

Evidencia de Funcionamiento

Link al repositorio: <https://github.com/SchroderCoder/auth-service>

Funcionamiento de contenedor:

```
PS C:\Users\CGare\Documents\projects\pruebatecnica> docker compose up --build
#15 resolving provenance for metadata file
#15 DONE 0.0s
[+] up 5/5
✓ Image pruebatecnica-auth-service      Built      4.7s
✓ Container pruebatecnica-cache-1       Running    0.0s
✓ Container pruebatecnica-mongo-1       Running    0.0s
✓ Container pruebatecnica-mongo-express-1 Running    0.0s
✓ Container auth-service                Recreated  1.6s
Attaching to auth-service, cache-1, mongo-1, mongo-express-1
```

Funcionamiento de test con Pytest

```
>> C:\Users\CGare\Documents\projects\pruebatecnica>
===== test session starts =====
platform linux -- Python 3.11.14, pytest-7.4.4, pluggy-1.6.0
rootdir: /usr/src/app
plugins: cov-4.1.0, mock-3.12.0
collected 12 items

tests/unit/test_login.py ... [ 25%]
tests/unit/test_logout.py ... [ 50%]
tests/unit/test_register.py ... [ 75%]
tests/unit/test_validate_token.py ... [100%]

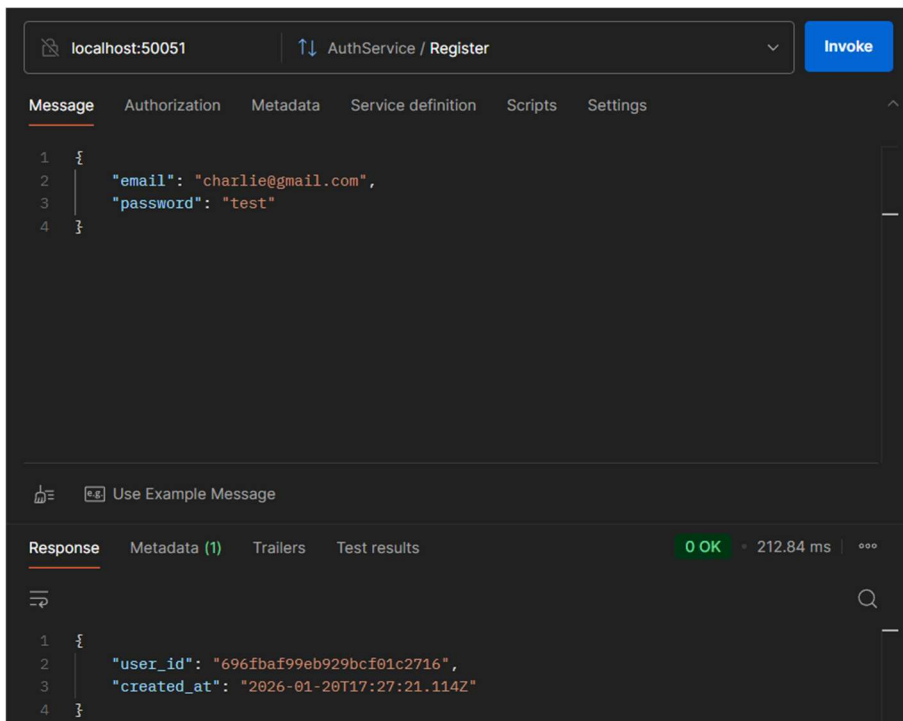
----- coverage: platform linux, python 3.11.14-final-0 -----
Name                               Stmts  Miss  Cover
-----
domain/security/jwt_service.py       13      0  100%
domain/security/password_hasher.py   11      0  100%
domain/services/auth_service.py      34      1   97%
tests/unit/conftest.py               14      0  100%
tests/unit/test_login.py             19      0  100%
tests/unit/test_logout.py            11      0  100%
tests/unit/test_register.py          14      0  100%
tests/unit/test_validate_token.py     13      0  100%
-----
TOTAL                                129      1   99%

===== 12 passed in 0.70s =====
PS C:\Users\CGare\Documents\projects\pruebatecnica>
```

