

Interfaces Web - Forum IPT.

Relatório de Trabalho Prático Interfaces Web.

**António Gonçalves n °23787,
Paulo —----- n° XXXX**

Resumo

O nosso trabalho prático para a unidade curricular "Interfaces Web" envolveu a implementação de uma aplicação utilizando uma stack um tanto quanto personalizada composta por React.js para o frontend, Azure Functions para a criação de endpoints e PostgreSQL hospedado na AWS como banco de dados.

A arquitetura da aplicação foi estruturada de forma a aproveitar as vantagens de cada tecnologia escolhida bem como a disponibilidade delas serem gratuitas. O React.js foi utilizado para desenvolver a interface do utilizador, proporcionando uma experiência interativa e responsiva aos utilizadores finais.

Para gerenciar a lógica do backend, foram utilizadas Azure Functions, que são funções serverless (cloud). Estas funções foram responsáveis por criar os endpoints necessários para manipulação de dados e interação com a base de dados.

A base de dados escolhida foi o PostgreSQL, hospedado na AWS (Amazon Web Services).

A combinação destas tecnologias proporcionou uma solução mais robusta para o desenvolvimento da nossa aplicação web. O React.js através dos seus componentes oferece uma interface coesa e amigável, às Azure Functions trazem o conforto de garantir a execução eficiente das operações no backend, e o PostgreSQL na AWS oferece uma base de dados confiável e segura e relacional. A integração destas tecnologias possibilitou desenvolver uma aplicação completa, abrangendo desde a interface do utilizador até à gestão eficiente dos dados nos bastidores.

Índice

Resumo.....	5
Introdução.....	7
Como Iniciar a aplicação (https://cyb3rwolf945.github.io/PEWEB/).....	9
1º Forma.....	9
2º Forma.....	9
Disponibilização Online.....	10
Desafios.....	13
Conclusão.....	14
Bibliografia.....	16

Introdução

O projeto final da unidade curricular "Interfaces Web" exigiu a criação de uma aplicação react utilizando a plataforma proposta pelo professor “sheety”, para a criação das funcionalidades CRUD, bem como a nossa base de dados.

Apesar destas recomendações, o nosso grupo preferiu adotar um caminho diferente no desenvolvimento desta aplicação, utilizando ferramentas em associação ao react como MUI (Material UI) e Tailwind.

No backend, optou-se por utilizar Azure Functions, aproveitando suas capacidades serverless. Sendo que estas funções ficaram responsáveis por criar os endpoints necessários para a manipulação de dados e interação com a base de dados.

A escolha do PostgreSQL como base de dados, hospedada também em cloud na AWS, trouxe confiabilidade e segurança para o armazenamento e gerenciamento eficiente dos dados.

Em resumo, a combinação estratégica de React.js, Azure Functions e PostgreSQL na AWS permitiu-nos o desenvolvimento de uma aplicação web mais completa, destacando-se pela sua versatilidade em termos de funções e componentes funcionais no Frontend.

FrontEnd

React js

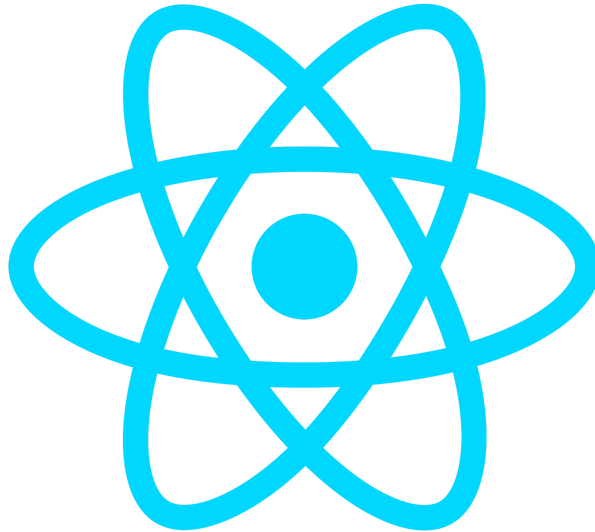


Figura 1 - React.js logotipo

O React.js é uma biblioteca JavaScript amplamente utilizada para desenvolvimento de interfaces web. A sua popularidade deve-se, em parte, à sua abordagem por componentes, onde é possível realizar a construção de uma página a partir da conjunção de vários componentes reutilizáveis, tornando a abordagem DRY (Don't Repeat Yourself) mais eficiente. Com o React, é possível construir interfaces interativas e responsivas de forma mais facilitada. Além disso, o conceito de virtual DOM contribui para um desempenho otimizado, atualizando apenas as partes necessárias da página.

