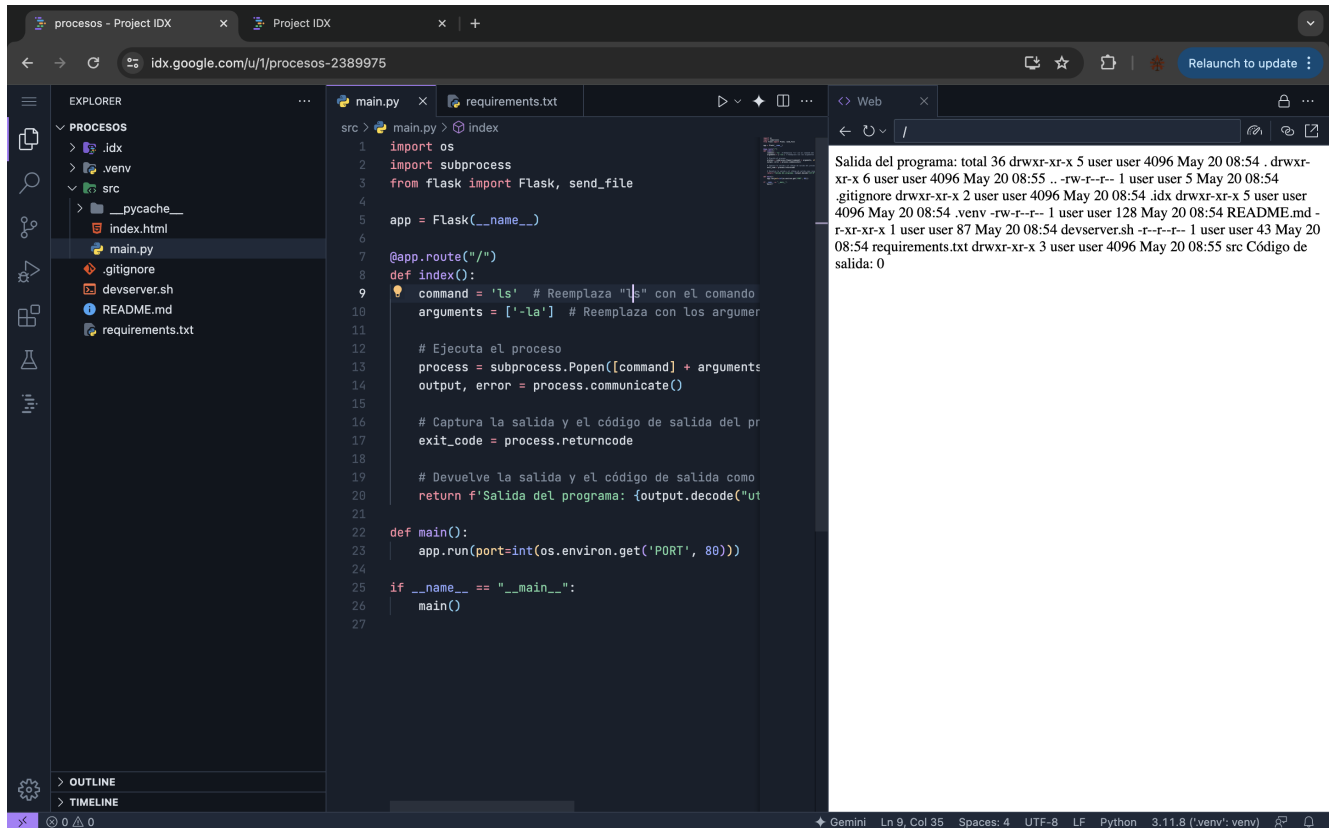


Lanzar un comando en Python con Flask

[crayon-6a9d4ed75e18b742191655/]



The screenshot shows a web browser with the URL `idx.google.com/u/1/procesos-2389975`. The browser displays the output of a Flask application. The output is a JSON object representing the output of the command `ls -la`.

```
Salida del programa: total 36 drwxr-xr-x 5 user user 4096 May 20 08:54 . drwxr-xr-x 6 user user 4096 May 20 08:55 .. -rw-r--r-- 1 user user 5 May 20 08:54 .gitignore drwxr-xr-x 2 user user 4096 May 20 08:54 .idx drwxr-xr-x 5 user user 4096 May 20 08:54 .venv -rw-r--r-- 1 user user 128 May 20 08:54 README.md -r-xr-xr-x 1 user user 87 May 20 08:54 devserver.sh -r--r-- 1 user user 43 May 20 08:54 requirements.txt drwxr-xr-x 3 user user 4096 May 20 08:55 src Código de salida: 0
```

The code in the background is a Flask application that runs on port 80. It has a single route `/` that executes the command `ls -la` and returns the output as a JSON object.

```
1 import os
2 import subprocess
3 from flask import Flask, send_file
4
5 app = Flask(__name__)
6
7 @app.route("/")
8 def index():
9     command = 'ls' # Reemplaza "ls" con el comando
10    arguments = ['-la'] # Reemplaza con los argumentos
11
12    # Ejecuta el proceso
13    process = subprocess.Popen([command] + arguments)
14    output, error = process.communicate()
15
16    # Captura la salida y el código de salida del proceso
17    exit_code = process.returncode
18
19    # Devuelve la salida y el código de salida como un objeto JSON
20    return f'Salida del programa: {output.decode("utf-8")} Código de salida: {exit_code}'
21
22 def main():
23     app.run(port=int(os.environ.get('PORT', 80)))
24
25 if __name__ == "__main__":
26     main()
27
```