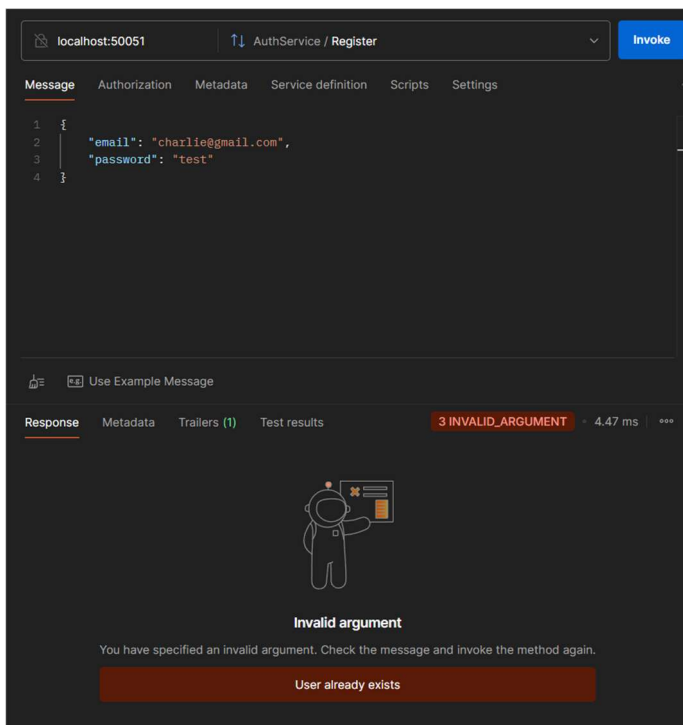
Test en postman

Register

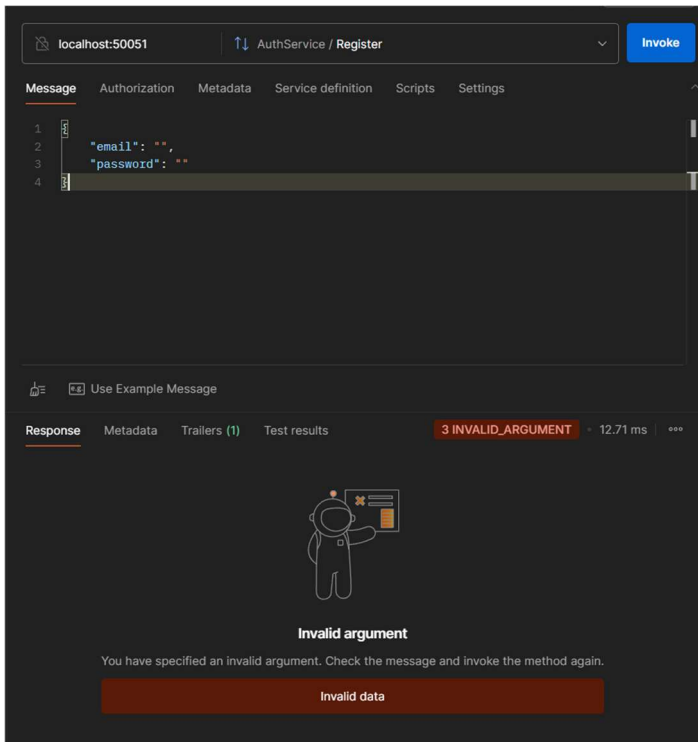
- Registro exitoso de nuevo usuario.



- Error si el correo ya existe.

