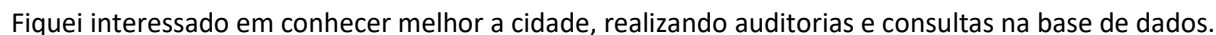


Data Source: *OpenStreetMap*



- Abreviação no logradouro (Ex.: Av. Coronel Francisco Heráclito Dos Santos)
- Logradouro sem informação se é rua ou avenida (Ex.: domingos benatto)
- Logradouro com número do endereço ou bairro (Ex.: José Maria da Silva paramos 551)
- Logradouro com erro de digitação (Ex.: rua Samanbaia -179- São Francisco -Araucária)
- CEP com formatação incorreta (Ex.: 81.200-230 ou 83020200)
- Bairros no lugar da cidade (Ex.: Batel, Rebouças ou Cajuru)
- Erro de digitação da cidade (Ex.: Curiiba, Curitba ou curitiba)

Nome das ruas

Criei um arquivo chamado “audit_street_name.py” com as funções onde realiza a auditoria do nome das ruas encontradas, na qual valida se a primeira palavra enquadra na variável “expected” do tipo lista.

```
expected = ["Rua", "Avenida", "Alameda", "Praça", "Rodovia", "Travessa", "Estrada", "Largo", "BR-116", "Linha", "Marginal", "Acesso", "Centro"]
```

Caso não se enquadre a primeira palavra do logradouro é adicionado na variável “street_types”, onde é corrigido depois com a função “update_name”.

Abaixo a função responsável pela correção dos problemas com nome dos logradouros.

```
def update_name(name, mapping):  
    name = name.split(' ')  
    for i in range(len(name)):  
        if name[i] in mapping:  
            name[i] = mapping[name[i]]  
            name[i] = string_case(name[i])  
        else:  
            name[i] = string_case(name[i])  
  
    name = ' '.join(name)  
  
    if name in mapping:  
        return mapping[name]  
  
    return name
```

Onde a variável “mapping” possui todo mapeamento do problema com as correções.

```
mapping = {"Av": "Avenida",  
          "Av.": "Avenida",  
          "R": "Rua",  
          "R.": "Rua",  
          "RUA": "Rua",  
          "Ana": "Rua Ana",  
          "Angelo Francisco Borato": "Rua Ângelo Francisco Borato",  
          "Comendador Franco": "Avenida Comendador Franco",  
          "Residencial": "Rua",  
          "Vicente Za": "Rua Carlos Vicente Zapxon",  
          "Domingos Benatto": "Rua Domingos Benatto",  
          "José Maria Da Silva Paramos 551": "Rua Domingos Jorge Velho",  
          "Rue Desembargador Motta - 2311 - Batel": "Rua Desembargador Motta",  
          "Rua Samambaia -179- São Francisco -Araucária": "Rua Samambaia"}
```

CEP

Na parte do CEP, encontrei formatos onde estavam como somente dígitos (Ex.: 82520120), com ponto e barra (Ex.: 81.200-230) ou com barra (Ex.: 80530-290).

Optei em deixar a padronização do CEP como: 5 dígitos barra 3 dígitos (Ex.: 80530-290).

Cidade

Para auditoria e correção da cidade, segui o mesmo processo do nome das ruas.

Criei uma variável chamado “expected” do tipo lista, onde contém os nomes das cidades que é esperado.

```
expected = ["Curitiba", "São José dos Pinhais", "Pinhais", "Colombo", "Campo", "Araucária", "Almirante", "Rio"]
```

Caso não se enquadre, é corrigido o nome da cidade através da variável “mapping”, onde está mapeado o problema e correção para cada problema.

```
mapping = {'Alto da Rua XV' : "Curitiba",  
          'Araucaria' : "Araucária",  
          'Batel' : "Curitiba",  
          'Cajuru' : "Curitiba",  
          'Curiiba' : "Curitiba",  
          'Curitba' : "Curitiba",  
          'Fanny' : "Curitiba",  
          'Jardim Amélia - Pinhais': "Pinhais",  
          'Rebouças' : "Curitiba",  
          'Sao José dos Pinhais': "São José dos Pinhais",  
          'São José dos Pinhais - PR': "São José dos Pinhais",  
          'São José dos Pinias': "São José dos Pinhais",  
          'Tarumã' : "Curitiba",  
          'curitiba' : "Curitiba"}
```

Overview dos dados

Informações sobre o tamanho dos arquivos que foi baixado e gerado durante o processo.

```
curitiba.osm ..... 74 MB  
curitiba.db ..... 52.9 MB  
nodes.csv ..... 27.1 MB  
nodes_tags.csv ..... 1.4 MB  
ways.csv ..... 3.3 MB  
ways_nodes.cv ..... 9.9 MB  
ways_tags.csv ..... 5 MB
```

Ideias em relação aos conjuntos de dados

Segue informações obtidas através de consultas realizadas na base de dados.

Quantidade de nodes

```
sqlite> SELECT COUNT(*) FROM nodes;
```

317890

Quantidade de ways

```
sqlite> SELECT COUNT(*) FROM ways;
```

53617

Quantidade de usuários

```
sqlite> SELECT COUNT(DISTINCT(e.uid))  
        FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
```

619

Top 5 contribuintes

```
sqlite> SELECT e.user, COUNT(*) as num  
        FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e  
        GROUP BY e.user  
        ORDER BY num DESC  
        LIMIT 5;
```

jump6024	77165
dclarks	75523
luisgustavo1968	42541
Daniel Lauer	21633
O Bicho Geográfico	13787

Top 10 pontos ou atrações

```
sqlite> SELECT value, COUNT(*) as num  
        FROM nodes_tags  
        WHERE key='amenity'  
        GROUP BY value  
        ORDER BY num DESC  
        LIMIT 10;
```

restaurant	287
school	265
fuel	192
pharmacy	184
bank	169
fast_food	132
place_of_worship	89
clinic	85
pub	79
taxi	48

Top 5 tipos de restaurante

```
sqlite> SELECT nodes_tags.value, COUNT(*) as num
        FROM nodes_tags
        JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value="restaurant") i
            ON nodes_tags.id=i.id
        WHERE nodes_tags.key="cuisine"
        GROUP BY nodes_tags.value
        ORDER BY num DESC
        LIMIT 5;
```

pizza	39
regional	38
japanese	17
steak_house	17
italian	16

Conclusão

Informações que obtive através do OpenStreetMap da cidade de Curitiba e região metropolitana são bem poucas ou com dados antigos se forem comparados com Google Maps, Waze ou Bing Maps.

Sem contar nos erros que foi encontrado no processo de auditoria, pois as informações foram inseridas manualmente por usuários que contribuem.

Sugestões para melhorias

Automatizar a inserção e validação de logradouro através do webservice dos Correios

Benefícios:

- Diminuir erro de digitação.
- Verificação que o logradouro existe de verdade.

Problemas esperados:

- Serviria somente para o Brasil.
- Caso o endereço que deseja inserir não esteja no webservice dos Correios, teríamos problemas.

Engajar usuários bonificando com pontos para subir de nível/status/ranking

Benefícios:

- Pontuação te dá um feedback em tempo real no que está contribuindo.
- Motiva a contribuir mais para subir de nível/status/ranking.

Problemas esperados:

- Chegará num ponto onde subir de nível não faz mais sentido.
- Mudança na forma de premiação.