

DISEÑO ORIENTADO A OBJETOS

Prof: Esp. Ing. Agustín Fernandez

Patrón MVC

- Patrón Modelo Vista Controlador (MVC)
- **Un poco de historia:** Trygve Mikkjel Heyerdahl Reenskaug (nacido en 1930): científico de la computación y profesor emérito de Noruega, de la Universidad de Oslo. El formuló el patrón modelo-vista-controlador (MVC) para el diseño de interfaz gráfica de usuario (GUI) de software en 1979 durante su visita al Centro de Investigación de Palo Alto de Xerox. Su primer proyecto de software importante, "Autokon" produce un exitoso programa de CAD/CAM, que luego se utilizó en los astilleros de todo el mundo durante más de 30 años.



Patrón MVC

- **Un poco de historia:** “MVC se concibió como una solución general al problema de los usuarios que controlan un conjunto de datos grandes y complejos. La parte más difícil fue dar con buenos nombres para los diferentes componentes de la arquitectura. Model-View-Editor fue el primer set que se nos ocurrió después de largas discusiones, en particular con Adele Goldberg, terminamos con los términos Modelo-Vista-Controlador”

[Trygve Reenskaug]

- Sitio personal: <http://heim.ifi.uio.no/~trygver/>



Patrón MVC

- ¿Qué es MVC?:

- MVC es un patrón que considera dividir una aplicación en tres **módulos** claramente identificables y con funcionalidad bien definida: El **Modelo**, las **Vistas** y el **Controlador**.
- MVC Es un **patrón de arquitectura** de las aplicaciones software que separa la lógica de negocio de la interfaz de usuario y facilita la evolución por separado de ambos aspectos incrementado **reutilización y flexibilidad**.
- Buschmann en Pattern Oriented Software Architecture (1996) ubica al patrón MVC dentro de la categoría de **Sistemas Interactivos**.
- Fowler en Pattern of Enterprise Application Architecture (2003) ubica al patrón MVC dentro de la categoría de **Patrones de Presentación Web**.

