ETS - Web Application Development

ECommerce

Especificación de la Arquitectura

04 de junio de 2015

CAPÍTULO 1

Especificación de la Arquitectura

1.1. Diseño en capas

La arquitectura del proyecto se diseñó con base en el patrón Modelo Vista Controlador (MVC) parar separar la lógica de negocios, los datos y la interfaz de usuario, esta forma de trabajo permitió reutilizar el código de algunos componentes y además podría facilitar el futuro mantenimiento.

En la figura 1.1 se muestra el diseño por capas de la aplicación y una aproximación a la manera en que interactúan. Podemos observar que el usuario solicita recursos a través de la capa de presentación, la cual a su vez solicita a la capa controlador que procese dicha solicitud, la capa controlador decidirá cómo debe procesarse la información y se auxilará de la capa modelo.
Finalmente cuando se obtiene el resultado, va pasando por cada una de las capas de regreso hasta llegar nuevamente a la capa presentación, la cual mostrará los resultados al usuario.

Como podemos ver caada una de las capas realiza su funcionamiento y delega la siguiente operación a alguna otra capa, lo cual permite separar los diferentes procesamientos a lo largo del sistema.

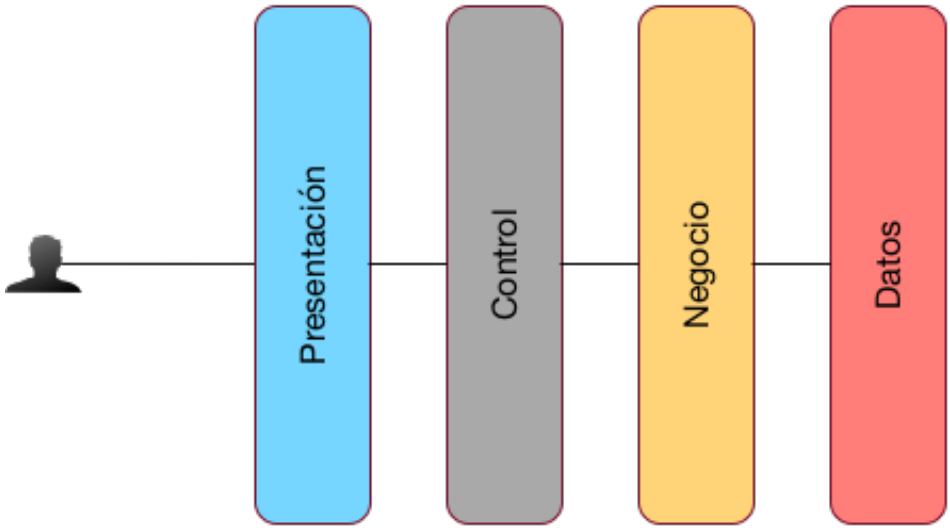

Figura 1.1: Capas

Debido a que cada capa presenta un corportamiento aislado y específico, se explicará a detalle cada una de ellas y su integración con las demás.En la figura 1.2 se muestra la arquitectura con los principales componentes de software utilizados.

1.1.1. Capa de presentación

La capa de presentación se utiliza para permitir al usuario interactuar con el sistema, a través de ella, se solicitarán recursos y se notificará cualquier cambio en el sistema por medio de mensajes. Su funcionamiento es obtener las acciones que el usuario realiza y enviarlas a la capa de control, con base en la respuesta obtenida esta capa actualizará su contenido, mostrando nuevos registro, registros actualizados, nueva vista y/o mensajes.

Los componentes de software utilizados en esta capa son los siguientes:

- Decorators, determina el conjunto de JSP's que constituirán la vista final del sistema.
- Layouts, determina la forma en que se agruparán los diferentes JSP's.
- JSP's, se encargan proveer vistas en HTML con base en una compilación previa del servidor.
- CSS's, se encargan de mostrar las vistas con un formato adecuado y agradable para el usuario.
- JavaScript, permite realizar procesamiento sencillo sobre determinadas acciones antes de llegar al servidor. Resulta altamente útil para realizar validaciones antes de ejecutar la petición y evitar que el servidor realice más procesamiento.
- JQuery, API que permite el uso de diferentes elementos preestablecidos basados en javascript, se utiliza para realizar comportamientos sobre la vista con base en peticiones realizadas por el usuario.

