

Taller de Desarrollo

Web 2024

Trabajo final de cursada

Elgart, Ximena

Leguizamón, Leonel

Índice

Introducción	3
Stack tecnológico	4
Funcionalidad	6
Listar autos disponibles en una ciudad entre dos fechas dadas.	6
Filtrado de autos disponibles en una ciudad entre dos fechas dadas.	7
Finalización de alquiler y visualización de alquileres previos	8
Listar todos los autos de una categoría (sin importar ciudad o fecha)	9
Listar autos mejor puntuados	10
Listar reviews de usuarios	10
API Rest y Endpoints	12
Endpoints de Autos (Cars)	12
Endpoints de Ciudades (Cities)	13
Endpoints de Reservas (Bookings)	14
Endpoints de Reseñas (Reviews)	14
Endpoints de Usuarios (Users)	15
Conclusiones	16

Introducción

Como proyecto para este taller de desarrollo web decidimos crear una página de alquiler de autos al estilo Hertz u otras cadenas de Rent-a-car similares, llamada Rentar.

La idea de la aplicación es que un usuario pueda ver los autos disponibles para alquiler en cada una de las sucursales establecidas en el país, según categorías, fechas, tipos de autos, etc. y, lógicamente, realizar reservas.

Además de lograr las funcionalidades mencionadas, el proyecto tiene por objetivo desarrollar habilidades de programación tanto en Frontend como en Backend, consiguiendo así, construir una aplicación denominada actualmente como Full Stack.

Stack tecnológico

La aplicación Rentar fue desarrollada utilizando el stack **MERN**, el cual incluye MongoDB, Express.js, React y Node.js, junto con **Docker** y **Mongoose** para facilitar el deployment y la gestión de datos, y Git/Github para el control de versiones y desarrollo colaborativo.

MongoDB es una base de datos NoSQL, orientada a documentos, que permite almacenar datos en un formato flexible y escalable.

Mongoose es una biblioteca de modelado de datos para MongoDB y Node.js. Proporciona una solución basada en esquemas para modelar datos, lo que facilita la validación y el manejo de datos en la base de datos.

En el contexto de Rentar, MongoDB se utiliza para almacenar la información de autos, usuarios, reservas, ciudades (sucursales) y reseñas, mientras que Mongoose se utiliza para el modelado de datos con esquemas. La especificación de cada uno de los esquemas se puede ver en el repositorio del proyecto en la ruta *Backend > models*.

Express.js es un framework para Node.js que facilita la creación de aplicaciones web y APIs. En Rentar, Express.js se encarga de manejar las rutas del servidor, procesar las solicitudes HTTP y comunicarse con la base de datos. Las rutas para cada endpoint se pueden visualizar en el repositorio del proyecto en la ruta *Backend > routes > api.js*.

React es una biblioteca de JavaScript para construir interfaces de usuario. En Rentar, React se utiliza para desarrollar el frontend, permitiendo a los usuarios interactuar con la aplicación de manera eficiente y dinámica. Los componentes y la interfaz se pueden ver en el repositorio del proyecto en la ruta *Frontend*.

Node.js es un entorno de ejecución para JavaScript, el cual permite ejecutar código JavaScript en el servidor, lo que facilita la creación de aplicaciones web completas utilizando un solo lenguaje de programación. En Rentar, Node.js se utiliza para desarrollar el backend de la aplicación. La configuración del servidor se puede

visualizar en el repositorio del proyecto, en el archivo *index.js* ubicado en la carpeta *Backend*.

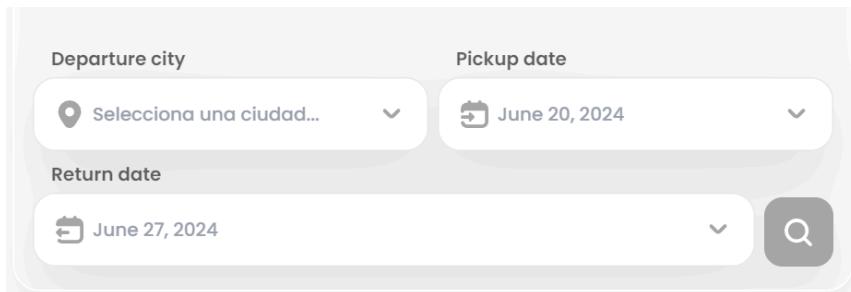
Docker es una plataforma que permite crear, desplegar y ejecutar aplicaciones en contenedores. En Rentar, Docker se utiliza para facilitar el desarrollo y despliegue de la aplicación, asegurando que el entorno de desarrollo sea el mismo en cualquier máquina donde se ejecute el contenedor, eliminando problemas de configuración y dependencias entre diferentes entornos de desarrollo.

