

1 INTRODUCCIÓN

1.1 OBJETIVOS Y MOTIVACIÓN

El objetivo para la realización del proyecto fue la creación de un Sistema Peer to Peer para la gestión de Catálogos UBL. Para este cometido era necesario, la creación de una red Peer to Peer, la gestión de una base de datos, y la utilización de UBL como formato para la representación de los catálogos y posterior comercio electrónico. Además, el proyecto debía de ser una aplicación con un buen entorno gráfico.

Para estas tareas se seleccionó JXTA como red Peer to Peer, UBL como formato para la representación de los catálogos, XQuery como base de datos capaz de trabajar con los ficheros UBL (xml con un espacio de nombres específicos) y Java como lenguaje de programación, utilizando además librerías gráficas.

Por tanto, los **objetivos** del proyecto, se podía dividir en lo siguiente:

- Estudio de JXTA como red Peer to Peer y desarrollo de la red Peer to Peer para la comunicación entre los nodos y el transporte de los catálogos UBL.
- Estudio de UBL y la creación de un Catálogo UBL que se ajustara a las necesidades.
- Estudio de XQuery y creación de una base de datos XQuery que pudiera almacenar y gestionar los catálogos de la red.
- Estudio de las librerías Swing de Java y el posterior desarrollo de un entorno gráfico que pudiera ejecutarse en cualquier máquina y sistema operativo.

La novedad de estas tecnologías propició un nuevo objetivo:

- La presentación de dicho proyecto en el Congreso “ColleECTer Iberoamérica 2008” celebrado en Madrid por la Universidad Politécnica, en Junio del 2008.

Se marcaron estos objetivos ya que reunían una serie de **características** que encajaban con lo que se buscaba:

- JXTA, era independiente de la plataforma y del sistema operativo. Además existía una implementación en Java que cumplía con estas características.
- UBL era un protocolo novedoso para el comercio electrónico, y cada vez más difundido por muchos países.
- XQuery nos permitía trabajar directamente con ficheros XML (UBL es XML con un específico espaciado de nombres) sin necesidad de hacer

ninguna conversión como hubiera ocurrido si se hubiese trabajado con una base de datos de tipo SQL.

- Librerías gráficas Swing, nos permitiría diseñar un entorno agradable para el usuario de manera que tuviera buena aceptación y se pudiera difundir rápidamente.

Para el desarrollo del proyecto se utilizó VMWare, ya que permite simular toda una red dentro de un mismo ordenador. Con VMWare se puede instalar tantas máquinas virtuales como se quiera y dentro de cada máquina virtual se puede instalar el Sistema Operativo deseado. En este proyecto, se utilizaron dos Sistemas Operativos Windows XP que actuaron como clientes de la aplicación y un Sistema Operativo Linux (se utilizó varias versiones hasta llegar a Ubuntu 8.10) que actuó como servidor.

Con todas estas máquinas virtuales se pudo simular toda una red dentro de un mismo ordenador. Sin embargo, para la correcta visualización del proyecto se utilizó un ordenador con una pantalla de 18,4", ya que, con una de 15,4" no era posible la visualización a la vez de todo el entorno.

