

Implementación de Microservicios con Autenticación, Gestión de Productos y Órdenes usando Spring Boot, Eureka y JWT

Alumno: Benites Vera Jorge Bryhan

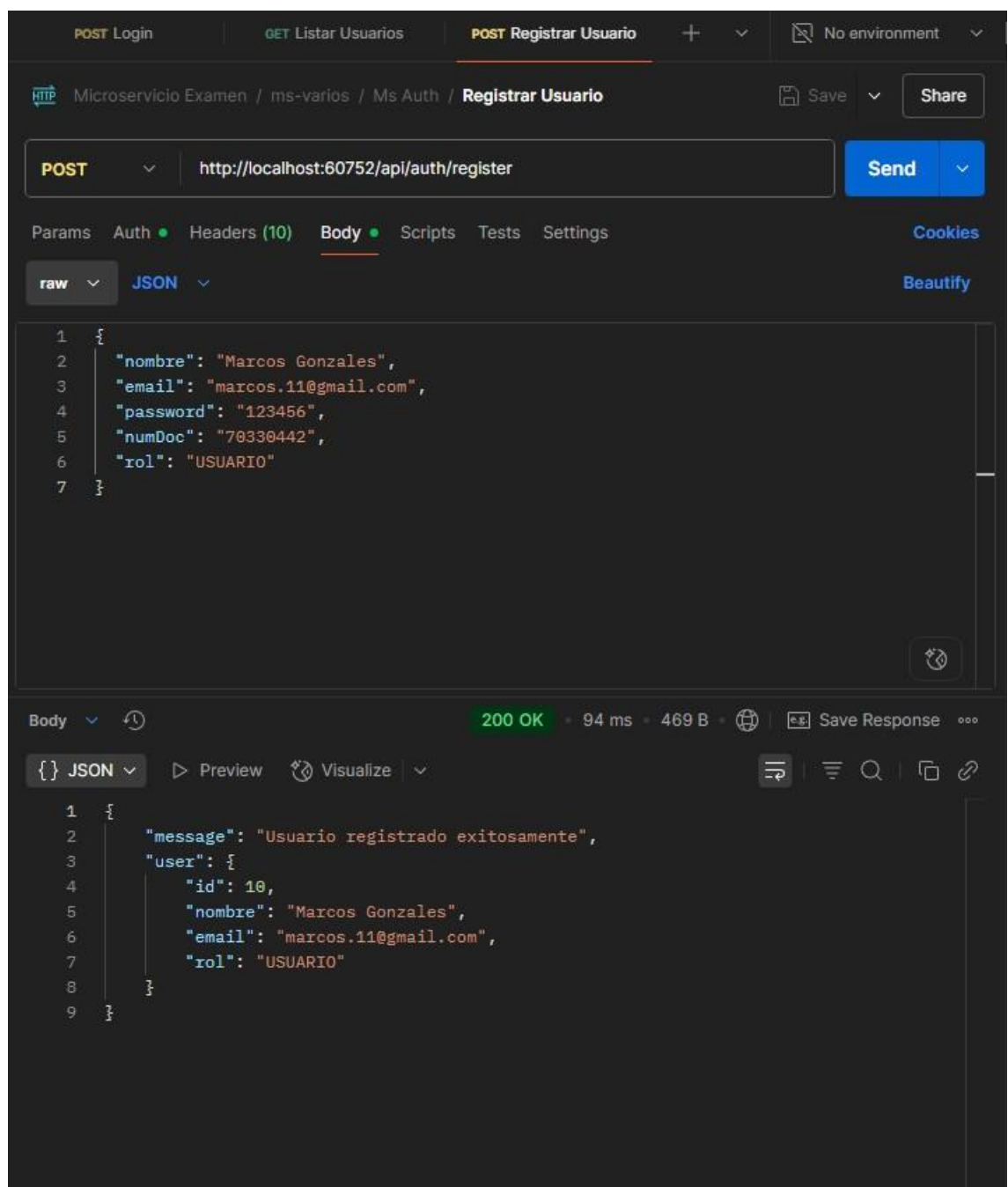
Link del repositorio : <https://github.com/jorgeben23/examen-ms-java>

El presente trabajo ha sido hecho con la siguiente tecnología Spring Boot 3.3.5, Java 17, JWT, Eureka además se hizo uso de herramientas como Postman para probar las APIS creadas.

