

PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática
Lucas Sampaio Leite



Criando um ambiente virtual e instalando o Flask

- Criar o ambiente virtual e instalar o Flask: **pipenv install Flask**

The screenshot shows a VS Code interface with a terminal window open. The terminal displays the output of a series of commands used to set up a virtual environment for a project named 'Projeto_Flask'. The output includes the creation of the virtual environment, the installation of Flask, and the installation of dependencies from a Pipfile.lock. Two green arrows point to the success messages: 'Successfully created virtual environment!' and 'Installing Flask... Installation Succeeded'.

```
clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy,
app_data_dir=/home/lucas/snap/code/192/.local/share/virtualenv)
added seed packages: pip==25.0.1, setuptools==75.3.2, wheel==0.45.1
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

✓ Successfully created virtual environment!
Virtualenv location: /home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG
Creating a Pipfile for this project...
Pipfile.lock not found, creating...
Locking [packages] dependencies...
Locking [dev-packages] dependencies...
Updated Pipfile.lock (7f7606f08e0544d8d012ef4d097dabdd6df6843a28793eb6551245d4b2db4242)!
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing Flask...
✓ Installation Succeeded
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (db4242)...
All dependencies are now up-to-date!
Upgrading Flask in dependencies.
Building requirements...
Resolving dependencies...
✓ Success!
Building requirements...
Resolving dependencies...
✓ Success!
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (ac8e32)...
All dependencies are now up-to-date!
Installing dependencies from Pipfile.lock (ac8e32)...
```

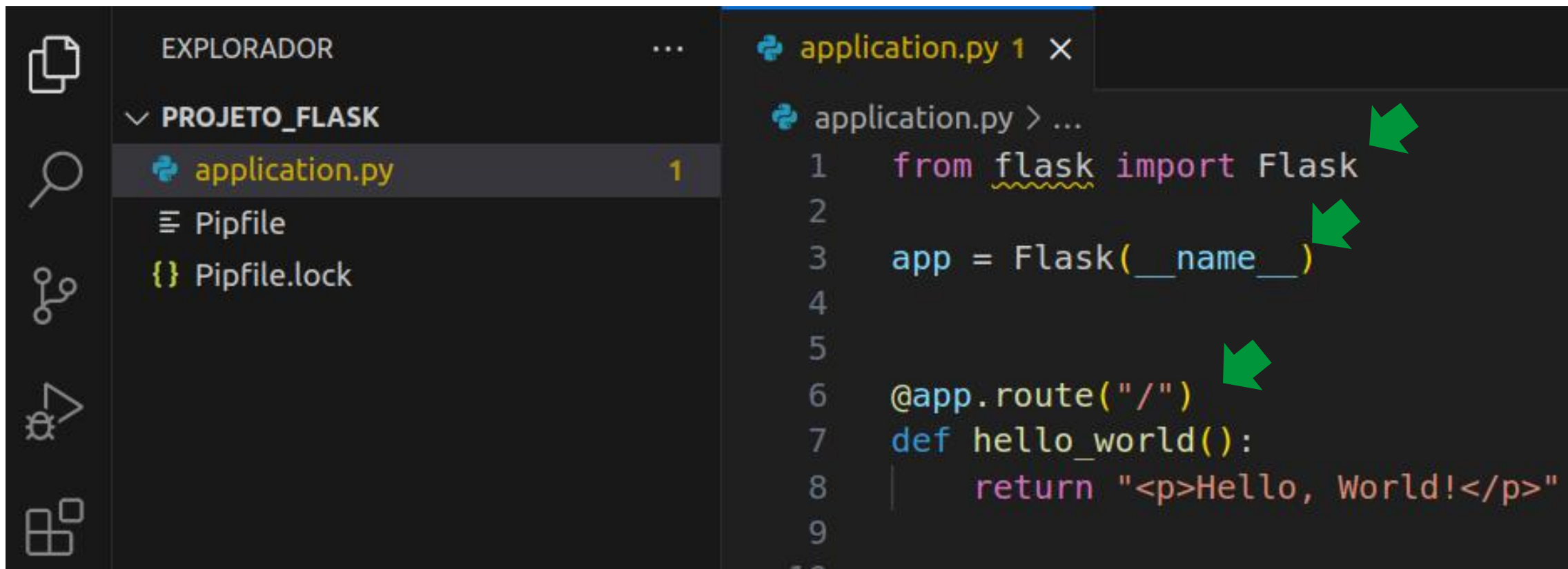
Verificando a instalação do Flask e suas dependências

- pipenv graph

```
Flask==3.0.3
├── blinker
├── click
├── importlib_metadata
│   └── zipp
├── itsdangerous
├── Jinja2
│   └── MarkupSafe
└── Werkzeug
    └── MarkupSafe
```

<https://flask.palletsprojects.com/en/stable/installation/#dependencies>

Primeiros passos com Flask



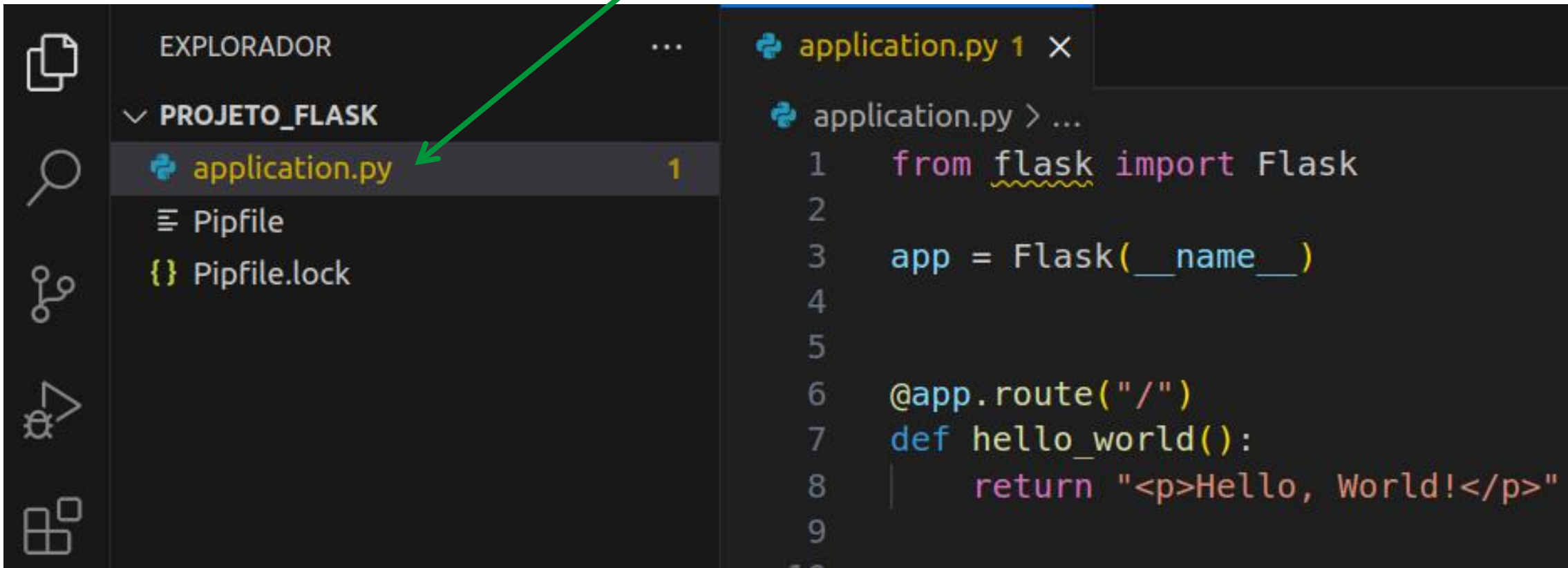
The image shows a code editor interface with a dark theme. On the left, the 'EXPLORADOR' (Explorer) sidebar shows a project named 'PROJETO_FLASK'. Inside this project, the file 'application.py' is selected and highlighted in yellow. Below it, 'Pipfile' and 'Pipfile.lock' are listed. The main editor area shows the code for 'application.py'. The code is as follows:

```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5
6  @app.route("/")
7  def hello_world():
8      return "<p>Hello, World!</p>"
9
```

Four green arrows point to specific parts of the code: the first points to 'Flask' in the import statement, the second points to the second underscore in '__name__', the third points to the opening parenthesis of the '@app.route' decorator, and the fourth points to the opening curly brace of the 'def hello_world()' function.

