P3: Limpando dados do OpenStreetMap

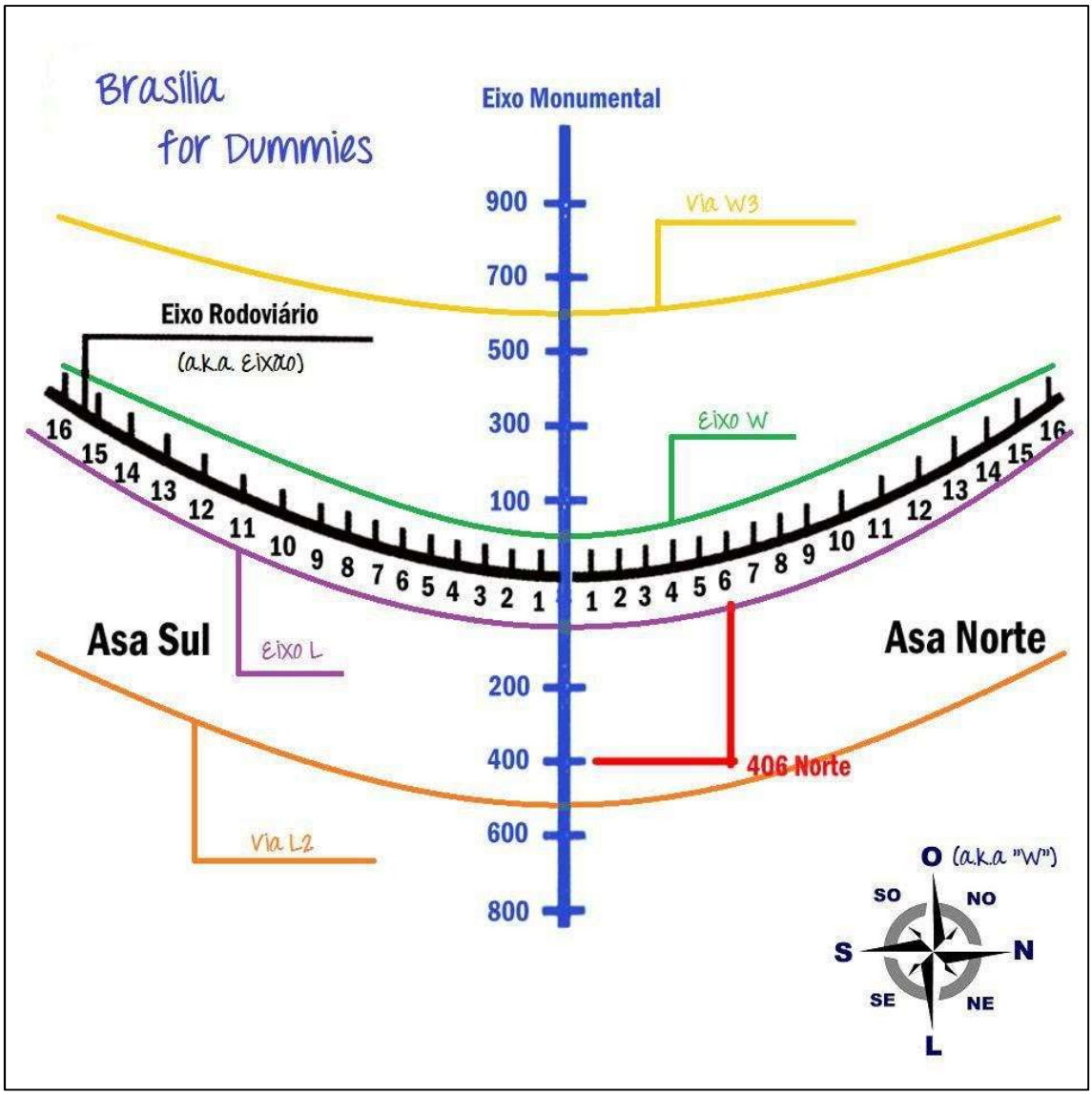
Daniel Senna Panizzo
Udacity

Área do mapa: Brasília, DF, Brasil

Há algum tempo, logo após terminar a faculdade e ingressar no mercado de trabalho, passei a dedicar meus dias de folga à um novo hobby: viajar. Conheci diversas cidades e, ao mesmo tempo em que me maravilhava com as novidades, acabei enfrentando uma dificuldade enorme de me localizar nas cidades que utilizam nomes de ruas. Como residente de Brasília, uma cidade famosa pelo seu planejamento, eu simplesmente não conseguia entender como era possível memorizar tantas ruas para que alguém se situasse em uma cidade.

Em uma cidade normal, um endereço nos indica uma posição ao longo de uma rua. Já em Brasília, um endereço indica a posição cartográfica de uma área. Se localizar por aqui é como jogar Batalha Naval, tão fácil que posso lhe ensinar agora. Veja abaixo a representação do mapa da cidade.

Agora, com uma simples imagem, vou lhe mostrar como encontrar qualquer endereço em Brasília sem nunca ter passado por aqui.



Ok, Brasília foi planejada mas não é perfeita. Não dá para realmente encontrar qualquer endereço com essa lógica, mas cobre a maior parte das necessidades de quem visita ou mora na cidade. Sabendo que a formação dos endereços daqui é diferente da maior parte das outras cidades, fiquei curioso para saber como seriam cadastradas estas informações no *OpenStreetMaps*.

PROBLEMAS ENCONTRADOS NO MAPA

Após o download dos dados da área de Brasília, processamento dos dados em Python e carga nos bancos SQL e NoSQL, podemos encontrar alguns problemas nos dados:

- Uso de siglas sem a descrição completa;
- Códigos postais não padronizados;

