



WHISP

2º Projeto

Tecnologias e Programação Web

23/24

Miguel Miragaia, 108317

Vasco Faria, 107323

Rodrigo Graça, 107634

Índice

Introdução.....	3
Objetivos	3
Aplicação e Usos	4
REST API	5
Execução da Aplicação.....	6
Contas de Teste	9
Conclusão	10

Introdução

Este projeto tem como propósito a criação de uma rede social elementar por meio da integração do Angular Framework e Django REST Framework. A meta central consiste em estabelecer uma plataforma que permita aos usuários se registarem, criarem posts, realizar uploads de imagens e interagirem entre si de maneiras diversas

Objetivos

Objetivos	Done
Development of a web application (client and server side), to allow the information access in an easy and intuitive way.	X
For the client side, it must be developed with Angular technology, to provide all needed user interfaces.	X
For server side, it must use Django REST Framework (DRF), to provide a set of REST Web Services, to provide all needed functionalities in back-end, to allow all kind of operations: data searching, insertion, updating, deleting and all subsequent validations	X
Development of an authentication system, using e.g. Django Rest Framework Authentication.	X
In order to be accessed from anywhere and use a real n-tier software architecture, the deployment can be done for the different application's layers. The server side (DRF) can be deployed at pythonanywhere.com, as before for the Django application, and the client side at Heroku	X (exceto Heroku)
A small report describing the main fundamental parts implemented, and any other information needed to run the application, is necessary.	X

Aplicação e Usos

A Whisp consiste numa rede social com as seguintes funcionalidades:

- Registo e autenticação de Users (Django Rest Framework Authentication);
- Troca de password
- Criação e visualização de Posts (Upload de Texto, Video, Imagem)
- Página de Feed (Posts de todos os users)
- Perfil de User (username, Name, email, Biography, Date joined, Last Login Date, User posts)
- Personalização de informações e Imagem no Perfil
- Remover Post (Permitido ao admin, e ao user a quem o Post pertence)

Autenticação: Criada através da geração de um token próprio do Django REST Framework que será associado ao User. Útil para operações futuras, operações que tenham a necessidade de autenticação.

Autorização: Nível de operações permitidas às diferentes entidades. (Criar Posts, Perfil, etc...)

Entidades: Users; Admin; Posts

Para um funcionamento correto da app é notaria a diferença nas funcionalidades de um User com login para um User sem login. Através do Token de autenticação são concebidas autorizações e desta forma um maior espectro de interações, como por exemplo a permissão de fazer um Post, visto que este está sempre associado a um User, seria impossível fazê-lo anonimamente.

REST API

A criação de endpoints de 4 categorias (POST; DELETE; GET; PUT) leva a um acesso simples aos valores do Django para cada entidade. Para um melhor entendimento do projeto é apresentado de seguida toda a Documentação da API criada:

ENDPOINT	METHOD	DESCRIPTION
posts/	GET	Retorna os posts de todos os users
posts/create/	POST	Criação de um post
profile/user/<number:id>/	PUT	Atualização de um perfil de um user por id.
profile/user/<number:id>/	GET	Obtenção de um perfil de um user por id.
change-password/	PUT	Troca de password de um user já existente.
user_info/<str:username>/	GET	Obtenção de informação por username.
user_info/<str:username>/	PUT	Atualização de informação por username.
rest-auth/login/	POST	Login de user existente para obtenção do token de autenticação de user.
rest-auth/registration/	POST	Registo de um novo user, fornece o token de autenticação de user.

Execução da Aplicação

A execução do sistema é permitida de 2 maneiras, sendo elas localmente ou por um Deployment.

Localmente

Para a execução localmente podemos executar os seguintes comandos:

- `"pip install -r requirements.txt"`
- `"source venv/bin/activate"` no diretorio Whisp
- `"python3 manage.py runserver"` no diretorio Whisp
- `"npm start"` no diretorio Whisp_Angular

Desta forma corremos o Django e o Angular localmente e acedendo através da porta 4200.

