DESARROLLO ORIENTADO A LA WEB

AUTOR: WALTER MADRIGAL CHAVES

NOVIEMBRE: 2020

San Marcos

Contenido

Introducción

Al interior del desarrollo de software, una persona suele enfrentarse con problemas típicos de este medio. Estos por lo general suelen volverse más complejos si no existió una correcta planeación del proyecto a trabajar. En la mayoría de las ocasiones el no realizar un diseño adecuado de la o las bases de datos a utilizar, los diferentes módulos que tendrá el software y la ausencia de documentación de código, entre otros, tienden a desencadenar inconvenientes de gran magnitud y llegar a puntos donde irremediablemente ya no hay marcha atrás.

Aunque hoy el desarrollo web es catalogado como una manera eficiente de construir aplicaciones de gran impacto, por la sencillez de su interpretación en cuanto a definición y diseño, suele generar experiencias poco positivas, cuando no se tienen en cuenta elementos como los mencionados en el anterior párrafo.

Es importante entonces mencionar que las soluciones de software deberán planearse debidamente, haciendo uso de estrategias de desarrollo acordes a las necesidades del producto final y los procesos abordados.

Para esto, hoy existen distintos patrones que ayudan a establecer criterios puntuales para generar productos de gran calidad y eficiencia en la administración de la información, entre estos encontramos el MVC (Modelo Vista Controlador). A continuación, se describirán una serie de elementos que proporcionan información valiosa, para entender con mayor precisión, la filosofía del desarrollo orientado a la web.

Arquitectura básica de una aplicación web

Este se traduce en la construcción y mantenimiento de sitios web. Allí se realizan tareas que van desde tener una apariencia impecable, un buen desempeño y un funcionamiento ágil para facilitar a los usuarios la interacción con múltiples bancos de información, hasta la interconexión con bases de datos y la disposición de grandes volúmenes de información en tiempo real.

Los desarrolladores web hacen uso por lo general de uno o varios lenguajes de programación, dependiendo de la experiencia y de lo que se requiera para llevar a cabo el proyecto. El desarrollo web hace referencia a dos partes fundamentales en la generación de proyectos de software: existe el front end (la vista que va del lado del cliente) y el back end (la parte del servidor).

Una aplicación web funcionará siempre y cuando posea una arquitectura que tenga al menos los elementos que se describen a continuación:

- **Navegador:** muestra la definición del cliente-servidor, que hace solicitudes de recursos y componentes a distintos servidores de tipo web, haciendo uso de una URL.
- **Servidor:** guarda de manera eficiente y organizada la estructura de la información del correspondiente sitio web, con el fin de servir los contenidos en relación con las solicitudes que se realizan a través del navegador.
- **Protocolo HTTP:** se basa en el TCP/IP, mediante el cual el navegador puede realizar múltiples solicitudes al

Cliente-servidor

Expresión utilizada en el ámbito de la informática, en la que se llama cliente al dispositivo que requiere ciertos servicios de un servidor. La idea de servidor, por su parte, alude al equipo que brinda servicios a las computadoras que se hallan conectadas con él mediante una red.

Protocolo

Instrucciones, normativas o reglas que permiten guiar una acción o establecer ciertas bases para el desarrollo de un procedimiento.

TCP/IP

Denominación que permite identificar al grupo de protocolos de red que respaldan a internet y hacen posible la transferencia de datos entre redes de ordenadores.
En concreto, TCP/IP hace referencia a los dos protocolos más trascendentes de este grupo: Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP).

servidor para que sean respondidas por este.

- **HTML:** hace referencia al formato básico de los archivos que por lo general componen una página web. Se basa en el uso de etiquetas y se utiliza para construir la forma como se muestra el contenido de una página a los usuarios.

- **CSS:** se denominan también como hojas de estilo. Se utilizan para mejorar a nivel de diseño, colores y formas, los contenidos y elementos estructurados a través de componentes HTML.

Tecnologías del servidor y del cliente

Un sitio o aplicación web, además de agrupar los requisitos básicos para funcionar, puede crearse haciendo uso de distintas tecnologías que pueden dividirse en dos importantes categorías:

- **Tecnologías del cliente:** permiten y facilitan la creación de interfaces de usuario basadas en HTML, CSS y JavaScript, que facilitan el establecimiento de una comunicación con el servidor. Aquí, el navegador funciona como un intérprete.