O uso de siglas em Brasília é muito comum, ao ponto de sabermos a sigla do local mas não sabermos o significado delas. Conhecer seu significado é crucial para entender a funcionalidade do endereço em questão, já que nesta cidade todos os setores possuem uma. Por exemplo, temos setores reservados para hotéis, hospitais, clubes e embaixadas. Algumas siglas são tão estranhas quanto SHCGN, significa "Setor de Habitações Coletivas e Geminadas Norte", que basicamente quer dizer que é uma área onde podem construir prédios ou casas residenciais.

A cultura de dar nome às ruas homenageando pessoas ilustres é inexistente em Brasília, o endereço é diretamente associado à função. É normal e aceitável que encontremos siglas no *OpenStreetMaps*, afinal, no dia-a-dia também só usamos elas. Porém, para os que não conhecem a cidade é importante oferecer um contexto do endereço, e por isso os endereços foram tratados para incluírem a descrição e sigla dos endereços.

Quanto aos códigos postais (CEPs), os cadastros foram feitos em diferentes estilos, usando pontos, espaços e hifens. Durante o tratamento dos dados, limpamos o CEP e deixamos apenas os números. Assim, podemos testar a validade dos dados e verificar que, apesar de poucos, nem todos os códigos postais iniciam com 7 (número inicial do CEP do Distrito Federal) ou possuem 8 dígitos.

	CONSULTA	RESULTADO
SQLite	SELECT key AS key, value AS value FROM nodes_tags WHERE key = 'postcode' AND value NOT LIKE '7%' UNION ALL SELECT key AS key, value AS value FROM ways_tags WHERE key = 'postcode' AND value NOT LIKE '7%' LIMIT 10	postcode,87200110
MongoDB	db.osm.find({"address.postcode": {"$not": /^7.*$/, "$exists": 1}},{"_id": 0, "address.postcode": 1})	{ "address" : { "postcode" : "87200110" } }

