Engenharia de Software

LEI/LDM 2018/2019

ARQUITETURA DE SOFTWARE

[19/11/2018]

[PIUC]

Tabela de versões

Versão	Data	Autores	Descrição
1.0	19/11/2018	Inês Moreira, Rita Garrido, Yussara Monteiro, João Carvalho, Michael Ortet	Documento inicial

Índice

1. Introdução

Este documento tem como objetivo dar a conhecer a arquitetura de software do produto. Para isso, começar-se-á por fazer a descrição do tipo de software utilizado no seu desenvolvimento, o motivo da escolha desse software e as aplicações a serem integradas no produto.

Posteriormente, serão mencionados os tipos de utilizadores a quem a aplicação se destina e, por fim, serão apresentados os diagramas que irão mostrar a relação entre utilizadores-aplicações-produto desenvolvido, traduzindo assim, a arquitetura de software.

2. Opções de software consideradas

2.1 Opções de frameworks consideradas

Para o desenvolvimento da plataforma web de investigação da Universidade de Coimbra foram consideradas duas plataformas de desenvolvimento em Python: Django e Flask. De seguida apresenta-se uma breve descrição de ambas as plataformas.

2.1.1 Django

O Django é uma *framework* que segue o padrão model-template-view. Destina-se à criação de aplicações de maior grau de complexidade e possui uma série de *packages* que permitem aos programadores a rápida implementação do produto. Uma vez que o Django inclui um ORM (Object-Relational Mapping), permite que bases de dados relacionais tais como o PostgreSQL, Oracle, MySQL e SQLite interajam com os dados gerados pela aplicação.

2.1.2 Flask

O Flask é uma *microframework* destinada a aplicações com requisitos mais simples. No entanto, é mais leve, rápida e proporciona bastante liberdade de escolha ao programador, no que diz respeito a livrarias externas e *plugins*, sendo desta forma, mais flexível.

2.2 Opções de bases de dados consideradas

2.2.1 SQLite

O SQLite é uma biblioteca desenvolvida em linguagem C que implementa uma base de dados SQL, relacional e auto-suficiente, ou seja, não depende de bibliotecas externas. Todas as operações são feitas localmente e o acesso aos dados é feito diretamente no local onde a base está armazenada. As linguagens como o Python conseguem aceder aos dados através de uma interface de acesso pública disponibilizada pelo fabricante. Deste modo, os dados são lidos e processados a cada requisição pelo motor da base de dados.

3. Software escolhido

Após a análise dos *software* considerados, a equipa de Implementação e de Testes escolheram como plataforma de desenvolvimento a *framework* Django.

A plataforma de desenvolvimento Flask deixa a cargo do programador a implementação da maioria das funcionalidades e por isso adequa-se melhor a equipas com alguma experiência no desenvolvimento de aplicações web e que pretendam soluções personalizadas. Uma vez que as equipas de Implementação e Testes não possuíam tal experiência e uma vez que a *framework* Django oferece liberdade suficiente para cumprir os requisitos do produto, considerou-se que a última seria a que permitiria alcançar os objetivos do produto de forma mais eficiente.

4. API's

4.1 Twitter API's

Para que a aplicação aceda a conteúdos do Twitter foi usada a biblioteca para Python Twython[1].

Para procurar *tweets* cuja *hashtag* esteja de acordo com os interesses do investigador foi usada a biblioteca Python Twitter Search API - search-tweets-python[2].

4.2 Reddit

Para que a aplicação aceda a conteúdos do Reddit foi usada a biblioteca para Python PRAW (The Python Reddit API Wrapper)[3].

[1] https://github.com/ryanmcgrath/twython
[2] https://github.com/twitterdev/search-tweets-python
[3] http://github.com/praw-dev/praw

5. Tipos de utilizadores

Existe apenas um utilizador, o investigador que se regista na plataforma PIUC com *username* e *password*. Nome, apelido, email, ORCID, *research interests, afiliation* e *research unit* são dados de disponibilização obrigatória aquando da criação da conta.