Funcionalidad

Como se mencionó en un principio, la página posee diversas funcionalidades acordes a un Rent-a-Car. En el siguiente listado se especifican las más importantes o principales.

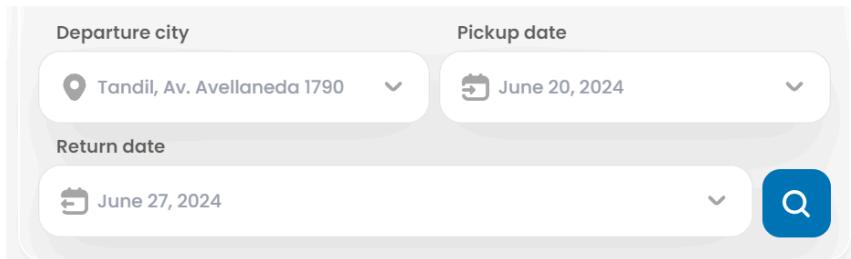
Listar autos disponibles en una ciudad entre dos fechas dadas.

Para poder listar los autos, el usuario debe obligatoriamente elegir una “ciudad de salida”. Una vez seleccionada, puede presionar el botón de búsqueda, que lo trasladará a otra pantalla donde se listarán los autos según las condiciones establecidas.



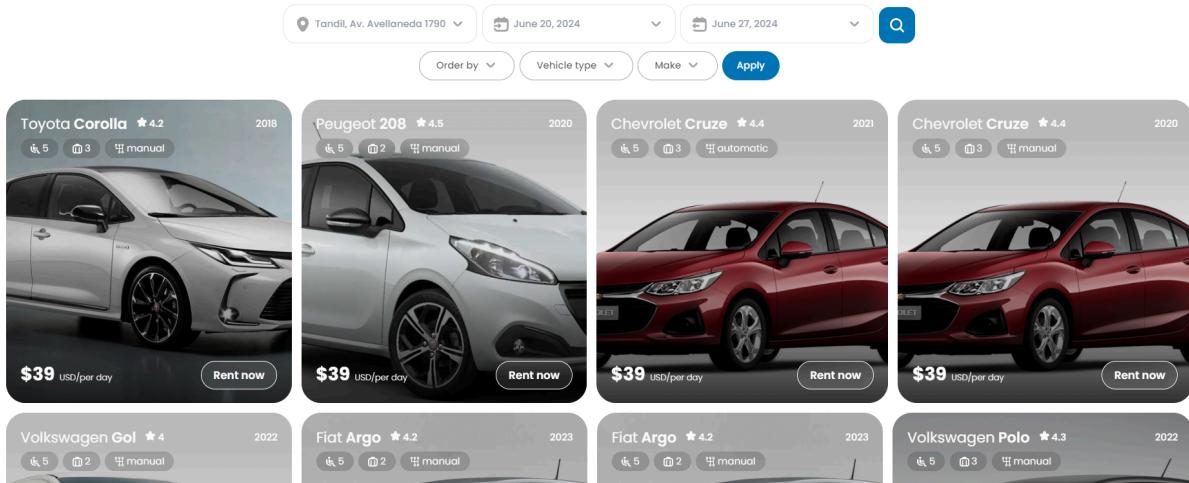
A screenshot of a search interface. It features three input fields: 'Departure city' with a placeholder 'Selecciona una ciudad...', 'Pickup date' set to 'June 20, 2024', and 'Return date' set to 'June 27, 2024'. A search button with a magnifying glass icon is located to the right of the return date field. All fields have dropdown arrows indicating they are interactive.

Img. 1: El usuario no ha seleccionado ciudad, en tanto no está habilitada la búsqueda.



A screenshot of the same search interface as above, but with the 'Departure city' field now populated with 'Tandil, Av. Avellaneda 1790'. The other fields ('Pickup date' and 'Return date') and the search button remain the same as in Img. 1.

