



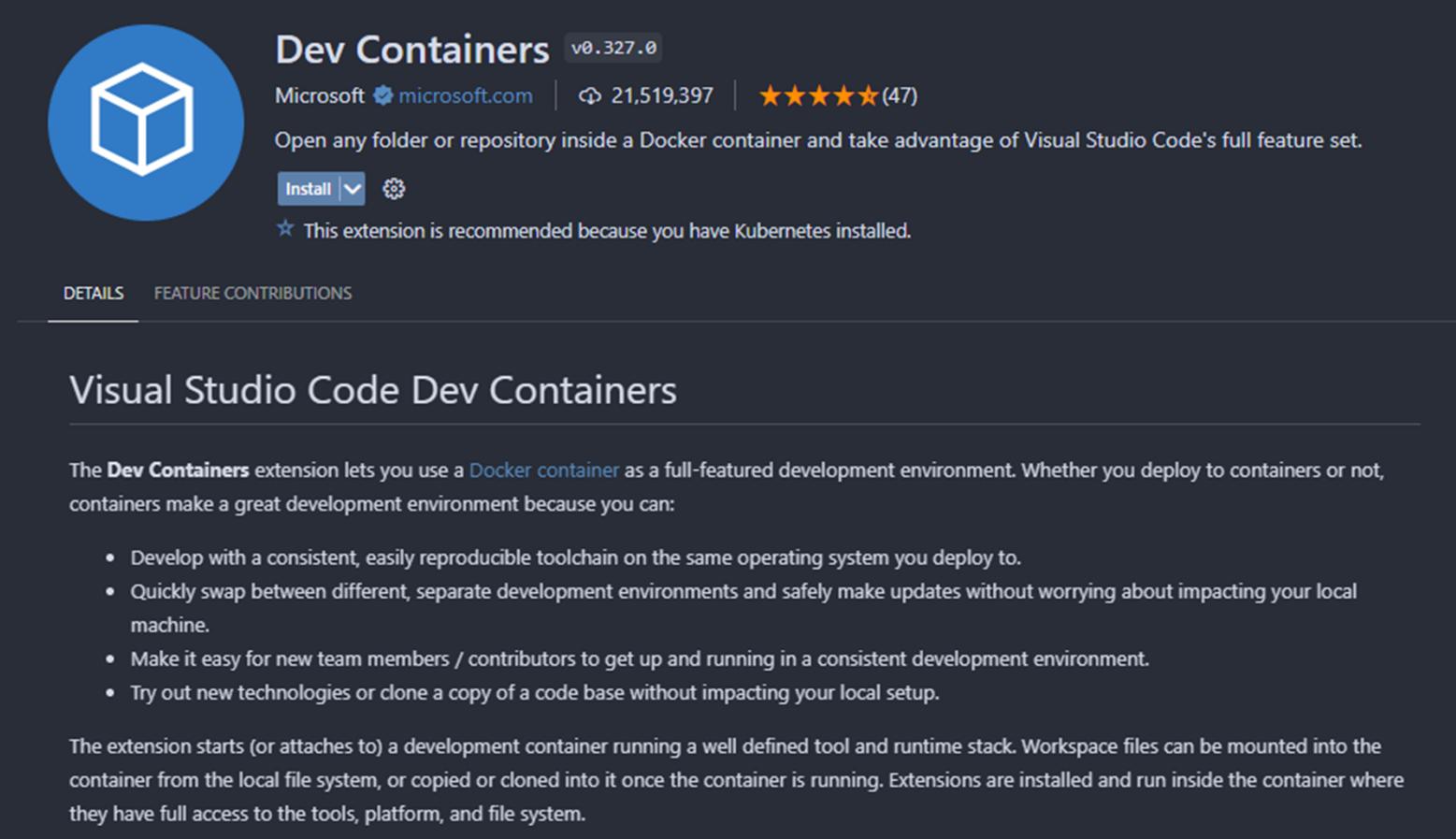
Flask

Framework de Python

¿Qué es Flask?