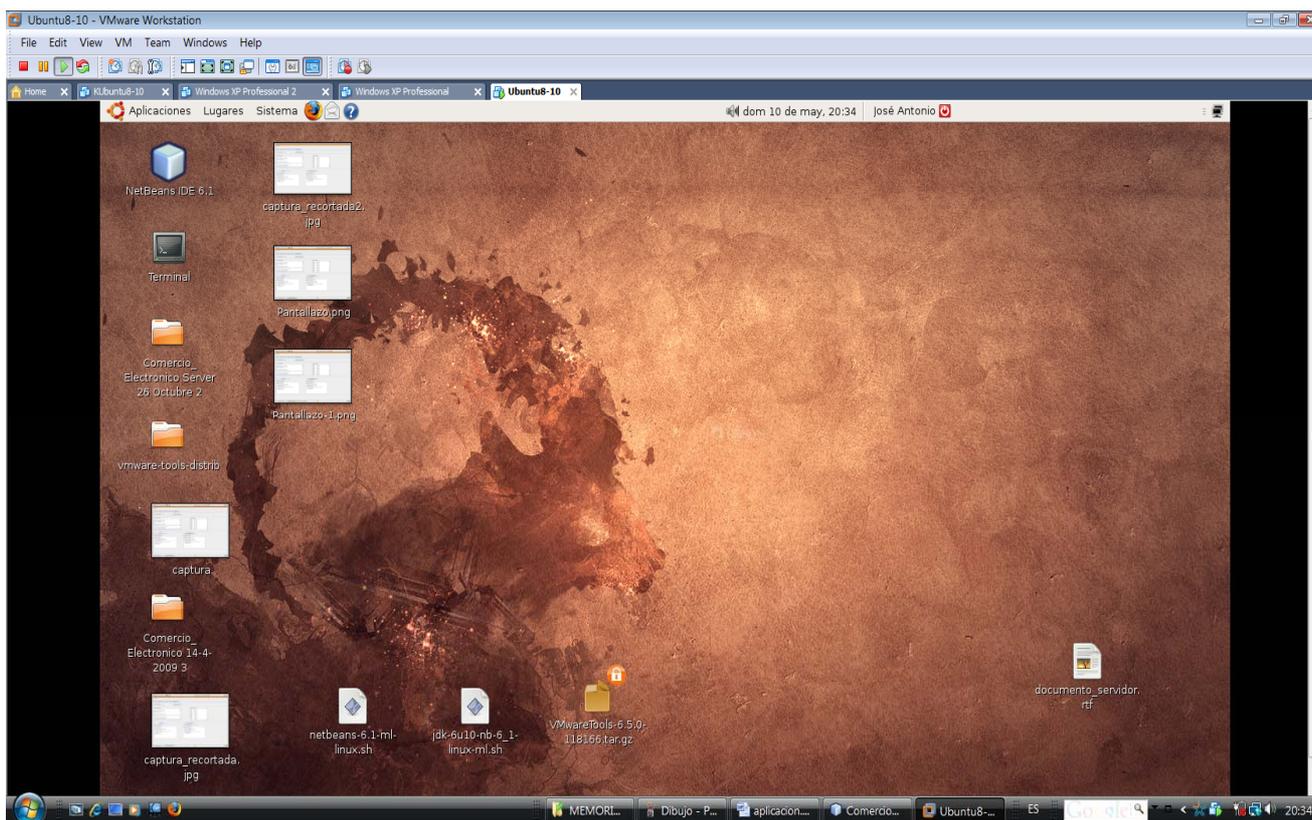


Figura 1-1: Entorno VMWare

Una vez obtenido el entorno óptimo para la simulación de la red y se había hecho el estudio de las distintas tecnologías (incluyendo la puesta en práctica de pequeñas aplicaciones para comprobar el buen funcionamiento de éstas), se desarrolló el entorno gráfico de la aplicación, donde se fueron incorporando las distintas tecnologías:

JXTA, UBL y XQuery. Este entorno gráfico, está constituido por cinco pestañas y a su vez la pestaña de “Catalog” está dividida en otras tres:

- Catalog
 - New Catalog
 - View Catalog
 - Search into catalog
- Connection Client
- Transaction
- Options
- Connection Server

Las pestañas de “Catalog/New Catalog” y “Catalog/View Catalog” se utilizan para la creación y vista/modificación de los distintos catálogos UBL.

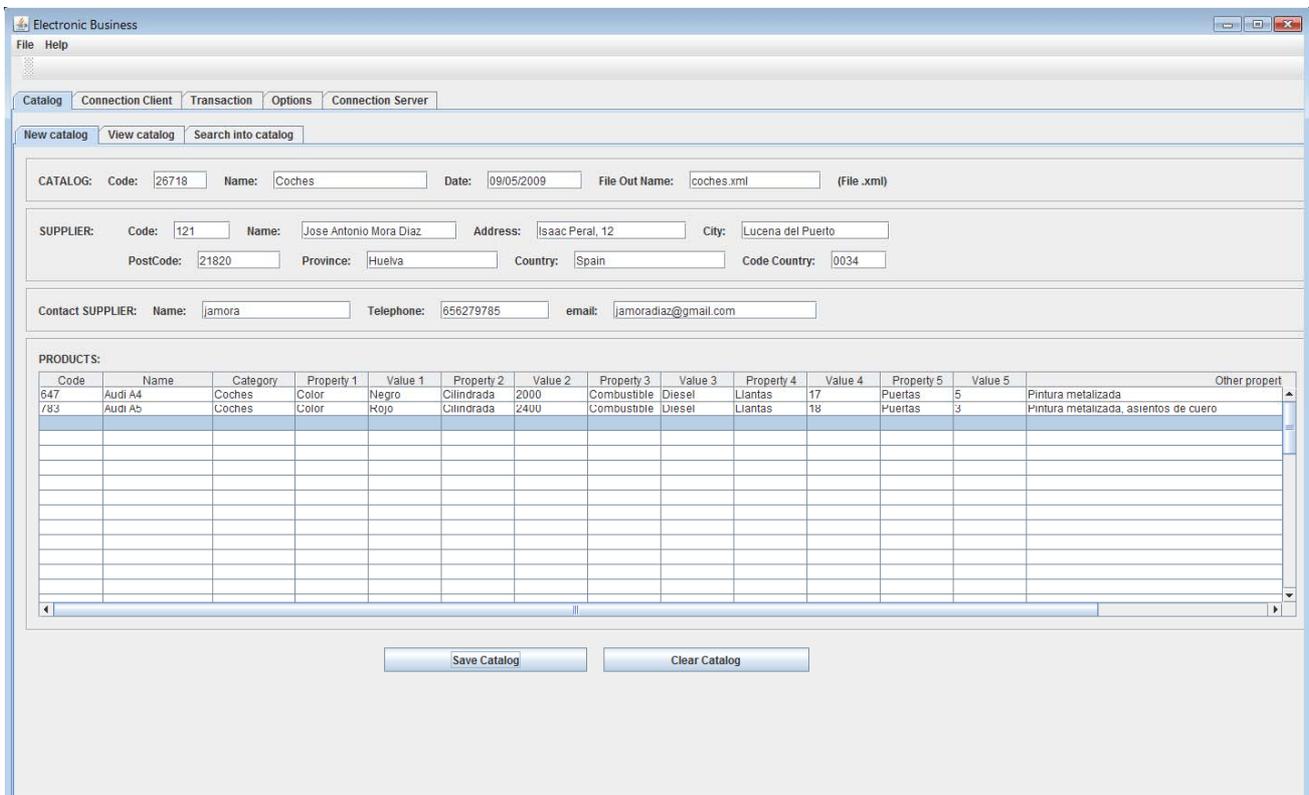


Figura 1-2: Entorno gráfico de la aplicación

Las pestañas de “Connection Client” y “Connection Server” se encargan de la comunicación JXTA: inicia tal comunicación y muestra el estado de la red (nodos conectados y mensajes enviados). La aplicación puede funcionar como cliente y como servidor. En el caso del cliente, la pestaña de “Connection Client”, permite añadir y eliminar catálogos a una lista que serán enviados al servidor para su registro o eliminación del registro de la base de datos. La pestaña de “Connection Server” permite la puesta en marcha de los servidores: servidor de registro (permite añadir catálogos a la

