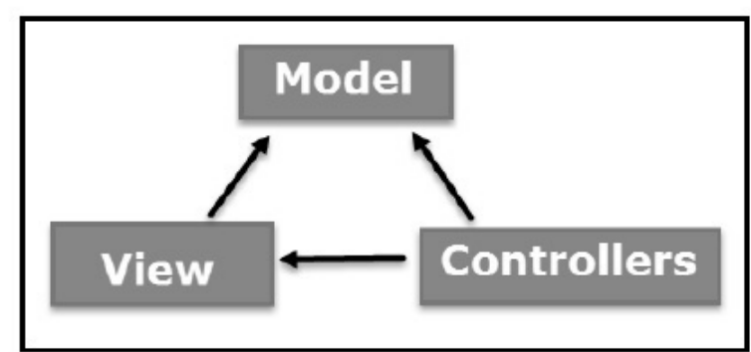
**MODELO MVC - Clase 9**

El Model-View-Controller (MVC) es un patrón arquitectónico que separa una aplicación en tres componentes lógicos principales: el modelo, la vista y el controlador. Cada uno de estos componentes está diseñado para manejar aspectos de desarrollo específicos de una aplicación. MVC es uno de los marcos de desarrollo web estándar de la industria más utilizados para crear proyectos escalables y extensibles.

**Componentes MVC**

Los siguientes son los componentes de MVC:



**Modelo**

El componente Modelo corresponde a toda la lógica relacionada con los datos con la que trabaja el usuario. Esto puede representar los datos que se transfieren entre los componentes View y Controller o cualquier otro dato relacionado con la lógica empresarial. Por ejemplo, un objeto Cliente recuperará la información del cliente de la base de datos, la manipulará y actualizará los datos en la base de datos o la utilizará para representar datos.

**Vista**

El componente Ver se utiliza para toda la lógica de la interfaz de usuario de la aplicación. Por ejemplo, la vista Cliente incluirá todos los componentes de la interfaz de usuario, como cuadros de texto, menús desplegables, etc., con los que interactúa el usuario final.

**Controlador**

Los controladores actúan como una interfaz entre los componentes Model y View para procesar toda la lógica comercial y las solicitudes entrantes, manipular datos utilizando el componente Model e interactuar con las Vistas para generar el resultado final. Por ejemplo, el controlador del Cliente manejará todas las interacciones y entradas de la Vista del Cliente y actualizará la base de datos usando el Modelo del Cliente. El mismo controlador se utilizará para ver los datos del Cliente.

**ASP.NET MVC**

ASP.NET admite tres modelos principales de desarrollo: páginas web, formularios web y MVC (controlador de vista de modelo). El marco ASP.NET MVC es un marco de presentación liviano y altamente comprobable que está integrado con las características existentes de ASP.NET, como páginas maestras, autenticación, etc. Dentro de .NET, este marco se define en el ensamblado System.Web.Mvc. La última versión de MVC Framework es 5.0. Usamos Visual Studio para crear aplicaciones ASP.NET MVC que se pueden agregar como plantilla en Visual Studio.

**Características de ASP.NET MVC**

ASP.NET MVC proporciona las siguientes características:

* Ideal para desarrollar aplicaciones complejas pero ligeras.
* Proporciona un marco extensible y conectable, que se puede reemplazar y personalizar fácilmente. Por ejemplo, si no desea utilizar el Razor integrado o el motor de visualización ASPX, puede utilizar cualquier otro motor de visualización de terceros o incluso personalizar los existentes.
* Utiliza el diseño basado en componentes de la aplicación al dividirlo lógicamente en componentes de modelo, vista y controlador. Esto permite a los desarrolladores gestionar la complejidad de los proyectos a gran escala y trabajar en componentes individuales.
* La estructura MVC mejora el desarrollo basado en pruebas y la capacidad de prueba de la aplicación, ya que todos los componentes pueden diseñarse en función de la interfaz y probarse utilizando objetos simulados. Por lo tanto, ASP.NET MVC Framework es ideal para proyectos con un gran equipo de desarrolladores web.
* Admite todas las amplias funcionalidades existentes de ASP.NET, como autorización y autenticación, páginas maestras, enlace de datos, controles de usuario, membresías, enrutamiento ASP.NET, etc.
* No utiliza el concepto de View State (que está presente en ASP.NET). Esto ayuda a crear aplicaciones, que son livianas y brindan control total a los desarrolladores.

