

PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática
Lucas Sampaio Leite



Deploy

- Deploy de um sistema web é o processo de colocar uma aplicação pronta para funcionar em um servidor, tornando-a acessível para os usuários pela internet.
- É a etapa em que o sistema sai do ambiente de desenvolvimento e passa a rodar em um ambiente de produção.



Deploy

- Etapas envolvidas em um deploy:
 - Subir os arquivos para um servidor
 - Configurar o servidor
 - Configurar o banco de dados
 - Definir ambiente de produção
 - Executar a aplicação

Deploy

- Onde o deploy pode ser feito:
 - Servidores próprios (VPS: DigitalOcean, Linode, AWS EC2).
 - Hospedagens compartilhadas.
 - Plataformas de deploy automático, exemplo:
 - Heroku
 - Railway
 - Render
 - Vercel (mais comum para front-end)
 - Netlify
 - AWS / Azure / GCP

Plataforma Render

- O Render é uma plataforma moderna de hospedagem em nuvem (cloud hosting) que permite fazer deploy de aplicações web, APIs, bancos de dados e serviços estáticos de forma simples.
- O Render aceita aplicações desenvolvidas com Flask, permitindo fazer deploy de APIs e sistemas web em poucos passos.



Deploy

Debug your Render services in Claude Code and Cursor.

[Try Render MCP >](#)

 Render

[Product](#)

[Pricing](#)

[Customers](#)

[Blog](#)

[Docs](#)

[Changelog](#)

[Company](#)

[Contact](#)

[Sign In](#)

[Get Started](#)

Your fastest path to production

Build, deploy, and scale your apps with
unparalleled ease – from your first user to
your billionth



[Get Started for Free >](#)

[Contact Sales >](#)



Deploy

 Lucas's workspace 

Projects

Search  

[+ New](#) [Upgrade](#)  

1 Choose service >

2 Configure >

3 Deploy

[Which to use?](#)

**Static Sites**

Static content served over a global CDN. Ideal for frontend, blogs, and content sites.

[New Static Site →](#)

**Web Services**

Dynamic web app. Ideal for full-stack apps, API servers, and mobile backends.

[New Web Service →](#)

**Private Services**

Web app hosted on a private network, accessible only from your other Render services.

[New Private Service →](#)

**Background Workers**

Long-lived services that process async tasks, usually from a job queue.

[New Worker →](#)

**Cron Jobs**

Short-lived tasks that run on a periodic schedule.

[New Cron Job →](#)

**Postgres**

Relational data storage. Supports point-in-time recovery, read replicas, and high availability.

[New Postgres →](#)

**Key Value**

Managed Redis®-compatible storage. Ideal for use as a shared cache, message broker, or job queue.

[New Key Value Instance →](#)

Deploy

  Lucas's workspace 

 Projects

Search   New  Upgrade  

Create a new Service Skip

1 Choose service >

2 Configure >

3 Deploy

Which to use?

**Static Sites**

Static content served over a global CDN. Ideal for frontend, blogs, and content sites.

[New Static Site →](#)

**Web Services**

Dynamic web app. Ideal for full-stack apps, API servers, and mobile backends.

[New Web Service →](#)

**Private Services**

Web app hosted on a private network, accessible only from your other Render services.

[New Private Service →](#)

**Background Workers**

Long-lived services that process async tasks, usually from a job queue.

[New Worker →](#)

**Cron Jobs**

Short-lived tasks that run on a periodic schedule.

[New Cron Job →](#)

**Postgres**

Relational data storage. Supports point-in-time recovery, read replicas, and high availability.

[New Postgres →](#) 

**Key Value**

Managed Redis®-compatible storage. Ideal for use as a shared cache, message broker, or job queue.

[New Key Value Instance →](#)

Deploy



Lucas's workspace

New Postgres

Search

+ New

Upgrade



Configure and deploy your new Database



Choose service



Configure



Deploy

Need help? Docs

Name

A unique name for your Postgres instance.

flask-blog-pg

Database Optional

The Postgres `dbname`

blogdb

User Optional

blogadmin

Region

Your services in the same [region](#) can communicate over a [private network](#).

Oregon (US West)

PostgreSQL Version

18

Datadog API Key Optional

The API key to use for sending metrics to Datadog. Setting this will enable Datadog monitoring.

Datadog Region Optional

The region key to use for sending metrics to Datadog.

US1 (default)

Deploy

Plan Options

Instance Type

Set your database's RAM and CPU. You can change your instance type later.



Free

For testing out Render Postgres

Free

\$0 / month

256 MB (RAM)

0.1 CPU

1 GB (Storage)



Basic

Reliability and performance for hobby projects. Starting at \$6 / month plus storage.



Pro

Perfect for production use cases at any scale. Starting at \$55 / month plus storage.



Accelerated

Memory-optimized plans offer significant performance improvements. Starting at \$160 / month plus storage.

Need a [custom instance type](#)? We support up to 1024 GB RAM, 128 CPUs, and 16 TB storage.

Deploy

Storage

Your database's capacity, in GB. You can increase storage later, but you can't decrease it.

Specify 1 GB or any multiple of 5 GB.

1 GB \$0 / month

Storage Autoscaling

When your database is 90% full, automatically increase its storage by 50% (rounded up to the nearest 5 GB).

Max one increase per 12 hours. [Learn more](#)

☐ Disabled \$0 / month

High Availability

Run a standby instance of your database and automatically fail over to it if the primary encounters an issue. [Learn more](#)

☐ Disabled

 Only available for **Pro** instances and higher.

Monthly Total

Database instances are billed and prorated by the second.

Instance: **Free** \$0 / month

Total \$0 / month



Create Database

Deploy

Lucas's workspace

Environment

flask-blog-pg

Info

MONITOR

Logs

Metrics

MANAGE

Apps

Recovery

Changelog

Invite a friend

Contact support

Render Status

My project / Production / flask-blog-pg

Search

+ New

Upgrade

?

L

POSTGRES SQL

flask-blog-pg

Free

Upgrade your instance

View docs

Connect

Service ID: dpg-d4bo8bndiees73dc8cq0-a

Info

Your database will expire on December 14, 2025. The database will be deleted unless you upgrade to a paid instance type.

General

Name

flask-blog-pg

A unique name for your database.

Edit

Creating since

a few seconds ago

Status

Creating

PostgreSQL Version

18

Region

Oregon (US West)

Deploy

Lucas's workspace

Environment

flask-blog-pg

Info

MONITOR

Logs

Metrics

MANAGE

Apps

Recovery

Changelog

Invite a friend

Contact support

Render Status

My project / Production / flask-blog-pg

Search

+ New

Upgrade

Help

Profile

POSTGRESQL

flask-blog-pg

Free

Upgrade your instance

View docs

Connect

Service ID: dpg-d4bo8bndiees73dc8cq0-a

Info

Your database will expire on December 14, 2025. The database will be deleted unless you [upgrade to a paid instance type](#).

General

Name

flask-blog-pg

A unique name for your database.

Edit

Created

3 minutes ago

Status

✓ Available

PostgreSQL Version

18

Region

Oregon (US West)

Deploy

Connections

Hostname

An internal hostname used by your Render services.

dpg-d4csdpggjchc73dk046g-a



Port

5432



Database

blogdb_vudo



Username

blogadmin



Password

.....



Internal Database URL

.....



External Database URL

.....