Deployment

Para o deployment foi seguido as diretrizes do projeto e enquanto a API REST era suportada através do PythonAnywhere, porém o fronted nao foi hospedado no Heroku devido às suas políticas não estarem a permitir que fizéssemos o deploy da nossa app. Posto isto o deploy do frontend foi feito no Netlify, e apesar das repetidas mensagens de succes no deploy e build sempre que tentávamos aceder remotamente pelo link fornecido obtiamos o popup de "Page Not Found".

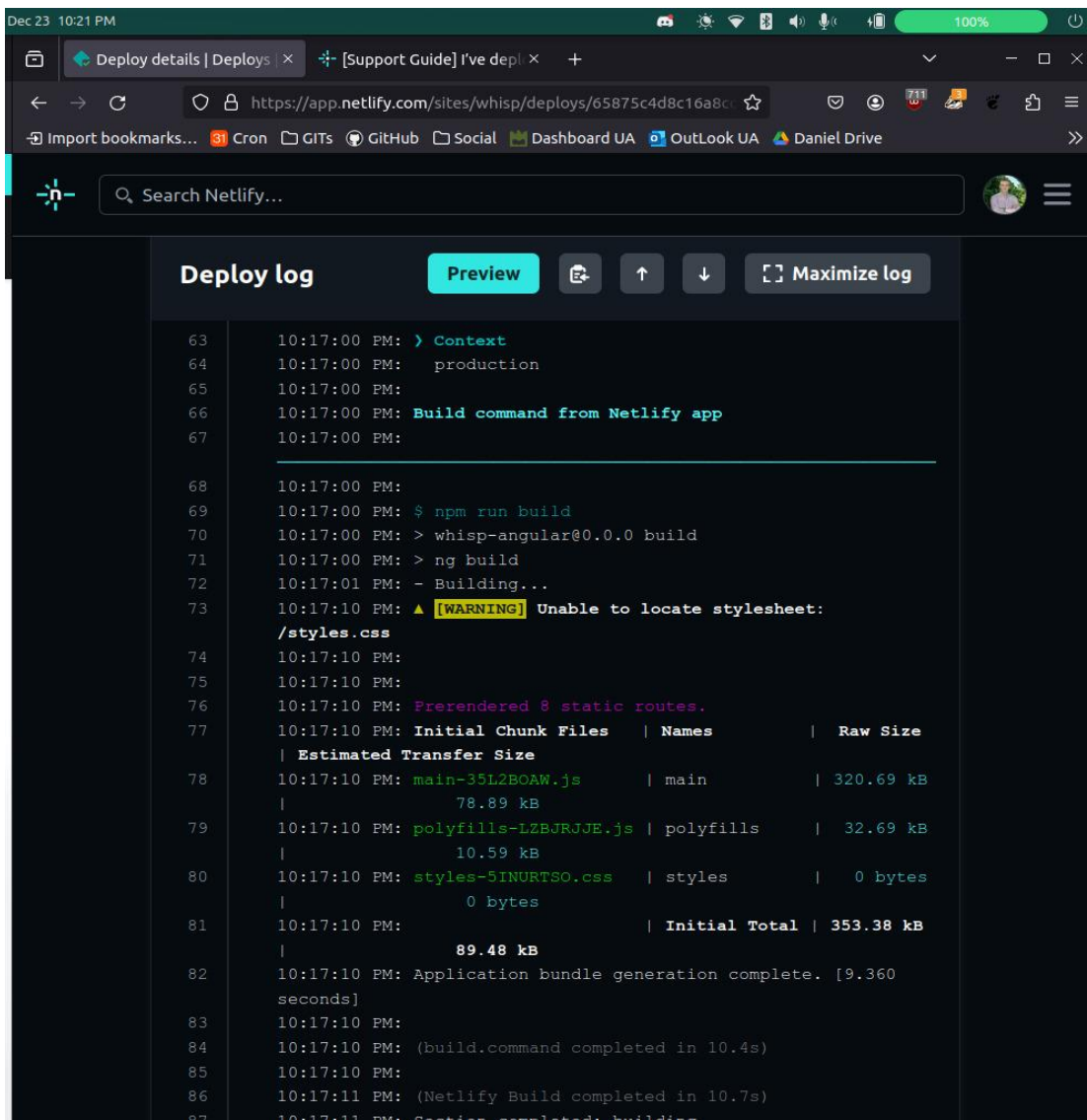
PythonAnywhere:

<http://vascomfaria.pythonanywhere.com/admin/>

Netlify:

<https://whisp.netlify.app>

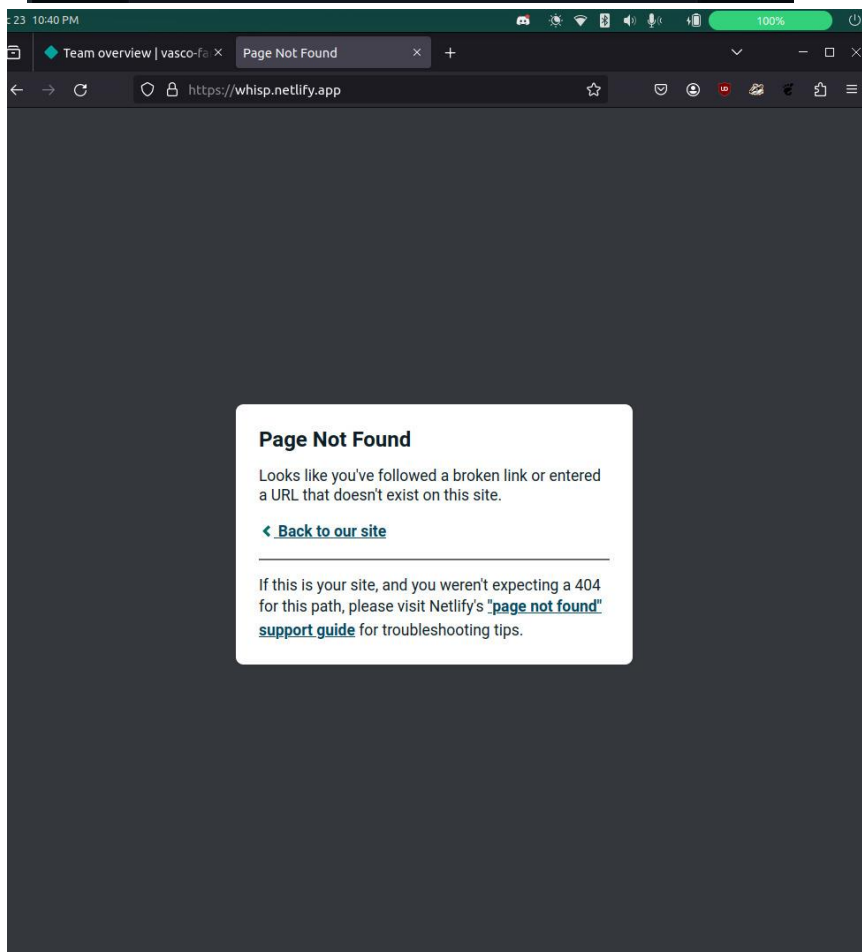
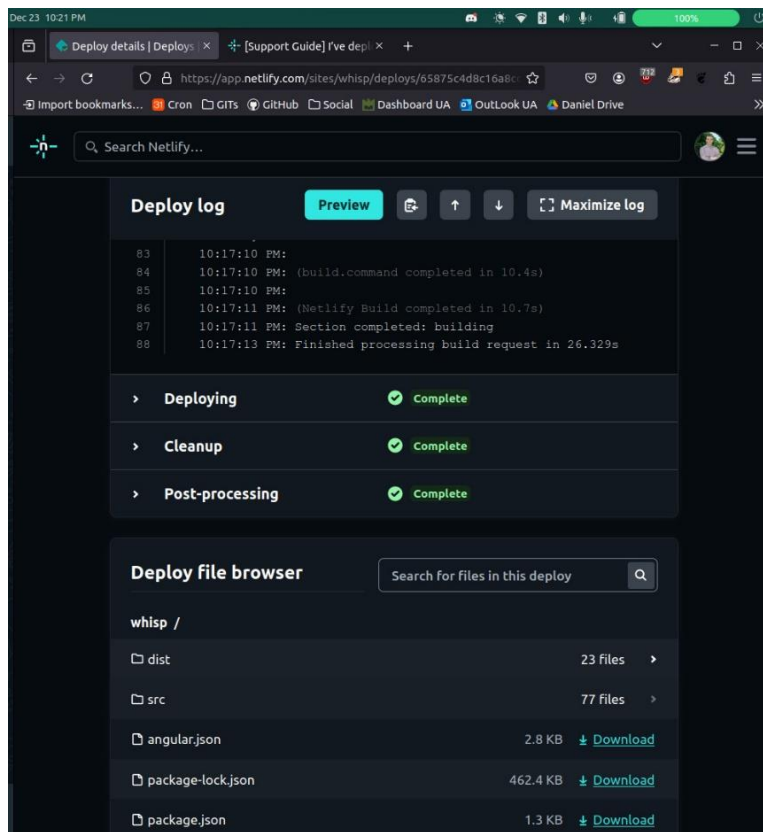
Para o bom funcionamento da app é necessário então correr o frontend Angular localmente e o usar o deploy do backend do pythonanywhere.



```
Dec 23 10:21 PM
Deploy details | Deploys x [Support Guide] I've deplo x +
https://app.netlify.com/sites/whisp/deploy/65875c4d8c16a8c
Import bookmarks... Cron GITs GitHub Social Dashboard UA Outlook UA Daniel Drive
Search Netlify...

Deploy log Preview [Icons] Maximize log

63 10:17:00 PM: > Context
64 10:17:00 PM: production
65 10:17:00 PM:
66 10:17:00 PM: Build command from Netlify app
67 10:17:00 PM:
68 10:17:00 PM:
69 10:17:00 PM: $ npm run build
70 10:17:00 PM: > whisp-angular@0.0.0 build
71 10:17:00 PM: > ng build
72 10:17:01 PM: - Building...
73 10:17:10 PM: ▲ [WARNING] Unable to locate stylesheet:
74 /styles.css
75 10:17:10 PM:
76 10:17:10 PM: Prerendered 8 static routes.
77 10:17:10 PM: Initial Chunk Files | Names | Raw Size
78 | Estimated Transfer Size
79 10:17:10 PM: main-35L2BOAW.js | main | 320.69 kB
80 | 78.89 kB
79 10:17:10 PM: polyfills-LZBJRJJE.js | polyfills | 32.69 kB
80 | 10.59 kB
80 10:17:10 PM: styles-5INURTSO.css | styles | 0 bytes
81 | 0 bytes
