

885216

UNIVERSIDAD AMERICANA DE ACAPULCO ¹
" EXCELENCIA PARA EL DESARROLLO "

**FACULTAD DE INGENIERÍA EN COMPUTACIÓN
INCORPORADA A LA UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**SITIO WEB DE UN PORTAFOLIOS FINANCIERO
CON ENTERPRISE JAVABEANS**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE :
LICENCIADO EN INGENIERIA EN COMPUTACIÓN
P R E S E N T A :
DANIEL HERRERA SALGADO**

DIRECTOR DE TESIS: ING. GONZALO TRINIDAD GARRIDO

ACAPULCO, GRO.

2003

**TESIS CON
FALLA DE ORIGEN**





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACIÓN DISCONTINUA

Contenido

Contenido	III
Agradecimientos	XIII
Introducción	XV
I Presentación del tema	1
1. Definición del problema	3
1.1. Introducción	3
1.2. Problemática	3
1.3. Motivaciones	7
1.3.1. Motivaciones tecnológicas	8
1.3.2. Motivaciones económicas	8
1.4. Resumen	9

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

IV CONTENIDO

2. Planteamiento de la hipótesis	11
2.1. Introducción	11
2.2. Hipótesis	11
II Antecedentes	13
3. El Mercado de Valores	15
3.1. Introducción	15
3.2. Mercado de Valores	16
3.2.1. Función de un Mercado de Valores	16
3.2.2. Clasificación	17
3.3. Índices de Mercado	18
3.3.1. Función de los Indicadores de Mercado	19
3.3.2. Clasificación	20
3.4. Valores financieros	21
3.4.1. Función de los valores financieros	22
3.4.2. Clasificación	22
3.5. Portafolios financiero	27
3.5.1. Función del portafolios financiero	27
3.5.2. Consideraciones para su creación	27

4. La tecnología hoy en día	31
4.1. Introducción	31
4.2. La Internet	32
4.2.1. Los inicios (ARPAnet)	34
4.2.2. El protocolo TCP/IP	37
4.2.3. La Internet hoy en día	39
4.3. La Web	41
4.3.1. Los inicios	43
4.3.2. Navegadores, HTML, HTTP y servidores Web	45
4.3.3. La Web hoy en día	48
4.4. La arquitectura cliente/servidor	49
4.4.1. Los inicios	50
4.4.2. Pros y contras del sistema cliente/servidor	52
4.4.3. El modelo cliente/servidor hoy en día	53
4.5. Programación de componentes	56
4.5.1. Los inicios	57
4.5.2. La Programación Orientada a Objetos	60
4.5.3. Los Componentes de Software	61
4.5.4. La programación de componentes hoy en día	62

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

VI CONTENIDO

III	Análisis de las tecnologías	63
5.	Tecnologías de desarrollo	65
5.1.	Introducción	65
5.2.	COM, DCOM, COM+ y .NET	65
5.2.1.	Antecedentes	66
5.2.2.	Características	68
5.2.2.1.	Características de COM	68
5.2.2.2.	Características de .NET	69
5.2.3.	Aspectos a considerar	70
5.3.	CORBA y CCM	71
5.3.1.	Antecedentes	71
5.3.2.	Características	73
5.3.2.1.	Características de CORBA	73
5.3.2.2.	Características de CCM	74
5.3.3.	Aspectos a considerar	75
5.4.	Enterprise JavaBeans	76
5.4.1.	Antecedentes	76
5.4.2.	Características de los EJB	77
5.4.3.	Aspectos a considerar	77

6. Comparativa entre tecnologías	79
6.1. Introducción	79
6.2. Aspectos a comparar	79
6.2.1. Precio	80
6.2.2. Soporte, Formación y Documentación	82
6.2.3. Plataformas soportadas	85
6.2.4. IDEs existentes	86
6.2.5. Protocolos de comunicación aceptados	87
6.3. Resultados	89
IV Descripción de la implementación	91
7. Solución propuesta (Prototipo)	93
7.1. Introducción	93
7.2. Descripción	94
7.3. Motivos de la selección	96
7.3.1. Cliente/servidor de 4 capas	96
7.3.2. Sistema operativo GNU/Linux	96
7.3.3. Base de datos PostgreSQL	97
7.3.4. Servidor de aplicaciones JOnAS	97
7.3.5. Servidor de páginas Web Tomcat	98

**TESIS CON
FALLA DE ORIGEN**

viii CONTENIDO

7.4. Integración de los elementos	98
7.4.1. Diseño básico	98
7.4.2. Creando los Enterprise JavaBeans	106
V Conclusión y perspectivas	109
8. Conclusión	111
9. Perspectivas y trabajo por realizar	113
VI Anexos	115
A. Códigos	117
B. Archivos de configuración	123
C. Pantallas	125

Índice de figuras

4.1. Relación entre redes e interredes.	33
4.2. La red ARPA en diciembre de 1969.	36
4.3. Comparación de los modelos TCP/IP y OSI/RM.	40
4.4. Navegador de páginas Web.	43
4.5. Proceso de visualización de una página Web.	47
4.6. Evolución del modelo cliente/servidor.	55
6.1. La IDE Java NetBeans.	87
7.1. Plataforma propuesta.	95
7.2. Página de inicio del Portafolios Financiero.	99
7.3. Formulario de registro de usuarios.	100
7.4. Página de ingreso al Portafolios.	102
7.5. Tabla de usuarios portadores de valores.	103
7.6. Fragmento de la hoja de estilo.	105
7.7. Fragmento de LoginServlet.	107

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

x INDICE DE FIGURAS

C.1. Arranque de Tomcat.	126
C.2. Deteniendo Tomcat.	127

Índice de cuadros

1.2. Crecimiento de la Internet.	4
3.2. Clasificaciones de mercados.	29
6.2. Comparación de acuerdo a su flexibilidad en precios.	81
6.4. Comparación de documentación, soporte y formación.	83
6.6. Comparación de plataformas soportadas.	85
6.8. Comparación de IDEs existentes.	86
6.10. Comparación de protocolos aceptados.	88
7.2. Descripción de la tabla del Portafolios.	102
7.4. Descripción de la tabla de Instrumentos.	103



Agradecimientos

A mi madre:
por su ejemplo de lucha e infinito amor.

A mi tía Maricela:
por su apoyo y amor.

A mi abuela Elena:
por su preocupación y amor.

A mi hermana Maribel:
por su amistad y amor.

A mi hermano Samuel:
por su buen humor y amor.

A mi tía Luz:
por sus cuidados y amor.

A mi tío Jorge:
por su ejemplo de dedicación.

TESIS CON FALLA DE ORIGEN

XIV AGRADECIMIENTOS

A mi padre:
por darme la vida.

A mi tía Estela:
por sus cuidados.

A mi primo Tomás:
por ser como un hermano mayor.

A mi amigo Irving:
por su apoyo.

A mi amigo Sandino:
por creer en mí.

A Mercedes:
por creer mucho en mí.

A Didier Donsez:
por su invitación al proyecto PEPiTA y a la Universidad de
Valenciennes.

Al ingeniero Gonzalo Trinidad Garrido:
por su valiosa asesoría.

A la Universidad Americana de Acapulco,
a la facultad de ingeniería y a mis profesores:
por ser trascendentales en mi formación profesional.

Introducción

El desarrollo de las finanzas en la WEB

Los rápidos avances en la computación y en las tecnologías de telecomunicaciones, particularmente Internet, han cambiado profundamente la dinámica de los mercados financieros. Más personas están negociando en línea a través de la Web en vez de usar los servicios tradicionales. Los empresarios y negociantes prefieren emplear Internet para ahorrar tiempo y aprovechar las ventajas que este medio proporciona. Una causa para ello es el acceso a información rápida y veraz, no en cuestión de días ni de horas, sino de segundos o minutos en el peor de los casos. La Internet y la Web facilitan la toma de decisiones oportunas a negociantes y empresarios.

En un mundo globalizado como en el que vivimos hoy en día, donde los efectos positivos o negativos de una economía ajena se reflejan en otros países. Tal fue el caso por ejemplo: del efecto tequila, el efecto vodka, el efecto zamba, el efecto dragón, el efecto tango y los sucesos del 11 de septiembre, que no sólo afectaron a las economías locales, sino que también a otras que mantenían sus intereses en ellas. La Internet y los

TESIS CON FALLA DE ORIGEN

XVI INTRODUCCIÓN

servicios de la Web representan una herramienta invaluable para conocer y actuar de acuerdo a lo que acontece a nivel mundial.

Las expectativas de las empresas hacia el mercado que representan los usuarios de la Web son muchas. Éstas expectativas incluyen, desde luego, a los servicios de corretaje y de asesoría financiera. Es cada vez más común encontrarse con sitios que ofrecen servicios de tipo financiero al navegar por la Web. Todo parece indicar que cada vez serán más y más los servicios que de este tipo se ofrezcan. Sin embargo, el interés empresarial por prestar estos servicios, no sólo radica en obtener más clientes o acaparar el mercado que representan (personas con un poder adquisitivo superior a la media poblacional), sino que también, se busca abaratar los costos de las transacciones.

Los Enterprise JavaBeans

Nuestra base cliente/servidor instalada está actualmente concebida por redes locales departamentales a servidor único. Ésta no puede hacer frente a la nueva demanda de millones de servidores y de aplicaciones intercambiando millares de millones de transacciones. Nuestra única esperanza reposa sobre soluciones cliente/servidor a tres niveles, utilizando los objetos repartidos o distribuidos y la Web. Los objetos distribuidos de la Web (JavaBeans, CORBA Beans, Enterprise JavaBeans y los componentes COM) están cambiando nuestra manera de concebir, de desarrollar, de condicionar, de distribuir, de licenciar y de mantener nuestro software cliente/servidor. [OHE99]

Los EJB (Enterprise JavaBeans) representan un acuerdo de la industria emergente de lo que una lógica de sistema de media capa es, y de cómo ésta debería ser representada. Tal y como Microsoft lo hace con su arquitectura de componentes pionera en el mercado, conocida como COM o DCOM, sólo que independiente de la plataforma, cosa de la cual COM no goza, pues únicamente puede desarrollarse con productos y para plataformas de Microsoft. Los EJB siguen la filosofía Java, es decir, buscar la independencia de la plataforma y de la integración. Por eso su famoso eslogan, "prográmelo una vez y úselo donde sea".

.NET es la nueva tecnología desarrollada por Microsoft. Presentada como un estándar y teniendo la característica de la independencia de lenguaje y el extenso uso de XML para crear y describir las aplicaciones. Sin embargo, por ser una tecnología relativamente nueva, existe muy poca información en comparación con los EJB y muchos más programadores conocen a Java que al lenguaje que .NET propone para desarrollar mejor sus aplicaciones, el nuevo lenguaje denominado como C#.

Los EJB también emplean XML o proporcionan las características necesarias para su utilización. Además, los EJB pueden ser integrados en conjunto con otras tecnologías libres para proporcionar una opción abierta y a muy bajo costo.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

XVIII INTRODUCCIÓN

Orientación del proyecto

Actualmente, el desarrollo de la industria informática fuera de la Internet es impensable, uno constata que la mayoría de las investigaciones están ligadas con la industria de redes. Pueden citarse como ejemplos: los objetos distribuidos, el paralelismo, y las tecnologías WAP. Por otro lado, se está obligado a utilizar las redes para transferir información importante. Por esta razón, los investigadores establecen exigencias cada vez mayores a las aplicaciones distribuidas.

Como numerosos servicios en la Web (finance.yahoo.com, Americtrade, etc.), el servicio bursátil que se plantea en esta memoria de tesis cuenta con un sistema de información. Este sistema proporciona a los cibernautas (portadores de valores) una asistencia en la administración de portafolios de valores bursátiles. Esta asistencia estará reducida al cálculo de los resultados de un portafolios compuesto de acciones bursátiles (como las de Vitro, IBM, Coca Cola, sólo por mencionar algunas) cotizadas en diversas bolsas del mundo y en los indicadores de los principales índices bursátiles (IPYC, CAC 40, DOW, NASDAQ, entre otros más).

El objetivo principal de este proyecto de investigación, en cuestión, es demostrar que con el uso de los Enterprise Java Beans, puede generarse un sitio Web de portafolios financiero que proporciona las ventajas de las aplicaciones basadas en componentes distribuidos, para garantizar su eficiencia y su fácil desarrollo y administración. La especificación de los

Enterprise Java Beans es tomada como modelo debido a las ventajas que Java proporciona, como lo son por ejemplo: la facilidad de programar y de poder trabajar sobre múltiples plataformas sin la necesidad de reescribir toda la aplicación (a diferencia de otros modelos como CORBA o Microsoft COM).

Los EJB son un modelo abierto que diferentes compañías proporcionan, incluso algunas lo hacen hasta de manera gratuita (JOnAS, JBoss). Existe una gran cantidad de programadores e información del lenguaje Java, por lo que el desarrollo, mantenimiento, optimización o actualización de una aplicación basada en los EJB están prácticamente garantizados.

Plan de esta memoria

La organización de esta memoria de tesis se presenta en 6 grandes partes, cada una estando subdividida en capítulos:

- La primera parte, denominada presentación del tema, aborda la problemática estableciendo las necesidades que existen actualmente para llevar a cabo la construcción de un sitio web de portafolios de valores financieros. También se presenta la hipótesis, donde se plantea que los Enterprise JavaBeans son una solución idónea para la implementación en la web de un servicio de portafolios bursátil que sea eficiente, de fácil desarrollo, fácil actualización y fácil mantenimiento.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

xx INTRODUCCIÓN

- La segunda parte muestra los antecedentes, describiéndose aquí, lo que es un mercado de valores y la tecnología existente en el ámbito computacional hoy en día, esto para tener una mejor perspectiva del contenido global de este documento.
- La tercera parte expone cuáles son algunas de las tecnologías de desarrollo de aplicaciones distribuidas basadas en componentes existentes hoy en día. También se muestra el análisis de alternativas para el desarrollo del sitio web de portafolios de instrumentos financieros, las ventajas y desventajas entre las diferentes alternativas.
- La cuarta parte describe la solución propuesta o desarrollo de la solución, presentando además, su implementación. Básicamente en este punto se presenta la tecnología seleccionada y cómo la integración de ésta misma se puede llevar a cabo con otras tecnologías, para obtener como resultado final un sitio lo suficientemente eficiente, de fácil desarrollo y administración.
- En la quinta parte se concluye y se describen algunos posibles trabajos futuros a realizar para obtener una solución actualizada, de acuerdo a las últimas tendencias tecnológicas.
- La última parte contiene los anexos de este documento, las referencias bibliográficas, así como un índice alfabético de los conceptos importantes.

Parte I
Presentación del tema

TESIS CON
FALLA DE ORIGEN

Capítulo 1

Definición del problema

1.1. Introducción

En el presente capítulo se pretende describir el problema y presentar algunas de las motivaciones para realizar el presente trabajo de tesis.

1.2. Problemática

En la actualidad el uso de la Internet es cada mayor. Uno puede apreciar que cada vez más y más sitios son dados de alta, además, el número de personas que tienen acceso a este servicio también se incrementa. Mas a medida que los usuarios aumentan, también aumentan las dificultades para generar aplicaciones que proporcionen un servicio eficiente y satisfactorio para los usuarios que lo emplean.

El cuadro 1.2, muestra el crecimiento de internet en rubros como son: el número de redes interconectadas, el número de computadoras, el número

TESIS CON FALLA DE ORIGEN

4 CAPÍTULO 1. DEFINICIÓN DEL PROBLEMA

Año	número de redes	número de computadoras	número de usuarios	número de administradores
1980	10	10^2	10^2	10^0
1990	10^3	10^5	10^6	10^1
2000	10^9	10^7	10^8	10^2

Cuadro 1.2: Crecimiento de la Internet.

de administradores y el número de usuarios. [CD00]

Nuestra base cliente/servidor instalada está actualmente concebida por redes locales departamentales a servidor único. Ésta no puede hacer frente a la nueva demanda de millones de servidores y de aplicaciones intercambiando millares de millones de transacciones. Nuestra única esperanza reposa sobre soluciones cliente/servidor a tres niveles, utilizando los objetos repartidos y la Web. [OHE99]

La gran mayoría de las aplicaciones está ideada como una aplicación cliente/servidor, lo que representa un problema si el número de usuarios de la aplicación es bastante elevado, tal y como ocurre en las aplicaciones basadas en la Web. Por eso, el uso de objetos distribuidos en plataformas cliente/servidor de n capas representa una solución idónea para construir sitios Web eficientes.

Para el diseño de un sitio Web el término eficiencia depende de un determinado punto de vista filosófico. Mientras que para algunos diseñadores este punto de referencia es el contenido, para otros puede ser la tecnolo-

gía, para algunos más los aspectos visuales o la usabilidad. [PT00]

Para ser considerado como un sitio Web eficiente; desde el punto de vista tecnológico, de desarrollo, de usabilidad y seguridad; un sitio Web debe cumplir con las siguientes características enlistadas a continuación:

La latencia. También conocida como tiempo de respuesta, tiene que ver con el tiempo que le toma al servidor responder a una petición realizada por el cliente. Mientras menor sea la latencia, más eficiente será considerado un sitio Web.

La disponibilidad. Se refiere al hecho de que cuando se desee acceder a la aplicación Web, ésta siempre esté disponible. A mayor disponibilidad, mayor eficiencia.

La escalabilidad. Tiene que ver con las facilidades que se proporcionen para realizar modificaciones a la presentación, el contenido y/o el funcionamiento de la aplicación Web.

La robustez. Si ocurriese alguna eventualidad o situación excepcional, el sitio Web debe reaccionar de manera apropiada. Una mayor capacidad de reacción equivale a mayor eficiencia.

La economía. Se refiere básicamente a hacer lo mismo o más con menos recursos monetarios. Desde luego, está estrechamente ligada con el tiempo de desarrollo porque a menor tiempo de desarrollo menores costos, pero también con la selección de la tecnología empleada en la construcción de la infraestructura física del sitio.

6 CAPÍTULO 1. DEFINICIÓN DEL PROBLEMA

La facilidad de uso. Sin importar la educación ni el nivel social del usuario, éste podrá hacer uso del sitio sin complicaciones.

La confidencialidad. Los datos de los clientes o usuarios sólo será única y exclusivamente accesible para ellos y para nadie más.

La integridad. Se refiere a la capacidad de proteger los datos de los clientes, además de los documentos que componen el sitio, contra modificaciones y accesos no autorizados.