PSQL Command

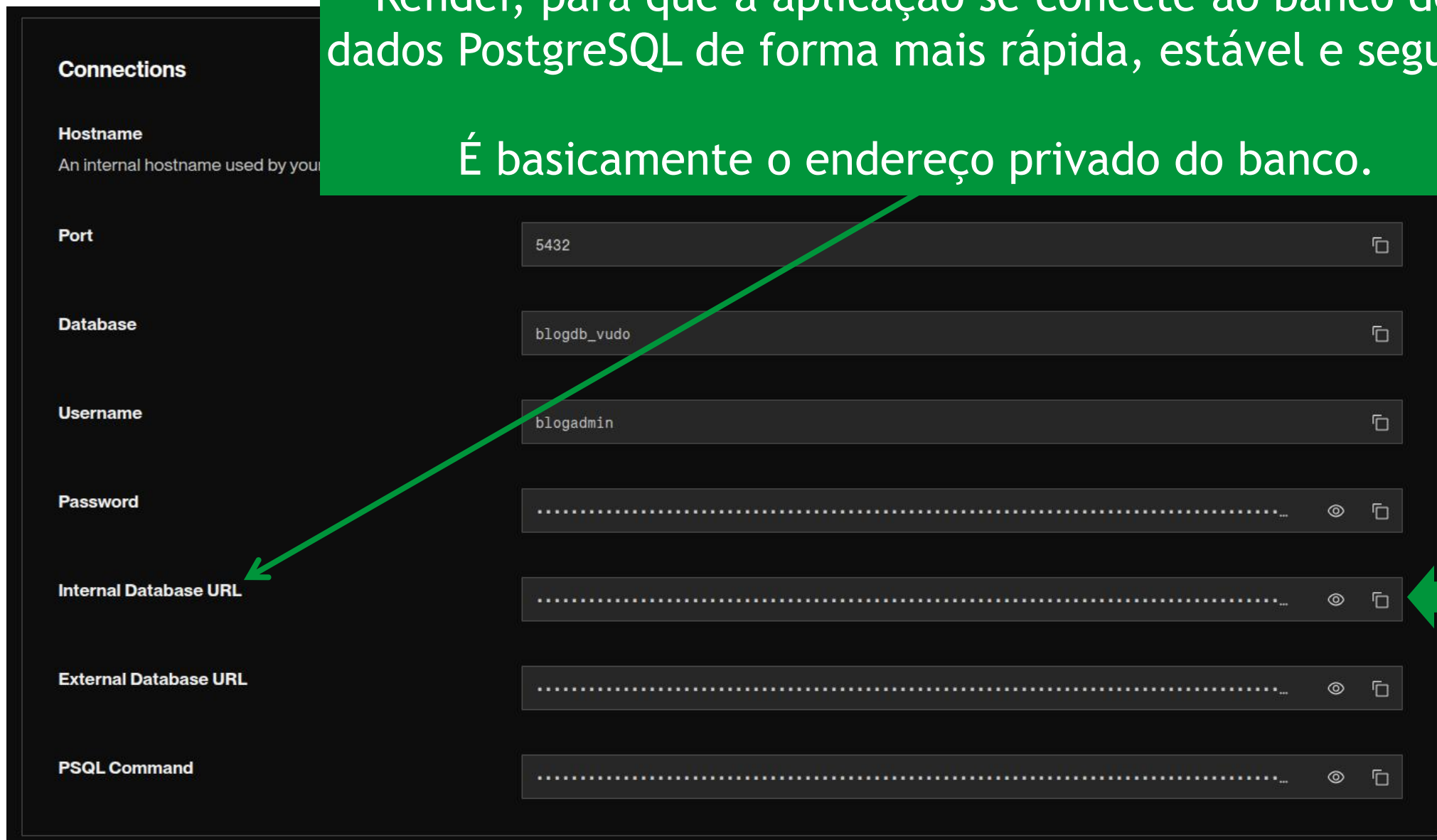
.....



Deploy

A Internal Database URL do Render é um endereço interno, usado somente dentro da infraestrutura do Render, para que a aplicação se conecte ao banco de dados PostgreSQL de forma mais rápida, estável e segura.

É basicamente o endereço privado do banco.



The screenshot shows the 'Connections' configuration page in the Render dashboard. It features a dark theme with a sidebar on the left containing the title 'Connections' and a description for 'Hostname'. The main area contains several configuration fields, each with a label, a text input, and a copy icon. The fields are: 'Port' (5432), 'Database' (blogdb_vudo), 'Username' (blogadmin), 'Password' (masked with dots), 'Internal Database URL' (masked with dots), 'External Database URL' (masked with dots), and 'PSQL Command' (masked with dots). A green arrow points from the text 'É basicamente o endereço privado do banco.' to the 'Internal Database URL' field. Another green arrow points to the right side of the 'Internal Database URL' field.

Connections	
Hostname	An internal hostname used by you
Port	5432
Database	blogdb_vudo
Username	blogadmin
Password	
Internal Database URL	
External Database URL	
PSQL Command	

Deploy



Search ^ K + New Upgrade ? L

Environment

flask-blog-pg

Info

MONITOR

Logs

Metrics

MANAGE

Apps

Recovery

My project / Production / flask-blog-pg

POSTGRESQL

flask-blog-pg

Free

Upgrade your instance →

Service ID: dpg-d4bo8bndiees73dc8cq0-a

Info

ⓘ Your database will expire on December 14, 2025. The database will be deleted unless you [upgrade to a paid instance type](#).

General

Name

A unique name for your database.

flask-blog-pg

Edit

Created

8 minutes ago

Static Site

Web Service

Private Service

Background Worker

Cron Job

Postgres

Key Value

Project

Blueprint

Deploy



Lucas's workspace

New Web Service

Search

+ New

Upgrade



New Web Service

Source Code

Git Provider

Public Git Repository

Existing Image

Connect Git provider

Connect your Git provider to deploy from your existing repositories.

GitHub

GitLab

Bitbucket

Name

A unique name for your web service.

example-service-name

Project Optional

Add this web service to a project once it's created.

Select a project...

Select an environment...

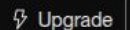
Language

Choose the runtime environment for this service.

Node

Deploy

  Lucas's workspace   New Web Service 

Search  ^ K    

New Web Service

Source Code

Git Provider

Public Git Repository

Existing Image

Connect Git provider

Connect your Git provider to deploy from your existing repositories.

 GitHub  GitLab  Bitbucket

Name

A unique name for your web service.

Project

Optional

Add this web service to a project once it's created.

Language

Choose the runtime environment for this service.

O Git Provider é o local onde o código está hospedado e de onde o Render vai buscar o repositório para fazer o deploy da aplicação.

Deploy

Render by **Render** would like permission to:

 **Verify your GitHub identity** (lucassampaioleite)

 **Know which resources you can access**

 **Act on your behalf**

[Learn more](#)

Resources on your account

 **Email addresses** (read)
View your email addresses

[Learn more about Render](#)

Cancel

Authorize Render

Authorizing will redirect to

<https://dashboard.render.com>

Search ^ K

+ New

Upgrade

?

L

Repository

Existing Image

Connect Git provider

Connect your Git provider to deploy from your existing repositories.

 GitHub

 GitLab

 Bitbucket

Select an environment...

Deploy

github.com/apps/render/installations/new/permissions:target_id=13689528



Install Render

Install on your personal account  lucassampaioleite

for these repositories:

☒ **All repositories** 

This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only).

☐ **Only select repositories**

Select at least one repository. Also includes public repositories (read-only).

with these permissions:

Search ^ K

+ New Upgrade ? L

Repository Existing Image

Connect Git provider

Connect your Git provider to deploy from your existing repositories.

 GitHub  GitLab  Bitbucket

Select an environment...

Deploy



Lucas's workspace

New Web Service

Search

+ New

Upgrade



New Web Service



Source Code

Git Provider

Public Git Repository

Existing Image

Search

Credentials (1)

lucassampaioleite /

lucassampaioleite /

lucassampaioleite /

lucassampaioleite /

lucassampaioleite /

lucassampaioleite /

lucassampaioleite /

Name

A unique name for your web service.

example-service-name

Project Optional

Add this web service to a project once it's created.

Select a project...

Select an environment...

Deploy

1

General

Owner *

 lucassampaioleite

 /

Repository name *

flask-blog

✓ flask-blog is available.

Great repository names are short and memorable. How about [fuzzy-garbanzo?](#)

Description

0 / 350 characters

2

Configuration

Choose visibility *

Choose who can see and commit to this repository

 Private

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

Off ☐

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

No license

Create repository

Deploy

lucassampaioleite / flask-blog



[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

**flask-blog** Private

[Watch 0](#) [Fork 0](#) [Star 0](#)



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

[Get started with GitHub Copilot](#)



Add collaborators to this repository

Search for people using their GitHub username or email address.

