Big Data para o Desenvolvimento Urbano Sustentável

Banco Interamericano de Desenvolvimento (BID)

Produto Final - Termo de Referência 8

Agosto de 2021

FICHA TÉCNICA

Objeto do Contrato	*Big Data* para o Desenvolvimento Urbano Sustentável
Data de Assinatura do Contrato	02/03/2018
Prazo de Execução	36 (trinta e seis) meses
Contratante	Banco Interamericano de Desenvolvimento - BID
Contratada	Fundação Getulio Vargas

Sumário

APRESENTAÇÃO

O presente documento corresponde ao **Produto final** referente ao **Termo de Referência 8 (TR 8) - Definição Técnica, Desenvolvimento, Prova e Implementação do Ambiente de Big Data (Repositório Único), incluindo a Aquisição de Servidores e Equipamentos, e Desenvolvimento de Ferramentas de Análise e Visualização para os Níveis Gerenciais e Operacionais**, da **Cooperação Técnica Regional Não Reembolsável nº ATN/OC-16463-RG** celebrada entre o **Banco Interamericano de Desenvolvimento (BID)** e a **Fundação Getulio Vargas**, objetivando o desenvolvimento do projeto ***Big Data* para o Desenvolvimento Urbano Sustentável**.

1. Introdução

A análise e o tratamento de grandes volumes de dados (ou *Big Data*) estão pressionando formuladores de políticas públicas urbanas pela incorporação de tecnologias cada vez mais avançadas no seu modelo lógico. Sistemas de monitoramento e predição em tempo real baseados em análise de dados não convencionais podem permitir que as cidades entendam a incidência de fenômenos urbanos com detalhes e precisão crescentes. Com a grande penetração de novas tecnologias e a expansão dos modelos *Smart Cities*, esta é uma tendência irreversível.

Embora a tecnologia tenha proporcionado a geração de quantidades massivas de dados, muitas vezes, a infraestrutura das cidades tem baixa capacidade de transformar os dados gerados em informações relevantes. Perde-se, assim, a oportunidade de solucionar problemas públicos que se aglutinam e de gerar desenvolvimento social e econômico por meio de políticas urbanas inteligentes. Por outro lado, empresas igualmente pressionadas por conquistar mercados no atual contexto de transformação digital acentuada, constroem produtos que não têm sido capazes de tornar as cidades mais inteligentes.

É nesse contexto que se insere o projeto ***Big Data* para o Desenvolvimento Urbano Sustentável** que tem como objetivo desenvolver um modelo piloto, replicável e escalável de governança e exploração de *big data* para as cidades de São Paulo (Brasil), Miraflores (Peru), Montevidéu (Uruguai), Quito (Equador) e Xalapa (México). O projeto está dividido em três componentes: **Componente I - Mapeamento do Ecossistema de Dados e Marcos Regulatórios**; **Componente II - Desenvolvimento de Recursos Legais e Institucionais para o Acesso e Análise de *Big Data***; e **Componente III - Desenvolvimento de Ferramentas de Análise e Visualização de Dados Massivos e Disseminação do Projeto**. Para tal, o projeto visa:

- Conhecer o ecossistema de dados existente e os marcos regulatórios em matéria de governança de dados para o acesso e a utilização nas cidades do projeto;
- Desenvolver ferramentas comuns de âmbito legal, técnico e institucional para acessar, integrar e analisar os dados;
- Testar modelos de governança a partir do desenvolvimento de projetos com foco em mudanças do clima e mobilidade em cinco cidades da região;
- Criar unidades de análise de *Big Data* em cidades da região; e

- Consolidar o conhecimento gerado de maneira a facilitar sua replicação e disseminação em nível regional.

Dentro do contexto do projeto, o **TR 8** é parte do **Componente III** e irá dar o apoio técnico para o desenvolvimento de um ambiente de dados para monitoramento e qualificação de serviços públicos nas cidades de São Paulo, Montevidéu, Quito, Xalapa e Miraflores e tem como objetivo a criação e o desenvolvimento do protótipo de um ecossistema tecnológico especializado que seja capaz de tratar e transformar *big data* em desenho de políticas públicas urbanas baseado em evidências. A componente de políticas públicas é chave nesse desenvolvimento e precisa ser amplamente documentada durante o processo.

Além desta introdução, este relatório é composto pelo Capítulo 2, onde é discutido brevemente as etapas anteriores em que foram discutidas quais seriam os protótipos desenvolvidos neste TR. o Capítulo 3 apresenta as necessidades de hardware e software para permitir que as cidades coloquem em produção os protótipos a serem desenvolvidos. O produto discute o mínimo necessário para que se implemente uma política de inovação aberta pensando nos protótipos específicos desenvolvidos, mas também para o suporte em geral dessa política pública. O Capítulo 4 que se divide em quatro seções e apresenta os protótipos desenvolvidos, a saber, (i) Plataforma de Transparência, (ii) Estimação de Poluição do Ar, (iii) Apoio ao Transporte Público (GTFS e Comparação de velocidades de carros e ônibus) e ; (iv) Apoio à Gestão do Trânsito. Cabe ressaltar que os produtos refletem o desenvolvimento da tecnologia, bem como decisões tomadas em conjunto com as cidades no decorrer do período compreendido do projeto, de modo que as atividades realizadas foram pertinentemente adaptadas e executadas a fim de atingir o objetivo geral do projeto. Outro ponto importante é que o desenvolvimento dos protótipos se deu através de um processo de construção em conjunto com as cidades e buscaram refletir as necessidades e a disponibilidade de dados e informações de cada uma delas. Por fim, o Capítulo 5 relata os principais aspectos da Capacitação realizada após o desenvolvimento dos protótipos.

2. Antecedentes

Neste item, serão apresentados os passos realizados com vistas à definição das pesquisas, a serem desenvolvidas no âmbito deste **Termo de Referência 8**, e os resultados alcançados.

2.1 Sessões de *Design Thinking*

Na ocasião do **II Encontro Regional** do projeto ***Big Data* para o Desenvolvimento Urbano Sustentável**, realizado na cidade de Miraflores, no Peru, em 4 e 5 de dezembro de 2019, foram realizadas sessões de *design thinking*, com a participação ativa dos representantes das cinco cidades que integram o projeto. Os resultados alcançados serviram de base para a definição das pesquisas a serem desenvolvidas no **TR 8**.

Nessas sessões, os participantes das cidades foram incentivados a interagir entre si e com os especialistas que conduziram o processo, de modo a mapear os problemas que podem ser corrigidos por políticas públicas baseadas nos dados disponíveis. Estando o participante no centro do problema, promove-se empatia e experimentação, combinando análise e intuição.

Para o desenvolvimento dos trabalhos, os representantes foram divididos em grupos, preferencialmente com pessoas de diferentes cidades, promovendo maior aproximação entre elas e a discussão de ideias diferentes trazidas à luz de realidades distintas. Objetivou-se a busca de uma solução comum a ser apresentada pelo grupo para todos os presentes.

A pergunta principal proposta na primeira etapa das sessões de *design thinking* foi “Como os dados mapeados podem promover desenvolvimento local a problemas comuns nas cinco cidades?”. Por sua vez, na segunda etapa, foi apresentada uma pergunta mais voltada para as áreas de mobilidade e meio ambiente, dada a quantidade de informações disponíveis nessas áreas pelas cidades.

Deste modo, eles puderam especificar suas demandas em termos de produtos que esperam receber como resultado do projeto. Esse trabalho forneceu elementos que possibilitaram a definição conjunta das pesquisas a serem realizadas no contexto deste termo de referência, atingindo o objetivo das sessões de *design thinking*.

2.2 Videoconferência com as Cidades

Após a análise de todo o conteúdo levantado, em 18 de fevereiro de 2020, foi realizada uma videoconferência por Skype com os representantes das cidades que integram o projeto. Além do Comitê Diretivo (CD), também foram convidados os participantes do evento em Miraflores. Essa reunião teve como objetivo apresentar os resultados das sessões de *design thinking*, ocorridas no **II Encontro Regional**. A Ata de Reunião, em espanhol, foi encaminhada por meio de correio eletrônico, em 14 de março de 2020, para os integrantes das cidades de Miraflores, Montevidéu, Quito, São Paulo e Xalapa, e encontra-se no **Anexo** deste relatório.

Os resultados da discussão, em que todos os presentes tiveram a oportunidade de contribuir, foram as criações de propostas para Políticas de Transparência, prioritária na agenda dos governos utilizando a plataforma CKAN; de Planejamento Urbano (criação de arquitetura e engenharia de modelos padronizados e condensados em painéis para oferecer subsídios para o planejamento e gerenciamento de transporte público e uso de informações em tempo real para auxiliar a gestão pública a tomar decisões operacionais no campo da mobilidade urbana); além de uma Política de Capacitação.

2.3 Resultados Obtidos

Muitas vezes, tecnologias são desenvolvidas para serem utilizadas pelos gestores públicos descontextualizadas das realidades locais e incoerentes com a lógica da política pública que requer um nexo causal bem estruturado entre insumos, ações e resultados. Para contornar essa falha, é necessária a construção de tais tecnologias no contexto de sistemas inovativos que requerem a interação da academia científica, modelando hipóteses para que o setor produtivo tecnológico desenvolva soluções em parceria sinérgica com um governo orientado pela missão de dissipar os problemas urbanos com ferramentas tecnológicas.

Em trabalho conjunto com as cidades, por meio de exercício de *design thinking* com os representantes e da discussão na videoconferência, foram debatidos os problemas e levantadas e exploradas algumas possibilidades de aplicação de inovações na gestão urbana que buscassem resolvê-las. Além disso, nas etapas anteriores deste projeto foram mapeados e qualificados os dados abertos existentes nas cidades, analisados seus modelos regulatórios, apresentadas recomendações para publicação de dados e para geração de valor público e construída uma plataforma para armazenamento padronizado.

Dessas informações resultaram as possibilidades de protótipos que estão sendo construídos e viabilizados neste **TR 8**. A ideia geral é que seja desenvolvida a partir de ferramentas de *big data* e *machine learning* uma plataforma que seja capaz de prestar serviços dentro do conceito de inovação aberta que sejam úteis para as cidades. A partir da proposta co-desenhada com as cidades, ficou definido que seriam prototipadas quatro aplicações que compõem esse produto:

- **Produto 4.1 - Plataforma de Transparência**

  Será criada uma plataforma a partir dos dados que já estão em CKAN (vide Termo de Referência 6 (TR 6) - Desenvolvimento da Ferramenta de Ambiente Online de Registro, Validação, Disposição de Dados (AOCD) para o Projeto Big Data para o Desenvolvimento Urbano Sustentável), adicionando algumas usabilidades à base como a capacidade de visualização e tabulação desses dados de maneira simples pelos cidadãos e a possibilidade de as cidades adicionarem informações de indicadores e de desenvolvimento de projetos urbanos de interesse público.

- **Produto 4.2 - Estimativa de Poluição do Ar**

  A poluição do ar é um problema recorrente nas grandes cidades do mundo com consequências sobre doenças respiratórias, diminuição da expectativa de vida entre outras. Usando uma vez mais o *feed* colaborativo do Waze em combinação com os dados do Open Street Maps, plataforma de dados abertos, é possível montar um modelo de emissões de poluentes para as cidades. Para as cidades onde há dados de emissão através de sensores, será possível implementar aprendizado de máquina para aprimorar os modelos preditivos em áreas maiores tipicamente cobertas pelos sensores.