La confiabilidad de la información. Está relacionada con las dos anteriores, pero además, si el sitio Web proporcionase información como noticias, estadísticas, resultados de encuestas, etc., entonces esta información contenida en el sitio deberá ser confiable.

Todas las cualidades anteriores se pueden considerar simples y sencillas, pero se requiere de una considerable y ardua labor de parte de los diseñadores, desarrolladores y administradores del sitio, para garantizar el cumplimiento de dichos aspectos. Aunque la latencia y la disponibilidad, sólo por ejemplificar, tienen una mayor relación con la infraestructura que con el desarrollo, un buen diseño y desarrollo de la aplicación favorecen a que ésta tenga un buen desempeño en tales rubros. La seguridad del sistema se garantiza si se satisfacen la confidencialidad, la integridad y la disponibilidad.

Por otro lado, y como ya sabemos, la información es un recurso importantísimo para las personas y, desde luego, también para las empresas. Con base en la información que se dispone son tomadas las decisiones. Por

ello, mientras más disponible, oportuna y confiable sea la información, las decisiones que se toman tendrían que ser mejores. La Internet y los servicios de la Web representan una herramienta invaluable para actuar de acuerdo a lo que acontece a nivel mundial. Lo que es más evidente en un mundo globalizado como en el que vivimos hoy en día, donde los efectos positivos o negativos de una economía ajena se reflejan también en otros países.

Las personas que tienen la oportunidad de poder ahorrar o invertir su capital excedente, requieren de información que les permita decidir cuáles son las mejores opciones para obtener una mejor retribución en sus inversiones. Para ello, algunas de estas personas adquieren periódicos y/o revistas financieras. Por supuesto que la gran mayoría de estas personas, por no decir que todas, pueden tener acceso a una computadora conectada a Internet, de no ser así, estarían pagando un costo de oportunidad por la posibilidad de estar mejor informados a un menor costo y con la misma facilidad con que se lee un periódico.

1.3. Motivaciones

Se puede decir que existen dos principales puntos de vista que sirvieron de motivación para la realización de este trabajo de investigación: el primero, como ingeniero, desde el punto de vista tecnológico; y el segundo, como ciudadano preocupado por la economía y el impacto de este rubro en la implementación tecnológica de las empresas, desde el punto de vista económico.

TESIS CON
FALLA DE ORIGEN

1.3.1. Motivaciones tecnológicas

La motivación primordial, desde este punto de vista, fue la de **desarrollar un sitio Web de manera práctica y sencilla, con base en una plataforma cliente/servidor de n capas y basada en componentes distribuidos que solucione los problemas que se presentan en los sitios Web de servidor único. Además, se busca que la solución garantice las características de construcción para que ese sitio Web pueda ser considerado eficiente, al menor costo posible y que permita una fácil administración.** Por administración se refiere a llevar a cabo las tareas de optimización, actualización y mantenimiento de la aplicación Web y no de los componentes de hardware que componen físicamente el sitio Web, ni del contenido.

1.3.2. Motivaciones económicas

La última tendencia del mercado es que las compañías están empezando a hacer negocios en la red más grande del mundo: Internet. La razón es el gran tamaño de la población de Internet. [GB94]

Si bien es cierto que la Internet no es una solución mágica para la economía de los países, empresas y/o personas, si representa una oportunidad de ampliar los mercados, de emplearla como un nuevo canal de distribución y consumo de productos y como un medio de comunicación eficiente y barato. De esta manera, la Internet permite reducir los costos de las transacciones.

A principios de la década de los 30 una llamada de tres minutos de Londres a Nueva York costaba 300 dólares; actualmente cuesta un dólar y si la información se transmite por Internet, baja su costo hasta un centavo. [PL98]

Entonces, ¿Por qué no aprovechar las ventajas que la Internet representa a través de la Web? ¿Por qué no abaratar los costos de las transacciones de esta forma? Como ingenieros en computación ¿Por qué no somos copartícipes en el incremento, a través de la Internet, de los flujos económicos y financieros del país? ¿Por qué no plantear una manera de crear un sitio Web financiero que proporcione las ventajas de las aplicaciones basadas en componentes que tenga un costo inferior y un desempeño igual o superior a las otras opciones?

1.4. Resumen

Por tanto, podemos resumir que la creación de un sitio Web que proporcione un servicio de portafolios financiero, que permita abaratar los costos de las transacciones y que garantice la calidad del sitio Web empleando una base cliente/servidor de n capas basada en los objetos repar-tidos de la Web, es un tópico altamente digno de tomar en consideración para un ingeniero en computación.

TESIS CON
FALLA DE ORIGEN

Capítulo 2

Planteamiento de la hipótesis

2.1. Introducción

En este capítulo se plantea la hipótesis, la cual establece la dirección y las limitantes del trabajo de investigación presentado en esta memoria de tesis.

2.2. Hipótesis

Existen tecnologías para el desarrollo de aplicaciones distribuidas basadas en componentes que son costosas, difíciles de utilizar y de administrar, por lo que el uso de los Enterprise JavaBeans es mucho más factible para desarrollar un sitio Web de portafolios financiero que cumpla con las características requeridas para que dicho sitio sea considerado como un sitio eficiente¹ de fácil desarrollo y administración.

¹Tal y como se describe eficiencia en la definición del problema.

Parte II
Antecedentes

Capítulo 3

El Mercado de Valores

3.1. Introducción

En el presente capítulo se presenta una descripción de qué es el mercado de valores y cómo se manejan los valores de inversión, los índices de mercado y, por supuesto, un portafolios financiero. De manera básica y sencilla se describen estos temas de gran importancia para un mejor entendimiento de esta memoria de tesis. No se tiene por objetivo presentar una información muy compleja ni mucho menos completa de los puntos tratados en este capítulo. Esto es debido a que éste es un trabajo de ingeniería en computación, no de economía, ni de finanzas. En cambio, este capítulo pretende exhibir un panorama general de los temas que se tratarán en él, para tener una idea más clara del contenido global de este documento.

3.2. Mercado de Valores

Un mercado de valores, o simplemente mercado, es un lugar donde se intercambian valores financieros. Un lugar donde se reúnen compradores y vendedores para determinar el precio de ciertos valores, abaratando así el costo de las transacciones.

Un mercado financiero es donde se intercambian activos financieros. Aunque la existencia de un mercado financiero no es una condición necesaria para la creación y el intercambio de un activo financiero, en la mayoría de las economías los activos financieros se crean y posteriormente se comercian en algún tipo de mercado financiero.

3.2.1. Función de un Mercado de Valores

Los mercados financieros, además de funcionar como enlace entre los emisores y los inversionistas, proporcionan las siguientes funciones económicas:

- La primera es que la interacción de compradores y vendedores en un mercado financiero determinan el precio y el rendimiento del activo negociado, llamado proceso de fijación de precios.
- La segunda es que los mercados proporcionan un mecanismo para que el inversionista venda un activo financiero, es decir, proporciona liquidez a los activos, aunque todos los mercados proporcionan alguna forma de liquidez, el grado de ésta es uno de los factores que caracteriza a los mercados.

- La tercera función económica de un mercado financiero es que reduce el costo de las transacciones: Los costos de búsqueda y los costos de información.

Los costos de búsqueda representan costos explícitos, tales como el gasto de dinero para anunciar la intención propia de vender o comprar un activo financiero; y costos implícitos, como el valor del tiempo gastado en encontrar una contraparte. Los costos de información son asociados con la apreciación de los méritos de invertir en un activo financiero, esto es, la cantidad y la probabilidad del flujo de efectivo que se espera sea generado. [FMF96]

3.2.2. Clasificación

Hay varias maneras de clasificar un mercado. El cuadro 3.2, nos muestra de manera breve algunas de las formas en que se pueden clasificar a los mercados financieros.

Los mercados primarios son aquellos mercados donde nuevos instrumentos de deuda son inicialmente vendidos (subastas de la Reserva Federal). Los mercados secundarios o mercados para instrumentos maduros, son aquellos mercados donde los instrumentos de deuda son revendidos (NYSE¹).

¹Por sus siglas en inglés *New York Stock Exchange*, es decir, intercambio de valores de Nueva York.

Otra manera de clasificar a los mercados es, como un mercado de subasta oral o un mercado continuo. En un mercado de subasta oral el comercio del mercado tiene lugar a intervalos de tiempo específicos. Los mercados continuos son mercados donde el comercio tiene lugar en términos continuos.

Una manera más de clasificar los mercados es determinar si es un mercado de agentes o de corredores. En un mercado de corredores un corredor actúa como un agente para un inversionista y compra o vende participaciones en representación del inversionista. El accionista negocia acciones con otros accionistas a pesar de utilizar un agente. En un mercado de agentes el agente compra o vende acciones para el inversionista utilizando el propio inventario del agente. El inversionista no negocia con otros inversionistas pero si con el agente que sirve de intermediario entre vendedores y compradores. [EG95]

3.3. Índices de Mercado

Un índice o indicador de mercado, es una medida basada en los precios y a menudo en número de participaciones vigentes de un número seleccionado de acciones que suponen transmitir información sobre el valor relativo de las participaciones comerciadas en un mercado. [FMF96]

Existen muchos índices de precios de acciones que proyectan y miden el desarrollo de los mercados de acciones, en cada país donde el comercio de acciones tiene lugar, existe por lo menos un índice de los movimientos

de precio de participaciones en general. Así por ejemplo, en México se tiene el IPC² de la Bolsa Mexicana de Valores. En Estados Unidos, como se tienen varias casas de bolsa, cada una de éstas posee su propio índice. Algunos ejemplos son: el DJIA,³ el *NYSE Composite Index*,⁴ el *NASDAQ⁵ Composite Index*, y algunos otros más. En Europa se tienen entre otros al francés CAC 40 y al británico FTSE 100 o también conocido como FOOTsie.

3.3.1. Función de los Indicadores de Mercado

Los indicadores de mercado de acciones realizan una variedad de funciones, por ejemplo, sirven de puntos de referencia para la evaluación del desarrollo de administradores de dinero profesionales, sirven como termómetros de la situación económica de una empresa, un grupo de ellas o incluso un país, también sirven de referencia a inversionistas para tomar mejores decisiones y así saber cómo, cuándo y dónde es conveniente invertir.

Básicamente los indicadores sirven para mostrar el movimiento de los precios de las participaciones de un grupo de empresas. Sin embargo, existen diferencias importantes en la magnitud de los movimientos, a pesar de que generalmente los índices de mercado suben y bajan en pa-

²Por sus siglas. Índice de Precios y Cotizaciones.

³De las siglas de *Dow Jones Industrial Average*, que quiere decir, Promedio Industrial Dow Jones.

⁴Significa índice compuesto en inglés.

⁵De sus siglas en inglés *National Association of Securities Dealers Automated Quotation*, que significa, Cotización Automatizada de la Asociación Nacional de Negociantes de Certificados de inversión.

TESIS CON
FALLA DE ORIGEN

trones equitativamente similares. Estas diferencias se deben a que cada indicador aplica y mide una faceta diferente del mercado de acciones. [FMF96]

3.3.2. Clasificación

Los indicadores de mercados de acciones pueden clasificarse en tres grupos, los cuales se enlistan a continuación:

- Los índices producidos por las bolsas de valores basados en todas las acciones comerciadas en las casas de bolsa.
- Los índices producidos por las organizaciones que seleccionan subjetivamente las acciones para ser incluidas en los índices.
- Los índices cuya selección de acciones está basada en una medida objetiva, tal como la capitalización de mercado de la compañía, la cual es una medida de tamaño de la empresa, es decir, el valor de mercado total de las acciones comunes que se calcula multiplicando el precio por participación de las acciones en el mercado por el número de participaciones vigentes.

Dentro del primer grupo de la clasificación de los índices mencionados en el listado anterior, podemos encontrar por ejemplo al IPC de la Bolsa Mexicana de Valores, al *NYSE Composite Index* de la *New York Stock Exchange* y el *NASDAQ Composite Index* de la *National Association of Securities Dealers*, que a pesar de no ser una casa de bolsa representa todas las acciones seguidas por el sistema NASDAQ en su índice.

Algunos de los índices pertenecientes al segundo grupo son el *Dow Jones Industrial Average* construido a partir de las 30 más grandes compañías industriales de acciones de primera clase comerciadas en la NYSE. Las compañías son seleccionadas por *Dow Jones & Company*, editores del *Wall Street Journal*;⁶ dentro de este grupo encontramos también al *Nikkei 225 Stock Average* basado en las 225 compañías más grandes de la primera sección del mercado de Tokio, calculado y publicado por la empresa de información financiera *Nihon Keizai Shimbun Inc.*

Por último, ejemplos del tercer grupo son los *Russell Index* producidos por la *Frank Russell Company*, un consejero para fondos de pensión y otros inversionistas institucionales. [FMF96]

3.4. Valores financieros

Los valores financieros o activos financieros son activos intangibles. Un activo, en términos generales, es cualquier posesión que tiene valor en un intercambio. Los activos pueden ser clasificados como tangibles e intangibles. Un activo tangible es aquél cuyo valor depende de propiedades físicas particulares, ejemplos de éstos serían edificios, terrenos o maquinaria. Los activos intangibles, por el contrario representan obligaciones legales sobre algún beneficio futuro. Su valor no tiene relación con la forma, física ni de cualquier otro tipo, en que estas obligaciones estén

⁶ Periódico financiero estadounidense que trata temas de la Bolsa de valores de Wall Street.

TESIS CON
FALLA DE ORIGEN

registradas. [FMF96]

3.4.1. Función de los valores financieros

Los activos financieros tienen dos funciones económicas principales las cuales se enlistan a continuación:

- La transferencia de fondos de aquellos que tienen un excedente para invertir, hacia aquellos que los necesitan para invertirlos en activos tangibles.
- Transferir fondos en forma tal, que se redistribuya el inevitable riesgo asociado con el flujo de efectivo, el cual se genera por los activos tangibles, entre aquellos que buscan y aquellos que proporcionan los fondos.

3.4.2. Clasificación

Los valores financieros o instrumentos de inversión se pueden clasificar así como se clasifican a los mercados. Recordemos que es en los mercados donde se comercializan los valores y es de acuerdo a las características de éstos como se clasifican los mercados. Aunque es conveniente profundizar en la clasificación de acuerdo al tipo de obligación. Así tenemos, que los instrumentos de valores se clasifican en: instrumentos de deuda e instrumentos de obligación. Sin embargo, algunos valores pueden caer dentro de ambas categorías. A continuación se mencionan algunos de los diferentes valores que existen:

Acciones: Una obligación en acción (también llamada obligación residual o simplemente acción común) compromete al emisor del activo financiero a pagar al poseedor una cantidad con base en las ganancias, si las hubiere, después de que se haya pagado a los poseedores de los instrumentos de deuda.

Bonos: Los bonos son instrumentos de deuda donde el emisor acuerda pagar al poseedor o inversionista a pagos fijos una cantidad de dinero establecida de antemano cada determinado tiempo hasta cubrir el adeudo en su totalidad.

Certificados: Documentos oficiales emitidos generalmente por los gobiernos como instrumento de deuda en los que el emisor se compromete a pagar una cantidad fija al cumplimiento de una fecha previamente definida.

Títulos: Los títulos de inversión es otra forma de llamar a los instrumentos de inversión, sin embargo, otro tipo de títulos son los títulos de propiedad los cuales son activos tangibles, es decir, la inversión está en el valor que en sí tiene la propiedad.

Según la CONDUSEF⁷ dependiendo de la Casa de Bolsa o la institución financiera que se seleccione para realizar operaciones, se tendrá acceso a los instrumentos de inversión y servicios como los siguientes:

1. Sociedades de inversión

⁷Comisión Nacional para la Protección y Defensa de los Usuarios de Servicios Financieros.

TESIS CON
FALLA DE ORIGEN

**TESIS CON
FALLA DE ORIGEN**

24 CAPÍTULO 3. EL MERCADO DE VALORES

- 2. Sociedades de inversión en instrumentos de deuda (acciones)**
- 3. Sociedades de inversión comunes (acciones)**
- 4. Sociedades de inversión de capitales de riesgo (acciones)**
- 5. Mercado de capitales**
 - Acciones
 - Certificados de Participación Ordinarios sobre acciones (CPOs sobre acciones)
 - Obligaciones convertibles en acciones
 - Warrants (títulos opcionales)
- 6. Mercado de dinero**
 - Papeles Gubernamentales
 - a)* Certificados de la Tesorería de la Federación (CETEs)
 - b)* Certificados de la Tesorería de la Federación denominados en UDIs (UDICETEs)
 - c)* Bonos de Desarrollo del Gobierno Federal (BONDEs)
 - d)* Bonos de Desarrollo del Gobierno Federal denominados en UDIs (UDIBONOs)
 - e)* Bonos Ajustables del Gobierno Federal (AJUSTABONOs)
 - f)* Bonos de la Tesorería de la Federación (TESOBONOs)
 - g)* Instrumentos emitidos en el extranjero a cargo del Gobierno Federal sujetos a negociación en el país (BONOS UMS)

- h)* Pagarés de Indemnización Carretera con aval del Gobierno Federal (PIC-FARAC)
- i)* Bonos de Regulación Monetaria (BREMs)
- j)* Bonos de Protección al Ahorro (BPAs)
- Papeles Bancarios
 - a)* Pagarés con Rendimiento Liquidable al Vencimiento (PRLVs)
 - b)* Certificados de Depósito a Plazo (CEDEs)
 - c)* Bonos Bancarios (Bonos)
 - d)* Bonos Bancarios para el Desarrollo Industrial (BONDI)s
 - e)* Aceptaciones Bancarias (ABs)
 - f)* Papeles con Aval Bancario
- Papeles Privados
 - a)* Pagarés Financieros
 - b)* Papel Comercial simple o indizado
 - c)* Pagaré de Mediano Plazo
 - d)* Bonos de Prenda
 - e)* Certificados de Participación Ordinarios de deuda o amortizables (CPO deuda)
 - f)* Certificados de Participación Inmobiliarios (CPIs)
 - g)* Bonos Estructurados (Bono empresarial)
 - h)* Obligaciones

7. Mercado de Metales

- Certificado de plata (CEPLATA)

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

26 CAPÍTULO 3. EL MERCADO DE VALORES

- Centenarios de oro
 - Onzas troy de plata
8. Sistema Internacional de Cotizaciones (SIC) y Valores Extranjeros
- Acciones de empresas ubicadas en el extranjero (listado SIC)
 - Títulos representativos de deuda cotizados en el extranjero e inscritos en la Sección Especial del RNVI
 - Servicios
 - Información de emisoras: asistencia a las asambleas, derechos corporativos y patrimoniales, prospectos, reportes trimestrales, resultados de asambleas, precios, análisis, etc.
 - Asesoría en materia de valores: recomendaciones, integración de portafolios personalizados, manejo discrecional de cuentas, etc.
 - Operaciones especiales: préstamo de valores, ventas en corto, operaciones de arbitraje, ofertas públicas de compra, etc.
 - Venta de acciones y obligaciones de TELMEX
 - Futuros y opciones extrabursátiles (productos derivados fuera de bolsa)
 - Pagos de tarjetas
 - Depósitos bancarios
 - Servicios de ingeniería financiera y banca de inversión
 - Fideicomisos

- Administración de fondos de pensiones o jubilaciones Fondos de promoción bursátil

3.5. Portafolios financiero

Un portafolios financiero, portafolios de inversiones o carpeta de inversiones, es el conjunto de inversiones de una persona o una empresa. Un portafolios contiene diferentes tipos de instrumentos de inversión que conforman todas o algunas de las inversiones que posee un inversionista o un grupo de inversionistas.