[Invite collaborators](#)

Quick setup — if you've done this kind of thing before

HTTPS SSH

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# flask-blog" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:lucassampaioleite/flask-blog.git
git push -u origin main
```

Deploy

 Lucas's workspace  New Web Service  Search  + New  Upgrade  ? 

New Web Service

Source Code

Git ProviderPublic Git RepositoryExisting Image

 Search

 Credentials (1) 

 lucassampaioleite / flask-blog 2m ago 

 lucassampaioleite /

 lucassampaioleite /

 lucassampaioleite /

 lucassampaioleite /

 lucassampaioleite /

 lucassampaioleite /

Name

A unique name for your web service.

Project Optional

Add this web service to a project once it's created

 Select a project... 

 /

 Select an environment... 

Deploy

  Lucas's workspace   New Web Service 

 Search  K  + New  Upgrade  ?  L

Source Code

lucassampaioleite / flask-blog • 2m ago  Edit

Name
A unique name for your web service.

flask-blog

Project Optional
Add this web service to a **project** once it's created.

 Select a project...  /  Select an environment... 

Language
Choose the **runtime environment** for this service.

Python 3 

Branch
The Git branch to build and deploy.

main 

Region
Your services in the same **region** can communicate over a **private network**. You currently have services running in **Oregon**.

 Oregon (US West) 1 existing service 

Root Directory Optional
If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a **monorepo**.

e.g. `src`

Deploy

Build Command

Render runs this command to build your app before each deploy.

```
$ pip install -r requirements.txt
```

Start Command

Render runs this command to start your app with each deploy.

```
$ gunicorn src.wsgi:app
```

Instance Type

For hobby projects

Free 512 MB (RAM)
\$0 / month 0.1 CPU

⚠ Upgrade to enable more features

Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.

For professional use

For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

Starter 512 MB (RAM)
\$7 / month 0.5 CPU

Standard 2 GB (RAM)
\$25 / month 1 CPU

Pro 4 GB (RAM)
\$85 / month 2 CPU

Pro Plus 8 GB (RAM)
\$175 / month 4 CPU

Pro Max 16 GB (RAM)
\$225 / month 4 CPU

Pro Ultra 32 GB (RAM)
\$450 / month 8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

Deploy

O gunicorn é um servidor web de produção usado para executar aplicações Python, como Flask ou Django, em ambientes reais de produção.

Build Command

Render runs this command to build your app before each deploy.

```
$ pip install -r requirements.txt
```

Start Command

Render runs this command to start your app with each deploy.

```
$ gunicorn src.wsgi:app
```

Instance Type

For hobby projects

Free
\$0 / month

For professional use

For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

Starter
\$7 / month

512 MB (RAM)
0.5 CPU

Standard
\$25 / month

2 GB (RAM)
1 CPU

Pro
\$85 / month

4 GB (RAM)
2 CPU

Pro Plus
\$175 / month

8 GB (RAM)
4 CPU

Pro Max
\$225 / month

16 GB (RAM)
4 CPU

Pro Ultra
\$450 / month

32 GB (RAM)
8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

O wsgi.py é necessário porque ele funciona como o ponto de entrada padrão para servidores web Python

Deploy

Build Command

Render runs this command to build your app before each deploy.

```
$ pip install -r requirements.txt
```

Start Command

Render runs this command to start your app with each deploy.

```
$ gunicorn src.wsgi:app
```

Instance Type

For hobby projects

Free
\$0 / month

For professional use

For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

Starter
\$7 / month

Pro
\$85 / month

Pro Max
\$225 / month

16 GB (RAM)
4 CPU

Pro Ultra
\$450 / month

32 GB (RAM)
8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

O build command (comando de build) é o comando que o Render executa para preparar o projeto antes de rodá-lo.

Ele é essencial para: instalar dependências, compilar arquivos, gerar artefatos necessários para a execução e preparar o ambiente da aplicação

Deploy

 Lucas's workspace 

New Web Service 

Q Search  K

+ New  Upgrade  

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

Pro \$85 / month	4 GB (RAM) 2 CPU	Pro Plus \$175 / month	8 GB (RAM) 4 CPU
Pro Max \$225 / month	16 GB (RAM) 4 CPU	Pro Ultra \$450 / month	32 GB (RAM) 8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

DATABASE_URL

.....



+ Add Environment Variable

 Add from .env

> **Advanced**Deploy Web Service 

Deploy



Lucas's workspace

← Dashboard

 flask-blog

 Events

 Settings

MONITOR

 Logs

 Metrics

MANAGE

 Environment

 Shell

 Scaling

 Previews

 Disks

 Jobs

 flask-blog

Search ^ K

+ New

 Upgrade





 WEB SERVICE

flask-blog

Python 3

Free

Upgrade your instance →

Service ID: `srv-d4bp2dogjchc73ctsh5g`

 lucassampaioleite / flask-blog `main`

`https://flask-blog-iokt.onrender.com`

Connect

Manual Deploy

 Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now](#)

 Your service has not been deployed because the GitHub repository is empty. Make a commit before retrying.

 Filter events 31

 No events from the past 90 days match the selected filters.

Deploy

...or create a new repository on the command line

```
echo "# flask-blog" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:lucassampaioleite/flask-blog.git
git push -u origin main
```



```
Initialized empty Git repository in /home/lucas/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog/.git/
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git add README.md
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git commit -m "first commit"
[master (root-commit) 770ec88] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git branch -M main
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git remote add origin git@github
b.com:lucassampaioleite/flask-blog.git
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:lucassampaioleite/flask-blog.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Deploy

 lucassampaioleite / flask-blog 

  |      

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

 **flask-blog** Private

 Watch 0   Fork 0   Star 0 

 main   1 Branch  0 Tags



[Add file](#) 

[Code](#) 

 **lucassampaioleite** first commit

770ec88 · 6 minutes ago  1 Commit

 README.md first commit 6 minutes ago

 **README**



flask-blog

About

No description, website, or topics provided.

-  Readme
-  Activity
-  0 stars
-  0 watching
-  0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Deploy

```
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    Pipfile
    Pipfile.lock
    __pycache__/
    application.py
    controllers/
    extensions/
    instance/
    models/
    views/

nothing added to commit but untracked files present (use "git add" to track)
○ lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```

Deploy

```
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

Pipfile

Pipfile.lock

__pycache__/
application.py

controllers/

extensions/

instance/

models/

views/

O __pycache__ é uma pasta criada automaticamente pelo Python para armazenar arquivos compilados do seu código, com extensão .pyc.

nothing added to commit but untracked files present (use "git add" to track)

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

Deploy

- O `.gitignore` é um arquivo especial usado pelo Git para dizer quais arquivos ou pastas devem ser ignorados — ou seja, não serão adicionados, versionados ou enviados para o repositório.
- `gitignore.io` é uma ferramenta online gratuita que cria automaticamente arquivos `.gitignore` completos e atualizados para qualquer linguagem, framework ou IDE.

Deploy

 | gitignore.io



gitignore.io

Crie arquivos .gitignore úteis para o seu projeto.

Criar

[Código-fonte](#) | [Documentação](#)

gitignore.io

Deploy



🔗 **gitignore.io**

Crie arquivos .gitignore úteis para o seu projeto.



Python X

Flask X

Criar



[Código-fonte](#) | [Documentação](#)

Deploy

```
# Created by https://www.toptal.com/developers/gitignore/api/python,flask
# Edit at https://www.toptal.com/developers/gitignore?templates=python,flask

### Flask ###
instance/*
!instance/.gitignore
.webassets-cache
.env

### Flask.Python Stack ###
# Byte-compiled / optimized / DLL files
__pycache__/*
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
```

Deploy



INSTITUTO
FEDERAL
Baiano

```
.gitignore
1  # Created by https://www.toptal.com/developers/gitignore/api/python,flask
2  # Edit at https://www.toptal.com/developers/gitignore?templates=python,flask
3
4  ### Flask ###
5  instance/*
6  !instance/.gitignore
7  .webassets-cache
8  .env
9
10 ### Flask.Python Stack ###
11 # Byte-compiled / optimized / DLL files
12 __pycache__/
13 *.py[co]
14 *.py.class
15
16 # C extensions
17 *.so
18
19 # Distribution / packaging
20 .Python
21 build/
22 develop-eggs/
23 dist/
24 downloads/
```

Deploy

```
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git status
```

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```



```
.gitignore  
Pipfile  
Pipfile.lock  
application.py  
controllers/  
extensions/  
models/  
views/
```

Deploy

- O psycopg2-binary é o driver que permite que sua aplicação Python (Flask) se conecte a um banco PostgreSQL, que é o banco padrão oferecido pelo Render.
 - Em outras palavras, é o "conector" que permite ao Flask conversar com o PostgreSQL.
- O gunicorn é o servidor web recomendado para rodar aplicações Flask em produção.