- **Tecnologías de servidor:** facilitan la implementación de comportamientos de una aplicación web en el servidor. En este ámbito se hace uso de lenguajes de programación como Java, PHP, .NET, Python, Ruby on Rails, Django, Groovy, Node.js, entre otros.

Arquitecturas web de contenido dinámico y estático

Por lo general se menciona que los elementos mostrados en un sitio o página web hacen referencia a uno de tipo estático, cuando no es permitida la interacción del usuario. Por otro lado, los sitios que invitan a que el usuario

interactúe a través de distintos componentes como mapas, botones, formularios u otros elementos, se refieren a contenido de tipo dinámico.

Servidor y cliente estático: se realiza una solicitud en el servidor haciendo uso del navegador a través del protocolo HTTP, dicho servidor convierte la correspondiente URL en una ruta de disco y retorna el archivo al navegador, el cual lo renderiza mostrando con simplicidad la estructura del sitio en HTML y todo el contenido como videos y fotos al igual que su diseño gracias a los elementos del CSS.

Para este escenario no ha de existir interacción alguna utilizando JavaScript, ya que el servidor retorna los mismos recursos siempre, es decir, que la página web es totalmente estática.

Servidor dinámico y cliente estático: este modelo está compuesto por tres capas: el servidor web, una base de datos y el navegador. Cuando al servidor le llega una solicitud, retorna elementos contenidos en el disco o ejecuta el código que le permita generar el recurso de forma dinámica, ese código, por lo general, lleva a cabo consultas a la base de datos con el fin de recuperar información. De esta forma genera contenidos en el sitio web dinámicamente.

Servidor estático y cliente dinámico: este contenido se encuentra almacenado por lo general en el servidor de manera estática en su disco duro principal, aunque el cliente actúa de forma dinámica ya que las páginas utilizadas por él incluyen un código basado en JavaScript. Dicho código tiene la capacidad de realizar efectos y acciones de tipo gráfico como ocultar y mostrar información, y desplegar contenido interactivo adaptándolo a las necesidades visuales del usuario, entre otras tareas.

Servidor y cliente dinámico: se han de combinar los elementos de los dos casos anteriormente mencionados, ya que se hace uso de JavaScript para comportamientos interactivos y efectos gráficos al igual que para llevar a cabo

solicitudes en segundo plano (AJAX) y hacer uso de aplicaciones de página única con API Rest.

JavaScript es un lenguaje que puede utilizarse para evitar la recarga completa de la página al dar clic sobre un enlace, haciendo que se realicen peticiones al servidor web en un segundo plano (ocultas a la vista del usuario). Cuando la información resultante de la solicitud llega al navegador, JavaScript refresca exclusivamente las partes que se necesitan de la página en mención.

Esta técnica es denominada AJAX (Asynchronous JavaScript and XML), que permite mejorar la experiencia del usuario ya que, al llevar a cabo las solicitudes al servidor, este podría retornar partes de HTML que han sido generadas de forma dinámica, recursos estáticos en disco como mostrar errores, PDF, imágenes, información construida en JSON o XML, o simplemente cambiar colores.

Tecnologías del lado del cliente de tipo estándar

El W3C (World Wide Web Consortium) hace referencia a la comunidad internacional que elabora estándares abiertos, los cuales permiten asegurar que la web crezca a largo plazo. Allí puede encontrarse HTML5, CSS, AJAX, scripting, gráficos, protocolos de accesibilidad, video, audio, XML y web de tipo semántica, entre otros.

HTML (Hypertext Markup Languaje): brinda un conjunto de información organizada en párrafos, secciones, imágenes y títulos. Actualmente la versión difundida y trabajada es HTML5, la cual ofrece diferentes bibliotecas que permiten la inclusión de contenido multimedia, comunicaciones, canvas y elementos concurrentes.

CSS (cascading style sheets): se conoce como el encargado de distribuir los elementos y el correspondiente estilo como tipos de fuentes, colores y fondos, por mencionar algunos en archivos HTML, SVG y XML, así como interfaces de

usuario que hacen uso de diversas tecnologías.

Scripting: permite programar las páginas con diferentes scripts propios de diversos lenguajes. Sin embargo, por lo general es utilizado JavaScript, el cual va modificando la página, gracias a la capacidad que posee para ejecutar código al interactuar con ella.

Inicialmente JavaScript tenía características de lenguaje interpretado; sin embargo, en la actualidad también puede ejecutarse haciendo uso de máquinas virtuales en los navegadores, que aumentan la velocidad y eficiencia de ejecución de memoria. Este lenguaje de tipo dinámico está netamente basado en prototipos (orientado a objetos).