3.5.1. Función del portafolios financiero

Un portafolios tiene por función facilitar el análisis del comportamiento de todos los instrumentos que conforman el portafolios. De esta manera, se puede conocer el comportamiento de todas las inversiones en conjunto, proporcionando una perspectiva general de las ganancias y/o pérdidas que, de manera integral, proporcionan aquellas inversiones incluidas dentro del portafolios.

3.5.2. Consideraciones para su creación

La creación de un portafolios depende de la manera en que el inversionista prefiera supervisar o controlar sus instrumentos de inversión. Así por ejemplo, un inversionista puede crear un portafolios sólo para instrumentos de deuda, otro para instrumentos de obligación, o poner todos

TESIS CON
FALLA DE ORIGEN

sus instrumentos en un único portafolios. Es el propio inversionista, de acuerdo a su comodidad o a su capacidad de supervisión, quien decide cómo generar su o sus portafolios financieros.

Para la creación de un portafolios debemos considerar que nunca será lo mismo supervisar unos cuantos instrumentos de inversión, que cientos o miles de ellos, es por esto que, cuando se tienen una cantidad significativa de instrumentos de inversión, es recomendable manejar varios portafolios para el control de éstos instrumentos.

Clasificación por la naturaleza de la obligación
Mercado de deuda
Mercado de acción
Clasificación por vencimiento de la obligación
Mercado de dinero
Mercado de capitales
Mercado primario
Mercado secundario
Clasificación por entrega inmediata o futura
Mercado spot o en efectivo
Mercado derivado
Clasificación por estructura organizacional
Mercado de subasta
Mercado de mostrador
Mercado intermediario

Cuadro 3.2: Clasificaciones de mercados.

TESIS CON
FALLA DE ORIGEN

Capítulo 4

La tecnología hoy en día

4.1. Introducción

A continuación se presentan algunos antecedentes computacionales, pues es muy importante contar con las bases de lo que son: la Internet, la Web, la arquitectura cliente/servidor de n niveles o multicapas, la programación orientada a objetos (POO) y la programación basada en componentes; para tener un mejor entendimiento de este trabajo de investigación. Los temas abordados en el presente capítulo pretender ofrecer una idea más clara y precisa de lo que se tratará a lo largo de la presente memoria de tesis.

El propósito de este capítulo es la de presentar los conceptos o definiciones de cada uno de los temas mencionados en el párrafo anterior, acompañados por una breve reseña histórica y una sencilla descripción de la situación actual. Estos tópicos son fundamentales para el entendimiento y desarrollo de este trabajo de investigación. Empero, no se pretende

profundizar en las temáticas antes mencionadas, pues cada una de éstas, por sí mismas, son muy extensas e inclusive dignas de una investigación más exhaustiva.

4.2. La Internet

La Internet¹ puede verse como la infraestructura que conforma la red más grande de todas y que incluye todas aquellas redes, líneas de comunicación (satélites, líneas telefónicas, antenas de radio, etc.), servidores y computadoras de todo el mundo. La Internet es quizá, la mayor revolución tecnológica y de comunicación del siglo XX.

Internet es la interred² más grande que se haya creado jamás. Está compuesta por más de 20,000 redes, conecta alrededor de 2,000,000 de computadoras y la acceden alrededor de 25,000,000 de personas diariamente. Internet también es el experimento de interoperabilidad más grande y exitoso de la raza humana. Cientos de fabricantes de hardware y miles de programadores han construido productos para TCP/IP³ en general, así como productos con ciertas características especiales para Internet. [GB94]

Internet es, simplemente, una red de computadoras localizadas alrededor del mundo. La interred más extensa y ampliamente conocida del

¹Del inglés, *International Network*, o sea, Red Internacional.

²Red de redes.

³Por sus siglas en inglés *Transmission Control Protocol/Internet Protocol*, que en español significa, Protocolo de control de transmisión/protocolo de Internet.

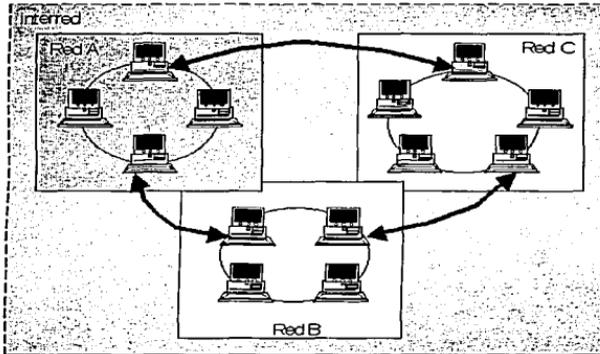


Figura 4.1: Relación entre redes e interredes.

planeta, Internet (escrita con I mayúscula), conecta más de veinte mil computadoras en 150 países. Dentro de ella existen miles de tipos de computadoras, las cuales utilizan una variedad similar de software de red. [JLW96]

Cuando dos o más computadoras son conectadas de tal forma que puedan comunicarse se forma una red. También se pueden conectar dos o más redes, cuando esto sucede, se forma una interred o internet.

La figura 4.1, arriba, muestra la relación entre redes e interredes.

4.2.1. Los inicios (ARPAnet)

Todo empezó en el año de 1969 gracias a la tecnología militar de los Estados Unidos de América (EUA). ARPAnet, la red de computadoras de ARPA,⁴ que se comunicaba por medio de paquetes de información y cuya finalidad era permitir a los diferentes nodos de la red, compuestos por computadoras no heterogéneas, comunicarse con los demás nodos sin ser estrictamente fundamentales para el funcionamiento global de la red. De esta forma, en el caso de que ocurriera una falla en algunos de los nodos de la red, éstos no podían seguir funcionando, pero si la red conformada por los nodos restantes.

La idea de una red de tales características, surgió en el año de 1962 por Paul Baran, investigador de la *RAND Corporation*,⁵ debido al sabotaje de varias antenas de comunicación militares en Utah en el año de 1961. A esta idea la denominó “la papa caliente”, la cual consistía en dividir el mensaje en paquetes que se enviaban lo más rápido posible, de ahí su nombre. Sin embargo en ese entonces nadie sabía conectar computadores distintos entre sí.

Para 1966 ARPA contaba con varios proyectos costosos que involucraban el uso de diferentes computadoras, por lo que surgió la idea de conectarlas en red y reducir de esa manera los costos. En 1968 se aprobó

⁴Por sus siglas en inglés, *Advanced Research Projects Agency*, es decir, Agencia de Proyectos de Investigación Avanzada.

⁵Que proviene de *Research and Development Corporation*, o sea, Corporación de investigación y desarrollo.

un presupuesto de 2.2 millones de dólares que permitirían a la ARPA concretar la idea. ARPA sometió a concurso el proyecto y una firma consultora de Massachussets llamada BBN, conformada por Bolt, Beranek y Newman consiguió el contrato. La tarea era superar la heterogeneidad de las computadoras.

Para lograr superar la barrera de la heterogeneidad, se pensó en una especie de computadora traductora denominada IMP.⁶ Nace también en este momento el NCP,⁷ el protocolo base de ARPAnet. También es cuando nacen los RFC,⁸ la manera en que la Internet es renovada y modificada todavía en la actualidad. [FF00][STB00]

En septiembre de 1969 se realiza el primer enlace con IMP en la UCLA.⁹ Para el mes de diciembre del mismo año, la red ARPAnet estaba conformada por 4 nodos. En un principio, sólo algunas universidades y laboratorios, que colaboraban en los programas de defensa del Pentágono, tenían acceso a la ARPAnet.

La figura 4.2., muestra como se encontraba conformada la red ARPA a finales de 1969. La UCLA fue el primer nodo de la red. Contaba con una computadora Sigma 7. Después le siguió el SRI¹⁰ que empleaba un

⁶Por sus siglas en inglés, *Interface Message Processor*, Interfaz procesadora de mensajes.

⁷*Network Control Protocol*. Protocolo de control de red.

⁸*Request For Comments*: Solicitud de comentarios.

⁹*University of California at Los Angeles*: Universidad de California en Los Angeles.

¹⁰*Stanford Research Institute*: Instituto de investigación de Standford.

TESIS CON
FALLA DE ORIGEN

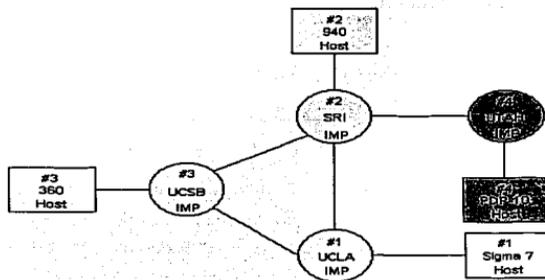


Figura 4.2: La red ARPA en diciembre de 1969.

sistema 940. Por último se agregaron la UCSB¹¹ con un sistema 360, y la universidad de Utah con una computadora PDP-10. [RFC15]

Cavilando en cómo sería posible conectar otras redes a la red ARPA durante un congreso en el verano de 1973 Vint Cerf junto con Bob Kahn meditaron en un protocolo de comunicación basado en datagramas. Asimismo pensaron que se necesitaría una computadora que funcionara como compuerta (gateway). Se madura la idea anterior con el paso de los días y entonces nace el protocolo de comunicación TCP.¹² Se dice también que es cuando la Internet nació.

El TCP pretendía permitir la conexión de otras redes a la ARPAnet,

¹¹ *University of California at Santa Barbara*: Universidad de California en Santa Bárbara.

¹² *Transmission Control Protocol*: Protocolo de control de transmisión.

considerando también a las redes que aún no habían sido construidas. En 1978 se decide separar el TCP en dos protocolos. El TCP y el IP.¹³ El primero se encargaría de descomponer y recomponer el mensaje en paquetes, mientras que el segundo se encargaría del envío de cada uno de los paquetes a su destino.

La ARPAnet envejecía, nadie esperaba en 1969 que el número de nodos sobrepasaría los 255, sin embargo, a finales de los 70s era evidente que lo haría y que se necesitaba un cambio. Es durante este mismo periodo que la BSD¹⁴ incluye el TCP/IP, algo realmente importante porque muchas personas utilizaban esta distribución basada en Unix en sus sistemas no tan poderosos ni tan costosos.

Un año importante para la Internet y al cual muchos individuos consideran como la verdadera fecha de su nacimiento, es cuando finalmente ARPAnet adopta el protocolo TCP/IP en el año de 1983. [FF00]

4.2.2. El protocolo TCP/IP

Internet se basa en una colección de protocolos llamada conjunto de protocolos TCP/IP. Un conjunto de protocolos es una colección de protocolos cooperativos y complementarios. El conjunto de protocolos TCP/IP incluye los protocolos TCP (Protocolo de Control de Transmisión) y el IP (Protocolo Internet), así como otros varios. Todos ellos trabajan en

¹³Internet Protocol: Protocolo de Internet.

¹⁴Berkeley Software Distribution: Distribución de Software Berkeley.



TESIS CON FALLA DE ORIGEN

conjunto para transmitir información por Internet. [JC95]

Fue gracias a las universidades y laboratorios de investigación en Estados Unidos que se desarrolló el protocolo TCP/IP, pues fue un trabajo desarrollado en conjunto por esas universidades y centros de investigación, para obtener un grupo de protocolos que pudieran utilizarse como el estándar en las comunicaciones por red. Por este motivo de estandarización podemos considerar al TCP/IP como parte fundamental en el desarrollo de la Internet. El éxito del TCP/IP fue tal, que la ARPAnet comenzó a ser llamada red TCP/IP y así fue llamada por algunos años a partir de 1983, año en que ARPAnet adoptó el protocolo TCP/IP.

Los protocolos de aplicaciones de Internet en un principio eran sólo de tres tipos y lo seguirían siendo hasta 1991: SMTP¹⁵ para la mensajería electrónica, FTP¹⁶ para la transferencia de archivos y el Telnet¹⁷ para conectarse a distancia a una computadora distante desde una terminal. No había Web.¹⁸ Estos protocolos tenían dos propiedades: eran abiertos, es decir, independientes de los constructores particulares, y podían funcionar con todo tipo de computadoras; además eran públicos, por lo que todo mundo tenía acceso a su especificación. [FF00]

En la figura 4.3., en la página siguiente, podemos apreciar los diferentes

¹⁵ *Simple Mail Transfer Protocol*: Protocolo simple de transferencia de correo.

¹⁶ *File Transfer Protocol*: Protocolo de transferencia de archivos.

¹⁷ Palabra compuesta por la palabra griega *Tele*, que significa a distancia; y la palabra inglesa *network*, que significa red.

¹⁸ *World Wide Web*, *WWW* o simplemente *Web*, es el nombre que se le dan a los archivos enlazados alrededor del mundo y que asemejan a una telaraña.

protocolos que conforman el TCP/IP, además de una comparativa entre el modelo TCP/IP y el modelo OSI/RM¹⁹ desarrollado por la ISO²⁰ en el año de 1980.

La aparición del TCP/IP fue un suceso trascendental para el desarrollo de la Internet puesto que sus protocolos permitían a los investigadores y científicos alrededor del mundo transferir archivos usando el FTP, enviar correos electrónicos por medio del protocolo SMTP, acceder y trabajar remotamente en una computadora utilizando el Telnet. Aunque, otro suceso significativo fue la creación de la Web, la cual ha permitido la masificación del uso de la Internet, convirtiéndola en lo que es en la actualidad.

4.2.3. La Internet hoy en día

¿Qué es la Internet actualmente? Si se adopta el punto de vista del especialista en Telecomunicaciones se trata, la mayoría de las veces, de un conjunto de redes autónomas, interconectadas a través de una infraestructura de comunicaciones que se había ideado para transportar la telefonía vocal. [STB00]

Ahora es un tópico decir que Internet ha revolucionado la manera de localizar información y personas. Casi todo el mundo ha ido a Internet por cuestiones de investigación y de comunicación, tanto para negocios

¹⁹De sus siglas en inglés, *Open Systems Interconnection/Reference Model*, que quiere decir, Modelo de referencia de interconexión de sistemas abiertos.

²⁰Por las siglas en inglés, *International Standards Organization*, es decir, Organización de estándares internacionales.

TESIS CON
FALLA DE ORIGEN

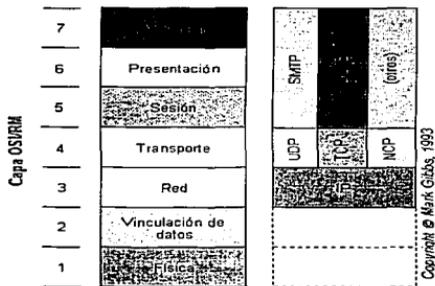


Figura 4.3: Comparación de los modelos TCP/IP y OSI/RM.

como para utilización personal. [PT97]

En la actualidad la Internet ya no es utilizada únicamente por expertos e investigadores como en sus inicios. Hoy en día, la Internet se ha convertido en un recurso de gran importancia para las empresas, centros de educación, para la población que tiene acceso a ella, y para los gobiernos.

Es cierto que aún quedan muchas cosas por hacer para mejorar o incrementar los servicios que Internet puede ofrecer. Por ejemplo: mejorar el protocolo TCP/IP, mejorar las líneas de comunicación para ofrecer un mayor ancho de banda y así poder enviar un flujo mayor de datos por segundo, pero se trabaja día a día en mejorar ese mundo virtual llamado Internet.

Una muestra de los esfuerzos por mejorar Internet. Es el consorcio denominado Internet2²¹ conformado por universidades, industrias y gobiernos que pretende hacer mejoras en aplicaciones, infraestructura de comunicaciones y protocolos. Como es el caso del muy mencionado IPv6 (Protocolo Internet versión 6).

Debido a que ni la Internet, ni la Web pertenecen a compañía o nación alguna, los estándares de éstos son aprobados por una combinación del *World Wide Web Consortium (W3C*²²) y la *Internet Architecture Board (IAB*²³). [JLW96]

La ICANN,²⁴ la IETF²⁵ y la ISOC²⁶ son esenciales para la operación de Internet. La W3C no está directamente involucrada con los trabajos de Internet. Desarrolla estándares técnicos para la Web, cuya popularidad ha sido la mayor fuerza detrás de la difusión de la Internet. [CR01]

4.3. La Web

Mucha gente confunde los términos de Internet y la Web pensando que

²¹<http://www.internet2.org>

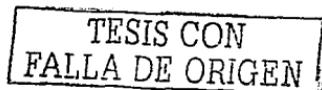
²²W3C: Consorcio de la WWW, Consorcio conformado por industrias privadas y universidades para guiar el desarrollo técnico de la Web y el conjunto de sus estándares.

²³IAB: Junta responsable de la evaluación, comprobación y aceptación de los estándares de Internet conformado por: la IETF, la IESG, la IRTF y la IRSG.

²⁴Internet Corporation for Assigned Names and Numbers: Corporación de Internet para la asignación de nombres y números.

²⁵Internet Engineering Task Force: Fuerza de tareas ingenieriles de la Internet.

²⁶De la composición de *Internet Society*, que significa: Sociedad de la Internet.



TESIS CON FALLA DE ORIGEN

son la misma cosa; lo cual es absolutamente incierto. Existen diferentes servicios que son empleados gracias a Internet, por ejemplo: el correo electrónico, la telefonía por Internet, la mensajería instantánea, la transferencia de archivos electrónicos y la misma Web entre algunos otros servicios más.

