Módulo StoreFront de Eguana

Erika Aizaga Barzola[1], Federico Domínguez Bonini[2], Wendy Ramos Dávila[3], Luís Muñoz[4]

[1]Ingeniera en Computación 2005; email: eaizaga@hotmail.com
[2]Ingeniero en Computación 2005; email: fexadom@gmail.com
[3]Ingeniera en Computación 2005; email: wramos96@hotmail.com
[4]Director de Tópico, Ingeniero en Computación, Escuela Superior Politécnica del Litoral, 2000; email: juniormunoz@yahoo.es

RESUMEN

Eguana es un proyecto inspirado en el concepto de E-Procurement. Entre muchas cosas, E-Procurement busca una convergencia entre proveedores y compradores en una empresa. Eguana implementa esta convergencia usando J2EE, demostrando el poder y alcance de esta tecnología. StoreFront es el módulo de Eguana encargado de manejar la interacción directa entre compradores y vendedores. En este módulo se usa el espectro completo de la tecnología J2EE desarrollado con diversas herramientas de código abierto, sirviendo así como una prueba de concepto de esta tecnología.

Eguana es hasta el momento el único proyecto E-Procurement código abierto de propósito general disponible. Eguana implementa el concepto E-Procurement de una manera económica, lo cuál permite su uso en negocios pequeños y medianos.

Eguana is a project inspired in the E-Procurement concept. Among many things, E-Procurement seeks the convergence of buyers and suppliers in a company. Eguana implements this convergence using J2EE while demonstrating the power and reach of this technology. StoreFront is the module in Eguana responsible for handling the direct interaction between buyers and providers. StoreFront uses the complete spectrum of the J2EE technology and was developed with a diverse number of open source tools, thus serving as a proof of concept of this technology.

Eguana is so far the only open source, generic E-Procurement project available. Eguana economically implements the E-Procurement concept, making it accessible to medium and small businesses.

INTRODUCCION

E-Procurement, en términos sencillos "solicitud o gestión electrónica", se define como la automatización de los procesos de solicitudes de una organización utilizando aplicaciones basadas en el Web.

Así como los sistemas ERP[1] desarrollan tareas para automatizar sus procesos internos, E-procurement reúne a compradores y proveedores geográficamente dispersos para interactuar y ejecutar transacciones de compra sobre el Internet.

En un sistema E-procurement basado en el Web, cada paso del proceso de solicitud ocurre electrónicamente; por tanto, estas aplicaciones bajan los costos de procesos e inventarios, extienden el alcance de los proveedores y mejoran el acceso de los mismos por parte de los clientes.

La plataforma Java 2 Enterprise Edition (J2EE) es una especificación para el diseño, desarrollo, ensamblaje y despliegue de aplicaciones empresariales y aplicaciones basadas en el Web. El objetivo principal de J2EE es crear un simple modelo estándar de desarrollo para aplicaciones empresariales utilizando componentes basados en el modelo de aplicación. Por tanto, la plataforma J2EE consiste de un conjunto de servicios, interfaces de programación de aplicaciones (APIs[2]) y protocolos que proveen la funcionalidad para desarrollar aplicaciones Web multicapas. J2EE y Java son una creación de la compañía Sun Microsystems.

Eguana es un sistema de información comercial orientado a la administración y soporte del proceso de compra y venta entre empresas en el Web.

Basado fundamentalmente en el concepto de los sistemas E-Procurement e implementado con tecnología J2EE, Eguana brinda a compradores y vendedores un sitio de encuentro en Internet donde pueden realizar solicitudes de productos de manera electrónica, tanto a través de un proceso de compra simple como de una subasta o licitación, donde el sistema proporcionará los principios propios de estos dos tipos de actividades comerciales de una manera ágil y eficiente.