- **Produto 4.3 - Apoio ao Transporte Público**

  Nesse produto, o potencial de se explorar os dados de transporte público como uma ferramenta de inovação aberta será explorado. A partir da implementação de um gerador de GTFS participativo, os códigos abertos disponibilizados por diversas instituições serão adaptados para os municípios que ainda não contam com dados padronizados para os trajetos de ônibus da sua cidade. Para as cidades que possuem

a posição dos ônibus em tempo real, será possível criar algumas visualizações de comparações de tempo de deslocamento por ônibus e de carros.

- **Produto 4.4 - Apoio à gestão de trânsito**

  Nesse tema, são trabalhados dois painéis. O primeiro pretende aproveitar o *feed* colaborativo do Waze para gerar um alerta de acidentes para as cidades, através de um indicador de mudança abrupta na velocidade dos veículos e do próprio *feed* do Waze, que apresenta também dados de acidentes, dependendo de informações dos usuários do aplicativo. Essas bases poderiam auxiliar na coleta de informações de acidentes, tais quais suas características, visualização em tempo real e geração de alertas para agentes de trânsito e saúde se deslocarem mais rapidamente ao local.

  O segundo pretende criar indicadores de congestionamento para as cidades a partir dos dados do Waze. Essa plataforma tem o potencial de dar agilidade e aumentar a área atualmente monitorada de trânsito.

Por fim, o **Produto 5** deste termo de referência prevê a fase de capacitação da equipe. O produto faz um compilados das informações instrucionais apresentados nos produtos anteriores e entrega um manual de uso e implementação de todos os aplicativos gerados nos produtos supramencionados e realiza uma série de webinários para capacitar os gestores públicos no uso das plataformas e, também, no uso de inovação aberta para o setor público. Os webinários pretendem utilizar os protótipos desenvolvidos mais como um exemplo do potencial da inovação aberta para a melhoria no fornecimento de serviços do setor público do que como um fim em si mesmo.

3. Infraestrutura de Captação e Armazenamento de Dados, Estrutura de Carga de dados e ETL

3.1 Conceitos e Definições

Big Data, ou “Megadados” (em português), em tecnologia da informação, é o termo utilizado para se referir a um grande volume de dados armazenados, que são gerados em alta velocidade e variedade. Essas informações necessitam de ferramentas não convencionais para o seu armazenamento e processamento.

Na era atual da informação, quando se fala em grandes volumes de dados, a referência é de valores da ordem de Tera, Peta, Exa bytes ou até mais. Somente quando novos dados são gerados com uma periodicidade muito alta, diariamente ou semanalmente, por exemplo, convém necessariamente a definição de *Big Data*.

3.1.1 O Tempo Real

Quando se ouve o termo “tempo real”, o senso comum leva à visão de um sistema que mostra resultados de forma imediata (como em milésimos de segundo), mas na computação este termo tem um significado mais abrangente. Um sistema de tempo real se refere a uma tarefa, ou a um conjunto delas, que possuem um tempo de execução rígido independente da carga do sistema.

Este tempo pode ser muito curto ou não, isso dependerá dos requisitos do projeto. O sistema deverá cumprir a tarefa no prazo ou informar imediatamente que não poderá fazê-la. Sistemas de *trading* têm um requisito de milésimos de segundo para computar uma transação. Em um sistema de notificação de acidentes de trânsito, o requisito de tempo pode ser 10 minutos, por exemplo.

3.1.2 Computação na Nuvem e Infraestrutura acerca do Volume de Dados

O termo “nuvem” faz referência à internet. Porque é mais simples imaginá-la como uma nuvem, do que como uma emaranhada rede de conexões, máquinas e dados. Resumindo, a computação em nuvem inclui servidores, armazenamento, banco de dados, rede, software, análise e

inteligência, tudo como um serviço contratável sob demanda. Este tema é muito presente quando se trata de *Big Data*, porque é muito mais simples escalar um negócio quando ele reside na "nuvem".

De acordo com o Instituto Gartner (2019), estima-se que é possível que haja um total de 40 trilhões de gigabytes de dados no mundo. Isso significa 2,2 milhões de Terabytes de novos dados gerados todos os dias. Às vezes é difícil ter uma ideia do que isso significa, mas se fosse guardada toda essa informação que é gerada em um único dia em CDs empilhados, ter-se-ia o equivalente a duas vezes a distância da Terra até a Lua.

Mas, simplificando, imagine que sua solução vá gerar 1 Terabyte de dados por dia, o que é bem comum. Dados de trânsito e mobilidade costumam gerar uma quantidade de dados dessa ordem. Hoje em dia é comum encontrar discos rígidos de 1 Terabyte nas prateleiras. Se você escolhesse montar seu próprio data center, precisaria adquirir constantemente um novo equipamento para dar conta da quantidade de dados.

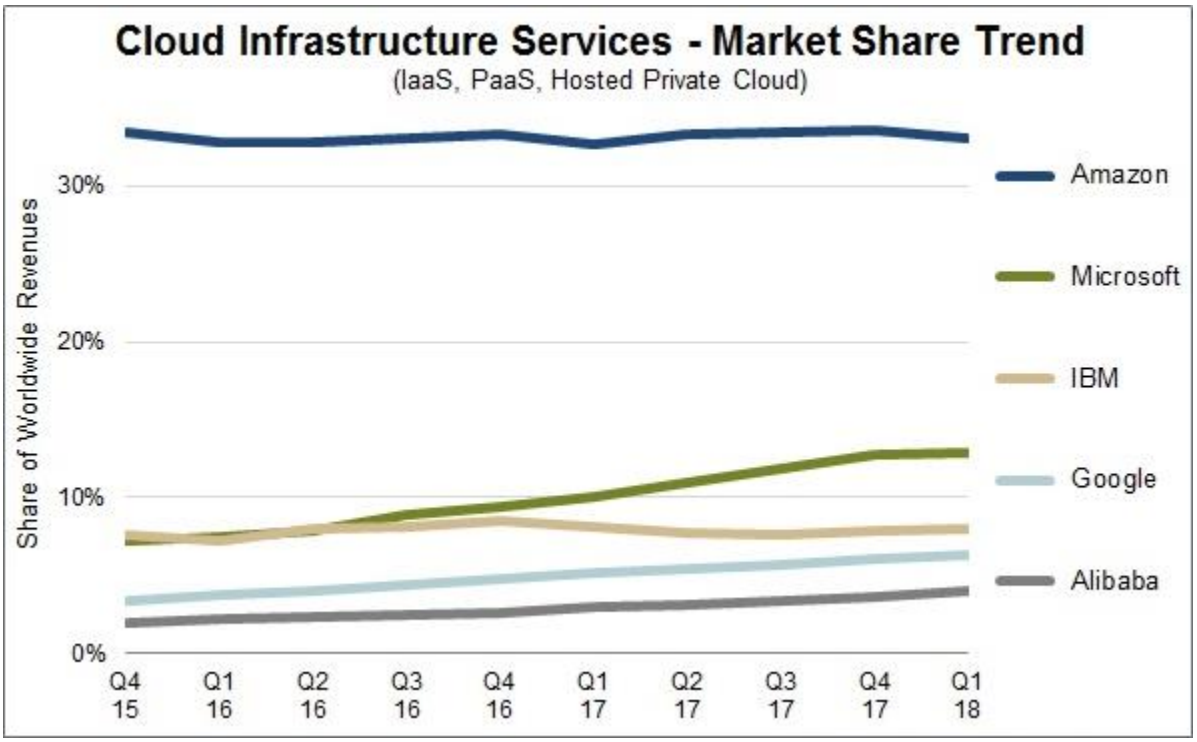
Existem soluções de armazenamento na nuvem que são flexíveis nesse sentido. Não é preciso se preocupar com o volume de dados, pois esses sistemas são automaticamente escaláveis à sua demanda por novos espaços de armazenamento. O mesmo se aplica à questão do processamento. Se um super servidor fosse adquirido com o hardware mais potente disponível e fosse utilizado para processar os dados uma vez na semana, o retorno financeiro da sua solução não ultrapassaria os seus gastos necessários com a infraestrutura. Existem soluções mais viáveis economicamente que trabalham com o conceito de máquinas virtuais alugáveis que ajustam a demanda de acordo com o processamento necessário, pagando somente pelo tempo de uso.

Com o crescimento da popularidade dos termos *Big Data* e *Cloud Computing* (Computação na Nuvem), os gigantes tecnológicos como Google, Microsoft e Amazon enxergaram excelentes oportunidades de negócio. A Google possui a Google Cloud Platform, a Microsoft possui a Microsoft Azure e a Amazon possui a Amazon Web Services (ou AWS).

Para este documento serão apresentadas especificamente as soluções apresentadas pela Amazon na AWS. Essa escolha deve-se à dominância da empresa no mercado de *Cloud Computing*. Um estudo do Synergy Research Group (2018) mostrou que a AWS teve participação

de 33% das receitas deste mercado entre 2015 e 2018, mesmo com esse tipo de serviço quase triplicando de tamanho no mesmo período.

Figura 3.1.2.1 - Gráfico Relacionando Porcentagem de Receitas dos Serviços de Cloud no Mundo



Fonte: Synergy Research Group

Outro estudo mais recente, realizado pela Canalys Cloud Channels Analysis (2020), reportou que em 2019 a AWS compete com 32,4% do mercado, Azure com 17,6%, Google Cloud com 6%, Alibaba Cloud com 5,4% e outras soluções de *cloud* com 38,5%.

Embora exista uma clara preferência para o desenvolvimento do projeto pelas soluções da AWS, também é possível que as cidades utilizem as soluções da Google Cloud Platform, Microsoft Azure e afins para implementar os projetos que são descritos aqui, pois existe uma equivalência entre os serviços disponibilizados por elas e é possível alternar entre as soluções por um baixo custo de transição. As cidades podem avaliar quais as soluções são mais convenientes para si, tendo, portanto, autonomia para utilizarem outras opções.

Amazon Simple Storage Service (Amazon S3)

"O S3 é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade de dados, segurança e performance. Isso significa que clientes de todos os tamanhos e setores podem usá-lo para armazenar qualquer volume de dados em uma grande variedade."[1]

Considerar o S3 é interessante quando faz sentido para o seu problema armazenar os arquivos em formato de texto, e/ou até JSON e CSV (compactados ou não). Existem soluções, como Amazon Athena e Amazon Glue, que trabalham diretamente com o S3 para analisar e processar os dados direto no seu formato de arquivo. Porque no *Big Data* é importante que o processamento seja levado até o dado e não o contrário.