1. Microservicio ms-auth: Autenticación y Autorización

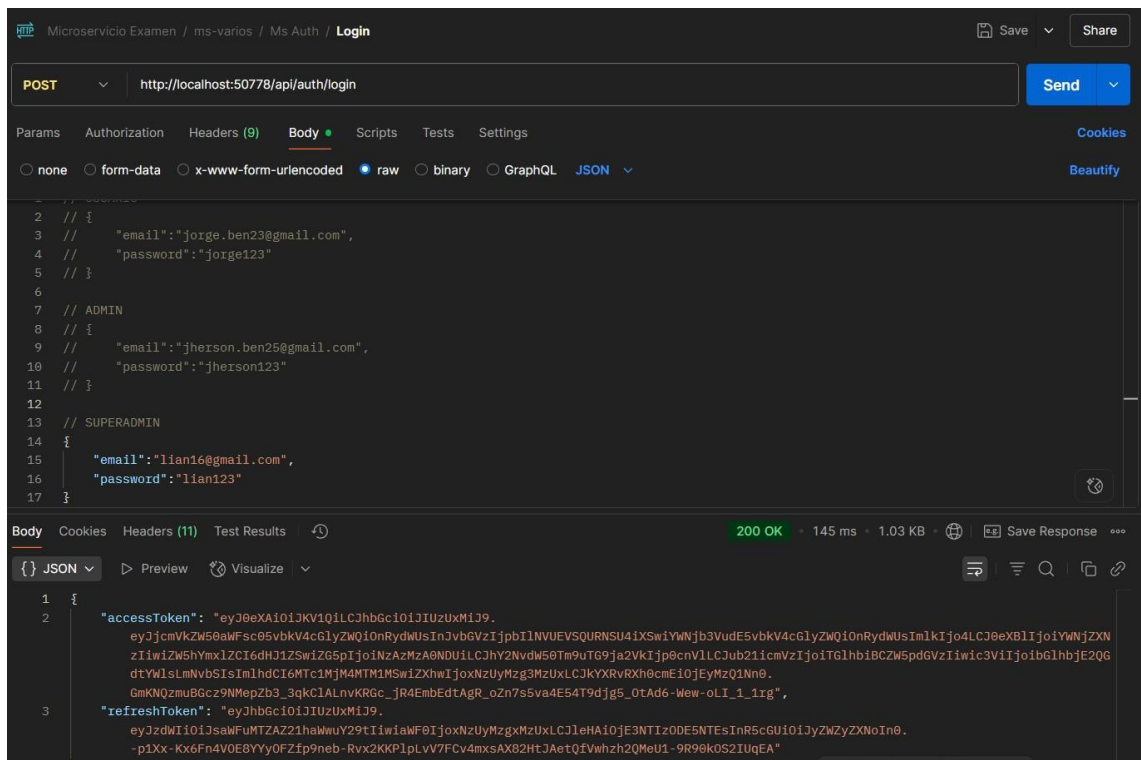
1.1. Registro (/api/auth/register)

Registra un nuevo usuario con nombre, email, contraseña, documento y rol. Es un endpoint público.



