

MATERIAL EXTRA · SESIÓN 1

APIs sin misterio.

Application Programming Interface explicada desde cero — con hamburguesas, código real y tres APIs que podés probar hoy sin configurar nada.

El misterio de las APIs

Cuando empezás en tecnología, escuchás la palabra API en todos lados. Te dicen que es vital, pero las explicaciones son tan abstractas que cuesta digerirlas.

En términos sencillos, **una API (Application Programming Interface) es el puente que permite que dos aplicaciones se hablen entre sí.**

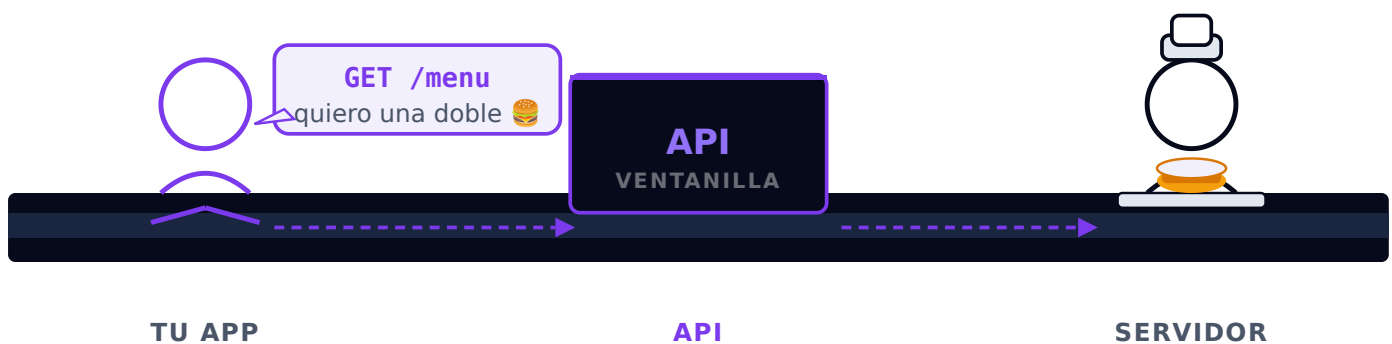


La Aplicación A no necesita saber cómo funciona la B por adentro. Solo necesita saber **qué pedirle y cómo pedirlo**. La API define ese contrato.

Aprendamos con comida rápida

Imaginá esto: vas a pedir una hamburguesa a **CloudBurger**. Llegás con hambre y ya sabés qué querés, pero no podés entrar directo a la cocina a armar tu hamburguesa ni a sacar ingredientes de la heladera.

Hay una **barrera lógica** por organización y seguridad. Necesitás que alguien comunique tu pedido.



En el software pasa exactamente lo mismo. Tu aplicación no puede entrar directo a la base de datos de otra empresa. **La API es la ventanilla — el punto de entrada controlado.**

La cajera (La API)

La cajera toma tu orden (**Request**), la pasa a la cocina y te la devuelve lista (**Response**). Vos no necesitás saber cómo se arma la hamburguesa — solo disfrutás el resultado.



En el software, **la API conecta aplicaciones sin que tengan que conocer sus códigos internos**. Cuando Instagram carga tu ubicación de Google Maps — hay una API en el medio procesando todo.

REQUEST

Lo que vos pedís, con todos los datos necesarios para que el servidor entienda qué hacer.

RESPONSE

Lo que recibís: los datos, un código de estado, y a veces un mensaje de error si algo falló.

La petición HTTP Request

Cuando pedís algo, no decís "dame comida". Revisás el menú y elegís algo específico. En programación, esas opciones se llaman **Endpoints** — URLs específicas que el servidor tiene disponibles.

CloudBurger · API MENU

ENDPOINTS DISPONIBLES — V1.0

GET	/menu	Ver la carta completa
POST	/hacer-pedido	Crear una nueva orden
PUT	/pedido/{id}/extra	Modificar un pedido
DELETE	/cancelar/{id}	Cancelar una orden

GET — LEER DATOS

Ver el menú de precios sin hacer ningún cambio. Es como hojear la carta.

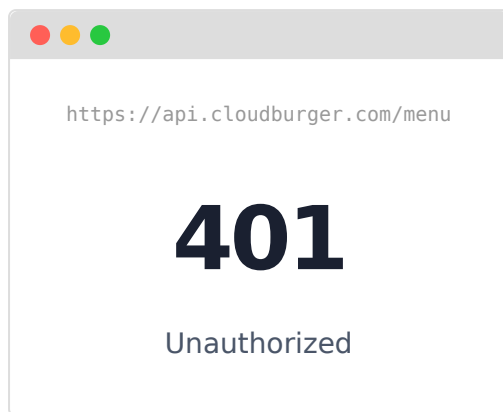
POST — CREAR ALGO NUEVO

Hacer un pedido nuevo. Enviás datos y el servidor crea un registro.