	CONSULTA	RESULTADO
SQLite	SELECT key AS key, value AS value FROM nodes_tags WHERE key = 'postcode' AND LENGTH(value) != 8 UNION ALL SELECT key AS key, value AS value FROM ways_tags WHERE key = 'postcode' AND LENGTH(value) != 8 LIMIT 10	postcode,73088 postcode,72231
MongoDB	db.osm.find({"address.postcode": {"$exists": 1}, "$where": "this.address.postcode.length != 8"},{"_id": 0, "address.postcode": 1})	{ "address" : { "postcode" : "73088" } } { "address" : { "postcode" : "72231" } }

Pelo menos podemos confirmar a consistência dos dados ao verificar que não há CEPs diferentes para a mesma latitude e longitude.

	CONSULTA	RESULTADO
SQLite	SELECT n.lat AS lat, n.lon AS lon, COUNT(nt.id) AS qtd FROM nodes n INNER JOIN nodes_tags nt ON nt.id = n.id WHERE nt.key = 'postcode' GROUP BY n.lat, n.lon	-

	HAVING qtd > 1;	
MongoDB	db.osm.aggregate([{"$match": {"address.postcode": {"$exists": 1}}}, {"$group": {"_id": "$pos", "postcode": {"$addToSet": "$address.postcode"}}}, {"$unwind": "$postcode"}, {"$group": {"_id": "$_id", "count": {"$sum": 1}}}, {"$match": {"count": {"$gt": 1}}}])	-

RESUMO DOS DADOS

Esta seção contém estatísticas básicas sobre o conjunto de dados e as consultas utilizadas para adquiri-las.

TAMANHO DOS ARQUIVOS

Nome do Arquivo	Tamanho do Arquivo
nodes.csv	40.600 KB
nodes_tags.csv	1.241 KB
ways.csv	6.020 KB
ways_nodes.csv	15.157 KB
ways_tags.csv	6.527 KB
brasilia_brazil.osm.json	165.016 KB

NÚMERO DE *NODES*

	CONSULTA	RESULTADO
SQLite	SELECT COUNT(*) FROM nodes;	469773
MongoDB	db.osm.find({"type": "node"}).count()	469773

NÚMERO DE *WAYS*

	CONSULTA	RESULTADO
SQLite	SELECT COUNT(*) FROM ways;	94443
MongoDB	db.osm.find({"type":"way"}).count()	94443

NÚMERO DE USUÁRIOS ÚNICOS

	CONSULTA	RESULTADO
SQLite	SELECT COUNT(DISTINCT e.uid) FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;	659
MongoDB	db.osm.distinct('created.uid').length	659

