



# APLICACIONES MONOLÍTICAS VS APLICACIONES POR SERVICIOS



# APLICACIONES MONOLÍTICAS

Una aplicación monolítica es un tipo de arquitectura de software en la que todos los componentes y funcionalidades de la aplicación se desarrollan, implementan y despliegan como una única unidad.

## CARACTERÍSTICAS

**Monolito único:** La aplicación se compone de una sola unidad de ejecución. Todas las funcionalidades están integradas en un solo código base.

**Despliegue conjunto:** Las actualizaciones o cambios en la aplicación requieren la implementación de la aplicación completa.

Las aplicaciones monolíticas tienen ventajas en términos de simplicidad y facilidad de desarrollo inicial. Sin embargo, a medida que las aplicaciones crecen en complejidad y tamaño, pueden surgir desafíos, especialmente en términos de escalabilidad y mantenimiento.

# APLICACIONES POR SERVICIOS

1

## APLICACIONES BASADAS EN SERVICIOS

Están construidas utilizando servicios externos o internos para proporcionar funcionalidades específicas.

2

## ARQUITECTURA DE MICROSERVICIOS

Este enfoque arquitectónico implica descomponer una aplicación en pequeños servicios independientes, cada uno enfocado en realizar una tarea específica.

# ¿QUE ES UN SERVICIO?

## SERVICIO WEB (WEB SERVICE).

En un sentido más amplio, un servicio web es una tecnología que permite la comunicación entre aplicaciones a través de la web.

## VERBO HTTP.

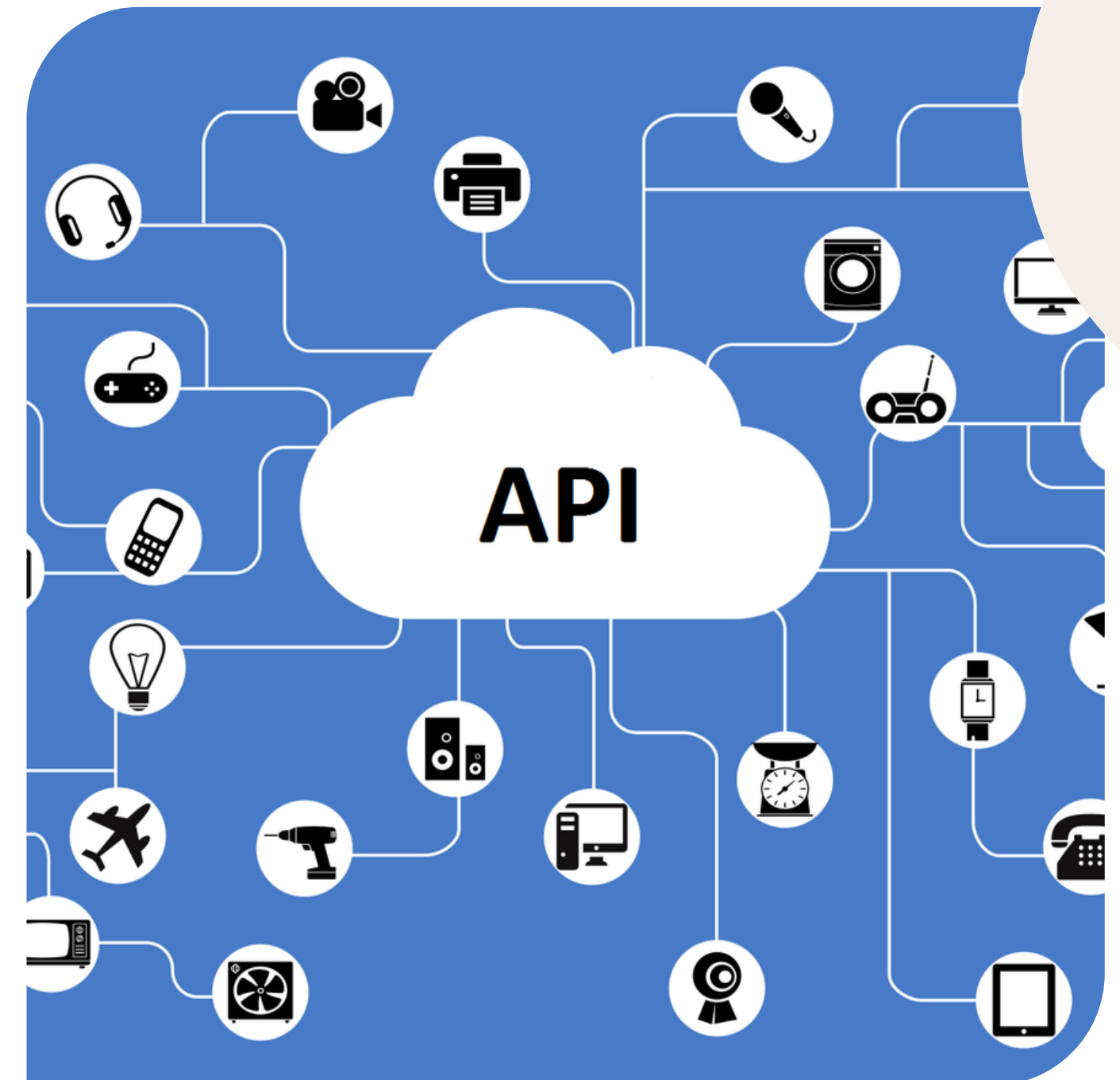
En el contexto de las operaciones en una API o servicio web, el término "verbo" se refiere a los métodos HTTP que se utilizan para realizar acciones en un recurso.

Los verbos representan las acciones que se pueden realizar en un servicio web o API. Cada verbo tiene un propósito específico, como recuperar datos, enviar datos, actualizar recursos o eliminar recursos.

# ¿QUE ES API?

API significa Interfaz de Programación de Aplicaciones. Es un conjunto de reglas y definiciones que permite que diferentes software se comuniquen entre sí. En otras palabras, una API define cómo distintos componentes de software deben interactuar.

facilitan la integración de diferentes sistemas y aplicaciones, permitiendo que trabajen juntos de manera eficiente.



# ENDPOINT

En el contexto de las API se refiere a un punto final o una URL específica a la que se puede acceder para realizar operaciones o intercambiar datos. En otras palabras, un endpoint es una interfaz expuesta por una API que define la ubicación de un recurso o servicio particular.

## EJEMPLO.

en una API de servicios web para pronósticos del tiempo, podrías tener un endpoint para obtener el pronóstico actual, otro para obtener el pronóstico a futuro y otro para obtener el historial de pronósticos. Cada uno de estos endpoints tendría una URL única que los identifica.

# VERBOS HTTP

- **GET:** Solicita una representación del recurso especificado. Se utiliza para recuperar información.
- **POST:** Envía datos para ser procesados a un recurso especificado. Se utiliza para enviar datos y solicitar que se guarden en el servidor.
- **PUT:** Actualiza un recurso existente o crea uno si no existe. El cuerpo de la solicitud generalmente contiene la representación actualizada del recurso.
- **DELETE:** Elimina el recurso especificado. Se utiliza para solicitar la eliminación de un recurso en el servidor.
- **PATCH:** Aplica modificaciones parciales a un recurso. Se utiliza para actualizar parcialmente un recurso.
- **OPTIONS:** Proporciona información sobre las opciones de comunicación para el recurso especificado. Se utiliza para describir las opciones de comunicación disponibles para el recurso.
- **HEAD:** Similar a GET, pero solicita solo los encabezados de la respuesta, sin el cuerpo del mensaje. Se utiliza para obtener información sobre el recurso sin solicitar su representación completa.