

WOMEN WHO CODE® /medellín

FROM HERO TO SUPERHERO

BACKEND CON NODEJS

AGRADECER A...

Nuestro patrocinador y tutores voluntarios

PATROCINADOR

softserve

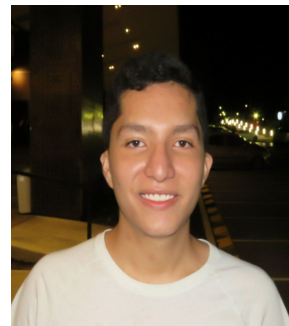
TUTORES VOLUNTARIOS



Edwin



Habid



Jean



Jose

TUTORES VOLUNTARIOS



**Habid Enrique
Manzur
Restrepo**

Soy ingeniero DevOps con más de 12 años de experiencia en el área de TI Actualmente lidero un equipo para una startup en USA.

Soy un gamer apasionado, mago de la automatización de procesos y me encantan los videojuegos, la música clásica y el buen heavy metal.

Cuando no estoy trabajando o jugando, disfruto de una buena taza de café negro, como mi alma y con mi cumpleaños acercándose, espero conseguir pronto una Xbox Series X de regalo.

NUESTROS VALORES

01 **PUNTUALIDAD**
El tiempo de todos es oro

02 **ORDEN**
Dejar todo mejor de lo que encontramos: limpio y ordenado.

03 **RESPETO**
Como invitados respetamos las reglas de nuestros anfitriones, en este caso Softserve

04 **COLABORACIÓN**
Apoyarnos y ayudarnos para terminar como un solo equipo

VIVIR LA CULTURA WWCODE

Que tus actos hablen más que tus palabras

01

Theoretical Lesson

CRUD orientado a los
principios REST

02

Features of the Topic

Manejo de BD PostgreSQL
para CRUD

03

TIPS

PostgreSQL
Sequelize
Sequelize-cli

04

PRACTICAL EXERCISES

Implementar una operación
CRUD con PostgreSQL

TABLE OF CONTENTS

Antes de empezar

Repasemos otras consultas SQL, pero esta vez es personal...

Queries Avanzadas

Volvamos al [playground](#) y ejecutemos más [queries](#)

A young couple is walking outdoors on a sunny day. The woman, on the left, has long blonde hair and is wearing a white t-shirt with a blue denim jacket over her shoulders. She is smiling and looking towards the man. The man, on the right, has a beard and is wearing a bright yellow t-shirt. He is also smiling and looking at the woman, with his hand slightly raised as if in conversation. The background is a blurred outdoor setting with trees and buildings.

Hora del break

GLOSARIO

Repasemos términos que usaremos antes de
iniciar

Glosario

WWW

01

Sistema global de información que permite el acceso y la compartición de recursos en **Internet** mediante **URLs**

HTTP

02

Protocolo para **transferir** información en la web entre **clientes** y **servidores**.

Middleware

03

Programa que se ejecuta entre la solicitud del cliente y la respuesta del servidor, y puede realizar una variedad de tareas, como autenticación, validación, registro de solicitudes, manejo de errores, entre otras.

CRUD

Orientado a los principios REST

¿QUÉ ES CRUD Y REST?

CRUD

Abreviación de **Crear**, **Leer**, **Actualizar** y **Eliminar**, las cuatro operaciones **básicas** que se pueden realizar en una **base de datos** para interactuar con los datos.

REST

Conjunto de **principios** para **diseñar** y **desarrollar** aplicaciones web **escalables** y **flexibles**. Se basa en la idea de utilizar recursos identificables con **URLs únicas** y realizar operaciones a través de verbos **HTTP** estandarizados.

