Principais conceitos do Backend e configurações iniciais com Flask

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Configuração do Poetry
- Configuração do Flask
- Conceitos principais de Backend e para que cada um serve
- Bancos de dados
- Segurança
- Escalabilidade

Configuração do Poetry

bashonwindows install instructions

```
curl -sSL
https://raw.githubusercontent.com/python-poetry/poetry/master/
get-poetry.py | python
```

Após a instalação ser concluída, acesse a pasta bin do Poetry, na pasta do seu usuário, e adicione nas variáveis de ambiente. dentro do PATH o caminho de execução.

Fonte: https://python-poetry.org/docs/

Configuração do Flask com Poetry

Após a instalação e configuração do Poetry, iremos seguir o passo a passo abaixo:

- Selecione a pasta em que criar o novo diretório e execute o comando: **poetry new --name app --src NOME_DA_PASTA**
- Acesse o diretório e remova os itens desnecessários;
- Adicione a configuração inicial da virtualenv usando o comando: **poetry config --local virtualenvs.in-project true**
- Adicione o Flask na aplicação executando o comando: poetry add flask

Configuração do Flask com Poetry

- Dentro da pasta src/app, no arquivo __init__.py, iremos adicionar o trecho de código inicial para compilar a aplicação:

```
from flask import Flask

app = Flask(__name__)

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

Configuração do Flask com Poetry

- Feito o passo anterior, agora iremos criar um arquivo na raiz do projeto, chamado app.py, com o seguinte trecho de código:

```
from src.app import app


if __name__ == '__main__':
  app.run()
```

- Agora, para executar a aplicação, iremos acessar a pasta .venv, utilizando o comando: **.\.venv\Scripts\activate.ps1 (Windows)**
- Após estar no ambiente virtualizado, é somente executar o comando: **poetry run flask run**

Configuração do Flask com Poetry

- Organização de configurações dentro do projeto para trabalhar com diversos ambientes:
- Iremos criar um arquivo chamado .env na raiz e iremos adicionar o seguinte código:

```
FLASK_APP=app.py
FLASK_ENV=development
```

- Nesse momento, iremos adicionar uma biblioteca para trabalhar com environments do projeto, utilizando o comando: **poetry add python-dotenv**
- Na pasta src/app, iremos criar uma nova pasta chamada config e dentro dela um arquivo chamado __init__.py e adicionaremos o seguinte trecho de código:

```
import os
from dotenv import load_dotenv,
find_dotenv
load_dotenv(find_dotenv())
```

Configuração do Flask com Poetry

- Ainda no mesmo arquivo src/app/config/__init__.py, adicionaremos o restante do código abaixo:

```
class Development(object):
    DEBUG = True
    TESTING = False
class Production(object):
    DEBUG = False
    TESTING = False
app_config = { 'development': Development, 'production': Production}
```

Configuração do Flask com Poetry

- Por fim, nessa parte iremos retornar para o __init__.py da pasta src/app e iremos adicionar o seguinte código:

```
import os
from src.app.config import app_config


#Após a execução da variável app
app.config.from_object(app_config[os.getenv('FLASK_ENV')])
```

Conceitos principais de Backend e para que cada um serve

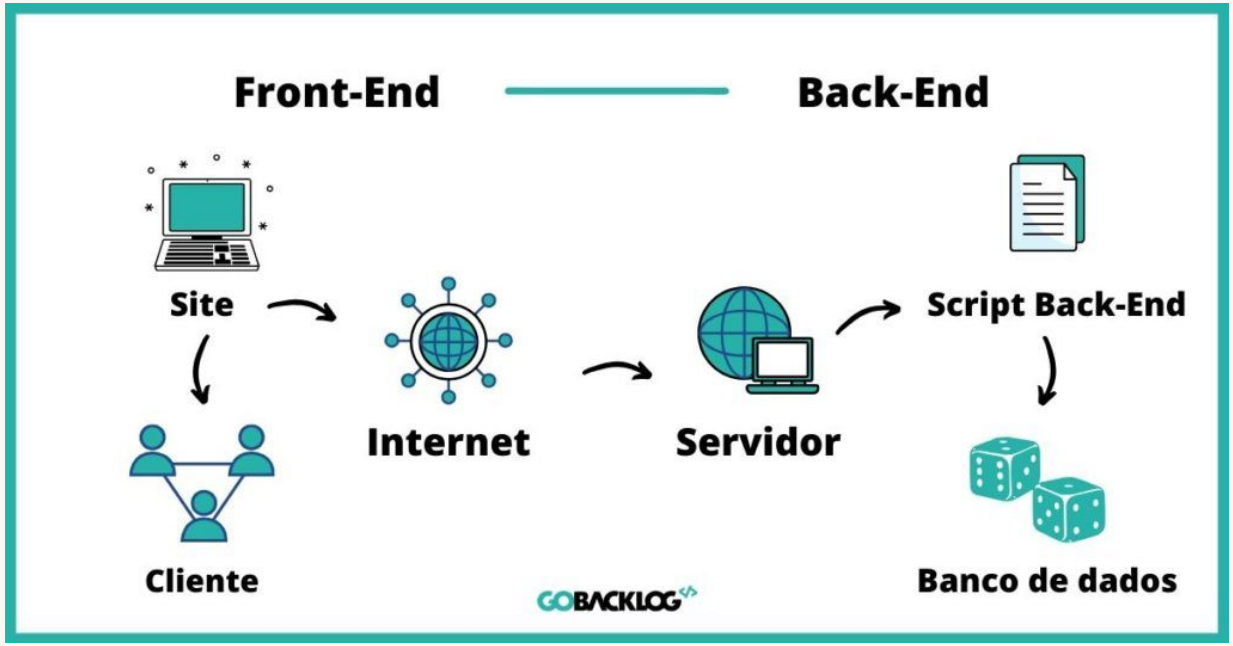
Desde que trabalhamos com aplicações Frontend, vimos a necessidade de tratar os dados fornecidos e muitas vezes regras de negócio do projeto que estão no projeto, portanto, por mais que alteremos nosso Frontend, a regra do sistema permanece a mesma de acordo com os dados fornecidos pelo Backend, mas será que não existe mais conceitos envolvidos?

Bancos de dados

Os bancos de dados, por sua vez, são como uma enorme fonte de informações, e é por meio deles que as aplicações se tornam dinâmicas:

- MySQL: É um sistema open-source de gerenciamento de base de dados relacional.
- PostgreSQL: É um sistema gerenciador de banco de dados objeto relacional, desenvolvido como projeto de código aberto.
- MongoDB: De acordo com a documentação oficial do MongoDB, o Mongo é um banco de dados baseado em documentos com escalabilidade e flexibilidade, tornando mais simples as consultas e indexação.
- DynamoDB: O Amazon DynamoDB é um banco de dados de chave-valor NoSQL, sem servidor e totalmente gerenciado, projetado para executar aplicações de alta performance em qualquer escala.

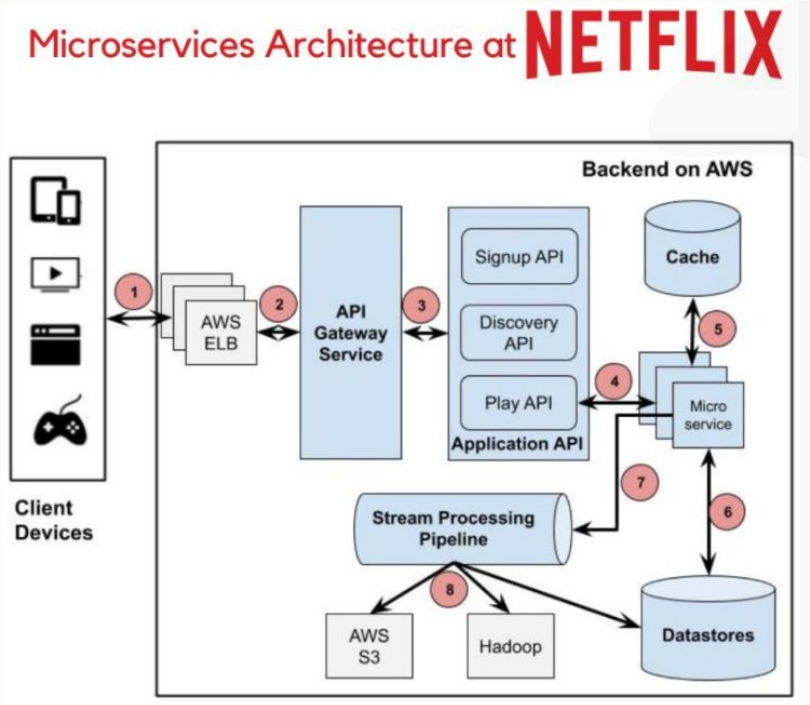
Segurança



Fonte: https://gobacklog.com/blog/back-end-guia-para-empreendedores/

Escalabilidade

Desenvolver um sistema escalável significa que este sistema possui a capacidade de crescer de acordo com o crescimento do negócio ao qual ele está associado.



Fonte: https://smartstudios.io/blog/serving-200-million-online-subscribers-the-netflix-way/

DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!

SENAI

<LAB365>