

**Universidad Manuela Beltrán**

**Facultad de ingeniería**

**Software**

**Virtual**

**ingeniería Web**

**Diana Toquica**

**Diego Andres Rodriguez Martin**

**Bogotá D.C.**

**2025**

### Actividad 3. Mini Proyecto de Integración Continua

El proyecto Autos DARM consiste en el desarrollo de una aplicación web con arquitectura REST, diseñada para la gestión de usuarios dentro de un concesionario de autos. El sistema permite registrar nuevos usuarios, almacenar sus datos y autenticar el acceso a la interfaz principal de la página.

Además, el proyecto implementa pruebas automatizadas y un pipeline de integración continua (CI), garantizando la calidad y estabilidad del código mediante validaciones automáticas cada vez que se actualiza el repositorio.

#### Tecnologías utilizadas

- **Frontend:** HTML, CSS y JavaScript (interfaz de registro e inicio de sesión).
- **Backend simulado:** JSON Server, utilizado para crear una API REST sencilla.
- **Pruebas automatizadas:** Jest y Supertest para validar los endpoints de la API.
- **Gestor de dependencias:** NPM (Node Package Manager).
- **Control de versiones:** Git y GitHub.
- **Integración continua:** GitHub Actions.
- 

#### Desarrollo del proyecto

##### Diseño y planificación

Se definió el alcance del sistema: crear una interfaz que permita registrar e iniciar sesión de usuarios.

Se adoptó una arquitectura cliente-servidor con un backend simulado mediante JSON Server, que expone un endpoint principal /users para almacenar la información en un archivo db.json.

##### Implementación

Se construyó una página principal (index.html) con un formulario de inicio de sesión y registro.

Los datos de los usuarios se envían al servidor mediante peticiones fetch (POST/GET).

El servidor JSON Server almacena los registros en db.json, funcionando como una base de datos ligera.

## Registro Autos DARM

Nombre

Correo

Rol

Asesor

Contraseña

Registrar

[¿Ya tienes cuenta? Inicia sesión](#)

## Registro Autos DARM

Nombre

Correo

Rol

Asesor

Contraseña

Registrar

[¿Ya tienes cuenta? Inicia sesión](#)

## Registro Autos DARM

Nombre

Correo

Rol

Asesor

Contraseña

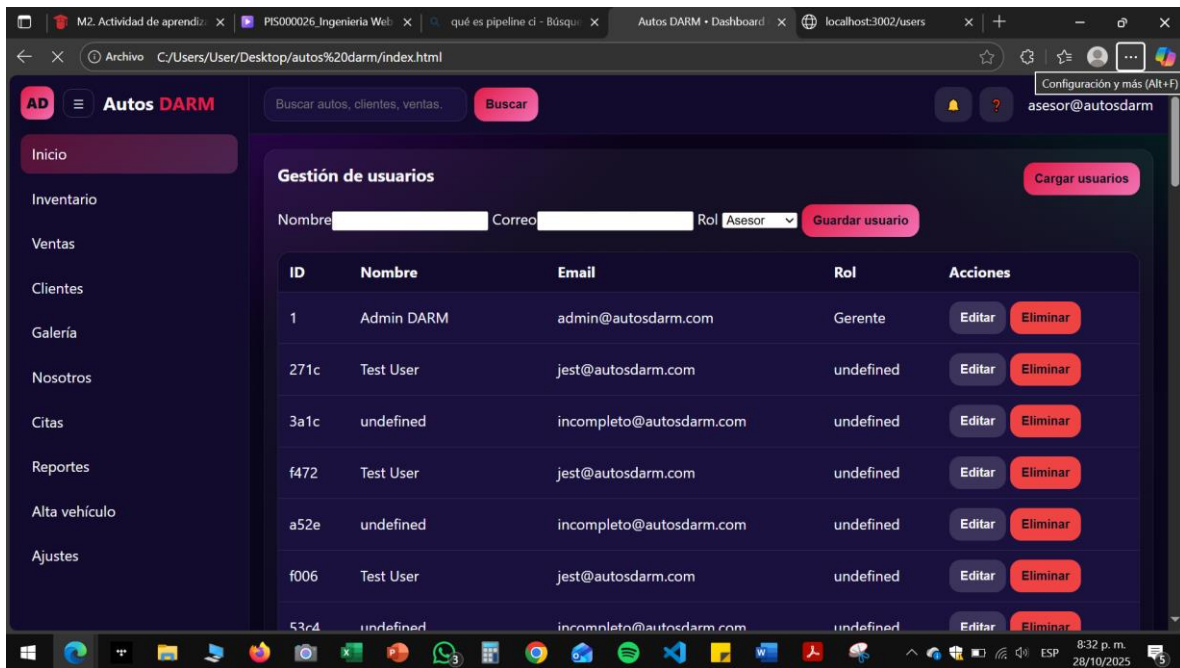
Registrar

✓ Registro exitoso, redirigiendo...

