# Guia rápida de usuario

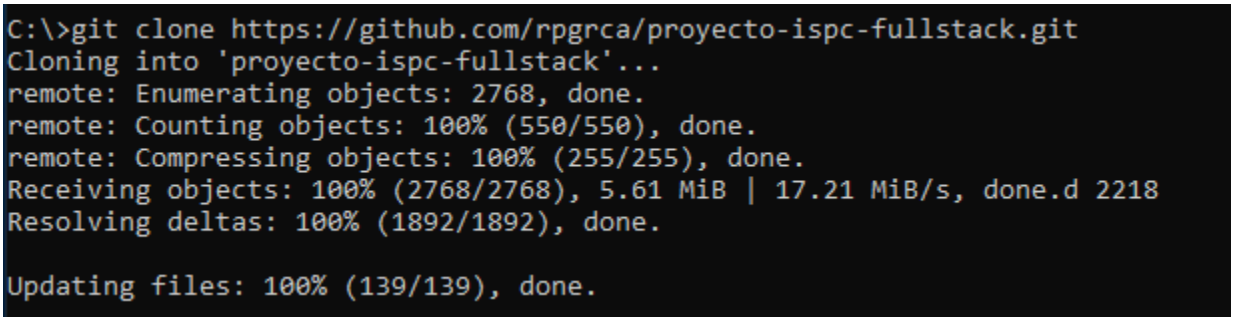
La presente es una guía de usuario para la puesta en funcionamiento del sistema para subastas Bidón.

1. Clonación de repositorio
2. Instalación de paquetes
3. Pruebas unitarias
4. Configuración
5. Ejecución del sistema
6. Utilizando Swagger UI
7. Utilizando Postman

## Clonación de repositorio

Para realizar la clonación del repositorio es suficiente con ejecutar la siguiente línea de comando desde una consola que tenga *git* instalado.

**# git clone https://github.com/rpgrca/proyecto-ispc-fullstack.git**




## Instalación de paquetes

Una vez obtenido el código fuente hay que ingresar al directorio creado e instalar los paquetes desde una consola que tenga *Python* instalado.

**# cd proyecto-ispc-fullstack**
**# pip install -r requirements.txt**

```
C:\>cd proyecto-ispc-fullstack

C:\proyecto-ispc-fullstack>pip install -r requirements.txt
Collecting anyio==3.6.2
  Downloading anyio-3.6.2-py3-none-any.whl (80 kB)
     ---------------------------------------- 80.6/80.6 kB 563.1 kB/s eta 0:00:00
Collecting asgiref==3.4.1
  Downloading asgiref-3.4.1-py3-none-any.whl (25 kB)
Collecting click==8.0.4
  Downloading click-8.0.4-py3-none-any.whl (97 kB)
     ---------------------------------------- 97.5/97.5 kB 1.9 MB/s eta 0:00:00
Collecting contextlib2==21.6.0
  Downloading contextlib2-21.6.0-py2.py3-none-any.whl (13 kB)
```

Asegurarse que todos los paquetes se terminaron de instalar.

```
Successfully installed anyio-3.6.2 asgiref-3.4.1 attrs-22.1.0 click-8.0.4 colorama-0.4.6 contextlib2-21.6.0 contextvars-
2.4 coverage-6.5.0 ddt-1.6.0 fastapi-0.83.0 h11-0.13.0 idna-3.4 immutables-0.19 importlib-metadata-4.8.3 iniconfig-1.1.1
 mysql-connector-2.2.9 mysql-connector-python-8.0.31 packaging-21.3 pluggy-1.0.0 protobuf-3.20.1 py-1.11.0 pydantic-1.9.
2 pyparsing-3.0.9 pytest-7.1.3 pytest-cov-4.0.0 python-multipart-0.0.5 six-1.16.0 sniffio-1.2.0 starlette-0.19.1 tomli-2
.0.1 typing_extensions-4.1.1 uvicorn-0.16.0 zipp-3.6.0
```

## Ejecución de pruebas unitarias

Es posible ejecutar las pruebas unitarias para asegurarse que todas las dependencias han sido correctamente instaladas.