Deploy

- `pipenv install psycpg2-binary`

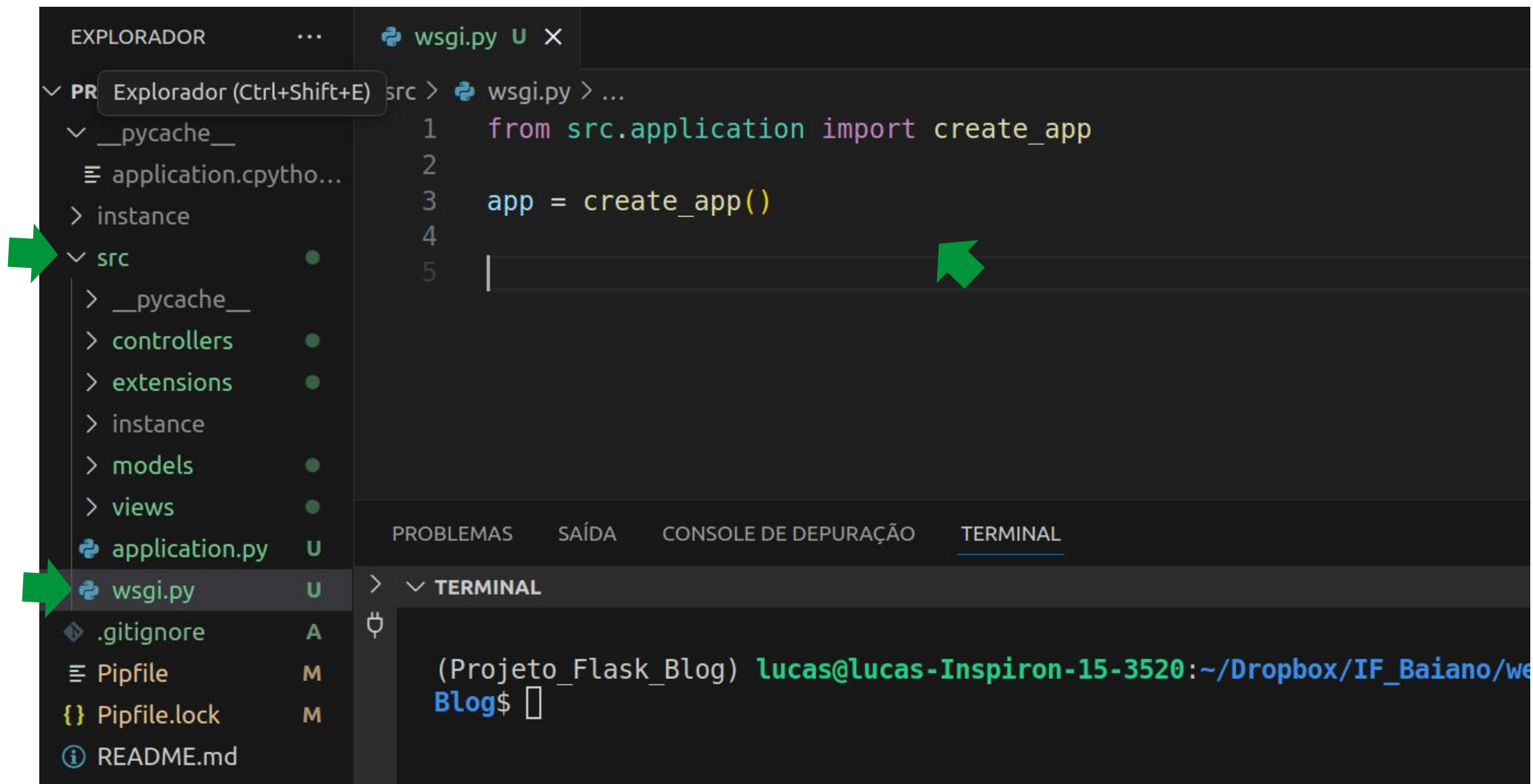
```
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ pipenv install p
sycpg2-binary
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing psycpg2-binary...
✓ Installation Succeeded
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (659eb1)...
All dependencies are now up-to-date!
Upgrading psycpg2-binary in dependencies.
Building requirements...
Resolving dependencies...
✓ Success!
```

Deploy

- pipenv install gunicorn

```
● lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ pipenv install gunicorn
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing gunicorn...
✓ 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 (3c5272)...
All dependencies are now up-to-date!
Upgrading gunicorn in dependencies.
Building requirements...
Resolving dependencies...
✓ Success!
```

Deploy

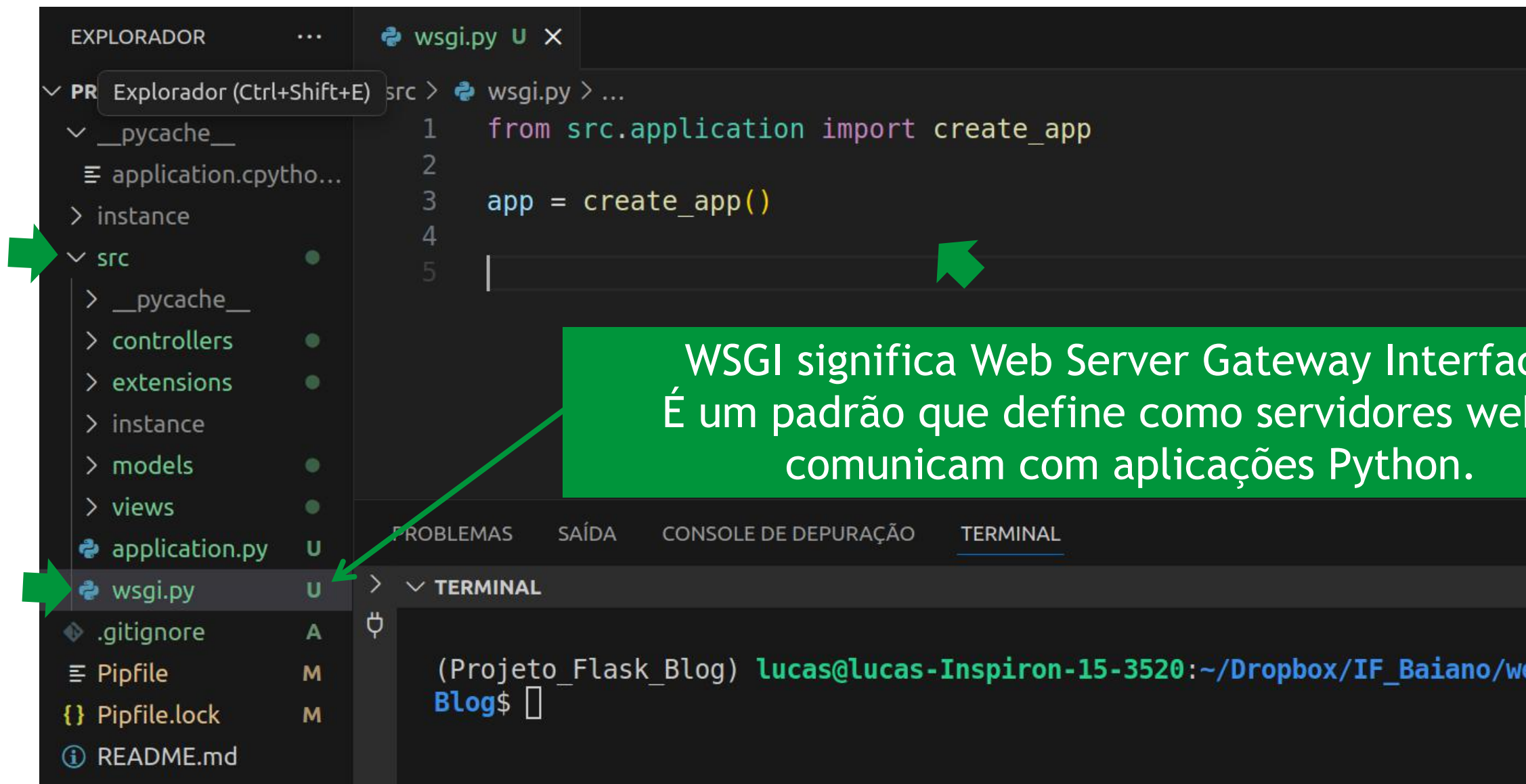


The screenshot displays the Visual Studio Code (VS Code) interface during the deployment of a Flask application. The Explorer panel on the left shows the project structure, with the `src` directory expanded. A green arrow points to the `src` directory, and another green arrow points to the `wsgi.py` file. The main editor shows the `wsgi.py` file with the following code:

```
1 from src.application import create_app
2
3 app = create_app()
4
5
```

The bottom panel shows the TERMINAL view. The terminal output indicates the current directory is `~/Dropbox/IF_Baiano/we` and the prompt is `Blog$`. A green arrow points to the terminal area.

