Desarrollo de aplicaciones Web utilizando herramientas FLOSS[1]
Una experiencia en el CENPAT[2]-CONICET

Gustavo SAMEC, Renato MAZZANTI
Centro Nacional Patagónico – Servicio Centralizado de Computación
Bvrd. Brown 3700, Puerto Madryn, Chubut, Argentina.
Tel/Fax 54-02965-450401
{gsamec, renato}@cenpat.edu.ar

RESUMEN

El proyecto tiene como objetivo evaluar distintas herramientas FLOSS, definir una arquitectura de software para el desarrollo de aplicaciones Web y realizar un desarrollo aplicando las herramientas y arquitectura seleccionada.
Actualmente existen varios proyectos de ley que fomentan la utilización de herramientas FLOSS en entes estatales. Por otro lado otras instituciones de investigación científica han priorizado el uso de este tipo de herramientas [1]. Siguiendo este lineamiento, nuestra intención es utilizar las mismas en el desarrollo de aplicaciones que el CENPAT necesita informatizar.

1. INTRODUCCION

A la hora de abordar este proyecto, nos propusimos dar solución a un problema concreto: informatizar el sistema de licencias del personal. El incremento del personal en los últimos años y la complejidad de las leyes laborales hacen muy difícil llevar un control efectivo en forma manual. Por otro lado, resulta muy laborioso generar informes en tiempo y forma acerca del estado de los empleados.

A los efectos de atender a las necesidades más urgentes de la organización, se decidió dividir el proyecto en dos etapas:

Etapa 1: Desarrollar un sistema Web accesible a través de la Intranet que permita: Automatizar el registro de información y su validación de acuerdo a las leyes vigentes[3]. Consultar datos personales y de personas a cargo. Generar informes. En todos los casos se habilitará el acceso al mismo de acuerdo al rol que ocupa el usuario en la organización.

Etapa 2: Extender la funcionalidad del sistema para: Habilitar el pedido de licencias a través del sistema y desarrollar su *workflow* de autorizaciones. Exportar datos a distintos formatos compatibles con paquetes Office. Agregar componentes de seguridad y permitir su acceso desde Internet.

Actualmente está completa la primera etapa del proyecto, encontrándose el sistema en producción. Este año está planificado avanzar en la segunda etapa del mismo.

---

[1] Free/Libre Open Source Software
[2] Centro Nacional Patagonico
[3] Decreto N° 3413/79 y modificatorias del CONICET

2. PROYECTO

2.1 Características Generales

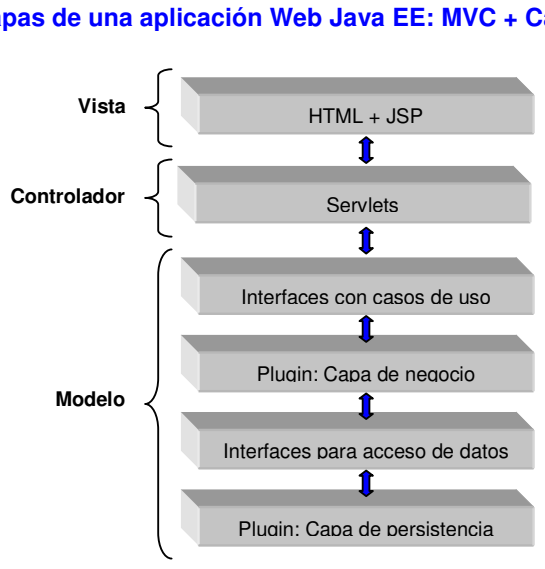
Dada las características del proyecto se desarrolló una aplicación Web usando tecnología J2EE 5 [4,9]. La misma se encuentra instalada en un servidor Linux. Como servidor de aplicaciones usamos Tomcat [19] y como servidor de base de datos MySql [18]. Los clientes acceden a la aplicación desde un browser (Firefox, IE) a través de la Intranet.

La aplicación fue diseñada siguiendo la arquitectura que fijan los patrones Modelo-Vista-Controlador (MVC) y Capas (Layers) [2,3]. El objetivo principal de esta arquitectura es separar las distintas capas de desarrollo facilitando el mantenimiento y la evolución de las aplicaciones.

Para la capa de presentación (la vista) se eligió el framework JavaServer Faces (JSF) [5,6] que permite la elaboración de pantallas, mapeo entre los formularios y sus clases en el servidor, validación, conversión, gestión de errores, internacionalización e incluir componentes más complejos.