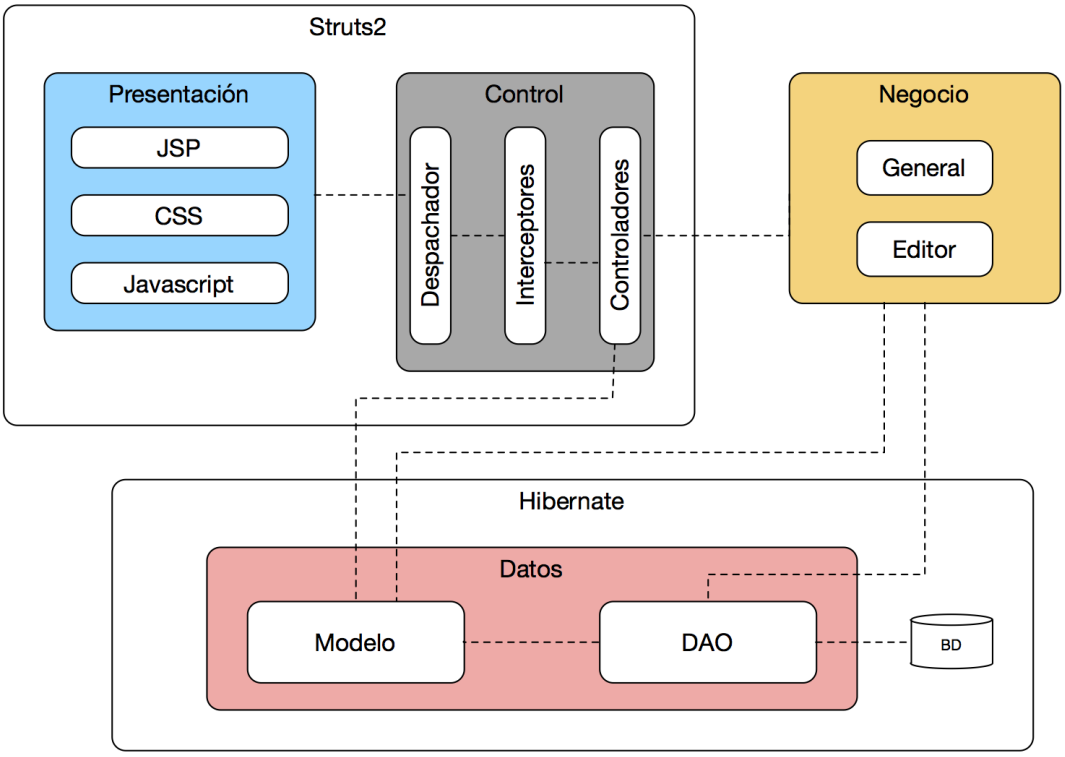

Figura 1.2: Arquitectura

- DataTable, API de JQuery que provee mecanismos para la creación de tablas con un funcionamiento preestablecido, como por ejemplo la paginación y las búsquedas sobre la tabla.
- JSON, formato permite el envío y recepción de datos entre la presentación y el control
- Struts2, framework que funciona como capa intermedia entre el controlador y la presentación, por parte de la presentación se encarga de pre-procesar los JSP's con base en información del controlador y finalmente constuir una vista para el usuario.

1.1.2. Capa de control

La capa de control se utiliza para procesar los datos proporcionados por la capa de presentación, con base en los recursos solicitados y un conjunto de datos, esta capa decidirá cómo deben procesarse los datos, una vez tomada la decisión deberá enviarlos a la siguiente capa para que sean procesados, con base en la respuesta obtenida reportará a la capa de control que cambios deben realizarse para notificar al usuario.

Los componentes de software utilizados en esta capa son los siguientes:

- JSON, formato permite el envío y recepción de datos entre la presentación y el control
- Interceptores, conjunto de clases que se encargan de interceptar las peticiones y procesar determinada información antes de llegar al controlador. Resulta bastante útil cuando es necesario validar repetidamente antes de procesar los datos.
- Controladores, conjunto de clases que procesarán las peticiones realizadas por el usuario por medio de la capa de presentación. Cuando el controlador tiene una respuesta lo notifica a la capa de presentación y se ejecuta el cambio visible para el usuario.
- Struts2, framework que funciona como capa intermedia entre el controlador y la presentación, por parte del control se encarga de automatizar las peticiones a realizar, brindando mecanismos que con base en determinado action se ejecuta determinada parte del código, esta parte es medular pues permite que el desarrollo sea considerablemente más rápido.