**# pytest main_tests.py**

```
C:\proyecto-ispc-fullstack>pytest main_tests.py
============================================= test session starts =============================================
platform win32 -- Python 3.11.0, pytest-7.1.3, pluggy-1.0.0
rootdir: C:\proyecto-ispc-fullstack
plugins: anyio-3.6.2, cov-4.0.0
collected 223 items

main_tests.py ..................................................................................................... [ 43%]
..................................................................................................................... [ 94%]
.............                                                                                                        [100%]

============================================= 223 passed in 0.69s =============================================
```

## Configuración

Por defecto el sistema correrá con persistencia temporal utilizando listas y diccionarios. Es posible editar main.py y cambiar de persistencia temporal a persistencia en base de datos.

```
#db = CreadorDeBasesDeDatosMySql(["localhost", "root", "gTp8xT2!", "bidon_subastas"]).construir()
db = CreadorDeBasesDeDatosTemporales().construir()
```

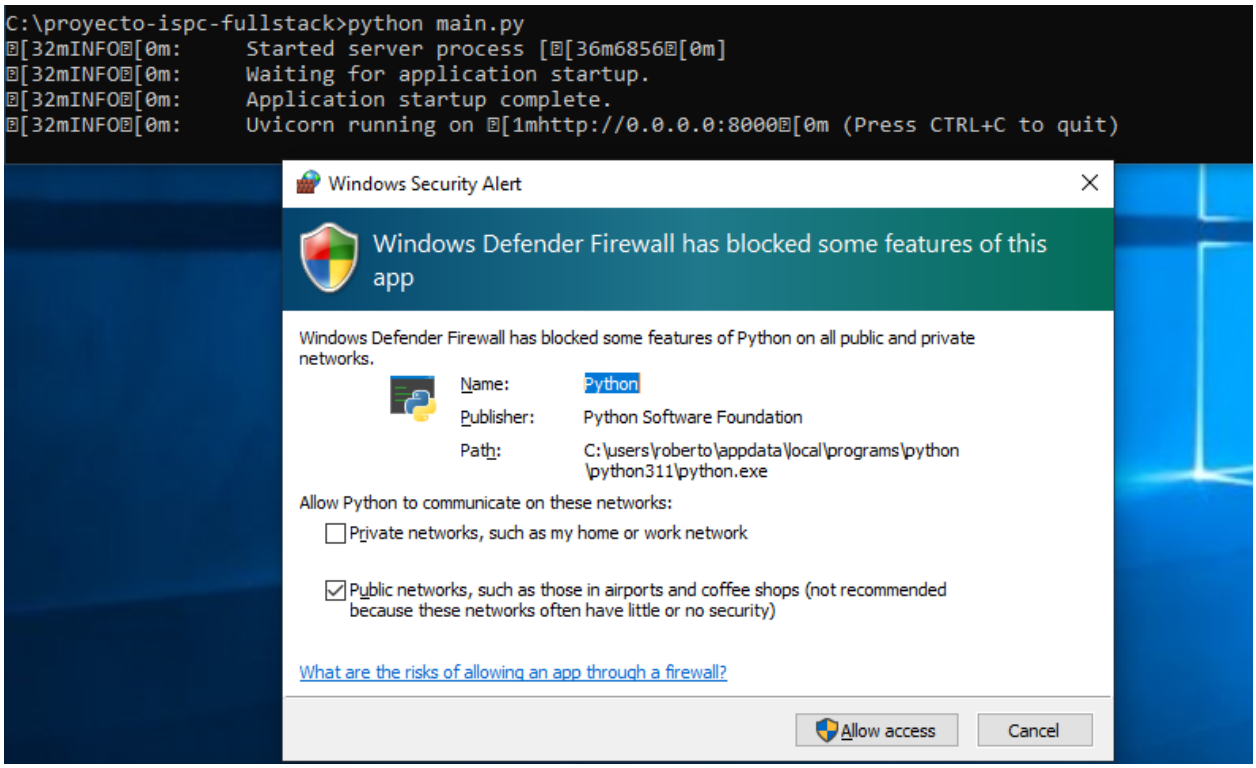
Simplemente se puede descomentar una línea y comentar la otra. Para la base de datos es necesario configurar pasando cuatro datos: el host, el usuario, el password y el nombre de la base de datos a utilizar.

