

SEGUNDA FASE PROCESSO SELETIVO LABTRANS/UFSC PARA DESENVOLVEDOR BACK-END.

A segunda fase do processo seletivo consiste em construir uma estrutura backend utilizando Python e Sqlite para receber e manipular dados de um levantamento de uma rodovia extraídos utilizando visão computacional. Não é necessário criar interface gráfica, as funções podem ser executadas direto pelo console.

Entrada

A entrada do sistema será um arquivo .csv com resultados de múltiplos vídeos. A Tabela 1 mostra a estrutura do arquivo.csv, onde cada linha corresponde a um item detectado em determinado vídeo.

name	highway	UF	item	latitude	longitude	exp_km_calc
GH010080	282	SC	Rocada	YY	XX	27.0
GH039704	349	BA	Trinca	YY	XX	938.0
GH039704	349	BA	Trinca	YY	XX	940.0
GH010080	282	SC	Placa	YY	XX	27.0
GH010080	282	SC	Rocada	YY	XX	28.0
GH010080	282	SC	Faixa Central	YY	XX	28.0
GH010080	282	SC	Faixa Lateral	YY	XX	28.0
GH010080	282	SC	Placa	YY	XX	30.0

Tal csv deve ser populado em uma tabela chamada **results**, a qual deverá possuir os mesmos campos do csv e também um **id** sequencial e único para cada resultado.

Obs.: Coordenadas em latitude e longitude em graus decimais (WGS84).

Saída

A saída do sistema também será um arquivo.csv, entretanto com os itens agrupados por km e rodovia (highway) como mostra a Tabela 2, onde cada linha corresponde a um quilômetro da rodovia. Vale ressaltar que uma rodovia pode ser levantada por um ou mais

vídeos, portanto é necessário se atentar em juntar resultados de diferentes vídeos se necessário. Deverá ser exportado um csv para cada rodovia.

highway	km	buraco	remendo	trinca	placa	drenagem
116	469.0	0	0	1	4	6
116	470.0	2	2	2	6	6
116	471.0	3	0	2	4	6
116	472.0	0	0	0	4	6
116	473.0	2	0	2	5	6
116	474.0	3	2	2	4	6

- **highway:** Rodovia selecionada para exportação dos resultados <str>;
- **km:** Quilômetro correspondente, apenas uma linha por km <double>.
- **buraco:** Número de instâncias de buraco por km <int>;
- **remendo:** Número de instâncias de remendo por km <int>;
- **trinca:** Número de instâncias de trinca por km <int>;
- **placa:** Número de instâncias de placa por km <int>;
- **drenagem:** Número de instâncias de drenagem por km <int>.

Dados disponibilizados:

- 4 levantamentos de rodovias, em um único [arquivo.csv](#).
- pub_202304A (SNV202304A), disponível em [DNIT-Cloud](#)

Prazo:

Uma semana (7 dias), a contar a partir do recebimento deste documento.

Considerações:

Caso julgue necessário, é permitido o uso de quaisquer linguagens e bibliotecas adicionais para composição do programa desde que cumpra os requisitos obrigatórios;

Requisitos Obrigatórios:

- ☐ Utilizar **Python** com **Peewee** - Valor: 1,5pts;

- ☐ Utilizar **Sqlite** - Valor: 1,5pts;
- ☐ Criar tabela **results** no banco de dados - Valor: 1,0pt;
- ☐ Exportar resultados em um csv agrupados corretamente - Valor: 1,0 pt.

Requisitos Bônus:

- ☐ Criar uma **API** com **Tornado, Django ou similar** (Neste caso, tira a obrigatoriedade da utilização do **Peewee**) - Valor: 2,0pt ;
- ☐ Utilizar **Postgres + PostGIS** - Valor: 2,0pt; (Substitui o **sqlite**)
- ☐ Criar tabelas auxiliares no banco de dados:
 - ☐ Criar tabela **vídeos** com km Ini e Km Final de cada vídeo no banco de dados - Valor: 0.5pt;
 - ☐ Criar tabela **rodovias** com km Ini e Km Final (de acordo com os km que foram levantados) de cada rodovia no banco de dados - Valor: 0.5pt;
- ☐ Criar scripts de criação automática das tabelas (init.sql) - Valor: 0.5pt;
- ☐ Criar tabelas do tipo **View** para auxiliar em queries - Valor: 0.5pt;
- ☐ Utilizar **Docker** - Valor: 1.0pt;
- ☐ Utilizar **ElectronJs**- Valor: 1.0pt;
- ☐ Criar função para calcular o KM de determinado item - Valor: 1.0pt; para isso utilize a **'latitude'** e **'longitude'** e a camada do **SNV202304A** (a qual possui os traçados e quilômetros de referências de rodovias federais do Brasil), use a coluna **'exp_km_calc'** (valor inteiro do km calculado) como referência para validar seus resultados. (Dica: `Utilize shapely.geometry.LineString().project(point,normalized=True)`)
- ☐ Permitir consulta do tipo qual Km de determinada rodovia possui maior incidência de determinado item - Valor: 1.0pt;
- ☐ Criar interface gráfica com **vueJs** - Valor: 1.0pt;
- ☐ Utilize **OpenLayers** ou **Leaflet** para mostrar os dados levantados em um mapa - Valor: 1.0pt

Entrega:

- **Vídeo:** O vídeo deverá ser auto-explicativo, ou seja, mostrando todas as funcionalidades do sistema, cumprimento dos requisitos e funcionalidades bônus (se houver), vale ressaltar que as funcionalidades não mostradas no vídeo, por mais que funcionais, não serão consideradas na pontuação final. (Sugestão: Grave a própria tela



enquanto descreve as funcionalidades verbalmente e mostre a aplicação funcionando com diferentes vídeos e rodovias.)

- Formato: Link do youtube (Não listado) ou Link para download.
- **Código fonte (incluindo o banco de dados .db e os arquivos .csv de saída)**
 - Formato: Link do github ou gitlab ou Link para download de um arquivo zip.

Avaliação:

- A pontuação final ocorrerá pela soma dos requisitos cumpridos e mostrados em vídeo;
- O não cumprimento de um requisito obrigatório acarretará em uma penalização correspondente ao valor do requisito em sua pontuação final;
- Mesmo não cumprindo os requisitos obrigatórios o candidato poderá pontuar sob os requisitos bônus, tendo em vista a penalização descrita no item anterior;