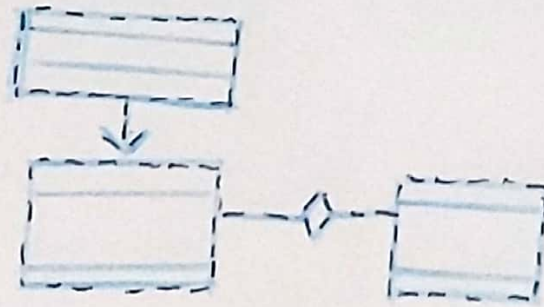


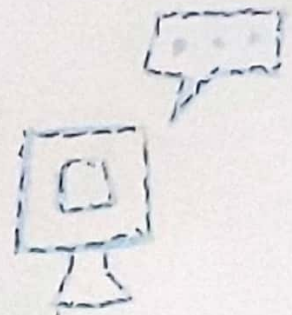
# ¿Qué son los Patrones de Diseño?

Los Patrones de diseño tratan de resolver los problemas relacionados con la interacción entre interfaz de usuario, lógica de negocio y los datos.



## Características

- Proporcionar catálogos de elementos reusables en el diseño.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño



# MVC (Modelo Vista Controlador)

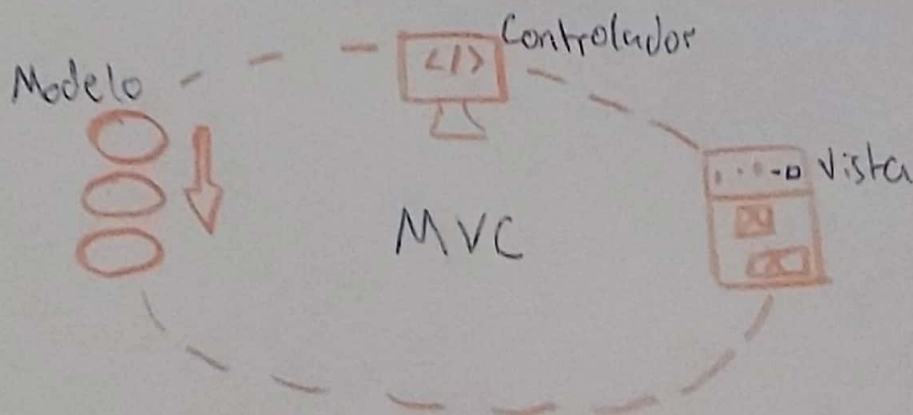
## Concepto MVC

Es un patrón de diseño que se estructura mediante tres componentes: Modelo, vista y controlador. Este patrón tiene como principio que cada uno de los componentes está separado en diferentes objetos.

Modelo: Esta capa contiene el conjunto de clases

Vista: Esta capa contiene la interfaz de usuario

Controlador: Esta capa es la intermedia entre la vista y el Modelo.





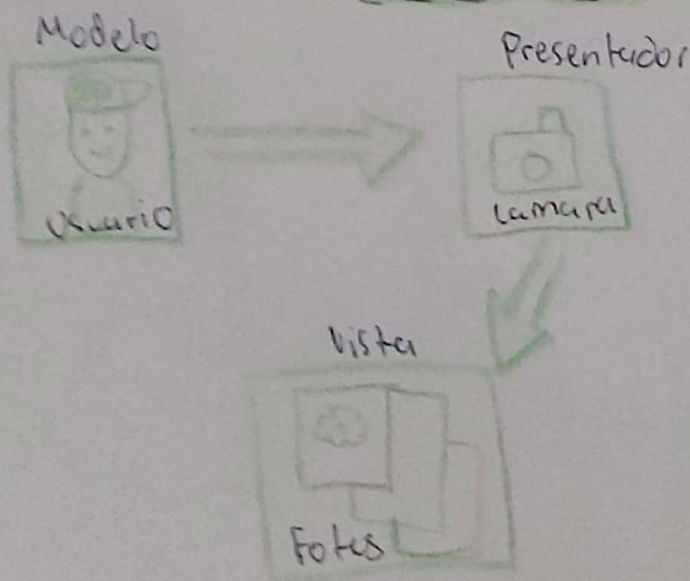
# MVP (Modelo Vista Presentador)

Es una derivación del patrón arquitectónico MVC y es utilizado para construir interfaces de usuarios.

Modelo: Esta capa contiene las claves

Vista: Esta capa contiene la interfaz de usuario

Presentador: permite conectar la interfaz gráfica con los datos

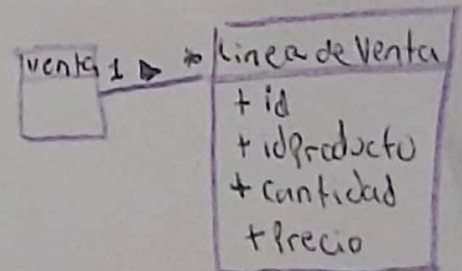


# GRASP (Patrones Generales de Software Para Asignar Responsabilidades)

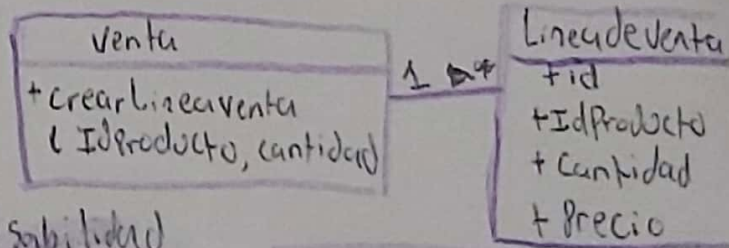
Los patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos.