## Ejecución del sistema

Una vez configurado (en caso de ser necesario) se puede ejecutar el sistema utilizando la siguiente línea de comando:

**# python main.py**

En Windows es posible que pida permiso para ejecutar la aplicación.




Una vez ejecutado el programa esperará conexiones por parte del cliente (web browser, por ejemplo).

Si la conexión elegida fue a base de datos MySQL y los datos de conexión son correctos (la base de datos está corriendo y es accesible por el programa) el programa creará su propia base de datos si ésta no existe, o utilizará la existente en caso de que encuentre una. Por consiguiente para borrar los datos grabados en su totalidad y empezar con una base limpia es suficiente borrar la base de datos con una sentencia drop:

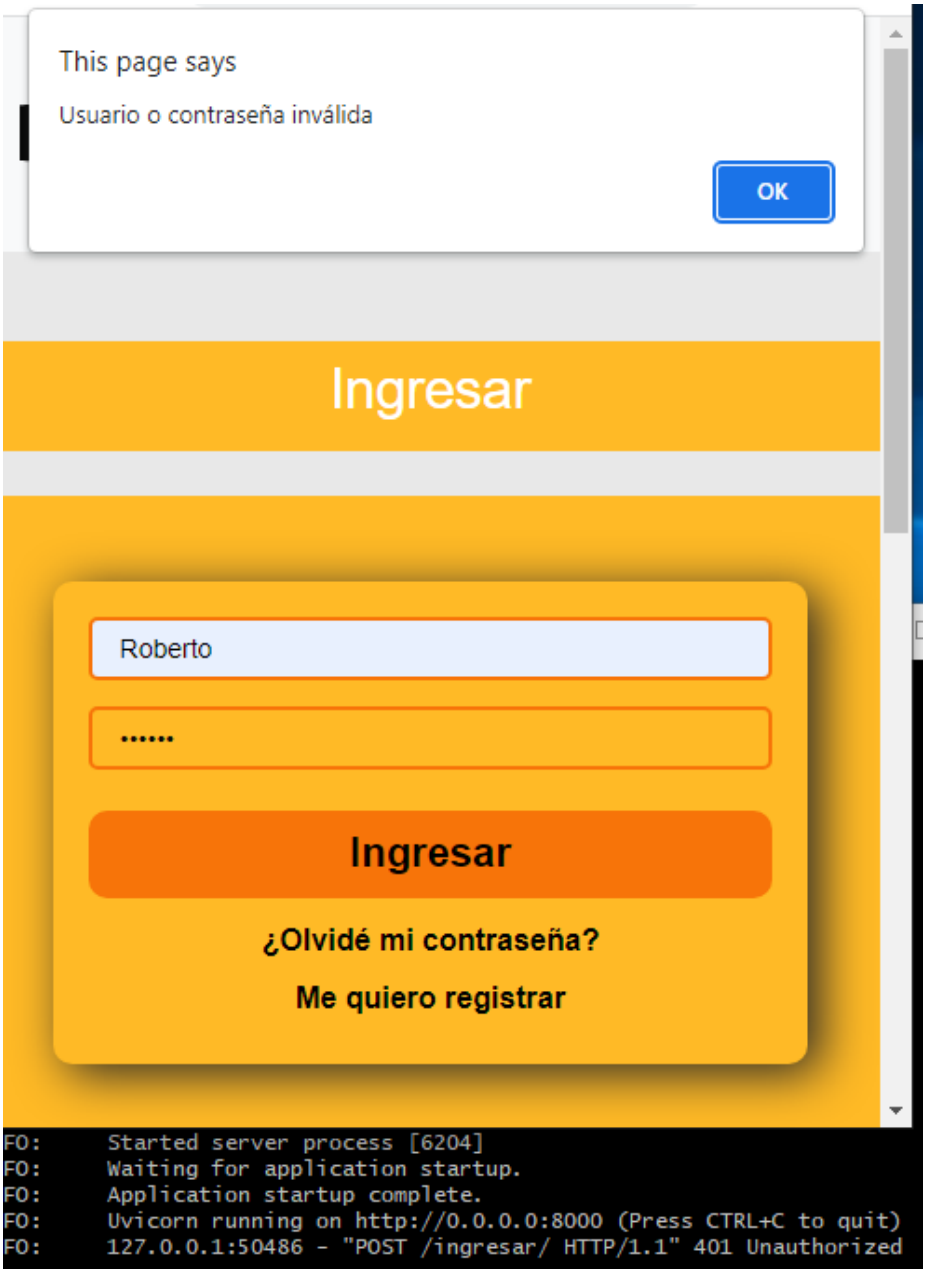
**mysql> drop database bidon_subastas;**

