REINGENIERÍA DE LA SECCION CINE DEL PORTAL MIPUNTO.COM

Benjamín Arismendi G.,Marianella Aveledo M.

Universidad Simón Bolívar
Departamento de Procesos y Sistemas
A.P.: 89000
Caracas, Venezuela.
Email:maveledo@usb.ve

Abstract. En el presente artículo, se expone la experiencia vivida en Venezuela con un proyecto de reingeniería que se realizó a un portal que maneja entre otros tópicos, una aplicación para cines llamado MIPUNTO.COM de la empresa Telcel Bellsouth.En este sentido se planteó la reingeniería de la aplicación cine del portal para la migración de la misma hacia una arquitectura multicapas basada en el patrón Modelo-Vista –Controlador[1].

1. Introducción

El avance de las tecnologías relacionadas con Internet ha hecho que se abran nuevos horizontes para los negocios y para las empresas, ya que esta tecnología presenta grandes bondades. Con la creciente popularidad de Internet, ha surgido en las empresas la necesidad de utilizar aplicaciones Web para hacerse más competitivas en su negocio.

Telcel Bellsouth [2] a través del portal MIPUNTO.COM, busca la eficiencia de todas sus aplicaciones para ser mas competitivos con los servicios que brindan. Por lo tanto, se empezó a detectar la necesidad de llevar estas aplicaciones hacia un estándar arquitectónico. Mediante este cambio, que se denomina reingeniería, se pretende crear bases sólidas en los sistemas, permitiendo la escalabilidad de una forma natural, unificando criterios de diseño, optimizando el trabajo a la hora de realizar mantenimiento a dichas aplicaciones.

Este proyecto de reingeniería se basó en dos aspectos principales; en primer lugar, en la migración de la aplicación de cine, hacia una arquitectura basada en el patrón Modelo-Vista-Controlador (MVC) [1], que permita mantener dicha aplicación bajo el estándar de la organización y que a su vez permita realizar labores de mantenimiento y crecimiento de ésta de una forma mas simplificada y organizada. Y en segundo lugar, en la creación de nuevas administraciones delegadas que permitan el manejo de promociones y comisiones, a través de mecanismos de configuración sencillos, sin la necesidad de realizar cambios en el código de la aplicación.

El presente artículo consta de las siguientes secciones: la Sección 1 es la presente y constituye la introducción del trabajo, la Sección 2 presenta el proyecto y sus objetivos, en la Sección 3 se encuentra la solución que se propuso, la Sección 4 describe el proceso de implementación llevado a cabo, en la Sección 5 se exponen las conclusiones y finalmente en la Sección 6 se presentan las referencias bibliográficas.

2 Planteamiento del Problema

MIPUNTO.COM es un portal de contenido venezolano, que busca satisfacer las cuatro principales necesidades de los usuarios dentro de Internet, definidas como: información, servicios, entretenimiento y sentido de comunidad.
Este portal, es una plataforma de gestión de contenido, y su estrategia principal es conocer las características y necesidades de cada uno de los usuarios para ofrecerles servicios de contenido adaptados a sus necesidades.
El objetivo general del proyecto es la Reingeniería del Flujo de Compras de la sección Cine del portal MIPUNTO.COM y una extensión de las administraciones delegadas de la sección cine, permitiendo el manejo de comisiones, y aprovechando el modelo de datos actual para el manejo de cambios de precio y aforos.
Los objetivos específicos del proyecto son los siguientes:

- Analizar el sistema actual y los requerimientos de los usuarios finales, entendimiento del alcance funcional del proyecto y las implicaciones operativas, técnicas y de negocio.
- Diseñar el sistema. Donde se deben definir las estructuras de datos, los módulos y los procesos involucrados en la compra de entradas.
- Desarrollar los módulos y programar los componentes para la aplicación.
- Ejecutar las pruebas unitarias y de integración.

Uno de los principales problemas hallados, fue la utilización de archivos planos para el almacenamiento de información importante como lo son el aforo de cada función y el costo de los boletos.
La administración delegada de promociones no existía, por lo cual se invertían muchos recursos en el mantenimiento de la aplicación al momento en que surgía un cambio en alguna promoción y/ó comisión, los cuales se presentan con cierta regularidad, o al momento en que alguna de estas no aplicaba.
En ese sentido, la reingeniería de la sección Cine del portal MIPUNTO.COM se basó en primer lugar, en resolver aquellos problemas asociados al diseño e implementación del sistema actual, mediante un análisis y diseño basado en la identificación de oportunidades de mejoras en el proceso de compra de boletos, creando una arquitectura que permitiera el cambio de precios, comisiones, aforos y promociones a través de la administración delegada; y en segundo lugar, proponer e implementar la arquitectura para un nuevo modelo de datos que sea capaz de manejar todos los conceptos relacionados con la operación del espectáculo del cine con el dinamismo y cambios que por naturaleza este posee.