DOM (Document Object Model): modelo de objetos del documento el cual consta de una biblioteca de tipo API, que se utiliza en la administración del documento HTML el cual es cargado en el navegador, facilitando así la inserción y eliminación de elementos, y la gestión de eventos.

Tecnologías para desarrollo web de tipo no estándar

Durante algún tiempo, la ausencia de tecnologías abiertas para la realización de distintos eventos asociados a contenidos y comportamientos multimedia hizo que varias de las llamadas tecnologías propietarias lograran ocupar este espacio, por lo general, por iniciativa de organizaciones o fábricas de software. Dentro de las más conocidas se pueden mencionar:

- **Adobe Flash:** tecnología usada para insertar elementos de multimedia interactivo en las páginas web, la cual predominó bastante tiempo de forma gratuita para los usuarios, pero de tipo cerrado y propietario para quienes cumplían su labor como desarrolladores. Ellos debían costear la respectiva licencia para poder utilizarla.

- **Java Applets:** fueron los precursores de los elementos Flash, pero debido a que se presentaron múltiples prácticas anticompetitivas por parte de Sun Microsystems y Microsoft, se centraron fuertemente en la administración de servidores de aplicaciones, así que de igual forma como el anterior ítem, dejó de usarse.

- **Microsoft Silverlight:** la mayor apuesta de Microsoft con la cual buscó competir fuertemente con Adobe Flash; sin embargo, el soporte era demasiado limitado en entornos distintos a Windows.

Tecnologías de tipo servidor

Aunque las normas son relevantes en los navegadores web debido a la gran importancia que esta cobra en cuanto a compatibilidad con cualquier dispositivo, dichos estándares no suelen ser tan necesarios en el servidor, ya que cada institución configurará el suyo con el conjunto de componentes tecnológicos que más se ajuste a sus propias necesidades.

Por lo general, en el servidor se puede hacer uso de tecnologías abiertas o propietarias para la generación de aplicaciones orientadas a la web. En la actualidad existen varias de ellas, entre las que más se utilizan están Java EE, PHP y componentes .NET, y dentro de las que menos se utilizan se pueden mencionar Grails (Groovy), Ruby on Rails, Perl, Django (Phyton) y ColdFusion, entre otras. A continuación, se describen algunas de ellas:

PHP: tecnología que tiene su propio lenguaje. Fue construida por PHP Group con uso de licencia abierta. Es la tecnología del lado del servidor a través de la cual se han logrado implementar más servidores en la nube a nivel mundial. Esta es multiplataforma y logra integrarse con facilidad con MySQL y Apache en ambientes Linux, gracias a un paquete de distintas herramientas conocido como LAMP.

Java EE: conjunto de componentes que se basan en Java, desarrollados por una asociación de instituciones lideradas por IBM, Oracle y Red Hat. Es bastante usada a nivel empresarial, ya que la mayoría de las herramientas para desarrollo son software de tipo libre, además, existen organizaciones de empresas y desarrolladores que se dedican a realizar complementos y en general recursos compatibles.

ASP.NET: allí puede encontrarse una versión mejorada del ASP clásico, la cual se encuentra integrada en la tecnología .NET, al lado de lenguajes como C# o Visual Basic. Posee una licencia propietaria que es de uso exclusivo para plataformas Windows, además cuenta con una gran asociación de desarrolladores que, aunque es más reducida que la de otras alternativas, genera gran cantidad de contenido para la generación de soluciones de software.

Bases de datos

Las bases de datos más utilizadas y populares en el desarrollo de soluciones de software de tipo web son aquellas de tipo relacional. Aunque en el mercado se encuentran muchas bases de datos de este tipo, tanto para software libre como propietario, unas de las más utilizadas son Oracle, MS SQL Server, MySQL, Derby y PostgreSQL, entre otras. Las aplicaciones web deben tener características de tolerancia a fallos y escalabilidad, por esto, hoy se viene trabajando en un nuevo conjunto de bases de datos que se denominan NoSQL, aquí pueden mencionarse: Cassandra, Riak, y Redis.

Sistemas gestores de contenido

Desde hace ya largo tiempo, se vienen imponiendo los sistemas que gestionan contenidos, también conocidos como CMS (content management systems), los cuales hacen referencia a conjuntos de aplicaciones tipo web configuradas y

prediseñadas en principio para la creación y por ende la administración de contenidos en línea.