Amazon Athena

"O Amazon Athena é um serviço de consultas interativas que facilita a análise de dados no Amazon S3 usando SQL padrão. O Athena não precisa de servidor. Portanto, não há infraestrutura para gerenciar e você paga apenas pelas consultas executadas. Basta apontar para os dados no Amazon S3, definir o *schema* e iniciar as consultas usando SQL padrão. Com o Athena, não há necessidade de trabalhos complexos de ETL para preparar dados para análise. Isso permite que qualquer pessoa com experiência em SQL analise conjuntos de dados em grande escala com facilidade e rapidez."[2]

As soluções que tratam do processamento de grandes volumes de dados são como "facas de dois gumes". Imagine a estrutura para processar 1 Terabyte de planilhas de dados em 10 minutos. Para testar um arquivo de somente uma linha, será preciso esperar ainda os mesmos 10 minutos para conseguir uma resposta. Isso acontece porque esses sistemas foram pensados de forma que seja indiferente processar uma linha ou um milhão de linhas. Geralmente, estudos com *Big Data* têm prazos de um dia, uma semana ou um mês para que as análises de uma determinada população sejam entregues, então esperar alguns minutos ou até horas para obter os resultados é algo comum no *Big Data*.

[1] https://aws.amazon.com/pt/s3/

[2] https://aws.amazon.com/pt/athena/

Amazon Kinesis

“O Amazon Kinesis facilita a coleta, o processamento e a análise de dados de *streaming* em tempo real, permitindo que você obtenha *insights* oportunos e reaja rapidamente às novas informações. O Amazon Kinesis oferece recursos essenciais para processar dados de *streaming* em qualquer escala de forma econômica, além da flexibilidade de escolher as ferramentas mais adequadas aos requisitos dos aplicativos. Com o Amazon Kinesis, você pode consumir dados em tempo real como vídeo, áudio, *logs* de aplicativos, *clickstreams* de sites e dados de telemetria de IoT para *machine learning*, análises e outros aplicativos. O Amazon Kinesis permite processar e analisar dados assim que são recebidos e responder instantaneamente, em vez de aguardar a conclusão da coleta de dados para poder iniciar o processamento.” [3]

A AWS também possui soluções que se encaixam quase que perfeitamente no universo do *Big Data*. O Amazon Kinesis é uma delas. Ela é uma solução para coleta e processamento de dados em tempo real. Ela é feita para se escalar automaticamente com a sua demanda por novos dados. Um dos grandes requisitos para processar quantidades massivas de dados e obter respostas em tempo real é o processamento em paralelo. É fortemente necessário que o sistema seja capaz de trabalhar paralelamente com a geração de dados.

3.1.3 Estatística Aplicada ao Big Data

Quando se quer fazer perguntas aos dados, é importante decidir o quão certos se quer estar sobre a resposta. Imagine que se queira calcular a porcentagem de homens e mulheres de uma base de dados que armazena os dados de consumo de vários usuários. Uma inteligência artificial foi criada especialmente para analisar o padrão de consumo e inferir se aquele consumidor é homem ou mulher.

Como um exemplo, é possível analisar uma base com 488 milhões de consumidores e encontrar o percentual de 32,05% de homens e 19,48% de mulheres em 12 horas de processamento. As porcentagens não somam 100%, pois algoritmos de inteligência artificial também podem chegar

[3]https://aws.amazon.com/pt/kinesis/

a resultados inconclusivos sem ter certeza se o consumidor analisado é um homem ou uma mulher.

A questão é que, dada esta população de 488 milhões de consumidores, se for optado por ter 99% de certeza, é possível analisar apenas uma amostra aleatória de 16 mil consumidores e chegar a um resultado aproximado de 32,36% de homens e 19,66% de mulheres, o que é aceitável. A questão é que a incerteza de 1% pode poupar 12 horas de processamento pesado de *Big Data*, em troca de uma solução mais simples que dá o resultado aceitável em alguns minutos.

3.1.4 Veracidade e Valor no Big Data

No *Big Data*, os dados precisam ser verídicos, ou seja, não podem advir de opiniões, e sim de fatos. Nesse universo, pode-se definir um fato como sendo um dado gerado por uma fonte de informação confiável, como um sensor, uma imagem etc. Um GPS, por exemplo, emite as coordenadas de latitude e longitude de um determinado objeto, e isso é fato, não é pesquisa de opinião.

É claro que numa aplicação de geolocalização é bem comum que 10% das respostas de um sensor de coordenadas seja latitude zero e longitude zero e, a não ser que seu sensor esteja no meio do mar ao sul da África, essa é uma resposta que você não vai querer pesar em suas análises. Para isso, também é importante que haja uma inteligência para tratamentos de erro dos dados gerados. Muitas dessas soluções utilizam inteligência artificial para traçar volumes que delimitam dados verídicos e dados não-verídicos.

Por fim, quando se refere ao Valor no *Big Data*, algumas citações explicam claramente: quanto maior a riqueza de dados, mais importante é saber realizar as perguntas certas no início de todo processo de análise (Brown, Eric, 2014).

É necessário estar focado para a orientação do negócio, o valor que a coleta e análise dos dados trará para o negócio. Não é viável realizar todo o processo de *Big Data* se não se tem questionamentos que ajudem o negócio de modo realístico. Da mesma forma, é importante estar atento aos custos envolvidos nessa operação, o valor agregado de todo esse trabalho desenvolvido, coleta, armazenamento e análise de todos esses dados têm que compensar os custos financeiros envolvidos (Taurion, 2013).

3.2 Requisitos para Estrutura de Arquitetura e Dimensionamento de Custos

Cobertos os termos, conceitos e algumas soluções básicas para se trabalhar com Big Data, um aprofundamento nos requisitos do projeto para estruturar uma arquitetura e dimensionar seus custos é o próximo passo.

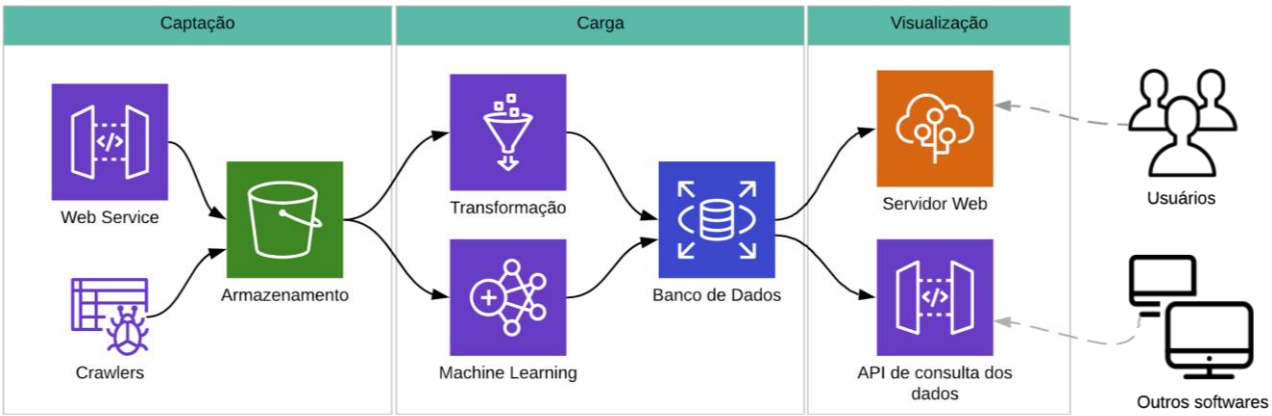
Para entender os requisitos no sistema, primeiro deve-se visualizar as metas supramencionadas definidas pelo projeto:

- A criação de uma **plataforma de transparência** que possibilitará a visualização e tabulação desses dados de maneira simples pelos cidadãos e a possibilidade de as cidades adicionarem dados sobre o desenvolvimento de projetos urbanos, permitindo aos cidadãos acompanhar o andamento do projeto;
- Um sistema para **estimar a poluição do ar**, usando os dados colaborativos do Waze em combinação os dados do Open Street Maps;
- Um sistema de apoio ao transporte público, com a implementação de um **gerador de GTFS participativo**:
  - Os códigos abertos disponibilizados por diversas instituições serão adaptados para as cidades que ainda não contam com dados padronizados para os trajetos de ônibus da sua cidade; e
  - Para as cidades que possuem a posição dos ônibus em tempo real, será possível criar algumas visualizações de comparações de tempo de deslocamento por ônibus e de carros.
- Apoiar a **gestão do trânsito** por meio de:
  - Um sistema para identificação de acidentes utilizando dados disponibilizados pelo Waze, indicadores de mudança abrupta de velocidade e também as informações disponibilizadas pelos usuários.
  - O desenvolvimento dos índices de congestionamento para apoio à gestão do trânsito incorre em disponibilidade de dados do Waze para as vias que se desejam monitorar

3.2.1 Arquitetura do Projeto

Deve-se separar o projeto em três grandes partes, para que seja possível decidir melhor como cada um dos processos facilitará o trabalho da etapa seguinte. Veja na Figura 3.2.1.1, como ocorre essa separação de tarefas. O processo de captação busca os dados puros e os armazena. O processo de carga transforma e aprende com os dados, armazenando-os de forma estruturada. A visualização exibe os dados e os deixa acessíveis para a população.

Figura 3.2.1.1 – Processo de Captação, Carga e Visualização.



Fonte: Elaboração Própria.

Captação

Para exemplificar melhor, na Captação define-se como acessar os dados puros e armazená-los para o uso no passo seguinte. Se os seus dados já estão facilmente acessíveis, esta etapa se torna desnecessária, e seria possível implementar uma solução de Carga diretamente.

Outro caso de uso é quando não existe a possibilidade de capturar os dados para sua solução utilizando uma integração elegante com API Gateways, Webservices ou Webhooks. Nos casos em que os dados estão disponíveis na internet; às vezes em sites, links, arquivos e imagens. Pode-se automatizar o acesso a esses dados através de Web Crawlers. Essas ferramentas de software funcionam como robôs capazes de navegar na internet e extrair qualquer tipo de informação contida no texto e enviá-la para um local de processamento. A Google utiliza crawlers para manter toda a sua indexação de páginas atualizada. Scrapy é uma biblioteca famosa, escrita

em Python e que facilita a criação de Crawlers. Neste documento, será estudada especificamente esta etapa.

Carga

A única diferença entre a implementação da Captação para a Carga é que, na Captação, a preocupação é apenas em obter os dados puros de alguma forma e armazená-los exatamente como estão. Na Carga, já se preocupa em transformar os dados. Nessa etapa também se integram algoritmos de *machine learning* com o propósito de estruturar o acesso à informação para a etapa de visualização.

Visualização

Quando chega essa etapa, os dados já estão em uma base de dados que pode ser facilmente integrada com alguma aplicação web para mostrar gráficos, mapas, estudos, etc. Também é possível criar APIs de consulta de dados para que outros softwares possam fazer análises nos dados apresentados.

3.2.2 Estimativa de Preço

Nesta seção do documento, realiza-se a estimativa de custo relativas somente à etapa descrita como “Captação”. Os cálculos também serão realizados como um exemplo num número limitado de requisições, todavia os custos da solução em cenários reais podem escalar para 10 vezes mais o preço estimado.

Foram selecionados alguns serviços da Amazon Web Services que irão compor uma versão piloto do produto. O primeiro deles é o Amazon API Gateway, que é um serviço para criação de API Rests. Esse serviço atuará como a porta de entrada dos dados em uma possível integração via Web Service. O nível gratuito do Amazon API Gateway inclui um milhão de chamadas de API recebidas para as APIs REST por mês durante até 12 meses. Se o número de chamadas por mês for excedido, haverá cobrança de acordo com as taxas de uso do API Gateway.

