

LECTURE 4

PLATFORM BASED DEVELOPMENT

ACKNOWLEDGEMENTS

- A bunch of Websites where I took all the pictures without asking. Apologies :)
- Carrie Anne @ CrashCourse

TODAY'S LECTURE

@FRANCISCOVILCHEZV

- ¿Qué sucede en el backend?
- Conexión entre el backend y la base de datos
- HTTP protocol
- Cómo se accede a las funciones del backend?
- Qué sucede en el front end?
- Conexión entre el frontend y backend
- Arquitectura completa
- Web Application Stack

¿QUÉ SUCEDE EN EL BACKEND?

¿QUÉ SUCEDA EN EL BACKEND?

- El backend esta encargado de obtener información de la base de datos (u otras fuentes), y en algunos casos aplicar alguna lógica.
- Algunas lógica adicional puede ser: hacer conversiones de algunos valores numéricos, filtrar información que el usuario no tiene permisos para acceder, convertir archivos de imágenes a bits, etc.
- Estas funciones del backend son invocadas desde el frontend (la página web o app móvil), y ellos se encargan de mostrar la información que el backend le proporciona.

```
✓ function getProducts(){
  var products = getProductsFromDatabase();
  return products;
}

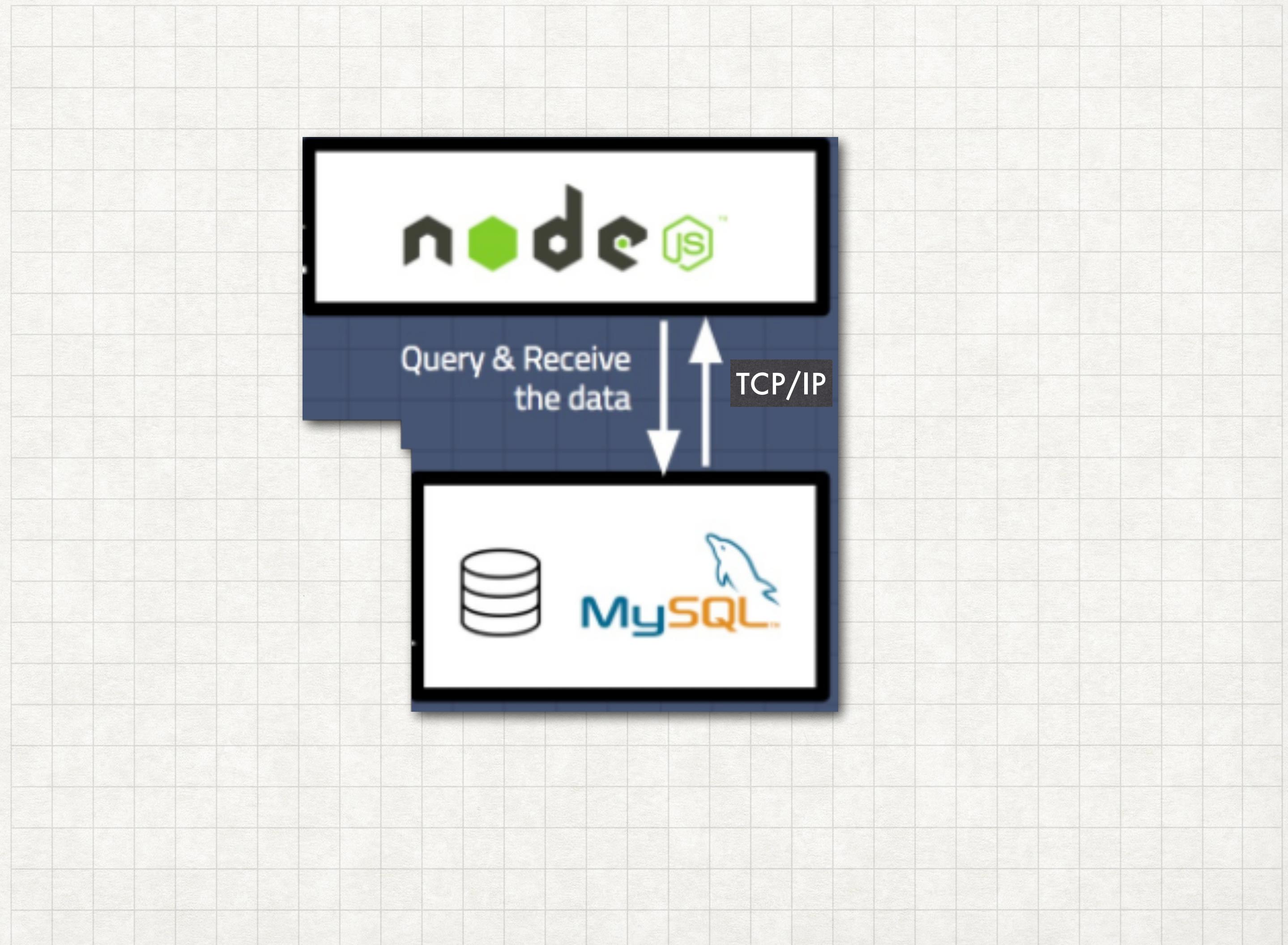
✓ function getProductsFromDatabase(){
  ✓ return mysql.query(
    "SELECT product_id, product_name " +
    " FROM product "
  );
}
```

CONEXIÓN ENTRE EL BACKEND Y BASE DE DATOS

CONEXIÓN ENTRE EL BACKEND Y BASE DE DATOS

CONEXIÓN

- La comunicación se hace a través del protocolo TCP/IP (mayoría de casos). Gracias a ello, nuestro programa puede recibir una confirmación del envío del query, e incluso recibir la información solicitada.
- Los mismos lenguajes que usted escribía en el terminal para el MySQL, el lenguaje backend lo envía a través de TCP (indica el puerto y host que debe recibirla para llegar al MySQL).



CONEXIÓN ENTRE EL BACKEND Y BASE DE DATOS LIBRERÍAS

- El backend generalmente utiliza una librería que le permite mandar las sentencias SQL a la base de datos con mayor facilidad.
- Sin la librería, la creación de la comunicación TCP/IP entre el backend y la BD se haría desde cero. En el mundo de la programación, constantemente encontrará librerías ya creadas que realizan algunas de las tareas que usted desea.