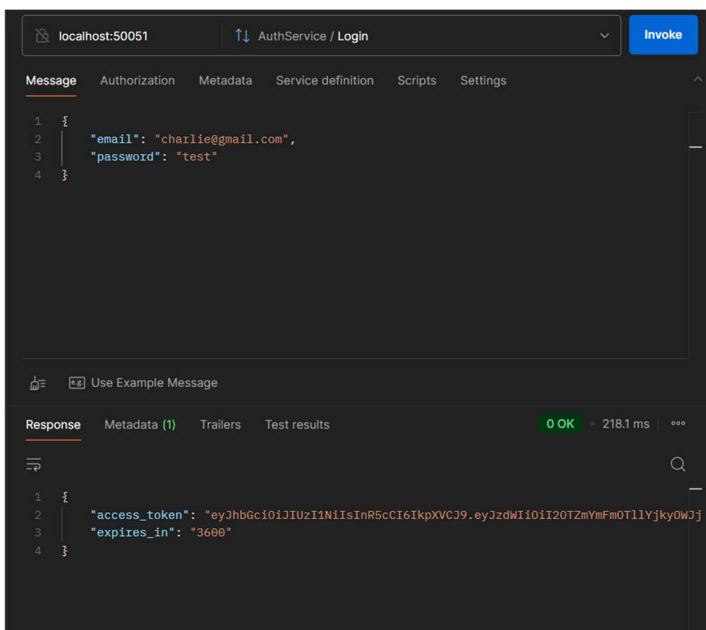


- Error si los campos son inválidos.

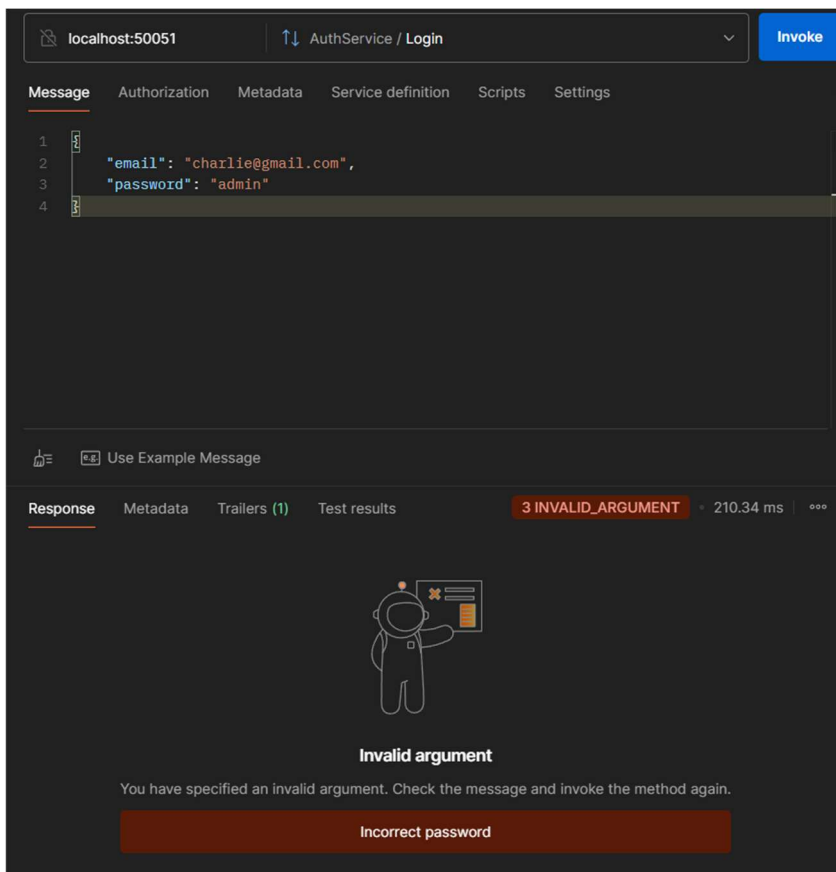


Login

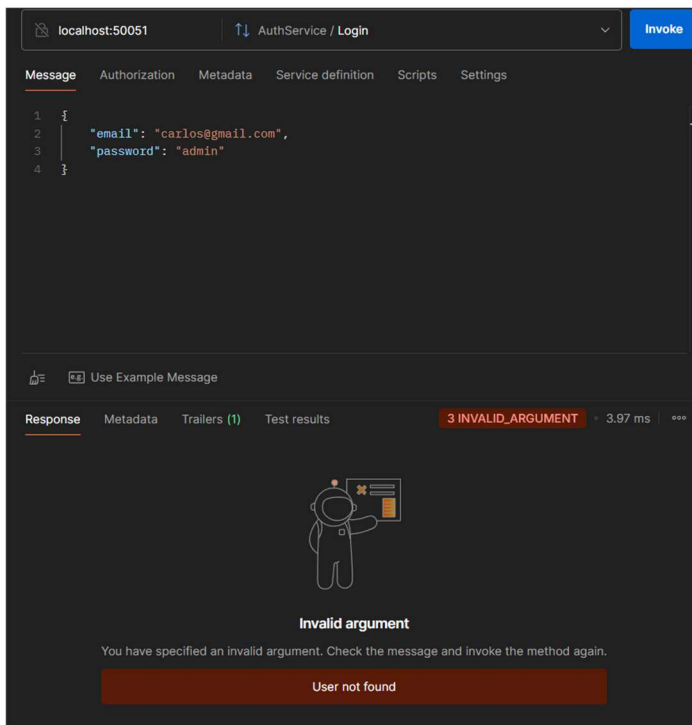
- Login exitoso con credenciales válidas.