Los CMS hacen uso de varias de las tecnologías que se han mencionado con anterioridad, que han evolucionado para llegar a convertirse en el nuevo modelo de desarrollo web, adaptando y configurando módulos a través de una sencilla, pero completa interfaz de usuario en ambiente web. De igual forma, dicho sistema permite administrar independiente y autónomamente el diseño y el contenido, facilitando la modificación de su estructura a través de temas o plantillas.

En el mercado, se pueden encontrar gran cantidad de CMS con objetivos y enfoques distintos. Entre estos puede mencionarse: Plone (JavaScript), Moodle (PHP), Drupal (PHP), WordPress (PHP), PrestaShop (PHP), Liferay (Java) y Joomla (PHP).

Servlet

Puede definirse como una tecnología que permite crear aplicaciones web dinámicas, es decir, programas que le facilitan al usuario interactuar con la aplicación de forma eficiente y sencilla. El interactuar hace referencia propiamente a tareas como consultas, inserción y eliminación de datos.

De una forma más técnica, un servlet es un pequeño programa que ha sido escrito en Java y admite peticiones o solicitudes haciendo uso del protocolo HTTP. Por lo general un servlet recibe peticiones desde un navegador web, las procesa y retorna una respuesta a dicho navegador, por lo general en HTML.

Para lograr realizar estas tareas, es probable que haga uso de las clases incluidas en el propio lenguaje Java. Ese programa es el intermediario entre el cliente y la información almacenada en una base de datos

Ventajas

Dentro de las ventajas en el uso de los servlets se puede mencionar:

- **Eficiencia:** las solicitudes hechas por un cliente crean hilos y no nuevos procesos como solía ocurrir con los CGI (Common Gateway Interface) convencionales.
- **Potencia:** son desarrollados en Java, de allí que pueden emplearse todas las clases y en general herramientas disponibles para dicha plataforma.
- **Seguridad:** distintos elementos al interior de los servlets son controlados por la máquina virtual de Java. También es importante mencionar que la mayoría de los problemas de seguridad que se encuentran en los CGI, no se evidencian en los servlets.
- **Seguridad:** una ventaja bastante grande es que pueden ser utilizados sobre cualquier sistema operativo y en la mayoría de los servidores orientados a la web.
- **Precio:** todo el software requerido para el correcto funcionamiento de un servlet es gratis.
- **Elegancia:** el código Java puede trabajarse de forma modular u orientado a objetos, lo que lo hace bastante simple.

JSP (JAVA SERVER PAGES)

Como una traducción literal podría definirse como páginas de servidor Java. Esta es una tecnología que está orientada puntualmente a la creación de páginas web haciendo uso de la programación en Java. Al utilizar JSP, se pueden crear aplicaciones web que se ejecuten en múltiples servidores web, de

distintas plataformas, ya que como su código está basado en Java facilita la adaptación a esos entornos.

Las páginas JSP se componen de código HTML/XML combinado con etiquetas especiales que permiten elaborar scripts de servidor, obviamente usando sintaxis Java. Esto quiere decir que basta con un sencillo editor de texto para escribir las sentencias propias de JSP.

Ventajas

- El JSP dinámico está escrito en Java, lo que permite y facilita una integración total con módulos Java y hacer uso de un motor de páginas basado en servlets del mismo lenguaje. La escritura del código resulta de fácil lectura, comprensión y mantenimiento.

- Las JSP son más convenientes para aquellos desarrolladores que trabajan con Java. Aquellas personas que no están familiarizadas con los lenguajes de Microsoft pueden utilizar JSP para iniciarse en el desarrollo de páginas web de tipo dinámico. JSP les va a permitir utilizar funciones, sentencias para trabajo con bases de datos y en general código que normalmente se utiliza bajo lenguaje Java. Además, brinda amplia compatibilidad con servidores de internet que trabajen bajo Linux y bajo plataformas licenciadas.

- Con respecto a la parte dinámica, al no estar escrito en Visual Basic, en comparación por ejemplo con ASP, es más fácil de usar tanto en el ambiente de desarrollo, como en el espacio del usuario.

Desventajas

JSP es una tecnología que puede denominarse como antigua, por lo que actualmente no se usa tanto como otros lenguajes ya sean PHP o lenguajes basados en tecnología .NET. Debido a esto, los desarrolladores no crean

activamente complementos o nuevas bibliotecas para el idioma. Si una persona debe trabajar con lenguajes nuevos complementando su tarea con el uso de JSP, es posible que le genere dificultades al intentar encontrar las librerías necesarias para el idioma en cuestión.