The top-left screenshot shows the npmjs.com page for the mysql package. It includes the introduction, a note about the license (100% MIT), and a code example for connecting to MySQL using Node.js:

```
var mysql      = require('mysql');
var connection = mysql.createConnection({
  host     : 'localhost',
  user     : 'me',
  password : 'secret',
  database : 'my_db'
});

connection.connect();

connection.query('SELECT 1 + 1 AS solution', function (error, results, fields) {
  if (error) throw error;
  console.log('The solution is: ', results[0].solution);
});

connection.end();
```

The bottom screenshot shows the dev.mysql.com page for Connector/Python, specifically the section '5.1 Connecting to MySQL Using Connector/Python'. It includes a note about the connect() constructor and a code example for connecting to MySQL using Python:

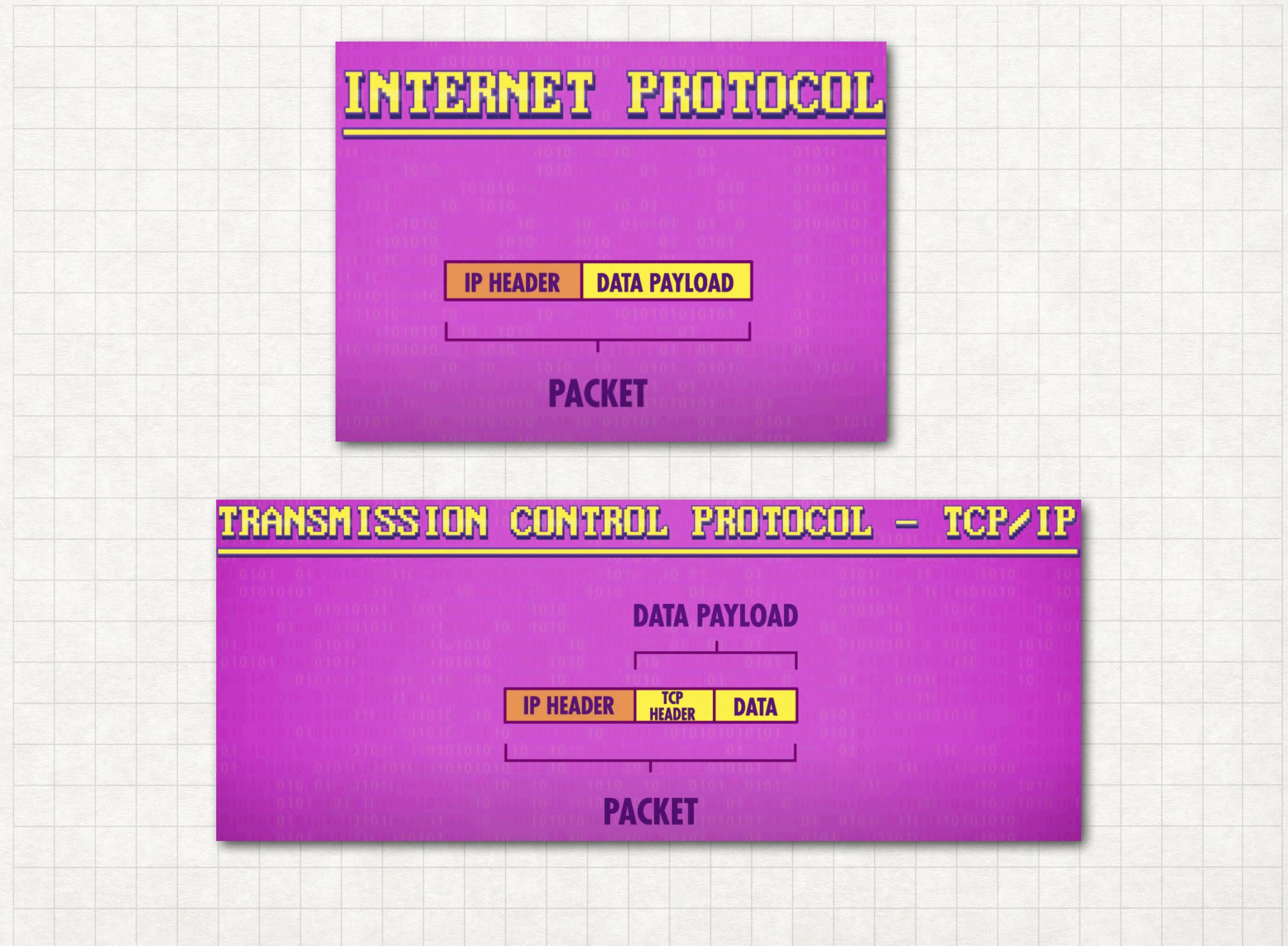
```
1 import mysql.connector
2
3 cnx = mysql.connector.connect(user='scott', password='password',
4                               host='127.0.0.1',
5                               database='employees')
6
7 cnx.close()
```

HTTP PROTOCOL

HTTP PROTOCOL

REMEMBER

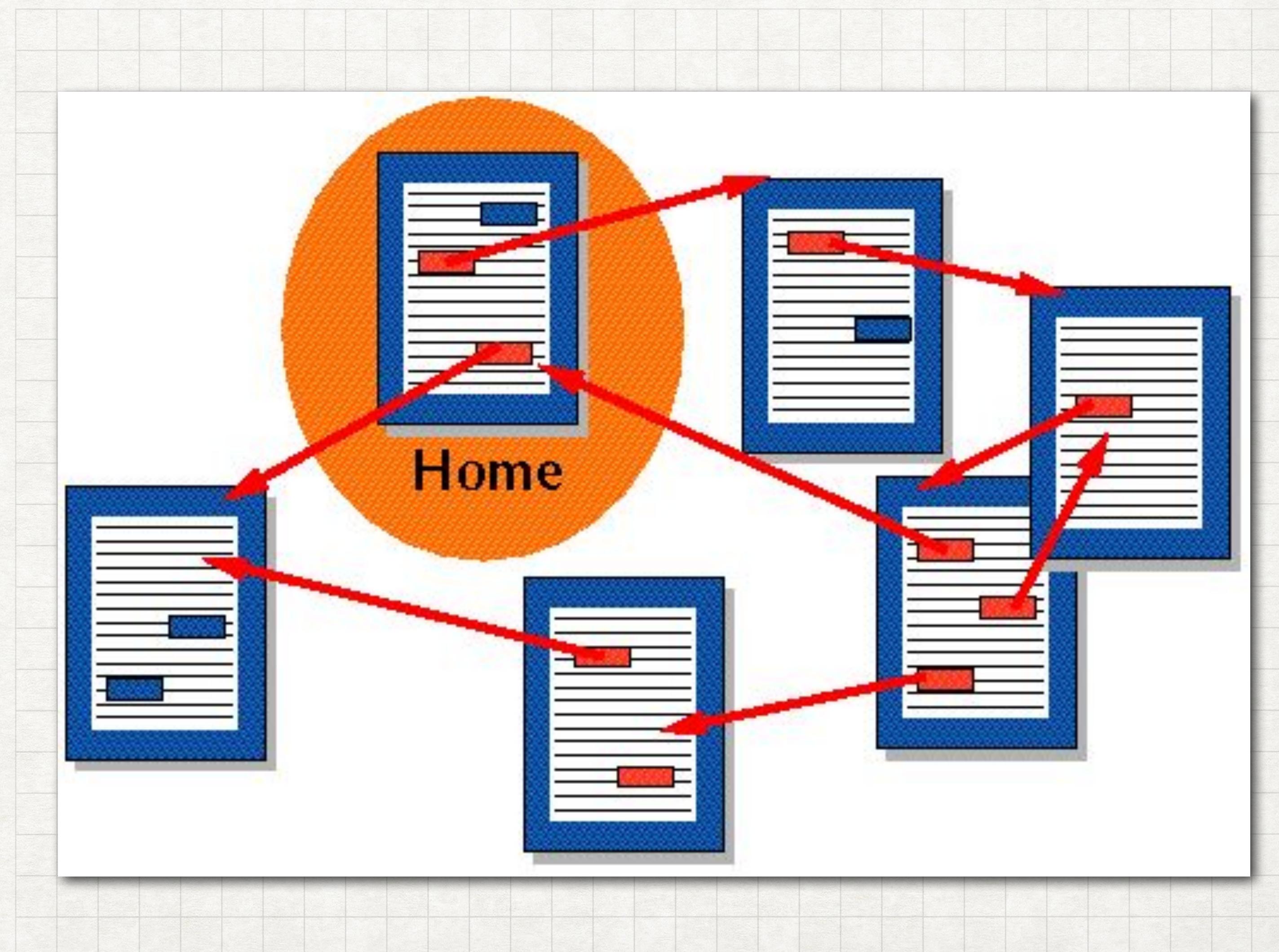
- Protocol: Conjunto de reglas y pautas para comunicar data.
- Por ejemplo, el protocolo IP, en el cual necesitabas especificar la dirección de la computadora a quien querías enviarle la data.
- Otros protocolos que hemos aprendido son los protocolos TCP y UDP. Ambos construidos sobre el protocolo IP.