Como Iniciar a aplicação (<https://cyb3rwolf945.github.io/PEWEB/>)

Existem 2 formas de iniciar o projeto localmente, ambas envolvem a criação de um web server a diferença é uma estar incorporada no próprio python e a outra está incorporada no VS Code como um plugin facilitando a sua utilização.

1º Forma

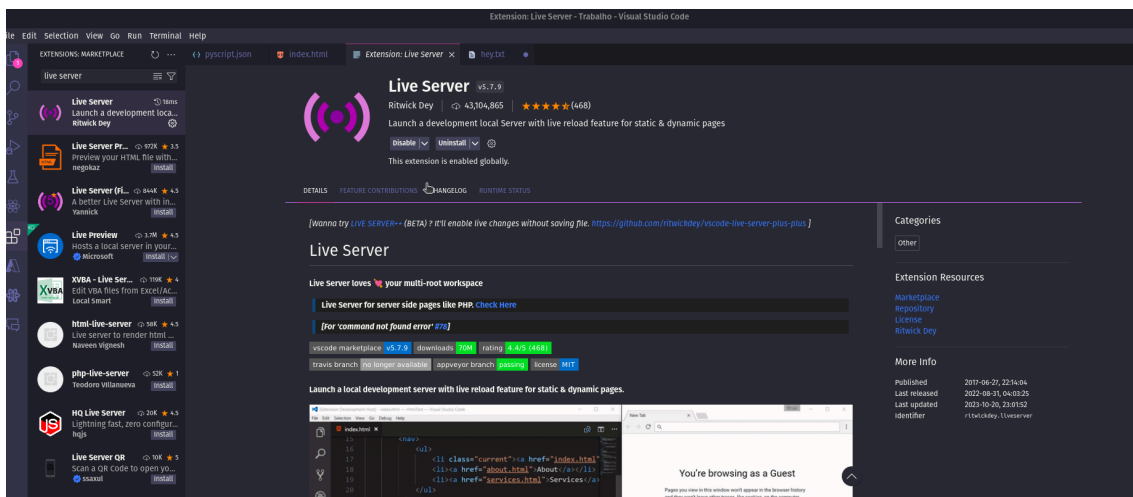
A primeira forma para ser possível iniciar o projeto localmente usando o servidor HTTP implementado no Python 3, basta ter o Python instalado no computador. No root da pasta onde se encontram os arquivos, execute o seguinte comando a partir do terminal:

```
python3 -m http.server
```

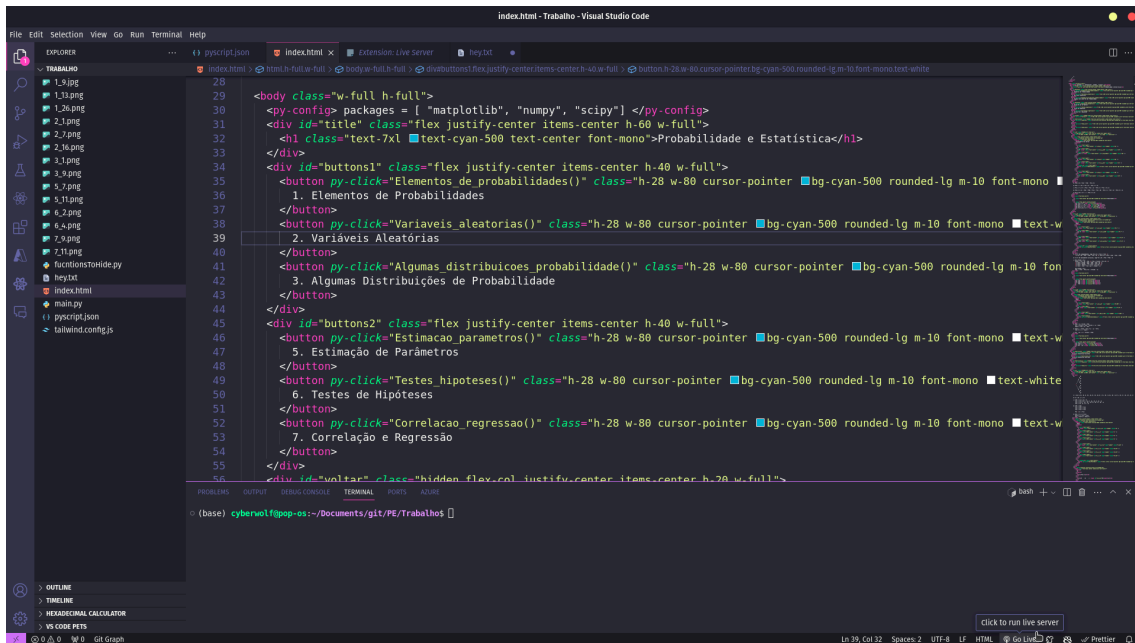
Este comando iniciará um servidor web local. Em seguida, basta abrir o navegador no url “http://localhost:8000”. Lembre-se de atualizar a página (CTRL-R) para visualizar quaisquer atualizações que você tenha feito nos arquivos.