Deploy



The image shows a Visual Studio Code editor interface. On the left, the Explorer sidebar shows a project structure with a `src` directory containing `application.py` and `wsgi.py`. A green arrow points to the `src` directory, and another points to `wsgi.py`. The main editor displays the content of `wsgi.py`:

```
1 from src.application import create_app
2
3 app = create_app()
4
5
```

A green box with white text explains WSGI:

WSGI significa Web Server Gateway Interface.
É um padrão que define como servidores web se comunicam com aplicações Python.

At the bottom, the Terminal panel shows the command prompt:

```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/we
Blog$
```

Deploy

```
from src.extensions import db
from src.extensions import jwt
```

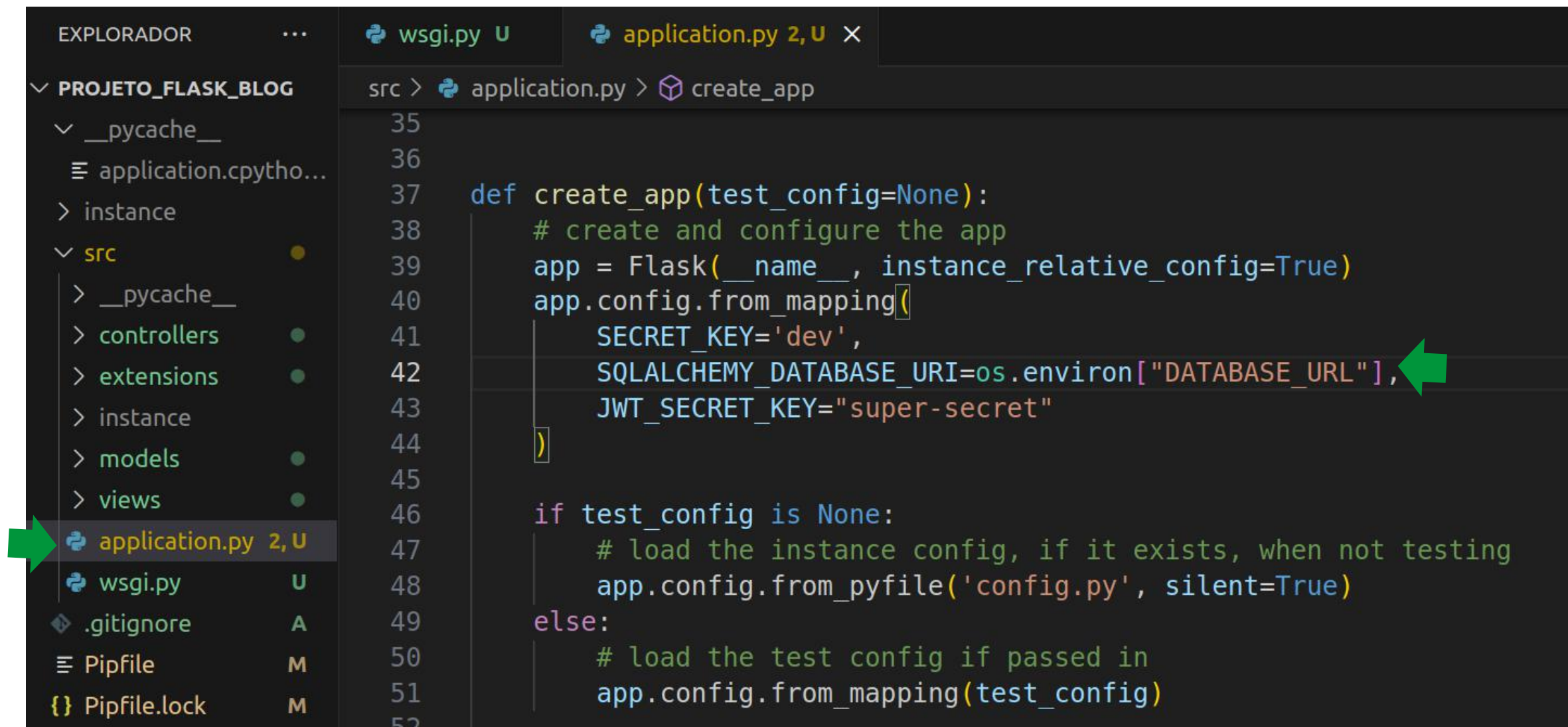
Corrigir erros de import:

- src.models
- src.controllers
- src.views
- src.extensions



```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ flask --app src.application run --debug
* Serving Flask app 'src.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: 110-182-683
```

Deploy



EXPLORADOR ...

PROJETO_FLASK_BLOG

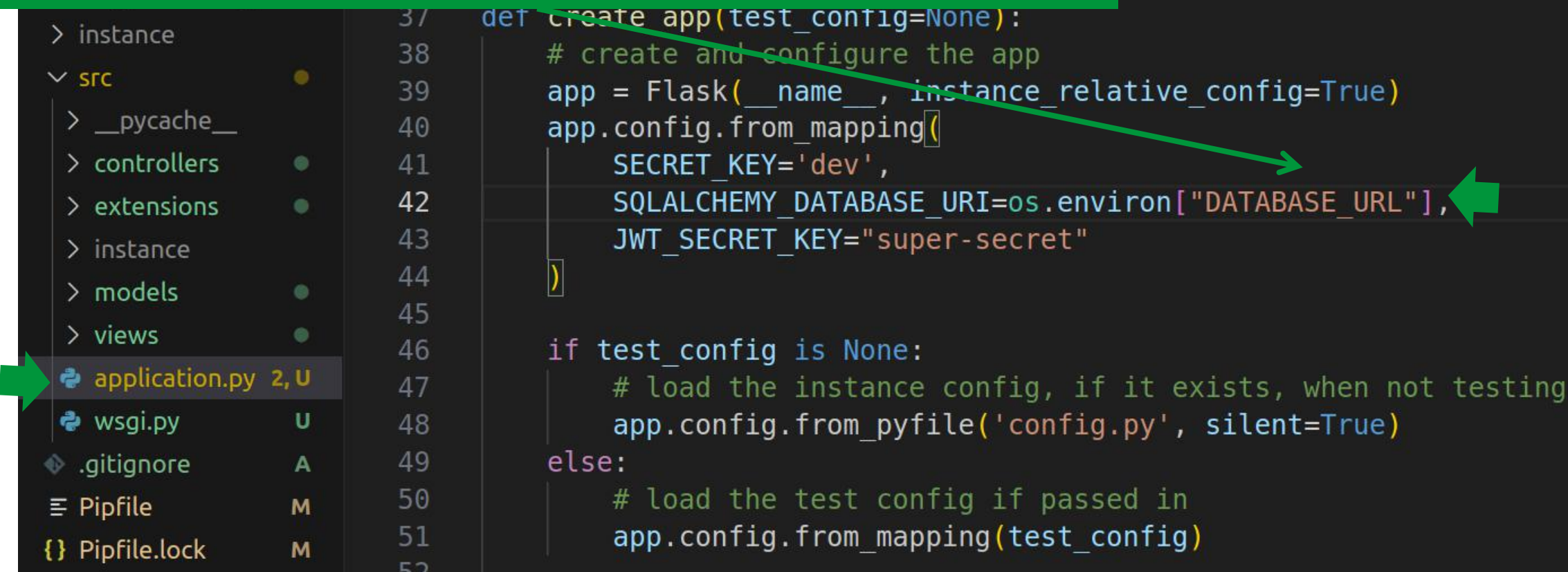
- __pycache__
- application.cpytho...
- instance
- src
 - __pycache__
 - controllers
 - extensions
 - instance
 - models
 - views
 - application.py 2, U**
 - wsgi.py
 - .gitignore
 - Pipfile
 - Pipfile.lock

src > application.py > create_app

```
35
36
37 def create_app(test_config=None):
38     # create and configure the app
39     app = Flask(__name__, instance_relative_config=True)
40     app.config.from_mapping(
41         SECRET_KEY='dev',
42         SQLALCHEMY_DATABASE_URI=os.environ["DATABASE_URL"],
43         JWT_SECRET_KEY="super-secret"
44     )
45
46     if test_config is None:
47         # load the instance config, if it exists, when not testing
48         app.config.from_pyfile('config.py', silent=True)
49     else:
50         # load the test config if passed in
51         app.config.from_mapping(test_config)
52
```