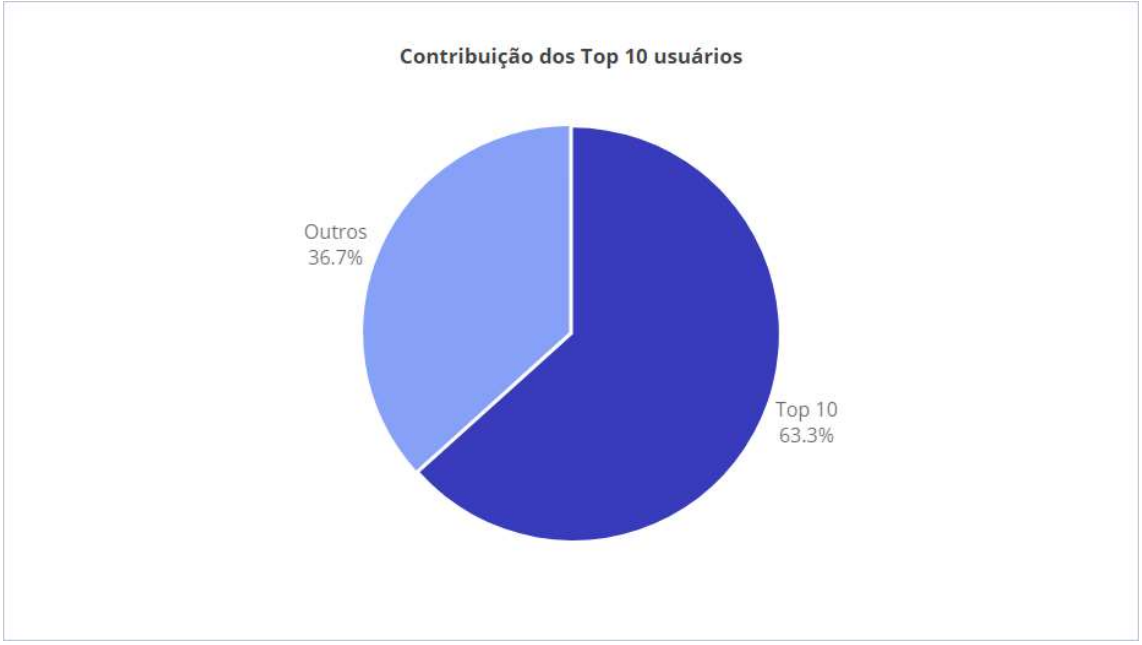
TOP 10 CONTRIBUIDORES

	CONSULTA	RESULTADO
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 10;	erickdeoliveiraleal,130873 Linhares,45897 street0501,40582 MAPconcierge,38129 teste18,29037 wille,18577 Rusleykcruz,14687 woodpeck_repair,13691

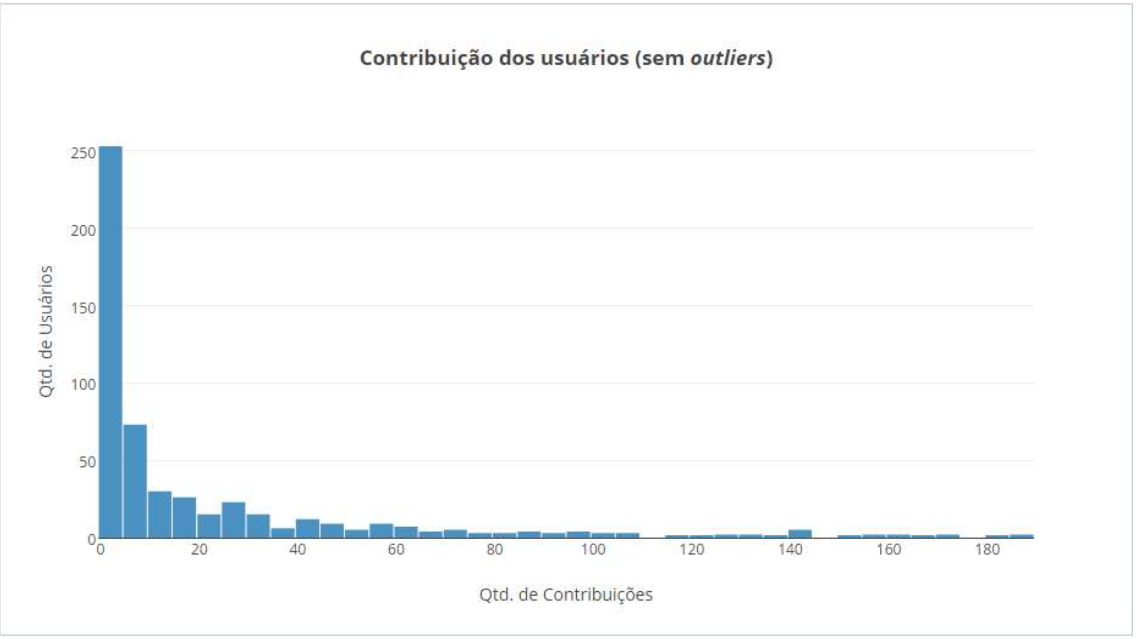
		jadson_reis,13070 charliekowacks,12754
MongoDB	db.osm.aggregate([{"$group": {"_id": "$created.user","count": {"$sum": 1}}} ,{"$sort": {"count": -1}}, {"$limit": 10}])	{ "_id" : "erickdeoliveiraleal", "count" : 130873 } { "_id" : "Linhares", "count" : 45897 } { "_id" : "street0501", "count" : 40582 } { "_id" : "MAPconcierge", "count" : 38129 } { "_id" : "teste18", "count" : 29037 } { "_id" : "wille", "count" : 18577 } { "_id" : "Rusleykcruz", "count" : 14687 } { "_id" : "woodpeck_repair", "count" : 13691 } { "_id" : "jadson_reis", "count" : 13070 } { "_id" : "charliekowacks", "count" : 12754 }

