y la secuencia finaliza. Si el origen es válido, entonces se continúa con el proceso cargando los datos de envío del Cliente, los datos de facturación actuales del Cliente, el detalle del pedido y los datos del servicio de envío seleccionado en la <<navigation class>> *Resumen Pago*. La secuencia continúa con la edición opcional de los datos de facturación del Cliente. El Cliente accede a la <<navigation class>> *Registro Cliente* desde *Pago de Productos* indicando que la operación a realizar es la de modificación de los datos de facturación del Cliente. Cuando el control llega a *Registro cliente*, se evalúa la acción y de acuerdo a esta, se cargan los datos actuales del Cliente para que puedan ser modificados. El Cliente introduce la nueva información de facturación a través del <<form>> *Datos Cliente*. Esta información se reenvía a la <<navigation class>> *Registro Cliente* en donde se actualiza el registro del Cliente con los nuevos datos de facturación. Una vez realizada la modificación, *Registro Cliente* redirecciona el control de navegación a la <<navigation class>> *Pago de Productos* para continuar con el proceso de pago. Una vez de vuelta en *Pago de Productos*, el Cliente debe continuar con el procedimiento de pago introduciendo la información de la tarjeta bancaria sobre la cual se debe cargar el monto del pedido. El Cliente introduce esta información mediante el <<form>> *Info Pago Prod*. La información se envía en modo seguro a la <<navigation class>> *Proceso Pago Seguro* del subsistema GTr para que sea este elemento quien gestione la transacción con la terminal Bancaria.

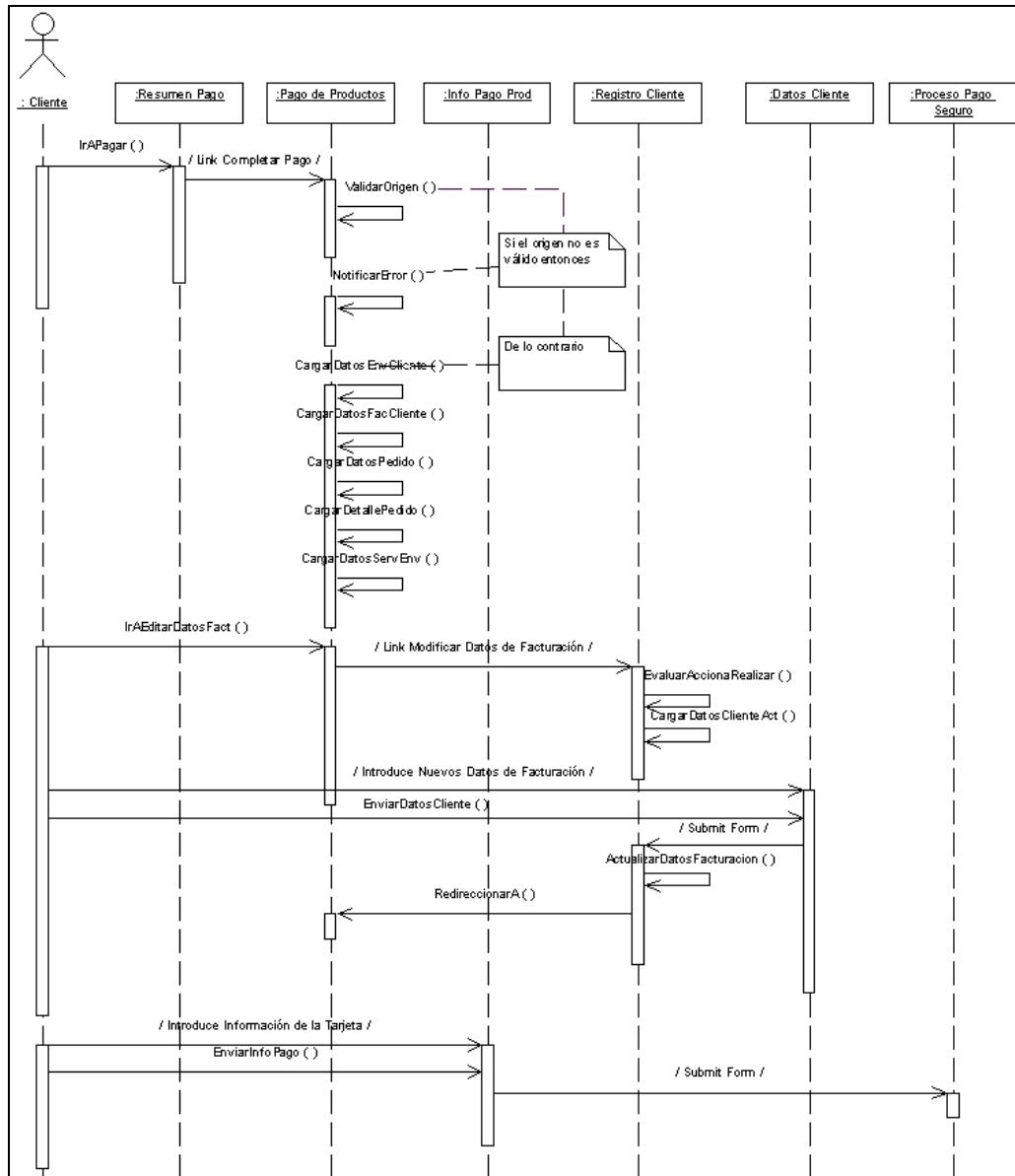


Figura 6.8: Secuencia Cambios en datos de facturación y captura de datos de la tarjeta

6.2.8 Secuencia de Gestor de Transacciones: Procesamiento de Pago Seguro de Productos

Con esta secuencia, que se muestra en la figura 6.9, se complementa el proceso del pago del pedido de un Cliente. Esta secuencia comienza cuando el <<form>> *Info Pago Prod*, del subsistema ePlaza, remite la información de la tarjeta bancaria del Cliente hacia la <<navigation class>> *Proceso Pago Seguro* del subsistema Gestor de Transacciones. Cuando la información es recibida por la <<navigation class>> *Proceso Pago Seguro*, lo primero que se realiza es la verificación acerca del origen de la información. Esto es un mecanismo de seguridad para tener certidumbre que información bancaria recibida proviene del subsistema ePlaza y no de un origen distinto malintencionado. Si el origen no es válido, entonces se notifica que un error ha ocurrido y se termina la secuencia. En el caso esperado de que el origen sea válido, entonces la secuencia continúa evaluando la acción a realizar, esto ya que mediante esta misma <<navigation class>> se procesa el pago de comisiones. Una vez que se evalúa la acción y se determina que es el pago de un pedido por parte de un Cliente, *Proceso Pago Seguro* carga los datos de la cuenta de banco de la tienda que esta efectuando la venta. La secuencia continua con las operaciones *CrearInstancia_e24TranPipe()*, *Preparar_e24TranPipe()* y *ExecTrans_e24TranPipe()*. Con estas operaciones se crea una instancia del objeto COM *e24TranPipe* que es el encargado de realizar la transacción con la terminal bancaria. Una vez creado, se prepara dicho objeto con los datos de la cuenta de banco de la Tienda y con el monto de la operación. *ExecTrans_e24TranPipe()* ejecuta la transacción construyéndola en el formato requerido. Esta cadena es la que se envía a través del canal seguro mediante la operación *Perform_Transaction()* del *e24TranPipe*. Con esta función el objeto *e24TranPipe* se comunica con la terminal bancaria y espera una cadena de respuesta. Cuando esta respuesta llega, *e24TranPipe* la notifica a *Proceso de Pago Seguro*, en donde se evalúa dicha cadena de respuesta para verificar si la transacción fue realizada con éxito o no. Dependiendo de esta respuesta se registra el error de pago o el pago exitoso. Independientemente de la respuesta, la secuencia continúa con el registro de la operación realizada, se guarda el movimiento en la bitácora, guarda la información de la transacción y crea la cadena que se envía como respuesta de la transacción al subsistema ePlaza. La secuencia finaliza cuando la navegación retorna al subsistema ePlaza mediante el redireccionamiento hacia la <<navigation class>> *Confirm Pago Prod* o la <<navigation class>> *Error Pago Prod* dependiendo si la transacción fue o no exitosa. En el redireccionamiento se incluye la cadena de de respuesta que fue creada con el resultado de la operación.

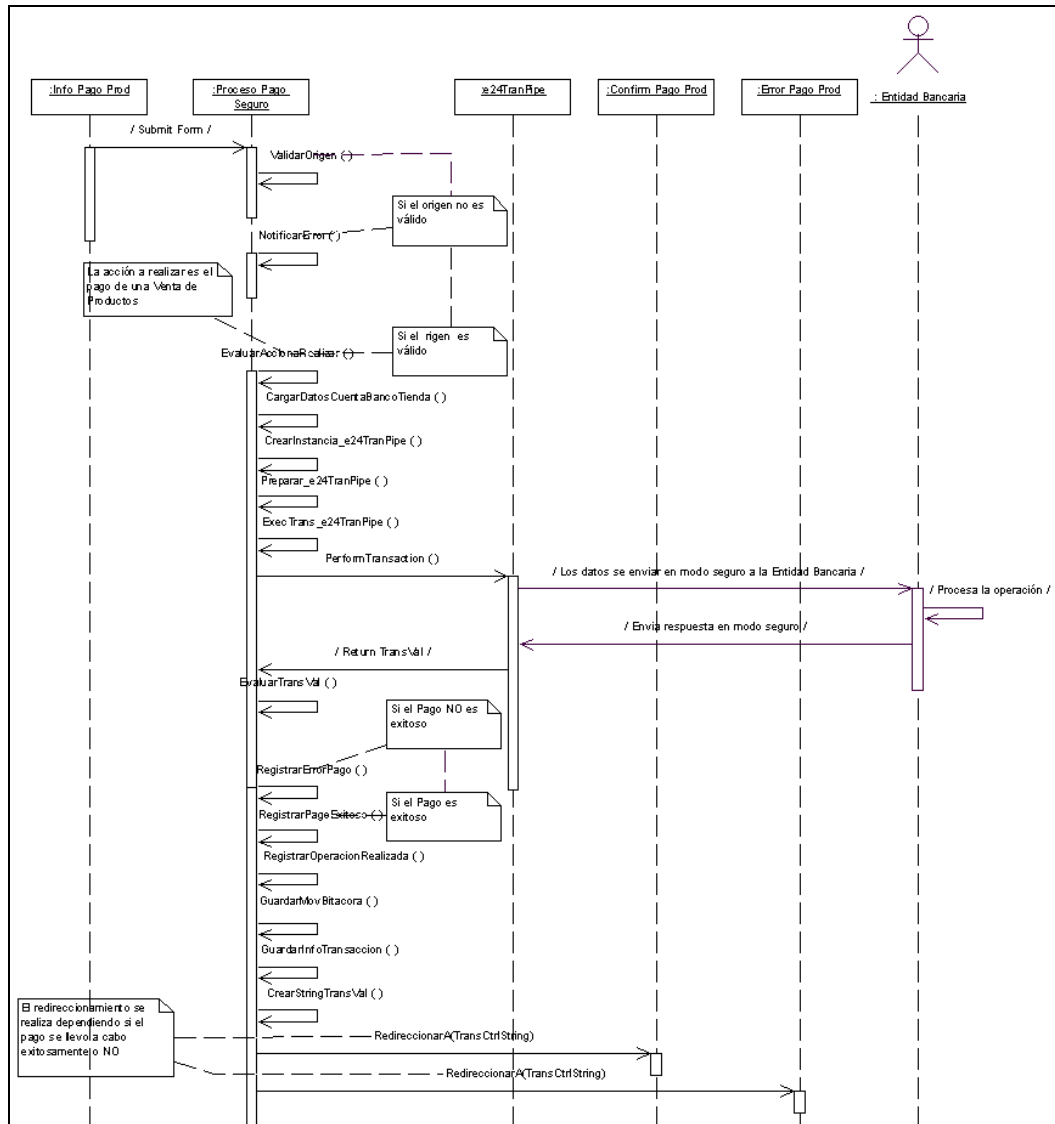


Figura 6.9: Secuencia Procesamiento de Pago Seguro de Productos

6.2.9 Secuencia de ePlaza: Autenticación de Usuario Vendedor

Esta secuencia, que se muestra en la figura 6.10, describe el proceso de autenticación de un usuario Vendedor para acceder al subsistema Gestor de Tienda. La secuencia inicia cuando el usuario Vendedor accede a la <<navigation class>> *Home CCV* y a partir de ella navega hacia la <<navigation class>> *Login Vendedor*. Una vez en la <<navigation class>> *Login Vendedor*, en el <<form>> *Datos Registro Vendedor* se carga la lista de tiendas que ya se encuentran asociadas al CCV. El Vendedor debe introducir sus datos de acceso que le fueron proporcionados por el Administrador del CCV. El Vendedor introduce estos datos por medio del <<form>> *Datos Registro Vendedor*, dentro de los cuales se incluye la selección de la Tienda asociada a la cual el Vendedor desea acceder, el nombre de usuario y la contraseña correspondiente. Esta información es enviada hacia la <<navigation class>> *Autenticación Usuario*. Cuando en *Autenticación Usuario* se recibe esta información, evalúa la acción a realizar, en este caso se trata de la autenticación de un usuario del tipo Vendedor. Hecho esto la <<navigation class>> *Autenticación Usuario* realiza la autenticación del usuario Vendedor utilizando los datos proporcionados mediante la operación *RealizarAutenticacion()*, en donde se realiza una consulta a la fuente de datos de la aplicación. Esta operación devuelve una respuesta que se evalúa en la operación *EvaluarRespuesta()*. En base a esta evaluación la secuencia puede tomar dos distintos flujos alternativos. Si la respuesta no es favorable, es decir, que los datos de registro proporcionados por el Vendedor no encuentran una correspondencia en las fuentes de datos entonces se debe notificar un error, finalizando de esta manera la secuencia. Por el contrario, si la respuesta es favorable, entonces *Autenticación Usuario* realiza la operación *IniciarSesionVendedor()* con la que se inicializan las variables de sesión con los datos de registro del usuario Vendedor, finalizando la secuencia redireccionando la navegación hacia la <<navigation class>> *Home Gestor Tienda* mediante la operación *RedireccionarA()*, en donde se proporcionan al Vendedor diversas opciones para la administración de su tienda en el CCV.

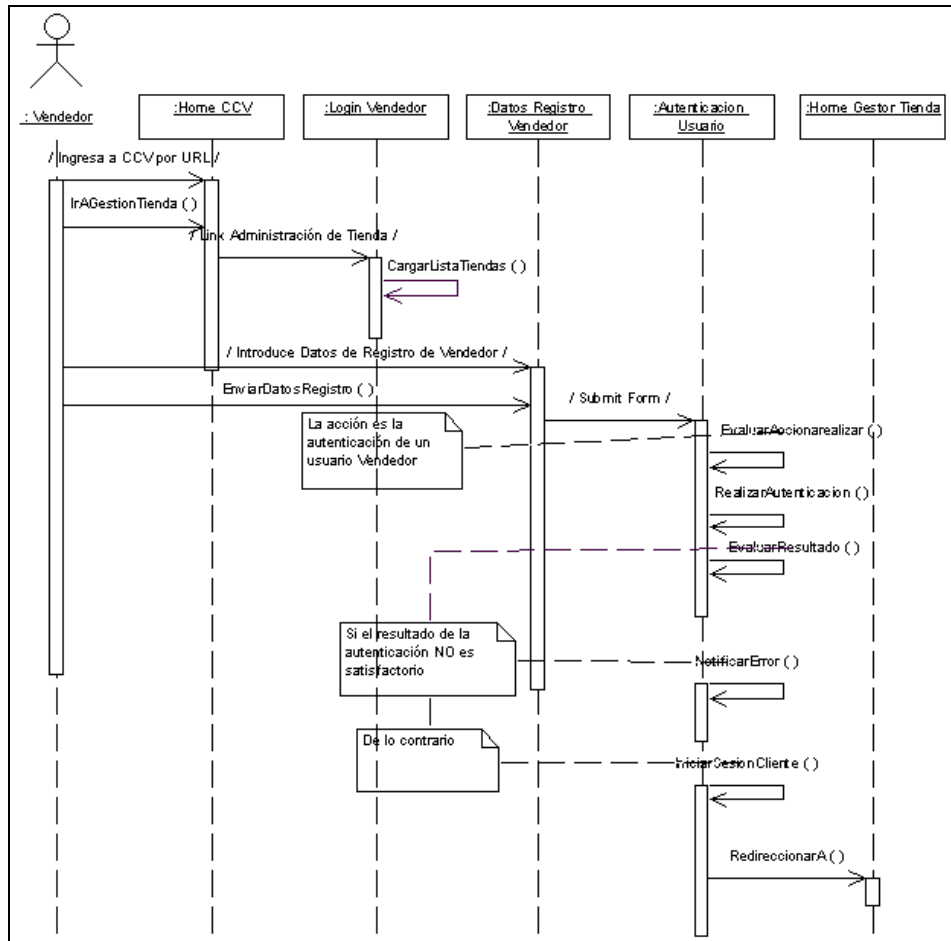


Figura 6.10: Secuencia Autenticación de usuario Vendedor

6.2.10 Secuencia de Gestor de Tienda: Modificación de Configuración de Tienda

Esta secuencia, que se muestra en la figura 6.11, describe la manera de cómo un usuario Vendedor puede modificar la configuración de la tienda asociada al CCV. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda* una vez que se ha autenticado. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Modificar Conf Tienda*. En esta <<navigation class>> se encuentran asociados los elementos requeridos para llevar a cabo la modificación de la configuración. Una vez en *Modificar Conf Tienda* se carga la configuración actual de la Tienda en los campos correspondientes del <<form>> *Datos Conf Tienda*. Sobre este <<form>> el Vendedor introduce los nuevos valores de la configuración, tales como la combinación de colores, estructura de la tienda, textos personalizados, entre otros. Entre estos valores de configuración existen algunos campos que requieren la localización de un archivo de imagen en disco. Para ello, de manera opcional, el Vendedor puede invocar al explorador de archivo mediante la operación *CambiarRutaArchivo()* para explorar el disco y localizar el nuevo archivo. De esta forma, el control de la secuencia lo toma el <<component>> *Explorador Archivos*. El Vendedor entonces examina el disco y selecciona el archivo deseado, esto mediante las operaciones *Explorar()* y *Seleccionar()* del *Explorador Archivos*. Una vez que el Vendedor selecciona el archivo, entonces el <<component>> *Explorador Archivos* deposita el path del archivo seleccionado al <<form>> *Datos Conf Tienda*. Ahora, la secuencia puede tener tres flujos alternativos, ya que el Vendedor puede elegir entre previsualizar los cambios realizados, guardar los cambios realizados o bien ignorar los cambios realizados a la configuración. Si ejecuta la operación *Previsualizar()* del <<form>> *Datos Conf Tienda*, entonces los datos del formulario se reenvían a la <<navigation class>> *Modificar Conf Tienda* en donde se recuperan estos datos y se presenta al Vendedor la previsualización de los cambios realizados a la configuración. La secuencia finaliza con esta previsualización, pero cabe señalar que en la implementación se sugiere que el control de navegación permanezca en la <<navigation class>> *Modificar Conf Tienda*, de tal forma que el Vendedor, después de previsualizar los cambios, pueda guardar la configuración, ignorar los cambios o modificar de nuevo los datos de configuración. Si el Vendedor decide guardar la nueva configuración, entonces se dispara la operación *EnviarDatosConfTienda()* del <<form>> *Datos Conf Tienda*. Entonces los datos del formulario se reenvían a la <<navigation class>> *Modificar Conf Tienda* y se guarda en la fuente de datos la nueva información de la configuración de la tienda. Al igual que en la previsualización, la secuencia finaliza con la actualización de la configuración pero cabe señalar que en la implementación se sugiere que el control de navegación permanezca en la <<navigation class>> *Modificar Conf Tienda*, de tal forma que el Vendedor después de haber guardado los cambios pueda modificar de nuevo los datos de configuración. Por último, si el Vendedor desea descartar los cambios que haya realizado sobre los campos del <<form>> *Datos Conf Tienda*, entonces se dispara la operación *IgnorarCambios()* y la configuración original es restablecida.

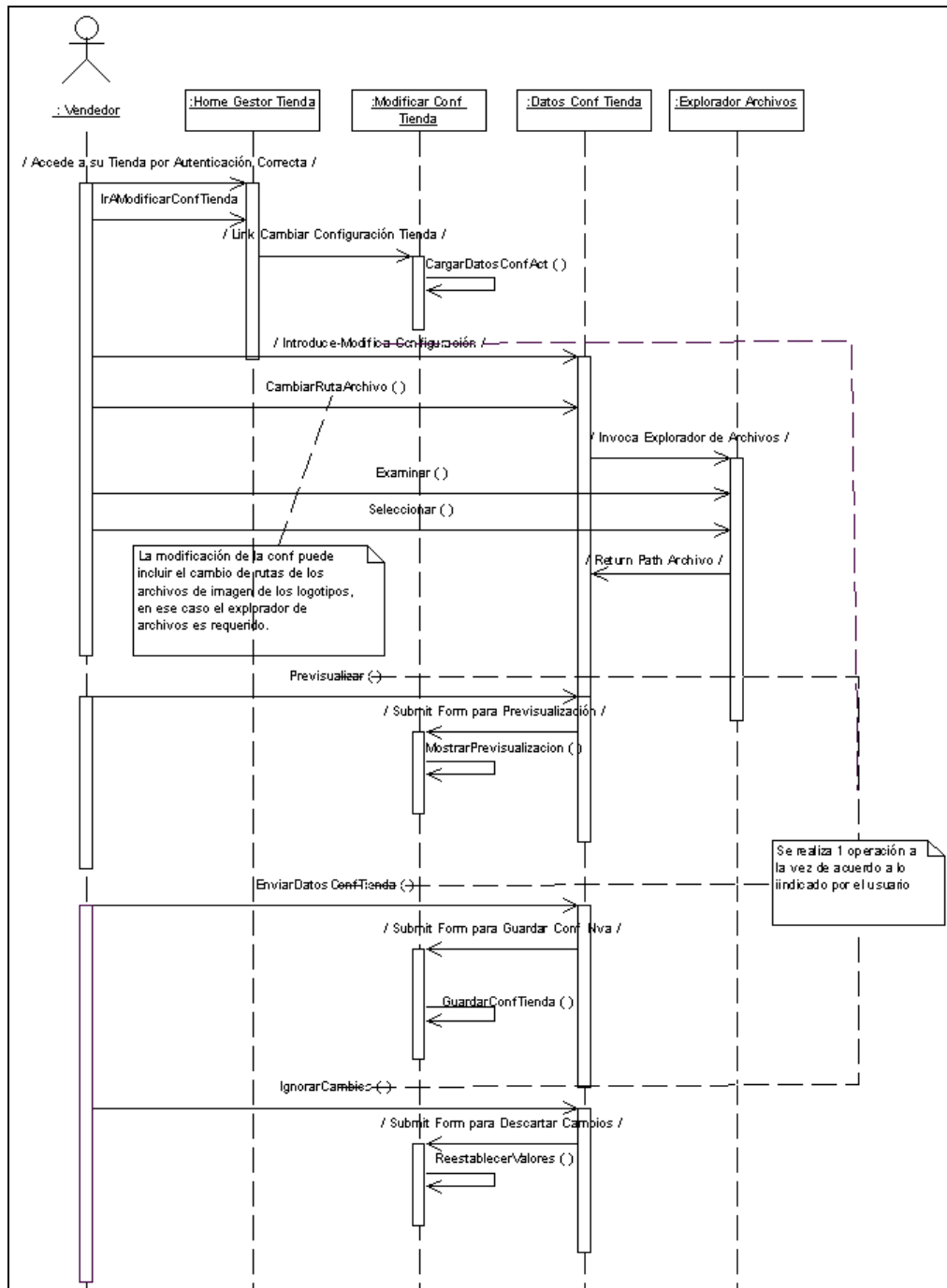


Figura 6.11: Secuencia Modificación de Configuración de Tienda

6.2.11 Secuencia de Gestor de Tienda: Adición de una nueva categoría al catálogo

Esta secuencia, que se muestra en la figura 6.12, describe el procedimiento de adición de una nueva categoría al catálogo de productos de la tienda por parte del usuario Vendedor, administrador de dicha tienda. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda*, una vez que se ha autenticado de manera correcta como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Catálogos*. En esta <<navigation class>> se carga la lista del catálogo con todas las categorías actuales de la tienda. Para agregar una nueva categoría, el usuario Vendedor dispara la operación *IraRegistroCategoriaNueva()* para navegar hacia la <<navigation class>> *Registro Categoría*. Cuando *Registro Categoría* se carga, se evalúa la acción a realizar que, en este caso, es la agregación de una nueva categoría al catálogo. El usuario Vendedor, a través del <<form>> *Datos Categoría*, introduce la información de la nueva categoría a agregar y dispara la operación *EnviarDatosCategoría()*. Esta información se reenvía hacia la <<navigation class>> *Registro Categoría* en donde se valida que dicha información sea la apropiada y, en base a esta evaluación, la secuencia prosigue en cualquiera de dos flujos alternativos. Si la validación no es satisfactoria entonces se notifica el error y la secuencia termina. Por el contrario, si la validación de la información es correcta, la secuencia continua en la <<navigation class>> *Registro Categoría* con la operación *GenerarIDCategoria()*, con la cual se genera el identificador único de la nueva categoría a registrar y por último, la secuencia concluye con la operación *RegistrarCatNueva()* con la que se efectúa el registro de la nueva categoría en la fuente de datos del CCV. La categoría recién creada no contiene ningún producto; los productos se agregarán a la categoría cuando se lleve a cabo el registro del producto.