Deploy

A URL de conexão do banco de dados está sendo carregada de uma variável de ambiente chamada **DATABASE_URL**



```
37 def create_app(test_config=None):
38     # create and configure the app
39     app = Flask(__name__, instance_relative_config=True)
40     app.config.from_mapping(
41         SECRET_KEY='dev',
42         SQLALCHEMY_DATABASE_URI=os.environ["DATABASE_URL"],
43         JWT_SECRET_KEY="super-secret"
44     )
45
46     if test_config is None:
47         # load the instance config, if it exists, when not testing
48         app.config.from_pyfile('config.py', silent=True)
49     else:
50         # load the test config if passed in
51         app.config.from_mapping(test_config)
```

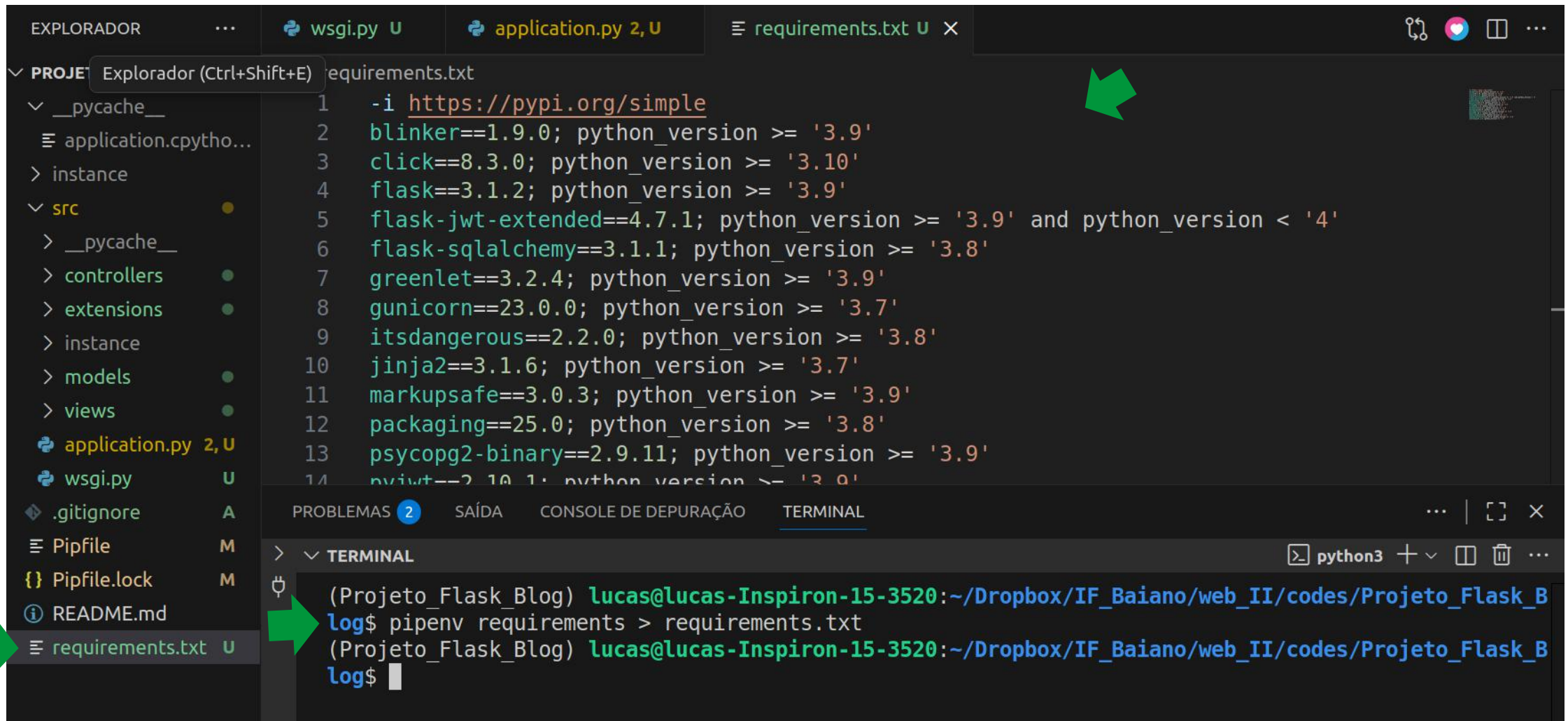
> instance

▼ src

- > __pycache__
- > controllers
- > extensions
- > instance
- > models
- > views
- 🔗 application.py 2, U
- 🔗 wsgi.py U
- 📄 .gitignore A
- 📄 Pipfile M
- 📄 Pipfile.lock M

Deploy

- pipenv requirements > requirements.txt



The screenshot shows a code editor with the following content:

```
EXPLORADOR  ...  wsgi.py U  application.py 2, U  requirements.txt U X
```

PROJECT Explorer (Ctrl+Shift+E)

- __pycache__
- application.cpytho...
- instance
- src
 - __pycache__
 - controllers
 - extensions
 - instance
 - models
 - views
 - application.py 2, U
 - wsgi.py U
- .gitignore A
- Pipfile M
- Pipfile.lock M
- README.md
- requirements.txt U

requirements.txt

```
1 -i https://pypi.org/simple
2 blinker==1.9.0; python_version >= '3.9'
3 click==8.3.0; python_version >= '3.10'
4 flask==3.1.2; python_version >= '3.9'
5 flask-jwt-extended==4.7.1; python_version >= '3.9' and python_version < '4'
6 flask-sqlalchemy==3.1.1; python_version >= '3.8'
7 greenlet==3.2.4; python_version >= '3.9'
8 gunicorn==23.0.0; python_version >= '3.7'
9 itsdangerous==2.2.0; python_version >= '3.8'
10 jinja2==3.1.6; python_version >= '3.7'
11 markupsafe==3.0.3; python_version >= '3.9'
12 packaging==25.0; python_version >= '3.8'
13 pycopg2-binary==2.9.11; python_version >= '3.9'
14 pyjwt==2.10.1; python_version >= '3.9'
```

PROBLEMAS 2 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

python3 + v [] X

TERMINAL

```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ pipenv requirements > requirements.txt
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```

Deploy

```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git add .
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git commit -m "added init_db"
[main elab334] added init_db
 1 file changed, 8 insertions(+)
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 398 bytes | 398.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:lucassampaioleite/flask-blog.git
   748c06f..elab334  main -> main
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```



Lucas's workspace

flask-blog

Search

+ New Upgrade ?