IDEIAS ADICIONAIS

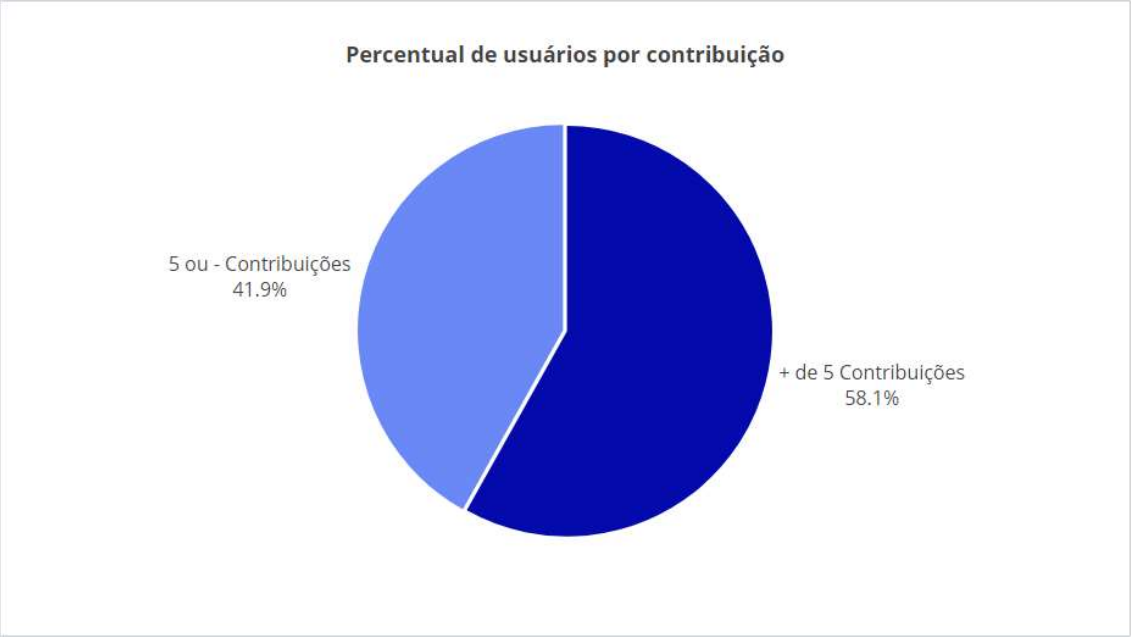
As contribuições feitas para o mapa de Brasília no OpenStreeMaps é desbalanceada, ou seja, são poucos usuários responsáveis pela maioria das contribuições. No gráfico abaixo podemos observar que cerca de dois terços de todas as contribuições foram feitas por dez usuários. Muito provavelmente por se tratarem de contribuições automatizadas, já que entre estes usuários podemos encontrar nomes como “MAPconcierge”, “teste18” e “woodpeck_repair”.



Ao analisarmos o histograma de contribuição dos usuário, comprovamos a concentração de usuários entre aqueles que fizeram entre uma e cinco contribuições.



Felizmente, mais da metade dos usuários fizeram mais do que cinco contribuições. Isto pode indicar que existe uma comunidade em crescimento de mantedores dos dados da cidade no *OpenStreetMap*.