lo que provocará que el sistema vuelva a crear la base de datos y las tablas nuevamente.

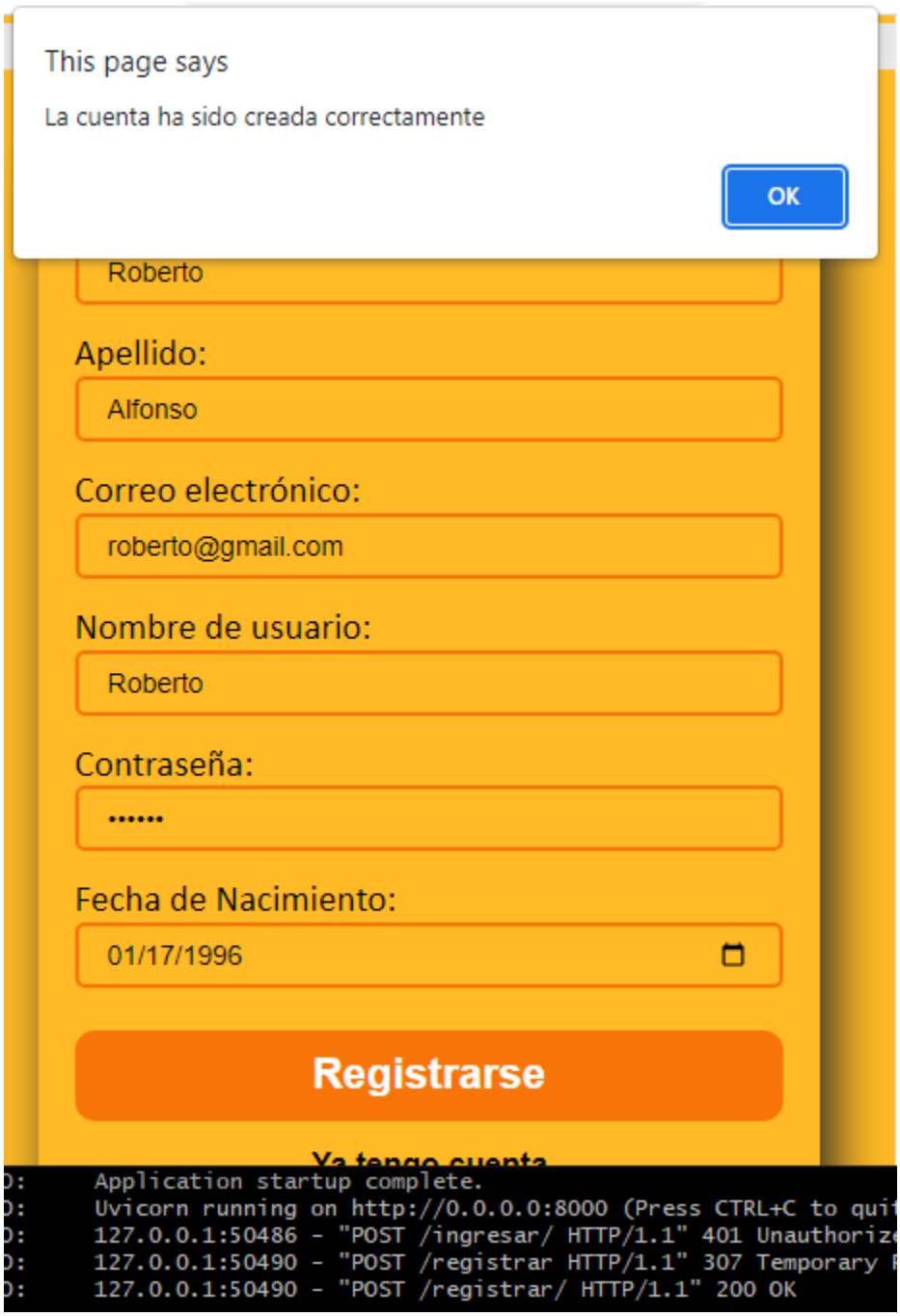
## Utilizando el sitio web

Es posible entrar al directorio de la vista y navegar a partir del archivo index.html. Como prueba de concepto la registración, login y recuperación de claves impactan en el servidor back-end.