- Framework web creado por el desarrollador austriaco Armin Ronacher en 2010.
 - Flask es un microframework para Python basado en Werkzeug que permite crear aplicaciones web de todo tipo rápidamente.
- Flask solo incluye el motor de plantillas Jinja y una biblioteca llamada “tool”. Sin embargo, ofrece la posibilidad de integrar funciones de terceros.
- Flask está bajo una licencia BSD. Es gratuito y de código abierto.

Primera aplicación Flask... pero antes una extensión muy útil.



The screenshot shows the Microsoft Store page for the "Dev Containers" extension. The extension is version v0.327.0, developed by Microsoft, with over 21,519,397 installations and a 5-star rating from 47 reviews. The description states: "Open any folder or repository inside a Docker container and take advantage of Visual Studio Code's full feature set." There is an "Install" button and a note: "★ This extension is recommended because you have Kubernetes installed." Below the store page, there is a section titled "Visual Studio Code Dev Containers" which explains the benefits of using Docker containers as a development environment. It lists several advantages, such as consistency, reproducibility, and ease of use for new team members. The text also describes how workspace files can be mounted into the container and how extensions are run inside it.

Dev Containers v0.327.0

Microsoft [microsoft.com](#) | 21,519,397 | ★★★★★(47)

Open any folder or repository inside a Docker container and take advantage of Visual Studio Code's full feature set.

[Install](#) ⚙

★ This extension is recommended because you have Kubernetes installed.

[DETAILS](#) [FEATURE CONTRIBUTIONS](#)

Visual Studio Code Dev Containers

The **Dev Containers** extension lets you use a [Docker container](#) as a full-featured development environment. Whether you deploy to containers or not, containers make a great development environment because you can:

- Develop with a consistent, easily reproducible toolchain on the same operating system you deploy to.
- Quickly swap between different, separate development environments and safely make updates without worrying about impacting your local machine.
- Make it easy for new team members / contributors to get up and running in a consistent development environment.
- Try out new technologies or clone a copy of a code base without impacting your local setup.

The extension starts (or attaches to) a development container running a well defined tool and runtime stack. Workspace files can be mounted into the container from the local file system, or copied or cloned into it once the container is running. Extensions are installed and run inside the container where they have full access to the tools, platform, and file system.

Primera aplicación Flask

```
[kralos]--[y/main =]
[D:\umar\Asignaturas\Semestre 23-24 A\912_TI_1\Notas_TI1\code\flask\project_0]
● ► docker images
REPOSITORY          TAG      IMAGE ID   CREATED    SIZE
python              3.11.7-alpine3.18  8b683f39f4af  6 days ago  52.1MB
gitlab/gitlab-ce   latest   0943abb03ebc  11 days ago  2.87GB
```

The screenshot shows a code editor interface with three tabs: Dockerfile, run.py, and requirements.txt.

- Dockerfile:**

```
FROM python:3.11.7-alpine3.18
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
EXPOSE 3000
CMD python ./run.py
```
- run.py:**

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=int("3000"), debug=True)
```
- requirements.txt:**

```
flask
```

Primera aplicación Flask

```
[kralos]--[¶]main = __name__
[D:\umar\Asignaturas\Semestre 23-24 A\912_TI_1\Notas_TI1\code\flask\project_0]
● ▶ docker build -t fflaskp0 .
[+] Building 18.8s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 174B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.11.7-alpine3.18
=> [1/4] FROM docker.io/library/python:3.11.7-alpine3.18
=> [internal] load build context
=> => transferring context: 235B
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:15e6c3d50e09b6fe3cd0c23332620726dd6b9b08e38f7c868498e16aaa701448
=> => naming to docker.io/library/fflaskp0

