



TESIS DE MAGISTER
EN INGENIERIA DEL SOFTWARE

MERCADOS VIRTUALES

AUTOR: LIC. MARCELO PETRONIO

DIRECTORES

DR. RAMON GARCÍA MARTÍNEZ (ITBA)

M.ING. BIBIANA ROSSI (ITBA)

DRA. ANA MARÍA MORENO (UPM)

BUENOS AIRES, 2003

Indice

1. INTRODUCCIÓN	9
1.1. ¿De qué trata esta tesis?	9
1.2. A quién esta dirigido este texto	9
1.3. Organización del Material	10
2. INTRODUCCIÓN A LOS MERCADOS VIRTUALES	15
2.1. Introducción a los Mercados Virtuales	16
2.2. ¿Porqué los mercados virtuales?.....	18
2.3. Modelos de Comercio Electrónico	19
2.4. Modelos de Negocio de los Mercados Virtuales	21
2.5. Características de un Buen Mercado Virtual	22
2.6. Tecnología de Internet	24
2.7. Protocolo de Transferencia de Hipertexto (HTTP).....	26
2.8. Lenguaje de Definición de Hipertexto (HTML)	27
2.9. Aplicaciones de Internet.....	28
2.10. Lenguaje de Marcas Extendido (XML).....	28
2.11. Servicios de Internet	29
2.12. Consideraciones Finales	30
3. DEFINICIÓN GENERAL DEL PROYECTO.....	33

3.1.	Descripción del Problema: El proceso actual.....	34
3.2.	Proceso Propuesto	35
3.3.	Análisis Comparativo	37
3.4.	Metodología a seguir	39
3.5.	Selección del Ciclo de Vida	40
4.	PLANIFICACIÓN GENERAL DEL PROYECTO	45
4.1.	Estimación por Puntos de Función.....	45
4.2.	Control de la Configuración.....	55
4.3.	Planificación del Trabajo.....	67
5.	ANÁLISIS DEL SISTEMA DE INFORMACIÓN	73
5.1.	Descripción y Objetivos	73
5.2.	Definición del Sistema.....	75
5.3.	Establecimiento de Requisitos	78
5.4.	Identificación de Subsistemas de Análisis	96
5.5.	Análisis de los Casos de Uso	98
5.6.	Análisis de Clases.....	107
5.7.	Elaboración del Modelo de Datos	121
5.8.	Definición de Interfaces de Usuario.....	122
5.9.	Análisis de Consistencia y Especificación de Requisitos.....	142
5.10.	Cuestiones Específicas al Desarrollo de Internet.....	149
6.	DISEÑO DEL SISTEMA DE INFORMACIÓN	153
6.1.	Definición de la Arquitectura del Sistema.....	153
6.2.	Identificación de Subsistemas de Diseño	164
6.3.	Diseño de la Arquitectura de Soporte.....	174
6.4.	Diseño de Casos de Uso Reales	190

6.5.	Diseño de Clases.....	198
6.6.	Diseño Físico de Datos.....	220
6.7.	Verificación y Aceptación de la Arquitectura del Sistema.....	228
6.8.	Generación de Especificaciones de Construcción.....	228
6.9.	Diseño de la Migración y Carga Inicial de Datos	231
6.10.	Especificación Técnica del Plan de Pruebas	233
6.11.	Establecimiento de Requisitos de Implantación	263
6.12.	Presentación y Aprobación del Diseño del Sistema de Información	264
7.	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN	269
7.1.	Preparación del Entorno de Generación y Construcción	269
7.2.	Generación del Código de los Componentes	271
7.3.	Ejecución de las Pruebas Unitarias.....	273
7.4.	Ejecución de las Pruebas de Integración	274
7.5.	Ejecución de las Pruebas del Sistema	275
8.	IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA DE INFORMACIÓN	281
8.1.	Establecimiento del Plan de Implantación	283
8.2.	Formación Necesaria para la Implantación	289
8.3.	Incorporación del Sistema al Entorno de Operación	290
8.4.	Carga de Datos al Entorno de Operación	292
8.5.	Pruebas de Implantación del Sistema.....	292
8.6.	Pruebas de Aceptación del Sistema.....	294
8.7.	Establecimiento del Acuerdo de Nivel de Servicio.....	294
8.8.	Presentación y Aprobación del Sistema.....	296
9.	BIBLIOGRAFÍA Y GLOSARIOS	299

9.1. Bibliografía	299
9.2. Glosario de Abreviaturas	301
9.3. Glosario de Términos	302
10. ANEXO I - INSTALACIÓN Y OPERACIÓN DEL SISTEMA DE INFORMACIÓN	305
10.1. Instalación del Sistema.....	305
10.2. Operación del Sistema de Información.....	307
10.3. Instalación del Componente Cliente	307
11. ANEXO II - MANUAL DEL USUARIO DEL SISTEMA.....	311
11.1. Ingreso al Sistema	311
11.2. Creación de un Pedido de Cotización	313
11.3. Consulta de Pedidos de Cotización y Creación de la Cotización	317
11.4. Consulta de Cotizaciones.....	319
11.5. Creación de una Orden de Compra	320
11.6. Consulta de Ordenes de Compra	323
12. CONCLUSIONES.....	327

Capítulo I

Introducción y

Convenciones

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Introducción
Fase de Origen:	Introducción a los Mercados Virtuales
Nombre o Código del Elemento de Configuración de Software:	Introducción a los Mercados Virtuales (IMV)
Versión N°: 1.0.1	Entrega N°: 02
Fecha de Última Modificación:	9/02/2002
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

1. Introducción

1.1. ¿De qué trata esta tesis?

Este proyecto trata sobre el desarrollo de una aplicación del tipo mercado virtual para la compra y venta de materiales eléctricos. Este desarrollo se basa en la utilización de tecnologías de Internet y la metodología de desarrollo Métrica versión 3.

Este trabajo persigue varios objetivos que se detallan en el desarrollo de los primeros capítulos. A modo de introducción se puede decir que los mismos se encuentran relacionados con la idea de desarrollar un sistema distribuído sobre la Internet, con tecnología orientada a objetos de forma tal de plantear las bases para este tipo de desarrollo y brindar una guía de desarrollo de estas aplicaciones utilizando Métrica versión 3.

Por otro lado, se busca definir cuestiones específicas de este tipo de sistemas que serán tratadas de manera especial para ser tenidas en cuenta en futuros desarrollos.

1.2. A quién esta dirigido este texto

Este proyecto apunta a un público muy variado. Podemos hacer una gran división entre:

- Usuarios finales del sistema.
- Usuarios finales de la tesis.

Entre los primeros encontramos tanto a compradores como vendedores de materiales eléctricos que encuentren útil el sistema para

resolver cuestiones específicas de la logística de adquisición / provisión de productos.

En el segundo grupo podemos mencionar a ingenieros de software y profesionales del área de tecnología en general, que puedan utilizar este proyecto como guía para la implementación de futuros desarrollos, y cátedras de diseño e ingeniería de software que utilicen la documentación generada para su estudio como caso de referencia para este tipo de sistemas.

1.3. Organización del Material

El material presentado consta de una estructura de 8 capítulos, incluyendo el actual. Conceptualmente el material se divide en dos grandes partes. La primera parte, capítulos del 1 al 3, abarcan cuestiones de introducción a los conceptos, principales, introducción a la problemática a abarcar y la forma de encarar la solución. La segunda parte, capítulos 4 al 8, abarca los grandes grupos de actividades específicos de Métrica versión 3, necesarios para obtener una solución a dicha problemática (el sistema en cuestión).

El capítulo II, Dominio del Proyecto, introduce al lector en los conceptos básicos sobre los que se apoya el resto del material. Se tratan temas tales como definiciones, clasificaciones y ejemplos de mercados virtuales y comercio electrónico en general, así también como algunas estadísticas extraídas de análisis de mercado que ejemplifican la importancia de una solución informatizada.

El capítulo III, Definición General del Proyecto, introduce al lector en la problemática relacionada con la gestión de compras y ventas de materiales eléctricos y da una descripción de la mejora esperada a partir de la implementación del presente desarrollo. Se detalla el proceso de compra - venta, se identifican problemas a solucionar en cada etapa y se especifica una potencial solución a cada uno. El capítulo termina con una tabla comparativa entre la situación actual y la esperada al concluir el proyecto.

El capítulo IV, Planificación General del Proyecto, trata sobre cuestiones más específicas al desarrollo de software. Se estima el tamaño del proyecto aplicando la técnica de puntos de función, se define el esquema de control de la configuración a utilizar en el resto del proyecto y se planifica el resto de las actividades.

El capítulo V, Análisis del Sistema de Información, define el trabajo a realizar en cuanto a los requisitos del sistema. De utilizan diversas técnicas del desarrollo orientado a objetos (diagramas de análisis, casos de uso, etc) para determinar el alcance del sistema y establecer su posterior desarrollo.

El capítulo VI, Diseño del Sistema de Información, especifica y justifica los criterios de diseño a utilizar para el desarrollo del sistema. En este capítulo es donde más se intenta aportar en cuanto a guía de implementación de este tipo de sistemas, ya que es en este momento donde encontramos la diferencia sustancial entre este tipo de tecnología y el resto. Se diseña y especifica el sistema de información.

El capítulo VII, Construcción del Sistema de Información, se construyen los componentes que conforman la solución. Además, se procede a ejecutar los distintos tipos de prueba necesarios para verificar la calidad del sistema. Se definen las necesidades de formación del usuario final y la carga inicial de datos.

El capítulo VII, Implantación y Aceptación del Sistema de Información, se describen las actividades para la entrega y aceptación del sistema en su totalidad y para el paso a producción del mismo.

Como ya se ha planteado con anterioridad, es uno de los objetivos de este proyecto servir como base para futuros desarrollos basados en esta tecnología, es por esto que los capítulos estarán organizados teniendo en cuenta tres cuestiones:

Cada capítulo sigue la secuencia de actividades planteada por la metodología Métrica versión 3. En cada una de estas actividades se introduce al lector en un resumen teórico que indica los objetivos de cada etapa y las herramientas a utilizar. En cada actividad también de presenta el desarrollo de la misma que comprende los diagramas, documentos y especificaciones necesarias para el proyecto. Al concluir cada capítulo se presenta un apartado que describe las adaptaciones y cuestiones específicas para el desarrollo de aplicaciones de Internet.

Este último apartado intenta tener un valor teórico / práctico que justifica las decisiones tomadas a lo largo del capítulo e informa al lector sobre otras posibles decisiones o criterios a utilizar en casos similares.

Estos apartados especiales hacen su aparición en tanto y en cuanto se identifiquen estas salvedades o tenga sentido la aclaración.

Capítulo II

Introducción a los Mercados Virtuales

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Introducción
Fase de Origen:	Introducción a los Mercados Virtuales
Nombre o Código del Elemento de Configuración de Software:	Introducción a los Mercados Virtuales (IMV)
Versión N°: 1.0.2	Entrega N°: 05
Fecha de Última Modificación:	30/01/2002
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

2. Introducción a los Mercados Virtuales

En este capítulo se presentan los conceptos principales del dominio del conocimiento relacionados al presente proyecto de tesis.

Para comenzar se hace una introducción a los mercados virtuales. Los mercados virtuales están muy relacionados con el desarrollo en el uso de Internet y la evolución del comercio electrónico en cualquiera de sus variantes.

Una vez comprendidos los fundamentos del mercado electrónico y el concepto de mercado virtual, se introduce al lector en el tema de tesis propiamente dicho: un mercado virtual para compra y venta de materiales eléctricos. Se presentan tablas comparativas que relacionan y comparan la problemática actual y la solución propuesta.

A continuación se detalla el plan de proyecto. Se identifica el ciclo de vida que se utilizará y se indican las actividades a realizar y su cronograma de ejecución.

Luego, una vez identificados los productos que resultan de las actividades definidas en el cuarto apartado se define el esquema de gestión de la configuración de los componentes del proyecto. Para ello se introduce el tema, se identifican los elementos a gestionar y se especifica la política de gestión de la configuración a utilizar.

2.1. Introducción a los Mercados Virtuales

Para que el lector pueda comprender el dominio del conocimiento de la tesis, es necesario comprenda varios conceptos relacionados con el comercio electrónico.

A continuación se presentan algunas definiciones extraídas de publicaciones del Gartner Group [Gartner Group 2001a].

2.1.1. Los negocios electrónicos (e-Business)

Se define como e-business a *"cualquier proceso de negocio realizado mediante el uso de una red de computadoras"*. Si bien esta definición es suficientemente genérica como para abarcar un amplio espectro de sistemas y soluciones que permitan desarrollar algún tipo de negocio, en la actualidad se utiliza este nombre para hacer referencia al software que utiliza la tecnología de Internet para unir a las partes involucradas en dicha operación de negocios.

Por ejemplo, supongamos una compañía de correo, que entre otras actividades, se dedica a enviar paquetes alrededor del mundo. Esta compañía podría desarrollar un sistema implantado sobre Internet que permita a sus clientes identificar el estado, ubicación y fecha de arribo de sus paquetes. Esta información podría ser consultada tanto por el que envía como el que recibe el paquete. De esta forma se podrían reducir drásticamente los llamados a las oficinas comerciales que tengan la intención de averiguar dicha información.

2.1.2. El comercio electrónico (e-Commerce)

Se define como comercio electrónico a *"comprar y vender sobre una red de computadoras"*. El comercio electrónico es un subconjunto del universo planteado por la definición anterior. En este caso existe una transacción comercial entre las partes involucradas. Es decir, existe una transferencia de fondos (compra-venta). En la definición anterior se habla de negocios sin que necesariamente se trate de transacciones económicas.

Un ejemplo puede ser un sistema desarrollado por un vendedor de productos minoristas implantado sobre Internet que permita a sus clientes revisar el catálogo de sus productos, pedirlos y pagarlos en forma electrónica.

2.1.3. Los mercados virtuales (e-Marketplace)

Según el Gartner Group [Gartner Group, 2001b] "los mercados virtuales son portales que permiten a una comunidad de usuarios interactuar en una variedad de actividades comerciales dentro del mismo". También agrega que los modelos de negocios planteados por dichos sitios se basan en compras, ventas y subastas. Para finales del año 2000 se contabilizaron 1000 de estos sitios y se predicen 3000 para finales del 2002 (datos para América Latina) [Gartner Group, 2001c].

Los mercados virtuales son, por lo general, sitios en Internet que brindan la posibilidad de relacionar un gran número de empresas, clientes y proveedores, en un sólo punto de encuentro donde comprar y vender los productos de interés. Las compañías involucradas en el desarrollo de los mercados virtuales, los definen como verdaderos ecosistemas globales y completos, que enlazan automáticamente toda la cadena de operaciones, desde los proveedores hasta el consumidor final.

Estos espacios están conformados por distintos operadores de una actividad común donde pueden realizar sus procesos de negocios, activar órdenes de compra, producción y comercialización, ofrecer sus productos y servicios, ofrecer excedentes de stock, establecer acuerdos, presentar ofertas o demandas concretas, etc.

Por lo general, se habla de agrupaciones verticales de servicios o productos. Dentro de la jerga del comercio electrónico, se habla de soluciones verticales cuando las mismas están íntimamente ligadas a un tipo específico de negocio. Un ejemplo puede ser el presente proyecto de desarrollo de un sistema de mercados virtuales para vendedores – compradores mayoristas de materiales eléctricos.

Las soluciones horizontales, en cambio, se encuentran relacionadas a una tecnología específica más que a un tipo de negocio. Un buen ejemplo de soluciones horizontales es el correo electrónico que es una tecnología destinada a intercambiar mensajes a través de la Internet, pero no atiende cuestiones específicas a un mercado particular.

Un mercado virtual debe facilitar la localización de productos y servicios (quizás dispersos por Internet o bien no informatizados) de un mercado y permitir que los mismos puedan ser vendidos o consumidos por

compradores a través de la Internet. Todo el ciclo de la venta – compra debería estar auto contenido, desde la búsqueda de un producto hasta su compra y envío.

2.2. ¿Porqué los mercados virtuales?

La utilización de sistemas de mercados virtuales, trae acarreadas ciertas ventajas para ambas partes (compradores y vendedores). A continuación se detallan algunas de ellas.

Para el **comprador**, utilizar un sistema de este estilo permite:

- Reducción de tiempos de búsqueda de productos y servicios, ya que encuentra en un solo lugar toda la información en lugar de buscarla en catálogos, cámaras de comercio ó, en el mejor de los casos, dispersa en Internet.
- Llegar a una mayor cantidad de proveedores de la que sería posible en forma directa (buscando en el mercado).
- Integrar los sistemas que administran la cadena de provisión con sistemas de sus proveedores para lograr una provisión automática.

Desde el punto de vista del **proveedor** se habla de:

- Una cantidad creciente de potenciales clientes dado que la cantidad de usuarios de Internet se acrecienta día a día. En América Latina se estiman en 9 millones los usuarios de Internet hacia fines del año 2001 [Gartner Group, 2001c].
- Mejorar el servicio brindado a sus clientes reduciendo el tiempo de ciertas operaciones, o bien, facilitando las mismas. Un buen ejemplo consiste en una terminal portuaria que permite a sus clientes conocer a cada momento la tarifa que debe abonar por importar su mercadería y permitir que la misma se pague por medio de una transferencia electrónica de dinero, sin tener que concurrir a una oficina de aduana con una suma aproximada (dado que desconocía anteriormente el monto a abonar) y con el dinero en efectivo (con el riesgo que esto implica).

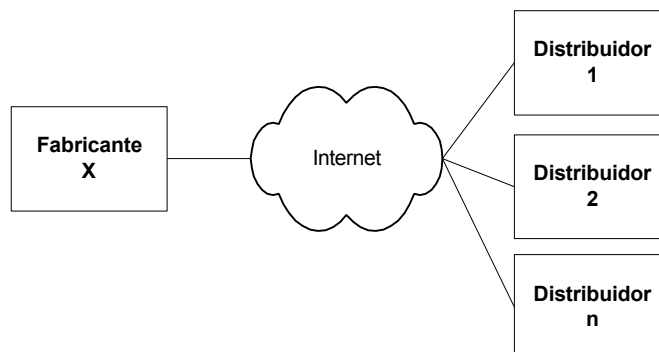
Este tipo de comercialización es una buena aplicación de tecnología para reducir costos y tiempos de ventas de productos y servicios, pero no deja de ser un canal de ventas adicional a los tradicionales, al menos, en tanto y en cuanto no se produzca un cambio radical en la forma de realizar negocios hoy en día. Y como todo canal de ventas, acomodar a la empresa a una nueva forma de comercialización trae acarreados nuevos problemas de diversas índoles, por ejemplo, cuestiones de logística relacionadas a vender productos a clientes en todo el mundo.

De todas formas, es un canal interesante. Según datos del Gartner Group el volumen de negocios realizado por sistemas de comercio electrónico en América Latina para el año 2000 fue de mil millones de dólares y se predice que crecerá a diez mil millones para fines del 2004 [Gartner Group, 2001c].

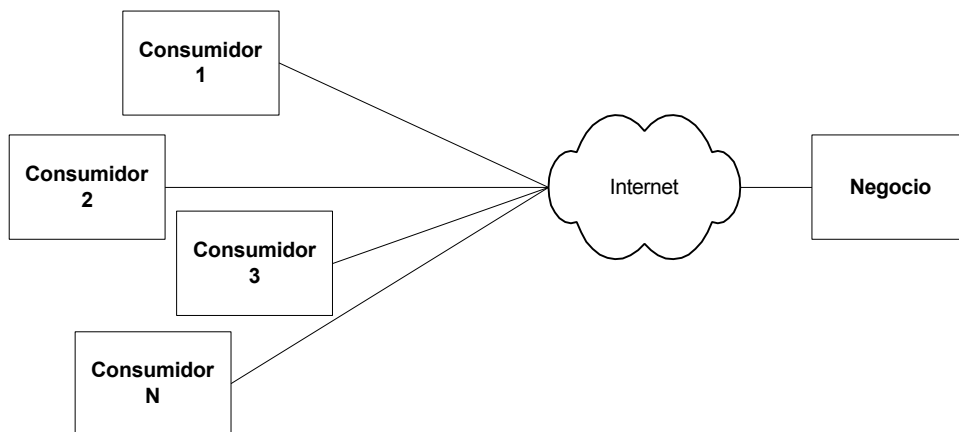
2.3. Modelos de Comercio Electrónico

Dentro del ámbito del comercio electrónico se encuentran 2 variantes básicas de las cuales se desprende una gama de combinaciones. Los modelos básicos son:

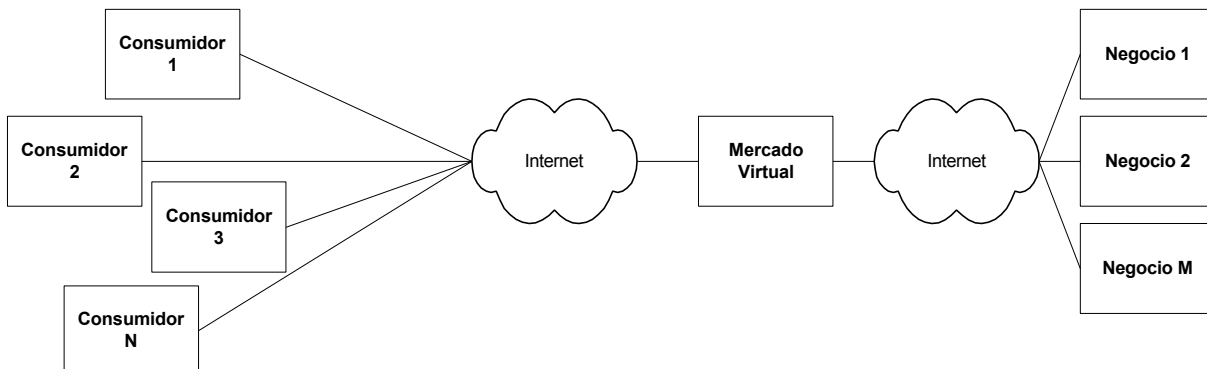
- **Negocio-a-Negocio** (del inglés Business-to-Business o B2B): son aquellos casos donde las partes que intervienen son en su totalidad empresas. Ej: un fabricante que recibe pedidos de sus distribuidores.



- **Negocio-a-Consumidor** (Business-to-Consumer o B2C): son aquellos casos donde una de las partes es un negocio y la otra esta formada por consumidores finales del producto de dicho negocio (por lo general una gran cantidad). Ej: un negocio de electrónica que vende a sus clientes por Internet.



En el caso de los mercados virtuales, se habla de una variante conocida como Negocio-a-Negocio-a-Consumidor (B2B2C). El mercado virtual hace las veces de intermediario entre negocios (en este caso vendedores de materiales eléctricos) que aceptan pedidos desde un sitio de venta (el negocio del mercado virtual) que agrupa pedidos de un sector de los consumidores (los consumidores de materiales eléctricos).



No siempre los mercados virtuales obtienen su ganancia a partir de comprar y vender. En muchas oportunidades se verifica que los mercados virtuales son empresas que no intervienen en dichas actividades sino más bien son un canal de distribución adicional administrado por empresas de corte netamente informático que proveen de la infraestructura tecnológica necesaria, que de otra forma sería inaccesible por los negocios y los consumidores.

2.4. Modelos de Negocio de los Mercados Virtuales

En la actualidad se pueden verificar varios modelos de negocio, entre los que se pueden mencionar:

2.4.1. Ganancias por cobro de un porcentaje sobre las transacciones

En este caso se trata de transacciones efectuadas sobre Internet gravadas con un importe por parte de la empresa poseedora del mercado virtual.

Como ejemplo se puede pensar en un comprador y un vendedor de un producto X. En el momento en que la venta se concreta, la empresa dueña del mercado virtual obtiene una ganancia porcentual del precio de la venta. Este esquema tiene como ventaja que tanto el demandante como el oferente no tienen un canon fijo que abonar, sino más bien, el canon es directamente proporcional a los negocios que se realicen por este medio. Esto reduce la barrera de entrada a este tipo de servicios. En este caso, el propietario del sitio asume el riesgo que significa implementar el mercado virtual, dado que si no se realizan suficientes ventas por medio del sistema, el mismo puede no ser rentable.

2.4.2. Pago de cuotas por suscripción de los usuarios

En esta variante, la cantidad de transacciones puede ser ilimitada, ya que bien los oferentes o demandantes ó ambos pagan por utilizar el sistema. Si bien esta opción reduce el riesgo para el poseedor del sitio, es verdad que también eleva la barrera de entrada para los usuarios: si el sistema no es lo realmente útil que los usuarios esperan, pronto estos prescindirán del mismo.

Un ejemplo de este caso es un portal que cobra a los proveedores de productos un canon por cada producto publicado en el catálogo, ó bien, a los usuarios una cuota mensual por utilizar el sistema.

2.4.3. Ingresos por publicidad y comportamiento de los usuarios

Este tipo de sitios intenta ser gratuito para los compradores y vendedores. Los usuarios pueden ingresar y comerciar todo lo que quieran. La única restricción es que deban ver constantemente publicidad de productos de distinta índole. En los sitios más modernos se utilizan tecnologías de personalización de usuarios, derivadas de la minería de datos (del inglés data mining), que permiten ver qué es lo que el cliente busca y en base a dicha información, proponer publicidad específica para cada usuario. En este caso, los ingresos del sitio provienen exclusivamente de la venta de dicha publicidad y de los datos obtenidos a partir del comportamiento de los usuarios dentro del sistema: ¿qué miran? ¿cada cuánto entran al sistema? ¿qué edad tienen los compradores del producto X?

Si bien estas alternativas parecen excluyentes entre sí, los verdaderos mercados virtuales hacen uso de variantes y mezclas de las mismas.

Esos sitios terminan siendo un producto que adquieren tanto compradores como vendedores y debe brindar una serie de ventajas comparativamente superior a la compra tradicional para tener una mayor cantidad de subscriptores al servicio.

Los mercados virtuales, en definitiva, son una implementación de las tecnologías de colaboración entre empresas.

2.5. Características de un Buen Mercado Virtual

Un mercado virtual es un modelo de negocios nuevo y con ello nada sencillo de implementar como negocio, aunque una vez en marcha representa una herramienta muy poderosa para las partes que intervienen en su uso. Para que un emprendimiento de este tipo tenga chances de ser exitoso, la lista de características que debe cumplir es larga.

A continuación se tratan algunas de estas (las más significativas):

2.5.1. Conocimiento y experiencia en el negocio

No cualquiera puede crear un e-Marketplace. Para comenzar, se necesita tener un conocimiento específico en la industria a la que este mercado pertenece. El mercado electrónico, para tener éxito, tiene que ser más eficiente y conveniente que las formas tradicionales de intercambiar bienes y servicios para las empresas. Por lo tanto, es imprescindible conocer el "negocio" o bien contar con especialistas en el tema para identificar las potenciales áreas de mejora o reducción de costos de un determinado negocio.

2.5.2. Estandarización de la oferta

Es necesario tener la capacidad técnica y comercial de definir un "lenguaje único" en base a un catálogo de productos y servicios que abarque todo el mercado de punta a punta. Sin esto, las partes que negocian pueden entender de forma distinta las ofertas, generando incongruencias e inconsistencias que terminan indefectiblemente en negocios que se deshacen luego de acordados, problemas y decepciones por parte de los participantes. Es por esto que el manejo de un catálogo, que unifica los criterios de definición, presentación, evaluación, y calidad de los productos y servicios que se comercializarán en el mercado virtual es un requisito imprescindible. Esta estandarización de la oferta no es una actividad trivial y su complejidad se encuentra relacionada con el mercado en cuestión.

2.5.3. Habilitar y facilitar la negociación

Un mercado virtual, así como un mercado real, es un lugar donde si bien están definidas las reglas del juego, no está definido a priori de qué forma o en qué condiciones se van a cerrar las transacciones. De hecho, las grandes transacciones requieren negociaciones complejas y los resultados distan mucho de la forma "vendo este producto a este precio". Condiciones de entrega, pago, beneficios adicionales, canjes, exclusividades, y muchos otros elementos están siempre en juego a la hora de negociar. El mercado virtual debe brindar las herramientas para que las negociaciones abarquen todas estas características de una forma flexible (permitir una negociación tan fluida como la actual) y confiable (habilitar mecanismos de autenticación de las partes).

2.5.4. Niveles de aprobación

En las compras convencionales, es probable que la aprobación de una compra o de un descuento en una venta, sean responsabilidades de distintos perfiles en distintos niveles del organigrama de las empresas. El mercado virtual no es la excepción. Por lo tanto, el sistema deberá brindar las herramientas que permitan dichos niveles de aprobación e interacción entre partes de una misma empresa.

2.5.5. Arbitraje

En muchos casos, el mercado virtual debe autodepurarse. Como en cualquier mercado, el mercado en sí mismo se depura haciendo correr la voz sobre el incumplimiento de cierto proveedor o comprador. Es deseable entonces que el mercado virtual contenga mecanismos que permitan identificar el incumplimiento de una de las partes de un negocio cerrado dentro del sistema, para informar, quizás en un proceso de compra posterior, a potenciales negociantes de la fiabilidad de un participante. Tiene que estar definido quién va a validar o invalidar el cumplimiento de cada una de las partes, quién va a ser el árbitro de la disputa y que tipo de acciones este puede tomar, para validar la transacción, deshacerla y eventualmente poner penalidades a una o ambas partes. Sería bueno que dicho sistema también pueda ejecutar dichas penas. Ej: cobro de una multa.

2.5.6. Disponibilidad sobre estado financiero de las partes

En muchos casos es deseable, además de la información sobre la participación histórica de un miembro dentro del mercado virtual, su información financiera actual. Esta información es necesaria, por ejemplo, para gestionar la aprobación de una compra muy importante. Esto se hace manifiesto, por sobre todo, en sistemas como el presente donde la consideración de "mercado mayorista" hace pensar en volúmenes de venta importantes. Los proveedores no quieren venderle a clientes incobrables. Este punto es importante a la hora de brindar confianza a los participantes sobre el uso del sistema.

2.6. Tecnología de Internet

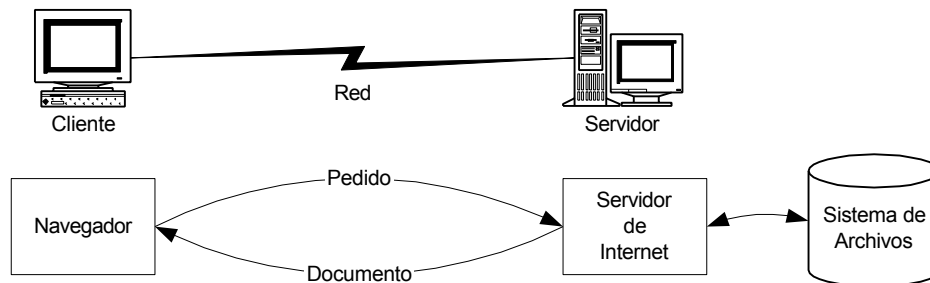
Constantemente en el desarrollo del material se hace referencia a ciertas tecnologías que son utilizada por una aplicación de Internet (del inglés

"Web Application"). Este apartado brinda una breve introducción a cada una de ellas para interiorizar al lector sobre las mismas. En caso de tratarse de un lector experto en las mismas se puede proceder al próximo capítulo aunque se recomienda al menos una lectura rápida de los siguientes puntos.

Las aplicaciones de Internet evolucionaron desde los originales sitios de Internet (del inglés "Web Site"). Los primeros sitios fueron creados por Tim Berner-Lee en el CERN (el laboratorio europeo de física de partículas) y consistían en sistemas distribuidos de hipermedia que permitían a los investigadores acceder directamente desde sus computadoras a la información publicada por sus colegas investigadores. Esta información era accedida de a trozos utilizando una herramienta instalada en la computadora cliente llamada navegador (del inglés "browser"). Con esta, el usuario puede requerir documentos almacenados en otros computadores y visualizarlo en su pantalla. Para ver un documento, el usuario debe arrancar el navegador e indicar la ubicación del mismo ó URL ("Uniform Resource Location", o ubicación uniforme del recurso). El pedido es enviado a la máquina que contiene el documento; este pedido es recibido por otro componente de software llamado servidor de Internet (del inglés "Web Server"). Este servicio identifica el documento pedido por el navegador, lo busca en su sistema de archivos y lo devuelve hacia el cliente.

Estos documentos conforman una hipermedia dado que contienen referencias ("links") hacia otros documentos que pueden estar en el mismo servidor, o bien, en otros nodos de la red que contienen dicha información. De allí el nombre de "web" o tela de araña con el que se denomina en muchos casos a Internet. Esta forma de publicación de documentos ha demostrado ser simple ya que el usuario simplemente haciendo un click del ratón sobre una referencia obtiene el documento deseado (el navegador se encarga de hacer el trabajo).

Sistema de Internet Básico



Una aplicación de Internet extiende esta funcionalidad ya que si bien el resultado final es el mismo (un documento en el navegador), el servidor contiene lógica de negocios que genera páginas específicas a tal fin.

2.7. Protocolo de Transferencia de Hipertexto (HTTP)

Los navegadores y los servidores se comunican por medio de un protocolo especial denominado HTTP (del inglés "Hyper Text Transfer Protocol") que define exactamente como el navegador debe realizar un pedido a un servidor de Internet. Este protocolo se encuentra especificado por el Consorcio de Internet (W3C) [W3Ca]. El cliente incluye dentro del protocolo, la ubicación del documento en el servidor. Este es un protocolo ASCII (es decir, conformado por caracteres simples reconocibles por un humano).

Nuevamente, el HTTP hace uso del URL para ubicar un documento. El URL está conformado por el tipo de protocolo a utilizar ("http://"), el servidor donde buscar, el puerto por el cual comunicarse con el servidor y la ubicación del archivo. Un ejemplo se presenta a continuación de un pedido al servidor de páginas del ITBA para proveer la página principal del sitio:

```
http://www.itba.edu.ar/index.html
```

Unas de las grandes ventajas que aporta este modelo son su robustez y capacidad de tolerancia ante fallos. Esto se obtiene por el uso del protocolo HTTP que no es orientado a la conexión. Es decir, una vez que un pedido es satisfecho, la conexión entre el navegador y el servidor es terminada. De esta forma, el servidor no está obligado a mantener información del cliente, lo que permite atender una cantidad mayor de clientes que del modo tradicional.

Cuando se habla de Internet, el protocolo HTTP realiza sus peticiones sobre el protocolo TCP/IP (formalmente dos protocolos: Transfer Control Protocol e Internet Protocol) que también es altamente tolerante a fallas por su forma de comunicación basada en paquetes. El TCP es una implementación del modelo de referencia para comunicaciones OSI (Open Systems Intercommunications) adoptado oficialmente como un estándar internacional por la Organización Internacional de Estándares (ISO).

Existen algunas variantes del protocolo HTTP como ser el HTTPS que utiliza el mecanismo SSL (Secure Sockets Layer) para proveer encriptación a los mensajes enviados entre cliente y servidor y evitar así la lectura de la información mediante el envío y recepción de mensajes.

2.8. Lenguaje de Definición de Hipertexto (HTML)

Además de ponerse de acuerdo en el protocolo de comunicaciones entre el servidor y el navegador, el navegador necesita mostrar el documento obtenido en la pantalla de la computadora cliente. Dicho formateo es realizado por el navegador. Aquí es donde aparece el HTML (Hiper Text Markup Language) [W3Cb] que se utiliza como lenguaje estándar para la escritura de documentos en Internet.

Un documento HTML contiene tanto la información a mostrar, como el formato con la cual debe ser mostrada. Para ello utiliza marcas (del inglés "tag") que definen que porciones del documento utilizan un formato específico, como ser tablas, negrita, tamaño de letra, justificación, etc. Toda esta información esta preparada para mostrarse en un monitor, lo cual trae serios problemas a la hora de generar un reporte impreso de dicha información, si es necesario cierto nivel de precisión en la copia impresa. Para estos casos, se deben utilizar otros componentes que ayuden a lograr dicho fin.

Dado que en cierto momento las aplicaciones de Internet no solo necesitaban bajar documentos, sino también enviar datos a un servidor, el estándar HTML fue enriqueciéndose agregando campos de entrada, botones, listas y otros tipos de componentes que en conjunción con los formularios permiten enviar información a un servidor remoto. Esta característica es indispensable para construir aplicaciones.

2.9. Aplicaciones de Internet

Las aplicaciones de Internet hacen uso de distintas tecnologías para hacer que el contenido de las páginas sea dinámico. Con el concepto de páginas, la información publicada en un documento cambiaba solamente al publicar una nueva versión del mismo. Ahora, se necesita que la misma página contenga información que cambie dependiendo de cada pedido. Además, el envío de información a un servidor afectará cierta lógica de negocios subyacente que tendrá que ser reflejada, por ejemplo, en una base de datos.

Es necesario hacer una distinción entre una Aplicación de Internet y un Sitio de Internet. La diferencia radica en la posibilidad que tiene el usuario de cambiar el estado de la lógica de negocios en el servidor (datos). Si esta lógica no existe, se habla de un sitio. En aquellos casos donde el usuario puede hacer dicho cambio por medio de un navegador de Internet, se habla de una aplicación.

2.10. Lenguaje de Marcas Extendido (XML)

Uno de los grandes problemas del HTML consiste en que tanto los datos como el formato de presentación de los mismos se encuentra mezclado en un único archivo. De esta forma, resulta casi imposible tener un cliente capaz de procesar información en el cliente de forma eficiente. Esto resulta en que cada vez que ocurre una pequeña modificación de datos, toda la página debe ser cargada nuevamente.

XML [W3C] es un lenguaje de marcas tal como HTML. Pero hace la división entre 3 tipos de archivos:

- XML: datos.
- DTD (Data Type Definition): especificación de la estructura de datos.
- XSL (Extended Stylesheet Language): formato de presentación.

Además se definen los parseadores y procesos de transformación para generar, por ejemplo, un archivo HTML a partir de los datos (XML) y un formato de salida (XSL).

En la actualidad, XML es una herramienta importante en cuanto al desarrollo del presente proyecto, por varios motivos:

- Se lo plantea como estándar para las comunicaciones entre aplicaciones.
- Se adapta fácilmente al paradigma de orientación a objetos permitiendo comunicar objetos entre cliente y servidor fácilmente.
- Existen estándares de comunicación basados en XML y HTTP lo que permite una comunicación sobre Internet (SOAP).

XML es un estándar especificado y mantenido por W3C lo que lo convierte en un fuerte estándar del mercado en general.

2.11. Servicios de Internet

En la actualidad, son muchas más las tecnologías involucradas en el desarrollo de una aplicación de Internet. Muchas de ellas, ni siquiera constan de una interfaz basada en un navegador y HTML, pero continúan usando la tecnología de comunicaciones subyacente (HTTP) para integrar el servidor con el cliente (ej: ICQ - www.mirabilis.com).

Los servicios de Internet (WebServices) son, conceptualmente, no aplicaciones sino servicios disponibles sobre Internet. De esta forma inclusive otros servicios o aplicaciones pueden accederlo simplemente por estar conectados a Internet.

El concepto de este tipo de servicios se baja en la idea de un objeto servidor que reside en un servidor de Internet y clientes de cualquier tipo que ejecutan métodos en el servidor por medio de pedidos y respuestas XML sobre pedidos y respuestas HTTP.

Estos pedidos y respuestas XML utilizan el estándar SOAP (Single Object Access Protocol) que al igual que el HTTP establece como deben pedirse y responderse los mensajes pero a nivel de aplicación.

Esto permite tener interfaces de usuario más flexibles y proveer más funcionalidad en los clientes al no tener que estar limitados al uso de HTML y las capacidades de un navegador.

A lo largo de este material, se verá que si bien la aplicación es una aplicación de Internet, se utiliza un cliente Java que utiliza SOAP para comunicarse con el servidor que contiene la lógica de negocios y los datos de la aplicación.

2.12. Consideraciones Finales

Como se ha comentado a lo largo de este capítulo de introducción, un mercado virtual es un nuevo canal de distribución de productos que hoy por hoy complementa a los canales tradicionales. Según estudios de mercado y tendencias del Gartner Group [Gartner Group, 2001c], esta tendencia se acrecentará en los próximos años, con lo cual en breve no se hará distinción de negocios tradicionales y negocios electrónicos, sino que estos últimos se llamarán simplemente negocios (ya que en algún momento cercano será la única forma de negociar para las empresas).

Por otra parte, un mercado virtual es por lo general una aplicación de Internet. Existen diversas tecnologías para desarrollar un mercado virtual. La definición de cuales utilizar dicta en gran parte el éxito del proyecto (a nivel técnico).

Capítulo III

Definición General del Proyecto

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Definición General
Fase de Origen:	Definición General del Sistema
Nombre o Código del Elemento de Configuración de Software:	Definición General del Sistema (DGS)
Versión N°: 1.0.2	Entrega N°: 05
Fecha de Última Modificación:	08/07/2002
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

3. Definición General del Proyecto

El presente proyecto se enfoca en desarrollar un mercado virtual para optimizar el proceso de adquisición de materiales eléctricos. A continuación se presenta una introducción al problema y las soluciones para la aplicación a desarrollar.

Para comenzar, se analizan los perfiles de usuario del sistema:

- El **consumidor** de materiales eléctricos: es el minorista, distribuidor, constructora que quiere comprar materiales eléctricos.
- El **proveedor** de materiales eléctricos: es el mayorista o fabricante de materiales eléctricos.

La gestión de compra involucra un proceso de interacción previo a la concreción de la compra de los materiales eléctricos. Esto se debe a que se manejan volúmenes de mercadería importantes, donde no sólo el precio es importante a la hora de comprar, sino también la disponibilidad, tiempo de entrega, calidad de los materiales, entre otros.

3.1. Descripción del Problema: El proceso actual

El proceso de compra-venta actual que relaciona a los compradores con los vendedores se describe en los siguientes pasos:

<i>Identificación de potenciales proveedores</i>	Los consumidores de materiales eléctricos deben identificar potenciales proveedores por medio de investigaciones de mercado que pueden ser llevadas a cabo por personal del departamento de compras, o bien adquiriendo una investigación realizada por una consultora. Esto se debe a que existe una gran variedad de materiales con prestaciones equivalentes en el mercado y gran cantidad de proveedores que los suministran. Por otro lado, es posible que necesiten más de un proveedor para satisfacer su demanda.
<i>Pedido de cotización a los proveedores</i>	Una vez identificados los proveedores, es necesario emitir un pedido de cotización. El mismo es un documento que contiene la especificación de los productos requeridos, su cantidad, tiempo de entrega, entre otros datos. El pedido de cotización debe ser enviado a los potenciales proveedores por algún medio como ser correo, fax, correo electrónico. De cualquier manera, este debe ser evaluado en forma manual dada la diversidad de formatos que puede adquirir uno de estos pedidos.

<i>Análisis de las cotizaciones presentadas por los proveedores</i>	En esta etapa, se analizan las cotizaciones enviadas por los potenciales proveedores para verificar que cumplan con las especificaciones del pedido de cotización. En este momento se decide por un proveedor. La decisión del proveedor puede realizarse en base al análisis de distintas variables que dependerán del criterio seleccionado por el comprador. Para simplificar el proceso diremos que las variables a analizar son: precio, fecha de entrega y cantidad disponible para entregar.
<i>Confección de la orden de compra definitiva</i>	Una vez recibidas las cotizaciones de los proveedores, y seleccionado con a cuales de ellos comprar, hay que ensamblar la orden de compra definitiva. Es probable que se opte por rearmar los pedidos, ya que cada proveedor puede tener artículos en mejor situación, con lo cual, la necesidad puede ser satisfecha por varios proveedores y no uno solo. Se confecciona la orden de compra con el formato y características que impone el proveedor. Cabe destacar, que si bien puede haber lugar a una nueva ronda de negociación, dicha posibilidad se descarta para simplificar la descripción de la situación actual.
<i>Colocación de la orden de compra</i>	Las órdenes de compra generadas son enviadas a los proveedores. Al igual que al enviar el pedido de cotización, las órdenes de compra pueden ser enviadas por diversos medios, aunque con los mismos problemas que en dicho paso.

3.2. Proceso Propuesto

Desde el punto de vista del mercado virtual, el proceso puede plantearse bajo el siguiente esquema:

<i>Identificación de potenciales proveedores</i>	Los clientes del sistema no tendrán que realizar estudios de mercado ni búsquedas de proveedores en el mercado. Muchos de los mismos se encontrarán catalogados en la base de datos del sistema y ya relacionados con los productos a cotizar. De esta forma, cuando el usuario cargue un pedido de cotización, el sistema informará todos los potenciales proveedores asociados a cada uno de los productos que conformen dicho pedido de cotización.
<i>Pedido de cotización a los proveedores</i>	El pedido de cotización se generará a través del sistema. Para ello el mismo contará con un catálogo de productos en el cual se podrán ver todas las especificaciones necesarias por el cliente. Este catálogo de productos será único para todos los proveedores. Dicha unificación surge como el resultado de un proceso de catalogación formal que escapa a los alcances del presente proyecto. Por lo tanto el sistema no contará con herramienta alguna de catalogación y gestión de catálogos; se presentará una base de datos de prueba con ciertos datos ya disponibles. Dichos datos se analizarán el apartado de Carga Inicial de Datos en el capítulo correspondiente. Los atributos de los productos a tener en cuenta para el presente proyecto, no intentarán guardar correlación con aquellos necesarios en la realidad. Por el contrario, intentan ser suficientes para el desarrollo del proyecto. Una vez que se genere el pedido de cotización, el sistema enviará el pedido a todos los proveedores subscriptos a los productos pedidos. El sistema enviará un e-mail al proveedor para notificarlo sobre la existencia de novedades en el sistema.

Análisis de las cotizaciones presentadas por los proveedores	El cliente ingresará al sistema y podrá consultar las cotizaciones presentadas por los potenciales proveedores. Esta consulta presentará toda la información disponible en el sistema de forma tal de permitir su análisis por parte del cliente.
Confección de la orden de compra definitiva	El cliente genera las órdenes de compra. Para ello el sistema lo asistirá en dicho proceso. El usuario seleccionará cada ítem del pedido de cotización original y el sistema mostrará las cotizaciones relacionadas. El usuario seleccionará una de dichas cotizaciones para continuar con el próximo producto hasta terminar. De esta forma, se generan las distintas órdenes de compra.
Colocación de la orden de compra	Una vez generadas las órdenes de compra, el sistema las almacenará en su base de datos. Adicionalmente se enviará un mensaje por correo electrónico para notificar a los proveedores la existencia de novedades en el mercado virtual.

3.3. Análisis Comparativo

El cuadro a continuación, permite comparar ambos modelos de mercado:

Mercado Actual	Mercado Virtual
Identificación de potenciales proveedores	
Identificar por medio de estudios de mercados o investigaciones por parte del departamento de compras a los potenciales proveedores.	Los proveedores ya se encuentran catalogados en el mercado virtual.

<p>Si la cantidad de proveedores es muy grande, es posible no llegar a todos o al mejor proveedor.</p>	<p>No importa la cantidad de proveedores, siempre es posible llegar a todos los catalogados y por lo tanto al mejor, dentro del catálogo del sistema.</p>
<p><i>Pedido de cotización a los proveedores</i></p>	
<p>Los pedidos de cotización no se arman por medio de sistemas, sino manualmente y teniendo en cuenta el formato impuesto por el potencial proveedor.</p> <p>Hay que enviar el pedido de cotización por fax o correo.</p>	<p>El pedido de cotización se escribe utilizando el sistema, con lo cual ya se tiene referencia de los productos disponibles y de ser necesario, toda la información del producto para ser accedida en línea.</p> <p>El pedido de cotización es enviado y accedido por el proveedor en forma instantánea. Un mensaje de correo electrónico lo notifica de la existencia de nuevos pedidos de cotización en el sistema.</p>
<p><i>Análisis de las cotizaciones presentadas por los proveedores</i></p>	
<p>El formato de la respuesta del proveedor no es estándar.</p> <p>La verificación de quién es el mejor proveedor es estrictamente manual.</p>	<p>El formato de la respuesta del proveedor es estándar (el sistema la impone).</p> <p>El sistema puede ordenar los presupuestos priorizando lo que el usuario considera más importante (precio, tiempo de entrega, cantidad, etc.)</p>
<p><i>Confección de la orden de compra definitiva</i></p>	
<p>Una vez identificado el/los proveedores, se confecciona la(s) orden de compra definitiva. Las mismas deben tener la información y formato impuestos por el proveedor.</p>	<p>El sistema asiste a seleccionar los productos a pedir por cada proveedor y se encarga de generar la orden con la información y formato para cada proveedor.</p>

Colocación de la orden de compra	
El comprador envía las órdenes de compra por fax, correo, correo electrónico, etc.	El usuario envía las órdenes de compra generadas en la tarea anterior presionando el botón "enviar". En un momento posterior, podrán agregarse mecanismos adicionales de firma digital para autenticar las partes, aunque esta funcionalidad esta fuera del alcance del proyecto actual.

3.4. Metodología a seguir

Como primer paso se realiza la selección del *ciclo de vida* a utilizar para el desarrollo del proyecto. Además como producto resultante se obtendrá el conjunto de tareas a realizar en cada fase del proyecto.

Una vez definidas las actividades, será necesario darles un marco temporal, lo cual se logrará con el *plan de proyecto*.

Luego, se realizará la planificación de las tareas de *gestión de la configuración* conducentes a establecer y mantener la integridad de los productos generados durante el proyecto de desarrollo software y a lo largo de todo el ciclo de vida del producto.

Sobre la base del plan de proyecto y siguiendo con los lineamientos metodológicos de Métrica 3, se comenzará con el análisis y desarrollo del plan considerando las primeras dos fases: *análisis de requisitos del sistema* y *especificación funcional del sistema*, utilizando las técnicas y herramientas de orientación a los objetos que se consideran en la metodología Métrica v3.

Luego serán desarrolladas las fases de *diseño técnico del sistema*, *diseño de componentes del sistema*, *desarrollo de procedimientos de usuario* y finalmente la de *pruebas, implantación y aceptación del sistema*.

Métrica v3 contempla el desarrollo para distintas tecnologías relacionadas con la arquitectura cliente/servidor, la orientación a objetos y el uso de bases de datos relacionales.

Cubre tanto desarrollos estructurados como desarrollos orientados a objetos. Para este último caso, Métrica v3 se basa en algunas de las

propuestas de otras metodologías orientadas a objetos, especialmente al tener en cuenta la mayoría de las técnicas que contempla UML 1.2 (Unified Modeling Language).

Ello se nota en que se han integrado las actividades exclusivas de la orientación a objetos dentro de la estructura de Métrica v3, definiendo así una metodología mixta. También se han reforzado las actividades y ampliado las técnicas teniendo en consideración las implicaciones de la tecnología Cliente/Servidor y el desarrollo de interfaces de usuario basadas en entornos gráficos.

Si bien la metodología no define explícitamente su uso para sistemas de arquitectura de varias capas con tecnología de Internet, el presente proyecto plantea su aplicación para el desarrollo del problema definido con anterioridad, el cual se encuentra relacionado con el comercio electrónico. Es necesario recordar que uno de los objetivos de la presente tesis es servir como base específica para desarrollar sistemas distribuidos sobre Internet.

3.5. Selección del Ciclo de Vida

Según el Proceso Unificado [Jacobson 1999] el ciclo de vida aconsejado debe tener como cualidades el hecho de ser iterativo e incremental. La producción de un software del tipo de un mercado virtual, puede llevar meses o inclusive años. Esto se debe a que siendo un mercado totalmente nuevo, y por lo tanto poco estable, aparecen constantemente nuevas funcionalidades a soportar por el software. Esto se debe a la disponibilidad de nueva tecnología, la necesidad planteada por los clientes, ó simplemente que la competencia ya implementó dicha funcionalidad.

Es por eso, que es conveniente dividir el proyecto en otros más pequeños que llamaremos iteración. El carácter de incremental del producto se basa en que cada iteración debería mejorar al producto, ya sea agregando funcionalidad, o bien, mejorando alguna de sus cualidades: performance, robustez, etc.

Este proyecto de tesis formaliza la primer iteración de un proceso evolutivo de tal índole.

La definición de utilizar a Métrica v3 como metodología de desarrollo y el paradigma de la orientación a objetos, plantea un universo de actividades a

desarrollar. El subconjunto de actividades a desarrollar en el presente proyecto definen una secuencia de actividades que puede observarse a continuación:

Gestión de la configuración del sistema

Especificar la política de gestión de la configuración para los componentes del sistema

Análisis del sistema de información

Definir el sistema
Establecer los requisitos del sistema
Desarrollar el modelo de negocio/modelo de dominio
Modelar los casos de uso
Especificar dichos casos de uso
Describir los subsistemas de análisis
Describir las interfaces entre subsistemas
Modelar las clases de análisis
Especificar el comportamiento de las clases de análisis
Analizar la realización de los casos de uso
Definir las interfaces de usuario
Analizar la consistencia y especificación de los requisitos
Especificar el plan de pruebas

Diseño del sistema de información

Definir la arquitectura del sistema
Diseñar la realización de los casos de uso
Modelar las clases de diseño
Especificar el comportamiento de las clases de diseño
Diseñar la interfaz de usuario

- Diseñar el modelo físico de datos
- Diseñar la carga inicial de datos
- Establecer los requisitos de implantación

Construcción del Sistema de Información

- Preparar el entorno de desarrollo
- Generar el código de los componentes
- Ejecutar las pruebas unitarias
- Ejecutar las pruebas de integración
- Ejecutar las pruebas del sistema
- Construir los componentes de carga inicial de datos

Implantación y Aceptación del Sistema

- Establecer el plan de implantación
- Incorporar el sistema al entorno de operación
- Cargar los datos al entorno de operación
- Ejecutar las pruebas de implantación del sistema
- Presentar y aprobar el sistema
- Pasar el sistema a producción

Todas estas actividades se subdividen en sub-tareas. Durante el desarrollo del proyecto (en los documentos que corresponda) se justificará la realización o no de las mismas.

En este contexto, la metodología Métrica v3 proporciona un conjunto de métodos y técnicas, que sirven de guía en la obtención de los distintos productos del ciclo de vida del software.

Por continuidad con Métrica v2.1, la notación empleada en la versión 3 es la misma para aquellas técnicas que son comunes en ambas versiones, en el caso de desarrollos orientados a objetos se utilizará la notación UML.

Capítulo IV

Planificación General del Proyecto

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Planificación
Fase de Origen:	Planificación General del Sistema
Nombre o Código del Elemento de Configuración de Software:	Planificación General del Sistema (PGP)
Versión N°: 1.1.1	Entrega N°: 04
Fecha de Última Modificación:	17/11/2001
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

4. Planificación General del Proyecto

En el presente capítulo se tratan los aspectos del proyecto relacionados a la estimación y planificación del proyecto.

A dicho fin, se comienza por la aplicación del método de Puntos de Función para obtener el tamaño del proyecto, luego se planifican las tareas que componen el proyecto, para luego concluir en la planificación de la gestión de la configuración.

4.1. Estimación por Puntos de Función

Los puntos de función miden el software cualificando la funcionalidad que proporciona externamente, basándose en el diseño lógico del sistema.

Entre los objetivos que se buscan con su aplicación podemos mencionar:

- Medir lo que el usuario pide y lo que el usuario recibe.
- Medir independientemente de la tecnología utilizada en la implantación del sistema.
- Proporcionar una métrica de tamaño que de soporte al análisis de la calidad y la productividad.
- Proporcionar un medio para la estimación del software.
- Proporcionar un factor de normalización para la comparación de distintos software.

La aplicación de la técnica de los puntos de función comprende los siguientes pasos:

- Definición de los límites del sistema.
- Definición de parámetros.
- Valoración de la complejidad.
- Análisis de las características generales del sistema.

4.1.1. Definición de los Límites del Sistema

El límite es utilizado para definir el alcance del sistema y ayudar a identificar los parámetros externos. Existen tres variantes en cuanto al uso de los puntos de función, en este caso se utilizarán basándose en el Límite inicial del proyecto a desarrollar, dado que la estimación se hará sobre los requisitos de un sistema que no existe aún.

Para este caso, la fórmula de cálculo será:

$$FPA = FP \times AF$$

Donde:

FP = es el número de puntos de función sin ajustar de la aplicación.

AF = es el factor de ajuste de la aplicación.

4.1.2. Definición de Parámetros

Para poder determinar la existencia de los componentes que contribuirán al total final, hay que determinarlos previamente.

Para el presente proyecto se utilizará la definición realizada por el IFPUG en 1994.

A continuación se presenta una tabla que identifica los parámetros identificados durante el análisis por casos de uso (definidos en el apartado 4.2 "Definición del Sistema" en el capítulo IV Análisis del Sistema de Información):

Capítulo IV - Planificación General del Proyecto
PGP Versión 1.1.1

Parámetro	Tipo	Complejidad
Ingreso al sistema	Entrada	Baja
Menú principal del comprador	Consulta Externa	Baja
Menú principal del proveedor	Consulta Externa	Baja
Carga de pedido de cotización	Entrada	Baja
Carga de ítem de pedido de cotización	Entrada	Media
Carga de cotización	Entrada	Baja
Carga de ítem de cotización	Entrada	Media
Consulta de pedidos de cotización	Consulta Externa	Media
Consulta detalle de un pedido de cotización	Consulta Externa	Media
Consulta de cotizaciones	Consulta Externa	Media
Consulta detalle de una cotización	Consulta Externa	Media
Carga de orden de compra	Entrada	Baja
Carga de ítem de orden de compra	Entrada	Alta
Tabla de usuarios	Fichero lógico interno	Baja
Tabla de productos	Fichero interfaz externo	Baja
Tabla de productos por proveedor	Fichero lógico interno	Baja
Tabla de pedidos de cotización	Fichero lógico interno	Medio
Tabla de ítems de pedidos de cotización	Fichero lógico interno	Medio
Tabla de cotizaciones	Fichero lógico interno	Medio
Tabla de ítems de cotizaciones	Fichero lógico interno	Medio
Tabla de órdenes de compra	Fichero lógico interno	Medio
Tabla de ítems de órdenes de compra	Fichero lógico interno	Medio
Buscador de productos	Consulta Externa	Alta
Mensaje de correo electrónico sobre novedades en el sistema	Salida	Baja

4.1.3. Valoración de la Complejidad

Para cada uno de los parámetros externos se ha de indicar su complejidad como baja, media o alta. Para las entradas, salidas y consultas, se puede evaluar su complejidad en función del número de campos que contengan y del número de ficheros a los que hagan referencia. Para los ficheros, por el contrario, su complejidad vendrá dada en función del número de registros y de campos que tengan.