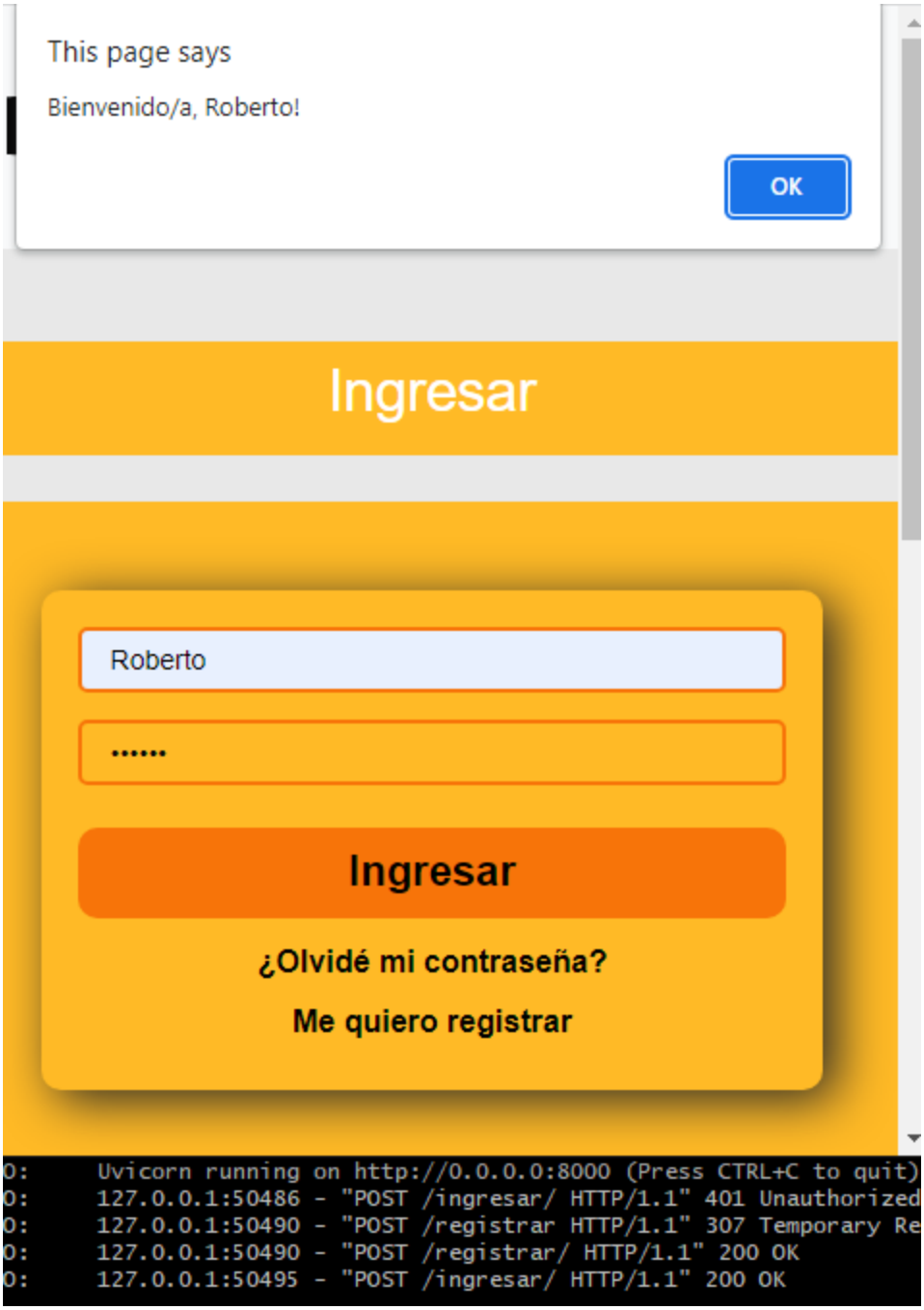
Al intentar realizar el login con una cuenta como “Roberto” con clave “123456” el sistema dirá que no es posible hacerlo, con el servidor retornando un error 401, prohibido.



Uno entonces puede ir a la pantalla de registración y crear dicho usuario completando los datos. También se verá cómo impacta en el servidor, y éste esta vez retornará 200, ok.