HTTP PROTOCOL

HYPertext

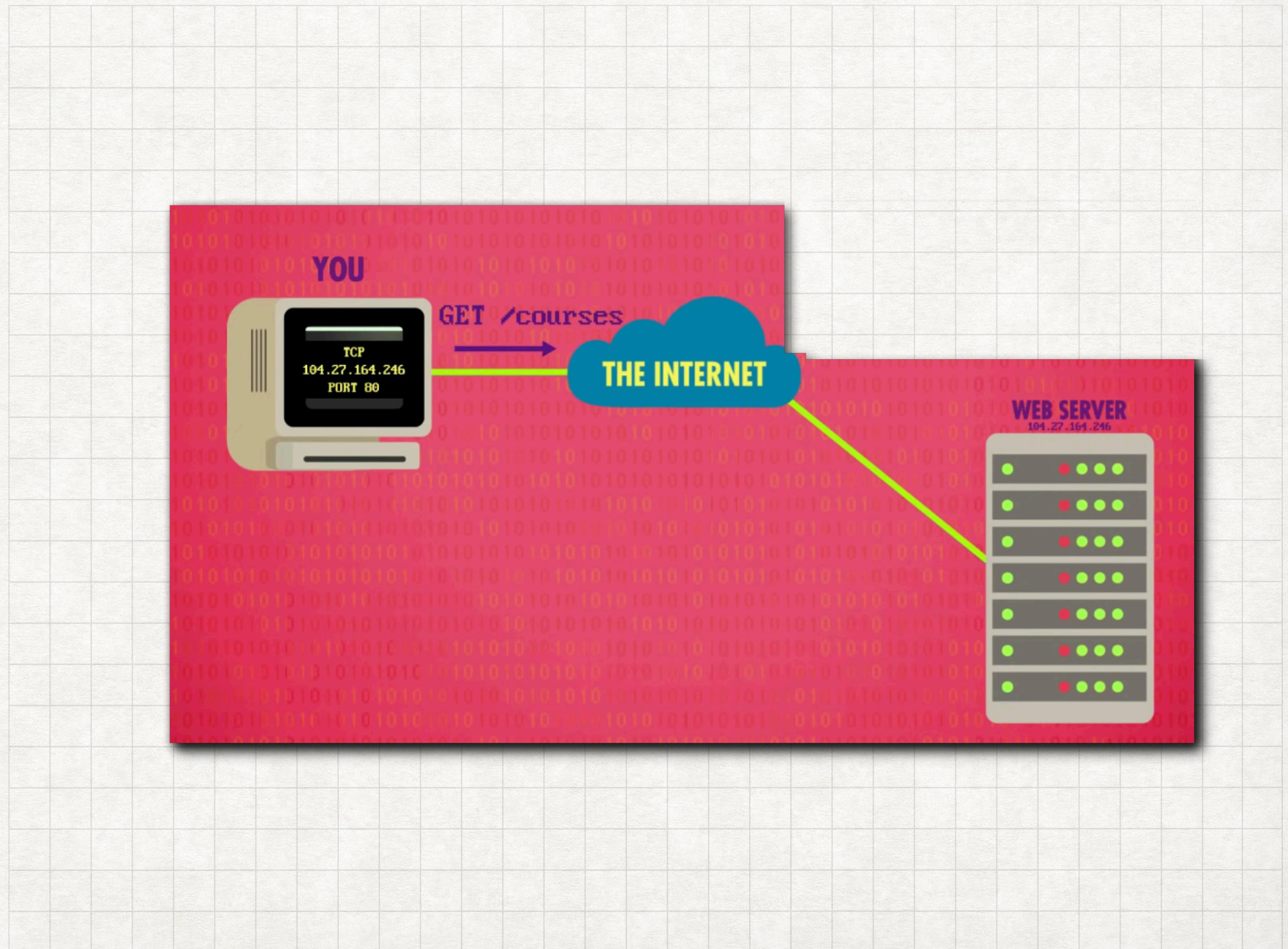
- El protocolo HTTP (Hypertext Transfer Protocol), fue creado para poder enviar hipertexto (hypertext) entre computadoras.
- Hypertext es simplemente un documento de texto, el cual tiene hyperlinks (links) que redireccionan hacia otros documentos.
- Para poder definir esos links en el documento, se desarrolló un lenguaje para marcar dichos elementos (lenguaje de marcador). El lenguaje se llamó Hypertext Markup Language (HTML).