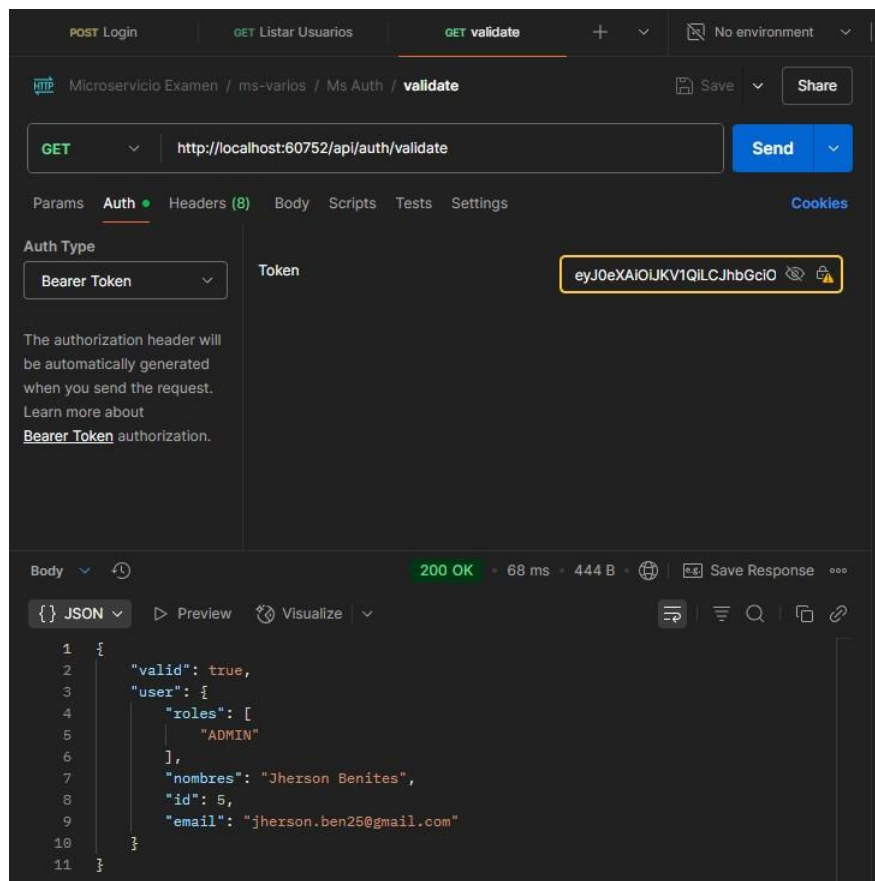
1.2. Login (/api/auth/login)

Autentica al usuario con su email y contraseña. Retorna un JWT accessToken y refreshToken.



1.3. Validación (/api/auth/validate)

Valida un JWT enviado por header. Retorna datos del usuario y sus roles si el token es válido.



2. Microservicio ms-productos: Gestión de Productos

2.1. Crear producto (POST /api/productos/registrar)

Registra un nuevo producto. Requiere autenticación y rol ADMIN o SUPERADMIN.

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8080/api/productos/registrar
- Body (raw JSON):**

```
1 {  
2   "nombre": "Borrador",  
3   "precio": 1.0,  
4   "categoria": "Utiles"  
5 }  
6
```
- Response:** 200 OK, 127 ms, 226 B. The response body (raw JSON) is:

```
1 {  
2   "id": 5,  
3   "nombre": "Borrador",  
4   "precio": 1.0,  
5   "categoria": "Utiles"  
6 }
```

POST Registrar

No environment

Microservicio Examen / ms-varios / Ms Productos / Registrar

Save

Share

POST

http://localhost:8080/api/productos/registrar

Send

Params Auth Headers (10) Body Scripts Tests Settings

Cookies

raw

JSON

Beautify

```
1 {  
2   "nombre": "Borrador",  
3   "precio": 1.0,  
4   "categoria": "Utiles"  
5 }  
6
```

Body

403 Forbidden

457 ms

234 B

Save Response

{ } JSON

Preview

Visualize

```
1 {  
2   "error": "Acceso denegado. Se requiere rol ADMIN o SUPERADMIN"  
3 }
```

2.2. Listar productos (GET /api/productos/listar)

Lista todos los productos disponibles. Protegido, requiere rol ADMIN o SUPERADMIN.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/api/productos/listar
- Auth Type:** Bearer Token
- Token:** eyJ0eXAIQlJKV1QILCJhbGciOi...
- Status:** 200 OK
- Response Body (JSON):**

```
[
  {
    "id": 2,
    "nombre": "Cuaderno",
    "precio": 3.0,
    "categoria": "Utiles"
  },
  {
    "id": 1,
    "nombre": "Lapiz",
    "precio": 0.5,
    "categoria": "Utiles"
  },
  {
    "id": 4,
    "nombre": "Cartuchera",
    "precio": 2.0,
    "categoria": "Utiles"
  },
  {
    "id": 5,
    "nombre": "Borrador",
    "precio": 1.0,
    "categoria": "Utiles"
  }
]
```

2.3. Actualizar producto (PUT /api/productos/actualizar/{id})

Actualiza un producto por ID. Requiere rol ADMIN o SUPERADMIN.