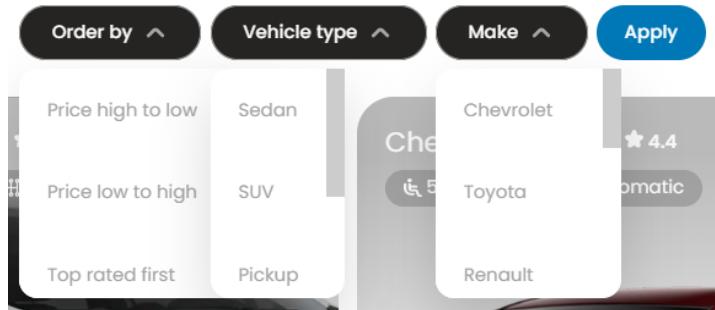
Img. 2: El usuario ha seleccionado la ciudad, en tanto se habilita la búsqueda.



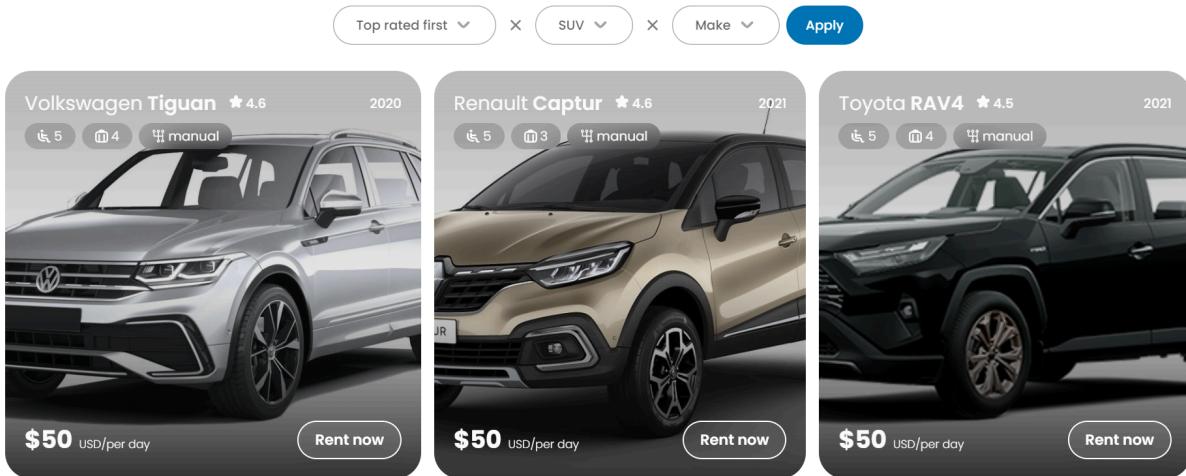
Img. 3: Listado de vehículos disponibles en Tandil, del 20 al 27 de Junio.

Filtrado de autos disponibles en una ciudad entre dos fechas dadas.

Una vez en la página donde se listan los vehículos, el usuario puede filtrar según diversos criterios, seleccionar otras fechas, otra ciudad, y clickear en el auto de su gusto para ser redireccionado a la página de finalización de renta.



Img. 4: Filtros disponibles para aplicar.



Img. 5: Resultado de ordenar según puntaje y filtrado solo de SUV.

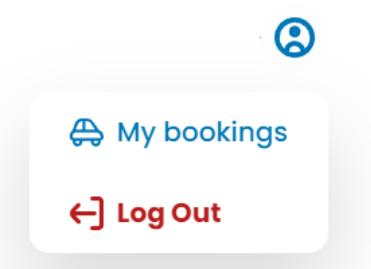
Finalización de alquiler y visualización de alquileres previos

Cuando el usuario encuentra un vehículo de su gusto (y haya seleccionado una ciudad) puede clickear en “Rent Now” para ser redirigido a la finalización de alquiler. Aquí se detalla el período de alquiler, la ciudad, y el precio final.



Img. 6: Página para finalizar alquiler. El usuario debe estar logueado para poder alquilar un auto.

Una vez que el usuario finaliza el alquiler del vehículo, esto se refleja en la página de “My Bookings” (opción disponible si clickeamos en el ícono de usuario arriba a la derecha).