Tabela 3.2.2.1 - Amazon API Gateway - Estimativa de Custo

Amazon API Gateway	
Número de solicitações (por mês)	**Preço (por milhão)**
Próximos 667 milhões	4,25 USD
Próximos 19 bilhões	3,61 USD
Mais de 20 bilhões	2,29 USD

Fonte: Elaboração Própria a partir de Dados da Amazon

O segundo serviço é o Amazon Kinesis Data Stream, discutido anteriormente. No Amazon Kinesis Data Stream, o preço é definido a partir da Unidade de Carga PUT (que equivale a 25 KB). Por exemplo, um registro de 5 KB contém uma Unidade de Carga PUT, um registro de 45 KB contém duas Unidades de Carga PUT e um registro de 1 MB contém 40 Unidades de Carga PUT. A unidade de carga PUT é cobrada com base em uma taxa por milhão de unidades de carga PUT.

Outra unidade utilizada é o estilhaço, uma unidade básica de *throughput* de um *stream* de dados do Amazon Kinesis. Especifica-se o número de fragmentos necessários no *stream* de acordo com os requisitos de taxa de transferência. Para cada fragmento, é cobrada uma taxa horária.

Tabela 3.2.2.2 - Amazon Kinesis Data Stream - Estimativa de Custo

Amazon Kinesis Data Stream	
Unidades de carga PUT, por 1.000.000 de unidades	0,028 USD
Hora de estilhaço (entrada 1 MB/segundo, saída 2 MB/segundo)	0,03 USD

Fonte: Elaboração Própria a partir de Dados da Amazon

Por fim, o Amazon S3 cobra pelo armazenamento e pela transferência dos dados. O ato de inserir os dados não contabiliza custos de transferência, mas quando se direciona esses dados armazenados para um ETL, haverá uma taxa cobrada pela transferência.

Tabela 3.2.2.3 - Amazon S3 - Standard (Armazenamento Padrão) - Estimativa de Custo

Amazon S3 - Standard (Armazenamento Padrão)	
Primeiros 50 TB/mês	0,0405 USD por GB
Próximos 450 TB/mês	0,039 USD por GB
Mais de 500 TB/mês	0,037 USD por GB

Fonte: Elaboração Própria a partir de Dados da Amazon

Será preciso estimar alguns valores para calcular o custo mensal da etapa de captação. Para isto, assume-se que a aplicação receberá 1.000 (mil) eventos por minuto e que cada evento terá em média 10 Kilobytes de dados.

Tabela 3.2.2.4 - Comparativo de Custos

Amazon API Gateway	1.000 chamadas/minuto * 60 minutos/hora * 730 horas/mês = 43.000.000 chamadas/mês * 4,25 USD/milhão = **= 186,15 USD / mês**
Amazon Kinesis Data Stream	1.000 registros/minuto / (60 segundos/minuto) = = 16.7 registros /segundo * 2628000 segundos/mês = 43.800.000 PUT/mês * 0.000000028 USD = **1,23 USD**. 1 estilhaço * 730 horas/mês = 730.00 estilhaço-hora/ mês * 0.03 USD = **21,90 USD**. **= 23,13 USD / mês**
Amazon S3	10 KB * 1.000 chamadas/minuto * 60 minutos/hora * 24 horas/dia * 30 dias/mês = 432 GB/mês = = 438 GB * 0,0405 USD = **= 17,74 USD / mês**
TOTAL	**227,02 USD / mês**

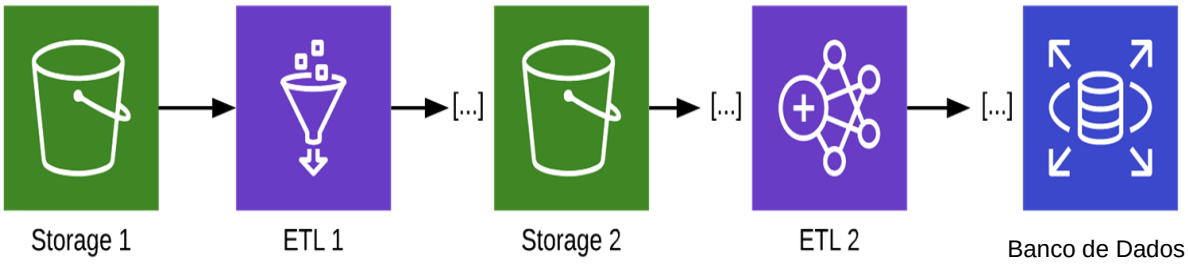
Fonte: Elaboração Própria a partir de Dados da Amazon

Como visto neste exemplo, recebendo apenas 1.000 requisições por minuto e armazenando, por mês, 432 Gigabytes de dados, é estimado o custo mensal de 227 dólares americanos. Embora esta seja uma estimativa para uma aplicação piloto, os serviços escolhidos são facilmente escaláveis. Então, para uma demanda maior, com Terabytes de dados armazenados por mês, e milhares de requisições acontecendo por segundo, é possível alterar os parâmetros utilizados no cálculo, chegando a valores aproximadamente 10 vezes maiores.

3.3 Soluções de ETL

ETLs são estruturas bem versáteis que podem ser acopladas uma após a outra criando um pipeline de transformação dos dados, como na Figura 3.1.1.

Figura 3.3.1 - Diagrama Exemplificando como Encadear Diferentes ETLs com Implementações Diferentes em um Grafo.



Fonte: Elaboração Própria.

Aws Glue

"O AWS Glue é um serviço de extração, transformação e carga (ETL) gerenciado que facilita a preparação e a carga de dados para análises pelos clientes. [...]. Basta indicar ao AWS Glue os dados armazenados na AWS que ele os descobre e armazena os metadados associados (ex.: definição e esquema de tabela) no AWS Glue Data Catalog. Uma vez catalogados, os dados são disponibilizados imediatamente para pesquisas, consultas e ETL."[4]

O AWS Glue é uma excelente opção quando se trabalha com as próprias soluções da Amazon. Ela se integra perfeitamente com o Amazon S3, extraindo os dados de lá para inseri-los em alguma solução de banco de dados relacional ou não-relacional de sua preferência: como Amazon RDS, Amazon Redshift ou até devolver novamente para um diretório no Amazon S3.

[4] "https://aws.amazon.com/pt/glue"

Apache Spark

O Spark é framework de software de código aberto muito robusta que surgiu como uma melhoria ao Hadoop. De acordo com o site oficial “O Apache Spark é um mecanismo de análise unificado para processamento de dados em larga escala”. Ou seja, o Spark é muito mais que uma ferramenta de ETL. Portanto, pode-se utilizar suas ferramentas em várias partes da arquitetura, como na Carga e na Visualização.

A estrutura do Spark também é facilmente integrada com os serviços da Amazon, como o Amazon EMR. Se optar por o serviço de cloud Microsoft, é possível utilizar o Azure Databricks que também gerencia um ambiente na nuvem com a estrutura para executar o framework.

Outras ferramentas de ETL

As ferramentas de ETL existem há talvez mais de 20 anos. É possível listar aqui algumas desses softwares que moldaram a história do Business Intelligence (BI):

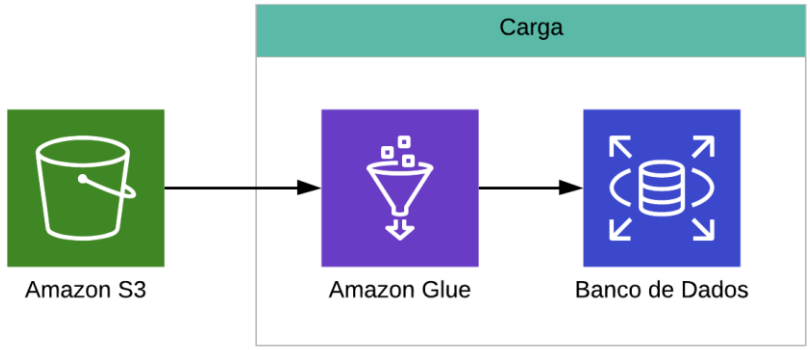
- IBM InfoSphere DataStage;
- Oracle Warehouse Builder;
- Pervasive Data Integrator;
- PowerCenter Informatica;
- SAS Data Management;
- Talend Open Studio;
- SAP Data Services;
- Microsoft SSIS;
- Syncsort DMX;
- CloverETL;
- Jaspersoft;
- Pentaho;
- Nifi.

3.4 Arquitetura de Carga

Agora que foram apresentadas algumas soluções de ETL, é apresentado um modelo de arquitetura voltado para a etapa de Carga descrita anteriormente. É importante lembrar que está sendo modelado uma arquitetura para um projeto piloto e com vistas sempre a escalabilidade do produto para o ponto em que seja capaz de processar Terabytes de dados.

Olhando para o projeto ainda numa perspectiva simplificada, inicialmente só é necessário um único ETL para carregar os dados que foram armazenados em um repositório no Amazon S3 como arquivos, e transferi-los para um banco de dados onde passaram a ser registros.

Figura 3.4.1 - Arquitetura Simplificada para a Segunda Fase do Projeto.



Fonte: Elaboração Própria

Integrar o Apache Spark à solução é interessante, mas ainda não há a necessidade para isso. O Amazon Glue trabalha muito bem o Amazon S3 e pode rapidamente transferir Terabytes de dados de um local para outro.

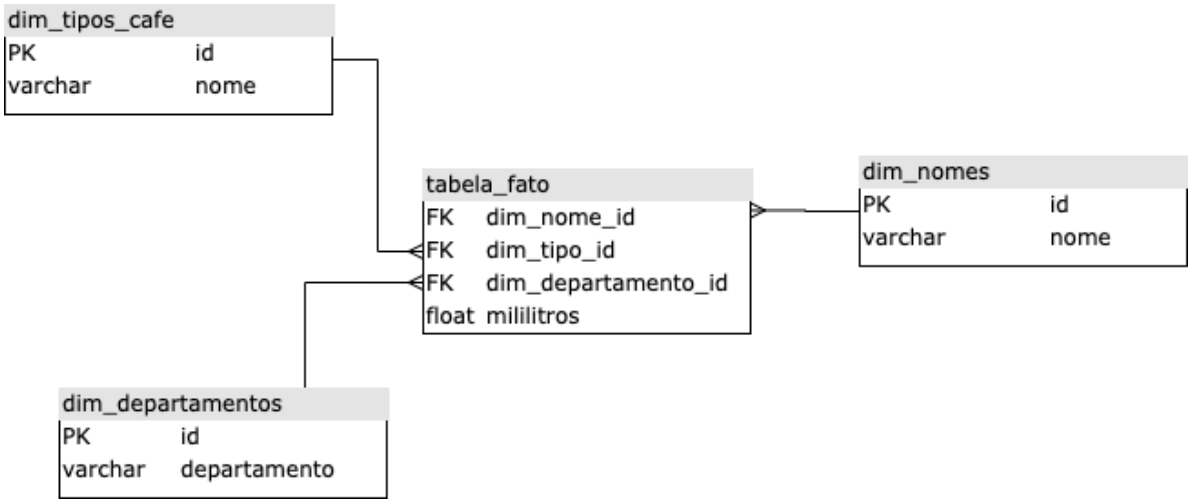
3.5 Topologia Estrela

Ainda não foi decidido qual solução de banco de dados utilizar. Mas antes de fazer isso, pode-se introduzir uma metodologia de modelagem de dados utilizada para armazenar grandes volumes de dados (data warehouses) em banco de dados relacionais, sem perder performance.