REST es CRUD

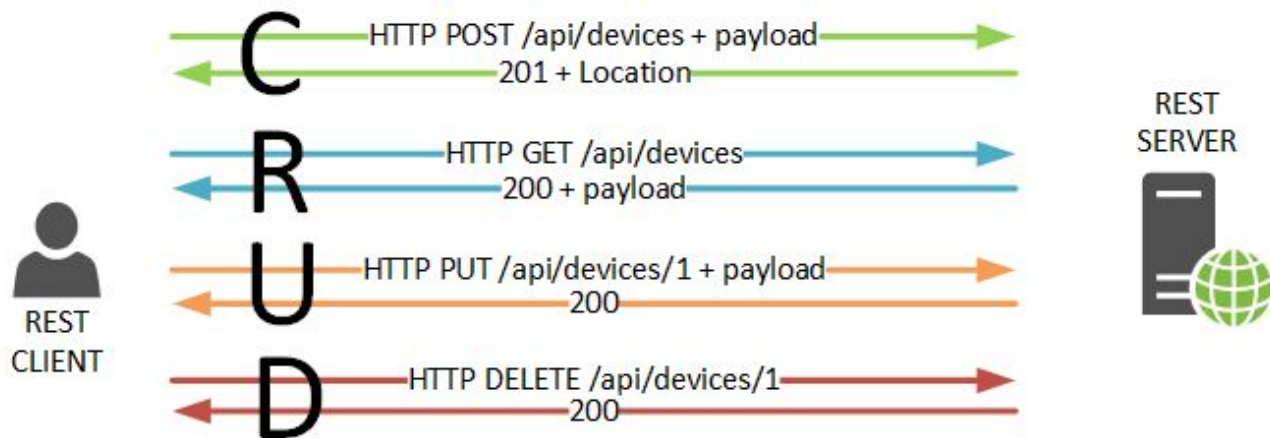
- Se usa para **exponer** de forma **segura** una Base de datos en la **web**
- **Sencillo** y simple
- Se trata de privilegiar **protocolos** y **recursos** sobre APIS

HTTP	SQL	CRUD
POST	INSERT	Create
GET	SELECT	Read
PUT / PATCH	UPDATE	Update
DELETE	DELETE	Delete

EJEMPLOS DE USO

GET	<code>/programming-languages</code>	→ <code>getMultiple()</code>
GET	<code>/programming-languages/:id</code>	→ <code>getOne(id)</code>
POST	<code>/programming-languages</code>	→ <code>create()</code>
PUT	<code>/programming-languages/:id</code>	→ <code>set(id)</code>
PATCH	<code>/programming-languages/:id</code>	→ <code>partialUpdate(id)</code>
DELETE	<code>/programming-languages/:id</code>	→ <code>remove(id)</code>

EJEMPLOS DE USO



“

Recuerda que esto no es camisa de fuerza y siempre puedes modificar el diseño para usar nuevos recursos si se quedan sin métodos HTTP

— EL PROFE



MVC

¿Con qué se come?

Arquitectura MVC

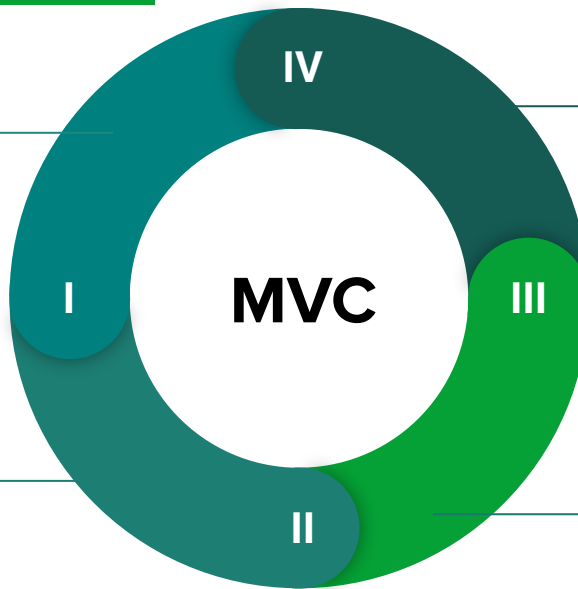
Cliente REST

Solicita información al servidor y recibe los resultados

Controlador

Se encarga de manejar las interacciones del usuario y coordinar el flujo de datos entre el modelo y la vista.

