

# Qt clase 3



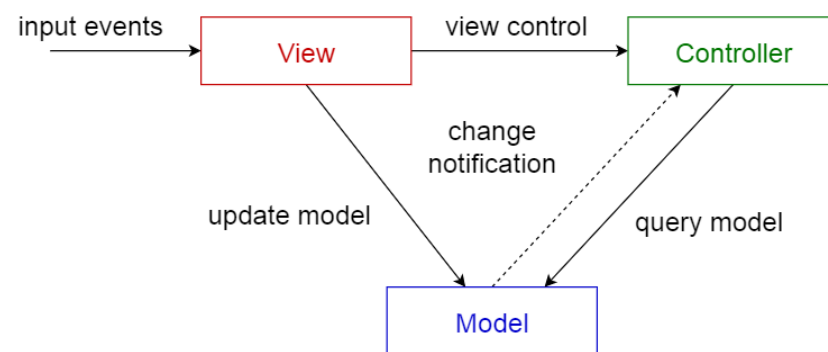


# Arquitectura de Software

- *Un **patrón arquitectónico** es una solución general y reutilizable a un problema común en la arquitectura de software dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de software pero tienen un alcance más amplio.*
- *Existen variadas arquitecturas*
  - 1. Patrón de capas
  - 2. Patrón cliente-servidor
  - 3. Patrón maestro-esclavo
  - 4. Patrón de filtro de tubería
  - 5. Patrón de intermediario
  - 6. Patrón de igual a igual
  - 7. Patrón de bus de evento
  - 8. Modelo-vista-controlador**
  - 9. Patrón de pizarra
  - 10. Patrón de intérprete

# Patrón de modelo-vista-controlador

- Este patrón, también conocido como patrón MVC, divide una aplicación interactiva en 3 partes, como
  - **modelo** — contiene la funcionalidad y los datos básicos
  - **vista** : muestra la información al usuario (se puede definir más de una vista)
  - **controlador** : maneja la entrada del usuario
- Esto se hace para separar las representaciones internas de información de las formas en que se presenta y acepta la información del usuario. Desacopla los componentes y permite la reutilización eficiente del código.
- **Uso**
  - Arquitectura para aplicaciones World Wide Web en los principales lenguajes de programación.
  - Marcos web como [Django](#) y [Rails](#) .





# El controlador es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener\_tiempo\_de\_entrega ( nueva\_orden\_de\_venta )".