base de datos XQuery) y servidor de búsquedas (permite la aceptación de peticiones de búsquedas en la base de datos y el envío de los resultados de la búsqueda).

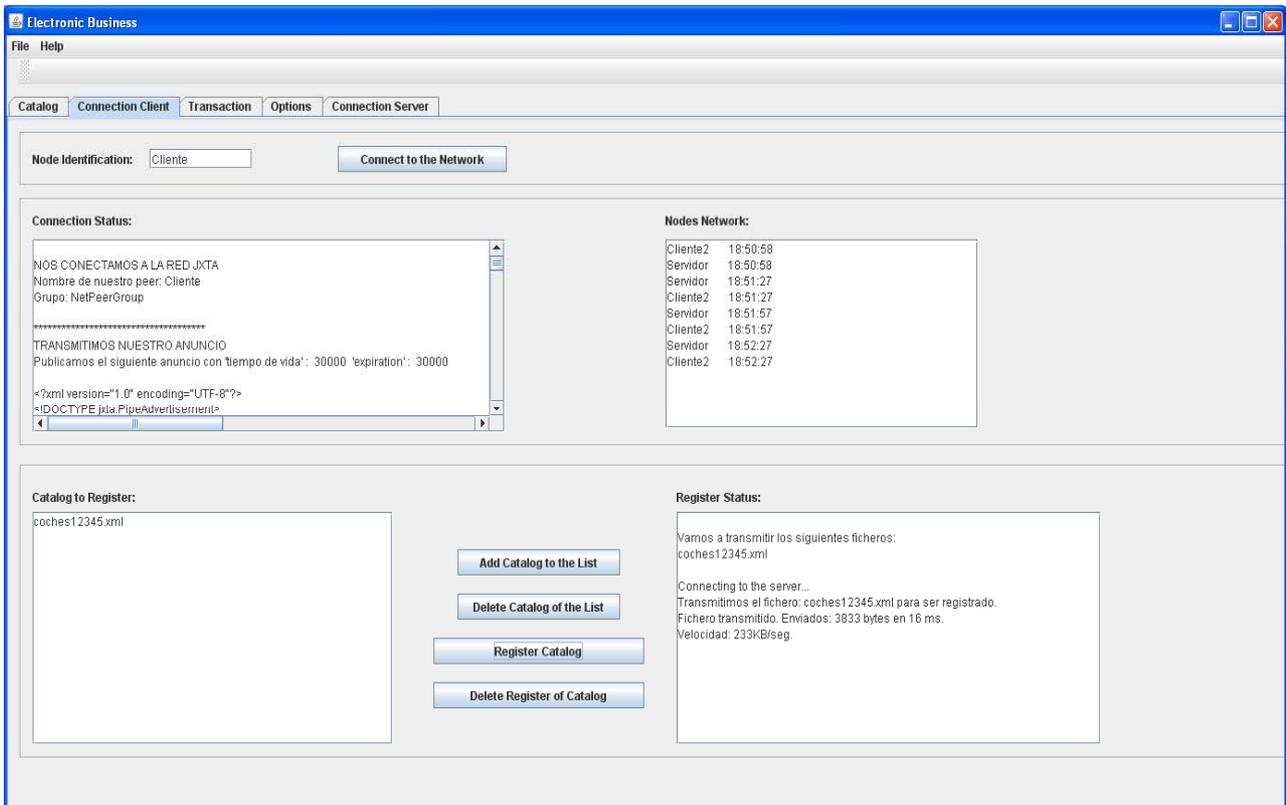


Figura 1-3: Conexión JXTA del Cliente

La pestaña de “Catalog/Search into catalog” permite la búsqueda de catálogos en la base de datos XQuery, ya sea de manera local, como de manera remota. En la aplicación, cada cliente tiene su base de datos local y además existe un servidor donde los distintos usuarios pueden registrar sus catálogos.