Directivas JSP

Las directivas JSP permiten configurar cualquier información que se use en la página JSP. Por ejemplo: importar clases, incluir una página JSP en otra o definir una página de error. Existen dos tipos de directivas fundamentales en JSP: page e include.

Directiva page

Normalmente se utiliza para el establecimiento de ciertas propiedades a la página JSP. Hace uso de atributos como sesión, buffer, import, autoflush, isThreadSafe, erroPage, entre otros.

Directiva include

Permite realizar la inserción de contenido de otro archivo en una JSP.

MVC

MVC o Modelo-Vista-Controlador hace referencia a un patrón de arquitectura de software el cual hace uso de 3 componentes (modelos, vistas y controladores). Su objetivo principal es separar la lógica del negocio, de la lógica de la vista en una aplicación. Esta es una arquitectura bastante importante ya que se utiliza tanto en componentes gráficos básicos, como en sistemas robustos de tipo empresarial.

En la actualidad, la mayoría de los frameworks hacen uso de MVC (o por lo menos alguna adaptación de este) para su arquitectura, entre ellos puede mencionarse a Django, Ruby on Rails y AngularJS.

Modelos

Aquí, los objetos de modelo son los componentes que logran la implementación de toda la lógica del dominio de datos de la respectiva aplicación. Por lo general, los objetos de modelo recuperan y guardan el estado de dicho modelo dentro de una base de datos. Podría plantearse a manera de ejemplo que un objeto llamado product pudiese recuperar la información de una base de datos, administrarla y posteriormente escribir estos datos en una tabla x de una base de datos de cualquier motor.

En aplicaciones más pequeñas, el modelo por lo general hace referencia a una separación conceptual más que una física. Por ejemplo, si dicha aplicación solo toma una parte de los datos y los envía a la respectiva vista, el software no requiere tener un nivel de modelo físico ni las correspondientes clases que hacen parte del proyecto. Es decir que los datos seleccionados logran asumir el rol de un objeto del modelo en mención.

Vistas

Hacen referencia a los componentes que muestran la interfaz de usuario del correspondiente software. Por lo general, dicha interfaz se desarrolla a partir de los datos que están contenidos en el modelo. Por ejemplo, puede tenerse una vista en modo edición de una tabla de clientes, la cual muestra listas desplegables y cuadros de texto, al igual que casillas de selección, teniendo como referencia el estado que actualmente tenga un objeto tipo cliente.

Controladores

Son los componentes, que como su nombre indica, controlan la interacción del usuario, trabajan de la mano del modelo y finalmente toman una vista para generar la correspondiente interfaz del usuario. En las aplicaciones MVC, dicha vista se limita solo a mostrar información; el controlador se encarga de

administrar y responder a los datos que le proporciona el usuario y su interacción.

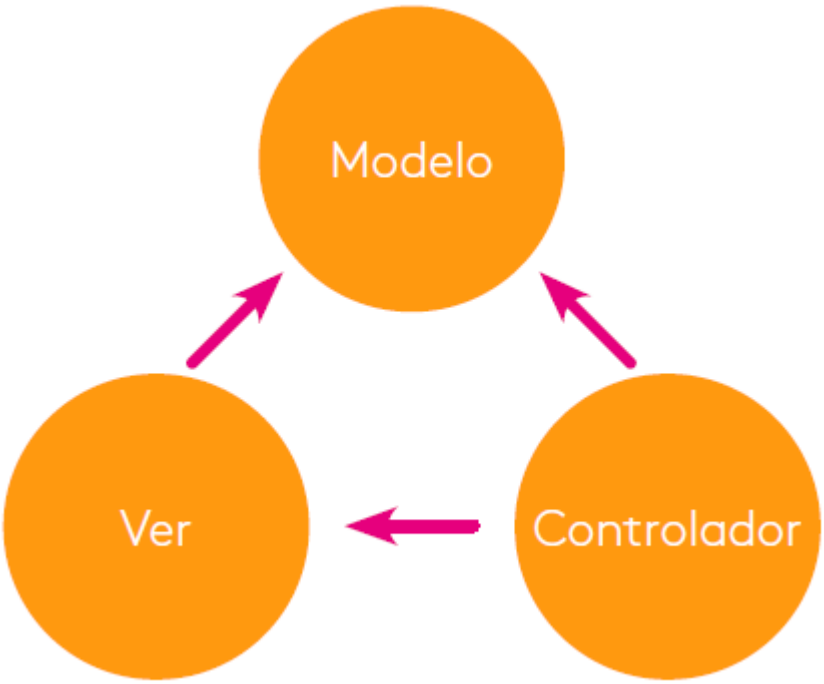
Para el caso, el controlador es quien gestiona los valores de la cadena de consulta y envía estos parámetros al respectivo modelo, el cual a su vez puede usarlos para realizar consultas a la respectiva base de datos.