Dashboard

flask-blog

Events

Settings

MONITOR

Logs

Metrics

MANAGE

Environment

Shell

Scaling

Previews

Disks

Jobs

Changelog

Invite a friend

Contact support

Service ID: `srv-d4bp2dogjchc73ctsh5g`

lucassampaioleite / flask-blog main

`https://flask-blog-iokt.onrender.com`

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. Upgrade now

November 15, 2025 at 8:49 AM ✓ Live
748c06f added application

All logs Search Live tail GMT-3

Nov 15 08:52:47 AM [15/Nov/2025:11:52:47 +0000] "HEAD / HTTP/1.1" 404 0 "-" "Go-http-client/1.1"

Nov 15 08:52:53 AM ==> Your service is live

Nov 15 08:52:53 AM ==>

Nov 15 08:52:53 AM ==> Available at your primary URL <https://flask-blog-iokt.onrender.com>

Nov 15 08:52:53 AM ==>

Nov 15 08:52:53 AM ==>

Nov 15 08:52:54 AM [15/Nov/2025:11:52:54 +0000] "GET / HTTP/1.1" 404 207 "-" "Go-http-client/2.0"

Nov 15 08:53:09 AM [15/Nov/2025:11:53:09 +0000] "GET / HTTP/1.1" 404 207 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36"

Nov 15 08:53:10 AM [15/Nov/2025:11:53:10 +0000] "GET /favicon.ico HTTP/1.1" 404 207 "https://flask-blog-iokt.onrender.com/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36"

Nov 15 08:53:17 AM [15/Nov/2025:11:53:17 +0000] "GET /users HTTP/1.1" 308 273 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36"

Nov 15 08:53:17 AM [15/Nov/2025:11:53:17 +0000] "GET /users/ HTTP/1.1" 401 39 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36"

Deploy

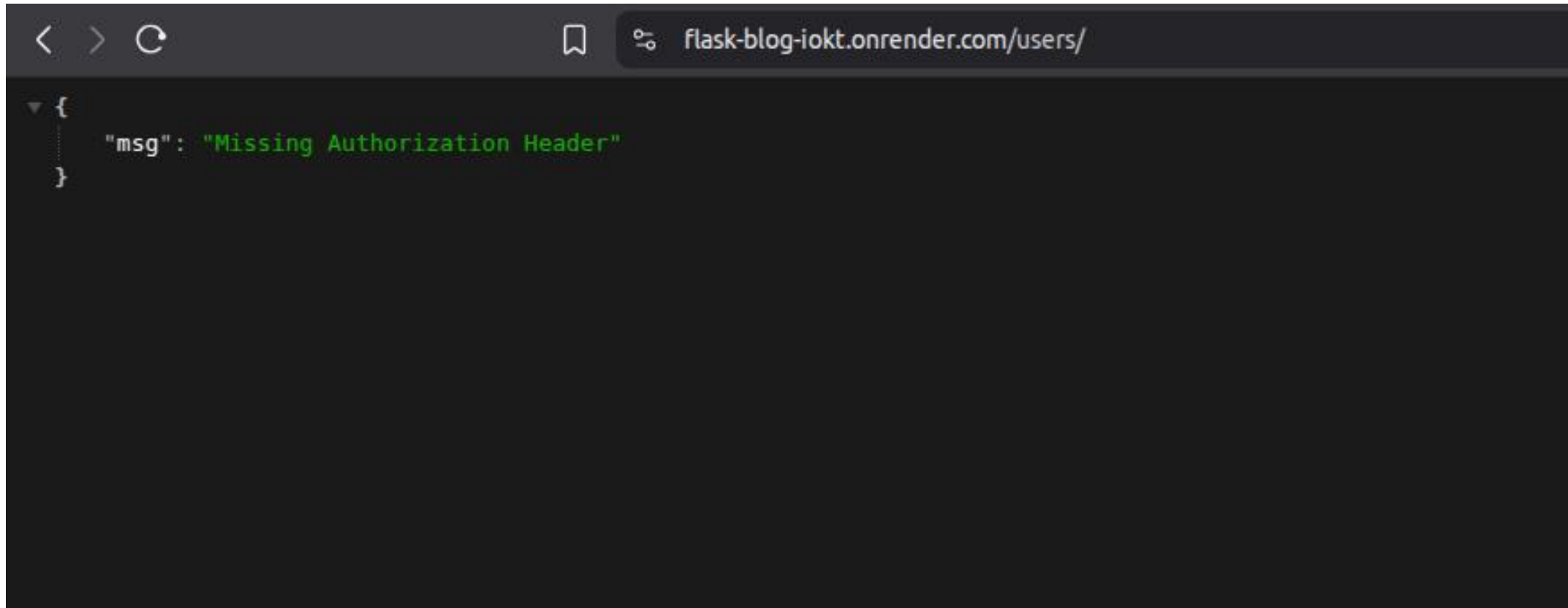


flask-blog-iokt.onrender.com

Not Found

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

Deploy



```
< > ↻ flask-blog-iokt.onrender.com/users/
{
  "msg": "Missing Authorization Header"
}
```

Deploy

Team Workspace

New Import

Collections

Environments

Flows

APIs

History

Search collections

Blog

user

post

auth

POST login

POST login Copy

POST logout

roles

credit-application-system

opencep.com material-geologist...

RESTful API with Java 17 and Springboot 3

udemy_spring_boot_course

User_API

POST login

GET list_users

POST logout

POST login Copy

No environment

Blog / auth / login Copy

Save

Share

POST

https://flask-blog-iokt.onrender.com/auth/login

Send

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Schema

Beautify

```
1 {
2   "username": "admin",
3   "password": "admin123"
4 }
```

Body

Cookies

Headers (12)

Test Results

500 Internal Server Error

345 ms

531 B

Save Response

HTML

Preview

Debug with AI

```
1 <!doctype html>
2 <html lang=en>
3 <title>500 Internal Server Error</title>
4 <h1>Internal Server Error</h1>
5 <p>The server encountered an internal error and was unable to complete your request. Either the server is
  overloaded or
6   there is an error in the application.</p>
```

Deploy



INSTITUTO
FEDERAL

Maiano

Lucas's workspace

flask-blog

Search

+ New Upgrade

Dashboard

flask-blog

Events

Settings

MONITOR

Logs

Metrics

MANAGE

Environment

Shell

Scaling

Previews

Disks

Jobs

Changelog

Invite a friend

Contact support

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

Upgrade now

All logs

Search

Live tail

GMT-3

Nov 15 08:58:24 AM zsd7m File "/opt/render/project/src/.venv/lib/python3.13/site-packages/sqlalchemy/engine/default.py", line 951, in do_execute

Nov 15 08:58:24 AM zsd7m cursor.execute(statement, parameters)

Nov 15 08:58:24 AM zsd7m ~~~~~

Nov 15 08:58:24 AM zsd7m sqlalchemy.exc.ProgrammingError: (psycopg2.errors.UndefinedTable) relation "user" does not exist

Nov 15 08:58:24 AM zsd7m LINE 2: FROM "user"

Nov 15 08:58:24 AM zsd7m ^

Nov 15 08:58:24 AM zsd7m [SQL: SELECT "user".id AS user_id, "user".username AS user_username, "user".email AS user_email, "user".password_hash AS user_password_hash,

Nov 15 08:58:24 AM zsd7m "user".role_id AS user_role_id

Nov 15 08:58:24 AM zsd7m FROM "user"

Nov 15 08:58:24 AM zsd7m WHERE "user".username = %(username_1)s

Nov 15 08:58:24 AM zsd7m LIMIT %(param_1)s]

Nov 15 08:58:24 AM zsd7m [parameters: {'username_1': 'admin', 'param_1': 1}]

Nov 15 08:58:24 AM zsd7m (Background on this error at: <https://sqlalche.me/e/20/f405>)

Nov 15 08:58:24 AM zsd7m 127.0.0.1 - - [15/Nov/2025:11:58:24 +0000] "POST /auth/login HTTP/1.1" 500 265 "-" "PostmanRuntime/7.49.0"

Nov 15 09:14:25 AM zsd7m [2025-11-15 12:14:25 +0000] [56] [INFO] Handling signal: term

Nov 15 09:14:25 AM zsd7m [2025-11-15 12:14:25 +0000] [57] [INFO] Worker exiting (pid: 57)

Nov 15 09:14:26 AM zsd7m [2025-11-15 12:14:26 +0000] [56] [INFO] Shutting down: Master

Nov 15 09:14:25 AM zsd7m [2025-11-15 12:14:25 +0000] [56] [INFO] Handling signal: term

Nov 15 09:14:25 AM zsd7m [2025-11-15 12:14:25 +0000] [57] [INFO] Worker exiting (pid: 57)

Nov 15 09:14:26 AM zsd7m [2025-11-15 12:14:26 +0000] [56] [INFO] Shutting down: Master

Nov 15 09:14:25 AM zsd7m [2025-11-15 12:14:25 +0000] [56] [INFO] Handling signal: term

Nov 15 09:14:25 AM zsd7m [2025-11-15 12:14:25 +0000] [57] [INFO] Worker exiting (pid: 57)

Nov 15 09:14:26 AM zsd7m [2025-11-15 12:14:26 +0000] [56] [INFO] Shutting down: Master

Need better ways to work with logs? Try the [Render CLI](#), [Render MCP Server](#), or set up a [log stream integration](#)