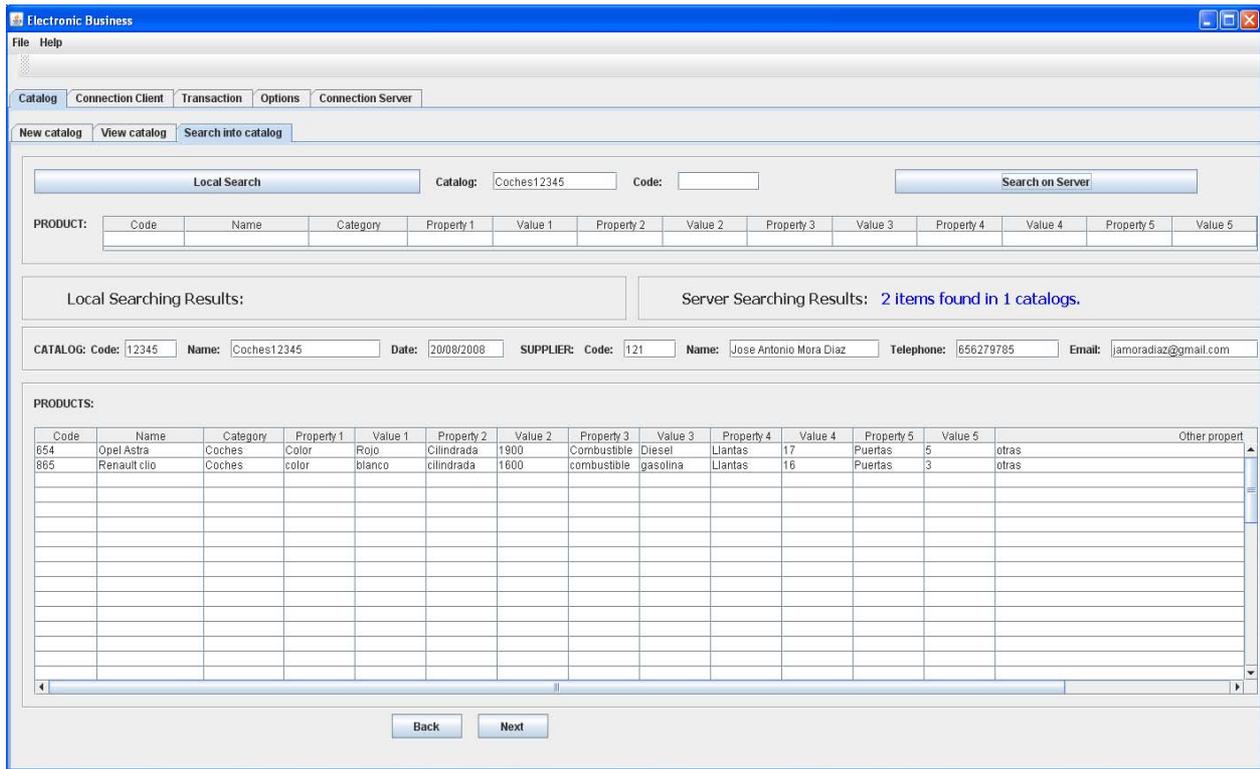


Figura 1-4: Búsqueda de catálogos

Para la base de datos **XQuery** se utilizó la implementación **Qizx** que es de libre distribución. Esta tiene un entorno gráfico para uso y gestión de la base de datos, pero en la aplicación se utiliza sólo el API de dicha base de datos. Esta implementación Qizx (local y remota) realizará las búsquedas a través de ficheros .xql recibidos, los cuales incorporan lenguaje XQuery, que la base de datos interpreta.

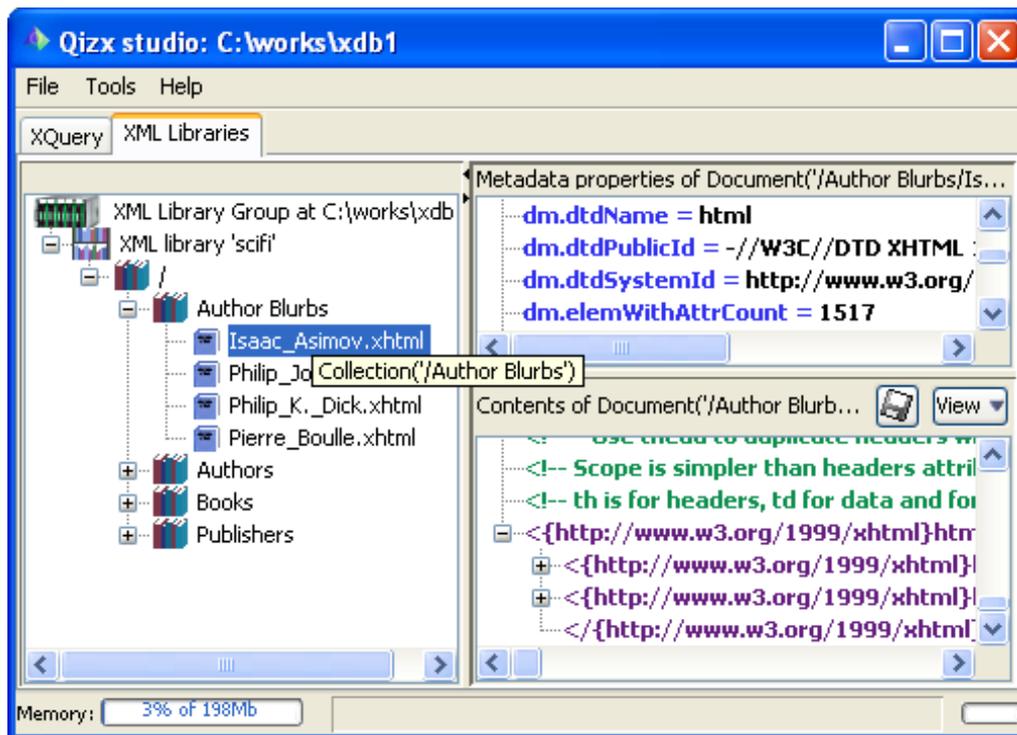


Figura 1-5: Entorno gráfico Qizx

Para la presentación de los catálogos **UBL** en la aplicación se ha utilizado el Catálogo de UBL 2.0, sin embargo, para que se pudieran representar en el entorno gráfico desarrollado, se limitó el catálogo genérico UBL a unos campos. Este Catálogo limitado se le llamó plantilla UBL (Esta plantilla como el programa utilizado para generarla se describe en el capítulo 8.6.1: Programas utilizados-UBL Swinger).