A continuación se identificará el tipo de punto de función para cada funcionalidad del sistema y luego se valorará la complejidad del mismo.

A partir de la especificación de requisitos se puede obtener la siguiente tabla resumen de los parámetros de cálculo.

Tipo de Parámetro	Complejidad	Cantidad Identificada	Peso	Total
Entrada	Alta	1	6	6
	Media	2	4	8
	Baja	4	3	12
Salida	Alta	0	7	0
	Media	0	5	0
	Baja	1	4	4
Fichero lógico interno	Alta	0	15	0
	Media	6	10	60
	Baja	3	7	21
Fichero lógico externo	Alta	0	10	0
	Media	0	7	0
	Baja	1	5	5
Consultas	Alta	1	6	6
	Media	4	4	16
	Baja	2	3	6
Total de Puntos de Función sin Ajustar				144

4.1.4. Análisis de las Características Generales del Sistema

Una vez obtenidos el total de puntos de función sin ajustar, debe realizarse un ajuste del mismo en función de las características generales del sistema.

A continuación se presenta el resultado de dicho análisis.

4.1.4.1. Comunicación de datos

La aplicación soporta más de un ordenador front-end (cada usuario del sistema, tanto proveedor como comprador), pero siempre sobre un mismo protocolo de comunicaciones (TCP/IP). El valor de complejidad es 4.

4.1.4.2. Funciones distribuidas

El procesamiento del sistema funciona en forma distribuida, donde la transferencia de datos en ambas direcciones ocurre en línea. Esto se debe a que existen componentes que funcionan tanto en el cliente como en el servidor de aplicaciones.

El valor de la complejidad es 4.

4.1.4.3. Rendimiento

Los requisitos de rendimiento y diseño han sido definidos y revisados pero no requieren ninguna acción especial.

El valor de la complejidad es 4.

4.1.4.4. Configuraciones fuertemente utilizadas

La aplicación cuenta con componentes tanto en el cliente como en el servidor. Si bien estos son multiplataforma, existen ciertas características a tener en cuenta que limitan la funcionalidad del sistema, como ser protocolos de comunicaciones y capacidad del servidor.

El valor de la complejidad es 5.

4.1.4.5. Frecuencia de transacciones

Si bien con un uso intensivo del sistema la cantidad de transacciones y su frecuencia pueden ser elevadas, no existe una definición del período punta de transacciones. Por lo tanto, el valor de la complejidad es 0.

4.1.4.6. Entrada on-line de datos

El 100% de las transacciones del sistema es en línea.

El valor de la complejidad es 5.

4.1.4.7. Diseño para la eficiencia del usuario final

Existen varias funciones a tener en cuenta para el uso por parte del usuario final, como ser: uso de ventanas, ratón, menús, scrolling, etc. Si bien estas existen, no plantean cuestiones u objetivos específicos que requieran el uso de herramientas especiales.

El valor de la complejidad es 3.

4.1.4.8. Actualización on-line

Existe una actualización importante de los ficheros lógicos internos del sistema. De todas formas, no se han tenido en cuenta cuestiones de recuperación ante caídas en forma automática.

La valoración de la complejidad es 3.

4.1.4.9. Procesos complejos

El sistema incluye la utilización de archivos de auditoría. Luego, la valoración de la complejidad es 1.

4.1.4.10. Reutilización

La aplicación será empaquetada expresamente y/o documentada para ser fácilmente reusable. La aplicación será adaptada por el usuario final por medio de parámetros de mantenimiento, o bien, por un administrador. Este punto es importante dado que cada usuario del sistema implantará los componentes cliente y deberá poder usarlos sin problemas.

La valoración de la complejidad es 5.

4.1.4.11. Facilidad de instalación

El componente cliente debe ser instalado en los equipos de los usuarios finales. Para ello se preverá un esquema de instalación y distribución de nuevas versiones que funcionen en forma automática.

La valoración de la complejidad es 5.

4.1.4.12. Facilidad de operación

En este caso, no se definieron por parte del usuario necesidades especiales de operación o respaldo distintas a las normales.

Por lo tanto, según la especificación de puntos de función, la valoración a asignar es 1.

4.1.4.13. Instalación en múltiples sitios

Como ya se ha indicado, parte de la aplicación estará instalada en cada uno de los clientes y proveedores del mercado virtual. Dichos componentes serán multiplataforma (para poder ser ejecutados en distintos sistemas operativos), y se actualizarán automáticamente.

Valoración de la complejidad 5.

4.1.4.14. Facilidad de cambio

No existen especificaciones por parte de los usuarios en cuanto a flexibilidad en las consultas o gestión por parte del usuario de datos relativos al negocio. La valoración de la complejidad es 1.

4.1.5. Cálculo del Factor de Ajuste

La tabla a continuación, presenta un resumen del análisis del apartado anterior, a partir del cual se obtiene el factor de ajuste necesario para ajustar los puntos de función.

Característica	Valor de complejidad
Comunicación de datos	4
Funciones distribuidas	4
Rendimiento	4
Configuraciones fuertemente utilizadas	5
Frecuencia de transacciones	1
Entrada on-line de datos	5
Diseño para la eficiencia del usuario final	3
Actualización on-line	3
Procesos complejos	1
Reutilización	5
Facilidad de instalación	5

Facilidad de operación	1
Instalación en múltiples sitios	5
Facilidad de cambio	1
Total	47

Luego, el cálculo del factor de ajuste:

$$AF = 0,65 + (0,01 \times 47) = 1,12$$

Los puntos de función ajustados surgen del cálculo a continuación:

$$FPA = FP \times AF = 144 \times 1,12 = 161,28.$$

Como resultado, entonces, la presente aplicación tiene 161 puntos de función.

4.1.6. Estimación del Esfuerzo

Resta como trabajo el cálculo del esfuerzo en horas hombre a partir de los puntos de función ajustados. El mejor método de estimación consiste en basarse en los registros históricos de la misma compañía que desarrolla el software.

En el caso de no contar con dichos registros, como lo es el caso actual, es necesario recurrir a alguna fuente adicional de información. En su tratado sobre casos de uso y puntos de función [Longstreet 2001], David Longstreet propone una tabla con valores promedio de esfuerzo para desarrollo de sistemas basados en casos de uso y desarrollados con tecnología cliente-servidor con ambientes gráficos. La tabla se presenta a continuación:

Tamaño en Puntos de Función	Horas por Punto de Función	Número Potencial de Casos de Uso
10	1-2	1 o 2
100	3-5	5-10
1000	9-11	20-35
5000	15-18	50-65
10000	25-35	200-350

Es claramente evidente que la segunda fila refleja el proyecto actual, dado que se habla de 161 PFA y 7 casos de uso iniciales. Por lo tanto, asumiendo una postura pesimista en cuanto al cálculo de esfuerzo por punto de función, se puede hacer el siguiente análisis

$$161 \text{ PFA} \times 5 \text{ hs} \times \text{ hombre} = 805 \text{ hs} \times \text{ hombre}$$

Si se asume una base de dedicación de tesis de 5 horas diarias, se estima la duración del proyecto en 8 meses de duración, aproximadamente. Esta estimación estipula 8 meses de trabajo neto y no contempla posibles desviaciones de tiempo en base a cuestiones de corrección de entregables y tiempos muertos dentro del proyecto.

4.1.7. Cuestiones Específicas al Desarrollo de Sistemas de Internet

Es necesario tener en cuenta la complejidad inherente a la cantidad de tecnologías y lenguajes de programación involucrados en el desarrollo de sistemas de Internet. Una gran porción del material disponible sobre tablas de referencia de esfuerzo para la construcción de sistemas se basa en la idea de la construcción de sistemas basados en las tecnologías involucradas.

Según Longstreet [Longstreet 2001], la productividad en el desarrollo de sistemas en lenguaje COBOL puede llegar a ser hasta 20 veces menor que la presentada en la tabla de referencia utilizada en el punto 4.1.6.

Estas diferencias se basan principalmente en la productividad de los ambientes de desarrollo modernos que permiten la construcción visual de pantallas, reportes y componentes, a pesar de que los sistemas cliente servidor contienen varios lenguajes de programación y tecnologías. Por lo general, uno que permite la construcción de interfaces de usuario y lógica de negocios y otro de base de datos para la construcción de los procedimientos almacenados en el SGBD.

Pero al hablar de sistemas de Internet, se introducen cuestiones de comunicaciones y una mayor cantidad de lenguajes involucrados. Por ejemplo, se podría usar HTML y JavaScript para la construcción de las interfaces de usuario, Java para la construcción del modelo de negocios residente en el servidor de aplicaciones y SQL para la construcción de procedimientos almacenados. Desde el punto de vista de las comunicaciones, ya no sólo se utiliza una conexión SQL-TCP/IP para conectarse a la base de datos, sino que también se utiliza HTTP o SOAP

para interactuar entre la interfaz de usuarios y los componentes en el servidor de aplicaciones.

Estas cuestiones deben ser tomadas en cuenta a la hora de estimar un proyecto de estas características en el caso de no contar con componentes ya disponibles para la simplificación de la construcción de las distintas capas de software y las comunicaciones entre estas. En el presente proyecto, las interfaces se construyen como en un sistema cliente servidor y se utilizan componentes para la comunicación con el servidor, con lo cual el sistema toma una forma parecida a la de un sistema cliente servidor convencional y por eso se consideran válidas las mediciones de esfuerzo presentadas.

En el caso de tratarse de un sistema de Internet más convencional, será necesario buscar valores que se adapten a dicha forma de construcción, o bien, evaluar el esfuerzo por separado por cada capa de software.

4.2. Control de la Configuración

La gestión de la configuración es una disciplina cuyo objetivo es mantener la integridad de los productos que se obtienen a través del ciclo de vida de los sistemas de información.

Dicha integridad es soportada por un conjunto de actividades y controles recogidos en un plan denominado plan de gestión de la configuración. La existencia o no de este plan para un sistema de información dependerá de las propias necesidades del sistema, o de la posibilidad de que dicho plan se encuentre establecido previamente como método estándar y sea aplicable a lo largo del ciclo de vida del software. En el caso particular del presente proyecto de tesis, se define un plan de gestión de la configuración propicio para el mismo.

Para definir un plan de gestión de la configuración es necesario analizar el apoyo que contempla la metodología a utilizar. En este caso, se utiliza Métrica V.3, donde la interfaz se establece a través de dos clases distintas de actividades.

Las primeras son **actividades de definición**, orientadas a establecer las necesidades y estrategia de configuración del sistema de información tratado, en definitiva, están orientadas a establecer su plan de gestión de la configuración.

El segundo grupo conforman las **actividades de control** de la configuración. Forman la interfaz en sí misma, a través de la cual se transfiere y recibe información de control.

La relación entre un tipo u otro de actividades es muy directa, en la medida en que las actividades de definición establecen los criterios por los que se rigen las actividades de interfaz que determinan los procesos y el nivel de control de la configuración.

Según sea el caso, puede existir una descoordinación entre actividades de interfaz y actividades de definición en el desarrollo de un sistema. Esto es debido, al momento en que se realizan las actividades de definición del plan de gestión de la configuración.

4.2.1. Actividades de Definición

Este tipo de actividades tiene como objetivo la obtención de un producto denominado plan de gestión de la configuración. El contenido del plan establece las necesidades de configuración, a través de controles (control de versión, control de estado, control de procesos, etc.), y la identificación de los productos a controlar.

Estas actividades se realizan embebidas dentro de las propias actividades de Métrica V.3 a lo largo de sus procesos y obtienen como resultado productos propios que deberán ser registrados a través de las actividades de interfaz, como un producto más de Métrica V.3, siempre y cuando se establezca esta necesidad.

4.2.2. Actividades de Interfaz

Como ya se ha dicho, estas actividades forman la propia interfaz en sí, y permiten mantener la consistencia entre los productos de desarrollo y la gestión de cambios.

Potencialmente, las actividades de interfaz afectan a casi todas las actividades del desarrollo que obtengan productos o necesiten información de los mismos.

Otra diferencia respecto a las actividades de definición, es que no presentan una secuencialidad pura, estableciendo ciclos para algunos tipos de productos, es decir, interactúan continuamente con las propias actividades de Métrica V.3, controlando y gestionando sus productos y estableciendo

versiones de los mismos hasta que el producto se encuentra correctamente finalizado o aceptado. Según se van generando los productos a lo largo de las actividades de un proceso, se registran en el sistema de gestión de la configuración con el estado correspondiente (por ejemplo, en estado de "elaboración").

Durante la realización de las distintas actividades, los productos, en función de su naturaleza, van pasando por los controles de estado que establece su ciclo de vida, registrándose en el sistema de gestión de la configuración. De esta manera, el control de estados y de procesos establece los criterios para determinar si los productos continúan su ciclo de vida o por el contrario se produce una vuelta hacia atrás. De ser así, se volvería al punto anterior, es decir, se realizaría un cambio de versión del producto en cuestión y se volvería a registrar en el sistema (siguiendo el ejemplo en estado "elaboración").

Este ciclo se repite hasta que el producto completa su ciclo de estados y llega a su aceptación, momento en el cual se vuelve a registrar el resultado en el sistema de gestión de la configuración.

No todos los productos disponen de un mismo control de estado, éste dependerá del tipo de producto de que se trate y su relación con otros productos. El plan de gestión de la configuración establecerá los productos y sus tipos de estado.

Otro criterio de control establecido por la gestión de configuración es el control de procesos, que facilita el conocer la situación de un sistema de información a lo largo de su ciclo de vida. Para establecer adecuadamente este control, las actividades de interfaz registran el conjunto de productos que se obtiene de un sistema de información como un producto más, de esta forma, se le pueden atribuir estados que permiten controlar cada uno de los procesos de Métrica V.3 a lo largo de su ciclo de vida. Existen tres productos de estas características:

- El plan de sistemas de información como concepto que engloba todos los productos del proceso Planificación de Sistemas de Información (PSI) este proceso no aplica para el presente proyecto de tesis dado que no forma parte de un plan maestro.

- El conjunto de productos que conforman el sistema de información obtenidos en los procesos Análisis del Sistema de Información (ASI), Diseño del Sistema de Información (DSI), Construcción del Sistema de Información (CSI) e Implantación y Aceptación del Sistema (IAS) que sí se desarrollarán en la presente tesis.

4.2.3. Beneficios del Control sobre la Configuración

Todo esto se traduce en una serie de beneficios que revierten sobre el sistema de información, tanto a nivel interno de apoyo al desarrollo del sistema como a nivel global de resultados finales.

El control sobre la ubicación de los productos facilita su localización y el mantenimiento del sistema, reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo. Asimismo, permite el control del sistema como producto global a lo largo del ciclo de vida y consigue reducir el número de errores de adaptación del sistema, lo que se traduce en un aumento de calidad del producto y de la satisfacción del cliente.

El grado o nivel de implementación que debe conferirse a una gestión de la configuración dependerá de los objetivos y necesidades que la organización se plantee al respecto. La interfaz de la gestión de configuración con Métrica v3, presenta el grado más alto de implementación, cubriendo todos los procesos de la metodología.

En el caso particular del presente proyecto, se utilizarán las técnicas de gestión de configuración para los entregables que surjan a partir de las actividades de análisis, diseño y puesta en producción del presente proyecto de tesis.

4.2.4. Definición del Plan de Gestión de la Configuración

Dado que es necesario, en este caso debe definirse un plan de gestión de la configuración para el sistema de información. A continuación se presentan los aspectos a tener en cuenta por el plan y su planificación.

4.2.4.1. Identificación de componentes

Los productos a controlar, en este caso, se encuentran relacionados con aquellos que se generen en las diversas etapas del proyecto. Para ello

se presenta a continuación una tabla que indica los productos generados en cada etapa cuya configuración será gestionada.

Los documentos, serán presentados en formato PDF.

Los mismos tendrán un rótulo en la carátula que indique:

Información para el Control de la Configuración	
Proyecto: NOMBRE DEL PROYECTO (CODIGO DE PROYECTO)	Línea Base: XXXXXXXXXXXXXXXX
Fase de Origen:	XXXXXXXXXXXXXXXXXx
Nombre o Código del Elemento de Configuración de Software:	XXXXXXXXXXXXXXXXXX
Versión N°: X.X.X	Entrega N°: XX
Fecha de Última Modificación:	XX/XX/XXXX
Autor:	XXXXXXXXXXXXXXXXXXXX
Estado:	XXXXXXXXXX

En todas las páginas del documento se repetirá información como ser:

PROYECTO.NOMBRE DEL DOCUMENTO
CODIGO DEL DOCUMENTO Y VERSIÓN

Para la identificación de componentes se utilizará la siguiente nomenclatura:

Proyectos: los proyectos se indicarán como PRXXXXX donde PR son constantes y las cuatro X son caracteres referentes al nombre del proyecto o alguna otra forma de identificación. En este caso todos los documentos indicarán la constate PRTESIS que los identificará como componentes del proyecto de tesis.

Líneas Base: se indicaran como LBFXXX-XXX-NN donde LB son constantes, XXXX identifican al proyecto en cuestión, XXX identifican el tipo de producto y NN indica el número de versión. Ya se ha mencionado que los documentos se

4.2.4.2. Ubicación y localización de los productos

Dado que todos los productos a generar en este proyecto son de índole electrónica (documentos, software, diagramas, etc.) los mismos se ubicarán en una carpeta indicando su pertenencia al proyecto de tesis y se entregan en un CD-ROM que contiene todos los componentes del sistema.

4.2.4.3. Definición del ámbito y alcance del control de la configuración

Hay que destacar que el control de la configuración para el proyecto será de muy alto nivel. Esto se debe a que existe un alto grado de acoplamiento entre los componentes generados en cada fase del proyecto y que una modificación en alguno de ellos radica en la modificación del documento que lo contiene en muchas de sus secciones. Por lo tanto, se define un documento general por cada etapa (según las plantea Métrica v3) y los componentes que contendrá.

Etapa	Presentación	Contenido
Planificación	Documento de Planificación del Sistema de Información	Plan de proyecto. Plan de gestión de la configuración.
Análisis	Documento de Análisis del Sistema de Información	Diagramas de casos de uso Especificación de los casos de uso.
Diseño	Documento de Diseño del Sistema de Información	Diagramas de clases. Diagramas de secuencia.

		Diagramas de interacción. Plan de pruebas.
Construcción	Documento de la Construcción del Sistema de Información	Código fuente de los componentes del sistema. Código de los componentes para la carga inicial de datos. Resultado de la ejecución del plan de pruebas.
Implantación	Documento de Implantación del Sistema de Información	Plan de implantación. Componentes del sistema (binarios).

Dado que cada uno de los documentos contiene la totalidad de los componentes generados en cada etapa, los mismos harán las veces de línea base de la configuración.

Se puede decir entonces que se definen las siguientes líneas base para el sistema:

- **Línea base funcional:** se establece al finalizar la fase de especificación de requisitos. Estará representada por el documento de Análisis del Sistema de Información.
- **Línea base de diseño:** se establece al finalizar la fase de diseño; estará representada por el documento de Diseño del Sistema de Información y los diversos productos generados durante el diseño (programas de creación, prototipos de pantallas, etc.).
- **Línea base de producto:** se establece al finalizar la fase de pruebas y estará representada por el documento de Construcción del Sistema de Información, el plan de pruebas, los casos de prueba, los comandos SQL necesarios para las pruebas y los programas a probar.

- **Línea base operativa:** se establece al finalizar la fase de implantación. En este caso estará representada por los componentes binarios, manuales y el documento de Implantación del Sistema de Información.

4.2.4.4. Identificación de los Componentes

En este apartado se trata la idea de cómo identificar los documentos. Para tal fin se definen a continuación las nomenclaturas que identifican a cada tipo de documento:

Componente	Código
Introducción a los Mercados Virtuales	IMV
Definición General del Proyecto	DGP
Planificación General del Proyecto	PGP
Análisis del Sistema de Información	ASI
Diseño del Sistema de Información	DSI
Construcción del Sistema de Información	CSI
Implantación del Sistema de Información	ISI
Componente Cliente	CLI
Componente Servidor	SER

Además, se deben definir el resto de los componentes del sistema. En este caso, y dado que la herramienta de desarrollo contiene un esquema de versionado de componentes fuentes y binarios, no se definen en este documento esquemas tales. Para tal caso, ver el apartado anterior que define dicho esquema de versionado.

4.2.4.5. Definición de las reglas de versionado de los productos y los criterios de actuación

Para cada caso se definirá el criterio a tener en cuenta para generar una nueva versión del producto y cómo esta debe ajustarse al tipo de cambio.

Los componentes del sistema (en este caso tanto documentos como software) serán versionados teniendo en mente una identificación de versión formada por 3 números con el formato X.X.X

4.2.4.6. Versión Principal

El primer número indica la versión del componente a nivel de grandes implementaciones. Se puede asumir un número 0 como un número válido durante el desarrollo de la primer versión del documento. Un cambio en este dígito es solamente el resultado de un gran cambio en la aplicación, como ser:

- Cambios significativos en su forma de operar.
- Gran cantidad de funcionalidad añadida tal que se pueda asumir que el producto ha cambiado radicalmente.
- Una reescritura del producto.
- Un nuevo proyecto que se arma exclusivamente en torno al producto.

4.2.4.7. Versión de Funcionalidad

Este número de versión cambia cada vez que se cambia funcionalidad en el producto. Un buen ejemplo de esto puede ser el agregado de nuevas consultas al sistema, nuevos reportes, cambios realizados para mejorar su performance, etc.

En definitiva, cualquier tipo de cambio que no sea un ajuste menor o bien la corrección de un error.

4.2.4.8. Versión de Correcciones y Ajustes

El tercer y último número se utiliza para identificar la corrección de errores. Podemos asumir como errores a manifestaciones en el producto, errores de ortografía o semántica en un documento, un campo mal ubicado en una pantalla. Al reparar cualquiera de estos, el objeto de la corrección debe incrementar su número de versión de correcciones y ajustes.

Los cambios en las versiones variarán con la evolución de los componentes hasta el momento de su aprobación.

4.2.4.9. Versionado de Componentes de Software

Para el presente proyecto, la mayor cantidad de componentes a construir se relacionan con clases de objetos. Pocos componentes quedan fuera de dicha categoría y podemos citar: scripts para la creación de la base de datos, scripts de inicialización de la base de datos, etc.

Las clases tanto de interfaz, control y dominio que conformen el sistema se generarán dentro de un ambiente integrado de desarrollo (en inglés IDE). En este caso particular, la herramienta de desarrollo será IBM VisualAge for Java. Se recomienda el uso de dicha herramienta porque entre otras características importantes, contiene un esquema de versionado de los componentes en todos sus niveles: por clase, por paquete de clases y por proyecto.

De esta forma se define a continuación un esquema de versionado para dichos componentes.

Los componentes del sistema deberán existir dentro de un único proyecto denominado "ITBA Tesis de Master". Todas las versiones (en cualquiera de los niveles antes mencionados) deberán constar de 3 dígitos: X.X.X. El primero indicará la versión global del sistema (cuantas veces se ha iniciado un proceso de desarrollo o de mantenimiento radical sobre el sistema. El segundo indica cambios en la funcionalidad del sistema y el tercero corrección de errores. Podemos anticipar entonces que una vez terminado el sistema nos encontraremos con la versión 1.0.X dado que pueden surgir ciertos cambios a raíz de los resultados del plan de pruebas.

4.2.4.10. Definición del ciclo de estados para cada tipo de producto y los criterios de trazabilidad entre los mismos

Dadas las características del proyecto, los documentos estarán en uno de dos estados:

- **En edición:** el componente esta siendo desarrollado o bien corregido. Es un estado de evolución.
- **Aprobado:** por el director de tesis.

La trazabilidad de los componentes tiene que ver con la identificación de la historia de su evolución. Para este caso, esto significa poder identificar qué documento sirvió de base a otro. Dado que se utiliza métrica versión 3 para el presente desarrollo, será necesario referirse a la documentación relacionada para verificar la trazabilidad. Por otro lado, no se guardará historia alguna de trazabilidad de versiones para ningún componente del sistema.

4.2.4.11. Identificación de la información necesaria de control para auditoría del Control de la Configuración

En este apartado se definen las cuestiones relacionadas a la información que haría falta guardar para una potencial auditoría de la gestión de la configuración. Como ejemplo se puede mencionar: almacenar información sobre las personas que trabajaron en un determinado documento, almacenar fecha y hora de todas las modificaciones, etc.

Dado que en este caso todos los documentos son generados y modificados por el alumno, no se indagará en detalles al respecto, aunque cabe recordar que existe software comercial que automatiza esta cuestión (ej: TeamConnect de IBM).

4.2.5. Especificación del Entorno Tecnológico para la Gestión de Configuración

En esta tarea se define el entorno tecnológico de soporte al plan de gestión de la configuración y se determinan los componentes hardware y software que van a permitir la mecanización de los procesos y controles que establece el plan. Si bien es cierto que existen herramientas que dan soporte automático a dicho proceso, este no es el caso dado que existe una única persona que genera, mantiene y modifica los componentes del proyecto.

De todas maneras, hay que recalcar que en proyectos donde participa una gran cantidad de profesionales y usuarios, estas herramientas se vuelven indispensables para evitar problemas como ser repetición de documentos, pérdidas de versiones o modificaciones de código fuente, esquemas de base de datos, etc.

En este caso se utilizará una planilla electrónica que contenga el mapa de configuración de los componentes del sistema y se utilizarán rótulos en los

documentos que indiquen la información relevante al control de la configuración. No se utilizarán herramientas informáticas específicas.

4.2.6. Línea Base Funcional

La línea base funcional es un documento que indica en forma global y única las versiones de todas los componentes del sistema. En pocas palabras, para el presente proyecto se generará una tabla resumen que indique la sigla del componente correspondiente a la versión actual del sistema y su correspondiente versión. De esta forma, será fácil saber, por ejemplo, qué componentes de software deben ser instalados en el momento de su puesta a producción, qué información de análisis utilizar para entender el sistema actualmente en explotación, etc.

De ahora en más se hará referencia a dicho documento como *Configuración Global del Sistema*. Este documento deberá ser constantemente actualizado en cada proceso de revisión, o bien, con la generación de un nuevo documento o una nueva versión de uno ya existente.

Por fines prácticos, en el presente proyecto se presentará un único documento hacia el final de proyecto que indique cuál es finalmente la versión de cada uno de los componentes que conformen la tesis.

Cabe destacar que todos los componentes serán considerados como "en desarrollo" en tanto y en cuanto no figuren en el documento de Configuración Global del Sistema. Por otro lado, la presencia de un componente en el mismo denota su carácter de aprobado.

A continuación se presenta un ejemplo de documento:

<i>Código de Componente</i>	<i>Tipo de Componente</i>	<i>Descripción del Componente</i>	<i>Versión Aprobada del Componente</i>
ASI	Documento	Análisis del Sistema de Información	1.0.0
DSI	Documento	Diseño del Sistema de Información	1.0.2
CLI	Binario	Ejecutables	1.1.0

4.2.7. Registro de Instalaciones

El sistema tendrá un componente cliente que deberá ser distribuido e instalado en la ubicación. Para mantener registro de ello, deberá generarse un registro de instalaciones. Si bien la instalación y explotación del sistema no se encuentra dentro del alcance del presente proyecto, se define a continuación un registro de instalación típico para su potencial uso.

Registro de Instalaciones			A fecha/...../.....
Código de Producto	Instalación / Cliente	Fecha de Instalación	Versión N°

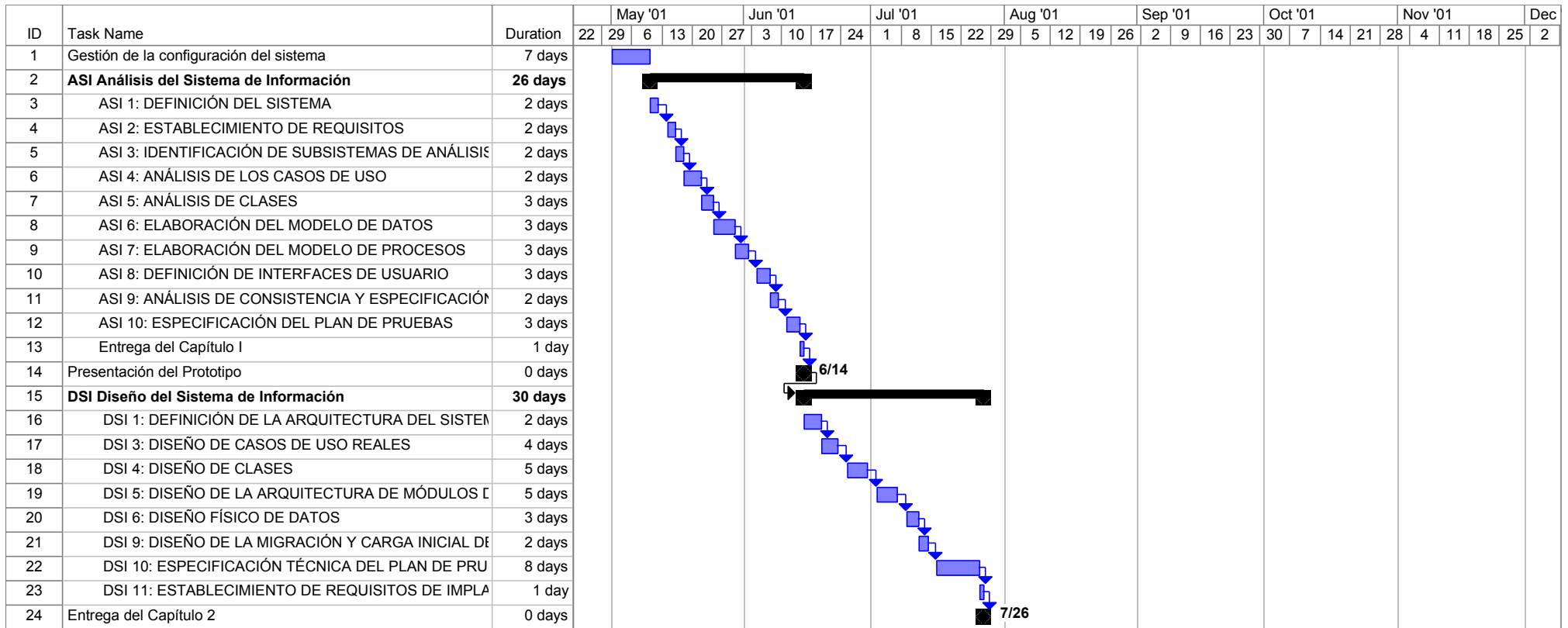
4.3. Planificación del Trabajo

En esta actividad se asignan fechas para la realización de las distintas actividades que conforman el proyecto. Esta asignación no debe hacerse en base a suposiciones por parte del líder de proyecto, sino que deben basarse en algún método de estimación.

Por lo general estas técnicas de estimación se basan en datos recolectados en proyectos anteriores de forma tal de conocer el costo (en término de recursos necesarios) para la construcción de los distintos componentes del sistema.

Dado que se carece de dicha base de datos para el presente proyecto, el plan que se presenta a continuación es de carácter tentativo en cuanto a la planificación temporal de las actividades definidas para el ciclo de vida del presente proyecto.

Capítulo IV - Planificación General del Proyecto
PGP Versión 1.1.1



Capítulo V

Análisis del Sistema de Información

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: LBANALISIS
Fase de Origen:	Análisis del Sistema de Información
Nombre o Código del Elemento de Configuración de Software:	Análisis del Sistema de Información (ASI)
Versión N°: 1.0.3	Entrega N°: 06
Fecha de Última Modificación:	14/02/2003
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

5. Análisis del Sistema de Información

En el presente capítulo se desarrollan las cuestiones relacionadas con el análisis del sistema de información. Para ello se parte de la base de describir el proceso y establecer objetivos, para luego explotar cada una de las actividades necesarias.

5.1. Descripción y Objetivos

El objetivo de este proceso es la obtención de una especificación detallada del sistema de información, de forma que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema.

Al ser Métrica v3 una metodología que cubre tanto desarrollos estructurados como orientados a objetos, las actividades de ambas aproximaciones están integradas en una estructura común.

En la primera actividad, Definición del Sistema (ASI 1), se lleva a cabo la definición inicial del sistema de información. Se delimita el alcance del sistema, se genera un catálogo de requisitos generales y se describe el sistema mediante unos modelos iniciales de alto nivel. También se identifican los usuarios que participarán en el proceso de análisis, determinando sus perfiles y responsabilidades, así como la dedicación que va a ser necesaria por su parte y, se elabora el plan de trabajo a seguir.

La definición de requisitos del nuevo sistema se realiza principalmente en la actividad Establecimiento de Requisitos (ASI 2). El objetivo de esta actividad es definir un catálogo de requisitos detallado que permita describir con precisión el sistema de información, y que además sirva de base para comprobar que es completa la especificación de los modelos obtenidos en las

actividades Identificación de Subsistemas de Análisis (ASI 3), Análisis de Casos de Uso (ASI 4), Análisis de Clases (ASI 5), y Definición de Interfaces de Usuario (ASI 8).

Hay que hacer constar que estas actividades pueden provocar la actualización de dicho catálogo, aunque no se reflejará como producto de salida en las tareas de dichas actividades, ya que el objetivo de las mismas no es crear el catálogo sino definir modelos que soporten los requisitos de dicho catálogo.

Para la obtención de requisitos:

Se tomará como punto de partida el catálogo de requisitos y los modelos elaborados en la actividad Definición del Sistema (ASI 1), que se completarán mediante sesiones de trabajo con los usuarios.

Las sesiones de trabajo tienen como objetivo obtener la información necesaria para la especificación detallada del nuevo sistema. Las técnicas que ayudan a la recopilación de esta información pueden variar en función de las características del proyecto y tipos de usuario a entrevistar. Entre ellas podemos citar las reuniones, entrevistas, *Joint Application Design (JAD)*, etc.

Durante estas sesiones de trabajo se propone utilizar la especificación de los casos de uso como ayuda y guía en el establecimiento de requisitos. Esta técnica facilita la comunicación con los usuarios, y en el análisis orientado a objetos, constituye la base de la especificación.

Se identificarán los requisitos no funcionales del sistema, es decir, las facilidades que ha de proporcionar el sistema, y las restricciones a que estará sometido, en cuanto a rendimiento, frecuencia de tratamiento, seguridad y control de acceso, etc.

Esta información se incorpora también al catálogo de requisitos.

En la actividad Identificación de Subsistemas de Análisis (ASI 3), se estructura el sistema de información en subsistemas de análisis, para facilitar la especificación de los distintos modelos y la traza de requisitos.

En paralelo, se generarán los distintos modelos que servirán de base para el diseño; en el caso de análisis estructurado, se procederá a la elaboración y descripción detallada del modelo de datos y de procesos, y en el caso de un análisis orientado a objetos, del modelo de clases y el comportamiento dinámico, mediante el análisis de los casos de uso. Se especificarán,

también, todas las interfaces entre el sistema y el usuario, como formatos de pantallas, diálogos, formatos de informes y formularios de entrada.

En la actividad Análisis de Consistencia y Especificación de Requisitos (ASI 9), se realiza una verificación y validación de los modelos, con el fin de asegurar que son:

- Completos: cada modelo obtenido contiene toda la información necesaria recogida en el catálogo de requisitos.
- Consistentes: cada modelo es coherente con el resto de los modelos
- Correctos: cada modelo sigue unos criterios de calidad predeterminados, respecto a la técnica utilizada, calidad de los diagramas, elección de nombres, normas de calidad, etc.

Como resultado de esta actividad se elabora la Especificación de Requisitos Software (ERS), producto que constituye un punto de referencia en el desarrollo del software para formalizar las peticiones de cambio sobre los requisitos inicialmente especificados.

En la actividad Especificación del Plan de Pruebas (ASI 10), se establece el marco general del plan de pruebas, y se inicia su especificación, que se completa en el proceso Diseño del Sistema de Información (DSI).

La participación activa de los usuarios es una condición imprescindible para el análisis del sistema de información, ya que dicha participación constituye una garantía de que los requisitos identificados son comprendidos e incorporados al sistema y, por tanto, de que éste será aceptado. Para facilitar la colaboración de los usuarios, se podrán utilizar técnicas interactivas, como diseño de diálogos y prototipos, que permiten al usuario familiarizarse con el nuevo sistema y colaborar en la construcción y perfeccionamiento del mismo.

5.2. Definición del Sistema

Esta actividad tiene como objetivo hacer una descripción del sistema, delimitando su alcance, estableciendo las interfaces con otros sistemas e identificando a los usuarios representativos del sistema. Dicho objetivo se logra desarrollando las subactividades que se presentan a continuación.

5.2.1. Determinación del Alcance del Sistema

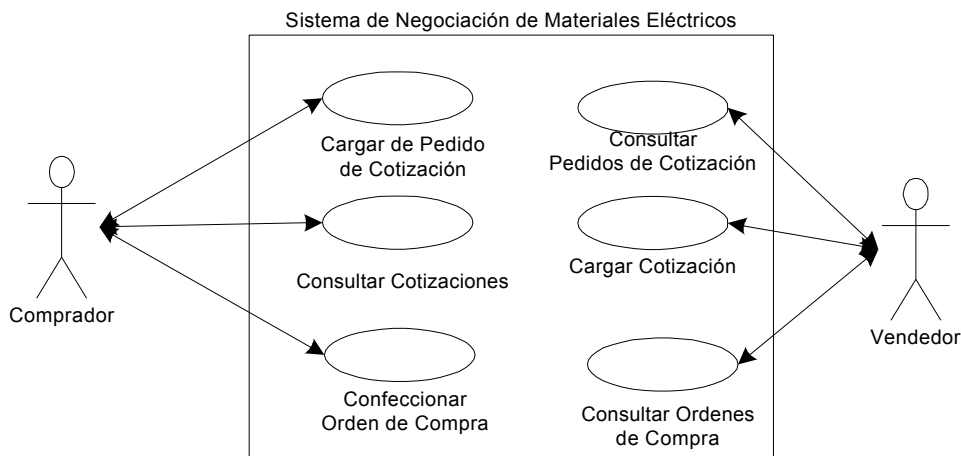
En esta tarea se delimita el sistema de información indicando qué procesos quedan dentro de su ámbito y se identifican las entidades externas al sistema que aportan o reciben información. Asimismo, se obtiene un modelo conceptual de datos identificando las entidades y relaciones que formarán parte del sistema de información objeto de este análisis.

Para el caso de análisis orientado a objetos, antes de la captura de requisitos a través de los casos de uso, puede ser conveniente establecer el contexto del sistema a partir del modelo de negocio. El modelo de negocio especifica los procesos a los que se quiere dar respuesta en el sistema de información, en forma de casos de uso de alto nivel.

En esta actividad se inicia, también, la definición del catálogo de requisitos.

A medida que se van generando los productos anteriores, se recomienda la definición de un glosario de términos, del ámbito de negocio, con el fin de conseguir una mayor precisión en la especificación del sistema de información. El glosario será un catálogo de términos general y común a todos los procesos y, susceptible de ser entrada o salida en cualquier tarea, de modo que en las restantes tareas se omitirá la referencia al mismo, por sencillez.

A continuación, el modelo de casos de uso de alto nivel:



5.2.2. Identificación del Entorno Tecnológico

El objetivo de esta tarea es definir, a alto nivel, el entorno tecnológico que se requerirá para dar respuesta a las necesidades de información, especificando los posibles condicionantes y restricciones.

Como resultado de esta tarea, el catálogo de requisitos se enriquece con los requisitos relativos al entorno tecnológico. Se define entonces el entorno tecnológico bajo el cual se desarrollará la aplicación con los requisitos relacionados.

En lo que refiere a la presente aplicación, es necesario tener en cuenta que la misma forma parte del desarrollo de una tesis de master, con lo cual, la misma deberá poder ser ejecutada dentro del ámbito de una universidad con las restricciones presupuestarias que de ello se deduce. Esta restricción indica que el software a utilizar para el desarrollo y sobre todo el software necesario para su posterior ejecución deberá ser en lo posible gratuito (por lo menos para un uso académico).

En base a esta premisa, se deberá utilizar software del tipo gratuito (GNU) y bajo el sistema operativo LINUX, también gratuito.

Requerimientos de Hardware y Software

Descripción del Requerimiento	Nro. Req.	Volatilidad	Necesidad	Prioridad
El sistema debe ser desarrollado utilizando software de tipo GNU (gratuito para uso académico) de manera de poder ser utilizado posteriormente por las universidades como implementación de referencia.	R001	Baja	Alta	1
El sistema debe poder ser instalado en un equipo del tipo PC (hardware).	R002	Baja	Alta	1
El sistema debe ser desarrollado para el sistema operativo LINUX.	R003	Baja	Alta	1
Para las comunicaciones entre componentes del sistema se deberá utilizar el protocolo TCP/IP	R004	Baja	Alta	1
El sistema debe ser desarrollado para el sistema operativo Microsoft Windows.	R005	Baja	Alta	1
Los componentes del sistema deberán cumplir con las restricciones de rendimiento aún al ser utilizados sobre una red de 10 Mbps.	R006	Baja	Alta	1
El sistema deberá ser desarrollado para su uso en	R007	Baja	Alta	1

Interenet (simulado dentro del ambiente de la universidad)				
El sistema deberá correr en una modalidad ICQ para facilitar su uso.	R008	Baja	Alta	1

5.3. Establecimiento de Requisitos

En esta actividad se lleva a cabo la definición, análisis y validación de los requisitos a partir de la información especificada por el usuario, completando el catálogo de requisitos obtenido en la actividad Definición del Sistema (ASI 1). El objetivo de esta actividad es obtener un catálogo detallado de los requisitos funcionales y no funcionales (de rendimiento, restricciones de diseño, seguridad, etc.), a partir del cual se pueda comprobar que los productos generados, en las actividades de modelización, se ajustan a los requisitos de usuario.

Esta actividad se descompone en un conjunto de tareas que, si bien tienen un orden, exigen continuas realimentaciones y solapamientos entre sí y con otras tareas realizadas en paralelo. No es necesaria la finalización de una tarea para el comienzo de la siguiente. Lo que se tiene en un momento dado es un catálogo de requisitos especificado en función de la progresión del proceso de análisis. Por tanto, cuando este catálogo de requisitos se requiera como producto de entrada para otra tarea, no se especificará la tarea concreta en la que se obtiene como producto de salida, sino que se referirá al estado más avanzado de dicho catálogo.

Para el caso, por tratarse de un sistema orientado a objetos, se utilizará la técnica de obtención de requisitos basada en los casos de uso.

5.3.1. Obtención de Requisitos

Se recoge información de los requisitos que debe cumplir el software desde el punto de vista de la funcionalidad, así como de aspectos no funcionales que se deben satisfacer para obtener una interacción eficaz con otros elementos del sistema, como el usuario, el hardware u otros productos software. En la definición de los requisitos no funcionales, que en general servirán de base para establecer los niveles de servicios que exigirá el sistema, se tendrán en cuenta, si existen, las posibles restricciones del

entorno, tanto hardware como software, que puedan afectar al sistema de información.

También se definen las prioridades a asignar a los requisitos, considerando los criterios de los usuarios acerca de las funcionalidades a cubrir. Los principales tipos de requisitos que se deben de especificar son, por ejemplo:

- Funcionales
- Rendimiento
- Seguridad
- Implantación
- Disponibilidad del sistema

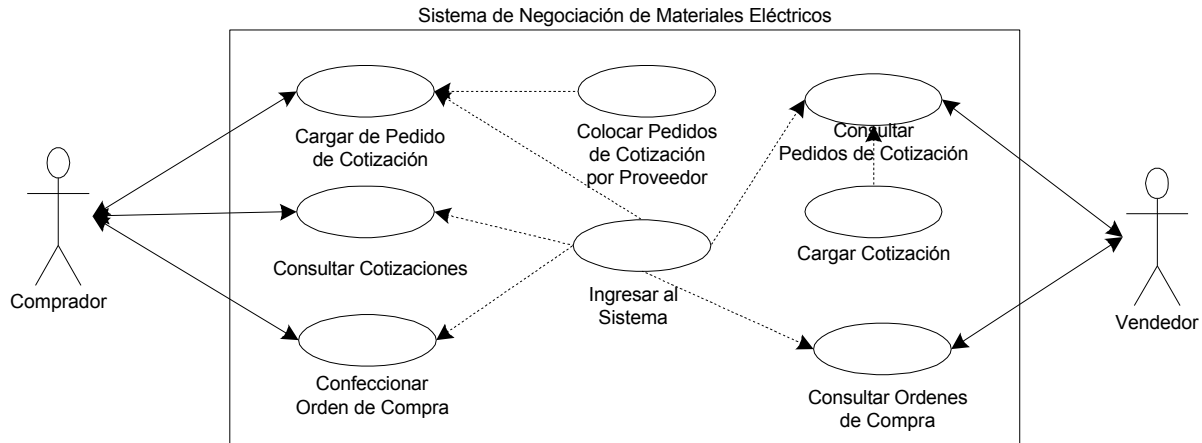
En el caso de orientación a objetos, se especificarán además los casos de uso asociados a los requisitos funcionales.

Los casos de uso son una técnica de especificación de requisitos, válida tanto en desarrollos estructurados como en orientación a objetos, aunque en este último caso se propone como técnica obligatoria al ser necesaria como referencia a lo largo de todo el ciclo de vida. En esta tarea se elabora el modelo de casos de uso, identificando:

- Actores
- Casos de uso
- Breve descripción de cada caso de uso

Los productos obtenidos en la tarea Determinación del Alcance del Sistema (ASI 1.1), se tomarán como referencia durante la obtención de requisitos, de forma que todos los requisitos especificados se encuentren dentro del ámbito del sistema de información.

A continuación se define el diagrama de casos de uso para el sistema a desarrollar:



En el diagrama se pueden identificar claramente los grandes tipos de usuarios que utilizarán el sistema de mercados virtuales.

Se definen:

- *Compradores de materiales eléctricos*: es la persona / organización que ingresa al sistema impulsado por la necesidad de adquirir materiales eléctricos en el mercado mayorista.
- *Vendedores de materiales eléctricos*: son las personas u organizaciones cuyo objetivo es vender materiales eléctricos en el mercado mayorista.

Los casos de uso definen la funcionalidad que debe brindar el sistema. A continuación se detalla una breve explicación sobre cada caso de uso, explicación que será profundizada en la próxima tarea de la metodología:

- *Ingresar al sistema*: la información conservada dentro del sistema es privada y crítica. Por lo tanto un punto funcional muy importante es identificar al usuario que esta intentando ingresar al sistema para filtrar la información relacionada con el mismo. Así también como la funcionalidad, dado que el vendedor usará ciertos casos de uso distintos al comprador.
- *Cargar pedido de cotización*: el comprador de materiales eléctricos ingresará un pedido de cotización. Para ello deberá buscar los productos, ingresar la cantidad necesitada, un precio potencial y una fecha de entrega. Una vez ingresados los productos se guardará el pedido de cotización.

- *Colocar pedidos de cotización por proveedor*: el sistema deberá identificar quienes son los proveedores que potencialmente pueden proveer los productos, según una catalogación previa de los mismos, y generar los pedidos de cotización dentro del sistema.
- *Consultar pedidos de cotización*: los proveedores ingresarán al sistema para identificar los pedidos de cotización a responder.
- *Cargar cotización*: una vez identificada la cotización a responder, el proveedor deberá ingresar en firme la cantidad de unidades del producto en cuestión, así como el precio en firme y la fecha de entrega.
- *Confeccionar orden de compra*: el comprador, una vez que obtenga todos los pedidos de cotización, deberá ingresar el o los pedidos en firme. El sistema deberá facilitar dicha confección.
- *Consultar órdenes de compra*: los proveedores, ingresarán al sistema para consultar los pedidos que les hayan sido solicitados.
- *Consultar cotizaciones*: en cualquier momento, los compradores pueden ingresar al sistema y consultar las cotizaciones que los proveedores hicieron sobre cada uno de sus pedidos de cotización.

Resumiendo, se presentan a continuación los requisitos que se agregan al catálogo de requisitos:

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RF01	Cargar un pedido de cotización	Baja	Alta	5
RF02	Colocar pedidos de cotización por proveedor	Baja	Alta	5
RF03	Consultar pedidos de cotización	Baja	Alta	5
RF04	Cargar una cotización	Baja	Alta	5
RF05	Confeccionar una orden de compra	Baja	Alta	5
RF06	Consultar órdenes de compra	Baja	Alta	5
RF07	Consultar cotizaciones	Baja	Alta	5

Por otro lado se definen los requisitos no funcionales relacionados con la aplicación:

Requerimientos de Cumplimiento de Normas y Estándares

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFN01	El sistema debe ser desarrollado utilizando la metodología Métrica v3 teniendo en cuenta las características propias de la orientación a objetos.	Baja	Alta	5
RNFN02	Los componentes tanto cliente como servidor deben ser desarrollados en lenguaje Java para asegurar su portabilidad y robustez.	Baja	Alta	5
RNFN03	Los componentes de servidor deberán ser desarrollados utilizando el estándar de desarrollo propuesto por Sun para aplicaciones de Internet.	Baja	Alta	5

Requerimientos de Seguridad

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFS01	Los usuarios del sistema deben ingresar su nombre de usuario y clave para ingresar al sistema.	Alta	Alta	2
RNFS02	Las claves de los usuarios debe ser guardada en forma encriptada.	Baja	Alta	2

Requerimientos de Auditoría

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFA01	El sistema debe guardar información de cada usuario que ingresa al sistema (fecha, hora, usuario).	Baja	Alta	1
RNFA02	El sistema debe guardar información sobre cada acceso al servidor (independientemente del usuario) para poder verificar el volumen de transacciones (fecha, hora, transacción).	Baja	Alta	1

Requerimientos de Organización

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFO01	El software y las herramientas utilizadas para ejecutar el sistema deben ser gratuitas de forma tal de poder utilizar este proyecto como material de estudio.	Baja	Alta	1

Requerimientos de Backup, Contingencias y Recuperación de Errores

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFB01	Los datos de gestión del sistema deberán encontrarse centralizados de forma tal de facilitar su backup y recuperación.	Baja	Alta	5

Requerimientos de Rendimiento

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFR01	Ninguna operación del sistema puede superar los 5 segundos de espera para el usuario.	Baja	Alta	1
RNFR02	El sistema deberá ser diseñado de forma tal de permitir el acceso concurrente de gran cantidad de usuarios.	Baja	Alta	5

5.3.2. Especificación de Casos de Uso (Tarea ASI 2.2)

El objetivo de esta tarea es especificar cada caso de uso identificado en la tarea anterior, desarrollando el escenario. Para completar los casos de uso será preciso especificar información relativa a:

- Descripción del escenario, es decir, cómo un actor interactúa con el sistema, y cual es la respuesta obtenida.
- Precondiciones y postcondiciones.
- Identificación de interfaces de usuario.
- Condiciones de fallo que afectan al escenario, así como la respuesta del sistema (escenarios secundarios).

A continuación, se tratará la especificación en detalle de los casos de uso.

Ingresar al Sistema (CU01)

Descripción

El usuario deberá ingresar su nombre de usuario y su clave de acceso para ingresar al sistema. Si el usuario es válido (existente en la base de datos del sistema) y la clave es la indicada para dicho usuario, entonces el sistema procederá a mostrar el menú principal del sistema personalizado para el usuario en cuestión (proveedor o comprador).

Precondiciones

El par de datos [usuario, clave] debe relacionarse con un usuario dado de alta previamente en el sistema. Cabe aclarar que en esta versión del software, aquellos nuevos usuarios que quieran comprar o vender por el sistema deberán ser registrados por el administrador del sistema. Esto se debe a que sería fácil que un vendedor ingrese como un comprador para utilizar al sistema como mecanismo para averiguar los precios y disponibilidad de la competencia, lo cual repercutiría en una reducción en cuanto al uso por parte de los proveedores honestos y consecuentemente la baja del sistema. En tanto y en cuanto se popularice el uso de tecnologías de autenticación en Internet (certificados digitales x.509 por ejemplo) la incorporación de nuevos usuarios al sistema podría ser automatizable.

Postcondiciones

El usuario se encuentra "autenticado" en el sistema.

Identificación de Interfaces de Usuario

Ingreso al Sistema



Menú Principal del Comprador



Menú Principal del Vendedor



Condiciones de Fallo

1. El nombre de usuario puede ser erróneo (inexistente en la base de usuarios).
2. La clave no es la correspondiente al nombre de usuario ingresado.

Cargar Pedido de Cotización (CU02)

Descripción

El usuario seleccionará la opción de "Crear un Nuevo Pedido de Cotización" desde el menú principal. El sistema mostrará una pantalla con los datos del nuevo pedido a crear: la lista de productos ingresados en el pedido de cotización. En este momento el usuario podrá agregar nuevos productos como así eliminar productos ya ingresados. En el caso de seleccionar ingresar un nuevo producto, el sistema permitirá buscar los productos en la base de datos, ya sea por medio de su código o bien por su descripción. Una vez identificado el producto, el usuario deberá ingresar la cantidad solicitada, la fecha de entrega y el precio tentativo. En el caso de borrar productos, el usuario seleccionará los productos a eliminar y el sistema los borrará del presente pedido de cotización. Una vez confeccionado el pedido, el sistema procederá a almacenarlo en la base de datos.

Precondiciones

1. El usuario debe estar autenticado en el sistema.

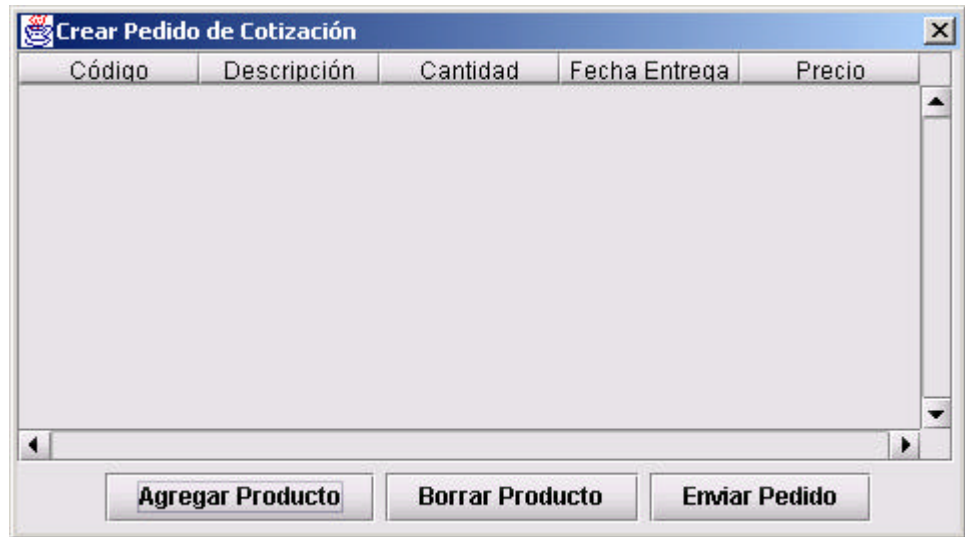
2. El usuario debe ser del tipo "comprador".

Postcondiciones

1. El pedido de cotización se encuentra ingresado.

Identificación de Interfaces de Usuario

Esta prime interfaz, presenta un pedido de cotización en blanco para luego mostrar todos los productos sobre los cuales el usuario necesita una cotización.

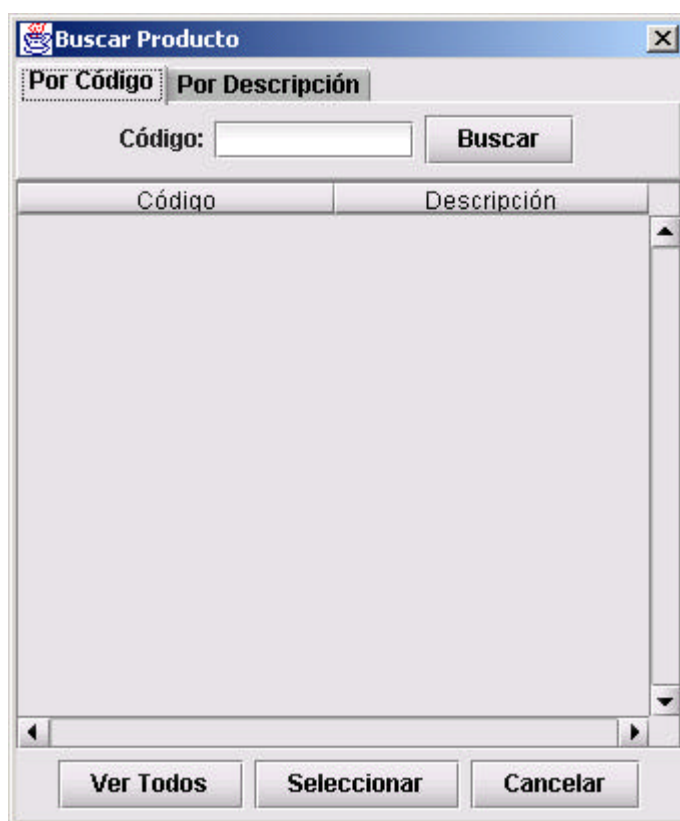


Al agregar un producto, el sistema abrirá el siguiente diálogo de carga.



Al presentarse la necesidad de seleccionar un producto, aparece el siguiente diálogo que permite buscarlos por código o descripción. No se presentan búsquedas por negocio o proveedor para que todos los

proveedores tengan las mismas posibilidades de participar en la cotización. En la presente versión no es posible excluir proveedores (por alguna razón particular).



Condiciones de Fallo

1. Al buscar un producto por código, el mismo puede ser inválido.
2. Al cargar la fecha de entrega, esta puede no ser una fecha válida (anterior a la actual, una combinación de día-mes-año inválida, etc.).
3. Al cargar la cantidad, esta puede no ser válida (la cantidad se define como un número entero y positivo).
4. Al borrar productos de la lista, puede no haberse seleccionado ningún producto.

Colocar pedidos de cotización por proveedor (CU03)

Descripción

Este caso de uso es desarrollado por el sistema en forma automática. Una vez almacenado un pedido de cotización en la base de datos, el sistema deberá identificar para cada producto que conforme el pedido de cotización, a todos los proveedores catalogados con el mismo. Luego deberá armar un pedido de cotización para cada uno de ellos teniendo en cuenta todos los productos del pedido de cotización que cada uno puede brindar.

