

Entregable Programacion Orientada aspectos

Christian Daniel Avila Sánchez.

Instituto tecnologico y de estudios superior de monterrey

Motivación: Creación de un proyecto Web que aplique los conocimientos obtenidos en el transcurso del curso de Programación orientada a Aspecto, así como su unión con el framework de Spring MVC + Spring AOP

1 INTRODUCCIÓN

La programación orientada aspectos, AOP, es un nuevo paradigma que esta cobrando fuerza ya que cambia la manera de atacar a los "*Cross cutting concerns*" puesto que ocupa un modelado de la información transversal, que corta a los diferentes módulos que componen la aplicación.

La AOP al ser un paradigma reciente, en comparación a objetos o estructurado, aun no esta totalmente estandarizado y existen diferentes frameworks para casi todos los lenguajes de programación que existen.

A lo largo de este documento nos centraremos en el Framework de Spring para JAVA. Este framework es el de mayor adopción en la comunidad de desarrolladores por la facilidad de uso y la amplia gama de posibilidades que se pueden desarrollar.

Durante los siguientes puntos se detallara el proyecto final para el curso de Programación Orientada a Aspectos, que consiste en desplegar información sobre los diferentes meteoritos que han caído en el planeta de los cuales se tiene registro. La información se obtuvo de fuentes públicas y solo es de carácter informativo.

2 METHODS

En esta sección se detallara los elementos utilizados para el desarrollo del proyecto y se explicará cuales fueron los retos en cada una de las etapas.

Se propone utilizar un modelo por capas, que consiste en tres:

- Capa de datos.
- Capa de negocio.
- Capa Vista.

Cada una de estas capas tiene una tarea específica, con lo cual se extrae la lógica del negocio de la capa vista y datos.

Para la capa de datos se utilizó un data set público, este se obtuvo desde el sitio web de Visualizing [1], que servirá de base del proyecto y será la información que se desplegara en la capa vista.

A su vez para terminar de describir la capa de Datos se propuso utilizar una base de datos orientadas a documentos, CouchDB [2]. Este motor al ser una base de datos NoSQL la forma en como almacena y recupera la información es muy diferente. CouchDB utiliza los métodos REST para el manejo de la información (documentos). Los datos que conoce CouchDB son de tipo JSON con lo cual se pueden realizar consultas mediante código JavaScript y mapas de reducción.

Para la capa de negocio se propone el uso del framework Spring, que es el framework mas aceptado por la comunidad de java. La parte que nos interesa ocupar es Spring MVC, que es orientado a web por lo cual encaja a las necesidades del proyecto. Adicional para los "*Cross cutting concerns*" se utilizará Spring AOP, que es la parte orientada a Aspectos, toma como base AspectJ y lo integra con su modelo de inyección de dependencias y anotaciones características de Spring.

Por ultimo en la capa vista se utilizará el framework D3 Data Driven Documents [4], para mostrar los datos de una manera dinámica y a su vez amigable al usuario. Este framework ocupa documentos JSON para la creación de los elementos visuales que van acorde con la base de datos CouchDB.

Para poder unir todos estos elementos se ocuparon elementos adicionales como es couchdb4j [5] que es un framework para conectarse a CouchDB desde Java, este framework es amigable ya que ocupa llamadas HTTP con métodos RESTfull para la obtención de los datos. Adicional a este elemento se crearon scripts para la consola que ocupa el comando **curl** para crear e insertar los datos desde un archivo de Excel. Este script esta junto el código del proyecto.

2.1 Utilización de las partes

Spring MVC se utilizó para la poder manejar las llamadas del cliente, capa vista. Se creó un Servlet que responde a la ruta principal y uno extra donde se puede realizar la consulta directo en CouchDB por año. Estos dos Servlets son "cortados" por un aspecto que escribe en consola cada uno de los accesos con la fecha y hora en que se accede al recurso. Esto funciona como auditoría y mantener un log de las consultas a base de datos.

En la capa vista se agrego un cuadro de texto y un botón de consulta para verificar un año en especial y verificar los datos que se cuentan.

3 R5RESULTS

Los resultados de este proyecto, como anteriormente se comentó, es una página web dinámica que permite realizar una búsqueda por año

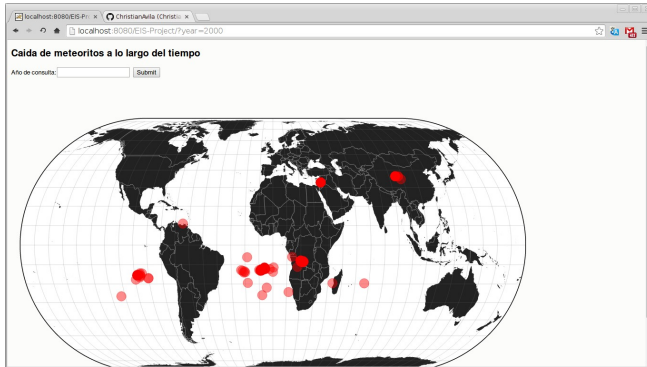


Figura 1. Ejemplo del resultado del proyecto

En la siguiente imagen se muestra el cuadro de texto donde el usuario puede realizar la consulta por año.



Figura 2. Cuadro de búsqueda por año.

En la siguiente imagen se muestra la evidencia del registro de cada uno de los accesos, esto se logró mediante un aspecto que corta cada cada controlador.

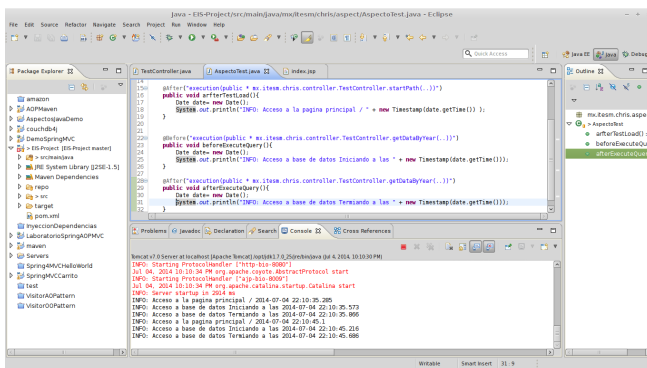


Figura 3. Bitácora de acceso de datos

ACKNOWLEDGEMENTS

Durante la realización de este proyecto se aprendió la parte de Aspectos orientados a un proyecto funcional, adicional a los aspectos se incursiono en dos tecnologías de reciente creación, las bases de datos orientadas a documentos y D3js.

Para concluir con este trabajo, quiero recalcar que estas tecnologías, se encuentran en constante desarrollo y evolución, por lo cual es la tendencia para el futuro y que a su vez todos tenemos que tener en mente. Este proyecto solo es una pequeña parte del potencial que está a la espera de ser utilizado por las industrias, con lo cual llevaría a programas de software con mayor calidad tanto en la parte de experiencia del usuario, a si como estabilidad y mantenimiento del mismo.

REFERENCES

- [1] Visualizing.org, <http://www.visualizing.org/datasets/meteorite-landings>, 2014
- [2] Apache CouchDB, <http://couchdb.apache.org/> 2014
- [3] Spring, <http://projects.spring.io/spring-framework/> 2014
- [4] D3js <http://d3js.org/> 2014
- [5] Couchdb4j, <https://github.com/mbreese/couchdb4j> 2014
- [6] Ramnivas, L. (2010). Aspect J in Action Enterprise AOP with Spring Applications. : Manning Publications Co.