La capa controlador esta implementada utilizando Faces Servlet que provee JSF, que mediante un archivo XML permite su configuración.

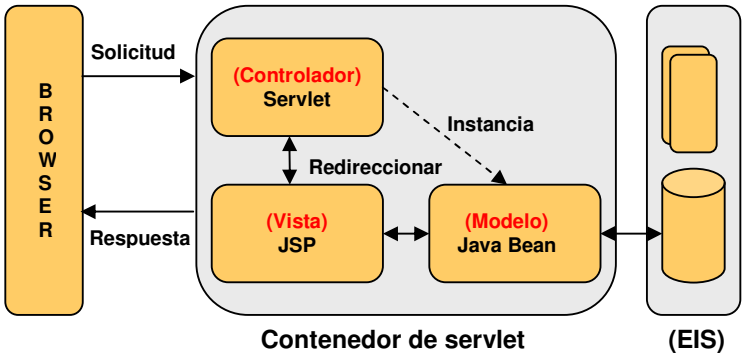
Para las capas de negocio y persistencia se optó por una solución basada en servicios. La persistencia de clases se sustenta en DAO[4], que mantiene aislada la capa de persistencia de la capa de negocio. Ambas capas se implementaron usando POJO[5], clases simples de Java sin dependencias de ningún framework. Para la capa de persistencia se utilizó Hibernate[7] como herramienta de mapeo objeto-relacional.



[4] Data Access Object
[5] Plain Old Java Object

Se utilizó MVC nivel 2, en esta versión la arquitectura esta centrada en servlets. Las solicitudes se pasan desde el browser del cliente al controlador (que es un servlet). El controlador decide a que vista (JSP[6]) se pasará la petición. La vista invoca a métodos en el modelo (Java Bean), que pueden acceder a la capa EIS[7] y devuelve el objeto de respuesta al contenedor Web para que éste lo pase al browser del cliente.

MVC Nivel 2 (Arquitectura centrada en Servlet)



2.2 Herramientas empleadas

J2EE 5: (Java 2 Enterprise Edition) plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java. Proporciona un potente conjunto de APIs que reduce el tiempo y la complejidad del desarrollo y mejora el rendimiento de las aplicaciones. [9].

JavaServer Faces: estándar presentado por Sun para el desarrollo de la capa de presentación Web. Forma parte de la especificación J2EE 5 [10].

MyFaces: proyecto de la fundación Apache que ofrece una implementación en código abierto de JavaServer Faces y un amplio conjunto de componentes adicionales. En nuestro caso utilizamos componentes de Tomahawk [11].

Hibernate: motor de persistencia de código abierto. Permite mapear un modelo de clases a un modelo relacional sin imponer ningún tipo de restricción en ambos diseños [12].

C3P0: permite establecer un pool de conexiones de código abierto. Fácil de emplear y configurar con Hibernate [13].

Log4J: permite gestionar todos los aspectos de los logs. Utilizado en el desarrollo para debug código y en producción para registrar accesos y modificaciones de datos para su posterior auditoria [14].

JUnit: Estándar en Java en código abierto para la creación de tests unitarios [15].

JasperReports: herramienta para la creación de informes [8, 16], utilizada junto con iReports, front-end gráfico que facilita la edición de los mismos [17].

---

[6] Java Server Pages
[7] Executive Information System

MySQL: sistema de gestión de base de datos relacional junto con herramientas GUI[8] tales como MySQL Administrador y MySQL QueryBrowser para facilitar las tareas de administración de la misma. Se utilizó el driver MySQL Connector/J para la conexión de Java con MySQL [18].

Tomcat: servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets [19].

Eclipse: plataforma de desarrollo de software. Este IDE permite adicionar módulos (plugin) que extiende su funcionalidad, integra y facilita el desarrollo de las aplicaciones [20].

2.3 Problemas durante el desarrollo

La utilización de Hibernate nos llevo a modificar el diseño de la base de datos a los efectos de controlar la concurrencia y mejorar su eficiencia.

El driver para conexiones que provee Hibernate funcionó adecuadamente en la fase de desarrollo pero trajo problemas de conexión cuando la aplicación se puso en producción. La utilización de C3P0 solucionó estos inconvenientes.

Para la utilización de componentes Tomahawk se necesita incluir un filtro, Extensions Filter, en el web.xml de la aplicación, el mismo trae conflicto en la visualización de los reportes PDF, generados por JasperReports, en el navegador. Para solucionar este problema fue necesario incluir un servlet que se ejecuta por fuera de las especificaciones de JavaServer Faces.