El sistema Eguana para cumplir con las expectativas del mercado competitivo actual y proveer todas las herramientas necesarias para un rápido y completo comercio electrónico, posee cuatro módulos que separan las diferentes funciones de un portal de compras:

- ✓ Administración
- ✓ StoreFront
- ✓ Subasta y Licitación
- ✓ Reportes Gerenciales

CONTENIDO

Arquitectura de Eguana

Los sistemas que gestionan procesos de compras en Internet, se ajustan a un tipo de aplicación común: aplicaciones de comercio electrónico basados en Web. Por lo tanto, Eguana está implementada en una arquitectura empresarial estándar para este tipo de aplicaciones, es

[1] ERP: Entreprise Resource Planning
[2] API: De sus siglas en inglés Application Programmable Interface

decir, una arquitectura multicapas con tecnología EJB[3] que es una propuesta de una arquitectura que propone separar la lógica de presentación de la lógica de negocios basándose en componentes configurables y reutilizables, esto se explicará al detalle en la sección de implementación del módulo StoreFront.

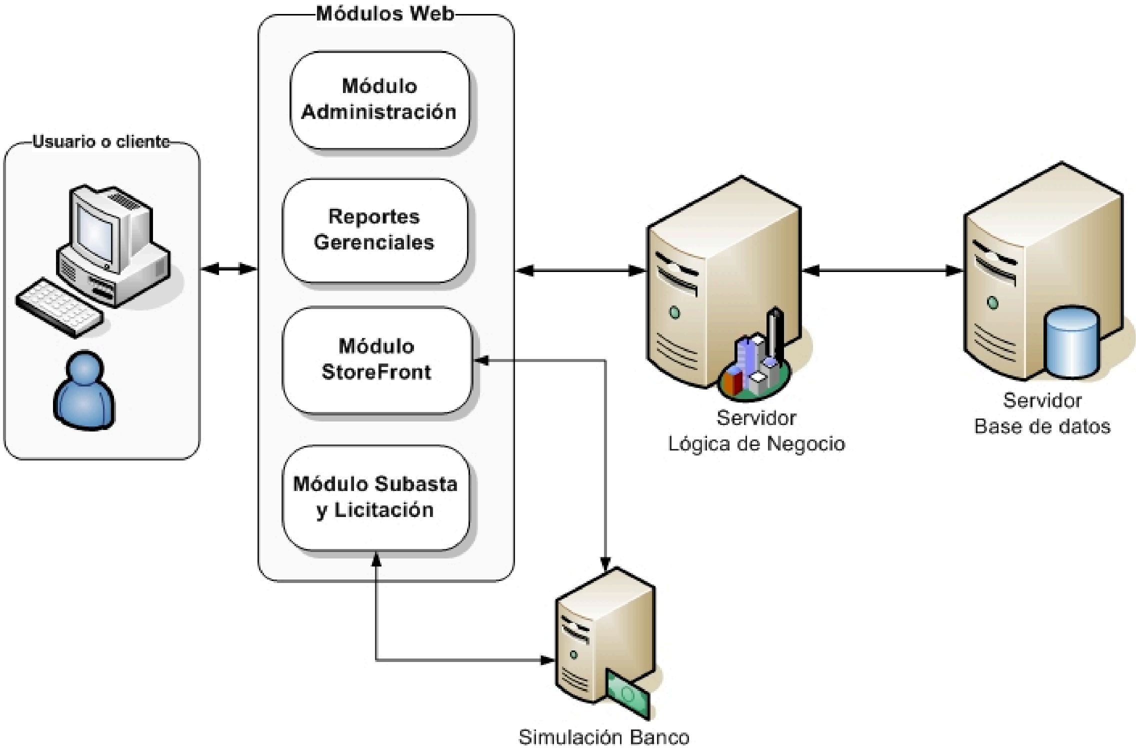

Figura 1: Esquema de Arquitectura Eguana

La aplicación tendrá instalado un servidor de acceso Web, que sirva páginas Web con componentes Java, que son los estándares del mercado en este tipo de operaciones. En cualquier caso, es imprescindible que el medio de acceso sea un navegador de páginas HTML[4], tales como Internet Explorer o Netscape.