- Error con credenciales incorrectas.

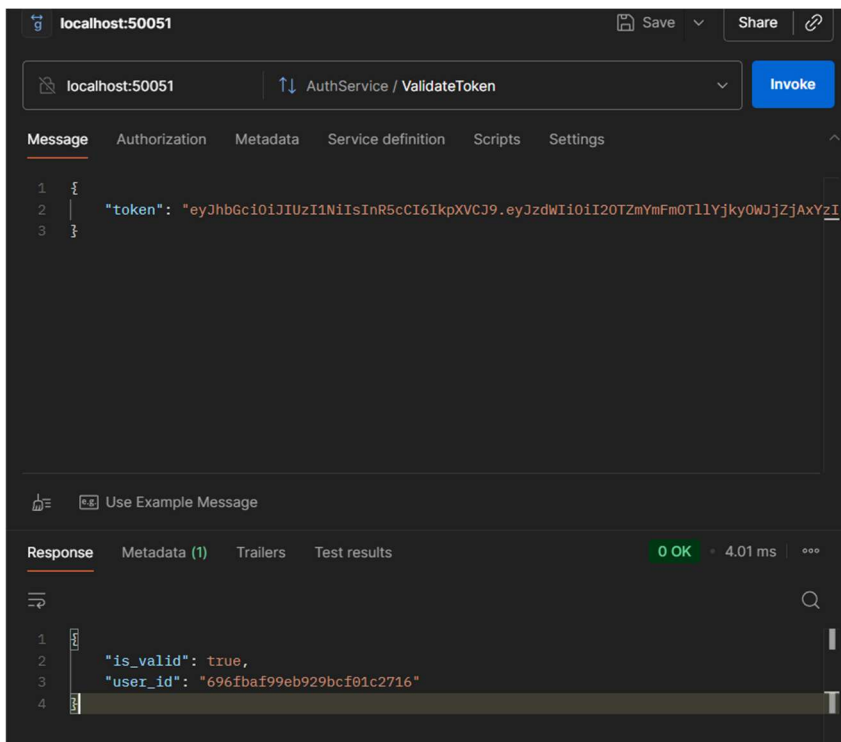


- Error si el usuario no existe.

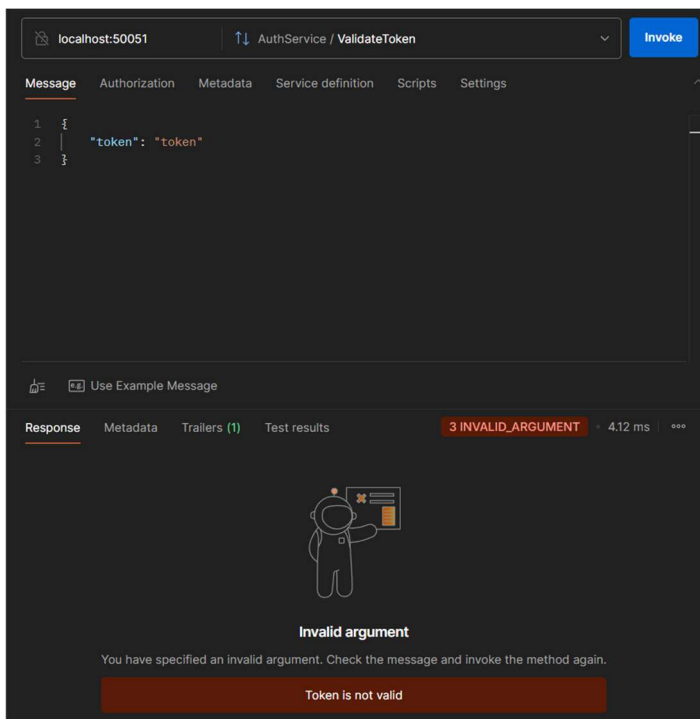


Validate Token

- Token válido devuelve identidad correcta.