Primeiros passos com Flask

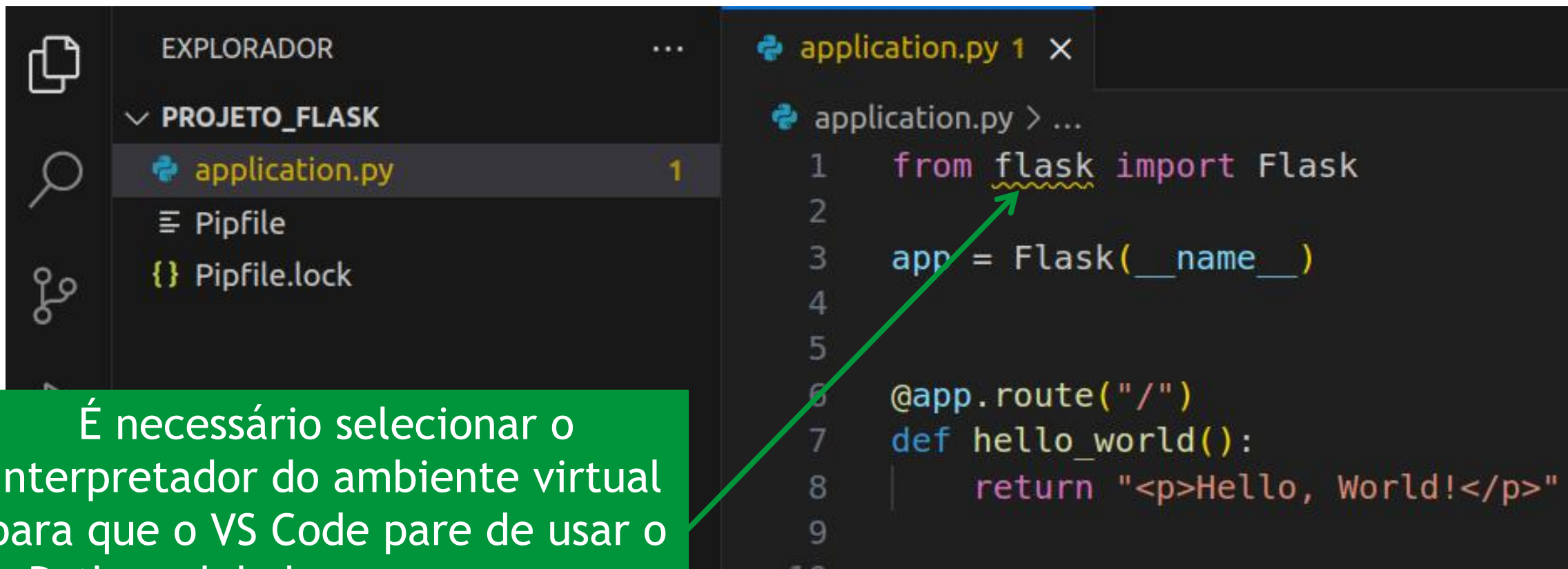
Não nomear como flask.py.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project named 'PROJETO_FLASK' containing a file 'application.py' (highlighted with a green arrow), a 'Pipfile', and a 'Pipfile.lock'. The main editor area displays the code for 'application.py'.

```
application.py 1 X
application.py > ...
1  from flask import Flask
2
3  app = Flask(__name__)
4
5
6  @app.route("/")
7  def hello_world():
8      return "<p>Hello, World!</p>"
9
```

Primeiros passos com Flask



EXPLORADOR

PROJETO_FLASK

application.py 1

Pipfile

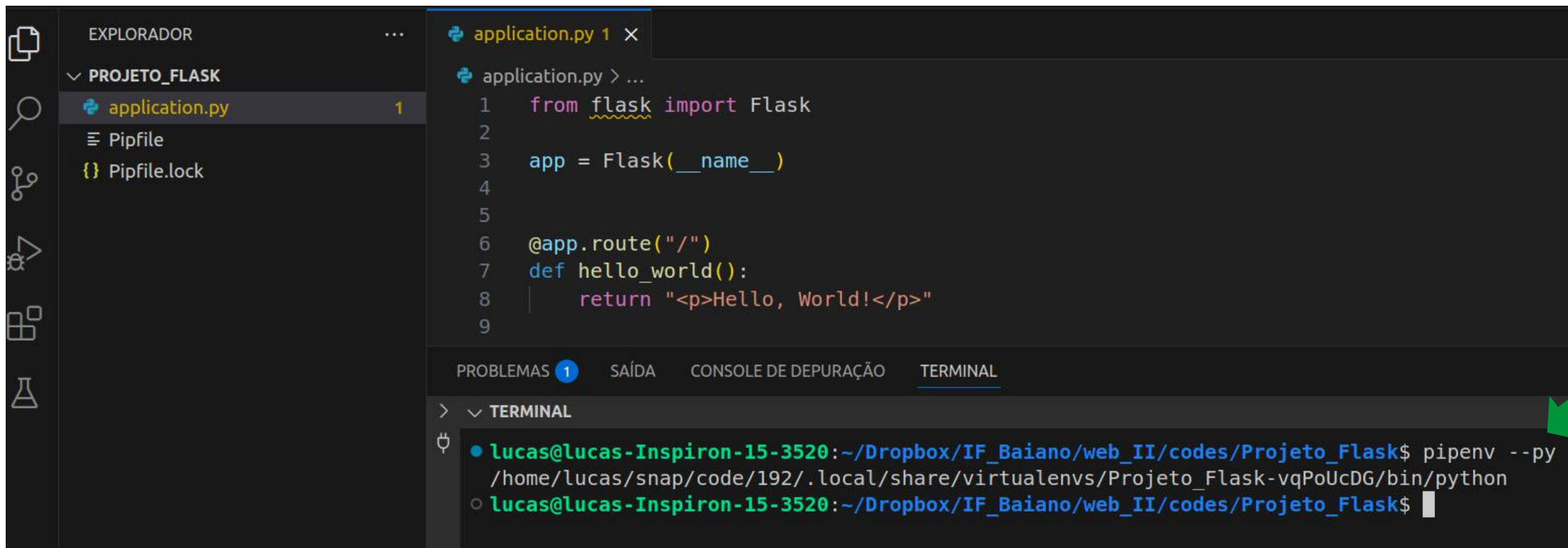
Pipfile.lock

```
application.py 1 X
application.py > ...
1  from flask import Flask
2
3  app = Flask(__name__)
4
5
6  @app.route("/")
7  def hello_world():
8      return "<p>Hello, World!</p>"
9
10
```

É necessário selecionar o interpretador do ambiente virtual para que o VS Code pare de usar o Python global e passe a usar o ambiente isolado com suas dependências.