Here are your bookings

	Volkswagen Gol 2022	Sedan	Tandil, Av. Avellaneda 1790	2024-06-15	2024-06-20	Ended	\$195
	Mercedes-Benz C220 2022	Sport	Mar del Plata, Av. Luro 1790	2024-06-15	2024-06-22	Active	\$623
	Volkswagen Tiguan 2020	SUV	Tandil, Av. Avellaneda 1790	2024-06-20	2024-06-27	Active	\$350

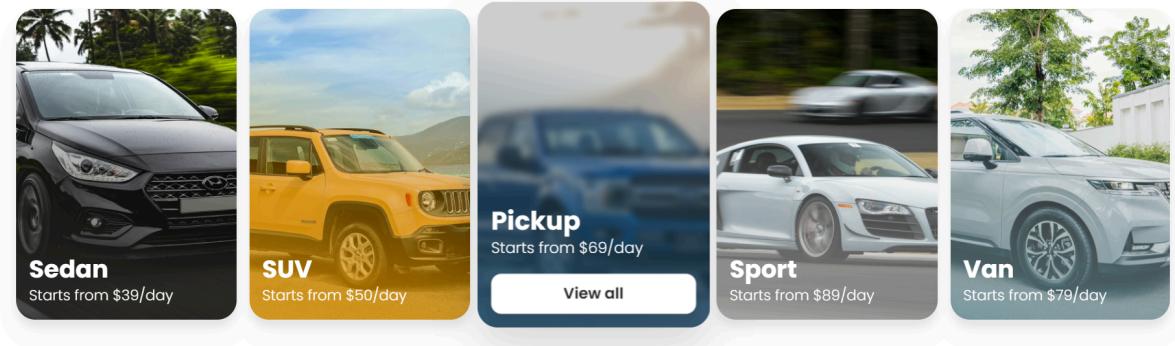
Img. 7: Página de listado de alquileres. Se informan detalles de la reserva, el precio final, y si está activa, terminada o lista para retirar.

Listar todos los autos de una categoría (sin importar ciudad o fecha)

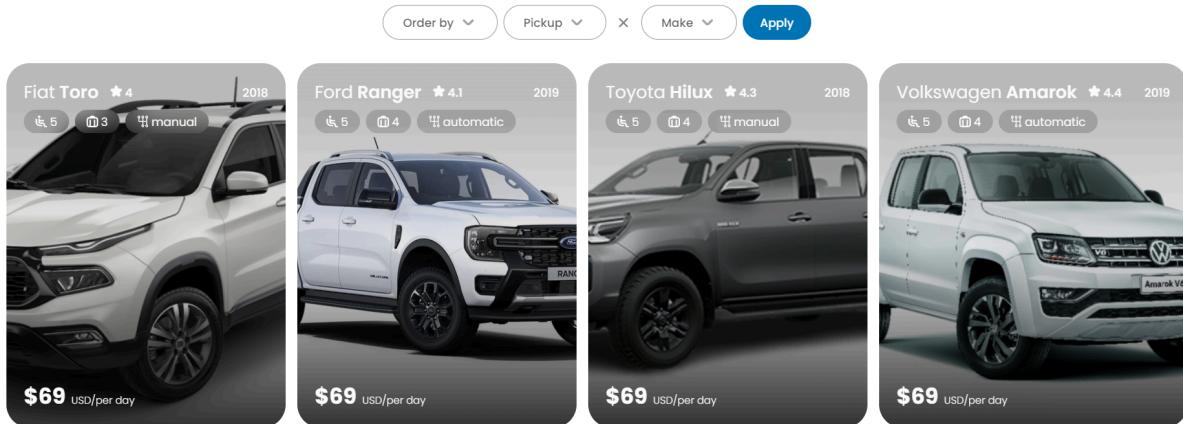
Si el usuario sólo desea ver los autos disponibles, o precios, o solo está inspeccionando la página, puede hacerlo sin necesidad de establecer fechas ni ciudades de salida gracias la funcionalidad de listar todos los autos de una categoría, disponible en la página principal.

Your adventure, your choice

Find the one that best suits you



Img. 8: Apartado para seleccionar la categoría. Al clickear el botón, será redirigido a la página de listado de autos.