Contiene métodos o funciones que reciben las solicitudes del usuario, acceden al modelo para obtener o actualizar datos, y renderizar la vista adecuada.



Vista

Representa la capa de presentación de la aplicación.

Se encarga de la interfaz de usuario y la visualización de los datos.

Contiene archivos de plantilla, como HTML, CSS y JavaScript, que definen la apariencia y comportamiento de la interfaz.

Modelo

Representa la capa de datos de la aplicación.

Se encarga de manejar la lógica de negocio y el acceso a los datos.

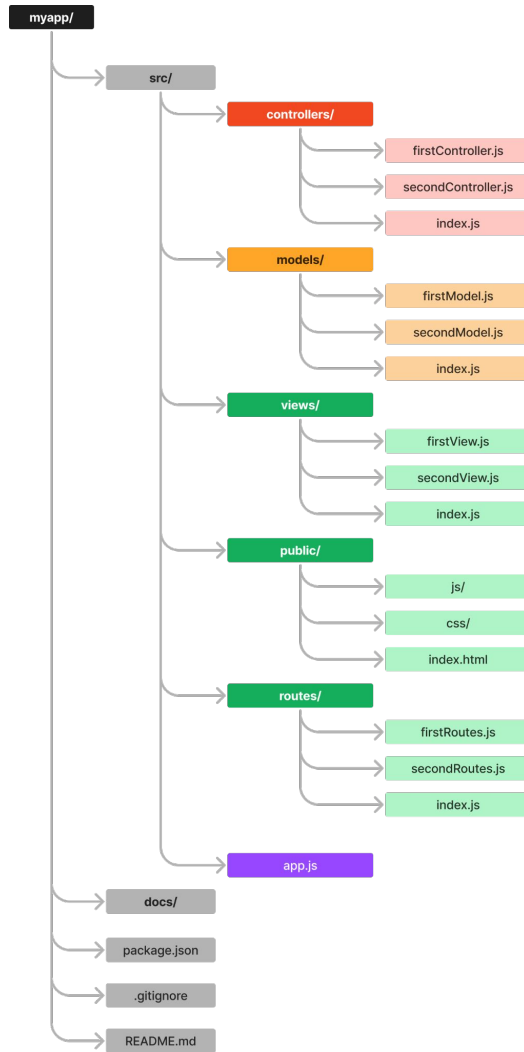
Contiene clases o estructuras que representan los objetos de datos y métodos para manipularlos.