Primeiros passos com Flask

- Obtendo o caminho do ambiente virtual: `pipenv --py`

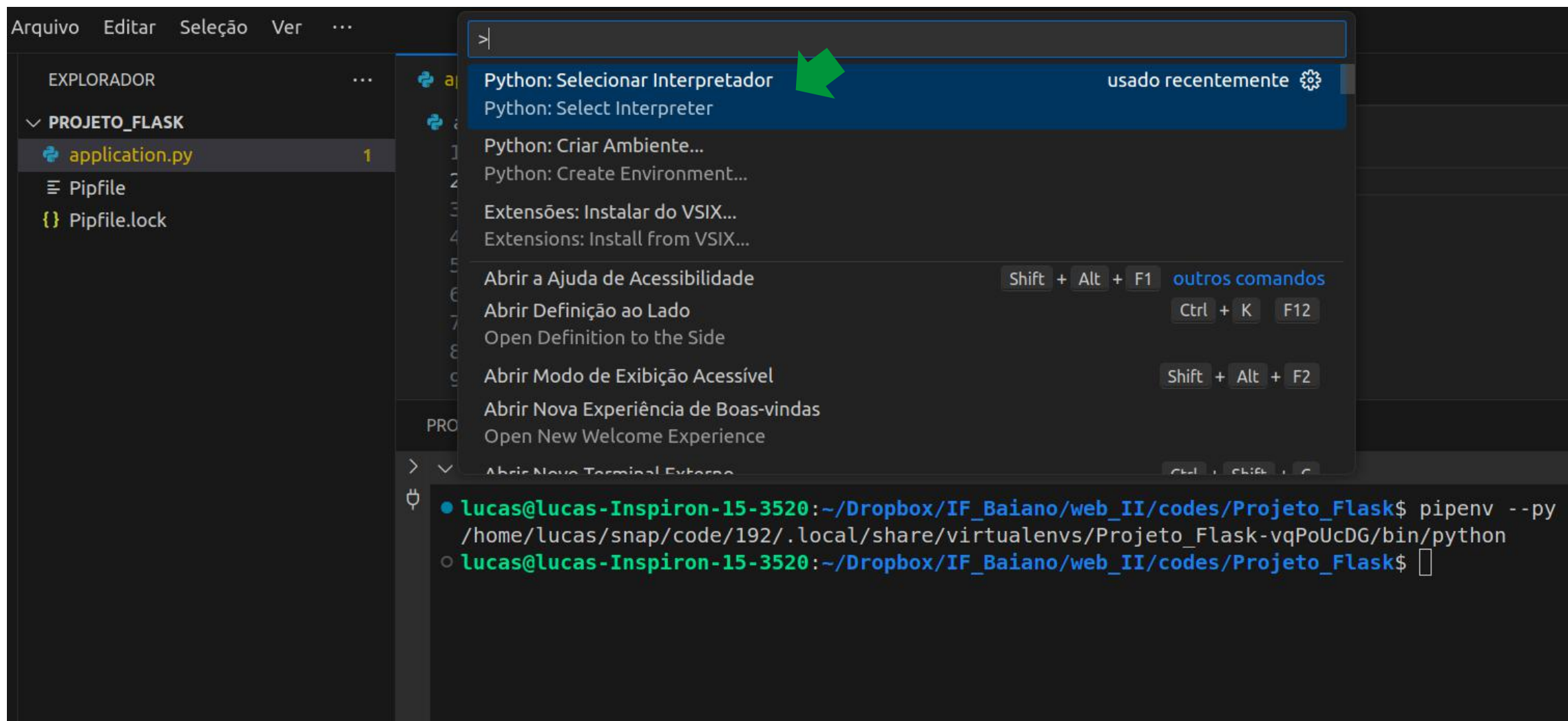


The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project named 'PROJETO_FLASK' containing 'application.py', 'Pipfile', and 'Pipfile.lock'. The main editor displays the contents of 'application.py', which is a simple Flask application with a single route. The bottom panel shows the 'TERMINAL' tab, where the command `pipenv --py` has been executed, returning the path to the virtual environment's Python interpreter. A green arrow points to the terminal output.

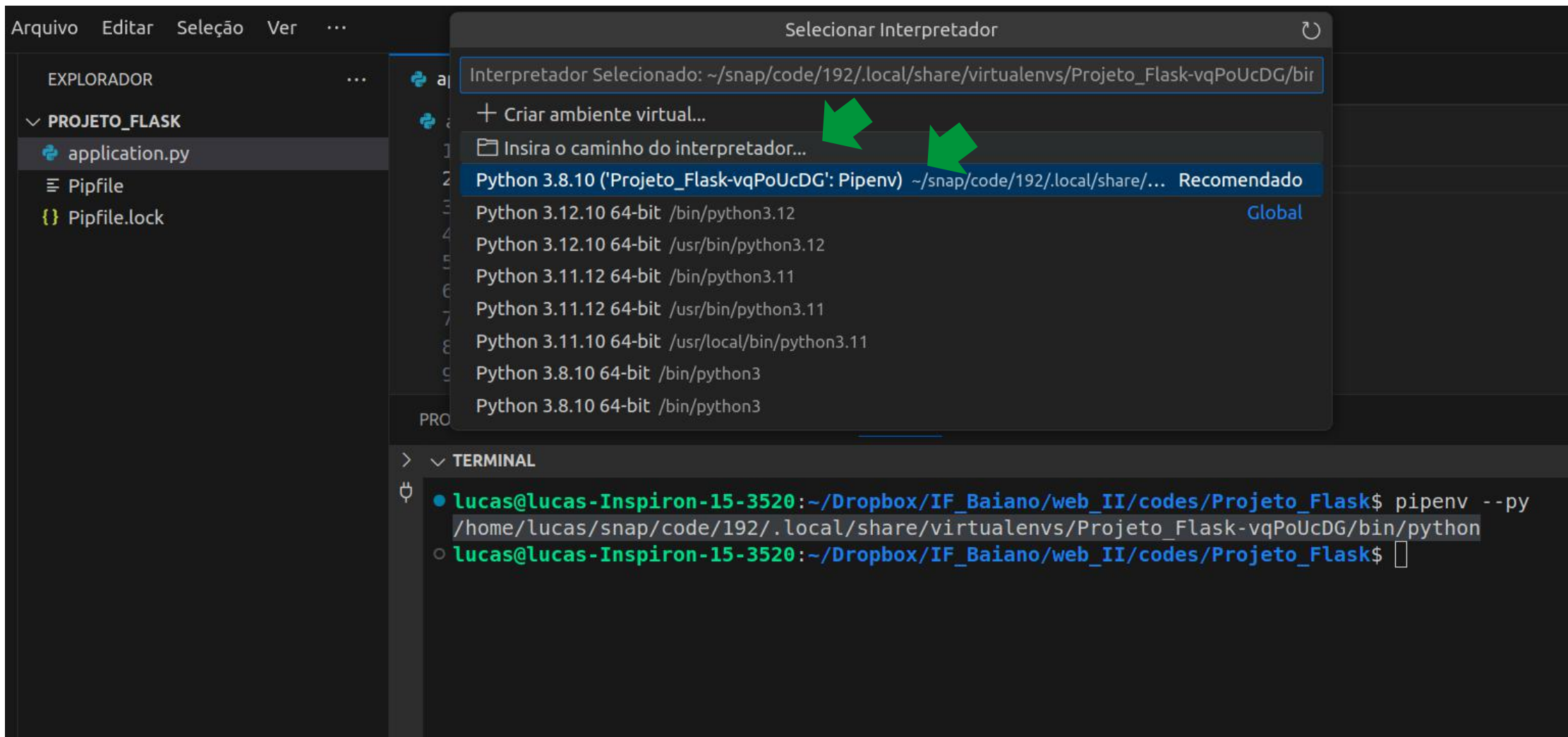
```
application.py 1 x
application.py > ...
1  from flask import Flask
2
3  app = Flask(__name__)
4
5
6  @app.route("/")
7  def hello_world():
8      return "<p>Hello, World!</p>"
9

PROBLEMAS 1  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL
>  ✓ TERMINAL
• lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask$ pipenv --py
  /home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/bin/python
○ lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask$
```

Primeiros passos com Flask



Primeiros passos com Flask



Arquivo Editar Seleção Ver ...

EXPLORADOR ...

PROJETO_FLASK

application.py

Pipfile

Pipfile.lock

Selecionar Interpretador

Interpretador Selecionado: ~/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/bin/python

+ Criar ambiente virtual...

Insira o caminho do interpretador...