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://localhost:8080/api/productos/actualizar/1
- Body (JSON):**

```
{
  "nombre": "Lapiz",
  "precio": 0.50,
  "categoria": "Utiles"
}
```
- Response:** 403 Forbidden (62 ms, 234 B)
- Response Body (JSON):**

```
{
  "error": "Acceso denegado. Se requiere rol ADMIN o SUPERADMIN"
}
```

POST RegistrarGET ListarPUT ActualizarDEL Eliminar

Microservicio Examen / ms-varios / Ms Productos / Actualizar

SaveShare

PUThttp://localhost:8080/api/productos/actualizar/1Send

ParamsAuthHeaders (10)BodyScriptsTestsSettingsCookies

rawJSONBeautify

```
1 {
2   "nombre": "Lapiz",
3   "precio": 0.70,
4   "categoria": "Utiles"
5 }
6
```

Body200 OK170 ms223 BSave Response

{ }JSONPreviewVisualize

```
1 {
2   "id": 1,
3   "nombre": "Lapiz",
4   "precio": 0.7,
5   "categoria": "Utiles"
6 }
```


2.4. Eliminar producto (DELETE /api/productos/eliminar/{id})

Elimina un producto por ID. Requiere rol ADMIN o SUPERADMIN.

The screenshot displays a REST client interface with the following components:

- Method and URL:** DELETE `http://localhost:8080/api/productos/eliminar/4`
- Auth Type:** Bearer Token
- Token:** `κR3XyiZSaKILfzi9u87fx4KjHw`
- Response:** 403 Forbidden (51 ms, 234 B)
- Response Body (JSON):**

```
{  "error": "Acceso denegado. Se requiere rol ADMIN o SUPERADMIN"}
```


POST Registrar

GET Listar

PUT Actualizar

DEL Eliminar

No environment

Microservicio Examen / ms-varios / Ms Productos / Eliminar

SaveShare

DELETE

http://localhost:8080/api/productos/eliminar/4

Send

Params

Auth

Headers (8)

Body

Scripts

Tests

Settings

Cookies

Auth Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

eyJ0eXAiOiJKV1QiLCJhbGciOi...

Body

200 OK • 915 ms • 209 B • Save Response

{ } JSON

PreviewVisualize

```
1 {
2   "message": "Producto eliminado exitosamente"
3 }
```

3. Microservicio ms-ordenes: Gestión de Órdenes

3.1. Crear orden (POST /api/ordenes)

Crea una nueva orden asociada al usuario autenticado. Requiere rol USUARIO.

The screenshot shows a REST client interface with the following details:

- Request:**
 - Method: **POST**
 - URL: `http://localhost:8081/api/ordenes`
 - Body (JSON):

```
1 {
2   "usuarioId": 5,
3   "productosIds": [2],
4   "observaciones": "Entrega urgente por favor"
5 }
```
- Response:**
 - Status: **403 Forbidden**
 - Time: 482 ms
 - Size: 235 B
 - Body (JSON):

```
1 {
2   "error": "Acceso denegado. Solo usuarios pueden crear órdenes"
3 }
```

POST Crear Orden

No environment

Microservicio Examen / ms-varios / Ms Ordenes / Crear Orden

Save

Share

POST

http://localhost:8081/api/ordenes

Send

Params Auth Headers (10) Body Scripts Tests Settings

Cookies

raw

JSON

Beautify

```
1 {
2   "usuarioId": 1,
3   "productosIds": [2],
4   "observaciones": "Entrega urgente por favor"
5 }
```



Body

201 Created

392 ms

433 B



Save Response



{ } JSON

Preview

Visualize



```
1 {
2   "id": 5,
3   "usuarioId": 3,
4   "productosIds": [
5     2
6   ],
7   "fecha": "2025-07-14T01:13:38.1811163",
8   "estado": "PENDIENTE",
9   "total": 3.0,
10  "productos": [
11    {
12      "id": 2,
13      "nombre": "Cuaderno",
14      "precio": 3.0,
15      "categoria": "Utiles",
16      "descripcion": null,
17      "disponible": null
18    }
19  ],
20  "mensaje": "Orden creada exitosamente"
21 }
```

3.2. Listar órdenes (GET /api/ordenes)

Lista todas las órdenes creadas. Acceso restringido a usuarios con rol ADMIN o SUPERADMIN. Primera imagen usando Rol de USUARIO y segundo con ROL Valido

The screenshot shows a Postman interface for the endpoint `http://localhost:8081/api/ordenes` using the `GET` method. The **Authorization** tab is selected, showing **Bearer Token** as the auth type. The token field is filled with a masked string. The **Body** tab is also visible, showing a JSON response with an error message: `{ "error": "Acceso denegado. Se requiere rol ADMIN o SUPERADMIN" }`. The status bar indicates a **403 Forbidden** response with a 43 ms latency and 234 B of data.