Como complemento a la realización del proyecto, se realizó su presentación en el en el **Congreso “ColleECTer Iberoamérica 2008”** celebrado en Madrid por la Universidad Politécnica, en Junio del 2008. Primero, se presentó un primer informe y una vez aceptado el proyecto se realizó un póster para su presentación en el congreso. Una pequeña imagen del póster puede verse en la Figura 1.6.

A Peer-To-Peer system to perform e-procurement transactions with UBL

Josef Ignacio Hótz Ordoño, Teresa Ariza Gómez, Jose Antonio Mora Díaz, Miguel Rodríguez Iglesias
 University of Seville, Engineering School, Telematics Area
 joseignaciohotorodono@hotmail.com, matere@itaj.us.es

Abstract

Typical e-procurement applications rely on centralized architectures such as e-marketplaces or standalone. In both cases, business information is stored and managed by private companies, implying a permanent presence of third parties in every step. A P2P network offers a different approach to a B2B system, because it deploys cooperative networks where participants can interact directly, with no need of external third parties. Searching for products, a regular task in all e-procurement systems, can also be easily executed and improved in a P2P environment. An e-catalog management system on top of a P2P overlay network is introduced, using UBL as business language. Supplier peers create and manage UBL e-catalogs which are stored in dedicated server-peers. Customer-peers perform complex searches for products with XQuery, sending queries to the e-catalog servers. Once the right supplier is found, the customer can initiate e-commerce transactions like purchase orders with the supplier. Security concerns are also discussed, using a customized protocol to implement digital encryption and verification.

Key words: P2P, UBL, e-commerce

System behaviour

Basic message exchange:

Figure 1: Basic messaging interaction within P2P network in the e-catalog system.

The message choreography follows this sequence:

1. Each peer ("A", "B", and "C") joins the P2P network. "A" identifies itself as an available catalog server.
2. "B" as e-catalog client wants to register a catalog with a product named "Acme", so it asks the P2P network which catalog servers are connected and available.
3. The P2P network replies with a JXTA message containing the desired list. "A" is included in the returned list.
4. "B" as client sends to "A" the UBL Catalog through a JXTA message. "A" stores it locally.
5. "C" wants to purchase a lot of 100 "Acme" units. Firstly it searches for e-catalog servers, as seen on 2. and 3. "A" is included in the returned list.
6. "C" makes a XQuery with filtering parameters and sends it to the catalog servers through a JXTA message.
7. Each catalog server performs the product search within their stored e-catalogs according to the query. "A" replies to "C" with the results of the Xquery, which includes "B" as the seller of "Acme".
8. "C" knows now that "B" has available the product "Acme", so it sends an UBL purchase Order to "B" wrapped in a JXTA message.
9. "B" as e-commerce server receives and processes the Purchase Order. If OK, it sends a Order Response reply to "C".

3. Development

To implement this research we have developed an easy program for inexperienced persons. We can create and edit catalogs and look for products in these catalogs.

Some images from this program are shown:

Catalog creation

JXTA connection

Transaction

1. Introduction

P2P + B2B = PB2B

A P2P network is a distributed computing model that enables decentralized collaboration by integrating devices (known as peers) into subnetworks which each peer can offer and consume services.

The Peer-to-Peer Model

B2B (Business-To-Business).

PB2B: enables companies to share information and execute business transactions through direct connections instead of centralized hubs.

JXTA (P2P) & UBL (B2B) & XQuery

JXTA is an open source peer-to-peer protocol specification designed by Sun Microsystems, which allows a wide range of devices - PCs, palmfrans, cell phones, PDAs.

UBL (Universal Business Language) is designed to provide a universally understood and recognized commercial syntax for legally binding business documents. It is a specific XML.

Xquery (Query language) is designed to query collections of XML data (UBL). It is semantically similar to SQL.

2. Research

The architecture consists of a group of e-catalog servers which enable decentralized product searches within the P2P network. Once a product is found, e-commerce transactions are executed in a decentralized way too.

The system are divided into three layers:

P2P layer: it manages all the P2P communication requirements. JXTA framework has been used to implement it.

E-catalog layer: It manages all the information about products, and it enables searches within e-catalogs and over the P2P network. UBL is the standard format to describe the catalogs and Xquery is used to perform complex searches.

E-commerce layer: It processes e-commerce information flow such as purchase orders over a P2P network. UBL has been selected as e-commerce protocol.

There are four possible roles:

E-catalog server: a peer that stores e-catalog files in UBL. It receives XQuery requests from clients to search for products inside the UBL catalogs, sending back the results.

E-catalog client: a peer who owns a catalog stored in a server and performs remote operations on it, such as updating, uploading or deleting.

E-commerce server: a peer who publishes products in e-catalogs and accepts UBL transactions.

