

# Definição do projeto

---

## Conceito da aplicação

Ezployee é uma plataforma útil e pertinente, que conecta empresas a potenciais futuros funcionários, sendo uma das suas principais funcionalidades a publicação de ofertas de emprego, com salários, modelos de trabalho, contactos da empresa para a candidatura, entre outras informações.

Ver site no [Python Anywhere](#). Caso o documento HTML não seja carregado, num primeiro acesso, refresque a página.

## Fundamentação da escolha

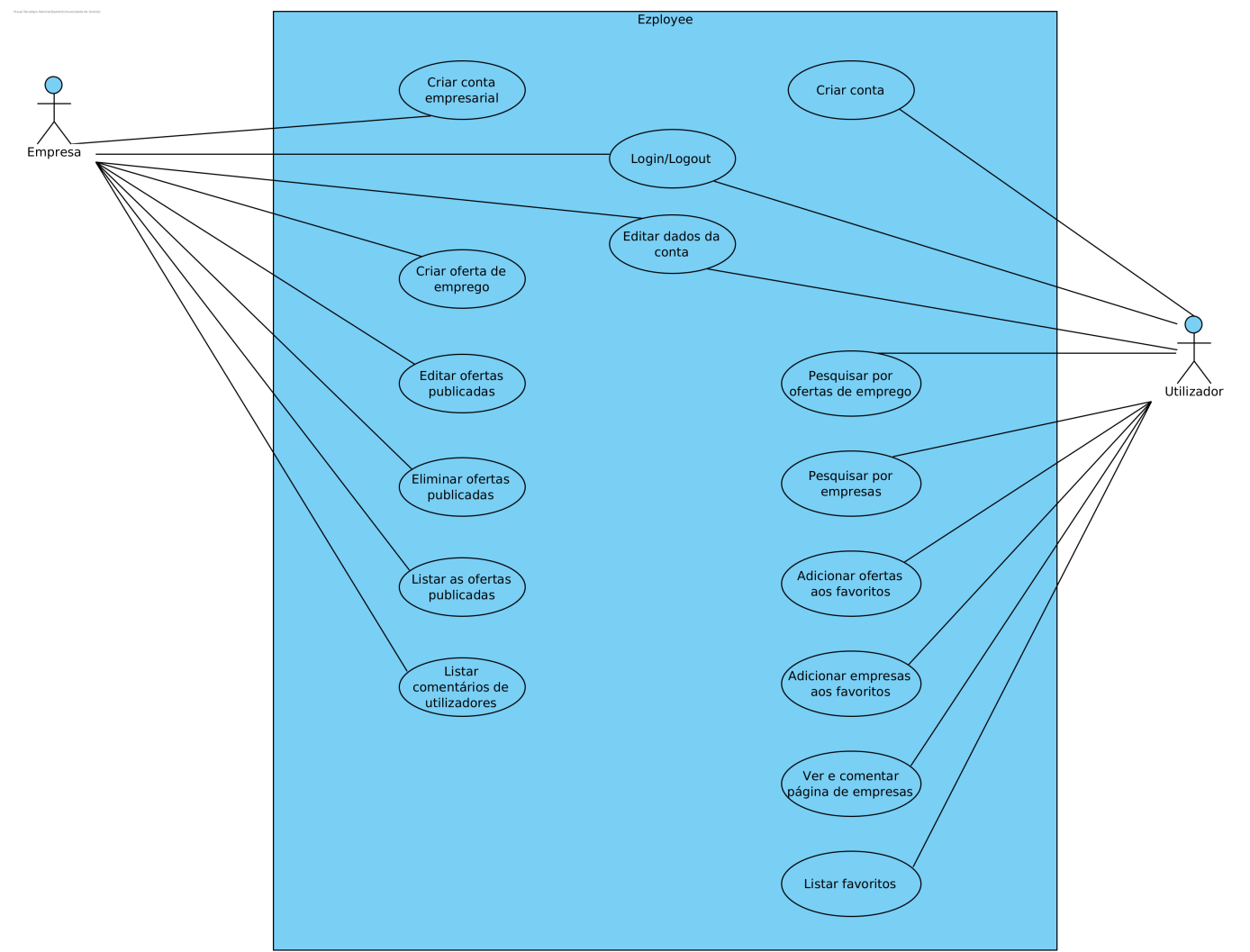
A transformação digital dos vários setores da sociedade é inegável e tem vindo a acelerar significativamente, sobretudo desde a pandemia. Perante a competitividade do mercado de trabalho, as empresas tiveram de se adaptar rapidamente a um novo paradigma. Não só o modelo de trabalho foi reformulado, passando a ser possível trabalhar remotamente, como também a procura e o recrutamento de novos funcionários sofreram alterações radicais, recorrendo-se frequentemente a plataformas digitais. Posto isto, o desenvolvimento de aplicações, que possibilitem a divulgação de ofertas de emprego e aproximem as empresas dos potenciais candidatos, é imperativo e motivou-nos para a escolha da temática deste projeto.

