Desafio Técnico Desenvolvedor FullStack

Grau de Dificuldade

Médio

Prazo para Entrega

Uma semana

O Problema

A organização CPBCorp necessita criar um módulo de reuniões em seu CRM para que os gestores possam ter uma visão melhor das atividades dos funcionários da empresa.

O aplicativo batizado de Meetings terá que se integrar ao cadastro de funcionários da empresa, hoje escrito em **Angular** com **Ionic** e **Django REST framework**.

Sobre a API

Os gestores irão acessar o sistema via django-admin, não será necessário implementar cadastro de usuários na interface. Além disso o cadastro de funcionários será o model de ***User*** padrão do framework e deve ser usado na aplicação para esse objetivo.

Requisitos do módulo

O model de Reunião deve conter os seguintes campos:
- Nome da reunião (*CharField*)
- Data (*DateTimeField*)
- Foto de referência (*ImageField*)
- Objetivo (*TextField*)
- Funcionários (*ManyToMany* com ***User***)

O cadastro de Reuniões será gerido pelos usuários administradores do Django Admin, com isso no cadastro de Reuniões deverá poder adicionar múltiplos Funcionários à mesma Reunião.

A aplicação Web/Mobile

A aplicação deverá ser híbrida entre web e mobile, usando o Ionic sob o Angular.
O requisito atual para essa interface é a mínima possível, contendo somente a tela de login e listagem de reuniões as quais estão associadas ao usuário atual.

Login

O login da interface web/mobile deverá acessar o django rest via Basic Auth para autenticar o funcionário.

Reuniões

Após a ação de login o funcionário deve ser redirecionado para a listagem de reuniões.
A listagem deverá conter as colunas referentes a **foto**, **nome** e **data**.
Essa listagem deve ser fornecida por um endpoint chamado */meetings/* via *GET*, que deverá ser criado no django rest trazendo as reuniões do usuário logado.

Desafios técnicos

- Usar **git** como controle de versão e gerar commits graduais ao andamento do desenvolvimento da aplicação.
- Usar o mesmo repositório para ambas aplicações (frontend e backend), separando por pastas diferentes.
- Pode-se usar SQLite como banco de dados, utilizar postgres é um **plus**.
- Será um **plus** utilizar Docker e docker-compose para organizar a aplicação em containers.

O projeto será apresentado pelo candidato(a) na própria máquina, não há necessidade de publicar o repositório no github e etc.

Boa sorte! 💚