Python 3.8.10 ('Projeto_Flask-vqPoUcDG': Pipenv) ~/snap/code/192/.local/share/... Recomendado

Python 3.12.10 64-bit /bin/python3.12 Global

Python 3.12.10 64-bit /usr/bin/python3.12

Python 3.11.12 64-bit /bin/python3.11

Python 3.11.12 64-bit /usr/bin/python3.11

Python 3.11.10 64-bit /usr/local/bin/python3.11

Python 3.8.10 64-bit /bin/python3

Python 3.8.10 64-bit /bin/python3

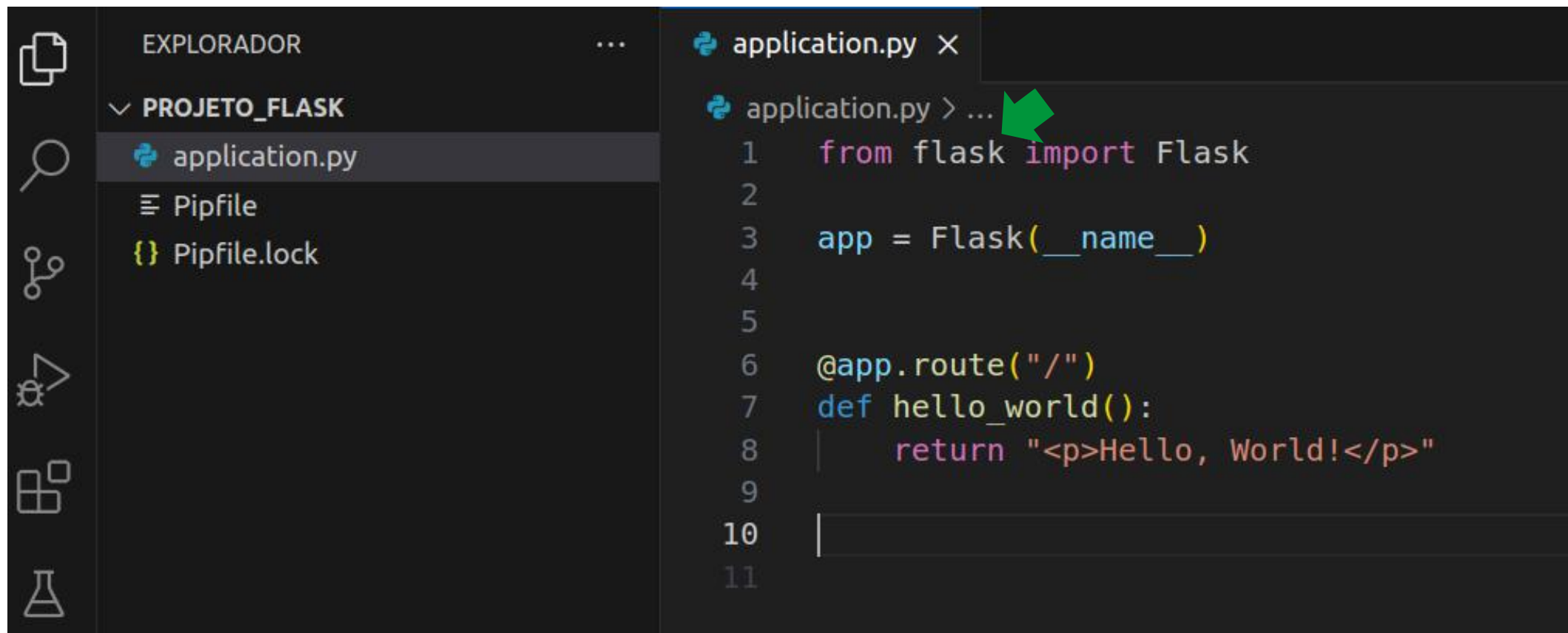
PROJETO_FLASK

TERMINAL

```
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask$ pipenv --py /home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/bin/python
```

```
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask$
```

Primeiros passos com Flask

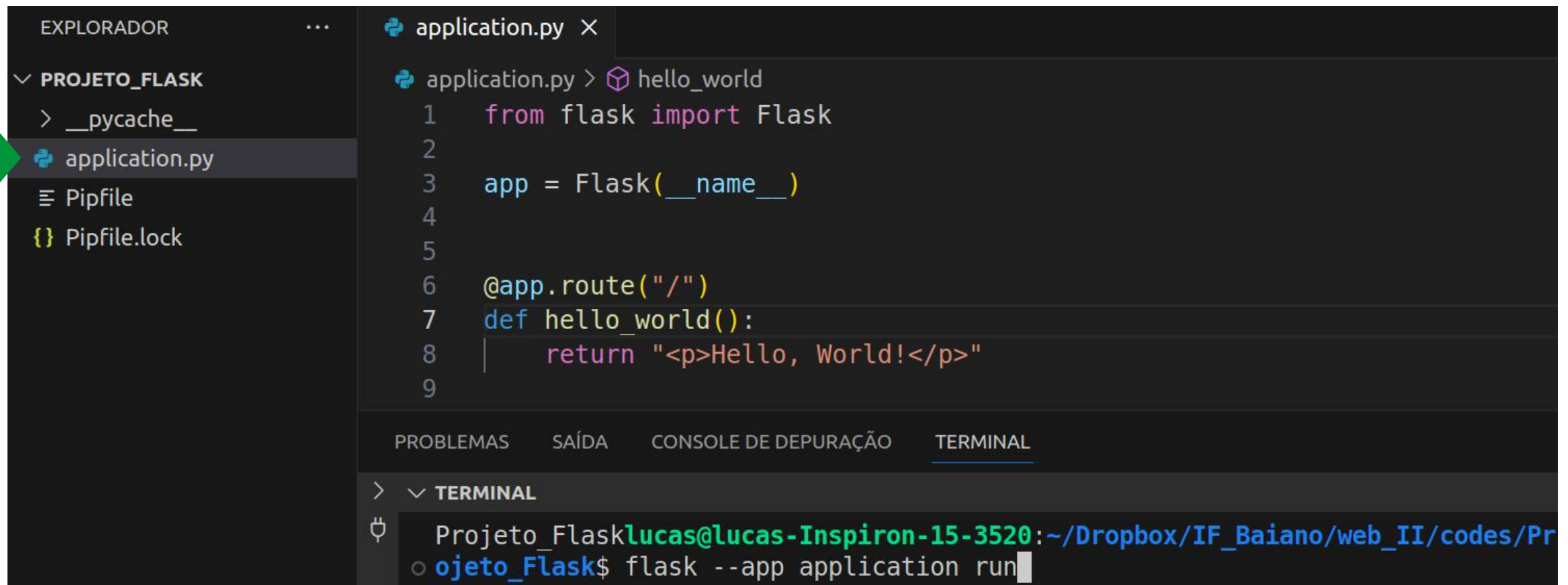


The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORADOR' (Explorer) pane shows a project named 'PROJETO_FLASK' containing 'application.py', 'Pipfile', and 'Pipfile.lock'. The 'application.py' file is selected. The main editor area shows the code for 'application.py' with line numbers 1 through 11. The code imports Flask, creates an app, and defines a route for the root path that returns 'Hello, World!'. A green arrow points to the 'import' keyword in the first line of code.

```
application.py x
application.py > ...
1  from flask import Flask
2
3  app = Flask(__name__)
4
5
6  @app.route("/")
7  def hello_world():
8      return "<p>Hello, World!</p>"
9
10 |
11
```

Primeiros passos com Flask

- Executando a aplicação: `flask --app application_name run`




The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORADOR' (Explorer) sidebar shows a project named 'PROJETO_FLASK' with files: '__pycache__', 'application.py' (highlighted with a green arrow), 'Pipfile', and 'Pipfile.lock'. The main editor area shows the code for 'application.py' with a 'hello_world' route. The code is as follows:

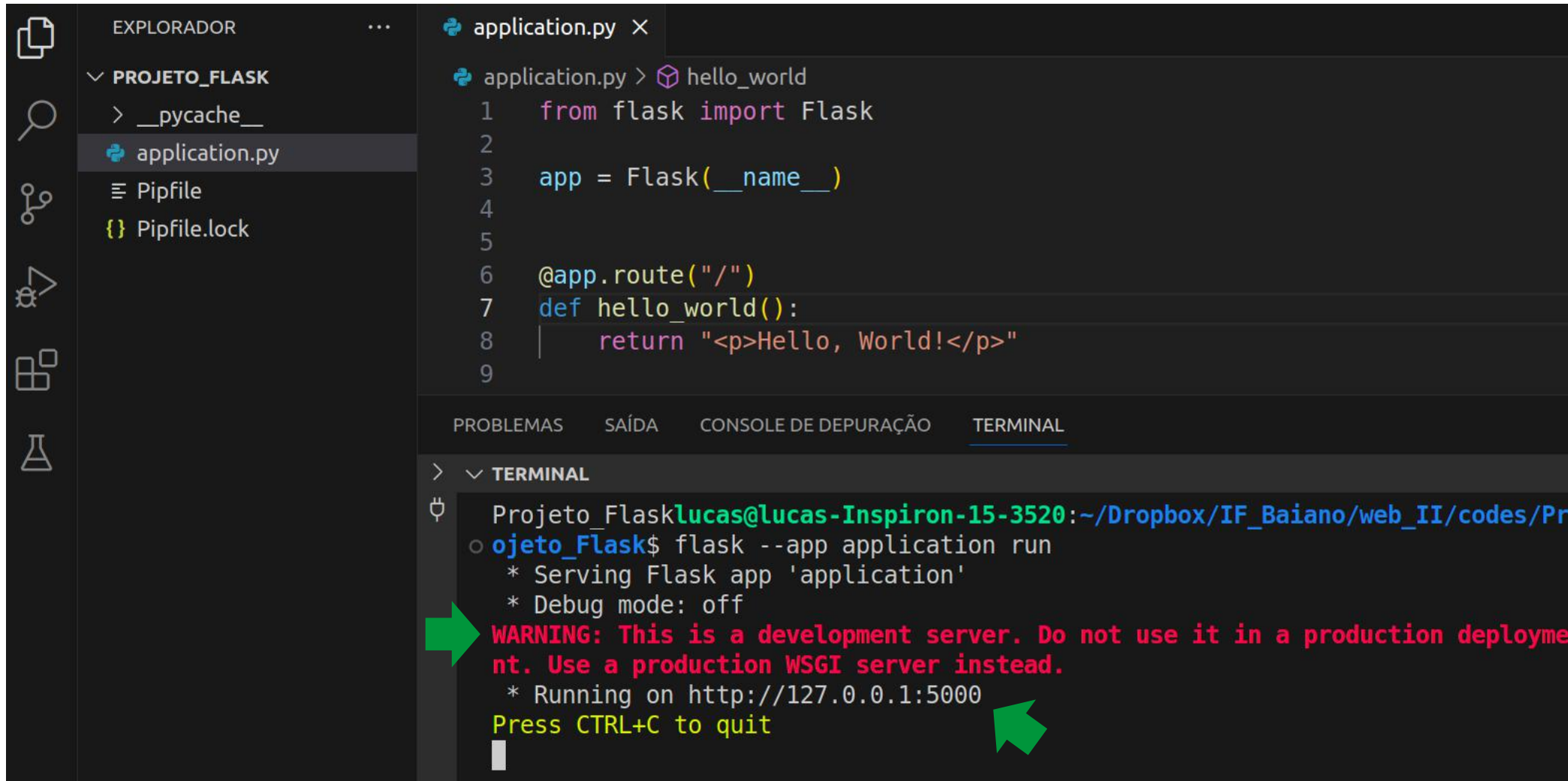
```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5
6 @app.route("/")
7 def hello_world():
8     return "<p>Hello, World!</p>"
9
```

Below the code editor, the 'TERMINAL' tab is active, showing the command to run the application:

```
Projeto_Flasklucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Pr
ojetto_Flask$ flask --app application run
```



Primeiros passos com Flask



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'PROJETO_FLASK' with files 'application.py', 'Pipfile', and 'Pipfile.lock'. The code editor displays the 'application.py' file with the following Python code:

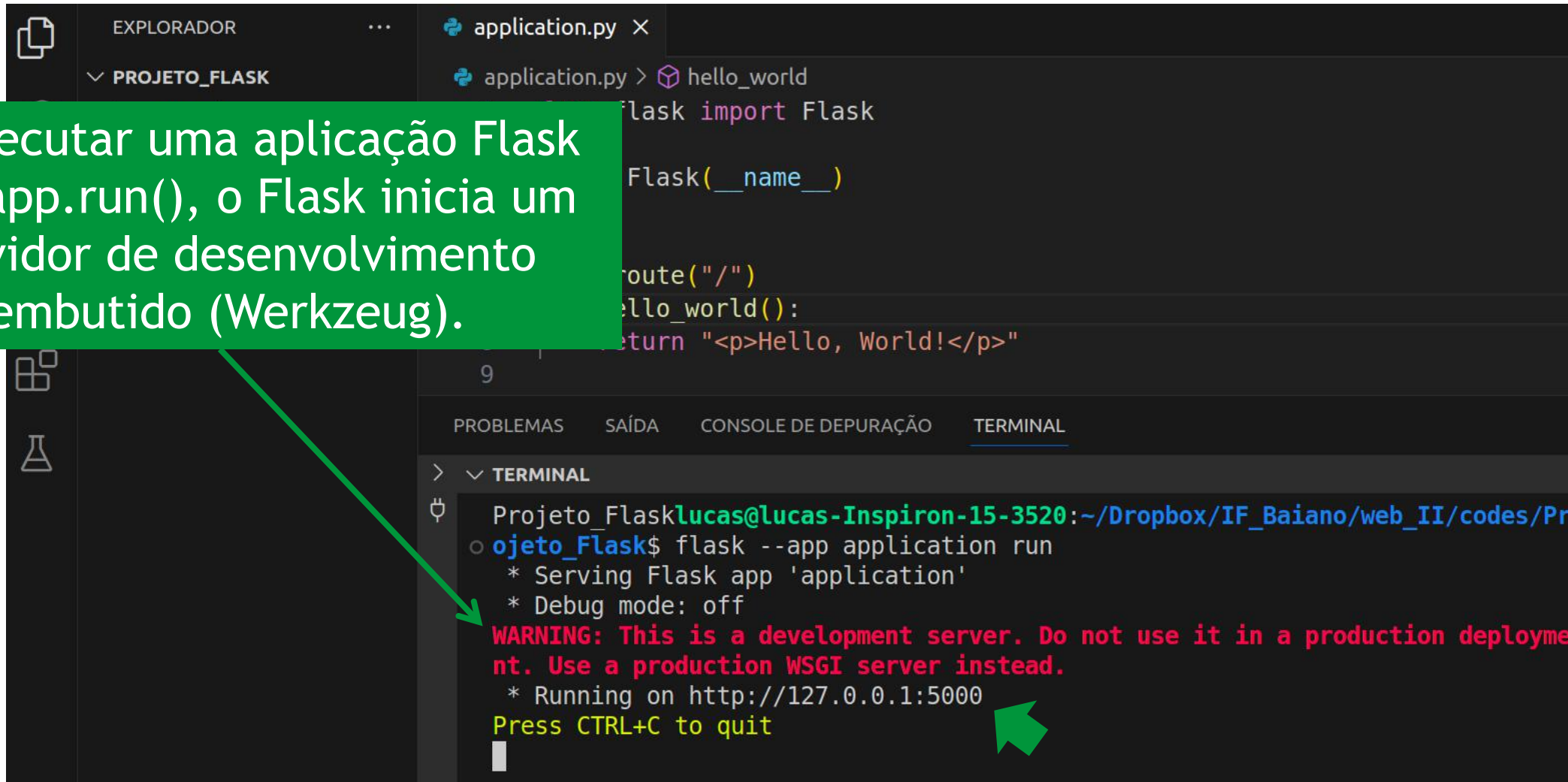
```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5
6 @app.route("/")
7 def hello_world():
8     return "<p>Hello, World!</p>"
9
```