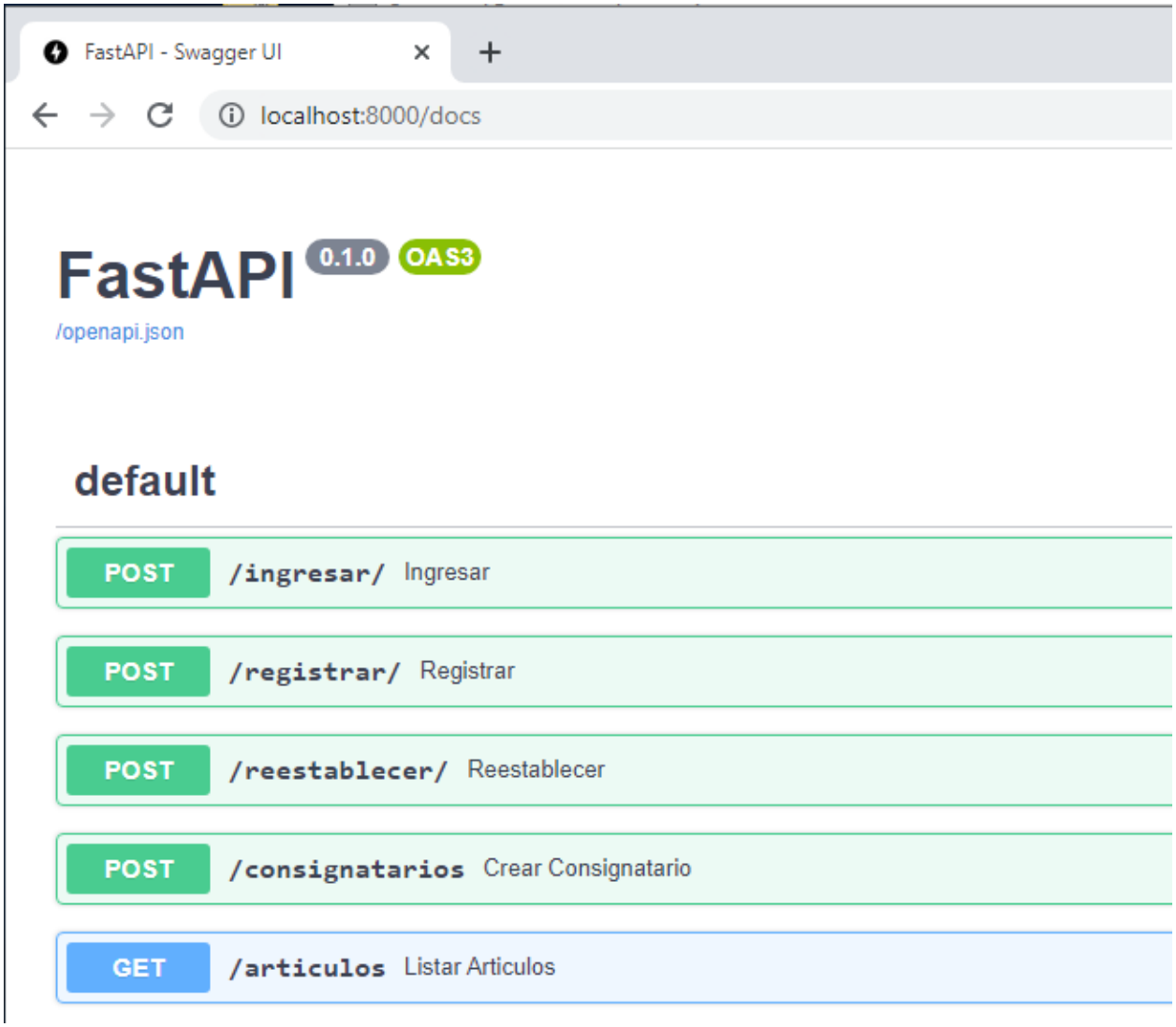

Finalmente al intentar entrar nuevamente con el mismo usuario y clave iniciales, esta vez se le permitirá al usuario ingresar al sistema.