Se encontraron bugs en componentes Tomahawk, muchos de los cuales ya están solucionados en nuevas versiones, para ello no solo hubo que actualizar varias bibliotecas de las clases que utiliza sino también migrar a nuevas versiones varias de las herramientas que comparten estas bibliotecas.

2.4 Resultados del Proyecto

Se logro seleccionar un conjunto de herramientas que satisfacen los requerimientos necesarios para el desarrollo de sistemas de información Web de pequeña y mediana envergadura.

Se definió un modelo de arquitectura por capas que facilitan el mantenimiento y evolución de las aplicaciones.

Se construyó un conjunto de componentes aplicables a desarrollos de características similares.

Se implementó la primera etapa del proyecto con un grado razonable de productividad.

Se adquirió conocimientos y experiencia en el uso de este tipo de herramientas que estamos volcando en la tutoría de alumnos que están trabajando en su tesina de la Licenciatura en Informática en la UNPSJB[9].

---

[8] Graphical User Interface

[9] Universidad Nacional de la Patagonia San Juan Bosco

3. CONCLUSIONES

En la actualidad encontramos una amplia variedad de herramientas FLOSS que cubren todas las fases de desarrollo de un sistema de información Web.

Muchas de ellas están desarrolladas por grupos muy activos que permanentemente mejoran y corrigen bugs en nuevas versiones.

La falta de soporte técnico se ve largamente compensada por una amplia comunidad de desarrolladores que participan en foros y proponen soluciones a inquietudes y problemas que se presentan.

La integración entre las mismas no siempre es la mejor, al igual que su facilidad de uso. Tareas que en otros productos comerciales se realizan con wizards o potentes editores visuales, en estas herramientas hay que hacerlo a mano o con editores rudimentarios. Si bien esto implica un esfuerzo adicional para el desarrollador, no afecta a la calidad final del producto obtenido y es muy probable que nuevas versiones de estas herramientas cubran estas falencias.

Los resultados hasta aquí obtenidos nos alientan a seguir utilizando estas herramientas e investigar otras (tales como Spring, Maven, etc.) para completar este proyecto y encarar futuros desarrollos de mayor envergadura.

Referencias:

[1] Clara Cala Rivero, Ángel L. Rodríguez Alcalde, José Ángel Barroso - Reflexiones sobre el Framework de desarrollo del Consejo Superior de Investigaciones Científicas – CSIC España 2006
[2] Juan Raposo Santiago - Introducción al Desarrollo de Aplicaciones Empresariales - Universidade Da Coruña, Departamento de Tecnoloxías da Información e as Comunicacións (TIC) - 2007
[3] Juan Medín Piñeiro, Antonio García Figueras – Comunicación 209 - Hacia una arquitectura con JavaServer Faces, Spring, Hibernate y otros frameworks - Jornadas sobre Tecnologías de la Información para la Modernización de las Administraciones Públicas (TECNIMAP) – Sevilla 2006
[4] Eric Jendrock, Jennifer Ball, Debbie Carson, Ian Evans, Scott Fordin, Kim Haase - The Java™ EE 5 Tutorial – Addison Wesley (2006)
[5] Kito D. Mann - JavaServer Faces in Action – Manning (2007)
[6] David Geary, Cay Horstmann - Core Javaserver Faces - Prentice Hall (2da Ed. 2007)
[7] Christian Bauer, Gavin King - Java Persistence with Hibernate – Manning (2005)
[8] David R. Heffelfinger - JasperReports for Java Developers – Packt (2006)

Referencias Web:

[9] Java EE at a Glance. http://java.sun.com/javaee/
[10] JavaServer Faces Technology. http://java.sun.com/javaee/javaserverfaces/
[11] The Apache MyFaces Project. http://myfaces.apache.org/
[12] Relational Persistence for Java and .NET. http://www.hibernate.org/
[13] c3p0:JDBC DataSources/Resource Pools. http://sourceforge.net/projects/c3p0
[14] Logging Services. http://logging.apache.org/log4j/
[15] JUnit. framework to write repeatable tests. http://junit.sourceforge.net/
[16] Embeddable open source Java reporting library.
http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/
[17] An open source graphical report designer for JasperReports.
http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/ireport/
[18] MySql. The world's most popular open source database. http://www.mysql.com/
[19] Apache Tomcat. http://tomcat.apache.org/
[20] Eclipse - an open development platform. http://www.eclipse.org/