The terminal at the bottom shows the command 'flask --app application run' being executed. The output indicates that the Flask app is running on http://127.0.0.1:5000. A red warning message is displayed: 'WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.' Two green arrows point to the warning message and the 'Press CTRL+C to quit' prompt.

```
> PROJETO_FLASK$ flask --app application run
* Serving Flask app 'application'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Primeiros passos com Flask

Ao executar uma aplicação Flask com `app.run()`, o Flask inicia um servidor de desenvolvimento embutido (Werkzeug).



The screenshot shows a code editor with a file named `application.py` open. The code in the editor is as follows:

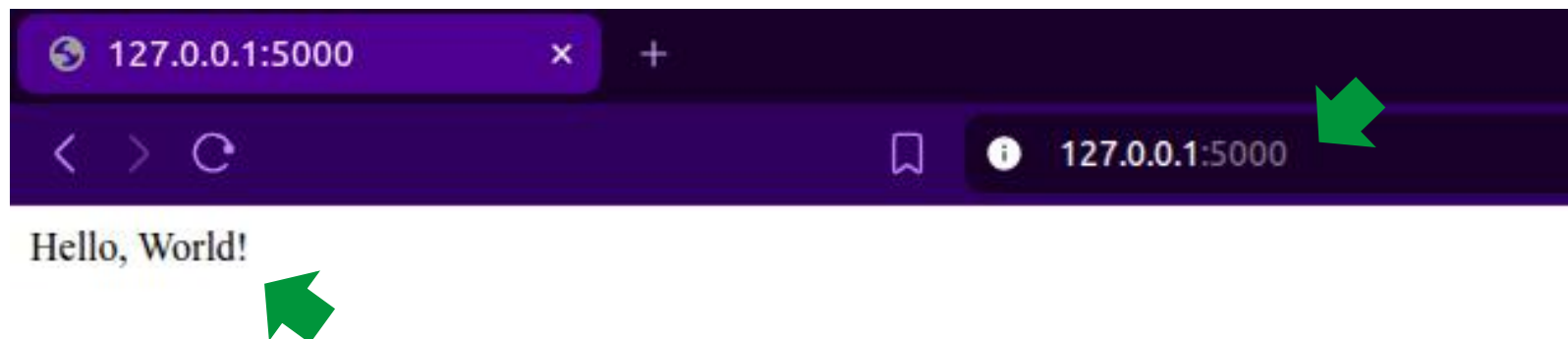
```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"
```

Below the code editor, the terminal window is visible. It shows the command `flask --app application run` being executed. The output indicates that the Flask app is running on `http://127.0.0.1:5000`. A green arrow points from the text box on the left to the terminal output, and another green arrow points to the URL `http://127.0.0.1:5000`.

```
Projeto_Flasklucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Pr
ojetos/Projeto_Flask$ flask --app application run
* Serving Flask app 'application'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployme
nt. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

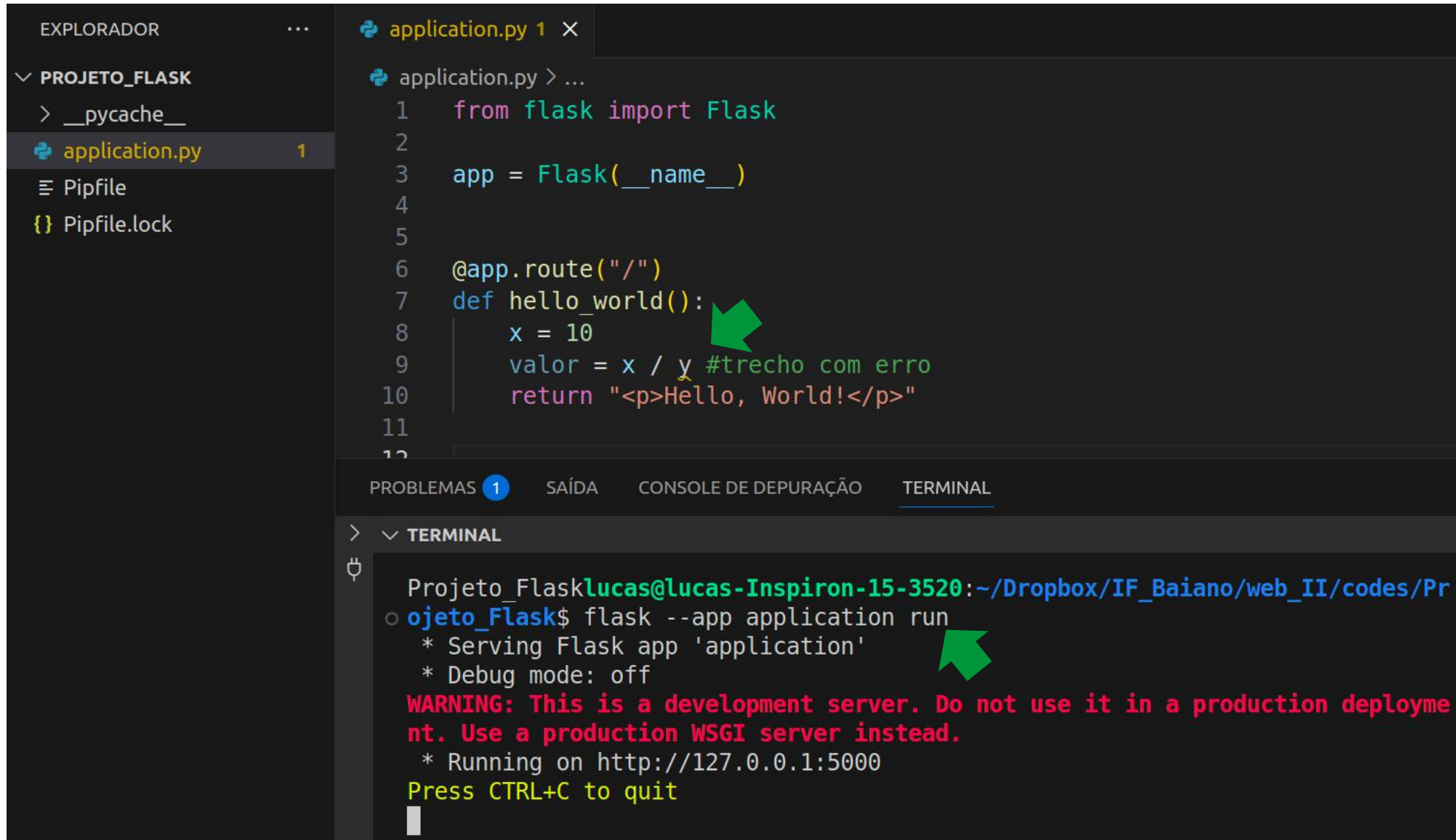

Primeiros passos com Flask



Primeiros passos com Flask

- O modo debug do Flask é uma configuração especialmente útil durante o desenvolvimento da aplicação. Ele facilita a identificação e correção de erros, oferecendo recursos que agilizam o processo de teste e iteração.
- Principais funcionalidades:
 - Recarregamento automático: a aplicação é reiniciada automaticamente sempre que um arquivo é modificado.
 - Mensagens de erro detalhadas: em caso de falha, o Flask exibe uma tela com a descrição do erro, pilha de chamadas (stack trace) e um console interativo para depuração.