Por lo tanto, puede considerar MVC Framework como un marco importante construido sobre ASP.NET que proporciona un gran conjunto de funcionalidades adicionales que se centran en el desarrollo y las pruebas basadas en componentes.

**AJAX**

La mejor tecnología para construir páginas web dinámicas es Ajax . El código JavaScript incrustado en la página HTML se utiliza para enviar solicitudes al servidor. En el lado del servidor, se requiere algún procesamiento para manejarlos, encontrar la información o almacenar los datos. Para hacer eso, necesitamos un marco especializado. El marco siempre tiene una parte de JavaScript y, a veces, una parte del lado del servidor en otro lenguaje de secuencias de comandos. Muchos de ellos existen en varios lenguajes de programación, en todos los entornos, pero retenemos aquí solo los más utilizados.

**¿Por qué un marco?**

En realidad, este marco es el motor Ajax descrito por JJ Garrett y destinado a suprimir la espera del usuario al acceder al servidor.

El marco proporciona funciones clásicas entre navegadores para usar el objeto XMLHttpRequest. Pero un marco puede ir más allá y permitir crear "aplicaciones web ricas", aplicaciones con una interfaz gráfica de usuario y otras funciones de software de escritorio que se ejecutan a través de un navegador, mientras intercambian datos con un servidor remoto.

**Características de un framework Ajax**

Cualquier framework Ajax es capaz de comunicarse con el servidor, y así, leer datos o enviarle datos o comandos. En el último caso, se requiere un script del lado del servidor.

Los marcos a menudo agregan componentes que hacen uso de la comunicación asíncrona con el servidor.

Los ejemplos clásicos son botones, paneles con pestañas, cuadrículas, cuadros de lista y otros widgets similares.

Un ejemplo más innovador, las "cajas", se implementan cada vez más, y Lightbox y Slimbox son dos de ellas. Hay galerías de imágenes que las colocan una al lado de la otra en la pantalla y que están haciendo uso de Ajax para mostrarlas instantáneamente.

El marco también puede ser controlado por el servidor y, en este caso, los componentes se crean en el servidor con un lenguaje de secuencias de comandos como PHP y se envían al navegador. Ajax se usa para transmitir las acciones del usuario a la parte del servidor y para manejar los resultados.

La capacidad de trabajar sin conexión que ofrece HTML 5 también es un complemento del marco Ajax.

**¿Qué framework Ajax elegir?**

Desea utilizar un marco Ajax para crear páginas dinámicas sin tener que reescribir todo el material usted mismo y probarlo en todos los navegadores...

Pero son tantos que se pregunta cuál usar. ¡No querrás hacer una mala elección y verte obligado a rehacer todas las páginas con un marco diferente!

Es menos probable que te arrepientas de la elección de un marco si adoptas uno de los más utilizados porque se complementan continuamente con nuevas extensiones y tienen un buen soporte por parte de los usuarios.

Entre estos, hay jQuery, bien documentado, Mootools que es modular, Ext JS que ofrece muchos widgets para RIA.

La interacción con un lenguaje del lado del servidor también debería guiar la elección.

**Ejercicios:**

* ¿En qué casos se aplica AJAX conocidamente?

AJAX se aplica para construir páginas web dinámicas que mejoran la experiencia del usuario al permitir la actualización de contenido sin recargar toda la página. Es útil en casos donde se requiere la interacción asíncrona con el servidor, como formularios de búsqueda en vivo, carga de contenido dinámico (como comentarios o secciones de noticias), validación de formularios, y la implementación de aplicaciones web ricas (RIA) con interfaces de usuario avanzadas, como galerías de imágenes interactivas (Lightbox, Slimbox) y widgets.

* Desarrolle brevemente MVC

El patrón arquitectónico **MVC (Model-View-Controller)** separa una aplicación en tres componentes principales:

* **Modelo**: Gestiona la lógica de los datos de la aplicación. Incluye la recuperación, manipulación y actualización de los datos, como la interacción con la base de datos.
* **Vista**: Representa la interfaz de usuario y muestra la información del modelo. Incluye elementos con los que el usuario interactúa, como cuadros de texto y botones.
* **Controlador**: Actúa como intermediario entre el modelo y la vista. Procesa las solicitudes del usuario desde la vista, realiza las operaciones necesarias sobre el modelo y actualiza la vista con el resultado.