

## Evidencias de pruebas funcionales

Proyecto **User creator**

Versión **1.0.0**

Creación de usuario con contraseña mal formada en longitud:

POST

http://localhost:8080/users/registro

Params

Authorization

Headers (8)

Body ●

Pre-request Script

Tests

S



none



form-data



x-www-form-urlencoded



raw



binary



GraphQL

```
1 {
2   "name": "Juan Rodriguez",
3   "email": "juan@rodriguez.org",
4   "password": "hunter2",
5   "phones": [
6     {
7       "number": "1234567",
8       "citycode": "1",
9       "countrycode": "57"
10    },
11    {
12      "number": "9876543",
13      "citycode": "2",
14      "countrycode": "58"
15    }
16  ]
17 }
```

Body

Cookies

Headers (13)

Test Results



Pretty

Raw

Preview

Visualize

JSON



```
1 [
2   {
3     "timestamp": "2023-12-13T19:09:36.672+00:00",
4     "code": 400,
5     "detail": "password : Password debe contener entre 8 a 12"
6   }
7 ]
```

Creación de usuario sin mínimo una mayúscula en password:

POST



http://localhost:8080/users/registro

Params

Authorization

Headers (8)

Body ●

Pre-request Script

Tests

S



none



form-data



x-www-form-urlencoded



raw



binary



GraphQL

```
1  {
2    "name": "Juan Rodriguez",
3    "email": "juan@rodriguez.org",
4    "password": "hunter22",
5    "phones": [
6      {
7        "number": "1234567",
8        "citycode": "1",
9        "countrycode": "57"
10     },
11     {
12       "number": "9876543",
13       "citycode": "2",
14       "countrycode": "58"
15     }
16   ]
17 }
```

Body

Cookies

Headers (13)

Test Results

C

Pretty

Raw

Preview

Visualize

JSON



```
1  {
2    "timestamp": "2023-12-13T19:10:39.344+00:00",
3    "code": 400,
4    "detail": "Password debe tener una letra mayúscula"
5  }
```

Registro exitoso:



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

- GraphQL

Body Cookies Headers (14) Test Results



```
2      "user": {  
3          "id": "21fdcb38-1431-4141-a078-ab94e9737f60",  
4          "name": "Juan Rodriguez",  
5          "email": "juan@rodriguez.org",  
6          "password": "$2a$10$6S7r1R3liuL4ysWjo5Fuj.ULU0GiEezwoteqa  
7          "createAt": "2023-12-13T16:12:05.006242",  
8          "lastLogin": "2023-12-13T16:12:05.006201",  
9          "active": true  
10     },  
11     "token": "eyJhbGciOiJIUzI1NiJ9.  
           eyJzdWIiOiJqdWVudmFuQHJvZHZpZ3Vlei5vcmciaLCJleHAiOjE3MDI0OTgzM
```

En GET: login se implementa el token creado en registro y se usa para acceder:

GET

⌵

http://localhost:8080/users/login

Params

Authorization ●

Headers (9)

Body ●

Pre-request Script

Tests

⋮

Type

Bearer Tok... ⌵

Token

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

Se obtienen los datos del usuario registrado:

GET



http://localhost:8080/users/login

Params

Authorization ●

Headers (9)

Body ●

Pre-request Script

Tests

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

1 {}

Body

Cookies

Headers (14)

Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "id": "21fdcb38-1431-4141-a078-ab94e9737f60",
3   "created": "2023-12-13T16:12:05.006242",
4   "lastLogin": "2023-12-13T16:15:19.295077",
5   "token": "eyJhbGciOiJIUzI1NiJ9.
      eyJzdWIiOiJqdWVzZ3VleWVlcmlLCJleHAiOjE3MDI0OTg1MjE1
      UiJ9fQ.qw4us5uTG9_wAKaWdDTYq_HRuXIGX3Jaft-8bxaQl3Q",
6   "name": "Juan Rodriguez",
7   "email": "juan@rodriguez.org",
8   "password": "$2a$10$6S7r1R3liuL4ysWjo5Fuj.ULU0GiEezwoteqasLDF",
9   "phones": [
10     {
11       "number": 1234567,
12       "citycode": 1,
13       "countrycode": "57"
14     },
15     {
16       "number": 9876543,
17       "citycode": 2,
18       "countrycode": "58"
19     }
20   ],
21   "active": true
22 }
```