- Token expirado o inválido devuelve error.

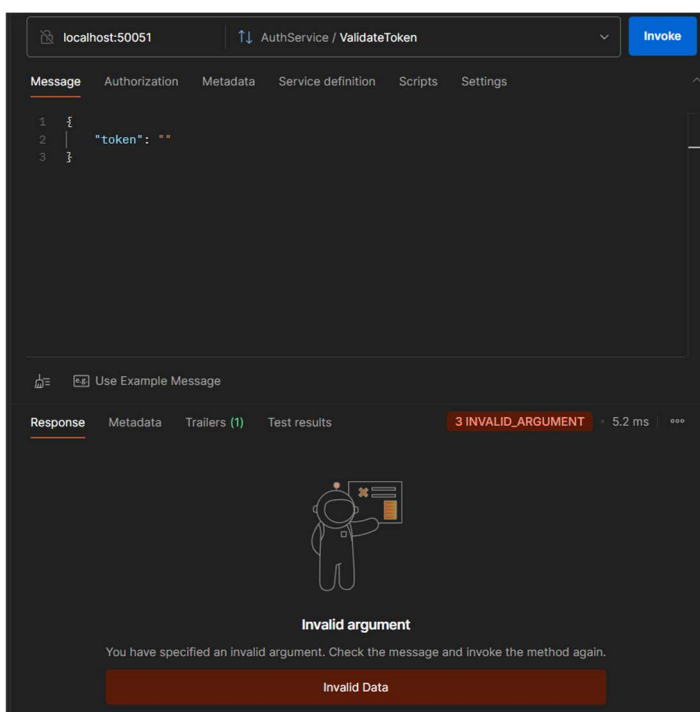


The screenshot shows a REST client interface with the following details:

- URL:** localhost:50051
- Method:** AuthService / ValidateToken
- Message:**

```
1 {  
2   "token": "token"  
3 }
```
- Response:** Metadata Trailers (1) Test results **3 INVALID_ARGUMENT** 4.12 ms
- Error Message:** Invalid argument
You have specified an invalid argument. Check the message and invoke the method again.
Token is not valid

- Token no presente devuelve error.



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

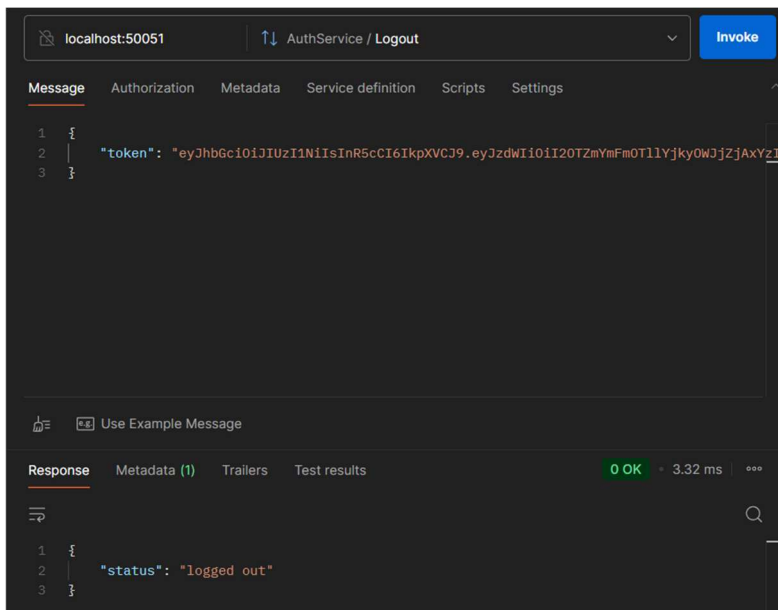
- URL:** localhost:50051
- Method:** AuthService / ValidateToken
- Message:**

```
1 {  
2   "token": ""  
3 }
```
- Response:** Metadata Trailers (1) Test results **3 INVALID_ARGUMENT** 5.2 ms
- Error Message:** Invalid argument
You have specified an invalid argument. Check the message and invoke the method again.
Invalid Data

