

ACÀMICA

Agenda

Daily

Programo: ORM

Buenas prácticas

Break

Programamos: Venta de televisores

Programan: Venta de televisores

Cierre



ORM

El ORM (Object Relational Mapping) es una técnica de programación que permite representar datos de una base de datos en objetos.

Daily



Daily



Sincronizando...

Toolbox



¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Challenge

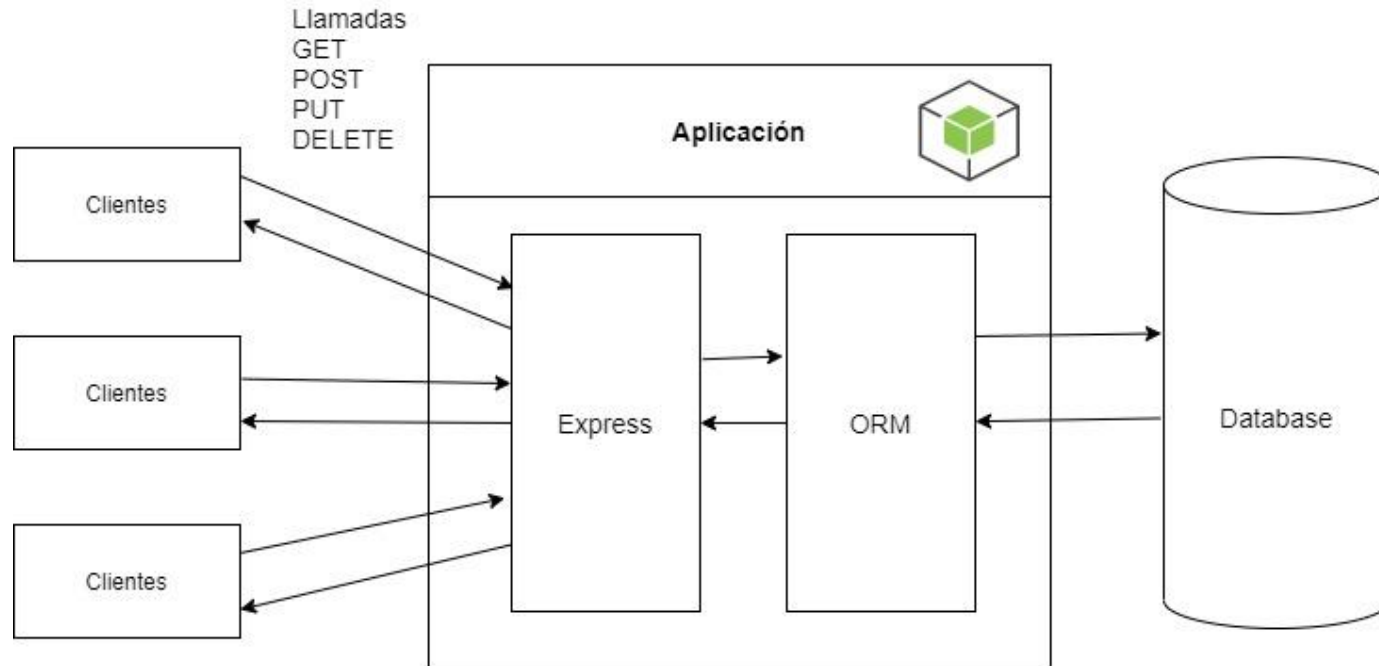


¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Teoría **ORM**



Mediante el **ORM** añadimos una capa extra con nuestra DB.