## Levantamento de requisitos

### Método utilizado

Como não conhecíamos ninguém que atuasse na área de Recursos Humanos, não era possível realizar uma entrevista, pelo que optámos por estudar sistemas análogos. A nossa principal referência foi o [Glassdoor](#), ainda que diferindo um pouco de algumas funcionalidades que implementámos. Por exemplo, nesta plataforma, as *reviews* são anónimas, contrastando com o nosso sistema, onde os comentários ficam associados a um nome de utilizador e a uma avaliação da empresa (numa escala de 1 a 5).

Diagrama de casos de utilização



**Atores:**

- Empresa
- Utilizador

Mapeamento dos requisitos funcionais (CaU) para *user stories***Empresa**

Use Case	User Story
Registar empresa	<b>Como</b> empresa, <b>quero</b> registar-me na plataforma, <b>para</b> poder publicar ofertas de emprego.
Login	<b>Como</b> empresa, <b>quero</b> fazer login na plataforma, <b>para</b> poder publicar ofertas de emprego.
Logout	<b>Como</b> empresa, <b>quero</b> fazer logout da plataforma, <b>para</b> poder sair da minha conta.
Publicar oferta	<b>Como</b> empresa, <b>quero</b> publicar uma oferta de emprego, <b>para</b> poder encontrar candidatos.
Editar oferta	<b>Como</b> empresa, <b>quero</b> editar uma oferta de emprego, <b>para</b> que atualizar a sua informação.
Eliminar oferta	<b>Como</b> empresa, <b>quero</b> eliminar uma oferta de emprego, <b>para</b> poder remover ofertas que já não estão disponíveis.
Ver perfil	<b>Como</b> empresa, <b>quero</b> ver o meu perfil, <b>para</b> poder ver a informação da minha empresa.
Editar perfil	<b>Como</b> empresa, <b>quero</b> editar o meu perfil, <b>para</b> poder atualizar a informação da minha empresa.
Ver comentários	<b>Como</b> empresa, <b>quero</b> ver os comentários, <b>para</b> poder ver a opinião dos utilizadores.

**Utilizador**

Use Case	User Story
Registar utilizador	<b>Como</b> utilizador, <b>quero</b> registar-me na plataforma, <b>para</b> poder pesquisar ofertas de emprego.
Login	<b>Como</b> utilizador, <b>quero</b> fazer login na plataforma, <b>para</b> poder pesquisar ofertas de emprego.
Logout	<b>Como</b> utilizador, <b>quero</b> fazer logout da plataforma, <b>para</b> poder sair da minha conta.
Pesquisar ofertas	<b>Como</b> utilizador, <b>quero</b> pesquisar ofertas de emprego, <b>para</b> poder encontrar uma oferta que se adeque às minhas necessidades, limitações e ambições.
Filtrar ofertas	<b>Como</b> utilizador, <b>quero</b> filtrar ofertas de emprego, <b>para</b> poder encontrar uma oferta que se adeque às minhas necessidades, limitações e ambições.
Pesquisar por empresas	<b>Como</b> utilizador, <b>quero</b> pesquisar por empresas, <b>para</b> poder encontrar uma empresa que se adeque às minhas necessidades, limitações e ambições.
Filtrar empresas	<b>Como</b> utilizador, <b>quero</b> filtrar empresas, <b>para</b> poder encontrar uma empresa que se adeque às minhas necessidades, limitações e ambições.
Ver detalhes da oferta	<b>Como</b> utilizador, <b>quero</b> ver os detalhes da oferta, <b>para</b> poder ver a informação da oferta.
Ver comentários na empresa	<b>Como</b> utilizador, <b>quero</b> ver os comentários na empresa, <b>para</b> poder ver a opinião dos utilizadores.
Ver detalhes da empresa	<b>Como</b> utilizador, <b>quero</b> ver os detalhes da empresa, <b>para</b> poder ver a informação da empresa.
Comentar empresa	<b>Como</b> utilizador, <b>quero</b> comentar uma empresa, <b>para</b> poder dar a minha opinião sobre a empresa.
Adicionar oferta (favoritos)	<b>Como</b> utilizador, <b>quero</b> adicionar aos favoritos uma oferta, <b>para</b> poder guardar a oferta para consulta posterior.
Adicionar empresa (favoritos)	<b>Como</b> utilizador, <b>quero</b> adicionar aos favoritos uma empresa, <b>para</b> poder guardar a empresa para consulta posterior.
Ver favoritos	<b>Como</b> utilizador, <b>quero</b> ver os meus favoritos, <b>para</b> poder ver as ofertas e empresas que guardei.
Ver perfil	<b>Como</b> utilizador, <b>quero</b> ver o meu perfil, <b>para</b> poder ver a informação da minha conta.
Editar perfil	<b>Como</b> utilizador, <b>quero</b> editar o meu perfil, <b>para</b> poder atualizar a informação da minha conta.