La World Wide Web (o WWW) es el contenedor de información electrónica más grande del mundo. En otras palabras, la WWW es una colección de millones de documentos enlazados que residen en computadoras a lo largo del mundo. Cuando utiliza su navegador para tener acceso a sitios en la Web, puede revisar texto, gráficos, vídeo, e incluso escuchar audio (todos juntos conocidos por hipermedios). Internet es el vehículo que permite a las computadoras alrededor del mundo comunicarse entre sí. La Web, por lo tanto, es un sistema de comunicación global que permite a las computadoras transferir electrónicamente información basada en múltiples medios por Internet. [JLW96]

Como puede apreciarse entonces, la Web es tan sólo un servicio de la Internet que ofrece un sin número de páginas elaboradas en HTML u otros lenguajes y/o formatos que permiten ser visualizadas por medio de navegadores, tales como: Galeon, Mozilla, *Netscape Navigator* o *Internet Explorer*. Precisamente, es gracias a la facilidad de emplear los navegadores y de visualizar las páginas Web en HTML²⁷ empleando el protocolo HTTP,²⁸ que se empieza a expandir la Web y hacerse más po-

²⁷ *HyperText Markup Language*, que en español significa: Lenguaje de marcación de hipertexto.

²⁸ *HyperText Transfer Protocol*, que significa en español: Protocolo de transferencia de



Figura 4.4: Navegador de páginas Web.

pular la Internet. Es por eso, quizá, la confusión de muchos.

La figura 4.4, arriba, muestra al navegador de páginas Web llamado Galeon que forma parte de los paquetes de Gnome, una distribución de ambiente de escritorio de la GNU.²⁹

4.3.1. Los inicios

La Web o World Wide Web fue concebida en Europa por Tim Berners-Lee por allá del año de 1989 en el CERN³⁰ de Génova, pero no fue sino hasta 1992 que se empezaron las primeras presentaciones públicas en el hipertexto.

²⁹ Acrónimo recursivo que quiere decir *GNU is Not Unix*, es decir, GNU No es Unix.

³⁰ *Conseil Européen pour la Recherche Nucléaire*, del francés, y significa: Consejo Europeo para la Investigación Nuclear.

TESIS CON FALLA DE ORIGEN

CERN de Ginebra. Debido a que tales publicaciones estaban al alcance de todos desde el CERN, se hicieron cada vez más y más populares entre los usuarios de Internet, la inmensa gran mayoría, los investigadores como Tim Berners-Lee.

Durante años, antes de la Web, los usuarios almacenaron documentos en archivos o sistemas dispersos por todo Internet. Para tener acceso a ellos, ejecutaban un programa especial, denominado FTP. Con FTP, los usuarios podían transferir con rapidez archivos de un sistema a otro. Desafortunadamente, a medida que aumentó la cantidad de archivos, se dificultó a los usuarios localizar los documentos que necesitaban. Como resultado, Internet contenía millones de documentos que la mayoría de los usuarios desconocía. Al enlazar documentos relacionados, la Web facilitó a los usuarios localizarlos y moverse entre ellos. [JLW96]

La historia comenzó en marzo de 1989, cuando Tim Berners-Lee propuso en el CERN un proyecto de gestión de la información que tuvo escaso eco. Un año después, volvió a someter su propuesta y obtuvo algunos materiales para realizar su sistema. A finales de 1990 presentó su primera propuesta y en 1992 se realizaron las primeras presentaciones. Su difusión fue fulgurante, pasando de veintiséis servidores a finales de 1992 a doscientos en octubre de 1993. [FF00]

En 1993 la aparición de Mosaic, ancestro universitario de Netscape, en el NCSA (*National Center for Supercomputing Applications*³¹), hizo des-

³¹En español, centro nacional para aplicaciones de supercomputo.

pegar espectacularmente a la World Wide Web. [STB00]

Los navegadores no eran otra cosa que programas que solicitaban al servidor de páginas Web, por medio del protocolo HTTP, documentos en HTML los cuales contenían las reglas en que serían desplegados los datos contenidos en el documento solicitado.

4.3.2. Navegadores, HTML, HTTP y servidores Web

Los navegadores son aplicaciones cliente que solicitan a los servidores de páginas Web documentos escritos en HTML empleando el protocolo HTTP. Al recibir el documento por medio de este mismo protocolo, el navegador interpreta el documento HTML y despliega su información de acuerdo a las reglas establecidas por HTML.

Para generar un documento Web, un navegador espera encontrar el conjunto de normas acordado que deberá de seguir. Esto es, para desplegar correctamente un documento, el navegador busca la estructura proporcionada por el HTML. HTML es un lenguaje especial que sirve para crear documentos Web con hiperligas³² (enlaces a otros documentos Web). HTML no es un lenguaje de programación, sino un conjunto de normas utilizadas para dar formato a un documento. [JLW96]

El HTML se derivó de SGML.³³ SGML es un estándar internacional aprobado en 1986. Por tanto, HTML heredó las características de inde-

³²De la palabra en inglés *hyperlinks*.

³³*Standard Generalized Markup Language*: Lenguaje estándar de marcación generalizado.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

pendencia del sistema operativo y del hardware propias de SGML.

Para poder enviar documentos HTML a través de la Internet, fue necesario un conjunto de normas que permitiera la descomposición en paquetes de los documentos en HTML para ser enviados por medio de la Internet. HTTP es el protocolo de red que fue específicamente desarrollado para este propósito.

HTTP es un protocolo inestable, es decir, navegador y servidor deben establecer y después romper una única transacción. Ni el navegador, ni el servidor, son responsables de recordar el último estado de conexión. HTTP se basa en acciones cliente-solicitud y servidor-respuesta. El cliente establece una conexión TCP/IP con un servidor enviando un mensaje de solicitud de conexión por Internet. Si el servidor está disponible, recibe este mensaje del cliente y establece la conexión. [JC95]

La figura 4.5, exhibe el proceso de visualización de una página Web. Una vez establecida la conexión, el cliente desde un navegador solicita un documento de HTML especificándolo vía un URL.³⁴ El navegador interpreta el URL. Por medio del protocolo HTTP, realiza la petición al servidor de páginas Web al que se refiere el cliente en el URL. El servidor recibe la petición, basado en el HTTP establece que documento HTML le fue solicitado y le responde al cliente enviándole el documento vía HTTP. El navegador recibe la respuesta por medio del protocolo HTTP y recompone el documento HTML, el cual es interpretado para desple-

³⁴ *Uniform Resource Locator*: Localizador uniforme de recursos.

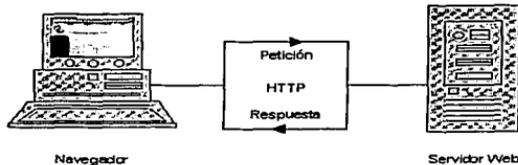


Figura 4.5: Proceso de visualización de una página Web.

gar la información contenida en el documento tal y como se definió por HTML. Es importante mencionar que, una vez atendida la petición del cliente, el servidor rompe la conexión TCP/IP, lo que le permite atender otras solicitudes.

Los servidores Web son parte fundamental de la Web, básicamente es en los servidores donde se almacenan las páginas Web. Pero un servidor Web no es nada más el hardware, también es el software que permite aceptar las solicitudes del cliente, enviar los documentos solicitados y administrar los documentos y directorios que conforman el sitio Web. Algunos servidores de páginas Web que son utilizados, son por ejemplo: Apache, AOLServer, Zope, que son de fuente abierta; los propietarios iPlanet de la alianza Netscape y Sun Microsystems, WebSphere de IBM, y IIS³⁵ de Microsoft que se incluye con Windows NT/2000 y sólo puede ser empleado con tecnología y bajo plataformas Microsoft.

³⁵ *Internet Information Server* o Servidor de información de Internet en español.

4.3.3. La Web hoy en día

El éxito del que disfruta la Web en la actualidad no puede asociarse a un solo hecho únicamente, pues para empezar la Web respondía a la necesidad de un grupo de físicos para compartir sus trabajos de investigación. Además, el CERN estaba conectado a Internet lo que permitió que las universidades y laboratorios conectados a Internet adoptaran también esa manera de compartir documentos ganando así una cantidad considerable de usuarios. Y desde luego que también influyó, la enorme capacidad y el gran empeño de Tim Berners-Lee que no sólo se limitó a especificar su sistema, sino que lo llevo a la práctica y demostró que funcionaba.

En la actualidad la Web no se basa tan sólo en documentos escritos en HTML, existen una gran variedad de aplicaciones y de utilerfas que han venido a revolucionar la Web, haciéndola menos plana y mucho más vistosa y atractiva para los usuarios.

Hoy en día hablamos de *plugins*,³⁶ *scripts*,³⁷ hojas de estilo,³⁸ DHTML,³⁹ XHTML,⁴⁰ XML,⁴¹ entre otras aplicaciones y aditivos que tienen la intención de proveer una Web mucho más agradable. Pero el modelo de comu-

³⁶ Añadidos para presentar cierto tipo de archivos en un navegador de páginas Web o alguna otra aplicación.

³⁷ *Scripts* o guiones: Instrucciones de lenguajes interpretados que permiten algunas características no incluidas por el HTML.

³⁸ En inglés *Style Sheets*.

³⁹ *Dynamic HTML*: HTML Dinámico.

⁴⁰ *eXtended HTML*: HTML extendido.

⁴¹ *eXtended Markup Language*: Lenguaje de marcación extendido.

nicación que se utiliza todavía, sigue siendo el modelo cliente/servidor, donde un cliente hace la petición al servidor, el servidor responde y termina la comunicación.

4.4. La arquitectura cliente/servidor

Las redes fueron creciendo manteniendo una arquitectura base que fue nada más y nada menos que la arquitectura cliente/servidor, por eso no es de extrañar que la base de la Web sea aún en la actualidad, la arquitectura cliente/servidor.

La mayoría de los programadores del Web utilizan el modelo cliente/servidor para diseñar programas para la capa de red (aplicación). La comunicación de red requiere una conexión entre dos computadoras o programas que se comunican entre sí. Una conexión de red la constituyen ambos extremos (o lados) en el proceso de comunicación. [JLW96]

La arquitectura cliente/servidor define una relación entre el cliente y el servidor. El cliente no puede ser una terminal tonta. Debe tener su propia capacidad de procesamiento, pero no tiene que soportar toda la carga como sucede generalmente en el modelo de servidor de archivos. [PT97]

Para evitar un costo excesivo en los protocolos orientados hacia la conexión como OSI o TCP/IP, lo usual es que el modelo cliente/servidor se base en un protocolo solicitud/respuesta sencillo y sin conexión. [TA95]

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

4.4.1. Los inicios

La arquitectura cliente/servidor surge de la necesidad de compartir los recursos, de esta manera los clientes podían acceder a los recursos con los que contaba el servidor (generalmente un equipo mucho mejor equipado y con los mejores recursos tecnológicos que los equipos de los clientes), y así poder reducir costos y aprovechar al máximo los recursos como son: disco, memoria y procesador, sólo por mencionar algunos.

En un principio surgieron las redes de computadoras para compartir recursos que eran extremadamente costosos. De esta manera más de un equipo de cómputo podía aprovechar los recursos con los que se contaban como: las impresoras y el espacio en disco.

El término cliente/servidor surge cuando se conectan a las supercomputadoras⁴² varias terminales tontas alrededor de ellas. Se denominaban terminales tontas porque ningún proceso era llevado en ellas, sólo eran receptáculos de salida de la computadora principal. [MB00]

Las primeras aplicaciones de red estaban basadas en los servidores de archivo y de disco que almacenaban una gran cantidad de información gracias a sus enormes dispositivos de almacenamiento físico, en comparación con los dispositivos de almacenamiento del cliente que muchas veces consistía tan sólo de un disco flexible o una cinta magnética.

⁴²O en inglés *mainframes*.

Muchas de las primeras aplicaciones de red fueron soluciones de cajón. Si ofrecían una base de datos para usuarios múltiples, por ejemplo, tenían la aplicación frontal del usuario⁴³ y el mecanismo⁴⁴ de la base de datos en la misma PC.⁴⁵ La única porción en red era la base de datos que se encontraba en el servidor. [GB94]

Las aplicaciones de escritorio llegaron a ser más poderosas gracias a la evolución de las computadoras de escritorio. La gente necesitaba compartir información de manera más rápida. Esto dió paso a servidores más poderosos, los cuales respondían a las solicitudes de los clientes y las procesaban. Esto es lo que la gente comienza a llamar proceso cliente/servidor. Básicamente un diseño de dos capas, donde el cliente envía solicitudes y el servidor las responde. Toda la lógica del negocio está en el nivel de aplicación del cliente.⁴⁶

El diseño a dos capas prevalece en la actualidad. Éste es conocido también como cliente pesado⁴⁷ porque todos los procesos de las aplicaciones se hacen al nivel del cliente. Después de un tiempo los servidores se convirtieron en parte importante para las empresas debido a sus tareas. Utilizaban lo mejor de lo mejor en hardware y con impresionante velocidad. Los procesos de las aplicaciones se movieron a los servidores. El cliente podría ser sólo la interfaz gráfica del usuario (GUI⁴⁸) y la aplica-

⁴³También aplicación frontal o en inglés *Front-end*.

⁴⁴También aplicación posterior o *Back-end* en inglés.

⁴⁵*Personal Computer*: Computadora personal.

⁴⁶llamado también cliente ligero o en inglés *Light Client*.

⁴⁷*Heavy Client* en inglés.

⁴⁸De sus siglas en inglés *Graphic User Interface*.

TESIS CON FALLA DE ORIGEN

ción principal o lógica de negocios podría ser procesada en el servidor. Éste podría hacer las llamadas necesarias a otros servidores de bases de datos o archivos según se necesitara. Esto dio origen al nacimiento del diseño en tres capas o cliente ligero. [MB00]

4.4.2. Pros y contras del sistema cliente/servidor

Éstas son las ventajas de los sistemas cliente/servidor:

- La seguridad es mayor porque los datos de la base de datos del servidor son accedidos de manera indirecta. Los usuarios no pueden ver en realidad los archivos de datos a menos que se les dé acceso explícito.
- Se puede mejorar el rendimiento, ya que un servidor de aplicación posterior bien diseñado puede proporcionar una mejor condición de usuarios múltiples y, por lo tanto, un mejor rendimiento. En el caso de los servidores de bases de datos, las PCs cliente no tienen que leer todos los datos de la base de datos para encontrar lo que quieren; pueden enviar solicitudes al servidor, y el servidor entrega solamente lo que necesitan.
- La efectividad en relación con el costo es mucho mayor. Los clientes sólo necesitan el poder suficiente para ejecutar adecuadamente la aplicación frontal. (Cuando se requiere de un alto rendimiento, el servidor de base de datos puede implementarse en una PC muy poderosa y proporcionalmente costosa.)

Las desventajas de los sistemas cliente/servidor son estas:

- **Complejidad:** Los sistemas cliente/servidor generalmente no son fáciles de configurar ni administrar.
- **Requerimientos:** Para dar servicio a muchos usuarios, el componente cliente de un sistema cliente/servidor frecuentemente necesita ejecutarse en una PC costosa. Las aplicaciones de servidor tienden a ser grandes y complejas, y generalmente necesitan mucha memoria.
- **Costo:** El rendimiento del servidor se reduce conforme aumenta el número de usuarios. Para recuperar los altos niveles de rendimiento, el software del servidor tal vez tenga que ejecutarse en una máquina dedicada especialmente a ese servicio. Así que donde una vez había un servidor de archivos general que también ejecutaba el software de servidor de base de datos, ahora habría un servidor de archivos general y un servidor de base de datos especial, elevando el costo del hardware por lo menos al doble. [GB94]

4.4.3. El modelo cliente/servidor hoy en día

El modelo cliente/servidor ha revolucionado pasando de un modelo de 2 capas o niveles a un modelo de 3 o más capas, donde las máquinas más poderosas manejan la lógica de negocios y las máquinas más lentas, las de los clientes, únicamente despliegan la salida.

La Web es un ejemplo selecto de cliente ligero, donde un navegador (cliente ligero) hace peticiones a un servidor de páginas Web que las

procesa y envía las respuestas de regreso al navegador, el cual despliega la información. [MB00]

Gracias a la disminución de los precios del hardware y al gran éxito de la Internet, muchas empresas han decidido tener una presencia en la Web, por lo que han pasado de una arquitectura a dos capas a una arquitectura de tres o más capas.

Una arquitectura o modelo de n capas, también llamado modelo a multicapas, es aquel modelo donde las diferentes aplicaciones requeridas por un cliente son procesadas en un servidor dedicado solamente a esa labor. Esto permite un mayor rendimiento en el desempeño de la aplicación además de una administración más sencilla. Cada capa es representada por un servidor aparte de la capa representada por el cliente.

La figura 4.6, en la página siguiente, muestra como ha sido la evolución de los diferentes modelos de cliente/servidor. Comenzando con el modelo básico de 2 niveles o capas, hasta el modelo de n niveles, pasando lógicamente por el modelo de 3 capas. En la figura n es igual a 4, es decir, las 3 capas de los servidores más la capa del cliente.

Se debe considerar que mientras una aplicación de modelo de negocios tenga más niveles su rendimiento será mayor, pero también el costo es mayor ya que cada servidor tiene un precio considerable. Aunque en la actualidad se pueden usar PCs como servidores, gracias a que ya existen

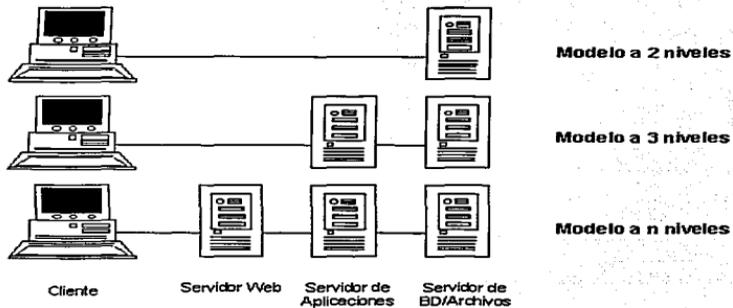


Figura 4.6: Evolución del modelo cliente/servidor.

TESIS CON
FALLA DE ORIGEN

computadoras con procesadores muy veloces y gran cantidad de memoria. Lo cual reduciría el costo inicial, pero se tiene que considerar que en ocasiones el desempeño de servidores caros es mayor, recuperando la inversión en un lapso de tiempo relativamente corto y superando a los servidores de bajo costo inicial.