Instalar nuestro servidor

La instalación es sencilla, consiste de solo 3 pasos:

1. **Crea** la carpeta del proyecto, por ejemplo **server**
`mkdir server`
`cd server`
2. **Inicia** un proyecto Node
`npm init`
3. **Instala** Sequelize
`npm i sequelize`
4. **Instala** el cliente de tu DB
`npm i mysql2`



Configura tu conexión a la DB

```
Terminal  Help  • app.js - Untitled (Workspace) - Visual Studio Code

JS app.js  ●

server > JS app.js > ...
1  const { Sequelize, DataTypes, Model } = require('sequelize');
2  const sequelize = new Sequelize('database', 'user', 'pass', {
3    |   host: 'localhost',
4    |   dialect: 'mysql'
5  });
6
7  //controla estar conectado
8  sequelize.authenticate().then(() =>{
9    |   console.log("Conexión ok")
10 | })
11 |   .catch(()=>{
12 |     console.log("Conexión fallida")
13 |   })
14
```

Crea tu modelo

```
14
15  class Usuarios extends Model {}
16
17  Usuarios.init({
18    nombre: DataTypes.STRING, // campo nombre de la BD
19    apellido: DataTypes.STRING // campo apellido de la BD
20  }, { sequelize, modelName: 'usuarios' } /* Tabla usuarios */
21  );
22
23
```



Realiza tus **operaciones**

```
23  
24   (async () => {  
25       await sequelize.sync();  
26       const datos = {nombre: 'Daniel', apellido: 'Segovia'}  
27       const got = await Usuarios.create(datos);  
28       console.log(got.toJSON());  
29   })();  
30
```



Inicia tu proyecto



```
MINGW64:/c:/Users/Daniel Segovia/Desktop/DWBE/server

Daniel Segovia@DESKTOP-RDG4RBT MINGW64 ~/Desktop/DWBE/server
$ node app.js
Executing (default): SELECT 1+1 AS result
Executing (default): CREATE TABLE IF NOT EXISTS `usuarios` (`id` INTEGER NOT NULL auto_increment, `nombre` VARCHAR(255), `apellido` VARCHAR(255), `createdAt` DATETIME NOT NULL, `updatedAt` DATETIME NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB;
Conexión ok
Executing (default): SHOW INDEX FROM `usuarios`
Executing (default): INSERT INTO `usuarios` (`id`,`nombre`,`apellido`,`createdAt`,`updatedAt`) VALUES (DEFAULT,?,?,?,?,?);
{
  id: 5,
  nombre: 'Daniel',
  apellido: 'Segovia',
  updatedAt: 2021-01-10T20:24:14.098Z,
  createdAt: 2021-01-10T20:24:14.098Z
}
```

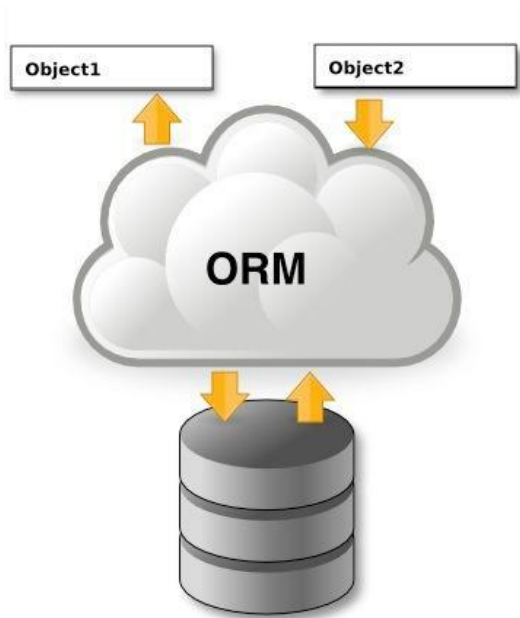
Programo

squad leads



ORM con Sequelize

Veamos a una demostración en vivo de cómo trabajar con este ORM.



Buenas prácticas



Base de datos grandes

Utilizar ORM es muy ágil para tus proyectos pero si tu base de datos es muy grande será costoso para el servidor mapear tu DB. ¡Tenlo en cuenta a la hora de elegir utilizarlo!



A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup is placed on a matching white saucer. In the background, a white napkin and a silver spoon are visible, though they are out of focus. The overall lighting is soft and even, highlighting the textures of the coffee and the smooth surface of the cup.

¡BREAK!