Na topologia em estrela, os eventos vão para uma tabela chamada de “Tabela Fato” ou “Tabela Espinha”, que registra os acontecimentos. Cada atributo ou outra informação, fora a métrica que se quer contabilizar, vai para as “Tabelas dimensões”.

Para exemplificar melhor, você tem uma base de dados que contabiliza toda vez que alguém da empresa tomar um café. Cada registro conta com o tipo do café - se foi expresso, cappuccino, latte, ao leite - também registra a quantidade de mililitros, a data em que ele bebeu e o nome do funcionário, o departamento dele. Para transformar estes registros em uma topologia estrela é preciso identificar onde estão as métricas e as dimensões. A métrica vai depender do que você quer contabilizar. Neste exemplo, deseja-se saber qual funcionário está bebendo mais capuccino, então a nossa métrica são os mililitros de café. As outras informações precisam ser extraídas dos registros para formas as tabelas dimensões. Deste modo a lista de nomes de funcionário irá receber a sua própria tabela, assim como também a lista de tipos de café. Na tabela fato só ficam as chaves estrangeiras que apontam para as dimensões (Figura 3.5.1).

Figura 3.5.1 - Diagrama de uma Topologia Estrela Exemplificando o Problema do Cafezinho.



Fonte: Elaboração Própria

As tabelas dimensões são necessariamente muito menores em número de registros do que a tabela fato exatamente pelo fato de serem agregações. Para descobrir qual funcionário está bebendo mais capuccino pode-se primeiramente consultar a dimensão de tipos de café "*dim_tipos_cafe*" e buscar o ID do tipo de café com o nome "capuccino". Com este ID pode-se filtrar a tabela fato pelo ID do tipo de café e somar as métricas de mililitros de forma mais otimizada.

É muito importante que as chaves estrangeiras na tabela fato estejam propriamente indexadas da maneira que mais são acessadas. Por exemplo, se você costuma filtrar com frequência o tipo de café e o departamento, poderia criar um índice de ambas as colunas.

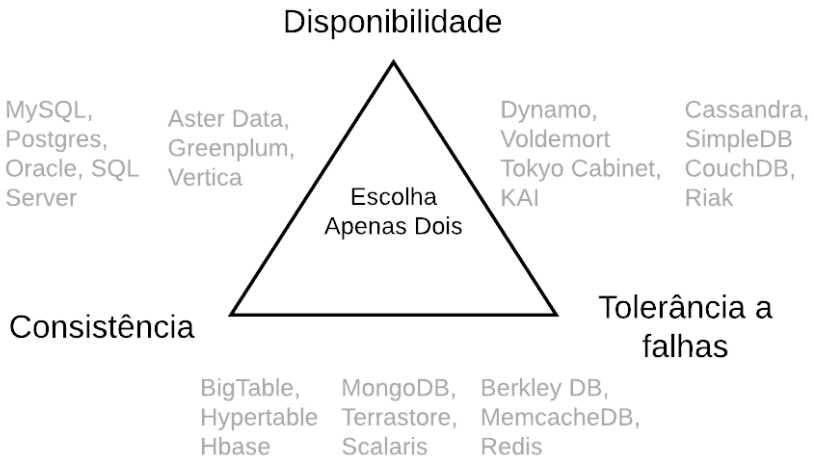
3.6 Consistência, Disponibilidade e Tolerância de Partição

De acordo com o Teorema CAP (Consistency, Availability, Partition Tolerance), ou Teorema de Brewer. É impossível para um armazenamento de dados distribuído fornecer simultaneamente mais de duas das três garantias:

- Consistência: cada leitura recebe sempre a visão mais recente do banco de dados.
- Disponibilidade: o sistema sempre atende as leituras e escritas.
- Tolerância de Partição / a falhas: o sistema continua operando, mesmo que aconteça alguma falha na rede (quebra de conectividade).

Essa teoria é usada para analisar várias plataformas de Big Data. Na Figura 3.6.1 separam-se os tipos de banco de dados que se encaixam nos pares de garantias do Teorema de Brewer.

Figura 3.6.1 – Utilização do Teorema CAP para Guiar Escolhas de Banco de Dados.



Fonte: Elaboração Própria

Consistência e disponibilidade

Sistemas que implementam este tipo de abordagem, normalmente necessitam de forte consistência e altíssima disponibilidade. Isso representa um risco para a aplicação que não sabe lidar com falhas em algum dos nós. Caso essa falha aconteça, o sistema fica indisponível até que os membros do cluster se estabeleçam novamente. Esse tipo de abordagem é chamada de otimista.

Disponibilidade e Tolerância a Falhas

Sistemas que implementam este tipo de abordagem, onde requer alta disponibilidade, escalabilidade e tolerância a particionamento, precisam abrir mão da consistência e aplicar o conceito chamado: Consistência Eventual. A intenção desse tipo de implementação é que esses tipos de sistemas permitam mais escritas e tenham seus dados sincronizados em um momento posterior; o que pode ter uma janela tempo de inconsistência.

Consistência e Tolerância a falhas

São sistemas de abordagem pessimista, o que significa dizer que escritas podem ser negadas. Sistemas que optam por essa estratégia, precisam abrir mão da disponibilidade, mesmo que pouca. Caso exista um particionamento e o sistema não entre num consenso, a escrita pode ser rejeitada/negada. Há um forte trabalho feito nas plataformas que implementam essa abordagem, para que nada de errado aconteça.

3.7 Bancos de Dados Orientados a Colunas

São diferentes dos bancos de dados “tradicionais” pela forma de armazenar os dados. É como se você aplicasse uma transposição nos dados de uma tabela. As linhas viram colunas e as colunas viram linhas. E é assim que funciona. Bancos de dados orientados a colunas armazenam cada coluna separadamente, diferente daquela visão de que cada registro ocupa uma linha. Este tipo de layout de dados tem bastante prestígio no universo do Big Data porque comumente trabalha-se com informações que contém dezenas de colunas e quando são efetuadas consultas, a preocupação é somente com poucas delas. No caso, somente as que fazem sentido para a métrica desejada. Da perspectiva do usuário, o banco de dados funciona da mesma forma. Ainda

assim, não se destaca a necessidade o uso de organizar os dados em uma topologia estrela para extrair ainda mais proveito da estrutura.

Amazon Redshift

"[...] Com o Redshift, você pode consultar petabytes de dados estruturados e semiestruturados em data warehouses, bancos de dados operacionais e seus data lakes usando SQL padrão. O Redshift permite salvar facilmente os resultados das suas consultas de volta no data lake do S3 usando formatos abertos, como o Apache Parquet, para análises adicionais de outros serviços analíticos como o Amazon EMR, Amazon Athena e Amazon SageMaker."[5]

Apache Cassandra

Cassandra é um banco de dados orientado a colunas que trabalha com Tolerância a Falhas e Disponibilidade. Ele supera alternativas populares do NoSQL em benchmarks e aplicativos reais, principalmente por causa da arquitetura do sistema. O rendimento de leitura e gravação aumenta linearmente à medida que novas máquinas são adicionadas, sem tempo de inatividade ou interrupção nos aplicativos.

"Algumas das maiores implantações de produção incluem a Apple, com mais de 75.000 nós armazenando mais de 10 PB de dados, Netflix (2.500 nós, 420 TB, mais de 1 trilhão de solicitações por dia), o mecanismo de busca chinês Easou (270 nós, 300 TB, mais de 800 milhões solicitações por dia) e eBay (mais de 100 nós, 250 TB)." [6]

Apache HBase

HBase é um banco de dados também orientado a colunas que trabalha com Tolerância a Falhas e Consistência. É distribuído, não-relacional, escalável e de código aberto. Foi desenvolvido para hospedar grandes tabelas com bilhões de linhas e milhões de colunas. Funciona integrado com sistema de arquivos distribuídos como Google File System e HDFS.

[5] https://aws.amazon.com/pt/redshift/
[6] https://cassandra.apache.org/

3.8 Ferramentas de Visualização

Esta etapa é a mais visível ao usuário em um projeto de Big Data. Não adianta ter as soluções mais perfeitas de pipeline de dados em tempo real, processamento, machine learning se você não apresenta esses dados de forma coerente. Porque o que se quer com tudo isso é gerar insights. Um insight é um termo utilizado na área de Business Intelligence para designar uma ideia que você extraiu dos dados e irá utilizá-la para tomar uma decisão no seu negócio. Uma boa visualização em dados de trânsito, por exemplo, pode ser decisiva para as cidades decidirem entre aumentar a frota de ônibus ou reorganizar os trajetos.

Amazon QuickSight

"O Amazon QuickSight é um serviço de inteligência de negócios rápido e baseado na nuvem que facilita a entrega de insights a todos em sua organização. Como um serviço gerenciado, o QuickSight permite que você crie e publique facilmente painéis interativos que incluem o Insights de ML. Os painéis podem ser acessados de qualquer dispositivo e incorporados a aplicativos, portais e sites."[7]

Com o QuickSight é possível criar seus próprios dashboards administrativos e deixá-lo acessível para os usuários. É possível integrá-lo com o Amazon Athenas, Amazon Redshift, entre outros armazéns de dados.

Microsoft Power BI

O Microsoft Power BI também é uma ferramenta de visualização compatível com os serviços da Microsoft Azure. Com esse serviço é possível fornecer visualizações interativas com uma interface para que os usuários criem os seus próprios relatórios e dashboards.

[7] "https://aws.amazon.com/pt/quicksight/"

CKAN

É uma ferramenta de código aberto que te ajuda a organizar e publicar coleções de dados. Como um gerenciador de conteúdo como o Wordpress, só que para dados ao invés de posts. É usado por governos nacionais e locais, instituições de pesquisa e outras organizações que coletam muitos dados.

Magda

O Magda é um sistema de catálogo de dados (muito semelhante ao CKAN) que fornecerá um único local onde todos os dados de uma organização podem ser catalogados, enriquecidos, pesquisados, grandes ou pequenos, disponíveis como arquivos, bancos de dados ou APIs. O Magda fornece uma visão única de todos os dados de interesse de um usuário, independentemente de onde os dados estão armazenados ou de onde foram originados. O sistema pode rastrear rapidamente fontes de dados externas, rastrear alterações, fazer aprimoramentos automáticos e fazer notificações quando ocorrem alterações, oferecendo aos usuários de dados um balcão único para descobrir todos os dados disponíveis para eles.

Solução Customizada

Também existe uma alternativa de construir a sua própria solução de visualização. Uma página web, um aplicativo, uma planilha interativa, uma simulação tridimensional, entre outros. Nesses casos, o foco é gerar uma experiência única e simplificada para o usuário. Somente é recomendado seguir por esse caminho quando já se utilizou as alternativas de serviços mencionados acima para validar as ideias do projeto.

3.9 APIs de Consulta de Dados

APIs de consultas possibilitam a criação de modelos de negócio e ferramentas em cima dos dados disponibilizados. Desenvolvedores de software, pesquisadores e empresas poderão contribuir com novas soluções e diferentes insights com suas perspectivas. Com esse tipo de abordagem, dá-se liberdade para a própria comunidade otimizar o ambiente em que vive.