Los módulos Web que conforman el sistema (Módulo de Administración, Módulo StoreFront, Módulo Subasta y Licitación, Reportes Gerenciales), han sido diseñados para operar cada uno independiente del otro, pero con la capacidad de integrarse, para cubrir todo el ciclo de vida de la actividad comercial en un portal de compras.

Para realizar transacciones o consultas en cada uno de los módulos, estos deberán interactuar con un servidor que contiene la lógica de negocio, y un servidor de base de datos.

El servidor de lógica de negocio será el que contenga las reglas definidas para la ejecución de cada transacción y ejecute las transacciones. El servidor de base de datos será donde residen los datos de la empresa, deberá contar con una base de datos MySQL[5], para ejecución de las transacciones. Los entornos de aplicaciones más modernos para estos ambientes son J2EE y .Net. Ambos servidores deberán ser de disponibilidad 100%.

---

[3] EJB: **Enterprise Java Beans**. Es un conjunto de componentes Java, desarrollados por Sun MicroSystems, que permiten desarrollar aplicaciones distribuidas y multicapas.
[4] HTML**: Hypertext Markup Language**. Lenguaje para escribir documentos en el web.
[5] MySql: Base de datos código abierto.

Los módulos StoreFront, Licitación y Subasta interactúan con la simulación de un sistema E-payment (módulo bancario) definido como la realización electrónica del pago en la cuenta del banco del proveedor.

<u>Módulo StoreFront</u>

El módulo "StoreFront" es la sección del sistema que se encarga de implementar la funcionalidad de "tienda", de ahí su nombre (Store = tienda, front = frente). La funcionalidad de tienda es básicamente:

1. Ingresar a la tienda (ingreso al sistema, login)
2. Buscar en las perchas el o los producto que deseo comprar (navegación en el catalogo)
3. Separar los productos que deseo comprar en una canasta o carrito de compras. (Carro de compras virtual).
4. Llevar los productos en el carrito a la cajera. (Orden de compra virtual).
5. Pagar los productos. (Transacción de pago).
6. La cajera me entrega el o los productos cancelados. (Orden de entrega virtual).

Además de implementar estos pasos análogos a una tienda o supermercado físico debemos implementar funcionalidad exclusiva de una tienda virtual Web, la cuál es:

7. Automatización de comunicación entre comprador y vendedor vía email.
8. Comunicación al vendedor del estado de sus ventas vía email y mensajes del sistema.
9. Comunicación al comprador del estado de sus compras vía email y mensajes del sistema.
10. Comunicación al comprador y al vendedor de eventos de la transacción. Ejemplo: anulación de orden, negación de crédito, etc.

<u>Implementación del Módulo StoreFront</u>

Al módulo StoreFront de Eguana, durante el análisis, se lo puede dividir en secciones de funcionalidad. A continuación se listan las respectivas secciones:

- ✓ Ingreso y salida del sistema
- ✓ Catálogo
- ✓ Carrito de compra
- ✓ Proceso de compra
- ✓ Proceso de despacho
- ✓ Proceso de cancelación

Para la implementación usamos las siguientes plataformas y herramientas código abierto:

- ✓ Sistema operativo: Linux
- ✓ Base de datos: MySQL 3.23
- ✓ Servidor de aplicaciones: JBOSS 3.2.4 (J2EE 1.3)
- ✓ Herramienta de desarrollo: Eclipse 3.0

A continuación se presenta una breve descripción de la implementación de cada una de estas secciones:

Ingreso y salida del sistema.- Esta sección se encarga de validar el usuario y clave de los usuarios de Eguana. Además se encarga de mantener la seguridad y confidencialidad de la información de autenticación. Aquí se definen que secciones de la aplicación serán protegidas con acceso restringido.

A continuación se presentará la página de inicio del sistema, donde se encuentra el formulario de inicio de sesión en el cual el usuario ingresa sus datos de usuario y contraseña. El usuario debe haber sido registrado previamente por su empresa en el sistema y, dependiendo de su tarea se le habrá asignado un usuario, clave y roles.

