

# Fundamentos de Desarrollo Backend

¡Bienvenidos, aprendices, al emocionante mundo del desarrollo Backend! En este taller exploraremos los pilares fundamentales que sustentan las aplicaciones y servicios que usamos a diario. Desde las APIs que conectan sistemas hasta los protocolos que aseguran nuestras transacciones en línea, el Backend es el motor invisible que hace posible la tecnología moderna.

Este espacio de aprendizaje no solo busca proporcionarles conocimientos teóricos, sino también fomentar su curiosidad, habilidades de análisis y pensamiento crítico. A través de investigaciones, ejemplos prácticos y casos de estudio, desarrollaremos una comprensión sólida de los conceptos clave que les permitirán construir soluciones eficientes y escalables en el futuro.

Nuestro objetivo es que al finalizar este taller puedan:

- Entender qué es y cómo funciona el Backend.
- Dominar los fundamentos de las APIs y su uso en el desarrollo.
- Reconocer la importancia de los protocolos y las peticiones HTTP.
- Conocer patrones de arquitectura de software aplicados a diferentes necesidades.
- Diseñar y proponer soluciones innovadoras que impacten positivamente en el mundo digital.

Recuerden: este taller es una invitación a explorar, preguntar y crear. Cada paso los acercará un poco más a ser profesionales confiables y creativos en el mundo del desarrollo de software. ¡Manos a la obra!

## 1. Descubriendo el Backend

- ¿Qué crees que significa "Backend" en desarrollo de software? Investiga ejemplos reales de sistemas con Backend (por ejemplo, un banco o una red social).

## 2. Explorando APIs

- Define qué es una API. ¿Qué ventajas y desventajas encuentras al usarlas en el desarrollo?
- Investiga dos ejemplos de APIs famosas (por ejemplo, Google Maps y Twitter). ¿Cómo ayudan a las empresas?

## 3. Backend vs. API

- Imagina que estás explicándole a un amigo las diferencias entre Backend y API. ¿Cómo lo harías? Usa analogías si es necesario.

## 4. Tipos de API

- Investiga y compara los siguientes tipos de API:

- Pública.
- Privada.
- HTTP API.
- REST API.
- SOAP API.

- Relaciona cada tipo con un ejemplo práctico. ¿Dónde crees que se usan?

## 5. ¡Es tu turno!

- Con tus propias palabras, explica por qué deberíamos usar APIs en nuestros desarrollos.

## 6. Protocolos HTTP y HTTPS

- ¿Qué son y cómo funcionan estos protocolos?
- Encuentra tres sitios web y verifica cuál usa HTTPS y cuál HTTP. ¿Qué diferencias notas?

## 7. Seguridad en la web

- ¿Qué ventaja tienen los sitios con protocolo HTTPS? Relaciónalo con un caso real donde se necesite seguridad (por ejemplo, compras en línea).

## 8. Peticiones HTTP: Tus herramientas básicas

- ¿Qué son y cuáles son los métodos de solicitud más comunes? (GET, POST, PUT, DELETE). Describe para qué sirve cada uno.
- Da un ejemplo práctico donde usarías cada tipo de solicitud.

## 9. Cómo funciona una petición HTTP

- Explica, paso a paso, cómo funciona una petición HTTP desde que se envía hasta que se recibe una respuesta.

## 10. Desglosando una petición HTTP

- Detalla qué información contiene una petición HTTP. ¿Cómo crees que podrías usar esa información en un desarrollo?

## 11. Códigos de estado HTTP

- Investiga los códigos de respuesta HTTP. Agrúpalos (1xx, 2xx, 3xx, 4xx, 5xx) y da ejemplos prácticos de cuándo se usaría cada uno.

## 12. Explorando APIs públicas *Nota: para resolver esta pregunta, ver Actividad: Caso de Estudio con APIs Públicas al final de este documento*

- Investiga tres APIs públicas:

- ¿Cuál es la funcionalidad principal de cada una?
- Identifica y explica dos tipos de peticiones (GET, POST, PUT o DELETE) en cada API.
- Describe cómo funcionan las APIs.