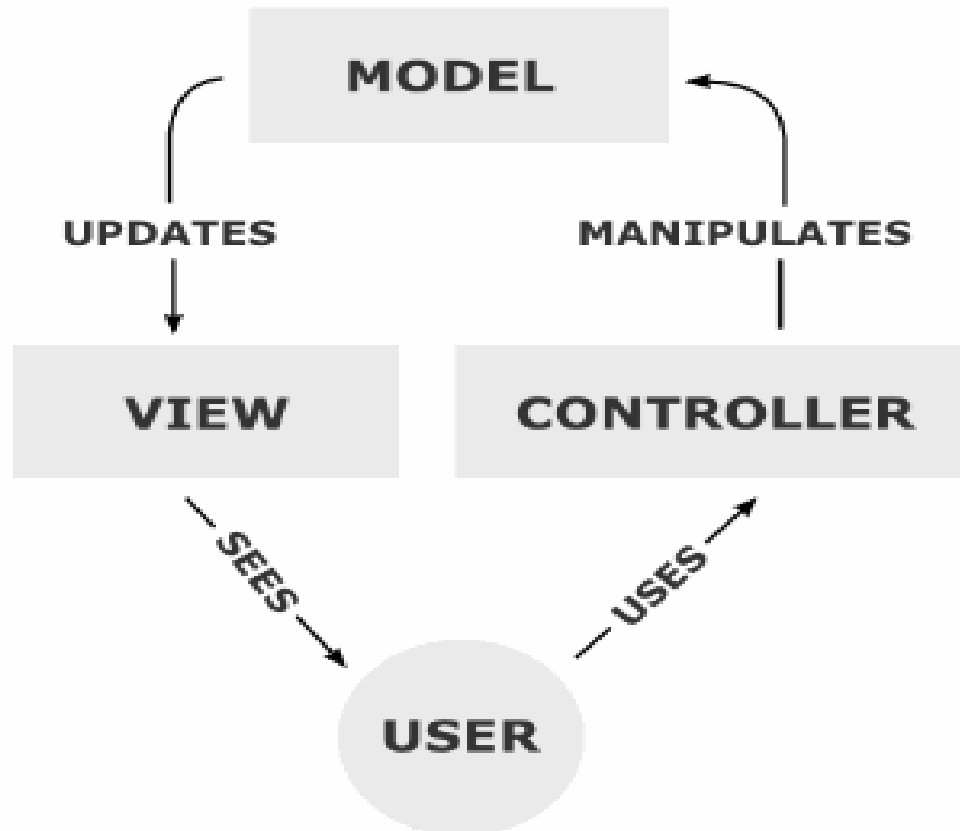
Patrón MVC

- **Utilizado en multiples frameworks:**

- Java Swing
- JME
- JavaFX
- Java Server Faces
- XForms (Formato XML estándar del W3C para la especificación de un modelo de proceso de datos XML e interfaces de usuario como formularios web)
- GTK+ (escrito en C, toolkit creado por Gnome para construir aplicaciones gráficas, inicialmente para el sistema X Window)
- ASP.NET MVC Framework (Microsoft)
- Google Web Toolkit (GWT, para crear aplicaciones Ajax con Java)
- Apache Struts (framework para aplicaciones web)
- Ruby on Rails (framework para aplicaciones web con Ruby)
- Etc, etc, etc

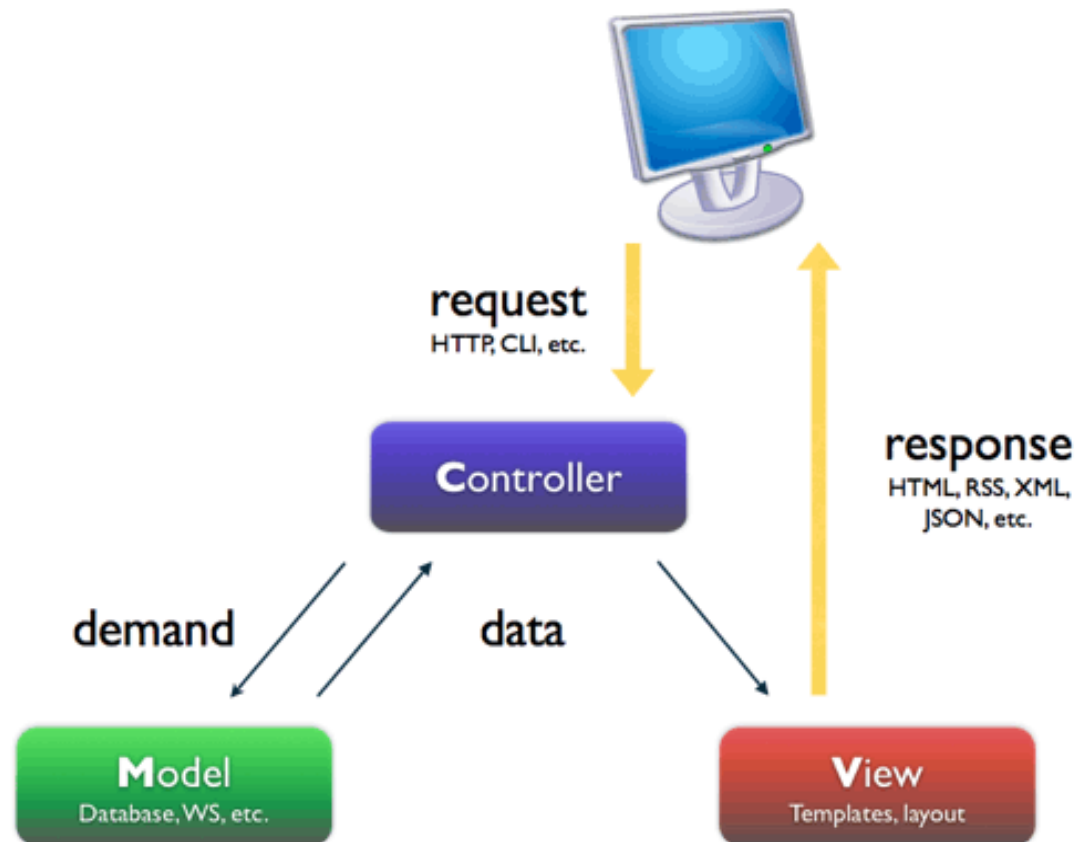
Patrón MVC

- Típicamente:



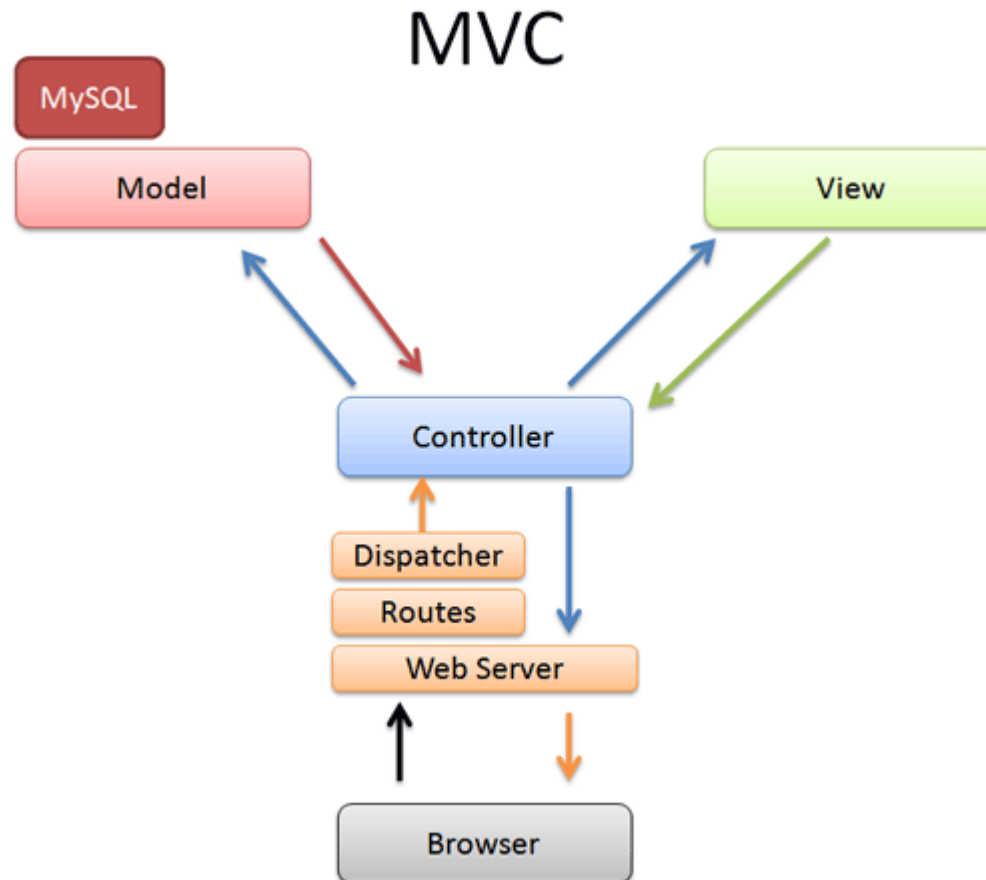
Patrón MVC

- ¿Qué diferencias hay con el anterior?