ESTRUCTURA DE CARPETAS

Sí, por fin

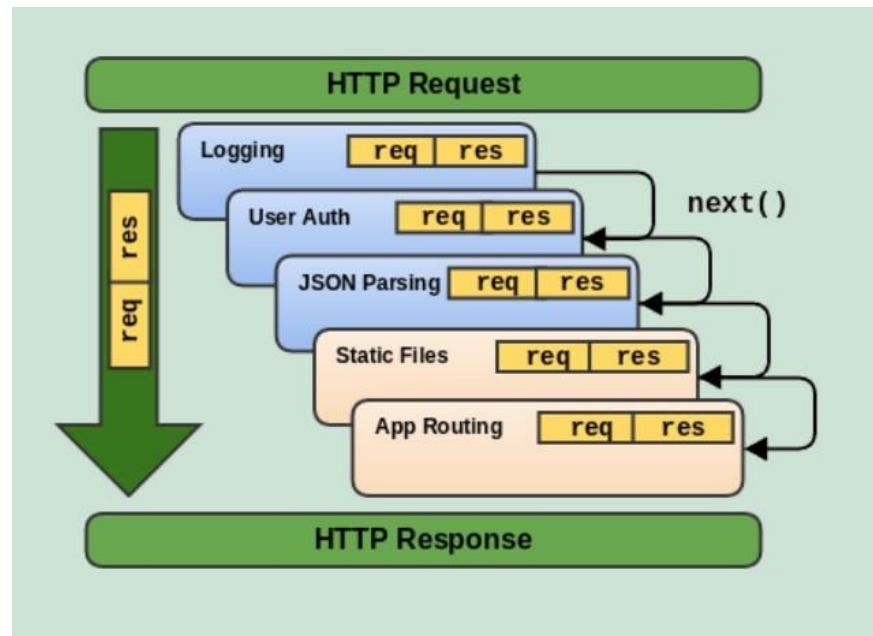
ESTRUCTURA

La arquitectura **Modelo-Vista-Controlador (MVC)** separa las diferentes **responsabilidades** de una aplicación en **tres** capas **principales**, lo que facilita la **gestión** y **mantenimiento** del código, y permite una mayor **reutilización** y **escalabilidad** del mismo.



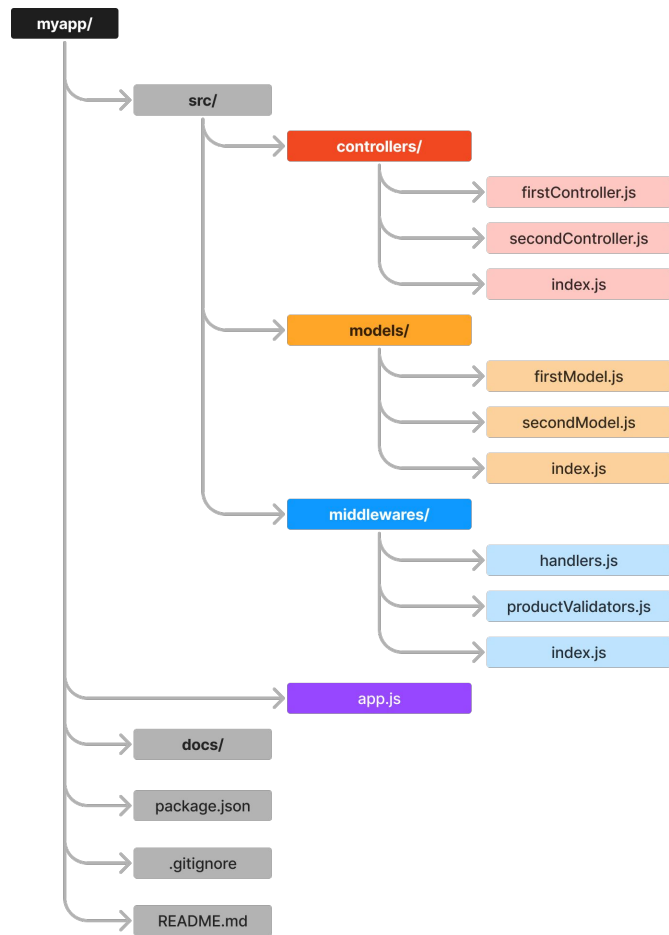
Middlewares

En el contexto de una aplicación Node.js que utiliza el patrón MVC, los middlewares se utilizan comúnmente para realizar tareas específicas en la capa del controlador antes de que los datos se procesen en la capa del modelo o se envíe como respuesta al cliente.



¡Para tener en cuenta!

La base a usar en el proyecto es la presentada en esta diapositiva, la anterior es de ejemplo y también separando los **middlewares** (errores, not found, joi, etc) en su propia carpeta



Taller en clase

Implementemos un CRUD avanzado con PostgreSQL

TAREA

1. Traer el proyecto actualizado con la estructura de carpetas para la clase 6 o al menos las carpetas ya creadas.
2. Para la clase 7, investigar patrones de diseño más usados en NodeJs y tener el proyecto al día con todo lo visto hasta ahora.