Ferramentas como Magda e CKAN, já proporcionam formas simples de acessar os dados publicados por meio de API Rests, mas elas não foram desenvolvidas para realizar consultas customizadas e voltadas para respostas em tempo real de forma a suportar a demanda de acesso. Para esses casos, existem outras formas de disponibilizarmos acesso a esses dados.

API Serverless

Uma API Rest Serverless desenvolvida especificamente para acessar os dados é uma abordagem possível. Nesse caso, o foco é performance e escalabilidade para um alto número de acessos de leitura aos dados. O termo “serverless”, que quer dizer “sem servidor” é um modelo de execução de computação em nuvem no qual o provedor executa o servidor e gerencia dinamicamente a alocação de recursos da máquina. O preço é baseado na quantidade real de recursos consumidos por um aplicativo, e não nas unidades de capacidade pré-adquiridas. O AWS Lambda é um exemplo de serviço de execução do código serverless da Amazon.

3.10 Estimativa de Preço (Visualização)

Para estimar os custos de uma solução de visualização, serão elencados os serviços para um produto piloto. Para isso, vamos utilizar o Amazon Quick Sight como um dashboard de visualização que exibe os dados disponibilizados por uma base de dados orientada a colunas como Amazon Redshift.

No Amazon Quick Sight, a definição do preço se baseia em duas entidades: o Autor e o Leitor. O autor é quem gerencia e configura visualizações no dashboard. O leitor seria o usuário que só tem permissão para visualizar os gráficos disponibilizados para consultas e insights. Para o leitor, também entra o conceito de uma “sessão” de acesso. Uma sessão compreende o período de 30 minutos de acesso do leitor.

Tabela 3.10.1: Amazon Quick Sight - Enterprise Edition – Estimativa de Custo

Amazon Quick Sight - Enterprise Edition	
Preço por Autor	18 USD / mês
Preço por Leitor	0,30 USD até 5 USD por sessão por mês

Fonte: Elaboração Própria a partir de Dados da Amazon

Tabela 3.10.2: Amazon Quick Sight – Standard Edition – Estimativa de Custo

Amazon Quick Sight - Standard Edition	
Preço por Autor	12 USD / mês
Preço por Leitor	N/A

Fonte: Elaboração Própria a partir de Dados da Amazon

A versão “Standard Edition” já servirá adequadamente para o projeto piloto, então se faz a precificação a partir dela. No Quick Sight, é possível visualizar dados de vários tipos de fontes diferentes. Mas para essa simulação, imagina-se que os dados já estão normalizados e armazenados no Redshift.

O Amazon Redshift usa o conceito de “Nó”. Um nó significa uma máquina virtual que processará uma consulta feita. Quanto mais “nós”, mais opções existem de distribuir o processamento em diferentes instâncias. Para essa simulação, utilizam-se 4 instâncias. Cada uma com 2 núcleos de CPU e 15 GB de memória RAM (dc2.large).

Tabela 3.10.3: Amazon Redshift – Standard Edition – Estimativa de Custo

Amazon Redshift - Standard Edition	
dc2.large (2 CPU, 15 GiB RAM)	0,25 USD / hora
ds2.xlarge (4 CPU, 31 GiB RAM)	0,85 USD / hora
ra3.4xlarge (12 CPU, 96 GiB RAM)	3,26 USD / hora
dc2.8xlarge (32 CPU, 244 GiB RAM)	4,80 USD / hora
ds2.8xlarge (36 CPU, 244 GiB RAM)	6,80 USD / hora
ra3.16xlarge (48 CPU, 384 GiB RAM)	13,04 USD / hora

Fonte: Elaboração Própria a partir de Dados da Amazon

As instâncias trabalham por demanda. Inicializam no momento da primeira consulta e cessam suas atividades uma vez que a demanda finaliza. Ainda assim, é calculado imaginando que os nós trabalharão incessantemente durante o mês.

Tabela 3.10.4: Estimativa de Custo de Visualização

Amazon Quick Sight (Standard Edition)	4 autores * 12 USD / mês **= 48,00 USD / mês**
Amazon Redshift	4 nós * 0,25 USD / hora * 730 horas / mês **= 730 USD / mês**
TOTAL	**778 USD / mês**

Fonte: Elaboração Própria a partir de Dados da Amazon

4. Desenvolvimento de Plataformas de Gestão Urbana

Agora que já se conhecem os aspectos conceituais e a infraestrutura mínima para o desenvolvimento das plataformas, pode-se trabalhar efetivamente na criação dos protótipos que visam ajudar a resolver problemas de gestão urbana.

Como explicado anteriormente, o Produto 4 é composto por quatro plataformas e ele é o catalisador de trabalhos desenvolvidos anteriormente nesse projeto: a fase de mapeamento do ecossistema de dados e a estrutura regulatória das cidades do Componente I e a fase de Desenvolvimento dos Recursos Legais e Institucionais para Acesso e Análise de Big Data, explorado no Componente II. Ademais ele é fruto de das interações e discussões realizadas com as cidades ao longo do projeto que deu origem e sugeriram os protótipos objetos desse TR 8. Ao longo deste componente, os representantes dos governos locais juntamente com a academia e a empresa de tecnologia tem aberto um canal co-criação dos produtos.

As questões de gestão urbana que esse projeto pretende prototipar são resultados dos exercícios de *design thinking* com os representantes das cinco cidades latino-americanas. Nesse encontro foram discutidos os principais problemas comuns entre as cidades e ao final foram listadas algumas possíveis políticas a serem desenvolvidas durante o período do projeto: Produto 4.1 - Plataforma de Transparência; Produto 4.2 - Estimativa de Poluição do Ar; Produto 4.3 - Apoio ao Transporte Público e Produto 4.4 - Identificação de Acidentes.

O objetivo é construir protótipos com potenciais para replicabilidade capazes de fazer frente aos problemas urbanos experimentados pelas cidades latino-americanas. Cada um dos produtos elencados acima foi desenvolvido considerando a complexidade do produto, bem como aspectos particulares de cada cidade, seja por necessidade ou por disponibilidade de dados de cada local que possibilitou o desenvolvimento de mais ou menos funcionalidades. Sendo assim, os produtos atingirão diferentes níveis de maturidade tecnológica limitados pelo período de duração do projeto para a fase de desenvolvimento e de testes adicionais para que fosse colocada em produção e uso se assim as cidades desejarem. A principal intenção do projeto é demonstrar as capacidades de desenvolvimento de produtos tecnológicos a partir de dados que estão disponíveis para os gestores e seus potenciais usos. Apesar de temas comuns relativos aos desafios urbanos, cada municipalidade enfrenta uma situação peculiar e eventualmente um mesmo produto pode ser explorado de uma forma diferente, adicionando ou não mais funcionalidades ao que já foi criado.

Justifica esse projeto o modelo lógico que ancora tais políticas públicas de inovação aberta que indica haver soluções tecnológicas inovadoras capazes de resolver ou minimizar diversos problemas urbanos comuns às cidades-piloto desse projeto. Esses problemas foram mapeados, bem como levantados e qualificados todos os dados abertos existentes nessas cidades de modo que tais protótipos sejam agora construídos e viabilizados num processo de co-criação para responder às questões que de fato sejam de interesse das regiões envolvidas no projeto.

Um Novo Paradigma sobre o Uso de Dados em Políticas Públicas

Esse projeto se insere no contexto de estudos sobre políticas públicas com base em métricas e evidências que resultem em melhores resultados e mais transparência para o cidadão. Apesar de um crescente uso de dados para as decisões de políticas públicas, este ainda é muito limitado na maioria das cidades, com a divulgação de informações com baixa periodicidade, atrasos e coletadas de forma bastante custosa. Ao final do processo de coleta eles acabam não sendo tratados e utilizados inteligentemente em políticas públicas urbanas.

Em geral, tecnologias são desenvolvidas para serem utilizadas pelos gestores públicos descontextualizadas das realidades locais e incoerentes com a lógica das políticas públicas que requerem um nexo causal bem estruturado entre insumos, ações e resultados. Para contornar essa falha, é necessária a construção de tais tecnologias no contexto de sistemas inovativos que requerem a interação da academia científica, modelando hipóteses para que o setor produtivo tecnológico desenvolva soluções em parceria sinérgica com um governo orientado pela missão de dissipar os problemas urbanos com ferramentas tecnológicas. Tais parcerias resultariam, por exemplo, na aplicação de *machine learning* e inteligência artificial na gestão pública a partir da massiva quantidade de dados existentes e coletáveis com grande frequência – ou até mesmo em tempo real – em ações e políticas que melhorem a gestão urbana.

Este projeto representa, portanto, não apenas um trabalho especializado em desenvolvimento de estruturas de tecnologia, mas sobretudo de uma mudança de paradigma sobre o uso de dados em políticas públicas, com vistas à capacitação completa sobre a relevância, a utilização das plataformas que serão desenvolvidas, sendo o seu objetivo final a aplicação dessas informações em intervenções de políticas públicas e aumento da transparência e participação da sociedade.

Almeja-se que ao longo do desenvolvimento dos protótipos possamos refletir, analisar e recomendar a partir dos dados existentes os tipos de aplicação que os gestores públicos podem

fazer no estágio atual e as oportunidades de continuidade de aplicação futura para melhorar a gestão pública e a implementação de políticas públicas com base em dados confiáveis, acurados e com alta frequência de divulgação.

4.1 Plataforma da Transparência

A Transparência e Comunicação com o Cidadão

A plataforma de Transparência é o primeiro protótipo desenvolvido no TR8. Este produto se refere à criação de uma plataforma a partir dos dados que já estão em CKAN (TR6) adicionando algumas usabilidades à base. Ele pretende agregar em sentido amplo usabilidades que permitam dar mais transparência aos cidadãos uma vez que trazem a possibilidade de as cidades poderem adicionar informações relevantes à população, sejam dados e indicadores das mais diversas áreas de interesses como também informações sobre desenvolvimento de projetos urbanos.

Sua concepção é um resultado dos esforços empreendidos nas etapas anteriores desse projeto, trazendo elementos importantes do **Produto do TR1** – em que foram realizadas análise do ambiente de dados das cidades participantes do projeto; do **Produto do TR2**, onde foram analisadas a disponibilidade e qualidade dos dados das cidades, dos trabalhos **do TR6** que criou a plataforma CKAN (o Dataurbe), do exercício de *Design Thinking* e Webinário onde os representantes das cidades puderam se reunir e conjuntamente refletir sobre os desafios de gestão pública que podem ser solucionados a partir de propostas tecnológicas.

Do ponto de vista de políticas públicas, essa plataforma pretende contribuir no sentido de melhorar a transparência da gestão pública, viabilizar a participação popular na tomada de decisão governamental, melhorar a comunicação com o cidadão e seu engajamento e aumentar a *accountability*.

A transparência estimula a participação social, permite ao cidadão a possibilidade de acesso à informação pública e aos atos praticados pela administração pública e aproxima o cidadão à gestão de seus representantes e é, portanto, uma importante ferramenta para a prática da democracia.