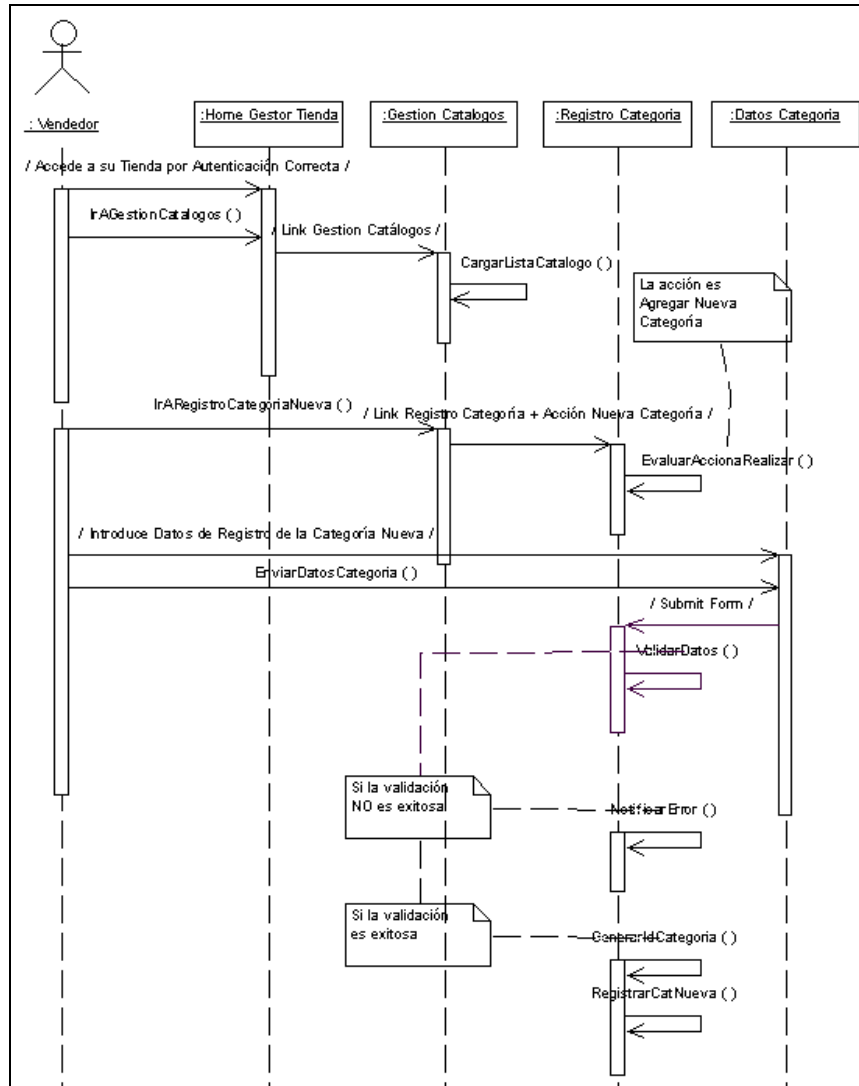


Figura 6.12: Secuencia Adición de una nueva categoría al catálogo

6.2.12 Secuencia de Gestor de Tienda: Edición de una categoría existente

Esta secuencia, que se muestra en la figura 6.13, describe la edición de una categoría ya existente en el catálogo de productos por parte del usuario Vendedor de una tienda. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda*, una vez que se ha autenticado de manera correcta como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Catálogos*. En esta <<navigation class>> se carga la lista del catálogo con todas las categorías actuales de la tienda. Este listado es el <<index>> *Catálogo*. De este <<index>> de categorías, el Vendedor selecciona a una de ellas para editarla mediante la operación *Editar()* asociada a dicha categoría. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Categoría*. Cuando *Registro Categoría* se carga se evalúa la acción a realizar que, en este caso, es la edición de una categoría ya existente en el catálogo. Subsecuentemente, se cargan los datos actuales de la categoría a editar en el <<form>> *Datos Categoría*. Una vez que los datos de la categoría aparecen cargados en el <<form>>, el usuario Vendedor modifica la información de la categoría y, una vez hecho los cambios deseados, dispara la operación *EnviarDatosCategoría()*. Esta información se reenvía hacia la <<navigation class>> *Registro Categoría* en donde se valida que dicha información sea apropiada y, en base a esta evaluación, la secuencia prosigue en uno de dos flujos alternativos. Si la validación no es satisfactoria entonces se notifica el error y la secuencia termina. Por el contrario, si la validación de la información es correcta, la secuencia concluye con la operación *ActualizarCat()* de la <<navigation class>> *Registro Categoría*, con la cual se lleva a cabo la actualización del registro en la fuente de datos de la categoría editada. Cabe sugerir que en la implementación, si una categoría tiene productos asociados a ella, la edición de la misma debe incluir la persistencia de este vínculo aun cuando la categoría sea editada, o bien únicamente se permita editar categorías vacías.

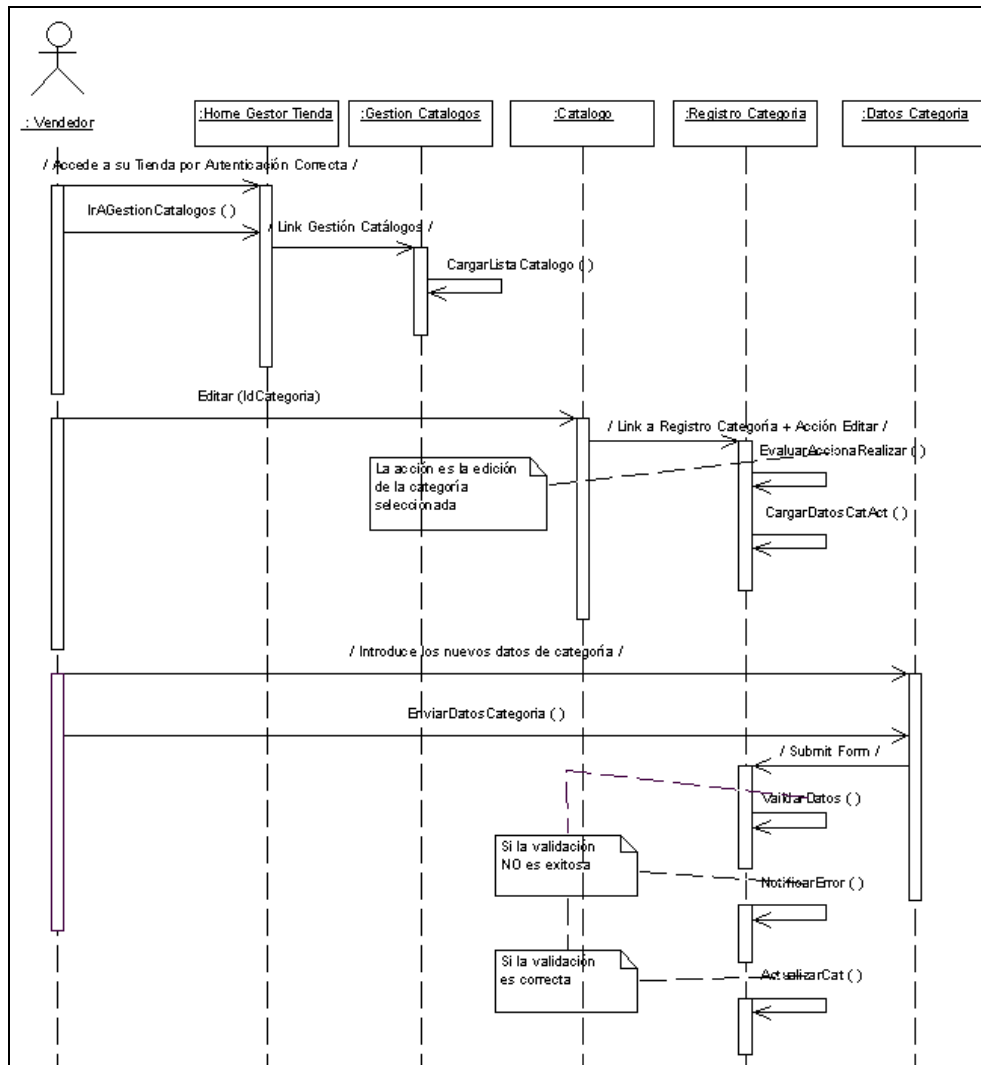


Figura 6.13: Secuencia Edición de una categoría existente

6.2.13 Secuencia de Gestor de Tienda: Eliminación de una categoría existente

La eliminación de una categoría existente en el catálogo de productos parte del usuario Vendedor se describe en esta secuencia. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda* una vez que se ha autenticado de manera correcta como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Catálogos*. En esta <<navigation class>> se carga la lista del catálogo con todas las categorías actuales de la tienda. Este listado es el <<index>> *Catálogo*. De este <<index>> de categorías, el Vendedor selecciona a una de ellas para eliminarla mediante la operación *Eliminar()* asociada a dicha categoría. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Categoría*. Cuando *Registro Categoría* se carga, se evalúa la acción a realizar que, en este caso, es la eliminación de una categoría ya existente en el catálogo. La secuencia concluye con la operación *EliminarCat()* de la <<navigation class>> *Registro Categoría*, con la cual se lleva a cabo la eliminación (marcado) del registro en la fuente de datos de la categoría editada. Cabe sugerir que en la implementación, si una categoría tiene productos asociados a ella, la eliminación de la misma no debe ser posible, o bien, los productos asociados a dicha categoría deben moverse a una categoría general preexistente que no pueda ser eliminada.

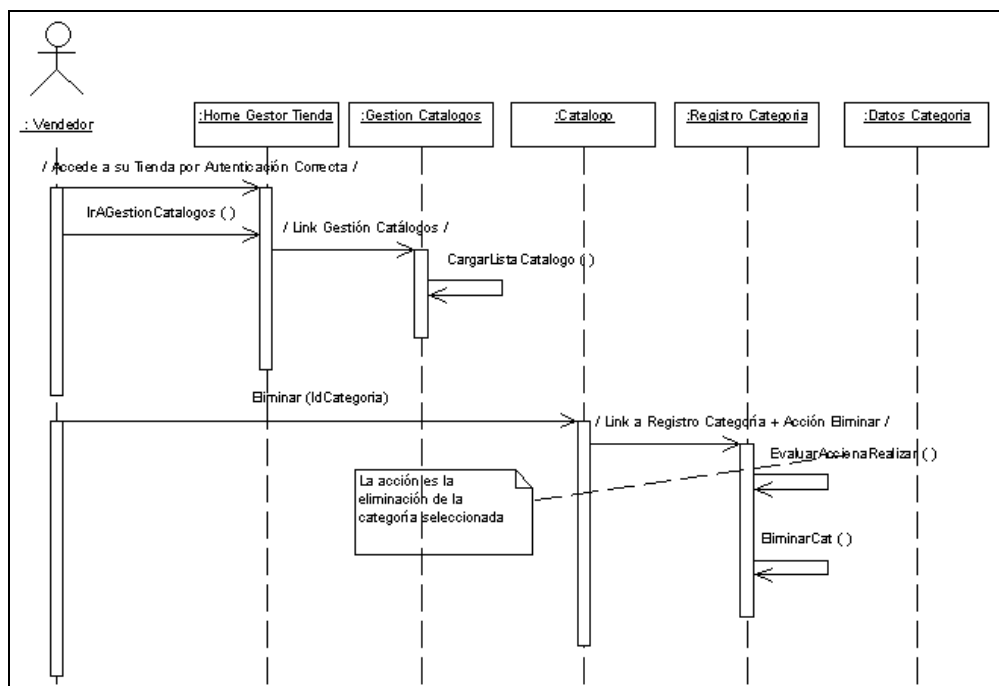


Figura 6.14: Secuencia Eliminación de una categoría existente

6.2.14 Secuencia de Gestor de Tienda: Adición de un producto nuevo al catálogo

Esta secuencia, que se muestra en la figura 6.15, describe el procedimiento de adición de un nuevo producto al catálogo de productos de la tienda por parte del usuario Vendedor, administrador de dicha tienda. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda* una vez que se ha autenticado de manera correcta como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Catálogos*. En esta <<navigation class>> se carga la lista del catálogo con todas las categorías actuales de la tienda. Para agregar un nuevo producto, el usuario Vendedor dispara la operación *IraRegistroProductoNuevo()* para navegar hacia la <<navigation class>> *Registro Producto*. Cuando *Registro Producto* se carga, evalúa la acción a realizar que, en este caso, es la agregación de un nuevo producto al catálogo. El usuario Vendedor, a través de los campos definidos en el <<form>> *Datos Producto*, introduce los datos del nuevo producto a agregar. Entre los datos requeridos para el registro se encuentra la descripción del producto, el precio, especificaciones, categoría a la cual se va a asociar, fecha de publicación, archivo de imagen asociado, entre otras. En la introducción de esta información se definen dos secuencias opcionales. La primera de ellas es para utilizar el <<component>> *Explorador Archivos*. Por medio de este, el usuario Vendedor obtiene el path del archivo de la imagen del producto. Una vez que ha explorado el disco y seleccionado el archivo, el path es depositado en el <<form>> *Datos Producto*. La segunda secuencia opcional describe el uso del <<component>> *Selector Fecha* para obtener la fecha de publicación y la fecha de registro. Por medio de este <<component>>, el usuario selecciona la fecha de publicación de un calendario, la cual, se deposita en un formato estándar en el <<form>> *Datos Producto*. Una tercera secuencia opcional, es la previsualización del registro del producto. Esta secuencia se lleva a cabo cuando el Vendedor requiere esta previsualización mediante la operación *Previsualizar()* del <<form>> *Datos Producto*. Los datos de registro se envían hacia la <<navigation class>> *Registro Producto* en donde se utilizan para mostrar la previsualización. La secuencia del registro continúa, en su flujo básico, cuando el usuario Vendedor dispara la operación *EnviarDatosCategoría()* el <<form>> *Datos Producto*. Esta información es enviada hacia la <<navigation class>> *Registro Producto* en donde se valida que dicha información sea correcta. Si la validación no es satisfactoria entonces se notifica el error y la secuencia termina. Por el contrario, si la validación de la información es correcta, la secuencia continua en la <<navigation class>> *Registro Producto* con las operaciones *SubirImagen()* y *GenerarImagenCh()*. Con estas operaciones se suben los archivos de imagen especificados y se genera la imagen pequeña de tamaño uniforme que se utiliza en la descripción del producto. La secuencia continua con la operación *GenerarIdProductoNvo()* con la cual se genera el identificador único del nuevo producto a registrar y por último, la secuencia concluye con la operación *RegistrarProductoNvo()* con la que se efectúa el registro de la nueva categoría en la fuente de datos del CCV.

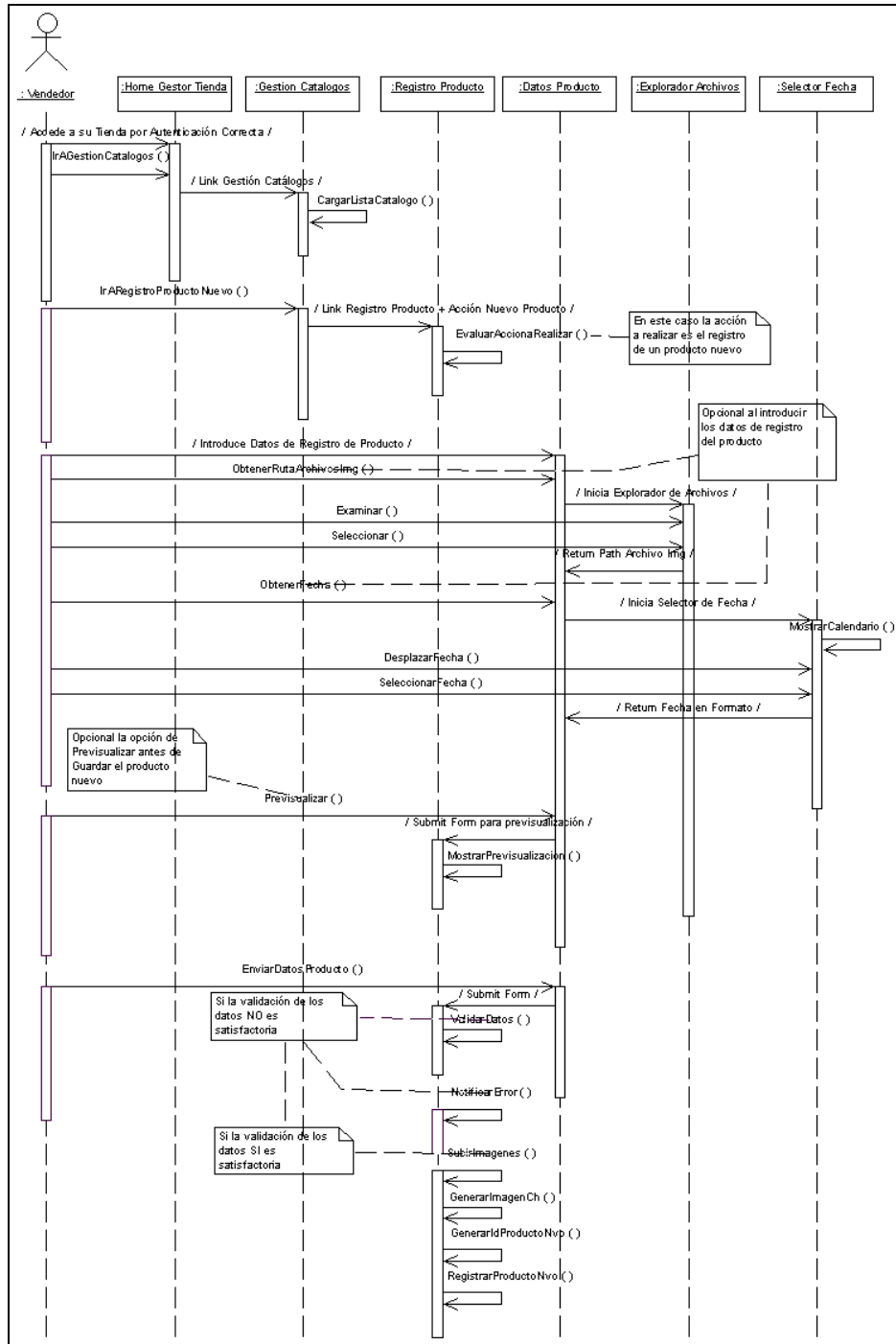


Figura 6.15: Secuencia Adición de un producto nuevo al catálogo

6.2.15 Secuencia de Gestor de Tienda: Edición de un producto existente

Esta secuencia describe el procedimiento de edición de un producto ya registrado en el catálogo de productos de la tienda. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda* una vez que se ha autenticado de manera correcta como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Catálogos*. En esta <<navigation class>> se carga el <<index>> *Catálogo*, que es un listado de todas las categorías existentes en la tienda. De este listado, el usuario Vendedor selecciona, mediante la operación *VerProductosdeCategoria()*, a una de ellas para ver el listado de productos que contiene. Al hacer esto se carga el <<index>> *Catagorias*. Este <<index>> es un listado de los productos que contiene la categoría seleccionada. De este listado el usuario Vendedor selecciona, mediante la operación *Editar()* asociada a dicha categoría, al producto el cual va a ser editado. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Producto*. Cuando *Registro Producto* se carga se evalúa la acción a realizar que, en este caso, es la edición de un producto ya existente en el catálogo. Subsecuentemente, se cargan los datos actuales del producto a editar en el <<form>> *Datos Producto*. Una vez que los datos del producto aparecen cargados en el <<form>>, el usuario Vendedor realiza los cambios deseados en el registro del producto. En la introducción de los nuevos datos, al igual que cuando se dio de alta al producto, se tienen la posibilidad de utilizar a los <<component>> *Explorador Archivos* y *Selector Fecha*. La secuencia de utilización de estos elementos es la misma que se describió en la secuencia de adición de un producto nuevo. En la edición de un producto existente también se incluye la secuencia opcional que contempla la posibilidad que el usuario Vendedor pueda visualizar el registro del producto antes de guardar los cambios. Una vez que ha introducido los nuevos datos, el Vendedor dispara la operación *EnviarDatosProducto()* del <<form>> *Datos Producto*. Esta información se envía hacia la <<navigation class>> *Registro Producto* en donde se valida que dicha información sea apropiada y, en base a esta evaluación, la secuencia prosigue en uno de dos flujos alternativos. Si la validación no es satisfactoria entonces se notifica el error y la secuencia termina. Por el contrario, si la validación de la información es correcta, la secuencia continua en la <<navigation class>> *Registro Producto* con las operaciones *SubirImagen()* y *GenerarImagenCh()*. Con estas operaciones se vuelven a subir los archivos de imagen especificados y se genera la imagen pequeña de tamaño uniforme que se utiliza en la descripción del producto. La secuencia concluye con la operación *ActualizarProducto()* con la que se efectúa la actualización del registro del producto en la fuente de datos del CCV.

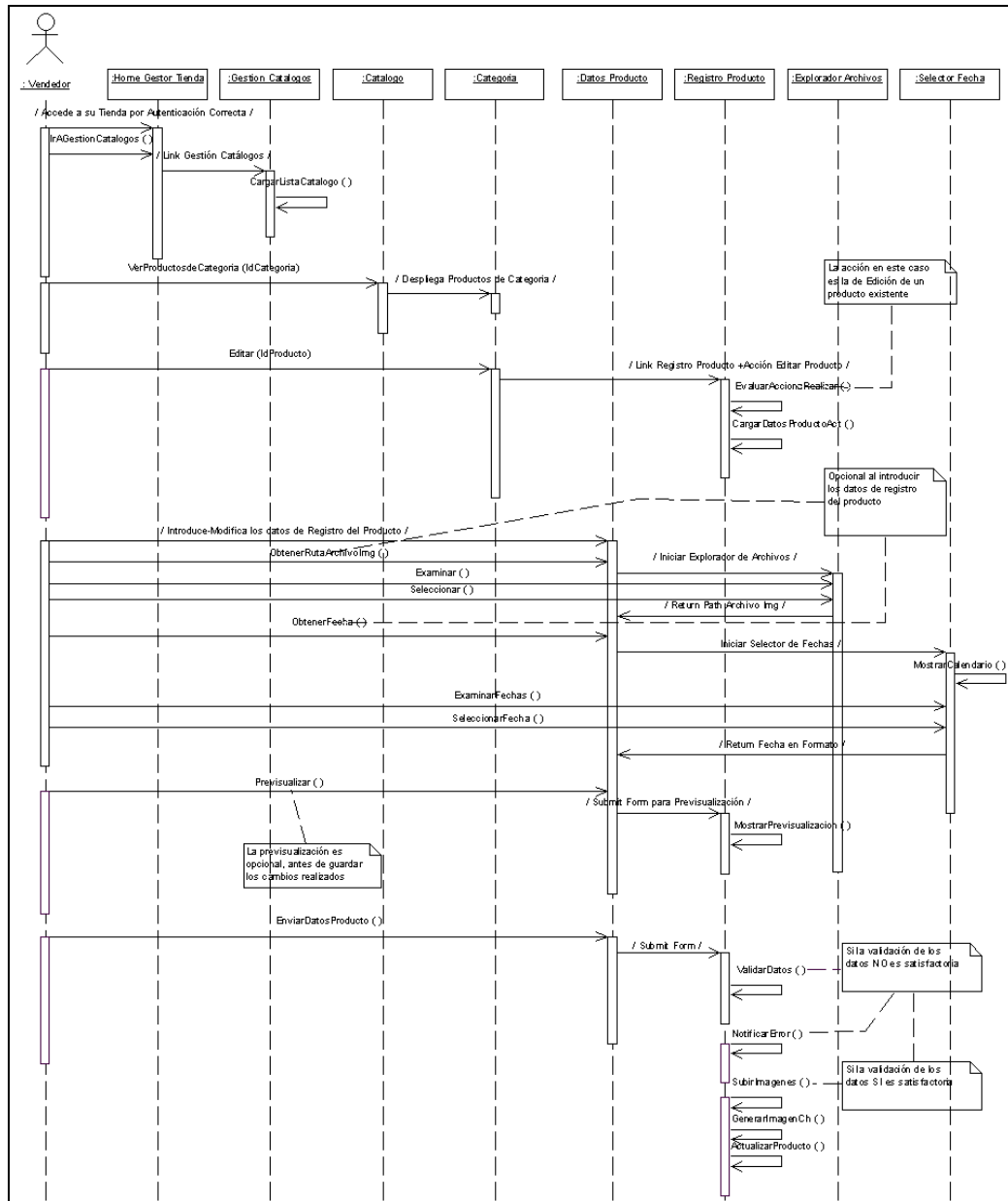


Figura 6.16 Secuencia Edición de un producto existente

6.2.16 Secuencia de Gestor de Tienda: Copiado/Movimiento de un producto existente a otra categoría

Esta secuencia para el procedimiento del copiado o cortado de un producto de una categoría a otra dentro del catálogo de la tienda asociada al CCV se describe en la figura 6.17. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda* una vez que se ha autenticado de manera correcta como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Catálogos*. En esta <<navigation class>> se carga el <<index>> *Catálogo*, que es un listado de todas las categorías existentes en la tienda. De este listado, el usuario Vendedor selecciona, mediante la operación *VerProductosdeCategoría()*, a una de ellas para ver el listado de productos que contiene. Al hacer esto se carga el <<index>> *Categorías*. Este <<index>> es un listado de los productos que contiene la categoría seleccionada. Dependiendo de la selección del usuario Vendedor la secuencia continua con uno de dos flujos alternativos. El primero de estos flujos se inicia cuando el usuario Vendedor selecciona la opción *CopiarA()* asociada al producto del listado que desea que sea copiado a otra categoría. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Producto*. Cuando *Registro Producto* se carga, se evalúa la acción a realizar que, en este caso, es el copiado de un producto de una categoría a otra. La secuencia continua con la introducción de la categoría destino a la cual va a ser copiado el producto. Esto lo realiza el usuario mediante el <<form>> *CopiarA*. Hecho esto, el Vendedor dispara la operación *EnviarDatoCopia()* del <<form>> *CopiarA()*. Esta información se envía hacia la <<navigation class>> *Registro Producto* en donde se recupera la información contenida en el <<form>> y se realiza el copiado del producto con la operación *CopiarProductoA()*, finalizando la secuencia de copiado. El segundo flujo alternativo se inicia cuando el usuario Vendedor, en vez de seleccionar la opción *CopiarA()*, selecciona la opción *MoverA()* asociada al producto del listado que desea que sea movido a otra categoría. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Producto*. Cuando *Registro Producto* se carga se evalúa la acción a realizar que, en este caso, es el movimiento de un producto de una categoría a otra. La secuencia continua con la introducción de la categoría destino a la cual va a ser movido el producto. Esto lo realiza el usuario mediante el <<form>> *MoverA*. Hecho esto, el Vendedor dispara la operación *EnviarDatoMov()* del <<form>> *MoverA*. Esta información se envía hacia la <<navigation class>> *Registro Producto* en donde se recupera la información contenida en el <<form>> y se realiza el movimiento del producto de una categoría a otra mediante la operación *MoverProductoA()*, finalizando la secuencia del movimiento.