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
[kralos]--[¶]main = __name__
[D:\umar\Asignaturas\Semestre 23-24 A\912_TI_1\Notas_TI1\code\flask\project_0]
● ▶ docker images


| REPOSITORY       | TAG               | IMAGE ID     | CREATED        | SIZE   |
|------------------|-------------------|--------------|----------------|--------|
| fflaskp0         | latest            | 15e6c3d50e09 | 55 seconds ago | 68.3MB |
| python           | 3.11.7-alpine3.18 | 8b683f39f4af | 6 days ago     | 52.1MB |
| gitlab/gitlab-ce | latest            | 0943abb03ebc | 11 days ago    | 2.87GB |


```

Primera aplicación Flask

```
[kralos]--[¶main = ● ]
[D:\umar\Asignaturas\Semestre 23-24 A\912_TI_1\Notas_TI1\code\flask\project_0]
● ► docker run --rm -d -p 3000:3000 fflaskp0
ef3c1fb50a2598fb57ad04243a73fce8a610e2691232ec9c357474e0cc8ca60f
```

Logs Inspect Bind mounts Exec Files Stats

2023-12-11 11:06:09 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead

.

2023-12-11 11:06:09 * Running on all addresses (0.0.0.0)

2023-12-11 11:06:09 * Running on http://127.0.0.1:3000

2023-12-11 11:06:09 * Running on http://172.17.0.2:3000

2023-12-11 11:06:09 Press CTRL+C to quit