[¿Ya tienes cuenta? Inicia sesión](#)

Impresión con sangría

```
{
  "email": "incompleto@autosdarm.com"
},
{
  "id": "7275",
  "nombre": "diego rodriguez",
  "email": "diego98@autosdarm.com",
  "rol": "Gerente",
  "password": "0000"
},
{
  "id": "f6d9",
  "nombre": "Test User",
  "email": "jest@autosdarm.com",
  "password": "12345"
},
{
  "id": "47a0",
  "email": "incompleto@autosdarm.com"
},
{
  "id": "6e74",
  "nombre": "Test User",
  "email": "jest@autosdarm.com",
  "password": "12345"
},
{
  "id": "d7a0",
  "email": "incompleto@autosdarm.com"
},
{
  "id": "fff9",
  "nombre": "EDUARDO LOZANO ",
  "email": "erl@autosdarm.com",
  "rol": "Asesor",
  "password": "12345"
}
```



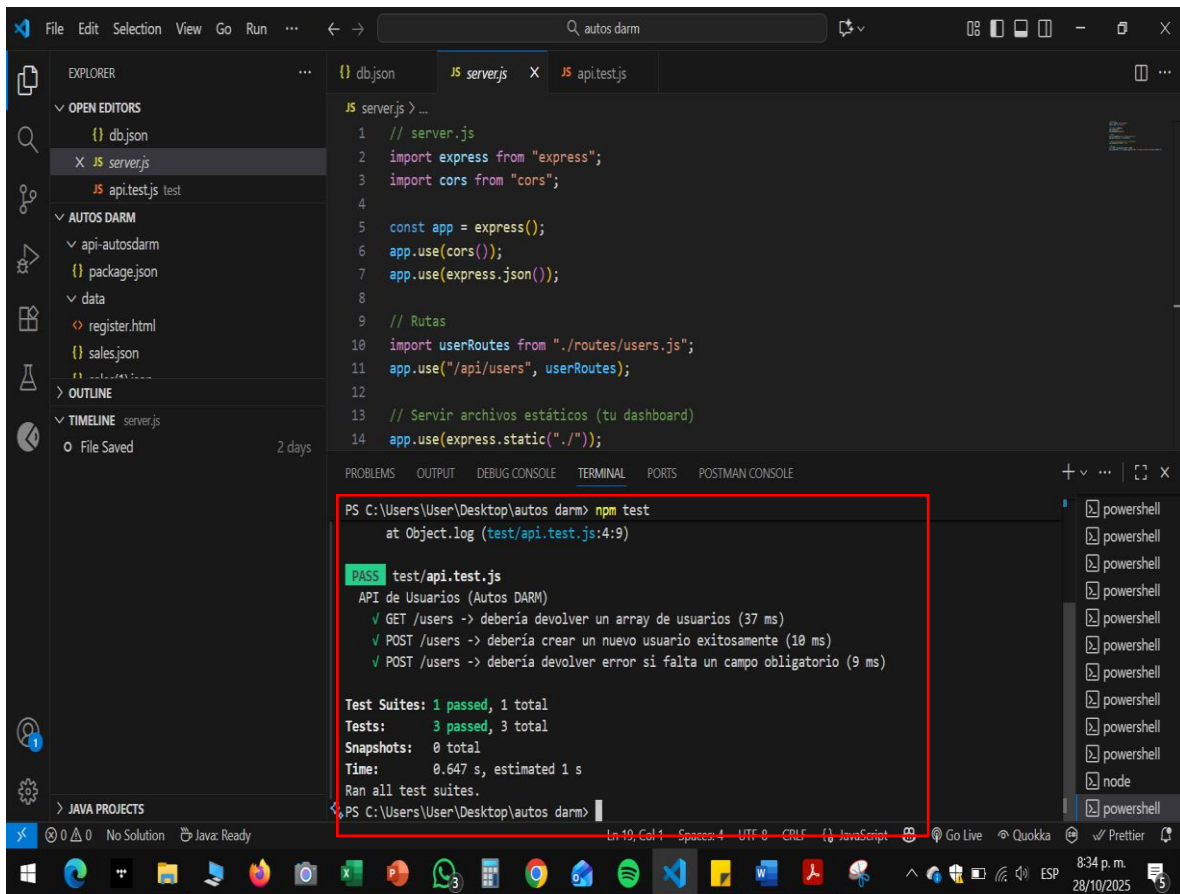
## Pruebas y validación

Se implementaron pruebas automatizadas con Jest y Supertest para verificar la funcionalidad de la API.

Estas pruebas cubren:

- GET /users: Devuelve la lista de usuarios.
- POST /users: Crea un nuevo usuario correctamente.
- Manejo de errores ante datos incompletos.

Los resultados se validaron con códigos de estado HTTP (200, 201, 400, 500) y se ajustaron las pruebas para reflejar el comportamiento real de JSON Server.



## Integración continua (CI)

Se configuró un pipeline CI utilizando GitHub Actions, el cual ejecuta automáticamente los siguientes pasos cada vez que se realiza un push o pull request:

1. Clona el repositorio.
2. Instala las dependencias (npm install).
3. Ejecuta las pruebas (npm test).
4. Informa si todas las pruebas pasaron o si hubo fallos.

Esto garantiza que el código que se integre en la rama principal esté probado, validado y libre de errores.

## Resultados obtenidos

- Se logró una API REST funcional con operaciones básicas de usuarios.
- Se implementó un sistema de pruebas automatizadas exitoso.

- El pipeline de CI se integró correctamente, permitiendo validación continua.
- Se documentó la arquitectura y los flujos del sistema con diagramas y documentación técnica.

## Conclusión

El proyecto Autos DARM demuestra la aplicación práctica de conceptos de arquitectura REST, automatización de pruebas y metodologías DevOps.

A través del pipeline de CI, se logró un proceso de desarrollo más seguro, controlado y eficiente, garantizando la calidad del software en cada etapa.