**Actividad 2 - Planeación de la problemática**

Corporación Universitaria Iberoamericana.

Ingeniería de software.

# Métodos de construcción de software (22042024\_C2\_202431).

Angel Stiven Toro Fuentes

Mayo 2024

Introducción:

En esta actividad se quiere dar a entender y expresar en modo Wiki 5 modelos de arquitectura de la siguiente manera:

Modelo de capas:

El Modelo de capas es una forma de organizar un sistema en diferentes niveles, donde cada capa se encarga de una tarea específica y sólo interactúa con la capa inmediatamente superior o inferior a ella.

1. Capa de presentación: Esta es la parte que ves y usas, como la interfaz gráfica de una aplicación o una página web. Se encarga de mostrar la información al usuario.
2. Capa de lógica de negocio: Aquí es donde ocurre la magia. Se encarga de procesar la información y aplicar las reglas del negocio. Por ejemplo, si estás comprando un producto en línea, esta capa se asegura de que el proceso de compra siga todas las reglas.
3. Capa de acceso a datos: Esta capa se encarga de interactuar con la base de datos. Guarda y recupera la información necesaria para que la capa de lógica de negocio pueda hacer su trabajo.

Cada capa tiene un rol específico, lo que facilita la modificación y el mantenimiento del sistema. Si tienes que cambiar cómo se muestra algo, solo modificas la capa de presentación sin tocar las demás.

Modelo de MVC:

El modelo vista controlador es una forma popular de organizar aplicaciones, especialmente en el desarrollo web:

1. Modelo: Representa los datos y la lógica del negocio. Es como el cerebro de la aplicación. Si estás manejando una lista de contactos, el modelo se encarga de almacenar y manipular esos contactos.
2. Vista: Es la parte que el usuario ve. Muestra la información del modelo en un formato que el usuario puede entender. Siguiendo el ejemplo, la vista mostraría la lista de contactos en tu pantalla.
3. Controlador: Actúa como intermediario entre el modelo y la vista. Recibe las acciones del usuario (como hacer clic en un botón), actualiza el modelo si es necesario y luego actualiza la vista para reflejar los cambios.

Este modelo ayuda a separar las preocupaciones: el modelo maneja los datos, la vista muestra la información y el controlador coordina todo. Esto hace que el código sea más fácil de entender y mantener.

Modelo cliente servidor:

El Modelo cliente servidor describe cómo se comunican los dispositivos en una red:

1. Cliente: Es el dispositivo o programa que solicita información o servicios. Por ejemplo, cuando usas tu navegador web para visitar un sitio, tu navegador es el cliente.
2. Servidor: Es el dispositivo o programa que proporciona la información o servicios solicitados por el cliente. En el ejemplo anterior, el servidor es el sitio web al que accedes.

El cliente envía solicitudes al servidor, y el servidor responde con la información que se solicitó. Este modelo permite que múltiples clientes se conecten a un único servidor para obtener datos o servicios.

Modelo Microservicios:

El modelo Microservicios es una forma de construir aplicaciones dividiéndolas en pequeños servicios independientes:

1. Servicios independientes: Cada microservicio se encarga de una parte específica de la funcionalidad de la aplicación. Por ejemplo, un microservicio podría manejar los pagos, otro podría gestionar los usuarios y otro podría encargarse de las recomendaciones.
2. Comunicación: Los microservicios se comunican entre sí mediante API Esto permite que cada servicio se desarrolle, despliegue y escale de forma independiente.
3. Escalabilidad y flexibilidad: Como cada microservicio es independiente, puedes escalar solo los servicios que necesitan más recursos y actualizar uno sin afectar a los demás.

Este modelo permite una gran flexibilidad y hace que sea más fácil manejar aplicaciones grandes y complejas al dividirlas en partes más manejables.

Modelo maestro esclavo:

El Modelo maestro esclavo es una forma de organizar tareas o procesos donde hay una entidad principal (maestro) y una o más entidades secundarias (esclavos):

1. Maestro: Es el control central que coordina y gestiona el trabajo. Por ejemplo, en una base de datos, el maestro puede ser el servidor principal que distribuye las tareas.
2. Esclavo: Son las entidades que reciben instrucciones del maestro y ejecutan tareas específicas. En el caso de la base de datos, los esclavos pueden ser servidores que replican la información del maestro y responden a consultas.

Este modelo ayuda a distribuir la carga de trabajo y a mejorar la eficiencia, ya que el maestro se encarga de la coordinación y los esclavos realizan las tareas asignadas.

URL repo con ejemplos realizados en JavaScript:

[AngelStivenToro/patrones\_arquitectura-software (github.com)](https://github.com/AngelStivenToro/patrones_arquitectura-software)