2023-12-11 11:06:09 * Restarting with stat

2023-12-11 11:06:09 * Debugger is active!

2023-12-11 11:06:09 * Debugger PIN: 117-229-387

2023-12-11 11:06:09 * Serving Flask app 'run'

2023-12-11 11:06:09 * Debug mode: on

2023-12-11 11:06:55 172.17.0.1 - - [11/Dec/2023 17:06:55] "GET / HTTP/1.1" 200 -

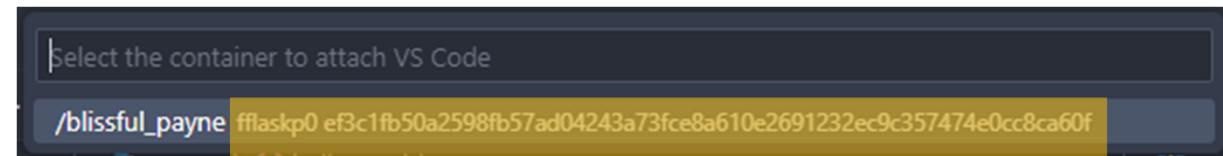
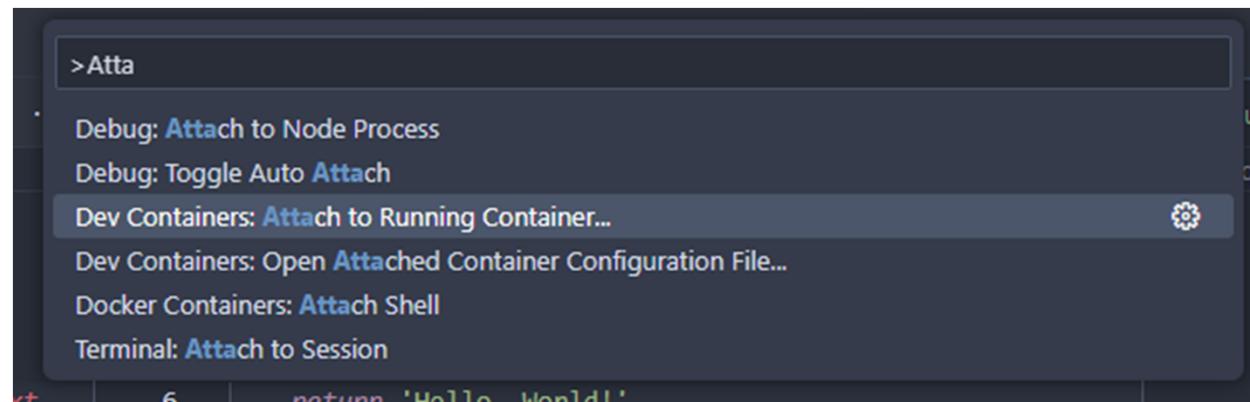


localhost:3000

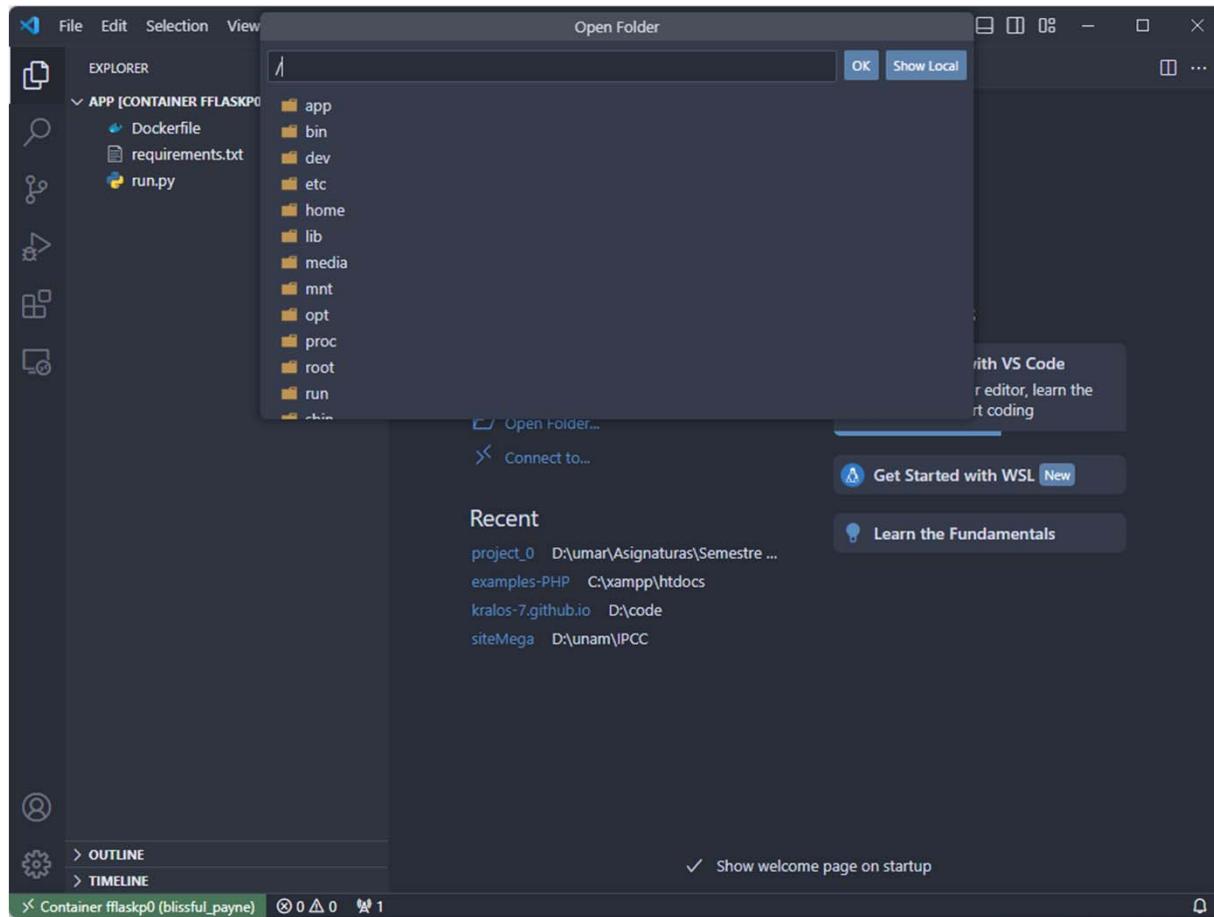
localhost:3000

Hello, World!

Primera aplicación Flask



Primera aplicación Flask



Primera aplicación Flask

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following interface elements:

- File Bar:** File, Edit, Selection, View, ...
- Search Bar:** app [Container fflaskp0 (blissful_payne)]
- Explorer Panel:** Shows a folder named "APP [CONTAINER FFLASKP...]" containing "Dockerfile", "requirements.txt", and "run.py".
- Editor Panel:** Displays the "run.py" file content:

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route('/')
5  def hello_world():
6      return 'Hello, World!'
7
8  if __name__ == "__main__":
9      app.run(host="0.0.0.0", port=int("3000"), debug=True)
```