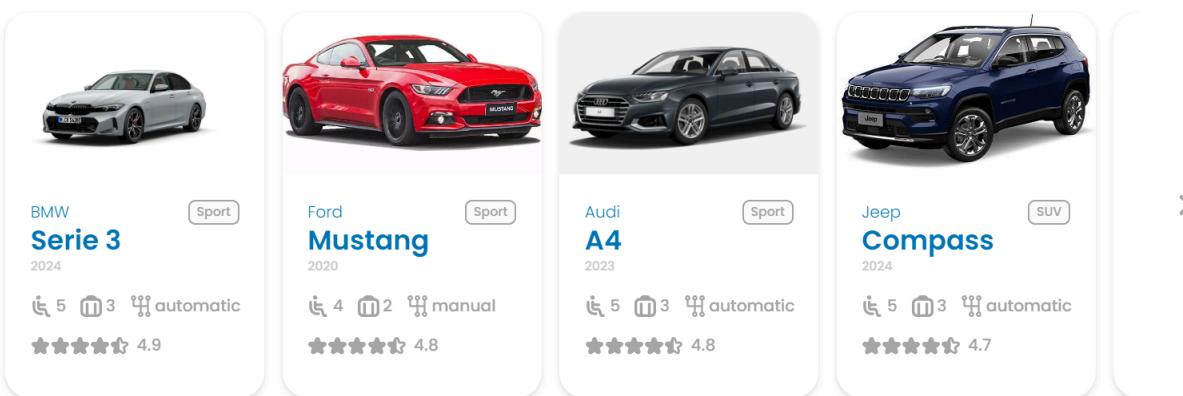
Img. 9: Página de listado de autos. Lista todos los vehículos de la categoría seleccionada. Podemos apreciar que al no haber seleccionado una ciudad, no es posible ir a la página de finalización de renta.

Listar autos mejor puntuados

En este apartado de la página principal, simplemente se listan los 10 vehículos mejor puntuados de toda la plataforma, independientemente de su ciudad.

Our top rated vehicles by clients

+12K Reviews +100 Cars with 4.5+ stars +50K Rented cars



Img. 10: Apartado donde se listan los 10 vehículos mejor puntuados.

Listar reviews de usuarios

Si bien la plataforma no permite el posteo de opiniones actualmente (podría permitirlo en un futuro), tiene activa la funcionalidad para mostrar las opiniones que hagan, eventualmente, los usuarios.

Thousands of clients trust in us, join them!

I had a wonderful experience with Rent.ar! The team was incredibly helpful and friendly, ensuring that the rental process was quick and easy. The vehicle I received was pristine and performed perfectly. I will definitely be using Rent.ar for all my future car rentals!



John Doe

★ 4.7 Avg. ratings

I had a wonderful experience with Rent.ar! The team was incredibly helpful and friendly, ensuring that the rental process was quick and easy. The vehicle I received was pristine and performed perfectly. I will definitely be using Rent.ar for all my future car rentals!



Maria Rodriguez

★ 4.4 Avg. ratings

I had a wonderful experience with Rent.ar! The team was incredibly helpful and friendly, ensuring that the rental process was quick and easy. The vehicle I received was pristine and performed perfectly. I will definitely be using Rent.ar for all my future car rentals!



Juan Perez

★ 4.6 Avg. ratings

Img. 11: Reviews de usuarios

API Rest y Endpoints

La aplicación Rentar utiliza una API Rest para la comunicación entre el frontend y el backend. La API está diseñada para manejar las operaciones CRUD relacionadas con los autos, ciudades, reservas, reseñas y usuarios. A continuación se detallan los endpoints desarrollados y sus funcionalidades.

Endpoints de Autos (Cars)

POST /cars - *findCars*

- Descripción: busca autos disponibles entre un rango de fechas y en una ciudad especificada.
- Parámetros: ciudad, fecha de comienzo y fecha de finalización de la reserva en el cuerpo de la solicitud.
- Respuesta: lista de autos que coinciden con los criterios.

GET /cars/category/:category - *findCarsByCategory*

- Descripción: obtiene los autos que pertenecen a la categoría especificada (Sedan, SUV, Pickup, Van, Sport).
- Parámetros: categoría en la URL.
- Respuesta: lista de autos pertenecientes a la categoría especificada.

GET /cars/toprated - *findTopRatedCars*

- Descripción: obtiene los 10 autos mejor valorados.
- Parámetros: -
- Respuesta: lista de autos con las mejores valoraciones.

GET /cars/make/:make - *findCarsByMake*

- Descripción: obtiene los autos que pertenecen a la marca especificada.
- Parámetros: marca en la URL.
- Respuesta: lista de autos de la marca especificada.