El modelo MVC colabora en la creación de soluciones que separan distintos aspectos de esta (lógica de negocios, lógica de entrada y lógica para la interfaz de usuario), al mismo tiempo que brinda un ligero acople entre estos elementos. El modelo logra especificar dónde debe ubicarse cada lógica al interior del software. La lógica de la interfaz de usuario hace parte de la vista, la lógica de entrada hace parte del controlador y la lógica de negocios de igual forma hace parte del modelo.

Dicha separación facilita la administración de la complejidad cuando se compila una solución, ya que le permite centrarse en cada momento en solo un atributo de la implementación.

Para el caso, puede centrarse en la vista sin condicionarse por la lógica de negocios. El vago acoplamiento entre estos tres componentes que conforman una aplicación que usa MVC, de igual forma colabora en el favorecimiento del desarrollo en paralelo. Es decir, un desarrollador de software puede dedicarse a trabajar en la vista, otro desarrollador podría ocuparse de la lógica que tiene el controlador mientras que un tercero podría centrarse en la lógica de negocios para el modelo respectivo.

Imagen 1 Modelo de diseño MVC

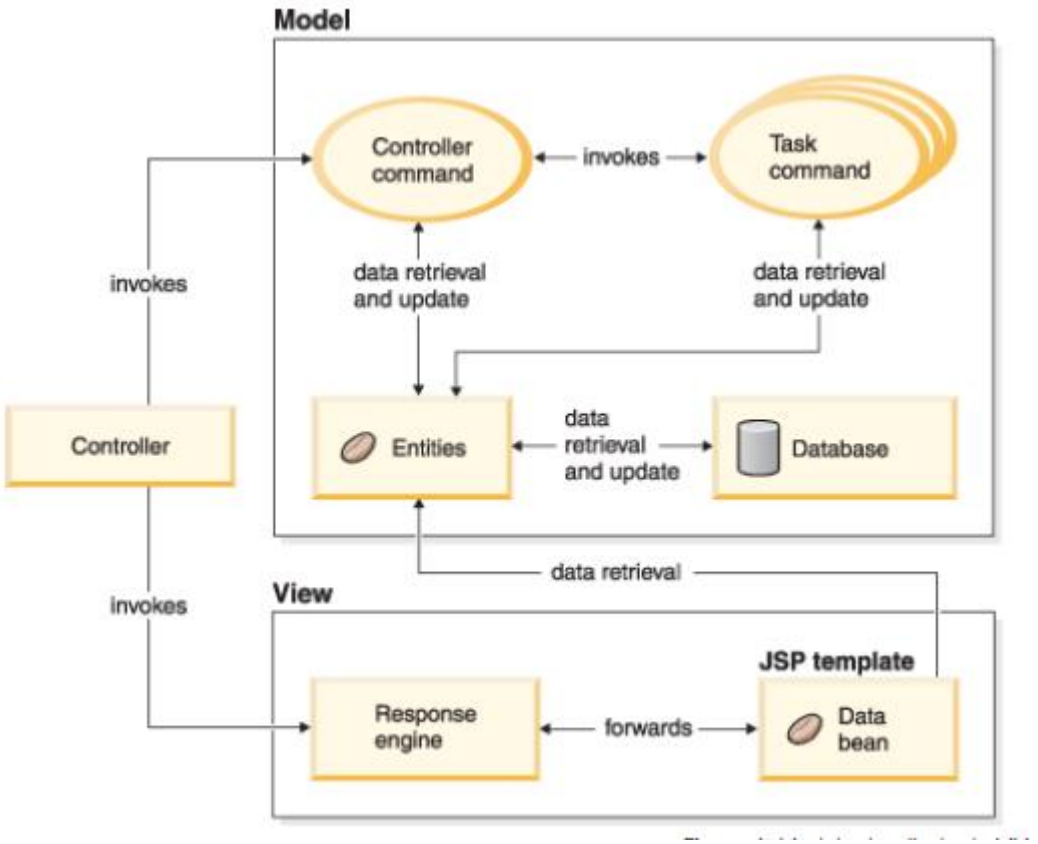


Fuente: https://bit.ly/1W74QHc

En la actualidad múltiples aplicaciones utilizan este patrón, con algunas variaciones. En algunos casos, las aplicaciones combinan el controlador y la vista en una clase, debido a que ya se encuentran estrechamente unidos. Independiente de la solución, siempre se recomienda establecer la separación de los datos de su correspondiente presentación. De esta manera se simplifica la estructura de una aplicación y además facilita la reutilización de código.