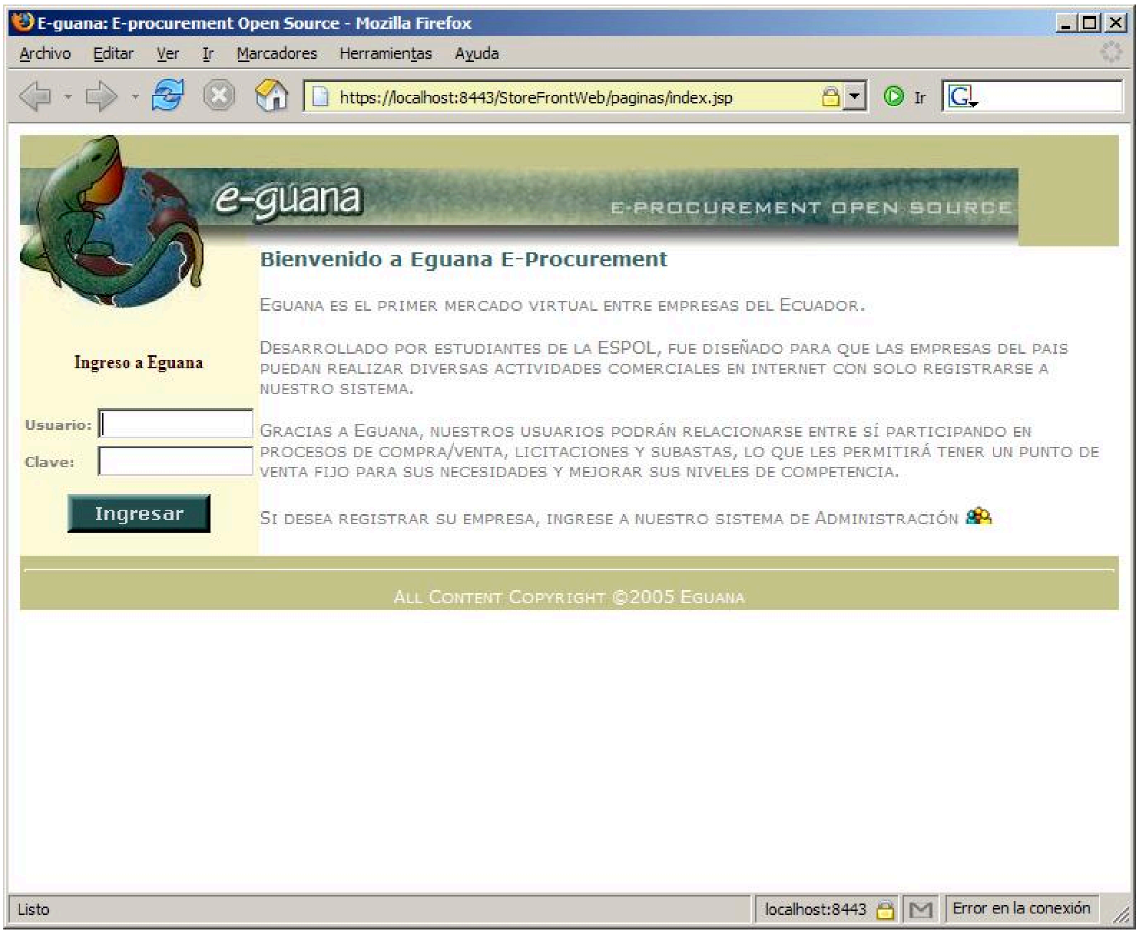

Figura 1: Página de Inicio de Eguana

Una vez que el proceso de autenticación ha validado al usuario y tiene permiso para ingresar al sistema, se verifican qué roles tiene asignado (autorización), los cuales permitirán observar diferentes opciones del menú principal, de acuerdo a la siguiente tabla:

Tabla I: Roles vs. Opción de menú

Rol	Opción de Menú
Comprador	Compras Ver mis Órdenes
Vendedor	Órdenes Asignadas

Se usaron además las siguientes tecnologías en esta sección:

1. HTTPS y SSL: Protocolo seguro de comunicación de datos en el Web.

2. JAAS: Java Authentication and Authorization Service, es un conjunto de APIs usado para autenticación y autorización.

Catálogo.- En el catálogo se implementa:

- ✓ Búsqueda de productos de acuerdo a un criterio definido por el usuario.
- ✓ Búsqueda de empresas vendedoras de productos de acuerdo a un criterio definido por el usuario.
- ✓ Búsqueda de categorías de productos de acuerdo a un criterio definido por el usuario.
- ✓ Vista de productos por empresa.
- ✓ Vista de productos por categoría.
- ✓ Vista de detalles del producto.
- ✓ Completa navegación dentro del catalogo entre categorías, empresas y productos.
- ✓ Facilitar la comunicación con las empresas vendedoras desde el catálogo.
- ✓ Fácil y rápida vista del precio del producto.
- ✓ Fácil compra.

Se usa directamente JDBC y JSP para manejar la búsqueda de información directamente en la base de datos:

Figura 2: Sección de Compras con el Catálogo

Carrito de compras.- El carrito de compras es el segundo paso en el proceso de compras luego del catálogo. El carrito debe ser lo más transparente e inocuo para el usuario, tiene que ser extremadamente fácil de usar y flexible. Su código programático debe ser lo suficientemente modular para ser reutilizado por otros módulos de Eguana.

La funcionalidad que el carrito de compra en Eguana presenta es:

- ✓ Añadir productos de diferentes empresas en el catálogo a un carro de compras virtual que solo existe en la memoria del servidor de aplicaciones.
- ✓ Cambiar la cantidad de ítems y borrarlos del carrito de compras.
- ✓ Mostrar un total de la compra al ver el carrito de compras y durante la navegación en el catálogo.
- ✓ Validar con el inventario el número de ítems de una empresa que se desea pedir.

Se usaron las siguientes tecnologías:

1. JavaBeans
2. EJB: Enterprise Java Beans

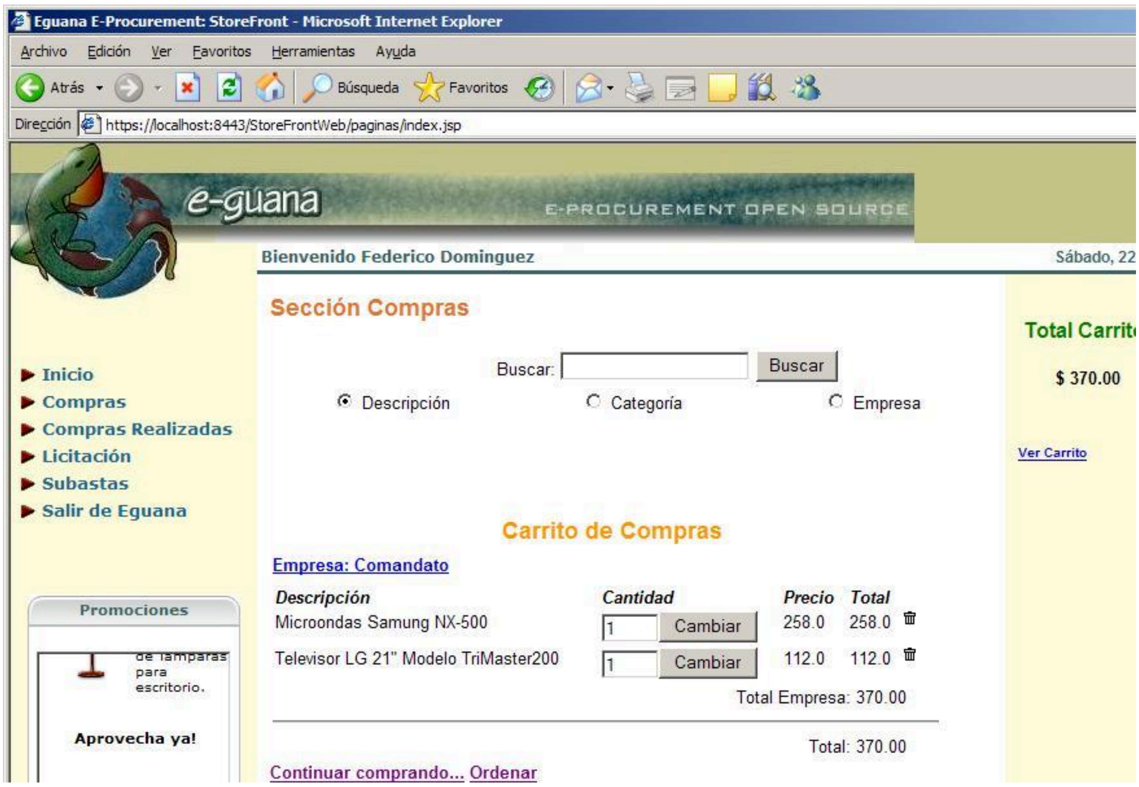

Figura 3: Carrito de Compras

Proceso de compra, cancelación y despacho.- El proceso de Compras en el sistema considera los siguientes puntos en su funcionalidad:

- ✓ Ingresar en el módulo StoreFront.
- ✓ Escoger la opción compras, la cual le presentará un catálogo de productos registrados en el sistema.
- ✓ Escoger del catálogo de productos los que desea adquirir, los cuales se encuentran categorizados por descripción, categoría y empresa.
- ✓ Añadir al carro de compras los productos que desee adquirir.
- ✓ Confirmar los productos seleccionados y que se encuentran en el carro de compras del sistema.
- ✓ Realizar la orden de compra de los productos que han sido seleccionados por el usuario que ha realizado la compra en el sistema EGUANA.
- ✓ Crear una orden de compra por cada una de las empresas en las que se estén seleccionando los productos.
- ✓ Enviar un mensaje a la aplicación bancaria para realizar los respectivos débitos correspondientes a los valores de las órdenes realizadas.
- ✓ Asignar un vendedor a cada orden de compra.
- ✓ Emitir notificación de e-mail a cada usuario vendedor.

Además los usuarios comprador y vendedor pueden cancelar las ordenes hasta el momento en que son despachadas por el vendedor. Cuando la orden es despachada o cancelada se envían mensajes de confirmación a comprador y vendedor.

Figura 4: Consulta de Órdenes

Se usaron las siguientes tecnologías:

1. JSP y Sofia (Framework)
2. EJB: Enterprise Java Beans
    a. MDB: Message Drive Beans
    b. CMP EJB: Container Manager Persistance EJB
3. JNDI: Java Naming and Directory Interface
4. JavaMail
5. JMS: Java Messaging System

CONCLUSIONES

Una vez culminado el proyecto se pueden emitir las siguientes conclusiones:

- ✓ Los modelos de arquitectura definen fundamentalmente las características principales de un sistema y determinan la manera de codificarlo. Por ello, es sumamente importante seleccionar aquel modelo que se ajuste mejor al sistema a desarrollar. En particular, el Modelo de Arquitectura de Software de n Capas, contempla la independencia entra las diversas capas, lo cual permite que se hagan cambios en cualquiera de estas sin afectar a las demás. Este modelo fomenta la modularidad y la reusabilidad de los elementos del sistema.

- ✓ La aplicación puede ser dividida entre varios servidores físicos, distribuyendo diferentes capas en distintas ubicaciones.

