

## **Desafio Softfocus**

### **Web Developer Python - Pleno/Sênior**

O Proagro Fácil é um sistema da Softfocus que facilita o gerenciamento de Proagro (Programa de Garantia da Atividade Agropecuária). O Proagro é um programa administrado pelo Banco Central do Brasil, que visa exonerar o produtor rural de obrigações financeiras relativas a operações de crédito, em casos de ocorrência de perdas nas lavouras. Estas perdas podem ser ocasionadas por fenômenos naturais, como chuva excessiva, geada, granizo, etc.

No Proagro Fácil, uma das principais etapas para a solicitação de Proagro é o cadastro da comunicação da perda ocorrida, onde o analista de Proagro irá informar os dados sobre o produtor rural, sobre a lavoura e sobre o evento que provocou a perda. É muito importante que essas informações sejam preenchidas corretamente para que o produtor tenha o benefício aprovado.

Neste desafio, você irá criar uma versão simplificada da comunicação de perda.

#### **Critérios essenciais**

1. A solução deve ser desenvolvida em Python (utilize o framework de sua preferência);
2. Utilizar um framework front-end (Angular, React, Vue, Ember JS, etc);
3. Criar uma interface agradável e intuitiva (utilizando frameworks CSS, tais como: Bootstrap, Materialize, Semantic UI, etc);
4. Implementar testes automatizados;
5. Fazer deploy do projeto (ex.: Heroku, Surge.sh, Pythonanywhere, AWS, etc) e disponibilizar a URL para acesso;
6. Disponibilizar a aplicação em forma de API para consulta e manipulação das comunicações de perda;
7. Disponibilizar a documentação da API (Swagger, Apiary, Document360, etc);
8. A solução deve possibilitar o cadastro, visualização, atualização e exclusão de uma comunicação de perda;
9. Os dados devem ser salvos em um dos banco de dados: Postgres, MySQL, MongoDB ou Firebase;
10. A comunicação de perda deve ter os seguintes campos:
  - a. Nome do produtor rural;
  - b. E-mail do produtor rural;
  - c. CPF do produtor rural;
  - d. Localização da lavoura (latitude e longitude);
  - e. Tipo da lavoura (milho, soja, trigo, feijão, etc);
  - f. Data da colheita;
  - g. Evento ocorrido, sendo os eventos possíveis:
    - i. CHUVA EXCESSIVA
    - ii. GEADA

- iii. GRANIZO
- iv. SECA
- v. VENDAVAL
- vi. RAIO

11. Quando o analista estiver cadastrando uma nova comunicação de perda, queremos garantir a veracidade do evento informado. Por isso, caso já exista um cadastro no banco de dados, com mesma data, cuja localização esteja em um raio de 10km da localização da nova comunicação de perda e for um evento divergente do que já consta no banco de dados, o analista deverá ser informado;
12. O projeto deverá conter validações para que o CPF e e-mail informados sejam válidos (feito em Javascript);
13. Deve ser possível realizar a busca de uma comunicação de perda pelo CPF do produtor (front-end);
14. O projeto deverá ser disponibilizado em repositório online, como Github, Gitlab, etc;
15. O repositório deve conter um arquivo README explicando como utilizar o projeto.
16. Qualquer funcionalidade extra será bem vinda.

#### **Critério opcional**

- Para cada Comunicação de Perda, criar um Mapa pinando/marcando as Coordenadas Cadastradas.