



Flask

RESTAPI COM FLASK

MICROFRAMEWORK WEB

COMPONENTES

Fabiano de Santana

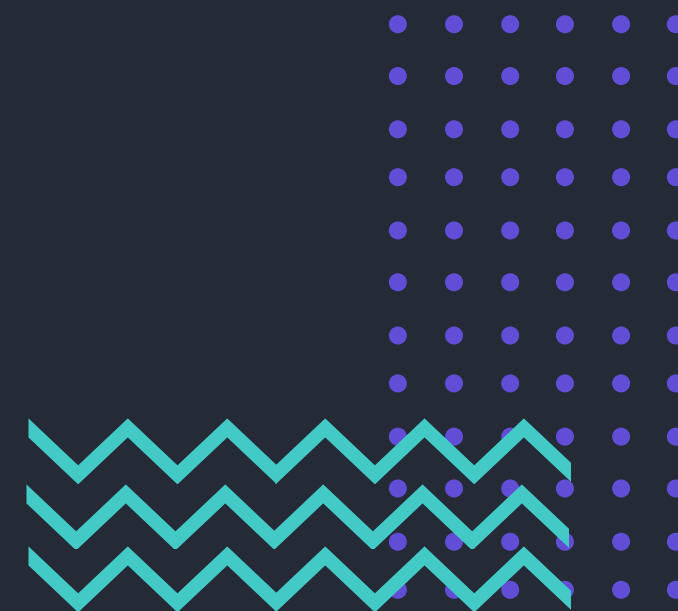
Leôncio Ferreira

Sywrah Gabriella



O QUE É UM MICROFRAMEWORK?

**SIGNIFICA QUE O OBJETIVO É MANTER
O NÚCLEO SIMPLES, MAS EXTENSÍVEL.**

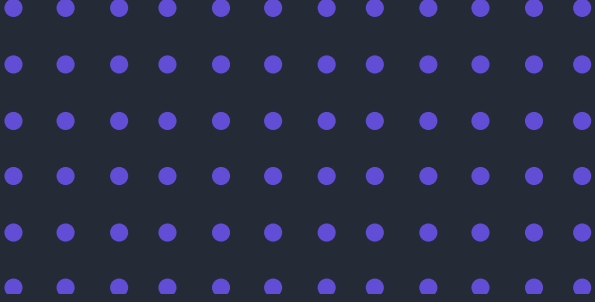






QUEM É FLASK NA FILA DO PÃO?

**FLASK É UM SIMPLES, MAS PODEROSO
MICROFRAMEWORK QUE PERMITE CRIAR
DE FORMA EFICIENTE APLICAÇÕES WEB.**



- 
- 
- **DEPOIS DE CONFIGURADO O AMBIENTE CRIAMOS UM ARQUIVO APP.PY PARA TESTAR**
 - **CONFIGURAMOS O NOME DO APP UM VARIÁVEL DE AMBIENTE .**
 - **RODAMOS ENTÃO O FLASK QUE IDENTIFICA O APP E RODA NO LOCALHOST DO COMPUTADOR.**
- 

ACOMPANHAR O CODIGO

[HTTPS://GITHUB.COM/FABIO
MSRS/FLASK_PROJECT](https://github.com/fabioMSRS/FLASK_PROJECT)



HELLO WORLD

A ESTRUTURA

- NO FLASK EXISTE O CONCEITO DE MODULOS , EM QUE EXISTE UMA PASTA PRINCIPAL PARA OS ARQUIVOS E MODULOS QUE SÃO USADOS PELO APP.
- PARA CRIAR UM MODULO BASTA CRIAR UMA PASTA COM "__INIT__.PY" DENTRO.

CRIANDO O MODULO

- **DENTRO DA PASTA PRINCIPAL CRIAMOS UM DIRETORIO COM O NOME DO NOSSO MODULO.**
- **CRIAMOS O ARQUIVO `__init__.py`**
- **NA RAIZ DO PROJETO CRIAMOS UM ARQUIVO `index.py`**
- **ESTE SERÁ A PORTA DE ENTRADA DO NOSSO PROJETO.**

INICIANDO O APP

- NA RAIZ DO PROJETO CRIAMOS UM ARQUIVO BOOTSTRAP.SH.
- ESTE ARQUIVO É UM SCRIPT PARA RODAR O APP AUTOMATICAMENTE

CRIANDO UM ENDPOINT

- NO ARQUIVO INDEX.PY NOS VAMOS DEFINIR AS ROTAS DA APLICAÇÃO JUNTO COM OS METODOS COM SUAS RESPECTIVAS RESPOSTAS.
- AS ROTAS SÃO DEFINIDAS PELO DECORADOR `@APP.ROUT('URL',METHODS[])`
- E RETONAMOS UM JSON COM AS RESPOSTAS

TESTANDO

HORA DE MODELAR

- DENTRO DO MODULO DO NOSSO PROJETO , CRIAMOS UM NOVO MODULO DESTINADO AOS MODELOS.
- DENTRO DESTES MODULO NOS CRIAMOS NOSSOS MODELOS.
- JUNTO DE CADA CLASSE É PRECISO CRIAR UM ESQUEMA DA MESMA PARA MAPEAR OS SEUS CAMPOS.

ESQUEMAS

- SERÃO USADO NA SERIALIZAÇÃO E DESSECCIALIZAÇÃO DOS NOSSOS OBJETOS.
- ISTO É FEITO ATRAVÉS DO PACOTE MARSHMALLOW.

ORGANIZAÇÃO

- PARA QUE NOSSO MODELOS NÃO FIQUEM EXPOSTOS NOS ENDPOINTS DA API, VAMOS CRIAR UMA CLASSE INTERMEDIARIA.

LÁ E DE VOLTA OUTRA VEZ

- NO INDEX.PY DA NOSSA API NOS CRIAMOS OS ENDPOINTS DA APLICAÇÃO.
- ESTE ENDPOINT VÃO RECEBER OS OBJETOS ME JSON E GUARDADOS COMO OBJETOS.
- ALEM DE RECUPERAR ESTES OBJETOS E ENTREGAR UM JSON.

TA PRONTO O SORVETINHO

CONSIDERAÇÕES FINAIS

CONSIDERAÇÕES FINAIS

- O DESENVOLVIMENTO COM FLASK É BEM ÁGIL E SIMPLIFICADO.
- NOSSA API POSSUI APENAS O ESSENCIAL.
- MAS AO MESMO TEMPO É EXTREMAMENTE ESCALÁVEL, POIS SE SURGIR A NECESSIDADE APENAS É ADICIONADO NOVOS PACOTE.

CONTRAS

- **DEPENDENDO DA SITUAÇÃO PODE SE TER A SENSACÃO DE QUE ESTÁ SEMPRE FALTANDO ALGO.**
- **CASO UM PROJETO JÁ INICIE COM UM ESCOPO GRANDE, É NECESSÁRIO VASCULHAR ATRÁS DE MUITAS DEPENDENCIAS PARA COMEÇAR A IMPLEMENTAR.**

REFERÊNCIAS

- [HTTP://FLASK.POCOO.ORG/](http://flask.pocoo.org/).
- [HTTPS://AUTH0.COM/BLOG/DEVELOPING-RESTFUL-APIS-WITH-PYTHON-AND-FLASK/](https://auth0.com/blog/developing-restful-apis-with-python-and-flask/)

FIM.