Precondiciones

1. El usuario debe estar autenticado.
2. El usuario debe ser de tipo "comprador".
3. El pedido de cotización debe contener por lo menos un producto.

Postcondiciones

1. El pedido se encuentra "procesado".
2. Cada potencial proveedor tiene disponible su pedido de cotización.

Identificación de Interfaces de Usuario

No existen interfaces de usuarios asociadas a este caso de uso.

Condiciones de Fallo

1. Pueden no existir proveedores para un producto determinado en algún momento de la explotación del sistema.

Consultar Pedidos de Cotización (CU04)

Descripción

El proveedor ingresa al sistema; en este momento el sistema mostrará información sobre los clientes que han solicitado pedidos de cotización. Al seleccionar uno de estos pedidos el sistema mostrará su detalle, como ser: fecha de ingreso del pedido, nombre del potencial comprador y valoración del mismo (la sumatoria del producto de cantidad de artículos x su precio unitario).

Precondiciones

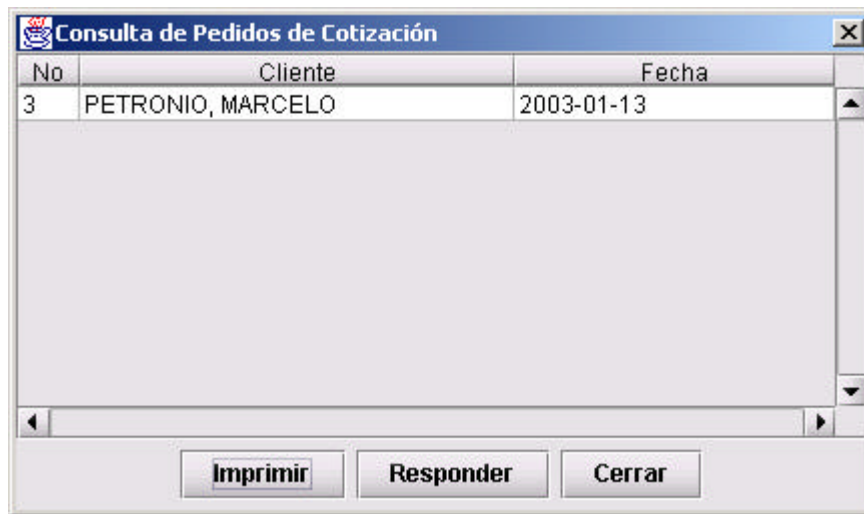
1. El usuario debe encontrarse autenticado por el sistema.
2. El usuario debe ser del tipo "vendedor".

3. Deben existir pedidos de cotización asignados al proveedor en cuestión.

Postcondiciones

No se observan.

Identificación de interfaces de Usuario



Condiciones de Fallo

No se observan.

Cargar Cotización (CU05)

Descripción

Este caso de uso se presenta como una extensión del anterior. Esto significa que añade funcionalidad a la consulta de pedidos de cotización. Para el caso, el usuario se encuentra consultando sus pedidos de cotización y puede seleccionar uno para contestarlo. El sistema comenzará un proceso por el cual guiará al usuario producto por producto de la cotización mostrando los datos cargados por el comprador (producto, fecha de entrega, precio tentativo, cantidad) y permitirá que el proveedor pueda confirmar dichos datos o bien cambiarlos por los que están a su alcance brindar. Una vez terminado el proceso, se encontrará disponible para el comprador la cotización.

Precondiciones

1. El usuario debe encontrarse autenticado por el sistema.
2. El usuario debe ser del tipo "proveedor".
3. Debe existir por lo menos un pedido de cotización para el proveedor en cuestión.

Identificación de Interfaces de Usuario

Producto	Fecha	Fecha Rta	Cantidad	Cant. Pedida	Precio	Precio Pedido
LAMPARA 10...	2003-01-14	2003-01-14	10.0000	10.0000	10.00	10.00

	Solicitado	Confirmado
Fecha de Entrega:	2003-01-14	14/01/2003
Cantidad:	10.0000	10.0000
Precio Unitario:	10.00	10.00

Condiciones de Fallo

1. El precio indicado por el proveedor no es válido (el precio es un número positivo).
2. La fecha de entrega ingresada por el proveedor no es válida (la cantidad es un número positivo).
3. La cantidad ingresada por el proveedor no es válida.
4. La fecha de vencimiento del presupuesto no es válida.

Consultar Cotizaciones (CU06)

Descripción

El comprador ingresa al sistema y selecciona la opción de consultar las cotizaciones que los proveedores hicieron sobre la base de sus pedidos. El sistema muestra una lista de todos los pedidos de cotización realizados: fecha, número, valuación, cantidad de pedidos de cotización enviados, cantidad de cotizaciones recibidas.. El usuario selecciona uno de sus pedidos y el sistema muestra una página con el detalle de la oferta de cada uno de los proveedores versus los datos ingresados por el comprador.

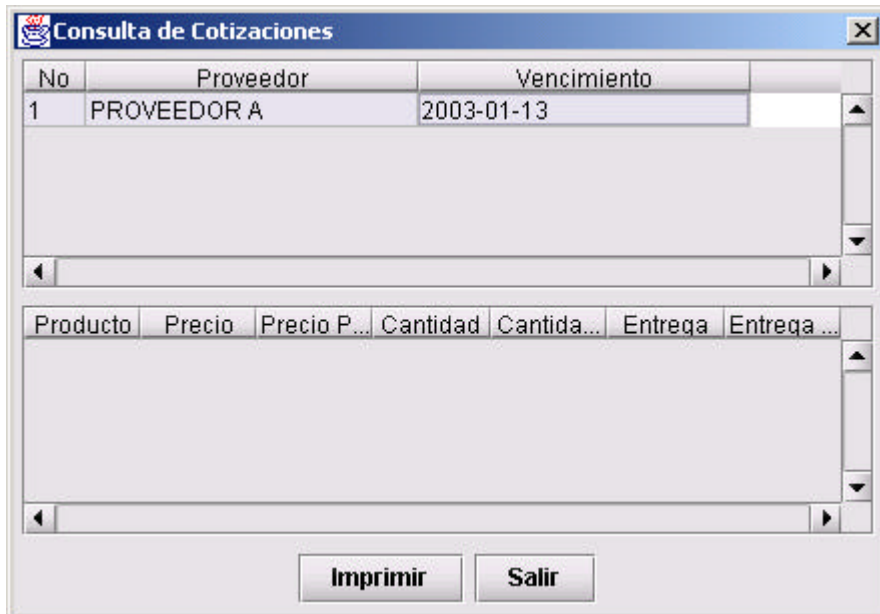
Precondiciones

1. El usuario debe encontrarse autenticado por el sistema.
2. El usuario debe ser de tipo "comprador".
3. El usuario debe haber ingresado por lo menos un pedido de cotización.

Postcondiciones

No se observan.

Identificación de Interfaces de Usuario



Condiciones de Fallo

1. El usuario puede presionar consultar detalle sin haber seleccionado una cotización.

Confeccionar Orden de Compra (CU07)

Descripción

Una vez que el comprador ha decidido cerrar una orden de compra, el mismo selecciona la opción de "Confeccionar una Orden de Compra". El sistema muestra todas los pedidos abiertos (sin una correspondiente orden de compra). El usuario selecciona uno y a continuación el sistema lleva al usuario por cada uno de los productos pedidos originalmente y una lista de los proveedores que cotizaron. El usuario seleccionará un proveedor para continuar con el siguiente producto. Al terminar el sistema mostrará una lista conteniendo as ordenes de compra a generar. Al confirmar dicha información, el sistema enviará las ordenes de compra a los proveedores.

Precondiciones

1. El usuario debe encontrarse autenticado por el sistema.
2. El usuario debe ser de tipo "comprador".
3. El usuario debe haber ingresado por lo menos un pedido de cotización en el sistema.
4. Para cada pedido de cotización debe haber por lo menos una cotización, aunque esta no cubra todos los productos.

Postcondiciones

1. El pedido de cotización se encuentra cerrado (no se aceptan más cotizaciones).
2. Las ordenes de compra se encuentran disponibles para los proveedores.

Identificación de Interfaces de Usuario

Producto	Cantidad	Precio	Entrega	Proveedor
----------	----------	--------	---------	-----------

Buttons: **Agregar**, **Borrar**, **Cancelar**, **Cargar desde Cotización**, **Enviar**

Producto: LAMPARA 100W

Proveedor	Cantidad	Precio	Entrega
-----------	----------	--------	---------

Cotizaciones:

Cantidad:

Precio:

Entrega: 13/01/2003

Buttons: **Aceptar**, **Cancelar**

Condiciones de Fallo

1. Pueden no existir cotizaciones para un pedido.
2. Puede haber productos no cotizados por ningún proveedor.

5.3.3. Análisis de Requisitos (Tarea ASI 2.3)

En esta tarea se realiza un proceso de análisis de la información capturada en esta actividad, para detectar inconsistencias, ambigüedades, duplicidad o escasez de información, etc.

También se analizan las prioridades establecidas por el usuario y se asocian los requisitos relacionados entre sí. El análisis de los requisitos y de los casos de uso asociados permite identificar funcionalidades o comportamientos comunes, reestructurando la información de los casos de uso a través de las generalizaciones y relaciones entre ellos.

Esta actividad se lleva a cabo en las reuniones de avance y análisis con el tutor de tesis. Por lo tanto no existen entregables en este apartado dado que en sí, esta actividad, resulta en nuevas versiones de los componentes ya desarrollados.

5.3.4. Validación de Requisitos (Tarea ASI 2.4)

Mediante esta tarea, los usuarios confirmarán que los requisitos especificados en el catálogo de requisitos, así como los casos de uso, son válidos, consistentes y completos.

Esta tarea es llevada a cabo con el / los usuarios del sistema. Por lo tanto se ha decidido que la misma consistirá en la corrección por parte del director de tesis sobre el trabajo realizado hasta el momento. De tal forma, no indagaremos más en la presente actividad, dado que dichas correcciones se reflejarán en posteriores versiones de los diagramas y documentos antes generados.

De todas formas es necesario destacar que muchas mejoras al sistema han sido resultado de sugerencias dadas por el **Ing. Ignacio Perez** quien se ha ofrecido como usuario para la validación del sistema, de ahora en más referenciado como "Usuario Líder".

Estas modificaciones abarcan varios aspectos, algunos que por su magnitud escapan a la posibilidad de desarrollarlos para la presente tesis, pero muchos que contribuyeron a la validación de requisitos y la formulación de muchos otros.

5.4. Identificación de Subsistemas de Análisis

El objetivo de esta actividad es facilitar el análisis del sistema de información llevando a cabo la descomposición del sistema en subsistemas. Se realiza en paralelo con el resto de las actividades de generación de modelos del análisis, por lo tanto, se asume una realimentación y ajuste continuo con respecto a la definición de los subsistemas, sus dependencias y sus interfaces.

A continuación se describe el análisis ya refinado y final del sistema a desarrollar:

5.4.1. Determinación de Subsistemas de Análisis

La descomposición del sistema en subsistemas debe estar, principalmente, orientada a los procesos de negocio, aunque también se pueden adoptar otros criterios lógicos. Entre los criterios que pueden ayudar a su identificación se encuentran los siguientes:

- Homogeneidad de procesos
- Servicios comunes
- Prioridad
- Afinidad de requisitos
- Localización geográfica.

Métrica v3 plantea formas tentativas para identificar subsistemas de análisis tanto para el análisis estructurado como para el análisis orientado a objetos.

En nuestro caso corresponde el segundo, donde se identifican y definen las dependencias entre subsistemas analizando los elementos compartidos entre ellos o las interfaces entre subsistemas. En el caso de que un subsistema se decida abstraer para su análisis como una unidad con una funcionalidad concreta, se podrá opcionalmente definir la interfaz de dicho subsistema, para poder delimitar su comportamiento y utilización en el modelo general del sistema. Por tanto, se establece como obligatoria la asociación entre subsistemas indicando sólo la dependencia. Además, opcionalmente, se propone la especificación de la interfaz de subsistemas de análisis, y la definición del comportamiento del sistema.

En ambos casos, se asignarán los requisitos y casos de uso a cada uno de los subsistemas identificados, actualizando el catálogo de requisitos.

5.4.1.1. Descripción de los subsistemas de análisis

Para el caso de nuestro sistema, es posible identificar claramente dos grandes conjuntos de funcionalidad o subsistemas:

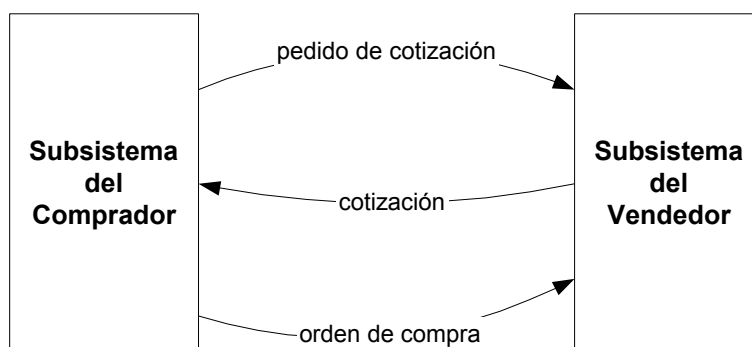
- Subsistema del comprador
- Subsistema del vendedor

En este caso la división se basa en los principios de "servicios comunes" y "homogeneidad de procesos" propuestos por Métrica v3. Se pueden desechar los otros principios por la siguiente explicación. El sistema a desarrollar consiste en una secuencia de pasos necesarios para llegar a una orden de compras (del inglés *workflow* - ver en el capítulo 1 el apartado que trata los tipos de mercados virtuales), que consisten en: pedir cotizaciones, responderlas por parte de los proveedores, seleccionarlas y confeccionar las órdenes de compra por parte de los consumidores y ser recibidas y procesadas por los proveedores.

Esto significa que cada funcionalidad que brinda el sistema genera una postcondición que es una precondition para otra. Por eso no es posible determinar la división de subsistemas basada en afinidad de requisitos, dado que si bien muchos son compartidos por todos los casos de uso (ej: usuario autenticado por el sistema) otros no lo son y estos definen la característica de secuencia propia del sistema. Por otro lado no se habla de prioridad dado que existe una secuencia clara de actividades a desarrollar, todas igualmente prioritarias e importantes. Al estar hablando de un proceso totalmente electrónico y accesible por Internet, no es posible agrupar funcionalidad sobre una base de localización geográfica alguna ya que no hace diferencia alguna para el sistema.

5.4.1.2. Descripción de interfaces entre subsistemas

Ahora bien, para hablar sobre la interfaz entre los subsistemas debemos hablar de las objetos (información) que viajan entre ambos. A continuación se presenta un diagrama que explica dicha interacción.



Cabe destacar que existen otros flujos de información (Ej.: productos ofrecidos) que si bien son parte de la información que manejan ambos subsistemas, no se representan en el diagrama por quedar fuera del presente desarrollo los componentes necesarios para mantenerlos interactivamente. Los mismos serán cargados oportunamente por los procedimientos de carga inicial de datos.

5.4.2. Integración de Subsistemas de Análisis

El objetivo de esta tarea es la coordinación en la elaboración de los distintos modelos de análisis de cada subsistema, asegurando la ausencia de duplicidad de elementos y la precisión en la utilización de los términos del glosario. Esta tarea se realiza en paralelo con el resto de las actividades de elaboración de modelos del análisis, y permite tener una visión global y unificada de los distintos modelos.

Para el presente proyecto no se han separado los subsistemas para su análisis. Por lo tanto esta actividad carece de sentido.

5.5. Análisis de los Casos de Uso

El objetivo de esta actividad, que sólo se realiza en caso de desarrollos orientados a objetos, es identificar las clases cuyos objetos son necesarios para realizar un caso de uso y describir su comportamiento mediante la interacción de estos objetos.

Esta actividad se lleva a cabo para cada uno de los casos de uso contenidos en un subsistema de los definidos en la actividad Identificación de Subsistemas de Análisis (ASI 3). Las tareas de esta actividad no son realizadas de forma secuencial sino en paralelo, con continuas realimentaciones entre ellas y con las realizadas en las actividades

Establecimiento de Requisitos (ASI 2), Identificación de Subsistemas de Análisis (ASI 3), Análisis de Clases (ASI 5) y Definición de Interfaces de Usuario (ASI 8).

5.5.1. Identificación de Clases Asociadas a un Caso de Uso

En esta tarea se comienzan a identificar los objetos necesarios para desarrollar el caso de uso basándose en la especificación que tenemos del mismo. A partir del estudio del caso de uso, se extrae una lista de objetos candidatos a ser clases. Es posible que inicialmente no se disponga de la información necesaria para identificar todas, por lo que se hace una primera aproximación que, durante esta actividad y en el proceso de diseño, se irá refinando. Además, algunos de los objetos representan mejor la información del sistema si se les identifica como atributos en vez de como clases. Para poder diferenciarlas es necesario estudiar el comportamiento de esos objetos en el diagrama de interacción y además se deben tener en cuenta una serie de reglas como puede ser el suprimir palabras no pertinentes, con significados vagos o sinónimos.

Una vez definidas cada una de las clases se incorporan al modelo de clases de la actividad Análisis de Clases (ASI 5), donde se identifican sus atributos, responsabilidades y relaciones.

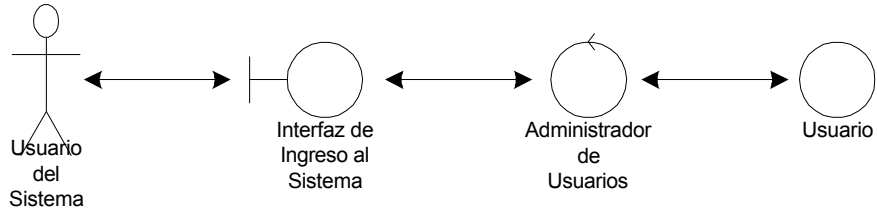
Las clases que se identifican en esta tarea pueden ser de entidad, interfaz de usuario y control. Las clases entidad representan la información manipulada en el caso de uso. Las clases de interfaz de usuario se utilizan para describir la interacción entre el sistema y sus actores. Suelen representar abstracciones de ventanas, interfaces de comunicación, formularios, etc. Las de control son responsables de la coordinación, secuencia de transacciones y control de los objetos relacionados con un caso de uso.

5.5.1.1. Diagramas de Clases

En este apartado, se utilizará la nomenclatura UML para realizar los diagramas de clases para cada uno de los casos de uso. Se comenzará por diferenciar los subsistemas de análisis y el diagrama de clases para cada uno de ellos. Si bien estos diagramas sufren un proceso iterativo de refinamiento, se presentan los diagramas definitivos para el sistema.

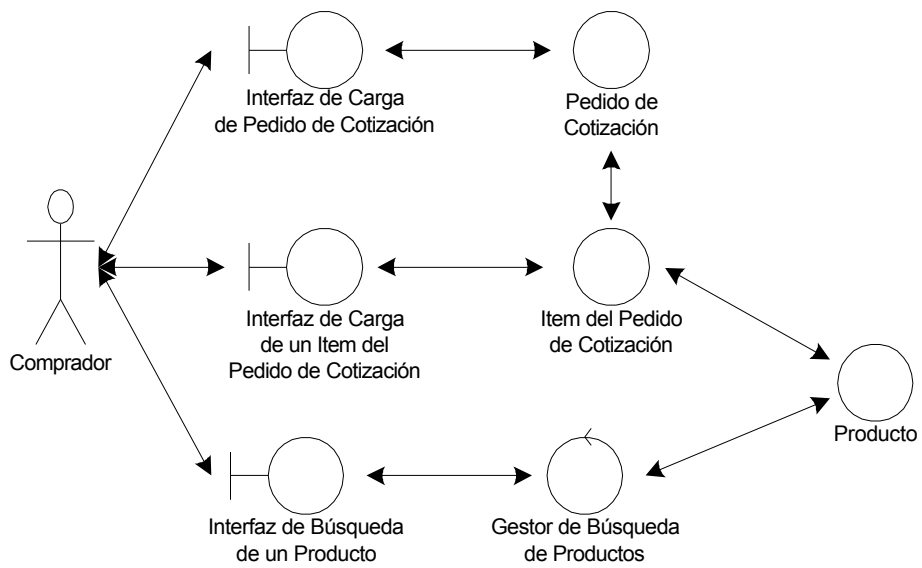
Casos de Uso Comunes a Ambos Subsistemas

Ingreso al Sistema (CU01)

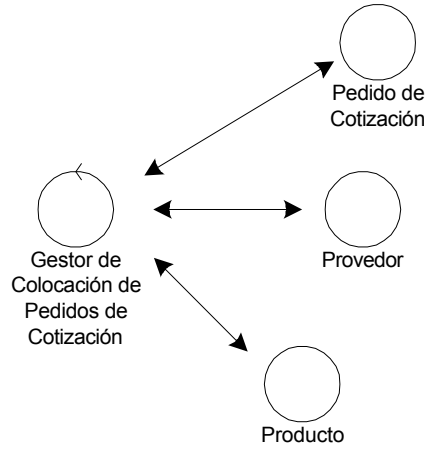


Subsistema del Comprador

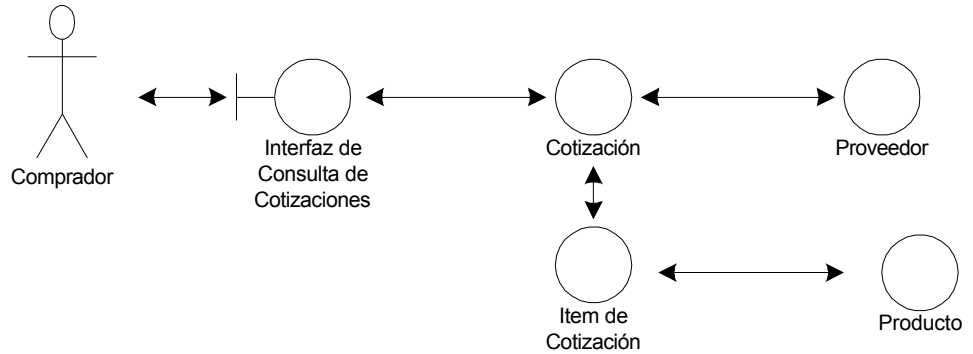
Cargar Pedido de Cotización (CU02)



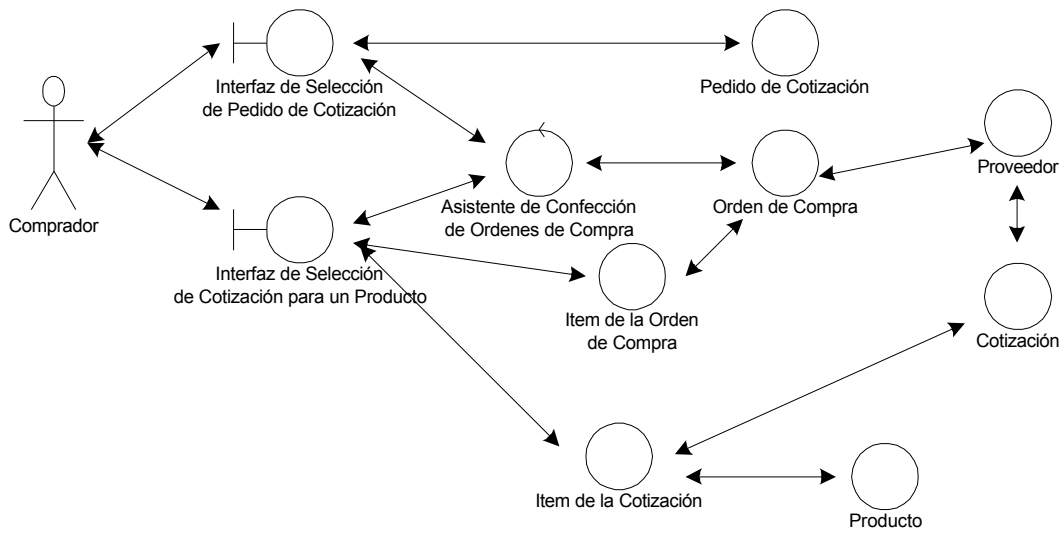
Colocar Pedido de Cotización por Proveedor (CU03)



Consultar Cotizaciones (CU04)

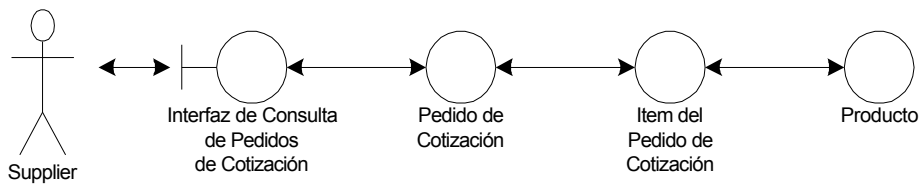


Confeccionar Orden de Compra (CU07)

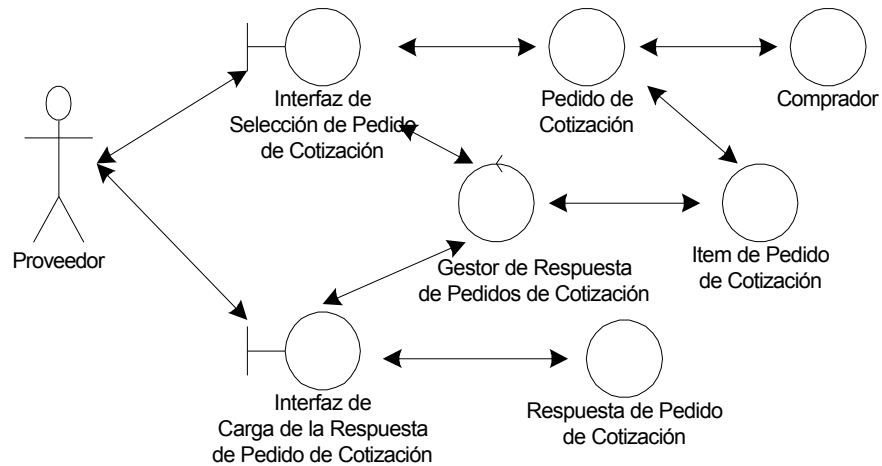


Subsistema del Vendedor

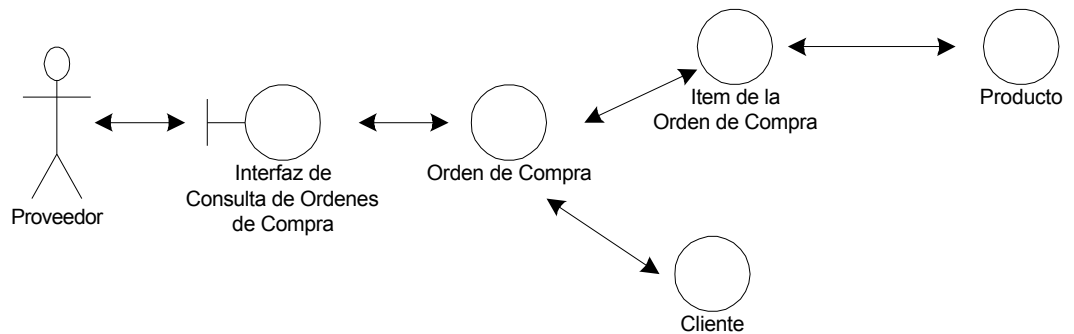
Consultar Pedidos de Cotización (CU04)



Cargar Cotización (CU05)



Consultar Ordenes de Compra (CU08)



5.5.2. Descripción de la Interacción de Objetos (Tarea ASI 4.2)

El objetivo de esta tarea es describir la cooperación entre los objetos utilizados para la realización de un caso de uso y que fueron identificados en la tarea anterior.

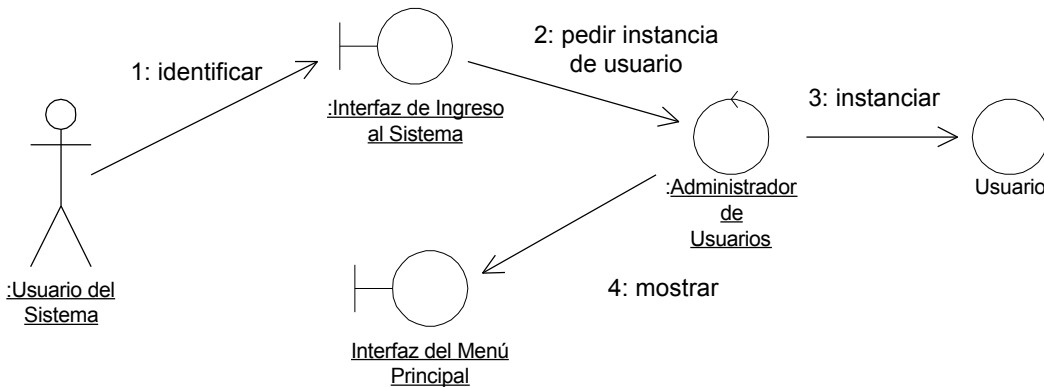
Para representar esta información se usan diagramas de interacción que contienen instancias de los actores participantes, objetos y la secuencia de mensajes intercambiados entre ellos. Se pueden establecer criterios para determinar que tipo de objetos y mensajes se van a incluir en este diagrama, como por ejemplo: si se incluyen objetos y llamadas a bases de datos, objetos de interfaz de usuario, de control, etc. Estos diagramas pueden ser tanto de secuencia como de colaboración, y su uso depende de sí se quieren centrar en la secuencia cronológica o en cómo es la comunicación entre los objetos.

Si bien el enfoque antes mencionado es el relevante a Métrica v3, la metodología UML plantea que los diagramas de secuencia corresponden con la etapa de diseño y el de colaboración a la etapa de análisis. Es por eso que se optará en este caso por utilizar diagramas de colaboración, dejando para un momento posterior el uso de los diagramas de secuencia.

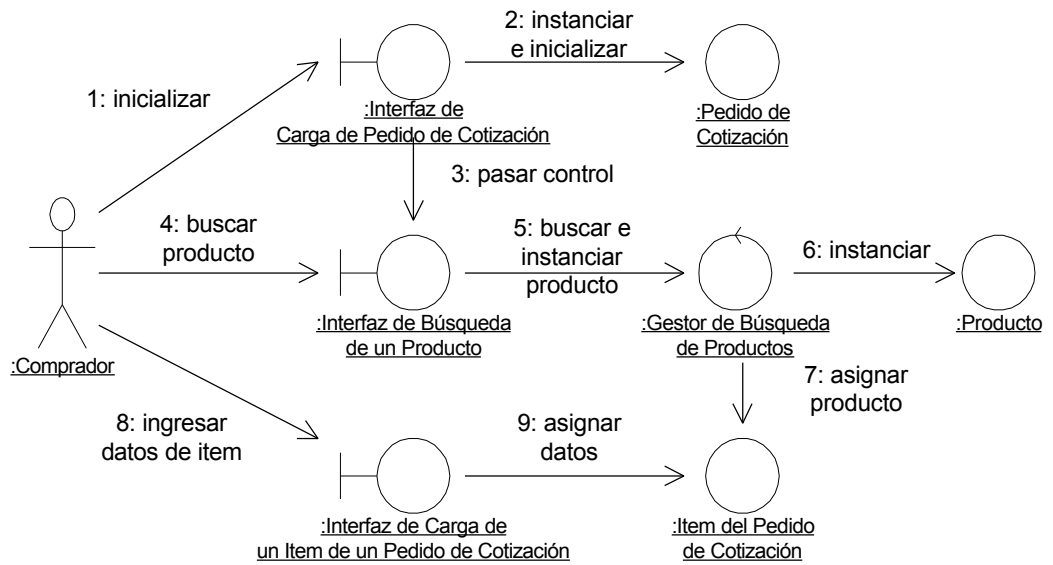
En aquellos casos en los que se especifique más de un escenario para un caso de uso puede ser conveniente representar cada uno de ellos en un diagrama de interacción.

También es recomendable, sobre todo en el caso anterior, completar los diagramas con una descripción textual.

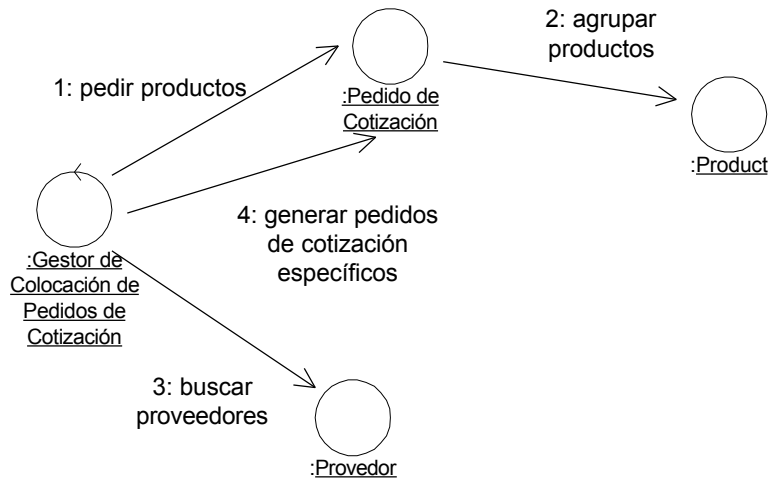
Ingresar al Sistema (CU01)



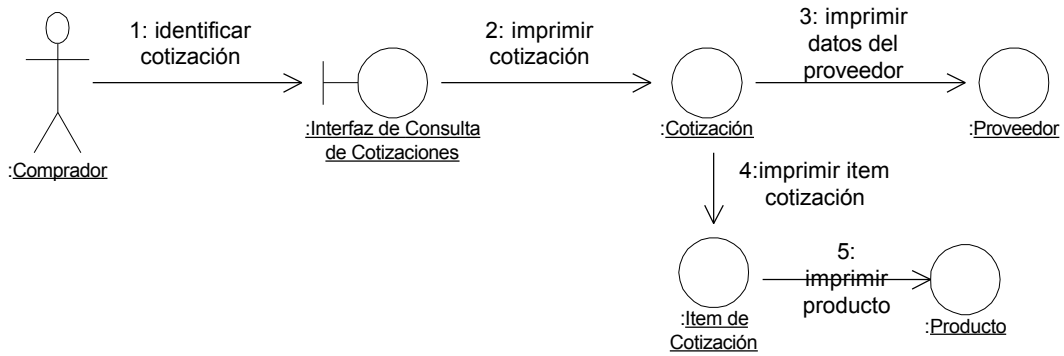
Cargar Pedido de Cotización (CU02)



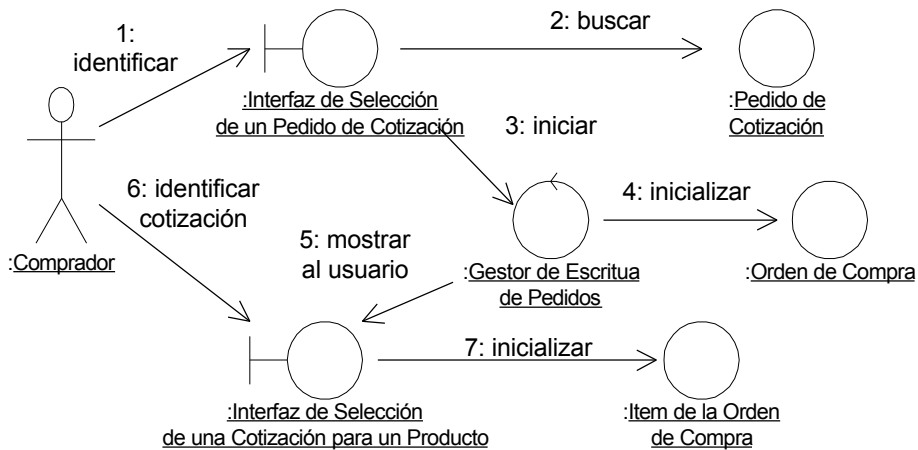
Colocar Pedido de Cotización por Proveedor (CU03)



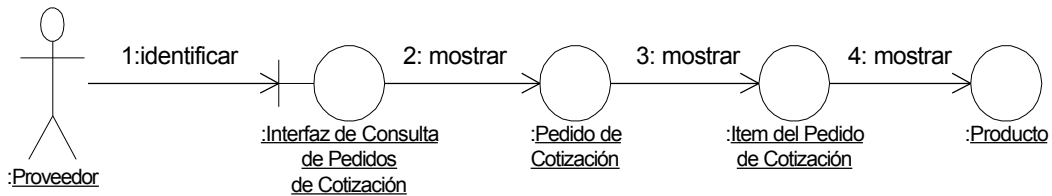
Consultar Cotizaciones (CU06)



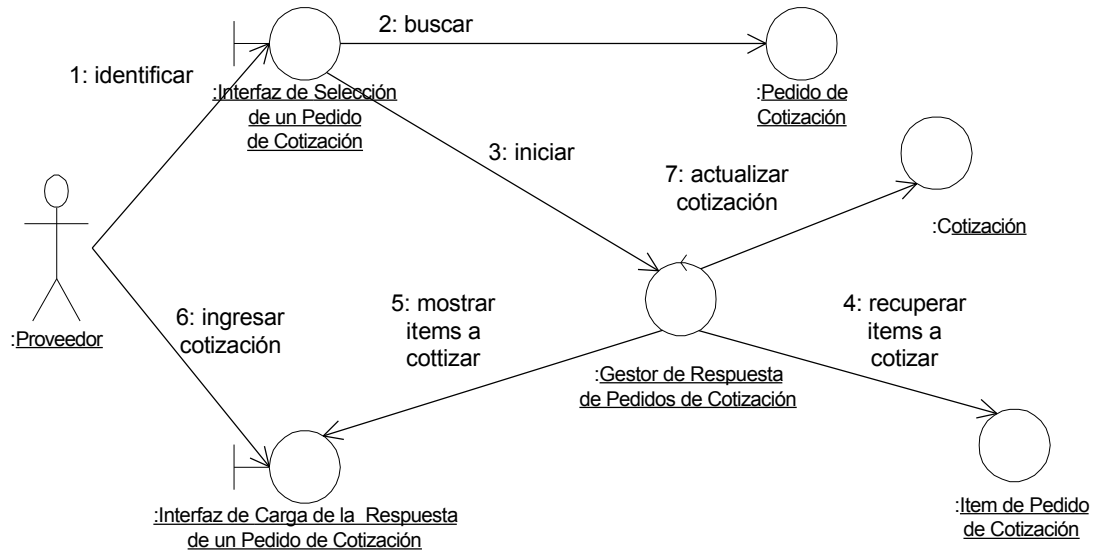
Confeccionar Orden de Compra (CU07)



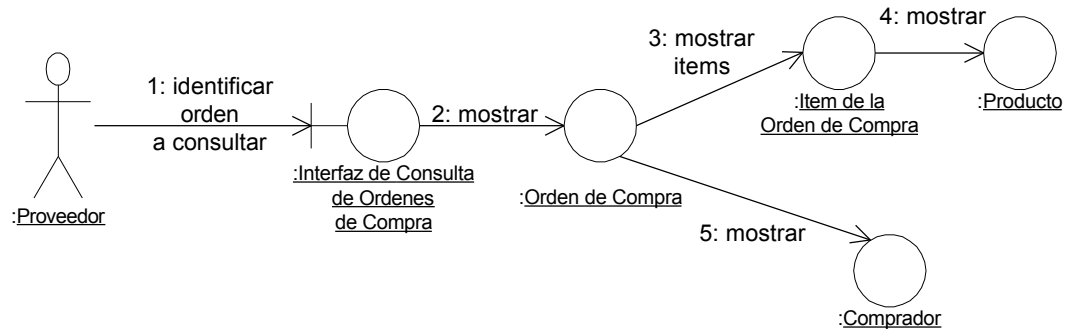
Consultar Pedidos de Cotización (CU04)



Cargar Cotización (CU05)



Consultar Ordenes de Compra (CU08)



5.6. Análisis de Clases

Teniendo en cuenta las clases identificadas en la actividad Análisis de los Casos de Uso (ASI 4), se elabora el modelo de clases para cada subsistema. A medida que avanza el análisis, dicho modelo se va completando con las clases que vayan apareciendo tanto del estudio de los casos de uso, como de la interfaz de usuario necesaria para el sistema de información.

El objetivo de esta actividad es describir cada una de las clases que han surgido identificando las responsabilidades que tienen asociadas, sus atributos y las relaciones entre ellas.

Según Jacobson [Jacobson 1992], el proceso de análisis debe llevar los siguientes pasos:

Paso	Descripción
Nombre de Clase	Se identifican las clases participantes y se define un nombre que identifica a cada una.
Atributos Lógicos	Se identifican los atributos que componen a cada clase en particular. No se representan aquellos atributos que serán necesarios para implementar una asociación.
Asociaciones Estáticas de Instancias	Se identifican las asociaciones estáticas, es decir, aquellas que representan composición.
Herencia	Se identifica la posibilidad de herencia entre clases basándose en su nivel de cohesión en cuanto al negocio y a los atributos de las clases identificadas.
Asociaciones Dinámicas de Instancias	Se identifican las asociaciones de objetos basadas en como estos se relacionan y actúan para llevar a cabo distintas responsabilidades.
Operaciones	Se identifican las operaciones (métodos) de los objetos.

Este proceso es sometido a un refinamiento iterativo hasta alcanzar un cierto nivel de conformidad. En los siguientes apartados se presenta el producto final obtenido de dicho refinamiento.

A continuación, se muestra una tabla que compara las actividades planteadas por Métrica 3 y su equivalente según la metodología UML

Actividad en Métrica 3	Actividad en UML
Identificación de Responsabilidades y Atributos	Nombre de Clase Atributos Lógicos Operaciones
Identificación de Asociaciones y Agregaciones	Asociaciones Estáticas de Instancias Asociaciones Dinámicas de Instancia
Identificación de Generalizaciones	Herencia

A continuación se presenta el detalle de las actividades según Métrica 3.

5.6.1. Identificación de Responsabilidades y Atributos

El objetivo de esta tarea es identificar las responsabilidades y atributos relevantes de una clase.

Las responsabilidades de una clase definen la funcionalidad de esa clase y están basadas en el estudio de los papeles que desempeñan sus objetos dentro de los distintos casos de uso. A partir de estas responsabilidades pueden comenzar a encontrarse las operaciones que van a pertenecer a esa clase. Estas deben ser relevantes, simples y deben participar en la descripción de la responsabilidad.

Los atributos de una clase especifican propiedades de la clase y se identifican por que están implicados en sus responsabilidades. Los tipos de estos atributos deberían ser conceptuales y conocidos en el dominio.

De manera opcional se elabora una especificación para cada clase, que incluye: la lista de sus operaciones y las clases que colaboran para cubrir esas operaciones y una descripción de las responsabilidades, atributos y operaciones de esa clase.

Para aquellas clases cuyo comportamiento dependa del estado en el que se encuentren se realiza, también de manera opcional, un diagrama de transición de estados.

Una cuestión muy importante en los desarrollos orientados a objetos consiste en que la implementación de los mismos debe ser coherente y consistente dentro del ambiente en el cual se implementa. Los lenguajes de implementación orientados a objetos suelen estar basados en una librería de clases. La misma está implementada utilizando el lenguaje y semántica del idioma inglés.

Por lo tanto es necesario implementar nuestras clases y componentes basándonos en el idioma inglés también para soportar posibles usos de polimorfismo, entre otras cualidades de los lenguajes orientados a objetos. Si bien estas cuestiones aparecen en el momento de implementar nuestro sistema, es necesario tenerlas en cuenta desde este punto para evitar futuras

inconsistencias entre los modelos de análisis y diseño con la implementación. Si bien los diagramas de análisis antes construidos utilizan denominaciones en español para facilitar su comprensión por parte de los usuarios.

Por este motivo, a continuación se define una tabla con el nombre de la clase a implementar (en idioma inglés) y un alias que es el nombre dado en los documentos de análisis.

Clase	Responsabilidad	Atributos
Pantalla de Ingreso al Sistema	Validar el ingreso al sistema por parte de un usuario.	nombre de usuario clave
Administrador de Usuarios	Administrar la información a cerca de los usuarios del sistema.	usuarios del sistema
Usuario	Representa a un usuario del sistema.	nombre de usuario clave de acceso
Comprador	Representar al comprador de productos en el mercado virtual.	identificador nombre dirección de correo electrónico
Pantalla de Carga de Pedido de Cotización	Permitir la edición de un pedido de cotización.	pedido de cotización
Pantalla de Carga de un Ítem de un Pedido de Cotización	Permitir la edición de un ítem de un pedido de cotización.	ítem de un pedido de cotización
Pantalla de Búsqueda de Productos	Permitir al usuario interactuar con el componente de búsqueda de un producto dentro del catálogo.	buscador de productos producto seleccionado
Pedido de Cotización	Representar un pedido de cotización dentro del sistema.	proveedor comprador fecha de carga número ítems
Ítem del Pedido de	Representar un ítem de un	producto

Cotización	pedido de cotización dentro del sistema.	cantidad precio fecha de entrega
Producto	Representar a un producto de compra venta del mercado virtual de materiales eléctricos.	código descripción
Gestor de Búsqueda de Productos	Centralizar la gestión de las búsquedas sobre el catálogo de producto.	catálogo de productos
Proveedor	Representar al proveedor de materiales eléctricos.	identificador nombre dirección de correo electrónico
Gestor de Colocación de Pedidos de Cotización	Gestionar la colocación de pedidos de cotización a los potenciales proveedores de un pedido de cotización genérico.	proveedores por producto pedido de cotización a colocar
Pantalla de Consulta de Cotizaciones	Permitir al usuario consultar las cotizaciones realizadas sobre uno de sus pedidos de cotización.	pedidos de cotización realizados por el usuario cotizaciones por pedido de cotización
Cotización	Representar una cotización realizada por un proveedor dentro del mercado virtual.	proveedor comprador fecha de creación ítems pedido de cotización
Item de una Cotización	Representar un ítem de una cotización realizada por un proveedor dentro del mercado virtual.	producto cantidad precio fecha de entrega ítem del pedido de cotización al que se responde

Pantalla de Selección de Un Pedido de Cotización	Permitir a un usuario seleccionar un pedido de cotización realizado por el mismo.	usuario pedidos de cotización pedido de cotización seleccionado
Interfaz de Selección de Cotización Para Un Producto	Permitir a un comprador seleccionar un pedido de cotización realizado para él sobre un producto específico.	comprador producto cotizaciones
Gestor de Confección de Ordenes de Compra	Ayudar a un comprador a completar una orden de compra.	comprador pedido de cotización base cotizaciones presentadas
Orden de Compra	Representar en el mercado virtual al documento de orden de compra.	identificador fecha de creación proveedor comprador ítems de la orden de compra
Ítem de la Orden de Compra	Representar en el mercado virtual a un ítem de una orden de compra.	identificador ítem de una cotización
Interfaz de Consulta de Pedidos de Cotización	Permitir a un proveedor consultar los pedidos de cotización pendientes de contestar.	proveedor pedidos de cotización pendientes
Interfaz de Consulta de Ordenes de Compra	Permitir a un proveedor consultar las órdenes de compra que le fueron colocadas en el mercado virtual.	proveedor ordenes de compra
Interfaz de Carga de Cotización	Permitir a un proveedor cargar una cotización	Proveedor Pedidos de cotización

Para facilitar la trazabilidad de los componentes del sistema, se presenta a continuación una tabla de identifica desde que caso de uso se extrae cada clase:

Clase	Casos de Uso Involucrados
Pantalla de Ingreso al Sistema	Ingresar al Sistema (CU01)
Administrador de Usuarios	Ingresar al Sistema (CU01)
Usuario	Ingresar al Sistema (CU01)
Comprador	Cargar Pedido de Cotización (CU02) Consultar Cotizaciones (CU06) Consultar Ordenes de Compra (CU08) Confeccionar Orden de Compra (CU07)
Interfaz de Carga de Pedido de Cotización	Cargar Pedido de Cotización (CU02)
Interfaz de Carga de un Item de un Pedido de Cotización	Cargar Pedido de Cotización (CU02)
Interfaz de Búsqueda de Productos	Cargar Pedido de Cotización (CU02)
Pedido de Cotización	Cargar Pedido de Cotización (CU02) Colocar Pedido de Cotización por Proveedor (CU03) Confeccionar Orden de Compra (CU07) Consultar Pedidos de Cotización (CU04) Cargar Cotización (CU05)
Item del Pedido de Cotización	Cargar Pedido de Cotización (CU02) Consultar Pedidos de Cotización (CU04) Cargar Cotización (CU05)
Producto	Cargar Pedido de Cotización (CU02) Colocar Pedido de Cotización por Proveedor (CU03)

	Consultar Cotizaciones (CU06) Consultar Pedidos de Cotización (CU04) Consultar Ordenes de Compra (CU08)
Gestor de Búsqueda de Productos	Cargar Pedido de Cotización (CU02)
Proveedor	Colocar Pedido de Cotización por Proveedor (CU03) Consultar Cotizaciones (CU06) Cargar Cotización (CU05) Consultar Ordenes de Compra (CU08)
Gestor de Colocación de Pedidos de Cotización	Colocar Pedido de Cotización por Proveedor (CU03)
Interfaz de Consulta de Cotizaciones	Consultar Cotizaciones (CU06)
Cotización	Consultar Cotizaciones (CU06) Cargar Cotización (CU05)
Item de una Cotización	Consultar Cotizaciones (CU06) Cargar Cotización (CU05)
Interfaz de Selección de Un Pedido de Cotización	Confeccionar Orden de Compra (CU07) Cargar Cotización (CU05)
Interfaz de Selección de Cotización Para Un Producto	Confeccionar Orden de Compra (CU07)
Gestor de Confección de Ordenes de Compra	Confeccionar Orden de Compra (CU07)
Orden de Compra	Confeccionar Orden de Compra (CU07) Consultar Ordenes de Compra (CU08)

Item de la Orden de Compra	Confeccionar Orden de Compra (CU07) Consultar Ordenes de Compra (CU08)
Interfaz de Consulta de Pedidos de Cotización	Consultar Pedidos de Cotización (CU04)
Interfaz de Consulta de Ordenes de Compra	Consultar Ordenes de Compra (CU08)
Interfaz de Carga de Respuesta de Un Pedido de Cotización	Cargar Cotización (CU05)

5.6.2. Identificación de Asociaciones y Agregaciones (ASI 5.2)

En esta tarea se estudian los mensajes establecidos entre los objetos del diagrama de interacción para determinar que asociaciones existen entre las clases correspondientes.

Estas asociaciones suelen corresponderse con expresiones verbales incluidas en las especificaciones.

Las relaciones surgen como respuesta a las demandas en los distintos casos de uso y para ello puede existir la necesidad de definir agregaciones y herencia entre objetos. Una asociación está caracterizada por los papeles que desempeña, su direccionalidad, que representa el sentido en el que se debe interpretar, y su cardinalidad, que representa el número de instancias implicadas en la asociación. Ambas características pueden obtenerse a partir de la especificación de los casos de uso.

A medida que se establecen las relaciones entre las clases se revisa la especificación de subsistemas de análisis en la actividad Identificación de Subsistemas de Análisis (ASI 3), para conseguir optimizar los subsistemas.

Para entrar en detalle sobre este punto, en lo que se refiere al presente proyecto de tesis, podemos recordar que las clases de objetos se dividen en 3 grandes categorías: interfaz, dominio y control. Por lo general, los lenguajes de programación orientados a los objetos vienen acompañados de librerías de clases. Estas contienen implementaciones orientadas a objetos de las características más importantes de las interfaces de usuario.

Es por esta razón, que cualquier desarrollo de interfaces estará fuertemente ligado a las clases provistas por el lenguaje en cuestión. Durante el análisis se ha optado por indagar en cuestiones relacionadas con las clases de control y dominio y postergar las definiciones relacionadas con la interfaz de usuario para un momento posterior en cuanto se trate el diseño detallado. Luego, a continuación se trata el tema de asociaciones y agregaciones para dichos objetos.

Se entiende por asociación de objetos a la identificación de necesidad de cooperación entre los mismos para poder llevar a cabo una responsabilidad. Esto puede ser visto como las flechas que unen las clases de análisis en los diagramas de colaboración antes descriptos. En este caso es necesario estudiar cuidadosamente dichas conexiones dado que posteriormente indicarán referencias y agregaciones entre objetos.

Cada uno de los participantes en una asociación debe llevar a cabo cierta actividad, la cual queda determinada por el rol que ocupa. La agregación, es un caso particular de asociación también conocida como relación "part-of". Es decir, donde el rol de uno de los participantes es "ser parte de" otro (es el caso más general). En este caso se habla de composición.

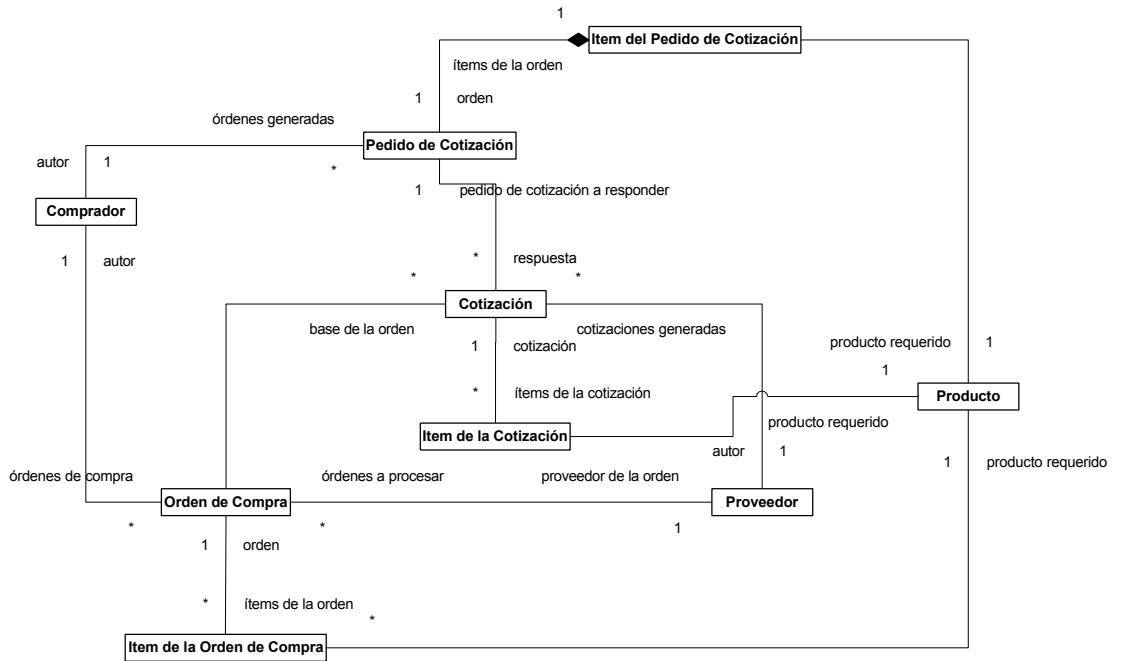
Se pueden citar tres casos de asociaciones:

- Conceptos que se contienen físicamente unos a otros.
- Conceptos que están compuestos mutuamente.
- Conceptos que conforman un conjunto conceptual de objetos (ej: una familia que consta de madre, padre, hijos, etc.).

A medida que se avanza con el análisis, se identifican ambos tipos de relaciones en nuestro modelo de objetos. Algunas surgen de forma bastante obvia; un buen ejemplo puede ser hablar de ítem que conforman un comprobante (orden de compra, pedido de cotización, etc.). En este caso hablamos de una composición de objetos. Esta relación queda definida como composición por varios factores entre los cuales se pueden identificar uno bien determinante como ser: no tiene sentido la existencia de ítems de comprobantes sin el comprobante en sí mismo. Esta relación, como vemos, presenta una cierta similitud con la relación entre entidades y entidades dependientes en el modelo relacional. Otra asociación más tradicional se presenta al hablar de usuarios del sistema y comprobantes.

A continuación, el detalle de las asociaciones entre clases de dominio:

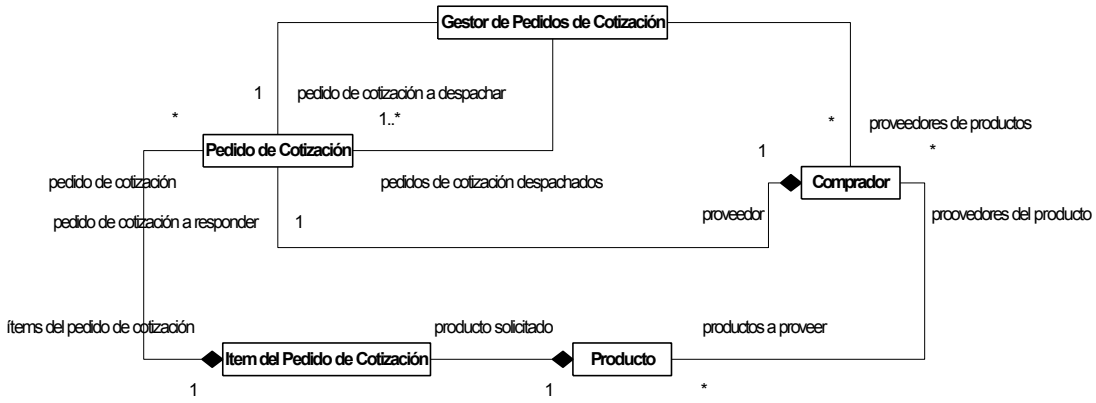
Diagrama de relaciones entre clases del dominio



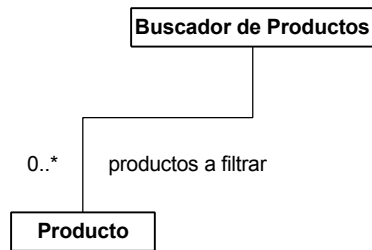
Las clases de control, por su parte, son responsables de administrar los flujos de trabajo necesarios para implementar un caso de uso. Por lo general, utilizan a las clases de dominio como materia prima y resultado de su operación.

Por esta razón las relaciones de estas clases se analizan en forma individual.