Uma vez autenticado, o utilizador pode consultar uma lista de *posts* do Twitter, de acordo com os interesses de investigação (*research interests)* colocados no processo de registo. Pode marcar determinados *posts* como favoritos e consultá-los a partir da lista respectiva.

Existe a possibilidade de guardar determinados endereços web como *bookmark*, acessíveis depois a partir da lista de *bookmarks*.

O investigador pode ainda, depois de registado, iniciar sessão no Twitter e fazer tweets a partir da plataforma PIUC e iniciar sessão no Reddit e ver o último post de cada *subreddit* que segue.

6. Descrição da arquitetura

6.1 Contexto do sistema

O diagrama do contexto do sistema é utilizado para mostrar os tipos de utilizador do produto e as ações que poderão vir a desempenhar. Serve também para dar a conhecer as aplicações que externas que irão permitir cumprir as funcionalidades especificadas no SRS (Software requirements Specification) (V2.2).

Como se pode observar, existe apenas um tipo de utilizador (o investigador) e as aplicações associadas são o Reddit e o Twitter. As funcionalidades a que o utilizador tem acesso estão descritas abaixo, bem como a utilidade do Twitter e do Reddit.

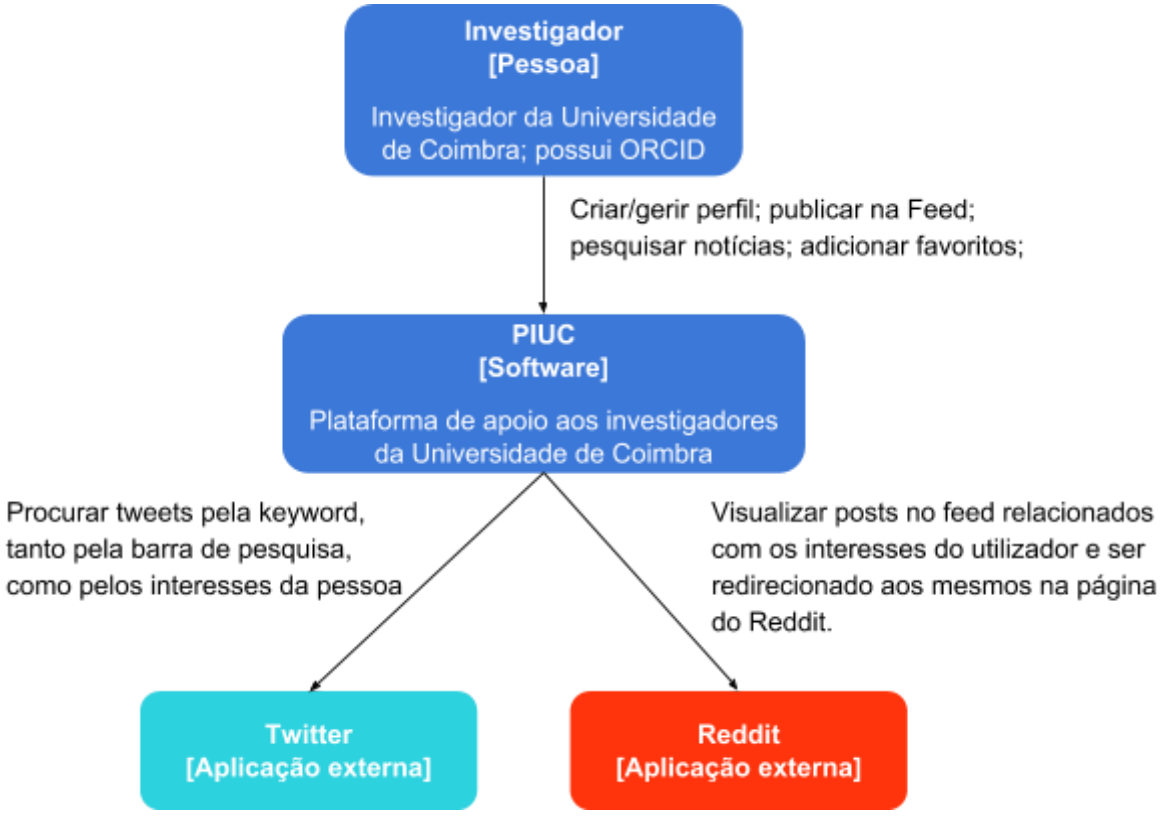

Figura 1 - Diagrama do contexto do sistema.

6.2 Containers

Neste diagrama é apresentado o *container* utilizado e o modo como funciona a sua interação com a aplicação e o utilizador. A descrição das ligações e de cada componente é apresentada no esquema abaixo.

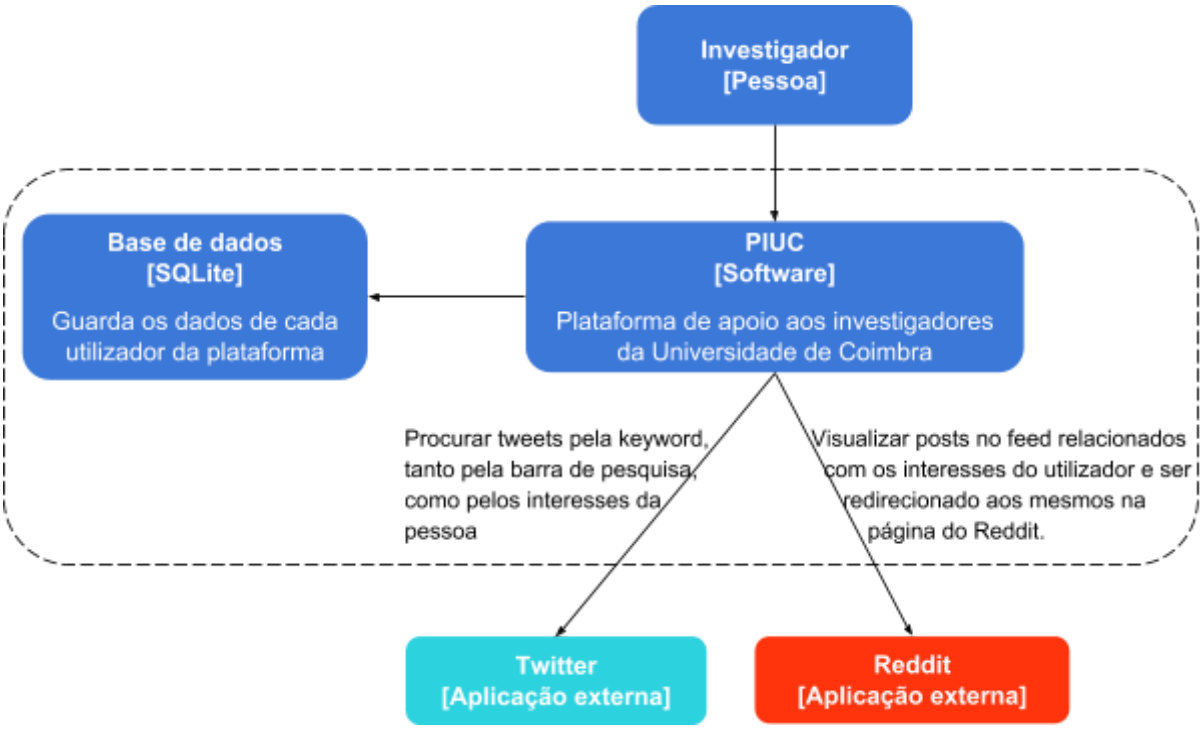

Figura 2 - Diagrama de *containers*.