Visto os dados acima, acredito que a comunidade do OSM gostaria de cumprir, pelo menos, dois objetivos:

1. Aumentar as contribuições de usuários únicos;
2. Garantir a qualidade dos dados que são cadastrados;

Para alavancar as contribuições dos usuários o uso de técnicas de *gamefication* ou recompensas no OSM poderia alcançar o efeito desejado. Isto envolveria o esforço de uma equipe de desenvolvedores para aplicar o novo modelo e remodelar a interface, tornando o esforço da contribuição algo intuitivo e fácil.

Já para garantir a qualidade dos dados cadastrados, poderiam ser utilizados uma coletânea de bancos de dados de endereços *gold standard*. Isso envolveria a busca e uso de dados oficias fornecidos por diferentes países, possibilitando verificar a precisão e plenitude das contribuições. Esta verificação poderia ser aplicada de acordo com a posição (latitude e longitude) da contribuição. Porém, este é um cenário complexo de ser alcançado, devido aos diferentes padrões de endereço que serão encontrados e uma provável necessidade de georreferenciamento dos dados.

EXPLORAÇÃO DE DADOS ADICIONAIS

TOP 10 TIPOS DE LOCAIS

	CONSULTA	RESULTADO
SQLite	SELECT v.value ,SUM(v.count) as count FROM (SELECT value as value ,1 as count FROM nodes_tags WHERE key='amenity' UNION ALL SELECT value as value ,1 as count FROM ways_tags WHERE key='amenity') v GROUP BY value ORDER BY count DESC LIMIT 10;	parking,1184 school,937 restaurant,756 place_of_worship,406 fuel,352 fast_food,345 bar,263 pharmacy,241 bank,215 police,201
MongoDB	db.osm.aggregate([{"$match": {"amenity": {"$exists": 1}}}, {"$group": {"_id": "$amenity", "count": {"$sum": 1}}} ,{"$sort": {"count": -1}} ,{"$limit": 10}])	{ "_id" : "parking", "count" : 1184 } { "_id" : "school", "count" : 937 } { "_id" : "restaurant", "count" : 756 } { "_id" : "place_of_worship", "count" : 406 } { "_id" : "fuel", "count" : 352 } { "_id" : "fast_food", "count" : 345 } { "_id" : "bar", "count" : 263 } { "_id" : "pharmacy", "count" : 241 } { "_id" : "bank", "count" : 215 } { "_id" : "police", "count" : 201 }

PRINCIPAL TIPO DE RESTAURANTE

	CONSULTA	RESULTADO
SQLite	SELECT nt.value ,COUNT(*) as num FROM nodes_tags nt JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value='restaurant') i ON nt.id=i.id WHERE nt.key='cuisine' GROUP BY nt.value ORDER BY num DESC LIMIT 1;	pizza,98
MongoDB	db.osm.aggregate([{"$match": {"type": "node" ,"amenity": {"$exists": 1} ,"cuisine": {"$exists": 1} ,"amenity": "restaurant"}} ,{"$group": {"_id":	{ "_id" : "pizza", "count" : 98 }

	"$cuisine" ,"count": {"$sum": 1}}} ,{"$sort": {"count": -1}} ,{"$limit": 1}])	

QUADRA COM MAIOR QUANTIDADE DE BARES

	CONSULTA	RESULTADO
SQLite	SELECT nd1.value AS addr, COUNT(nd1.id) AS qtd FROM nodes_tags nd1 WHERE nd1.type LIKE '%addr%' AND (nd1.value LIKE '%CLN%' OR nd1.value LIKE '%CLS%') AND nd1.id IN (SELECT id FROM nodes_tags WHERE key = 'amenity' AND value = 'bar' UNION SELECT id FROM ways_tags WHERE key = 'amenity' AND value = 'bar') GROUP BY nd1.value ORDER BY 2 DESC LIMIT 1;	Comércio Local Norte (CLN) 408,6
MongoDB	db.osm.aggregate([{"$match": {"address.street": {"$exists": 1, "$regex": /.*CLN\|CLS.*/}, "amenity": {"$exists": 1, "$regex": /^bar$/}}}, {"$group": {"_id": "$address.street", "count": {"$sum": 1}}}, {"$limit": 1}])	{ "_id" : "Comércio Local Norte (CLN) 408", "count" : 6 }