HTTP PROTOCOL

HYPertext

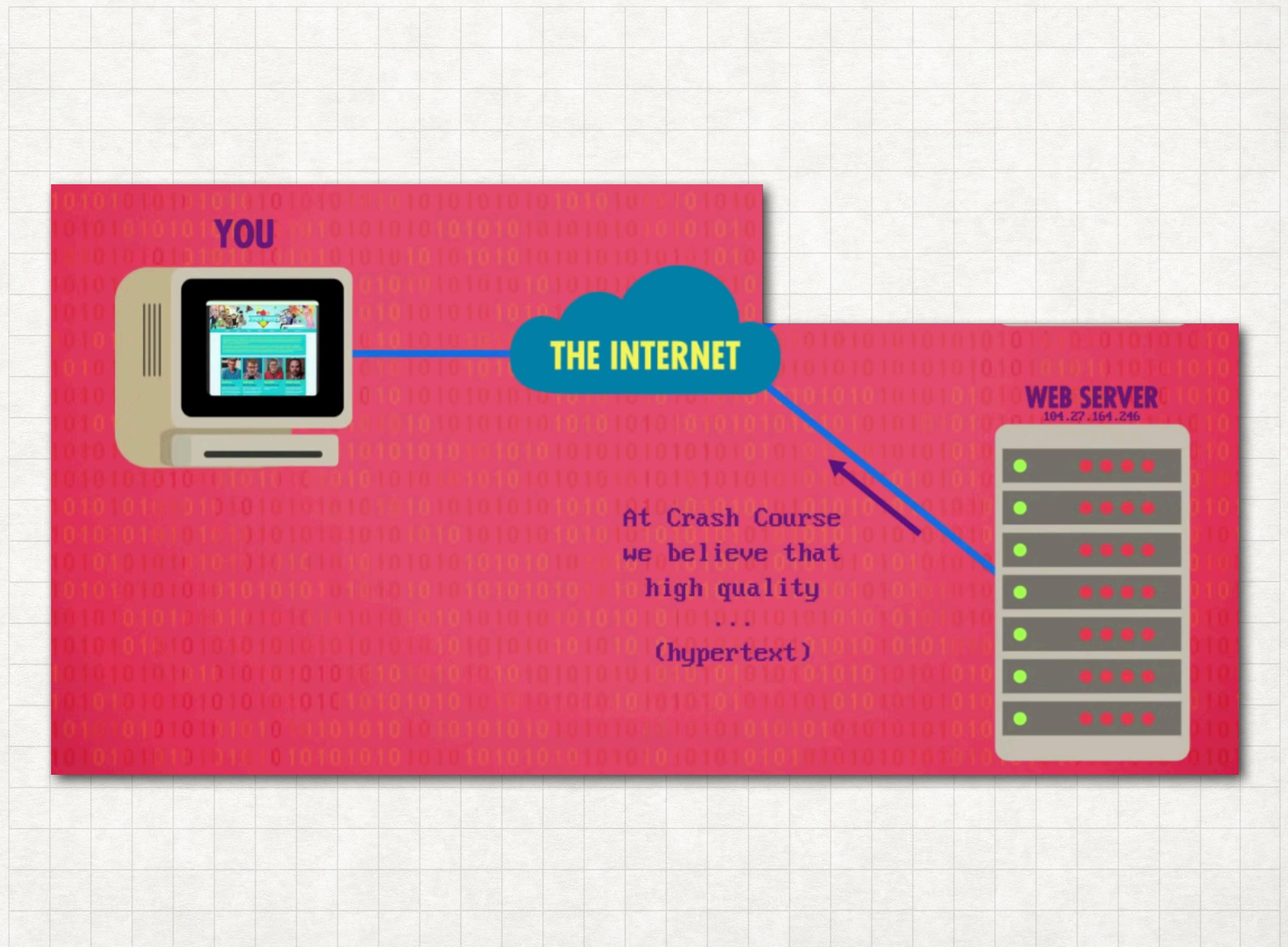
- El documento (o también conocido como página) que se quiere acceder, está referenciado por una URL (Uniform Resource Locator). Adicionalmente se indica una operación (o método) que se desea realizar, por ejemplo GET.
- HTTP está construido sobre TCP!!!
- Para hacer la comunicación, la computadora necesita especificar los datos de TCP (IP y Port), además de los datos de HTTP (URL y operación).



HTTP PROTOCOL

WEB SERVER

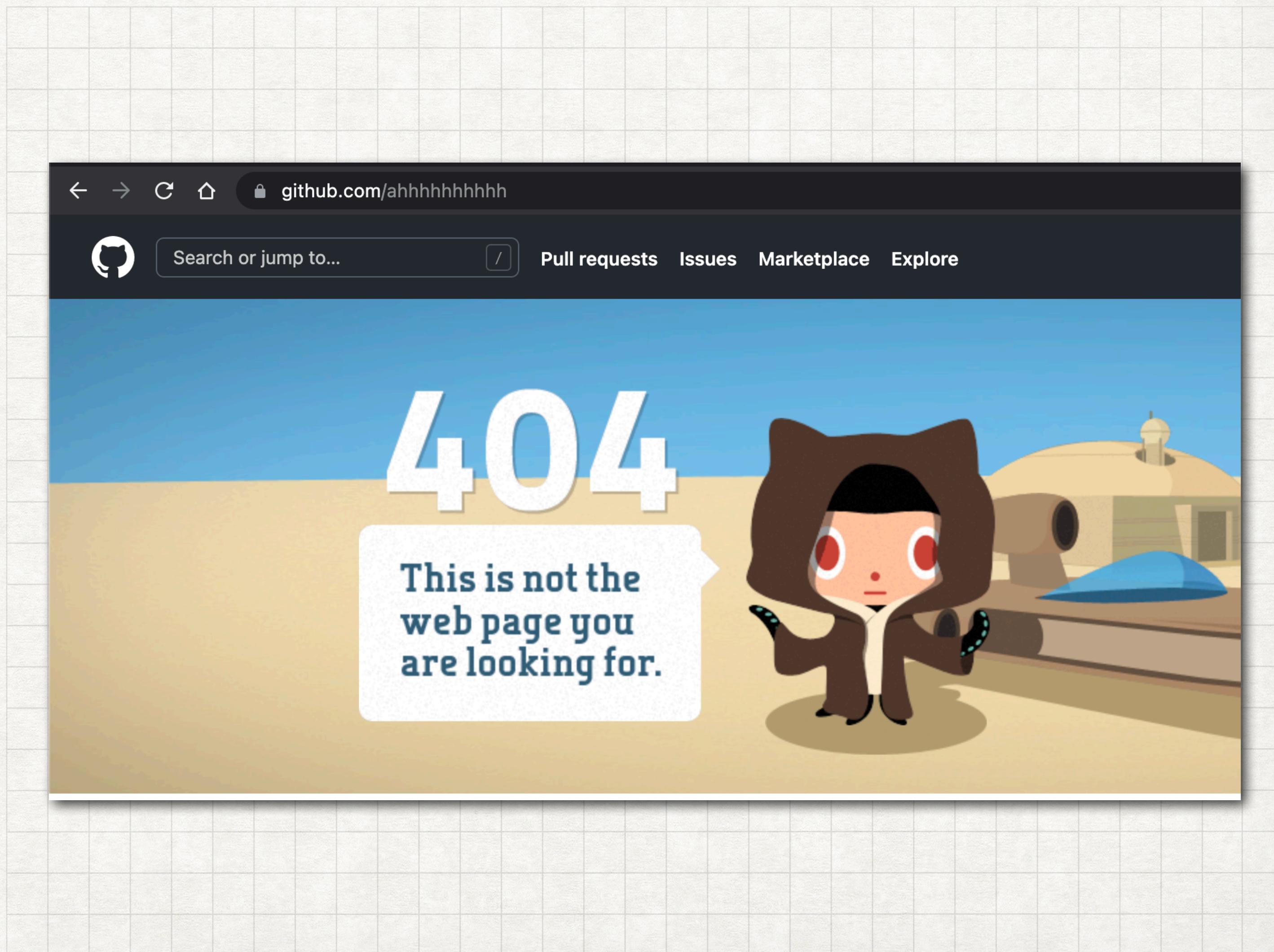
- El Web Server es un programa instalado en una computadora, que se encarga de devolver el archivo correspondiente a la petición HTTP que recibe.
- En otras palabras, devuelve el documento de texto correspondiente a la URL .
- Al ser un programa, tiene un puerto asignado (por defecto, 80)
- Adicionalmente al documento, también devuelve otra información, como un status code, que indica el resultado de la operación. (Por ejemplo, 404, documento no encontrado).
- Existe mucha otra data transferida en una petición HTTP, lo cual iremos viendo durante el curso.