2º Forma

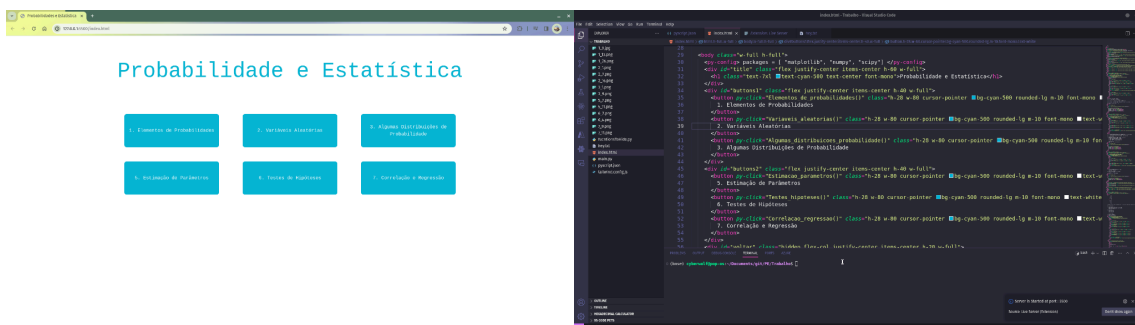
Esta segunda forma permite facilitar a vida ao programador na hora dos testes e da inicialização da aplicação fazendo uso da extensão do Visual Studio code designada por “Live Server”.



Onde uma vez instalada é só abrir o ficheiro HTML no VS code e carregar no canto inferior direito onde diz “Go Live”.

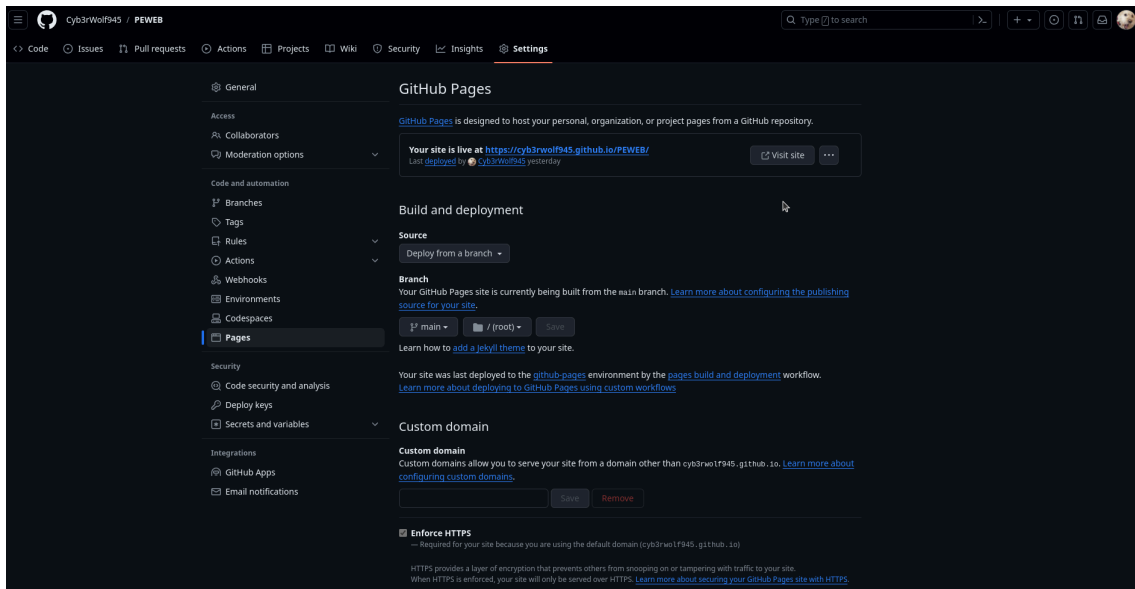


Após clicar na respectiva opção uma janela no navegador será aberta automaticamente.



Disponibilização Online

Visto que queríamos tornar este trabalho o mais interativo e fácil de usar possível e visto ser uma solução web, decidimos usar o Github Pages sendo este um serviço oferecido pelo GitHub que permite aos seus utilizadores hospedar páginas web diretamente a partir de seus repositórios do GitHub.



Podendo a nossa solução ser acedida através do respectivo link:
<https://cyb3rWolf945.github.io/PEWEB/>

Desafios

Conclusão

Bibliografia