PIZZARIAS NAS PROXIMIDADES DA MINHA QUADRA

	CONSULTA	RESULTADO
MongoDB	db.osm.createIndex({"pos": "2dsphere"}) db.osm.find({"type": "node", "amenity": {"$exists": 1}, "cuisine": {"$exists": 1}, "cuisine": "pizza", "pos": {$nearSphere: { $geometry: { type: "Point", coordinates: [-47.8933383,-15.7451663] } , $maxDistance: 1000 }}})	{ "_id" : ObjectId("58d1cfa712fc241fa6d6c912") , "cuisine" : "pizza" , "amenity" : "restaurant" , "name" : "La Fornicella" , "created" : { "changeset" : "44934285" , "user" : "wille" , "version" : "2" , "uid" : "360183" , "timestamp" : "2017-01-05T19:53:48Z" } , "pos" : [-47.8926383, -15.750439] , "type" : "node" , "id" : "3200842313" } { "_id" : ObjectId("58d1cfa612fc241fa6d5a2cc") , "cuisine" : "pizza" , "amenity" : "restaurant" , "name" : "Valentina" , "created" : { "changeset" : "20793373"

		, "user" : "erickdeoliveiraleal" , "version" : "1" , "uid" : "463504" , "timestamp" : "2014-02-26T16:06:12Z" } , "pos" : [-47.8875238, -15.745009] , "phone" : "3340-6868" , "address" : { "street" : "Comércio Local Norte (CLN) 214 BL A LJ 9 11" , "place" : "Asa Norte" } , "type" : "node" , "id" : "2691239630" }

CONCLUSÃO

O OpenStreetMaps é uma iniciativa formidável e alinhada com os interesses crescentes das comunidades *Open Data*. Assim como em outras iniciativas *open*, as dificuldades se concentram na qualidade das contribuições e engajamento da comunidade. O engajamento, em especial, é importante para tratar exceções como Brasília, que fogem ao padrão de formação de endereços encontrado na maioria das cidades.

Uma das possíveis soluções para melhorar a qualidade dos dados inseridos no OpenStreetMaps é a atualização da interface de contribuição dos usuários, tornando-a mais intuitiva e recompensadora. O uso de dados *gold standard* de cada país também pode fazer parte desta atualização, sugerindo opções reconhecidamente oficiais aos contribuintes durante a inserção dos dados.

REFERÊNCIAS

- https://docs.google.com/document/d/1F0Vs14oNEs2idFJR3C_OPxwS6L0HPliOii-QpbmrMo4/pub
- https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md
- https://wiki.openstreetmap.org/wiki/OSM_XML
- https://wiki.openstreetmap.org/wiki/Elements
- https://docs.mongodb.com/manual/core/2dsphere/
- https://docs.mongodb.com/manual/reference/operator/query/near/#op._S_near
- https://docs.mongodb.com/manual/tutorial/geospatial-tutorial/
- http://regexr.com/
- https://www.codeschool.com/courses/breaking-the-ice-with-regular-expressions
- https://mapzen.com/data/metro-extracts/metro/brasilia_brazil/
- http://siglasbsb.alanmol.com.br/p/siglas.html
- http://www.tutorialspoint.com/sqlite/index.htm
- https://sqlite.org/lang.html
- https://stackoverflow.com/questions/29577713/string-field-value-length-in-mongodb/29578020
- https://stackoverflow.com/questions/18501064/mongodb-aggregation-counting-distinct-fields