

MÓDULO DE ADMINISTRACIÓN DE E-GUANA

Paola Acosta Carvajal¹, Maria de Lourdes Barreno Valdivieso², Andrés Guerrero Vasco³, Luis Muñoz⁴

¹ Ingeniero en Computación 2006

² Ingeniero en Computación 2006

³ Ingeniero en Computación 2006

⁴ Director de Tópico. Ingeniero en Computación, Escuela Superior Politécnica del Litoral

RESUMEN

El módulo de Administración de E-Guana¹ es uno de los módulos constituyentes del sistema E-Guana. Este es un sistema de E-Procurement, es decir, es un sistema que automatiza los procesos de adquisición de una empresa usando aplicaciones Web.

Este módulo provee una parte importante de los servicios que debe proveer todo sistema de E-Procurement, entre éstos se tiene: el mantenimiento del catálogo de productos, el mantenimiento de usuarios, el control de acceso y seguridad a través de roles de usuario. Un servicio adicional provisto por este proyecto es el control del pago de las transacciones resultantes de las compras generadas por los módulos de E-Guana: Store Front y Licitación & Subastas.

El módulo de Administración de E-Guana es por si solo un sistema que demuestra el uso de la tecnología J2EE utilizando el paradigma MVC (Modelo Vista Controlador). El modelo lo constituyen Enterprise JavaBeans, la vista está formada por Java Server Pages y etiquetas del framework Sofia en tanto que el control lo realizan Servlets y clases Java provistas por Sofia.

ABSTRACT

E-Guana's Administration module is one conforming module of the E-Guana system. This is an E-Procurement system, in other words, is a system that automates an organization's procurement processes using Web applications.

This module provides an important part of the services that must be provided by any E-Procurement system, which includes products catalog maintenance, user maintenance, access control and security trough user roles.

An additional service provided by this project is payment management, resulting from buyer's orders generated by Store Front and Auctions & Reverse Auctions modules.

E-Guana's Administration module is by itself a system that shows the use of J2EE technology using the MVC (Model View Controller) paradigm. Model is handled by Enterprise JavaBeans, view are Java Server Pages and Sofia framework tags meanwhile control is made by Servlets and Sofia Java classes

INTRODUCCION

El E-Procurement (electronic procurement) define la automatización de los procesos de adquisición de bienes y servicios de una organización, usando aplicaciones basadas en el Web². Esta automatización resulta en varias ventajas para ambos participantes del proceso de compra. Entre éstas se pueden indicar: la reducción de los tiempos requeridos para la transacción, la estandarización de catálogos y reglas de negocio, mayor acceso a proveedores, operabilidad global, reducción de costos e incremento de productividad.

Por lo anotado un sistema de E-Procurement indudablemente trae beneficios nada despreciables para una organización, y es por esto que existen en el mercado varias soluciones de este tipo, aunque, entre otras cosas, tienen en común ser software propietario. Es ahí donde E-Guana ve la oportunidad de entrar al mercado como la solución E-Procurement de Código Abierto. Para alcanzar esto se utilizaron herramientas, de la misma forma, de Código Abierto y basadas en la tecnología J2EE de Java.

El presente artículo se centra en la arquitectura del sistema y la forma de implementación, más específicamente en el módulo de Administración del mismo.

CONTENIDO

1 Descripción de los módulos de E-Guana

E-Guana está dividido en cinco módulos: Administración, StoreFront, Licitación & Subastas, Reportes y Pagos de Transacciones.

Para facilitar la comprensión de la tarea que cumple cada uno de estos módulos dentro del contexto de E-Procurement, la figura 1 presenta la relación entre cada uno de ellos con los componentes que debe tener todo sistema de este tipo.