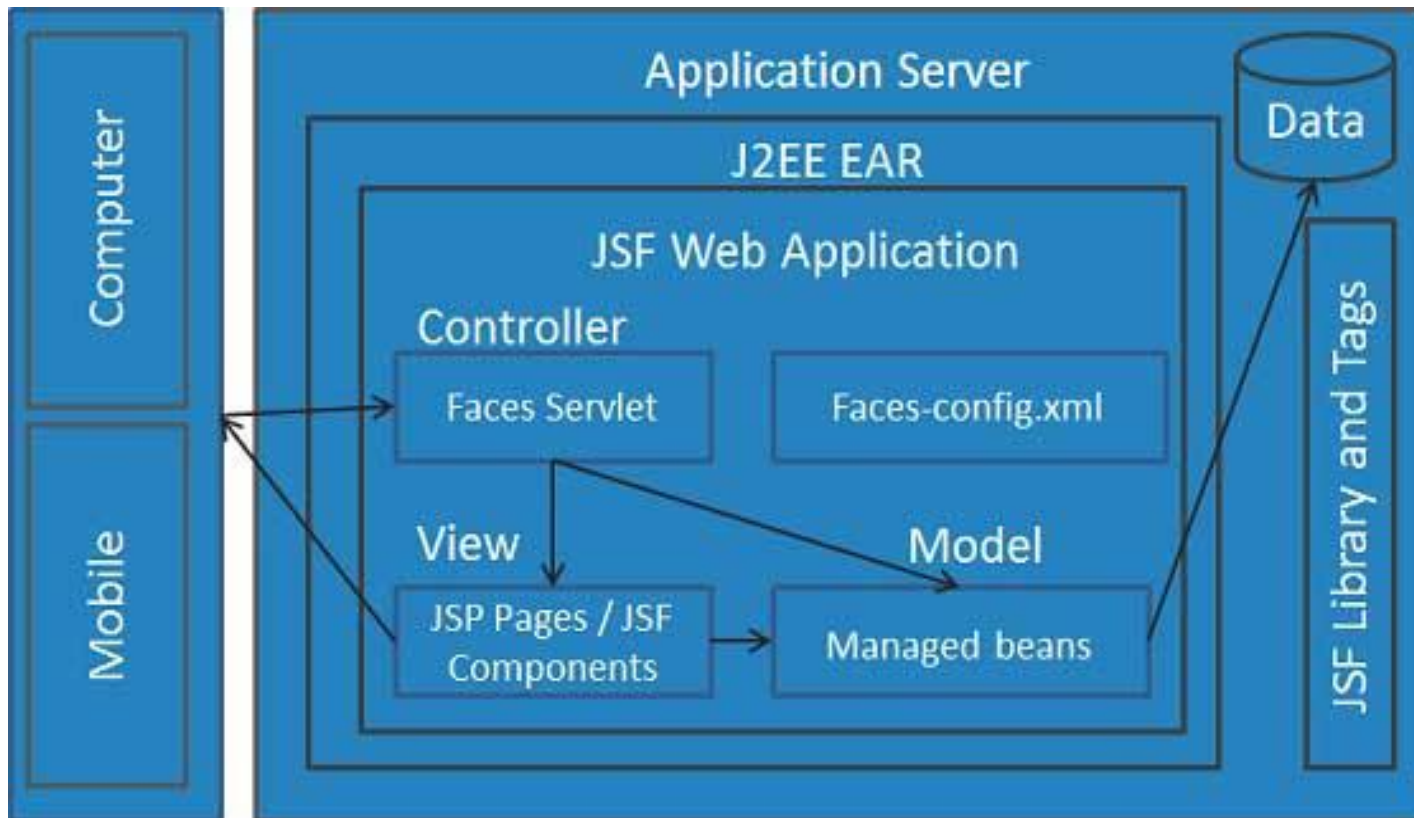
Patrón MVC

- ¿Y con este otro?



Patrón MVC

- Framework web MVC (Java Server Faces) basado en este patrón:



Patrón MVC

- **El modelo:**

- El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar sin tener en cuenta **ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo.**
- Idealmente el modelo desconoce la existencia de las vistas y del controlador. Pero en la práctica deberán existir interfaces que permitan a los módulos comunicarse entre sí.
 - Según la descripción inicial de SmallTalk el modelo se descompone en 2 submódulos:
 - **El modelo del dominio.**
 - **El modelo de la aplicación.**
 - Esta descomposición es la base de la implementación de la librería Swing de Java.

Patrón MVC

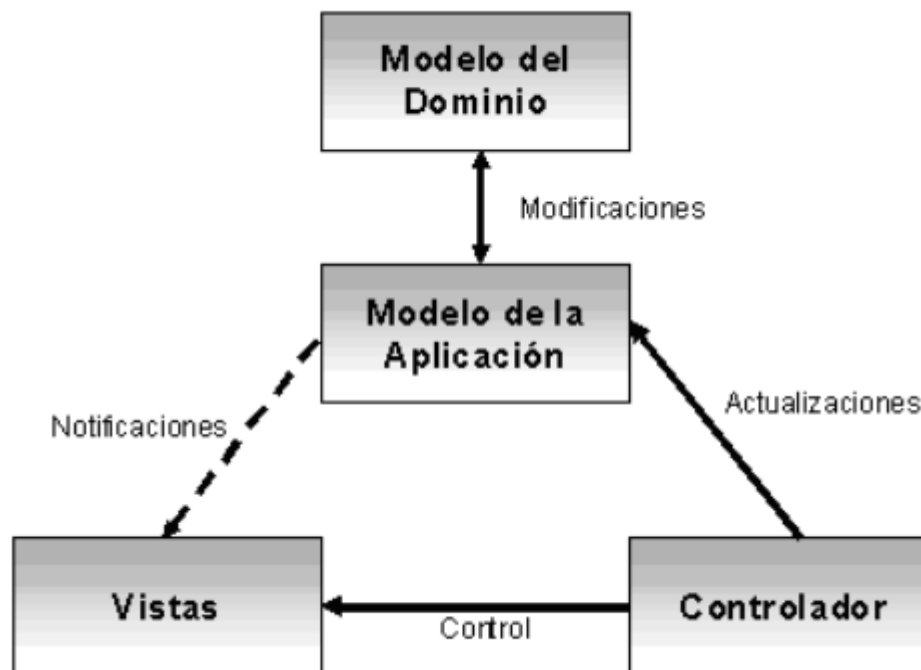
- **El modelo del dominio:** es el conjunto de clases que un ingeniero modela al analizar el problema que desea resolver; por ejemplo: el cliente, la factura, la temperatura, la hora, etc. **El modelo del dominio no debe tener relación con nada externo a la información que contiene.**
- **El modelo de la aplicación:** es un conjunto de clases que se relacionan con el modelo del dominio, que tienen conocimiento de las vistas y que implementan los mecanismos necesarios para notificar a estas ultimas sobre los cambios que se pudieren dar en el modelo del dominio.