1.1.3. Capa de negocio

La capa de negocio se encarga de procesar los datos con base en la lógica del negocio. Esta capa determinará si es necesario realizar una transacción con la base de datos, o informar a la capa de control que hubo un error, si es necesario realizar una transacción se deberá realizar una solicitud a la capa de datos.

Los componentes de software utilizados en esta capa son los siguientes:

- Negocios, conjunto de clases que procesarán los datos enviados desde la capa de control, en ellas se implementa toda la lógica de negocio, con base en el procesamiento realizado y las reglas de negocio establecidas notificará a la capa de control el estado de la información, es decir, notificará si hubo cambios o si no hubo debido a que no se cumplieron determinados requisitos. Todo lo realiza delegando las transacciones (en caso de cumplirse los requisitos) al modelo.

1.1.4. Capa de datos

La capa de datos es la encargada de realizar las transacciones a la base de datos. Con base en las solicitudes realizadas por la capa de negocio, el único funcioamiento de esta capa de proveer mecanismos para realizar inserciones y extracciones de datos. Los componentes de software utilizados en esta capa son los siguientes:

- Hibernate, framework que automatiza el acceso a datos de la aplicación, para el caso de java, implementa la especificación JPA. Hibernate se encarga de asociar clases (modelos) a tablas en la base de datos, de este modo busca que cada miebro de la clase se encuentre asociado a algún dato en la base de datos. Con base en la asociación anterior, automatiza la creación de sentencias o querys, pues procvee mecanismos para la inserción y extracción de datos.
- MySQL, gestor de base de datos que aloja la base de datos del sistema, se encuentra gestionada directamente por Hibernate.
- DAO's, patrón de diseño que permite encapsular los mecanismos de accesso a datos.
- Modelos, clases que utiliza Hibernate para automatizar la inserción y extracción de datos con base en los mapeos.
- Mapeos, especificaciones de cómo es que se interrelacionan las clases (modelos) en el sistema. Se encuentran estrictamente basados en el esquema de la base de datos, ver figura 1.3.

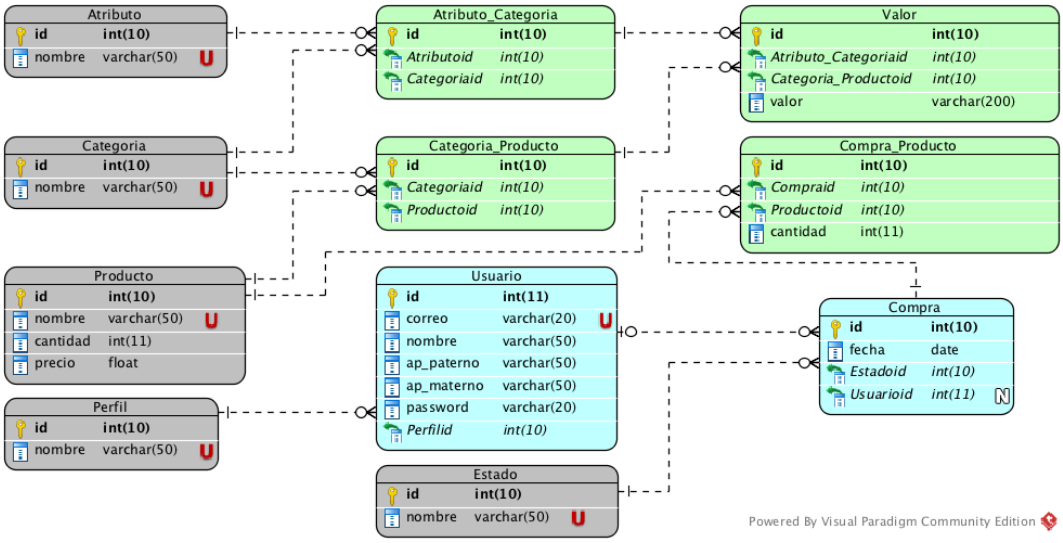

Figura 1.3: Esquema ECommerce