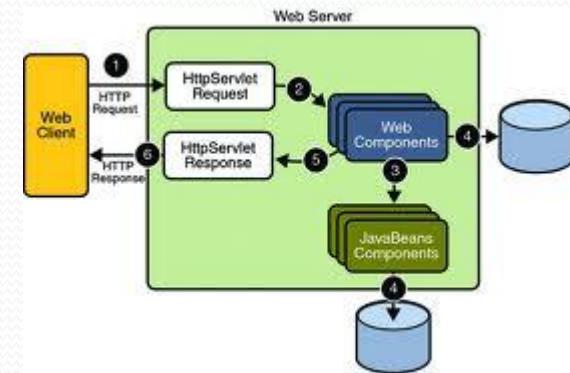


# MVC

**Expositor:** Ing. Wilmer Richard Perfecto Manrique

# Agenda

- INTRODUCCION
- PATRONES DE DISEÑO
- PATRON MVC
- CONCLUSIONES



# INTRODUCCION



# Introducción



# Patrones de Diseño



# Patrones de Diseño

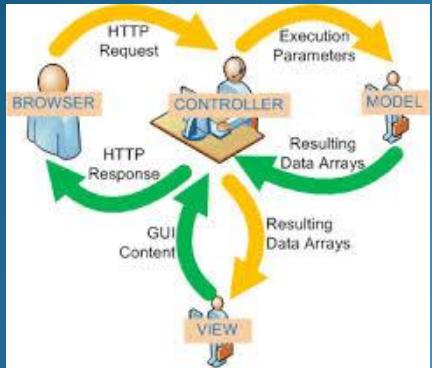
- Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos.
- Los patrones de diseño son soluciones basadas en la experiencia y que se ha demostrado que funcionan.
- Con los patrones de diseño el desarrollo es mas ordenado, mas rápido, mas fácil de entender.
- Los patrones de diseño, brindan una solución ya probada y documentada a problemas de desarrollo de software con contextos similares.

# Patrones de Diseño

- Los elementos de un patrón de diseño son:
  - Nombre
  - El problema (cuando aplicar un patrón)
  - La solución (descripción abstracta)
  - Las consecuencias (costos y beneficios)

# Patrones de Diseño

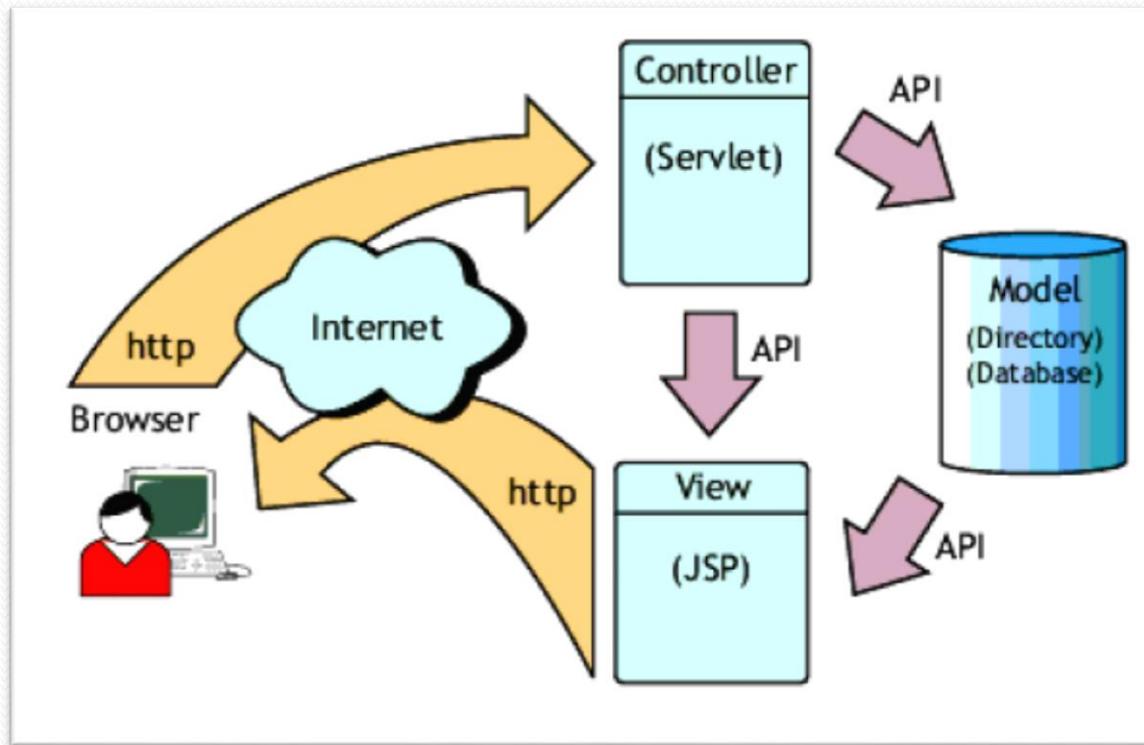
- Clasificación de los Patrones
  - **Patrones Creacionales:** Inicialización y configuración de objetos.(MVC)
  - **Patrones Estructurales:** Separan la interfaz de la implementación. Se ocupan de
    - cómo las clases y objetos se agrupan, para formar estructuras más grandes.
  - **Patrones de Comportamiento:** Más que describir objetos o clases, describen la comunicación entre ellos.