Patrón MVC

- **Las vistas:** son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo. Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación.
- **El controlador:** es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario. A partir de estos mensajes, el controlador se encarga de modificar el modelo (o de abrir y cerrar vistas). El controlador puede tener acceso al modelo y a las vistas, pero las vistas y el modelo no deberían conocer de la existencia del controlador.

Patrón MVC

- Entonces según Smalltalk la estructura queda:



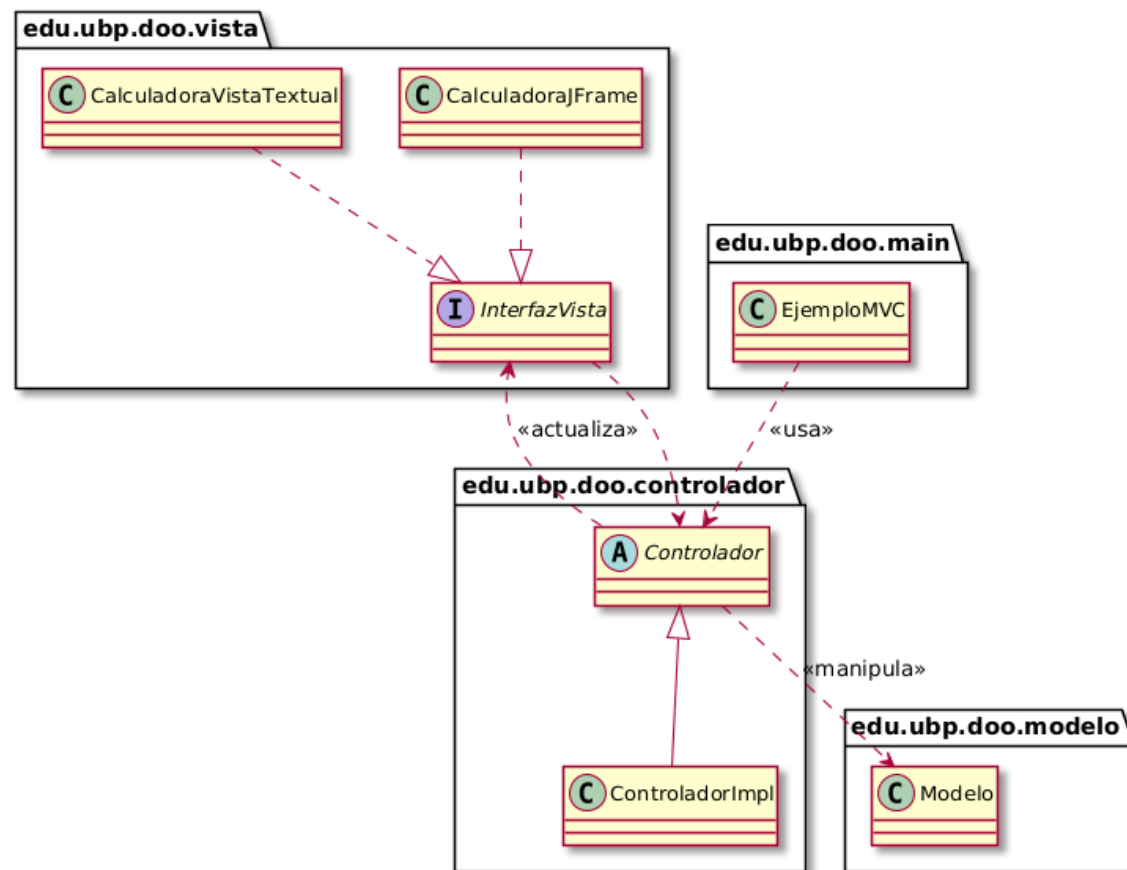
Patrón MVC

- **Flujo de control típico:**

- El usuario realiza una acción en la interfaz
- El controlador trata el evento de entrada
- El controlador notifica al modelo la acción del usuario, lo que puede implicar un cambio del estado del modelo
- Se «genera» una nueva vista. La vista toma los datos del modelo
 - El modelo no tiene conocimiento directo de la vista
- La interfaz de usuario espera otra interacción del usuario, que comenzará otro nuevo ciclo

Patrón MVC

- Ejemplo calculador para sumar y restar 2 números:
- Proyecto NetBeans: ejemploMVC



Patrón MVC

- **Ejemplo calculador para sumar y restar 2 números:**
 - Si tenemos otra vista llamada CalculadoraVistaTextual
 - ¿Qué hay que cambiar en el modelo y el control para utilizar la vista textual en vez de la vista swing?
 - ¿Qué hay que cambiar en el programa principal?