HTTP PROTOCOL

WEB BROWSER

- El Web Browser es la aplicación en tu computadora que te permite obtener estos documentos de hipertexto de un web server, dependiendo de la solicitud enviada:
 - El hostname/IP (GitHub.com)
 - La URL (/ahhhhhhhhhh)
 - La operación/método (GET por defecto)
 - El puerto (80 por defecto)
- El web browser asimismo, se encarga de mostrar (renderizar) dicho hypertext document en tu pantalla.
- En el curso usaremos otros programas para realizar estas peticiones HTTP (peticiones de hipertexto a un Web Server, también conocido como HTTP request), entre ellos Postman.

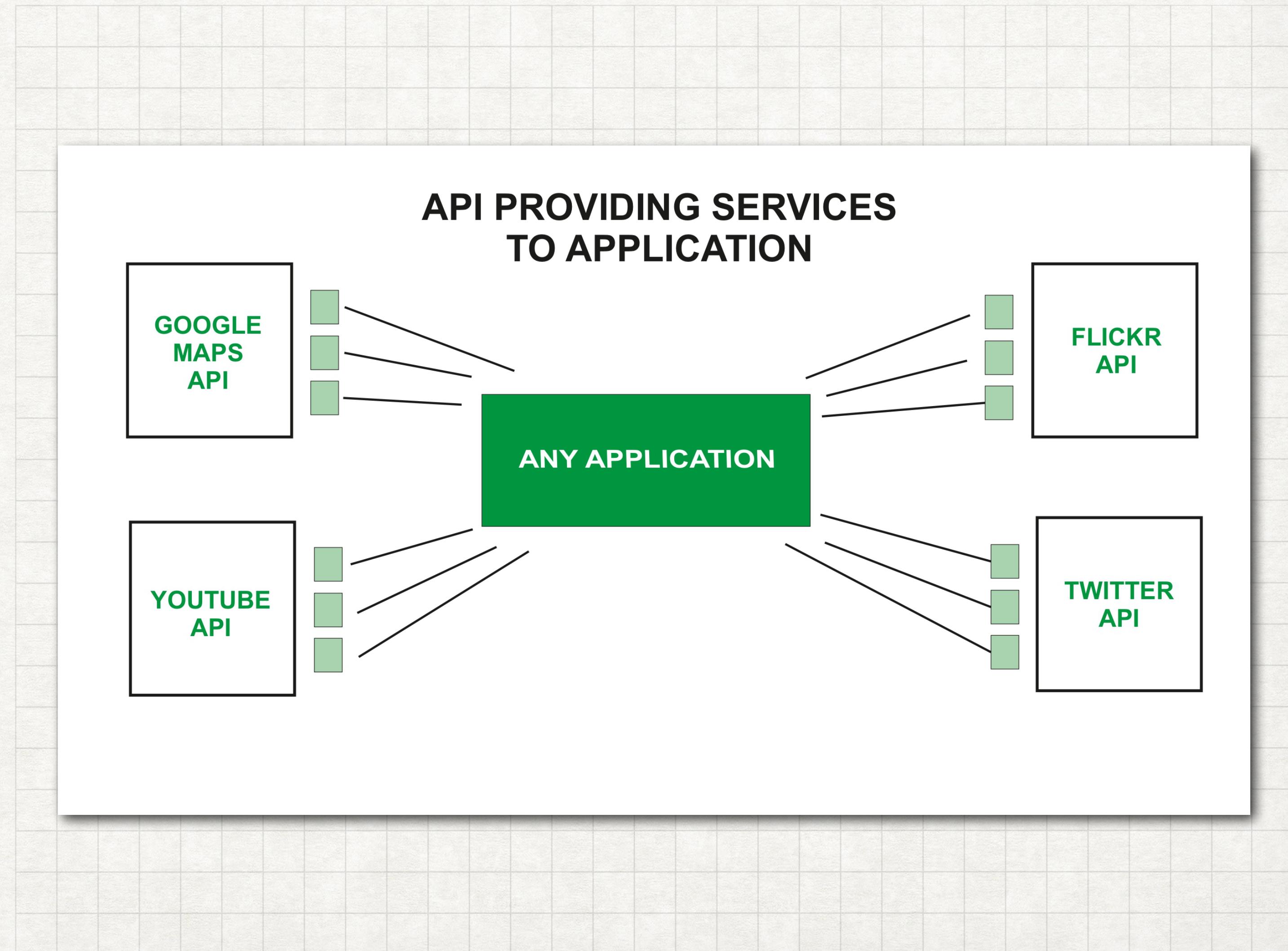


¿CÓMO ACCEDEMOS A LAS FUNCIONES DEL BACKEND?

¿CÓMO ACCEDEMOS A LAS FUNCIONES DEL BACKEND?

HTTP TO THE RESCUE (1)

- La forma en que usted “expone” sus funciones, y le permite a otros programas acceder a ellas, es conocido como la API (Application Programming Interface) de su proyecto.
- Por ejemplo, Facebook tiene/tenía una API que le permite a programadores acceder a sus funciones que muestran información de usuarios, páginas, eventos, etc.
- Un tipo de API son conocidas como Web API, en la cual sus funciones son accedidas a través de peticiones HTTP.
- La computadora donde sus funciones son almacenadas y ejecutadas es conocido como Application Server.



¿CÓMO ACCEDEMOS A LAS FUNCIONES DEL BACKEND?