Al software que se utiliza para hacer funcionar o comunicar a los diferentes servidores de cada capa y a la capa del cliente se le llama *middleware*. El *middleware*⁴⁹ puede considerarse como el pegamento que hace funcionar el modelo cliente/servidor de multicapas. El *middleware* es lo que permite la comunicación con el servidor de componentes distribuidos.

Es en los servidores de objetos o componentes, también llamados servidores de aplicaciones que corren del lado del servidor, donde los componentes de software de la aplicación distribuida son almacenados, ejecutados y entregados.

4.5. Programación de componentes

La programación de las computadoras nunca ha sido una tarea sencilla, pero gracias al esfuerzo de mucha gente y a los avances tecnológicos, esta tarea ha pasado a ser más llevadera para las personas que se dedican a programar.

A lo largo de la historia computacional han aparecido muchos métodos

⁴⁹puede traducirse como software de enmedio.

de programación, sin mencionar a los lenguajes de programación que buscaban resolver ciertos problemas específicos, o como algunos otros lenguajes de propósito general que se podían emplear para resolver múltiples problemas.

Sin embargo, una de las metodologías con más auge entre los programadores por su facilidad de uso y su disponibilidad para la reutilización de código, es la programación orientada a objetos (POO⁵⁰), que consiste básicamente en coger el problema y a los actores que intervienen en éste para representarlos como objetos y resolver el problema con base en su interacción entre ellos, como en la realidad misma.

Uno de los últimos paradigmas en programación, ha sido la de construir aplicaciones a través de componentes de software. Un componente es un módulo lógico autónomo sin llegar a ser una aplicación completa. Los componentes facilitan la construcción de aplicaciones ya que se pueden integrar con otros componentes, sin importar quien los desarrolló, y formar así una aplicación completa. En una analogía, un componente de software equivaldría a un componente electrónico que se une a otros componente para formar un dispositivo electrónico que equivaldría de manera análogamente inversa a una aplicación.

4.5.1. Los inicios

En un principio las enormes computadoras eran programadas con tarje-

⁵⁰en inglés *OOP*, es decir, las siglas de *Object-Oriented Programming*.

TESIS CON FALLA DE ORIGEN

tas perforadas que emulaban los ceros y unos del lenguaje que la máquina interpretaba y después convertía en una instrucción. Los 1's y 0's del lenguaje máquina representaban la ausencia o presencia de voltaje, mismo que hacía funcionar a las computadoras. Posteriormente, para facilitar la labor de los programadores, se crearon mnenónicos que representaban una instrucción específica del código máquina, fue entonces cuando nacieron los ensambladores y el código ensamblador.

El código ensamblador representó una gran ayuda para los programadores, aunque aún era muy complicado y extenuante realizar programas que satisficieran las necesidades que se presentaban, las cuales eran cada vez mayores. Además de que el lenguaje ensamblador dependía de la arquitectura y diseño de la computadora. Es por tal motivo que algunos programadores comienzan a desarrollar lo que se conoce como lenguajes de programación de alto nivel.

En la programación de alto nivel se generaron instrucciones que son más fáciles de entender para los seres humanos que para las computadoras. Cada instrucción, representada por una palabra en inglés generalmente, correspondía a una serie de instrucciones en ensamblador y a cadenas de unos y ceros en el lenguaje de bajo nivel. Por medio de los compiladores o interpretes los programas de alto nivel son traducidos a instrucciones que la máquina puede reconocer.

Los compiladores son programas que toman todo el código de alto nivel, lo convierten en código ensamblador (algunos hasta optimizándolo)

y generan un programa ejecutable que la máquina puede interpretar y ejecutar. Los interpretes también toman el código de alto nivel, sólo que a medida que lo van interpretando lo van convirtiendo a código de máquina y van ejecutando las instrucciones correspondientes. Los lenguajes de alto nivel son independientes de la plataforma en cierta modo, pues es el compilador o interprete el encargado de hacer la traducción a código máquina.

Aunque lenguajes como C o C++ cuentan con compiladores en todas las plataformas existentes, para ejecutar los programas escritos bajo una plataforma A en otra distinta B, se requiere hacer algunos cambios en el código del programa de la plataforma A y recompilarlo en la nueva plataforma B. El lenguaje Java es una excepción, debido a que puede ser interpretado o compilado y ejecutado bajo cualquier plataforma que cuente con una máquina virtual Java sin necesidad de modificar código alguno.

Existen un sin número de diferentes lenguajes de alto nivel. Cada lenguaje utiliza algún método de programación que se caracteriza por la manera en que se encara a los problemas. Se tienen por ejemplo al método funcional que utiliza funciones matemáticas para resolver el problema a programar, un lenguaje con soporte funcional es FORTRAN.⁵¹ También se tiene el método de programación procedural que descompone el problema, representando cada una de las partes que componen el problema y la solución como procedimientos, un ejemplo de lenguaje con

⁵¹del inglés: *FORmula TRANslator*, es decir, traductor de fórmulas.



soporte procedural es C.

Desde luego no se puede dejar de mencionar al método o modelo de programación que más éxito ha tenido entre los programadores por su forma tan intuitiva de descomponer y solucionar los problemas, se trata del método de la POO.

4.5.2. La Programación Orientada a Objetos

El modelo de programación orientada a objetos, o POO⁵² para abreviar, consiste en representar todo como objetos. Los objetos tienen atributos, propiedades y realizan ciertas tareas. Las tareas que un objeto puede realizar son denominadas como métodos.

En la programación orientada a objetos el concepto fundamental es, como no, el objeto. Un objeto puede ser definido como un conjunto de datos más un conjunto de subprogramas que rigen el comportamiento sobre estos datos. Los datos y las operaciones están estrechamente ligados en esa entidad fundamental: el objeto. En la terminología más clásica de la programación se denomina miembro de un objeto⁵³ tanto a un elemento de datos que proporciona su estado, como a una función que opera sobre ellos. [SB02]

Aunque no existe un consenso sobre lo que es un objeto, éste debe cumplir con los siguientes paradigmas:

⁵² en inglés *Object-Oriented Programming* u OOP.

⁵³ A los datos se les llama *data members* y a las funciones *function members*.

- Encapsulación.
- Herencia.
- Polimorfismo.
- Instanciación.

La tecnología de orientación a objetos busca conseguir desarrollos más rápidos y más fáciles de mantener. A últimas fechas esta finalidad se ha visto enormemente complicada por las arquitecturas distribuidas que pretenden integrar multitud de sistemas heterogéneos. Parece que una de las soluciones es la creación de componentes que puedan ser utilizados en el diseño y desarrollo de cualquier aplicación.

4.5.3. Los Componentes de Software

Un componente es un módulo lógico autónomo que puede ser instalado en diferentes plataformas. Un componente no está ligado a ninguna implementación, programa o lenguaje en particular. Para que un programa sea considerado estrictamente como componente debe cumplir las siguientes características:

- Puede ser instalado en diferentes plataformas.
- Puede ser configurado.
- Debe ser capaz de autodescribirse.
- No debe ser una aplicación completa.

- Posee una interfaz bien definida.
- Permite la notificación de eventos.

El utilizar componentes tiene las ventajas de que facilita la actualización de versiones del software. Si consideramos a los componentes de software como bloques independientes, los cuales se pueden ir ensamblando de manera que se integre una aplicación completa.

4.5.4. La programación de componentes hoy en día

La programación basada en componentes es en la actualidad parte fundamental de las aplicaciones distribuidas. La Web es una herramienta que ha permitido el intercambio de componentes, beneficiando el crecimiento del uso de componentes de software. Empero, asimismo los componentes han beneficiado a la Web proporcionando una manera distribuida de crear aplicaciones para ésta.

Los componentes de software proporcionan una manera muy efectiva de incrementar la escalabilidad y el desempeño de los sitios Web.

En la actualidad les damos el nombre de servicios Web a aquellas aplicaciones distribuidas que emplean a la Web y a la Internet como elementos indispensables de la aplicación. Entornos de componentes como COM, .NET, CORBA, CCM y los EJB se integran con otras herramientas para crear servicios Web basados en componentes.

Parte III

Análisis de las tecnologías

Capítulo 5

Tecnologías de desarrollo

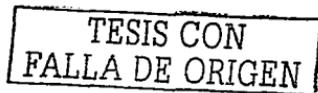
5.1. Introducción

En el presente capítulo se presentan algunas definiciones y características de las tecnologías de desarrollo de aplicaciones distribuidas basadas en objetos y componentes de Microsoft, la OMG y SUN Microsystems, es decir, se presentan COM¹ y .NET desarrolladas por Microsoft; CORBA y CCM desarrolladas por la OMG; y por último se presenta a la tecnología desarrollada por SUN Microsystems, los Enterprise JavaBeans.

5.2. COM, DCOM, COM+ y .NET

A continuación se presenta una descripción de las tecnologías de desarrollo de aplicaciones basadas en objetos y componentes que Microsoft (la empresa de software norteamericana) ha sacado al mercado.

¹Por COM se entiende a esta tecnología o cualquiera de sus variaciones, por ejemplo: DCOM y COM+, tal y como se explica en la sección 5.2.



TESIS CON FALLA DE ORIGEN

5.2.1. Antecedentes

Cómo Microsoft utiliza una nomenclatura no muy clara para sus productos, a continuación se presenta una descripción breve de la tecnología desarrollada por esta empresa norteamericana.

COM, o Modelo de Objetos Componentes,² es el primer desarrollo del gigante del software Microsoft para el manejo de componentes. COM pretendía proporcionar a los programadores, objetos o componentes que facilitaran la labor de ellos. Sin la necesidad de invertir mucho tiempo en la elaboración de los aspectos de presentación los programadores podían enfocarse más en el aspecto medular de su programa.

COM es una especificación para construir componentes OLE.³ Es una especificación orientada a objetos para definir interfaces: los puntos de contacto entre los componentes. Las interfaces COM ofrecen una única solución al problema de conectar componentes de software ejecutándose en diferentes procesos separados por un límite de espacio de direcciones. Las interfaces son a la vez un contrato de servicios y la conexión binaria entre el proveedor del servicio y el usuario del servicio. [PD01]

DCOM⁴ es lo mismo que COM sólo que con la diferencia de que DCOM es utilizado para realizar aplicaciones distribuidas. Es por eso la letra

²en inglés, *Component Object Model*.

³de sus siglas de las palabras en inglés: *Object Linking and Embedding*, que significan: incrustación y vinculación de objetos.

⁴de sus siglas en inglés *Distributed Component Object Model*, que quiere decir: Modelo de objetos componentes distribuidos.

“D” que antecede a COM. DCOM provee un mecanismo para soportar conexiones fuera del proceso, es decir, entre distintos equipos informáticos.

COM aparece en 1993, y a partir de 1997 Microsoft comienza a hablar de COM+ que sería la evolución de COM, pero que finalmente fue como se le denominó a la nueva versión del servidor transaccional de Microsoft o MTS.⁵ MTS permite a los programadores implementar transacciones dentro de sus aplicaciones de una manera sencilla y más barata que con los sistemas transaccionales tradicionales. [PD01]

MTS se creó para subsanar una serie de servicios que no presentaba COM, pero COM nunca se modificó para trabajar con MTS. Por tal motivo la integración de ambos se realizó en COM+, incluido en Windows 2000. [TJM01]

Finalmente, a principios del 2002 Microsoft lanza al mercado lo que se supone sería COM+ y terminó siendo llamado .NET. Éste proporciona una serie de herramientas que facilitan el desarrollo de aplicaciones Web y la migración de aplicaciones que no tienen una presentación Web a una de tipo Web y viceversa. La gran diferencia de .NET con respecto a las tecnologías pasadas desarrolladas por Microsoft es la independencia de la plataforma y del lenguaje de programación.

Para facilitar un poco la comprensión de este trabajo, cuando se haga

⁵por las siglas de las palabras en inglés: *Microsoft Transaction Server*.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

referencia a COM se trata de la integración de COM, DCOM y COM+ que permiten a los programadores ignorar las diferencias basadas exclusivamente en la localización del componente, así como proporcionar acceso a la funcionalidad del sistema operativo, cosas como colas y transacciones, consiguiendo que los programadores centren su atención en la propia tarea.

5.2.2. Características

En este punto se dividen las características en dos, las características de COM⁶ y las características de .NET, esto básicamente por la diferencia respecto a la independencia, tanto de la plataforma, como del lenguaje de este último producto de Microsoft. Aunque es cierto, de alguna manera, que .NET es la evolución de COM, es preferible hablar por separado de cada una de estas tecnologías.

5.2.2.1. Características de COM

Facilidad de uso. COM fue desarrollado para proporcionar a los programadores una manera sencilla de realizar aplicaciones, empleando componentes comunes, con IDEs amistosas que permiten arrastrar y pegar componentes.

Reutilización de código. Una de las ventajas que poseen todas la tecnologías de desarrollo basadas en componentes es la reutilización de

⁶Refiriendonos por COM a COM, DCOM o COM+.

componentes ya sea propios o de los adquiridos al comprar una licencia de un IDE.

Uso de componentes de terceros. Además de poder reemplazar componentes propios y de Microsoft, se pueden conseguir, comprándolos o de manera gratuita, componentes desarrollados por terceras personas como desarrolladores independientes o empresas de desarrollo.

5.2.2.2. Características de .NET

.NET cuenta con las mismas características de COM, pero además cuenta con algunas características extras, las cuales son explicadas a continuación:

Estándar. .NET fue presentado como un estándar EMAC por lo que se pueden realizar implementaciones de .NET de forma independiente, un ejemplo de ello es MONO. MONO es una implementación de .NET realizada por XIMIAN, empresa desarrolladora de soluciones de código fuente libre.

Independencia del lenguaje. (CLR⁷) Quizá la característica más atractiva de .NET sea que se pueden desarrollar aplicaciones en cualquier lenguaje de programación para el cual exista un CLR. El CLR compila el código fuente y crea un lenguaje intermedio denominado MSIL.⁸

⁷Common Language Runtime.

⁸de las siglas de *Microsoft Intermediate Language*, es decir, lenguaje intermedio de Microsoft.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

Independencia de la plataforma. (JIT⁹). Para cualquier plataforma en la que exista un JIT, .NET podrá ser utilizado. El JIT es una especie de máquina virtual que hace la función de interprete entre la aplicación .NET ya traducida en MSIL y la plataforma, generando el código máquina particular de la plataforma.

Uso de XML. XML ha venido ganando terreno en las aplicaciones. .NET ha hecho uso de XML para darle una mayor flexibilidad en el manejo de datos independientes de la plataforma.

Compatibilidad con COM. Sin lugar a dudas ésta es una característica importante para mantener la compatibilidad con aplicaciones desarrolladas en el pasado con COM, la anterior tecnología de componentes de Microsoft.

5.2.3. Aspectos a considerar

Aunque COM representa una herramienta bastante útil para desarrollar aplicaciones ensamblando componentes, aún existen algunos inconvenientes para los programadores como es el hecho de algunas incompatibilidades entre las implementaciones del cliente y el servidor si éstos están desarrollados en dos diferentes lenguajes como Visual C++ y Visual Basic.

⁹Just-in-time, o compilación justa en tiempo.

Es algo inapropiado concluir respecto a .NET porque es muy poco el tiempo que tiene en el mercado, sin embargo, con base en la lectura de múltiples artículos y libros que van desde los promocionales hasta los antagonicos, pasando por los veraces y netamente comparativos, se puede decir que .NET es una estrategia inteligente y bien planeada de Microsoft para recuperar el mercado de desarrollo de soluciones distribuidas y en especial para aquellas que hacen uso de la Internet.

.NET fue desarrollado por Microsoft por lo que es una plataforma propietaria, pero como también fue presentado como un estándar EMAC otros distribuidores pueden realizar soluciones 100% compatibles con .NET. Un ejemplo de una implementación de .NET es MONO, cuya licencia pretende ser libre. MONO está siendo desarrollado por Ximian, una de las empresas de software libre más reconocidas internacionalmente.

5.3. CORBA y CCM

5.3.1. Antecedentes

Ahora se muestran brevemente algunas definiciones y siglas propias de la tecnología de *CORBA*.¹⁰

CORBA fue desarrollado por la *OMG*¹¹ como una alternativa abierta

¹⁰ *Common Object Request Broker Architecture*, es decir, Arquitectura intermediadora de solicitud de objetos comunes.

¹¹ *Object Management Group*, es decir, Grupo de Administración de objetos. Grupo conformado por compañías competidoras de Microsoft que buscaban una alternativa estándar y económica a COM.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

a COM. Independiente de la plataforma y del sistema operativo empleado. La gran diferencia entre COM y CORBA es que éste último fue creado como un estándar que todo el mundo podía seguir.

La parte esencial de CORBA se encuentra en el ORB.¹² El ORB se emplea para conectar objetos entre sí vía una red. [JJ98]

IIOP.¹³ El protocolo estándar de comunicación de CORBA que permite la comunicación entre ORBs.

El *IDL*¹⁴ es un lenguaje que permite definir las interfaces necesarias para que el objeto cliente pueda comunicarse con el ORB que se comunicará a su vez con el objeto remoto.

*OTS*¹⁵ es uno de los servicios que CORBA ofrece y que permite el empleo de transacciones. Es la implementación de la OMG del MTS de Microsoft. OTS del mismo modo que CORBA también fue presentado como un estándar.

CCM, de las siglas de *CORBA Component Model*,¹⁶ es una extensión de CORBA para emplear componentes además de objetos. CCM está fuertemente inspirado en la especificación de los *Enterprise JavaBeans* (EJB) y fue definido a partir de la versión de CORBA 3.0.

¹² *Object Request Broker*, que en español quiere decir, negociador de solicitudes de objetos.

¹³ *Internet Inter-ORB Protocol*, es decir: Protocolo Inter-ORB Internet.

¹⁴ *Interface Definition Language*, o en español, lenguaje de definición de interfaces.

¹⁵ de *Object Transaction Service*, o sea, servicio de transacción de objeto.

¹⁶ Modelo de componentes CORBA.

*CIDL*¹⁷ es la extensión del IDL para el manejo de componentes, lo que explica la letra "C" en referencia a los componentes. Al igual que CCM está definido desde CORBA 3.0.

5.3.2. Características

A continuación se presentan las características de CORBA y CCM. Primero se presentan las características de CORBA¹⁸ y sus servicios y posteriormente se muestran las características de CCM.