Deploy

```
class User(db.Model):  
    id: Mapped[int] = mapped_column(primary_key=True)  
    username: Mapped[str] = mapped_column(  
        db.String(80), unique=True, nullable=False)  
    email: Mapped[str] = mapped_column(db.String(120), nullable=True)  
    password_hash: Mapped[str] = mapped_column(db.String(300), nullable=False)
```

Aumentar o tamanho do campo da hash

Deploy

```
@click.command("init-db")
def init_db_command():
    with current_app.app_context():
        init_db()

def init_db():
    with current_app.app_context():
        db.create_all()

        # cria a role "admin" se não existir
        role_admin = Role.query.filter_by(name="admin").first()
        if not role_admin:
            role_admin = Role(name="admin")
            db.session.add(role_admin)
            db.session.commit()
            click.echo("Role 'admin' criada!")

        # cria um usuário "admin" se não existir
        if not User.query.filter_by(username="admin").first():
            user = User(username="admin",
```

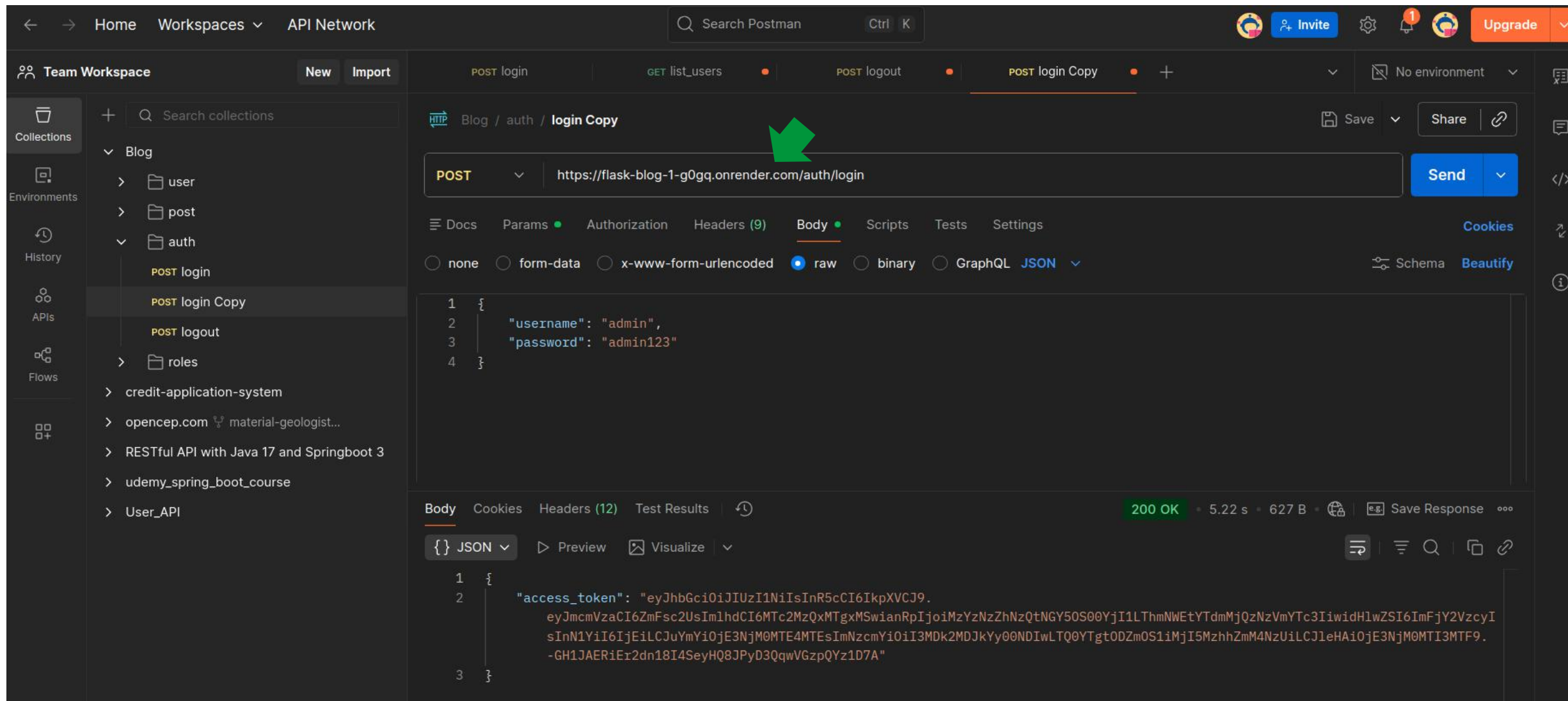
Em application.py,
função create_app()

```
with app.app_context():
    init_db()
```

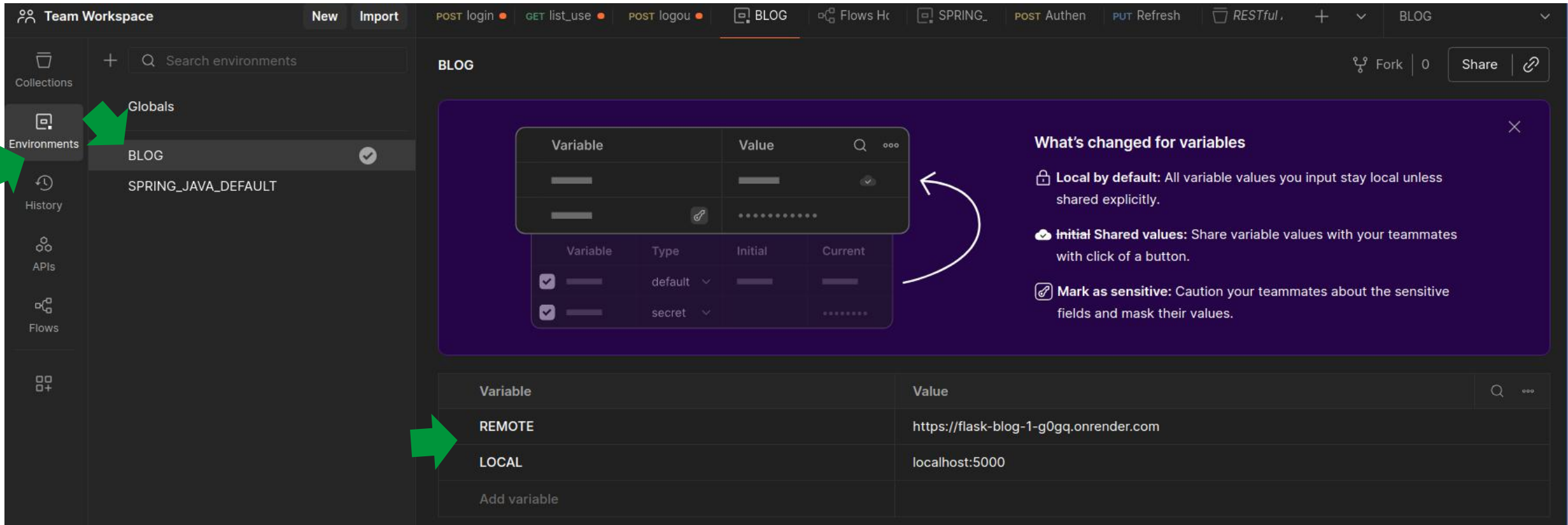
Deploy

```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git add .
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git commit -m "added init_db"
[main elab334] added init_db
1 file changed, 8 insertions(+)
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 398 bytes | 398.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:lucassampaioleite/flask-blog.git
748c06f..elab334 main -> main
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```

Deploy



Deploy



Team Workspace

New Import

POST login GET list_use POST logou BLOG Flows Hc SPRING_ POST Authen PUT Refresh RESTful, + - BLOG

Search environments

Collections

Environments

Globals

BLOG

SPRING_JAVA_DEFAULT

History

APIs

Flows

BLOG

Fork 0 Share

Variable	Value
	

What's changed for variables

- ☐ **Local by default:** All variable values you input stay local unless shared explicitly.
- ☐ **Initial Shared values:** Share variable values with your teammates with click of a button.
- ☐ **Mark as sensitive:** Caution your teammates about the sensitive fields and mask their values.

Variable	Type	Initial	Current
☒	default		
☒	secret		

Variable	Value
REMOTE	https://flask-blog-1-g0gq.onrender.com
LOCAL	localhost:5000
Add variable	

Deploy

[illegible]

Dúvidas



PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática
Lucas Sampaio Leite