HTTP TO THE RESCUE (2)

- Mediante la ayuda de un framework (librerías) para Web API, usted puede “ligar” URLs y método HTTP (e.g. GET) a alguna función que haya creado.
- Algunos ejemplos de frameworks para crear Web APIs son:
 - Flask
 - SpringBoot
 - Express
- Al “ejecutar” la definición de su API, el OS le asigna un puerto mediante el cual recibirá las peticiones HTTP.
- El framework se encarga de redireccionar la petición HTTP (HTTP request) a la función correspondiente.
- La definición de la Web API varía dependiendo del framework que use.

```
# api_definition.py

router.get('/users', getUsers)
router.get('/professors', getProfessors)
router.get('/courses', getCourses)
```

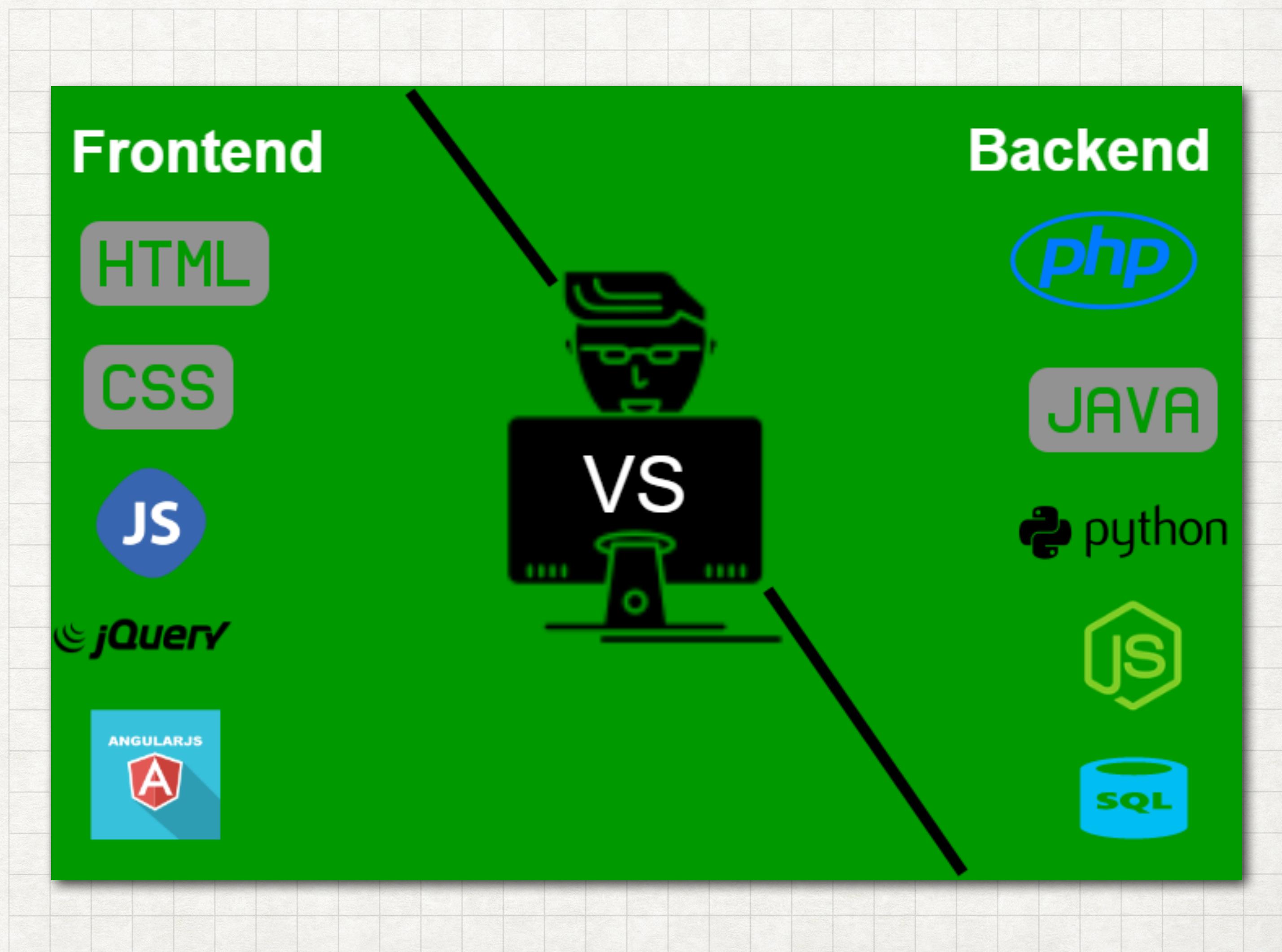
```
# users.py

def getUsers(httpData):
    return getUsersFromDatabase()
```

¿QUÉ SUCEDE EN EL FRONTEND?

¿QUÉ SUCEDE EN EL FRONTEND?

- El Frontend está encargado de la parte visual de la aplicación (móvil o web).
- Entre las funciones que realizan pueden estar incluidas:
 - Realizar llamadas al backend para obtener datos.
 - Mostrar los datos obtenidos desde el backend de manera clara.
 - Solicitar información al usuario.
 - Enviar datos al backend para su almacenaje y/o procesamiento.



