

## **Capítulo II. Análisis de herramientas y tecnologías de desarrollo.**

Dentro del desarrollo de la tesis el proceso de modelado del sistema fue hecho con el lenguaje de Modelo de Objetos llamado UML (Unified Modeling Language) [Unified Modeling Language Tutorial], la programación del sistema fue desarrollado con un lenguaje orientado a objetos, el cual puede ser Java, C++ u otro lenguaje.

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

La ventaja de utilizar UML como lenguaje de modelado es que es el lenguaje de modelado mas utilizado y además reúne una colección de las mejores prácticas en la ingeniería que han sido utilizadas con éxito para modelar sistemas grandes y complejos.

Ya que el software desarrollado será utilizado en el Web y para construir portales de Web el lenguaje utilizado será java, ya que es un lenguaje de alto nivel orientado a objetos que cuenta con diferentes tecnologías entre las cuales se encuentran los Servlets y Java

Server Pages (JSP). Además se usaran otras tecnologías para construir sitios en Internet como lo son javascript, HTML, ASP, XML entre otras.

**Servlets** [CORE Servlets and Java Server Pages]: Esta tecnología puede llevar a cabo todas las funciones de un programa de CGI. Este se ejecuta dentro de la máquina virtual de Java junto con el servidor Web por lo cual no requiere de la descarga de un programa externo. Debido a que Java cuenta con la capacidad de descargar clases en el tiempo de ejecución. Los Servlets pueden ser utilizados en el momento en que se soliciten.

Los servlets permiten tener un control total de la información que se va a desplegar en el HTML, además utiliza como medio de presentación documentos HTML que son contruidos "on the fly", es decir. En tiempo real. Otra ventaja es que los servlets se encuentran en el servidor y no saturan la maquina del cliente. Una posible desventaja es el tiempo que tardan en desplegarse las páginas HTML por medio de los servlets, ya estos son quienes escriben el código.

**Java Server Pages (JSP)** [Java Server Programming Edition]: La implementación actual de JSP está limitada como un lenguaje script. El diseño de JSP permitirá a otros lenguajes scripts ser soportados en un futuro. JSP ayuda en la creación de nuevos tags que son interpretados por el procesador JSP, permitiendo la extensión del lenguaje. Esta nueva propuesta tiene que ver con la codificación dentro de la propia página. Una vez que se accede a la misma, se ejecuta el código contenido, y la información desplegada en el cliente pasa a ser HTML puro generado en el momento de la consulta.

Los Java Server Pages nos permitirán tener código java para la manipulación de los datos y poder construir paginas de manera dinámica, que se actualicen directamente con la información de la base de datos. Finalmente el usuario vera el sitio como un documento HTML.

**HTML** [Programación en Web], Hyper Text Markup Lenguaje o lenguaje de definición de marcas es un lenguaje sencillo que permite marcar los documentos de hipertexto mediante unas etiquetas especificas, de este modo conseguimos darle a los documentos una cierta estructura.

**JavaScript** [Programación en Web]. Es un lenguaje compacto de scripts basado en objetos, para desarrollar aplicaciones cliente-servidor en Internet. El Navegador Netscape interpreta las declaraciones de JavaScript inmersos en las páginas de HTML y LiveWire permite la creación de aplicaciones cliente-servidor similar a los programas CGI (Common Gateway Interface).

## II.1 Estrategia de Diseño.

La arquitectura de una aplicación es la vista conceptual de la estructura de esta. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como esta distribuido este código.

Existe una arquitectura de aplicaciones de tres-capas, basadas en componentes, la cual fue presentada por Windows DNA (Universal Access Data), Llamada “Model View Controller Technology” [Model View Controller Technology]. Donde la meta es unificar las aplicaciones cliente y servidor y las aplicaciones basadas en la Web, lo cual es posible para aplicaciones de cualquier tamaño.

La arquitectura “Model View Controller Technology” de tres capas se muestra en la siguiente gráfica, cuenta con servicios específicos en cada capa que se comunican entre sí.

