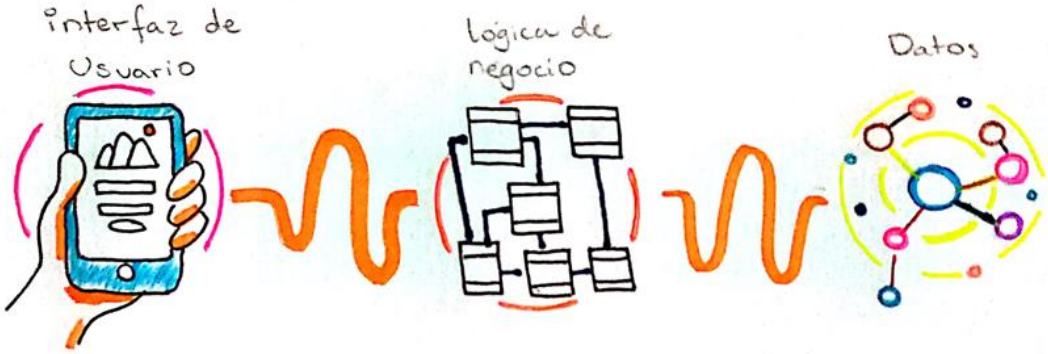
Capitulo 4:

Patrones de diseño.

En el area de programación se pueden desarrollar ciertos patrones para resolver problemas relacionados entre :



Los patrones mas utilizados para resolver estos problemas son:

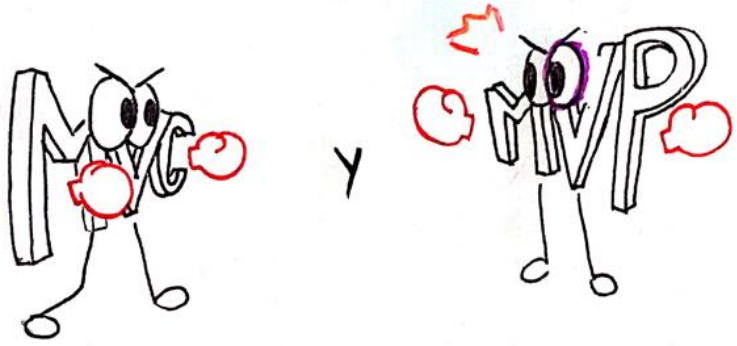
MVC y **MVP**

(Modelo Vista Controlador) (Modelo Vista Presentador)

Beneficios de su uso :

- facilita el desarrollo de una aplicacion
- facilita su mantenimiento
- Código reutilizable
- Cada una de las capas tiene un uso especifico.
- Estos patrones nos permite encaminar a una solución ya antes diseñada.

A continuación se realizara una comparación entre estos modelos.



MVC (modelo Vista controlador)

Conocido por la comunidad de desarrolladores

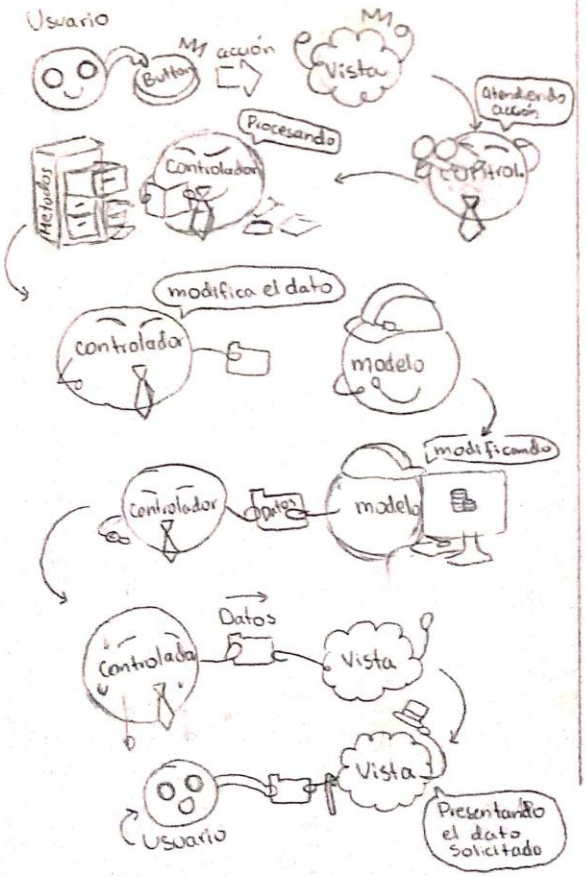
Se utilizan 3 capas:

Modelo: Aquí se define la estructura de datos con los que se trabajaran, además del almacenamiento y persistencia de datos.

Vistas: Podremos encontrar la interfaz de usuario.

Controlador: Capa intermediaria, entre Vista y modelo, responde eventos, capturandolos por la interaccion del usuario en la interfaz, para procesarlo en la peticion y solicitar datos para mostrarlo en interfaz.

Este modelo actua de la siguiente manera:



MVP (modelo Vista presentador)

Este patrón toda la logica de prestación de la interfaz reside en el "Presentador"

Modelo: Almacena los datos y encontramos la logica de negocio de la app, realizando peticiones al servidor para actualizar datos y entregarlo al presentador

Vista: Representando por pantalla los datos

Presentador: Intermediario entre modelo y vista.

COMPARANDO

MVC

- El controlador puede tener multiples vistas para mostrar datos
- Contiene la lógica de negocio del sistema y es el intermediario entre modelo y vista.

MVP

- Cada vista tiene asignada su presentador responsable de gestionar su lógica de presentación.
- Se encarga de la lógica representación de la vista y es el intermediario entre el modelo y vista

• Patron Observer

Este patrón se basa en dos objetos :

Observables

Son capaces de informar a los Subscriptores al observable y que deseen ser notificados sobre cambios de estado de esos Objetos.

Observadores

Son objetos que se suscriben a los Objetos observables y que solicitan ser notificados cuando el estado el estado de estos observables Cambien.