

Bootcamp Backend Code Review N° 1

🚩 Introducción

La fase de desarrollo no culmina simplemente con la escritura de código. Es por eso que además del sprint, sumamos esta instancia de code review como parte de tu proceso global de aprendizaje.

🤔 ¿En qué consiste este "Code Review Day"?

Es un espacio de 10 a 15 minutos de duración, en el que te vas a encontrar por meet con uno de los facilitadores del Bootcamp para discutir y presentar una solución a uno de los ejercicios propuestos sobre una concesionaria de autos; que se detallan más abajo.

Durante este proceso, no solo tendrás que dar una solución al problema, sino también explicarla, justificar tus decisiones y responder a preguntas técnicas o conceptuales que puedan surgir.

🎯 El objetivo de esta instancia es doble:

- Permitirte demostrar tu capacidad técnica y tu comprensión de los conceptos aprendidos.
- Darte una idea aproximada e hipotética de cómo se realizan las revisiones de código en entornos profesionales reales, preparándote para situaciones similares en el mundo laboral.

🛠️ Planteo del problema

Estás a cargo de desarrollar una API para una concesionaria de vehículos. Cada vehículo 🚗 en la concesionaria posee detalles únicos como:

- identificador
- marca
- modelo
- matrícula
- color
- año de fabricación
- velocidad máxima
- capacidad de personas

- tipo de combustible
- tipo de transmisión
- dimensiones: largo y ancho
- peso

Para facilitar tu trabajo, ya cuentas con un proyecto base. En este, se ha preestablecido la estructura de datos y el almacenamiento correspondiente. Además, ya se ha codificado el endpoint para **listar todos los vehículos**.

Listar todos los vehículos

- **Como:** Usuario de la API.
- **Quiero:** Obtener una lista de todos los vehículos.
- **Para:** Visualizar el inventario completo.
- **Endpoint:** GET /vehicles
- **Respuestas:**
 - 200 OK: Devuelve una lista con todos los vehículos.
 - 404 Not Found: No hay vehículos en la concesionaria.

Tu tarea principal es mejorar y expandir esta API para que gestione eficientemente toda la información de los vehículos.

A continuación dependiendo en qué Bootcamp estes, puedes encontrar el proyecto base a utilizar:

- Proyecto Base para Bootcamp JAVA: [Scaffolding Java](#)

A continuación, te presento una lista de los endpoints que podrías implementar:

1. Añadir un vehículo

- **Como:** Usuario de la API.
- **Quiero:** Registrar un nuevo vehículo.
- **Para:** Ampliar el inventario de la concesionaria.
- **Endpoint:** POST /vehicles
- **Respuestas:**
 - 201 Created: Vehículo creado exitosamente.
 - 400 Bad Request: Datos del vehículo mal formados o incompletos.
 - 409 Conflict: Identificador del vehículo ya existente.

2. Buscar vehículos por color y año

- **Como:** Usuario de la API.
- **Quiero:** Listar vehículos por color y año.
- **Para:** Filtrar vehículos basados en esas especificaciones.
- **Endpoint:** GET /vehicles/color/{color}/year/{year}
- **Respuestas:**
 - 200 OK: Devuelve una lista de vehículos que cumplen con los criterios.
 - 404 Not Found: No se encontraron vehículos con esos criterios.

3. Buscar vehículos por marca y rango de años

- **Como:** Usuario de la API.
- **Quiero:** Listar vehículos de una marca específica fabricados en un rango de años.
- **Para:** Realizar búsquedas más detalladas en el inventario.
- **Endpoint:** GET /vehicles/brand/{brand}/between/{start_year}/{end_year}
- **Respuestas:**
 - 200 OK: Devuelve una lista de vehículos que cumplen con los criterios.
 - 404 Not Found: No se encontraron vehículos con esos criterios.

4. Consultar velocidad promedio por marca

- **Como:** Usuario de la API.
- **Quiero:** Conocer la velocidad promedio de vehículos de una marca específica.
- **Para:** Comparar las prestaciones de diferentes marcas.
- **Endpoint:** GET /vehicles/average_speed/brand/{brand}
- **Respuestas:**
 - 200 OK: Devuelve la velocidad promedio de la marca.
 - 404 Not Found: No se encontraron vehículos de esa marca.

5. Añadir múltiples vehículos

- **Como:** Usuario de la API.
- **Quiero:** Registrar varios vehículos al mismo tiempo.
- **Para:** Facilitar la carga masiva de datos en el inventario.

- **Endpoint:** POST /vehicles/batch
- **Respuestas:**
 - 201 Created: Vehículos creados exitosamente.
 - 400 Bad Request: Datos de algún vehículo mal formados o incompletos.
 - 409 Conflict: Algún vehículo tiene un identificador ya existente.

6. Actualizar velocidad máxima de un vehículo

- **Como:** Usuario de la API.
- **Quiero:** Modificar la velocidad máxima de un vehículo específico.
- **Para:** Mantener el inventario actualizado con las características correctas.
- **Endpoint:** PUT /vehicles/{id}/update_speed
- **Respuestas:**
 - 200 OK: Velocidad del vehículo actualizada exitosamente.
 - 400 Bad Request: Velocidad mal formada o fuera de rango.
 - 404 Not Found: No se encontró el vehículo.

