

# Bootcamp de Full Stack

Bienvenidos a la clase N°29

- Async / Await
- JSON
- Protocolo HTTP
- Proyecto Integrador F1 (i2)

# JavaScript

## Async / Await

Los operadores **async/await** son una forma moderna y sencilla de manejar código asíncrono. Permite escribir funciones que realizan tareas que toman tiempo sin bloquear la ejecución del programa. Aunque el proceso sigue siendo asíncrono, el código se lee de forma clara y secuencial, con una apariencia similar al código síncrono.

Usos comunes:

- Esperar temporizadores (setTimeout, delay)
- Esperar operaciones con archivos
- Esperar operaciones con APIs y bases de datos

Sintaxis:

```
async function getData() {  
  try {  
    const timer = await getTimer();  
    console.log(timer)  
  } catch (error) { console.error(error.message); }  
}
```

# JavaScript

## JSON

**JSON** (JavaScript Object Notation) es un formato de texto liviano y fácil de leer, diseñado para almacenar y transmitir datos de forma estructurada. Aunque se basa en la sintaxis de objetos de JavaScript, es independiente del lenguaje, lo que permite que muchos lenguajes (Python, Java, PHP, etc) los puedan leer y generar. Es ampliamente utilizado para intercambiar información entre aplicaciones web y servidores.

### Reglas:

- Los datos están en pares clave-valor: "clave": valor
- Las claves van siempre entre comillas dobles
- Los valores pueden ser: string, number, boolean, array y objects

### Sintaxis:

```
const personObject = { name: "Juan", age: 30 };  
const personJson = JSON.stringify(personObject);
```

```
const personJson = '{"nombre":"Sergio","edad":30}';  
const personObject = JSON.parse(personJson);
```

# HTTP

## Fundamentos

### ¿Qué es HTTP?

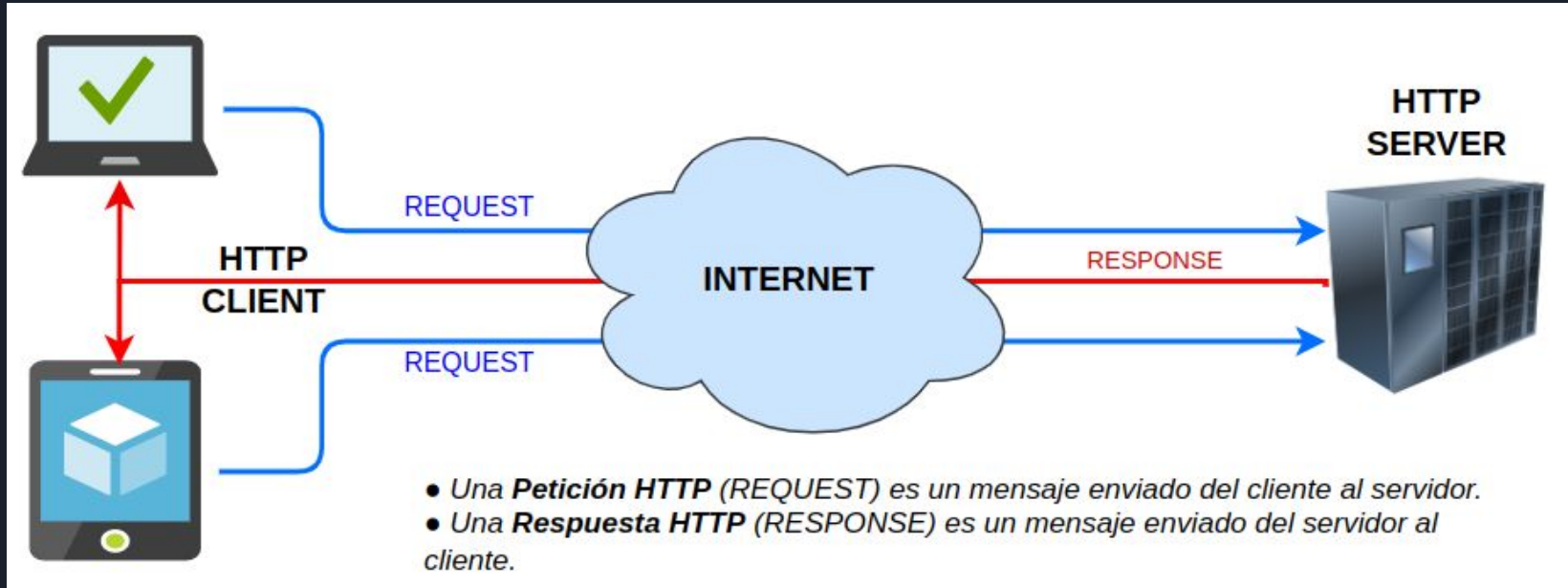
El **HyperText Transfer Protocol** es un protocolo para el intercambio de información en la red; es decir, que define las reglas de ese intercambio. Es un protocolo para transferencias de red bajo el modelo Cliente/Servidor.

El Cliente envía una petición y recibe una respuesta. El Servidor recibe una petición y envía una respuesta asociada. Por defecto, el Cliente inicia la comunicación y el Servidor responde.

Una característica importante de HTTP es que se trata de un protocolo sin estado ¿Qué significa esto? En lo que concierne a las reglas de este protocolo, toda la información está contenida en el intercambio.

# HTTP

## Fundamentos



# HTTP

## Fundamentos

### HTTP METHODS

- GET
- POST
- PUT (total)
- PATCH (parcial)
- DELETE

### STATUS CODES

- 200 Ok
- 201 Created
- 400 Bad request
- 401 Unauthorized (autenticación)
- 403 Forbidden (permisos)
- 404 Not found
- 409 Conflict
- 422 Unprocessable Entity
- 500 Internal Server Error

# BREAK

Descansemos 10 minutos



# Proyecto Integrador F1

## Iteración N°2



Para llevar a cabo el proyecto integrador F1, accede al documento de la consigna a través del siguiente enlace:

<https://docs.google.com/document/d/181methzV3GUL-YCBNp-qE94qE8-eEiA9Q169gyzUvSs/edit?usp=sharing>



# CIERRE DE CLASE

Continuaremos en la próxima clase