CONEXIÓN ENTRE EL FRONTEND Y BACKEND

CONEXIÓN ENTRE EL FRONTEND Y BACKEND

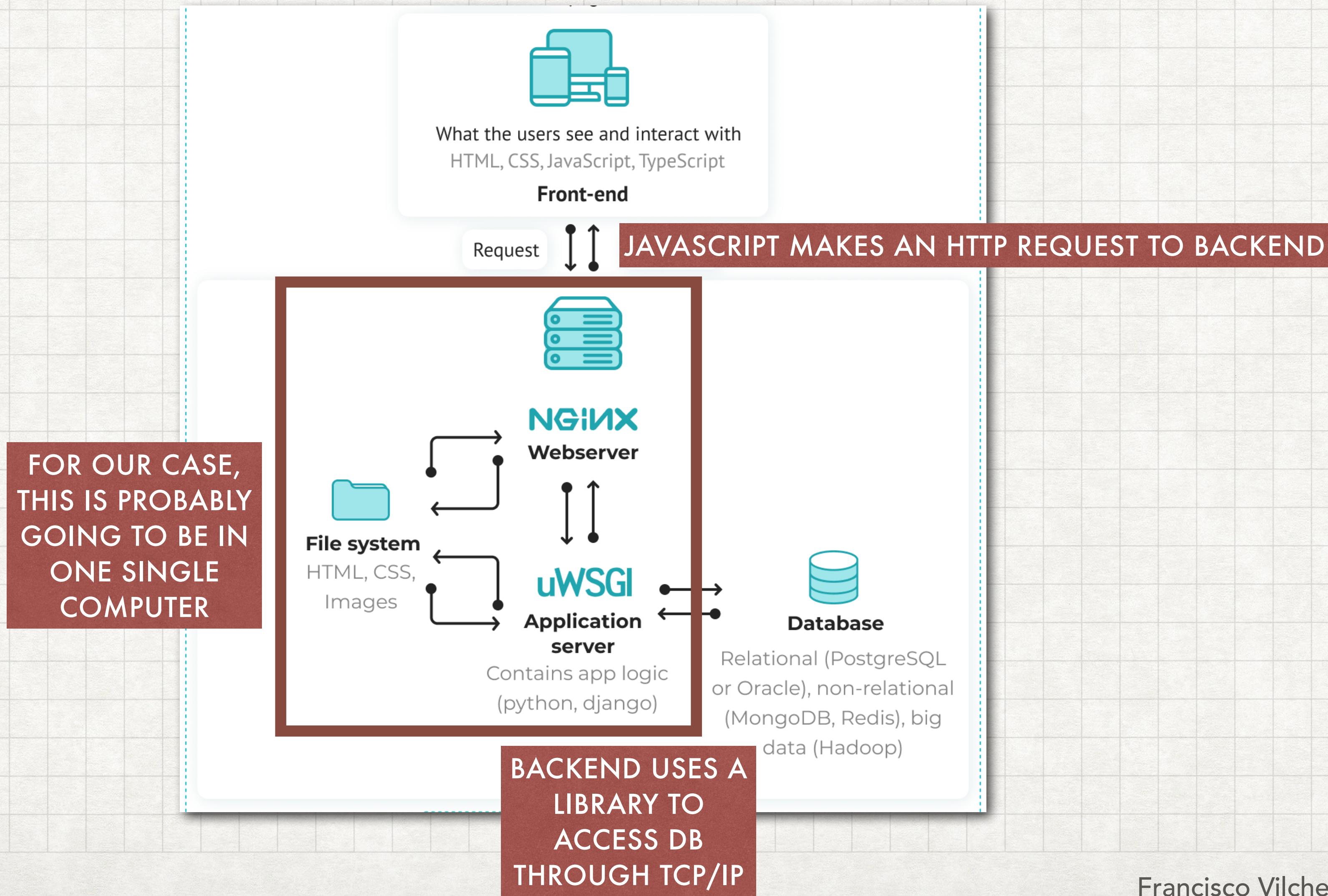
- Ya que nuestro backend API es accessible a través del protocolo HTTP, el Frontend debe realizar una petición HTTP para acceder a alguna de las funciones del backend.
- La información que debe detallar es:
 - Host/IP + PORT
 - Endpoint (URL)
 - Method: GET, POST, PUT, DELETE, PATCH, etc...
- La petición se hace desde el Javascript de nuestra página web. Dependiendo del framework utilizado, la sintaxis varía.

```
const Http = new XMLHttpRequest();
const url='https://jsonplaceholder.typicode.com/posts';
Http.open("GET", url);
Http.send();

Http.onreadystatechange=function(){
  if(this.readyState==4 && this.status==200){
    console.log(Http.responseText)
  }
}
```