Primeiros passos com Flask



The image shows a VS Code editor interface. On the left, the Explorer pane shows a project named 'PROJETO_FLASK' with files '__pycache__', 'application.py', 'Pipfile', and 'Pipfile.lock'. The main editor displays 'application.py' with the following code:

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5
6 @app.route("/")
7 def hello_world():
8     x = 10
9     valor = x / y #trecho com erro
10    return "<p>Hello, World!</p>"
11
12
```

A green arrow points to the variable 'y' on line 9, indicating a NameError. Below the editor, the TERMINAL pane shows the command 'flask --app application run' being executed. The output shows the server starting on http://127.0.0.1:5000. A green arrow points to the 'WARNING' message in the terminal output.

PROBLEMAS 1 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

Projetos_Flasklucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Pr
ojetos_Flask\$ flask --app application run
* Serving Flask app 'application'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit

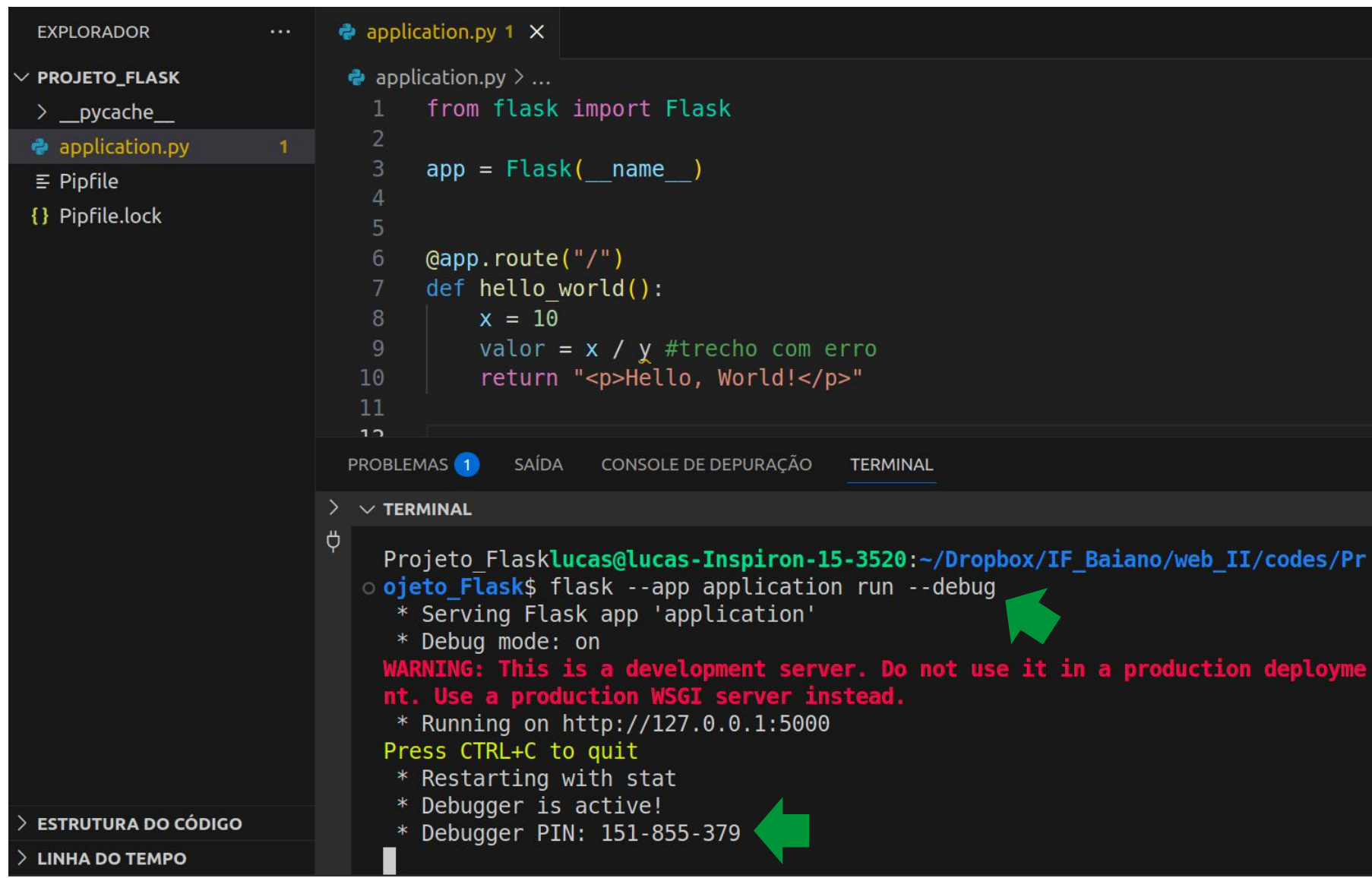
Primeiros passos com Flask



Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

Primeiros passos com Flask



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'PROJETO_FLASK' with files like '__pycache__', 'application.py', 'Pipfile', and 'Pipfile.lock'. The main editor displays the content of 'application.py', which is a simple Flask application with a route '/' that returns 'Hello, World!'. The terminal at the bottom shows the command 'flask --app application run --debug' being executed, and the output indicates that the server is running on http://127.0.0.1:5000 and that debug mode is on. A warning message is also displayed: 'WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.' Two green arrows point to the terminal output: one points to the command and the other points to the warning message.

```
EXPLORADOR ... application.py 1 X
PROJETO_FLASK
  > __pycache__
  application.py 1
  Pipfile
  Pipfile.lock

1 from flask import Flask
2
3 app = Flask(__name__)
4
5
6 @app.route("/")
7 def hello_world():
8     x = 10
9     valor = x / y #trecho com erro
10    return "<p>Hello, World!</p>"
11
12

PROBLEMAS 1 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL
> ▼ TERMINAL
Projetos_Flasklucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Pr
ojetos_Flask$ flask --app application run --debug
* Serving Flask app 'application'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 151-855-379
```


Primeiros passos com Flask



NameError

NameError: name 'y' is not defined

Traceback (most recent call last)

```
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1498, in __call__
    return self.wsgi_app(environ, start_response)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1476, in wsgi_app
    response = self.handle_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1473, in wsgi_app
    response = self.full_dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 882, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 880, in full_dispatch_request
    rv = self.dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 865, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args) # type: ignore[no-any-return]
File "/home/lucas/Dropbox/IF_Baiano/web_III/codes/Projeto_Flask/application.py", line 9, in hello_world
    valor = x / y #trecho com erro
```

NameError: name 'y' is not defined

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

Primeiros passos com Flask

NameError

NameError: name 'y' is not defined

Traceback (most recent call last)

```
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1498, in __call__
    return self.wsgi_app(environ, start_response)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1476, in wsgi_app
    response = self.handle_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1473, in wsgi_app
    response = self.full_dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 882, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 880, in full_dispatch_request
    rv = self.dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 865, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args) # type: ignore[no-any-return]
File "/home/lucas/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask/application.py", line 9, in hello_world
    valor = x / y #trecho com erro
```

NameError: name 'y' is not defined

Open an interactive python shell in this frame

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

Primeiros passos com Flask



NameError

NameError: name 'y' is not defined

Traceback (most recent call last)

```
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1495, in __call__
    return self.wsgi_app(environ, start_response)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1495, in __call__
    response = self.handle_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1495, in __call__
    response = self.full_dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1495, in __call__
    rv = self.handle_user_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1495, in __call__
    rv = self.dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 865, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args) # type: ignore[no-any-return]
File "/home/lucas/Dropbox/IF_Baiano/web_III/codes/Projeto_Flask/application.py", line 9, in hello_world
    valor = x / y #trecho com erro

[console ready]
>>>
```

NameError: name 'y' is not defined

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

Primeiros passos com Flask

NameError

NameError: name 'y' is not defined

Traceback (most recent call last)

```
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1498, in __call__
    return self.wsgi_app(environ, start_response)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1476, in wsgi_app
    response = self.handle_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 1473, in wsgi_app
    response = self.full_dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 882, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 880, in full_dispatch_request
    rv = self.dispatch_request()
File "/home/lucas/snap/code/192/.local/share/virtualenvs/Projeto_Flask-vqPoUcDG/lib/python3.8/site-packages/flask/app.py", line 865, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)  # type: ignore[no-any-return]
File "/home/lucas/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask/application.py", line 9, in hello_world
    valor = x / y #trecho com erro

[console ready]
>>> x
10
>>> |
```





NameError: name 'y' is not defined

Primeiros passos com Flask

- API significa Application Programming Interface (Interface de Programação de Aplicações).
- É um conjunto de regras que permite que diferentes softwares se comuniquem entre si.
- Na prática, uma API web permite que clientes (como navegadores, apps ou outros sistemas) enviem requisições e recebam respostas com dados.
- Exemplo:
 - GET /temperatura?cidade=Bonfim → retorna a temperatura atual em Bonfim.
 - POST /alerta → envia um alerta meteorológico.

Primeiros passos com Flask

- Roteamento é o processo de associar uma URL específica a uma função do servidor que trata aquela requisição.
- No Flask, o roteamento é feito com o decorador `@app.route()`.