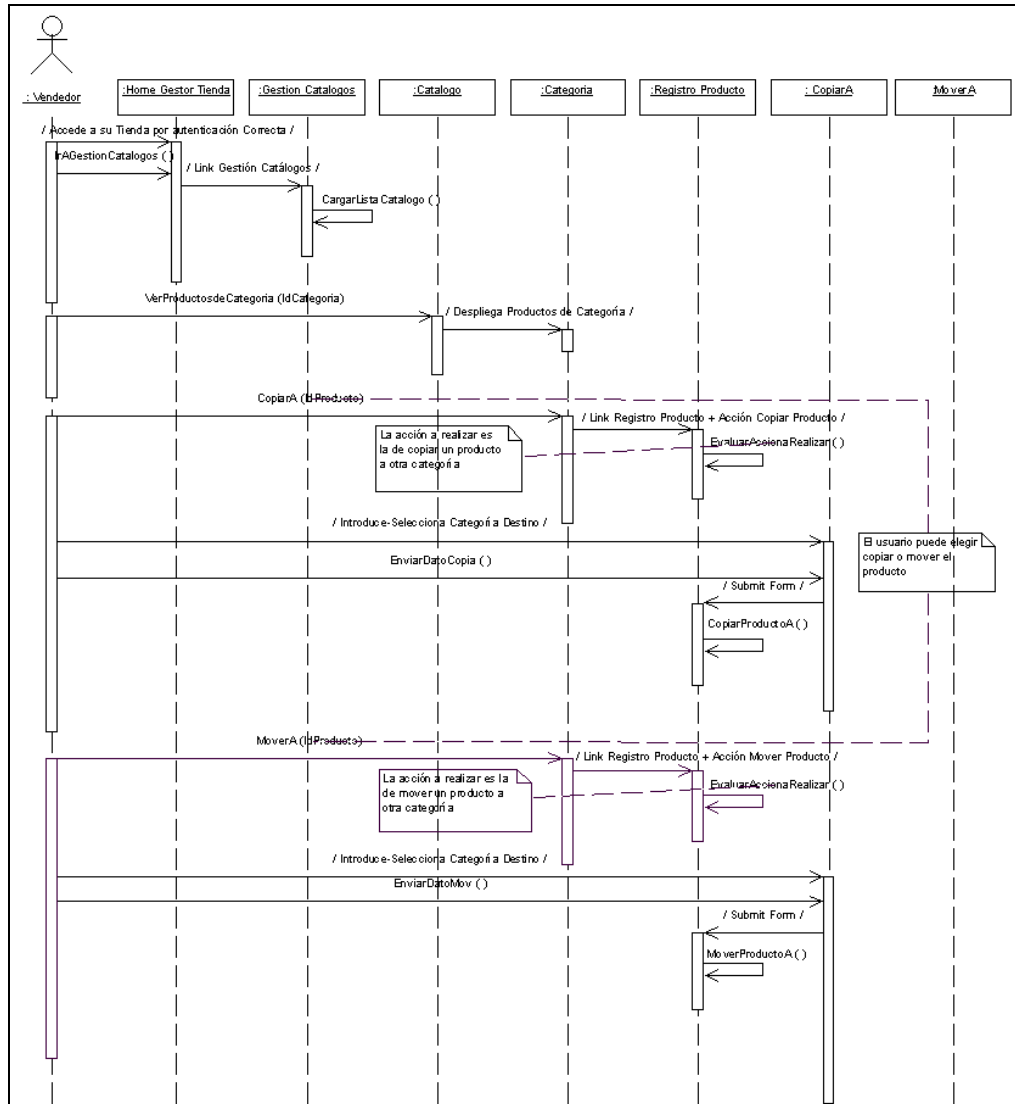


Figura 6.17: Secuencia Copiado/Movimiento de un producto existente a otra categoría

6.2.17 Secuencia de Gestor de Tienda: Eliminación de un producto existente

La secuencia para el procedimiento de eliminación de un producto del catálogo de la tienda asociada al CCV se describe en la figura 6.18. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda* previa autenticación correcta como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Catálogos*. Esta <<navigation class>> carga el <<index>> *Catálogo*, que es un listado de todas las categorías existentes en la tienda. De este listado, el usuario Vendedor selecciona, mediante la operación *VerProductosdeCategoria()*, a una de estas categorías para ver el listado de productos que contiene. Al hacer esto se carga el <<index>> *Categoría*. Este <<index>> es un listado de los productos que contiene la categoría seleccionada. La secuencia prosigue cuando el usuario Vendedor selecciona a una de ellas para eliminarla mediante la operación *Eliminar()* asociada a dicho producto. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Producto*. Cuando *Registro Producto* se carga se evalúa la acción a realizar que, en este caso, es la eliminación de un producto. La secuencia concluye con la operación *EliminarProducto()* de la <<navigation class>> *Registro Producto*, con la cual se lleva a cabo la eliminación (marcado) del registro en la fuente de datos del producto seleccionado.

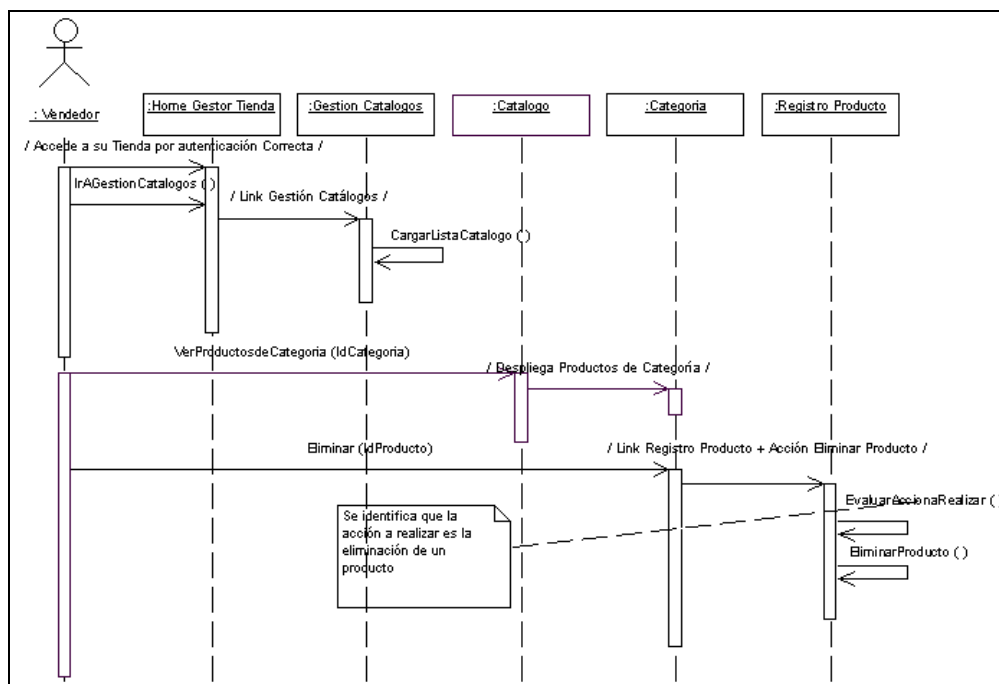


Figura 6.18: Secuencia Eliminación de un producto existente

6.2.18 Secuencia de Gestor de Tienda: Administración de Pedidos

Esta secuencia, que se muestra en la figura 6.19, describe la manera en que un usuario Vendedor puede administrar los pedidos que le han sido realizados. Esta gestión se lleva a cabo mediante la adición y consulta de movimientos realizados sobre los pedidos. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda*, previa autenticación como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Gestión Pedidos*. En esta <<navigation class>> se carga el <<index>> *Pedidos de Tienda*, que es un listado de todos los pedidos que han sido realizados en la tienda. El Vendedor tiene la opción de filtrar únicamente los pendientes por surtir. De este listado, el usuario Vendedor selecciona, mediante la operación *VerPedido()*, a uno de estos pedidos. Al hacer esto se carga la <<navigation class>> *Detalle Pedido*, en donde se cargan los detalles del Cliente que realizó el pedido, los datos del pedido, el detalle del pedido y el listado de movimientos realizados hasta el momento en el pedido. El Vendedor tiene la posibilidad de añadir nuevos movimientos al pedido. Por ejemplo, un movimiento puede ser un acuse de recibo del pedido, que el pedido esta en proceso de embarque, o bien que ya fue enviado. El usuario Vendedor introduce la información del movimiento a través del <<form>> *Nuevo Movimiento*. Los datos del movimiento se envían a la misma <<navigation class>> *Detalle Pedido*. Una vez que *Detalle Pedido* recibe la información se agrega el movimiento listado de movimientos del pedido, se envía una notificación al Cliente del movimiento realizado y por último se concluye la secuencia refrescando el <<index>> de movimientos.

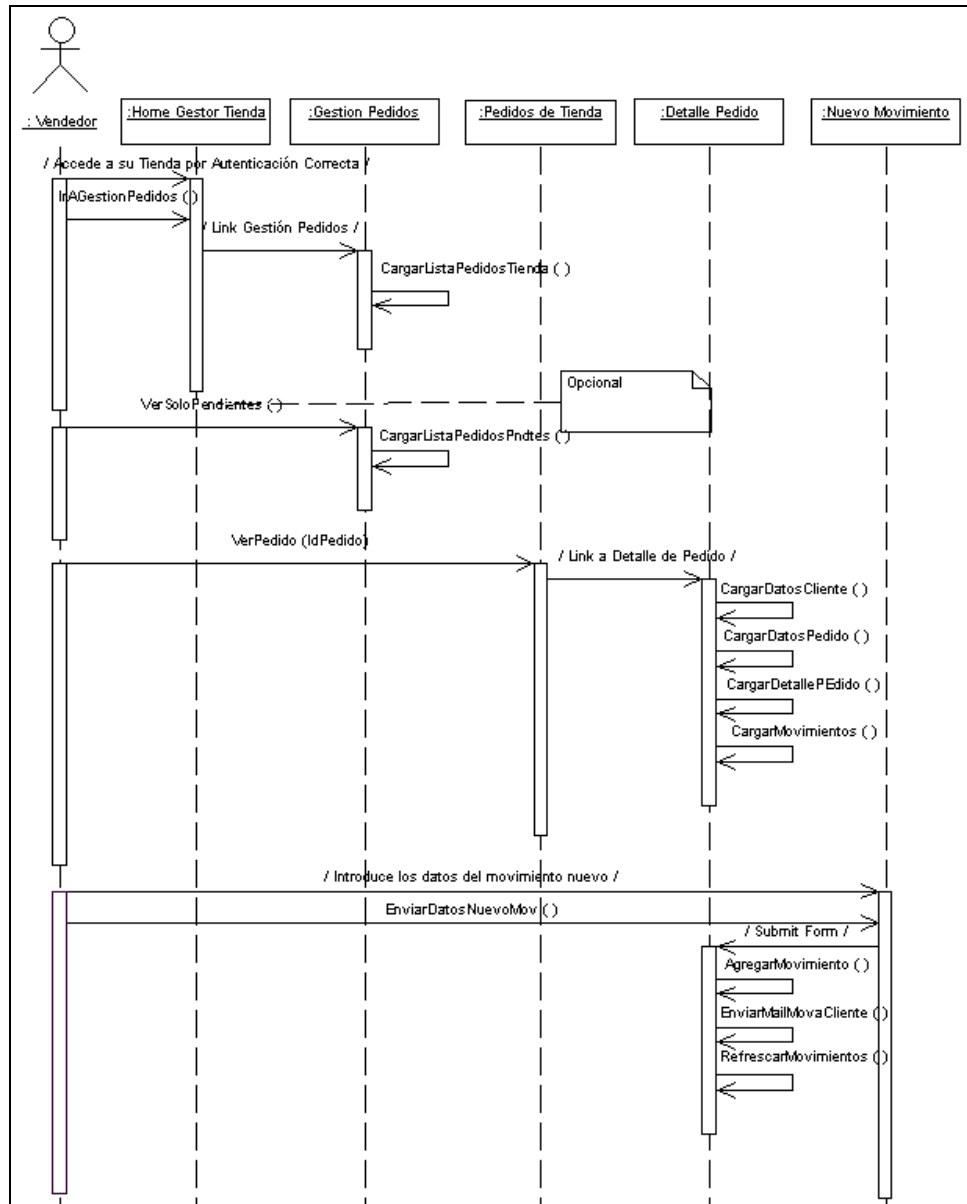


Figura 6.19: Secuencia Administración de Pedidos

6.2.19 Secuencia de Gestor de Tienda: Pago de Comisiones

En la figura 6.20 se muestra la secuencia del procesamiento del pago seguro de comisiones por parte del usuario Vendedor hacia el CCV. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Gestor Tienda*, previa correcta autenticación como un usuario válido. Desde *Home Gestor Tienda* el Vendedor navega hacia la <<navigation class>> *Pago de Comisiones*. Esta <<navigation class>> valida la <<navigation class>> de procedencia mediante la operación *ValidarOrigen()*, esto como mecanismo de seguridad. Si el origen no es válido, entonces se trata de un intento de pago malintencionado, por lo que se notifica la ocurrencia del error, y la secuencia finaliza. Si el origen es válido, entonces la secuencia continúa en la <<navigation class>> *Pago de Comisiones* con el proceso cargando los datos de la tienda y los datos del estado de cuenta sobre el cual se va a efectuar el pago de las comisiones. Para continuar con el procedimiento, el Cliente introduce, mediante el <<form>> *Info Pago Comision*, la información de la tarjeta bancaria con la cual se va a efectuar el pago. La información se envía en modo seguro a la <<navigation class>> *Proceso Pago Seguro* del subsistema Gestor de Transacciones para que sea este elemento quien gestione la transacción con la terminal Bancaria.

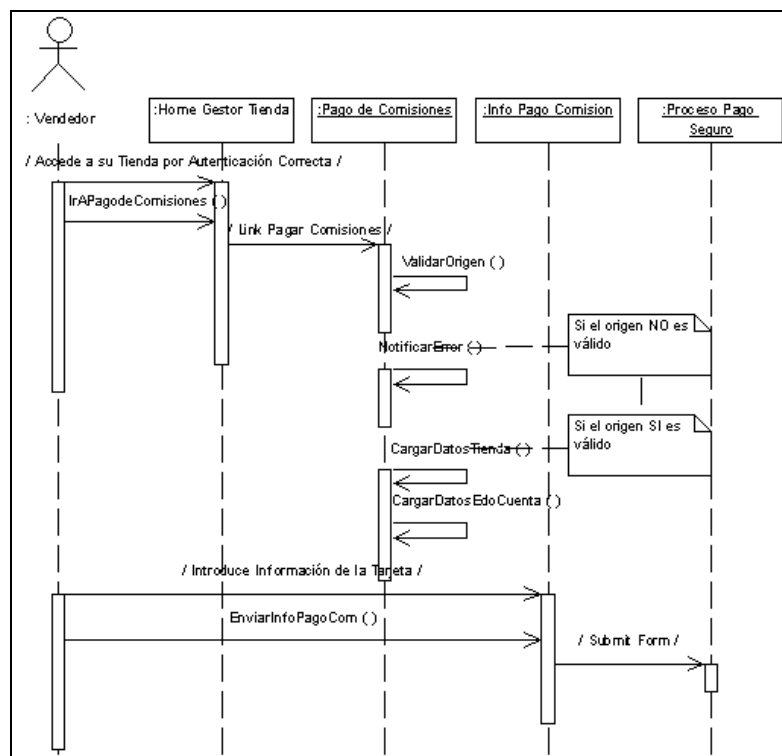


Figura 6.20: Secuencia Pago de Comisiones

6.2.20 Secuencia de Gestor de Transacciones: Procesamiento de pago seguro de Comisiones

Esta secuencia se muestra en la figura 6.21 Esta secuencia comienza cuando el <<form>> *Info Pago Comision*, del subsistema Gestor de Tienda, remite la información de la tarjeta bancaria del usuario Vendedor (administrador de una tienda) hacia la <<navigation class>> *Proceso Pago Seguro* del subsistema Gestor de Transacciones. Cuando la información es recibida por la <<navigation class>> *Proceso Pago Seguro*, lo primero que se realiza es la verificación acerca del origen de la información. Esto es un mecanismo de seguridad para tener certidumbre que información bancaria recibida proviene del subsistema Gestor de Tienda y no de un origen distinto malintencionado. Si el origen no es valido, entonces se notifica que un error ha ocurrido y se termina la secuencia. En el caso esperado que el origen sea válido, entonces la secuencia continúa evaluando la acción a realizar, esto ya que mediante esta misma <<navigation class>> *Proceso Pago Seguro* se procesa también el pago de pedidos por parte de los Clientes. Una vez que se evalúa la acción y se determina que es el pago de comisiones por parte de un usuario Vendedor, *Proceso Pago Seguro* carga los datos de la cuenta de banco de la tienda que esta realizando el pago de las comisiones que han generado sus ventas a través del CCV. La secuencia continua con las operaciones *CrearInstancia_e24TranPipe()*, *Preparar_e24TranPipe()* y *ExecTrans_e24TranPipe()*. Con estas operaciones crea una instancia del objeto COM *e24TranPipe* que es el encargado de realizar la transacción con la terminal bancaria. Una vez creado, se prepara dicho objeto con los datos de la cuenta de banco de la Tienda y con el monto de la operación. *ExecTrans_e24TranPipe()* ejecuta la transacción construyéndola en el formato requerido. Esta cadena es la que se envía a través del canal seguro mediante la operación *Perform_Transaction()* del *e24TranPipe*. Con esta función *e24TranPipe* se comunica con la terminal bancaria y espera una cadena de respuesta. Cuando esta respuesta llega, *e24TranPipe* la notifica a *Proceso de Pago Seguro*, en donde se evalúa dicha cadena de respuesta para verificar si la transacción fue realizada con éxito o no. Dependiendo de esta respuesta se registra el error de pago o el pago exitoso de la comisión. Independientemente de la respuesta, la secuencia continúa con el registro de la operación realizada, se guarda el movimiento en la bitácora, se guarda la información de la transacción y se crea la cadena que se envía como respuesta de la transacción al subsistema Gestor de Tienda. La secuencia finaliza cuando la navegación retorna al subsistema Gestor de Tienda mediante el redireccionamiento hacia la <<navigation class>> *Confirm Pago Comisión* o la <<navigation class>> *Error Pago Comisión* dependiendo si la transacción fue o no exitosa. En el redireccionamiento se incluye la cadena de respuesta que fue creada con el resultado de la operación.

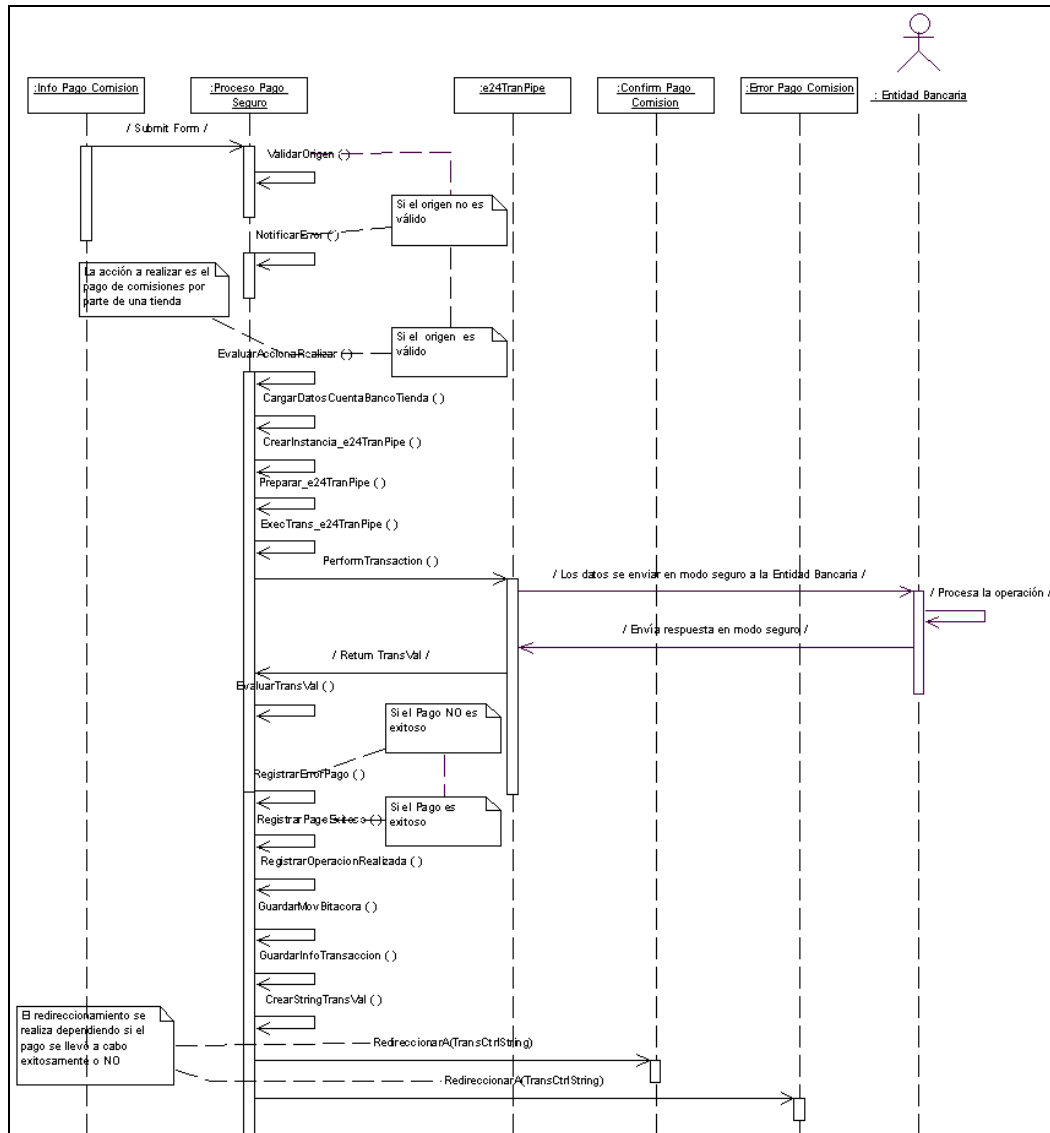


Figura 6.21: Secuencia Procesamiento de pago seguro de Comisiones

6.2.21 Secuencia de ePlaza: Autenticación de Usuario Administrador CCV

Esta secuencia describe el proceso de autenticación de un usuario Administrador CCV para acceder al subsistema de Administración CCV. La secuencia inicia cuando el usuario Administrador accede a la <<navigation class>> *Login Admon* mediante direccionamiento URL. Una vez en *Login Admon*, el Administrador debe introducir sus datos de acceso por medio del <<form>> *Datos Registro Admon*. Esta información es enviada hacia la <<navigation class>> *Autenticación Usuario*. Cuando en *Autenticación Usuario* se recibe esta información se evalúa la acción a realizar, en este caso se trata de la autenticación de un usuario del tipo Administrador. Hecho esto la <<navigation class>> *Autenticación Usuario* realiza la autenticación del usuario Administrador, la cual devuelve un resultado. Si el resultado no es favorable, entonces se debe notificar un error finalizando de esta manera la secuencia. Si el resultado es favorable, entonces *Autenticación Usuario* realiza la operación *IniciarSesionAdmon()* con la que se inicializan las variables de sesión con los datos de registro del Administrador, finalizando la secuencia redireccionando la navegación hacia la <<navigation class>> *Home Administración CCV* mediante la operación *RedireccionarA()* en donde se proporcionan al Administrador diversas opciones para operación del CCV.

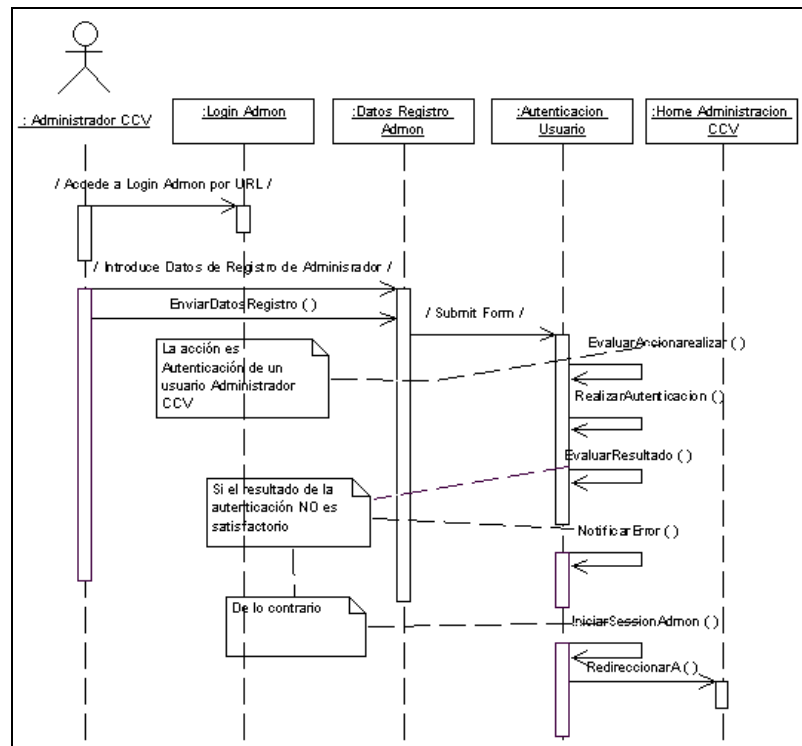


Figura 6.22: Autenticación de usuario Administrador CCV

6.2.22 Secuencia de Administración CCV: Modificación de la Configuración de CCV

Esta secuencia, que se muestra en la figura 6.23, describe la manera de cómo un usuario Administrador del CCV puede modificar la configuración del CCV en su conjunto. La secuencia inicia cuando el Vendedor accede a la <<navigation class>> *Home Administración CCV* una vez que se ha autenticado de manera exitosa. Desde *Home Administración CCV* el Administrador navega hacia la <<navigation class>> *Modificar Conf CCV*. En esta, se encuentran asociados los elementos requeridos para llevar a cabo la modificación de la configuración tales como los <<form>> *Datos Conf Servidor* y *Datos Conf BdD*. Cuando se carga la <<navigation class>> *Modificar Conf CCV* se efectúa la lectura de los <<file>> *CCVConf* y *BdDConf*. Estos elementos representan archivos en donde se guardan los datos de la configuración del CCV. Los datos leídos de estos archivos se cargan en los campos de los <<form>> respectivos. La secuencia continua con la introducción por parte del usuario de los nuevos parámetros de la configuración del servidor. Una vez que han sido completados los campos requeridos, el Administrador solicita la operación *EnviarDatosInfoServ()* para que la nueva configuración sea guardada. Los datos se envían a la <<navigation class>> *Modificar Conf CCV* en donde mediante la operación *GuardarArchivo_InfoServ()* se actualiza el contenido del <<file>> *CCVConf*. A continuación el flujo prosigue con introducción por parte del usuario de los nuevos parámetros de la configuración de la aplicación. Una vez que han sido completados los campos requeridos, el Administrador solicita la operación *EnviarDatosConfBdD()* para que la nueva configuración sea guardada. Los datos se envían a la <<navigation class>> *Modificar Conf CCV* en donde mediante la operación *GuardarArchivo_ConfBdD ()* se actualiza el contenido del <<file>> *BdDConf*.