E-commerce client: a peer that searches for products in the e-catalogs held by the servers, and performs UBL transactions (typically purchase orders) with the e-commerce server.

A single peer is able to assume two or more roles at the same time.

Security aspects

To assure a secure e-commerce system we have implemented the following customized protocol using JXTA Security Libraries:

4. Conclusion

The described system forsakes the centralized approach of traditional e-procurement systems, exploring decentralized collaboration between trading partners. Instead of a single centralized e-catalog server, our system (PB2B) allows multiple and independent servers to join and exit the network dynamically. There is a reasonable degree of centralization, because of the searching optimization: queries are sent only to e-catalog servers to avoid flooding the network with unnecessary messages. Our implementation lies on widespread technologies which increase compatibility and reduce adoption costs, such as UBL, XQuery and JXTA.

COLLECTER 2008
Iberoamérica
Colección de Comercio
Intelectual y de Comercio
Intelectual y de Comercio
http://www.iberocollecter.org/

Figura 1-6: Póster Congreso "COLLECTer Iberoamérica 2008"

El carácter novedoso de todos estos objetivos, ya que se utilizaban tecnologías muy actuales, fue la principal motivación en el desarrollo del proyecto.

La planificación del proyecto, así como la explicación de los protocolos utilizados (JXTA, UBL, XQuery) y el desarrollo de la aplicación, son tratados con más detalles en los siguientes capítulos.

1.2 PLANIFICACIÓN TEMPORAL

El tiempo empleado para el desarrollo de este proyecto ha sido de aproximadamente 19 meses, tiempo transcurrido desde principios del mes de Septiembre de 2007 a finales del mes de Junio de 2009, ya que se excluyen los meses de Julio y Agosto y la mayor parte del mes de Diciembre de 2008, que fueron de vacaciones. Además, hay que aclarar que las horas empleadas por día ha sido de aproximadamente 4 horas, por lo que realmente, el tiempo efectivo en realizar el proyecto ha sido de 9 meses aproximadamente.

Dentro del proyecto se ha diferenciado en color verde las actividades relacionadas con la exposición del proyecto en un Congreso, organizado por la Universidad de Madrid: Collecter Iberoamérica 2008.

Memoria

1-7

Podemos ver la distribución temporal del mismo en el siguiente diagrama de Gantt:

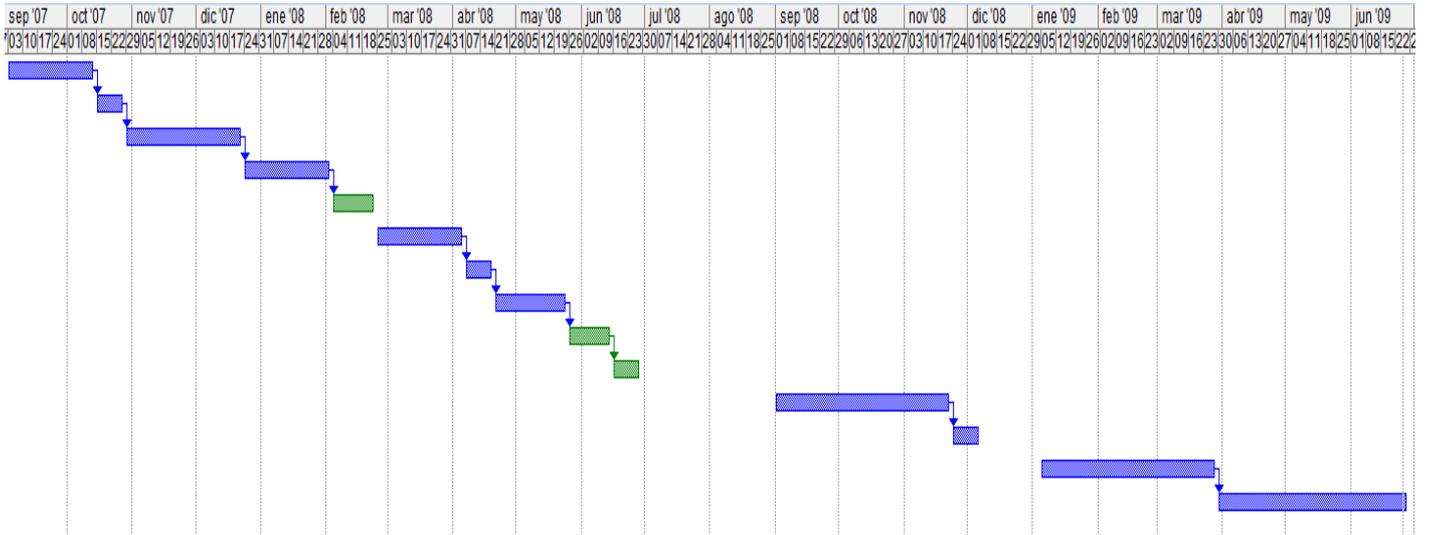


Figura 1-7: Planificación temporal del proyecto desarrollado

Cada una de estas barras se corresponde con las tareas que se muestran en el siguiente cuadro (Figura 1-8), en el cual se especifica la duración y el objetivo de cada una de estas tareas (Igualmente se distinguen las actividades relacionadas con el Congreso, que van con texto azul):