7. Listar vehículos por tipo de combustible

- **Como:** Usuario de la API.
- **Quiero:** Obtener una lista de vehículos según el tipo de combustible.
- **Para:** Realizar búsquedas basadas en la fuente de energía del vehículo.
- **Endpoint:** GET /vehicles/fuel_type/{type}
- **Respuestas:**
 - 200 OK: Devuelve una lista de vehículos que usan ese tipo de combustible.
 - 404 Not Found: No se encontraron vehículos con ese tipo de combustible.

8. Eliminar un vehículo

- **Como:** Usuario de la API.
- **Quiero:** Borrar un vehículo de la concesionaria.
- **Para:** Mantener el inventario actualizado.
- **Endpoint:** DELETE /vehicles/{id}
- **Respuestas:**
 - 204 No Content: Vehículo eliminado exitosamente.
 - 404 Not Found: No se encontró el vehículo.

9. Buscar vehículos por tipo de transmisión

- **Como:** Usuario de la API.

- **Quiero:** Obtener una lista de vehículos basada en su tipo de transmisión (manual, automática, etc.).
- **Para:** Filtrar vehículos según mis preferencias de conducción.
- **Endpoint:** GET /vehicles/transmission/{type}
- **Respuestas:**
 - 200 OK: Devuelve una lista de vehículos con ese tipo de transmisión.
 - 404 Not Found: No se encontraron vehículos con ese tipo de transmisión.

10. Actualizar el tipo de combustible de un vehículo

- **Como:** Usuario de la API.
- **Quiero:** Modificar el tipo de combustible de un vehículo específico.
- **Para:** Mantener el inventario actualizado ante cambios en las características del vehículo.
- **Endpoint:** PUT /vehicles/{id}/update_fuel
- **Respuestas:**
 - 200 OK: Tipo de combustible del vehículo actualizado exitosamente.
 - 400 Bad Request: Tipo de combustible mal formado o no admitido.
 - 404 Not Found: No se encontró el vehículo.

11. Obtener la capacidad promedio de personas por marca

- **Como:** Usuario de la API.
- **Quiero:** Conocer la capacidad promedio de personas de los vehículos de una marca específica.
- **Para:** Comparar la capacidad de diferentes marcas y decidir una compra.
- **Endpoint:** GET /vehicles/average_capacity/brand/{brand}
- **Respuestas:**
 - 200 OK: Devuelve la capacidad promedio de personas de la marca.
 - 404 Not Found: No se encontraron vehículos de esa marca.

12. Buscar vehículos por dimensiones

- **Como:** Usuario de la API.
- **Quiero:** Listar vehículos basados en un rango de dimensiones (largo, ancho).
- **Para:** Encontrar vehículos que se adapten a mis necesidades de espacio.
- **Endpoint:** GET
/vehicles/dimensions?length={min_length}-{max_length}&width={min_width}-{max_width}
- **Respuestas:**

- 200 OK: Devuelve una lista de vehículos que cumplen con los criterios de dimensiones.
- 404 Not Found: No se encontraron vehículos con esas dimensiones.

13. Listar vehículos por rango de peso

- **Como:** Usuario de la API.
 - **Quiero:** Listar vehículos basados en un rango de peso.
 - **Para:** Encontrar vehículos que cumplan con mis requisitos de carga o preferencias.
 - **Endpoint:** GET /vehicles/weight?min={weight_min}&max={weight_max}
 - **Respuestas:**
 - 200 OK: Devuelve una lista de vehículos que están en el rango de peso especificado.
 - 404 Not Found: No se encontraron vehículos en ese rango de peso.
-

Guía de trabajo

Para concluir, te ofrecemos algunas pautas que te ayudarán a prepararte de la mejor manera para el día de tu encuentro con los facilitadores. Estas recomendaciones podrían ser de gran utilidad para ti. ¡Aprovéchalas!

1. ¿Qué esperamos que aprendas de esta instancia? 🤖

- a. **Recordar (Conocer):** Identificar y describir los componentes básicos de una API, tales como endpoints, handlers y queries, y reconocer la estructura estándar de un proyecto de software.
- b. **Comprender:** Explicar la importancia de las convenciones de código y el papel que desempeñan en la mantenibilidad y legibilidad del software. Además, interpretar las solicitudes del usuario y traducirlas en funcionalidades técnicas.
- c. **Aplicar:** Implementar endpoints específicos en la API basados en los ejercicios proporcionados, utilizando técnicas adecuadas de filtrado y ordenación en el repositorio.
- d. **Analizar:** Diferenciar entre un código bien estructurado y uno que necesita mejoras, examinando la lógica de negocio, las queries y la seguridad de los endpoints.
- e. **Evaluar:** Justificar decisiones tomadas durante el desarrollo, considerando eficiencia, seguridad y usabilidad. Además, revisar y evaluar el código de pares para identificar áreas de mejora o refactoring.

2. Algunas recomendaciones: 📝

- a. Antes de la entrevista, repasa los ejercicios y asegúrate de entender cada problema y posibles soluciones.
- b. Prepárate para justificar y explicar tus decisiones.
- c. Mantente abierto a feedback. El proceso no solo es evaluativo sino también formativo; se trata de aprender y mejorar.

¡Éxitos! 🍀💪🌟

¡Esperamos verte brillar ✨✨ durante el "CodeReview Day"!