

# **Estructura de Requisitos y Patrones de Diseño**

Presentación sobre  
conceptos clave en diseño  
de software

# Fases para el Desarrollo software.

- 1) Requerimientos → Funcionales  
No Funcionales
- 2) Diseño
- 3) Implementación / Codificación
- 4) Pruebas
- 5) Despliegue.

¿Cómo?



Patrones  
Diseño

¿Qué?

# Estructura de Requisitos

- **Requisitos funcionales:** Definen las funciones específicas del sistema.
- **Requisitos no funcionales:** Atributos de calidad como rendimiento, seguridad.
- **Requisitos del sistema:** Capacidades globales y restricciones tecnológicas.

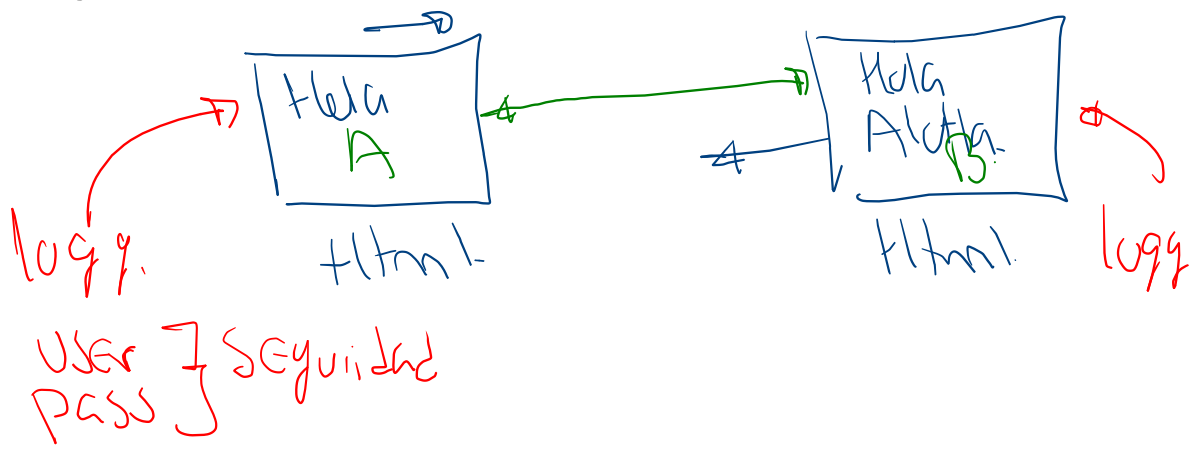
## Metodologías comunes:

- UML (Unified Modeling Language)
- User stories y casos de uso → SCUM

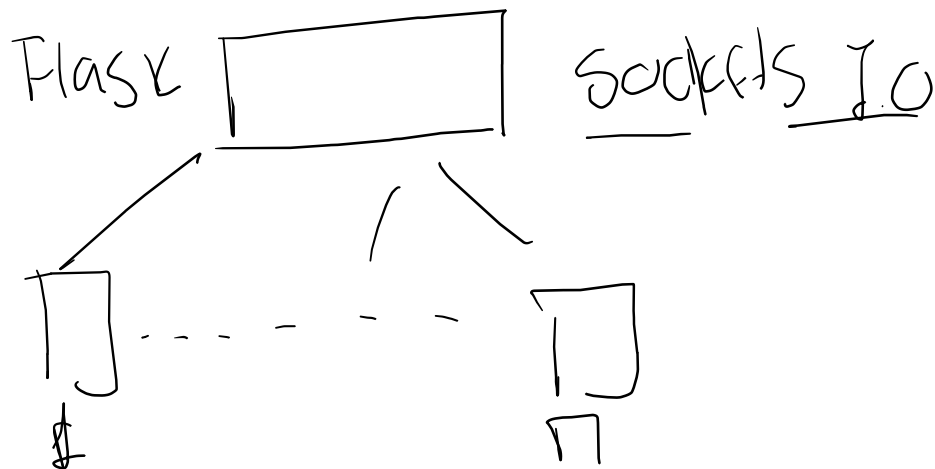
Ejemplo.