5.3.2.1. Características de CORBA

Al ser una respuesta de la OMG a COM, lógicamente CORBA presenta las mismas características que COM como son: la reutilización de código y el uso de componentes de terceros pero además muestra las características que se mencionan a continuación:

Estándar. CORBA fue concebido desde un principio como un estándar de la OMG al que todos podían apegarse. Es reconocido internacionalmente y admitido por casi todas las principales marcas.

Independencia de la plataforma. (*ORB*) Existen ORBs disponibles para un gran número de sistemas operativos por lo que el soporte para las

¹⁷del inglés Component Interface Definition Language, que en español quiere decir Lenguaje de definición de interfaz de componentes.

¹⁸Por CORBA entenderemos a todas aquellas versiones anteriores al CORBA 3.0, este último define a CCM.

múltiples plataformas existentes está casi garantizada.

Independencia del lenguaje. Se pueden conectar objetos desarrollados en cualquier lenguaje de programación siempre y cuando se pueda suministrar a los objetos un fragmento adaptador del IDL. La OMG ha estandarizado mapeos entre IDL y lenguajes como C, C++, Java, COBOL, Smalltalk, Ada, Lisp, Python, entre otros lenguajes más.

5.3.2.2. Características de CCM

Al estar basado fuertemente en las especificaciones de los EJB posee las características que se incluyen en los EJB, pero además CCM posee las características de CORBA extendidas al uso de componentes de software como son:

El uso de XML. Al igual que en los EJB, XML es empleado para definir los tipos de documentos (*DTD*¹⁹). CCM crea un archivo XML que tiene como finalidad guardar la información necesaria para describir el componente.

Mapeo con Java. Permite a los desarrolladores construir aplicaciones enteramente en Java y a continuación generar la forma IDL de las clases Java de manera automática. Esto permite que aplicaciones binarias accedan a aplicaciones Java usando RMI²⁰ sobre IIOP.

¹⁹De las siglas en inglés de *DocumentType Definition*.

²⁰Por las siglas en inglés de *Remote Method Invocation*. Invocación de métodos remoto.

Real-Time ORB.²¹ Este nuevo ORB permite al desarrollador tener un control mas directo sobre la asignación de recursos.

5.3.3. Aspectos a considerar

CORBA es una herramienta poderosa para la integración de aplicaciones, sin embargo, la lentitud con la que algunos de los proveedores de servicios CORBA adoptan los estándares hace que pocos sean los que se atreven a emplearlo en las empresas.

Aunque CORBA es un estándar, existen algunas incompatibilidades entre ORBs de diferentes proveedores ya que algunos proveedores agregan funcionalidades no estandarizadas, por lo que pueden existir algunos problemas de comunicación entre ORBs distintos causando que la aplicación no funcione adecuadamente.

No existen marcos de aplicación²² amigables para el usuario, por lo que CORBA tiene pocos usuarios.

CCM no cuenta con tantas implementaciones como CORBA, de hecho, sólo se consideró en esta investigación a la primera implementación de CCM creada por la Universidad de Lille en Francia, llamada OpenCCM. Sin embargo, no todas las especificaciones de CCM están implementadas en OpenCCM, aunque no se duda que así será en un futuro.

²¹ORB de tiempo real.

²²en inglés *application framework*.

TESIS CON
FALLA DE ORIGEN

5.4. Enterprise JavaBeans

5.4.1. Antecedentes

A pesar de ser coparticipes en el proyecto de la OMG para crear una solución para la creación de aplicaciones distribuidas, al ver que los avances eran lentos, *SUN Microsystems* (SUN) decide crear su propia solución. Nace así RMI²³ para el lenguaje multiplataforma de SUN, *Java*.

RMI permite acceder a un objeto en una máquina remota, permitiendo así la creación de aplicaciones distribuidas, pero relativamente limitadas. Posteriormente SUN crea los *Enterprise JavaBeans* (EJB) que proporcionan una solución distribuida más elegante y basada en el modelo cliente/servidor.

Los EJB fueron desarrollados por SUN como una alternativa a COM y a CORBA. Es una especificación para componentes almacenados en un servidor. Esta especificación establece una manera estándar del cómo crear y administrar componentes para una aplicación distribuida orientada a objetos. [AW99]

Pueden verse a los EJB como la evolución de los *JavaBeans*. Los *JavaBeans* son un estándar para definir componentes para aplicaciones del lado del cliente. Los EJB también son un estándar para definir componentes para aplicaciones, pero del lado del servidor.

²³Por las siglas en inglés de *Remote Method Invocation*. Invocación de métodos remoto.

5.4.2. Características de los EJB

Los EJB también cuentan con las características de reutilización de código y el uso de componentes de terceros, pero además cuentan con características propias, tales como lo son:

La independencia de la plataforma. (JRE²⁴) Existen máquinas virtuales de Java para casi todas las plataformas existentes. Por eso, cualquier aplicación escrita en Java puede correrse en la plataforma para la que existe

Uso de XML. Java fue uno de los primeros lenguajes en adoptar a XML, de hecho los Enterprise JavaBeans se definen en XML, es decir, en XML se guarda la información necesaria para describir el componente o Bean como también se conocen a los componentes de Java.

Seguridad. Java proporciona librerías para seguridad, además los EJB copian de JDK 1.1²⁵ el concepto de identidad que establece que dependiendo de la identidad asociada son los derechos que tiene cada Bean.

5.4.3. Aspectos a considerar

Java cuenta con un número muy extenso de usuarios. Por lo que existen una gran cantidad de páginas Web, artículos, libros, capacitación y asesoría respecto al lenguaje y a sus diferentes herramientas, incluyendo

²⁴Java Runtime Environment, Ambiente de tiempo de ejecución Java.

²⁵Java Development Kit version 1.1, es decir, versión 1.1 del paquete para desarrollo Java.

TESIS CON FALLA DE ORIGEN

por supuesto a los EJB.

Al ser abierto, todos pueden tener acceso a las especificaciones de los EJB. Pero a diferencia de CORBA, al ser controlado por SUN, no tiene los problemas de incompatibilidades y las nuevas características o funcionalidades son adoptadas rápidamente y no de manera lenta como en CORBA.

Capítulo 6

Comparativa entre tecnologías

6.1. Introducción

En este capítulo se hace una comparativa entre las diferentes tecnologías de creación de componentes de media capa. Se presentan cuadros comparativos donde se consideran aspectos que resultan importantes a la hora de decidir que tecnología emplear.

6.2. Aspectos a comparar

Algunos aspectos que consideraremos para realizar las comparaciones entre las tecnologías de desarrollo de aplicaciones distribuidas basadas en componentes ya antes mencionadas son los siguientes:

- Precio.
- Soporte, documentación y formación.

TESIS CON
FALLA DE ORIGEN

- Plataformas soportadas.
- IDEs existentes.
- Protocolos de comunicación aceptados.

6.2.1. Precio

Sin duda, dada la situación económica por la que se atraviesa en México, el aspecto de los dineros es un factor determinante a la hora de seleccionar una tecnología cualesquiera.

Aunque el precio de cada una de las tecnologías es variable de acuerdo a los proveedores y a los elementos que deseamos integrar. A continuación se presenta un cuadro (6.2), en el que se comparan algunas de las opciones que se tienen para realizar una aplicación Web. No se establece un precio fijo porque depende de los elementos a aplicar en una plataforma en particular. Pero se considera el hecho de que para algunas tecnologías existen varios vendedores y las alternativas de precios son mayores.

La forma en que se evalúa es con base en las alternativas en precio de los sistemas operativos, las bases de datos y los IDEs que funcionan para cada una de las tecnologías, pero medularmente en la existencia de alternativas en el servidor de aplicaciones. Las calificaciones a otorgar en orden descendente son: Excelente, Bueno, Regular y Malo.

Respecto a COM cabe mencionar que no existe una implementación de

Tecnología	Flexibilidad en precios
COM	Malo
.NET	Regular
CORBA	Buena
CCM	Regular
EJBs	Excelente

Cuadro 6.2: Comparación de acuerdo a su flexibilidad en precios.

código fuente abierto¹ o de software libre.² Por lo que el precio de sus productos es único y relativamente mayor a las tecnologías no propietarias. Además, el hecho de que COM no tenga múltiples proveedores hacen que su precio sea definido por Microsoft en una especie de monopolio, por lo cual se considera que no proporciona una flexibilidad de acuerdo a las necesidades particulares de la aplicación a desarrollar. Por lo tanto, calificamos como malo el precio de COM.

En referencia a .NET es el mismo caso que COM, aunque Mono pretende ser la implementación libre de .NET, aún no es del todo funcional por lo que se le da la calificación de regular. Dicha calificación puede mejorar a medida que aparezcan implementaciones de .NET (CLR's e IDE's independientes de Microsoft) ya que fue presentado como un estándar EMAC que puede ser implementado por otras compañías u organizaciones independientes de Microsoft.

CORBA es mucho más flexible respecto al número de proveedores de ORBs y de IDLs, aunque existen algunos problemas de incompatibili-

¹o en inglés *Open Source*.

²en inglés *Free Software*.

TESIS CON
FALLA DE ORIGEN

dades entre algunos ORBs nada debería ocurrir si sólo se emplean las funcionalidades estándares y se dejan a un lado las funcionalidades extras ofrecidas por los distintos fabricantes de ORBs. Por lo que se otorga una calificación de bueno a esta tecnología.

CCM no posee tantos vendedores como CORBA, sin embargo, dispone de una implementación de código abierto como lo es OpenCCM, pero no está del todo completa de acuerdo a las especificaciones de la OMG. Por lo tanto, CCM obtiene una calificación de regular respecto a la flexibilidad en precios. Su calificación puede mejorar a medida que se incremente el número de implementaciones.

Los EJB tienen más tiempo en el mercado que CCM y .NET por lo que cuenta con varios proveedores, algunos de ellos libres como es el caso de JBoss o algunos otros abiertos como en el caso de JONAS. Si además agregamos los varios IDEs existentes para crear las aplicaciones, algunos de ellos con licencia pública como NetBeans, y el gran apoyo que ha recibido de la comunidad de software libre para ayudar a la integración con otras herramientas libres como Apache, Tomcat, MySQL, PostgreSQL, entre muchas otras más, los EJB obtienen la calificación de excelencia en la flexibilidad de precios.

6.2.2. Soporte, Formación y Documentación

Una empresa siempre quiere contar con gente preparada y capaz o por lo menos que sepa que puede encontrarla cuando lo requiera. Por eso el

soporte es un aspecto que se debe considerar, además de la existencia de formación y documentación para capacitar a los empleados que trabajarán con la tecnología seleccionada.

La manera en como se realiza la valuación de cada tecnología en este punto tiene que ver con el número de empresas que proporcionan el servicio de soporte y formación, así como por el número de publicaciones, considerando que a mayor oferta menor costo. Las calificaciones otorgadas de acuerdo a la oferta pueden ser: Excelente, Bueno, Regular y Malo.

Existen una variedad muy grande de publicaciones respecto a COM, .NET, CORBA, CCM y EJB, se tienen publicaciones impresas y electrónicas tanto de los propios creadores de las tecnologías como también de editores y analistas independientes. Empero, por ser relativamente nuevas, CCM y .NET no cuentan con tantas publicaciones como COM y los EJB. CORBA no cuenta con tantas publicaciones como los EJB, mas en muchas ocasiones dentro de las publicaciones de Java se abarcan temas de CORBA.

Tecnología	Documentación	Soporte	Formación
COM	Excelente	Bueno	Buena
.NET	Buena	Bueno	Buena
CORBA	Buena	Bueno	Regular
CCM	Buena	Regular	Mala
EJB	Excelente	Bueno	Buena

Cuadro 6.4: Comparación de documentación, soporte y formación.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

84 CAPÍTULO 6. COMPARATIVA ENTRE TECNOLOGÍAS

En referencia a la capacitación y entrenamiento Microsoft provee de cursos y certificaciones a sus usuarios tanto de COM como de .NET. Sin embargo, el precio de una certificación es relativamente costosa, por lo que al de por sí solo alto costo de las tecnologías, se tiene que agregar el costo por capacitación de los desarrolladores. Algunas empresas distintas a Microsoft proporcionan este mismo servicio, pero aún así los precios son elevados. Por tal motivo las calificamos con buenas notas.

CORBA y CCM cuentan con una cantidad bastante reducida y casi nula variedad de prestadores de servicio de formación respectivamente, por eso son calificadas en este tópico con calificaciones de regular y mala respectivamente.

SUN proporciona certificaciones y varios cursos de Java y sus tecnologías, además existen también algunas empresas diferentes a SUN que proporcionan el mismo servicio. Aunque los precios no dejan de ser elevados, lo mismo ocurre para las otras alternativas por lo que se le otorga la nota de buena respecto a la formación.

En el soporte, todas obtienen una calificación buena ya que tanto sus creadores, como empresas independientes ofrecen sus servicios de solución de problemas. La única excepción es CCM, cuya implementación (OpenCCM) ofrece soporte únicamente a través de una lista de correo electrónico moderada por el grupo GOAL de la Universidad de Lille.

6.2.3. Plataformas soportadas

En un ambiente Web, en el que existen un sin número de plataformas diferentes, el hecho de poder emplear cualquiera de las muy diversas plataformas existentes es también un factor sumamente importante a considerar porque algunas plataformas son más robustas o más baratas que otras.

A continuación compararemos las tecnologías de desarrollo con base en las plataformas en que pueden emplearse.

Tecnología	Plataformas soportadas
COM	Windows y Solaris
.NET	Windows y Linux
CORBA	Varias
CCM	Windows y Linux
EJB	Varias

Cuadro 6.6: Comparación de plataformas soportadas.

COM es soportado por las plataformas en las que los sistemas operativos Windows y Solaris pueden correrse. En el caso de .NET por el momento se puede correr en las plataformas en que se tengan instalados Windows o Linux³ (gracias a la implementación de Ximian). CORBA y los EJB están soportados en las plataformas más variadas en las que existen ORBs y JVMs, por ejemplo: Linux, Solaris, Windows y otras más. CCM por lo pronto es soportado por Windows y Linux.

³Linux es un sistema operativo del tipo Unix, como BSD, por lo que también Unix y sus variantes están incluidos.

TESIS CON FALLA DE ORIGEN

Tecnología	IDEs existentes
COM	Visual Studio, JBuilder, Delphi, C++Builder
.NET	Visual Studio .NET
CORBA	C++Builder, JBuilder, Delphi, Visual Cafe, etc.
CCM	C++Builder, JBuilder, Delphi, Visual Cafe, etc.
EJB	JBuilder, Visual Cafe, VisualAge, NetBeans, etc.

Cuadro 6.8: Comparación de IDEs existentes.

6.2.4. IDEs existentes

El desarrollo de aplicaciones requiere de un tiempo considerable, más aún cuando se trata de proyectos muy grandes o complejos, por eso siempre resulta útil contar con un ambiente de desarrollo de interfaces o por sus siglas en inglés IDE.⁴ Un IDE permite un desarrollo mucho más rápido, reduciendo así en forma considerable los costos de desarrollo estrechamente ligados al costo total de la aplicación.

COM ofrece varias IDEs para el desarrollo de aplicaciones como: Visual Basic, Visual C++ y Visual J, incluidos en Visual Studio. Además algunas empresas distintas a Microsoft permiten crear componentes COM en sus IDEs, como es el caso de Borland Delphi, JBuilder y C++Builder.

.NET ofrece IDEs incluidas en el Visual Studio .NET, que por lo pronto soporta a los lenguajes Visual C++, Visual Basic y Visual J.

CORBA y CCM no poseen IDEs propias, mas pueden emplear las interfaces de los diversos lenguajes que soportan a CORBA y a CCM, por

⁴*Interface Development Environment*, es decir, ambiente de desarrollo de interfaces.

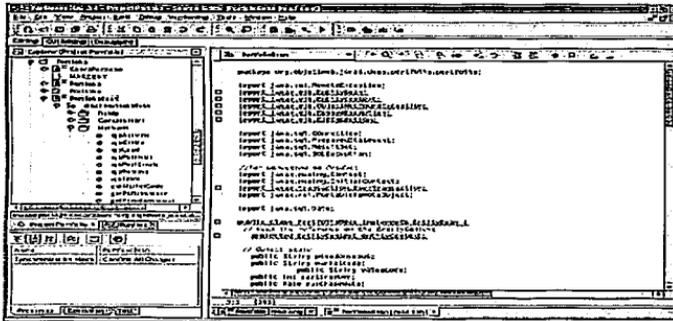


Figura 6.1: La IDE Java NetBeans.

ejemplo, C++-Builder, JBuilder, Delphi, Visual Cafe y muchos más. Lo importante aquí es respetar los mapeos entre los lenguajes y el IDL.

Java es uno de los lenguajes que ofrece más IDEs propietarias y de fuente abierta o libres. Las más destacadas son, por ejemplo, VisualAge, Visual Cafe, JBuilder y NetBeans. Existen una cantidad considerable de IDEs de fuente abierta o libres para Java, pero una de las mejores es sin duda NetBeans (Figura 6.1).

6.2.5. Protocolos de comunicación aceptados

Los componentes de software requieren comunicarse con otros componen-

TESIS CON FALLA DE ORIGEN

Tecnología	Protocolos soportados
COM	ORPC
.NET	ORPC
CORBA	IIOP
CCM	IIOP, RMI
EJB	IIOP, RMI

Cuadro 6.10: Comparación de protocolos aceptados.

tes para llevar a buen término las tareas que se les tienen asignadas, sino no tendría caso hablar de una aplicación distribuida, es por eso que considerar los diferentes protocolos de comunicación que pueden emplearse resulta de mucha valía.

COM y .NET están ligados por su compatibilidad entre ambos, es decir, clientes COM pueden acceder a servidores .NET y clientes .NET puede acceder a servidores COM, por eso ambos soportan el protocolo ORPC⁵ para establecer una comunicación entre componetes.

CORBA emplea el protocolo IIOP para la comunicación entre los componentes vía los ORBs.

CCM también hace uso del IIOP, pero además por estar estrechamente relacionado con los EJB, soporta el protocolo RMI.

EJB utiliza para comunicar componentes remotos tanto a RMI como a IIOP, esto por su relación cercana con CORBA y a que Java es uno de

⁵Object Remote Procedure Call, es decir: llamada a procedimientos remotos de objetos.

los lenguajes que más apoyo y soporte ha dado a CORBA.

6.3. Resultados

Cierto es que aún existen aspectos a tomar en consideración. Sin embargo, en esta investigación se examinan rubros como la diversidad en todos y cada uno de los puntos considerados en el capítulo. A mayor diversidad mayores son las variantes de funcionamiento y costo asociado a la implementación de la tecnología de componentes.

