



INSTRUCTOR: NAIN ZUÑIGA PORTO

APREDENDIZ: CRISTIAN DAVID ALARCON HUERFANO

SERVICIO NACIONAL DE APRENDIZAJE – SENA

Evidencia de producto: GA7-220501096-AA5-EV04: API del proyecto

PRUEBAS DE TESTING CON POSTMAN API del Proyecto TechRepair GA7 220501096  
AA5 EV04

TECNOLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE

FICHA: 2977354

FECHA: 23 de noviembre de 2025

SOACHA, CUNDINAMARCA

## Objetivo

Realizar testing de las APIs de autenticación (Register y Login) desarrolladas en la evidencia EV03, verificando que funcionan correctamente mediante la herramienta Postman.

## Endpoints Testeados

### ENDPOINT: POST /api/auth/register

**Descripción:** Servicio para registrar un nuevo usuario en el sistema.

**Método:** POST

**Status Code Esperado:** 201 Created

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:3000/api/auth/register`. The request method is `POST`. The body contains the following JSON:

```

1
2   "name": "cristian",
3   "email": "cristian.alarcon@iteperformance.com",
4   "password": "Performance2925"
5

```

The response status is `201 Created` with a response time of `898 ms` and a size of `524 B`. The response body is:

```

1
2   "message": "Registro exitoso - usuario creado correctamente",
3   "success": true,
4   "code": "REGISTRATION_SUCCESS",
5   "user": {
6     "id": "6923b24edf30ee3c4867dba3",
7     "name": "cristian",
8     "email": "cristian.alarcon@iteperformance.com",
9     "createdAt": "2025-11-24T01:18:06.212Z"
10   }
11

```

## Observaciones:

- El usuario fue registrado correctamente
- Se asignó un ID único en MongoDB
- La contraseña fue hasheada por bcrypt
- Se generó automáticamente el timestamp de creación

## ENDPOINT: POST /api/auth/register (Segundo Usuario)

**Descripción:** Registro de un segundo usuario de prueba.

**Método:** POST

**URL:** <http://localhost:3000/api/auth/register>

**Request Body (JSON):**

**Status Code Esperado:** 201 Created

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, Explore, and a search bar labeled "Search Postman". On the right side, there are buttons for "Sign in" and "Create Account". The main workspace shows a history of requests under the "Today" section, with three entries for "POST http://localhost:3000/api/auth/register". A specific request is selected in the center panel, with its URL being "http://localhost:3000/api/auth/register". The "Body" tab is active, showing the JSON payload:

```

1  {
2    "name": "Gabriella",
3    "email": "gabriella.londoño@itegperformance.com",
4    "password": "Performance2025"
5  }

```

Below the request details, the "Body" tab is selected in the bottom navigation bar, along with "Cookies", "Headers (8)", and "Test Results". The "JSON" dropdown is set to "Pretty". The response section shows the status "201 Created" and the response body:

```

1  {
2    "message": "Registro exitoso - usuario creado correctamente",
3    "success": true,
4    "code": "REGISTRATION_SUCCESS",
5    "user": {
6      "id": "6923b591c13df3117d1ef656",
7      "name": "Gabriella",
8      "email": "gabriella.londoño@itegperformance.com",
9      "createdAt": "2026-11-24T01:32:01.352Z"
10    }
11  }

```

At the bottom of the interface, there are buttons for "Create collections in Postman", "Create a Collection", "Dejar de compartir", and "Ocultar". A message "Activar Windows" with a link "Ve a Configuración para activar Windows." is also visible.

## Observaciones:

- Segundo usuario registrado exitosamente
- Confirmación de funcionamiento correcto del endpoint
- Base de datos almacenando múltiples usuarios correctamente

## ENDPOINT: POST /api/auth/login

**Descripción:** Servicio para autenticar un usuario y generar token JWT.

**Método:** POST

**URL:** <http://localhost:3000/api/auth/login>

**Status Code Esperado:** 200 OK

The screenshot shows the Postman interface with a successful API call to `http://localhost:3000/api/auth/login`. The request body is set to `JSON` and contains the following data:

```

1
2
3
4
5
{
  "email": "gabriella.londoño@itegperformance.com",
  "password": "Performance2025"
}

```

The response body is displayed in `Pretty` format:

```

1
2
3
4
5
{
  "message": "Autenticación satisfactoria",
  "success": true,
  "code": "AUTHENTICATION_SUCCESS",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZC16IjY5MjN1NTkxYxEzZGYzMTE3ZDF1ZjY1NiIsImVtYWlsIjoiz2FicmlbOxLmxvbmRvv7FvQGl0ZwdwZXJmb3JtYW5jZS5jb29ilC3uW1l1joIR2FicmlbOxhXiwiwMFOIjoxNzYzOTQ4MDc5LCJleHA10jE3NjQmMzQ0MT19.SXLYTWS0kmhLKQVQb-Jr0RpZX18u38UDG6c5IVv4fE",
  "user": {
    "id": "6923b591c13df3117dief656",
    "email": "gabriella.londoño@itegperformance.com",
    "name": "Gabriella"
  }
}

```

## Observaciones:

- Login exitoso
- Token JWT generado correctamente
- Token puede usarse para autenticar futuras requests
- Datos del usuario retornados correctamente

## Resultados del Testing

Endpoint	Método	Status	Resultado
/api/auth/register (Cristian)	POST	201 Created	EXITOSO
/api/auth/register (Gabriella)	POST	201 Created	EXITOSO
/api/auth/login	POST	200 OK	EXITOSO

## Características Verificadas

- **Validación de Datos:** El servidor valida que se envíen todos los campos requeridos
- **Encriptación de Contraseña:** Las contraseñas se hashean con bcrypt antes de guardarse
- **Generación de Token JWT:** Se generan tokens válidos al hacer login
- **Almacenamiento en Base de Datos:** Los usuarios se guardan correctamente en MongoDB
- **Códigos de Estado HTTP:** Se retornan los códigos correctos (201 para registro, 200 para login)
- **Respuestas JSON:** Las respuestas tienen la estructura correcta con mensajes, success y datos

## Conclusiones

1. Los endpoints de autenticación funcionan correctamente
2. El sistema de registro permite crear múltiples usuarios
3. El sistema de login genera tokens JWT válidos
4. La seguridad está implementada correctamente (bcrypt para contraseñas)
5. La arquitectura en capas está funcionando adecuadamente
6. La base de datos MongoDB almacena los datos correctamente

**Estado Final: TODOS LOS TESTS PASARON EXITOSAMENTE**