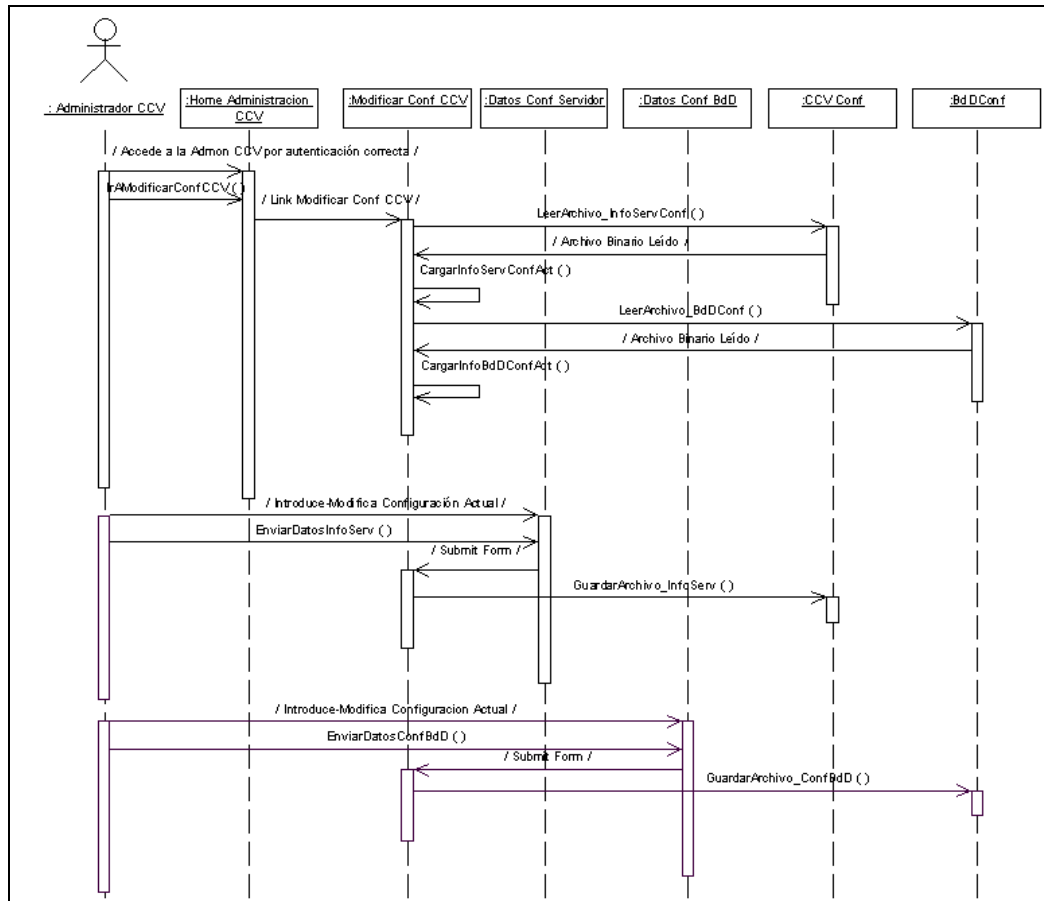


Figura 6.23: Secuencia Modificación de la Configuración de CCV

6.2.23 Secuencia de Administración CCV: Consulta de Estado de Cuenta de Tienda

La secuencia que se muestra en la figura 6.24, describe la manera en como un usuario Administrador consulta el estado de cuenta de una tienda asociada al CCV. La secuencia inicia cuando el Administrador accede, previa autenticación exitosa, a la <<navigation class>> *Home Administración CCV*. Desde *Home Administración CCV* el Administrador navega hacia la <<navigation class>> *Gestión de Cuentas y Comisiones*. La secuencia continúa con el Administrador navegando hacia la <<navigation class>> *Gestión Cuentas Tienda*, en donde despliega el <<index>> *Cuentas Tiendas* que es un listado de todas las cuentas asociadas al CCV. A cada elemento de este listado se asocia un <<form>> *Datos Periodo a Visualizar*. Mediante este <<form>> el Administrador selecciona el mes y el año del estado de cuenta de la tienda que desea consultar. Estos parámetros son enviados a la <<navigation class>> *Estado de Cuenta Tienda*. La secuencia continua en esta <<navigation class>> en donde se cargan los datos generales de la tienda, se realiza el despliegue del estado de cuenta solicitado y se realiza el cálculo de los totales. La secuencia concluye con la verificación del estatus del estado de cuenta, el despliegue de la información de pago de la Tienda, y el monto de las comisiones generadas en el período del estado de cuenta.

6.2.24 Secuencia de Administración CCV: Modificación de status de Tienda

La secuencia que se muestra en la figura 6.25, describe el procedimiento de modificación de status de una tienda asociada al CCV por parte de un usuario Administrador. La secuencia inicia cuando el Vendedor accede, previa autenticación exitosa, a la <<navigation class>> *Home Administración CCV*. Desde *Home Administración CCV* el Administrador navega hacia la <<navigation class>> *Gestión de Cuentas y Comisiones*. Posteriormente, el Administrador navega hacia la <<navigation class>> *Gestión Cuentas Tienda*. En esta <<navigation class>> se carga el listado de las cuentas de las tiendas, una por cada tienda. Este listado es el <<index>> *Cuentas Tiendas*. A partir de este <<index>> el Administrador selecciona, mediante la operación `IrASatutTienda()`, la tienda sobre la cual desea realizar la modificación del estatus respectivo. Ya en la <<navigation class>> *Status Tienda* se carga el status general y el status financiero de la tienda en cuestión. La secuencia continua con la introducción por parte del usuario de los nuevos status para la tienda mediante el <<form>> *Datos Status Tienda*. Los nuevos status son enviados a la <<navigation class>> *Status Tienda*, en donde se actualizan los nuevos status y se envía una notificación del cambio de status realizado, vía correo electrónico, al usuario Vendedor administrador de la tienda.

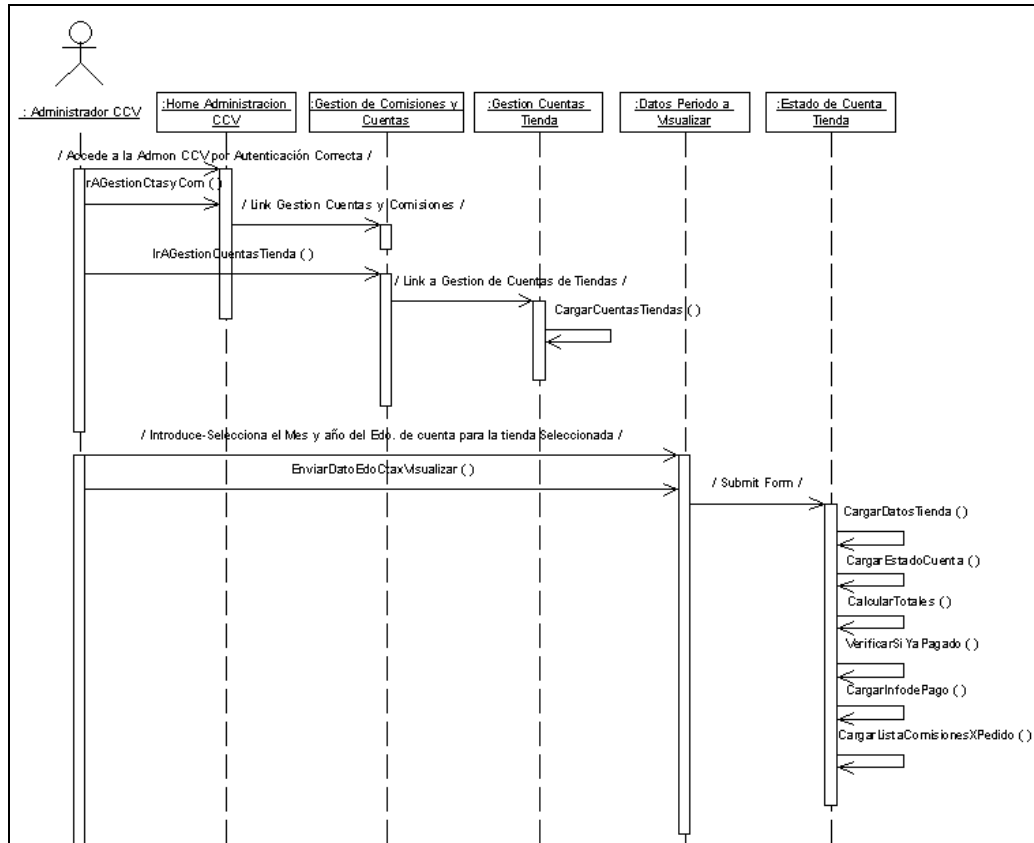


Figura 6.24: Secuencia Consulta de Estado de Cuenta de Tienda

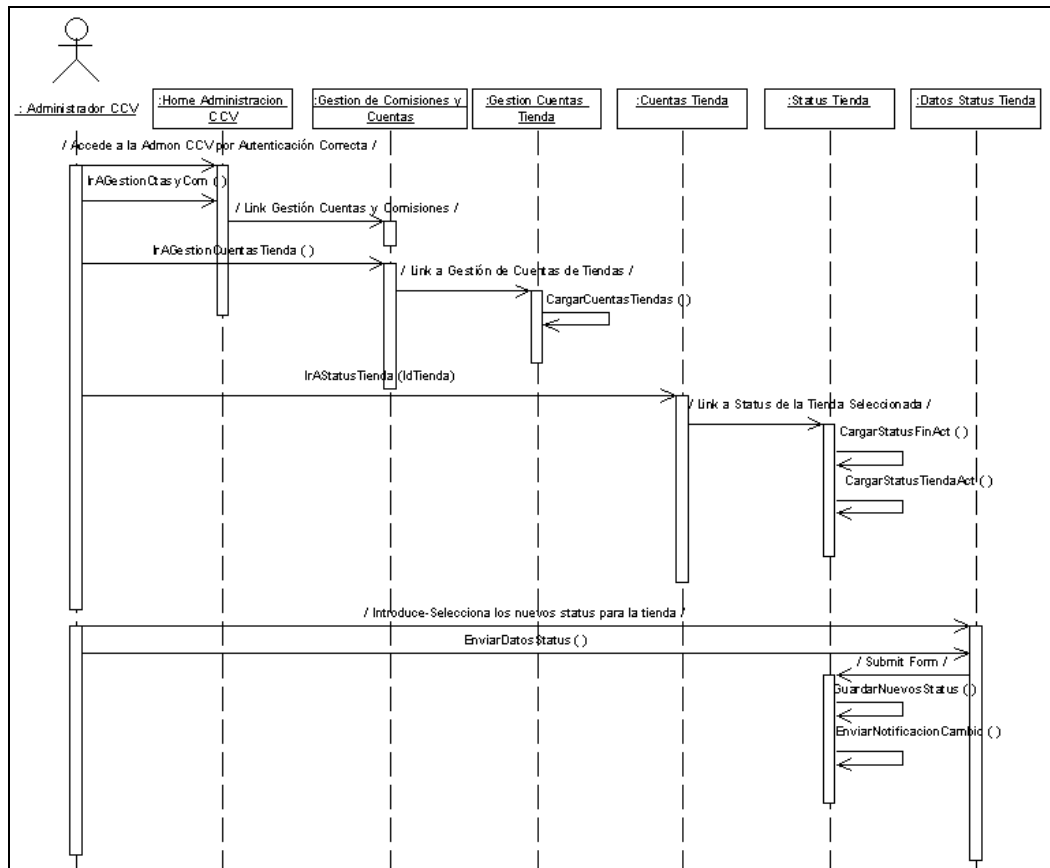


Figura 6.25: Secuencia Modificación de status de tienda

6.2.25 Secuencia de Administración CCV: Adición de una nueva tienda al CCV

Esta secuencia, que se muestra en la figura 6.26, describe el procedimiento de adición de una nueva tienda al CCV por parte del usuario Administrador de CCV. La secuencia inicia cuando el usuario Administrador accede, previa autenticación correcta, a la <<navigation class>> *Home Administración CCV*. Desde *Home Administración CCV* el Administrador navega hacia la <<navigation class>> *Manejo Padrón de Tiendas*. En esta <<navigation class>> se carga la lista de de tiendas ya registradas al CCV. Para agregar un nuevo producto, el usuario Administrador dispara la operación *IraRegistroTiendaNueva()* para navegar hacia la <<navigation class>> *Registro Tienda*. Cuando *Registro Tienda* se carga, evalúa la acción a realizar que, en este caso, es la agregación de una nueva tienda al CCV. El usuario Administrador, a través de los campos definidos en el <<form>> *Datos Tienda*, introduce los datos de la nueva tienda. Entre los datos requeridos para el registro se encuentran los datos generales de la tienda, domicilio, fecha de registro, etc. En la introducción de esta información se definen una secuencia opcional para la captura de las fechas mediante el <<component>> *Selector Fecha*. La secuencia de utilización de este componente se ha descrito en otras secuencias del modelo. La secuencia del registro de la nueva tienda continúa, en su flujo básico, cuando el usuario Administrador dispara la operación *EnviarDatosTienda()* el <<form>> *Datos Tienda*. Esta información es enviada hacia la <<navigation class>> *Registro Tienda* en donde se valida que dicha información sea correcta. Si la validación no es satisfactoria entonces se notifica el error y la secuencia termina. Por el contrario, si la validación de la información es correcta, la secuencia continua en la <<navigation class>> *Registro Tienda* con la operación *GenerarIdTienda()* con la cual se genera el identificador único de la nueva tienda a registrar y a continuación se ejecutan las operaciones para registrar la tienda nueva en la fuente de datos, asociar los logotipos iniciales a la tienda nueva, crear los directorios para el manejo de los archivos de la tienda y para crear los usuarios asociados a la tienda con sus respectivos passwords. Hecho esto, el usuario Administrador introduce, por medio del <<form>> *Datos Banco Tienda*, la información bancaria de la tienda. Esta información es enviada hacia la <<navigation class>> *Registro Tienda* donde se recupera y se efectúa el registro. En la implementación se sugiere que esta última secuencia sea únicamente de acceso local restringido, con el propósito de que la información bancaria que se utiliza no viaje a través de la red, protegiendo de esta manera la confidencialidad.

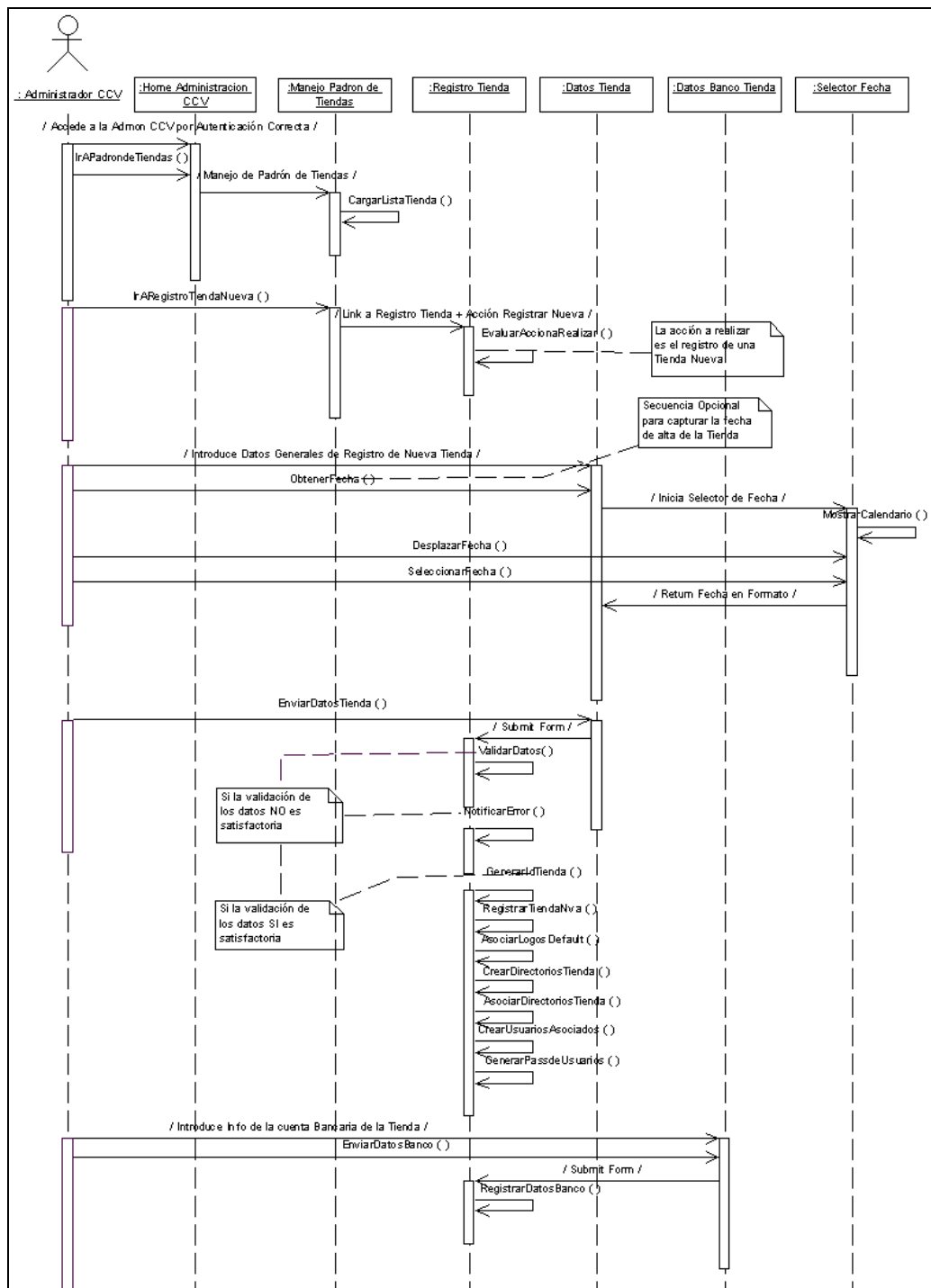


Figura 6.26: Secuencia Adición de una nueva tienda al CCV

6.2.26 Secuencia de Administración CCV: Edición del registro de una tienda existente

La secuencia que se muestra en la figura 6.27, describe el procedimiento de edición del registro de una tienda asociada al CCV por parte del usuario Administrador de CCV. La secuencia inicia cuando el usuario Administrador accede, previa autenticación correcta, a la <<navigation class>> *Home Administración CCV*. Desde *Home Administración CCV* el Administrador navega hacia la <<navigation class>> *Manejo Padrón de Tiendas*. En esta <<navigation class>> se carga la lista de de tiendas ya registradas al CCV. De este listado el usuario Administrador selecciona, mediante la operación *Editar()*, a la tienda cuyo registro desea sea modificado. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Tienda*. Cuando *Registro Tienda* se carga se evalúa la acción a realizar que, en este caso, es la edición del registro de una tienda ya existente en el CCV. Subsecuentemente, se cargan los datos generales y bancarios actuales de la tienda a editar en los <<form>> *Datos Tienda* y *Datos Banco Tienda*, respectivamente. La secuencia continua con la introducción de la nueva información general por parte del usuario. Esta información es enviada a la <<navigation class>> *Registro Tienda* en donde se recupera y se realiza la actualización del registro de la tienda. La secuencia concluye con la introducción de la nueva información bancaria por parte del usuario. Esta información es enviada también a la <<navigation class>> *Registro Tienda* en donde se recupera y se realiza la actualización del registro de los datos bancarios de la tienda de la tienda. Al igual que cuando se realiza el registro inicial de la tienda, para la introducción los datos fecha se hacer uso del <<component>> *Selector Fecha*, cuya secuencia de uso se ha descrito en diagramas anteriores. También se sugiere que en la implementación que esta secuencia sea únicamente de acceso local restringido, con el propósito de que la información bancaria que se utiliza no viaje a través de la red, protegiendo de esta manera la confidencialidad.

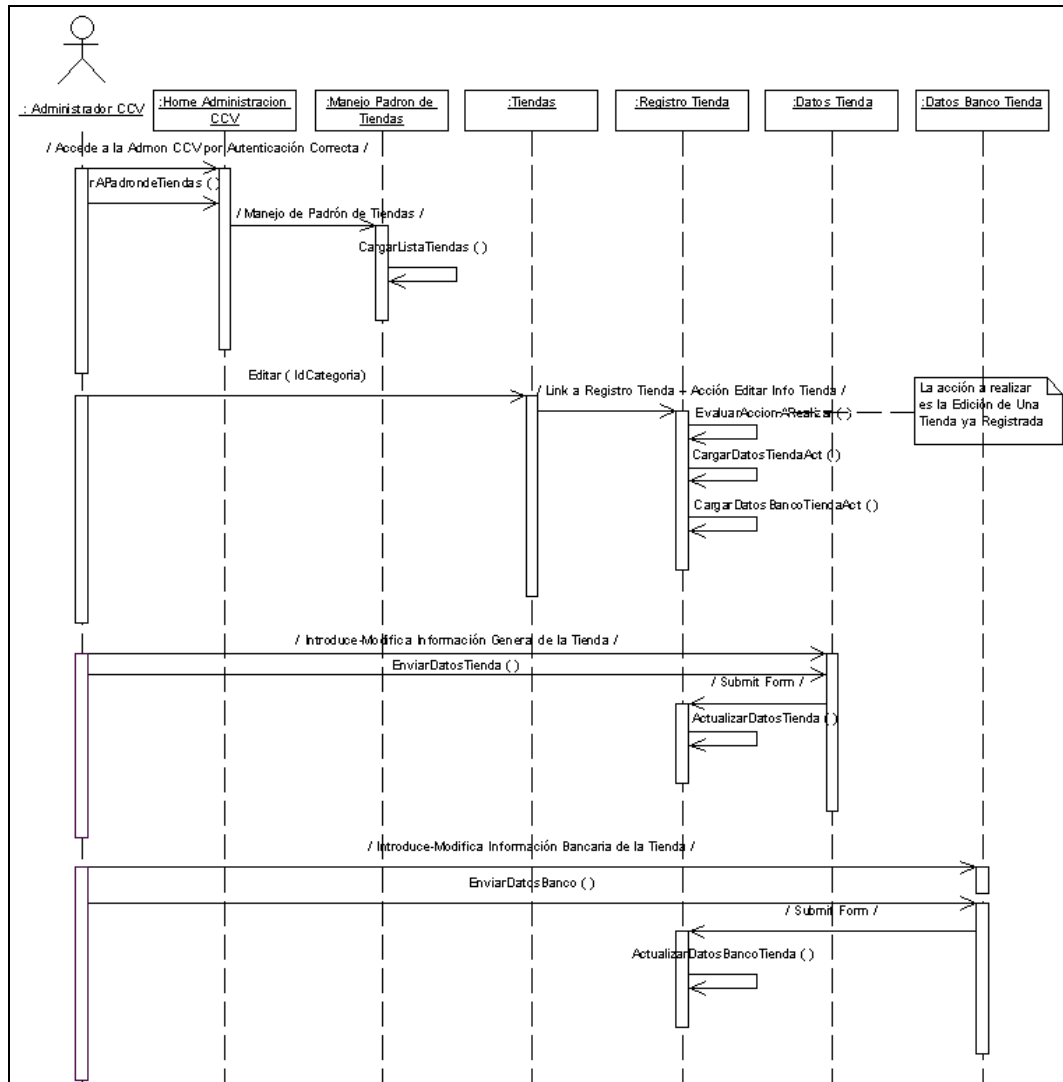


Figura 6.27: Secuencia Edición del registro de una tienda existente

6.2.27 Secuencia de Administración CCV: Eliminación del Registro de una tienda existente

La figura 6.28 describe el procedimiento de eliminación del registro de una tienda existente asociada al CCV por parte del usuario Administrador. La secuencia inicia cuando el usuario Administrador accede, previa autenticación correcta, a la <<navigation class>> *Home Administración CCV*. Desde *Home Administración CCV* el Administrador navega hacia la <<navigation class>> *Manejo Padrón de Tiendas*. En esta <<navigation class>> se carga el <<index>> *Tiendas*. Este <<index>> es un listado de las tiendas registradas al CCV. De este listado el usuario Administrador selecciona, mediante la operación *Eliminar()*, a la tienda cuyo registro desea eliminarse. Esta operación hace navegar al Vendedor a la <<navigation class>> *Registro Tienda*. Cuando *Registro Tienda* se carga se evalúa la acción a realizar que, en este caso, es la eliminación de un producto ya existente en el catálogo. La secuencia concluye con la operación *EliminarTienda()* de la <<navigation class>> *Registro Tienda*, con la cual se lleva a cabo la eliminación (marcado) del registro de la tienda en la fuente de datos del CCV.

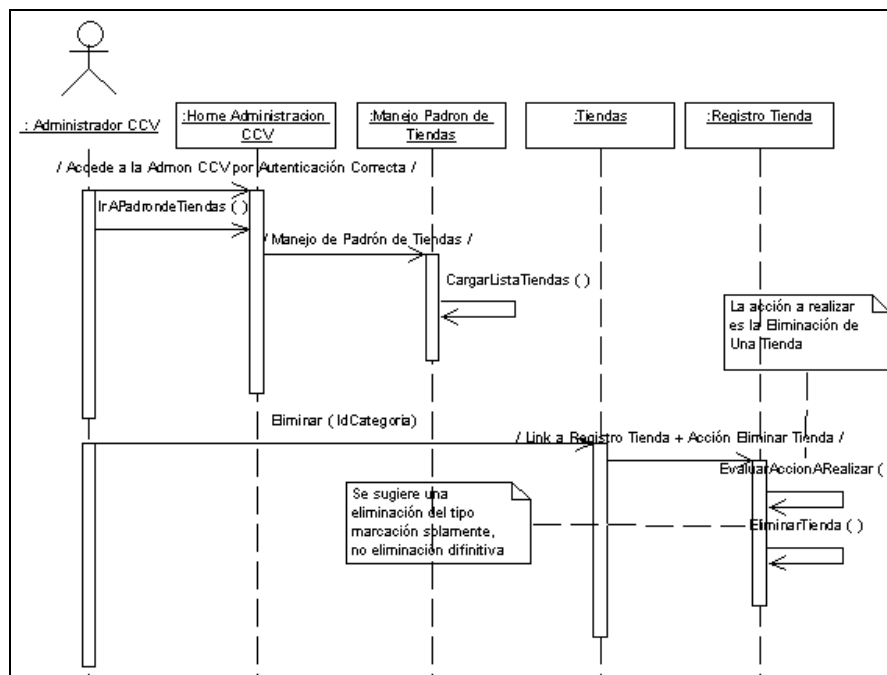


Figura 6.28: Secuencia Eliminación del Registro de una tienda existente

6.2.28 Secuencia de Administración CCV: Envío de Mail a un miembro del CCV

La figura 6.29 describe el procedimiento por medio del cual un usuario Administrador CCV envía un correo electrónico a un miembro registrado del CCV. La secuencia inicia cuando el usuario Administrador accede, previa autenticación correcta, a la <<navigation class>> *Home Administración CCV*. Desde el Administrador navega hacia la <<navigation class>> *Enviar MailA*. En esta <<navigation class>> se carga el directorio completo de todos los posibles destinatarios registrados en el CCV, ya sean Clientes o Vendedores. También se carga el directorio de cuentas que pueden ser utilizadas para enviar el mail. Estos directorios se cargan dentro del <<form>> *Datos Mail* para que el usuario Administrador pueda realizar la selección del destinatario como de la entidad emisor. La secuencia continua con la introducción de los datos del correo por parte del usuario Administrador. Estos datos incluyen la selección del o de los destinatarios y de la cuenta que se utiliza para enviar el correo. Estos datos se envían hacia la <<navigation class>> *Enviar MailA* en donde ejecuta la operación *EnviarMailaDestino()*, la cual, realiza el envío del correo a los destinatarios seleccionados.