- Bottom Status Bar:** Container fflaskp0 (blissful_payne), 0 △ 0, 1, Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, Python.
- Bottom Extension Bar:** A notification about the Python extension: "Do you want to install the recommended 'Python' extension from Microsoft for the Python language?". It includes "Install" and "Show Recommendations" buttons.

Primera aplicación Flask

The screenshot shows a code editor interface with a dark theme. On the left, the Explorer sidebar displays a project structure for a Docker container named 'APP [CONTAINER FFLASKP0 (BLISSFUL_PAYNE)]'. The files listed are 'Dockerfile', 'requirements.txt', and 'run.py'. The 'run.py' file is currently open in the main editor area. The code is as follows:

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route('/')
5  def hello_world():
6      return 'Hello, World!, Im in the container 👍'
7
8  if __name__ == "__main__":
9      app.run(host="0.0.0.0", port=int("3000"), debug=True)
```

Below the editor, a browser window is open at 'localhost:3000'. The page content is identical to the code output: 'Hello, World!, Im in the container 👍'.

Primera aplicación Flask

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) running on a Mac. The title bar indicates the window is titled "app [Container flaskp0 (blissful_payne)]".

The Explorer sidebar on the left shows a project structure for "APP [CONTAINER flaskp0 (BLISSFUL_PAYNE)]" containing "Dockerfile", "requirements.txt", and "run.py".

The main editor area displays the content of "run.py":

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!, I'm in the container !'

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=int("3000"), debug=True)
```

The Terminal tab at the bottom shows the output of a command run inside the Docker container:

```
/app # ls
Dockerfile      requirements.txt  run.py
/app #
```

The status bar at the bottom provides information about the file: "Ln 6, Col 49", "Spaces: 4", "UTF-8", "CRLF", "Python 3.11.7 64-bit".

Primera aplicación Flask

The screenshot shows a terminal window within a code editor interface, likely VS Code, connected to a Docker container named 'flaskp0' (alias 'blissful_payne'). The terminal output is as follows:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World! I'm in the container!'

/app # apk add --update nodejs npm
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/community/x86_64/APKINDEX.tar.gz
(1/7) Installing brotli-libs (1.0.9-r14)
(2/7) Installing c-ares (1.19.1-r0)
(3/7) Installing icu-data-en (73.2-r2)
Executing icu-data-en-73.2-r2.post-install
*
* If you need ICU with non-English locales and legacy charset support, install
* package icu-data-full.
*
(4/7) Installing icu-libs (73.2-r2)
(5/7) Installing nghttp2-libs (1.57.0-r0)
(6/7) Installing nodejs (18.18.2-r0)
(7/7) Installing npm (9.6.6-r0)
Executing busybox-1.36.1-r5.trigger
OK: 79 MiB in 47 packages
/app # npm install -D tailwindcss
added 83 packages in 8s

15 packages are looking for funding
  run `npm fund` for details
/app # npx tailwindcss init
Created Tailwind CSS config file: tailwind.config.js
/app #
```

Primera aplicación Flask ...reloaded

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows a project structure under "NOTAS_TI1": code, balanceo, docker-composer, flask\project_0 (containing docker-compose.yml, Dockerfile, requirements.txt, run.py), nodejs, ollama, rasa, docs, imgs, .gitignore.
- docker-compose.yml:** The file content is as follows:

```
services:
  app:
    build: .
    container_name: flask_app0
    command: python ./run.py
    ports:
      - 3000:3000
    volumes:
      - ./app
```

- TERMINAL:** Displays the output of running "docker-compose up".