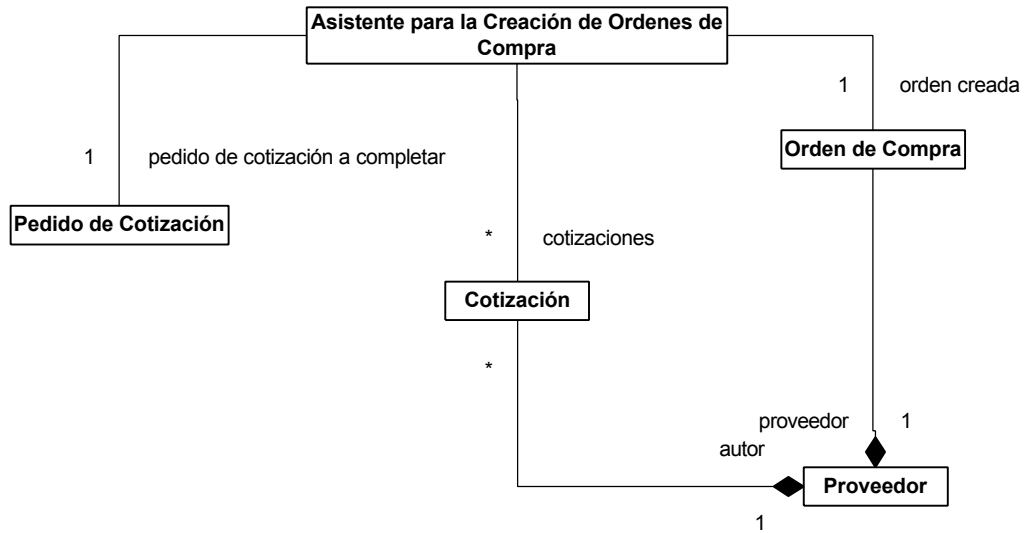
Gestor de Asignación de Pedidos de Cotización



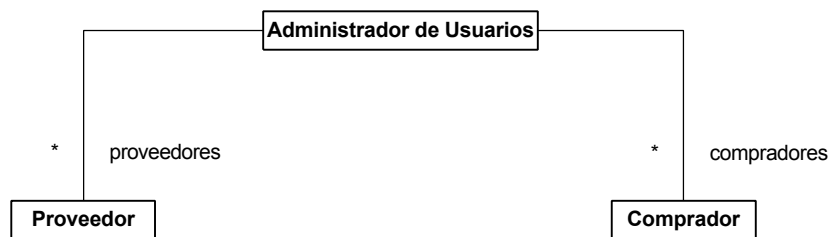
Buscador de Productos



Asistente para la Creación de Ordenes de Compra



Gestor de Usuarios



5.6.3. Identificación de Generalizaciones (ASI 5.3)

El objetivo de esta tarea es representar una organización de las clases que permita una implementación sencilla de la herencia y una agrupación semántica de las diferentes clases, basándose siempre en las normas y estándares definidos por métrica v3, y en nuestro caso teniendo en cuenta a UML en forma adicional.

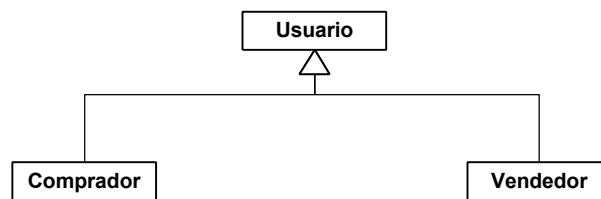
La identificación de generalizaciones se utiliza durante el análisis para extraer comportamiento compartido y común entre diferentes clases de

análisis. Las generalizaciones obtenidas en esta parte del proceso de desarrollo deben ser de nivel alto y conceptual dado que su objetivo debe ser mantener el modelo de análisis simple y de fácil comprensión.

A continuación se analizan las generalizaciones obtenidas:

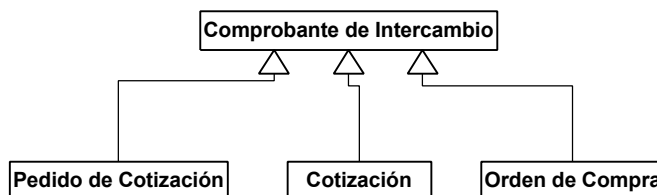
Usuarios

Los usuarios del sistema, ya identificados como compradores (purchaser) y vendedores (supplier) tienen un comportamiento común. Ambos deben, por ejemplo, ingresar al sistema y para ello deben poseer una cuenta (nombre de usuario y clave de acceso). Además, deben compartir documentos entre ambos y saber manejarlos de acuerdo a su responsabilidad frente a cualquiera de ellos (ej: un comprador emite un pedido de cotización y el proveedor debe contestarla). Teniendo en cuenta estas premisas, se presenta una primer generalización:



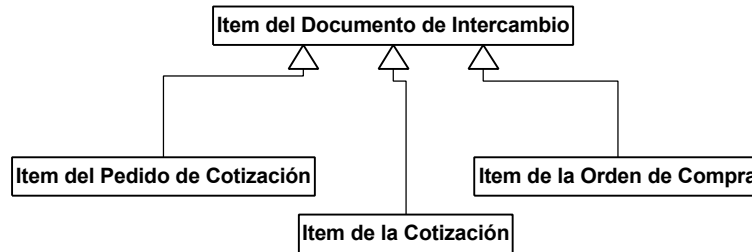
Documentos

Los documentos que maneja el sistema definitivamente presentan también generalizaciones. En principio, todos son documentos. Todos tienen una fecha de emisión, un autor (sea tanto un comprador como un vendedor), un número que los identifique unívocamente dentro del sistema, todos hacen referencias a un conjunto de ítems que los conforman. Así se presenta la siguiente generalización:



Por otra parte, los documentos presentan ítems. Estos ítems, independientemente de su comportamiento a posteriori, presentan

referencias comunes a un documento, un producto, la cantidad, precio y fecha de entrega solicitadas, etc. Por lo tanto, se puede afirmar que se ha identificado una tercer generalización:



5.7. Elaboración del Modelo de Datos

Según se detalla en el documento "Introducción" de Métrica v3, esta actividad no es contemplada para los desarrollos orientados a los objetos. De todas formas y dada su importancia, se plantea a continuación una justificación que hace las veces de introducción a la filosofía adoptada que se detallará cuando se llegue a la etapa de diseño del sistema.

A la hora de generar un modelo de datos para la presente aplicación, es necesario tener en cuenta la característica de orientada a objetos de la misma. Esto se debe a que existe extensa bibliografía relativa a como generar una base de datos que permita el almacenamiento de los objetos de la aplicación. En estos casos no se estaría hablando de generar un modelo lógico desde los requisitos, o bien, generar un diagrama entidad relación que identifique las entidades del sistema.

Desde esta perspectiva, es necesario tomar como base en el modelo de clases, generado con anterioridad, donde, entre otras cosas, se pueden observar las entidades del sistema representadas por clases.

Es por esta razón se utilizan las técnicas y metodologías de generación de esquemas de datos para sistemas orientados a objetos según el enfoque que proponen Blaha y Premerlani [Blaha 1998].

En síntesis, la metodología plantea que no existen diferencias entre el diseño de objetos y el de bases de datos durante el análisis. Esto se debe a que en ambos casos se trabaja con entidades, atributos y relaciones, con lo

cual debería ser indistinto trabajar con diagramas de clases o diagramas entidad relación. En cambio, sí se plantean cuestiones efectivamente distintas cuando se habla de diseño ya que el universo de implementación del código es extremadamente distinto al de las bases de datos. Se tendrá en cuenta dicha cuestión en el momento del diseño.

5.8. Definición de Interfaces de Usuario

En esta actividad se especifican las interfaces entre el sistema y el usuario: formatos de pantallas, diálogos e informes, principalmente. El objetivo es realizar un análisis de los procesos del sistema de información en los que se requiere una interacción del usuario, con el fin de crear una interfaz que satisfaga todos los requisitos establecidos, teniendo en cuenta los diferentes perfiles a quienes va dirigido.

Al inicio de este análisis es necesario seleccionar el entorno en el que será operativa la interfaz considerando estándares internacionales y de la instalación, y establecer las directrices que serán aplicables en los procesos de diseño y construcción. El propósito es construir una interfaz de usuario compatible con sus necesidades, flexible, coherente, eficiente y sencilla de utilizar, teniendo en cuenta facilitar la portabilidad. Se identifican los distintos grupos de usuarios de acuerdo a las funciones que realizan, conocimientos y habilidades que poseen y características del entorno en el que trabajan.

La identificación de los diferentes perfiles permite conocer mejor las necesidades y particularidades de cada uno de ellos.

Asimismo, se determina la naturaleza de los procesos que llevan a cabo (en lotes o interactivos) y para cada proceso interactivo se especifica qué tipo de información requerirá el usuario para completar su ejecución realizando, para ello, una descomposición en diálogos que refleje la secuencia de la interfaz de pantalla.

Finalmente se define el formato y contenido de cada una de las interfaces de pantalla especificando su comportamiento dinámico.

Se propone un flujo de trabajo muy similar para desarrollos estructurados u orientados a objetos, coincidiendo en la mayoría de las tareas, si bien es cierto que en orientación a objetos, al identificar y describir cada escenario en la especificación de los casos de uso, se hace un avance muy significativo en la toma de datos para la posterior definición de la interfaz de usuario.

Como resultado de esta actividad se generará la especificación de interfaz de usuario, como producto que engloba los siguientes elementos:

- Principios generales de la interfaz
- Catálogo de perfiles de usuario
- Descomposición funcional en diálogos
- Catálogo de controles y elementos de diseño de interfaz de pantalla
- Formatos individuales de interfaz de pantalla
- Modelo de navegación de interfaz de pantalla
- Formatos de impresión
- Prototipo de interfaz interactiva (opcional)
- Prototipo de interfaz de impresión (opcional)

Para el presente proyecto se ha decidido utilizar un modelo de prototipos de maqueta para definir las interfaces de usuario con el usuario líder. Las pantallas resultantes aparecen en la especificación de casos de uso.

5.8.1. Especificación de Principios Generales de la Interfaz

El objetivo de esta tarea es especificar los estándares, directrices y elementos generales a tener en cuenta en la definición de la interfaz de usuario, tanto para la interfaz interactiva (gráfica o carácter), como para los informes y formularios impresos.

En primer lugar se selecciona el entorno de la interfaz interactiva (gráfico, carácter, etc.), siguiendo estándares internacionales y de la instalación, y se determinan los principios de diseño de la interfaz de usuario.

Existe una serie de estándares en el mundo que especifican este punto al a perfección. Para el caso se adhiere al uso del estándar "JAVA Look And Feel Design Guidelines" [Sun Microsystems 1999a] que define como deben ser y comportarse las interfaces gráficas de usuario programadas en lenguaje Java (el lenguaje con el cual se implementará la aplicación).

5.8.1.1. Directrices generales en cuanto a la interfaz y aspectos generales de interacción

El presente proyecto será construido con una interfaz gráfica. Esto es debido a el tipo de software a desarrollar. La idea es lograr un producto que quede instalado en la PC de trabajo y que avise constantemente al usuario sobre novedades en el sistema y permitir que dicho usuario pueda rápida y fácilmente completar su parte del trabajo.

Por eso es el objetivo desarrollar una interfaz amigable, organizada y donde el usuario pueda, prácticamente, utilizar toda la funcionalidad del sistema con el *mouse* y con muy poca interacción mediante el teclado (para evitar errores).

Nuevamente, se toma como base la especificación de Sun Microsystems para el desarrollo de aplicaciones de este tipo [Sun Microsystems 1999a].

5.8.1.2. Principios de composición de pantallas y criterios de ubicación de los distintos elementos dentro de cada formato

Las pantallas serán concebidas así y como se presentan con anterioridad en el análisis de casos de uso. Para ello se generarán pantallas que respeten un cierto estándar, salvo el menú principal, cuyo objetivo es muy distinto al resto.

Para todas las pantallas:

- Los botones de acción (aceptar, cancelar, agregar, etc.) deben estar ubicados en la parte inferior.
- El área central de la pantalla debe contener los componentes de trabajo específicos a la funcionalidad de la misma. Puede darse el caso de un formulario donde el usuario ingresa datos, en cuyo caso dicha área central contendrá los componentes de ingreso de información, o bien, una consulta donde se muestra una tabla con los elementos consultados.
- El área superior de la pantalla contendrá los elementos de acciones especiales o parametrización. Un ejemplo puede ser hablar de una barra de acción (copiar, pegar, etc.) o bien un rango de fechas de filtro para la consulta que se encuentra en el área central.

5.8.1.3. Normas para los mensajes de error y aviso, codificación, presentación y comportamientos

Los mensajes de error, aviso y codificación se presentarán siguiendo el estándar de las interfaces gráficas modernas: diálogos del tipo pop-up que aparecerán con un icono que simbolice el tipo de mensaje (error, advertencia, confirmación, etc.) y un texto que indique el mensaje en sí mismo.

5.8.1.4. Normas para la presentación de ayudas

La ayuda del sistema se presentará en formato HTML. No se ha previsto la implementación de la misma para el presente proyecto de desarrollo, por lo cual no se harán más explicaciones al respecto, aunque hay que destacar que la misma debería constar de páginas HTML dispuestas en algún servidor de Internet de forma tal de poder mantener la información actualizada en un único punto y de esa forma lograr que los usuarios siempre tengan la última versión del sistema de ayudas.

El mismo debería constar de un índice que agrupe las áreas temáticas cubiertas por la ayuda y un buscador que permita llegar a cierta información rápidamente.

5.8.1.5. Informes Impresos

En el presente desarrollo se pueden identificar los siguientes informes impresos:

- Pedido de cotización.
- Cotización.
- Orden de Compra.

Si bien es la idea principal del sistema es que toda la información que este administra este comprendida dentro de las pantallas del mismo, es verdad que ciertos procesos manuales fuera del alcance del mismo puedan requerir de esta información en forma impresa. Para ello se definen a continuación los formatos de impresión para estas salidas.

Al tratarse de un sistema distribuido sobre Internet, es necesario tener en cuenta las complicaciones que pueden aparecer al intentar distribuir reportes que serán impresos en diversas tecnologías de impresión y sistemas operativos. Esta cuestión es tratada en profundidad en el Diseño del Sistema de Información, dejando para esta actividad la especificación de información y distribución de la misma en una hoja de reporte.

Pedido de Cotización

Fecha de Ingreso y Número: **XX/XX/XXX - XXXXXX**

Cliente: **XXXXXXXXXXXXXXXXXXXXXX**

Fecha de Impresión: **XX/XX/XXXX**

Productos a Cotizar			
Código de Producto	Descripción del Producto	Fecha de Entrega	Cantidad Requerida
XXXX	XXXXXXXXXXXXXXXXXX	XX/XX/XXXX	XX.XX
XXXX	XXXXXXXXXXXXXXXXXX	XX/XX/XXXX	XX.XX
XXXX	XXXXXXXXXXXXXXXXXX	XX/XX/XXXX	XX.XX

Cotización

Fecha de Ingreso del Pedido de Cotización y Número: **XX/XX/XXXX - XXXXXX**

Fecha de Ingreso de la Cotización y Número: **XX/XX/XXXX - XXXXXX**

Solicitante: **XXXXXXXXXXXXXXXXXXXXXX**

Proveedor: **XXXXXXXXXXXXXXXXXXXXXX**

Fecha de Impresión: **XX/XX/XXXX**

Cotización Válida Hasta: **XX/XX/XXXX**

Productos Cotizados				
Código de Producto	Descripción del Producto	Fecha de Entrega (Solicitada/Cotizada)	Cantidad Requerida (Solicitada/Cotizada)	Precio Unitario
XXXX	XXXXXXXXXXXXXXXXXXXXXX	XX/XX/XXXX	XX.XX	XXXX.XX

		XX/XX/XXXX	XX.XX	
--	--	------------	-------	--

Orden de Compra

Fecha de Ingreso: **XX/XX/XXXX - XXXXXX**

Solicitante: **XXXXXXXXXXXXXXXXXXXX**

Proveedor: **XXXXXXXXXXXXXXXXXXXX**

Fecha de Impresión: **XX/XX/XXXX**

Productos Cotizados				
Código de Producto	Descripción del Producto	Fecha de Entrega	Cantidad Requerida	Pre Uni
XXXX	XXXXXXXXXXXXXXXXXXXX	XX/XX/XXXX	XX.XX	XXX

5.8.2. Especificación de Formatos Individuales de la Interfaz de Pantalla (ASI 8.3)

El objetivo de esta tarea es especificar cada formato individual de la interfaz de pantalla, desde el punto de vista estático. Para cada proceso interactivo identificado en la especificación de los casos de uso, y teniendo en cuenta los formatos estándar definidos en la tarea Especificación de Principios Generales de la Interfaz (ASI 8.1), se definen los formatos individuales de cada interfaz de pantalla requerida para completar la especificación de cada diálogo.

Para el caso, dado que hablamos de un desarrollo orientado a objetos, estos formatos individuales irán completando las especificaciones de los casos de uso.

También se considerará el catálogo de requisitos, para especificar las interfaces relacionadas con las consultas.

En la definición de cada interfaz de pantalla se deben definir aquellos aspectos que se consideren de interés para su posterior diseño y construcción:

- Posibilidad de cambio de tamaño, ubicación, modalidad (modal del sistema, modal de aplicación), etc.
- Dispositivos de entrada necesarios para su ejecución.
- Conjunto y formato de datos asociados, identificando qué datos se usan y cuáles se generan como consecuencia de su ejecución.
- Controles y elementos de diseño asociados, indicando cuáles aparecerán inicialmente activos e inactivos al visualizar la interfaz de pantalla.

Dado que la interfaz será de tipo gráfico, será necesario tener en cuenta los dispositivos de entrada/salida típicos, como ser: mouse y teclado. Esta consideración será tenida en cuenta para todas las pantallas del sistema.

A continuación se detalla el análisis de las interfaces de usuario desde el punto de vista estático:

Ingreso al Sistema



Característica	Descripción
Tipo	Modal de aplicación
Ubicación	Centrada
Cambio de tamaño	NO
Datos usados	Nombre de usuario Clave de usuario
Datos generados	Se genera un usuario del sistema

Menú Principal para Comprador



Característica	Descripción
Tipo	Marco de Aplicación
Ubicación	Definida por el usuario
Cambio de tamaño	SI
Datos usados	Lista de pedidos de cotización emitidos. Lista de pedidos de cotización recibidos. Lista de órdenes de compra emitidas.
Datos generados	

Menú Principal para Proveedor



Característica	Descripción
Tipo	Marco de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	SI
Datos usados	Pedidos de cotización recibidos. Pedidos de cotización respondidos.
Datos generados	

Nuevo Pedido de Cotización

The screenshot shows a software window titled "Crear Pedido de Cotización". Inside the window, there is a table with the following headers: "Código", "Descripción", "Cantidad", "Fecha Entrega", and "Precio". The table body is currently empty. Below the table, there are three buttons: "Agregar Producto", "Borrar Producto", and "Enviar Pedido".

Característica	Descripción
Tipo	Marco de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	SI
Datos usados	Lista de items del pedido de cotización: <ul style="list-style-type: none"> • Código del producto • Descripción del producto • Cantidad requerida • Fecha de entrega • Precio tentativo
Datos generados	Un nuevo pedido de cotización

Nuevo Item de Pedido de Cotización

The image shows a dialog box titled "Agregar Item". It has four input fields: "Producto" with the value "<SIN SELECCION>", "Cantidad" (empty), "Fecha de Entrega" with the value "13/01/2003", and "Precio Estimado" (empty). There are "Aceptar" and "Cancelar" buttons at the bottom.

Característica	Descripción
Tipo	Modal de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	NO
Datos usados	<ul style="list-style-type: none">• Producto a pedir• Cantidad• Fecha de entrega• Precio estimado
Datos generados	Un nuevo item de pedido de cotización

Buscador de Productos

Buscar Producto

Por Código Por Descripción

Código: Buscar

Código	Descripción
--------	-------------

Ver Todos Seleccionar Cancelar

Característica	Descripción
Tipo	Modal de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	SI
Datos usados	Código o descripción parcial del producto a buscar. Lista de productos encontrados.
Datos generados	Producto seleccionado.

Carga de Ordenes de Compra

Característica	Descripción
Tipo	Modal de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	SI
Datos usados	Producto a pedir. Lista de cotizaciones recibidas para ese producto. Cantidad, precio y fecha de entrega a contratar.
Datos generados	Una nueva orden de compra.

Responder Pedido de Cotización

Responder Pedido de Cotización

Número de Pedido de Cotización: 3

Fecha del Pedido de Cotización: 2003-01-14

Solicitante: PETRONIO, MARCELO

Fecha de Vencimiento del Presupuesto: 14/01/2003

Producto	Fecha	Fecha Rta	Cantidad	Cant. Pedida	Precio	Precio Pedido
LAMPARA 10...	2003-01-14	2003-01-14	10.0000	10.0000	10.00	10.00

No Cotizar Item Editar Item Enviar Respuesta

Característica	Descripción
Tipo	Modal de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	SI
Datos usados	Lista de productos a cotizar.
Datos generados	

Responder Item de Pedido de Cotización

Característica	Descripción
Tipo	Modal de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	SI
Datos usados	Se muestran los datos del pedido de cotización y se cargan los datos de respuesta (fecha de entrega, cantidad y precio).
Datos generados	Un nuevo item de cotización.

Consulta de Ordenes de Compra

Cliente	Fecha
PETRONIO, MARCELO	2003-01-13

Producto	Cantidad	Precio	Fecha de E...
LAMPARA 100W	10.0000	10.00	2003-01-14

Imprimir Cerrar

Característica	Descripción
Tipo	Modal de Aplicación
Ubicación	Definida por el usuario.
Cambio de tamaño	SI
Datos usados	Detalle de la orden de compra: Cliente. Productos pedidos.
Datos generados	Una nueva orden de compra.

5.8.3. Especificación del Comportamiento Dinámico de la Interfaz

El objetivo de esta tarea es especificar los flujos entre los distintos formatos de interfaz de pantalla, y también dentro del propio formato. Este comportamiento se especifica mediante un modelo de navegación de interfaz de pantalla.

Especificación de Formatos Individuales de la Interfaz de Pantalla (ASI 8.3), se establece la entrada lógica de los datos y las reglas de validación incluyendo dependencia de valores (reflejo de los requisitos de validación de sistema).

Se analiza y determina la secuencia de acciones específicas para completar cada diálogo, tal y como se ejecutará en el ámbito de la interfaz, así como las condiciones que deben cumplirse para su inicio y las posibles restricciones durante su ejecución. El comportamiento vendrá dirigido y representado por los controles y los eventos que provocan su activación.

Se identifican aquellos diálogos o formatos que se consideren críticos para el correcto funcionamiento del sistema, basándose en el número de usuarios, frecuencia de uso, datos implicados, alcance de las funciones asociadas al diálogo, diálogos comunes a diferentes funciones, marco de seguridad establecido en los requisitos del sistema, etc.

Para los diálogos o comportamientos complejos de interfaz se propone la técnica de diagrama de transición de estados, siendo suficiente en la mayoría de los casos una especificación del comportamiento con matrices control/evento/acción, detallándose la acción con una descripción textual.

Para el presente proyecto se identifica como crítico al diálogo desde el cual se colocan las ordenes de compra. Para ello se decide utilizar una matriz control/evento/acción para especificar las características propias de este diálogo.

Confección de Orden de Compra		
Control	Evento	Acción
Botón Agregar	Botón Presionado	Abrir un diálogo en blanco para la edición de un ítem de orden de compra. Al ser confirmado el diálogo este ítem deberá ser agregado a la orden de compra.
Botón Borrar	Botón Presionado	Verificar que exista un ítem seleccionado. Si no lo existe, notificar del error al usuario mediante un diálogo de mensajes. De existir, solicitar al usuario la confirmación de la baja mediante un mensaje que especifique: "¿ Esta Ud. seguro de borrar el ítem de la orden de compra ?". El usuario deberá responder con Si o No. De contestar afirmativamente, el ítem deberá ser borrado. De lo contrario, nada debe ocurrir.
Botón Cancelar	Botón Presionado	Verificar que nada haya sido cargado en la orden de compra. De existir elementos cargados, preguntar al usuario: "¿Esta Ud. seguro de cancelar la orden de compra?". De responder afirmativamente, se debe cerrar el diálogo y volver a la pantalla anterior. De lo contrario, nada debe ocurrir.
Botón Enviar	Botón Presionado	Verificar que existe por lo menos un ítem cargado en la orden de compra. De ser así, preguntar al usuario: "¿Esta Ud. seguro de que desea enviar la orden de compra?". De responder afirmativamente, la misma debe ser enviada. De lo contrario, nada debe ocurrir.

Editor de Item de Orden de Compra		
Control	Evento	Acción
Selector de Producto	Botón Presionado	Abrir el buscador de productos. El usuario selecciona un producto. Buscar las cotizaciones no vencidas que existen para este producto y completar la lista de cotizaciones. En caso de no existir cotizaciones, informar al usuario por medio de un mensaje de diálogo: "El producto no posee cotizaciones asociadas."
Lista de Cotizaciones.	Doble Click sobre un Elemento.	Recuperar el elemento de la lista y con este completar los campos cantidad, precio y fecha de entrega.
Botón Aceptar	Botón Presionado	Verificar que se haya seleccionado un producto, que el mismo tenga cotizaciones asociadas y que una de estas haya sido seleccionada. De no cumplirse con estas condiciones, avisar al usuario al respecto por medio de un diálogo de mensaje. De cumplirse las precondiciones, el diálogo deberá cerrarse.
Botón Cancelar	Botón Presionado	Cerrar el diálogo.

5.8.4. Especificación de Formatos de Impresión (ASI 8.5)

El objetivo de esta tarea es especificar los formatos y características de las salidas o entradas impresas del sistema.

El formato de los reportes impresos ya ha sido especificado en la actividad 5.1.8.5 del presente capítulo.

A este formato se agrega como especificación que los reportes serán presentados en formato HTML. Cuando un usuario seleccione la opción imprimir de un comprobante, se abrirá un navegador de Internet con el informe a imprimir.

De esta forma, el sistema no deberá manejarse con complejidades de impresión (drivers, modelos de impresoras, tamaño del fuente, etc.) propios de el equipo "cliente" y podrá presentar los informes en un formato portable y genérico.

5.9. Análisis de Consistencia y Especificación de Requisitos

El objetivo de esta actividad es garantizar la calidad de los distintos modelos generados en el proceso de Análisis del Sistema de Información y asegurar que el usuario y el analista tienen el mismo concepto del sistema.

Para ello se deben realizar ciertas tareas, como ser : verificar la calidad técnica de cada modelo, asegurar la coherencia entre los distintos modelos y validar el cumplimiento de los requisitos.

Esta actividad tiene mucho más sentido en equipos donde varios ingenieros del software trabajan en paralelo sobre distintas actividades de la metodología. Si bien tienen también importancia en proyectos de un solo ingeniero (como es este caso), dichas actividades se han realizado continuamente en todo el desarrollo a modo de refinamiento de la información obtenida y especificada. Por otro lado, existen controles y correcciones sobre el material de tesis que también apuntan a dicho fin. Por lo tanto no se profundizará en las tareas asociadas a dicho fin.

También se elabora en esta actividad la Especificación de Requisitos Software (ERS), como producto para la aprobación formal, por parte del usuario, de las especificaciones del sistema, el cual sí se desarrollará en detalle a continuación.

5.9.1. Elaboración de la Especificación de Requisitos Software

A continuación, la especificación de requisitos de software para el presente proyecto.

5.9.1.1. Objetivos

Indicar los objetivos del sistema, tanto de negocios como tecnológicos y políticos.

Utilizando la metodología Métrica V3, desarrollar un sistema de mercados virtuales que permita la interacción entre compradores y proveedores de materiales eléctricos. El sistema abarcará las actividades del circuito de compras que se presentan a continuación:

- Carga y envío de pedidos de cotización.
- Carga y envío de cotizaciones.

- Carga y envío de órdenes de compra.
- Consultas varias.

*Ver documentos de casos de uso.

5.9.1.2. Características Generales

Indicar características generales del sistema como ser tecnología a utilizar, ambiente de implementación, usuarios a los que esta destinado, etc.

El sistema se desarrollará con una interfaz del tipo cliente/servidor aunque utilizando una arquitectura y tecnología de comunicación estándar de Internet. Esto se realiza para maximizar la facilidad de uso por parte de los usuarios finales pero al mismo tiempo permitir que los mismos trabajen en forma distribuida sobre la red.

Los usuarios a los que está destinado el sistema se encuentran detallados en los documentos de análisis correspondientes (compradores y proveedores de materiales eléctricos).

5.9.1.3. Referencias

Indicar bibliografía, documentos, especificaciones, o cualquier tipo de referencias a otro material que no se encuentre detallado en el presente documento

Este punto se presenta dado que es parte de cualquier ERS. En este caso las referencias bibliográficas se encuentran en el capítulo correspondiente del presente documento.

5.9.1.4. Glosario de Términos y Abreviaturas

Indicar términos, conceptos del negocio o abreviaturas utilizadas en el resto del documento.

Idem al punto anterior.

5.9.1.5. Requerimientos Funcionales

Especifique los requerimientos relacionados con la funcionalidad de la aplicación.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RF01	Cargar un pedido de cotización	Baja	Alta	5
RF02	Colocar pedidos de cotización por proveedor	Baja	Alta	5
RF03	Consultar pedidos de cotización	Baja	Alta	5
RF04	Cargar una cotización	Baja	Alta	5
RF05	Confeccionar una orden de compra	Baja	Alta	5
RF06	Consultar órdenes de compra	Baja	Alta	5
RF07	Consultar cotizaciones	Baja	Alta	5

5.9.1.6. Requerimientos No Funcionales

Especifique los requerimientos relacionados con la funcionalidad de la aplicación.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNF01	Los componentes cliente serán desarrollados teniendo en cuenta soportar la carga de un único usuario por sesión de uso.	Baja	Alta	5
RNF02	Los componentes del servidor deberán poder soportar múltiples usuarios concurrentes al sistema.	Baja	Alta	5

5.9.1.7. Requerimientos de Interfaz del Usuario

Especifique requerimientos específicos de la interfaz del usuario. Por ejemplo, si el sistema necesita de una interfaz para soporte de touch-screen, una interfaz de acceso rápido a la funcionalidad, de carga masiva, etc.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RFNIU01	Interfaz dinámica de acceso rápido a la funcionalidad.	Baja	Alta	1
RNFIU02	Interfaz estándar para sistemas de mensajería instantánea en Internet (pequeñas ventanas que muestran resúmenes de información).	Media	Alta	1
RNFIU03	Interfaz que utilice componentes reusables que minimicen los errores de ingreso de datos.	Baja	Alta	1

5.9.1.8. Requerimientos de Interfaces Externas

Especifique requerimientos relacionados con la interacción con otros sistemas. Por ejemplo: “cada orden ingresada al sistema debe ser comunicada al sistema en casa central”.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFC01	Los componentes cliente y servidor del sistema deberán comunicarse utilizando un protocolo basado en tecnologías estándar de Internet: HTTP y XML. (comunicaciones no orientadas a la conexión)	Baja	Alta	5
RNFC02	El componente servidor deberá comunicarse con el sistema gestor de base de datos utilizando mecanismos estándar para dicho fin: SQL y TCP/IP (comunicaciones orientadas a la conexión).	Baja	Alta	5

5.9.1.9. Requerimientos de Hardware y Software

Especifique requerimientos de hardware y software. Por ejemplo: “el sistema debe estar desarrollado en lenguaje de programación Java y debe funcionar en un equipo SUN E3000 ...”.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFHS01	El sistema debe ser desarrollado utilizando software de tipo GNU (gratis para uso académico) de manera de poder ser utilizado posteriormente por las universidades como implementación de referencia.	Baja	Alta	1
RNFHS02	El sistema debe poder ser instalado en un equipo del tipo PC (hardware).	Baja	Alta	1
RNFHS03	El sistema debe ser desarrollado para el sistema operativo LINUX.	Baja	Alta	1
RNFHS04	Para las comunicaciones entre componentes del sistema se deberá utilizar el protocolo TCP/IP	Baja	Alta	1
RNFHS05	El sistema debe ser desarrollado para el sistema operativo Microsoft Windows.	Baja	Alta	1
RNFHS06	Los componentes del sistema deberán cumplir con las restricciones de rendimiento aún al ser utilizados sobre una red de 10 Mbps.	Baja	Alta	1
RNFHS07	El sistema deberá ser desarrollado para su uso en Internet (simulado dentro del ambiente de la universidad)	Baja	Alta	1
RNFHS08	El sistema deberá correr en una modalidad ICQ para facilitar su uso.	Baja	Alta	1

5.9.1.10. Requerimientos de Cumplimiento de Normas y Estándares

Especifique requerimientos de normas y estándares. Por ejemplo “los documentos presentados durante el desarrollo del proyecto deben cumplir las especificaciones de la metodología Métrica Versión 2.0 ...”.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFN01	El sistema debe ser desarrollado utilizando la metodología Métrica v3 teniendo en cuenta las características propias de la orientación a objetos.	Baja	Alta	5
RNFN02	Los componentes tanto cliente como servidor deben ser desarrollados en lenguaje Java para asegurar su portabilidad y robustez.	Baja	Alta	5
RNFN03	Los componentes de servidor deberán ser desarrollados utilizando el estándar de desarrollo propuesto por Sun para aplicaciones de Internet.	Baja	Alta	5

5.9.1.11. Requerimientos de Seguridad

Especifique requerimientos de seguridad a cumplir por el sistema. Por ejemplo “el usuario debe autenticarse vía usuario y clave, el tráfico entre clientes y servidores debe estar encriptado, etc”.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFS01	Los usuarios del sistema deben ingresar su nombre de usuario y clave para ingresar al sistema.	Alta	Alta	2
RNFS02	Las claves de los usuarios debe ser guardada en forma encriptada.	Baja	Alta	2

5.9.2. Requerimientos de Auditoría

Especifique requerimientos de auditoría. Por ejemplo: “El sistema debe guardar rastro de todas las operaciones efectuadas por un usuario”.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFA01	El sistema debe guardar información de cada usuario que ingresa al sistema (fecha, hora, usuario).	Baja	Alta	1
RNFA02	El sistema debe guardar información sobre cada acceso al servidor (independientemente del usuario) para poder verificar el volumen de transacciones (fecha, hora, transacción).	Baja	Alta	1

5.9.3. Requerimientos de Organización

Especifique requerimientos relacionados con la organización.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFO01	El software y las herramientas utilizadas para ejecutar el sistema deben ser gratuitas de forma tal de poder utilizar este proyecto como material de estudio.	Baja	Alta	1

5.9.4. Requerimientos de Backup, Contingencias y Recuperación de Errores

Especifique requerimientos de backup, contingencias y recuperación de errores. Por ejemplo, “el backup debe realizarse todos los días a las 5 am en forma automática. En caso de no poder realizarse el mismo, el administrador del sistema debe ser notificado vía email”.

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFB01	Los datos de gestión del sistema deberán encontrarse centralizados de forma tal de facilitar su backup y recuperación.	Baja	Alta	5

5.9.4.1. Requerimientos de Rendimiento

Especifique requerimientos de rendimiento. Por ejemplo: "Cada transacción debe ser almacenada en el sistema en un lapso no mayor a 5 segundos".

No	Descripción del Requerimiento	Volatilidad	Necesidad	Prioridad
RNFR01	Ninguna operación del sistema puede superar los 5 segundos de espera para el usuario.	Baja	Alta	1
RNFR02	El sistema deberá ser diseñado de forma tal de permitir el acceso concurrente de gran cantidad de usuarios.	Baja	Alta	5

5.10. Cuestiones Específicas al Desarrollo de Internet

En el análisis del sistema de información, se hace referencia a qué debe hacerse y no al cómo debe hacerse. Esto lo hace suficientemente abstracto como para poder independizarlo del modelo de implementación a especificar durante el diseño.

De todas formas, quedan especificados en el ERS una cantidad de requisitos que deben ser validados con un arquitecto de la solución. Se define como arquitecto a aquel rol que se encarga de definir una arquitectura de software capaz de cumplir con los requerimientos del sistema de información.

Por ejemplo, supongamos un sistema que presenta un requisito funcional de la siguiente manera: "Al cargar el usuario su domicilio, el sistema deberá automáticamente completar el campo del código postal correspondiente". Por otro lado, tenemos un requerimiento no funcional que dice: "Los componentes clientes del sistema deben estar implementado 100% utilizando HTML versión 3.2".

La idea de conseguir el código postal a partir de un domicilio no parece ser extremadamente compleja. Por otro lado, son muchos los sistemas que necesitan estar implementados en versiones específicas de HTML para poder ser soportados por la mayor cantidad de navegadores disponibles en el mercado. Sin embargo, al pensar en el comportamiento del HTML (no posee funcionalidad del lado cliente sin algún agregado) y del HTTP, es imposible completar el campo, ya que, para ello sería necesario realizar un pedido sobre el servidor. Dado que el HTML no contiene código ejecutable, es imposible hacer dicho pedido, al menos, sin enviar todo el formulario al

servidor para que este complete el código postal y vuelva. De imaginarse una página para un seguro o crédito bancario con varias direcciones, esto resulta inaceptable.

Este problema debería resolverse fácilmente valiéndose de un componente externo al HTML, pero dado que es un requisito que el sistema use solamente HTML, se verifica que estos requerimientos son excluyentes.

Es por este tipo de razones que es muy importante que el arquitecto se encuentre vinculado al equipo de proyecto que aprobará los requisitos del sistema para evitar que en futuras etapas del proyecto se detecten estas inconsistencias. Si bien este tipo de problemas puede ser encontrado en cualquier sistema, en los sistemas de Internet es más probable por estar desarrollados en muchas tecnologías distintas, varias capas de software y al desconocer en muchos casos las limitaciones del equipo cliente que probablemente no pertenezca a la empresa que solicita el sistema.

Otro ejemplo sucede cuando es necesario que un sistema permita a sus clientes abrir varias pantallas simultáneas sobre cierta información. Dependiendo del servidor de Internet y servidor de aplicaciones disponible, esto puede traer serios problemas de funcionamiento dado que los mismos utilizan un mecanismo para identificar a los clientes basados entre otras cosas en su dirección de Internet, la cual, obviamente, es la misma en todos sus navegadores.

Es fácil ver entonces como ciertos escenarios previstos en un caso de uso pueden volverse riesgosos en un desarrollo de Internet donde las limitaciones técnicas suelen ser mucho más severas que en desarrollos convencionales.

Es así que se presenta una primer diferencia con Métrica v3 que establece que la aprobación del ERS corra por cuenta del Jefe de Proyecto, Comité de Dirección y el Usuario Final, al que se agrega ahora el Arquitecto de la Solución.

Capítulo VI

Diseño del Sistema de Información

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: LBDISEÑO
Fase de Origen:	Diseño del Sistema de Información
Nombre o Código del Elemento de Configuración de Software:	Diseño del Sistema de Información (DSI)
Versión N°: 1.0.4	Entrega N°: 07
Fecha de Última Modificación:	14/01/2003
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

6. Diseño del Sistema de Información

En el presente capítulo se desarrollan las cuestiones relacionadas con el diseño del sistema de información. Se define la arquitectura del sistema, se diseñan los subsistemas de soporte necesarios para resolver cuestiones de comunicaciones y seguridad, los subsistemas de negocios, se realizan los diseños de clases y bases de datos y se especifica el plan de pruebas del sistema.

6.1. Definición de la Arquitectura del Sistema

En esta actividad se define la arquitectura general del sistema de información, especificando las distintas particiones físicas del mismo, la descomposición lógica en subsistemas de diseño y la ubicación de cada subsistema en cada partición, así como la especificación detallada de la infraestructura tecnológica necesaria para dar soporte al sistema de información.

El particionamiento físico del sistema de información se especifica identificando los nodos y las comunicaciones entre los mismos, con cierta independencia de la infraestructura tecnológica que da soporte a cada nodo.

Con el fin de organizar y facilitar el diseño, se realiza una división del sistema de información en subsistemas de diseño, como partes lógicas coherentes y con interfaces claramente definidas.

Se hace una distinción entre subsistemas específicos del sistema de información (en adelante, subsistemas específicos) y subsistemas de soporte, con la finalidad de independizar, en la medida de lo posible, las funcionalidades a cubrir por el sistema de información de la infraestructura que le da soporte. En la mayoría de los casos, los subsistemas específicos

proviene directamente de las especificaciones de análisis y de los subsistemas de análisis, mientras que los subsistemas de soporte provienen de la necesidad de interacción del sistema de información con la infraestructura y con el resto de los sistemas, así como de la reutilización de módulos o subsistemas ya existentes en la instalación.

Debido a que la definición de los subsistemas de soporte puede exigir la participación de distintos perfiles técnicos, se propone el diseño de ambos tipos de subsistemas en actividades distintas, aunque en paralelo.

Una vez identificados y definidos los distintos subsistemas de diseño, se determina su ubicación óptima de acuerdo a la arquitectura. La asignación de dichos subsistemas a cada nodo permite disponer, en función de la carga de proceso y comunicación existente entre los nodos, de la información necesaria para realizar una estimación de las necesidades de infraestructura tecnológica que da soporte al sistema de información. Este factor es especialmente crítico en arquitecturas multinivel o cliente/servidor donde las comunicaciones son determinantes en el rendimiento final del sistema.

Se propone crear un catálogo de excepciones en el que se especifiquen las situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema de información, y que se irá completando a medida que se avance en el diseño detallado de los subsistemas

En esta actividad también se establecen los requisitos, normas y estándares originados como consecuencia de la adopción de una determinada solución de arquitectura o infraestructura, que serán aplicables tanto en este proceso como en la Construcción del Sistema de Información (CSI).

Se detallan, a su vez, de acuerdo a las particularidades de la arquitectura del sistema propuesto, los requisitos de operación, seguridad y control, especificando los procedimientos necesarios para cumplir dichos requisitos. Como resultado de esta actividad se actualizan el catálogo de requisitos y el de normas y se generan los siguientes productos:

- Diseño de la arquitectura del sistema, como producto que engloba el particionamiento físico del sistema de información y la descripción de subsistemas de diseño.

- Entorno tecnológico del sistema, que a su vez, comprende la especificación del entorno tecnológico, las restricciones técnicas y la planificación de capacidades.
- Catálogo de excepciones
- Procedimientos de operación y administración del sistema
- Procedimientos de seguridad y control de acceso

6.1.1. Definición de Niveles de Arquitectura

En esta tarea se describen los niveles de la arquitectura software, mediante la definición de las principales particiones físicas del sistema de información, representadas como nodos y comunicaciones entre nodos.

Se entenderá por nodo cada partición física o parte significativa del sistema de información, con características propias de ejecución o función, e incluso, de diseño y construcción.

Para facilitar la comprensión del sistema, se recomienda identificar como nodos los elementos de infraestructura más significativos de la arquitectura en la que se va a implementar el sistema de información. Los elementos que se recomienda especificar son los siguientes:

- Gestores de datos
- Tipos de puesto cliente
- Tipos de dispositivos de impresión
- Monitores de teleproceso
- Servidores
- Comunicaciones.

La comunicación se expresará por una conexión entre nodos, indicando su carácter bidireccional o unidireccional, con las principales características de los protocolos o tipo de mensajes utilizados.

La especificación de los niveles de la arquitectura se realizará con el detalle suficiente como para permitir un diseño dirigido hacia una solución concreta, y en general no será preciso indicar en cada nodo detalles relativos al hardware, capacidad, rendimiento o configuraciones de tolerancia a fallos,

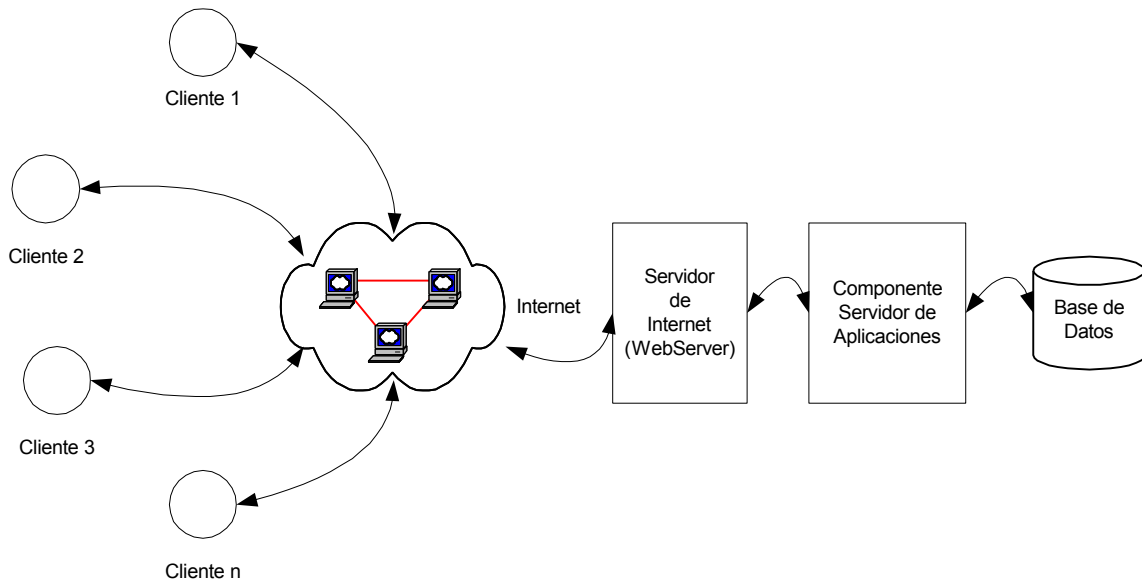
entre otros. Esta información se concretará en la tarea Especificación del Entorno Tecnológico (DSI 1.6).

Los criterios para diseñar la arquitectura se obtendrán de directrices tecnológicas y de integración según los estándares actuales de desarrollo (ver ERS), teniendo en cuenta, especialmente, aspectos no funcionales relacionados con:

- Usuarios: ubicación, movilidad, concurrencia, número, etc.
- Datos: variabilidad, volúmenes, necesidades de consolidación, seguridad, etc.
- Procesos: distribución, reutilización, concurrencia, carácter crítico, etc.

El sistema consta de múltiples nodos clientes que se comunican con un servidor centralizado. En este caso, el procesamiento de la información se encontrará distribuido entre los distintos nodos que componen la aplicación y los componentes servidores. A continuación se presenta un diagrama que muestra la distribución de los componentes de aplicación.

Diagrama de Distribución de Componentes



En el diagrama, pueden distinguirse claramente los componentes del sistema:

- **Cliente:** es un programa que contiene la cantidad mínima de componentes necesaria para operar el sistema. Si bien el mercado virtual se encontrará en ubicado en la Internet, el cliente utilizará una interfaz cliente servidor para comunicarse con el servidor (lo que facilitará la carga de datos y la interacción con el usuario)
- **Internet:** será el medio de transporte. Los componentes del sistema utilizarán la Internet para comunicarse con los componentes servidores y en futuras versiones para comunicarse entre sí.
- **Servidor de Internet:** es un tipo de servidor especializado en administrar comunicaciones utilizando el protocolo de comunicaciones HTTP (Hiper Text Transfer Protocol).
- **Componente Servidor de Aplicaciones:** es un tipo de servidor que brinda los recursos necesarios para ejecutar una aplicación. En el presente caso, es el lugar donde residirán los componentes de servidor de la aplicación.
- **Base de Datos:** la base de datos relacional donde el sistema guardará sus datos.

Esta distribución de los componentes del sistema se ha tomado teniendo en cuenta:

Los **usuarios** del sistema se encuentran distribuidos en el mundo. En principio, podemos decir que los mismos forman parte de distintas empresas que quieren comerciar entre sí, con lo cual se encontrarán en ubicaciones físicas necesariamente distintas.

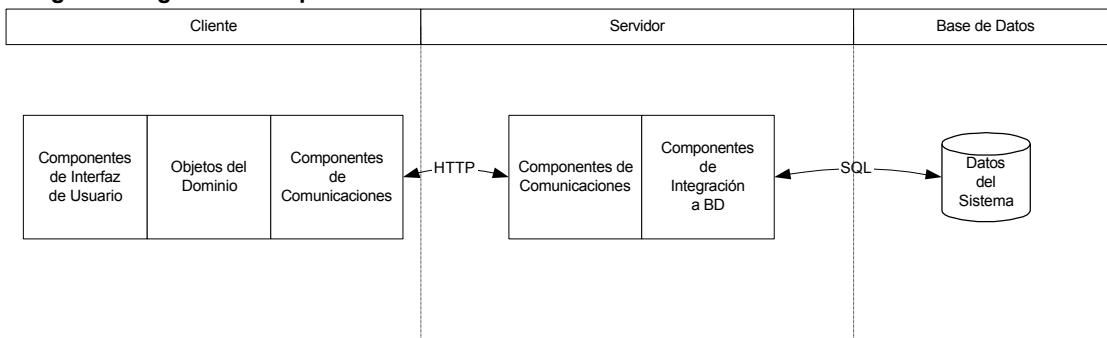
Los **datos** deben estar centralizados. Esto se manifiesta por el carácter de catálogo de productos que forma parte del sistema. Si bien no es el objetivo de esta tesis lograr un catálogo estricto según los conocimientos de áreas específicas que traten dicha cuestión, sí es necesario centralizar la administración de los productos disponibles para que el sistema "sepa" sobre su existencia y los potenciales proveedores de los mismos. Por otro lado, la información que debe manejar el sistema puede ser considerada crítica dado que de ello dependen tanto las ventas de cierto grupo de usuarios, como el abastecimiento de productos para otro grupo. La centralización de la información permitirá un único punto de "backup" para resguardar los datos así como la administración de la seguridad de los mismos. A medida que el sistema sea ampliamente utilizado, el volumen de los datos será

notoriamente creciente, con lo cual serán necesarios equipos más poderosos para la gestión de los mismos, y el hecho de centralizarlos permitirá tener un único punto de ajuste de hardware para soportarlos,

Los **procesos** se encontrarán distribuidos entre los componentes cliente y servidores de la aplicación. De esta forma los componentes cliente se encargarán de las cuestiones referentes a un usuario en particular (carga de datos, consultas, etc.) y los componentes servidores que tendrán que ser notoriamente más robustos dado que deberán soportar la concurrencia de múltiples usuarios y la gestión de los datos. Estos componentes se parecerán más a transacciones residentes en el servidor de aplicaciones. Por otro lado, es indispensable asegurar el correcto funcionamiento de los mismos y su alta disponibilidad dado que ningún nodo cliente del sistema funcionará correctamente si los componentes de servidor no se encuentran disponibles.

A partir de este análisis, podemos identificar un segundo diagrama lógico que se presenta a continuación:

Diagrama Lógico de Componentes



En este caso se ve que las distintas capas del sistema se comunican entre sí por medio de diversos mecanismos.

- **Comunicación Cliente-Servidor:** esta comunicación se lleva a cabo por medio del protocolo HTTP que funciona sobre el protocolo TCP/IP. Este mecanismo es el estándar para comunicaciones sobre Internet. Por lo tanto, para reutilizar la infraestructura subyacente a Internet, se utilizará el mismo mecanismo utilizado por los navegadores. La única diferencia se encuentra en que los componentes no hablarán HTML sino más bien XML para intercambiar datos. Estas comunicaciones no son orientadas a la conexión, es decir, por cada petición que necesita el componente cliente, se establece una conexión con el servidor, se pide la ejecución de una transacción y se cierra el canal de comunicaciones. Esto es debido a la imposibilidad de pensar en un

canal continuamente conectado a través de Internet y la necesidad de liberar recursos en el servidor para que permita atender la demanda de otros clientes.

- **Comunicación Servidor-BD:** en este caso, ambos componentes del sistema se encontrarán dentro de la misma red (probablemente en una zona desmilitarizada -DMZ-) y podrán hablar SQL sobre TCP/IP. Para ello será necesario utilizar un "driver" que permita ejecutar las consultas y actualizaciones necesarias sobre el sistema gestor de base de datos. Este mecanismo también es estándar para comunicaciones entre clientes y servidores. Esta comunicación será de índole "orientada a la conexión" donde una vez establecida la comunicación entre los componentes de servidor y sistema gestor de base de datos, la misma permanecerá activa al eterno. Se puede asumir dicha premisa ya que se parte de la base de que ambos componentes se encuentran en plataformas robustas (tanto de hardware, software y networking). Además resulta necesario de forma tal de minimizar el tiempo de respuesta de las transacciones.

Para este caso, es un requisito que las estaciones de trabajo en los clientes sean máquinas conectadas a Internet y que posean la capacidad de ejecutar un navegador. Si bien no será necesario el mismo para el uso del presente sistema, este se desarrollará teniendo en cuenta el consumo de memoria y procesador necesario por los navegadores de forma tal de permitir un uso adecuado del componente cliente.

6.1.2. Identificación de Requisitos de Diseño y Construcción

En esta tarea se especifican los requisitos no funcionales que están directamente relacionados con la adopción o diseño de una arquitectura o infraestructura concreta, y que pueden condicionar el diseño o la construcción del sistema de información.

Entre estos requisitos pueden estar los relacionados con lenguajes, rendimiento de los distintos elementos de la arquitectura, así como criterios de ubicación de módulos y datos en los distintos nodos.

Los nuevos requisitos se encuentran en el documento ERS en el anexo B.

- Los componentes del cliente serán desarrollados teniendo en cuenta soportar la carga de un único usuario por sesión de uso.

- Los componentes del servidor deberán poder soportar múltiples usuarios concurrentes al sistema.
- Requisitos de comunicaciones según se ha discutido en el apartado anterior.
- Requisitos de centralización, backup y recupero de datos centralizado.

6.1.3. Especificación de Excepciones

El objetivo de esta tarea es la definición de los comportamientos no habituales en el sistema, que reflejan situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema de información. Para ello, se establece previamente el nivel de especificación de las mismas así como los criterios de catalogación y clasificación.

Se propone su catalogación como ayuda para el diseño del sistema de información, así como guía en la especificación técnica de las pruebas, al permitir la generación de algunos casos de prueba de forma inmediata. Dicho catálogo se irá completando a partir de las actividades correspondientes al diseño detallado de los subsistemas.

Las excepciones se describen incluyendo, al menos, los siguientes conceptos:

- Tipo y descripción de la excepción
- Condiciones previas del sistema de información
- Elemento afectado (nodo, módulo, caso de uso)
- Respuesta del sistema de información

Elemento asociado a la respuesta esperada del sistema (módulo, clase, procedimiento, etc.).

Las excepciones que se proponen como obligatorias son las relacionadas con el funcionamiento general del sistema de información, habitualmente asociadas a:

- Nodos y comunicaciones del particionamiento físico del sistema de información. Este tipo de excepciones tendrá lugar cuando no estén disponibles los gestores de bases de datos o los recursos compartidos

del sistema (representados como nodos), cuando se produzcan fallos en las comunicaciones entre nodos, etc.

- Rangos o valores no válidos en la entrada de datos, como puedan ser atributos obligatorios, con formatos específicos, etc.

Se recomienda según el nivel de especificación que se establezca en cada caso, catalogar, asimismo, las excepciones particulares que se identifiquen en las actividades del diseño de detalle.

Catálogo de Excepciones

Tipo	Comunicaciones
Descripción	El componente cliente intenta conectarse al componente servidor y este no responde.
Condiciones Previas	El sistema no se encuentra conectado al servidor.
Elemento Afectado	Componente Cliente
Respuesta del Sistema	Mensaje de error al usuario indicando la imposibilidad de conectarse. "El sistema no puede conectarse con el servidor"

Tipo	Comunicaciones
Descripción	El componente servidor intenta conectarse con el sistema gestor de bases de datos pero el mismo no responde.
Condiciones Previas	El sistema no se encuentra conectado al sistema gestor de bases de datos. El componente servidor ha recibido una petición de un componente cliente para ejecutar una transacción.
Elemento Afectado	Componente Servidor
Respuesta del Sistema	El componente servidor debe notificar al componente cliente la imposibilidad de ejecutar una transacción. Este mensaje debe ser informado al usuario: "El servidor informa que es imposible ejecutar la transacción X"

Tipo	Comunicaciones
Descripción	El componente cliente se conecta con el componente servidor pero ocurre un error de comunicaciones en el medio de la transacción
Condiciones Previas	El componente cliente ejecuta una transacción en el componente servidor.
Elemento Afectado	Componente Cliente
Respuesta del Sistema	Mensaje de error al usuario indicando la imposibilidad de ejecutar correctamente la transacción: "Ha ocurrido un error de comunicaciones. Por favor, verifique que su equipo se encuentra correctamente conectado a Internet".

Tipo	Valor Requerido
Descripción	La no carga de un valor requerido. Cabe destacar que en el presente desarrollo todos los datos son requeridos para la gestión del sistema, por lo tanto se los engloba en un único tipo de excepción.
Condiciones Previas	Dato requerido no cargado
Elemento Afectado	Componente Cliente
Respuesta del Sistema	Mensaje de error al usuario indicando la necesidad de carga de dicho dato: "El dato X debe ser cargado. Por favor, verifique"

Tipo	Rango de Valores
Descripción	El único tipo de rango de valor inválido detectado consiste en que la fecha de entrega de un producto solicitada no puede ser menor a la fecha actual.
Condiciones Previas	Carga en progreso.
Elemento Afectado	Componente Cliente
Respuesta del Sistema	Mensaje de error al usuario indicando la necesidad de una fecha de entrega posterior a la fecha de carga. "Por favor, verifique que la fecha de fin del período sea mayor a la fecha de inicio"

Hay que destacar que ya es un requisito del sistema desarrollar o reutilizar componentes que validen la carga de datos específicos como ser: importes, fechas, etc. Por lo tanto no se tendrán en cuenta excepciones relacionadas con la carga de valores inválidos para cada dominio o bien un formato inválido.

Como se detalla en los siguientes apartados, estos componentes forman parte de una librería de comercio electrónico a reusar en esta aplicación.

6.1.4. Especificación de Estándares y Normas de Diseño y Construcción

En esta tarea se especifican los estándares técnicos y de nomenclatura, normas y recomendaciones, que generalmente están relacionados con la adopción o diseño de una arquitectura o infraestructura tecnológica concreta, y que pueden condicionar el diseño o la construcción del sistema de información.

La información recogida en el catálogo se deberá tener en cuenta en la elaboración de los productos resultantes del diseño y construcción del sistema de información siendo producto de entrada en todas las tareas, aunque se omitirá la referencia al mismo, por sencillez.