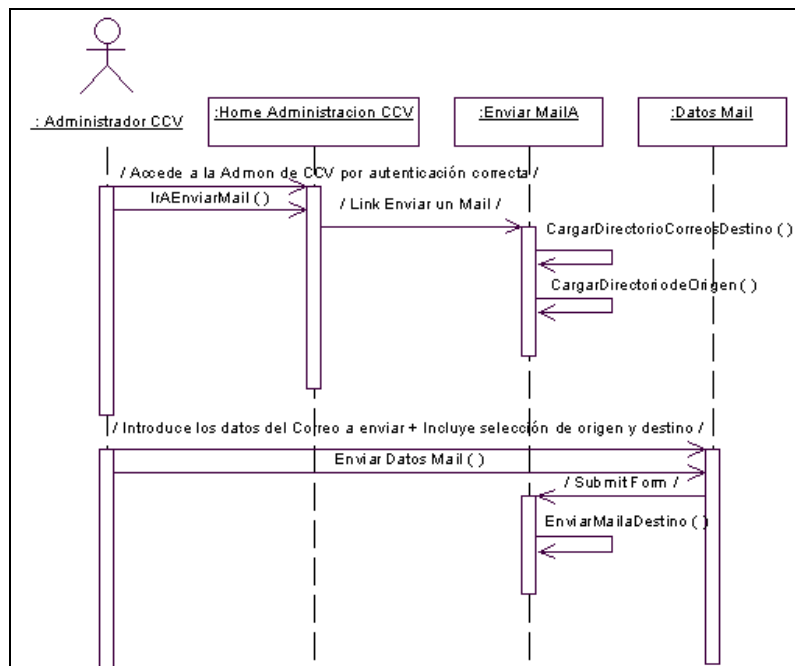


Figura 6.29: Secuencia Envío de Mail a un miembro del CCV

Capítulo 7

Escenarios Web

En este capítulo se describen los Escenarios de Navegación del CCV. Los diagramas UML que se utilizan para visualizar estos escenarios de navegación son los diagramas de estados. En una primera sección se describe la semántica de los elementos de modelado que se utilizan en los diagramas de estado. Posteriormente, se presentan los escenarios de navegación más significativos para el CCV.

7.1 Los escenarios Web y la semántica asociada a los diagramas de estados

Un escenario Web es una técnica para la descripción de situaciones de uso de un sitio Web con una meta o espacio concreto. Estas situaciones son el resultado de acciones llevadas a cabo por un personaje (usuario) o bien por la propia aplicación [Manchón03]. De acuerdo a las acciones que toma se toman en cuenta para modelar las situaciones que se pueden presentar, los escenarios pueden ser de tres tipos: escenarios que describen acciones de alta frecuencia, escenarios que describen acciones necesarias y escenarios que describen acciones marginales. Los escenarios que describen acciones de alta frecuencia describen situaciones muy frecuentes en la aplicación Web, pero triviales y poco trascendentes. Los escenarios que describen acciones necesarias describen situaciones que en algún momento se llevan a cabo dentro de la aplicación Web como parte de una funcionalidad de la misma. Por último, los escenarios que describen acciones marginales describen situaciones de poca incidencia, y por lo mismo fáciles de ignorar. Los escenarios Web que se modelan en este apartado describen las acciones necesarias de funcionalidades provistas por el CCV.

Por otra parte, un diagrama de estados de UML denota una secuencia de los estados que un objeto puede adquirir durante su vida, junto con acciones responsivas, disparando eventos y las condiciones asociadas para indicar transiciones. Tomando en cuenta lo anterior, en este apartado se utilizan los diagramas de estados para visualizar escenarios de navegación Web. Estos diagramas permiten visualizar la parte dinámica del modelo de navegación, especificando los eventos que disparan las transiciones, definen condiciones y explícitamente incluyen las acciones a ser realizadas [Koch02].

Debido a lo anterior, la semántica de los elementos que constituyen los diagramas de estado es distinta a la habitual. A continuación se describe la semántica de los elementos de modelado que aparecen en los diagramas de estado.

Elemento	Semántica
	Representa la situación inicial del escenario. Esta situación normalmente no tiene gran representatividad en la aplicación Web, pero se incluye como referencia inicial para el escenario Web que se está describiendo.
	Representa a una situación del escenario Web en la provisión de una funcionalidad. Cada situación involucra actividades y a las entidades que participan en ella. En la nomenclatura de estas situaciones se utilizan descriptores de las entidades que participan en la situación y de las actividades que realizan, concatenadas con el carácter +.
	Mediante las flechas se representan a eventos, acciones o circunstancias que provocan el traslado del flujo de una situación a otra. Las flechas se etiquetan con una descripción breve de dicho evento, acción o circunstancia.

En base a esta semántica, a continuación se presentan los escenarios Web más significativos para el CCV.

7.2 Escenarios Web del CCV

7.2.1 Escenario Web ePlaza, 1ra parte

En este escenario, que se muestra en la figura 7.1 se describe la operación del subsistema ePlaza de manera global. Cuando un usuario accede al Home CCV se carga el listado de tiendas y el formulario de búsqueda. En esta situación el Cliente puede seleccionar una Tienda o bien intentar una búsqueda. Si el Cliente intenta una búsqueda entonces, la situación cambia a Procesando Búsqueda. Cuando la búsqueda concluye, entonces se despliega la lista de los resultados, de donde el Cliente puede seleccionar un producto para visualizar los Detalles del producto. Ahora bien, si desde el Home CCV el Cliente selecciona una tienda específica, entonces la situación cambia hacia Home Tienda, en donde se carga el catálogo de la tienda. Si el Cliente ahora selecciona una categoría, se presenta el listado de categorías disponibles. En esta situación, si el Cliente selecciona a una categoría, entonces se presenta el listado de productos de la categoría seleccionada. Si el Cliente selecciona a un producto, entonces la situación cambia a la presentación de los detalles del producto seleccionado. En la situación Detalle del Producto, un Cliente puede decidir agregarlo al carrito de compras. Ya estando en el carrito, el Cliente puede elegir seguir comprando regresando a Home Tienda para seleccionar otro producto y adicionarlo al pedido, o realizar cambios sobre los productos que ya se encuentran en el carrito del Cliente. Por último, estando en el carrito, un Cliente puede decidir comprar el producto, entonces la situación cambia para iniciar el proceso de pago en Login Cliente. Si el Cliente es nuevo entonces, la situación cambia para realizar que el Cliente sea registrado. Cuando el registro se realiza de manera correcta se inicia el proceso de pago en su primera etapa. Por otro lado, si el Cliente no recuerda sus datos de acceso entonces se realiza un procedimiento para reenviar los datos de registro al Cliente vía correo electrónico. Cuando el Cliente introduce sus datos de registro, entonces se lleva a cabo la autenticación de usuario. Si es exitosa se inicia el proceso de pago en su primera etapa, de lo contrario se retoma la situación para que el Cliente vuelva a intentar autenticarse de manera correcta. Estando en la situación de la primera etapa del pago, el Cliente debe seleccionar un servicio de envío para su pedido. En la primera etapa de pago, el Cliente puede también elegir modificar los datos de envío del pedido. En la situación de la segunda etapa de pago, el Cliente puede elegir modificar los datos de facturación. En esta situación, el Cliente introduce los datos de la tarjeta con la cual desea efectuar el pago. Una vez que el Cliente acepta el cargo a la tarjeta, la información de la tarjeta es enviada para que se realice la transacción. La situación cambia entonces para procesar dicha operación. El proceso puede terminar con dos resultados. Si el resultado es exitoso, entonces la situación final es de confirmación de la operación efectuada. Por el contrario, si el resultado del pago intentado no es exitoso, entonces la situación final es la notificación del error ocurrido.

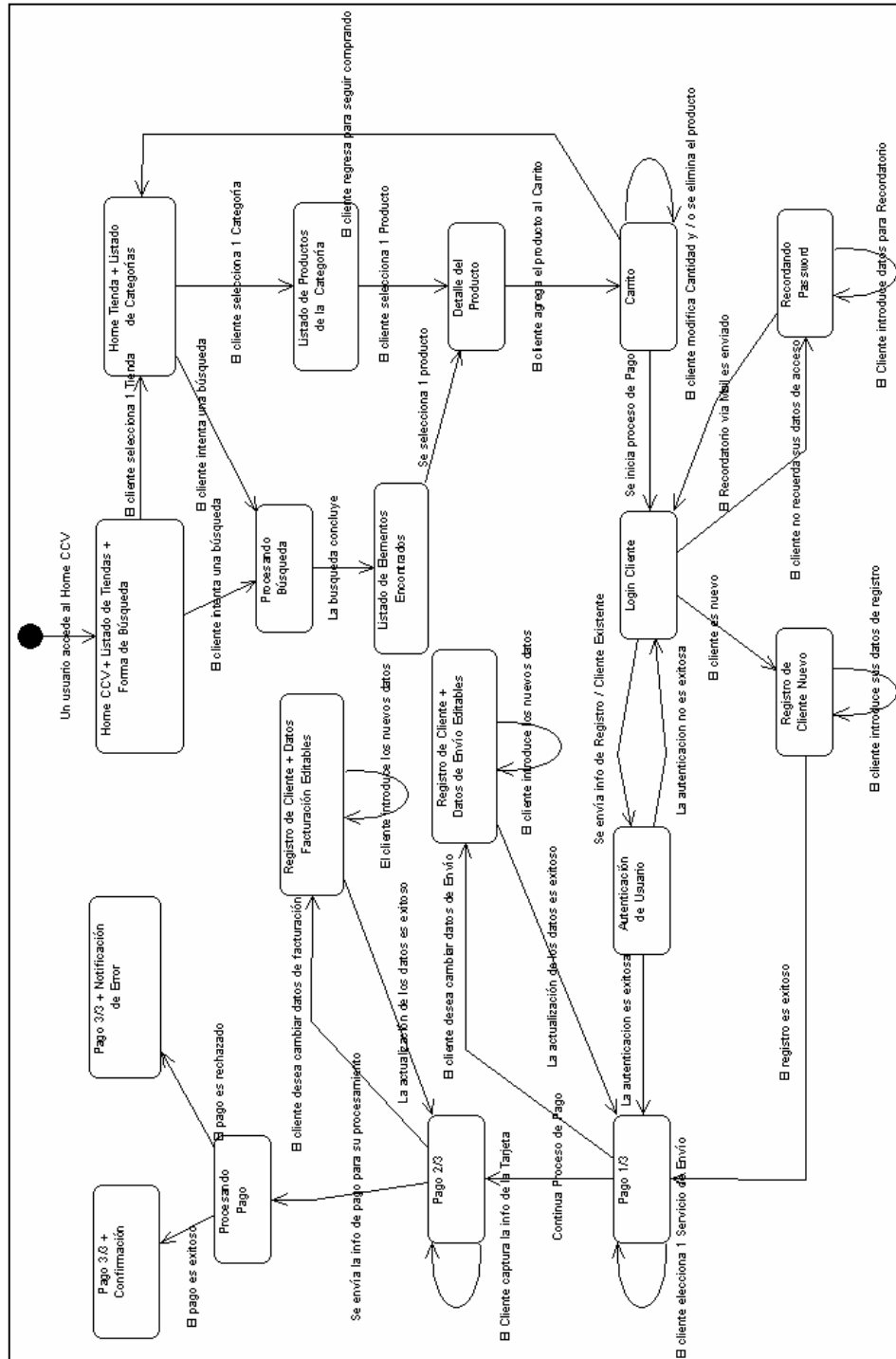


Figura 7.1: Escenario Web ePlaza 1ra parte

7.2.2 Escenario Web ePlaza, 2da parte

Este escenario se muestra en la figura 7.2. En esta segunda parte del escenario Web del subsistema ePlaza se muestran las situaciones asociadas con la funcionalidad de autenticación para los usuarios tipo Vendedor y Administrador CCV. De acuerdo a este escenario si un usuario Vendedor desea ingresar al subsistema Gestor de Tienda debe acceder al Home CCV. Desde esta situación el usuario Vendedor selecciona acceder a la interfase de ingreso del subsistema Gestor de Tienda. Esto provoca una transición hacia la situación en donde el usuario Vendedor introduce sus datos de registro. El proceso de autenticación se representa con una situación, después de la cual, en base al resultado de dicho proceso, se transita hacia el Gestor de Tiendas o de nuevo a la interfase de ingreso. En este escenario también se presenta la situación que representa al proceso de autenticación correcta de un usuario tipo Administrador. En este caso el usuario Administrador accede directamente, sin pasar por Home CCV, a la interfase de ingreso del subsistema de Administración del CCV en donde el Administrador introduce los datos de registro. El proceso de autenticación se representa con una situación, después de la cual, en base al resultado de dicho proceso, se transita hacia el subsistema de Administración del CCV o de nuevo a la interfase de ingreso.

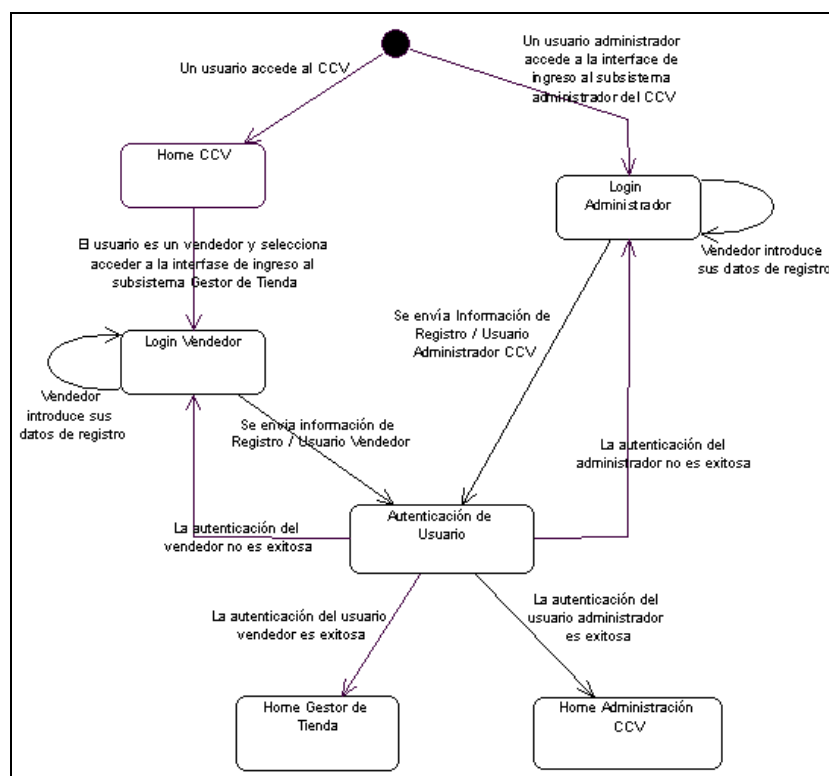


Figura 7.2: Escenario Web ePlaza, 2da parte

7.2.3 Escenario Web Gestor de Tienda, 1ra Parte

En esta primera parte del escenario Web del gestor de tienda, que se muestra en la figura 7.3, se describen las situaciones asociadas con la funcionalidad de modificación de la configuración de la tienda y de la gestión de catálogos. Una vez que un usuario del tipo Vendedor accede por autenticación correcta a su gestor de tiendas, la situación va a depender de la selección que haga el Vendedor en base a la funcionalidad que desea utilizar. En el caso que el Vendedor selecciona modificar la configuración de la tienda se carga el módulo correspondiente con los datos de la configuración actual. El Vendedor, en esta situación, puede modificar estos datos de la configuración actual y guardar los cambios realizados. Cuando los datos son guardados correctamente, entonces se presenta la configuración de la tienda con una confirmación de los cambios efectuados. La funcionalidad de la gestión del catálogo presenta varias situaciones. Cuando se accede a este módulo de la aplicación se presenta un listado de las categorías que se incluyen en el catálogo. En esta situación, el Vendedor puede seleccionar registrar una categoría nueva, registrar un producto nuevo, o bien, seleccionar a una categoría del listado para visualizar los productos asociados a dicha categoría o bien, para editarla o eliminarla. A partir de esta situación, el Vendedor también puede seleccionar modificar la configuración del stock. Cuando se selecciona ver el listado de productos de una categoría, las opciones para el usuario Vendedor en esa situación son cuatro; el usuario puede seleccionar a un producto del listado para editarlo, eliminarlo, copiarlo a otra categoría o moverlo a otra categoría. Cada una de estas funcionalidades presenta distintas situaciones, las cuales se presentan de acuerdo a las operaciones que el Vendedor realice en la aplicación.

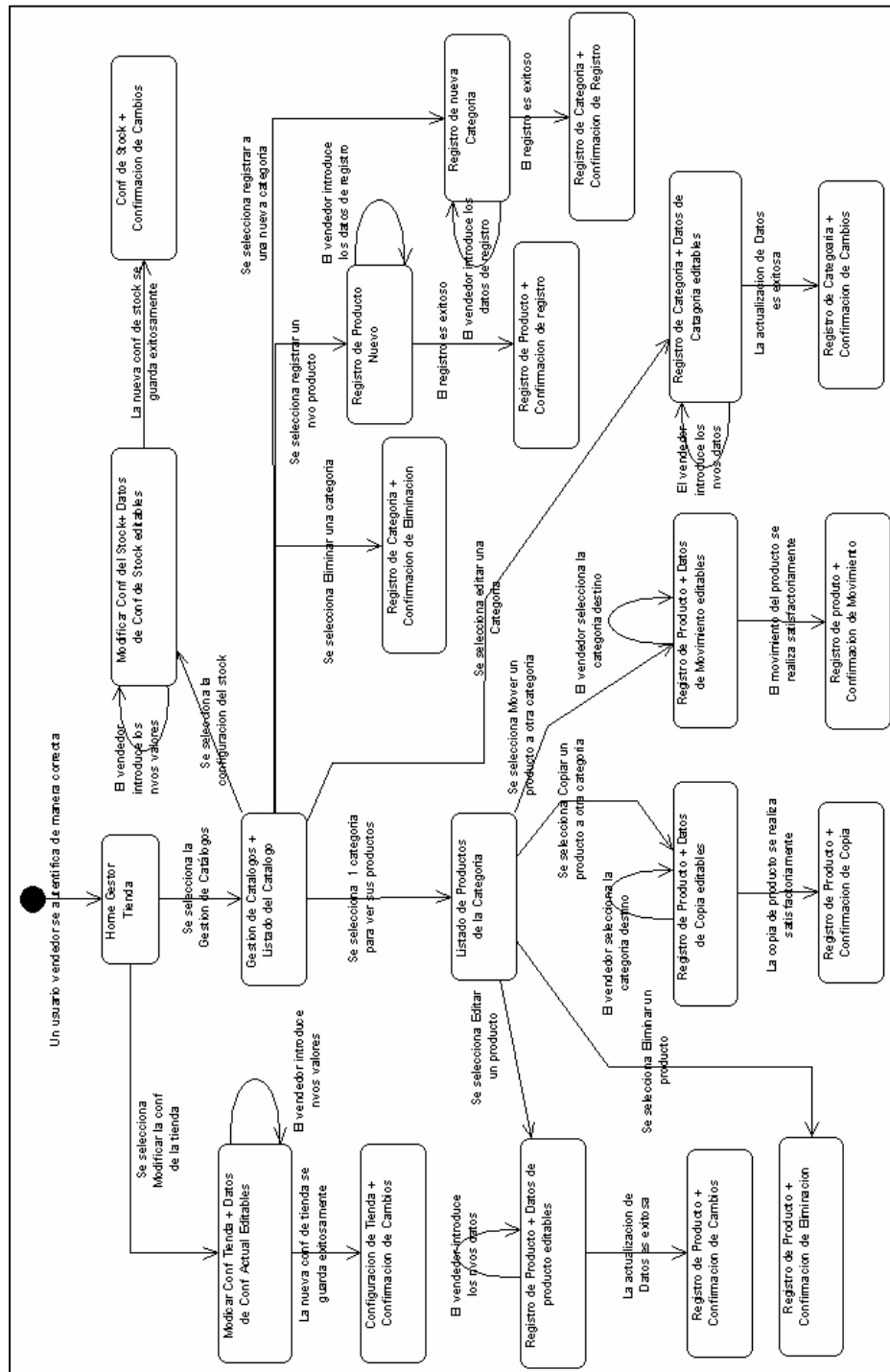


Figura 7.3: Escenario Web Gestor Tienda, 1ra parte

7.2.4 Escenario Web Gestor de Tienda, 2da Parte

En esta segunda parte del escenario Web del gestor de tienda, que se muestra en la figura 7.4, se describen las situaciones asociadas con un segundo bloque de funcionalidades provistas por este subsistema. Estas funcionalidades son las de gestión de servicios de envío, gestión de pedidos, envío de correos electrónicos, visualización de informes y el pago de comisiones. Una vez que un usuario del tipo Vendedor accede por autenticación correcta a su gestor de tiendas, la situación del subsistema va a depender de la selección que haga el Vendedor en base a la funcionalidad que desea utilizar. En el caso que el Vendedor selecciona operar sobre los pedidos de la tienda se carga el módulo correspondiente. Ya en esta situación del subsistema, si el Vendedor selecciona a uno de los pedidos del listado, entonces se dispara la transición a la gestión de los movimientos para ese pedido específico. Otra funcionalidad que se describe en este escenario es la de la gestión de los servicios de envío. Dentro de las situaciones que describen a esta funcionalidad se encuentran la adición de nuevos servicios de envíos, la modificación de servicios de envío ya existentes y la eliminación de servicios que ya no desea el Vendedor se encuentren a disposición de los Clientes. Cuando estando en Home Tienda el Vendedor selecciona acceder al panel de envío de correos, se transita hacia el panel correspondiente. Ya dentro de este panel, el Vendedor parametriza el correo que desea a enviar y lo envía. Si el correo se envía correctamente se transita a una situación de confirmación. La visualización de informes se describe con tres situaciones, la presentación de la funcionalidad, el despliegue del listado de colección de informes y la visualización de un informe específico. Mediante estas tres situaciones el Vendedor puede visualizar un informe de la colección de informes disponibles para la tienda. Por último, para la funcionalidad del pago de comisiones se muestran cuatro situaciones. En la primera de ellas el usuario introduce los datos de la tarjeta con la cual desea realizar el pago. En la segunda se representa el compás de espera del tiempo que se toma el intento de transacción bancaria. Con las dos últimas situaciones se representan las dos posibles respuestas, de éxito o fracaso, al intento de transacción.

7.2.5 Escenario Web Administración CCV, 1ra Parte

En esta primera parte del escenario Web del subsistema de Administración de CCV, que se muestra en la figura 7.5, se describen las situaciones asociadas con un primer bloque de funcionalidades provistas por este subsistema. Estas funcionalidades son las de configuración de CCV, gestión de cuentas de tienda, gestión de comisiones y cuentas, y administración del padrón de tiendas. Una vez que un usuario del tipo Administrador accede por autenticación correcta al subsistema de Administración de CCV, la situación va a depender de la selección que haga el usuario Administrador en base a la funcionalidad que desea utilizar. Si el Administrador selecciona modificar la configuración del CCV se carga el módulo correspondiente con los datos de la configuración actual del CCV. El Administrador, en esta situación, puede modificar estos datos de la configuración actual y guardar los cambios realizados. Cuando los datos son guardados correctamente, entonces se presenta la nueva configuración del CCV juntamente con una confirmación de los cambios efectuados. Por otra parte, cuando el Administrador selecciona trabajar sobre la configuración de comisiones y el manejo de cuentas se carga el módulo correspondiente. Para la configuración de las comisiones se presentan dos situaciones, una en la que el usuario visualiza la configuración actual y realiza los cambios y otra en la que se confirma que la nueva configuración de las comisiones ha sido correctamente guardada. El manejo de las cuentas se plantea en cuatro situaciones, primeramente una situación en la que el Administrador accede al panel principal de esta funcionalidad y a partir del cual, puede seleccionar a una tienda en específico para llevar a cabo la modificación de su estatus dentro del CCV o para visualizar un estado de cuenta específico. La modificación del estatus de la tienda involucra dos situaciones, una en donde el usuario Administrador visualiza el estatus actual y modifica el estatus de la tienda y una segunda situación en la que se confirman la realización satisfactoria de los cambios. La visualización de un estado de cuenta específico se describe en una situación en la que se despliega el estado de cuenta solicitado por el usuario Administrador. Por último, otra de las funcionalidades que se describe en este escenario es la del manejo del padrón de tiendas. Dentro de las situaciones que describen a esta funcionalidad se encuentran el registro y asociación de nuevas tiendas al CCV, la modificación del registro de las tiendas ya asociadas y la eliminación de tiendas.

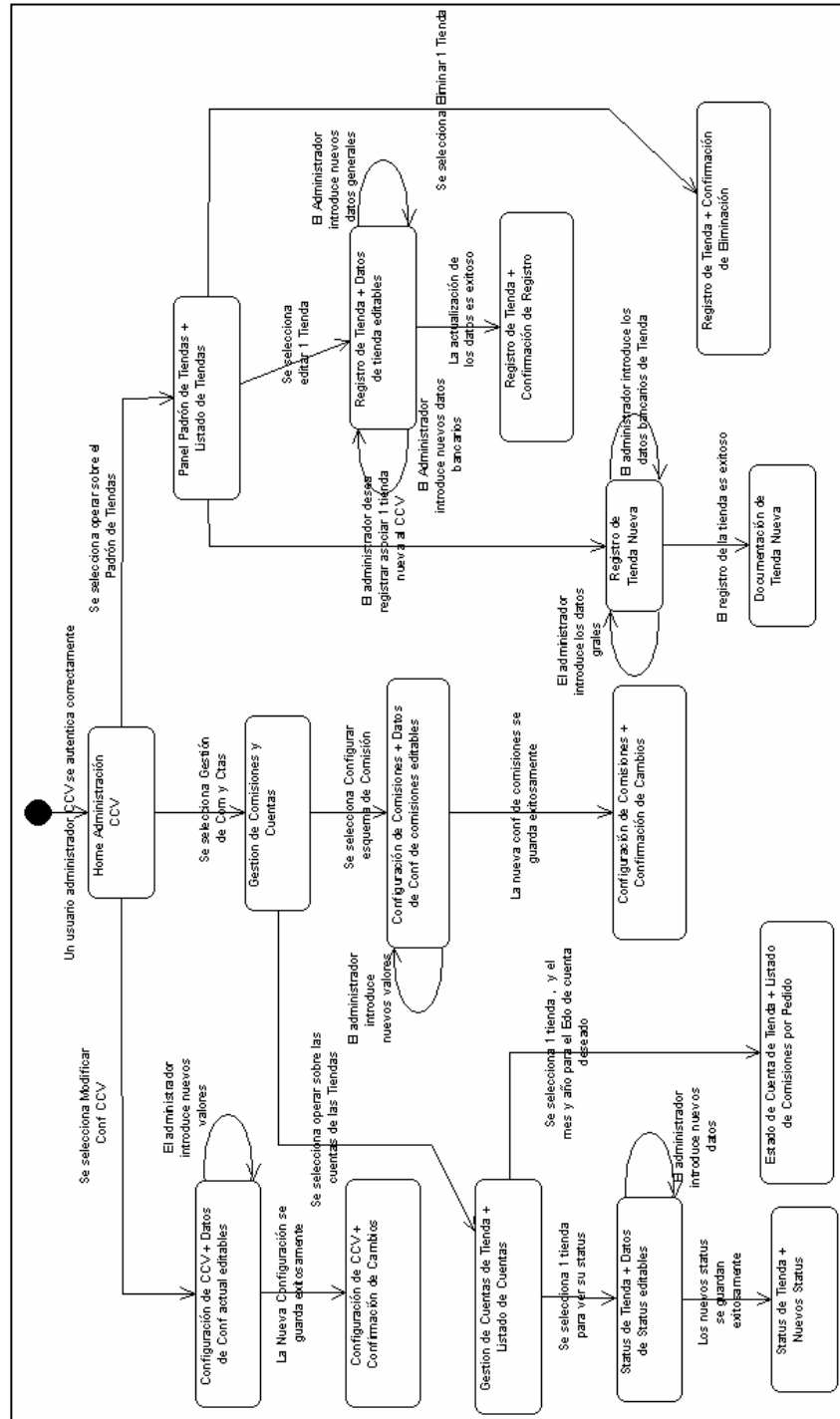


Figura 7.5: Escenario Web Administración CCV, 1ra parte

7.2.6 Escenario Web Administración CCV, 2da Parte

En esta segunda parte del escenario Web del subsistema de Administración de CCV, que se muestra en la figura 7.6, se describen el remanente de las situaciones asociadas a las funcionalidades provistas por el subsistema de Administración de CCV. Estas funcionalidades son la visualización de informes generales, la gestión de usuario y en el envío de correos electrónicos a usuarios registrados en el CCV. Una vez que un usuario del tipo Administrador accede por autenticación correcta al subsistema de Administración de CCV, la situación va a depender de la selección que haga el usuario Administrador en base a la funcionalidad que desea utilizar. Si el Administrador selecciona ver algún informe del CCV, entonces se carga el módulo correspondiente. El Administrador, en esta situación, puede seleccionar a una familia de informes para posteriormente seleccionar a un informe específico de dicho listado. Por otra parte, en la gestión de usuarios se presentan siete situaciones. La primera denota el acceso al panel principal de esta funcionalidad cuando un usuario Administrador selecciona trabajar con el padrón de usuarios. A partir de esta situación principal, y de acuerdo a la selección del usuario Administrador, se llevan a cabo las transiciones hacia las demás situaciones con las cuales se representan las tareas específicas de la gestión de usuarios. Existe una situación la presentación del listado de usuarios asociados a una tienda específica, para la confirmación de eliminación de un usuario, la edición de un usuario y la confirmación de cambios realizados. Por último, cuando estando en Home Administración CCV el Administrador selecciona acceder al panel de envío de correos, se realiza la transición hacia el panel correspondiente. Ya dentro de este panel, el Vendedor introduce los campos requeridos del correo electrónico y solicita su envío. Si el correo se envía correctamente se transita a una situación de confirmación.