Programamos

todos/as



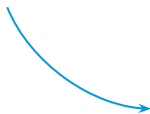
Programamos

Programemos la siguiente estructura:

Crea una API con dos rutas y utiliza ORM para trabajar con la base de datos:

- 1 La primera tipo **GET** que muestre todos los televisores en venta.
- 2 La segunda tipo **POST** para agregar un televisor.

Modelo de datos



Marca
Modelo
Pantalla
Smart (boolean)
Precio

Programman

trainees

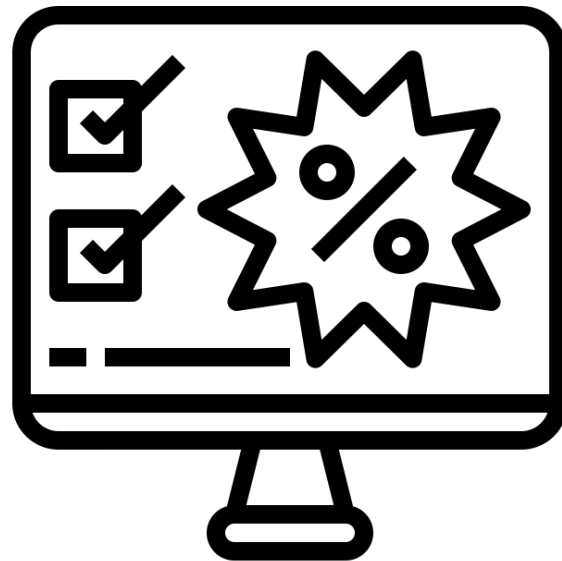


Programan

Venta de televisores

Se acercan las fiestas de fin de año y LocalTV necesita ayuda con la organización de su base de datos para organizar la venta de televisores.

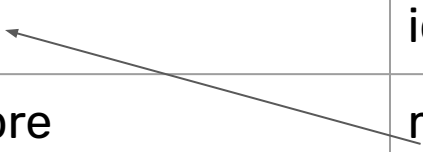
¡Ayúdalos!



Relación

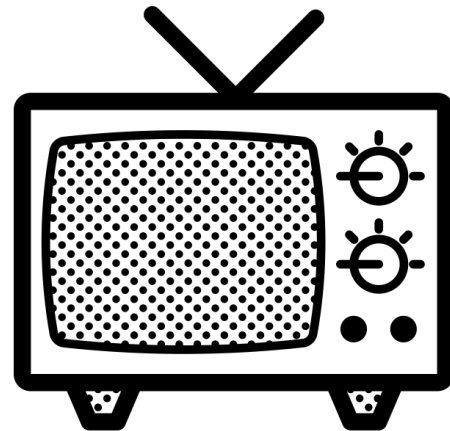
Si no lo haz hecho aún, genera una relación entre las tablas:

marcas	modelos
id	id
nombre	marca_id
	nombre



Todos los televisores de una marca

Crea un Endpoint que reciba la marca de un televisor y retorne todos los modelos.



Precios

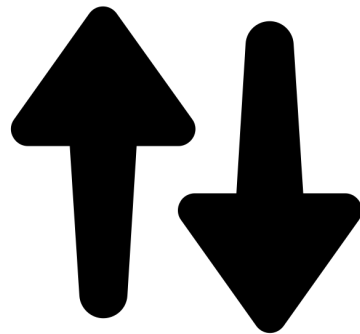
Crea 2 Endpoints, donde ambos reciban un parámetro de un número entero:

- El primer Endpoint debe devolver todos los televisores de precio más alto que el parámetro;
- El segundo debe devolver los televisores de menor precio al parámetro.



Order By

1. Googlea cómo hace un/a desarrollador para encontrar cómo ordenar los resultados a través del ORM.
2. Crea un Endpoint que ordene todos los televisores de menor a mayor precio.



Sprint project #2:

Mi primera API

Persistente



Sprint project #2

**¡Genera la conexión a
la DB en tu proyecto !**



Para la próxima

- 1) Termina el ejercicio de la meeting de hoy.
- 2) Lee la toolbox 27.
- 3) Resuelve el challenge.

ACÀMICA