13. ¡Piensa en grande! **Nota:** para resolver esta pregunta ver Actividad: Caso de Estudio con APIs Públicas al final de este documento

- Si pudieras crear una API pública que resolviera un problema, ¿cuál sería? Asegúrate de investigar si ya existe algo similar.

14. La arquitectura en sistemas de software

- Define qué es la arquitectura de un sistema de software. ¿Cómo crees que afecta al éxito de un proyecto?

15. Patrones de arquitectura

- Investiga y compara los siguientes patrones de arquitectura:
  - Microkernel.
  - Microservicios.
  - Arquitectura en capas.
  - Basada en eventos.
  - Basada en el espacio.
- Da un ejemplo de aplicación o sistema que usaría cada patrón.

16. Cliente vs. Servidor

- ¿Qué significa que un lenguaje de programación "corra del lado del cliente"? Da ejemplos de lenguajes y cuándo se usan.

17. Conociendo Node.js

- ¿Qué es Node.js y por qué es tan popular para el desarrollo backend? Da ejemplos de aplicaciones que podrías crear con él.

18. Entendiendo Middleware

- En tus palabras, explica qué es un middleware en Node.js. Imagina que lo estás explicando a alguien que nunca ha programado antes.

# Actividad: Caso de Estudio con APIs Públicas

En esta actividad, como aprendices, tendrán la oportunidad de explorar, analizar y proponer soluciones basadas en el uso de APIs públicas. El objetivo es que comprendan cómo estas herramientas facilitan la integración de funcionalidades en aplicaciones y cómo pueden ser utilizadas para resolver problemas del mundo real.

## ¿Qué haremos?

1. **Investigación de APIs Públicas (Pregunta 12):**
  - Seleccionen tres APIs públicas de su interés. Pueden ser APIs populares (como OpenWeather, Spotify o GitHub) o APIs menos conocidas, pero que resuelvan problemas interesantes.
  - Para cada API, respondan las siguientes preguntas:
    - ¿Cuál es la funcionalidad principal de la API? ¿Qué problema resuelve?
    - Identifiquen al menos dos tipos de peticiones (GET, POST, PUT o DELETE) y expliquen qué hacen.
    - Describan, de forma sencilla, cómo funciona la API y den un ejemplo de uso práctico.
2. **Creación de una API Ideal (Pregunta 13):**
  - Piensen en un problema real que creen que podría resolverse con una API, y propongan una idea innovadora para una API pública que aún no exista.
  - Responda las siguientes preguntas:
    - ¿Qué problema resolvería esta API?
    - ¿Qué funcionalidades principales tendría?
    - ¿Cómo ayudaría a los desarrolladores o usuarios finales?

## ¿Cómo lo haremos?

- **Organización por Equipos:** Divídanse en equipos pequeños (2-3 integrantes). Cada equipo seleccionará sus APIs y diseñará su propuesta.
- **Presentación Creativa:**
  - Cada equipo preparará una presentación breve (5-7 minutos) donde explique:
    - Las APIs que investigaron y sus hallazgos.
    - Su idea de API ideal, usando gráficos, diagramas o ejemplos para ilustrar su funcionalidad.
  - Usen herramientas como diapositivas, pizarras o dibujos en papel para apoyar su presentación.

## Criterios de Evaluación:

- Claridad en la explicación de las APIs investigadas.
- Creatividad e innovación en la propuesta de una nueva API.

- Capacidad para relacionar los conceptos teóricos con aplicaciones prácticas.
- Trabajo en equipo y habilidades de comunicación durante la presentación.

### **Propósito de la Actividad**

Esta actividad les permitirá:

- Comprender cómo funcionan las APIs en el mundo real.
- Analizar problemas desde la perspectiva de un desarrollador.
- Desarrollar habilidades para comunicar ideas técnicas de manera clara y efectiva.
- Fomentar la creatividad para idear soluciones tecnológicas.

¡Anímense a explorar y **sorprendernos** con sus propuestas! El mundo del desarrollo backend está lleno de posibilidades, y ustedes son los futuros creadores de soluciones innovadoras.