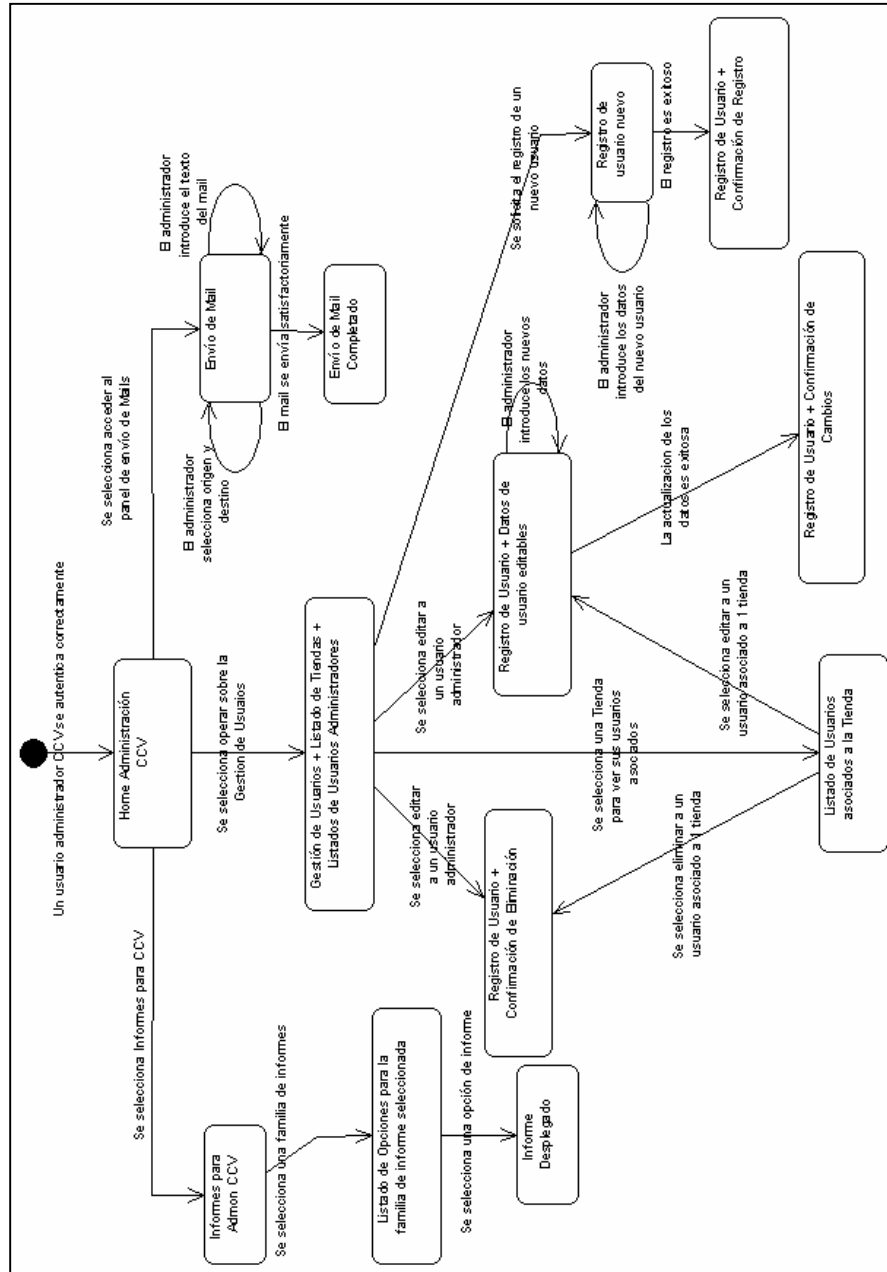


Figura 7.6: Escenario Web Administración CCV, 2da parte

Capítulo 8

Esquema de implementación

En este capítulo se presenta un único diagrama de componentes mediante el cual se proporciona una vista general del esquema de implementación que se propone para el CCV. En un primer apartado se describe la semántica asociada a los elementos de modelado que se utilizan en el diagrama. Posteriormente se muestra el diagrama de componentes y los comentarios correspondientes.

8.1 Semántica de elementos asociados al diagrama de componentes

Los elementos de modelado UML que aparecen en el diagrama de componentes que se presenta en este capítulo son: nodos, paquetes, componentes y asociaciones. Un nodo es un elemento físico que existe en el tiempo de ejecución y representa un recurso computacional disponible en la aplicación. Un nodo en el diagrama de componentes se representa con un cubo. Un nodo puede contener objetos y componentes que residen dentro de ese recurso de cómputo. Por otra parte, un componente representa una parte del sistema que proporciona la ejecución de una tarea o servicio específico y se representa como rectángulo con dos pestañas sobre el lado izquierdo. Por último, las asociaciones entre componentes se representan con una flecha. Cada asociación representa en el diagrama algún tipo de relación entre los componentes implicados. Además, estas asociaciones pueden enriquecer su semántica en el diagrama mediante la utilización de estereotipos que denoten con mayor claridad el tipo de relación que se establece entre los componentes y/o nodos [Koch02].

8.2 Esquema de Implementación del CCV

En el diagrama de componentes de la figura 8.1 se provee una vista genérica de la implementación del CCV. El diagrama se enfoca al CCV como elemento principal y las dependencias que sostiene con otros elementos requeridos para su correcto funcionamiento.

Esta propuesta de implementación se plantea con la utilización de tecnologías de uso frecuente en el ámbito del desarrollo de aplicaciones Web tales como PHP y MySQL. En el anteproyecto que se presenta en el próximo capítulo se hace referencia a ambas tecnologías. Como se puede observar la aplicación CCV reside en el nodo Web Server. La aplicación CCV, tal como se presenta en el modelo de navegación, consiste de una amplia colección de <<navigation class>>, entre otros elementos. Cuando se accede al CCV por medio de un navegador Web lo que realmente sucede es que se envía una solicitud al Web Server en donde se encuentra hospedada la aplicación. Cuando el Web Server recibe dicha solicitud, se inicia el proceso de ejecución para satisfacer dicha solicitud. El contenido dinámico debe ser interpretado por el Web Server y para ello se requiere, en este caso, de un módulo PHP. Con el soporte que provee este módulo se accede a la fuente de datos que reside en servidor dedicado a este fin. En este caso, se propone la utilización de un manejador de base de datos MySQL. Una vez que el código de servidor se interpreta, se accede a la base de datos para la ejecución de instrucciones SQL y se genera el contenido requerido, se envía al Web Browser del Cliente un código (HTML) que incluye tanto el contenido dinámico generado por el servidor, como el estático claramente interpretable por el Web Browser. Una vez que el Web Browser interpreta este código despliega como resultado la <<client page>> para que sea visualizada por el usuario [Conallen02]. Para algunas funcionalidades, específicamente aquellas que involucran la realización de pagos con tarjetas bancarias, el CCV también hace uso de un socket de seguridad que se establece con el protocolo SSL/TLS, por medio del cual se envía la información bancaria hacia la terminal bancaria y se obtiene la respuesta al intento de transacción.

Este esquema de implementación es el mas comúnmente usado en aplicaciones Web, especialmente en las de Comercio Electrónico. Al respecto de esto, en el próximo capítulo se presenta un anteproyecto de implementación en donde se refinan algunos aspectos asociados al esquema descrito en este diagrama.

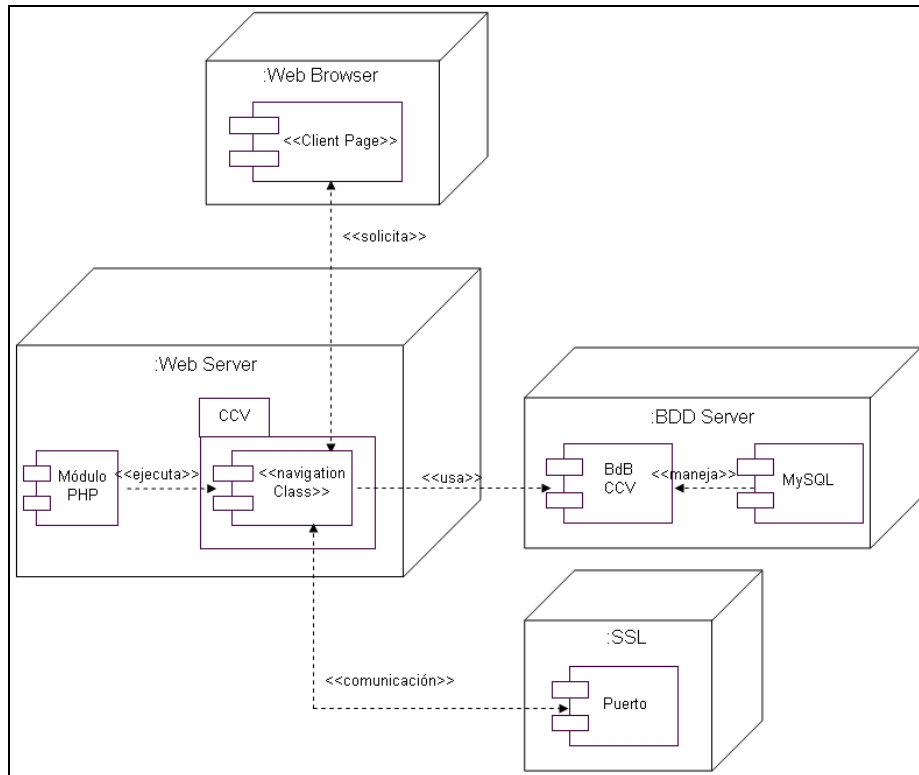


Figura 8.1: Diagrama de componentes, Esquema propuesto de implementaci3n para el CCV

Capítulo 9

Anteproyecto de Implementación

En este capítulo se presenta un anteproyecto de software de referencia para la implementación de este modelo de Centro Comercial Virtual. En este anteproyecto se incluyen propuestas a las cuestiones más elementales que se deben tomar en cuenta en el desarrollo de cualquier aplicación de software, y en este caso de las aplicaciones Web, tales como lo son: el equipo de trabajo, planificación de la implementación, las herramientas de desarrollo, el lenguaje de programación, etc.

9.1 El equipo de desarrollo

En el desarrollo de sistemas, no siempre es posible encontrar al personal idóneo que trabaje sobre un proyecto. Sin embargo, se puede reducir el riesgo de equivocarse en la selección de personal si en la planeación del proyecto se definen roles y perfiles del personal que debe conformar el equipo de trabajo. En este apartado se describen estos roles y funciones para el equipo de desarrollo propuesto para la implementación del modelo de CCV propuesto.

Para el desarrollo de este proyecto se propone un equipo de desarrollo integrado por personas que puedan desempeñar los siguientes roles: jefe de proyecto, arquitecto de software, desarrollador Web y un diseñador gráfico para Web [McConnell03]. A continuación se describen las actividades asociadas a cada uno de ellos.

- Jefe de Proyecto: El jefe de proyecto se propone como la figura clave en la planificación, ejecución y control del desarrollo de la aplicación proyecto y es quien ha de impulsar el avance del mismo mediante la toma de decisiones tendentes a la consecución de los objetivos. La principal función del jefe de proyectos se puede resumir en dirigir el equipo del que dispone para alcanzar los objetivos del proyecto. Más concretamente, podemos destacar las siguientes funciones específicas:
 - Planificación del proyecto en todos sus aspectos
 - Identificar las actividades a realizar por los demás miembros del equipo

- Identificar y poner a disposición del equipo de desarrollo los recursos necesarios para trabajar de manera óptima
 - Establecer canales de comunicación entre los miembros del equipo
 - Dirigir la integración de los resultados parciales generados por cada miembro del equipo
- Arquitecto de Software: El arquitecto de software es el encargado de proporcionar un control intelectual sobre la complejidad del sistema al permitir enfocar a los elementos esenciales y sus interacciones por sobre otros detalles. Cuando existe un modelo, como es el caso, el arquitecto de software es la figura encargada de dirimir las cuestiones que puedan presentarse en la interpretación del modelo de la aplicación y de supervisar que todas las características planteadas en el modelo sean correctamente implementadas.
- Desarrollador Web: Es el encargado de la implementación de rutinas, procedimientos y algoritmos mediante un lenguaje de programación Web que genere contenido dinámico y optimice la funcionalidad en una página Web. El desarrollo de páginas Web debe sujetarse al modelo de la aplicación y el desarrollador debe considerar la continua comprobación de los resultados con el arquitecto de software y con el jefe de proyecto.
- Diseñador Web: Es el encargado de diseñar visualmente la interfase de la aplicación Web. El objetivo principal de sus actividades es el de hacer atractiva y eficiente la parte visual de la aplicación mediante la correcta disposición de colores, formas y gráficos en la interfase. El diseñador Web, al igual que el desarrollador, debe sujetarse al modelo de la aplicación.

El número de individuos sugerido para la implementación del CCV es de, mínimo, cinco personas: un jefe de proyectos, un arquitecto de software, dos desarrolladores Web y un diseñador Web.

9.2 Planificación preliminar de la implementación

La construcción de un sistema de software es más que la escritura de código, pero definitivamente si es la parte más importante de los flujos de trabajo del equipo de implementación. En la práctica, los flujos de trabajo de la implementación deben tomar los artefactos del modelado y aplicar las herramientas de desarrollo del software a ellos. Las herramientas son típicamente editores y compiladores, aunque en el desarrollo de aplicaciones que operan en los ambientes Web pueden requerir hacer uso de otras herramientas y marcos que trabajo. Lo que se presenta a continuación es una propuesta acerca de la forma en que se debe desarrollar el CCV en base al modelo presentado.

En la figura 9.1 se describen las principales actividades respecto al proceso de desarrollo del CCV y el personal que realiza dichas actividades.

Ahora bien, en lo que respecta exclusivamente a la programación y diseño, se sugiere que se planifiquen las actividades en base a la secuencia que se muestra en la figura 9.2. En una primera instancia, se sugiere la definición de los criterios que tienen que ver con la apariencia de la aplicación y, de manera paralela, llevar a cabo el diseño y creación de la base de datos. Posteriormente, se debe proceder a instalar y configurar los servidores requeridos para el desarrollo de la aplicación. A la par, en lo que respecta al diseño, se debe desarrollar la colección de plantillas, hojas de estilos y gráficos. A continuación se debe proceder a programar el subsistema de componentes comunes, se integran todos los elementos de diseño y se publica un glosario con todas las especificaciones de diseño, como lo son, colores, tipos de letra, tamaños, etc. El subsistema de componentes comunes es el primero que debe programarse en virtud que los componentes definidos en este subsistema puedan ser utilizados por los demás subsistemas. El siguiente paso es la integración de los elementos de diseño con los funcionales y la creación del árbol de navegación. Hecho esto se tiene un primer prototipo para el CCV, el cual consiste de la estructura primaria de toda la aplicación sin ningún tipo de contenido dinámico. A partir de este punto, el diseño pasa a segundo término, y el trabajo pesado recae en la programación de todas las funcionalidades. Lo primero en programarse es el módulo de autenticación de usuario, ya que este es utilizado por cada uno de los subsistemas. El trabajo, a partir de este punto, se refiere a la programación del subsistema de Administración del CCV, el Gestor de Tienda, la ePlaza y el Gestor de Transacciones. Cabe acotar que el desarrollo de cada subsistema puede ser llevado a cabo de manera paralela si se cuenta con el número de desarrolladores importante, en virtud que son subsistemas dedicados a realizar un número específico de funcionalidades independientes entre ellas. Se sugiere la evaluación del prototipo cuando se haya completado al menos dos subsistemas y una evaluación al final cuando se hayan programado todos. Además, como se mencionó anteriormente, se requiere la calendarización y supervisión de las actividades que se realicen por parte del jefe de proyectos, así como también una colaboración estrecha del arquitecto de software con los desarrolladores y diseñadores a fin de respetar las especificaciones y los lineamientos definidos en el modelo.

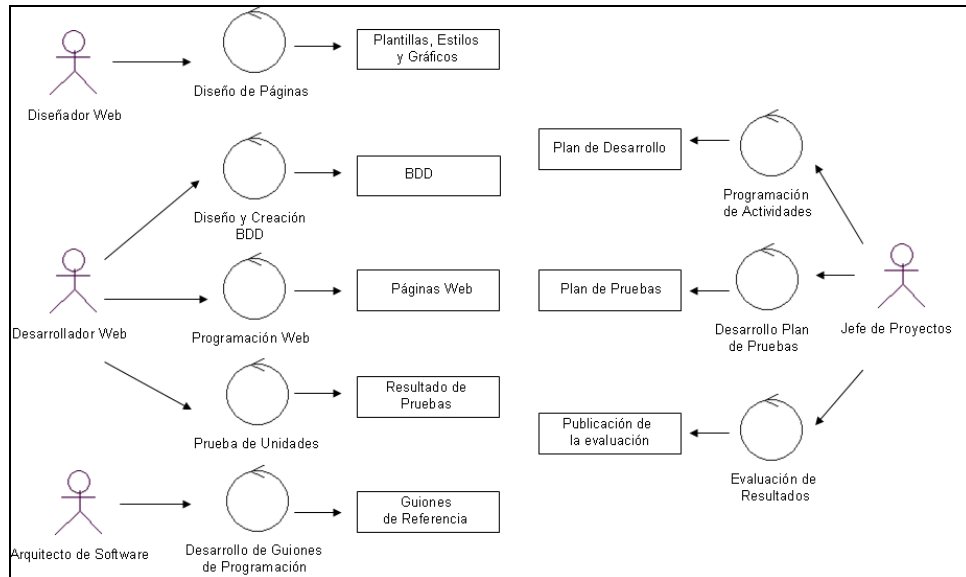


Figura 9.1: Actividades en el proceso de desarrollo de CCV

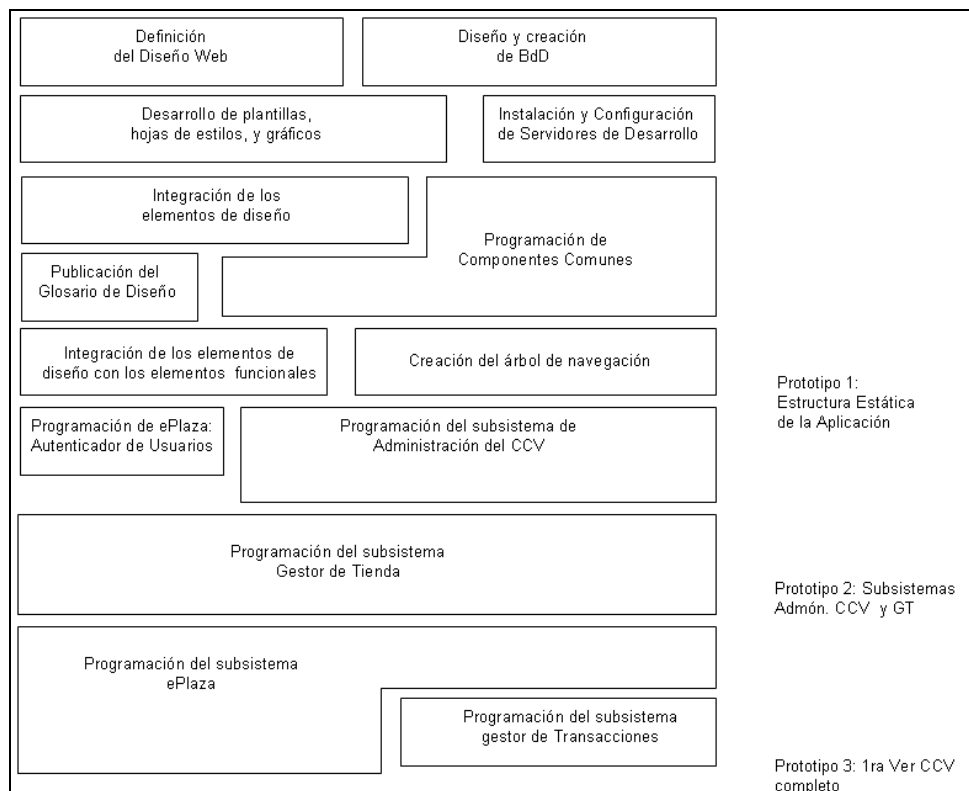


Figura 9.2: Planificación de Actividades en el proceso de desarrollo

9.3 Sugerencia de Lenguaje de programación, servidor de base de datos y servidor Web

El proceso de desarrollo de una aplicación Web, en lo que respecta a la implementación, depende en mucho de la correcta elección de estos tres elementos inherentes a cualquier aplicación Web. En este apartado se presenta una propuesta para cada uno de ellos la cual se plantea como la alternativa adecuada para el CCV. Cabe acotar que la sugerencia se realiza con base a la experiencia personal en el desarrollo de aplicaciones web y no en un análisis comparativo con otras suites de desarrollo.

9.3.1 PHP

PHP significa PHP: Pre-procesador de Hipertexto (PHP: Hypertext Preprocessor). PHP es un lenguaje de scripting embebido en HTML. Mucha de su sintaxis es tomada de C, Java y Perl con un par de características adicionales únicas y específicas de PHP. El propósito del lenguaje es permitir que los desarrolladores Web escriban páginas generadas dinámicamente con rapidez. Existen diferentes versiones de PHP. PHP 4 es la generación actual más estable de PHP, aunque ya se encuentra disponible la versión 5. PHP ha sido diseñado, desde su origen, para ser un lenguaje más seguro para escribir programas, a diferencia de Perl o C y, con la correcta selección de las opciones de configuración, del servidor de base de datos y el servidor Web, es posible conseguir una combinación equilibrada de libertad y seguridad que requieren las aplicaciones que operan en un ambiente naturalmente inseguro como lo es la Web.

La característica primaria de PHP es la generación dinámica de páginas. Pero en si, existen un par de razones muy importantes que justifican la popularidad y éxito de este lenguaje. La primera es el soporte de PHP para operar con una gran variedad de manejadores de bases de datos, como por ejemplo MySQL, Interbase, Oracle, PostgreSQL, Informix, entre muchos otras. La segunda es que PHP es multiplataforma, ya que funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server o Apache) de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra. Además de esto, PHP es libre, por lo que es un lenguaje de fácil acceso para todos. Esta información y más se encuentra disponible en la pagina oficial www.php.net. Estas características han convertido a PHP en el lenguaje más utilizado actualmente en el desarrollo de aplicaciones Web, de ahí que se proponga su utilización en la implementación del CCV.

9.3.2 MySQL

MySQL es un sistema de administración de bases de datos (SQL) desarrollado también bajo la filosofía de código abierto y es uno de los más populares en el desarrollo de aplicaciones Web. Entre las características disponibles en las últimas versiones (la versión 4 y la versión 5) se pueden destacar las siguientes: capacidad de manejar un amplio subconjunto del lenguaje SQL estándar, disponibilidad en gran cantidad de plataformas y sistemas, proporciona conectividad confiable y segura, permite la ejecución de transacciones y el manejo de claves foráneas, permite replicación y provee la capacidad de realizar búsquedas e indexaciones de campos de texto, entre otras características más. Además, MySQL es el servidor de base de datos más comúnmente utilizado con el lenguaje PHP. Es por esta robustez que, según las cifras del fabricante (MySQL AB), existen más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos. Esta información y más se encuentra disponible en la página oficial dev.mysql.com. Estas características de MySQL la hacen una excelente alternativa para que se utilice en la implementación del CCV.

9.3.3 Apache

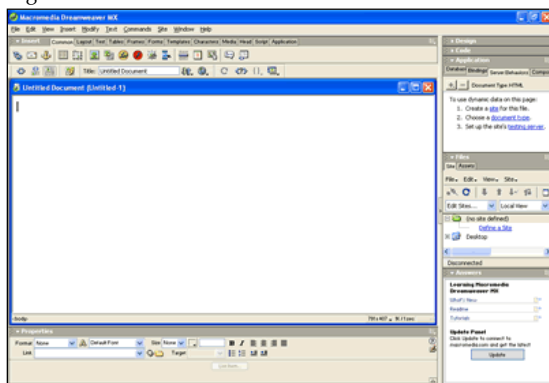
El Apache es un servidor Web código abierto multiplataforma. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches aplicados al servidor de NCSA (National Center for Supercomputing Applications, Centro Nacional de Aplicaciones de Supercómputo). Era, en inglés, “a patchy server” (un servidor parchado). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta, entre muchas otras, las siguientes características: utiliza mensajes de error altamente configurables, maneja bases de datos de autenticación, sus módulos se cargan de manera dinámica, soporta una gran cantidad de lenguajes (entre ellos PHP) y tiene un alto desempeño bajo concurrencia de operaciones. Además, en sus versiones 2.x, el núcleo de Apache tiene varias mejoras con respecto al núcleo de apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6. Esta información se encuentra disponible en la página oficial de Apache www.apache.org. En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft). Debido a esto, y su excelente desempeño cuando se utiliza juntamente con PHP y MySQL, se sugiere utilizar el servidor Web Apache en la implementación del CCV.