A questão da transparência já foi abordada anteriormente no **TR1 do Componente I** deste projeto. O TR1 explorou o Desenvolvimento de Relatório Regulatório com Lista e Crítica da

Legislação para o Uso de Dados Públicos e Privados nas Cidades. O documento fez um estudo do marco analítico de referência para orientar as cidades nos usos de dados na direção de se tornarem “cidades inteligentes”. Para tal, entre outras dimensões, foi analisado o nível de maturidade da transparência e participação nas cidades, classificando-as em cinco níveis: (i) incipiente; (ii) em formação; (iii) constituído; (iv) implementado; e (v) consolidado. A partir dessa análise foi possível fazer recomendações de acordo com o estágio em que a cidade se encontra.

De modo geral, verificou-se que as cidades apresentavam níveis intermediários de maturação da dimensão da transparência entre “em formação” e “implementado”.

Tabela 4.1.1: Resumo da Análise de Maturação das Ferramentas de Transparência

Cidade	**Transparência**	
	Nível de Maturidade	**Comentário**
Miraflores	Constituído	Existe marco normativo nacional e local determinando a transparência e o acesso à informação pública, bem como políticas locais de disponibilização de dados públicos.
Montevidéu	Implementado	Existe um marco normativo nacional e resolução local que regula transparência e o acesso à informação pública, bem como canais de solicitação de informação de fácil acesso.
Quito	Implementado	Considerou a qualidade e a variedade das informações disponibilizadas, o nível de maturidade “Implementado”. Importante avançar no sentido de informar como os dados gerados pelos cidadãos, como seus dados pessoais, são utilizados na formulação de políticas públicas e na tomada de decisão do município.
São Paulo	Implementado	Existe um marco normativo nacional e municipal para a transparência e o acesso à informação pública, bem como canais de solicitação de informação de fácil acesso
Xalapa	Constituído	Existência de marco normativo que promove a transparência e acesso a dados públicos

Fonte: FGV, Termo de Referência 1

Percebe-se que a maioria das cidades já colocaram esforços no sentido de promover a transparência e tem a compreensão da importância desse tipo de monitoramento como uma boa prática de gestão pública. É importante destacar que, para além da disponibilização de dados, a transparência requer que o cidadão tenha acesso à informação completa, objetiva, confiável e de qualidade e que sejam compreensíveis e que estejam disponíveis totalmente abertos de comunicação. Ela engloba atributos como acesso, abrangência, relevância, qualidade e confiabilidade (Figueiredo e Dos Santos 2013).

No sentido de dar confiabilidade e dinamismo à essa relação com o cidadão através da informação, e também da necessidade de lidar com um volume cada vez maior e mais frequente de dados, é importante um movimento em direção à promoção da inovação de dados abertos no uso da gestão pública urbana facilitando tanto o trabalho do gestor público que informa quanto dos cidadãos que as consomem. Assim, a transparência é uma via de mão dupla: de um lado a administração tem o dever de dar publicidade aos seus atos e, por outro, o cidadão tem o direito a ser informado. Por meio da informação disponível por meio eletrônico, desenvolve-se um controle preventivo, estimula-se a participação popular, torna-se o exercício do poder mais transparente e, portanto, mais democrático (Limberger 2007)

Nesse sentido, o protótipo da plataforma de transparência a ser desenvolvida no âmbito desse projeto tem por objetivo desenvolver as seguintes usabilidades aos dados: (i) possibilitar que as cidades adicionem dados relevantes à população dentro de uma estrutura que comporte a armazenagem de um volume grande de dados e adicionem informações sobre o desenvolvimento de projetos urbanos, permitindo aos cidadãos acompanhar o andamento do projeto, (ii) permitir que o gestor público construa ferramentas de visualização para análise e consumo próprio e dar a capacidade de visualização e tabulação desses dados de maneira simples pelos cidadãos.

4.1.1 Armazenagem, Difusão e Distribuição de Dados - Plataforma CKAN

A plataforma de transparência tem como ponto de partida o trabalho desenvolvido no TR6, onde foi desenvolvida a Plataforma CKAN, o Dataurbe. No contexto desse produto específico, o próprio CKAN já representa a primeira usabilidade que possibilita as cidades adicionarem dados relevantes de suas políticas em um ambiente que comporte um volume massivo de dados. Além

disso, o gestor tem a possibilidade de colocar na plataforma não apenas séries de indicadores, mas também as próprias informações sobre seus projetos em desenvolvimento para que a população possa acompanhar

Ainda que durante o período do projeto, a quantidade de dados lançados na plataforma ainda não seja muito grande, a ideia é que ela seja continuamente alimentada e que esteja preparada para suportar um volume grande de dados.

Desta forma, a seguir, retoma-se brevemente o conteúdo do **TR6** antes de prosseguir para a apresentação do manual de preparação de visualização de dados.

"O CKAN é uma plataforma livre para armazenamento e distribuição de dados, como planilhas e o conteúdo do código de banco de dados. A base de dados é mantida pela Open Knowledge. Considerada madura e amplamente utilizada pela comunidade de dados abertos, tem se mostrado eficiente para publicação, gerando vistas personalizadas, construção e identificação de metadados, e para compartilhar conjuntos de dados, a fim de facilitar o processo de abertura e estruturação de dados mais fácil e colaborativo" (FGV 2018). Assim, o desenvolvimento do CKAN em si já é um passo importante para a promoção de uma política de transparência de dados.

O objetivo do Dataurbe é de criar um ambiente de armazenamento, difusão e distribuição de dados abertos entre as cinco cidades envolvidas no projeto. Também serve como um ambiente de catalogação de dados em comum entre as cidades, de modo que seja possível que a comunidade se aproprie destes dados para fortalecer seu uso para o desenvolvimento de protótipos que tenham a finalidade de apoiar políticas públicas.

A ideia central era utilizar uma instância do CKAN com a capacidade de validar padrões de arquivos tabulares importados para a plataforma, com o fim de manter uma uniformidade de dados dentro da plataforma. Isto porque, a existência de um volume massivo de dados por si só, sem ter as condições necessárias para uso podem ser pouco úteis para a sociedade civil, e, em consequência, ter uma baixa capacidade de se transformarem em informações relevantes.

O potencial para melhoria da disponibilidade e uso dos dados das cidades foi verificado na etapa de estudo da qualidade de dados abertos **(TR2)** e também surgiu como assunto relevante nas discussões ocorridas nos encontros das cidades. Desta forma, surge a oportunidade da criação de um portal com dados abertos estruturados e comuns entre as cidades participantes do projeto,

de forma que os municípios possam publicar dados em comum entre eles, para que esses dados possam ser utilizados como base de comparação, e que possam ser utilizados para fortalecer evidências com dados para apoiar políticas públicas de gestão urbana.

A plataforma tem sido alimentada com os dados das cinco cidades, seguindo os protocolos apresentados no **TR6**. Isso nos dá a oportunidade para que essas informações sejam acessadas e trabalhadas por nossa equipe e pelas cidades futuramente, inclusive aumentando as possibilidades conforme mais informações sejam incluídas na plataforma. Durante o desenvolvimento deste produto, por exemplo, foi trabalhada adicionalmente uma solução para integração do Dataurbe com a plataforma Geosampa[8], mapa digital da cidade de São Paulo em formato aberto.

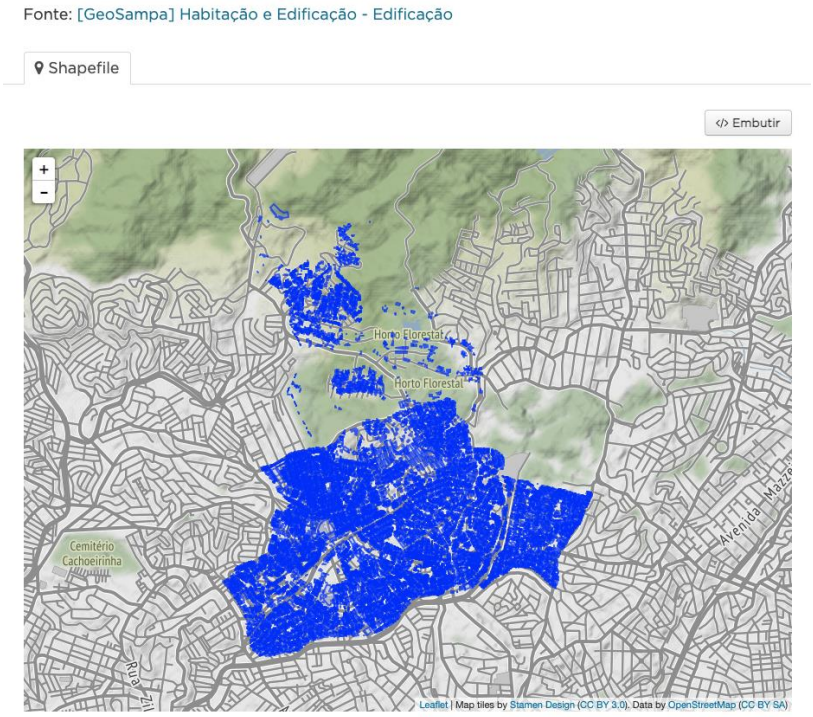
Integração com o GeoSampa

A fim de tornar o Dataurbe uma plataforma mais abrangente possível, realizou-se um trabalho de integração com a plataforma do GeoSampa. Para isso, foi construído uma ferramenta que importa de forma inteligente os dados disponibilizados abertamente no GeoSampa e os cadastra na forma de conjunto de dados do CKAN. A ferramenta foi desenvolvida de forma a não depender exclusivamente do Dataurbe e pode ser utilizada em quaisquer servidores CKAN. O código fonte e o vídeo explicativo sobre a instalação e execução da ferramenta podem ser encontrados nos anexos deste documento.

A imagem abaixo mostra uma visualização de um arquivo do tipo Shapefile extraído do GeoSampa e exibido diretamente no Dataurbe.

[8] http://geosampa.prefeitura.sp.gov.br

Figura 4.1.1.1: Visualização de Dados Extraídos do GeoSampa no Dataurbe



Fonte: Elaboração Própria

4.1.2 Publicando Projetos Públicos no Dataurbe

As cidades podem desejar atualizar seus cidadãos sobre os progressos de seus projetos e obras. Algumas cidades desenvolveram suas próprias páginas para publicar essas informações, como Montevidéu, por exemplo, por meio da plataforma ´Montevideo Mejora´[9].

Neste projeto, estuda-se como é possível utilizar o próprio Dataurbe para melhorar esta comunicação. Com isso, desenvolveu-se uma extensão à plataforma que tornou possível publicar para uma sessão de “Acompanhamento de Projetos” customizada especialmente para este fluxo. Assim as cidades poderão se utilizar de uma plataforma única para gerar estes relatórios.

[9] https://www.montevideo.gub.uy/SigesVisualizadorMM/faces/inicio.xhtml

Para isso, as próximas seções deste documento serão apresentadas na forma de um manual de instruções passo a passo. Este manual também está disponível no formato de vídeo e o link de acesso está disponível na seção de anexos.

Passo 1 - Autenticar a Sessão

Para começar, clique em Acessar - Dataurbe (appcivico.com). Cada gestor deverá ter credenciais[10] de acesso para o Dataurbe. Este passo é obrigatório para ter acesso às opções de cadastro protegidas. Entre com seu usuário e senha e clique em “Acessar”.