3 Solución Propuesta

Tal y como se expuso en el planteamiento del problema, el sistema que existía para el manejo de la aplicación de cine del portal MIPUNTO.COM, era un sistema basado en una arquitectura de tres capas donde existían páginas de la capa de presentación, clases pertenecientes a la capa de la lógica de la aplicación encargada de parte de la implementación de las reglas del negocio y la base de datos en la capa del repositorio de datos. Después de analizar la arquitectura existente, se encontró una gran divergencia en cuanto al estándar de modelaje de aplicaciones bajo el patrón MVC [3] en un entorno web, ya que las páginas de la capa de presentación incluían gran parte del procesamiento de información y de aplicación de reglas del negocio, funciones que debería delegar a la capa inferior. Por otra parte, también se observó como existían datos almacenados en archivos planos, lo cual está en contra del concepto de arquitectura que se pretende implantar.
La reingeniería de la sección de cine, se basó entonces, en una arquitectura que podría definirse como de n capas, ya que en realidad lo que hace es distribuir las funciones de la capa de la lógica de la aplicación en nuevas capas. Esta distribución se basa en los lineamientos del patrón MVC**.** Para la realización de la reingeniería se identificaron dos escenarios principales, los cuales contemplan los puntos mencionados anteriormente. Estos dos escenarios se describen a continuación:

3.1 Escenario 1: Integración con el nuevo modelo de compra

Este escenario contempla la integración del nuevo flujo de compra de cine con el modelo de compras actual, el cual representa un modelo genérico utilizado por la empresa para la venta de productos a través del portal.

3.2 Escenario 2: Construir el módulo de pago y registro de transacción

Este escenario contempla la construcción de un nuevo módulo de pago basado en las experiencias del modelo de compras, el módulo de pago del portal de voz y el antiguo método de pago de cine, donde se enmarque el proceso de pago en una transacción.

3.3 Escenario seleccionado

Luego de un proceso de análisis de los dos escenarios descritos anteriormente, se tomó la decisión de optar por el primer escenario, el de integración al modelo de compras. Finalmente, se tuvo el flujo de compras de la sección de cine del portal MIPUNTO.COM, bajo un patrón de n capas, haciendo uso de las bondades del Weblogic Server [4], a través de componentes que permiten definir transaccionalidad en las aplicaciones y además integrado al modelo genérico de compras.

4 Desarrollo

El proyecto se desarrolló siguiendo la metodología "Rational Unified Processs" (RUP) [5] y se realizó de acuerdo a un modelo de n capas que obedece al patrón MVC propuesto por J2EE [6], tal como fue explicado anteriormente, este patrón está conformado por 3 capas a saber: la Capa Vista, la Capa Modelo y la Capa Controlador. La Capa Vista se implementó mediante el uso de Struts, y su interacción con la Capa Controlador y la capa Modelo se realizó mediante el uso de Java RMI o RMI-IIOP que es una versión Java compatible con CORBA. En cuanto la Capa de Modelo, ésta hará uso de la tecnología Java Database Connectivity (JDBC), para comunicarse con el manejador de base de datos relacional.

5 Conclusiones

Una vez culminado el proyecto descrito en el presente artículo 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 Tres 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.
WebLogic Server, es un servidor de aplicaciones que funciona como plataforma para el desarrollo e implementación de aplicaciones empresariales multicapas. WebLogic Server centraliza los servicios requeridos por la aplicación, tales como: Servidor Web, componentes del negocio y el acceso a otros sistemas empresariales. Usando tecnologías como el "caching" y el pool de conexiones se mejora el uso de los recursos y el desempeño de las aplicaciones. Así mismo, WebLogic provee de un nivel de seguridad empresarial elevado y de poderosas facilidades de administración.

6 Referencias Bibliográficas

1. Burbeck, S.: Applications Programming in Smalltalk-80(TM):How to use Model-View-Controller(MVC).
2. TELCEL http://www.telcel.net.ve
3. Vikram,S. : Model View controller for Multiple ModeAccess. A White Paper, Analytica India Inc. 2001.
4. Girdley,M., Woolen, R.,Emerson, S. : J2EE Aplications and BEA Weblogic Server, Prentice Hall, INC. 2002.
5. Kruchten, P. :Architectural Blueprints. The 4+1 View model of software Architecture, http://www.rational.com,1996.
6. J2EE Paterns, Best Practices. http://www.corej2eepatterns.com/index.html