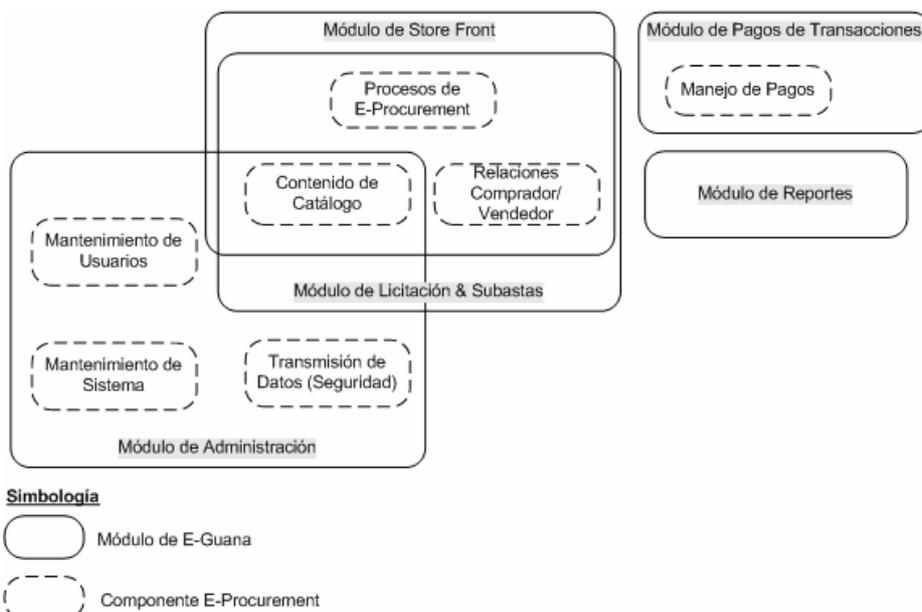


Figura 1 Relación de módulos de E-Guana con los componentes de un sistema de E-Procurement

A continuación se describe la función que realiza cada módulo de E-Guana.

Módulo de Administración: provee las herramientas para el mantenimiento de las principales entidades que soportan al sistema. Entre dichas entidades están el catálogo de productos, así como las empresas y sus usuarios.

Así mismo provee el mecanismo de seguridad que protege las funciones y datos del sistema. La seguridad está basada en la asignación de roles a los usuarios.

Módulo de Store Front: es el módulo de E-Guana encargado de manejar la interacción directa entre compradores y vendedores. Este módulo se encarga de implementar la funcionalidad de “tienda”, de ahí su nombre (Store = tienda, front = frente). Entre las principales funciones están:

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

El sistema mantiene en todo momento notificado a las empresas del estado de las órdenes, ya sea del lado del comprador o del vendedor, en el módulo de StoreFront, a través de notificaciones propias del sistema como por vía email. El proceso de compras llega a término cuando se ha realizado con éxito el proceso de pago y se da autorización al despacho.

Módulo de Licitación & Subastas: es módulo garantiza la realización de todos los servicios necesarios para la gestión de licitación y subasta.

La funcionalidad de *licitación* permite a las empresas utilizar Internet para solicitar y comparar precios de diversos proveedores. Las ofertas se pueden publicar en los tabloncillos de anuncios para fomentar la participación de una gama más amplia de posibles proveedores (publicación de una licitación). Con este módulo, se podrá realizar un aprovisionamiento global. Si necesita materiales específicos rápidamente, los postores entran en el sistema de E-Guana para leer los detalles de la llamada a licitación y para introducir sus ofertas, luego el sistema realiza la elección de la licitación ganadora y se emite la orden de compra.

Al finalizar el proceso de licitación y de la emisión de la orden de compra, se da por terminado el ciclo de compra-venta del bien, sin antes haber efectuado el pago.

El servicio de *subastas* permite a la empresa vendedora ofrecer sus productos o artículos y permitir que varios compradores hagan sus propuestas. La elección de la subasta ganadora será en base a la mejor propuesta ofrecida por el comprador, es decir, la de mayor precio.

Módulo de Pago de Transacciones: es el encargado del manejo de los débitos bancarios a las diferentes empresas, verificando los saldos en línea y asegurando así que

la venta sea exitosa. Por cada orden de compra generada se efectúa el pago electrónico. Este módulo sabe como comunicarse con la entidad bancaria que corresponda a las empresas que participan en el proceso de compra/venta.

Módulo de Reportes: este módulo permitirá generar los reportes de E-Guana y puede generar tres tipos de reportes:

- *Reportes de nivel operacional:* sirven para monitorear las actividades elementales y las transacciones de una organización.
- *Reportes de nivel administrativo:* apoyan las actividades administrativas, de monitoreo, de control y de toma de decisiones realizadas por los mandos medios.
- *Reportes de nivel estratégico:* promueven el cambio de los objetivos, procesos, productos, servicios y relaciones con el entorno para ayudar a la organización a obtener ventajas competitivas.

2 Diseño e Implementación

La tecnología sobre la que se implementó el módulo de Administración de E-Guana (así como los demás) fue Java 2 Enterprise Edition ó J2EE. Dicha tecnología surge de la creciente necesidad de los desarrolladores de escribir aplicaciones transaccionales distribuidas para la empresa, que provean la rapidez, seguridad y fiabilidad de la tecnología del lado servidor. La plataforma J2EE ofrece un modelo de aplicaciones multicapa distribuido, componentes reutilizables, un modelo unificado de seguridad, un control flexible de transacciones y soporte de *Web Services*.