La llave API Key

La cajera no va a atender a cualquiera que entre gritando pedidos desde la vereda. Primero necesita validar que sos cliente, que tenés crédito y que tu pedido es legítimo.

En el software real, para consumir los datos de un servidor necesitás identificarte. Esto lo hacés mediante una **API Key** o **Token de seguridad** en las cabeceras (**Headers**) de tu petición.



SIN API KEY

El servidor te devuelve **401 Unauthorized**. No te da los datos — ni siquiera te explica qué tiene disponible.

CÓMO SE ENVÍA

En el header de la petición:

`Authorization: Bearer TU_API_KEY`

Buen dato: las APIs de práctica del slide 10 son públicas — no requieren API Key. Perfectas para dar los primeros pasos sin configurar nada.

La respuesta

The Response

La cocina prepara el pedido y la cajera te lo entrega. Pero en internet, el servidor no te manda una bandeja física — te devuelve un formato de texto estructurado llamado **JSON**.

```
// Respuesta de POST /hacer-pedido
{
  "orden_id":    4589,
  "producto":   "Hamburguesa doble",
  "cantidad":   2,
  "extra_queso": true,
  "tiempo_est": "8 minutos",
  "estado":     "en preparación"
}
```

¿QUÉ ES JSON?

- Un formato de texto estructurado — no es código que "corre", es un formulario pre-llenado.
- Cada valor tiene **clave** (el nombre) y **valor** (el dato). Tu app lee las claves y decide qué una un mostrar.
- Los números van sin comillas. Los textos van entre comillas. Los `true` o `false` booleanos son

CÓDIGOS DE ESTADO HTTP

200 OK

Todo salió bien.

404 Not Found

No existe ese endpoint.

500 Server Error

Algo falló en el servidor.

¿Por qué importan?

Las APIs existen para **ahorrarnos trabajo** y no reinventar la rueda. Cuando programás una app de entregas, no lanzás satélites al espacio ni creás cartografía desde cero — te conectás a la API de Google Maps.

- **Google Maps API:** cualquier app puede mostrar mapas, calcular rutas y obtener tráfico en tiempo real. Sin lanzar un solo satélite.
- **Stripe API:** procesar pagos con tarjeta en minutos, sin construir toda la infraestructura bancaria.
- **OpenAI API:** integrar inteligencia artificial a tu proyecto con unas pocas líneas de código.
- **AWS API Gateway:** publicar tu propio backend como una API segura, escalable y lista para que cualquier app la consuma.

Aprender a consumir APIs te permite integrar bases de datos externas, procesar pagos seguros o conectar modelos de IA en tu proyecto en cuestión de minutos.

No tenés que construir cada pieza desde cero. Elegís el servicio que necesitás, leés su documentación y lo conectás. Eso es trabajar como un devops profesional.

Herramientas para probar APIs

No necesitás programar toda la interfaz visual de inmediato para validar si los datos llegan bien. Estas herramientas te permiten hacer peticiones y ver respuestas directamente:



Postman

La herramienta visual estándar en la industria. Simulá peticiones, guardá colecciones y revisá respuestas con interfaz gráfica completa.



Thunder Client

Extensión muy ligera que corre directo dentro de VS Code. Ideal para no salir del editor mientras desarrollás.



Fetch / Requests

Métodos nativos — fetch en JavaScript y requests en Python — para llamarlas desde tu código.

```
# Python – consumir una API en 3 líneas
import requests
res = requests.get("https://jsonplaceholder.typicode.com/posts/1")
print(res.json())
```

Tip para empezar: instalá Postman o Thunder Client y usálos con las APIs del próximo slide. Antes de escribir una línea de código propio, vas a entender exactamente qué devuelve cada endpoint.

¡A practicar!

La mejor forma de entender las APIs es haciendo tus propias pruebas. Estas 3 son **gratuitas, sin registro y sin configuraciones complejas** — perfectas para principiantes:

JSONPlaceholder

jsonplaceholder.typicode.com

Excelente para simular peticiones de posts, comentarios o usuarios ficticios. Tiene GET, POST, PUT y DELETE — todo responde como si fuera real sin guardar nada. Perfecta para aprender los métodos HTTP.

Open Library API

openlibrary.org/developers/api

Buscá información, autores y portadas de libros reales por título o ISBN. Datos reales, sin API Key. Buena para ver cómo se mandan parámetros en la URL (?title=...).

The Cat API / Dog API

thecatapi.com · dog.ceo/dog-api

Devuelve imágenes aleatorias de gatos o perros cada vez que la llamás. Divertida para aprender — 100% funcional para practicar fetch en JS o requests en Python.

Las APIs son la columna vertebral de las aplicaciones modernas.

Entenderlas a fondo te abre las puertas a crear proyectos mucho más completos, interactivos y profesionales — en cuestión de minutos, no de semanas.

No tenés que dominarlas todas. Solo tenés que saber **cómo pedirles lo que necesitás.**