81 10:17:10 PM: | Initial Total | 353.38 kB
82 | 89.48 kB
82 10:17:10 PM: Application bundle generation complete. [9.360
83 seconds]
83 10:17:10 PM:
84 10:17:10 PM: (build.command completed in 10.4s)
85 10:17:10 PM:
86 10:17:11 PM: (Netlify Build completed in 10.7s)
87 10:17:11 PM: Section completed: building
```



Contas de Teste

Foram desenvolvidas contas de teste para uma melhor experiência no website, ainda que este seja de alta intuição e permita uma adaptação e conhecimento rápido sem contexto.

Credenciais de teste:

User:

- **Username:** srpadre
- **Password:** amenamen

Admin:

- **Username:** admin
- **Password:** olaolaola

Conclusão

O projeto Whisp, desenvolvido com Angular e Django REST Framework, resultou em uma aplicação de rede social intuitiva. A implementação eficaz da autenticação com tokens, combinada com a REST API, proporcionou funcionalidades robustas de registo, autenticação e interação entre usuários.

O deployment bem-sucedido no PythonAnywhere e Heroku garante acessibilidade global. Apesar dos objetivos alcançados, identificamos oportunidades de melhoria contínua, incluindo aprimoramentos na interface do usuário e considerações adicionais de segurança.

Em resumo, o projeto Whisp destaca a eficiência da integração Angular-Django na criação de uma aplicação web funcional e moderna.