The screenshot shows the same Postman interface for the endpoint `http://localhost:8081/api/ordenes` using the `GET` method. The **Authorization** tab is selected, showing **Bearer Token** as the auth type. The token field is filled with a masked string. The **Body** tab is selected, showing a JSON response with a list of orders:

```
[
  {
    "id": 4,
    "usuarioId": 3,
    "productosIds": [
      2
    ],
    "fecha": "2025-07-12T23:05:05.618989",
    "estado": "PENDIENTE",
    "total": 3.0,
    "productos": null,
    "mensaje": null
  },
  {
    "id": 3,
    "usuarioId": 3,
    "productosIds": [
      1
    ]
  }
]
```

. The status bar indicates a **200 OK** response with a 171 ms latency and 453 B of data.

4. Eureka Server para microservicios


En este examen de microservicios, **Eureka actúa como el punto central donde se registran los siguientes servicios:**

- ms-auth: Servicio de autenticación y validación de tokens.
- ms-productos: Servicio de gestión de productos.
- ms-ordenes: Servicio de gestión de órdenes.

Todos estos servicios se comunican entre sí utilizando **Feign Client** y Eureka, sin necesidad de conocer la IP exacta del otro servicio.

← → ↺ localhost:8761

☆ 🔄 📄 🗑️ 👤

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2025-07-14T01:09:57 -0500
Data center	default	Uptime	11:50
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	8

DS Replicas

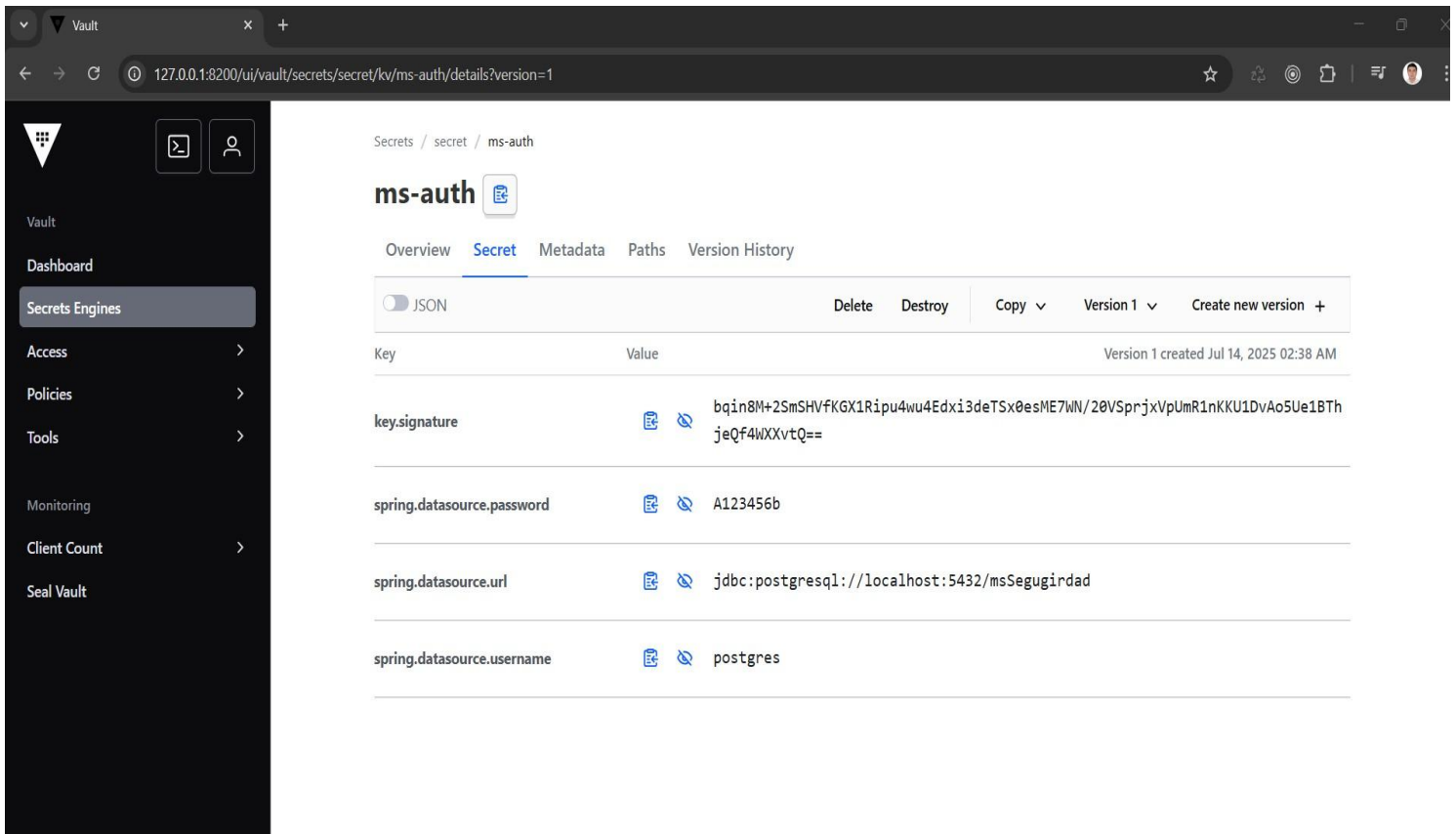
localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
MS-AUTH	n/a (1)	(1)	UP (1) - ms-auth:c5829be1-a74a-41da-8b3d-cc03a192d12a
MS-ORDENES	n/a (1)	(1)	UP (1) - ms-ordenes:641da42c-4488-4f8c-972c-3a1b814d9173
MS-PRODUCTOS	n/a (1)	(1)	UP (1) - ms-productos:6ea00327-c48a-4142-a2da-f653d1ae1ffb

