

# Bootcamp de Full Stack

Bienvenidos a la clase N°56

- Models
- ORM & ODM
- Mongoose:
  - Instalación
  - Configuración
- Implementación de DB en API
- Desafío Semanal F3-N18

# Backend

## Models

### ¿Qué son los Models en Bases de Datos?

Un model (o modelo) es una representación de la estructura de los datos que vamos a almacenar en la base de datos. Define qué campos tendrá cada registro, qué tipo de datos contendrá cada campo y a veces, reglas de validación o restricciones. En otras palabras, un model es como un molde o plantilla para nuestros datos, que asegura que la información se almacene de manera consistente.

### ¿Qué es un Model en MongoDB?

En MongoDB, un model es la representación de un documento dentro de una colección. Aunque MongoDB es flexible y no exige estructura fija, usar models permite definir esquemas, validar datos y facilitar operaciones sobre la colección.

# Backend

## ORM & ODM

### ¿Qué es un ORM?

Un ORM (Object-Relational Mapping) es una herramienta que permite trabajar con bases de datos relacionales usando objetos en tu lenguaje de programación. En lugar de escribir SQL, puedes crear, leer, actualizar y borrar registros mediante objetos y métodos, y el ORM se encarga de traducir esas operaciones a consultas SQL.

### ¿Qué es un ODM?

Un ODM (Object-Document Mapping) es similar al ORM, pero se utiliza en bases de datos NoSQL orientadas a documentos, como MongoDB. Permite trabajar con objetos en código que representan documentos, realizando operaciones como si fueran objetos normales, mientras el ODM se encarga de almacenarlos en la base de datos.

# Backend

## Mongoose

### ¿Qué es MongoDB Client?

Un client es la instancia que permite conectarse a la base de datos desde tu aplicación. Actúa como un puente entre tu código y MongoDB, y se encarga de:

- Establecer la conexión con el servidor de MongoDB o el clúster en la nube (Atlas).
- Enviar las operaciones (consultas, inserciones, actualizaciones, eliminaciones) a la base de datos.
- Recibir los resultados de esas operaciones y entregarlos a tu aplicación.

### ¿Qué es Mongoose?

Mongoose es un ODM para MongoDB en Node.js que permite definir esquemas y modelos, validar datos y realizar operaciones de manera sencilla y segura. Gracias a Mongoose, los documentos de MongoDB se pueden manejar como objetos de JavaScript, lo que simplifica tanto el desarrollo como la gestión de la base de datos. Cabe mencionar que, Mongoose no es un client en sí mismo, pero internamente usa un client de MongoDB para conectarse a la base de datos.



Documentación oficial de Mongoose <https://mongoosejs.com/docs/guide.html>

# Backend

## Implementación de DB en API

### Consigna

- Implementar una base de datos para la entidad Car dentro de la API REST que venimos desarrollando en Node.js. Emplear MongoDB como motor de base de datos y Mongoose como ODM. El objetivo es permitir crear, leer, actualizar y eliminar registros de coches de manera adecuada, siguiendo las buenas prácticas de desarrollo. Para lograrlo, será necesario definir el modelo de datos con Mongoose, configurar las rutas correspondientes en Express y desarrollar la lógica en los endpoints para gestionar cada una de las operaciones del CRUD.

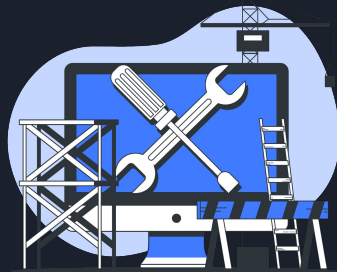
# BREAK

Descansemos 10 minutos



# Desafío Semanal

F3-N18



- **Aclaración:** Esta actividad es de **carácter opcional** y se lleva a cabo de forma **individual** o en **parejas de estudiantes**, fuera del horario de clase.
- **Evaluación:** La realiza el **tutor**.
- **Fecha límite:** La **entrega** se debe realizar antes de iniciar la **siguiente clase**.
- **Consigna y recursos:** El siguiente enlace contiene todo lo necesario para el desarrollo de la actividad:  
[https://drive.google.com/drive/folders/1fbOGaiGMaFmtRuQOgfcDU5p\\_DNhwKP7?usp=drive\\_link](https://drive.google.com/drive/folders/1fbOGaiGMaFmtRuQOgfcDU5p_DNhwKP7?usp=drive_link)
- **Entrega:** Se hace de forma individual en formato ZIP y se sube a la carpeta personal de Google Drive.
- **Registro de entrega:** Se debe registrar en este formulario: <https://forms.gle/VeRUKYcfjCtv4afXA>

# CIERRE DE CLASE

Continuaremos en la próxima clase