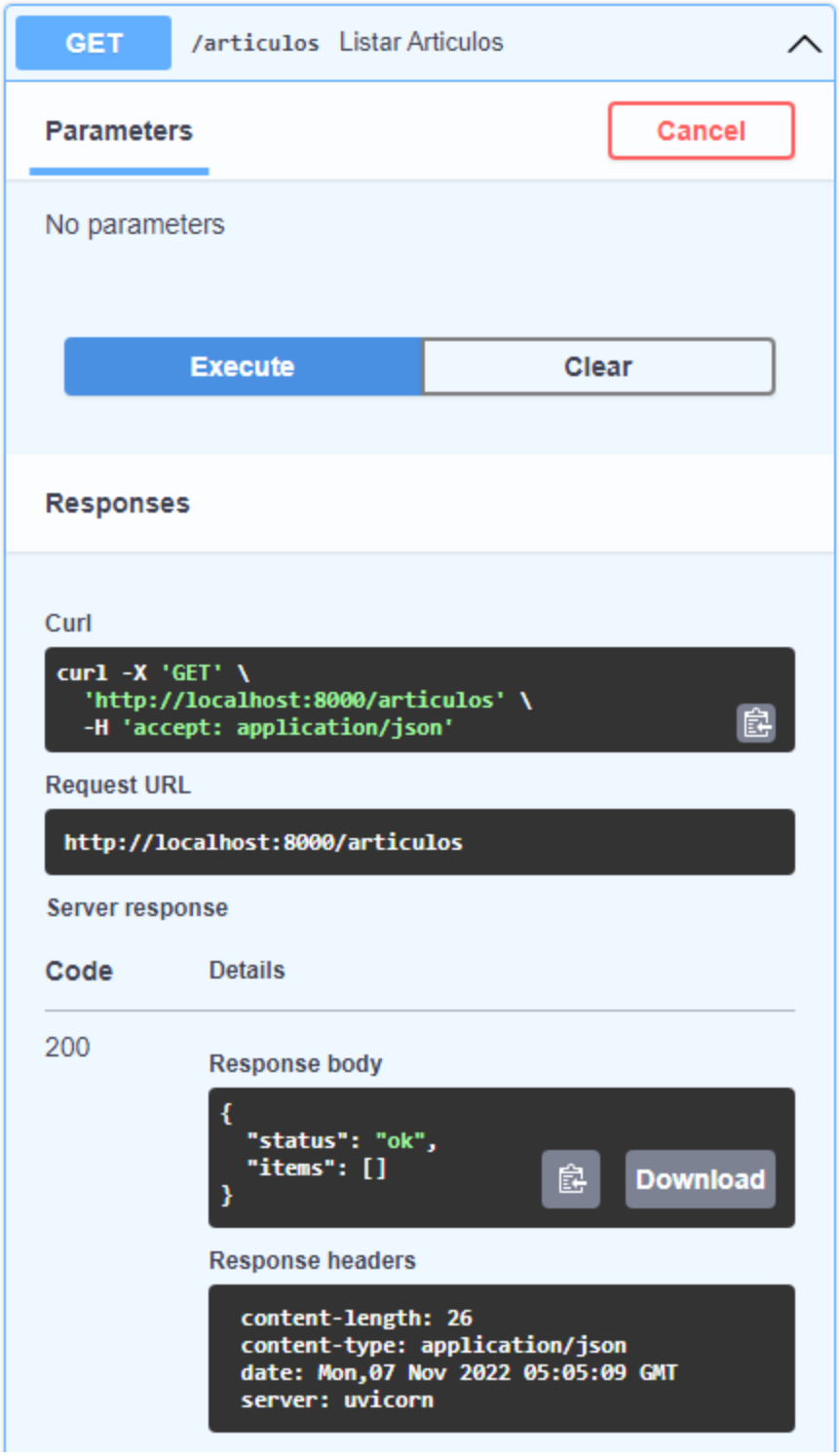

**MUY IMPORTANTE**: Esto es una prueba de concepto, por consiguiente no habrá una redirección a otra página ni se recordará al usuario logueado ya que no implementamos JWT ni similar, las páginas que se podrán navegar no cambiarán ya que el front-end es estático y no estamos utilizando un framework para hacerlo dinámico. Es simplemente una prueba de concepto.

## Utilizando Swagger UI

*FastAPI* tiene incorporado Swagger para poder interactuar con la API directamente desde un navegador, simplemente hay que ir a **localhost:8000/docs** desde uno.




Desde aquí es posible llamar las distintas APIs que posee el sistema. Como fue mencionado anteriormente el sitio web de Bidón solamente utiliza las tres primeras llamadas, el resto de los controladores están creados y funcionando pero no son accedidos desde la página web. Por ejemplo, es posible intentar obtener la lista de artículos utilizando el GET /articulos.




MUY IMPORTANTE: Por defecto se utiliza una base transcient o temporal (trabaja en memoria) por lo que para borrar los datos ingresados solo hay que apagar y prender el servidor como se indicó en el apartado *Ejecución del sistema*.

# Utilizando Postman

En el directorio Documentación se incluye una serie de scripts de prueba de Postman que pueden ser importados dentro de dicho programa. Es posible utilizar dichas pruebas como ejemplos o simplemente correrlas para verificar el funcionamiento del sistema.