

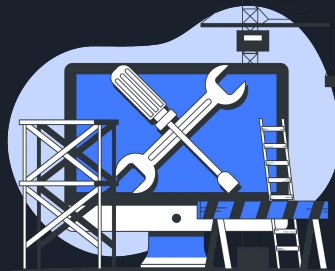
# Bootcamp de Full Stack

Bienvenidos a la clase N°59

- Desafío Semanal N°18
- Operadores de MongoDB:
  - Regex
- CRUD con Express y MongoDB
- Validación de parámetros

# Desafío Semanal N°18

## Pair Programming



- Esta actividad se resuelve **fuera de clase** entre dos estudiantes y es de **carácter obligatorio**.
- La misma, se **aprueba** con **5 o más** y es calificada por nuestros tutores.
- En el caso de obtener una calificación **menor a 5**, se debe **re-entregar** dicha actividad.
- La fecha límite para **entregar** es el Miércoles **20/03/2024**.
- El siguiente link, contiene la consigna y los recursos necesarios para su desarrollo:
- [https://drive.google.com/drive/folders/1HpfZGSjXXhJDdTtDj0womdna2RK3E-Mk?usp=drive\\_link](https://drive.google.com/drive/folders/1HpfZGSjXXhJDdTtDj0womdna2RK3E-Mk?usp=drive_link)
- La entrega se realiza de forma individual en formato zip y se sube en la carpeta personal de Google Drive.
- Por último y no menos importante, registrar la entrega en el form que recibieron por mail.

# Backend

## Operadores de MongoDB

Documentación de MongoDB: <https://www.mongodb.com/docs/manual/reference/operator/query/regex/>

Operadores de consulta + expresiones regulares:

- Obtener coches en donde la marca comienza con la letra R:  
`db.coches.find({ marca: { $regex: "^R" } })`
- Obtener coches en donde la marca finaliza con las letras at:  
`db.coches.find({ marca: { $regex: "at$" } })`
- Obtener coches en donde la marca contenga la letra o:  
`db.coches.find({ marca: { $regex: "o" } })`
- Obtener coches que coincidan con la marca foRd sin diferenciar entre mayúsculas y minúsculas:  
`db.coches.find({ marca: { $regex: "foRd", $options: "i" } })`



# Backend

## CRUD con Express y MongoDB

Consigna:

- Instalar la dependencia de MongoDB: `npm install mongodb --save`
- Crear conexión
- Utilizar variables de entorno, controllers y routers
- Implementar los métodos HTTP get, post, put y delete
- Emplear una validación del esquema de datos

# BREAK

Descansemos 15 minutos



# Backend

## Validación de parámetros

### ¿Qué es Joi?

Joi es una biblioteca de validación de datos utilizada comúnmente para validar la entrada de usuario y otros datos en una aplicación. Permite definir un esquema que describe la estructura y restricciones de los datos que se esperan recibir o manipular. Estas restricciones pueden incluir tipos de datos, valores mínimos y máximos, patrones de texto, y más.

Documentación oficial: <https://joi.dev/api/?v=17.12.2>

Consigna:

- Implementar Joi como middleware

# CIERRE DE CLASE

Continuaremos en la próxima clase