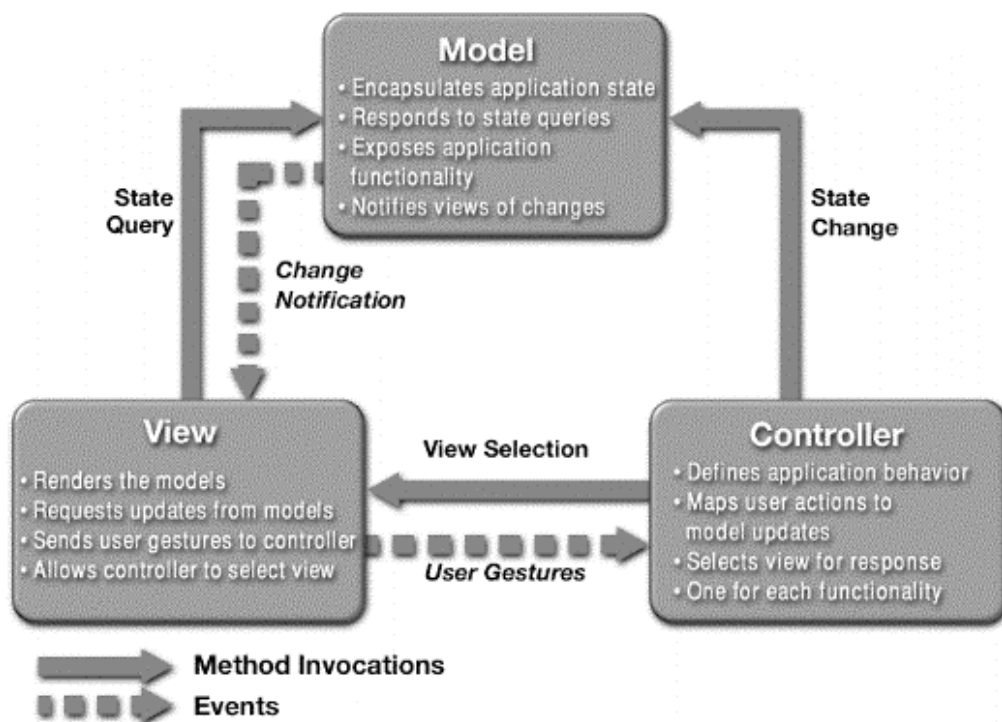


Figura 1. Diagrama del modelo Model-View-Controller

Las capas de arquitectura son las siguientes:

**Capa de presentación:** Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. También aseguran los servicios de negocios necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios.

Los servicios de presentación generalmente son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. Aún así, existen oportunidades para identificar servicios que residen en componentes separados.

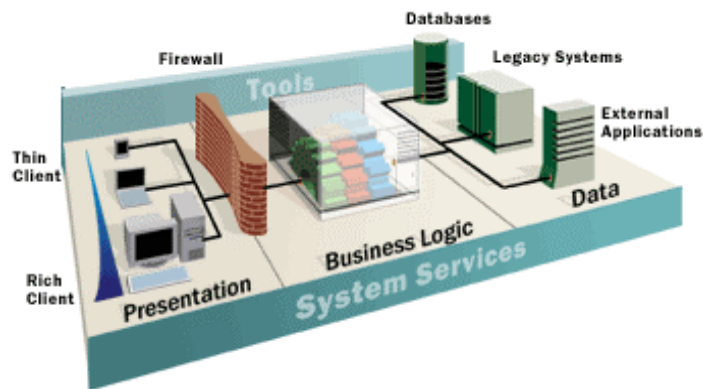


Figura 2. Ejemplo de aplicación del modelo “Model-View-Controller”

El cliente proporciona el contexto de presentación, generalmente un browser como Microsoft Internet Explorer o Netscape Navigator, que permite ver los datos remotos a través de una capa de presentación HTML.