ARQUITECTURA COMPLETA

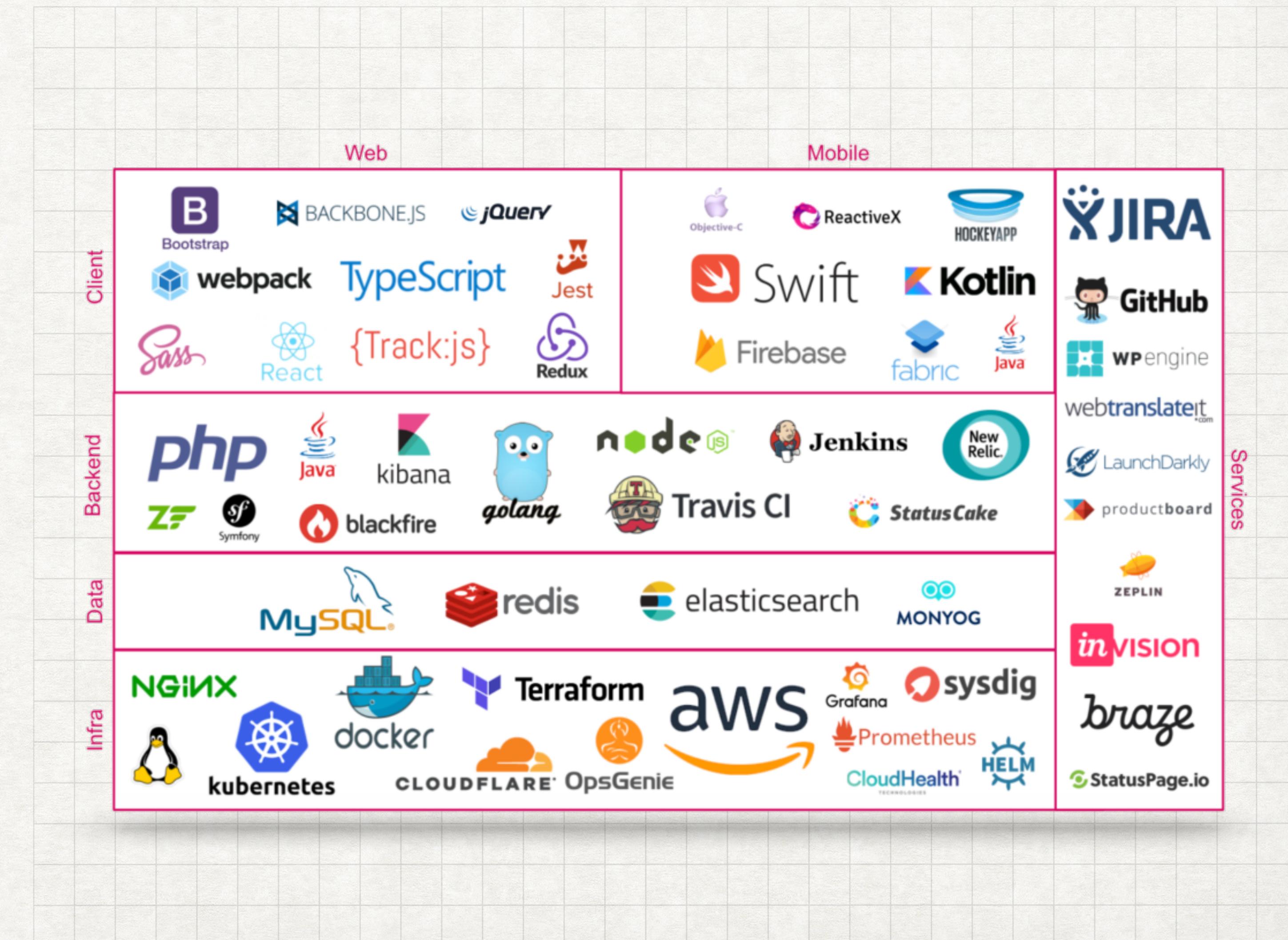
ARQUITECTURA COMPLETA WEB APPLICATION



APPLICATION STACK REVIEW

APPLICATION STACK

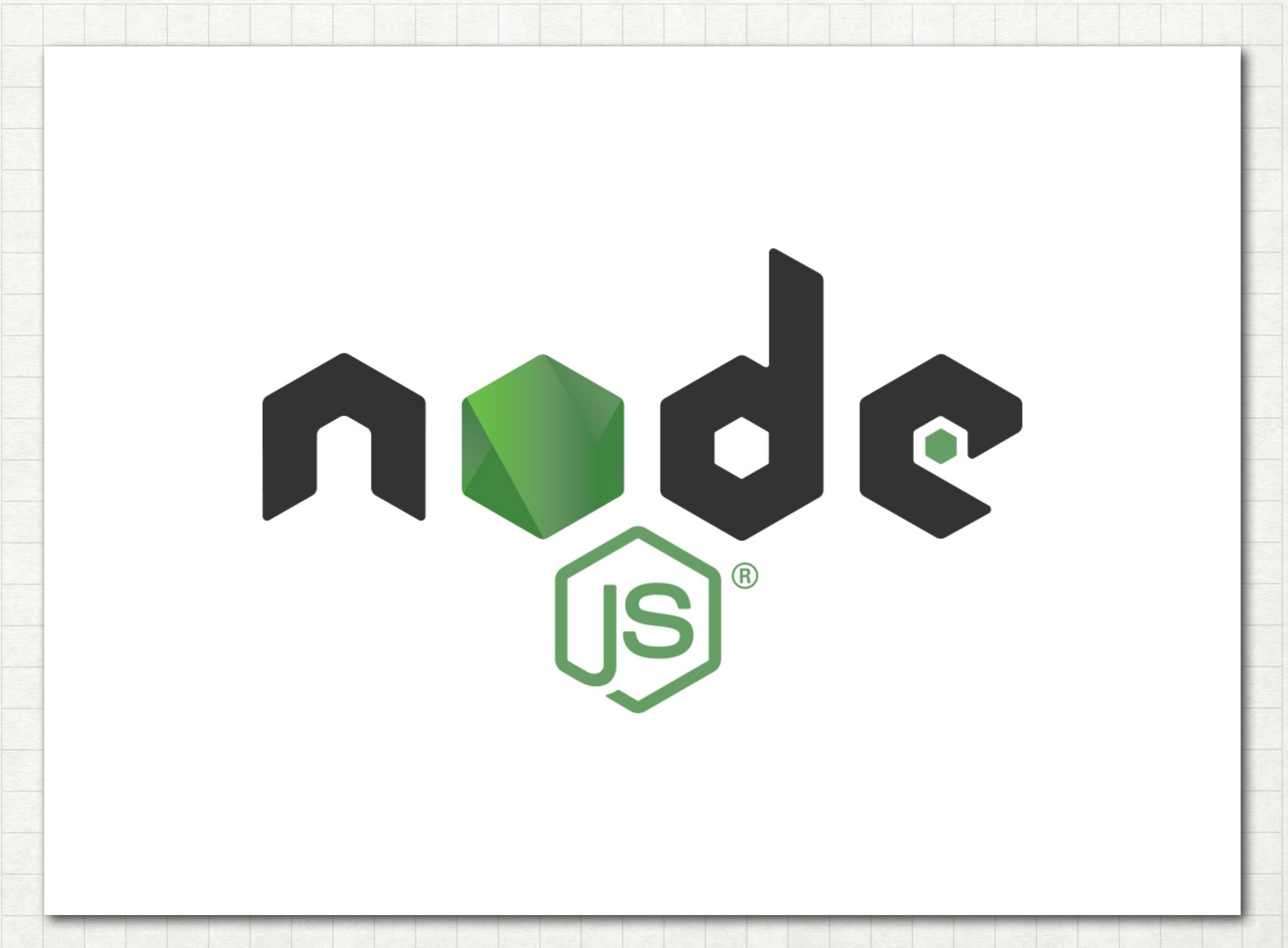
- Utilizamos el término `stack` para referirnos a las tecnologías utilizadas en la aplicación (web o mobile).
- En el lado back end, utilizaremos:
 - NodeJS (Javascript VM)
 - Express (NodeJS Framework)
- En el lado front end para Web, utilizaremos:
 - AngularJS (Framework JS, con Typescript)
 - Bootstrap (Framework CSS, opcional)
- Base de datos:
 - MySQL
- En el lado mobile, utilizaremos:
 - To be revealed in second part of the course.



NODEJS

NODEJS

- NodeJS es un “entorno de ejecución” para poder ejecutar Javascript de lado servidor.
- Inicialmente, Javascript era un lenguaje de programación sólo utilizado en el lado web, sin embargo, NodeJS permite ahora utilizar Javascript en el lado backend, de la misma manera que utilizamos Python, Java, C++, etc.
- De esta manera, los programadores no necesitan aprender un lenguaje adicional para programar sus funciones del lado backend, ya que pueden realizarla con sus conocimientos de Javascript.



EXPRESS

EXPRESS

- Express es un framework para NodeJS.
- Es utilizado para exponer funciones a través de HTTP.
- En términos más técnicos: Es utilizado para el desarrollo de Web APIs con NodeJS.
- Express te permite asignar un endpoint (URL) y un método HTTP a alguna función de javascript.
- Adicionalmente, express expone las funciones de tu proyecto a través de un puerto.



ANGULAR

ANGULAR

- Angular es un framework frontend para el desarrollo de aplicaciones web.
- Te brinda funciones ya creadas para que puedas fácilmente:
 - Manejar la navegación en tu aplicación
 - Comunicación con el backend
 - Modularizar tu código HTML, CSS y JS
 - Entre otras



TYPESCRIPT

TYPESCRIPT

- Typescript es un lenguaje de programación creado por Microsoft.
- En palabras simples, es Javascript, pero con algunas opciones adicionales, por ejemplo:
 - Poder declarar el tipo de dato a algunas de tus variables Javascript (aunque no es mandatorio)
 - Poder crear clases, interfaces, implementarlas, etc...
- La mayoría de frameworks Web usados actualmente utilizan Typescript.



BOOTSTRAP

BOOTSTRAP

OPCIONAL

- Bootstrap es un framework CSS.
- Te da un conjunto de components, estilos CSS y funciones Javascript ya creadas para poder agregar componentes visuales a tu aplicación web.



PREPARACIÓN PARA EL LAB

- Instalación de nvm y node LTS, y revisar los basics de NodeJS.
 - <https://github.com/franciscovilchezv/platform-based-development/tree/main/Labs/Lab4#before-the-lab>
- Instalación de Postman
 - <https://www.postman.com/downloads/>
- Repase la sintaxis de Javascript.

THANKS