

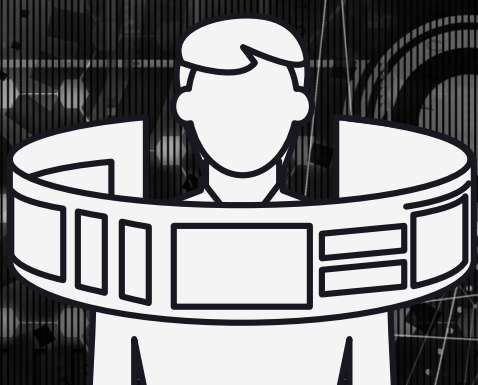
# APLICACIÓN MONOLÍTICA

está diseñado sin modularidad.



Arquitectura Todo en Uno: Todas las funciones de la aplicación se desarrollan, implementan y gestionan en un solo código base y se ejecutan en un solo proceso.

Despliegue Unitario: La aplicación monolítica se despliega como una unidad completa. Cualquier actualización o cambio en una parte de la aplicación generalmente requiere la implementación de la aplicación completa.



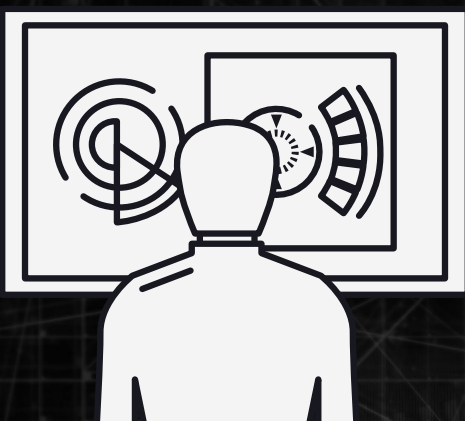
Escalar Verticalmente: La escalabilidad suele lograrse mediante la adición de recursos (como aumentar la potencia de procesamiento o la memoria) a la máquina que ejecuta la aplicación.

Desarrollo y Mantenimiento Simples: A veces, las aplicaciones monolíticas pueden ser más fáciles de desarrollar y mantener, especialmente para proyectos más pequeños o menos complejos.



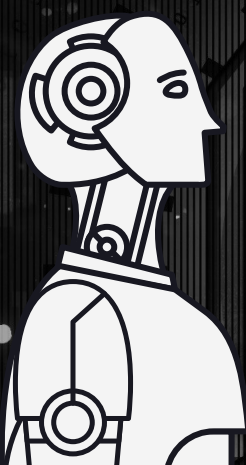
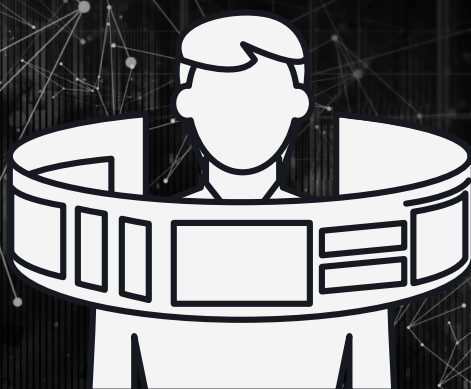


# APLICACIÓN MONOLÍTICA



Escalabilidad Limitada: La escalabilidad puede ser un problema, ya que la adición de recursos se vuelve más difícil a medida que la aplicación crece.

Despliegues Complejos: Actualizar o modificar partes específicas de la aplicación puede requerir despliegues completos, lo que puede causar tiempos de inactividad.

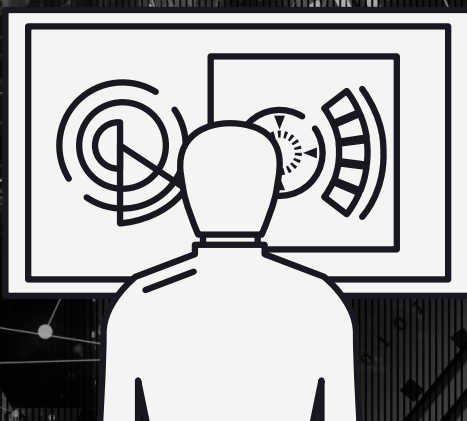


Dificultades en Equipos Grandes: En proyectos grandes, puede ser más difícil para equipos de desarrollo trabajar de manera colaborativa y eficiente en un único código base.



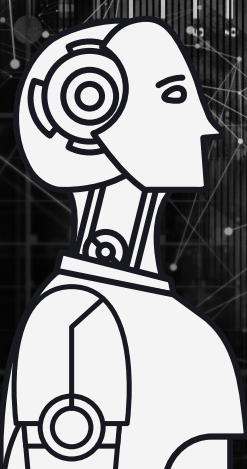
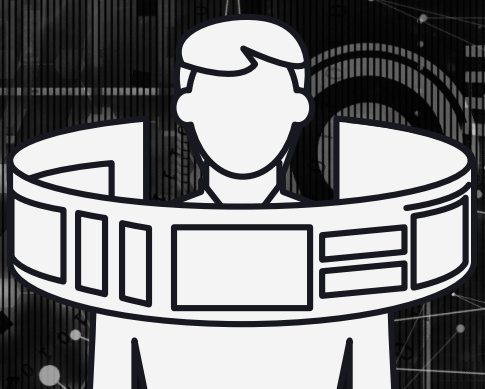
# APLICACIONES POR SERVICIOS

La Arquitectura Orientada a Servicios es un enfoque para desarrollar software que utiliza servicios como los componentes fundamentales para construir aplicaciones. En lugar de crear aplicaciones monolíticas donde todas las funcionalidades están integradas directamente,



Una API es un conjunto de reglas y definiciones que permite que diferentes software se comuniquen entre sí. Proporciona un conjunto de funciones y protocolos que permite a los desarrolladores acceder y utilizar las características o datos de una aplicación, servicio o plataforma.

**endpoint** se refiere a un punto de conexión específico donde las aplicaciones pueden comunicarse con un servicio o recurso particular. En otras palabras, un endpoint es una URL (Uniform Resource Locator) o URI (Identificador de Recursos Uniforme) que se utiliza para acceder a un servicio o recurso en la web.



Los verbos HTTP:son métodos que indican la acción que se debe realizar en un recurso determinado.

- 1.GET: Solicita la representación de un recurso específico. No debería tener ningún efecto secundario en el servidor.
- 2.POST: Envía datos para que sean procesados a un recurso específico. Puede llevar a cambios en el estado del servidor o generar nuevos recursos.
- 3.PUT: Actualiza un recurso existente o crea uno si no existe en la ubicación especificada.
- 4.DELETE: Elimina el recurso especificado.

"servicio" se refiere a una unidad de funcionalidad independiente y autónoma que está diseñada para realizar una tarea específica dentro de una aplicación más grande. Estos servicios son componentes lógicos que pueden ser accedidos y utilizados a través de una red, a menudo a través de una interfaz estandarizada.