- Será necesario desarrollar la aplicación en un lenguaje orientado a objetos de forma tal de facilitar la implementación y permitir una correspondencia entre el diseño y los componentes de

implementación. Por otro lado, dichos componentes deberán ser desarrollados en un lenguaje portable entre distintas plataformas para facilitar su uso y alcanzar una mayor cantidad de usuarios del sistema. Se propone el lenguaje de desarrollo Java.

- El sistema funcionará en forma distribuida. Por lo tanto necesitará de un medio de comunicación entre sus componentes. Para que el mismo pueda ser utilizado sobre la Internet, todos los medios de comunicación entre componentes serán desarrollados teniendo tecnologías abiertas de Internet (familia de protocolos TCP/IP).
- Los componentes se deben comunicar entre sí mediante el uso de un protocolo de comunicaciones (de nivel aplicativo). Para ello se utilizará el protocolo XML que es el estándar abierto más utilizado de la actualidad.
- La empresa Sun Microsystems [Sun02] plantea una serie de tecnologías y arquitecturas para utilizarlas en sistemas desarrollados para la Internet. El sistema a desarrollar deberá basarse en dichas premisas al momento de su diseño y concepción de forma tal de generar un software consistente con los estándares mundiales de desarrollo.

6.2. Identificación de Subsistemas de Diseño

En esta tarea se divide de forma lógica el sistema de información en subsistemas de diseño, con el fin de reducir la complejidad y facilitar el mantenimiento. Se toman como referencia inicial los subsistemas de análisis especificados en el proceso de Análisis del Sistema de Información (ASI).

La división en subsistemas de diseño se puede realizar con una continuidad directa de los modelos del análisis, o aplicando nuevos criterios de diseño, entre los que se pueden citar los siguientes:

- Facilidad de mantenimiento
- Reutilización de elementos del propio sistema o de la instalación
- Optimización de recursos (por ejemplo, líneas de comunicaciones)
- Características de ejecución (interactivo o por lotes)
- Funcionalidad común

- Aplicación de mecanismos genéricos de diseño a nivel de arquitectura.

Los subsistemas resultantes se califican como específicos o de soporte, asignando cada subsistema al nodo correspondiente.

Los subsistemas específicos contemplan las funcionalidades propias del sistema de información, mientras que los de soporte cubren servicios comunes proporcionando un acceso transparente a los distintos recursos. Estos últimos están relacionados con:

- Comunicaciones entre subsistemas
- Gestión de datos (acceso a bases de datos, ficheros, áreas temporales, importación y exportación de datos, sincronización de bases de datos, etc.)
- Gestión de transacciones
- Control y gestión de errores
- Seguridad y control de acceso
- Gestión de interfaz
- Interacción con los recursos propios del sistema.

La interacción del sistema de información con la infraestructura que le da soporte, así como con el resto de los sistemas y servicios de la instalación, puede originar la necesidad de nuevos subsistemas, módulos, clases o servicios no especificados en el análisis.

La definición del comportamiento externo de cada subsistema se completa durante el diseño de detalle con la especificación de su interfaz, así como con la dependencia entre subsistemas.

El diseño de detalle de los subsistemas identificados, por criterios de optimización y reutilización, puede aconsejar la reorganización y reubicación de los elementos que forman parte de cada subsistema y, a su vez, puede dar lugar a la identificación de nuevos subsistemas de soporte.

La ubicación de subsistemas en nodos y la dependencia entre subsistemas se especifica por medio de técnicas matriciales, o bien en lenguaje natural o seudo código.

Al entrar en un plano mayor de detalle dentro del ámbito del proyecto, resulta necesario tomar un enfoque distinto al tomado en el momento del

análisis para dividir el sistema en subsistemas. En el caso del análisis, dadas las características de la etapa, resultaba necesario dividir el sistema bajo ciertas heurísticas (allí mencionadas) que facilitaban el entendimiento del problema, dividiendo al sistema en dos grandes subsistemas utilizados para comprar y vender.

Al referirse al diseño del sistema, resulta necesario utilizar otras heurísticas tendientes a organizar los componentes del sistema, a fin de orientar su futura implementación. Esta división de subsistemas se puede observar en la figura diagrama lógico de componentes del apartado 5.1.1. El sistema es dividido en grandes módulos con funcionalidad bien específica. De esta forma se obtienen los siguientes subsistemas:

- Subsistema de Interfaz de Usuario
- Subsistema del Dominio del Sistema
- Subsistema de Comunicaciones
- Subsistema de Integración a la Base de Datos

Al hablar de subsistemas de diseño, se hace referencia a componentes compuestos de clases de diseño, casos de uso, interfaces y otros subsistemas (recursivos). Un subsistema provee interfaces que representan la funcionalidad que exportan en términos de operaciones.

Un subsistema debe ser cohesivo. Esto significa que sus componentes deben estar fuertemente relacionados. Además, los subsistemas deben encontrarse escasamente acoplados, es decir, la dependencia entre estos debe ser mínima.

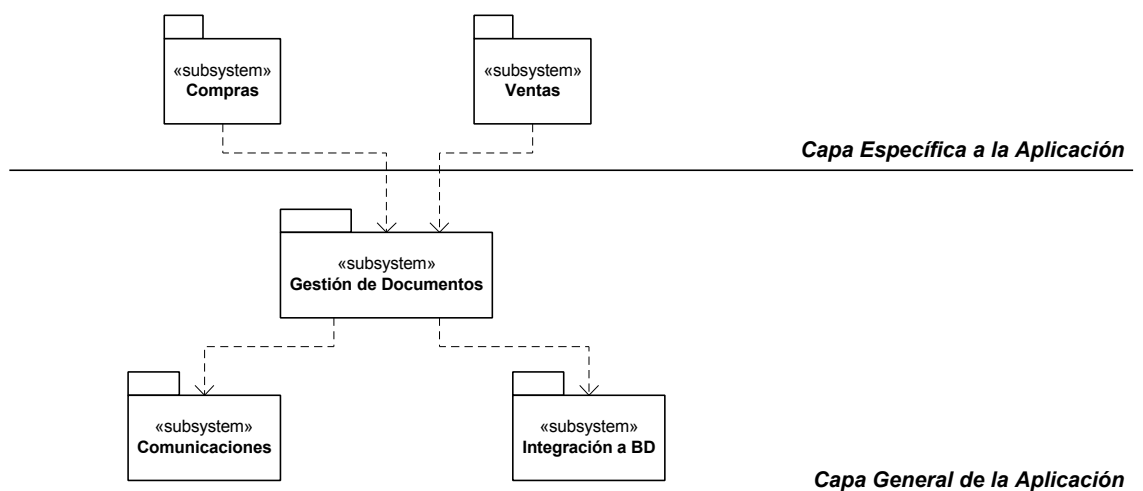
En este caso, el diseño resulta orientado al conocimiento de los mismos mediante interfaces más que en la implementación propiamente dicha.

Además, los subsistemas deben poseer las siguientes características:

- Los subsistemas deben representar la separación de distintos tópicos del diseño. En sistemas de gran tamaño los subsistemas pueden ser diseñados en forma separada por distintos equipos de proyecto. O bien, pueden reutilizarse subsistemas completos (ej: subsistema de comunicaciones).

- Los niveles más altos del sistema (por lo general los que modelan las capas de presentación y modelo de negocios) suelen tener una trazabilidad trivial al modelo de análisis.
- Los subsistemas pueden representar componentes de alto nivel en cuanto a la implementación del sistema se refieren. Es decir, proveen de interfaces claras de uso y pueden ser tenidos en cuenta como componentes en sí mismos, aunque queda claro que están contruidos por componentes de un nivel de granularidad más bajo y específico.
- Los subsistemas pueden incluso representar la reutilización de productos ya contruidos o tecnología ya disponible, en cuyo caso solo se modela lo necesario para adaptar su interfaz para el consumo en el sistema en construcción.
- Si bien no es el caso de este desarrollo, los subsistemas también pueden representar sistemas legados (heredados, ya existentes en el contexto de la compañía a utilizar el software a desarrollar) en cuyo caso también se modela la interfaz de acceso a la funcionalidad de dicho sistema.

Diagrama de Subsistemas



Como se observa en la figura, los distintos subsistemas conforman capas de la arquitectura. Por cuestiones de simplificación se han tenido en cuenta solamente aquellos subsistemas a desarrollar y no los ya

desarrollados (que conforman parte de la librería de clases propia del lenguaje de programación a utilizar).

En este caso es posible encontrar subsistemas específicos para los casos de Compras, Ventas y Gestión de Documentos. Por otro lado, sistemas de soporte para los casos de Comunicaciones e Integración a BD.

Como ya lo indicaba la definición de subsistema desde el punto de vista del proceso unificado, estos subsistemas no necesariamente son excluyentes para su posterior desarrollo. Esto es dado por el carácter de composición a partir de componentes triviales o bien de otros subsistemas. Esto se pondrá en manifiesto con posterioridad al ver que los subsistemas de Compras, Ventas y Gestión comparten un paquete común que conforma la implementación del dominio del problema (Pedidos de Cotización, Ordenes de Compra, Usuarios, etc.).

6.2.1. Especificación del Entorno Tecnológico

En esta tarea se definen en detalle los distintos elementos de la infraestructura técnica que dan soporte al sistema de información, determinando la implementación concreta de los nodos y comunicaciones especificados en la tarea Definición de Niveles de Arquitectura.

Se propone agrupar los elementos de la infraestructura en los siguientes conceptos:

- Hardware: procesadores, unidades de almacenamiento, estaciones de trabajo, etc.
- Software: sistemas operativos, subsistemas, sistemas de comunicaciones, gestores de bases de datos, sistemas de ficheros, software de base, herramientas y utilidades de gestión propias del sistema, etc.
- Comunicaciones: diseño de la topología de la red, protocolos, nodos de red, etc.

La definición de los distintos elementos puede generar restricciones técnicas que afecten al diseño o construcción del sistema de información.

Asimismo, se realiza una estimación de la planificación de capacidades (capacity planning) o se especifican los parámetros que Explotación y Sistemas precisen para realizar dicha planificación. Se indicarán, al menos, las necesidades previstas de:

- Almacenamiento: espacio en disco, espacio en memoria, pautas de crecimiento y evolución estimada del sistema de información, etc.
- Procesamiento: número y tipo de procesadores, memoria, etc.
- Comunicaciones: líneas, caudal, capacidades de elementos de red, etc.

Para poder determinar la planificación de capacidades, es necesario conocer los diseños detallados de los módulos/clases y escenarios, incluida la información de control en las comunicaciones, así como el diseño físico de datos optimizado, productos que se están generando en paralelo a esta actividad. También se tendrán en cuenta, cuando proceda, las estimaciones de volúmenes de datos propios de la migración y carga inicial de datos.

Los requisitos que se presentarán a continuación serán los necesarios para permitir la instalación del mismo y su uso para demostraciones y cuestiones de aprobación del proyecto de tesis y pruebas de aceptación del usuario.

6.2.1.1. Hardware

El sistema podrá ser ejecutado en equipos de distinta tecnología. Se preveen las siguientes configuraciones mínimas:

- Plataforma Intel: procesador Pentium II o superior. 128 Mb RAM. 2 Gb libres de espacio en disco. Placa de Red.
- Plataforma Sun: procesador UltraSparc II o superior. 128 Mb RAM. 2Gb libres de espacio en disco.

6.2.1.2. Software

- Plataforma Intel: sistema operativo RedHat Linux 7.1 ó Microsoft Windows 2000 Profesional.
- Plataforma Sun: sistema operativo Solaris 7 o superior.

Para ambos casos, el software adicional será:

- Java Software Development Kit versión 1.2 o superior.
- Servidor de aplicaciones IBM WebSphere ó Apache Tomcat.
- Base de datos MySQL versión 3 o superior.

6.2.1.3. Comunicaciones

Como ya se ha mencionado con anterioridad, el sistema esta preparado para su uso sobre Internet. Para las comunicaciones, debe encontrarse habilitado el tráfico TCP/IP entre todos los componentes del sistema en forma directa (es decir, sin utilizar proxies ni componentes de indirección alguno).

El sistema necesitará un ancho de banda mínimo de 32 Kbps para su correcto funcionamiento.

6.2.2. Especificación de Requisitos de Operación y Seguridad

El objetivo de esta tarea es definir los procedimientos de seguridad y operación necesarios para no comprometer el correcto funcionamiento del sistema y garantizar el cumplimiento de los niveles de servicios que exigirá el sistema en cuanto a la gestión de operaciones (servicios batch, seguridad, comunicaciones, etc.). Los niveles de servicio se especificarán formalmente en el proceso Implantación y Aceptación del Sistema (IAS).

Tomando como referencia los requisitos no funcionales establecidos para el sistema y teniendo en cuenta la arquitectura propuesta y las características del entorno tecnológico definido en esta actividad, se especifican los requisitos de seguridad y control de acceso necesarios para garantizar la protección del sistema y minimizar el riesgo de pérdida, alteración o consulta indebida de la información.

6.2.2.1. Acceso al sistema y a sus recursos

El sistema cuenta con una base de datos relacional para almacenar sus datos y ejecutar determinadas transacciones. El acceso a dichos datos, estructura de datos y transacciones se encuentra protegido por el mecanismo de autenticación básica provisto por el vendedor de la base de datos en cuestión. En este caso la base de datos provee un sistema de seguridad basado en usuario y contraseña y un mecanismo que permite legislar los equipos desde los cuales es posible conectarse (aún teniendo un par usuario y contraseña válido), o bien, delegar la autenticación del usuario al sistema operativo.

Los pares usuario/contraseña que utilizarán los usuarios del sistema son de nivel aplicativo, es decir, son administrados por la aplicación y no tienen sentido o uso a nivel de sistema operativo o base de datos.

Las librerías del sistema se almacenarán en archivos en el sistema operativo. El usuario tendrá acceso a las librerías en el componente cliente aunque no podrá hacerlo con las librerías del componente servidor donde se encuentran las transacciones.

6.2.2.2. Mantenimiento de la integridad y confidencialidad de los datos

La integridad de los datos se mantiene mediante el uso de transacciones ACID y por medio de un buen diseño de base de datos.

La confidencialidad de los datos se obtiene dado que la aplicación no permite ver los datos que no estén relacionados con el usuario autenticado que esta usando el sistema; y por otro lado, la base de datos se encuentra protegida detrás del servidor de aplicaciones además de que los usuario carecen de una forma de ingresar al equipo servidor y al servidor de base de datos.

6.2.2.3. Control y registro de accesos al sistema

Como parte de los requisitos de auditoria del sistema, se llevará un log (en el servidor) con todas las operaciones no triviales realizadas por el usuario, como ser:

- Ingresar al sistema
- Ejecutar una transacción

Dicho registro será en formato ASCII para poder ser consultado fácilmente sin ningún tipo de software adicional. Se define un nombre compuesto para el archivo de log, es decir, el mismo será *año-mes-día-vmarket.log*. De esta forma el log será distinto para cada día de operación del sistema, permitiendo su extracción para análisis y/o la eliminación del mismo en forma rutinaria para no acumular información innecesaria en el dispositivo de soporte.

6.2.2.4. Copias de seguridad y recuperación de datos y su periodicidad

Las bases de datos relacionales proveen mecanismos específicos para los resguardos de seguridad y la recuperación ante una eventual necesidad. Estos mecanismos varían desde el backup a nivel sistema de

archivos hasta copias replicadas en línea para cambiar el servidor de base de datos y continuar operando sin interrupciones.

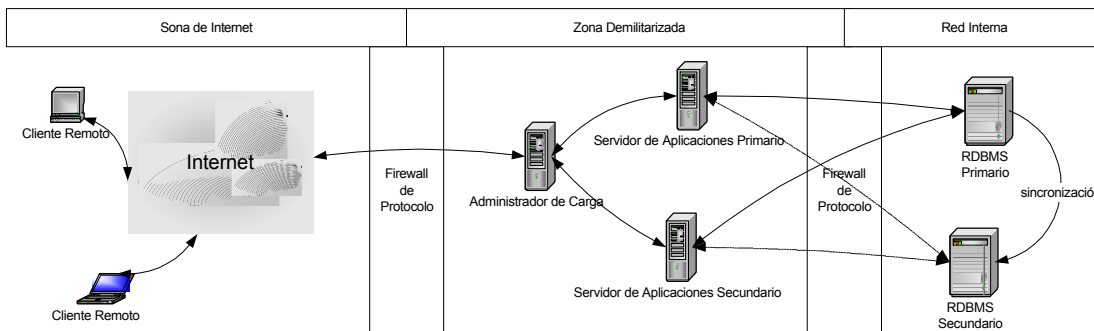
Dado que el proyecto actual esta involucrado dentro del marco de un trabajo académico, que no cuenta con presupuesto alguno y que el RDBMS utilizado carece de características avanzadas de backup y recupero, se recomienda un esquema de backup diario basado en copias del sistema de archivo en CD-ROM.

En el caso de tratarse de un sistema en producción, se recomienda el uso de un servidor adicional de alta disponibilidad con una copia sincronizada en línea con el RDBMS original (para garantizar la disponibilidad de datos). Por lo general el software de sincronización de bases de datos se basa en el registro de transacciones para sincronizar transacciones enteras una vez terminadas (commit).

También se puede hablar de la disponibilidad de varios servidores de aplicaciones para dar alta disponibilidad a los clientes, balanceo de carga entre los servidores, y un primer manejo de una contingencia ocasionada por una falla (en inglés fail-over). De esta forma, en caso de que uno de los servidores quede fuera de servicio el otro podría hacerse cargo del tráfico. O bien, ambos captar una porción del tráfico para balancear la carga sobre los equipos.

El diagrama a continuación ejemplifica lo antedicho:

Esquema de Seguridad



6.2.2.5. Recuperación ante catástrofes

Con la implementación de un esquema como el planteado en el punto anterior, se minimizan las posibilidades de catástrofes. Esto es aún más

evidente si los servidores secundarios se encuentran físicamente ubicados en sitios remotos de contingencia.

6.2.2.6. Tratamiento interactivo

El sistema funciona netamente bajo un modo de explotación interactivo. Como todo sistema de Internet, se espera que funcione las 24 horas del día, los 7 días de la semana; en una palabra: siempre.

Esto es posible gracias a la redundancia de servidores tanto de aplicaciones como de base de datos.

Por otro lado, dependiendo de la capacidad de los servidores se puede definir una cantidad máxima de clientes concurrentes (calidad de atención). Teniendo en cuenta los equipos planteados anteriormente (definición de hardware) el sistema tiene una capacidad de atención de 2000 usuarios concurrentes por servidor. Dado que la aplicación cliente se comunica en una forma no orientada a la conexión con el servidor, 2000 usuarios concurrentes no significa 2000 usuarios con la aplicación abierta, sino más bien, 2000 usuarios que al mismo tiempo disparan una transacción contra el servidor (en el mismo instante). Esto eleva la capacidad a un número muy superior.

6.2.2.7. Tratamiento por lotes

El sistema no contempla trabajos por lotes.

6.2.2.8. Control y planificación de trabajos

No existe planificación de trabajos ni control sobre los mismos. En todo caso el administrador de carga es el encargado de distribuir la ejecución de transacciones a los servidores que se encuentran dedicados a dicha gestión.

6.2.2.9. Recuperación y reanudación de trabajos

Los trabajos no pueden ser recuperados o reanudados. Simplemente, cada transacción puede ejecutarse satisfactoriamente, o no. En caso de una interrupción en el servicio sucederá lo siguiente:

Origen de Interrupción	Consecuencias
Fallo en el cliente que ocasiona la caída del componente cliente del sistema.	El sistema no guarda estado de sus objetos en el componente cliente. Con lo cual, cualquier transacción no terminada (enviada al servidor) será deshecha por

	completo. Al reactivarse el sistema, el usuario deberá volver a ejecutar los pasos hechos anteriormente.
Fallo en el componente servidor de aplicaciones que ocasione la caída del servidor.	El sistema tampoco almacena estado en el componente servidor, dado que este componente se basa en transacciones independientes cuya secuencia es ejecutada por el cliente. En caso de caerse este componente no se generarán inconsistencias. De existir transacciones abiertas, las mismas serán vueltas al estado consistente anterior.
Fallo en el componente de SGBD que ocasione la caída del componente.	La base de datos posee lógica de gestión de transacciones. Por lo tanto, una caída en este componente resultará en volver las transacciones abiertas al último estado consistente anterior.

6.3. Diseño de la Arquitectura de Soporte

En esta actividad se lleva a cabo la especificación de la arquitectura de soporte, que comprende el diseño de los subsistemas de soporte que se identificaron en la actividad de Definición de la Arquitectura del Sistema (DSI 1), y la identificación de los mecanismos genéricos de diseño. Estos últimos, sirven de guía en la utilización de diferentes estilos de diseño, tanto en el ámbito global del sistema de información, como en el diseño de detalle.

El diseño de los subsistemas de soporte, conceptualmente, es similar al diseño de los subsistemas específicos, aunque se aconseja la participación de perfiles técnicos, con una visión global de la instalación, y con unos objetivos claros de reutilización. De esta manera se consigue simplificar y abstraer el diseño de los subsistemas específicos de la complejidad del entorno tecnológico, dotando al sistema de información de una mayor independencia de la infraestructura que le da soporte.

Esta actividad se realiza en paralelo al diseño detallado, debido a que existe una constante realimentación, tanto en la especificación de los subsistemas con sus interfaces y dependencias, como en la aplicación de esqueletos o patrones en el diseño.

Los productos resultantes de esta actividad son:

1. Diseño detallado de los subsistemas de soporte
2. Mecanismos genéricos de diseño y construcción.

6.3.1. Diseño de Subsistemas de Soporte (DSI 2.1)

El objetivo de esta tarea es la especificación y diseño de los módulos / clases que forman parte de los subsistemas de soporte identificados en la tarea Identificación de Subsistemas de Diseño (DSI 1.5) y se llevará a cabo siempre y cuando no se disponga en la instalación de servicios comunes que respondan satisfactoriamente a los requisitos planteados.

El nivel de reutilización de los subsistemas de soporte y sus servicios es potencialmente alto, de modo que se intentará reutilizar, en la medida de lo posible, los subsistemas que ya existan en la instalación y se consideren viables. En cualquier caso, cuando proceda realizar el diseño de dichos subsistemas, se recomienda hacerlo con ese fin.

El diseño sigue las mismas pautas que el establecido para los subsistemas específicos, aunque con las siguientes particularidades:

- Generalmente será necesaria una descomposición de los subsistemas de soporte en servicios, entendiendo como tales módulos o clases independientes y reutilizables. Se recomienda realizar una descripción de la interfaz y del comportamiento de cada servicio previo a su diseño de detalle, que permita completar el diseño de los subsistemas específicos.
- La especificación y diseño de cada servicio, módulo o clase se realiza con las técnicas habituales de especificación y diseño de módulos o clases, o incluso opcionalmente si la simplicidad de los elementos lo aconseja, otros lenguajes de especificación, pseudocódigo o lenguaje natural.
- A medida que se lleva a cabo esta tarea pueden surgir comportamientos de excepción que deben contemplarse igualmente en el diseño, y que en función del nivel de especificación que se haya establecido, se incorporarán al catálogo de excepciones.

6.3.2. Diseño Detallado de los Subsistemas de Soporte

Existen principalmente dos subsistemas de soporte para la presente aplicación. Estos son los subsistemas de Comunicaciones y de Integración con la base de datos. El primero se relaciona con las comunicaciones entre

componentes cliente y servidor del sistema. El segundo se encarga de las comunicaciones de los componentes de negocios con la base de datos que en este caso es relacional.

No se profundiza en las cuestiones relacionadas al registro de auditoría ya que es una característica común de los servidores de Internet que es aprovechada para este sistema.

6.3.2.1. Subsistema de Integración con la Base de Datos

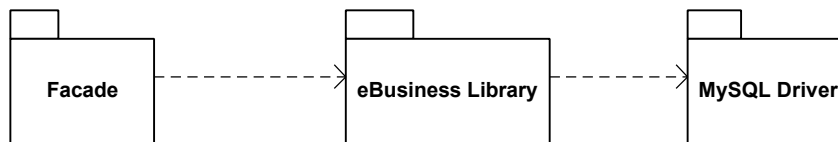
El objetivo del presente diseño detallado consiste en encapsular las cuestiones de implementación de las transacciones dentro del ámbito del servidor. Esta idea nos permite separar la necesidad de información de la base de datos, o bien, la ejecución de una transacción que impacte en la misma de los componentes relacionados al modelado de la problemática de negocios. Por otro lado, estos componentes no se relacionan con los componentes de comunicaciones.

Típicamente, para comunicarse con una base de datos es necesario utilizar un "driver". Este componente de software es desarrollado y suministrado por la compañía a cargo del desarrollo de la base de datos.

Si bien este driver permite comunicarnos con la base de datos, por lo general no tiene en cuenta cuestiones relacionadas con la escalabilidad del producto, entre otras. Es necesario tener en cuenta este requisito dado que los componentes servidores tienen que permitir una escalabilidad tal que permita a n clientes de la aplicación estar trabajando sobre la base de datos en forma concurrente. Por esta cuestión se utilizará una librería que permita obtener dicha capacidad, la cual ya se encuentra implementada y será reutilizada para el presente proyecto, referenciándola de ahora en más como "eBusiness Library".

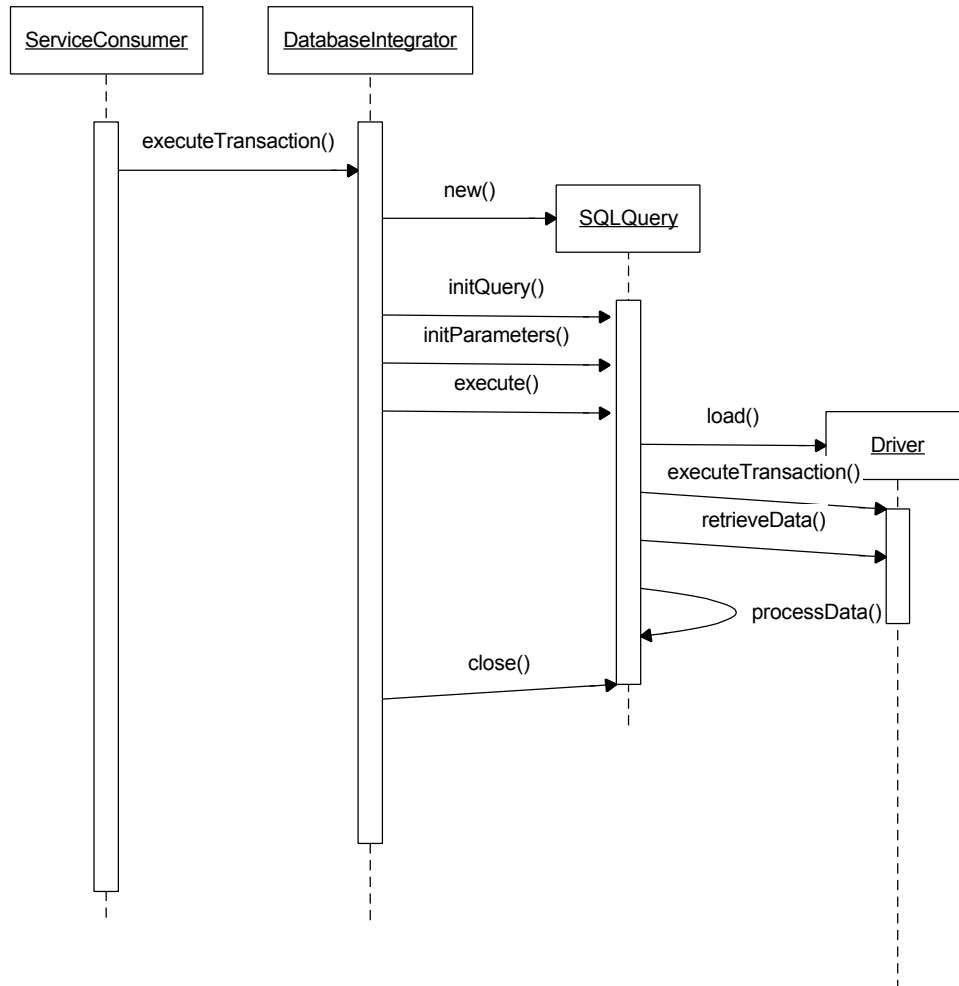
Teniendo en cuenta estas premisas, queda un único punto por modelar que radica en la API (Application Programming Interface) necesaria para sustentar el uso de la aplicación. Desde este punto de vista será necesario desarrollar una clase que implemente una interfaz de forma tal de ocultar la complejidad del subsistema de integración brindando métodos que ejecuten las transacciones relevantes. Según Gamma [Gamma1994], esta clase de interfaz sería un diseño "Facade".

Dependencias de paquetes del subsistema de integración a BD



Simplificando, la integración con la base de datos estará resuelta por una única clase que brinde un método para cada transacción a ejecutar sobre la base de datos (tanto de consulta como de impacto). La implementación de los métodos necesarios para dicho fin deberán seguir un esqueleto común, relacionado al siguiente diagrama de interacción:

Diagrama de Secuencia Genérico para Integración de Bases de Datos



En el diagrama se explica un ejemplo sobre el uso del subsistema de comunicaciones para la ejecución de una transacción en la base de datos centralizada en el servidor. Una instancia de una clase cualquiera que necesite consumir los servicios de la clase de integración ejecuta un método de la misma. En este caso, dicha clase delega la ejecución en la clase ConsultaSQL. La clase de integración instancia y parametriza una instancia de ConsultaSQL la cual encapsula el consumo del "driver" de base de datos, tal y como se esperaba. Esta capa, agrega además la posibilidad de intercambiar en un futuro la base de datos y por lo tanto el driver correspondiente, aislando al resto de la aplicación de dicha problemática. La capa intermedia convierte el resultado de la ejecución de la consulta o

transacción generando una representación intermedia independiente del driver. Devuelve dicho resultado a la capa de integración, la cual instanciará las clases necesarias y las devolverá al consumidor de servicios.

Este último paso es importante dado que se encuentra dentro del ámbito de una implementación orientada a objetos. En este caso la clase de integración es responsable por devolver instancias válidas del modelo de objetos en vez de filas de una tabla, propias de un modelo relacional. Si bien el SGBD seleccionado es relacional, esta conversión es realizada por la capa de integración que "habla" objetos contra los componentes clientes y SQL contra la base de datos según el siguiente diagrama:



Dado que esta capa de software se compone de una única clase, no tiene sentido profundizar en el diagrama de clases aunque sí especificar los métodos a implementar en la misma. No se hace referencia a variables de instancia ya que como buena clase de servicios, no posee un estado interno (la llamada a cada uno de sus métodos es transaccional y carece de estado).

<i>VirtualMarketImpl</i>		
authenticate	público	Autentica si un par usuario/contraseña corresponden a un usuario del sistema.
	Parámetros	IN - String - userName IN - String - password
	Resultado	Una instancia de comprador o proveedor según corresponda.
insert	público	Inserta un pedido de cotización en la base de datos e informa a los potenciales proveedores.
	Parámetros	IN - PricingOrderRequest - anOrder
	Resultado	Ninguno.
insert	privado	Inserta un ítem de un pedido de cotización en la base de datos.

	Parámetros	IN - PricingOrderRequestItem - anItem
	Resultado	Ninguno.
insert	público	Inserta una cotización en la base de datos e informa al solicitante.
	Parámetros	IN - PricingOrderRespense - anOrder
	Resultlado	Ninguno.
insert	privado	Inserta un ítem de una cotización en la base de datos.
	Parámetros	IN - PricingOrderResponseItem - anItem
	Resultado	Ninguno.
insert	público	Inserta una orden de compra en la base de datos y notifica a los proveedores.
	Parámetros	IN - PurchaseOrder - anOrder
	Resultado	Ninguno.
insert	privado	Inserta un ítem de una orden de compra en la base de datos.
	Parámetros	IN - PurchaseOrderItem - anItem
	Resultado	Ninguno.
retrieveNonExpiredPricingOrderResponses	público	Recupera todas las cotizaciones vigentes (no expiradas) para un cliente.
	Parámetros	IN - Customer - aCustomer
	Resultado	Un vector con todas las cotizaciones vigentes hechas para un cliente.
retrieveOpenPricingOrderRequests	público	Recupera todos los pedidos de cotización pendientes de cotización para un proveedor determinado.
	Parámetros	IN - Supplier - aSupplier
	Resultado	Un vector con todos los pedidos de cotización pendientes de respuesta.

retrievePossibleSuppliersFor	público	Recupera una lista de los proveedores que pueden proveer por lo menos uno de los ítems de un pedido de cotización.
	Parámetros	IN - PricingOrderRequest - anOrder
	Resultado	Un vector que contiene instancias de los potenciales proveedores de al menos uno de los ítems de un pedido de cotización.
retrievePricingOrderRequest	público	Recupera una instancia de pedido de cotización a partir de su identificador.
	Parámetros	IN - Integer - pricingOrderRequestId
	Resultado	Una instancia de pedido de cotización.
retrievePricingOrderRequestItems	público	Recupera los ítems de un pedido de cotización a partir del identificador del pedido de cotización.
	Parámetros	IN - Integer - pricingOrderRequestId
	Resultado	Un vector que contiene instancias de los ítems de un pedido de cotización.
retrievePricingOrderResponse	público	Recupera una instancia de una cotización a partir del identificador.
	Parámetros	IN - Integer - pricingOrderResponseId
	Resultado	Una instancia de cotización.
retrievePricingOrderResponseItems	público	Recupera los ítems de una cotización a partir del identificador de la cotización.
	Parámetros	IN - Integer - pricingOrderResponseId
	Resultado	Un vector con las instancias de los ítems de una cotización.
retrievePricingOrderResponsesIncluding	público	Recupera todas las cotizaciones hechas para un cliente que incluyan un producto en particular.

	Parámetros	IN - Integer - customerId IN - Integer - productId
	Resultado	Un vector con cotizaciones.
retrieveProductById	público	Recupera un producto a partir de su identificador.
	Parámetros	IN - Integer - productId
	Resultado	Una instancia de producto.
retrievePurchaseOrdersFor	público	Recupera las órdenes de compra para un proveedor.
	Parámetros	IN - Integer - supplierId
	Resultado	Un vector con ordenes de compra.
retrieveUserById	público	Recupera un usuario a partir de su identificador.
	Parámetros	IN - Integer - userId
	Resultado	Una instancia de proveedor o comprador.

6.3.2.2. Subsistema de Comunicaciones

En la actualidad existe una gran cantidad y variedad de métodos de comunicaciones disponibles de forma tal de permitir que nuestros componentes en el cliente se comuniquen con aquellos que se encuentran ejecutándose en el lado servidor. Entre estos métodos podemos encontrar:

- RMI (Remote Method Invokation)
- CORBA (Common Object Request Broker Architecture)
- XML-RPC
- SOAP (Simple Object Access Protocol)
- Enfoques mixtos.
- Otros

Los dos primeros se encuentran más orientados a comunicaciones entre componentes en una red de área local (LAN) dado que el enfoque

consiste en obtener una referencia a un objeto en el servidor para posteriormente invocar sus métodos bajo demanda (cada invocación implica tráfico de datos entre el componente cliente y el servidor).

Los otros, se utilizan en la actualidad para ejecutar componentes y servicios disponibles sobre la Internet, aunque su uso necesita de tecnologías y productos específicos.

Para este proyecto, se utilizará un enfoque mixto basado en el envío y recepción de mensajes XML sobre paquetes HTTP.

La implementación de un esquema de comunicaciones tal, trae ciertas consideraciones que deben ser tenidas en cuenta en el momento del diseño y posterior implementación.

Desde el punto de vista del diseño, es necesario identificar el punto de corte donde se diferencian los componentes que formarán parte del cliente y del servidor.

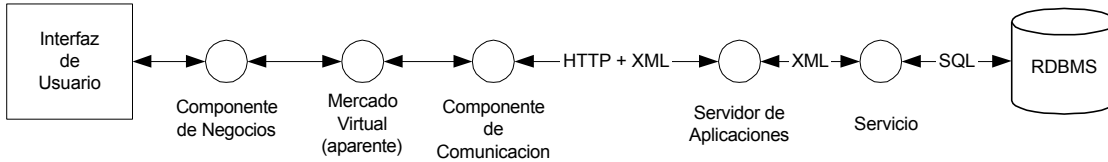
Con esta premisa en mente, se pueden clasificar a los componentes según el siguiente criterio:

- **Entidades.** Donde el estado está relacionado a la teoría de objetos como el conjunto de valores de sus variables de instancia. Son los objetos que representan las instancias de negocios (pedido de cotización, cliente, proveedor, orden de compra, producto, etc.). Cada instancia es distinta a las demás por lo que significa en el negocio.
- **Servicios.** Son los servicios que utiliza la aplicación. Estos representan las transacciones y los procesos de negocio necesarios para implementar la aplicación pero que pueden ser llamados en distinto orden o cantidad sin que esto signifique un cambio de comportamiento en su accionar. No es necesario identificar instancias de dichos objetos dados que los mismos son igualmente útiles.

Los componentes de servicios se encontrarán en el servidor, y los componentes que representen entidades vivirán principalmente en el cliente y viajarán desde y hacia el servidor (persistencia).

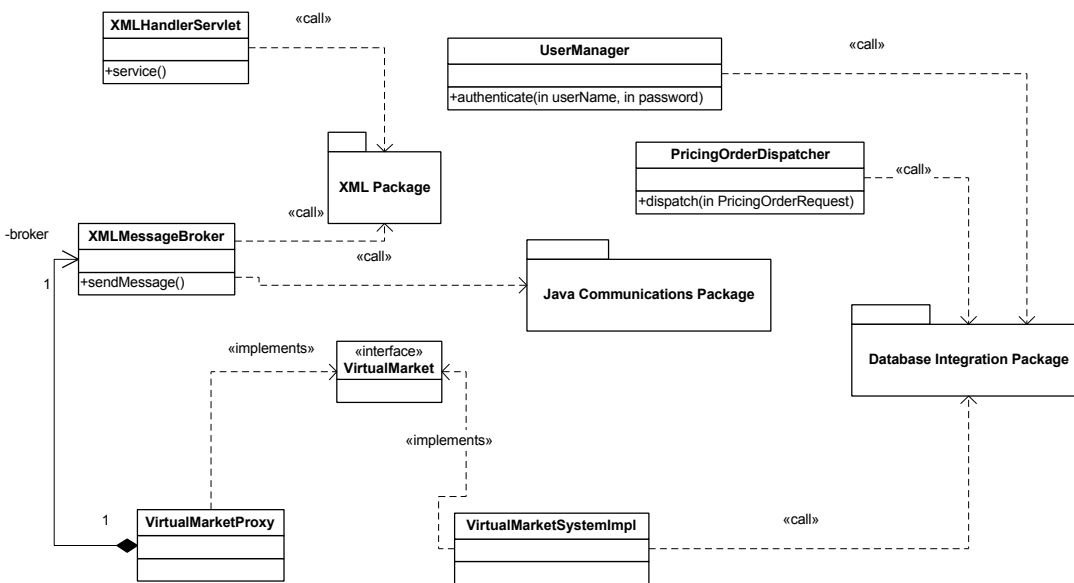
Teniendo en cuenta el diseño del modelo de integración con la base de datos, se define una nueva capa de objetos que representarán las reglas de negocio y otros que se encargarán de las comunicaciones propiamente dichas. La figura a continuación esquematiza dicha idea.

Diagrama Ejemplo del Modelo de Comunicaciones



Ahora, para hablar de un diseño detallado es necesario partir de definir las clases que intervienen en dicho modelo (diagrama estático de clases) y de la forma en que interactúan para llevar a cabo su responsabilidad (diagrama de interacción).

Diagrama de Clases del Paquete de Comunicaciones y Componentes del Servidor



Dentro del presente diseño, identificamos las clases e interfaces:

VirtualMarket

Para garantizar la portabilidad del código, se define una interfaz genérica que implementa todos los métodos necesarios para ejecutar los servicios en el servidor.

Esta premisa se basa en aquella heurística de la orientación a objetos que dice que dos objetos se conocen mediante un contrato o protocolo que es el conjunto de métodos que exportan [RebecaRevisar]. Por lo tanto, esta

interfaz específica (en forma abstracta) todos los métodos que deben ser implementados por el objeto que hará las veces de servidor.

VirtualMarketSystemImpl

Esta clase es la puerta de entrada a la funcionalidad en el servidor. Cualquier interacción de los componentes cliente con el servidor deben pasar por alguno de sus métodos. Esta clase implementa la interfaz VirtualMarket ya definida. Además, esta clase hace las veces de "Facade" [Gamma1994] es decir, encapsula todo el sistema compuesto por el servidor. En definitiva, las reglas de negocios y acceso a datos se hará mediante esta clase.

VirtualMarketProxy

Esta clase tiene la responsabilidad de parecer ser el mercado virtual. Toda la funcionalidad que se necesita invocar en el servidor es ejecutada contra una instancia de esta clase, la cual utiliza el XMLMessageBroker para generar los mensajes XML y enviarlos al servidor. Esta clase es una implementación del diseño Proxy [Gamma1994].

XMLMessageBroker

Esta clase se encarga de generar los mensajes XML y enviarlos al servidor dentro de un paquete HTTP, esperar la respuesta y devolverla al VirtualMarketProxy.

XMLHandlerServlet

Esta clase es una implementación del esquema de extensión de servidores de aplicaciones según Sun Microsystems [Sun Microsystems 1999b]. Es un componente que recibe el XML y ejecuta una transacción sobre el servicio necesario. Obtiene la respuesta y la devuelve en formato XML al cliente. El principal servicio a ejecutar está compuesto por todos los métodos implementados en VirtualMarketImpl.

PricingOrderRequestDispatcher

Esta clase surge del análisis. Su responsabilidad consiste en ubicar el/los potenciales pedidos de cotización a partir de un pedido generado por un cliente. Es por esto que hace uso del paquete de integración con bases de datos, para recuperar instancias de proveedores y productos y para guardar los pedidos de cotización generados.

UserManager

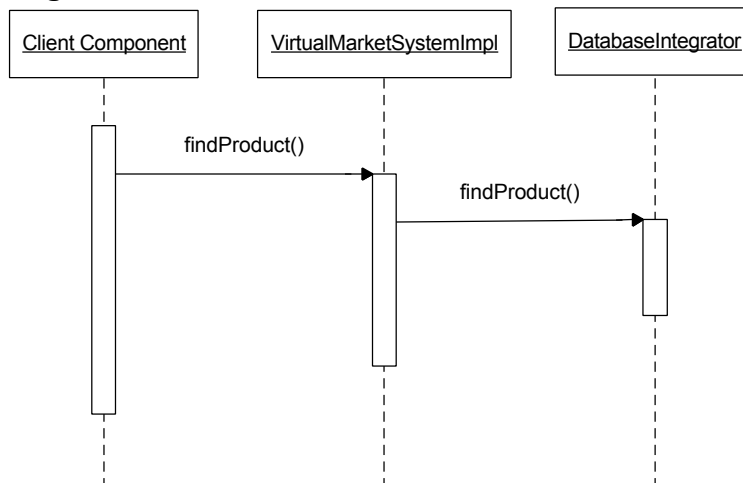
Esta clase tiene la responsabilidad de gestionar las cuestiones relacionadas con los usuarios. Desde este punto de vista, una de sus funciones principales es autenticarlos, es decir, dado un par (usuario, clave) identificar si corresponde a un usuario válido del sistema e instanciar la clase correspondiente (comprador o proveedor). En este caso dado que los usuarios se almacenan en la base de datos, este deberá utilizar al integrador de base de datos.

Diagramas de Interacción

Se pueden identificar claramente 3 escenarios de interacción dentro del componente servidor. Para comenzar, definimos esquemas genéricos de acceso a datos (petición de datos al servidor - consultas) y esquemas de funcionalidad específico. A continuación, se presentan los diagramas de interacción para los esquemas específicos y un diagrama genérico para el resto de los casos.

Para comenzar, el diagrama genérico que ejemplifica una consulta donde se busca un producto dado su código:

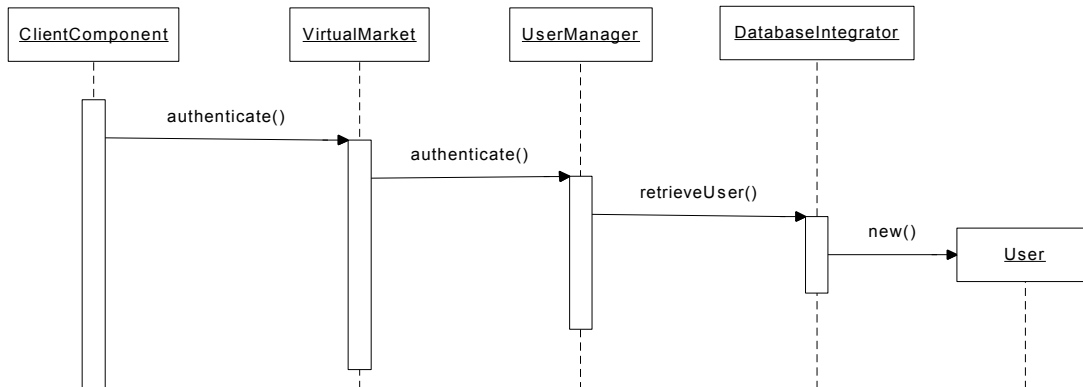
Diagrama de Interacción Genérico



Este esquema se repetirá para todos los métodos relacionados con búsquedas y actualizaciones de base de datos en el sistema (específicamente para todos los métodos exportados por el integrador de base de datos).

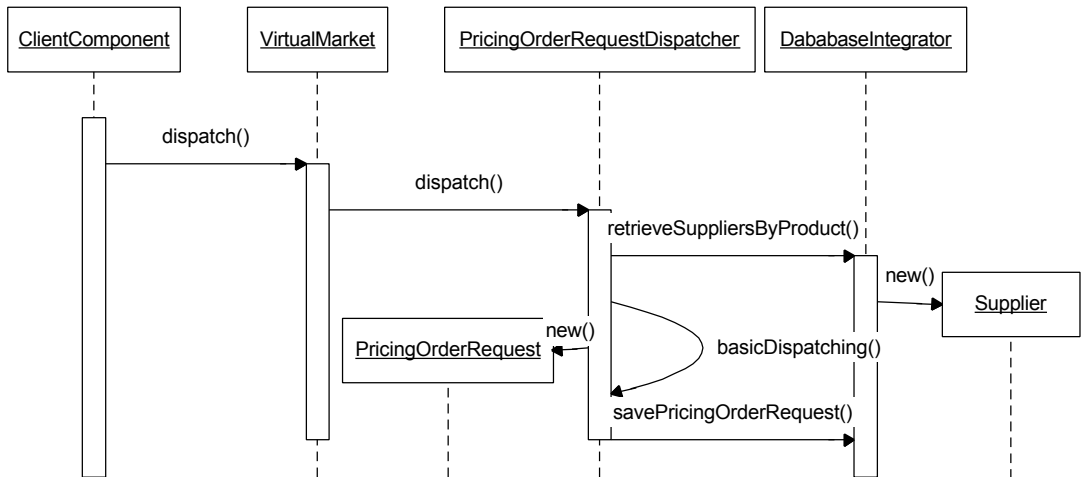
A continuación, se especifica el diagrama de interacción para la autenticación de usuarios.

Diagrama de Interacción para la Autenticación



En este diagrama puede verse claramente como el componente cliente (remoto) pide autenticar un usuario al VirtualMarketSystemImpl, quien delega la responsabilidad de autenticar usuarios al gestor de usuarios (UserManager). Este, a su vez dispara una transacción sobre la base de datos (a través del DatabaseIntegrator quien instancia un usuario de encontrarlo).

Diagrama de Interacción para la Asignación de Pedidos de Cotización



En estas figura puede verse el diagrama de secuencia para la asignación de pedidos de cotización. En este caso el componente cliente inicia una llamada remota al método para asignar los pedidos de cotización. Nuevamente este pedido es delegado en la clase designada a tal fin como ser el PricingOrderRequestDispatcher. Para ello debe pedir al DababaseIntegrator que obtenga los proveedores que puedan satisfacer por lo menos alguno de los productos que componen el pedido de cotización. Luego, se invoca a sí mismo para analizar para cada proveedor, qué puede proveer del pedido de cotización y de allí en más generar los pedidos de cotización para cada proveedor.

Luego, invoca nuevamente al DababaseIntegrator para que guarde dichos pedidos en la base de datos.

6.3.3. Identificación de Mecanismos Genéricos de Diseño

El objetivo de esta tarea es identificar y diseñar, en el caso de no existir en la instalación, esqueletos, patrones de diseño o guías de diseño. Estos mecanismos genéricos se definen a partir del estudio de comportamientos comunes relacionados, generalmente, con gestión de transacciones, persistencia de datos, control y recuperación de errores, utilización de recursos comunes, etc.

Los mecanismos genéricos de diseño son de aplicación tanto en la definición de la arquitectura del sistema como en el diseño de detalle de los subsistemas específicos y de soporte.

- Diseño de comunicaciones cliente / servidor utilizando SOAP
- Design Patterns [Gamma 1994]

6.4. Diseño de Casos de Uso Reales

Esta actividad tiene como propósito especificar el comportamiento del sistema de información para un caso de uso, mediante objetos o subsistemas de diseño que interactúan y determinar las operaciones de las clases e interfaces de los distintos subsistemas de diseño.

Para ello, una vez que se han identificado las clases participantes dentro de un caso de uso, se completan los escenarios que se recogen del análisis, incluyendo las clases de diseño que correspondan y teniendo en cuenta las restricciones del entorno tecnológico, esto es, detalles relacionados con la implementación del sistema. Es necesario analizar los comportamientos de excepción para dichos escenarios, algunos de ellos pueden haber sido identificados en el proceso de análisis aunque no se resuelven hasta este momento. Dichas excepciones se añadirán al catálogo de excepciones para facilitar las pruebas.

Algunos de los escenarios que se han detallado requerirán de nueva interfaz de usuario, por lo que es necesario diseñar el formato de cada una de las pantallas o impresos que se hayan identificado.

Es importante validar que los subsistemas que se han definido en la tarea Identificación de Subsistemas de Diseño (DSI 1.5), tienen la mínima interfaz con otros subsistemas. Por este motivo, se elaboran los escenarios a nivel de subsistemas y de esta forma se delimitan las interfaces necesarias para cada

uno de ellos, teniendo en cuenta toda la funcionalidad del sistema que recogen los casos de uso.

Además, durante esta actividad pueden surgir requisitos de implementación, que se recogen en el catálogo de requisitos.

Las tareas de esta actividad se realizan en paralelo con la actividad Diseño de Clases (DSI 4).

6.4.1. Identificación de Clases Asociadas a un Caso de Uso

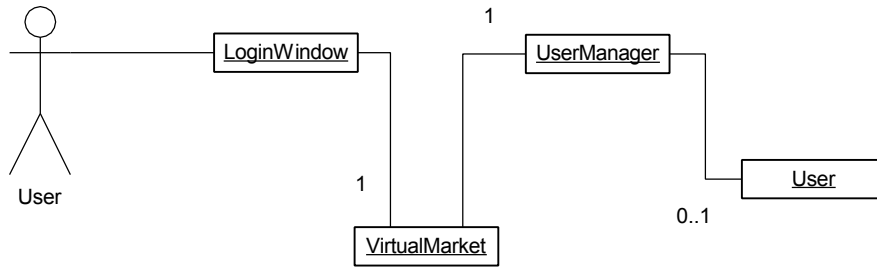
El objetivo de esta tarea es identificar, a partir del conjunto de clases que se han definido en la tarea Identificación de Clases Adicionales, las clases que intervienen en cada caso de uso. Estas clases se identifican a partir de las clases del modelo del análisis y de aquellas clases adicionales necesarias para el escenario que se está diseñando.

A su vez, a medida que se va estudiando la descripción de los casos de uso, pueden aparecer nuevas clases de diseño que no hayan sido identificadas anteriormente y que se incorporan al modelo de clases de la tarea Identificación de Clases Adicionales.

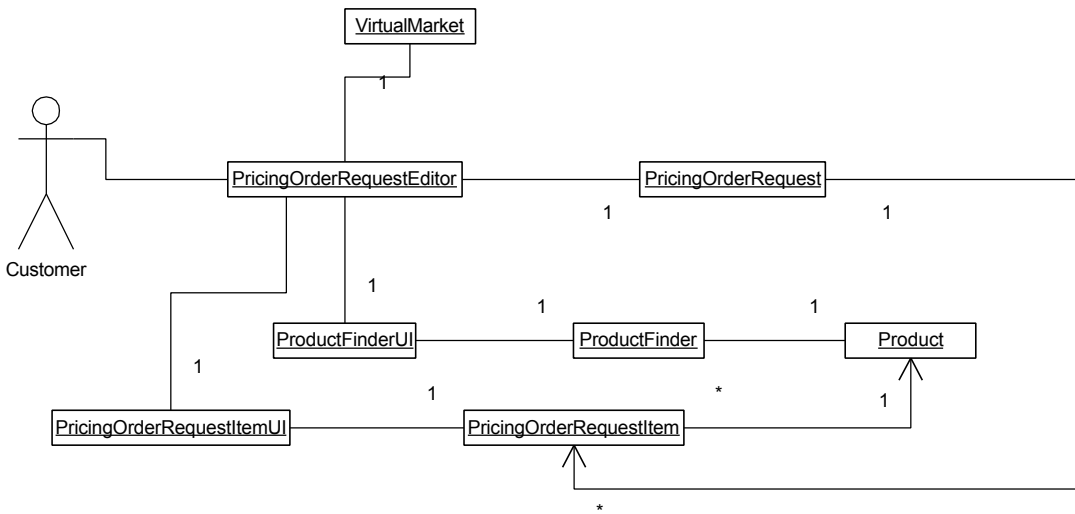
Si bien la metodología establecida en el Proceso Unificado establece utilizar todas las clases encontradas para todos los casos de uso y allí identificar las clases activas (active classes), dadas la cantidad de clases que se encuentran en este dominio, se opta por mostrar solamente las clases activas para cada caso de uso.

A continuación los diagramas de interacción para los distintos casos de uso involucrados.