El siguiente diagrama muestra cómo se aplica el patrón de diseño MVC a WebSphere Commerce. Este patrón se utiliza para las aplicaciones web y para los Rich Client y puede utilizar la infraestructura de servicios web o struts.

Imagen 2 Modelo detallado de MVC



Fuente: https://ibm.co/2xCo83k

Ventajas y desventajas del uso del patrón

Las vistas deben mostrar información actualizada. El desarrollador no debe preocuparse de estar solicitando la actualización de las vistas, ya que dicho proceso lo realiza de forma automática el modelo implementado en la solución.

Cualquier modificación que de alguna u otra manera afecta al dominio, como el aumento de datos contenidos o métodos genera también una actualización en el respectivo modelo al igual que las interfaces de usuario con las respectivas vistas, no toda la estructura de comunicación y de actualizaciones entre modelos.

MVC ha logrado demostrar que es un patrón de diseño muy bien definido ya que las aplicaciones que lo implementan presentan una fácil escalabilidad al

igual que un mantenimiento único si se comparan con otras aplicaciones basadas en patrones diferentes.

En el desarrollo de una aplicación, teniendo como patrón de diseño MVC, se hace necesario dedicar un mayor tiempo a los momentos iniciales de esa labor. Por lo general, este patrón exige al desarrollador realizar un número mayor de clases que los requeridos en otros ambientes de desarrollo.

Aun así, esta desventaja suele ser relativa ya que más adelante, en la etapa del mantenimiento del software, la aplicación que incluye MVC suele ser más fácil de mantener, escalar y actualizar que una solución que no lo tiene.

MVC necesita que exista una primera arquitectura sobre la que se deben construir interfaces y por ende clases, para actualizar y así comunicar los distintos módulos de un software. Esta arquitectura inicial deberá incluir al menos un mecanismo de eventos para que se logren brindar las correspondientes alertas que genera el modelo de aplicación: una clase vista, otra clase modelo y una clase de tipo controlador que hagan las tareas de notificación, comunicación y modificación que posteriormente serán transparentes en el momento del desarrollo de la solución.

MVC hace referencia a un patrón de diseño netamente orientado a objetos, esto quiere decir que la implementación es bastante costosa y complicada en lenguajes que poco o nada trabajan con ese paradigma.

El lenguaje Java y el patrón MVC

Java es un lenguaje de programación que da gran soporte para la organización de MVC. Ofrece dos clases que se encargan de realizar las correspondientes notificaciones de los cambios en los estados de los respectivos objetos. Estos objetos son el observer y el observable:

- **Observer:** hace referencia a cualquier objeto que requiera ser notificado cuando el estado de otro elemento (objeto) diferente sea modificado.

- **Observable:** se puede definir como un objeto que tiene un estado que representa interés y en el cual otro u otros objetos han demostrado dicho interés.

Las dos clases anteriormente descritas no solo se utilizan en la aplicación de MVC, tienen una utilidad mayor dentro de Java. Suelen ser bastante útiles en cualquier sistema en el que se requiera que algunos objetos sean notificados en aquellos instantes en que ocurran modificaciones en otros objetos.

El modelo hace referencia a un subtipo de observable y la vista de igual forma es un subtipo de observer. Esto se traduce en que estas dos clases manejan de forma adecuada la notificación de los cambios que requiere MVC y brindan un mecanismo adecuado por el cual las vistas logran ser notificadas de forma automática por medio de las modificaciones producidas al interior del modelo.

Las referencias al objeto modelo tanto en la vista como en el controlador permiten el acceso a los datos de dicho objeto modelo.

¿De qué manera trabaja una aplicación basada en MVC?

Captura de la solicitud a través del controlador

La aplicación recibe solicitudes que se centralizan en el respectivo controlador. Este se encarga de interpretar, teniendo como referencia la URL de la petición y el tipo de operación que se debe realizar. Por lo general, esto se realiza analizando el valor de alguno de los parámetros enviados, agregando a la URL la solicitud, que es utilizado con dicha finalidad.