Mediante el uso de componentes, se separa la programación que da acceso a los datos en las bases de datos y aplicaciones desde el diseño y otros contenidos de la página Web. Esto ayuda a asegurar que los desarrolladores estén libres para enfocarse en escribir su lógica de negocios en componentes sin preocuparse acerca de cómo se muestra la salida. Recíprocamente, esto da libertad a los diseñadores de usar herramientas familiares para modificar la interfaz.

La **capa de servicios de presentación** es responsable de:

- Obtener información del usuario.
- Enviar la información del usuario a los servicios de negocios para su procesamiento.
- Recibir los resultados del procesamiento de los servicios de negocios.
- Presentar estos resultados al usuario.

**2. Capa de Negocios:** Los servicios de negocios son el "puente" entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Cuando los datos necesarios residen en un servidor de bases de datos, garantizan los servicios de datos indispensables para cumplir con la tarea de negocios o aplicar su regla. Esto aísla al usuario de la interacción directa con la base de datos.

Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como introducir una orden de compra o imprimir una lista de clientes. Las reglas de negocio (business rules) son políticas que controlan el flujo de las tareas.

Como las reglas de negocio tienden a cambiar más frecuentemente que las tareas específicas de negocios a las que dan soporte, son candidatos ideales para encapsularlas en componentes que están lógicamente separados de la lógica de la aplicación en sí.

El nivel de servicios de negocios es responsable de:

- ? Recibir la entrada del nivel de presentación.
  
- ? Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar (por ejemplo, la preparación de impuestos por ingresos, el procesamiento de ordenes y así sucesivamente).
  
- ? Enviar el resultado procesado al nivel de presentación.

3. **Capa de Datos:** El nivel de servicios de datos es responsable de:

- ? Almacenar los datos.
  
- ? Recuperar los datos.
  
- ? Mantener los datos.
  
- ? La integridad de los datos.

Los servicios de datos tienen una variedad de formas y tamaños, incluyendo los sistemas de administración de bases de datos relacionales (SABDs), servidores de correo electrónico y sistemas de archivos tales como el Sistema de Archivos NTFS.

Este modelo “Model View Controller” es utilizado dentro de este proyecto de Tesis para modularizar tanto la presentación del sistema, el control de Datos y la comunicación entre este control de datos y la presentación o interfaz del sistema. Esto quiere decir que desarrollan componentes para la interfaz, componentes para la lógica de información y componentes para la comunicación entre la interfaz y la lógica de información.

La capa llamada Model es la Lógica de información, es decir las clases Java, que en este proyecto serán todas aquellas clases encargadas de hacer transacciones con la base de datos; Un ejemplo claro son los servlets de las secciones, los artículos, los templates y noticias, los cuales se encargan de agregar, eliminar o modificar información en la base de datos.

La capa llamada View son todos los templates que utilizará este sistema para desplegar la información, es decir, es la interfaz gráfica, la cual consta de archivos HTML.

Finalmente la capa Controller, como ya se explicó anteriormente realiza la comunicación para presentar los datos dentro de una interfaz, en este caso la capa controller son los JSP del sistema, ya que estos además de contener código HTML, contienen código Java el cual se comunica con los Java Beans y traen la información requerida para mostrarse dentro de la interfaz.

## **II.2 Características del Sistema.**

Este sistema será un servidor de aplicación el cual utilizará la arquitectura de tres capas, que son la capa de presentación que llamaremos FrontEnd, la capa de negocios o Middle Tier y la capa de datos o BackEnd. Como se mencionó anteriormente el FrontEnd es la interfase gráfica del cliente que el servidor de aplicación envía al browser; mientras el Middle Tier es el conjunto de programas de aplicación que reside en el servidor. Y el BackEnd es el servidor de base de datos en donde se guarda toda la información utilizada por el middle Tier.

La metodología del flujo de información (request-response) que hay entre el cliente el servidor y el servidor de aplicación, que el request del cliente va primero al servidor web y este le envía la información al servidor de aplicación que envía de regreso la respuesta al servidor web después de tomar acciones apropiadas. Finalmente el servidor web envía la información procesada de regreso al cliente.