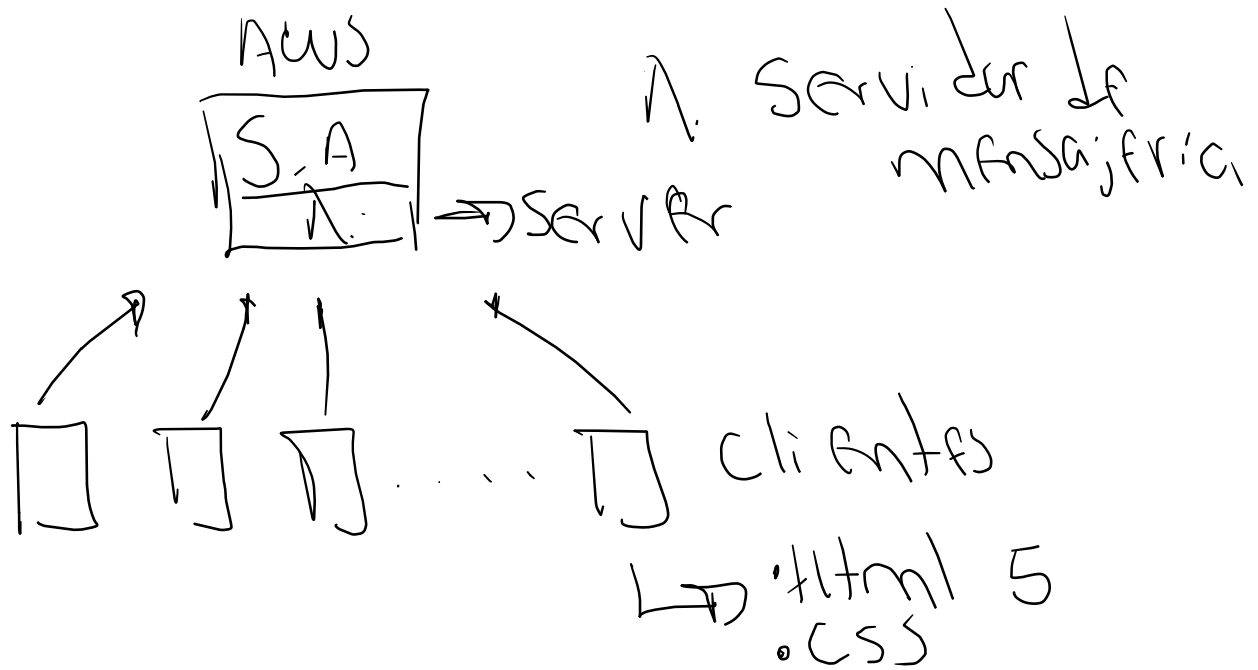
- Chat 2 Equipos. o más } Funcionales.
- modelo cliente servidor
  - TCP.
  - HTML.

¿cuantos?