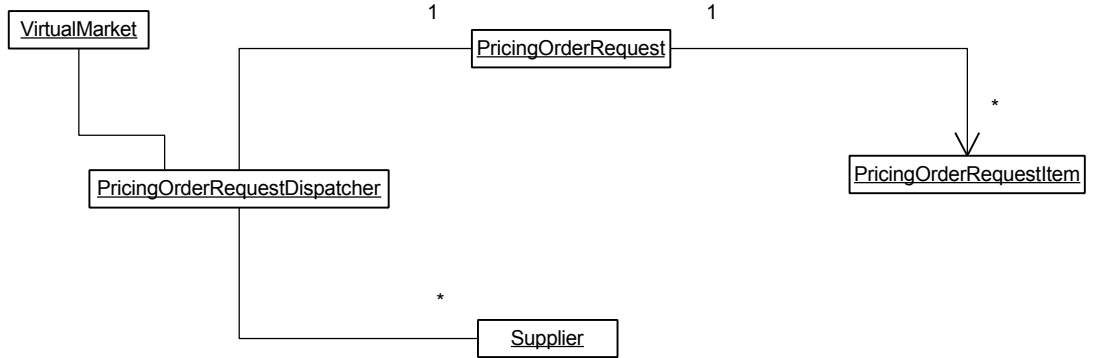
Ingreso al Sistema (CU01)



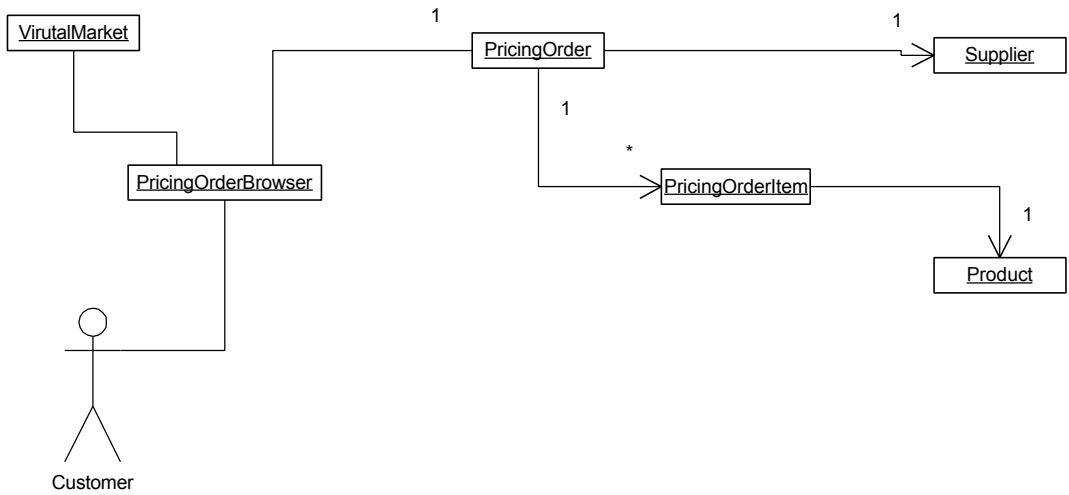
Carga del Pedido de Cotización (CU02)



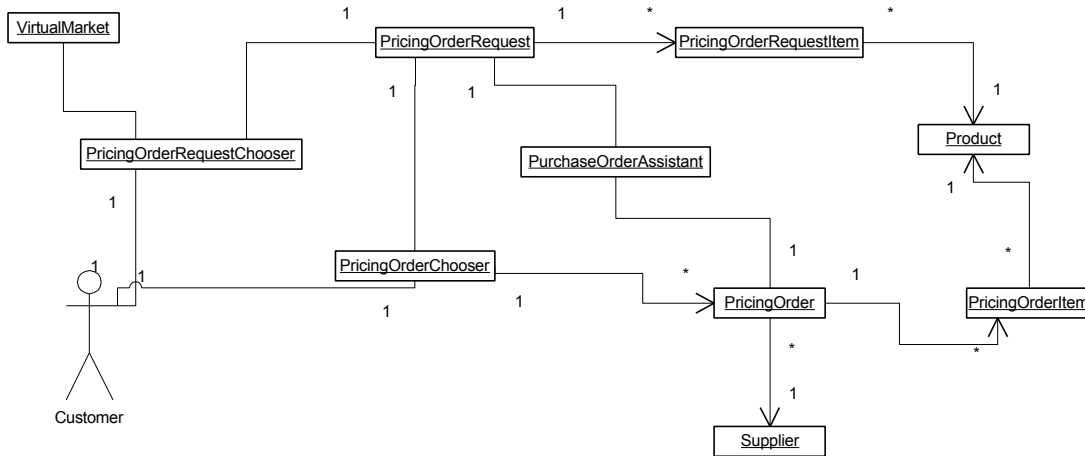
Colocar Pedido de Cotización por Proveedor (CU03)



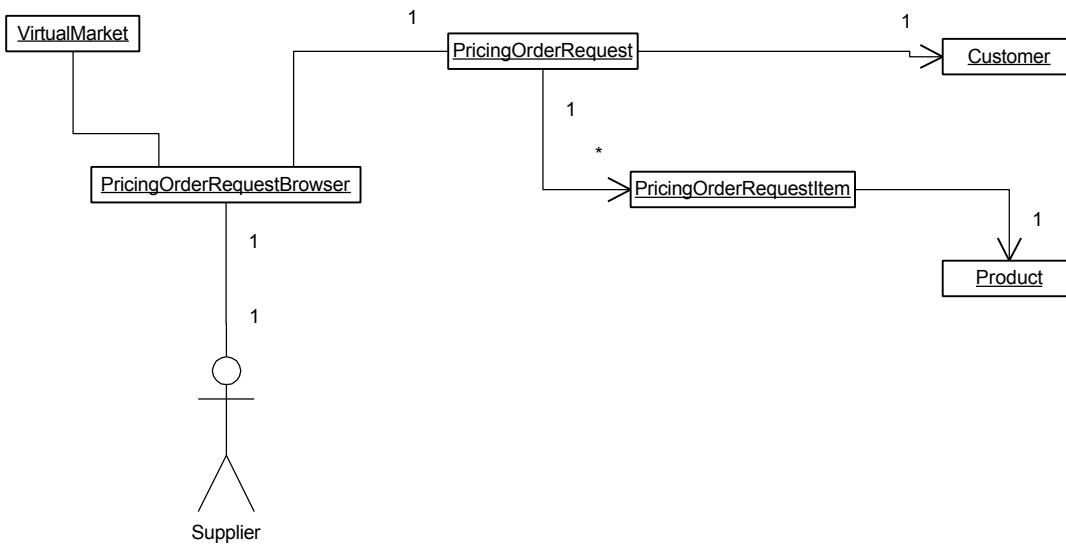
Consultar Cotizaciones (CU06)



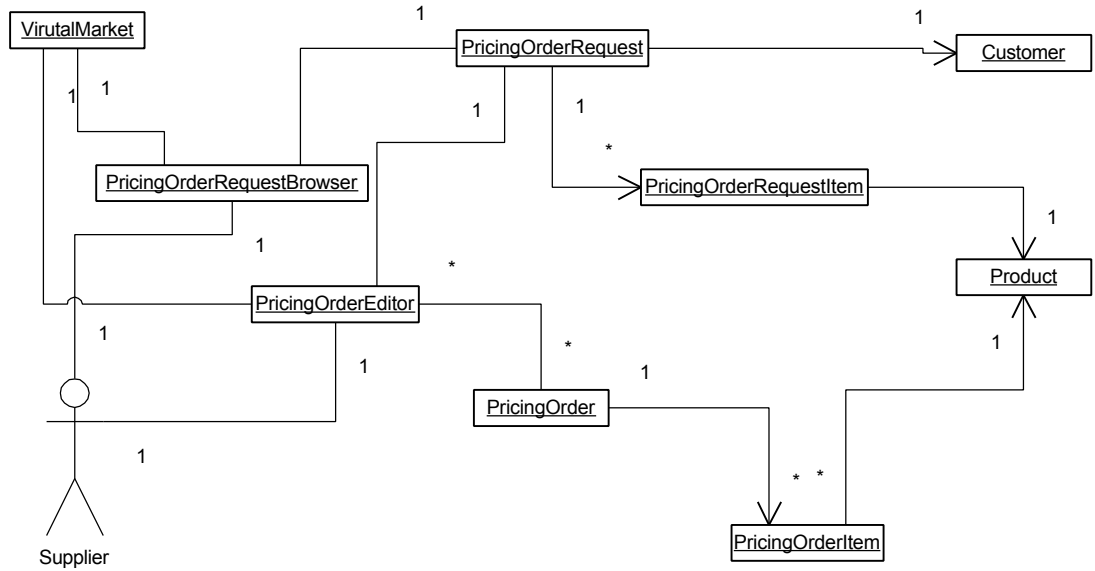
Confeccionar Orden de Compra (CU07)



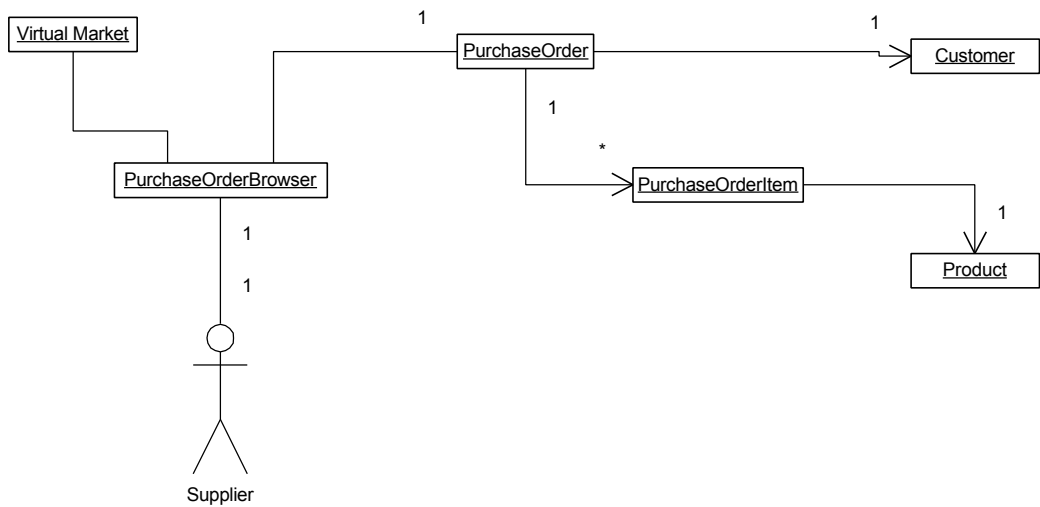
Consultar Pedidos de Cotización (CU04)



Cargar Cotización (CU05)



Consultar Ordenes de Compra (CU08)



6.4.2. Diseño de la Realización de los Casos de Uso

El objetivo de esta tarea es definir cómo interactúan entre sí los objetos identificados en la tarea anterior para realizar, desde un punto de vista técnico, un caso de uso del sistema de información. Para ello, se parte de los escenarios que se especificaron en el análisis y se detallan teniendo en cuenta que se deben llevar cabo sobre un entorno tecnológico concreto y unos mecanismos genéricos de diseño.

Durante el desarrollo de esta tarea, pueden surgir excepciones que se incluyen en el catálogo de excepciones, y que ahora quedan resueltas en los escenarios que corresponda.

Algunos de estos escenarios necesitarán nueva interfaz de usuario, por lo tanto, las clases de interfaz que se identifiquen se incorporan al modelo de clases de la tarea Identificación de Clases Adicionales (DSI 4.1), para realizar su diseño detallado.

También, se estudian los escenarios de los distintos casos de uso para identificar comportamientos comunes sobre los que se aplican mecanismos genéricos de diseño identificados en la tarea de Identificación de Mecanismos Genéricos de Diseño (DSI 2.2) o se puede decidir diseñar un subsistema de soporte que contenga dicho comportamiento, como un servicio.

El estudio de los comportamientos comunes identificados puede servir de ayuda al identificar o revisar la herencia entre clases en la tarea Diseño de la Jerarquía (DSI 4.5).

Como ya se ha visto en el apartado anterior, no aparecen nuevas clases de interfaz que deban ser tratadas en este punto. Aunque sí puede observarse que muchas de estas clases tienen un comportamiento común:

- Las clases definidas como "**chooser**" tienen la responsabilidad de mostrar al usuario un conjunto de instancias de objetos (TradeObjects) y permitirle seleccionar una de entre todas estas.
- Las clases identificadas como "**editor**" tienen la responsabilidad de editar uno de estos objetos.
- Las clases identificadas como "**browser**" tienen la responsabilidad de listar un conjunto de estos objetos.

En principio, podemos decir que hemos identificado comportamiento común en muchas clases de interfaz. Este comportamiento se verá reflejado en el modelo de clases por medio de "herencia" entre dichas clases. Este punto se tratará en detalle en el diseño de clases.

6.4.3. Revisión de la Interfaz de Usuario

El objetivo de esta tarea es realizar el diseño detallado del comportamiento de la interfaz de usuario a partir de la especificación de la misma que se obtuvo en el proceso de análisis y de acuerdo con el entorno tecnológico definido. Si se hubiera realizado un prototipo de la interfaz de usuario, éste se tomaría como punto de partida para el diseño.

Además se incluyen las ventanas alternativas o elementos de diseño que han aparecido nuevos como consecuencia del diseño de los escenarios definidos en la tarea anterior.

Además, se revisa: la interfaz de usuario, la navegación entre ventanas, los elementos que forman cada interfaz, sus características - que serán consistentes con los atributos con los que están relacionados -, su disposición, y cómo se gestionan los eventos relacionados con los objetos.

En aquellos casos en los que se realicen modificaciones significativas sobre la interfaz de usuario, será conveniente la validación de la misma por parte del usuario, siendo opcional la realización de un nuevo prototipo.

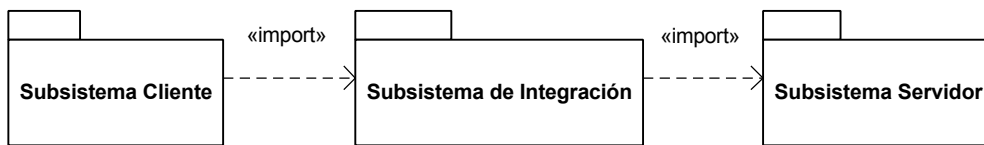
No existe mucho que agregar en este apartado, dado que si bien se han identificado mecanismos genéricos para utilizar en cuanto al diseño de las mismas (herencia) estas no han cambiado desde el punto de vista de la interfaz.

6.4.4. Revisión de Subsistemas de Diseño e Interfaces

El objetivo de esta tarea es describir cada caso de uso en términos de los subsistemas que participan en el caso de uso y las interfaces que se requieren entre ellos. Para un caso de uso se definen, además de los subsistemas y actores que intervienen en el mismo, los mensajes que intercambian los objetos de un subsistema con otro. Estos mensajes sirven para verificar y detallar las interfaces de cada subsistema, teniendo en cuenta todos los casos de uso en los que interviene, y completar de esta manera la definición de subsistemas establecida en la tarea Identificación de Subsistemas de Diseño

Como ya se ha tratado en apartados anteriores, los subsistemas detectados son el de integración con los componentes de servidor y el modelo del negocio. En todos los casos de uso se identifica el mismo patrón como puede presentarse en la figura a continuación:

Subsistemas de Diseño



6.5. Diseño de Clases

El propósito de esta actividad es transformar el modelo de clases lógico que proviene del análisis, en un modelo de clases de diseño. Dicho modelo recoge la especificación detallada de cada una de las clases, es decir sus atributos, operaciones, métodos, y el diseño preciso de las relaciones que se establecen entre ellas, bien sean de agregación, asociación o jerarquía. Para llevar a cabo todos estos puntos, se tendrán en cuenta las decisiones tomadas sobre el entorno tecnológico, y el entorno de desarrollo elegido para la implementación.

Se identifican las clases de diseño, denominadas clases adicionales en base al estudio de los escenarios de los casos de uso que se están realizando en paralelo, en la actividad Diseño de Casos de Uso Reales (DSI 3), y aplicando los mecanismos genéricos de diseño que se consideren convenientes por el tipo de especificaciones tecnológicas y de desarrollo. Entre ellas se encuentran clases abstractas, que integran características comunes con el objetivo de especializarlas en clases derivadas. Se diseñan las clases de interfaz de usuario, que provienen del análisis. Como consecuencia del estudio de los escenarios secundarios que se está realizando, puede aparecer nuevas clases de interfaz.

También hay que considerar, que por el diseño de las asociaciones y agregaciones, pueden aparecer nuevas clases, o desaparecer incluyendo sus atributos y métodos en otras, si se considera conveniente por temas de optimización. La jerarquía entre las clases se va estableciendo a lo largo de

esta actividad, a medida que se van identificando comportamientos comunes en las clases, aunque haya una tarea propia de diseño de la jerarquía.

Otro de los objetivos del diseño de las clases es identificar para cada clase los atributos, las operaciones que cubren las responsabilidades que se identificaron en el análisis, y la especificación de los métodos que implementan esas operaciones, analizando los escenarios del Diseño de Casos de Uso Reales (DSI 3). Se determina la visibilidad de los atributos y operaciones de cada clase, con respecto a las otras clases del modelo.

Una vez que se ha elaborado el modelo de clases se define la estructura física de los datos correspondiente a ese modelo, en la actividad Diseño Físico de Datos (DSI 6).

Además, en los casos en que fuera necesario una migración de datos de otros sistemas o una carga inicial de información, se realizará su especificación a partir del modelo de clases y las estructuras de datos de los sistemas origen.

Como resultado de todo lo anterior se actualiza el modelo de clases del análisis, una vez recogidas las decisiones de diseño.

Hay que destacar, que las actividades que se presentan a continuación, no tienen porqué desarrollarse secuencialmente. Al contrario, son más bien lineamientos a tener en cuenta en un sucesivo proceso de refinamiento iterativo sobre el modelo de objetos que componen la aplicación. Desde este punto de vista, se analizan las clases de objetos necesarias para soportar la interfaz y el dominio del sistema.

Para ello, se comienza por explicar en qué consisten las sub-actividades para terminar en el diseño detallado de las clases del modelo y el diseño detallado de las clases de interfaz. De tal forma, como resultado de esta fase se obtienen ambos diagramas de clases y una explicación de las cuestiones más significativas en ellos encontrados.

6.5.1. Identificación de Clases Adicionales

El objetivo de esta tarea es identificar un conjunto de clases que completen el modelo de clases analizado en el proceso anterior (clases y/o interfaces). Se tiene en cuenta que:

- Cada interfaz identificada en el análisis se corresponde en el diseño con una clase que proporcione esa interfaz.

- El conjunto de clases del análisis puede modificarse en función de las tecnologías de desarrollo utilizadas y de los mecanismos genéricos de diseño especificados.

Las clases de control deben contemplar la coordinación y secuencia entre objetos y en algunos casos contendrán lógica de negocio. En cualquiera de los casos se deben considerar cuestiones de distribución, de rendimiento, de transacción y de serialización. El diseño de las clases de entidad varía según el sistema de gestión de datos utilizado.

Las clases pueden ser construidas por el propio desarrollador, adquiridas en forma de bibliotecas, facilitadas por el entorno de trabajo o por el entorno tecnológico.

El diseño de las clases de interfaz de usuario depende de la tecnología específica que se esté utilizando. Así por ejemplo, la interfaz puede crearse a partir de los objetos gráficos disponibles en el entorno de desarrollo sin necesidad de que estos se contemplen en el modelo de clases correspondiente.

Entre las clases que se identifican a lo largo de esta tarea se encontrarán clases abstractas que reúnen características comunes a varias clases. Cada subclase aumentará su estructura y comportamiento con la clase abstracta de la que hereda.

Durante esta etapa se identifican entonces clases asociadas al diseño propiamente dicho. A continuación la lista de las clases encontradas y sus respectivas responsabilidades:

Clase	Responsabilidad
<i>PersistentObject</i>	La mayoría de los objetos de nuestra aplicación son persistentes. Esto significa que su ciclo de vida (en duración temporal) es mayor a la ejecución de los programas que la manejan. Esto se debe a que representan entidades del mundo real. Esta clase implementa el comportamiento común a esos objetos, como ser: poseer un identificador de base de datos, poder guardarse en una base de datos, etc.
<i>TradeObject</i>	El sistema en sí implementa un workflow de comprobantes de intercambio entre partes. Esta clase modela el comportamiento común de los objetos de intercambio.
<i>TradeObjectItem</i>	Los TradeObject, en nuestro caso, poseen ítems que los conforman. Dado que existen características específicas a cada tipo de ítem, cada uno tiene una representación en nuestro modelo de objetos. Esta clase modela las características comunes entre ellas.
<i>User</i>	Los usuarios pueden ser compradores o vendedores. De todas formas existen características comunes entre ellos, como ser la capacidad de usar el sistema, tener un usuario y password, etc. Es la responsabilidad de esta clase modelar dichas similitudes.
<i>Browser</i>	Es una clase abstracta de interfaz. Generaliza el comportamiento de todas aquellas interfaces de usuario que muestran una lista de datos.
<i>Chooser</i>	Otra clase abstracta de interfaz. Su responsabilidad es abstraer las características comunes de aquellas interfaces que permiten seleccionar un objeto a partir de una lista y realizar búsquedas para tal fin.
<i>Editor</i>	Generaliza el comportamiento de todos aquellos formularios utilizados para editar objetos del dominio.

6.5.2. Diseño de Asociaciones y Agregaciones

En esta tarea se completan las asociaciones entre las clases del modelo de clases del diseño, estudiando la secuencia de mensajes entre los objetos correspondientes en el diagrama de interacción de los escenarios definidos en la tarea Descripción de Interacciones entre Objetos (DSI 3.2).

Para definir las asociaciones partimos de las que fueron identificadas en proceso de análisis, teniendo en cuenta que:

- Se detallan las características de la asociación (papeles que desempeña, multiplicidad, etc.) según el entorno de desarrollo utilizado. Esta actividad ya ha sido tenida en cuenta y se refleja en el diagrama de clases consolidado del sistema.
- Las relaciones bidireccionales se transforman en unidireccionales, para simplificar la implementación del sistema. No se han encontrado relaciones bidireccionales, con lo cual, este punto se da por cumplido.
- Se modelizan rutas de acceso óptimas entre las asociaciones para evitar problemas de rendimiento. Si bien esta característica es de primordial importancia al hablar de sistemas productivos, no es tan así para este caso que se trata de un tema estrictamente académico. De todas formas, las rutas que se demuestran en los diagramas (por el símbolo de navegabilidad en UML) son óptimas.
- Se analiza la posibilidad de diseñar como clases algunas de las asociaciones. No se presenta el caso de necesitar clases para modelar relaciones, dado que las clases a utilizar para la implementación ya se encuentran desarrolladas y forman parte de la librería de clases del sistema (son triviales).

En este punto también se trata sobre como se implementarán las asociaciones. Para el presente desarrollo se pueden definir dos tipos de asociación a implementarse:

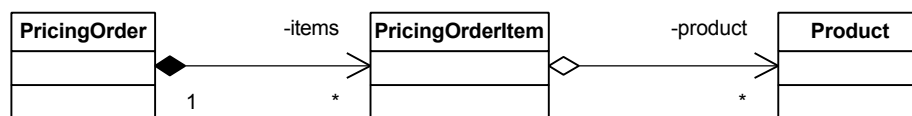
- Cuando la asociación de un objeto es contra uno de otra clase, la implementación es por medio a una referencia simple.
- Cuando la asociación es a 1 o más elementos, la misma se implementará utilizando la clase `java.util.Vector` del lenguaje Java, que permitirá mantener una colección de referencias a los objetos relacionados.

Todos los objetos del dominio son persistentes. Es decir, en algún momento de su ciclo de vida permanecen almacenados en una base de datos. Por ello, el hecho de recuperarlos no es trivial ya que indica un acceso a la base de datos. Por otro lado, es verdad que no tiene sentido que los mismos se carguen a memoria al cargar el objeto raíz (a partir del cual se

navegan) dado que esto implica un consumo indebido de memoria y recursos del equipo sin saber si efectivamente serán utilizados.

Para ello se plantea la siguiente estrategia de diseño, que se puede ver mejor con un ejemplo. Podemos extraer del modelo de clases el siguiente diagrama:

Diagrama de Estrategia de Asociaciones



En el pueden observarse dos relaciones:

- PricingOrder - PricingOrderItems: es una relación de composición. El pedido de cotización esta compuesto por ítems de un pedido de cotización. Según el apartado anterior esta asociación deberá ser implementada por medio de una colección de objetos.
- PricingOrderItem - Product: cada producto del pedido de cotización se encuentra compuesto aunque no exclusivamente de un producto del catálogo. Según el apartado anterior, esta asociación será implementada por una referencia directa.

Además, ambas asociaciones terminan en flecha. En la nomenclatura UML, esto significa que las asociaciones son navegables. Es decir, es posible pedirle a un ítem de un pedido de cotización su producto y este deberá devolverlo. En resumen, el diagrama muestra que en algún momento del código se puede llegar desde el pedido de cotización hasta el producto de cualquiera de sus ítems.

Ahora es el momento de diseñar como implementar estas relaciones que abundan en el modelo de objetos del presente proyecto.

El lenguaje de programación Java sugiere, entre otras normas de programación, que el acceso a las variables de instancia de un objeto debe realizarse siempre a través de un método de tipo setPropiedad (para asignar un valor) ó getPropiedad (para leer el valor de una propiedad).

Esta norma, es indispensable para implementar esta estrategia de implementación. Dado que un objeto que es navegable, muy probablemente no esté disponible en el momento en que se lo pida, sino más bien que la primer vez en ser pedido, este será cargado desde la base de datos. A este comportamiento, en la jerga de la orientación a objetos, se la conoce como *inicialización perezosa*.

La estrategia se basa entonces en:

- Al igual que en las bases de datos relacionales, los objetos que referencian a otros contendrán almacenada su clave primaria, pero como valor privado. La clave no deja de ser un objeto, con lo cual también contendrá métodos para asignar y leer su valor, aunque estos serán privados de la clase. Si suponemos una propiedad **product**, que es un objeto navegable (como en el ejemplo) tendremos una propiedad **productId** que contenga la clave foránea de la tabla donde se almacenan dichos objetos. Para simplificar el modelo en este proyecto todas las claves primarias serán tratadas como instancias de Integer (clase Java).
- La propiedad, en este caso **product**, será inicializada siempre con el valor nulo para indicar que la misma no está inicializada.
- Cuando se acceda a el método de lectura de la propiedad, nuevamente para el caso será **getProduct()** se verificará si la propiedad es nula y en ese caso se pedirá al objeto administrador que cargue la instancia para dicha clave primaria. Para el resto de las veces en que se solicite el producto, el mismo ya estará cargado en memoria. Un ejemplo en pseudocódigo se presenta a continuación:

```
Product getProduct {  
    si( producto es nulo )  
    entonces  
        producto = GestorDeInstancias.cargarInstancia(  
            getProductId() );  
    retornar producto;  
}
```

Este ejemplo se adecua a las relaciones n-1. Pero en el caso de las relaciones 1-n, como ser el del pedido de cotización con sus ítems, el parámetro que se envía al gestor de instancias es la clave primaria del extremo unario de la relación. El resultado de la ejecución de dicho método devolverá una colección de objetos.

El pseudocódigo:

```
Colección getItems {  
    si( items es nulo )  
    entonces  
        items = GestorDeInstancias.cargarInstancia(  
            this.getId() );  
    retornar items;  
}
```

Nótese como se utiliza la referencia "this" que es un puntero al objeto que implementa dicho método. De esa forma, dado que el mismo seguramente extenderá a PersistentObject, se recupera su clave primaria de forma tal de recuperar los ítems del pedido.

Este mismo patrón de diseño deberá ser utilizado para implementar todas las asociaciones navegables que aparecen en los diagramas de clases. De ahora en más se lo mencionará como "Patrón de Persistencia".

6.5.3. Identificación de Atributos de las Clases

El objetivo de esta tarea es identificar y describir, una vez que se ha especificado el entorno de desarrollo, los atributos de las clases.

Para identificar los atributos se revisa el modelo de clases del proceso anterior, considerando que, a partir de uno de ellos, puede ser necesario definir atributos adicionales. Para cada atributo identificado se define su tipo, con formatos específicos, y si existieran, las restricciones asociadas a ese atributo.

Como resultado de la aplicación del "Patrón de Persistencia", aparecen nuevos atributos de carácter implementativo en nuestro modelo de clases. Los mismos se definen a continuación:

<i>PersistentObject</i>		
Atributo	Tipo	Descripción
id	Integer	Clave primaria heredada por todos los objetos del dominio.

<i>TradeObject</i>		
Atributo	Tipo	Descripción
supplierId	Integer	Clave foránea del proveedor.
customerId	Integer	Clave foránea del cliente.

<i>TradeObjectItem</i>		
Atributo	Tipo	Descripción
productId	Integer	Clave foránea del producto al que hace referencia el ítem.
customerId	Integer	Clave foránea del cliente.

<i>PricingOrderRequestItem</i>		
Atributo	Tipo	Descripción
pricingOrderRequestId	Integer	Clave del pedido de cotización al que pertenece.

<i>PricingOrderResponsetItem</i>		
Atributo	Tipo	Descripción
pricingOrderResponseld	Integer	Clave de la cotización a la que pertenece.
pricingOrderRequestItemId	Integer	Clave del ítem del pedido de cotización al cual da respuesta este ítem de cotización.

<i>PurchaseOrderItem</i>		
Atributo	Tipo	Descripción
purchaseOrderId	Integer	Clave de la orden de compra a la que pertenece.
pricingOrderItemId	Integer	Clave del ítem de la cotización sobre la cual se efectuó este ítem de la orden de compra.

Nótese que muchas de estas clases tienen una relación de herencia entre sí, lo cual hace entrever que mucho comportamiento de persistencia será heredado también.

6.5.4. Identificación de Operaciones de las Clases

El objetivo de esta tarea es definir de forma detallada las operaciones de cada clase de diseño. Para ello, se toma como punto de partida el modelo de clases generado en el análisis, así como el diseño de los casos de uso reales y los requisitos de diseño que pueden aparecer al definir el entorno de desarrollo.

Las operaciones de las clases de diseño surgen para dar respuesta a las responsabilidades de las clases de análisis y además para definir las interfaces que ofrece esa clase.

Según el entorno de desarrollo utilizado, se describe cada operación especificando: su nombre, parámetros y visibilidad (pública, privada, protegida). Si el entorno de desarrollo lo permite, se tiene en cuenta la

posibilidad de simplificar el modelo de clases haciendo uso del polimorfismo y la sobrecarga de operaciones.

Para identificar las operaciones de aquellos objetos que presenten distintos estados, por lo que su comportamiento depende del estado en el que se encuentren, es recomendable realizar un diagrama de transición de estados, y traducir cada acción o actividad del mismo en una de estas operaciones.

En el presente desarrollo, no se encuentran objetos cuyo comportamiento cambien en función de su estado. Por lo cual no se desarrollarán diagramas de transición de estados.

Para comenzar con el análisis de los métodos de los objetos, hay que tener en cuenta una distinción:

- **Métodos triviales:** la especificación de Sun Microsystems para trabajar con el lenguaje de programación Java, define que deben existir ciertos métodos (set y get) que se utilizan para leer y escribir los valores de los atributos de los objetos. Dado que estos métodos son simples y serán generados automáticamente por la herramienta de desarrollo, no se documentarán dichos métodos de forma tal de facilitar el entendimiento del modelo.
- **Métodos no triviales:** son aquellos métodos que agregan valor al sistema, es decir, modelan reglas de negocio y comportamiento de las clases del sistema. Estos métodos sí serán diseñados y explicados a continuación.

<i>PersistentObject</i>		
isPersistent	público	Retorna verdadero en caso de que el receptor sea un objeto ya guardado en una base de datos. De lo contrario retorna falso.
	Parámetros	----
	Resultado	boolean
save	público	Guarda el objeto en la base de datos.
	Parámetros	----

	Resultado	----
--	------------------	------

<i>TradeObject</i>		
addItem	público	Agrega un ítem al documento.
	Parámetros	IN - TradeObjectItem
	Resultado	----
deleteItem	público	Borra un ítem del documento.
	Parámetros	IN - TradeObjectItem
	Resultado	----
items	público	Devuelve un iterador sobre los ítems del documento.
	Parámetros	----
	Resultado	Enumeration

<i>UserManager</i>		
authenticate	público	Autentica un usuario del sistema
	Parámetros	IN - String - userName IN - String - password
	Resultado	User
<i>PricingOrderRequestDispatcher</i>		
dispatch	público	Genera los pedidos de cotización necesarios a partir de uno genérico
	Parámetros	IN - PricingOrderRequest - template
	Resultado	void

<i>ProductFinder</i>		
findProductByCode	público	Busca un producto en el catálogo en base a su código de producto.
	Parámetros	IN - Integer - productCode
	Resultado	Product
findProductByDescription	público	Busca un producto en el catálogo en base a una cadena de caracteres a buscar en la descripción de los productos.
	Parámetros	IN - String - wildcard
	Resultado	Vector

<i>PurchaseOrderCreationAssistant</i>		
start	público	Comienza el ciclo de creación de las órdenes de compra.
	Parámetros	IN - PricingOrderRequest - baseOrder
	Resultado	void

<i>VirtualMarket</i>		
dispatch	público	Invoca a PricingOrderRequestDispatcher >> dispatch.
	Parámetros	IN - PricingOrderRequest
	Resultado	----
authenticate	público	Autentica un usuario del sistema
	Parámetros	IN - String - userName IN - String - password
	Resultado	User
insertPricingOrderRequest	privado	Inserta un pedido de cotización en la base de datos.
	Parámetros	IN - PricingOrderRequest
	Resultado	----

insertPricingOrderResponse	público	Inserta una cotización en la base de datos.
	Parámetros	IN - PricingOrderResponse
	Resultado	----
insertPurchaseOrder	público	Inserta una orden de compra en la base de datos.
	Parámetros	IN - PurchaseOrder
	Resultado	----
retrieveProduct	público	Recupera un producto del catálogo en base a su clave primaria.
	Parámetros	IN - Integer
	Resultado	Product
retrievePricingOrderRequests	público	Recupera los pedidos de cotización para un proveedor dado.
	Parámetros	IN - Integer - supplierId
	Resultado	Vector
retrievePricingOrderResponses	público	Recupera las cotizaciones realizadas para un pedido de cotización dado.
	Parámetros	IN - Integer - pricingOrderId
	Resultado	Vector
retrievePurchaseOrders	público	Recupera las órdenes de compra para un proveedor dado.
	Parámetros	IN - Integer - supplierId
	Resultado	Vector
RetrievePossibleSuppliersFor	Público	Recupera los posibles proveedores de un pedido de cotización (aquellos que proveen al menos alguno de los productos a cotizar).
	Parámetros	IN – PricingOrderRequest – order
	Resultado	Vector

Browser		
browse	público	Muestra una lista de TradeObjects.

	Parámetros	IN - Vector - tradeObjectsToBrowse
	Resultado	----

<i>Editor</i>		
edit	público	Abre una pantalla que permite editar el objeto recibido como parámetro.
	Parámetros	IN - TradeObject - aTradeObject
	Resultado	----

<i>Chooser</i>		
choose	público	Abre una pantalla y permite buscar un elemento en la base de datos.
	Parámetros	----
	Resultado	TradeObject
<i>LoginUI</i>		
login	público	Abre una pantalla que pida usuario y clave para autenticarlo.
	Parámetros	IN - String - userName IN - String - password
	Resultado	User

6.5.5. Diseño de la Jerarquía

El objetivo de esta tarea es revisar la jerarquía de clases que ha surgido en el modelo de clases a lo largo de las tareas anteriores y comprobar que esa jerarquía es viable según los mecanismos disponibles en el entorno de desarrollo utilizado.

Entre las modificaciones realizadas sobre la jerarquía se identifican clases abstractas que son superclases en las que se agrupan atributos y operaciones que se heredan por sus subclases.

6.5.6. Descripción de Métodos de las Operaciones

En esta tarea se describen los métodos que se usarán para detallar como se realizan cada una de las operaciones de una clase. Los métodos pueden especificarse mediante un algoritmo, usando pseudocódigo o lenguaje natural, y su implementación se basa en la secuencia de interacciones del diagrama de interacción en los que la clase aparezca o en la secuencia de transiciones del diagrama de transición de estados.

En la mayoría de los casos esta tarea no suele realizarse hasta el proceso de construcción, en el que se describen directamente en el lenguaje de programación que se vaya a utilizar.

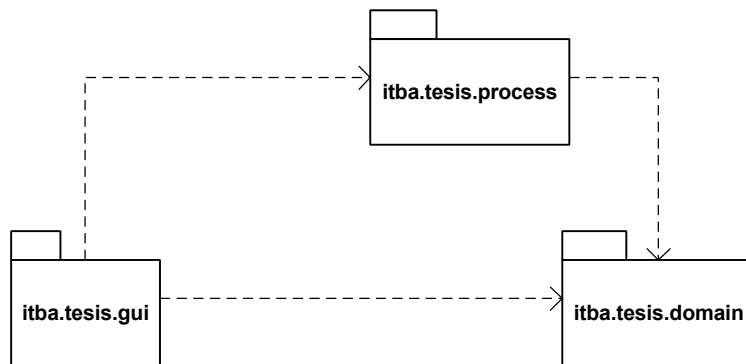
6.5.7. Diagrama de Clases del Dominio del Sistema

Antes de indagar sobre las clases que componen el sistema, es necesario mencionar el concepto de paquetes. Tanto la metodología UML como el lenguaje de programación Java (elegido para la implementación del presente proyecto) manejan el paralelismo conceptual de paquetes. Los paquetes son una forma de agrupar clases. Si bien físicamente no tienen porque conservar paralelismo alguno con cuestiones físicas de librería, sí intentan agrupar clases en base a criterios de cohesión o acoplamiento conceptual de las mismas. Para el presente proyecto, y dado que este es el diseño detallado, se divide el sistema en 3 paquetes principales:

Paquete	Descripción
Clases del Dominio (itba.tesis.domain)	Estas clases dominan fuertemente el dominio de negocio. Cada clase se corresponde con un concepto del negocio que es necesario modelar. Las relaciones entre estas clases, modelan las relaciones, restricciones y reglas de negocios.
Clases de Proceso (itba.tesis.process)	Estas clases modelan procesos dentro del dominio. Por lo general, estos procesos son de una importancia o complejidad tal que valen la pena su diseño e implementación fuera de una clase del dominio.
Clases de Interfaz Gráfica de Usuario (itba.tesis.gui)	Estas clases modelan la interfaz gráfica de usuarios. Es importante tener en cuenta el patrón de diseño Model-View-Controller para darse cuenta que el dominio de negocios debe estar completamente separado de los objetos de interfaz y los objetos que adaptan la interfaz a dichos objetos, tanto como la inversa.

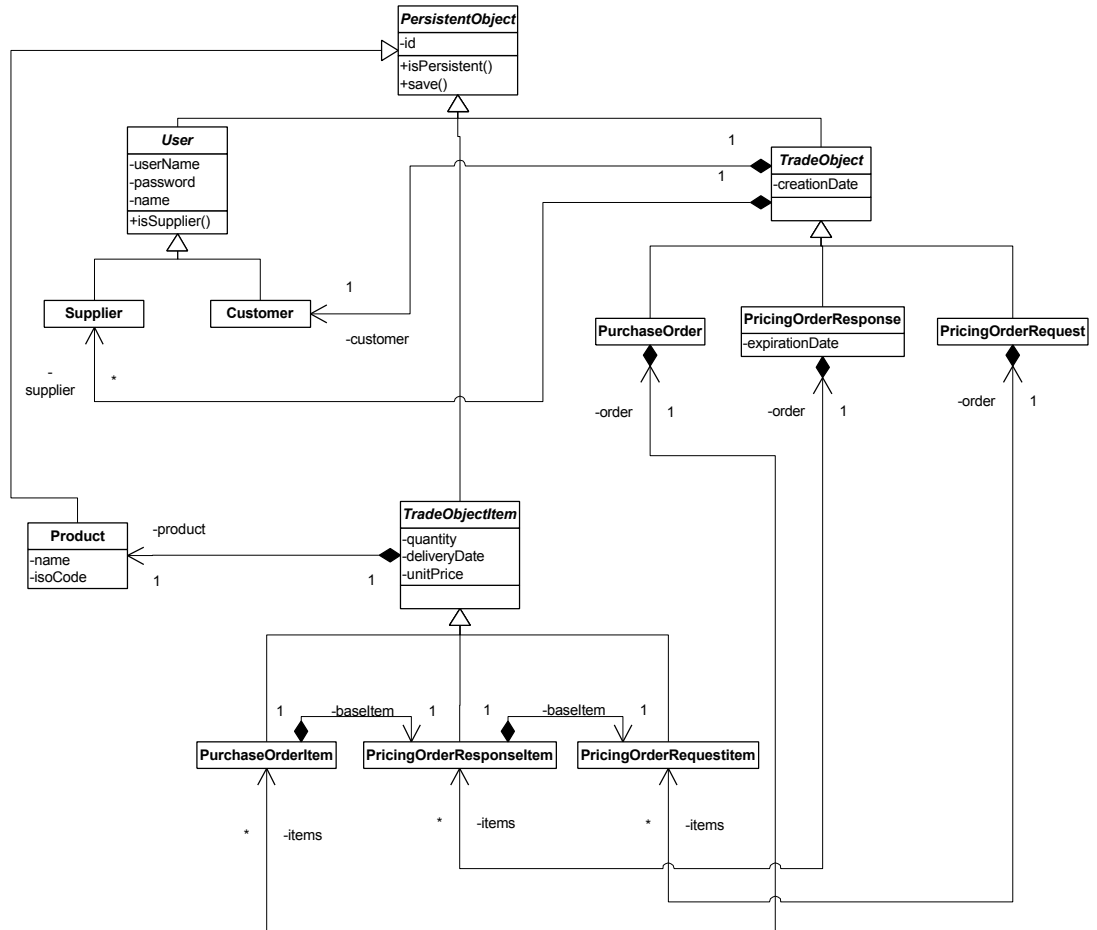
Si se establece el estereotipo de relación "import" (propia del lenguaje Java) como la necesidad de un paquete de importar otro para usar sus clases (dependencias) podemos ver el diagrama a continuación que la refleja en nomenclatura UML:

Paquetes del Sistema



Luego, a continuación, se explican los paquetes en detalle y las decisiones de diseño tomadas para cada uno de ellos.

Diagrama de Clases del Dominio del Sistema
(itba.tesis.domain)



En una primer instancia es necesario lo sucedido con los usuarios. Si bien estos pueden ser de dos tipos (cliente y proveedor), para el sistema resulta importante tratarlos como pertenecientes a una única clase (User).

Ambos poseen un nombre de usuario y clave para acceder al sistema, como así una razón social. Por lo tanto, se obtiene una primer jerarquía de usuarios.

Algo similar ocurre con los comprobantes que se deben manejar en el sistema. Todos ellos tienen cierto tipo de datos en común como ser la fecha de emisión, y ciertas relaciones comunes: todos ellos relacionan a un cliente con un proveedor. Si bien el significado de la relación varía dependiendo del documento, esto es una característica propia de cada clase y será el polimorfismo en encargado de administrarla. De esta forma se identifica una

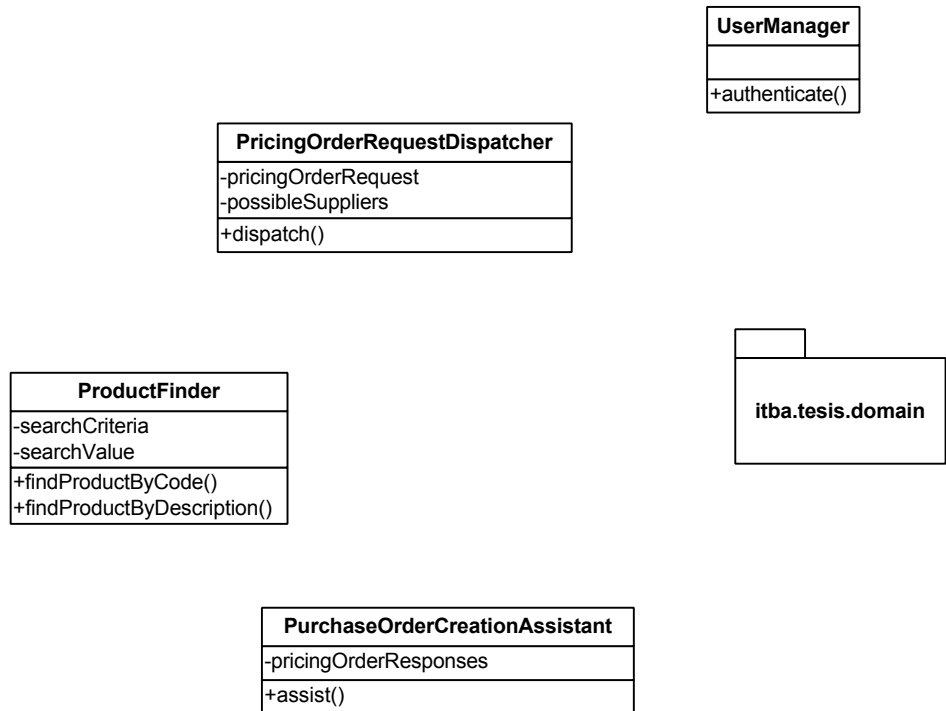
nueva clase abstracta "TradeObject" (objeto de intercambio) que contiene las características comunes a todos ellos. Por otro lado, se define una subclase para cada tipo de documento. Como se observa en el diagrama, si bien todos los documentos tienen ítems, cada uno de ellos tiene una relación de composición con un tipo específico de ítem.

Los ítems se encuentran agrupados también bajo una súper clase "TradeObjectItem". todos los ítems de los distintos documentos están compuestos por un producto y para el cual especifican una fecha de entrega, un precio unitario y una cantidad solicitada. Pero, se pueden observar otras relaciones. Las respuestas a los pedidos de cotización, se basan en un ítem del pedido de cotización (la toman como base). Por otro lado, los ítems de las órdenes de compra se basan en una respuesta a un pedido de cotización. Esta es la base de la justificación de la información mantenida por el sistema y queda establecida en el modelo de objetos por medio de dichas relaciones de diseño.

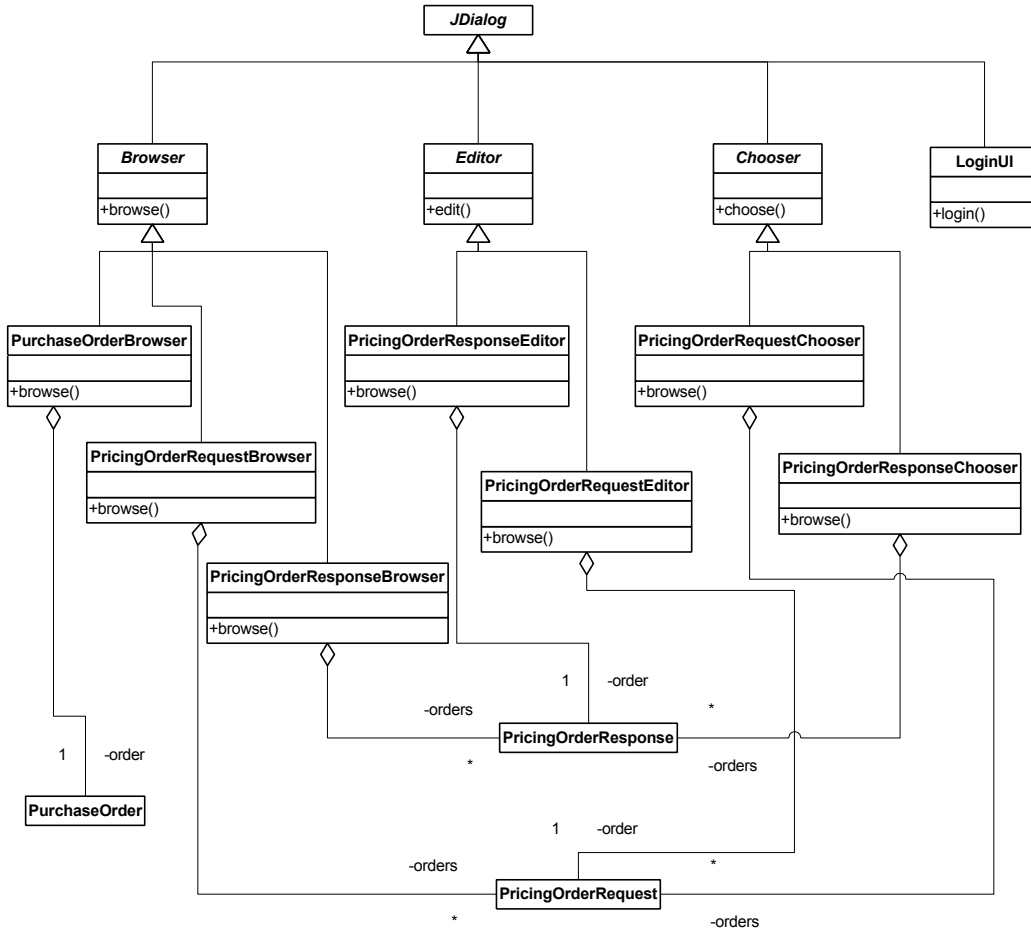
En algún momento, todos los objetos del modelo se vuelven persistentes. Esto significa que su ciclo de vida sobrevive al de la ejecución de los componentes del sistema. Por lo tanto, estos deben ser almacenados en algún dispositivo que asegure su supervivencia. La clase abstracta "PersistentObject" modela estas características para todos los objetos: la posesión de un identificador para su posterior relación y recupero (id), la posibilidad de preguntarle a cualquier objeto si es persistente (saber si ya se encuentra almacenado en algún medio de soporte) y la posibilidad de guardar los cambios sobre ellos hechos.

Si bien aquellos métodos heredados de clases abstractas deben ser obligatoriamente implementados por todas sus clases concretas, esto no se muestra en el diagrama para no aumentar la complejidad de entendimiento del mismo. Así también con los métodos necesarios para setear y obtener los valores de las variables de instancia (privadas) de los objetos.

Diagrama de Clases de Procesos (itba.tesis.process)



**Diagrama de Clases de Interfaz Gráfica de Usuario
(itba.tesis.gui)**



6.5.8. Resumen de Diferencias de Clases de Diseño y Análisis

En este apartado se presenta una tabla comparativa que detalla las clases que fueron agregadas durante el refinamiento en la etapa de diseño del sistema.

Las clases que figuran en ambas columnas, demuestran la trazabilidad de las mismas desde el análisis al diseño del sistema de información. Aquellas que figuran solamente en la etapa de diseño, son clases nuevas que aparecen en este momento del desarrollo.

Análisis	Diseño
Ingreso al Sistema	LoginUI
	UserManager
Usuario	User
Comprador	Customer
Editor de Pedido de Cotización	PricingOrderRequestEditor
Editor de Item de Pedido de Cotización	PricingOrderRequestItemEditor
Buscador de Productos	ProductFinderBrowser
Pedido de Cotización	PricingOrderRequest
Item del Pedido de Cotización	PricingOrderRequestItem
Producto	Product
	ProductFinder
Proveedor	Supplier
¿?	PricingOrderRequestDispatcher
Cotización	PricingOrderResponse
Item de la Cotización	PricingOrderResponseItem
Consulta de Cotizaciones	PricingOrderResponseBrowser
Selector de Cotizaciones	PricingOrderResponseChooser
Editor de Orden de Compra	PurchaseOrderEditor
Editor de Item de Orden de Compra	PurchaseOrderItemEditor
Orden de Compra	PurchaseOrder
Item de la Orden de Compra	PurchaseOrderItem
Consulta de Pedidos de Cotización	PricingOrderRequestBrowser
Consulta de Ordenes de Compra	PurchaseOrderBrowser
Editor de Cotización	PricingOrderResponseEditor
Editor de un Item de una Cotización	PricingOrderResponseItemEditor
	Editor
	Coser
	Browser
	VirtualMarket

	PersistentObject
	TradeObject
	TradeObjectItem

6.6. Diseño Físico de Datos

En esta actividad se define la estructura física de datos que utilizará el sistema, a partir del modelo de clases, de manera que, teniendo presente las características específicas del sistema de gestión de datos, concreto a utilizar, los requisitos funcionales y no funcionales establecidos para el sistema de información y las particularidades del entorno tecnológico, se consiga una mayor eficiencia en el tratamiento de los datos.

También, se analizarán los caminos de acceso a los datos, tanto de consulta como de actualización, que realiza cada clase a los datos persistentes del sistema, con el fin de mejorar los tiempos de respuesta y optimizar los recursos de máquina.

Las tareas de esta actividad se realizan de forma iterativa y en paralelo con las realizadas en las actividades Definición de la Arquitectura del Sistema (DSI 1), dónde se especifican los detalles de arquitectura e infraestructura y la planificación de capacidades, Diseño de la Arquitectura de Soporte (DSI 2), dónde se especifican y diseñan los servicios comunes que pueden estar relacionados con la gestión de datos (acceso a bases de datos, ficheros, áreas temporales, sincronización de bases de datos, etc.) y Diseño de Casos de Uso Reales y de Clases (DSI 3 y 4) dónde se especifica la lógica de tratamiento y las interfaces utilizadas.

En el caso de diseño orientado a objetos, esta actividad también es necesaria y la obtención del modelo físico de datos se realiza aplicando una serie de reglas de transformación a cada elemento del modelo de clases que se está generando en la actividad Diseño de Clases (DSI 4).

Asimismo, en esta actividad se considerarán los estándares y normas establecidas para el diseño, aplicando, cuando proceda, los mecanismos genéricos de diseño identificados en la tarea Identificación de Mecanismos Genéricos de Diseño (DSI 2.2).

6.6.1. Diseño del Modelo Físico de Datos

El objetivo de esta tarea es realizar el diseño físico de datos a partir del modelo de clases.

Como paso previo al diseño de la estructura física de datos, se analizan las peculiaridades técnicas del gestor de bases de datos o sistema de ficheros a utilizar y las estimaciones sobre la utilización y volumen de las ocurrencias de cada clase del modelo lógico de datos normalizado o modelo de clases. Además, si se ha establecido la necesidad de llevar a cabo una migración de datos, se tendrán en cuenta, también los volúmenes de las estructuras de datos implicadas en la conversión. Esta información servirá para decidir la mejor implementación del modelo de clases, así como para hacer una estimación del espacio de almacenamiento.

De acuerdo al análisis anterior, se determina cómo se van a convertir las clases en tablas considerando las relaciones existentes entre ellas, y los identificadores, definiendo sus claves primarias, ajenas, alternativas u otros medios de acceso en general.

Si bien en este punto también se consideran aspectos como bloqueo, compresión de datos, agrupamientos, etc. los mismos no son necesarios para el presente proyecto, con lo cual su análisis queda excluido.

Luego, para el presente proyecto se han tenido en cuenta las reglas de generación de tablas para la persistencia de objetos del libro "Object-Oriented Modeling and Design for Database Applications".

Las tablas a generar, son las siguientes:

<i>users</i>		
Campo	Tipo	Condicionantes
usr_id	INTEGER(10)	clave primaria
usr_name	VARCHAR(30)	
usr_login	VARCHAR(10)	
usr_password	VARCHAR(200)	Suficientemente grande como para soportar claves encriptadas y criptogramas
usr_mail	VARCHAR(50)	
usr_type	ENUM("Proveedor", "Comprador")	

<i>products</i>		
Campo	Tipo	Condicionantes
prod_id	INTEGER(10)	clave primaria
prod_desc	VARCHAR(30)	
prod_code	VARCHAR(10)	

<i>prod_x_sup</i>		
Campo	Tipo	Condicionantes
prod_id	INTEGER(10)	clave foránea
usr_id	INTEGER(10)	clave foránea

<i>pricing_order_requests</i>		
Campo	Tipo	Condicionantes
po_id	INTEGER(10)	clave primaria
po_date	DATE	
pur_id	INTEGER(10)	clave foránea
sup_id	INTEGER(10)	clave foránea
po_status	ENUM("OPEN", "CLOSED")	

<i>pricing_order_request_items</i>		
Campo	Tipo	Condicionantes
poi_id	INTEGER(10)	clave primaria
po_id	INTEGER(10)	clave foránea
prod_id	INTEGER(10)	clave foránea
poi_quantity	NUMERIC (10,4)	
poi_receival_date	DATE	
poi_unit_price	NUMERIC (10,4)	

<i>pricing_order_responses</i>		
Campo	Tipo	Condicionantes
por_id	INTEGER(10)	clave primaria
por_date	DATE	
sup_id	INTEGER(10)	clave foránea
pur_id	INTEGER(10)	clave foránea
po_id	INTEGER(10)	
por_expiration_date	DATE	

<i>pricing_order_response_items</i>		
Campo	Tipo	Condicionantes
pori_id	INTEGER(10)	clave primaria
por_id	INTEGER(10)	clave foránea
prod_id	INTEGER(10)	clave foránea
pori_quantity	NUMERIC (10,4)	
pori_receival_date	DATE	
pori_unit_price	NUMERIC(10,2)	

<i>purchase_orders</i>		
Campo	Tipo	Condicionantes
puro_id	INTEGER(10)	clave primaria
puro_date	DATE	
pur_id	INTEGER(10)	clave foránea
sup_id	INTEGER(10)	clave foránea

<i>purchase_order_items</i>		
Campo	Tipo	Condicionantes
puroi_id	INTEGER(10)	clave primaria
puro_id	INTEGER(10)	clave foránea
prod_id	INTEGER(10)	clave foránea
pori_quantity	NUMERIC(10,4)	
pori_receival_date	DATE	
pori_unit_price	NUMERIC(10,2)	

Luego, para la creación de dichas tablas en el gestor de bases de datos MySQL, serán necesarios los siguientes scripts:

```
// Tabla de productos
```

```
create table products(  
    prod_id INTEGER(10) primary key auto_increment,  
    prod_desc varchar(50),  
    prod_IEEE varchar(10)  
);
```

```
// Tabla de usuarios del sistema
```

```
create table users (  
    usr_id INTEGER(10) primary key auto_increment,  
    usr_name VARCHAR(30),  
    usr_login VARCHAR(10),  
    usr_mail VARCHAR(50),  
    usr_password VARCHAR(200), // grande para  
soportar criptograma  
    usr_type ENUM( "Proveedor", "Comprador" )  
);
```


// Tabla de relación entre productos y proveedores

```
create table prod_x_sup (  
    prod_id INTEGER(10),  
    usr_id INTEGER(10)  
);
```

// Tabla de pedidos de cotizaciones

```
create table pricing_order_requests (  
    po_id          INTEGER(10)          primary          key  
auto_increment,  
    po_date        DATE,  
    pur_id         INTEGER(10),  
    sup_id         INTEGER(10),  
    po_status      ENUM( "OPEN", "CLOSED" )  
);
```

// Tabla de ítems de pedidos de cotización

```
create table pricing_order_request_items (  
    poi_id         INTEGER(10)          primary          key  
auto_increment,  
    po_id          INTEGER(10),  
    prod_id        INTEGER(10),  
    poi_quantity   NUMERIC(10,4),  
    poi_receival_date DATE,  
    poi_unit_price NUMERIC(10,4)  
);
```

// Tabla de cotizaciones

```
create table pricing_order_responses (  
    por_id    INTEGER(10)          primary          key  
auto_increment,  
    por_date  DATE,  
    sup_id    INTEGER(10),  
    pur_id    INTEGER(10),  
    po_id     INTEGER(10),  
    por_expiration_date DATE  
);
```

// tabla de ítems de una cotización

```
create table pricing_order_response_items (  
    pori_id   INTEGER(10)          primary          key  
auto_increment,  
    por_id    INTEGER(10),  
    prod_id   INTEGER(10),  
    pori_quantity NUMERIC(10,4),  
    pori_receival_date DATE,  
    pori_unit_price NUMERIC(10,2),  
    poi_id    INTEGER(10)  
);
```

```
// tabla de órdenes de compra
create table purchase_orders (
    puro_id    INTEGER(10)          primary          key
auto_increment,
    puro_date  DATE,
    pur_id     INTEGER(10),
    sup_id     INTEGER(10)
);

// tabla de ítems de las órdenes de compra
create table purchase_order_items (
    puroi_id   INTEGER(10)          primary          key
auto_increment,
    puro_id    INTEGER(10),
    prod_id    INTEGER(10),
    puroi_quantity NUMERIC(10,4),
    puroi_receival_date DATE,
    puroi_unit_price NUMERIC(10,2),
    pori_id    INTEGER(10)
);
```

6.6.2. Optimización del Modelo Físico de Datos

En esta tarea se optimiza el diseño físico de datos, con el objetivo de mejorar el tiempo de respuesta en el acceso a datos persistentes, optimizar los recursos del sistema y, en consecuencia, garantizar que el diseño satisface las necesidades de tratamiento establecidas para el sistema de información en cuanto a que se ajusta a los requisitos de rendimiento exigidos.

A partir de la especificación de la secuencia de accesos, obtenida en la tarea anterior, de aquellas clases identificadas como críticas, se detectan las posibles mejoras con el fin de conseguir los niveles de rendimiento

establecidos y por lo tanto, una mayor eficiencia del sistema. Como resultado puede ser necesaria una desnormalización controlada que se aplica para reducir o simplificar el número de accesos a los sistemas de almacenamiento de datos.

Para el presente proyecto no será necesario modificar el modelo físico, dado que la base de datos obtenida en el paso anterior, es el resultado del proceso de creación de bases de datos relacionales para un modelo de objetos dado. Este modelo ya es óptimo para dicho esquema.

6.7. Verificación y Aceptación de la Arquitectura del Sistema

El objetivo de esta actividad es garantizar la calidad de las especificaciones del diseño del sistema de información y la viabilidad del diseño, previamente a la generación de las especificaciones de construcción.

Para cumplir dicho objetivo se llevarán a cabo las siguientes acciones:

- Verificación de la calidad técnica de cada modelo o especificación
- Aseguramiento de la coherencia entre los distintos modelos
- Aceptación del diseño de la arquitectura por parte de Explotación y Sistemas.

Para el caso del presente proyecto de tesis, estas actividades están reflejadas en las correcciones efectuadas por el director de tesis. El resultado de dichas correcciones se incorpora a los modelos, realimentándolos y generando nuevas versiones de los mismos.

6.8. Generación de Especificaciones de Construcción

En esta actividad se generan las especificaciones para la construcción del sistema de información, a partir del diseño detallado.