6.3 Componentes

No diagrama abaixo estão representados os componentes necessários para implementar os requisitos descritos no Documento SRS (V2.2) e a interação entre eles e o container do projeto. As descrições encontram-se no esquema.

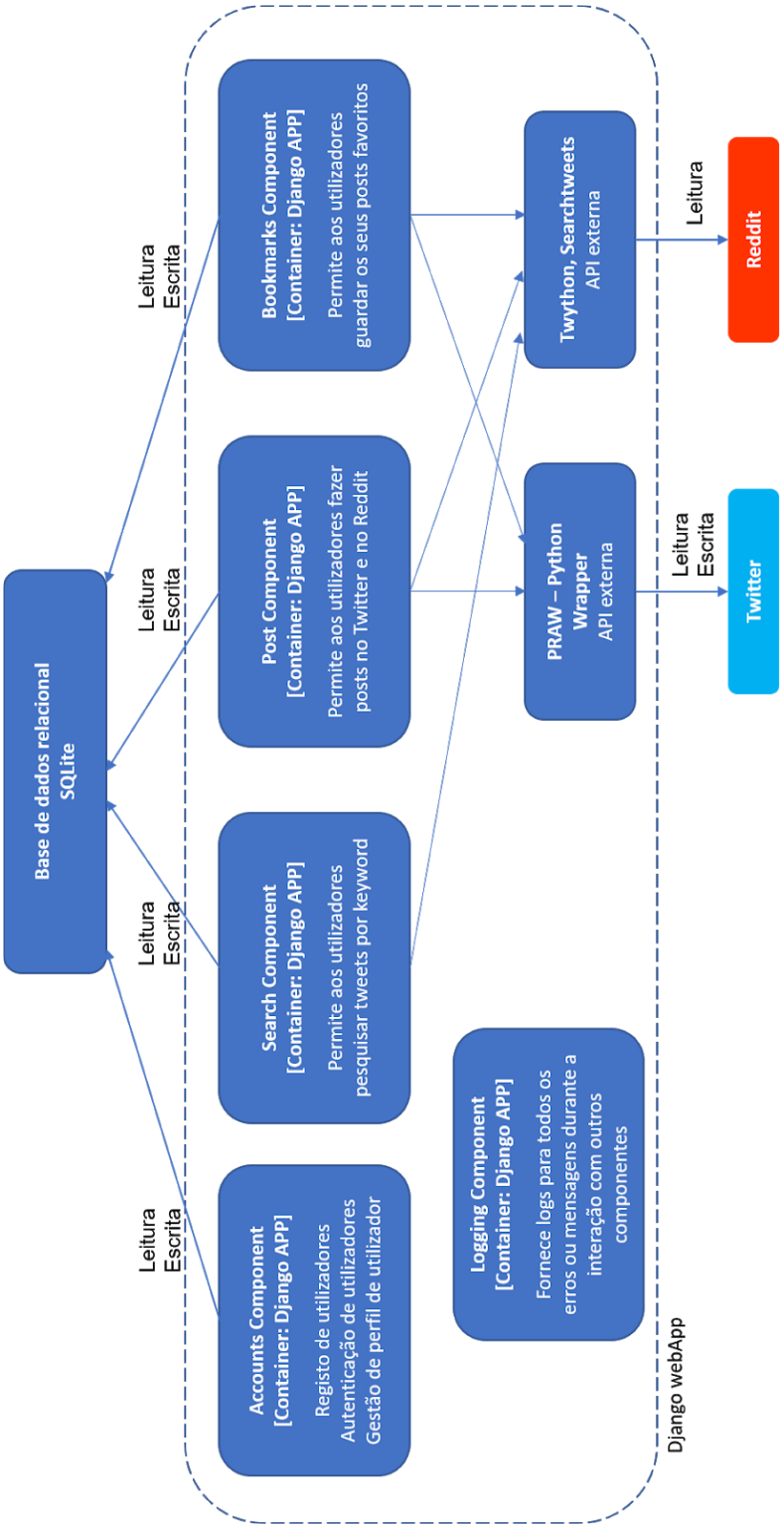

Figura 3 - Diagrama de componentes do sistema.