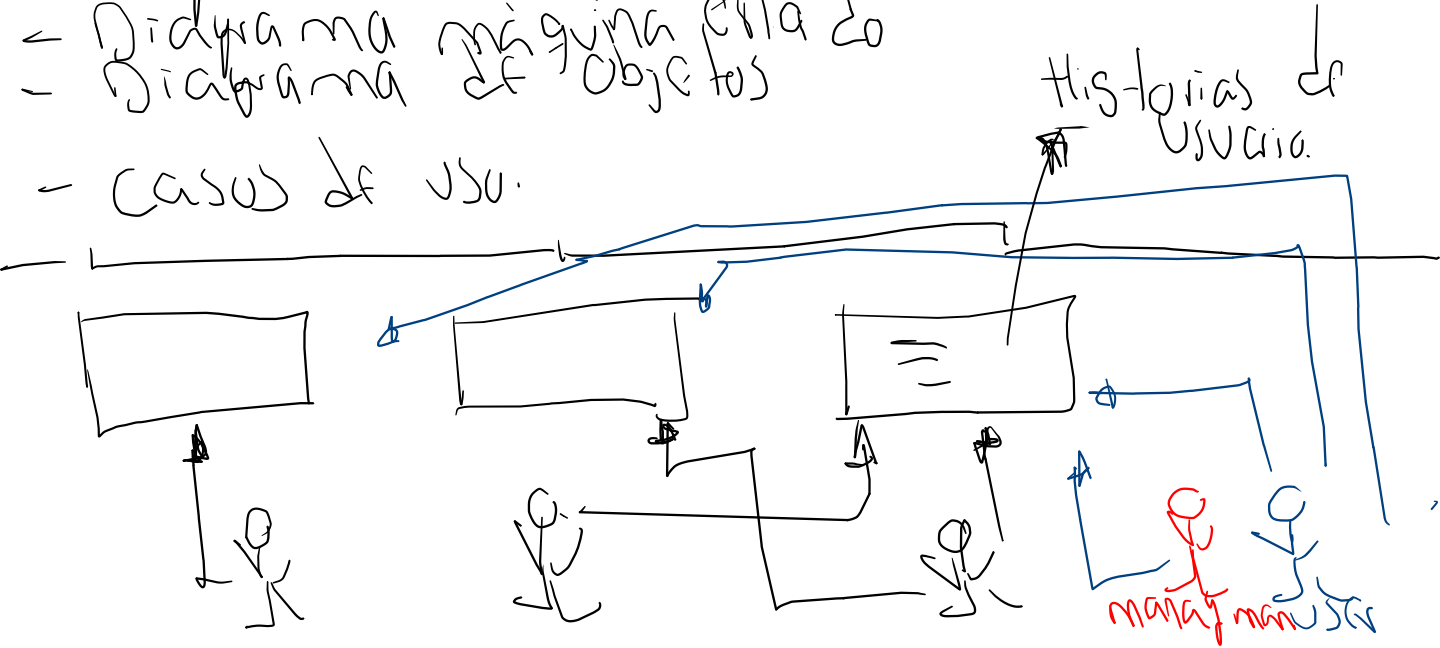
# Chat





# UML

- Diagrama de Clases
- Diagrama máquina estado
- Diagrama de Objetos
- Casos de uso.



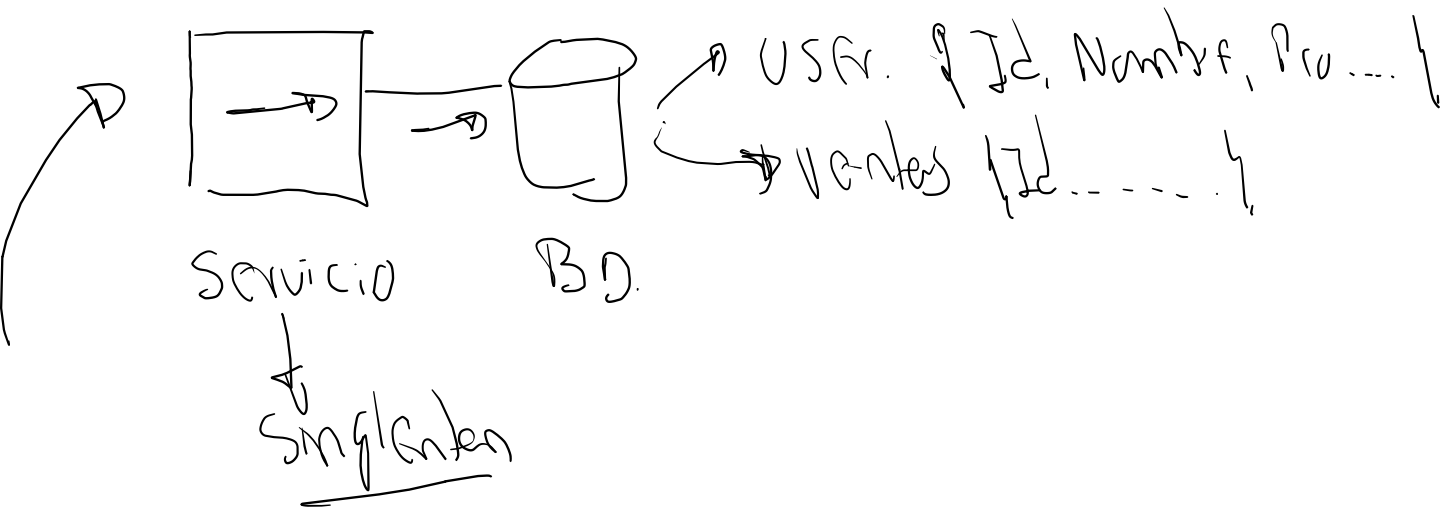
# Patrones de Diseño

- **Patrones creacionales:** Cómo crear objetos (Factory Method, Singleton).
- **Patrones estructurales:** Estructura de clases y objetos P.O.O (Adapter, Composite).
- **Patrones de comportamiento:** Interacción entre objetos (Observer, Strategy).



# Relación entre Requisitos y Patrones de Diseño

- Los requisitos definen el "qué", los patrones proporcionan el "cómo".
- Patrones ayudan a satisfacer requisitos funcionales y no funcionales.
- Ejemplo: El patrón Singleton puede abordar un requisito de escalabilidad.



# Conclusión

- La estructura de requisitos es clave para definir las necesidades del sistema.
- Los patrones de diseño permiten implementar soluciones eficientes y reutilizables.
- Su correcta combinación asegura sistemas robustos y mantenibles.