	Nombre de tarea	Duración	Comienzo	Fin
1	Búsqueda y estudio de la información referente a JXTA	30 días	lun 03/09/07	vie 12/10/07
2	Elección del entorno de trabajo: VMWare y Netbeans	10 días	lun 15/10/07	vie 26/10/07
3	Primeras pruebas de programación JXTA	40 días	lun 29/10/07	vie 21/12/07
4	Estudio y preparación de UBL	30 días	lun 24/12/07	vie 01/02/08
5	Preparación documentación para el Congreso	15 días	lun 04/02/08	vie 22/02/08
6	Estudio programación gráfica con Netbeans	30 días	lun 25/02/08	vie 04/04/08
7	Estudio de Xquery	10 días	lun 07/04/08	vie 18/04/08
8	Desarrollo del entorno gráfico	25 días	lun 21/04/08	vie 23/05/08
9	Preparación del Poster del Congreso	15 días	lun 26/05/08	vie 13/06/08
10	Preparación y presentación en el Congreso	10 días	lun 16/06/08	vie 27/06/08
11	Desarrollo de la red JXTA	60 días	lun 01/09/08	vie 21/11/08
12	Desarrollo de la plantilla UBL	10 días	lun 24/11/08	vie 05/12/08
13	Desarrollo de la base datos Xquery y búsquedas en ella	60 días	lun 05/01/09	vie 27/03/09
14	Desarrollo de la documentación	65 días	lun 30/03/09	vie 26/06/09

Figura 1-8: Distribución del tiempo empleado para cada tarea

Cada día que es mencionado en la tabla son realmente 4 horas, por lo que los días efectivos son justamente la mitad.

1.3 ORGANIZACIÓN DE LA DOCUMENTACIÓN

Esta memoria, relativa a la elaboración del proyecto fin de carrera, consta de los siguientes apartados:

Introducción: Motivación, Objetivos, Planificación y Organización de la documentación

Es este capítulo y se declara el porqué de este proyecto: motivación, objetivos, planificación y la organización de la documentación.

Peer to Peer

En este capítulo se define lo que es una red Peer to Peer, sus características y sus campos de aplicación (aplicaciones). Se hará también una comparación con las redes cliente-servidor.

JXTA

En este capítulo se define JXTA, cuales fueron sus objetivos y su historia. Posteriormente, se verá JXTA con detalle respecto a:

- Arquitectura de JXTA
- Terminología de JXTA
- Aspectos de red
- Protocolos JXTA
- JXTA en Java

XML

En este apartado se trata XML: Lenguaje Extensible de Marcas. Se define en que consiste XML y sus objetivos. Posteriormente se verán sus aspectos básicos:

- Tipos de documentos
 - Bien formados
 - Válidos
- Metalenguajes para definir XML
 - Definición de tipos de documento (DTD)
 - Esquemas (Schema XML)
- Procesado de XML (con y sin validación).

Los últimos puntos de XML tratarán su **sintaxis, las distintas tecnologías XML y su relación con Java.**

XQuery

En el capítulo 5 se trata XQuery: se empieza analizando la necesidad de XQuery y a continuación su definición: se ve requerimientos técnicos y un ejemplo de colección de datos.

Posteriormente, se explica cómo funcionan las consultas en XQuery:

- Reglas generales.
- Diferencias entre las cláusulas for y let.
- Funciones de entrada
- Expresiones condicionales
- Cuantificadores existenciales
- Operadores y funciones principales
- Comentarios

Qizx

Qizx se trata con detalle en el capítulo 6.

En el primer punto, se introduce Qizx y en los siguientes apartados se trata Qizx Studio y Qizx API.

En el apartado de Qizx Studio se explica como:

- Crear una biblioteca usando Qizx Studio
- Llenar una biblioteca de colecciones y documentos
- Peticiones Xquery sobre una biblioteca
- Copiar, Renombrar y Borrar Colecciones y Documentos.

Igualmente para Qizx API se muestra como:

- Crear una biblioteca (Library)
- Crear una colección e importar documentos en ella.
- Peticiones Xquery sobre la biblioteca.

Comercio Electrónico y UBL

En el capítulo 7 se trata el Comercio Electrónico y UBL.

En el primer punto, se empieza el **Comercio Electrónico**. Se realiza una introducción del comercio electrónico y se muestra sus ventajas. Posteriormente, se analiza junto a redes P2P y se comenta una pequeña descripción de los distintos estándares de comercio electrónico:

- EDI
- EDI-XML
- BizTalk

- RosettaNet
- xCBL
- ebXML
- cXML
- UBL

En el punto siguiente, se explica con detalle **UBL**. Se empieza con su definición, origen, estado y penetración de UBL en el mercado internacional.

A continuación, se analiza UBL 2.0:

- Contexto de uso de UBL
 - Requisitos generales de negocio
 - Procesos de negocios
 - Sourcing (Contratación)
 - Ordering (Petición de compra)
 - Fulfilment (Ejecución de la petición/pedido)
 - Billing (Facturación)
 - Payment (Pago)
 - Initiate Transport Services (Servicios de Transporte)
 - Certification of Origin of Goods (Certificado de Origen)
 - Report Status of Goods (Informe del Estado de los artículos)
 - Tipos de documentos
- Esquemas de UBL 2.0
- Esquemas de Catalogue
- Esquemas comunes de UBL

Aplicación

En este capítulo se explica la aplicación desarrollada en el proyecto. Se divide en varias partes:

Aplicación desarrollada

Se describe la aplicación desarrollada compuesta por 12 ficheros java. Se explica la funcionalidad y las Clases que constituyen cada uno de los ficheros siguientes, así como los métodos que componen cada una de las clases:

- Comercio_Electronico.java
- Catalogo.java
- Gestion_Catalogos.java
- Catalogo_Busqueda.java
- Gestion_Busqueda.java
- ConexionJxta.java
- Lista_Files_To_Register.java
- SocketClient.java
- SocketClient_Busqueda.java

- SocketServer.java
- SocketServer_Busqueda.java
- XQuery.java

Manual de Usuario

En este apartado se explica el entorno de usuario de nuestro programa: las distintas ventanas, opciones y botones que podemos encontrar y qué podemos hacer en cada uno de los casos.