6.4 Diagrama de componentes do utilizador

No diagrama abaixo estão representados os componentes necessários para implementar os requisitos descritos no Documento SRS (V2.2) e a interação entre eles e o utilizador da aplicação.

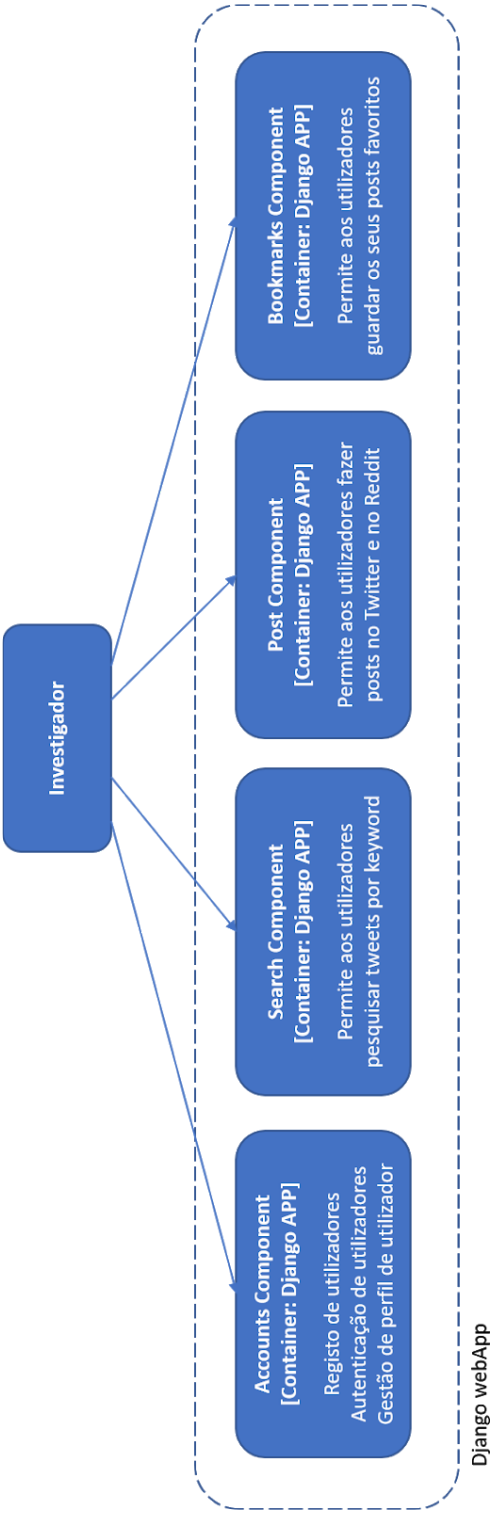

Figura 4 - Diagrama de componentes do utilizador.

