

Evidencia de conocimiento: GA7-220501096-AA5-EV04

API del Proyecto

EUNICE DE LA ROSA BASILIO

Instructor:

EDUWIN ANDRES FLOREZ OREJUELA

Análisis Y Desarrollo De Software

(2977367)

011/07/2025

Pruebas de API con Postman

Objetivo del documento

El objetivo es mostrar las pruebas realizadas a los endpoints de la API del proyecto, utilizando la herramienta Postman. Se verifica el correcto funcionamiento del registro de usuario, login, registro de producto y listado de productos, así como los posibles errores al omitir información requerida.

1. Registro de Usuario

Método: POST

Endpointhttp://localhost/sistema_inventario/api/usuarios/registrar.php

Descripción: Permite registrar un nuevo usuario en el sistema.

Cuerpo de la petición (JSON):

```
{  
  "nombre": "Eunice",  
  "correo": "eunice@gmail.com",  
  "clave": "123456"  
}
```

Respuesta esperada:

```
{"mensaje": "Usuario registrado con éxito"}
```

Captura de prueba exitosa

The screenshot shows the Postman interface with a collection of requests. The selected request is a POST to `http://localhost/sistema_inventario/api/usuarios/registrar.php`. The body is a JSON object with the following data:

```
1 {
2   "nombre": "Eunice",
3   "correo": "eunice@gmail.com",
4   "clave": "123456"
5 }
```

The response status is 200 OK, with a response time of 153 ms and a body size of 297 B. The response body is a JSON object:

```
{ "mensaje": "Usuario registrado con éxito" }
```

Buttons for 'Send', 'Save', 'Cookies', 'Beautify', 'Save Response', 'Pretty', 'Raw', 'Preview', and 'Visualize' are visible.

Captura de error al no enviar datos:

The screenshot shows the Postman interface with a collection of requests. The selected request is a POST to `http://localhost/sistema_inventario/api/usuarios/registrar.php`. The body is a JSON object with only one field:

```
1 {
2   "nombre": "Andrea"
3 }
```

The response status is 200 OK, with a response time of 32 ms and a body size of 291 B. The response body is a JSON object indicating an error:

```
1 { "error": "Faltan datos obligatorios" }
```

Buttons for 'Send', 'Save', 'Cookies', 'Beautify', 'Save Response', 'Pretty', 'Raw', 'Preview', 'Visualize', and 'HTML' are visible.

2. Login de Usuario

Método: POST

Endpoint: http://localhost/sistema_inventario/api/usuarios/login.php

Cuerpo de la petición (JSON):

```
{  
  
  "correo": "eunice@gmail.com",  
  
  "clave": "123456"  
}
```

Respuesta esperada:

```
{"mensaje": "login exitoso", "usuario": "Eunice"}
```

Captura de prueba exitosa:

The screenshot displays the Postman interface with a history of requests on the left and a detailed view of a successful POST request on the right.

History Panel (Left):

- Today:
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
 - POST http://localhost/sistema_inventario/api/usuarios/login.php
- Yesterday: (empty)

Request Detail Panel (Right):

- Method:** POST
- URL:** http://localhost/sistema_inventario/api/usuarios/login.php
- Body:** raw (Text)

```
1 {"correo": "eunice@gmail.com",  
2   "clave": "123456"  
3 }  
4
```
- Response:** 200 OK, 124 ms, 302 B. Status: OK. Response body (Pretty):

```
1 {"mensaje": "Login exitoso", "usuario": "Eunice"}
```

Footer: Console, Not connected to a Postman account

Captura de error por credenciales incorrectas:

The screenshot shows a REST client interface with a list of requests at the top. The selected request is a POST to `http://localhost/sistema_inventario/api/usuarios/login.php`. The request body is a JSON object with the following fields:

```
1 {
2   "correo": "correo_existente@example.com",
3   "clave": "clave_incorrecta"
4 }
```

The response status is `200 OK` with a response time of `52 ms` and a size of `292 B`. The response body is displayed in the 'Body' tab as:

```
1 {"error": "Credenciales incorrectas"}
```