# Patrón MVC



# Patrón MVC



# Patrón MVC

- Existen tres responsabilidades principales:
  - La **lógica de la aplicación** (qué hacer):
  - Los **datos** sobre los que se opera (sobre qué datos se opera y cómo se organizan esto).
  - La **presentación de la aplicación** (como se expone a sus usuarios, pueden ser otras aplicaciones).

# Patrón MVC (Relación entre..)

- **El controlador**

- El controlador es el que dirige la orquesta de la aplicación
  - Este define que se ha de hacer y cuando
  - Como se ha de responder ante un evento de la interfaz gráfica
  - Como se ha de responder ante un mensaje por la red
  - Como se han de producir los resultados
  - Que vistas se han de usar

# Patrón MVC

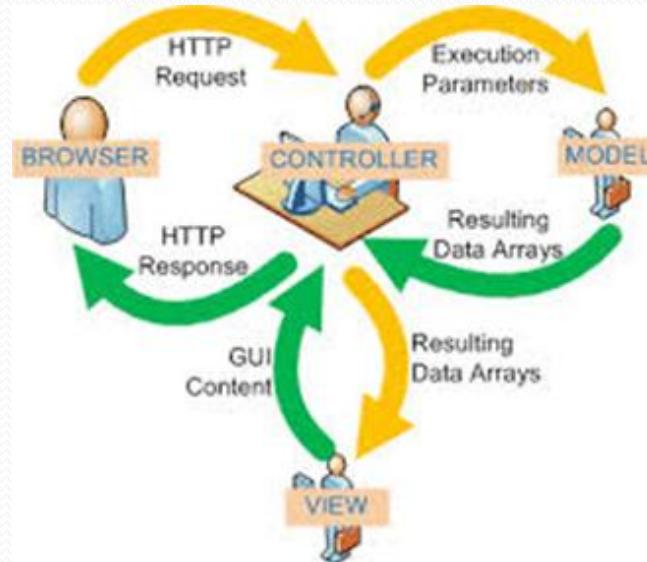
- El controlador
  - El controlador conoce tanto al modelo de datos subyacente como a la vista que se está usando
    - Necesita que hace, dado que tiene que orquestarlos.
    - Es posible que no conozca directamente a la vista misma, sino que sólo a una interfaz de esta, de tal manera de pueda cambiarla sin problemas
  - Recibe notificaciones por parte de la vista sobre cambios que han ocurrido.
    - Un usuario que hace click en un botón
    - La vista notifica (evento) al controlador, y es éste el que toma la acción que corresponda, potencialmente actualizando al modelo o a la vista.

# Patrón MVC

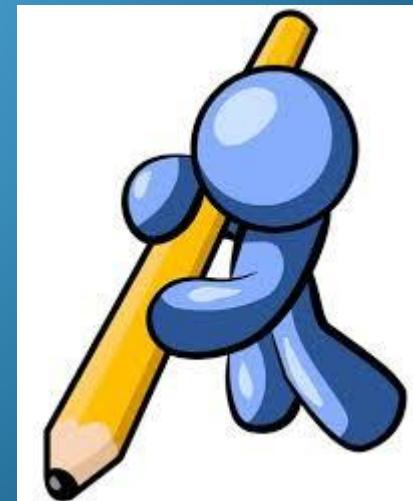
- El modelo
  - El modelo es la parte de la aplicación que representa los datos sobre los cuales se va a operar.
  - El modelo es actualizado/modificado por el controlador.
  - Es capaz de levantar notificaciones cuando sus datos cambian, de manera de que, quien sea que lo esté observando, se entere de los cambios
  - El modelo no conoce ni al controlador ni a la vista
  - El mismo modelo puede ser compartido por aplicaciones diferentes.

# Patrón MVC

- La vista
  - La vista es la frontera de comunicación con el mundo exterior.
  - La vista está expuesta al usuario.



# Conclusiones



# Conclusiones

- Patrones de Diseño:
  - Se debe utilizar patrones de diseño para obtener un desarrollo mas ordenado, mas rápido y mas fácil de entender
  - Se debe utilizar patrones de diseño para obtener soluciones simples y elegantes a problemas específicos, y comunes del diseño orientado a objetos.
  - Se debe utilizar patrones de diseño para obtener soluciones basadas en la experiencia
- Patrón MVC:
  - Todas las responsabilidades son diferentes e independientes
  - El patrón MVC nos dice cómo separarlas
  - El controlador se encarga de la lógica.
  - El modelo se encarga de los datos(Es capaz de avisar cuando cambia)
  - La vista se encarga de la interacción con el usuario



**Muchas Gracias..**

**Ing. Wilmer Richard Perfecto Manrique**