Esta aplicación utilizará la tecnología SUN, Java servlets y JSP, el protocolo HTTP y la salida o respuesta estará basada en el estándar de HTML4 y CSS nivel.

Esta aplicación entra en la categoría de servidores Web, ya que este tipo de servidores utiliza templates HTML y Scripts en las páginas de salida, incrustándoles información de una base de datos a la página. La ventaja de utilizar templates es si necesidad de cambiar algo en la configuración de aplicación se puede tener un aspecto totalmente diferente a un portal a otro, además que la información que el sistema escribe en

el template puede puesta de acuerdo al diseño gráfico del template, esto se podrá hacer gracias a que el sistema utilizara variables dentro del template y simplemente cambiando la ubicación de la variable, la información se despliegue donde se desea.

Para su funcionamiento esta aplicación requiere de una base de datos que puede ser:

?? Oracle.

?? Informix.

?? MS SQL Server.

?? Access.

Y las plataformas que soporta serán Windows y Unix que tengan instalado JDK, JRE y JSDK para utilizar el BackEnd, mientras que el FrontEnd puede ser visto con cualquier navegador.

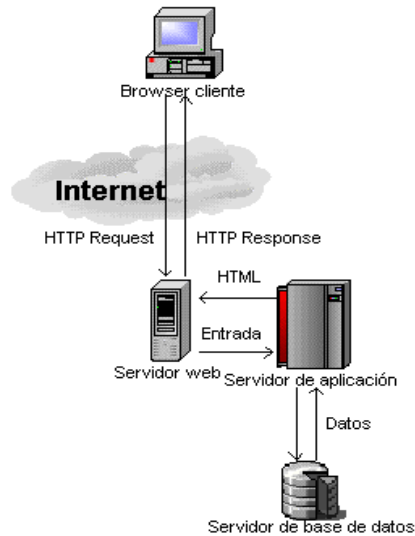


Figura 3. Diagrama de interacción del Sistema

Este sistema tiene un esquema de administración modular, es decir, existen varios módulos integrados y cada uno de ellos administrara cierta parte del portal. Gracias a este esquema se podrán agregar o eliminar módulos de acuerdo a las necesidades del cliente, permitiendo un fácil escalabilidad e implementación de nuevos módulos.

Los módulos que se desarrollaron para este sistema fueron:

- ?? Secciones: Crea el árbol de navegación de todas las secciones del portal, definiéndoles un template por sección y si la sección pertenece a otra sección o es una sección principal.
- ?? Artículos: En este modula se administra toda la información que se mostrara a los usuarios, un articulo podrá ser asociado a una o mas secciones del portal. Cada articulo tendrá un template relacionado de

la forma en que se quiere que se muestre dicha información, ya sea con campos de texto o imágenes.

?? Templates: Este modulo es el encargado de administrar las diferentes estructuras y formatos en los que se desplegaran los Artículos y secciones, también se definirán las imágenes utilizadas y la ubicación de cada template.

Para la construcción del sistema se utilizo el modelo de tres capas de Microsoft [Model View Controller Technology], el cual permite tener separados la presentación de los datos, el manejo de los datos y el lugar donde se almacenaran los datos. Este modelo nos permitirá tener un sistema modular y una de sus ventajas es que al eliminar o modificar la información solo se modificara la información de la capa correspondiente y no afectara a las demás módulos.

Además de utilizar este modelo de tres capas la construcción del software se realiza mediante módulos para lo cual nos permite agregar, eliminar o modificar módulos sin afectar a los demás módulos. Las tecnologías utilizadas para desarrollar el software fueron Java, servlets y JSP ya que son orientadas a objetos y nos permiten crear paginas en tiempo real y de manera dinámica sin cargar archivos en la maquina del cliente. El formato de presentación será el protocolo HTML, JavaScript y CSS2.