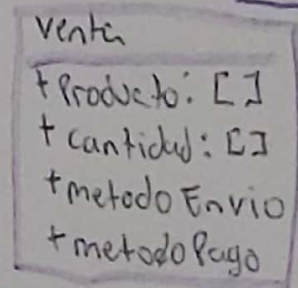
**Experto:** se encarga de asignar una responsabilidad al experto en información



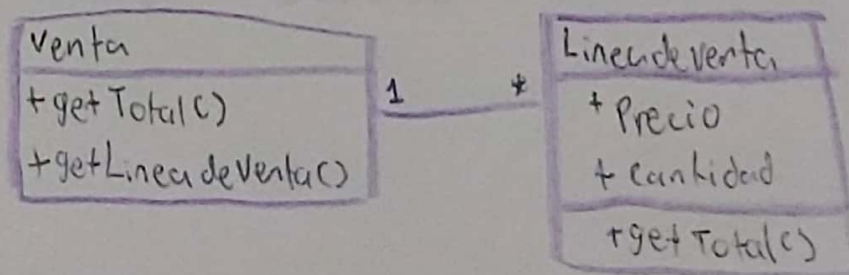
**Creador:** Este patrón implica que un objeto debe responsabilizarse de crear otros.



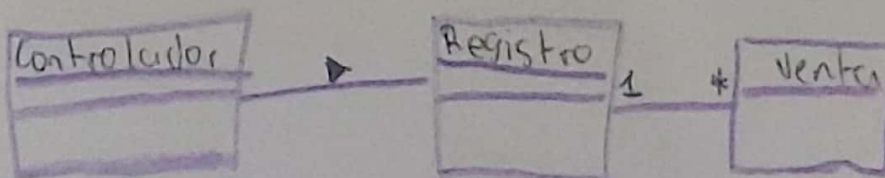
**Alta Cohesión:** Asigna una responsabilidad de forma tal que la cohesión siga siendo alta



**Bajo acoplamiento:** Se refiere a cuánto relación tiene los objetos entre sí.



**Controlador:** Es un patrón por el cual definimos objetos llamados controladores que independizan las interfaces con las acciones que haya que hacer





# GOF (Gang of Four)

Se clasifican en 3 categorías basadas en Propósito: Creacionales, estructurales y de comportamiento

## Creacionales

Fábrica abstracta, Constructor virtual, Método de Fabricación, Prototipo, Instancia única.

## Estructurales

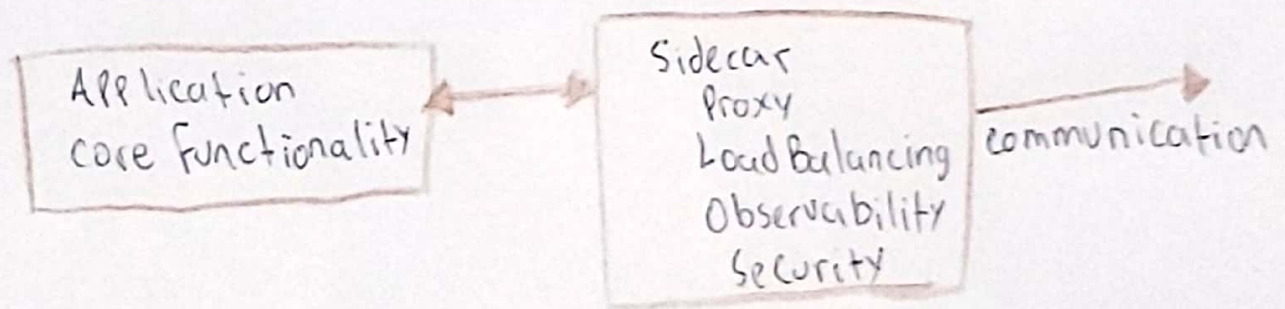
Adaptador, Puente, Objeto compuesto, Decorador, Fachada, Reso ligero, Proxy, Módulo.

## Comportamiento

Cadena de responsabilidad, Orden, Intérprete, Iterador, Mediador, Recuerdo, observador, Estado, Estrategia, Método Plantilla.

# SideCar

En este esquema, el componente de la aplicación se implementa en un proceso o contenedor separado. Esto ayuda a lograr la abstracción y la encapsulación.



## Cuando Usar este Patrón

- Cuando se trata de microservicios numerosos.
- Cuando se trata de aplicaciones heredadas.

## Cuando No Usar este Patrón

- Cuando se trata de un número limitado de servicios que necesitan comunicarse.
- Aplicaciones pequeñas.