Lógicamente J2EE utiliza Java como su lenguaje de programación, por lo que extiende la ventaja de independencia de plataforma al entorno de aplicaciones empresariales.

J2EE como plataforma de implementación permite gran flexibilidad en la forma desarrollar la solución, por lo que el siguiente paso es decidir de qué forma se la utilizará para lograr los objetivos de una manera rápida, confiable y mantenible. Es ahí donde surge la necesidad de escoger un patrón de diseño adecuado, siendo la respuesta MVC.

El patrón MVC (Modelo Vista Controlador). Este patrón de diseño clásico es usado frecuentemente por aplicaciones que necesitan la habilidad de mantener múltiples vistas de los mismos datos.

El patrón MVC se fundamenta en una clara separación de los objetos en una de tres categorías – modelos para mantener los datos, vistas para presentar todo o parte de los datos, y controladores para el manejo de eventos que afectan el modelo o vista(s)³.

Este patrón se puede plasmar de manera casi directa en un entorno J2EE. Los *beans* empresariales constituyen el modelo, la vista lógicamente son las páginas JSP mientras que el control lo realizan clases sencillas a través de *frameworks* de apoyo o directamente *Servlets*.

A continuación se muestra la figura 2, ésta representa la arquitectura del sistema, además señala el producto utilizado en cada componente de la misma.

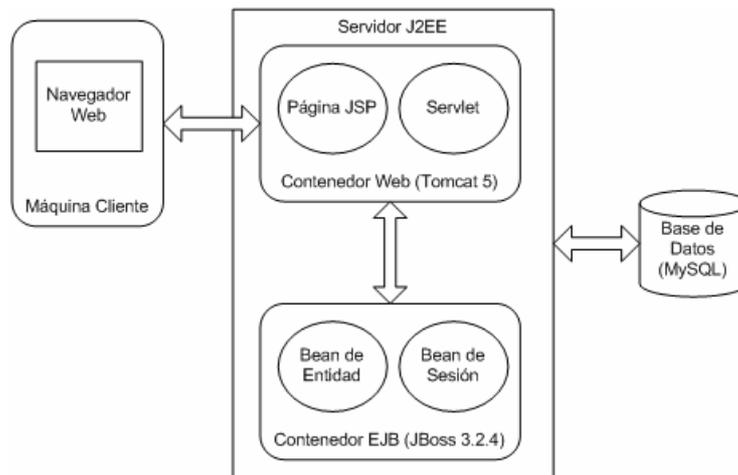


Figura 2 Arquitectura de E-Guana

El módulo de Administración de E-Guana implementa componentes tanto para cumplir los requerimientos del E-Procurement como otros para ofrecer una solución más flexible. Esta flexibilidad resultó en la clase denominada “unidad de negocio”, es decir, las entidades que conforman a cada empresa (Por ejemplo las Facultades dentro de la ESPOL). Cada unidad de negocio es la que agrupa a los usuarios que pertenecen a la empresa, además cada una de éstas, a través del usuario autorizado, puede emitir requerimientos de productos o servicios para que sean atendidos por la unidad principal de la empresa (esta unidad de negocio es identificada como la matriz de la empresa), ya sea por medio de una licitación o una compra directa.

El resto de clases componentes del sistema, implementadas por este módulo, corresponden a productos, categorías de productos y variables del sistema, que en conjunto se representan a continuación en la figura 3.