```
@app.route("/hello")
def hello():
    return "Olá, mundo!"
```

Primeiros passos com Flask

- Roteamento é o processo de associar uma URL específica a uma função do servidor que trata aquela requisição.
- No Flask, o roteamento é feito com o decorador `@app.route()`.

```
@app.route("/hello")  
def hello():  
    return "Olá, mundo!"
```

O Flask mapeia a URL para a função decorada, não importa o nome da função.

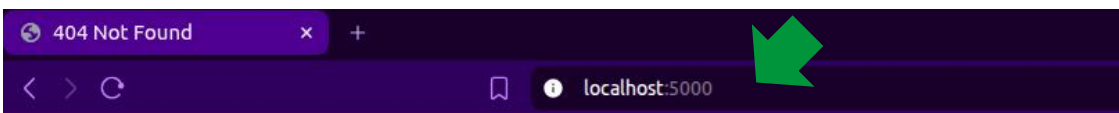
Primeiros passos com Flask

```
application.py > ...  
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  
6  @app.route("/")  
7  def hello_world():  
8      |   return "<p>Hello, World!</p>"  
9
```

Ausência de rota

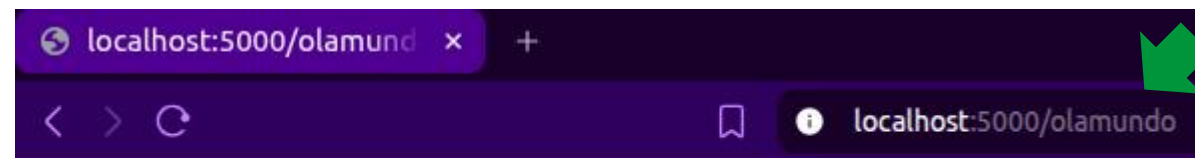
Primeiros passos com Flask

```
application.py > ...  
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  
6  @app.route("/olamundo")  
7  def hello_world():  
8      return "<p>Hello, World!</p>"  
9
```



Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.



Hello, World!

Primeiros passos com Flask

- Aplicações web modernas utilizam URLs significativas e amigáveis para facilitar a navegação e melhorar a experiência do usuário.
- Quando uma URL é clara e fácil de lembrar, os usuários têm mais chances de revisita-la e compartilhá-la. Isso contribui para o aumento do engajamento e da usabilidade da aplicação.
- Exemplo prático:
 - <https://meusite.com/page?id=42&tipo=x1>
 - <https://meusite.com/artigos/python-basico>

Primeiros passos com Flask

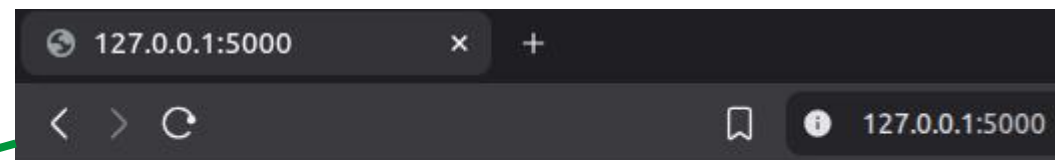
- Path parameters (ou parâmetros de caminho) são valores dinâmicos que fazem parte do caminho de uma URL em uma rota de um serviço web.
- Eles permitem que a URL seja mais flexível e dinâmica, recebendo dados diretamente nela, em vez de apenas nos parâmetros de consulta (query parameters).

Primeiros passos com Flask

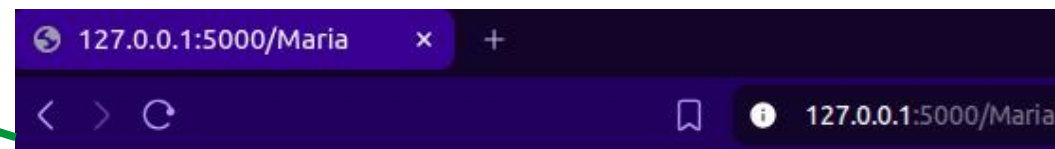
```
application.py > ...  
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  @app.route("/")  
6  def hello_world():  
7      return "<p>Hello, World!</p>"  
8  
9  @app.route("/<user>")  
10 def hello_user(user):  
11     return f"<p>Hello, {user}</p>"  
12
```

Primeiros passos com Flask

```
application.py > ...  
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  @app.route("/")  
6  def hello_world():  
7      return "<p>Hello, World!</p>"  
8  
9  @app.route("/<user>")  
10 def hello_user(user):  
11     return f"<p>Hello, {user}</p>"  
12
```



Hello, World!



Hello, Maria

Primeiros passos com Flask

```
from flask import Flask
from math import sqrt

app = Flask(__name__)

@app.route("/calc/<int:number>")
def square_root(number):
    if number < 0:
        return "Não é possível calcular a raiz quadrada de número negativo"
    else:
        result = sqrt(number)
        return f"A raiz quadrada de {number} é {result}"
```

Primeiros passos com Flask



A raiz quadrada de 5 é 2.24

Primeiros passos com Flask

- Conversão de tipos:

string	(padrão) aceita qualquer texto sem uma barra (/)
int	aceita inteiros positivos
float	aceita valores decimais (números de ponto flutuante) positivos
path	como string, mas também aceita barras (/)
uuid	aceita strings no formato UUID (Identificador Único Universal)

Primeiros passos com Flask

- URLs Únicas e comportamento de redirecionamento:

```
@app.route('/projects/')  
def projects():  
    return 'The project page'  
  
@app.route('/about')  
def about():  
    return 'The about page'
```

Ao acessar a URL sem a barra final (/projects), o Flask redireciona automaticamente para a URL com a barra final (/projects/).

Acessar a URL com a barra final (/about/) resulta em um erro 404 "Página não encontrada".

Exercícios

- Implemente uma aplicação Flask com endpoints que realizem operações básicas entre números reais.
 - Adição: `/calc/add/<float:a>/<float:b>`
 - Subtração:
 - Com dois números: `/calc/sub/<float:a>/<float:b>`
 - Com um número: `/calc/sub/<float:a>` → considera: $0 - a$
 - Multiplicação: `/calc/mul/<float:a>/<float:b>`
 - Divisão: `/calc/div/<float:a>/<float:b>`
 - Deve retornar erro se $b = 0$

Exercícios

Rota	Resposta
/calc/add/1.0/2.0	3.0
/calc/sub/5.0/2.0	3.0
/calc/sub/2.0	-2.0
/calc/mul/2.0/3.5	7.0
/calc/div/10.0/2.0	5.0
/calc/div/4.0/0.0	Erro: divisão por zero.

Exercícios

- Crie uma aplicação Flask com um endpoint que simula uma pequena agenda de contatos.
- A aplicação deve conter uma rota `/contato/<nome>` que receba um nome como parâmetro da URL e:
 - Verifique se o nome existe em um dicionário de contatos.
 - Se existir, retorne o telefone da pessoa.
 - Se não existir, retorne a mensagem: "Contato não encontrado."
 - O retorno pode ser texto simples.

Dúvidas



PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática
Lucas Sampaio Leite