Así por ejemplo, la flexibilidad del precio de la tecnología reduce los costos de la aplicación. Una mejor documentación, capacitación y/o soporte hacen a una tecnología más fácil de usar, modificar y mantener que otras. La robustez y la economía de las plataformas repercuten en el funcionamiento y costo de su implantación. Los IDEs hacen más fácil y menos costoso el desarrollo o modificación de la implementación. Los protocolos de comunicación permiten una mayor escalabilidad en el desarrollo de la aplicación, esto al poder emplear otros componentes que usen el mismo protocolo.

En el presente trabajo de investigación se puede apreciar que los aspectos considerados en este capítulo, nos proporcionan una visión clara para poder elegir la tecnología de componentes distribuidos idónea para el desarrollo del sitio Web del portafolios financiero. Una solución que no perjudique la economía de las empresas, pero que satisfaga las necesidades de proporcionar un entorno de desarrollo de aplicaciones distribuidas

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

90 CAPÍTULO 6. COMPARATIVA ENTRE TECNOLOGÍAS

altamente eficiente, fácil de utilizar y administrar.

Teniendo en cuenta las comparaciones realizadas en el capítulo, Los Enterprise JavaBeans resultan la mejor opción para la creación de una aplicación Web de portafolios financiero basada en componentes de software.

Parte IV

Descripción de la implementación

Capítulo 7

Solución propuesta (Prototipo)

7.1. Introducción

A continuación se presenta la plataforma propuesta para la realización del sitio Web de un portafolios financiero que satisface la problemática planteada. Esta selección se tomó de acuerdo al análisis de las tecnologías presentado en el capítulo anterior donde los EJB resultan el entorno de componentes más viable para la construcción del sitio Web de portafolios financiero por ser la tecnología de creación de aplicaciones distribuidas basadas en componentes en obtener los mejores resultados en cada una de las comparativas y consideraciones realizadas.

Los EJB ofrecen una gran variedad de posibilidades de integración. Poseen el mote de estandarizados (lo que facilita la administración de la aplicación) y están perfectamente controlados por SUN en el área referente a la compatibilidad entre proveedores. Quizá CORBA sea la opción más abierta e independiente de la plataforma, mas Java y los EJB



también lo son en cierta medida y presentan todas las ventajas ya mencionadas en el capítulo 6 que motivaron su elección.

Un ejemplo de las ventajas de los EJB, es que existen interfaces de desarrollo (IDEs) libres como NetBeans que se pueden descargar gratuitamente de la Internet y que no le piden nada en funcionalidad a sus similares propietarios como el Borland JBuilder y el Symantec Visual Cafe. Si no, siempre queda como una alternativa más (pero más difícil), la utilización de un simple editor de texto para escribir las líneas de código. Java y los EJB son sinónimos de flexibilidad.

7.2. Descripción

La plataforma propuesta está conformada por un modelo cliente/servidor a 4 capas. La capa del cliente en una computadora cualquiera con un visualizador de páginas Web es la primera capa. La capa del servidor Web conformada por el servidor de páginas Web, servlets y JSP, TOMCAT, es la segunda capa. La capa del servidor de aplicaciones, JOnAS, donde los Enterprise JavaBeans se encuentran almacenados y de donde son desplegados, es la tercera capa. Y por último la cuarta capa, la capa del sistema administrador de bases de datos (DBMS¹) PostgreSQL. El sistema operativo empleado en todos los servidores es GNU/Linux.

La figura 7.1, mostrada en la página siguiente, describe la plataforma

¹Por sus siglas en inglés de *Data Base Management System*.

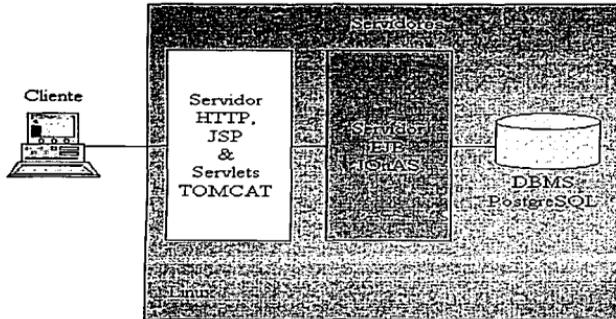


Figura 7.1: Plataforma propuesta.

seleccionada como solución. Del lado del cliente tenemos un navegador de páginas Web que permite acceder a los servidores donde está la implementación (servidor Web, servidor de aplicaciones y sistema administrador de base de datos). Gracias a la flexibilidad de los EJB se pueden intercambiar los elementos aquí presentados, pero para la realización del prototipo se seleccionaron estas herramientas de software por ser libres o abiertas y sobre todo costeables para un estudiante.

7.3. Motivos de la selección

En esta sección se amplían a detalle los motivos de la selección de cada uno de los componentes que conforman la solución propuesta, así como también se presenta una breve descripción de las herramientas utilizadas para la elaboración de la aplicación.

7.3.1. Cliente/servidor de 4 capas

El modelo cliente/servidor a 4 capas es mucho más eficiente que el modelo cliente/servidor tradicional empleado en la mayoría de los sitios Web. Separando la base de datos en una capa, las consultas se realizan con mayor rapidez, el número de tablas puede ser mayor y es más fácil de mantenerla. El servidor de aplicaciones y el servidor Web pueden trabajar en la misma máquina, ya que ambos requieren de la JVM² para su funcionamiento, pero es recomendable separarlos en distintos equipos, ya que esto permite incrementar el desempeño de cada uno de los servidores y decrementar el tiempo de respuesta.

7.3.2. Sistema operativo GNU/Linux

GNU/Linux³ es el sistema operativo escogido debido a la creciente popularidad de éste para implementarlo en los servidores. Linux ofrece niveles de seguridad mayores a Windows siempre y cuando se encuentre perfectamente configurado. Por ser abierto ofrece ventajas de bajo

²Máquina virtual de Java, de sus siglas en inglés de: *Java Virtual Machine*.

³GNU/Linux podría considerarse el nombre correcto de Linux, ya que Linux únicamente es el núcleo del sistema operativo.

costo. Existen distribuciones para casi todos los equipos y algunas de estas distribuciones están especializadas para trabajar como servidores, por ejemplo, RedHat y Debian. Empresas como IBM y HP están invirtiendo en Linux por lo que se considera cada vez más en serio como una alternativa barata a Windows NT/2000 y a Unix.

7.3.3. Base de datos PostgreSQL

La base de datos relacional abierta más robusta es sin duda PostgreSQL. Emplear una base de datos de fuente abierta nos da la libertad de no pagar ni un centavo más por licencias, las cuales muchas veces están limitadas a un número mínimo de conexiones. Y por si esto fuera poco, PostgreSQL ya viene incluida en la mayoría de las distribuciones de Linux.

7.3.4. Servidor de aplicaciones JOnAS

El servidor de aplicaciones JOnAS⁴ es un servidor abierto para crear aplicaciones en Java que sigue el estándar EJB. Cómo existe un número considerablemente grande de usuarios del lenguaje Java y de los Enterprise JavaBeans, consideramos a Java uno de los lenguajes más importantes a considerar en la realización de una aplicación Web. Existen varios servidores de aplicaciones Java, como WebSphere y WebLogic, pero su precio es mayor a los \$5,000 dólares. JOnAS, al ser abierto, puede integrarse a la perfección con GNU/Linux, PostgreSQL y Tomcat.

⁴por sus siglas en inglés de Java Open Application Server, es decir, servidor de aplicaciones abierto para Java.

7.3.5. Servidor de páginas Web Tomcat

Tal vez, el servidor Web con el porcentaje más alto del mercado sea Apache, pero no soporta servlets ni JSP. Aunque la misma *Apache Software Foundation*,⁵ dentro de su proyecto denominado *Jakarta*, creó un servidor Web que soporta servlets y JSP. Su nombre es Tomcat. Tomcat fue el servidor Web escogido porque así se pueden crear páginas que se ejecutan desde el servidor y que emplean la tecnología de Java denominada JSP⁶ que se encuentra incluida en la distribución J2EE⁷ junto con los EJB. Las JSP permiten la creación de páginas Web de contenido dinámico, lo cual resulta sumamente útil si se muestran tablas que puedan cambiar su contenido, como es el caso del portafolios financiero.

7.4. Integración de los elementos

A continuación se presenta cómo se integran los elementos para crear el portafolios financiero a 4 capas. La aplicación contiene muchas líneas de código, por lo que sólo se presentan algunas partes importantes y el funcionamiento general de la aplicación.

7.4.1. Diseño básico

Cuando un portador ingresa al sitio Web de portafolios financiero, lo primero que observa es la página de bienvenida, como se muestra en la

⁵en español: fundación de software Apache.

⁶*Java Server Pages*: páginas del lado del servidor de Java.

⁷*Java 2 Enterprise Edition*, es decir, Java 2 edición empresarial o jugando con el inglés un poco: edición Java para la empresa.

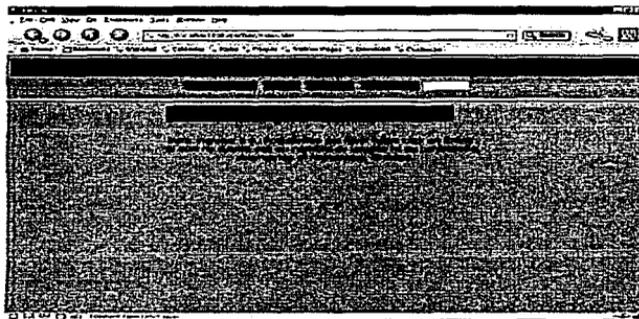


Figura 7.2: Página de inicio del Portafolios Financiero.

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

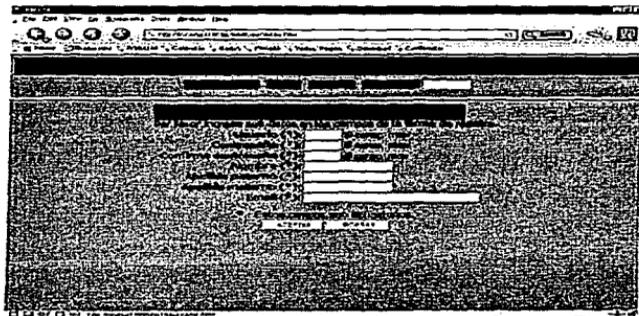


Figura 7.3: Formulario de registro de usuarios.

figura 7.2. En la página de bienvenida se invita al portador a ingresar si éste ya está registrado, en caso de no estarlo se le invita a registrarse para obtener su acceso al servicio.

Una vez que el usuario llena el formulario de registro, como el de la figura 7.3 en la página anterior, y sus datos son almacenados en la tabla correspondiente a los portadores de la base de datos del portafolios financiero, entonces el inversionista recién registrado puede acceder a ver sus portafolios financieros, siempre y cuando el nombre de usuario y la contraseña escritas en la página de ingreso, figura 7.4, sean correctos, de lo contrario aparecerá un mensaje de error.

Entrando en la aplicación es mostrado el contenido de la tabla del portafolios asociado al portador recién ingresado. Dicho contenido es generado dinámicamente utilizando las JavaServer Pages y presentado en la página Web correspondiente. Si el portador no ha creado ningún portafolios, entonces la tabla aparecerá vacía y el usuario tendrá que pasar a la página de creación de un nuevo portafolios si desea crear o agregar una nueva carpeta de inversiones.

Un portafolios posee los atributos que se aprecian en el cuadro 7.2, equivalen a campos en la tabla de la base de datos correspondiente a los portafolios financieros que han sido creados por el inversionista.

El valor de un portafolios está ligado, lógicamente, al valor de los instrumentos de inversión contenidos en éste. Los instrumentos de inversión

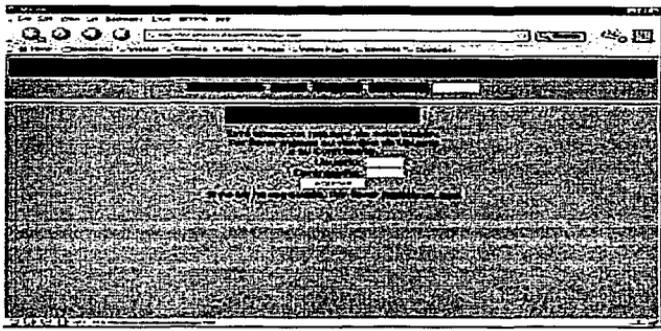


Figura 7.4: Página de ingreso al Portafolios.

Atributo	Descripción
Identificador	Identificador único asociado.
Nombre	Nombre asociado al portafolios.
Valor de compra	valor de compra en pesos.
Valor actual	valor actual en pesos.
Diferencia en \$	valor actual menos valor de compra.
Diferencia en %	porcentaje de ganancia.

Cuadro 7.2: Descripción de la tabla del Portafolios.

Atributo	Descripción
Símbolo	Establece el símbolo del instrumento.
Nombre	Nombre del instrumento en el mercado.
Precio	Precio actual del instrumento.
Variación	Variación respecto al día anterior.

Cuadro 7.4: Descripción de la tabla de Instrumentos.

también tienen atributos correspondientes a los campos de la tabla de los instrumentos contenidos en los portafolios creados por el cliente de la aplicación.

La tabla 7.4 presenta los atributos de los instrumentos de inversión y sus respectivas descripciones. Tanto la tabla del portafolios como la tabla de los instrumentos representan una tabla en la base datos portfolio.

```

portfolio=# CREATE TABLE porter (
portfolio(# login CHAR(8),
portfolio(# password CHAR(8),
portfolio(# name CHAR(20),
portfolio(# firstname CHAR(20),
portfolio(# lastname CHAR(20),
portfolio(# email CHAR(40),
portfolio(# );
CREATE
portfolio=# SELECT * FROM porter;
login | password | name | firstname | lastname | email | adress
-----+-----+-----+-----+-----+-----+-----
(0 rows)

```

Figura 7.5: Tabla de usuarios portadores de valores.

La creación de una tabla es ejemplificado en la figura 7.5. Ahí se repre-

TESIS CON FALLA DE ORIGEN

104 CAPÍTULO 7. SOLUCIÓN PROPUESTA (PROTOTIPO)

senta la tabla contenida en la base de datos creada con PostgreSQL que tiene por nombre: `portfolio`. Dentro de `portfolio` se crea una tabla `porter` que representa a los portadores de valores ya registrados en el servicio Web del portafolios financiero.

Las páginas Web en su mayoría están construidas con HTML simple y empleando hojas de estilo. Esto para reducir el tiempo de respuesta del servidor, pues emplear servlets o JSP requiere de emplear a la JVM para ejecutarlos, lo que incrementa el tiempo de respuesta, por lo tanto los servlets únicamente son usados para el envío de información, mientras que las JSP para crear páginas de contenido dinámico.

La figura 7.6, en la página siguiente, ejemplifica la hoja de estilo empleada en el sitio Web de portafolios financiero. Ésta hoja de estilo permite que todas las páginas del sitio tengan una presentación semejante sin tener que programar las cualidades requeridas en todas y cada una de las páginas del sitio.

Como ya se mencionó antes, para acceder a la aplicación del portafolios financiero, el portador de valores o usuario del sitio Web tiene que ingresar su nombre de usuario o alias y su contraseña. Para verificar que los datos introducidos en la página de ingreso corresponden a los datos del usuario contenidos en la tabla `porter` de la base de datos `portfolio` son los mismos, se realiza una verificación a través del uso de un servlet. La figura 7.7 presenta el fragmento del servlet de ingreso llamado `LoginServlet`.

```
/******  
/* Portfolio Service */  
/* Style Sheet for HTML documents */  
/******  
  
BODY { background-color: #2bc6a2;  
        color: black;  
        font-family: verdana, helvetica, arial, sans-serif  
    }  
  
P { font-family: verdana, helvetica, arial, sans-serif  
    }  
  
UL { font-family: verdana, helvetica, arial, sans-serif;  
      margin-left: 5%  
    }  
  
.title { background-color: #0f896d;  
          border-color: black;  
          border-style: solid;  
          border-width: medium;  
          font-size: 1.2em;  
          margin-right: 10%;  
          padding: .5%;  
          text-align: left  
    }  
  
...
```

Figura 7.6: Fragmento de la hoja de estilo.

7.4.2. Creando los Enterprise JavaBeans

Una tabla de la base de datos se representa como un Entity Bean. Para ejemplificar la construcción de este Bean, usamos al Bean que lleva por nombre portfolio y que está contenido en el paquete portfolio. Los códigos o fragmentos de las interfaces que componen al Entity Bean portfolio se pueden observar en el anexo de los códigos.

Para comenzar en la creación de un Entity Bean, se realiza la interfaz remota (ver Anexo A, algoritmo A.1). En la interfaz remota simplemente se definen los métodos que el Bean utilizará para obtener y escribir datos en la tabla. Por ejemplo getIdentifier, que obtiene el identificador único asociado al portafolios.

Después se crea la interfaz Home (ver Anexo A, algoritmo A.2). En Home se definen las operaciones que se pueden realizar con el Bean, por ejemplo, findAllPortfolios que busca todos los portafolios contenidos en la tabla portfolio de la base de datos portfolio.

Luego se construye el Enterprise Bean (ver Anexo A, algoritmo A.3). En el Enterprise Bean se establece el funcionamiento de todos los métodos o funcionalidades del Entity Bean Portfolio.

Y finalmente se define el BeanPK (ver Anexo A, algoritmo A.4). El

```
package org.objectweb.jonas.login.servlet;

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
...
public void doPost(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();

    // Get the user's login and password
    String login = req.getParameter("login");
    String password = req.getParameter("password");

    // Check the login and password for validity
    Porter porter=checkAndGetPorter(login, password, req);

    if (porter==null) {
        res.sendRedirect(req.getScheme() + "://" + req.getServerName() + ":" +
req.getServerPort() + invalidLoginDoc );
        return;
    } else {
        // Valid login. Make a note in the session object.
        HttpSession session = req.getSession(true);
        session.putValue("login.porter", porter);
    }
    ...
}
```

Figura 7.7: Fragmento de LoginServlet.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

108 CAPÍTULO 7. SOLUCIÓN PROPUESTA (PROTOTIPO)

BeanPK establece la llave primaria de búsqueda.⁸

Todas las interfaces anteriores están contenidas dentro de un mismo paquete, que en este caso es: `portfolio.portfolio`, esto es, se encuentran contenidos en el subdirectorío `portfolio` que a su vez está dentro del subdirectorío `portfolio`. Esto también es definido en el archivo XML que describe al Bean (ver Anexo A, algoritmo A.5).