```
[kralos]--[?]main = ● [D:\umar\Asignaturas\Semestre 23-24 A\912_TI_1\Notas_TI1\code\flask\project_0]
▶ docker-compose up
[+] Building 18.3s (9/9) FINISHED
⇒ [app internal] load .dockerignore
⇒ ⇒ transferring context: 2B
⇒ [app internal] load build definition from Dockerfile
⇒ ⇒ transferring Dockerfile: 174B
⇒ [app internal] load metadata for docker.io/library/python:3.11.7-alpine3.18
⇒ [app 1/4] FROM docker.io/library/python:3.11.7-alpine3.18
⇒ [app internal] load build context
⇒ ⇒ transferring context: 296B
⇒ CACHED [app 2/4] WORKDIR /app
⇒ [app 3/4] COPY . /app
⇒ [app 4/4] RUN pip install -r requirements.txt
⇒ [app] exporting to image
⇒ ⇒ exporting layers
⇒ ⇒ writing image sha256:37deca3f45040365bc388ef55a6a75c681b94caf4f9977f8c687854e60bbe702
⇒ ⇒ naming to docker.io/library/project_0-app
[+] Running 2/2
✓ Network project_0_default Created
✓ Container flask_app0 Created
● Attaching to flask_app0
flask_app0 | * Serving Flask app 'run'
flask_app0 | * Debug mode: on
flask_app0 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
flask_app0 | * Running on all addresses (0.0.0.0)
flask_app0 | * Running on http://127.0.0.1:3000
flask_app0 | * Running on http://172.18.0.2:3000
flask_app0 | Press CTRL+C to quit
flask_app0 | * Restarting with stat
flask_app0 | * Debugger is active!
flask_app0 | * Debugger PIN: 875-710-434
flask_app0 | 172.18.0.1 - - [11/Dec/2023 18:13:48] "GET / HTTP/1.1" 200 -
```

Primera aplicación Flask ...reloaded

```
● Attaching to flask_app0
flask_app0 | * Serving Flask app 'run'
flask_app0 | * Debug mode: on
flask_app0 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
flask_app0 | * Running on all addresses (0.0.0.0)
flask_app0 | * Running on http://127.0.0.1:3000
flask_app0 | * Running on http://172.18.0.2:3000
flask_app0 | Press CTRL+C to quit
flask_app0 | * Restarting with stat
flask_app0 | * Debugger is active!
flask_app0 | * Debugger PIN: 875-710-434
flask_app0 | 172.18.0.1 - - [11/Dec/2023 18:13:48] "GET / HTTP/1.1" 200 -
Gracefully stopping... (press Ctrl+C again to force)
[+] Stopping 1/1
  ✓ Container flask_app0 Stopped
flask_app0 exited with code 0
canceled
[[kralos]--[▶ main ≡ ● ][xSIGINT]
[D:\umar\Asignaturas\Semestre 23-24 A\912_TI_1\Notas_TI1\code\flask\project_0]
● ▶ docker-compose down
[+] Running 2/2
  ✓ Container flask_app0 Removed
  ✓ Network project_0_default Removed
```

Primera aplicación Flask ...reloaded

- Task:
 - Agregar el “tailwind”
 - Buscar un ejemplo de implementación de tailwind en flask

Referencias

- j2logo. (2019, February 25). *Tutorial de Flask en español: Desarrollando una aplicación web en Python*. J2LOGO. <https://j2logo.com/tutorial-flask-espanol/>
- *¿Qué es Flask Python? Un breve tutorial sobre este microframework*. (2023, March 1). IONOS Digital Guide; IONOS. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/flask/>