6.5 Diagrama da base de dados

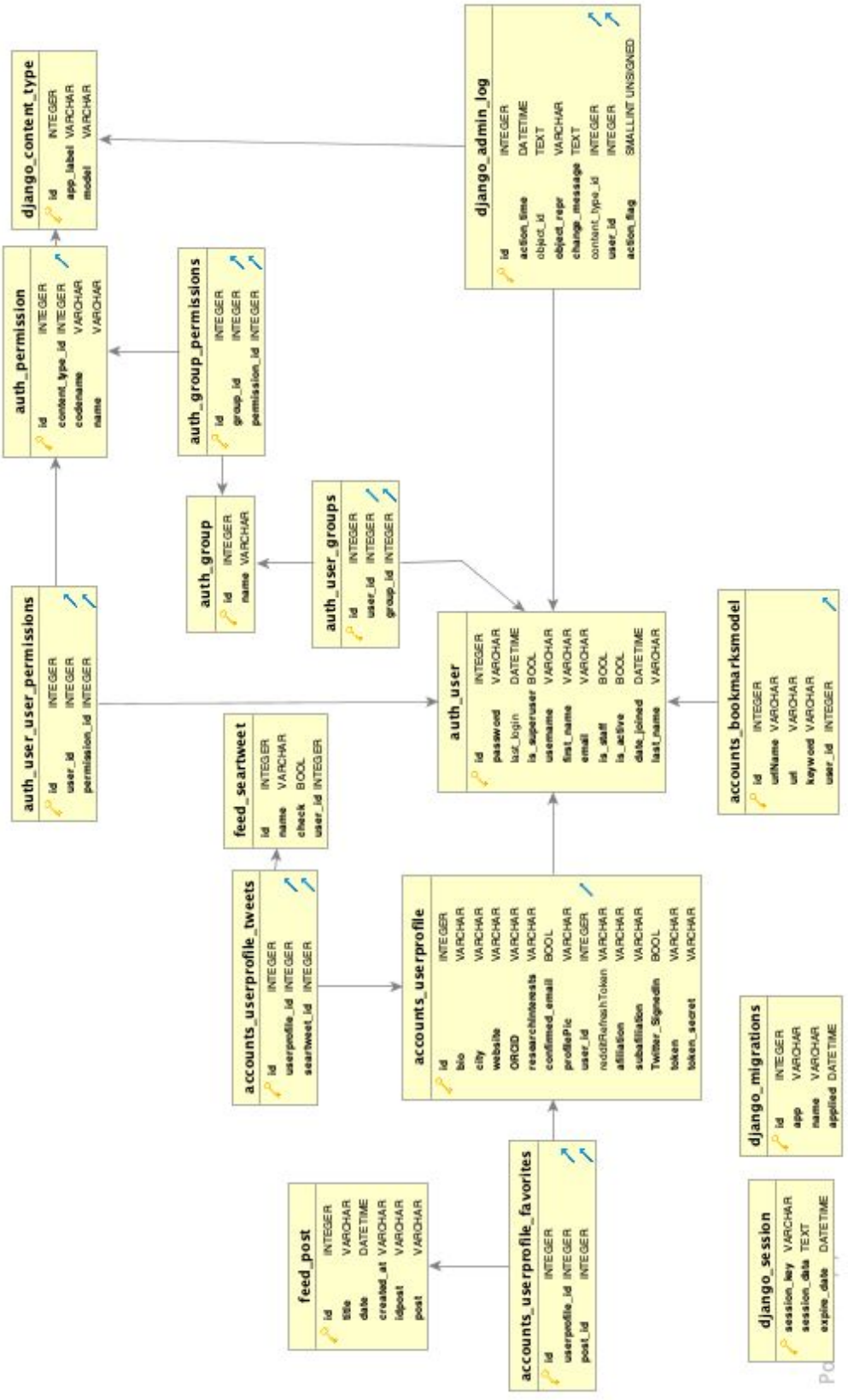

Figura 5 - Diagrama da base de dados gerado com a ferramenta DbVisualizer.

7. Conclusão

Ao longo deste documento apresentou-se a arquitetura de software da plataforma PIUC, incluindo as principais decisões em termos de linguagem de programação, *frameworks* e bibliotecas.

As escolha de Python e Django deveu-se à experiência prévia dos membros da equipa de Implementação e Testes com a linguagem Python e à pouca experiência no desenvolvimento de aplicações web.

O motor de base de dados utilizado actualmente é o SQLite, tanto em desenvolvimento como e em produção. O modelo de base de dados é especificado e actualizado recorrendo ao mecanismo de *Migrations* do Django[4].

Procurou detalhar-se, sob a forma de diagramas, aspectos importantes como os componentes de sistema, componentes de utilizador e o modelo de dados.

[4] https://docs.djangoproject.com/en/2.1/topics/migrations/