POST /cars/add - addCars

- Descripción: añade uno o varios autos nuevos.
- Parámetros: arreglo de autos enviados en el cuerpo de la solicitud.
- Respuesta: información de los autos añadidos.

DELETE /cars/:id - deleteCar

- Descripción: elimina un auto.
- Parámetros: id del auto en la URL.
- Respuesta: confirmación de la eliminación.

PUT /cars - updateCars

- Descripción: actualiza la información de un auto.
- Parámetros: datos del auto actualizados enviados en el cuerpo de la solicitud.
- Respuesta: información del auto actualizado.

Endpoints de Ciudades (Cities)

PUT /cities - updateCityAddress

- Descripción: actualiza la dirección de una sucursal de una ciudad.
- Parámetros: datos de la ciudad actualizados enviados en el cuerpo de la solicitud.
- Respuesta: información de la ciudad actualizada.

POST /cities - addCity

- Descripción: añade una nueva ciudad.
- Parámetros: datos de la ciudad enviados en el cuerpo de la solicitud.
- Respuesta: información de la ciudad añadida.

DELETE /cities/:id - deleteCity

- Descripción: elimina una ciudad.
- Parámetros: id de la ciudad en la URL.
- Respuesta: confirmación de la eliminación.

GET /cities - *findCities*

- Descripción: obtiene todas las ciudades.
- Parámetros: marca en la URL.
- Respuesta: lista de ciudades.

Endpoints de Reservas (Bookings)

POST /bookings - *createBooking*

- Descripción: crea una nueva reserva.
- Parámetros: datos de la reserva enviados en el cuerpo de la solicitud.
- Respuesta: información de la reserva creada.

DELETE /bookings/:id - *deleteBooking*

- Descripción: elimina una reserva.
- Parámetros: id de la reserva en la URL.
- Respuesta: confirmación de la eliminación.

GET /bookings/:user - *findBookings*

- Descripción: obtiene todas las reservas de un usuario específico.
- Parámetros: usuario en la URL.
- Respuesta: lista de las reservas del usuario especificado.

Endpoints de Reseñas (Reviews)

POST /reviews - *addReview*

- Descripción: añade una nueva reseña.
- Parámetros: datos de la reseña enviados en el cuerpo de la solicitud.
- Respuesta: información de la reseña creada.

DELETE /reviews/:id - *deleteReview*

- Descripción: elimina una reseña.
- Parámetros: id de la reseña en la URL.
- Respuesta: confirmación de la eliminación.

PUT /reviews - *updateReview*

- Descripción: actualiza una reseña.
- Parámetros: datos de la reseña actualizados enviados en el cuerpo de la solicitud.
- Respuesta: información de la reseña actualizada.

GET /reviews - *findReviews*

- Descripción: obtiene todas las reseñas.
- Parámetros: -
- Respuesta: lista de las reseñas.

Endpoints de Usuarios (Users)

POST /user - *addUser*

- Descripción: añade un nuevo usuario.
- Parámetros: datos del usuario enviados en el cuerpo de la solicitud.
- Respuesta: información del usuario creado.

DELETE /user/:id - *deleteUser*

- Descripción: elimina un usuario.
- Parámetros: id del usuario en la URL.
- Respuesta: confirmación de la eliminación.

PUT /user - *updateUser*

- Descripción: actualiza la información de un usuario.
- Parámetros: datos del usuario actualizados enviados en el cuerpo de la solicitud.
- Respuesta: información del usuario actualizado.

Conclusiones

El desarrollo de la aplicación Rentar con el stack MERN nos sirvió como una buena experiencia de programación web, siendo que teníamos poco o nulo conocimiento en las tecnologías utilizadas. Pudimos integrar MongoDB, Express.js, React y Node.js para crear una solución completa capaz de gestionar vehículos, ciudades, reservas, reseñas y usuarios.

Durante el proyecto, enfrentamos desafíos tanto en el frontend (puntualmente conexión y fetching de datos, ruteo y parámetros de ruta, reutilización de componentes), en el backend (creación de la API, conexión a la bbdd con mongoose) y en el deployment (contenerización con Docker Compose, y actualización dinámica con Docker Compose Watch).

Finalmente, esta experiencia nos permitió consolidar habilidades en desarrollo web y mejorar nuestra capacidad para trabajar en equipo y gestionar proyectos.