⁸en inglés *Primary Key* (PK).

Parte V

Conclusión y perspectivas

Capítulo 8

Conclusión

Esta investigación presenta una solución basada en una arquitectura cliente/servidor a 4 niveles o capas y en los componentes distribuidos. La solución responde a la problemática presentada por el uso de redes locales a servidor único, la cual no puede satisfacer la demanda incremental de transacciones.

Esta tesis tuvo el firme propósito de proporcionar una comparativa entre las diferentes opciones de tecnologías de desarrollo de aplicaciones distribuidas basadas en componentes.

En este trabajo de investigación se mostró que el uso de los Enterprise JavaBeans en el desarrollo de un sitio Web de portafolios financiero es una alternativa mucho más viable que sus contrapartes.

En la presente tesis la hipótesis es válida ya que se demostró que los Enterprise JavaBeans permiten desarrollar un sitio Web de un portafolio

TESIS CON FALLA DE ORIGEN

lios financiero eficiente (con un tiempo de respuesta aceptable, una alta disponibilidad, una escalabilidad superior, una mejor robustez, además de ser fácil de usar, seguro, confiable, integral y por sobre todo económico), de fácil desarrollo y administración.

Emplear los Enterprise JavaBeans es mucho más económico; seguro; fácil de desarrollar, de mantener y modificar. Todo gracias a su enorme flexibilidad y a la existencia de varios e importantes proveedores de servidores Web, servidores de aplicaciones y servidores de bases de datos que soportan a Java, que se integran a la perfección entre ellos, y por si esto fuera poco, son de fuente abierta y/o libres.

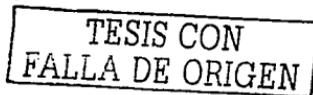
Capítulo 9

Perspectivas y trabajo por realizar

La seguridad es un elemento muy importante del desarrollo de aplicaciones basadas en la Web, pero extenso y complejo también. Aunque Java y los Enterprise JavaBeans proveen de mecanismos de seguridad a las aplicaciones, en la Web incrementar los niveles de seguridad nunca está por demás. Por tal motivo, se cree que un trabajo que podría realizarse en el futuro, sería incrementar los niveles de seguridad en un sitio Web empleando cortafuegos o firewalls, servidores carnada, SSL,¹ etc.

La tendencia tecnológica parece dirigirse al uso de equipos más pequeños y móviles como lo son los organizadores personales o PDA's y los teléfonos celulares para acceder a la Internet. Aunque existen máquinas virtuales de Java para varios de estos dispositivos, se podría extender la plataforma para permitir el acceso empleando estándares de comunicación tal y como lo es el WML.

¹ *Socket Security Layer*, en español, capa de seguridad en sockets.



Parte VI

Anexos

Apéndice A

Códigos

Aquí se presentan algunos códigos o fragmentos de partes importantes de la aplicación del sitio Web de portafolios financiero creado con los Enterprise JavaBeans.

TESIS CON
FALLA DE ORIGEN

118 APÉNDICE A. CÓDIGOS

Algorithm 1 Interfaz remota del Entity Bean Portafolio.

```
package portfolio.portfolio;

import java.rmi.RemoteException;
import javax.ejb.EJBObject;

/*
 * Remote interface for the Portfolio's bean
 */
public interface Portfolio extends EJBObject {
    // ----- Get methods -----
    public String getIdentifier() throws RemoteException;
    public String getName() throws RemoteException;
    public long getBuyValue() throws RemoteException;
    public long getActualValue() throws RemoteException;

    // ----- Set methods -----
    public void setIdentifier(String val_identifier) throws
RemoteException;
    public void setName(String val_name) throws RemoteException;
    public void setBuyValue(String val_buyvalue) throws RemoteException;
    public void setActualValue(int val_actualvalue) throws RemoteException;
}

```

Algorithm 2 Interfaz Home del Bean del Portafolio.

```
package portfolio.portfolio;

import java.rmi.RemoteException;
import java.util.Enumeration;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;
import javax.ejb.FinderException;

/*
 * Home interface for the Portfolio's bean
 */
public interface PortfolioHome extends EJBHome {
    public Portfolio create(String val_identifier, String val_name, long
val_buyvalue, long val_actualvalue) throws RemoteException,
CreateException;
    public Portfolio findByPrimaryKey(PortfolioBeanPK pk) throws
RemoteException, FinderException;
    public Enumeration findByIdentifier(String val_identifier) throws
RemoteException, FinderException;
    public Enumeration findAllPortfolios() throws RemoteException,
FinderException;
}
```

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

120 APÉNDICE A. CÓDIGOS

Algorithm 3 Fragmento del Entity Bean del portafolio.

```
public class PortfolioBean implements EntityBean {
    // Keep the reference on the EntityContext
    protected EntityContext entityContext;

    // Object state
    public String identifier;
    public String name;
    public long buyvalue;
    public long actualvalue;
    private static UserTransaction utx = null;

    /*===== ejbCreate methods =====*/
    public PortfolioBeanPK ejbCreate(String val_identifier, String
    val_name, long val_buyvalue, long val_actualvalue) {
        // Init object state
        identifier = val_identifier;
        name = val_name;
        buyvalue = val_buyvalue;
        actualvalue = val_actualvalue;
        return null;
    }

    public void ejbPostCreate(String val_identifier, String val_name, long
    val_buyvalue, long val_actualvalue) { }

    /*===== javax.ejb.EntityBean implementation =====*/
    public void ejbActivate() { }
    public void ejbLoad() { }
    public void ejbPassivate() { }
    public void ejbRemove() { }
    public void ejbStore() { }
    public void setEntityContext(EntityContext ctx) {
        // Keep the entity context in object
        entityContext = ctx;
    }
}
...

```

Algorithm 4 BeanPK del Portafolio.

```
package org.objectweb.jonas.portfolio.portfolio;
```

```
public class PortfolioBeanPK implements java.io.Serializable {
    public String identifier;
    public PortfolioBeanPK(String identifier) {
        this.identifier = identifier;
    }
    public PortfolioBeanPK() {
    }
    public boolean equals(Object other) {
        boolean isEqual = false;
        if (other instanceof PortfolioBeanPK) {
            isEqual = ( identifier == ((PortfolioBeanPK)other).identifier)
        }
        return(isEqual);
    }
}
```

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

122 APÉNDICE A. CÓDIGOS

Algorithm 5 Fragmento de la descripción del Portafolio.

```
<!DOCTYPE ejb-jar SYSTEM "/usr/local/J0nAS/xml/ejb-jar_1.1.dtd">
<ejb-jar>
  <enterprise-beans>
    <entity>
      <description>Deployment descriptor for the Portfolio bean
      </description>
      <ejb-name>Portfolio</ejb-name>
      <home>portfolio.portfolio.PortfolioHome
      </home>
      <remote>portfolio.portfolio.Portfolio
      </remote>
      <ejb-class>portfolio.portfolio.PortfolioBean
      </ejb-class>
      <persistence-type>Container</persistence-type>
      <prim-key-class>portfolio.portfolio.PortfolioBeanPK
      </prim-key-class>
      <reentrant>False</reentrant>
      <cmp-field>
        <field-name>identifier</field-name>
      </cmp-field>
      <cmp-field>
        <field-name>name</field-name>
      </cmp-field>
      <cmp-field>
        <field-name>buyvalue</field-name>
      </cmp-field>
      <cmp-field>
        <field-name>actualvalue</field-name>
      </cmp-field>
    </entity>
  </enterprise-beans>
  ...

```

Apéndice B

Archivos de configuración

Aquí se presentan algunos archivos de configuración de los distintos servidores de la aplicación del sitio web de portafolios financiero creado con los Enterprise JavaBeans.

TESIS CON FALLA DE ORIGEN

124 APÉNDICE B. ARCHIVOS DE CONFIGURACIÓN

Algorithm 6 Archivo de configuración de la variable de entorno.

```
# .bashrc

# User specific aliases and functions

export JAVA_HOME=/usr/local/j2sdk
export CATALINA_HOME=/usr/local/tomcat
export JONAS_ROOT=/usr/local/JONAS
export PATH=$PATH:$JONAS_ROOT/bin/unix:$JAVA_HOME/bin:$CATALINA_HOME/bin

# Source global definitions

if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```

Apéndice C

Pantallas

Aquí se presentan algunas de las pantallas más importantes de la aplicación del sitio web de portafolios financiero creado con los Enterprise JavaBeans.

TESIS CON FALLA DE C...

126 APÉNDICE C. PANTALLAS

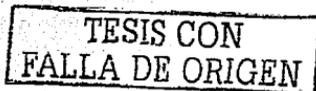
```
dherreaga@RHU3Rminal.org /home/dherreaga - intérprete de comandos - Esviste
Sesión Editar Vista Preferencias Ayuda
Croot@dherreaga@RHU3Rminal:~$ cd /opt/tomcat/bin
Croot@dherreaga@RHU3Rminal:~/bin$ ./startup.sh
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_BASE: /opt/tomcat/conf
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JAVA_HOME: /opt/jdk1.6
Croot@dherreaga@RHU3Rminal:~/bin$
```

Figura C.1: Arranque de Tomcat.



Bibliografía

- [AW99] Asbury Stephen, Weiner Scott R.,
Developing Enterprise Java Applications,
John Wiley & Sons, Inc.,
ISBN 0-47-132756-5, 1999.
- [CD00] Comer Douglas E.,
Internetworking with TCP/IP principles, protocols and archi-
tectures 4th edition,
Prentice Hall, Inc.,
ISBN 0-13-018380-6, 2000.
- [DD98] Deitel Harvey M., Deitel Paul J.,
Java How To Program second edition,
Prentice Hall, Inc.,
ISBN 0-13-899394-7, 1998.
- [EG95] Elton Edwing J., Gruber Martin J.,
Modern Portfolio Theory and Investment analysis 5th edition,
John Wiley & Sons, Inc.,
ISBN 0-471-00743-9, 1995.



TESIS CON FALLA DE ORIGEN

130 BIBLIOGRAFÍA

- [FMF96] Fabozzi Frank J., Mondigliani Franco, Ferri Michael G.,
Mercados e Instituciones Financieras primera edición,
Prentice Hall Hispanoamérica, S.A.,
ISBN 968-880-679-X, 1996.
- [GB94] Gibbs Mark, Brown Todd,
Absolute guide to networking,
Sams Publishing,
ISBN 0-672-30524-0, 1994.
- [JC95] Jamsa Kris, Cope Ken,
Internet programming,
Jamsa Press,
ISBN 1-884133-12-6, 1995.
- [JJ98] Jaworski Jamie,
Java 1.2 Unleashed,
Sams Publishing,
ISBN 1-57521-369-3, 1998.
- [JLW96] Jamsa Kris, Lalani Suleiman "Sam", Weakley Steve,
Web Programming,
Jamsa Press,
ISBN 1-884133-27-4, 1996.
- [MB00] Maslakowski Mark, Butcher Tony,
Teach yourself MySQL in 21 days,
Sams Publishing,
ISBN 0-672-31914-4, 2000.

- [OH98] Orfali Robert, Harvey Dan,
Client/Server Programming with Java and CORBA 2nd edition,
Jonh Wiley & Sons, Inc.,
ISBN 0-471-24578-X, 1998.
- [OHE99] Orfali Robert, Harvey Dan, Edwards Jeri,
The essential Client/Server survival guide 3rd edition,
Jonh Wiley & Sons, Inc.,
ISBN 0-47-131615-6, 1999.
- [PD99] Platt David S.,
Understanding COM +,
Microsoft Press,
ISBN 0-7356-0666-8, 1999.
- [PD01] Platt David S.,
Introducing Microsoft .NET,
Microsoft Press,
ISBN 0-7356-1377-X, 2001.
- [PL98] Pazos Luis,
La Globalización riesgos y ventajas,
Editorial Diana,
ISBN 968-13-3155-9, 1998.
- [PT97] Parnell Terè,
LAN TIMES Guide to building high-speed networks,

TESIS CON
FALLA DE ORIGEN

132 BIBLIOGRAFÍA

- Mc Graw-Hill, Inc.,
ISBN 0-07-8822-9, 1997.
- [PT00] Powell Thomas A.,
Web Design. The Complete Reference,
Mc Graw Hill, Inc.,
ISBN:0-07-212297-8, 2000.
- [SGG01] Sarang P. G., Gabhart Kyle, Griffin John,
Professional EJB,
Wrox Press, Inc.,
ISBN 1861005083, 2001.
- [TA95] Tanenbaum Andrew S.,
Distributed Operating Systems 1st edition,
Prentice Hall, Inc.,
ISBN 0-13-219908-4, 1995.
- [CR01] Comerford Richard, Keeping the Net Up, Revista IEEE Spec-
trum, Volumen 38, No.1, Enero del 2001.
- [FF00] Fluckiger François, Les recherchistes européens et son réseaux,
Revista La Recherche, No. 328, Febrero del 2000.
- [RFC15] C. Stephen Carr, Request For Comments number 15, Univer-
sity of Utah, 25 de septiembre de 1969.
- [SB02] Sánchez César, Bertrand Felipe. Programación Orientada a
Objetos en Ada, Revista Programadores en Acción, No. 4, Fe-
brero del 2002.

- [STB00] St-Arnaud Bill, Turcotte Bernard, Bjerring Andrew, Plaidoyer pour un Internet tout optique, Revista La Recherche, No. 328, Febrero del 2000.
- [TJM01] Torres José Miguel, COM+ en arquitectura DNA, Revista Sólo Programadores, No. 84, Enero del 2001.

TESIS CON
FALLA DE ORIGEN

Índice alfabético

- Acciones, 23
Apache Software Foundation, 98
aplicación frontal (front-end), 51
aplicaciones de red, 50
ARPA, 34-36
ARPAnet, 34-38
- back-end, 51
Baran Paul, 34
base de datos, 96
BBN, 35
Berners-Lee Tim, 43, 44, 48
Bonos, 23
BSD, 37
- código ensamblador, 58
código máquina, 58
CAC 40, 19
características CCM, 74
características COM, 68
características CORBA, 73
características de un componen-
te, 61
características .NET, 69
CCM, 65, 72, 75, 82
Cerf Vint, 36
CERN, 43, 44
Certificados, 23
CIDL, 73
cliente ligero, 52, 53
cliente pesado, 51
cliente-solicitud, 46
cliente/servidor, 49-53
CLR, 69
COM, 65, 66, 80
COM+, 67
compiladores, 58
componente, 61
componentes, 62
computera, 36
CONDUSEF, 23
confiabilidad de la información, 6
CORBA, 65, 71, 75, 81

- CORBA 3.0, 72
- DCOM, 66
- DHTML, 48
- diseño a dos capas, 51
- disponibilidad, 5
- DJIA, 19
- DTD, 74
- eficiencia, 5
- EJB, 65, 76, 78, 82
- Encapsulación, 61
- ensambladores, 58
- Enterprise JavaBeans, 65, 76, 78
- escalabilidad, 5
- Explorer, 42
- facilidad de uso, 6
- FOOTSIE, 19
- FORTRAN, 59
- front-end, 51
- FTP, 38, 39, 44
- FTSE 100, 19
- Galeon, 42, 43
- GNU, 43
- GUI, 51
- guión, 48
- Herencia, 61
- hoja de estilo, 48
- HTML, 45
- HTTP, 46
- IAB, 41
- ICANN, 41
- IDL, 72, 74
- IETF, 41
- IIOF, 72, 74, 88
- IMP, 35
- indicador de mercado, 18
- índice de mercado, 18
- Instanciación, 61
- integridad, 6
- interfaz gráfica del usuario, 51
- Internet, 32, 36-42, 44
- internet, 33
- Internet2, 41
- interpretes, 59
- IP, 37
- IPC, 19
- IPv6, 41
- ISO, 39
- ISOC, 41
- J2EE, 98

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

136 ÍNDICE ALFABÉTICO

- Jakarta, 98
- Java, 59, 74, 76, 77
- JavaBeans, 76
- JDK 1.1, 77
- JIT, 70
- JRE, 77
- JVM, 59

- Kahn Bob, 36

- latencia, 5
- lenguaje de alto nivel, 59
- lenguaje de máquina, 58
- lenguajes de propósito general, 57

- métodos, 60
- mecanismo (back-end), 51
- mercado, 16
- mercado de valores, 16
- mercado financiero, 16
- middleware, 56
- mnemónicos, 58
- modelo cliente/servidor, 53, 76
- modelo cliente/servidor a 3 capas, 96
- modelo cliente/servidor a multi-capas, 54
- modelo de 3 capas, 54

- MONO, 69, 71
- Mosaic, 44
- MSIL, 69
- MTS, 67

- NASDAQ, 19
- navegador, 46
- navegadores, 45
- NCP, 35
- NCSA, 44
- .NET, 65, 67, 71, 81
- Netscape, 44
- Netscape Navigator, 42
- NYSE, 19

- objeto, 60
- objetos, 60
- OLE, 66
- OMG, 71, 73, 76
- OpenCCM, 75
- OpenSource, 81
- ORB, 72, 73
- ORPC, 88
- OSI/RM, 39
- OTS, 72

- paradigmas de un objeto, 60
- plugin, 48

Polimorfismo, 61
 POO, 57, 60
 Portafolios, 27, 28
 programación, 56
 programación de alto nivel, 58
 programación de componentes, 62
 programación orientada a objetos,
 57
 programación procedural, 59
 RAND Corporation, 34
 Real-Time ORB, 75
 recursos, 50
 red, 33
 redes, 49, 50
 RFC, 35
 RMI, 74, 76
 robustez, 5
 script, 48
 seguridad, 6
 servicio web, 62
 servidor de aplicaciones, 56
 servidor de archivo, 50
 servidor de disco, 50
 servidor web, 47
 servidor-respuesta, 46

SGML, 45
 SMTP, 38, 39
 software libre, 81
 solicitud-respuesta, 46
 solución de cajón, 51
 supercomputadora, 50
 Títulos, 23
 tarjetas perforadas, 58
 TCP, 36, 37
 TCP/IP, 37-40, 46
 Telnet, 38, 39
 terminal tonta, 49
 terminal tonta, 50
 tiempo de desarrollo, 5
 Tomcat, 98
 UCLA, 35
 UCSB, 36
 Unix, 37
 URL, 46
 valores financieros, 21
 W3C, 41
 Web, 41-45, 48
 XHTML, 48
 XML, 48, 70, 74, 77

TESIS CON
 FALLA DE ORIGEN