3. Registro de Producto

Método: POST

Endpoint: http://localhost/sistema_inventario/api/productos/producto.php

Cuerpo de la petición (JSON):

```
{
  "nombre": "Teclado",
  "descripcion": "Teclado mecánico RGB",
  "cantidad": 5,
  "precio": 150000
}
```

Respuesta esperada:

```
{ "mensaje": "Producto registrado correctamente"}
```

Captura de prueba exitosa:

The screenshot shows the Postman interface with a history of requests on the left and a detailed view of a successful POST request on the right. The request is to the endpoint `http://localhost/sistema_inventario/api/productos/registrar.php`. The body is a JSON object: `{ "nombre": "Monitor", "descripcion": "Monitor LED 24 pulgadas", "cantidad": 5, "precio": 450.00 }`. The response status is 200 OK, with a time of 47 ms and a size of 303 B. The response body is `{ "mensaje": "Producto registrado correctamente" }`.

History: Today

- POST `http://localhost/sistema_inventario/api/productos/`
- POST `http://localhost/sistema_inventario/api/productos/`
- POST `http://localhost/sistema_inventario/api/productos/`
- POST `http://localhost/sistema_inventario/api/productos/`
- POST `http://localhost/sistema_inventario/api/productos/`
- POST `http://localhost/sistema_inventario/api/usuarios/lc`
- POST `http://localhost/sistema_inventario/api/usuarios/re`
- POST `http://localhost/sistema_inventario/api/usuarios/lc`
- POST `http://localhost/sistema_inventario/api/usuarios/lc`

Create collections in Postman
Use collections to save your requests and share them with others.

POST `http://localhost/sistema_inventario/api/productos/registrar.php`

Params Auth Headers (8) Body ● Pre-req. Tests Settings

raw JSON

```
1 {
2   "nombre": "Monitor",
3   "descripcion": "Monitor LED 24 pulgadas",
4   "cantidad": 5,
5   "precio": 450.00
6 }
```

Body 200 OK 47 ms 303 B Save Response

Pretty Raw Preview Visualize HTML

```
1 { "mensaje": "Producto registrado correctamente" }
```

Captura de error por datos incompletos:

The screenshot shows the Postman interface with a detailed view of a failed GET request. The request is to the endpoint `http://localhost/sistema_inventario/api/productos/registrar.php`. The body is a JSON object: `{ "nombre": "", "descripcion": "Sin nombre", "cantidad": 5 }`. The response status is 200 OK, with a time of 15 ms and a size of 291 B. The response body is `{ "error": "Faltan datos del producto" }`.

GET `http://localhost/sistema_inventario/api/productos/registrar.php`

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "nombre": "",
3   "descripcion": "Sin nombre",
4   "cantidad": 5
5 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 15 ms Size: 291 B Save Response

Pretty Raw Preview Visualize HTML

```
1 { "error": "Faltan datos del producto" }
```

4. Listar Productos

Método: GET

Endpoint: http://localhost/sistema_inventario/api/productos/listar.php

Respuesta esperada:

```
{
  "id": 1,
  "nombre": "Monitor",
  "descripción": "Monitor LED 24 pulgadas",
  "cantidad": 5,
  "precio": 450
},
```

Captura de prueba exitosa:

The screenshot shows the Postman interface with a GET request to `http://localhost/sistema_inventario/api/productos/listar.php` executed successfully. The response status is 200 OK, and the response body is a JSON array containing one object representing a product.

Query Params

Key	Value
Key	Value

Body

```
1 [
2   {
3     "id": 1,
4     "nombre": "Monitor",
5     "descripción": "Monitor LED 24 pulgadas",
6     "cantidad": 5,
7     "precio": "450.00"
8   }
9 ]
```

Conclusión

Las pruebas realizadas en Postman verifican el correcto funcionamiento de los endpoints desarrollados en la API del proyecto. Se registran usuarios, se validan accesos por login, se crean productos y se listan correctamente. También se muestran respuestas adecuadas ante errores de validación.