5. HashiCorp Vault en los Microservicios

Cada microservicio (como ms-auth, ms-ordenes, y ms-productos) deben utilizar HashiCorp Vault para gestionar sus secretos sensibles, como claves JWT o credenciales de base de datos. Esto garantiza una mayor seguridad y centralización de la configuración crítica. De momento se probó en el ms-auth dando resultados correctos en cosnomo.



```
Windows PowerShell
namespace="ID: root. Path: "
2025-07-14T02:35:45.979-0500 [INFO] rollback: Starting the rollback manager with 256 workers
2025-07-14T02:35:45.979-0500 [INFO] rollback: starting rollback manager
2025-07-14T02:35:45.980-0500 [INFO] core: restoring leases
2025-07-14T02:35:45.980-0500 [INFO] expiration: lease restore complete
2025-07-14T02:35:45.982-0500 [INFO] identity: entities restored
2025-07-14T02:35:45.982-0500 [INFO] identity: groups restored
2025-07-14T02:35:45.982-0500 [INFO] core: post-unseal setup complete
2025-07-14T02:35:45.982-0500 [INFO] core: vault is unsealed
2025-07-14T02:35:45.995-0500 [INFO] core: successful mount: namespace="" path=secret/ type=kv version="v0.24.0+builtin"
WARNING! dev mode is enabled! In this mode, Vault runs entirely in-memory
and starts unsealed with a single unseal key. The root token is already
authenticated to the CLI, so you can immediately begin using Vault.

You may need to set the following environment variables:

PowerShell:
$env:VAULT_ADDR="http://127.0.0.1:8200"
cmd.exe:
set VAULT_ADDR=http://127.0.0.1:8200

The unseal key and root token are displayed below in case you want to
seal/unseal the Vault or re-authenticate.

Unseal Key: sGdF6/Cb/JUe/RZzPq8XdTnMNbrBeh0kkozzkr7LP4Ug=
Root Token: hvs.AszFhgtRcjeASKRuf30JmaQ5

Development mode should NOT be used in production installations!
```

```
PS C:\Users\jorge> $env:VAULT_ADDR="http://127.0.0.1:8200"
PS C:\Users\jorge> vault kv get secret/ms-auth
=== Secret Path ===
secret/data/ms-auth

===== Metadata =====
Key                               Value
---                               -
created_time                      2025-07-14T07:38:12.5036027Z
custom_metadata                  <nil>
deletion_time                    n/a
destroyed                        false
version                          1

===== Data =====
Key                               Value
---                               -
key.signature                    bqin8M+2SmSHVfKGX1Ripu4wu4Edxi3deTSx0esME7WN/20VSprjxVpUmR1nKKU1DvAo5Ue1BThjeQf4WXXvtQ==
spring.datasource.password       A123456b
spring.datasource.url            jdbc:postgresql://localhost:5432/msSegugirdad
spring.datasource.username       postgres
PS C:\Users\jorge>
```