Estas especificaciones definen la construcción del sistema de información a partir de los componentes o unidades básicas de construcción (en adelante, componentes), entendiendo como tal, unidades independientes y coherentes de construcción y ejecución, que se corresponden con un empaquetamiento físico de los elementos del diseño de detalle, como pueden ser módulos, clases o especificaciones de interfaz.

La división del sistema de información en subsistemas de diseño proporciona, por continuidad, una primera división en subsistemas de construcción, definiendo para cada uno de ellos los componentes que lo integran. Si se considera necesario se podrá dividir a su vez en sucesivos niveles para mayor claridad de las especificaciones de construcción.

Las dependencias entre subsistemas de diseño proporcionan información para establecer las dependencias entre los subsistemas de construcción, y por lo tanto, definir el orden o secuencia que se debe seguir en la construcción y en la realización de las pruebas.

También se generan las especificaciones necesarias para la creación de las estructuras de datos en los gestores de bases de datos o sistemas de ficheros.

El producto resultante de esta actividad es el conjunto de las especificaciones de construcción del sistema de información, que comprende:

6.8.1. Especificación del Entorno de Construcción

El objetivo de esta tarea es la definición detallada y completa del entorno necesario para la construcción de los componentes del sistema de información.

La especificación del entorno se propone que se realice según los siguientes conceptos:

<i>Concepto</i>	<i>Definición</i>
Entorno tecnológico: hardware, software y comunicaciones.	El equipo de desarrollo debe ser un PC - INTEL Pentium III de 850 Mhz o superior. 256 Mb de Memoria RAM (mínimo) y un disco rígido con no menos de 4 Gb. Sistema operativo LINUX o Windows 2000. Base de datos MySQL. Protocolo TCP/IP.
Herramientas de construcción, generadores de código, compiladores, etc.	IBM VisualAge for Java 3.5. Java Developer Kit 1.2.2.
Restricciones técnicas del entorno.	No se observan.
Planificación de capacidades previstas, o la	No se observan.

información que estime oportuno el departamento de sistemas para efectuar dicha planificación.	
Requisitos de operación y seguridad del entorno de construcción.	No se observan.

6.8.2. Definición de Componentes y Subsistemas de Construcción

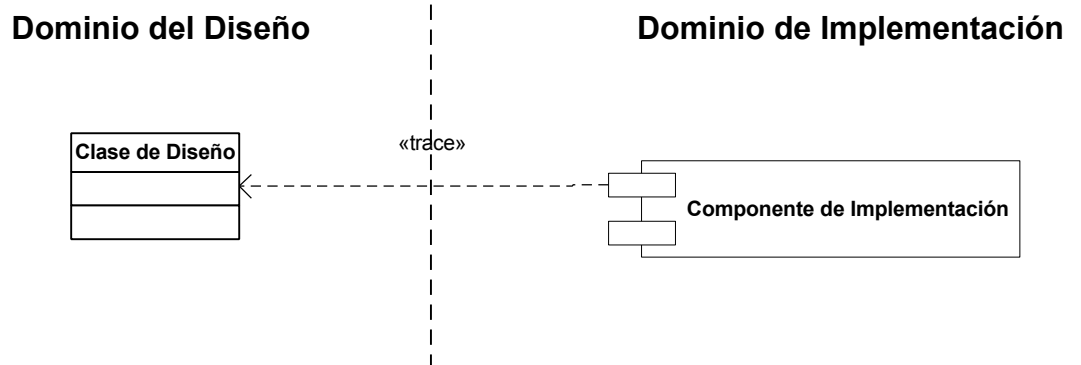
La especificación de los subsistemas de construcción se realiza a partir de los subsistemas de diseño, con una continuidad directa, permitiéndose a su vez un mayor nivel de detalle agrupando componentes en subsistemas dentro de un subsistema de construcción.

La definición de los componentes se realiza a partir del diseño de detalle de cada subsistema de diseño, mediante la agrupación de elementos del diseño de detalle.

En principio, cada clase corresponde con un componente, aunque se podrán agrupar o redistribuir clases en componentes, siguiendo otros criterios más oportunos como pueden ser:

- Optimización de recursos
- Características comunes de funcionalidad o de acceso a datos
- Necesidades especiales de ejecución: elementos críticos, accesos costosos a datos, etc.

En este caso, se define por dejar las clases tal y como se han diseñado. De esta forma cada clase que compone el diseño se encontrará representada por una clase en el dominio de la implementación y viceversa. Esta relación puede observarse en el siguiente diagrama UML:



Los subsistemas de construcción y las dependencias entre subsistemas y entre componentes de un subsistema recogen aspectos prácticos relativos a la plataforma concreta de construcción y ejecución.

<i>Aspecto</i>	<i>Definición</i>
Secuencia de compilación entre componentes	El ambiente de desarrollo compila los componentes incrementalmente. Es decir, a medida que se generan o modifican. Por lo tanto no existe la necesidad de especificar secuencia de compilación alguna.
Agrupación de elementos en librerías o paquetes	Los componentes (clases) serán distribuidos en archivos JAR (Java ARchive) que son de uso estándar al desarrollar con el lenguaje Java.

6.9. Diseño de la Migración y Carga Inicial de Datos

Esta actividad es necesaria para planificar la carga inicial de datos del sistema. Tal y como ya se había establecido en capítulos anteriores, la catalogación de productos así como la gestión de proveedores de productos, escapan al alcance del presente proyecto.

De todas formas, el sistema necesita de una carga inicial de datos para poder funcionar y demostrar su utilidad. Si bien no es el objetivo desarrollar una interfaz real con otros sistemas específicos a la catalogación, sí lo es cargar una cantidad de productos iniciales.

De acuerdo a la estructura física de los datos del nuevo sistema obtenida en la actividad Diseño Físico de Datos (DSI 6) y a las características

de la arquitectura y del entorno tecnológico propuesto en la actividad Definición de la Arquitectura del Sistema (DSI 1), se procede a definir y diseñar en detalle los procedimientos y procesos necesarios para realizar la carga inicial.

Como resultado de esta actividad se actualiza el plan de migración y carga inicial de datos con la información siguiente:

- Especificación del entorno de migración
- Definición de procedimientos de migración
- Planificación de la migración y carga inicial.

Es importante considerar que una carga inicial de información no tiene el mismo alcance y complejidad que una migración de datos, de modo que las tareas de esta actividad se llevarán a cabo en mayor o menor medida en función de las características de los datos a cargar.

6.9.1. Diseño de Procedimientos de Migración y Carga Inicial

El objetivo de esta tarea es la definición de los procedimientos necesarios para llevar a cabo la carga inicial de datos del sistema.

Como punto de partida, junto con los requisitos y especificaciones de migración y carga inicial se tendrá en cuenta el modelo físico de datos optimizado y su localización en los nodos, así como la definición del entorno tecnológico del sistema de información.

Los procedimientos asociados a la migración y carga inicial de datos son, principalmente los relacionados con la preparación, la realización y la posterior verificación del proceso.

Para el caso diremos que los datos se agruparán en archivos del tipo SCRIPT (archivos ASCII que contienen las instrucciones necesarias para la correcta ejecución de la carga inicial). Dichas instrucciones son sentencias de inserción en lenguaje SQL con los valores prefijados para comenzar las pruebas.

Para la ejecución de dichos archivos, se utilizarán las herramientas de administración provistas por el producto MySQL.

La carga inicial comprende el alta de:

- Usuarios del sistema (tanto compradores como proveedores).

- Productos (pseudo-catálogo).
- Y la tabla de relación entre proveedores y productos.

6.10. Especificación Técnica del Plan de Pruebas

En esta actividad se realiza la especificación de detalle del plan de pruebas del sistema de información, para cada uno de los niveles de prueba establecidos en el proceso Análisis del Sistema de Información:

- Pruebas unitarias
- Pruebas de integración
- Pruebas del sistema
- Pruebas de implantación
- Pruebas de aceptación.

Para ello se toma como referencia el plan de pruebas, que recoge los objetivos de la prueba de un sistema, establece y coordina una estrategia de trabajo, y provee del marco adecuado para planificar paso a paso las actividades de prueba. También será una referencia el plan de integración del sistema de información propuesto, opcionalmente, en la tarea Definición de Componentes y Subsistemas de Construcción.

El catálogo de requisitos, tanto funcionales como no funcionales, el catálogo de excepciones y el diseño detallado del sistema de información, permiten la definición de las verificaciones que deben ser realizadas en cada nivel de prueba para comprobar que el sistema responde a los requisitos planteados. La asociación de las distintas verificaciones a componentes, grupos de componentes y subsistemas, o al sistema de información completo, determina las distintas verificaciones de cada nivel de prueba establecido.

Las pruebas unitarias comprenden las verificaciones asociadas a cada componente del sistema de información. Su realización tiene como objetivo verificar la funcionalidad y estructura de cada componente individualmente.

Las pruebas de integración comprenden verificaciones asociadas a grupos de componentes, generalmente reflejados en la definición de subsistemas de construcción o en el plan de integración del sistema de

información y tienen como objetivo verificar el correcto ensamblaje entre los distintos componentes.

Las pruebas del sistema, de implantación y de aceptación corresponden a verificaciones asociadas al sistema de información, reflejando distintos propósitos en cada tipo de prueba:

Las pruebas del sistema son pruebas de integración del sistema de información completo, y permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen.

Las pruebas de implantación incluyen las verificaciones necesarias para asegurar que el sistema funcionará correctamente en el entorno de operación al responder satisfactoriamente a los requisitos de rendimiento, seguridad y operación, y coexistencia con el resto de los sistemas de la instalación, y conseguir la aceptación del sistema por parte del usuario de operación.

Las pruebas de aceptación van dirigidas a validar que el sistema cumple los requisitos de funcionamiento esperado recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información, y conseguir la aceptación final del sistema por parte del usuario.

Las pruebas unitarias, de integración y del sistema se llevan a cabo en el proceso Construcción del Sistema de Información (CSI), mientras que las pruebas de implantación y aceptación se realizan en el proceso Implantación y Aceptación del Sistema (IAS).

Como resultado de esta actividad se actualiza el plan de pruebas con la información siguiente:

- Especificación del entorno de pruebas
- Especificación técnica de niveles de prueba
- Planificación de las pruebas.

6.10.1. Especificación del Entorno de Pruebas

El objetivo de esta tarea es la definición detallada y completa del entorno necesario para la realización de las pruebas, tanto unitarias, de integración, del sistema, de implantación y de aceptación.

6.10.1.1. Entorno tecnológico

El hardware, software y cuestiones de comunicaciones a utilizar para el ambiente de pruebas es el mismo que se ha definido como requisito en el ERS para un entorno productivo.

Restricciones técnicas del entorno

Las mismas especificadas para el entorno de producción.

Requisitos de operación y seguridad del entorno de pruebas

Las mismas especificadas para el entorno de producción

Herramientas de prueba relacionadas con la extracción de juegos de ensayo, análisis de resultados, utilidades de gestión del entorno etc.

Se utilizarán las herramientas provistas por el proveedor del RDBMS para estas tareas. En especial para este caso se utilizará el programa de línea de comandos mysql.exe.

Planificación de capacidades previstas, o la información que estime oportuno el departamento técnico para efectuar dicha planificación.

No existe la necesidad de planificar capacidades superiores a las necesarias para realizar las pruebas de aplicación, dado que la arquitectura de hardware y software planteadas son más que suficientes para un uso intensivo del sistema.

Procedimientos de promoción de elementos entre entornos

El sistema, para sus pruebas, será instalado en un sistema operativo con capacidad de generar archivos TAR. De esta forma, una vez probado y aprobado, se generarán archivos de este tipo para cada directorio involucrado con el sistema y el mismo será enviado a un servidor de producción.

Procedimientos de emergencia y de recuperación, así como de vuelta atrás.

Se define como emergencia en la que haga falta una recuperación del sistema, a aquel caso en el que el servidor se ve dañado físicamente y por ende genera el mal funcionamiento de la aplicación. En este caso deberá ser necesario recuperar el sistema con el siguiente curso de acción:

1. Tomar los archivos de instalación del sistema.

2. Proceder a instalar el sistema en un nuevo servidor o bien el servidor reparado.
3. Recuperar la copia de resguardo (backup) de la base de datos. Este punto varía de acuerdo a la base de datos en cuestión. Para el caso del servidor MySQL, esta recuperación consiste en pisar los archivos de datos con la versión del backup. En servidores más complejos (Ej.: Oracle, Sybase, etc.) el proceso consiste en pisar los archivos de datos con la copia de seguridad y ejecutar un proceso de recuperación basado en el archivo de log de transacciones.
4. Iniciar el sistema.

6.10.2. Especificación Técnica de Niveles de Prueba

El objetivo de esta tarea es el diseño detallado de los distintos niveles de prueba, especificados en el plan de pruebas elaborado en el proceso Análisis del Sistema de Información (ASI).

El plan de integración del sistema de información, si se ha definido en la actividad Definición de Componentes y Subsistemas de Construcción (DSI 8.2), servirá de referencia para la elaboración detallada del plan de pruebas, principalmente las pruebas de integración y del sistema. En cualquier caso se especificará la estrategia de integración de dichas pruebas.

De acuerdo a la arquitectura del sistema propuesta y a las características intrínsecas del diseño del sistema de información, se definen en detalle las distintas verificaciones a realizar sobre el sistema, conforme a los niveles de prueba establecidos, teniendo en cuenta que una verificación puede ser aplicable a varios componentes o grupos de componentes.

Estas verificaciones deben cubrir aspectos funcionales y no funcionales, considerando las excepciones que puedan producirse, así como las soluciones de diseño adoptadas, tanto del propio diseño de detalle del sistema de información, como de la utilización de subsistemas de soporte propios de la instalación.

Las verificaciones a realizar se especifican detallando:

Ambito de aplicación (prueba unitaria, de integración, del sistema, de implantación o aceptación) y objetivo

Casos de prueba asociados: se definen en detalle los casos de prueba y se especifica como proceder en la ejecución de dichos casos, describiendo todas las entradas que se necesitan para ejecutar la prueba y las relaciones de secuencialidad que existan entre las entradas, así como todas aquellas salidas que se esperan obtener una vez ejecutado el caso de prueba y características especiales requeridas, como por ejemplo, tiempo de respuesta

Procedimientos de prueba: se determina el conjunto de pasos a seguir para asegurar que los casos de prueba se ejecutan adecuadamente, especificando:

- Casos de prueba a los que se aplica el procedimiento
- Recursos hardware y software necesarios para ejecutar el procedimiento
- Requisitos especiales o acciones necesarias para iniciar la ejecución
- Requisitos especiales o acciones necesarias a realizar durante la ejecución del procedimiento.

Entorno de prueba: herramientas adicionales, condicionantes especiales de

ejecución, etc.

Criterios de aceptación de la prueba

Análisis y evaluación de resultados.

Como resultado final se obtiene la relación de verificaciones que permiten comprobar el correcto funcionamiento de cada componente (pruebas unitarias), cada subsistema de construcción o conjunto de componentes (pruebas de integración), así como la integración del sistema de información en su globalidad (pruebas del sistema).

Además, se obtienen las verificaciones necesarias para garantizar que el sistema se ajusta a las necesidades para las que fue creado, de acuerdo a las características del entorno en el que se va a implantar (pruebas de implantación) y respondiendo satisfactoriamente a los requisitos especificados por el usuario (pruebas de aceptación).

6.10.3. Revisión de la Planificación de Pruebas

En esta tarea se completa y especifica la planificación de las pruebas, determinando los distintos perfiles implicados en la preparación y ejecución

de las pruebas y en la evaluación de los resultados, así como el tiempo estimado para la realización de cada uno de los niveles de prueba, de acuerdo a la estrategia de integración establecida.

Teniendo en cuenta que el sistema ha sido desarrollado con un enfoque de casos de uso, resulta un plan de pruebas que apunte a probar funcionalmente el sistema desde el punto de vista de los casos de uso, componentes de infraestructura y pruebas globales del sistema. Los tipos de prueba a realizar son:

- **Pruebas Unitarias:** prueban componentes específicos del sistema. Se preparará un plan de pruebas para el componente de comunicaciones.
- **Pruebas de Integración:** abarcan las pruebas por caso de uso (ya integrados al componente de comunicaciones y ejecutándose contra el servidor).
- **Pruebas de Sistema:** las pruebas serán ejecutadas y obtenidos sus resultados utilizando el mismo sistema como herramienta y probando el circuito total.

6.10.3.1. Pruebas Unitarias

El único componente a probar en forma independiente es el componente de comunicaciones. Las pruebas tienen por objetivo la ejecución de una transacción específica y evaluar el funcionamiento del componente tal y como se detalla en su diseño y en el catálogo de requisitos y excepciones.

Para ello, deberá construirse un programa especial que permita probar el componente. La transacción a ejecutar será la autenticación de usuario. El programa recibirá por línea de comandos un par usuario-contraseña y ejecutará la transacción de autenticación en el servidor. Como resultado imprimirá en la consola el nombre completo del usuario. El programa se llamará CommPrue.

Comunicaciones - CP001	
Objetivo	Probar una autenticación exitosa.
Entrada	mpetronio mpetronio
Salida	Marcelo Petronio
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Introducir en un intérprete de comandos</p> <pre>java emarket.bat</pre> <p>Al abrirse la pantalla de login ingresar como usuario mpetronio y clave mpetronio</p> <p>Se deberá observar el menú principal del sistema indicándose en la parte superior el nombre de usuario:</p> <pre>Marcelo Petronio</pre> <p>En el log del servidor (ubicado en el directorio de instalación y con el nombre de emarket.log) debe figurar una entrada:</p> <pre>dd/mm/yyy hh:mm:ss LOGIN Marcelo Petronio</pre>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>Levantar el servidor de aplicaciones.</p>

Comunicaciones - CP002	
Objetivo	Probar una autenticación errónea.
Entrada	mpetronio marcelo
Salida	Usuario inválido
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Introducir en un intérprete de comandos</p> <pre>java CommPrue mpetronio marcelo [Enter]</pre> <p>Se deberá observar la salida:</p> <pre>Usuario Inválido</pre> <p>En el log del servidor (ubicado en el directorio de instalación y con el nombre de emarket.log) debe figurar:</p> <pre>dd/mm/yy hh:mm:ss LOGIN Usuario Inválido</pre>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql.</p> <p>Levantar el servidor de aplicaciones.</p>

Comunicaciones - CP003	
Objetivo	Probar un error en comunicaciones.
Entrada	mpetronio mpetronio
Salida	Error de comunicaciones.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Ejecutar caso de prueba CP001.</p> <p>Se deberá observar la salida:</p> <pre>Error de comunicaciones.</pre>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>Bajar el servidor de aplicaciones.</p>

Buscador de Productos - CP001	
Objetivo	Probar búsqueda por código exitosa.
Entrada	BB02
Salida	La tabla debe mostrar un único producto de código BB02 y descripción LAMPARA 75W
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar opción "Crear pedido de cotización"</p> <p>Seleccionar opción "Agregar producto"</p> <p>Presionar el botón "...". Cuando aparezca la pantalla, ingresar en código el valor "BB02" y presionar buscar.</p> <p>Se deberá observar la salida:</p> <p>BB02 LAMPARA 75W</p>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>Ejecutar caso de prueba CP001</p>

Buscador de Productos - CP002	
Objetivo	Probar búsqueda por código errónea.
Entrada	XX02
Salida	Código Inexistente
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar opción "Crear pedido de cotización"</p> <p>Seleccionar opción "Agregar producto"</p> <p>Cuando aparezca la pantalla, ingresar en código el valor "XX02" y presionar buscar.</p> <p>Se deberá observar la salida:</p> <p>Código Inexistente</p>
Prerequisitos	Ejecución del script de prueba: ini_pru.sql

Buscador de Productos - CP003	
Objetivo	Probar búsqueda por descripción parcial.
Entrada	LAMPARA
Salida	Lista de productos que contienen la palabra LAMPARA en su descripción (BB01 y BB02).
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar opción "Crear pedido de cotización"</p> <p>Seleccionar opción "Agregar producto"</p> <p>Seleccionar la opción "Por Descripción"</p> <p>Cuando aparezca la pantalla, ingresar en código el valor "LAMPARA" y presionar buscar.</p> <p>Se deberá observar la salida:</p> <p>BB01 LAMPARA 100W</p> <p>BB02 LAMPARA 75W</p>
Prerequisitos	Ejecución del script de prueba: ini_pru.sql

Buscador de Productos - CP004	
Objetivo	Búsqueda por descripción sin productos
Entrada	Z
Salida	No existen productos que contengan "Z" en su descripción.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar opción "Crear pedido de cotización"</p> <p>Seleccionar opción "Agregar producto"</p> <p>Seleccionar la opción de búsqueda "Por Descripción".</p> <p>Cuando aparezca la pantalla, ingresar en código el valor "Z" y presionar buscar.</p> <p>Se deberá observar la salida:</p> <p>No existen productos que contengan "Z" en su descripción.</p>
Prerequisitos	Ejecución del script de prueba: ini_pru.sql

Buscador de Productos - CP005	
Objetivo	Probar búsqueda sin ingresar código de producto.
Entrada	
Salida	Debe ingresar un código de producto.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Cuando aparezca la pantalla, presionar buscar. Se deberá observar la salida: Debe ingresar un código de producto.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql

Buscador de Productos - CP006	
Objetivo	Búsqueda por descripción sin ingresar descripción
Entrada	
Salida	Debe ingresar una descripción.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Introducir en un intérprete de comandos <code>java ProdPrue [Enter]</code> Seleccionar la opción de búsqueda "Por Descripción". Presionar buscar. Se deberá observar la salida: Debe ingresar una descripción.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql

Buscador de Productos - CP007	
Objetivo	Presionar seleccionar sin seleccionar u producto.
Entrada	Buscador de Productos - CP003
Salida	Debe seleccionar un producto.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ejecutar el caso de prueba CP003. Luego, presionar <Seleccionar>. Debe aparecer el mensaje. Debe seleccionar un producto.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql

Buscador de Productos - CP008	
Objetivo	Selección de un producto.
Entrada	Buscador de Productos - CP007
Salida	Nombre del producto seleccionado.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ejecutar el caso de prueba CP007. Seleccionar un producto de la lista. Presionar <Seleccionar> Deberá aparecer impreso el nombre del producto seleccionado.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql

6.10.3.2. Pruebas de Integración

Las pruebas de integración tienen por objetivo encontrar fallas en el funcionamiento de los componentes y subsistemas del sistema, al funcionar en conjunto para proveer la funcionalidad deseada.

Dada la topología y funcionalidad del presente desarrollo, se harán las pruebas necesarias dentro de las pruebas del sistema. Esta definición se basa en las siguientes premisas:

- Los componentes de funcionalidad del usuario no pueden ser probados sin el uso del componente servidor y los medios de comunicaciones (ya probados en el apartado anterior).
- Los subsistemas de compra y venta serán probados dentro de las pruebas del sistema ya que de todas formas cada uno de los casos de uso genera y consume información de un proceso anterior que forma parte del subsistema complementario, es decir, son módulos que interactúan constantemente contra la interfaz del módulo complementario.

6.10.3.3. Pruebas del Sistema

Las pruebas de integración tienen como objetivo probar cada uno de los casos de uso implementados en la aplicación.

Ingreso al Sistema - CP001	
Objetivo	Probar autenticación exitosa de un comprador
Entrada	mpetronio mpetronio
Salida	Menú principal del comprador
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ejecutar el programa en el cliente <code>emarket.bat</code> y presionar [ENTER] Cuando aparezca la pantalla de ingreso teclear: mpetronio en el nombre de usuario y mpetronio en la clave. Presionar ACEPTAR. En el log del servidor debe figurar: <code>dd/mm/yyy hh:mm:ss LOGIN Comprador: Marcelo Petronio</code>
Prerequisitos	Ejecución del script de prueba: <code>ini_pru.sql</code> El servidor de aplicaciones y la base de datos deben estar corriendo.

Ingreso al Sistema - CP002	
Objetivo	Probar autenticación exitosa de un proveedor
Entrada	provA provA
Salida	Menú principal del proveedor
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ejecutar el programa en el cliente. Cuando aparezca la pantalla de ingreso teclear: provA en el nombre de usuario y provA en la clave. Presionar OK. En el log del servidor debe figurar: dd/mm/yyyy hh:mm:ss LOGIN Proveedor: Proveedor A
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo.

Ingreso al Sistema - CP003	
Objetivo	Probar autenticación errónea
Entrada	provA abc123
Salida	Usuario Inválido
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ejecutar el programa en el cliente. Cuando aparezca la pantalla de ingreso teclear: provA en el nombre de usuario y abc123 en la clave. Presionar OK. En el log del servidor debe figurar: dd/mm/yyyy hh:mm:ss LOGIN Usuario Inválido
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo.

Ingreso al Sistema - CP004	
Objetivo	Probar campos requeridos
Entrada	
Salida	Menú principal del proveedor
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ejecutar el programa en el cliente. Cuando aparezca la pantalla de ingreso presionar OK. Al ver el mensaje de error completar el login con provA. Presionar OK. Debe aparecer un mensaje de error indicando que faltan valores.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo.

Ingreso al Sistema - CP005	
Objetivo	Probar campos extremos.
Entrada	AAAAAAAAAAAAAAAA BBBB
Salida	Usuario Inválido
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ejecutar el programa en el cliente. Cuando aparezca la pantalla de ingreso teclear 15 letras A y en la clave 10 letras B Presionar OK.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo.

Cargar Pedido de Cotización - CP001	
Objetivo	Cargar un ítem válido.
Entrada	Un nuevo ítem para el pedido de cotización.
Salida	Lista de ítems con el nuevo producto agregado.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ingresa al sistema con el usuario "mpetronio" y clave "mpetronio". Presionar el botón <Crear Pedido de Cotización>. Seleccionar <Agregar Producto>. Buscar producto "CABLE CANAL". Cargar en cantidad 10. Cargar en fecha de entrega el 31 de diciembre de 2002. Cargar en precio estimado 10,50. Presionar <Aceptar>.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Cargar Pedido de Cotización - CP002	
Objetivo	Cargar un ítem válido pero con fecha inválida.
Entrada	Un nuevo ítem para el pedido de cotización con fecha inválida.
Salida	La fecha de entrega no puede ser anterior a hoy.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Seleccionar <Agregar Producto>. Buscar producto "CABLE CANAL". Cargar en cantidad 10. Cargar en fecha de entrega la fecha al día anterior al actual. Cargar en precio estimado 10,50. Presionar <Aceptar>. Deberá aparecer: La fecha de entrega no puede ser anterior a hoy.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Cargar Pedido de Cotización - CP003	
Objetivo	Cargar datos alfanuméricos en el precio y la cantidad.
Entrada	Un nuevo ítem para el pedido de cotización con fecha inválida.
Salida	La fecha de entrega no puede ser anterior a hoy.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar <Agregar Producto>.</p> <p>Buscar producto "CABLE CANAL".</p> <p>Cargar en cantidad AAAAA.</p> <p>Cargar en fecha de entrega la fecha al día anterior al actual.</p> <p>Cargar en precio estimado AAAA.</p> <p>Presionar <Aceptar>.</p> <p>Deberá aparecer:</p> <ol style="list-style-type: none"> 1) Ingrese una cantidad válida. 2) Ingrese un precio válido.
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>

Cargar Pedido de Cotización - CP004	
Objetivo	Cargar valores extremos en precio y cantidad.
Entrada	1.000.000.000 1.000.000.000
Salida	Ingrese un precio válido. Ingrese una cantidad válida.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar <Agregar Producto>.</p> <p>Buscar producto "LAMPARA 100W".</p> <p>Cargar en cantidad 1000000000.</p> <p>Cargar en fecha cualquier fecha válida.</p> <p>Cargar en precio estimado 1000000.</p> <p>Presionar <Aceptar>.</p> <p>Deberá aparecer:</p> <p>El precio unitario no puede superar los 1000000.</p> <p>Corregir el precio a 100.</p> <p>Presionar <Aceptar></p> <p>Deberá aparecer:</p> <p>Ingrese un precio válido.</p> <p>Corregir la cantidad a 1000</p> <p>Presionar <Aceptar></p> <p>El nuevo ítem deberá aparecer en la lista.</p>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>
Cargar Pedido de Cotización - CP005	
Objetivo	<p>Probar la carga exitosa de una cotización.</p> <p>Este caso de prueba también integra al asignador de pedidos de cotización.</p>
Entrada	Un pedido de cotización completo.
Salida	Verificar con SQL que se hayan generado los pedidos de cotización correspondientes (tablas de pricing_order_requests y

	pricing_order_request_items).
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Entrar en <Crear Pedido de Cotización></p> <p>Agregar los ítems:</p> <pre>BB01 1000 31/12/2002 100 BB02 250 31/12/2002 50</pre> <p>Presionar el botón "Enviar Pedido".</p> <p>En el log del servidor debe figurar:</p> <pre>dd/mm/yy hh:mm:ss PEDIDO COTIZACIÓN de Marcelo Petronio</pre> <p>Iniciar la utilidad de comandos, para eso es necesario ir al directorio donde se encuentra instalado el MySQL y escribir</p> <pre>bin\mysql + [ENTER] use ITBA + [ENTER]</pre> <p>Verificar la carga correcta del pedido de cotización en la base de datos por medio de las consultas:</p> <pre>select * from pricing_order_requests; select * from pricing_order_request_items;</pre>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>

Consultar Pedidos de Cotización - CP001	
Objetivo	Probar la Consulta de Pedido de Cotización.
Entrada	----
Salida	Una lista que contiene el pedido de cotización cargado en el caso de prueba anterior.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ingresar al sistema como usuario "provA" y clave "provA" Presionar <Consultar Cotizaciones> Deberá abrirse una pantalla con una lista de pedidos de cotización. Deberá haber un único pedido a nombre de Marcelo Petronio y con la fecha de carga. En el log del servidor debe figurar: <code>dd/mm/yyy hh:mm:ss CONSULTA COTIZACIONES de Proveedor A</code>
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Consultar Pedidos de Cotización - CP002	
Objetivo	Probar la selección de un pedido de cotización.
Entrada	----
Salida	Debe seleccionar un pedido de cotización.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Responder> sin haber seleccionado un pedido de cotización. Aparecerá el mensaje: <code>Debe seleccionar un pedido de cotización.</code>
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Consultar Pedidos de Cotización - CP003	
Objetivo	Probar la selección de un pedido de cotización.
Entrada	----
Salida	Pantalla de edición de pedido de cotización.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	No seleccionar un pedido de cotización. Presionar <Responder> sin haber seleccionado un pedido de cotización. Deberá aparecer la pantalla "Responder Pedido de Cotización"
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Cargar Cotización - CP001	
Objetivo	Probar la selección de un ítem.
Entrada	----
Salida	Debe seleccionar un ítem.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Editar Item> sin haber seleccionado uno de la lista Aparecerá el mensaje: Debe seleccionar un ítem.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Cargar Cotización - CP002	
Objetivo	Prueba de valores por defecto.
Entrada	----
Salida	Una pantalla con los valores por defecto correctamente cargados
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Seleccionar un ítem y presionar <Editar Item>. Deberá aparecer una pantalla con 2 columnas donde los valores de lo solicitado deben coincidir con lo confirmado.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Cargar Cotización - CP003	
Objetivo	Prueba de límite inferior de la cantidad.
Entrada	----
Salida	Debe ingresar una cantidad válida.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Ingresar en cantidad un valor menor a 1. Presionar <Aceptar>. Deberá aparecer una leyenda: La cantidad no puede ser menor a 1.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Cargar Cotización - CP004	
Objetivo	Prueba de límite superior de la cantidad.
Entrada	----
Salida	Debe ingresar una cantidad válida.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar un ítem y presionar <Editar Item>.</p> <p>Ingresar el valor 10.000.001.</p> <p>Presionar <Aceptar></p> <p>Deberá aparecer una leyenda.</p> <p>La cantidad no puede ser superior a 10.000.000.</p>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>

Cargar Cotización - CP005	
Objetivo	Prueba de corrección.
Entrada	Cantidad = 2000, Precio = 50
Salida	Lista actualizada con los valores correctos.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Seleccionar un ítem y presionar <Editar Item></p> <p>Ingresar una cantidad de 2000.</p> <p>Ingresar un precio unitario de 50.</p> <p>Presionar <Aceptar></p> <p>Deberá aparecer la tabla con los ítems de la cotización con los valores actualizados.</p>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>

Cargar Cotización - CP006	
Objetivo	Carga de una cotización.
Entrada	----
Salida	Verificar en base de datos.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Enviar> La pantalla de carga deberá cerrarse. En el log del servidor debe figurar: dd/mm/yyy hh:mm:ss COTIZACION de Proveedor A a Marcelo Petronio Verificar en la base de datos el correcto ingreso de la cotización con las consultas: <pre>select * from pricing_order_respnses; select * from pricing_order_response_items;</pre>
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Consulta de Cotizaciones - CP001	
Objetivo	Probar la consulta de cotizaciones.
Entrada	----
Salida	Cotización recién ingresada.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Ingresar al sistema como usuario mpetronio clave mpetronio.</p> <p>Presionar <Consultar Cotizaciones></p> <p>Se abrirá una pantalla con la cotización ingresada en el caso de prueba anterior.</p> <p>Al seleccionar una cotización, deberá aparecer en la lista inferior el detalle de ítems de esa cotización.</p> <p>En el log del servidor debe figurar:</p> <pre>dd/mm/yyy hh:mm:ss CONSULTA COTIZACIÓN de Marcelo Petronio</pre>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>

Generar Orden de Compra - CP001	
Objetivo	Ingresar Orden de Compra.
Entrada	----
Salida	Pantalla con una nueva orden de compra.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Presionar <Nueva Orden de Compra></p> <p>Deberá aparecer una pantalla de confección de orden de compra.</p>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>

Generar Orden de Compra - CP002	
Objetivo	Cargar un producto sin cotizaciones.
Entrada	Producto CC01
Salida	El producto no posee cotizaciones. Pida una cotización para este producto.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Agregar> Presionar el botón <...> a la derecha del campo de entrada de producto. Seleccionar el producto CC01 La lista de cotizaciones debe estar vacía. Presionar <Aceptar> Deberá aparecer el mensaje: El producto no posee cotizaciones. Pida una cotización para este producto.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Generar Orden de Compra - CP003	
Objetivo	Cargar un producto con cotizaciones vencidas.
Entrada	Producto BB01
Salida	El producto no posee cotizaciones. Pida una cotización para este producto.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Agregar> Presionar el botón <...> a la derecha del campo de entrada de producto. Seleccionar el producto BB01 La lista de cotizaciones debe estar vacía. Presionar <Aceptar> Deberá aparecer el mensaje: El producto no posee cotizaciones. Pida una cotización para este producto.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores. Cambiar la fecha del servidor a una 2 días adelante de la actual.

Generar Orden de Compra - CP004	
Objetivo	Aceptar orden de compra sin productos.
Entrada	----
Salida	No se puede generar una orden de compra sin solicitar productos.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Enviar> Deberá aparecer en la pantalla: No se puede generar una orden de compra sin solicitar productos.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.
Generar Orden de Compra - CP005	
Objetivo	Cargar un producto con cotizaciones.
Entrada	Producto BB01
Salida	Lista de productos con el nuevo producto cargado.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Agregar> Presionar el botón <...> a la derecha del campo de entrada de producto. Seleccionar el producto BB01 La lista de cotizaciones deberá mostrar la cotización realizada en los casos de prueba anteriores. Presionar <Aceptar> Deberá cerrarse la pantalla y aparecer un nuevo ítem en la lista a colocar por la orden de compra.
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Generar Orden de Compra - CP005	
Objetivo	Aceptar una orden de compra.
Entrada	----
Salida	Verificar en la base de datos la entrada de la orden de compra.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	Presionar <Enviar> La pantalla de carga deberá cerrarse. En el log del servidor debe figurar: dd/mm/yy hh:mm:ss ORDEN COMPRA de Marcelo Petronio Verificar en la base de datos la carga de la orden de compra con las consultas: <pre>select * from purchase_orders select * from purchase_order_items</pre>
Prerequisitos	Ejecución del script de prueba: ini_pru.sql El servidor de aplicaciones y la base de datos deben estar corriendo. Realizar casos de prueba anteriores.

Consultar Ordenes de Compra - CP001	
Objetivo	Consultar la orden de compra ingresada en el caso de prueba anterior.
Entrada	----
Salida	Lista con la orden de compra ingresada en el caso de prueba anterior.
Condiciones	No se permite el acceso a la base de datos por otros programas mientras se realizan las pruebas.
Procedimiento	<p>Ingresar al sistema como usuario "provA" y clave "provA"</p> <p>Presionar <Consultar Ordenes de Compra></p> <p>Aparecerá una lista que contiene la orden de compra ingresada en el caso anterior.</p> <p>Seleccionar la orden de compra.</p> <p>Aparecerá en la parte inferior de la pantalla una lista con el producto y condiciones seleccionado en el apartado anterior.</p> <p>En el log del servidor debe figurar:</p> <pre>dd/mm/yyy hh:mm:ss CONSULTA ORDEN COMPRA de Proveedor A</pre>
Prerequisitos	<p>Ejecución del script de prueba: ini_pru.sql</p> <p>El servidor de aplicaciones y la base de datos deben estar corriendo.</p> <p>Realizar casos de prueba anteriores.</p>

6.11. Establecimiento de Requisitos de Implantación

En esta actividad se completa el catálogo de requisitos con aquellos relacionados con la documentación que requerirá el usuario para operar con el nuevo sistema, y los relativos a la propia implantación del sistema en el entorno de operación.

La incorporación de estos requisitos permitirá ir preparando, en los procesos de Construcción del Sistema de Información (CSI) e Implantación y Aceptación del Sistema (IAS), los medios y recursos necesarios para que los usuarios, tanto finales como de operación, sean capaces de utilizar el nueva sistema de forma satisfactoria.

6.11.1. Especificación de Requisitos de Documentación de Usuario

En esta tarea se recoge toda la información necesaria para la especificación posterior de la documentación a entregar al usuario, que incluirá los manuales de usuario y, cuando proceda, los manuales de explotación.

La documentación del usuario se encuentra detallada en el ANEXO II "Manual del Usuario del Sistema de Información".

6.11.2. Especificación de Requisitos de Implantación

En esta tarea se especifican de forma detallada los requisitos de implantación, generalmente relacionados con la formación, infraestructura e instalación, con el fin de preparar y organizar, con la antelación suficiente, todos los recursos necesarios para la implantación e instalación del sistema de información.

Teniendo en cuenta las particularidades del sistema de información, se determinan los conocimientos o aptitudes adicionales que requerirán los usuarios finales para operar con el nuevo sistema, al margen de la funcionalidad soportada por el mismo. Como consecuencia se pueden establecer requisitos de formación indispensables, como condición previa, para el desarrollo del plan de formación que se elaborará en el proceso Implantación y Aceptación del Sistema (IAS).

Los requisitos de infraestructura e instalación hacen referencia a las necesidades especiales de equipamiento software, hardware y comunicaciones que exigirá el nuevo sistema, así como los tipos de elementos implicados en la instalación, que deberán tenerse en cuenta al especificar la estrategia de implantación, en el proceso Implantación y Aceptación del Sistema (IAS). Por lo tanto esta actividad será detallada en las primeras fases del proceso de Implantación y Aceptación del Sistema.

6.12. Presentación y Aprobación del Diseño del Sistema de Información

Para el caso del presente proyecto, esta actividad consiste en la presentación del documento de diseño del sistema de información al tutor de tesis a fin de evaluar el mismo.

Además se cuenta con la aprobación por parte del usuario líder que acompañó el desarrollo del sistema a lo largo del período de escritura de tesis

Capítulo VII

Construcción del Sistema de Información

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Construcción
Fase de Origen:	Construcción del Sistema de Información
Nombre o Código del Elemento de Configuración de Software:	Construcción del Sistema de Información (CSI)
Versión N°: 1.0.1	Entrega N°: 02
Fecha de Última Modificación:	08/07/2002
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

7. Construcción del Sistema de Información

7.1. Preparación del Entorno de Generación y Construcción

El objetivo de esta actividad es asegurar la disponibilidad de todos los medios y facilidades para que se pueda llevar a cabo la construcción del sistema de información.

Entre estos medios cabe destacar la preparación de los puestos de trabajo, equipos físicos y lógicos, gestores de bases de datos, bibliotecas de programas, herramientas de generación de código, bases de datos o ficheros de prueba, entre otros.

Las características del entorno de construcción y sus requisitos de operación y seguridad, así como las especificaciones de construcción de la estructura física de datos, se establecieron en la actividad Generación de Especificaciones de Construcción y constituirán el punto de partida para la realización de esta actividad.

7.1.1. Implantación de la Base de Datos Física o Ficheros

En esta tarea se crean los elementos del sistema gestor de base de datos, se inicializa la base de datos y se cargan los datos que se consideren necesarios en el espacio de almacenamiento previamente definido.

En este proyecto esta tarea consiste de instalar la base de datos MySQL y correr las sentencias de DDL que se indican en el capítulo del diseño del sistema.

La instalación de la base de datos consiste en los siguientes pasos

1. Bajar el archivo de instalación mysql-3.23.35a-win.zip desde la dirección www.mysql.com.
2. Descomprimir el archivo de instalación y ejecutar el programa setup.exe
3. Seguir los pasos planteados por el asistente de instalación (configuración por defecto).
4. Una vez instalada la base, es necesario levantar la base por medio de la siguiente instrucción: `<directorio de instalación>\bin\mysqladmin start`
5. Una vez instalado el software hay que ejecutar el programa de interfaz de comandos ubicado en `<directorio de instalación>\bin\mysql`
6. Luego, es necesario crear una base de datos para almacenar las tablas del sistema. Esto se logra ejecutando la instrucción: `create database mv`
7. Es necesario indicar que de ahora en más se utilizará dicha base de datos con la instrucción: `use mv`
8. Ahora es necesario ejecutar todas las instrucciones de DDL que se especifica en el diseño de sistemas de información.
9. La base de datos ya esta configurada y lista para ser usada.

7.1.2. Preparación del Entorno de Construcción

En esta tarea se prepara el entorno en el que se construirán los componentes del sistema de información. Para ello será necesario:

1. Instalar el entorno de programación IBM VisualAge for Java en su versión estándar (gratuita). Esta herramienta contiene los editores y compiladores necesarios para el desarrollo.
2. Instalar una máquina virtual Java para la ejecución de los programas generados (java.sun.com). La versión recomendada es 1.2.
3. Además, es necesario instalar un driver JDBC para acceder a la base de datos. Este es provisto con la base de datos.

Para mayor información sobre la instalación de las librerías dentro del entorno de desarrollo, es necesario consultar los manuales del producto.

7.2. Generación del Código de los Componentes

El objetivo de esta actividad es la codificación de los componentes del sistema de información, a partir de las especificaciones de construcción obtenidas en el proceso Diseño del Sistema de Información (DSI), así como la construcción de los procedimientos de operación y seguridad establecidos para el mismo.

En paralelo a esta actividad se desarrollan las actividades relacionadas con las pruebas unitarias y de integración del sistema de información, lo que permite una construcción incremental en el caso de que así se haya especificado en el plan de pruebas y en el plan de integración del sistema de información.

7.2.1. Generación del Código de Componentes

En esta tarea se genera el código correspondiente a cada uno de los componentes del sistema de información identificados en la tarea Definición de Componentes y Subsistemas de Construcción.

Para generar el código fuente se tendrán en cuenta los estándares de nomenclatura, codificación y calidad definidos por Sun Microsystems tal y como se especifica en el Diseño del Sistema de Información.

Con el fin de verificar que el código fuente especifica de forma correcta el componente se realiza su ensamblaje o compilación, verificando y corrigiendo los errores sintácticos, y el enlace del código objeto obtenido con las correspondientes bibliotecas.

7.2.2. Generación de los Procedimientos de Operación y Seguridad

El objetivo de esta tarea es especificar y preparar los procedimientos de operación y administración del sistema de información, así como los procedimientos de seguridad y control de acceso, necesarios para ejecutar el sistema una vez que se haya implantado y esté en producción.

Como ya se ha indicado antes, el sistema posee dos componentes físicos: cliente y servidor.

Este apartado radica en las cuestiones relacionadas al componente servidor ya que en este se encuentran las transacciones y datos del sistema.

Para ejecutar los servicios del lado servidor, se debe crear una cuenta del sistema específica para tal fin. Esta cuenta debe tener permiso de escritura, lectura y ejecución sobre el árbol de directorios donde se encuentra instalado el sistema. Se sugiere como nombre de cuenta `adminmv` (administrador del mercado virtual), y como directorio raíz desde el cual instalar a `/opt/mv`.

Dentro de este directorio se deben instalar tanto el servidor de aplicaciones como el servidor de base de datos.

Una nota especial de seguridad radica en que el servidor de internet debe levantarse con el usuario `adminmv` pero una vez iniciado debe pasar el proceso a un usuario sin permisos o bien, con permisos mínimos (para evitar que un usuario que logre acceso al servidor pueda dañarlo de alguna forma). Por lo general, este es el funcionamiento estándar de todos los servidores de Internet.

La aplicación necesita una cuenta para conectarse a la base de datos. Para ello se recomienda la creación de un usuario `adminmv` dentro del servidor de bases de datos. Esta cuenta debe tener permisos de administrador sobre la base de datos donde residen las tablas y datos del sistema. Se recomienda llamar a la base de datos `mv`.

Para la operación del sistema, se recomienda la creación de un usuario adicional que solamente tenga permisos de ejecución de SQL sobre las tablas pero no de ejecución de DDL (no puede crear o borrar tablas). También se deben restringir operaciones como TRUNCATE que eliminan todos los datos de una tabla. Nuevamente, en caso de un ingreso indebido, el intruso no podrá borrar información que imposibilite el funcionamiento del sistema.

El servidor de Internet debe estar protegido por un firewall que evite accesos indebidos al sistema. Como nota adicional para la administración de la seguridad se destaca que solamente debe encontrarse habilitado el acceso al equipo en el puerto de comunicaciones número 80 y solamente para el protocolo HTTP (o bien el puerto 443 si se utilizan conexiones seguras con el protocolo SSL). Cualquier otro tipo de conexión debe encontrarse prohibida.

Una vez creadas y configuradas las cuentas como se indica, se procede a levantar los servicios del sistema.

Para levantar dichos servicios se deben ejecutar las siguientes instrucciones (habiendo ingresado al sistema con la cuenta creada a tal fin):

Inicio del servicio de base de datos

```
mysqladmin -start
```

Inicio del servidor de internet

```
apachectl start
```

En este momento los servicios del sistema se encuentran iniciados y listos para su utilización.

7.3. Ejecución de las Pruebas Unitarias

En esta actividad se realizan las pruebas unitarias de cada uno de los componentes del sistema de información, una vez codificados, con el objeto de comprobar que su estructura es correcta y que se ajustan a la funcionalidad establecida.

En el plan de pruebas se ha definido el entorno necesario para la realización de cada nivel de prueba, así como las verificaciones asociadas a las pruebas unitarias, la coordinación y secuencia a seguir en la ejecución de las mismas y los criterios de registro y aceptación de los resultados.

7.3.1. Realización de las Pruebas Unitarias

El objetivo de esta tarea es comprobar el correcto funcionamiento de los componentes del sistema de información, codificados en la actividad Generación del Código de los Componentes y Procedimientos (CSI), conforme a las verificaciones establecidas en el plan de pruebas para el nivel de pruebas unitarias, en la actividad Especificación Técnica del Plan de Pruebas (DSI).

Para cada verificación establecida, se realizan las pruebas con los casos de pruebas asociados, efectuando el correspondiente análisis y evaluación de los resultados y generando un registro conforme a los criterios establecidos en el plan de pruebas.

A continuación se plantea una tabla resumen con el resultado de cada una de los casos de pruebas para los componentes definidos:

Componente	Caso de Prueba	Resultado
Comunicaciones	CP001	Correcto
Comunicaciones	CP002	Correcto
Comunicaciones	CP003	Correcto
Buscador de Productos	CP001	Correcto
Buscador de Productos	CP002	Correcto
Buscador de Productos	CP002	Correcto
Buscador de Productos	CP003	Correcto
Buscador de Productos	CP004	Correcto
Buscador de Productos	CP005	Correcto
Buscador de Productos	CP006	Correcto
Buscador de Productos	CP007	Correcto
Buscador de Productos	CP008	Correcto

7.4. Ejecución de las Pruebas de Integración

El objetivo de las pruebas de integración es verificar si los componentes o subsistemas interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

La estrategia a seguir en las pruebas de integración se establece en el plan de pruebas, dónde se habrá tenido en cuenta el plan de integración del sistema de información, siempre y cuando, se haya especificado en la tarea Definición de Componentes y Subsistemas de Construcción (DSI 8.2).

Esta actividad se realiza en paralelo a las actividades Generación del Código de los Componentes y Procedimientos (CSI 2) y Ejecución de las Pruebas Unitarias (CSI 3). Sin embargo, es necesario que los componentes objeto de las pruebas de integración se hayan verificado de manera unitaria.

Dado el grado de acoplamiento que presenta el sistema a desarrollar entre los subsistemas, no se realizarán las pruebas de integración. Dado que cada uno de los componentes (tanto desde la visión de los subsistemas de

análisis como los de diseño) necesita una entrada, o bien, genera una salida que es directamente consumida por otro componente. De esta forma, al haber probado los componentes de comunicaciones en las pruebas unitarias, ya es posible afirmar que es necesario probar el sistema en su totalidad.

7.5. Ejecución de las Pruebas del Sistema

El objetivo de las pruebas del sistema es comprobar la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.

En la realización de estas pruebas es importante comprobar la cobertura de los requisitos no funcionales, dado que su incumplimiento puede comprometer la aceptación del sistema por el equipo de operación responsable de realizar las pruebas de implantación del sistema, que se llevarán a cabo en el proceso Implantación y Aceptación del Sistema (IAS).

7.5.1. Preparación del Entorno de las Pruebas del Sistema

En esta tarea se preparan todos los recursos necesarios para realizar las pruebas del sistema, de acuerdo a las características del entorno establecidas en el plan de pruebas.

Para ello se asegura la disponibilidad del entorno y de los datos necesarios para ejecutar estas pruebas, se preparan las bibliotecas o librerías que se estimen oportunas para la realización de las mismas, así como los procedimientos manuales o automáticos asociados.

En el apartado Generación de los Procedimientos de Operación y Seguridad (7.2.2) se define como preparar un entorno de uso del sistema de información. Es necesario seguir dicha guía para generar un entorno de pruebas.

7.5.2. Realización de las Pruebas del Sistema

El objetivo de esta tarea es comprobar la integración de todos los subsistemas y componentes del sistema de información de acuerdo a las verificaciones establecidas para el nivel de pruebas del sistema.

Para cada verificación establecida, se realizan las pruebas con los casos de pruebas asociados, efectuando el correspondiente análisis e

informe de los resultados y generando un registro conforme a los criterios establecidos en el plan de pruebas.

A continuación se presenta una tabla con los resultados de las pruebas:

Componente	Caso de Prueba	Resultado
<i>Ingreso al Sistema</i>	CP001	Correcto
	CP002	Correcto
	CP003	Correcto
	CP004	Correcto
	CP005	Correcto
<i>Cargar Pedido de Cotización</i>	CP001	Correcto
	CP002	Correcto
	CP003	Correcto
	CP004	Correcto
	CP005	Correcto
<i>Consultar Pedidos de Cotización</i>	CP001	Correcto
	CP002	Correcto
	CP003	Correcto
<i>Cargar Cotización</i>	CP001	Correcto
	CP002	Correcto
	CP003	Correcto
	CP004	Correcto
	CP005	Correcto
	CP006	Correcto
	CP007	Correcto
<i>Consultar Cotizaciones</i>	CP001	Correcto
<i>Generar Orden de Compra</i>	CP001	Correcto
	CP002	Correcto
	CP003	Correcto
	CP004	Correcto

	CP005	Correcto
Consultar Ordenes de Compra	CP001	Correcto

7.5.3. Evaluación del Resultado de las Pruebas del Sistema

El objetivo de esta actividad es analizar los resultados de las pruebas del sistema de información y efectuar una evaluación, dónde se recoja el grado de cumplimiento de las mismas y las acciones recomendadas, siguiendo los criterios que se hayan establecido en el plan de pruebas del sistema de información.

En este caso decimos que se ha alcanzado el nivel de calidad deseado en el software dado que las salidas encontradas en el sistema son las salidas esperadas en el plan de pruebas.

Capítulo VIII

Implantación y Aprobación del Sistema de Información

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Implantación y Aceptación
Fase de Origen:	Implantación y Aceptación del Sistema de Información
Nombre o Código del Elemento de Configuración de Software:	Implantación y Aceptación del Sistema de Información (IASI)
Versión N°: 1.0.2	Entrega N°: 03
Fecha de Última Modificación:	11/10/2002
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

8. Implantación y Aceptación del Sistema de Información

Este proceso tiene como objetivo principal la entrega y aceptación del sistema en su totalidad y la realización de todas las actividades necesarias para el paso a producción del mismo.

En primer lugar, se revisa la estrategia de implantación, su alcance y en función de sus características, se define un plan de implantación y se especifica el equipo que lo va a llevar a cabo. Conviene señalar la participación del usuario de operación en las pruebas de implantación, del usuario final en las pruebas de aceptación.

Las actividades previas al inicio de la producción incluyen la preparación de la infraestructura necesaria para configurar el entorno, la instalación de los componentes, la activación de los procedimientos manuales y automáticos asociados y, cuando proceda, la migración o carga inicial de datos. Para ello se tomarán como punto de partida los productos software probados, obtenidos en el proceso Construcción del Sistema de Información (CSI) con su documentación asociada.

Se realizan las pruebas de implantación y de aceptación del sistema en su globalidad que responden a los siguientes propósitos:

Las pruebas de implantación cubren un rango muy amplio, que va desde la comprobación de cualquier detalle de diseño interno hasta aspectos tales como las comunicaciones. Se debe comprobar que el sistema puede gestionar los volúmenes de información requeridos, se ajusta a los tiempos de respuesta deseados y que los procedimientos de respaldo, seguridad e interfaces con otros sistemas funcionan correctamente. Se debe comprobar el comportamiento del sistema bajo las condiciones más extremas.

Las pruebas de aceptación se realizan por y para los usuarios y tienen como objetivo validar formalmente que el sistema se ajusta a sus necesidades.

Asimismo, se llevarán a cabo las tareas necesarias para la preparación del mantenimiento siempre y cuando se haya decidido que los sistemas de información implicados en la implantación van a ser objeto de mantenimiento. En cualquier caso, es necesario que la persona que vaya a asumir el mantenimiento conozca el sistema, antes de su incorporación al entorno de producción. Este no es el caso en cuanto al presente proyecto, dado que el mismo no tendrá una opción de mantenimiento, al menos, en el marco del presente proyecto de tesis.

Además se determinan los servicios que requiere el sistema que se va a implantar, especificando los niveles de servicio y el acuerdo que se adquiere una vez que se inicie la producción. Se distinguen los servicios de gestión de operaciones (servicios por lotes, seguridad, comunicaciones, etc.) y los servicios al cliente (servicio de atención a usuario, mantenimiento, etc.) que se deberán negociar en cuanto a recursos, horarios, coste, etc. Se fija el nivel con el que se prestará el servicio como indicador de la calidad del mismo.

Conviene señalar que la implantación puede ser un proceso iterativo que se realiza de acuerdo al plan que se establezca para el comienzo de la producción del sistema en su entorno de operación. Para establecer este plan se tendrá en cuenta:

- El cumplimiento de los requisitos de implantación definidos en la actividad
- Establecimiento de Requisitos y especificados en la actividad
- Establecimiento de Requisitos de Implantación.
- La estrategia de transición del sistema antiguo al nuevo: nuevamente, en el presente desarrollo no se encuentra un sistema legado desde el cual migrar, con lo cual estas definiciones no se presentan en los apartados posteriores.

Finalmente se realizan las acciones que sean necesarias para el inicio de la producción.

8.1. Establecimiento del Plan de Implantación

Se identifican los distintos sistemas de información que forman parte del sistema objeto de la implantación y se analizan, para cada uno de ellos, las posibles dependencias con otros proyectos que puedan condicionar el plan de implantación. En el caso del sistema de mercados virtuales, no se verifica dependencia alguna con otros proyectos de sistemas.

Una vez estudiado el alcance de la implantación y sus condicionantes, se decide si se puede llevar a cabo la implantación del sistema estableciendo, en su caso, la estrategia de forma definitiva que se concretará en el plan de implantación.

Se constituye el equipo de implantación determinando los recursos humanos necesarios para la propia instalación del sistema, para las pruebas de implantación y aceptación, y para la preparación del mantenimiento. Se identifican para cada uno de ellos, sus perfiles y niveles de responsabilidad.

8.1.1. Definición del Plan de Implantación

La estrategia de implantación del sistema se determina en base a la envergadura del sistema, es decir el número de sistemas de información implicados en la implantación y la cobertura geográfica, cuyo alcance dependerá de las características y complejidad de los sistemas de información que conforman el sistema objeto de la implantación.

En el caso del presente proyecto, el sistema se encontrará distribuido sobre la Internet. Si bien, los usuarios del sistema no formarán parte de una corporación donde sea posible educar formalmente a los usuarios para el uso del sistema, para este momento esta actividad sí es posible. Esto se debe a que como objetivo de este proyecto de tesis se decide formar al usuario líder (aquel que participó con el equipo de desarrollo a lo largo del proyecto) para realizar la prueba de aceptación del usuario.

Es por ello que la estrategia seleccionada consiste en capacitarlo en los distintos roles de usuario que se encuentran dentro del sistema. De esta forma, cuando nuevos usuarios aparezcan, el usuario líder podrá entrenarlos sin inconvenientes.

En esta actividad también se revisan los requisitos de implantación (instalación, infraestructura, formación) establecidos en la tarea Especificación de Requisitos de Implantación y los procedimientos implicados en la implantación establecidos para cada uno de los sistemas de información

en la tarea Especificación de Requisitos de Operación y Seguridad, con el fin de asegurar su adecuación a la estrategia global de implantación.

Una vez analizada la información anterior se define un plan de implantación que permita dimensionar adecuadamente el esfuerzo y los recursos necesarios para llevar a cabo con éxito la implantación.