9.4 Herramientas para la implementación de aplicaciones Web

El siguiente paso, una vez definidas las tecnologías a utilizar en la implementación de una aplicación Web, es seleccionar las herramientas más adecuadas para desarrollar la aplicación de la mejor manera posible. Es por eso que a continuación se comentan algunas sugerencias de herramientas útiles en el desarrollo de aplicaciones Web, enfocadas en el CCV.

9.4.1 Editor de Páginas Web: Macromedia Dreamweaver

Figura 9.3: Interfase Macromedia Dreamweaver



En la implementación del CCV, el uso de esta herramienta se propone para el diseñador Web y de los desarrolladores Web. Dreamweaver es la herramienta más utilizada en el sector del diseño y la programación Web, por sus funcionalidades, su integración con otras herramientas y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Esta herramienta cumple perfectamente el objetivo de

diseñar páginas con aspecto profesional de gran funcionalidad, y soporta gran cantidad de tecnologías como lo son las hojas de estilo, capas, DHTML, JavaScript, XML, ASP, PHP, entre varias más. La gran fortaleza de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que las rutinas de este programa (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en JavaScript-C lo que le ofrece una gran flexibilidad. Esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de JavaScript que hace que sea un programa muy fluido, y que permite que desarrolladores y editores Web puedan hacer extensiones para el programa. Además de estas características, tiene las funciones típicas de un editor de código fuente para la Web: un administrador de sitios para agrupar los archivos según el proyecto al que pertenezcan, un cliente FTP integrado que permite subir los archivos editados inmediatamente al sitio en Internet y la función de autocompletar y resaltado de la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP. La versión libre de este software es el D4L y funciona sobre plataforma Linux.

9.4.2 Editor de Gráficos digitales: Adobe Photoshop

En la implementación del CCV, el uso de esta herramienta se propone exclusivamente para el diseñador Web. Photoshop es un programa de edición y retoque de imágenes elaborada por la compañía de software Adobe, inicialmente para computadores Apple pero, en la actualidad, también para plataformas PC. Photoshop se ha convertido, casi desde sus comienzos, en el estándar mundial en retoque fotográfico, pero también se usa extensivamente en multitud de disciplinas del campo del diseño y fotografía, como diseño Web, composición de imágenes, estilismo digital, y básicamente en cualquier actividad que requiera el tratamiento de imágenes digitales. Consta de muchas facilidades para este tipo de tareas, tales como el soporte para tabletas gráficas, una amplia paleta de pinceles disponibles (así como la posibilidad de la edición de estos teniendo en cuenta múltiples parámetros: presión, rotación, dispersión, etc.), paletas de colores, rejillas auxiliares de orientación, etc. La versión mas reciente de este programa es la 9, la cual ha sido bautizada como Photoshop CS 2. Su utilización en la implementación del CCV se restringe a la edición de imágenes por parte del diseñador Web.

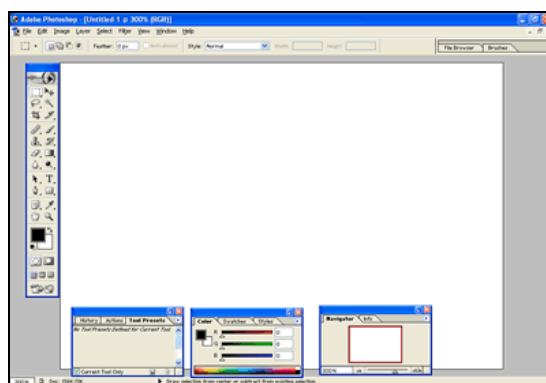


Figura 9.4: Interfase Adobe Photoshop

9.4.3 Editor UML: Rational Rose

En la implementación del CCV, el uso de una herramienta de este tipo se propone exclusivamente para el Arquitecto de Software. Rational Rose es una herramienta CASE que tiene como eje de su funcionamiento el Proceso Unificado de Desarrollo de Software, sin embargo, en virtud que el modelado de este CCV no se realizó estrictamente bajo esta metodología su uso en el proceso de implementación se restringe como un editor de UML para que el Arquitecto de Software pueda tener acceso al modelo del CCV que aquí se presenta y, de esta forma, poder supervisar la correcta implementación de las especificaciones del modelo. Ahora bien, puede utilizar otras herramientas de edición de UML como lo son Visual Paradigm, UML Studio, entre otros.

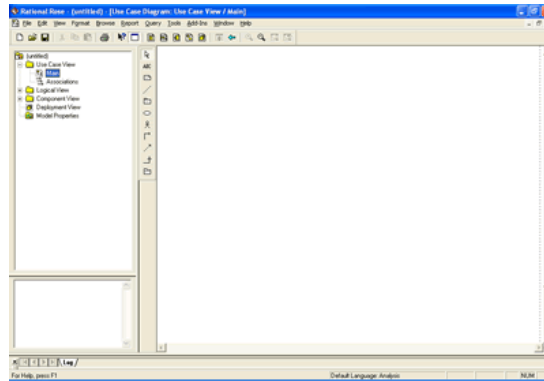
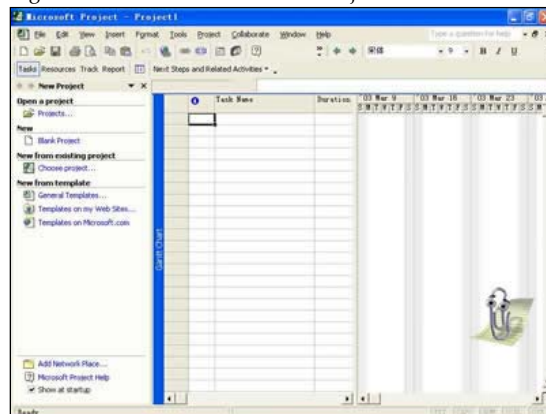


Figura 9.5: Interfase de Rational Rose

9.4.4 Software para la Administración de Proyectos: MSP

Figura 9.6: Interfase de Microsoft Project



En la implementación del CCV, el uso de esta herramienta se propone exclusivamente para el jefe de proyecto. El Microsoft Project (o MSP) es un programa del software de administración de proyectos desarrollado por Microsoft que se está orientado para auxiliar a administradores de proyectos de software en el desarrollo de planes, asignación de recursos y de personal a tareas, calendarización de tareas y procesos, seguimiento a procesos,

administración de presupuestos y análisis de cargas de trabajo. Además, Project puede reconocer a diversas clases de usuarios. Estas diversas clases de usuarios pueden tener diferentes niveles de acceso a los proyectos, a las vistas y a otros datos. La herramienta también permite administrar recursos y calcular costos de trabajo en las tareas que se programan. En la versión más reciente la herramienta ya da la posibilidad de poder administrar un proyecto a través de Internet. La utilización de esta herramienta en la implementación del CCV se restringe a la posibilidad de proporcionar al jefe de proyecto un instrumento para ejercer un mejor control y una mejor organización de las tareas que deben realizarse a lo largo del desarrollo del CCV.

9.5 Mecanismo de Prueba del Modelo

La mejor prueba de validez del modelado aquí expuesto es la construcción de un prototipo que se base en las especificaciones descritas a través de los diferentes diagramas. La construcción de ese prototipo es una tarea ardua que se delega a un trabajo futuro, sin embargo, es posible hacer una correspondencia entre los diferentes diagramas para proporcionar una idea de que es lo que se debe obtener como producto final. Es por eso que en este apartado se presenta el mapeo de una funcionalidad del CCV a través de la colección de diagramas que se presentan en este documento y la correspondencia de estos a lo que se desea obtener en la implementación. La funcionalidad que se toma como Caso de Ejemplo (CdE) es la de la compra de productos en una tienda.

En los siguientes diagramas se resaltan con marcas de tono a los elementos de modelado que en cada diagrama involucra al caso de ejemplo. Al final de este apartado, a partir de la colección de todos estos elementos resaltados, se presenta lo que en la práctica se desea obtener en la implementación.

9.5.1 El CdE en los diagramas de caso de uso

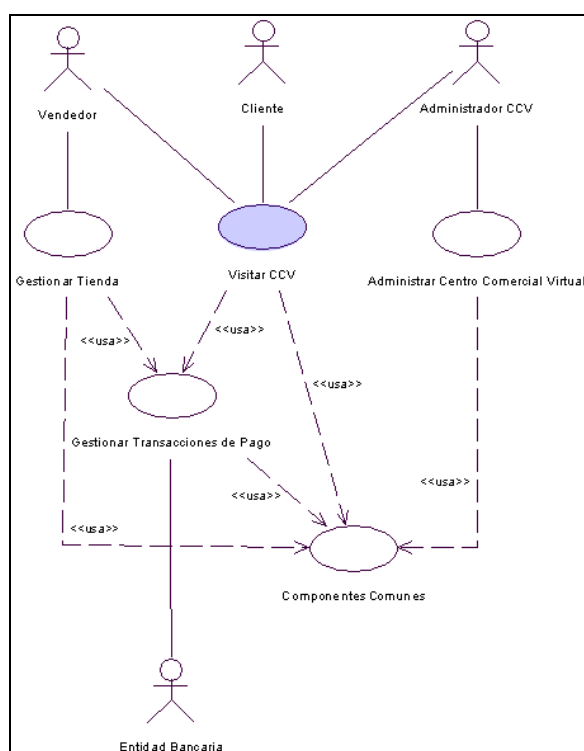


Fig 9.7: El CdE en el diagrama de casos de uso General

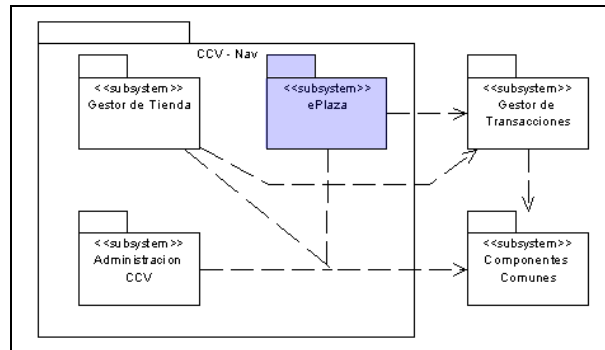


Fig 9.8: El CdE en la vista de subsistemas

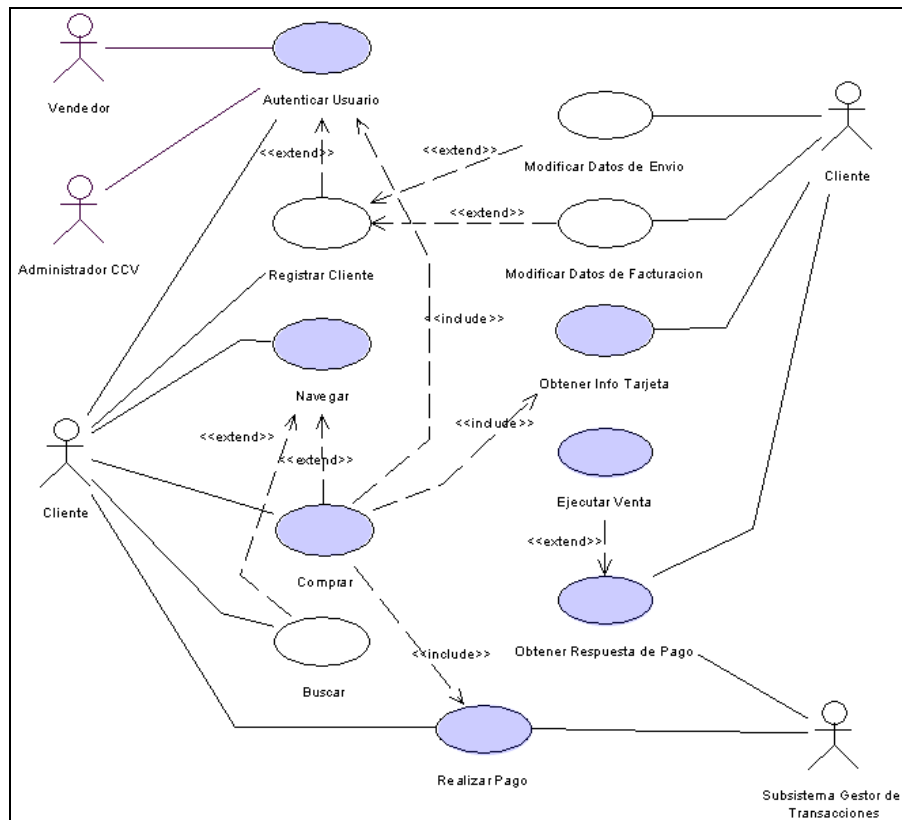


Fig 9.9: El CdE en el diagrama de casos de uso para el subsistema ePlaza

9.5.2 El CdE en el modelo lógico-conceptual

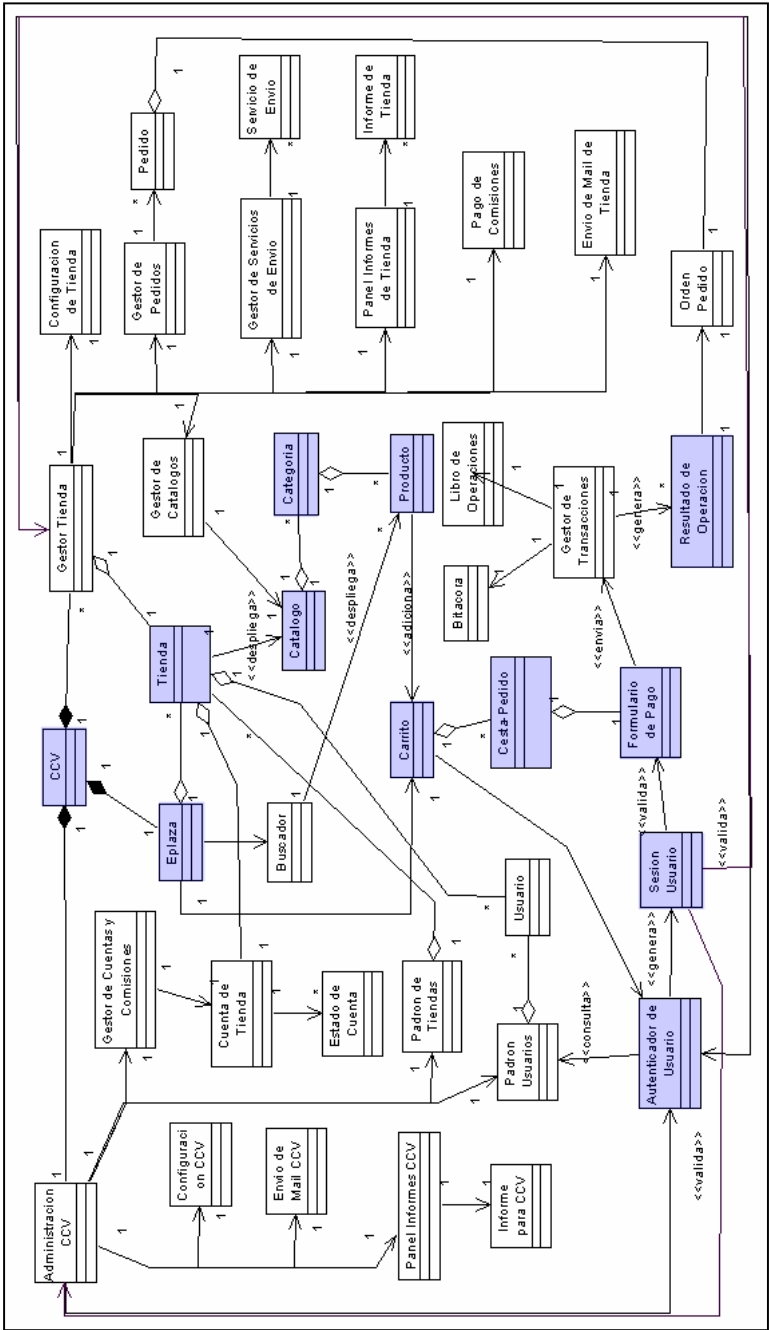


Fig 9.10: El CdE en el modelo lógico-conceptual

[illegible]

Fig 9.11: El CdE en el modelo de navegación

Fig 9.12: El CdE en la secuencia de adición de producto a carrito

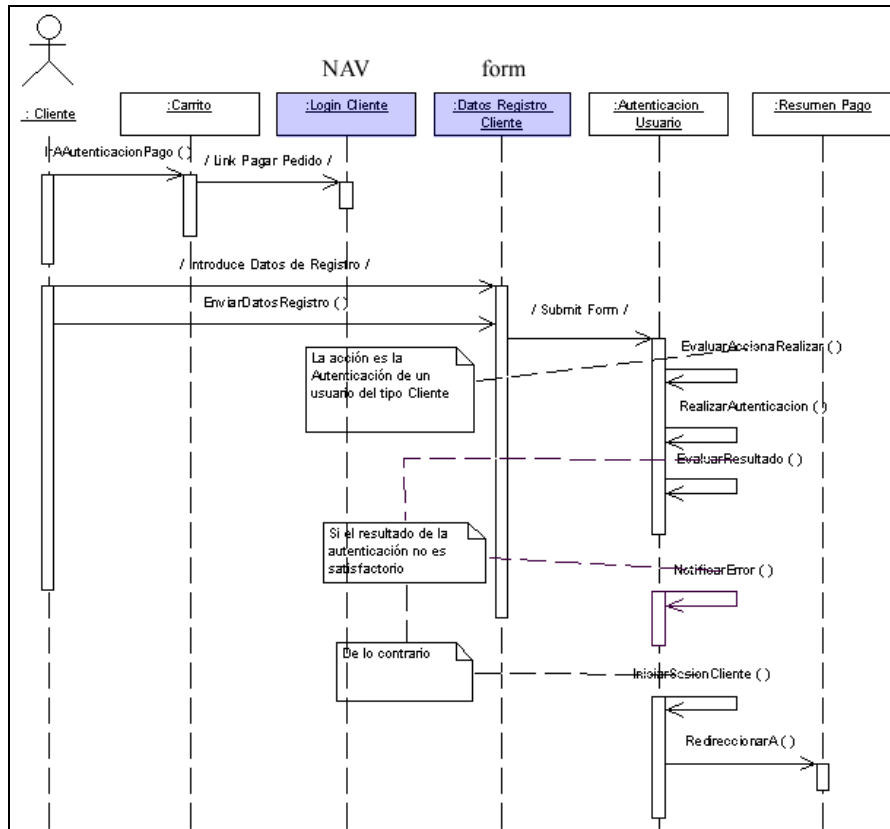


Fig 9.13: El CdE en la secuencia de autenticación de Cliente existente

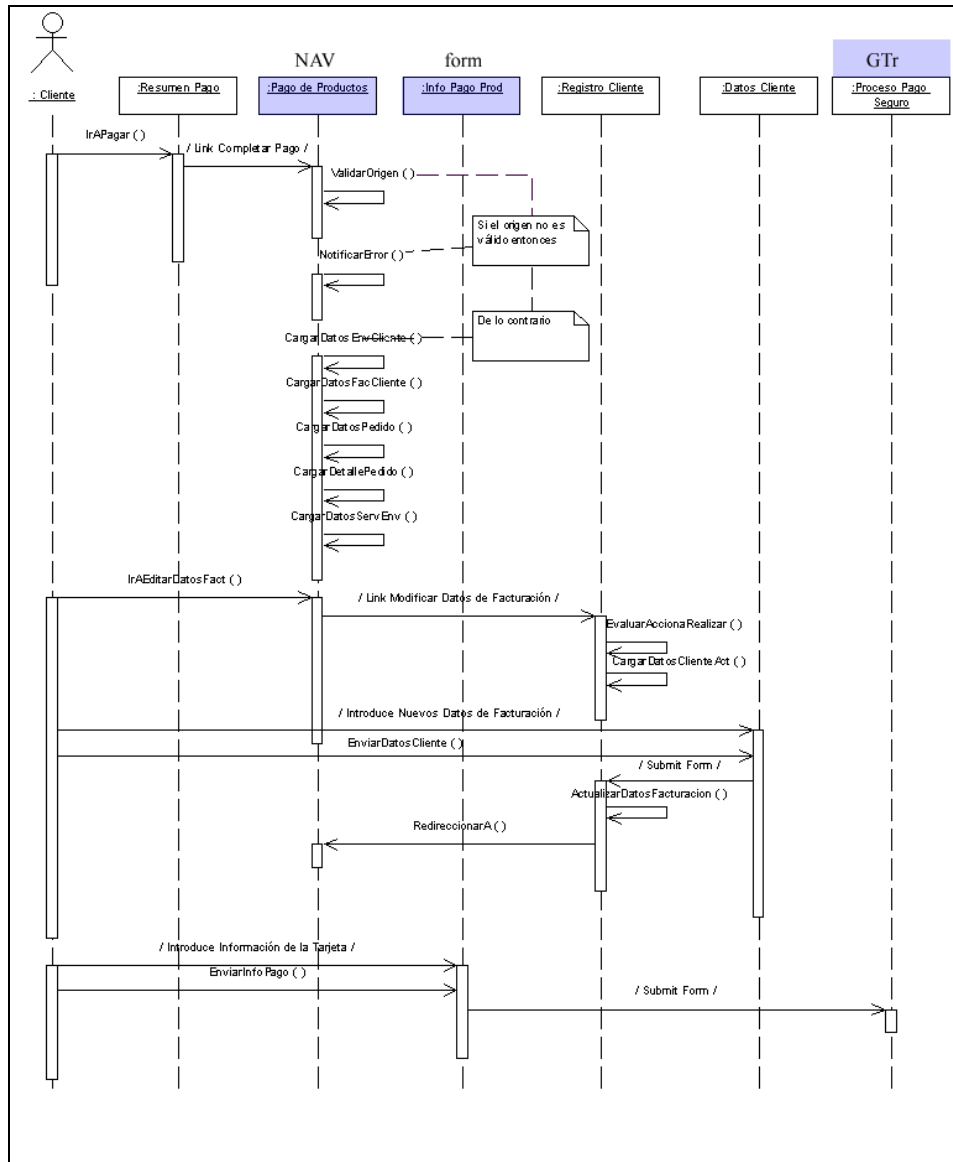


Fig 9.15: El CdE en la secuencia de Captura de datos de Tarjeta

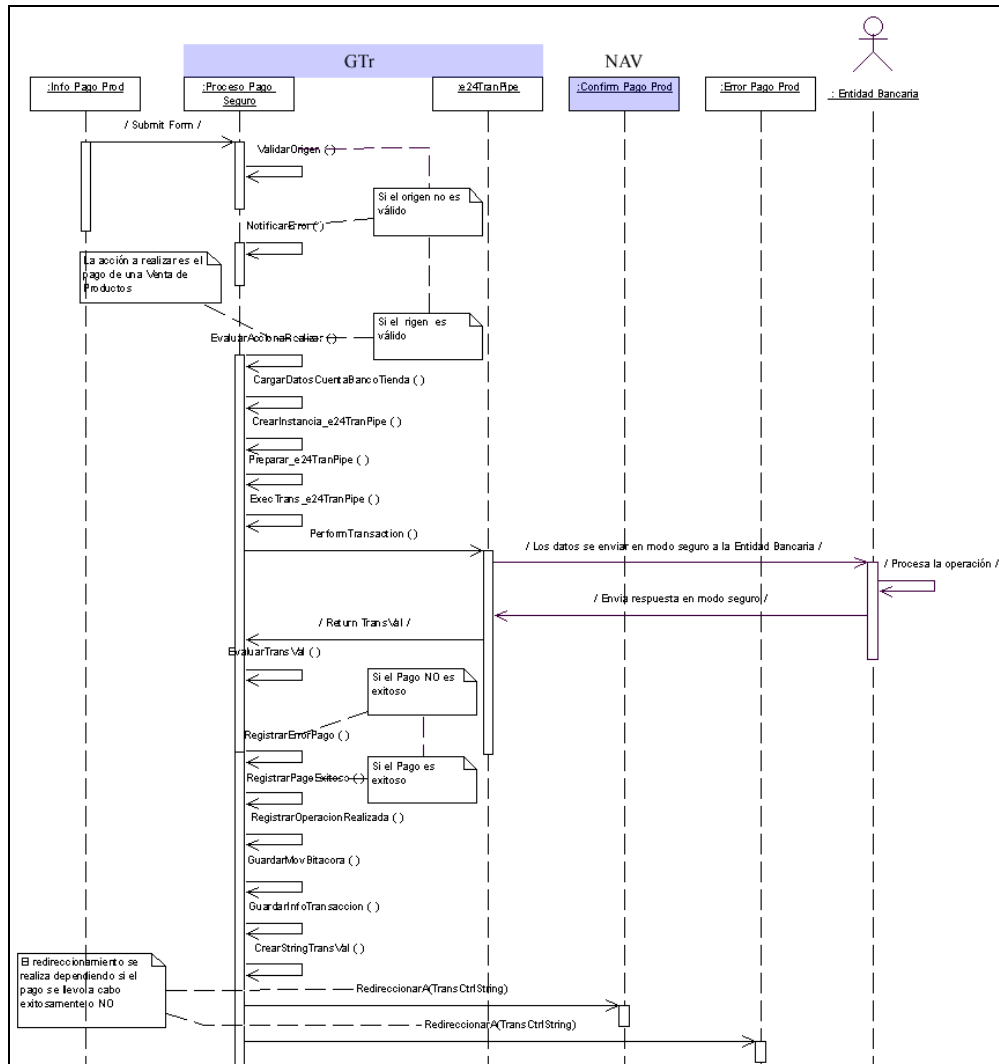


Fig 9.16: El CdE en la secuencia de Procesamiento de Pago seguro de Productos

[illegible]

9.5.6 El CdE en la práctica

En la figura 9.18 se observa la representación de la clase de navegación principal del CCV: el Home CCV. En la figura se pueden apreciar el formulario de búsqueda, el índice de tiendas y demás elementos descritos en la clase de navegación Home CCV. En este caso de ejemplo se asume que el Cliente selecciona a la tienda Libros +D.

Logo	Nombre	Descripción	
	Sport Shoes	Venta de Zapato deportivo	Ir a Tienda
	DVD Anime Online	Venta de DVD's de anime	Ir a Tienda
	Artesanias Cruz	Comercializacion de artesanias	Ir a Tienda
	Libros +D	Venta de libros: De novelas y Novedades	Ir a Tienda
	Enigma Cards	Venta de Cards coleccionables	Ir a Tienda

Fig 9.18: Representación del Home CCV

En la figura 9.19 se observa la representación de la clase de navegación del Home Tienda. En la figura se pueden apreciar el formulario de búsqueda que es incluido nuevamente, la descripción de la tienda, el logotipo, el nombre y demás elementos descritos en la clase de navegación Home CCV. Mención especial se hace de la inclusión del índice que describe el catálogo. En este caso de ejemplo se asume que el Cliente selecciona a la categoría novedades. Al hacer eso, se despliega el detalle de la categoría, es decir, el listado de productos que aparece en la figura 9.20. En este índice se asume que el Cliente seleccione ver el detalle del producto "Guardián de los arcanos"

Home CCV

Libros +D

[Home CCV](#)
[Ver Carrito](#)
 Palabra:
 Buscar en: En Tienda

Libros +D pone al alcance de nuestros clientes una gran variedad de novelas y libros en general de novedad. En nuestra tienda encontrarás siempre los más recientes exitos de la literatura en español. Le garantizamos que lo que usted compre en Libros +D le deleitara su intelecto y satisfacerá su gusto por la narrativa. Sea usted bienvenido, que tenga usted una excelente lecturaiiii.