# Tecnologias

---

## Geração de dados

Servimo-nos da ferramenta [Mockaroo](#), para gerar datasets conectados e coerentes, que nos permitissem testar as funcionalidades do Ezployee.

Depois de correr os comandos `python3 manage.py makemigrations app` + `python3 manage.py migrate`, populámos as nossa tabelas com os dados gerados.

## Estilização do site

- **Bootstrap** (inclui Popper.js): framework de desenvolvimento web, que oferece componentes para a construção da interface de utilizador (UI).
- **Google Fonts**, para fontes textuais.
- **Font Awesome**, com ícones.

### Porquê usar Bootstrap e não puro CSS?

- Responsividade da *viewport* (a área visível pelo utilizador).
  - Pré-definição de **breakpoints**: larguras da *viewport*, acima dos quais a disposição dos elementos do documento é alterada.
- Aparência moderna e agradável, que favorece a experiência do utilizador (UX).
- Uso de componentes pré-definidos, que incluem animações já implementadas. Com isto, o desenvolvimento do site é mais rápido, na medida em que o foco dos *developers*, ao nível do front-end, passa a ser a declaração e a programação dos elementos do site, ao invés do seu design.

## Autenticação dos utilizadores

Não usámos o objeto `django.contrib.auth.models.User`, pois queríamos outros atributos que esse não inclui. No entanto, por questões de segurança, não guardamos a password em texto livre, mas sim o seu hash. Para esse efeito, aplicámos os métodos `make_password` e `check_password` do módulo de autenticação do Django.

## Autocomplete na pesquisa de empresas/ofertas

**jQuery**, mais concretamente o **jQueryUI**: uma coleção de widgets para web bastante úteis.

## Formulários

Como aprendido nas aulas práticas, os **Django Forms** são formulários flexíveis, que permitem a validação de dados, a geração automática do HTML correspondente e até a associação direta a *models*, referenciando-os numa `class Meta`, dentro de uma subclasse do `django.forms.ModelForm`.

Com vista a aumentar o nosso controlo sobre o modo como os formulários são apresentados, instalámos os **django-widgets-tweaks**, que nos permite, entre outras funcionalidades, injetar classes CSS nos *forms*.

## Deployment da aplicação

---

Tal como solicitado, escolhemos o **Python Anywhere** como serviço remoto de hospedagem. É considerado *Platform as a Service*, uma vez que as máquinas virtuais que disponibiliza já vêm com a Bash e o Python instalados, suportam várias frameworks, como o Django, e têm uma Web Server Gateway Interface (WSGI) praticamente configurada.

## Base de Dados

---

Durante o desenvolvimento da aplicação, a nível local, utilizámos uma base de dados relacional embutida e baseada em ficheiros (**SQLite**). No contexto académico, é uma opção viável, agilizando o desenvolvimento de uma aplicação. Todavia, peca nas capacidades *multi-user*, ao não suportar concorrência, e na escalabilidade.

Como somos curiosos, quisemos explorar o PythonAnywhere e constatámos que este suporta a execução de servidores MySQL, para qualquer conta, ou PostgreSQL, para contas pagas. Escolhemos o **MySQL**, não só por termos uma conta gratuita, mas também por estar na 2º posição do ranking de *DB engines*, apenas atrás da Oracle.

Ver o [ranking de DB Engines](#).

Obviamente, o Django abstrai-nos dos comandos SQL. Contudo, podemos sempre interagir com o servidor da base de dados, através do comando `python3 manage.py dbshell`.

## Créditos

---

Nº mec.	Nome
102534	Rafael Gonçalves
102536	Leonardo Almeida
102778	Pedro Rodrigues