A continuación se analizan los distintos aspectos que debe tener en cuenta el plan:

8.1.1.1. Formación necesaria para la implantación tanto a usuarios finales como al equipo que se encargará de realizar las pruebas de implantación y aceptación del sistema

Se prevé capacitar al usuario líder en el uso del sistema de información de forma tal de que pueda utilizarlo para verificar que el mismo cumple con sus requisitos para posteriormente aceptar el sistema. Para ello será necesario entregar la documentación relacionada para que el mismo se interiorice sobre el sistema y la tecnología asociada al mismo. Por otro lado se prevé generar material relacionado al manual de uso (40 horas * hombre) y 4 charlas de 2 horas sobre el sistema funcionando.

Será necesario generar las cuentas de prueba asociadas a dicho fin.

8.1.1.2. Preparación de la infraestructura necesaria para la incorporación del sistema al entorno de operación

Como ya se ha indicado durante el diseño del sistema y la construcción del mismo, será necesario instalar ciertos componentes de software y la aplicación en sí. Además, es necesario configurar dichos componentes para el correcto funcionamiento de la aplicación.

Entre las tareas a desarrollar encontramos las siguientes:

Tarea	Rol	Esfuerzo
Implantación del servidor de aplicaciones necesario para ejecutar los componentes servidores de la aplicación.	Administrador de Aplicación e Infraestructura	1 hora
Implantación de la base de datos que necesita la aplicación.	Administrador de Base de Datos	1 hora
Instalación de los componentes binarios de la aplicación.	Administrador de Aplicación e Infraestructura	1 horas

Instalación de los componentes cliente de la aplicación.	Administrador de Aplicación e Infraestructura	1 hora
Instalación y verificación del correcto funcionamiento de las cuestiones de seguridad y comunicaciones.	Administrador de Seguridad y Comunicaciones	3 horas

8.1.1.3. Instalación de todos los componentes y procedimientos manuales y automáticos asociados a cada sistema de información implicado en la implantación

No se presentan componentes específicos como agentes o demonios que deban ser instalados y configurados.

8.1.1.4. Ejecución de los procedimientos de carga inicial y migración de datos

Para que el usuario pueda probar el sistema de información, será necesario hacer una carga inicial con productos de prueba.

Tarea	Rol	Esfuerzo
Ejecución de los procedimientos de carga inicial	Administrador de Base de Datos	1 hora

8.1.1.5. Realización de las pruebas de implantación y aceptación del sistema

Se necesitan generar las cuentas del sistema relacionadas a los perfiles de comprador y vendedor.

Tarea	Rol	Esfuerzo
Creación de cuentas del sistema	Administrador de Aplicación e Infraestructura	1 hora

8.1.1.6. Formalización del plan de mantenimiento.

La etapa de mantenimiento del sistema de información excede los límites del proyecto de tesis. De todas formas se indican ciertas cuestiones a tener en cuenta para la explotación del sistema de mercados virtuales.

Concepto	Explicación
Espacio de Almacenamiento	Las bases de datos relacionales asignan cierto espacio en disco para guardar los datos de sus tablas. Es necesario revisar periódicamente el espacio asignado, la cantidad de espacio disponible y la cantidad de espacio disponible en el sistema de archivos donde se encuentra instalado el sistema para asegurarse de que se cuenta con el necesario para el correcto funcionamiento del sistema.
Performance del Motor de Base de Datos	El sistema ha sido diseñado de forma tal de que la mayor cantidad de consultas que se ejecutan sobre el motor de gestión de bases de datos utilicen las claves primarias (IDs) de las tablas. Estas claves siempre están indexadas con lo cual se asegura un uso óptimo de los recursos del sistema. De todas formas y dependiendo la carga que tenga la aplicación, puede resultar necesario verificar el log de transacciones del gestor para verificar un tiempo de respuesta aceptable para todas las consultas del sistema.
Accesos Indevidos al Sistema	El sistema almacena una bitácora de errores y problemas detectados. Resulta necesario revisar periódicamente dicho registro para identificar la posible ocurrencia de problemas como intentos de ingreso al sistema (ingreso de clave incorrecta, uso de usuarios y claves inexistentes, etc.) de forma tal de poder aplicar contramedidas a tiempo.
Control de las Copias de Seguridad	Las copias de seguridad son el único recurso para reestablecer el sistema en caso de una caída del mismo. Es por ello importante verificar que las mismas se realizan correctamente para lo cual se aconseja recuperarlas periódicamente en otro equipo y probar que el sistema se encuentra disponible así como la consistencia de los datos.

8.1.1.7. Específico para sistemas de Internet

Es necesario destacar que los sistemas distribuidos sobre la Internet, suelen por lo general, tener usuarios distribuidos alrededor de amplias zonas

geográficas, ya que precisamente esa es una de las grandes ventajas de un sistema de estas características.

Se puede hacer una categorización de estos sistemas en base a la topología que posee el dominio de los usuarios:

Dominio de usuarios conocido

En muchas organizaciones se desarrollan sistemas de tecnología de Internet dado que posee varias sedes distribuidas y se quiere lograr que todas posean el mismo sistema.

Un buen ejemplo puede ser una compañía telefónica que posee distintas sedes en toda una región. Si los sistemas necesarios para operar una nueva sede se basan en esta tecnología, abrir una nueva sede (desde el punto de vista de sistemas) simplemente consiste en instalar un navegador en cada equipo (muchos sistemas operativos ya lo traen) y generar las cuentas de usuarios en el sistema. Esto se traduce en una ventaja competitiva para la organización: es muy rápido y poco costoso abrir una nueva sede.

En este caso, el dominio de usuarios es conocido. Esto se debe a que la gente que utilizará el sistema tiene un rol bien definido a nivel negocio, y es posible planificar su ingreso a la organización, por lo tanto, es posible darle formación formal sobre el uso del sistema.

Para este tipo de situaciones, se suele utilizar el enfoque del usuario líder al cual se lo entrena formalmente sobre el uso del sistema para que con posterioridad se encargue de la formación de los nuevos usuarios. Inclusive la documentación del sistema (manual del usuario) puede ser desarrollado por este usuario y contenida en un formato estándar de documento o manual impreso.

Este es el caso del actual proyecto de mercado virtual.

Dominio de usuarios desconocido

En otros casos, el dominio de usuarios es desconocido. Este punto se ejemplifica en los sistemas distribuidos sobre Internet donde se brinda un servicio al público pero se desconoce el perfil y conocimiento que cada usuario posee.

Un buen ejemplo en un sistema distribuido sobre Internet donde los clientes de la compañía telefónica antes mencionada, pueden ingresar para

consultar información sobre su consumo de línea telefónica y autogestionarla (Ej.: habilitar nuevos servicios ó configurar otros).

En este caso, se suele utilizar otro enfoque para la capacitación del usuario. En muchos casos, estos sistemas son utilizados por gente que se conecta en horarios marginales inclusive, dado que los sistemas de Internet se encuentran disponibles las 24 horas. Es por esto necesario poseer una mesa de ayuda para apoyar a los usuarios y contar con herramientas electrónicas de capacitación (en la jerga e-learning).

Existen productos comerciales y compañías dedicadas a proveer soluciones de aprendizaje electrónico, como por ejemplo :

En estos casos se presentan tutoriales y herramientas de ayuda como bases de conocimientos, incidencias, chat con soporte técnico y foros de discusión general donde el usuario puede depositar sugerencias, preguntas y leer casos anteriores en busca de posibles soluciones para su problema.

8.1.2. Especificación del Equipo de Implantación

Se constituye el equipo de trabajo necesario para llevar a cabo la implantación y aceptación del sistema, según el plan de implantación establecido en la tarea anterior.

Para ello, se identifican en función del nivel de esfuerzo requerido, los distintos participantes implicados en la implantación del sistema (usuarios, equipo técnico y responsable de mantenimiento), determinando previamente sus perfiles, responsabilidades, nivel de implicación y fechas previstas de participación a lo largo de toda la implantación.

A continuación se presenta una tabla que define los distintos perfiles involucrados en la implantación, su responsabilidad y su dedicación:

Rol	Perfil
Usuario Vendedor	Es el usuario que hará las veces de vendedor dentro del sistema. Dado que el mismo se encontrará disponible en forma remota, el plan de capacitación comprende la formación de un único usuario líder, encargado de entrenar a los sucesivos usuarios del sistema. Este usuario será el que ha participado con el equipo de proyecto a lo largo del desarrollo del sistema.

<i>Usuario Comprador</i>	Es el usuario que hace las veces de comprador de productos dentro del sistema.
<i>Administrador de Base de Datos</i>	Es un miembro del equipo de desarrollo que instalará y configurará los componentes de software necesarios para las pruebas de aceptación.
<i>Administrador de Seguridad y Comunicaciones</i>	Es un miembro del equipo de desarrollo encargado de verificar e instalar cuestiones específicas relacionadas con las comunicaciones y la seguridad del sistema.
<i>Administrador de Aplicación e Infraestructura</i>	Es un miembro del equipo de proyecto con conocimiento de sistemas operativos e infraestructura de software y hardware de Internet.

La dedicación de los distintos perfiles ya ha sido definida en la actividad anterior.

8.2. Formación Necesaria para la Implantación

En esta actividad se prepara y se imparte la formación al equipo que participará en la implantación y aceptación del sistema y se realiza el seguimiento de la formación de usuarios finales, cuya impartición queda fuera del ámbito de Métrica V.3. De esta formase asegura que la implantación se llevará a cabo correctamente.

Se determina la formación que va a ser necesaria para el equipo de implantación, en función de los distintos perfiles y niveles de responsabilidad identificados en la actividad anterior. Para ello, se establece un plan de formación que incluye los esquemas de formación correspondientes, los recursos humanos y de infraestructura requeridos para llevarlo a cabo, así como una planificación que quedará reflejada en el plan de formación. En el presente proyecto, los distintos roles necesarios para la implantación serán conformados por el usuario líder al cual se entrenará para poder llevar a cabo todas las actividades.

La formación para que los usuarios finales sean capaces de utilizar el sistema de forma satisfactoria ha sido establecida, previamente, en la actividad Definición de la Formación de Usuarios Finales (CSI 7).

8.3. Incorporación del Sistema al Entorno de Operación

En esta actividad se realizan todas las tareas necesarias para la incorporación del sistema al entorno de operación en el que se van a llevar a cabo las pruebas de implantación y aceptación del sistema.

Mientras que las pruebas unitarias, de integración y del sistema se pueden ejecutar en un entorno distinto de aquél en el que finalmente se implantará, las pruebas de implantación y aceptación del sistema deben ejecutarse en el entorno real de operación.

El propósito es comprobar, que el sistema, satisface todos los requisitos funcionales y no funcionales especificados por el usuario en las mismas condiciones que cuando se inicie la producción.

Por tanto, como paso previo a la realización de dichas pruebas y de acuerdo al plan de implantación establecido, se verifica que están disponibles todos los recursos necesarios para que se pueda realizar, adecuadamente, la instalación de todos los componentes que integran el sistema, así como la creación y puesta a punto de las bases de datos en el entorno de operación. Asimismo, se establecen los procedimientos de explotación y uso de las bases de datos de acuerdo a la normativa existente en dicho entorno.

8.3.1. Preparación de la Instalación

En esta tarea se verifica que esté disponible toda la infraestructura necesaria para configurar el entorno de acuerdo a los requisitos de implantación (instalación e infraestructura) y teniendo en cuenta los procedimientos de seguridad y control de acceso (mantenimiento de la integridad y confidencialidad de los datos, control de accesos al sistema, copias de seguridad y recuperación de datos, etc.), y operación y administración del sistema (estándares, recuperación y reanudación de trabajos, planificación de trabajos, etc.).

Una vez comprobada la idoneidad de los distintos elementos relacionados con la infraestructura, se realiza la instalación del software de base necesario para la incorporación posterior de los componentes asociados a los sistemas de información implicados en la implantación.

Dentro del contexto del presente proyecto de tesis, se deberá proceder a instalar el software indicado en la tabla a continuación:

Paquete de Software	Versión
RedHat Linux	7.2
MySQL Server	3.2
Java Development Kit	1.3.1

8.3.2. Realización de la Instalación

Teniendo en cuenta los estándares y normativas por los que se rige la organización en los entornos de operación y de acuerdo al plan de implantación establecido (ver ANEXO I – Instalación y Operación del Sistema), se realiza la instalación de todos los componentes del nuevo sistema de acuerdo a su ubicación física, establecida en el proceso Diseño del Sistema de Información (DSI), incluidos los procedimientos manuales y automáticos.

Asimismo, se prepara el entorno de datos identificando los sistemas de información que forman parte del sistema objeto de la implantación y, para cada uno de ellos:

- Se crean las bases de datos a partir del esquema físico elaborado en el proceso de construcción. Para ello es necesario ejecutar las rutinas de creación de tablas definidas en el capítulo

- Se establecen los procedimientos de explotación y uso de las bases de datos, es decir, la normativa necesaria para la utilización de las bases de datos, actualización, consulta, etc.

- Se revisan los procedimientos necesarios para realizar las copias de seguridad de los datos y de restauración de las copias indicando su frecuencia, así como los procedimientos de consolidación y sincronización de la información, éstos últimos cuando proceda

- Se preparan las autorizaciones de acceso a los datos para los distintos perfiles de usuarios.

Una vez comprobada la correcta instalación del nuevo sistema, se activan los procedimientos de operación, de administración del sistema, de seguridad y de control de acceso, que incluirán el arranque y cierre del sistema según la frecuencia establecida, planificación de trabajos,

recuperación y reanudación de trabajos, autorizaciones de acceso al sistema según los distintos perfiles de usuario, etc.

Además, si es necesaria una migración de datos se activarán también los procedimientos asociados.

8.4. Carga de Datos al Entorno de Operación

Teniendo en cuenta que los sistemas de información que forman parte del sistema a implantar pueden mejorar, ampliar o sustituir a otros ya existentes en la organización, será necesaria una carga inicial y/o una migración de datos cuyo alcance dependerá de las características y cobertura de cada sistema de información implicado. Por tanto, la necesidad de una migración de datos puede venir determinada desde el proceso Estudio de Viabilidad del Sistema (EVS), en la actividad Selección de la Solución (EVS 6), dónde se habrá establecido la estrategia a seguir en la sustitución, evaluando las opciones del enfoque de desarrollo e instalación más apropiados para llevarlo a cabo.

En cualquier caso, en la actividad Diseño de la Migración y Carga Inicial de Datos (DSI 9) se habrán definido y planificado los procesos y procedimientos necesarios para llevar a cabo la migración, realizándose su codificación en la actividad Construcción de los Componentes y Procedimientos de Migración y Carga Inicial de Datos (CSI 8).

Estas actividades se desarrollan con éxito dejando el software correctamente instalado en el ambiente productivo. No se presentan incidencias.

8.5. Pruebas de Implantación del Sistema

Se realizan las pruebas de implantación del sistema con el fin de comprobar el funcionamiento correcto del mismo en el entorno de operación y permitir al usuario, desde el punto de vista de operación, que determine la aceptación del sistema una vez instalado en su entorno real, según el cumplimiento de los requisitos no funcionales especificados.

Para ello, el responsable de implantación revisa el plan de pruebas de implantación y los criterios de aceptación del sistema, que ha elaborado previamente. Estas pruebas son realizadas por el usuario líder.

Una vez ejecutadas estas pruebas, el equipo de usuarios técnicos informa de las incidencias detectadas al responsable de implantación, el cual analiza la información y toma las medidas correctivas que considere necesarias para que el sistema dé respuesta a las especificaciones previstas, momento en el que operación lo da por probado.

Dada la naturaleza del presente proyecto, el sistema es probado nuevamente por el mismo usuario líder y ejecutando el mismo plan de pruebas utilizado para las pruebas anteriores.

8.5.1. Preparación de las Pruebas de Implantación

Se comprueba la disponibilidad de los recursos humanos y técnicos necesarios para realizar las pruebas de implantación y se revisan las verificaciones establecidas en el plan de pruebas. No se presentan incidencias en este caso.

Si fuera necesario, se crea algún caso de prueba adicional que se considere importante y que no se haya tenido en cuenta hasta entonces y se preparan las condiciones que permitan simular las situaciones límite previstas para las pruebas.

Se comunica el plan de pruebas de implantación al equipo responsable de llevarlas a cabo.

En conclusión, se utilizará el mismo plan de pruebas utilizado en las pruebas anteriores para verificar el correcto funcionamiento del sistema de mercados virtuales en el entorno de implantación.

8.5.2. Realización de las Pruebas de implantación

Se realizan las pruebas de implantación, de acuerdo a las verificaciones establecidas en el plan de pruebas definido en la actividad Especificación Técnica del Plan de Pruebas (DSI 10).

El usuario líder ejecuta nuevamente los casos de prueba del sistema sobre el entorno productivo y no se registran incidentes.

8.5.3. Evaluación del Resultado de las Pruebas de Implantación

El usuario líder no reporta anomalías en el funcionamiento del sistema con lo cual se dan por aprobadas las pruebas de implantación del sistema.

8.6. Pruebas de Aceptación del Sistema

Las pruebas de aceptación se llevan a cabo con el fin de validar que el sistema cumple los requisitos básicos de funcionamiento esperado y permitir al usuario que determine la aceptación del sistema. Por este motivo, estas pruebas son realizadas por el usuario final del sistema (en este caso el usuario líder) y es durante este periodo de tiempo, cuando debe plantear todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente.

El responsable del grupo de usuarios revisa los criterios de aceptación que se especificaron previamente en el plan de pruebas del sistema y dirige las pruebas de aceptación final que llevan a cabo los usuarios a los que corresponda. A su vez, éstos últimos deben elaborar un informe que el responsable del grupo de usuarios analiza y evalúa con el fin de determinar la aceptación o rechazo del sistema.

Estas pruebas fueron realizadas por el mismo usuario que acompañó el desarrollo del sistema, el mismo usuario que realizó las pruebas de implantación. No se registran incidencias con lo cual se da por aceptado el sistema por parte del usuario líder. El sistema ahora deberá ser usado por el tutor de tesis ya que su aprobación final define la aceptación del sistema en carácter definitivo.

8.7. Establecimiento del Acuerdo de Nivel de Servicio

Es conveniente que antes de la aprobación definitiva del sistema por parte del Comité de Dirección, se determinen los servicios que requiere el mismo, se especifiquen los niveles de servicio con los que se valorará la calidad de ese servicio y qué compromisos se adquieren con la entrega del sistema.

Para ello, en primer lugar, se negocian entre los máximos responsables del usuario y de operación, qué servicios y de qué tipo se van a prestar. Una vez acordados, se detallan los niveles de servicio definiendo sus propiedades funcionales y de calidad, y se establecen cuáles de ellas son cuantificables y qué indicadores se van a aplicar. Es importante señalar que los niveles de servicio son específicos para cada sistema de información que compone el sistema, y dependen del entorno de operación y de la localización geográfica en que se implante un sistema de información

concreto, pudiendo haber servicios básicos para todo el sistema o específicos para un sistema de información concreto.

Por último, se establece formalmente el acuerdo de nivel de servicio, considerando los recursos necesarios, plazos de restablecimiento del servicio, coste y mecanismos de regulación que están asociados a cada servicio especificado anteriormente.

Según el ámbito y el alcance de los tipos de servicio que se vayan a prestar, se determinarán los productos del ciclo de vida del software que son necesarios para poder establecer el acuerdo de nivel de servicio.

8.7.1. Identificación de los Servicios

Se identifican los tipos de servicio que requiere el sistema objeto de la implantación, en función de los sistemas de información que componen el sistema y su localización geográfica. También se consideran los requisitos no funcionales de los sistemas de información afectados por la implantación. Entre los tipos de servicio se pueden distinguir los servicios al cliente (servicio de atención a usuario, etc.) y servicios de gestión de operaciones.

Servicio	Nivel de Servicio
<i>Servicios en línea</i>	El sistema estará disponible durante 23 horas al día. Deberá ser dado de baja durante el lapso de una hora para realizar las copias de seguridad (esto se debe a la carencia de herramientas de uso corporativo que permiten realizar copias de seguridad en tiempo real). Fuera de este punto totalmente dependiente de las características provistas por el software seleccionado, el sistema estará disponible y con un tiempo de respuesta tal como se define en el diseño del sistema de información.
<i>Servicios por lotes</i>	El sistema de mercados virtuales no presenta procesos por lotes.
<i>Comunicaciones</i>	El sistema es ejecutado dentro de la red de área local disponible con lo cual no se

	presentan mayores inconvenientes de conectividad. En caso de presentarse una implementación sobre la Internet del sistema, deberá contratarse un servicio de alta disponibilidad y un servicio de contingencia (por lo general acceso remoto por módem) para asegurar que el sistema se encontrará disponible todo el tiempo.
Seguridad	En un entorno de producción, los equipos y firewalls cuentan con herramientas de administración y monitoreo del esquema de seguridad. Estas herramientas no se encuentran disponibles para el presente proyecto de tesis. De todas formas ya se ha indicado en apartados anteriores la necesidad de verificar el correcto funcionamiento del esquema de copias de seguridad y recuperación.

8.8. Presentación y Aprobación del Sistema

Una vez que se han llevado a cabo las pruebas de implantación y de aceptación y se ha fijado el acuerdo de nivel de servicio, el Comité de Dirección debe formalizar la aprobación del sistema. Para el caso del presente trabajo, la presentación y aprobación final del sistema consiste en la defensa del trabajo de tesis ante el tribunal oportunamente reunido a tal fin.

Capítulo IX

Bibliografía y Glosarios

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Bibliografía y Glosarios
Fase de Origen:	Bibliografía y Glosarios
Nombre o Código del Elemento de Configuración de Software:	Bibliografía y Glosarios
Versión N°: 1.0.1	Entrega N°: 03
Fecha de Última Modificación:	11/10/2002
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

9. Bibliografía y Glosarios

9.1. Bibliografía

- [Blaha 1998] "Object-Oriented Modeling and Design for Database Applications", Prentice Hall, Michael Blaha, William Premerlani, James Rumbaugh, 1998
- [Conallen 2000] "Building Web Applications with UML", Jim Conallen, Booch - Jacobson - Rumbaugh Series, Addison Wesley, 2000
- [Gamma 1994] "Design Patterns, Elements of Reusable Object-Oriented Software", Addison Wesley, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 1994.
- [Gartner 2001a] Group, "Evolution From e-Business to Business as Usual", Gartner Group, 2001.
- [Gartner 2001b] Group, "Habilitación de la Colaboración en la Cadena de Provisión", Gartner Group, 2001.
- [Gartner 2001c] Group, "El escenario de las tendencias de la industria: extendiendo la empresa a través del uso de la tecnología informática", Gartner Group, 2001.

Bibliografía y Glosarios

BG Versión 1.0.1

- [Jacobson 1992] "Object Oriented Software Engineering", Addison-Wesley, Jacobson, 1992.
- [Jacobson 1999] "The Unified Software Development Process", Addison Wesley, Jacobson, Rumbaugh, Booch, 1999.
- [Longstreet 2001] "Use Cases and Function Points", Longstreet Consulting Inc, David Longstreet.
<http://www.softwaremetrics.com/Articles/usecases.htm>
- [Métrica 1999] "Métrica Versión 3", Consejo Superior de Informática, 1999.
- [Sun Microsystems 1999a] "JAVA Look And Feel Design Guidelines", versión 1.0.2, Sun Microsystems, 1999.
<http://java.sun.com/products/jlf/ed1/dg/index.htm>
- [Sun Microsystems 1999b] "Java 2 Enterprise Edition", Sun Microsystems, 1999
- [W3Ca] "Hiper Text Transfer Protocol HTTP1/1", World Wide Web Consortium, www.w3c.org/Protocols/HTTP, 1999.
- [W3Cb] "Hiper Text Markup Language HTML 4.01", World Wide Web Consortium, www.w3c.org/Protocols/HTML, 1999.

9.2. Glosario de Abreviaturas

ACID	Atomicidad Consistencia Encapsulamiento Durabilidad
HTML	Hiper Text Markup Language (Lenguaje de denotación de hiper texto).
HTTP	Hiper Text Transfer Protocol (Protocolo de Transferencia de Hiper Texto).
IFPUG	International Function Point User Group (Grupo de Usuarios Internacional de Puntos de Función).
SOAP	Single Object Access Protocol (Protocolo Simple de Acceso a Objetos).
TCP/IP	Transport Control Protocol (Protocolo de Control de Transporte) / Internet Protocol (Protocolo de Internet).
W3C	World Wide Web Consortium (Consortio de Internet).
XML	eXtended Markup Language (Lenguaje de Denotación Extendido).

9.3. Glosario de Términos

e-Business	Cualquier proceso de negocio realizado mediante el uso de una red de computadoras.
e-Commerce	Proceso que involucra comprar y vender utilizando una red de computadoras.
e-Marketplace	Portal que permite a una comunidad de usuarios interactuar en una variedad de actividades comerciales.
Internet	Red mundial de computadoras.
Portal	Sitio en Internet que concentra información sobre un tópico específico.
Pantalla Modal de Sistema	Pantalla que conforma la interfaz gráfica del usuario que no permite utilizar otra pantalla hasta que esta se encuentre cerrada.
Pantalla Modal de Aplicación	Idem a Pantalla Modal del Sistema pero permite utilizar otras pantallas mientras esta se encuentra abierta.
Sitio de Internet	Conjunto de recursos accesibles por medio de un server de Internet.
Aplicación de Internet	Sitio de Internet donde el usuario puede modificar el estado de la lógica de negocios en el servidor utilizando un navegador.
Componente	Entidad de software auto - comprendida con una interfaz pública.

Anexo I

Implantación y Operación del Sistema de Información

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Implantación y Operación
Fase de Origen:	Diseño del Sistema de Información
Nombre o Código del Elemento de Configuración de Software:	Implantación y Operación del Sistema de Información (ANEXO I)
Versión N°: 1.0.0	Entrega N°: 01
Fecha de Última Modificación:	11/10/2002
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

10. ANEXO I - Instalación y Operación del Sistema de Información

En este anexo se describen los pasos necesarios para la instalación del sistema de información y la preparación de la infraestructura necesaria para su operación.

10.1. Instalación del Sistema

La instalación del sistema comprende las siguientes etapas:

- Instalación de la base de datos.
- Generación del esquema de base de datos.
- Instalación de los binarios del sistema.
- Instalación del componente cliente.

En los apartados contiguos se especifican los pasos que comprende cada una de estas etapas.

10.1.1. Instalación de la Base de Datos

En esta tarea se crean los elementos del sistema gestor de base de datos, se inicializa la base de datos y se cargan los datos que se consideren necesarios en el espacio de almacenamiento previamente definido.

En este proyecto esta tarea consiste de instalar la base de datos MySQL y correr las sentencias de DDL que se indican en el capítulo del diseño del sistema.

La instalación de la base de datos consiste en los siguientes pasos

Bajar el archivo de instalación `mysql-3.23.35a-win.zip` desde la dirección `www.mysql.com`.

Descomprimir el archivo de instalación y ejecutar el programa `setup.exe`

Seguir los pasos planteados por el asistente de instalación (configuración por defecto).

Una vez instalada la base, es necesario levantar la base por medio de la siguiente instrucción: `<directorio de instalación>\bin\mysqladmin start`

10.1.2. Generación del Esquema de Base de Datos

Una vez instalado el software hay que ejecutar el programa de interfaz de comandos ubicado en `<directorio de instalación>\bin\mysql`

Luego, es necesario crear una base de datos para almacenar las tablas del sistema. Esto se logra ejecutando la instrucción: `create database mv`

Es necesario indicar que de ahora en más se utilizará dicha base de datos con la instrucción: `use mv`

Ahora es necesario ejecutar todas las instrucciones de DDL que se especifica en el diseño de sistemas de información.

La base de datos ya esta configurada y lista para ser usada.

10.1.3. Instalación de los Binarios del Sistema

Para que el sistema pueda ser utilizado, es necesario iniciar los componentes del mismo. Al terminar los pasos descritos en el apartado anterior, la base de datos se encuentra creada y lista para ser utilizada.

Ahora es el momento de iniciar los componentes en el servidor. Para ello se deben descomprimir los archivos provistos en un archivo `emarket.tar` utilizando la siguiente sentencia: `tar xvf emarket.tar`

En este momento los binarios del sistema ya se encuentran instalados.

10.2. Operación del Sistema de Información

Para operar el sistema es necesario iniciar los componentes en el servidor. Iniciar el sistema en el servidor consiste en ejecutar los scripts de instalación en el siguiente orden:

```
listener
```

Y en otra ventana de terminar

```
startserver
```

En este momento el sistema se encuentra listo para iniciar las operaciones.

10.3. Instalación del Componente Cliente

El componente cliente se encuentra representado por una cantidad de archivos JAR (Java ARchieve) y un archivo de comandos que inicia el cliente. Además se provee de un archivo de configuración denominado `server.properties` el cual debe ser configurado para el correcto uso del sistema.

En este archivo se encuentran las siguientes entradas:

```
host=127.0.0.1
```

```
port=5001
```

En la primer línea se indica el nombre del servidor donde se encuentran funcionando los componentes del servidor del mercado electrónico. En el ejemplo se asume que el servidor se encuentra en el mismo equipo.

La segunda línea indica el puerto de comunicaciones donde se encuentra escuchando el componente servidor. El valor por defecto es 5001 y debería dejarse así salvo que este valor produzca algún tipo de conflicto con otro servicio.

Una vez correctamente configurado el archivo `server.properties`, se puede iniciar el cliente simplemente con el comando: `emarket`

Anexo II

Manual del Usuario del Sistema de Información

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Manuales de Usuario
Fase de Origen:	Diseño del Sistema de Información
Nombre o Código del Elemento de Configuración de Software:	Manual del Usuario del Sistema de Información (ANEXO II)
Versión N°: 1.0.1	Entrega N°: 02
Fecha de Última Modificación:	14/02/2003
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

11. ANEXO II - Manual del Usuario del Sistema

Este anexo es un breve manual de usuario que permite al principiante introducirse en las cuestiones propias al sistema de mercado electrónico en forma breve.

El sistema implementa la funcionalidad necesaria para soportar un ciclo completo de compras. Desde la solicitud de un pedido de cotización hasta la colocación de una orden de compra en una forma 100% digital e instantánea.

A continuación se detalla en cada apartado la forma de utilización de cada una de las funcionalidades. Cabe destacar que la disponibilidad de las distintas funcionalidades se encuentra relacionada con el rol con el cual el usuario fue dado oportunamente de alta dentro del sistema por parte del administrador.

11.1. Ingreso al Sistema

Como todo sistema de información, lo primero que el usuario debe realizar es identificarse frente al sistema. Una vez iniciado el cliente por medio del comando emarket, aparecerá la siguiente pantalla de ingreso al sistema:

ANEXO II - Manual del Usuario
ANEXO II - Versión 1.0.1



En esta pantalla el usuario debe ingresar su nombre de usuario y su clave de acceso para luego presionar el botón aceptar.

De encontrarse algún error el sistema lo notificará mediante una pantalla de error. De lo contrario, aparecerá una pantalla con el menú principal del sistema con las opciones propias del perfil asociado al usuario (comprador o vendedor en el mercado virtual).



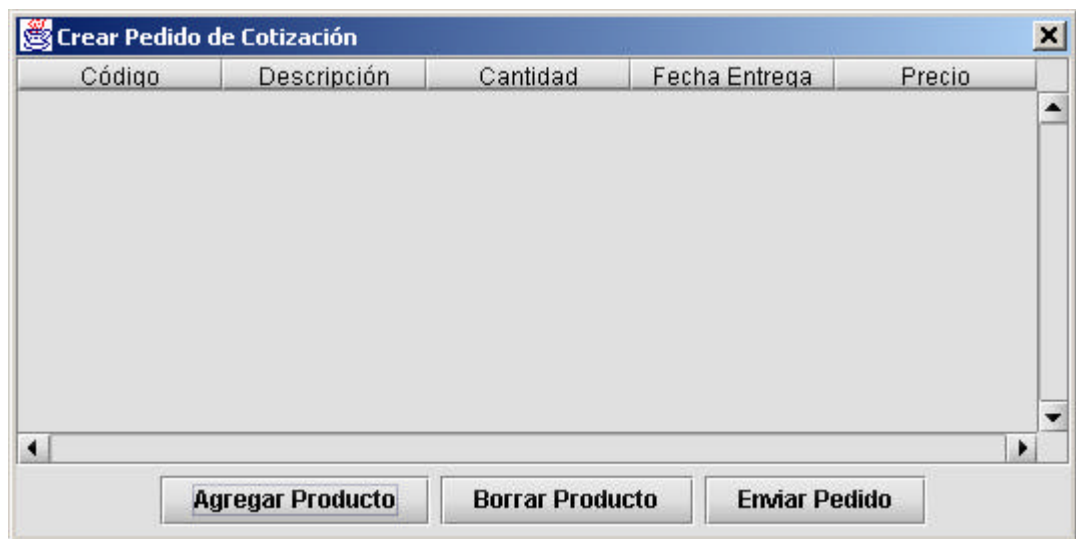
En este caso el sistema muestra el usuario con el cual se ingresó al sistema y su perfil con las opciones asociadas a un comprador dentro del mercado virtual.

Para terminar las operaciones bastará con presionar el botón "Salir del Sistema"

11.2. Creación de un Pedido de Cotización

Para iniciar el ciclo de ventas, el comprador de materiales eléctricos debe realizar un pedido de cotización. En este caso el comprador carga los productos de los cuales necesita una cotización.

Al seleccionar la opción "Crear Pedido de Cotización" del menú principal, aparecerá la siguiente pantalla:



En esta pantalla es posible ver la lista de productos seleccionados para crear la cotización. Para cargar un producto, el usuario debe presionar el botón "Agregar Producto" y el sistema responderá con la pantalla a continuación:

Agregar Item

Producto: <SIN SELECCION> ...

Cantidad:

Fecha de Entrega: 03/10/2002 ...

Precio Estimado:

Aceptar Cancelar

Esta pantalla permite la carga de los datos relacionados a un ítem de un pedido de cotización. Lo primero que el usuario debe hacer es seleccionar un producto presionando el botón de elipsis ("... ") próximo al campo producto.

En este momento el sistema abrirá la pantalla de búsqueda de productos que se describe a continuación:

Buscar Producto

Por Código Por Descripción

Descripción: LAM Buscar

Código	Descripción
BB01	LAMPARA 100W
BB02	LAMPARA 75W

Ver Todos Seleccionar Cancelar

Esta pantalla se encuentra especialmente diseñada para buscar productos dentro del catálogo que provee el sistema. Las búsquedas pueden ser realizadas bien por el código de producto o por su descripción. En la figura se muestra una búsqueda de todos los artículos cuya descripción contiene "LAM" y se muestran los productos encontrados.

Para seleccionar un producto, es necesario hacerle click en la lista con el mouse y presionar el botón "Seleccionar".

A continuación el sistema vuelve a la pantalla anterior con el siguiente resultado:



The image shows a Windows-style dialog box titled "Agregar Item". It contains the following fields and controls:

- Producto:** A text box containing "LAMPARA 100W" and a small square button with three dots to its right.
- Cantidad:** An empty text box.
- Fecha de Entrega:** A text box containing "03/10/2002" and a small square button with three dots to its right.
- Precio Estimado:** An empty text box.
- Buttons:** Two buttons at the bottom labeled "Aceptar" and "Cancelar".

El producto seleccionado, en este caso una lámpara de 100W, se muestra en el campo Producto. A continuación el usuario debe ingresar la cantidad solicitada y el precio estimado (el cual esta interesado en pagar).

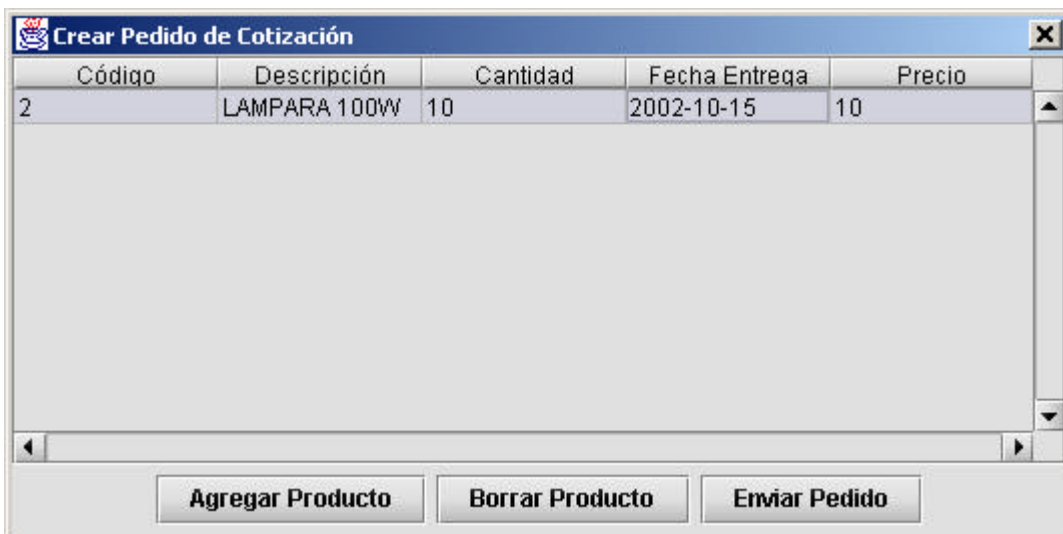
Para seleccionar una fecha de entrega, el usuario debe presionar el botón de elipsis (esta vez cercano al campo fecha de entrega) donde acto seguido aparecerá un calendario para la selección de la fecha como se muestra en la figura a continuación:



Con los botones de acción a los costados del mes y el año pueden cambiarse dichos valores. Para seleccionar una fecha el usuario debe hacer click con el ratón sobre un día en el calendario y presionar seleccionar.

Una vez completado el diálogo de carga de ítem del pedido de cotización el usuario debe presionar el botón "Aceptar".

En este momento el sistema muestra la lista de productos cargados para el pedido de cotización que el usuario ha ingresado hasta este momento según puede verse en la figura.



Una vez que el usuario ha cargado todos los productos que desea debe presionar el botón "Enviar Pedido" para que el mismo sea despachado a los potenciales proveedores de sus productos.

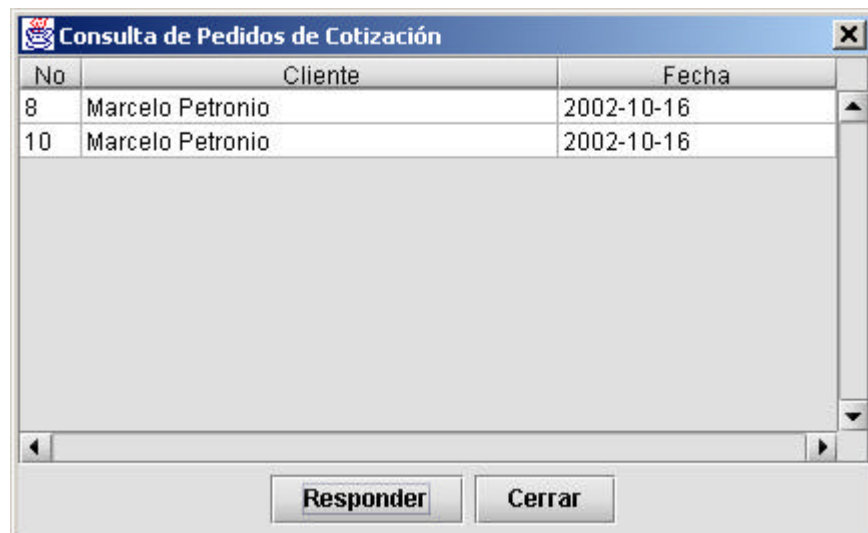
En este caso el sistema pedirá una confirmación sobre dicha transacción antes de proceder a enviar el pedido de cotización.



11.3. Consulta de Pedidos de Cotización y Creación de la Cotización

El proveedor de materiales puede consultar los pedidos de cotización solicitados por sus clientes por medio de la opción "Consultar Pedidos de Cotización" disponible en el menú principal.

El sistema muestra una lista de los pedidos de cotización ingresados en el sistema como se muestra en la figura a continuación:



En esta pantalla se muestra el número de pedido, el cliente que lo realizó y la fecha en la cual el pedido fue ingresado. Los pedidos de

cotización que se muestran son aquellos que aún no fueron respondidos por el proveedor.

Si el proveedor desea responder un pedido en particular, debe seleccionarlo de la lista y presionar el botón "Responder". A esta acción el sistema responde con la siguiente pantalla:

Producto	Fecha	Fecha Rta	Cantidad	Cant. Pedida	Precio	Precio Pedido
LAMPARA 10...	2002-10-03	2002-10-03	5.0000	5.0000	5.0000	5.0000
LAMPARA 10...	2002-10-03	2002-10-03	5.0000	5.0000	5.0000	5.0000

En esta pantalla se muestra el detalle del pedido de cotización a responder. Pueden verse datos como el número de pedido de cotización, la fecha de ingreso y el cliente que la solicitó.

Lo primero que debe hacer el proveedor es seleccionar una fecha de vencimiento de la cotización seleccionando una fecha en el campo "Fecha de Vencimiento del Presupuesto" según como se ha explicado en el apartado 13.2.

Una vez superada esta fecha, el sistema no permitirá al comprador colocar órdenes de compra en base a los datos suministrados por el proveedor.

A continuación, el proveedor debe confirmar o modificar los datos que el cliente solicita en base a cada producto, como ser el precio unitario, la fecha de entrega y la cantidad que es posible proveer.

Para ello el proveedor debe seleccionar un ítem de la lista y presionar el botón "Editar Item" apareciendo el diálogo a continuación:

	Solicitado	Confirmado
Fecha de Entrega:	2002-10-03	03/10/2002 ...
Cantidad:	5.0000	5.0000
Precio Unitario:	5.0000	5.0000

Aceptar Cancelar

En esta pantalla el sistema muestra los valores ingresados por el comprador y a la derecha una columna donde permite que el proveedor pueda modificarlos. Una vez hechas las modificaciones pertinentes el usuario debe presionar el botón "Aceptar".

Una vez que ha ingresado todos los valores para todos los productos, el usuario debe presionar el botón "Enviar" al cual el sistema contestará con una pantalla de confirmación de la operación:

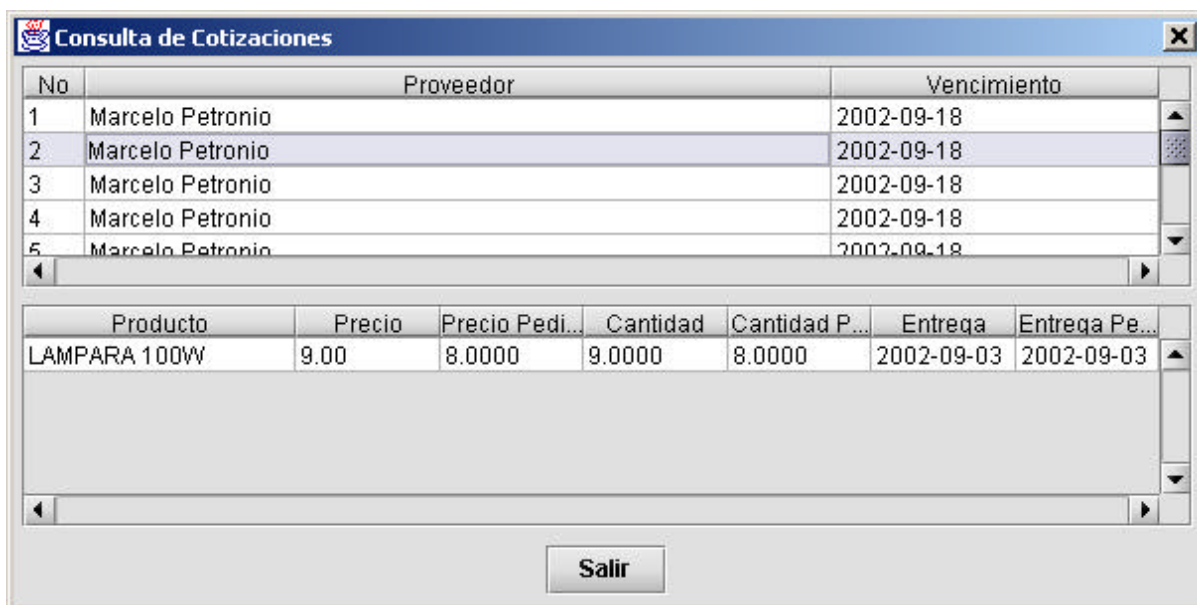
¿ Esta Ud. seguro de enviar la cotización ?

Si No

11.4. Consulta de Cotizaciones

El usuario comprador de materiales, estará interesado en todo momento de ver las cotizaciones ingresadas por sus potenciales proveedores.

Para ello deberá seleccionar la opción "Consultar Cotizaciones" en el menú principal del sistema. A ello el sistema responderá con la pantalla a continuación:



The screenshot shows a software window titled "Consulta de Cotizaciones". It contains two tables. The top table lists quotes with columns for "No", "Proveedor", and "Vencimiento". The bottom table shows details for a selected quote, with columns for "Producto", "Precio", "Precio Pedi...", "Cantidad", "Cantidad P...", "Entrega", and "Entrega Pe...". A "Salir" button is located at the bottom of the window.

No	Proveedor	Vencimiento
1	Marcelo Petronio	2002-09-18
2	Marcelo Petronio	2002-09-18
3	Marcelo Petronio	2002-09-18
4	Marcelo Petronio	2002-09-18
5	Marcelo Petronio	2002-09-18

Producto	Precio	Precio Pedi...	Cantidad	Cantidad P...	Entrega	Entrega Pe...
LAMPARA 100W	9.00	8.0000	9.0000	8.0000	2002-09-03	2002-09-03

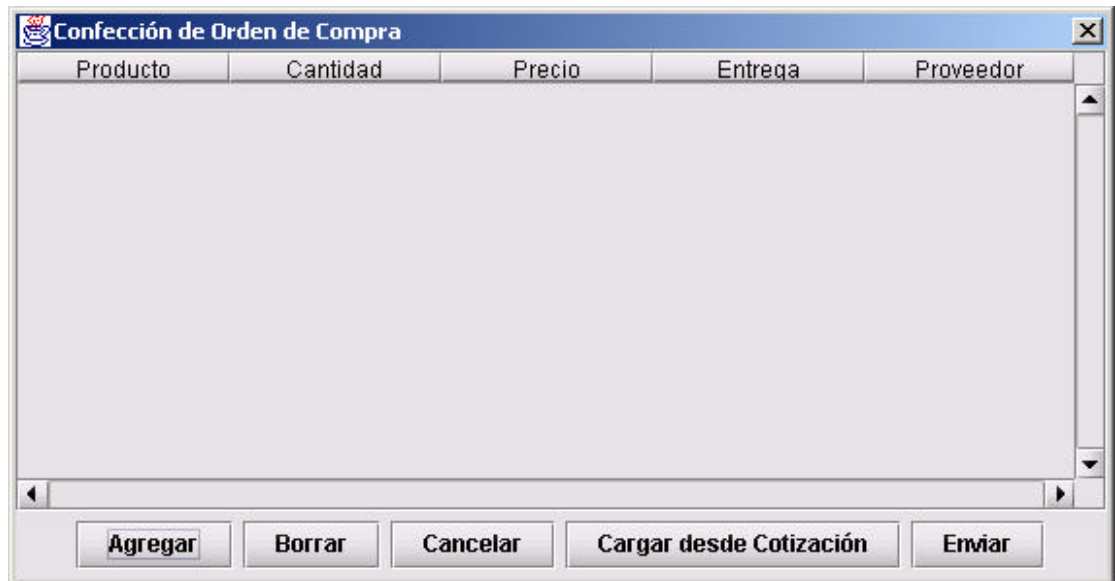
En esta pantalla pueden observarse dos secciones bien diferenciadas. En la parte superior una lista con todas las cotizaciones ingresadas por los distintos proveedores, detallando en número de cotización, el nombre del proveedor, y la fecha de vencimiento de dicho presupuesto.

Al seleccionar con el mouse una cotización, en la parte inferior de la pantalla el sistema muestra los productos cotizados y los valores solicitados oportunamente por el cliente versus los valores cotizados por el proveedor.

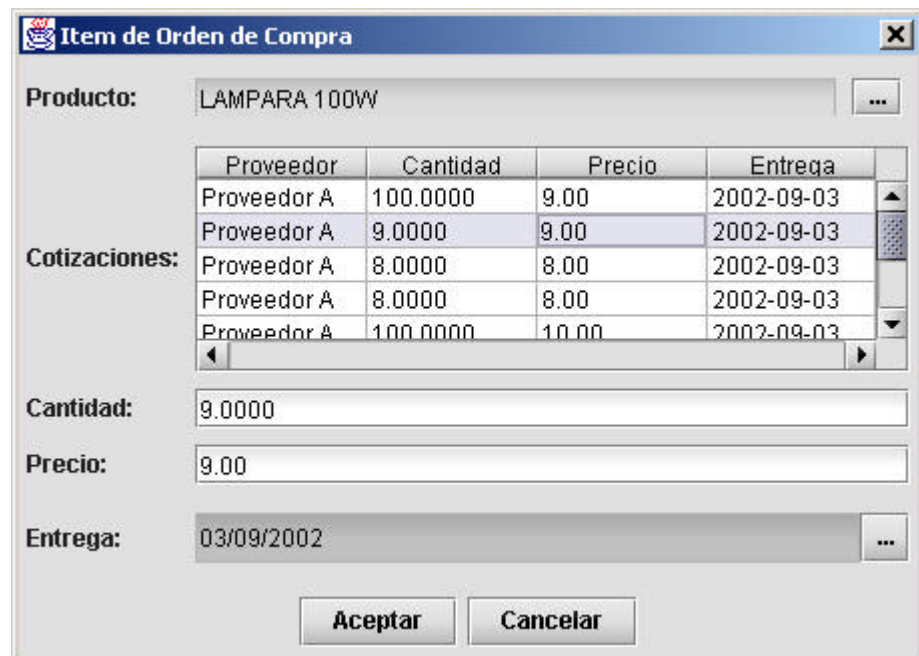
11.5. Creación de una Orden de Compra

En cuanto el comprador lo decida, podrá crear una orden de compra de materiales. Para ello deberá seleccionar la opción "Crear Orden de Compra" en el menú principal del sistema, apareciendo la pantalla a continuación:

ANEXO II - Manual del Usuario
ANEXO II - Versión 1.0.1



En esta pantalla (al igual que en el resto del sistema) aparece una lista de los productos que el comprador desea adquirir. Para agregar un producto a la orden de compra virtual, el usuario debe presionar el botón "Agregar". Entonces el sistema abrirá la pantalla del asistente de compra de productos como se muestra a continuación:



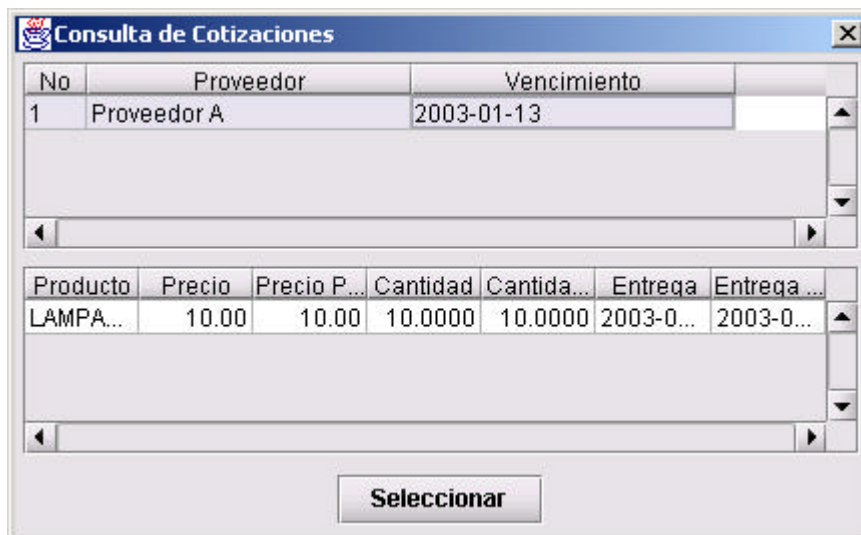
Lo primero que el usuario debe hacer es seleccionar un producto a comprar tal y como se explicó en el apartado 13.2.

Una vez seleccionado un producto, el sistema mostrará la lista de cotizaciones no vencidas realizadas por los distintos proveedores para el producto en cuestión.

Para seleccionar un ítem en particular bastará con hacer double-click sobre un ítem de la lista y el sistema cargará los valores deseados en el resto del formulario.

Para aceptar el ítem el usuario deberá presionar el botón de "Aceptar" y el producto será agregado a la lista de productos a comprar.

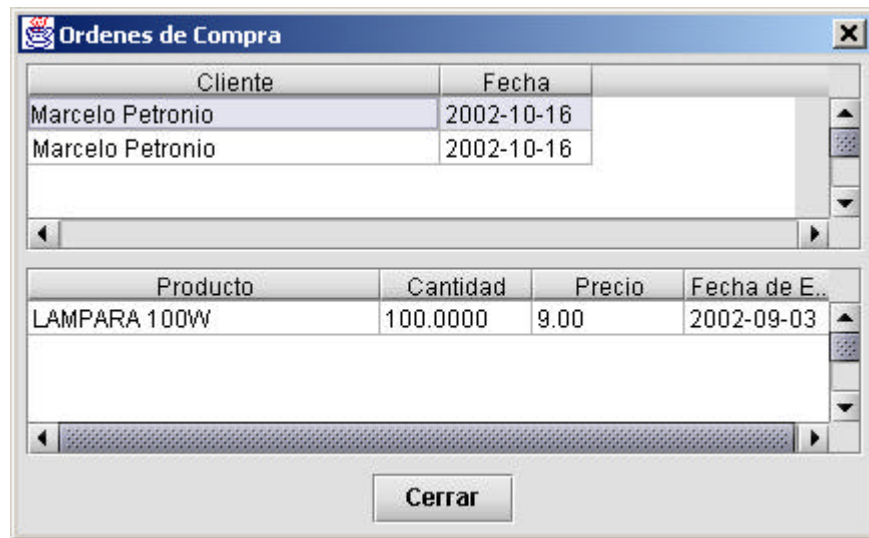
Otra modalidad de carga de productos, consiste en la posibilidad de cargar una orden de compra entera desde una cotización realizada por un proveedor. Para ello será necesario presionar el botón "Cargar desde Cotización" ubicado en la pantalla principal del sistema. Entonces, aparecerá una pantalla desde donde seleccionar una cotización realizada por un proveedor. Al presionar "Seleccionar" se confeccionará automáticamente una orden de compra basada en la cotización seleccionada.



Una vez ingresados todos los productos el usuario debe presionar el botón "Enviar" con el cual se generan las distintas órdenes de compra para los distintos proveedores seleccionados para cada uno de los productos.

11.6. Consulta de Ordenes de Compra

El proveedor de materiales posee una opción del sistema que permite consultar las órdenes de compra colocadas por sus clientes. Para ello debe presionar el botón "Consultar Ordenes de Compra" en el menú principal del sistema.



En esta pantalla se observan dos áreas bien definidas. En la superior se encuentran las órdenes de compra colocadas por los clientes. Al seleccionar una de la lista, el sistema muestra en la sección inferior el detalle de dicha orden para que el proveedor pueda procesarla.

Conclusiones

Información para el Control de la Configuración	
Proyecto: MERCADO VIRTUAL (MEVI)	Línea Base: Manuales de Usuario
Fase de Origen:	Diseño del Sistema de Información
Nombre o Código del Elemento de Configuración de Software:	Manual del Usuario del Sistema de Información (ANEXO II)
Versión N°: 1.0.1	Entrega N°: 02
Fecha de Última Modificación:	14/02/2003
Autor:	Lic. Marcelo Petronio
Estado:	TERMINADO

12. Conclusiones

12.1. Resultados Obtenidos

En el proceso de desarrollo de un mercado virtual se han identificado problemas en tres niveles: dominio de la aplicación, metodología y arquitectura.

En el nivel del dominio de la aplicación, se detecta

- La búsqueda costosa de proveedores.
- Actores de compraventa distribuidos en una extensa área geográfica.
- La no uniformidad en la especificaciones del producto.

En el nivel metodológico:

- No existen implementaciones de referencia para aplicaciones de mercados virtuales que utilicen la Metodología Métrica III.
- No hay referencias de cómo Métrica III se entalla a desarrollos distribuidos.

En el nivel de Arquitectura:

- La existencia de estándares rígidos para desarrollo de sistemas distribuidos (tres capas) lo que puede conllevar a un límite en la escalabilidad del sistema.

En esta tesis se propone una solución de implementación a cada uno de los problemas planteados.

Para los problemas identificados en el dominio de la aplicación se propone el desarrollo de un mercado virtual que proporcione:

- Herramientas para reducir los costos de búsqueda del proveedor.
- Un catálogo unificado de productos.

Para los problemas identificados en el nivel metodológico:

- Se plantea para el desarrollo una metodología que en su estructura sigue los lineamientos de Métrica III pero que se apoya en UML y el Proceso Unificado para aquellas cuestiones relativas a objetos no especificadas originalmente en la Metodología.
- En paralelo con el desarrollo de la aplicación, se propone una guía de recomendaciones para la implementación de sistemas distribuidos que soporten mercados virtuales.

Para los problemas identificados en la arquitectura:

- Se propone una arquitectura de n capas que integra las mejoras de la mensajería asincrónica entre sistemas a las arquitecturas clásicas ya disponibles.
- La arquitectura propuesta mejora la capacidad del sistemas al permitirle soportar picos de demanda.

Como un resultado adicional se puede señalar que se han tratado cuestiones relevantes a la persistencia del modelo de objetos en bases de datos relacionales, definiendo un modelo de diseño e implementación concreto a partir de recomendaciones disponibles en bibliografía, y en temas de comunicaciones asincrónicas entre componentes de la arquitectura.

12.2. Futuras Líneas de Desarrollo

Quedan como problemas abiertos:

- Verificar en otros dominios de aplicación que involucren procesamiento distribuido, que el uso integrado de Métrica III, UML y el Proceso Unificado repiten los resultados obtenidos en esta tesis.
- Evaluar los límites de degradación de los sistemas desarrollados con la arquitectura asincrónica propuesta.