¿Cómo se procesa la petición?

Cuando el controlador determina la operación a realizar, procede a ejecutar las correspondientes acciones, llamando para ello a los distintos métodos expuestos por el modelo. Según las acciones a realizar (dar de alta a un usuario del sistema), el modelo requerirá manejar la información enviada por el cliente en la solicitud.

Estos datos le serán brindados por el respectivo controlador. Asimismo, los resultados obtenidos en el modelo (un ejemplo puede ser la información que resulta de una búsqueda) serán entregados sin intermediarios al controlador.

Para que sea más fácil este intercambio de datos entre el modelo y el controlador, y luego entre el controlador y la respectiva vista, las soluciones basadas en MVC suelen utilizar JavaBeans. Estos pueden entenderse como clases que encapsulan un gran conjunto de datos que contienen varios métodos de tipo set/get que proporcionan fácil acceso a estos desde el exterior.

Generación de respuestas

Los resultados que retorna el modelo al controlador se almacenan por lo general en una variable de solicitud, aplicación o sesión, dependiendo del alcance que requieran tener. Posteriormente, el controlador llama a la respectiva página JSP, la cual se debe encargar de generar la correspondiente vista. La página deberá acceder a la variable de ámbito donde se encuentren guardados los resultados y los deberá utilizar para generar de forma dinámica una respuesta XHTML, la cual deberá ser enviada al cliente.

Hasta aquí, es importante recordar el gran impacto que se genera en los productos de software al utilizar patrones y estrategias que permiten organizar cada uno de los componentes que interactuarán en la solución. No obstante, el éxito no se centra exclusivamente en el MVC, sino que depende de otros elementos que se irán abordando con el transcurso del presente módulo.

Conclusiones y recomendaciones

- Las aplicaciones web necesitan para su funcionamiento cinco elementos importantes como lo son: navegador, el servidor, el protocolo HTTP, HTML y CSS, con estos elementos el programador puede jugar seleccionado versiones e implementaciones.

- Las arquitecturas web permiten hacer que un sitio web sea dinámico o estático, la decisión de cual utilizar va de la mano del estudio detallado del problema o situación a resolver.

- Existen diferentes tecnologías, prácticas y modelos para crear aplicaciones y sistemas web, la decisión de cual tomar al realizar un proyecto depende de muchos factores, es importante involucrar a los solicitantes, ver políticas institucionales, analizar sus competencias personales o del equipo, revisar el entorno, antecedentes, mercado, entre otros.

- El desarrollo de soluciones Web mejora día con día gracias a entes como W3C (World Wide Web Consortium), que a base de estándares procura un crecimiento uniforme y basado en buenas prácticas, como accesibilidad a persona con discapacidad, mejora en el posicionamiento de motores de búsqueda, multiplaforma de navegadores, multiplataforma de equipos, entre otros.

- Dentro de la gama de opciones para desarrollar ambientes web, se debe apuntalar conocimientos en una nueva tendencia llamada sistemas de gestión de contenidos. La facilidad de crear sistemas algunas veces sin conocimientos previos los hace los predilectos del mercado. Claro tienen sus inconvenientes, tal vez el más marcado es la poca flexibilidad que se tiene al desarrollar.

- El patrón MVC divide los elementos del software en tres capas Modelo, Vista y Controlador. Esto trae una gran cantidad de beneficios, como: simplicidad en el mantenimiento de los sistemas, sistemas escalables, reutilización de componentes, aprovechamiento de recursos, entre otros.

Referencias bibliográficas

Azpitarte, R., et ál. (2009). Introducción a la programación orientada a objetos con Java. Valencia, España: Editorial Universidad Politécnica de Valencia.

Deite, P., y Deitel, H. (2008). Java. Cómo programar. Recuperado de https://docs.google.com/file/d/0B1znxPN5ACgSbHhVU1RaVFh0cVU/

Morales, R. (2014). Lenguajes de programación; ¿qué son y papa qué sirven? Recuperado de https://colombiadigital.net/actualidad/ articulos-informativos/item/7669-lenguajes-de-programacion-queson-y-para-que-sirven.html

Universidad de Alicante, Departamento de Ciencia de la Computación e Inteligencia Artificial. (2014). Introducción a Java Server Faces. Recuperado de http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.pdf

San Marcos

www.usanmarcos.ac.cr

San José, Costa Rica