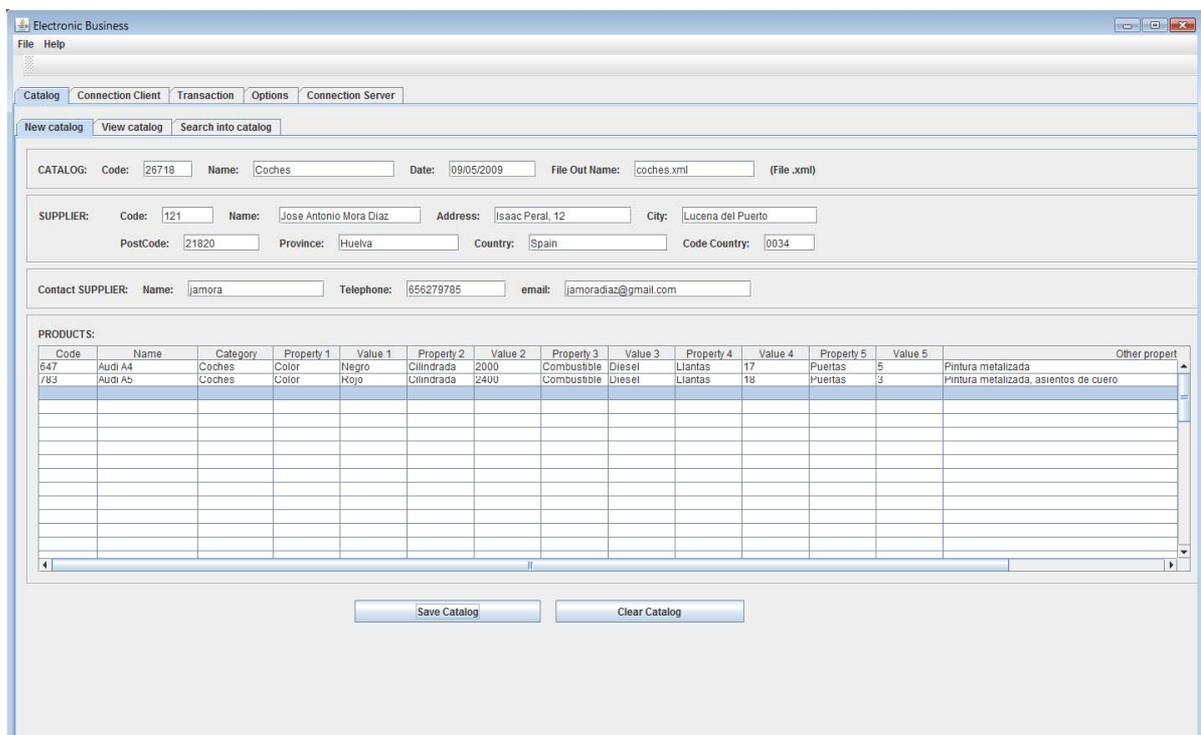


Figura 1-9: Entorno de Usuario

Diagramas de Casos de Uso

Se describen las actividades que se pueden hacer con nuestro programa:

Actividades locales:

- crear catálogo
- abrir/modificar catálogo
- buscar producto local.

Actividades remotas:

(Aquí se engloban las actividades que requiere de la participación de varios equipos)

- Conectar a la red JXTA.
- Iniciar Servidor de Registro (Servidor)
- Iniciar Servidor de Búsquedas (Servidor)
- Añadir/eliminar catálogos a la lista (requeridas por otras actividades remotas).
- Buscar productos en el servidor
- Registrar catálogos
- Eliminar catálogos del registro.

Diagrama de Clases

Se representa el diagrama de Clases de la aplicación.

Diagramas de Secuencia

En este apartado se verá los diagramas de secuencia de algunas de las actividades presentadas en el diagrama de casos.

- Creación de Catálogo
- Búsqueda Local
- Conexión JXTA
- Servidor de Registro
- Busca productos en el servidor
- Registrar productos

Programas utilizados

En este apartado se hace una descripción de los programas utilizados que son los siguientes:

- UBL Swinger: Se utiliza para crear plantilla UBL
- VMWare : entorno que nos simula una red de ordenadores
- Ubuntu 8.10: Sistema Operativo utilizado como servidor
- Netbeans IDE: entorno de desarrollo
- Visual Paradigm: SDE for Netbeans: Para crear diagramas UML

Manual de Instalación

Se describe como se distribuye la aplicación desarrollada y como puede ser instalada. También se describe como se instalan los programas utilizados.

Conclusiones

Se muestran las valoraciones finales de todo lo realizado y las líneas futuras de desarrollo.

Página intencionadamente dejada en blanco.