- ✓ El uso de una arquitectura J2EE permite que la solución sea escalable de acuerdo a las necesidades que se vayan planteando con la evolución y crecimiento del negocio. Adicionalmente esta misma arquitectura elegida posibilita que la aplicación sea portable e implementada sobre diferentes plataformas. Dada la importancia de permitir la expansión de la funcionalidad del sistema, la solución utiliza en todos sus módulos estándares que limitan el impacto de los cambios en el caso que sea necesario actualizar o crear nuevos componentes, reduciendo consecuentemente los costos y tiempos de desarrollo y mantenimiento.

- ✓ UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores. Al implementar un lenguaje de modelamiento común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

- ✓ Eguana y específicamente el módulo StoreFront logró demostrar el uso del espectro completo de las tecnologías basadas en J2EE y de las herramientas y plataformas de código abierto. No solo se demostró la utilidad de la tecnología J2EE y del movimiento “Open Source”, también se puso en evidencia su poder y alcance en la industria de desarrollo de software para empresas.

Lamentablemente durante el desarrollo de este proyecto también se puso en evidencia las limitaciones de estas tecnologías y el porque aún no han logrado vencer completamente a su competencia.

- ✓ La especificación J2EE 1.3 y 1.4, en especial la especificación EJB 2.x, es aún muy difícil de codificar y entender, incluso con la ayuda de herramientas como XDoclet y Sofia. La curva de aprendizaje de esta tecnología y su tiempo de desarrollo es la barrera más grande que existe para su total aceptación en el mercado. Tecnologías rivales como .Net de Microsoft sostienen todavía una ventaja en este aspecto.

- ✓ El soporte al desarrollador y al usuario para las tecnologías J2EE y herramientas código abierto todavía necesitan madurar. En algunos casos los sistemas de ayuda y soporte eran inexistentes o la información se encontraba desorganizada e inexacta, incluso cuando se pagaba por el acceso a la misma.

Afortunadamente la comunidad de Java ha tomado grandes pasos para remediar estos problemas. Eguana fue desarrollado con el estándar J2EE 1.3, el cuál aplica la especificación EJB 2.0. La última versión en producción de JBOSS, la 4.0, implementa el estándar J2EE 1.4 con EJB 2.1. Actualmente existe en estado de revisión pública el estándar J2EE 5[6] con EJB 3.0.

EJB 3.0 simplifica de una manera dramática la implementación de los EJBs. En esta nueva especificación se elimina, entre muchas cosas, el uso de las interfaces remotas, locales, y objetos "Home" y la necesidad del complicado descriptor de despliegue. Los EJBs de entidad son más parecidos a los POJOs (Plain Old Java Objects) y a los Java Beans. A pesar de no estar finalizada, esta especificación es implementada por JBOSS 4.0 de manera experimental. Con la ayuda de Hibernate, EJB 3.0 y J2EE 5 se posicionan fuertemente en la industria de desarrollo de software empresarial, ofreciendo un producto robusto, completo y simple a la comunidad de desarrolladores.

REFERENCIAS

1. E. Aizaga, F. Domínguez, W. Ramos, "Módulo StoreFront de Eguana" (Tópico, Facultad de Ingeniería en Electricidad y Computación, Escuela Superior Politécnica del Litoral, 2005)

[6] J2EE 5 es la versión siguiente a J2EE 1.4. La comunidad de Java, por simplificación, decidió no usar el número 1.5 en las últimas versiones de J2EE y Java SDK.

Guayaquil, lunes 24 de Octubre del 2005

Dr.
Paúl Carrión
Director del CICYT
Ciudad.

De nuestras consideraciones:

Por medio de la presente, hacemos la entrega formal del artículo para la Revista Tecnológica del CICYT, referente a nuestra tesis de grado, cuyo título es: "Módulo StoreFront de Eguana". Por esto, solicitamos nos conceda el certificado que corrobore esta entrega.

Agradecemos la atención a la presente.

Atentamente,

Ing. Luis Muñoz	Erika Aizaga B.
Director de Tópico	Mat # 199512807
Federico Domínguez B.	Wendy Ramos D.
Mat # 199710344	Mat # 199609611