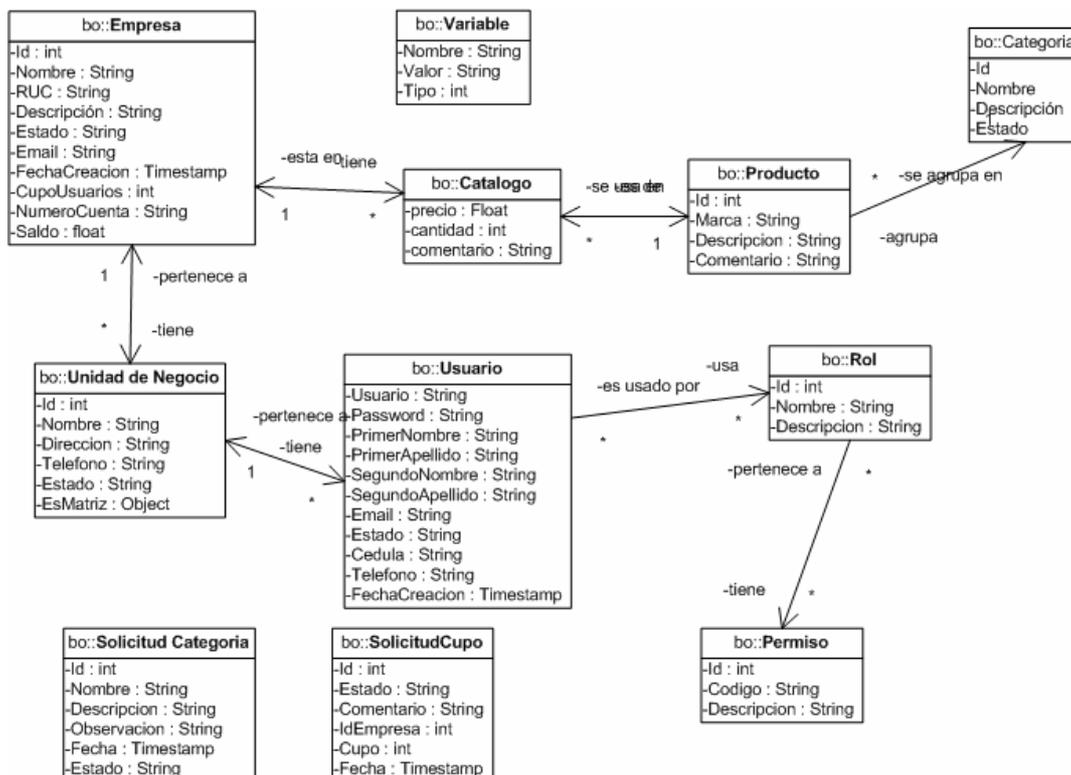


Figura 3 Diagrama de clases del módulo de Administración

Cada una de estas clases se implementa como un *bean* de entidad y a la par se implementa un *bean* de sesión que realiza todas las tareas mantenimiento. De esta forma si los componentes del sistema necesitan tener acceso a la funcionalidad provista por la capa de negocios de este módulo, tan sólo deben hacer uso de los servicios expuestos de lo *beans* de sesión.

La parte Web, como se indicó anteriormente, se implementó con páginas JSP ayudado del *framework* Sofia (Sofia permite el manejo de la vista a través de una arquitectura de componentes que pueden notificar su estado a clases Java de control). El intercambio de datos ente la capa Web y la capa de negocio (los *beans* empresariales) se realiza a través de clases de soporte conocidas como objetos de transferencia de datos (data transfer objects), que en su forma más simple es una clase que mapea un *bean* de entidad.

Un punto aparte corresponde a la implementación del módulo de Pagos de Transacciones. Aunque parte de su arquitectura es similar a lo hecho en el módulo de Administración, al ser éste el encargado de comunicarse con las entidades bancarias para los débitos o créditos resultantes de las compras y ventas, es necesario hacer uso de otro tipo de arquitectura para lograr esto.

Debe notarse claramente que la comunicación entre E-Guana y la entidad bancaria debe ser asincrónica, partiendo de la simple consideración de un cliente debe poder navegar dentro del sitio y hacer uso de todas las funciones sin tener que esperar a la respuesta del banco a un pedido de débito resultado de una compra. Afortunadamente J2EE provee los llamados *beans* manejados por mensajes, cuya función es atender o procesar los mensajes receptados por otros componentes creados con ese fin (colas de mensajes) y que, dependiendo del contenedor, pueden ser configurados tan fácilmente como a través de simples archivos XML.

La figura 4 corresponde a las clases utilizadas en el módulo de Pagos de Transacciones:

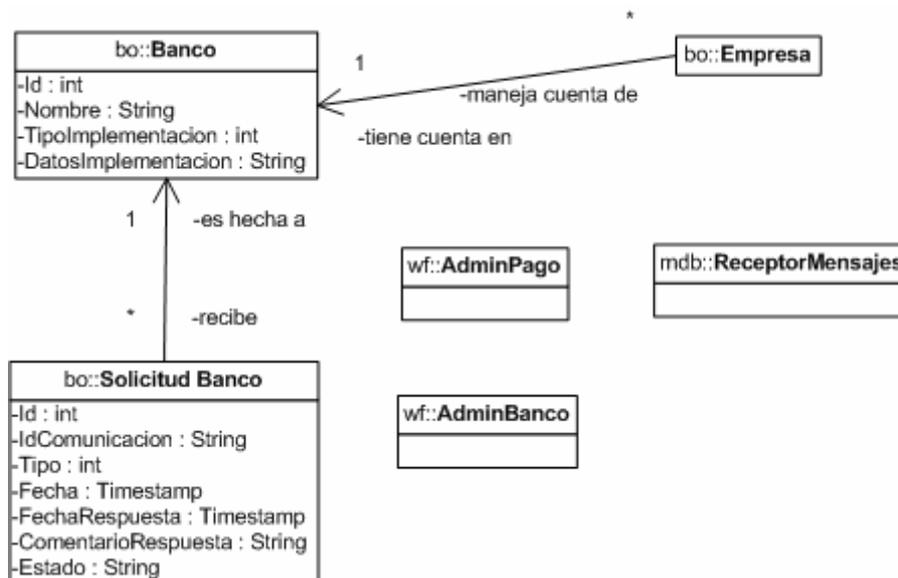


Figura 4 Diagrama de clases del módulo de Pagos de Transacciones

3. Configuración y Despliegue

Cada módulo Web corresponde a uno de los módulos de E-Guana, por lo tanto existirán cinco de estas aplicaciones. La configuración de seguridad de cada uno de ellos se resume en que el acceso solo se permitirá a usuarios registrados que tengan alguno de los roles soportados por el sistema (Administrador del Sistema, Administrador de la Empresa, Vendedor y Comprador). El módulo de Administración tiene lógicamente una sección que puede ser utilizada por el Administrador del Sistema.

Otro aspecto de la configuración Web, es la encriptación de la comunicación a través del protocolo HTTPS, para lograr esto tan solo es necesario asegurarse de modificar el descriptor de la aplicación Web para que la transmisión de datos sea identificada como confidencial.

Un punto adicional de configuración, es lo denominado *single sign-on*, esto quiere decir que cuando un usuario se autentique en una de las aplicaciones Web, no tenga que hacerlo nuevamente cuando se dirija a otra de las aplicaciones que conforman el sistema. Aunque esto no impide el funcionamiento correcto del sistema, sin duda es una función deseable para la comodidad del cliente. El *single sign-on* no es forma parte de la especificación Web por lo que es dependiente del contenedor Web en el que se haga el despliegue.

En cuanto al despliegue de la aplicación en el servidor de aplicaciones (JBoss), todos los componentes se empaquetan en una aplicación empresarial. Esta aplicación está conformada por el módulo EJB que contiene todos los *beans* empresariales y archivos descriptores y por cada uno de los módulos Web.

Completado lo anterior, tan solo es necesario ubicar el archivo de la aplicación empresarial en el directorio provisto para ese fin por el servidor JBoss.

CONCLUSIONES

Los sistemas de E-Procurement constituyen una alternativa que vale la pena ser considerada por empresas que quieren mejorar la relación con sus proveedores, así como por estos últimos si quieren ampliar su mercado y por consiguiente aumentar sus ganancias. Otro punto muy valioso a destacar es la considerable reducción de costos, para ambos entes participantes de las compras, como ejemplo el comprador puede evaluar productos muchos más rápidamente y el vendedor puede llegar a más clientes sin tener que gastar más.

E-Guana provee una solución E-Procurement de código abierto utilizando la arquitectura J2EE, lo que trae consigo varios beneficios, desde la reducción de costos hasta la independencia de productos, pudiendo adaptarse a las necesidades del cliente que adopte el sistema.

Indudablemente el uso de la tecnología J2EE es un desafío, pero que, luego de examinar los beneficios y los inconvenientes, vale la pena enfrentar. Este desafío es menor aún con las nuevas versiones de esta tecnología.

El patrón MVC es la solución más natural para el desarrollo de aplicaciones Web y más específicamente aquellas basadas en J2EE, ya que el mapeo es casi directo. De igual forma, la nueva versión de J2EE provee la tecnología Java Server Faces (para la capa Web) que hace más fácil la adopción de este patrón.

REFERENCIAS

- 1 P. Acosta Carvajal, L. Barreno Valdivieso, A. Guerrero Vasco, "Módulo de Administración de E-Guana" (Tópico, Facultad de Ingeniería en Electricidad y Computación, Escuela Superior Politécnica del Litoral, 2006).
- 2 S. Hamilton, J. Myron y J. Leech, octubre 2000, Entering the Dynamic New E-procurement Marketplace, <http://www.eJiva.com>.
- 3 eNode, Inc, septiembre 2002, Model-View-Controller Pattern, <http://www.enode.com/x/markup/tutorial/mvc.html>.