Logout

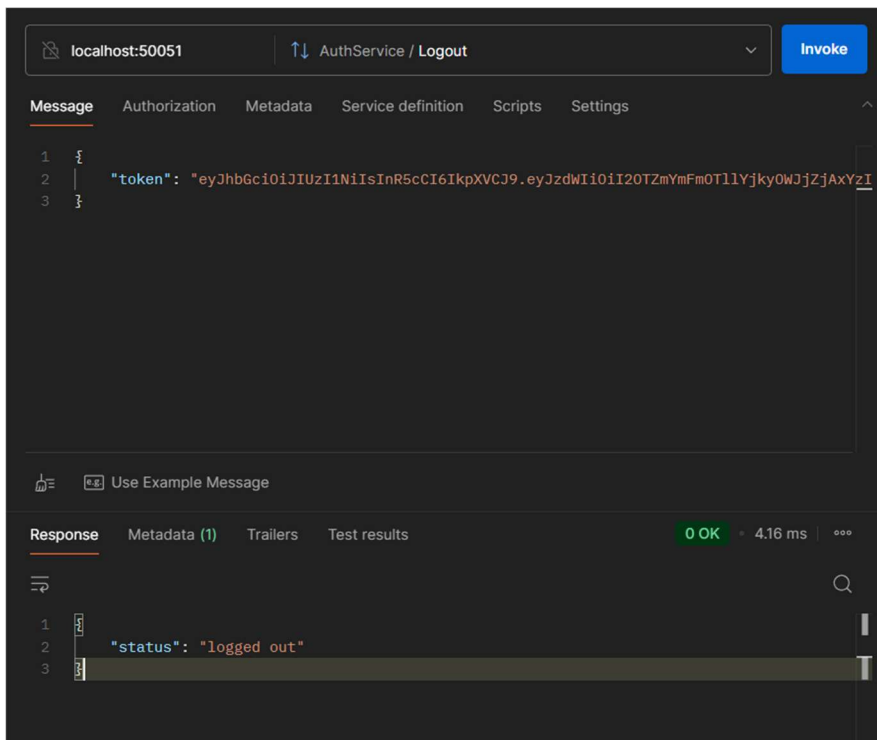
Utilizamos el mismo token obtenido de iniciar sesión

- Token activo es invalidado correctamente.

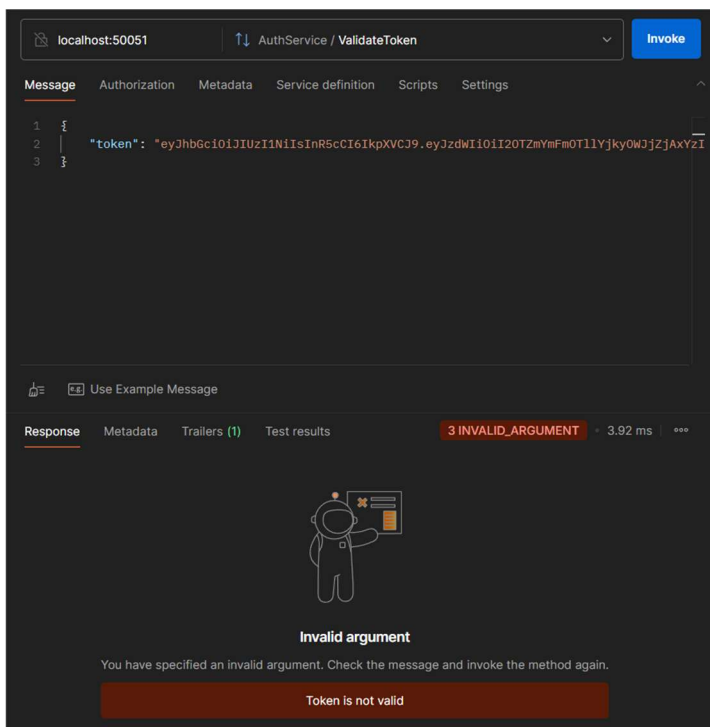


- Token ya invalidado no produce error

crítico.



- Validar que no se pueda usar el token tras
logout.



auth.proto