Catálogo

[Inicio>](#)

	Categoria	# Prod	
1	Novedades	(5)	Ver Productos
2	Clásicos	(50)	Ver Productos
3	Terror	(15)	Ver Productos
4	Viajeros de la Palabra	(45)	Ver Productos
6	Serie La dichosa Palabra	(18)	Ver Productos
7	Lo que + lee México	(60)	Ver Productos

Fig 9.19: Representación del Home Tienda

Categoria				
Inicio> Novedades>				
Ing	Producto	Precio	Descripción	
232	 Round de Sombra	\$ 200.00	La mas reciente Obra de Abraham Cruz Villegas, el texto es rico en cuestionamientos acerca de la vida misma.	
134	 Poesia para nada	\$ 50.00	Ignacio Sánchez presenta esta obra al mundo actual desde un punto de vista muy personal.	
567	 La mitad Oscura	\$ 149.00	Un asesinato.. una historia.	
231	 Guardian de los Arcanos	\$ 139.00	Una historia que gira en torno a la revuelta del pueblo hebreo contra los nazis.	
456	 La magia del perdón	\$ 99.00	Novela acerca del perdón como mejor medicina ante el duelo.	

Fig 9.20: Despliegue del index Categoría

A continuación en la figura 9.21 se presenta la clase de navegación para el detalle del producto seleccionado por el Cliente. En la figura se pueden apreciar los elementos descritos por la clase de navegación. Entre estos elementos se encuentra el formulario Agregar Producto a Carrito que contiene al botón correspondiente para esta tarea. Cuando el Cliente presiona este botón, el producto es enviado al carrito para continuar con el proceso de compra.

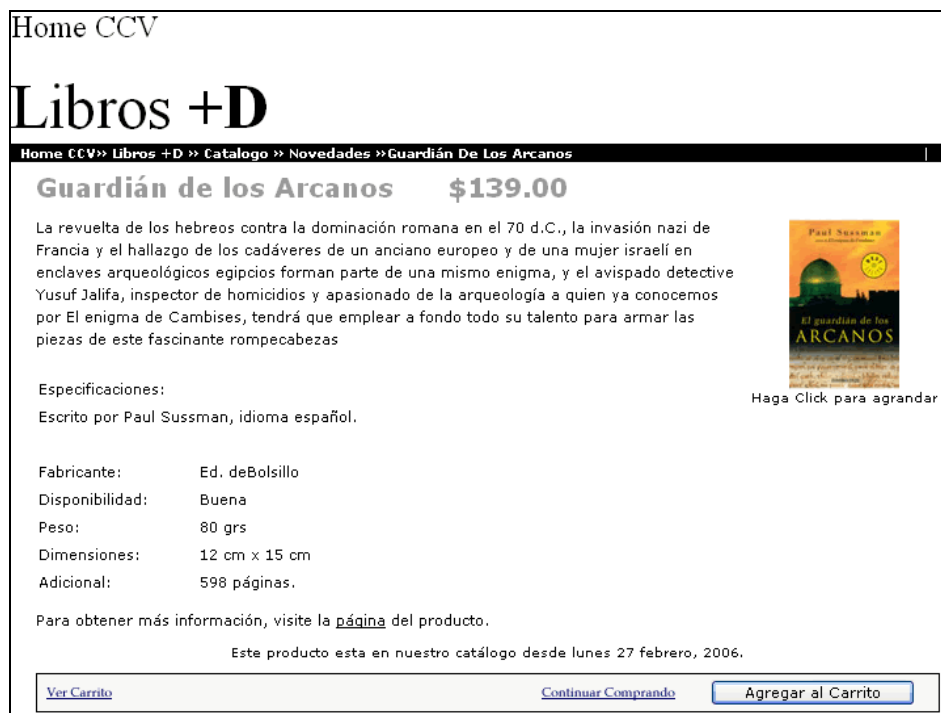


Fig 9.21: Representación del Detalle del Producto

En la figura 9.22 se presenta la clase de navegación para la clase de navegación Carrito. Esta clase de navegación, no obstante la apariencia, es a lo sumo compleja y, en la programación, requerirá de técnicas de programación complejas para su correcto funcionamiento. En general, el resultado de este trabajo de programación, en la práctica, deberá ser cercano a lo que se presenta en la figura 9.22. En la figura se puede destacar la manera en como se implementa el listado de productos. A cada elemento de este listado se asocia un formulario con los campos requeridos para llevar a cabo la actualización de cada línea. Con el propósito de visualizar el listado de productos del carrito, para el caso de ejemplo, se asume que además del producto “Guardián de los arcanos” el Cliente también agregó al carrito el producto “Poesía para nada”. Cuando el Cliente elige pasar a pagar se carga la clase de navegación Login cliente que se muestra en la figura 9.23.

Home CCV

Libros +D

Home CCV » Libros +D » Carrito

Pedido: 25631

Carrito

Eliminar Producto(s)	Cantidad	Precio	Importe
☐ Guardián de los Arcano La mas reciente Obra de Abraham Cruz Villegas, el texto es rico en cuestionamientos acerca de la vida misma.	1	Actualizar	\$139.00 \$139.00
☐ Poesía para nada Ignacio Sánchez presenta esta obra al mundo actual desde un punto de vista muy personal.	1	Actualizar	\$50.00 \$50.00

Artículos: 2

Subtotal: \$189.00
IVA: \$28.35
Total Pedido: \$217.35

[Continuar Comprando](#)
[Pasará a Pagar](#)

Fig 9.22: Representación del Carrito

En la figura 9.23 se presenta la clase de navegación para el logeo del Cliente. En el caso ideal el Cliente ya esta registrado por lo que introduce sus datos de acceso al formulario y presiona el botón enviar. Los datos son enviados a la clase de navegación Autenticar Usuario que es transparente al usuario, es decir, que dicha clase de navegación realiza el procesamiento de verificación los datos de acceso del Cliente y, de acuerdo al resultado de dicho proceso, redirecciona de manera automática hacia la clase de navegación correspondiente sin estacionarse en el navegador para que el Cliente interactué con ella. Este proceso se puede apreciar en la secuencia que describe la autenticación de un cliente existente. En el caso de ejemplo vamos a suponer que el Cliente se autentica correctamente y, de esta manera, el proceso de pago se inicia correctamente en la clase de navegación Resumen de Pago que se presenta en la figura 9.24.

The screenshot shows a web page for 'Libros +D'. At the top, it says 'Home CCV' and 'Libros +D'. Below this is a navigation bar with 'Inicio >> Libros +D >> Login'. The main heading is 'Login Cliente'. The page is divided into two sections: 'Cliente Existente' and 'Soy Nuevo Cliente'. The 'Cliente Existente' section has a message 'He comprado otras veces.' and two input fields for 'E-Mail:' and 'Contraseña:'. Below these is a link 'Recordar Datos de Acceso: Recordar'. The 'Soy Nuevo Cliente' section has a link 'Cliente Nuevo'. At the bottom right of the form area is an 'Enviar' button.

Home CCV

Libros +D

Inicio >> Libros +D >> Login

Login Cliente

Cliente Existente

He comprado otras veces.

E-Mail:

Contraseña:

Recordar Datos de Acceso: [Recordar](#)

Soy Nuevo Cliente: [Cliente Nuevo](#)

Fig 9.23: Representación del Carrito

En la figura 9.24 se presenta la clase de navegación Resumen Pago. En ella se puede observar los elementos descritos en el modelo. Entre estos elementos destacamos los datos de envío del producto y la posibilidad que tiene el Cliente de modificar estos datos. Se muestra también los datos del pedido y el detalle de los productos incluidos en él. Por último, se destaca el formulario para la selección del servicio de envío por parte del Cliente. Una vez que el Cliente ha seleccionado un servicio de envío de entre los disponibles, entonces puede continuar con el proceso de pago que prosigue con la obtención de los datos de la tarjeta bancaria.

Home CCV

Libros +D

Inicio » Libros +D » Resumen Pago

Resumen de Pago: Datos del Envío

Dirección de Entrega:

Jesús Ocaña Zúñiga
Rio Coatan 18
Los Laguitos
Tuxtla Gutiérrez Chiapas, México
C.P. 29028

[Editar Dirección de Envío](#)

Detalle Pedido: 25631

Cant	Producto(s)	Precio	Importe
1	Guardián de los Arcano La mas reciente Obra de Abraham Cruz Villegas, el texto es rico en cuestionamientos acerca de la vida misma.	\$139.00	\$139.00
1	Poesía para nada Ignacio Sánchez presenta esta obra al mundo actual desde un punto de vista muy personal.	\$50.00	\$50.00
	Servicio de Envío	\$0.00	\$0.00

Artículos: 2

Subtotal: \$189.00

IVA: \$28.35

Total Pedido: \$217.35

Servicio de Envío:

Servicios de Envío Disponibles:
-Seleccionar Servicio de Envío -

Comentarios de Envío :
-
Escriba sus comentarios
-

Costo: \$0.00

Agregar

[Pasarse a Pagar](#)

Fig 9.24: Representación de Resumen Pago

En la figura 9.25 se presenta la clase de navegación Pago de Productos. En esta clase de navegación el Cliente debe realizar el pago de los productos a través de su tarjeta bancaria. Antes, el Cliente tiene la posibilidad de modificar los datos de facturación de su registro. En esta clase de navegación se vuelven a desplegar los datos del pedido y el detalle de los productos incluidos en él. En esta clase también se incluye el costo del servicio de envío seleccionado en la fase previa. Por último, se destaca el formulario mediante el cual el Cliente introduce la información de la tarjeta bancaria con la cual desea realizar el pago del pedido. Una vez que el Cliente ha llenado los campos requeridos, procede a intentar el pago.

Home CCV

Libros +D

Inicio » Libros +D » Información de Pago

Informacion de Pago: Datos de Facturación

Datos de Facturación:

Jesús Ocaña Zúñiga
 Río Coatan 18
 Los Laguitos
 Tuxtla Gutiérrez Chiapas, México
 C.P. 29028

[Editar Datos de Facturación](#)

Detalle Pedido: 25631

Cant	Producto(s)	Precio	Importe
1	Guardián de los Arcano La mas reciente Obra de Abraham Cruz Villegas, el texto es rico en cuestionamientos acerca de la vida misma.	\$139.00	\$139.00
1	Poesía para nada Ignacio Sánchez presenta esta obra al mundo actual desde un punto de vista muy personal.	\$50.00	\$50.00
	Servicio de Envío: Nacional Terrestre-CajaMediana	\$50.00	\$50.00
Artículos: 2			

Subtotal: \$239.00
 IVA: \$35.85
Total Pedido: \$274.85

Datos Tarjeta de Credito

Titular Tarjeta:
 Núm. Tarjeta:
 Mes/Año Caducidad:
 Tipo:
 Monto a Cobrar: \$ 274.85

Pagar

Fig 9.25: Representación de Pago de Productos

La información de pago, de acuerdo al modelo, es enviada a la clase de navegación Proceso Pago Seguro. Al igual que autenticación usuario, Proceso pago seguro es transparente al usuario, es decir, no se estaciona en el navegador para que ningún usuario pueda interactuar con él. En esta clase de navegación se procesa la información de pago introducida por el Cliente y es enviada mediante la utilización del objeto e24TranPipe en modo seguro hacia la entidad la bancaria para efectuar transacción. Como resultado de este proceso, si el pago se llevó a cabo de manera exitosa, el Cliente

únicamente visualiza en su navegador la confirmación del pago. Este es el caso que ocurre en el caso de ejemplo y que se muestra en la figura 9.26. Si el pago no es exitoso, la notificación para el Cliente será de un error.

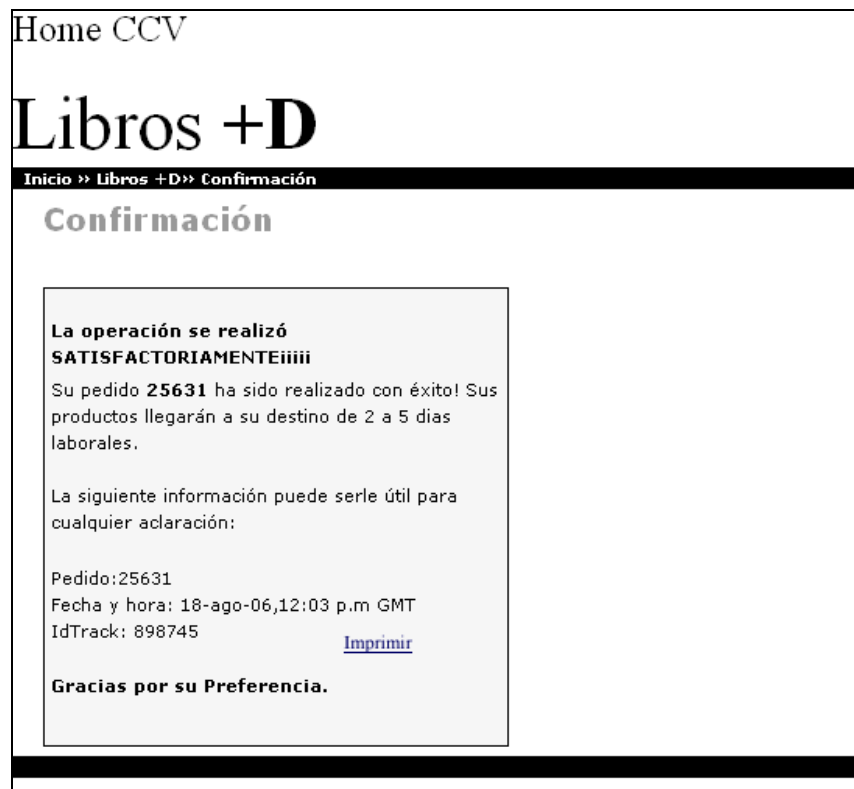


Fig 9.26: Representación de Confirm Pago Prod

Todas las funcionalidades del CCV, al igual que el caso de ejemplo expuesto en este apartado, se han modelado bajo el mismo patrón de lógica y metodología. En consecuencia, se puede afirmar que el modelo en UML de CCV desarrollado en esta tesis, revisado con la atención pertinente, constituye una referencia válida y útil en la construcción de un prototipo de implementación, así como también, un instrumento de documentación y verificación de la aplicación.

Conclusiones y Trabajo Futuro

Conclusiones

Para terminar, no resta más que presentar las conclusiones obtenidas del desarrollo de esta tesis, cuyo objetivo principal desde un inicio fue el de realizar un modelo en UML de una aplicación Web que constituyera una alternativa para que las pequeñas y medianas empresas de México puedan realizar comercio electrónico eficiente. En torno a este objetivo se concluye lo siguiente:

- Se realizó el análisis y diseño del Centro Comercial Virtual. Resultado de este trabajo se obtuvo: la especificación de requisitos, un modelo lógico-conceptual, un modelo de navegación, un modelo de interacción temporal, una colección de escenarios Web y un diagrama de distribución de la aplicación. Se incluye también un anteproyecto de implementación en el que se presentan algunos lineamientos generales para la futura implementación del CCV propuesto. Dicho anteproyecto consiste en una serie de sugerencias en relación a la conformación del equipo de desarrollo, una planificación preliminar del proceso de implementación, sugerencia de lenguaje de programación, de servidor de base de datos y de servidor Web, propuestas de herramientas de software apropiadas y un mecanismo de prueba para el modelo. Con lo realizado se cubrió el 100% de los objetivos planteados al inicio del desarrollo de la tesis.
- En la realización de cada uno de los diagramas del modelo del Centro Comercial Virtual aquí presentado se utilizó el Lenguaje Unificado de Modelado (UML) en el marco de la Ingeniería Web Basada en UML (UWE). UWE es una propuesta enfocada al modelado de aplicaciones Web basada en la extensión de la semántica del UML mediante la utilización de estereotipos. Tomando como referencia base a UWE se definió un perfil UML (extensión) basado en estereotipos adecuados para el modelado del Centro Comercial Virtual propuesto. En la definición de este perfil UML también se tomaron en consideración, aunque de manera minoritaria, otras propuestas de extensión del UML para el modelado de aplicaciones Web diferentes a UWE, como las utilizadas en [Conallen02] y [Baumeister02]. Con el perfil UML definido se logró la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso las aplicaciones Web.

- El modelo que se describe en esta tesis, la aplicación Web Centro Comercial Virtual, se diseñó para su utilización por usuarios humanos los cuales pueden ser del tipo Cliente, Vendedor o Administrador CCV. En este sentido, la principal actividad en el CCV de un usuario Cliente es la de comprar, la de un usuario Vendedor es la propiciar sus ventas y por último, la principal actividad de un usuario Administrador CCV es precisamente la de llevar a cabo tareas de administración en la operación del CCV en su conjunto. En virtud de que la operación del CCV incluye el pago electrónico, el CCV se asocia con una entidad bancaria para la realización de este tipo de tareas. Esta asignación de roles de usuarios en el CCV corresponde con la que comúnmente se toma en cuenta en el desarrollo de cualquier aplicación Web de comercio electrónico (Cliente, Vendedor, entidad bancaria). En este caso se incluyó al usuario Administrador CCV ya es necesario como figura de control sobre el CCV en general. Sin embargo, otro tipo de usuarios que en la vida real se asocian a una tienda (PyME) como los proveedores, mensajeros, empaques, etc., no tienen razón de ser en el modelo del CCV ya que dichos personajes corresponden a los procedimientos de cada tienda y no pueden ser proyectados en una generalización para todas las tiendas ya que factores como el giro o el tipo de productos es algo que varía de una tienda a otra. Es por lo anterior que este tipo de usuarios no fueron incluidos dentro del dominio de operación del CCV.
- Así también se realizó un trabajo de investigación importante respecto al marco teórico-conceptual asociado al desarrollo de esta tesis. En el capítulo correspondiente se presentan conceptos de Comercio Electrónico, la definición y características de las PyMEs mexicanas, los esquemas de comercio electrónico existentes orientados a las PyMEs Mexicanas y generalidades de las aplicaciones Web. Esta documentación en su conjunto constituye una fuente de información apropiada para el desarrollo de temas de investigación afines.
- Por último, la aportación de este trabajo se puede destacar desde dos puntos de vista. El primero de ellos tiene que ver con la propuesta de solución que se da al problema de la dificultad de acceso de las PyMEs mexicanas a recursos tecnológicos (específicamente de software) para la realización de comercio electrónico. En este sentido el modelo aquí presentado está enfocado para guiar el proceso de implementación del CCV el cual, una vez desarrollado, constituye una opción de solución al problema planteado. El segundo punto de vista está asociado con los métodos y técnicas avanzadas de desarrollo de software. El auge de los sistemas informáticos de la actualidad gira en torno a la Web. Al igual que cualquier otro tipo de aplicaciones, las aplicaciones Web requieren que sean desarrolladas apegándose a los lineamientos establecidos por la ingeniería de software para obtener productos de calidad. En este sentido este trabajo es un ejemplo robusto y completo acerca del cómo llevar a cabo el análisis y diseño de una aplicación Web utilizando el lenguaje UML.

De esta manera este trabajo constituye una referencia de consulta para todos aquellos desarrolladores de aplicaciones Web que deseen incluir en su proceso de desarrollo modelos en UML en la constante búsqueda de hacer cada día mejores productos.

Trabajo Futuro

El trabajo futuro inmediato es llevar a cabo el proceso de implementación del CCV propuesto, esto es, la construcción de un primer prototipo de CCV y la realización de pruebas sobre el mismo. Otros trabajos interesantes a considerarse tendrían que ver con mejoras al modelo de CCV aquí presentado. Entre estas mejoras podemos enumerar las siguientes:

- Modelar en el CCV la integración de otras opciones de pago distintas al pago con tarjeta de crédito, las cuales puedan coexistir y ser configurables por cada Vendedor. Por ejemplo opciones de prepago, pago contra-entrega y por transferencia bancaria.
- Incluir en el modelo de CCV módulos de entretenimiento como lo pueden ser: juegos en línea, zonas de descarga de recursos multimedios, blogs con temas de interés al público, etc. Esto con el propósito de pasar de los requerimientos netamente funcionales contemplados hasta este momento por el modelo hacia los requerimientos de mercado que garanticen el éxito de la aplicación para con los consumidores.
- Modelar en el CCV mecanismos de comunicación directa entre Clientes del CCV y los Vendedores. Por ejemplo foros de opinión, chat y servicios de mensajería.

Sobre la misma línea de interés de aplicaciones Web de comercio electrónico se puede tomar en cuenta el trabajo realizado hasta este momento como referencia para el desarrollo de proyectos de un siguiente nivel de complejidad. Por ejemplo, el modelado de una Comunidad Virtual de Comercio entre empresas (B2B). Estas comunidades aglutinan empresas comúnmente del mismo ramo, cada una con sistemas informáticos distintos e independientes, con el propósito de comerciar entre sí. Los problemas a superar en este esquema son varios, la comunicación entre sistemas diferentes, el desarrollo de una interfase común a todos los participantes y la negociación en las operaciones de compra venta entre las empresas. Por último, las líneas de investigación que se tomaron en cuenta para el desarrollo de esta tesis pueden retomarse, por ejemplo la evaluación de otras propuestas de notaciones, metodologías y técnicas de modelado de aplicaciones Web y el refinamiento de extensiones del UML para modelar aspectos dinámicos de las aplicaciones Web asociados con tecnologías específicas como XML, AJAX, entre muchas otras.

Bibliografía

- [AMIPCI04] Asociación Mexicana de Internet (AMIPCI), "Hábitos de los Usuarios de Internet en México, 2004", Presentación de Resumen Ejecutivo, 2004.
- [AMIPCI05] Asociación Mexicana de Internet (AMIPCI), "Primer Estudio AMIPCI de Comercio Electrónico en México 2005", Informe de Resultados, 2005. En www.amipci.org.mx.
- [Baumeister02] KOCH Nora, BAUMEISTER Hubert, HENNICKER Rolf, MANDEL Luis. "Extending UML to model navigation and presentation in Web Applications". Universidad de Munich Alemania, Mayo 2002
- [Benitez05] BENITEZ Víctor Hugo. "Pymes punto com". ". En Master-NET.net, Publicación sobre marketing, publicidad, e-commerce, diseño y promoción en Internet, 2005.
- [BRJUML00] BOOCH Grady, JACOBSON Ivar, RUMBAUGH James. "El Unificado de Modelado: Manual de Referencia". 1ra Edición, Ed. Addison Wesley, 2000.
- [Cachero00] CACHERO Cristina, GÓMEZ Jaime, PASTOR Oscar. "Extending an Object-Oriented Conceptual Modelling Approach to Web Application Design". CAISE 2000. Stocolmo Suecia, 2000.
- [Ceri01] CERI Stefano, MATERA Maristella, FRATERNALI Piero, MAURINO Andrea. "Designing multi-role, collaborative Web sites with WebML: a conference management system case study". 1er taller internacional sobre tecnologías de software orientadas a Web, Valencia España, 2001.
- [Clements02] CLEMENTS Paul, BACHMANN Felix, BASS Len, GARLAN David, IVERS James, LITTLE Reed, NORD Robert, STAFFORD Judy. "Documentig Software Architectures", 2da Edición, Ed. Addison Wesley, 2001.

- [Cohen00] COHEN Karen, ASÍN Larea Enrique. "Sistemas de Información para los Negocios. Un enfoque de toma de decisiones". 4ta Edición, Ed. McGraw Hill, 2000.
- [Conallen02] CONALLEN Jim. "Building Web applications with UML". Ed. Addison-Wesley, 2002.
- [DOFMX02] Diario oficial de la Federación, Estados Unidos Mexicanos, Fecha: viernes 13 de diciembre del 2002. Bajo responsabilidad del titular de la Secretaría de Gobernación.
- [Fernandez00] FERNÁNDEZ Juan, Coordinador de la Comisión Nacional de Comercio Electrónico de Cuba. Conferencia sobre el Comercio Electrónico en Cuba, Taller Nacional de Comercio Electrónico, La Habana Cuba, Diciembre 2000.
- [Gangopadhyay02] GANGOPADHYAY Aryya. "Managing Business with Electronic Commerce: Issues and Trends", 1ra Edición, Ed. Idea Group Publishing, 2002.
- [Garza00] GARZA Castaño Ricardo, Coordinador del Programa Emprendedor de la Universidad Autónoma de Nuevo León. "Creación de PyMEs: Objetivo Emprendedor". En Ingenierías, Octubre-Diciembre 2000, vol III No. 9.
- [González05] GONZÁLEZ Alejandro, Subsecretario de la Pequeña y Mediana Empresa dependiente de la Secretaría de Economía. Declaración proporcionada a Fernando Pedrero para el diario de circulación nacional "El universal" y publicada el día 18 de octubre del 2005.
- [Guerrero03] GUERRERO Luis. "Modelando Aplicaciones Web con UML". Taller de UML, Depto. de Ciencias de la Computación, Universidad de Chile, 2003.
- [IWStats05] Internet World Stats: Usage and Population Statistics. www.internetworldstats.com, 2005.
- [Koch01] KOCH Nora, "Software Engineering for adaptative Hypermedia Applications". Universidad de Munich Alemania, 2001.
- [Koch02] KOCH Nora, KRAUS Andreas. "The expressive Power of UML-based Web Engineering". Universidad de Munich Alemania, 2002.

- [Loshin04] LOSHIN Pete, VACCA John. "Electronic Commerce", 4ta Edición, Ed. Charles River Media, 2004.
- [Manchón03] MANCHÓN Eduardo. "Herramientas de diseño de aplicaciones Web: los personajes y escenarios". En Master-NET.net, Publicación sobre marketing, publicidad, e-commerce, y promoción en Internet, 2004.
- [Márquez04] MÁRQUEZ Rosell Jorge, Investigador del Centro de Desarrollo Regional en el Centro de Estudios Estratégicos del ITESM, Campus Guadalajara. "Reflexiones y tendencias del comercio electrónico". En Master-NET.net, Publicación sobre marketing, publicidad, e-commerce, diseño y promoción en Internet, 2004.
- [McConnell03] McCONNELL Steve. "Profesional Software Development". Ed. Addison-Wesley, 2003.
- [Méndez03] MÉNDEZ Morales José Silvestre. "Economía y la Empresa". 3ra Edición, Ed. McGraw Hill, 2003.
- [Potts01] POTTS Ricardo Flavio. "Comercio Electrónico: Herramientas para los negocios del tercer milenio", La Habana Cuba, 2001.
- [ProcyCom03] Procedimientos F.T. y Comunicaciones, S.L. "Comercio Electrónico para Comercios, PyMEs y Autónomos", versión 2003-12.
- [Rosenberg01] ROSENBERG Doug, SCOTT Kendall. "Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example". Ed. Addison Wesley, 2001.
- [Rossi00] ROSSI Gustavo, SCHWABE Daniel, LYARDET Fernando. "Web applications models are more than conceptual models". Proceedings of the web Engineering Workshop at WWWCM'99. París Francia, 2002.
- [Schwabe01] SCHWABE Daniel. "A conference Review System.", 1er taller internacional sobre tecnologías de software orientadas a Web, Valencia España, 2001.