Figura 4.1.2.1: Painel de autenticação da sessão

[10] As credenciais são fornecidas apenas aos representantes das cidades e devem ser solicitadas diretamente à coordenação do projeto.

Passo 2 – Acesse o Formulário

No menu superior, clique em “Conjunto de dados”. Clique no botão “Cadastrar Projeto Público”;

Figura 4.1.2.2: Funcionalidade que permite o cadastro de projetos públicos

Passo 3 – Preencha o Formulário

É possível alterar a língua do projeto e prosseguir a partir de agora com um exemplo de projeto extraído do site “Montevideo Mejora”.

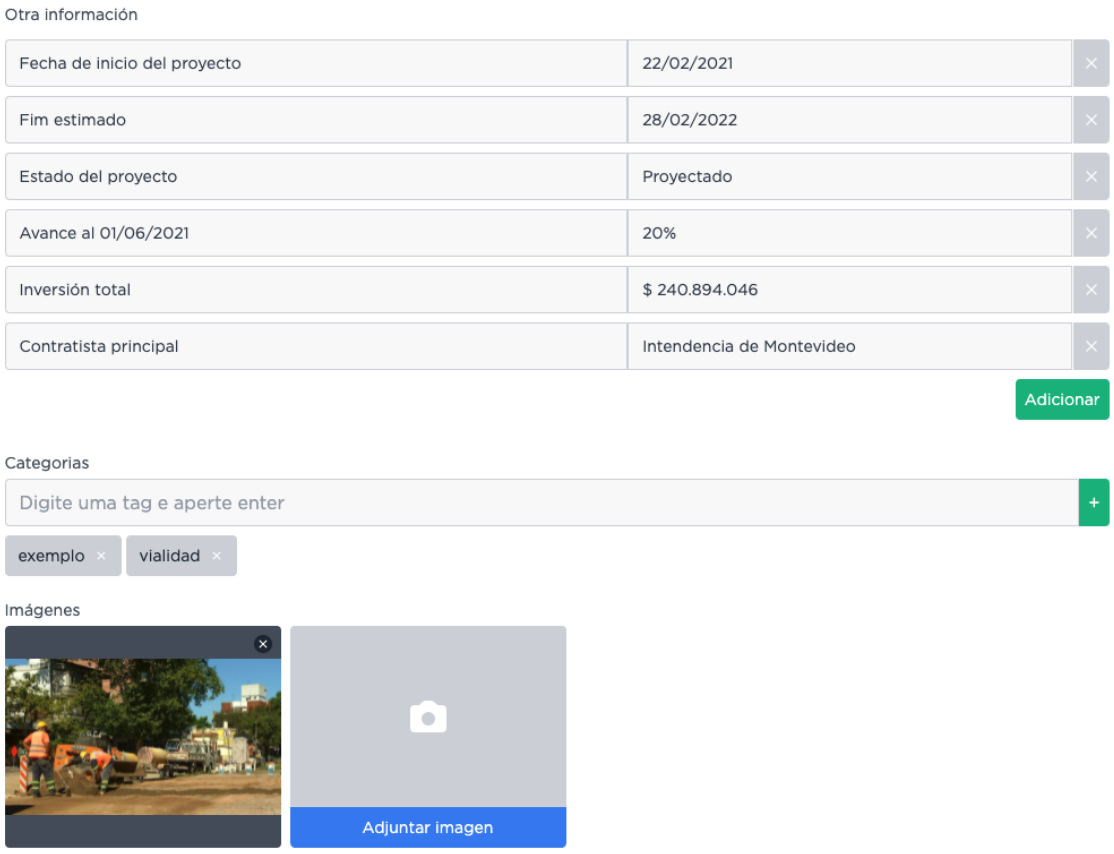
Figura 4.1.2.3: Funcionalidade que altera o idioma do projeto

O formulário foi construído para ser bem simples e intuitivo. O gestor deverá preencher conforme solicitado os dados relacionados ao projeto.

Figura 4.1.2.4: Painel de cadastro de projetos públicos

Adicione outras informações, categorias, imagens e arquivos.

Figura 4.1.2.5: Inclusão de outras informações acerca do projeto



Por fim, clique em “Cadastrar no Dataurbe” para finalizar a publicação.

Figura 4.1.2.6: Publicando no DataUrbe

Passo 4 – Visualize o Resultado

Pronto, agora o projeto está devidamente publicado no Dataurbe e disponível para ser acessado pelos cidadãos.

Figura 4.1.2.7: Painel de projetos públicos

Características Técnicas do Formulário de Cadastro de Projetos Públicos

O projeto foi desenvolvido separadamente ao CKAN utilizado no Dataurbe, utilizando algumas tecnologias recentes do mercado como “React” e “Next JS”. O acesso ao código fonte do projeto está disponível nos anexos deste documento. A escolha técnica da separação lógica com o Dataurbe se deve ao fato de facilitar a integração deste mesmo formulário a qualquer CKAN que disponibilize um link no formato requerido. O formulário apontará para o CKAN alvo e efetuará um cadastro de um conjunto de dados formatado para um projeto.

Para integrar um CKAN é necessário acessá-lo da seguinte forma:

```
[X]/api/auth?host=[Y]&apiKey=[Z]&lang=[W]
```

No lugar de:

Tabela 4.1.2.1: Passo-a-passo para o cadastro de projetos públicos

Token	Explicação	Exemplo
`[X]`	Preencha o servidor onde o formulário está hospedado.	`https://example.app.com`
`[Y]`	Preencha o servidor CKAN onde o projeto público será cadastrado.	`https://dataurbe.appcivico.com`
`[Z]`	A chave de API disponibilizada para cada usuário cadastrado no CKAN.	`abcd-efgh-ijkl-opqr`
`[W]`	A língua padrão no qual o formulário se apresentará. Os valores disponíveis são: ● “pt” (para português) ● “es” (para espanhol)	`pt`

	• “en” (para inglês)	

Um exemplo completo:

```
https://example.app.com/api/auth?host=https://dataurbe.appcivico.com/&apiKey=abcd-efgh-ijkl-opqr&lang=pt
```

Caso as informações não estejam corretas e a aplicação não consiga se comunicar com o CKAN, o formulário exibirá a seguinte notificação:

Figura 4.1.2.8: Notificação de falha na comunicação com o CKAN

Se estiver visualizando esta mensagem, revise o endereço do servidor CKAN e a chave de API informada.

4.1.3 Ferramentas de Visualização e Tabulação

Esta seção tem como finalidade apresentar a possibilidade de permitir aos cidadãos acessar as informações das cidades através do Dataurbe e de dar capacidade ao gestor público de analisar pequenos e grandes volumes de dado a partir dos recursos da Dataurbe.

As próximas seções apresentam em forma de manual as funcionalidades de visualização e tabulação do CKAN simplificadas e funcionalidades mais avançadas, caso o usuário - cidadão ou gestor – possua conhecimento de linguagem técnica. O passo a passo é apresentado a seguir e pode ser consultado nos links dos vídeos.

O manual a seguir tem como objetivo instruir os leitores sobre como realizar o download de dados na plataforma CKAN e mostrar as ferramentas e técnicas utilizadas para preparar os dados para visualização. Para isso, o procedimento se divide em dois: Simplificado e Avançado. De modo que a abordagem “Simplificada” não requer conhecimento avançado em tratamento de dados, linguagens de programação ou de uma infraestrutura na Amazon Web Services. Na abordagem “Avançada” utiliza-se uma linguagem técnica, cuja compreensão requer conhecimento prévio nas áreas de Tecnologia da Informação.

Manual – Procedimento Simplificado

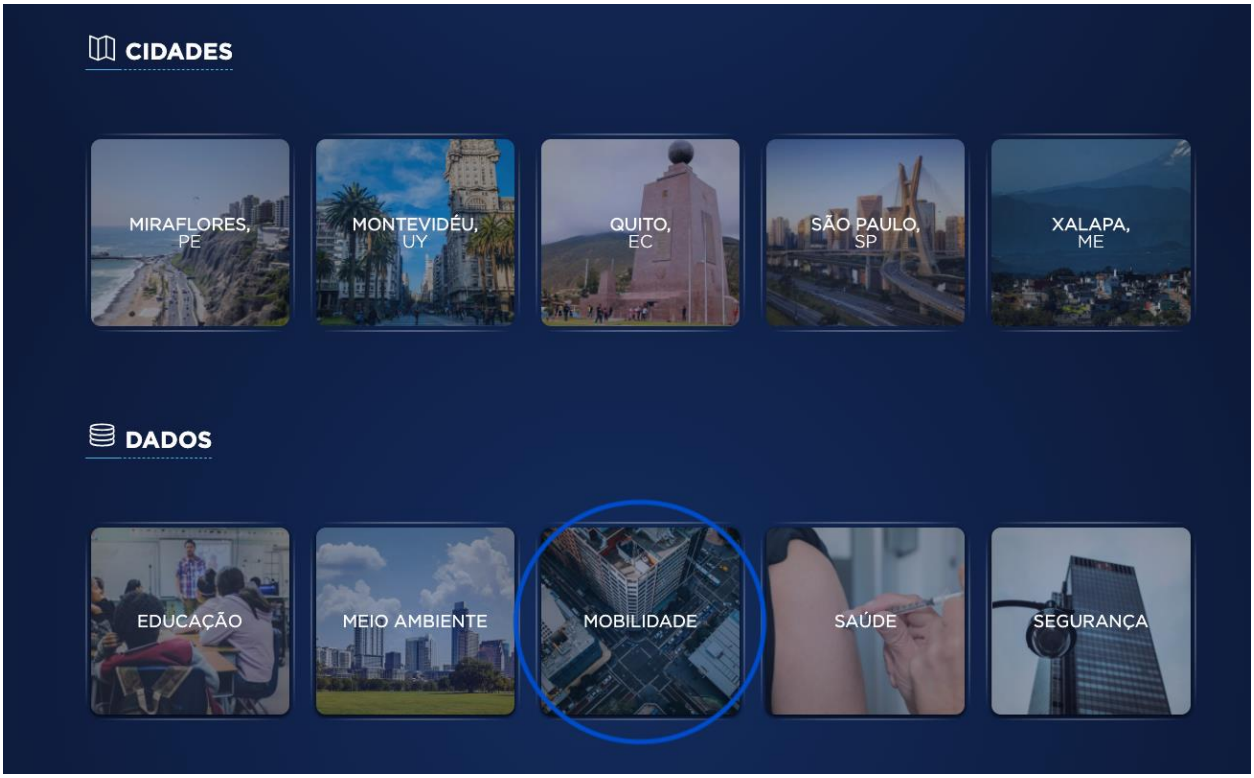
Consulta e Visualização de Dados no Dataurbe

A primeira e mais simples forma de visualizar os dados que estão disponíveis no Dataurbe, é diretamente pelo site. Alguns conjuntos de dados disponibilizam pré-visualizações direto na plataforma, mas outros não. Neste tópico descreve-se como acessar essas pré-visualizações para os conjuntos disponíveis e nos tópicos seguintes como proceder para os outros casos.

Passo 1 - Buscando uma Base de Dados

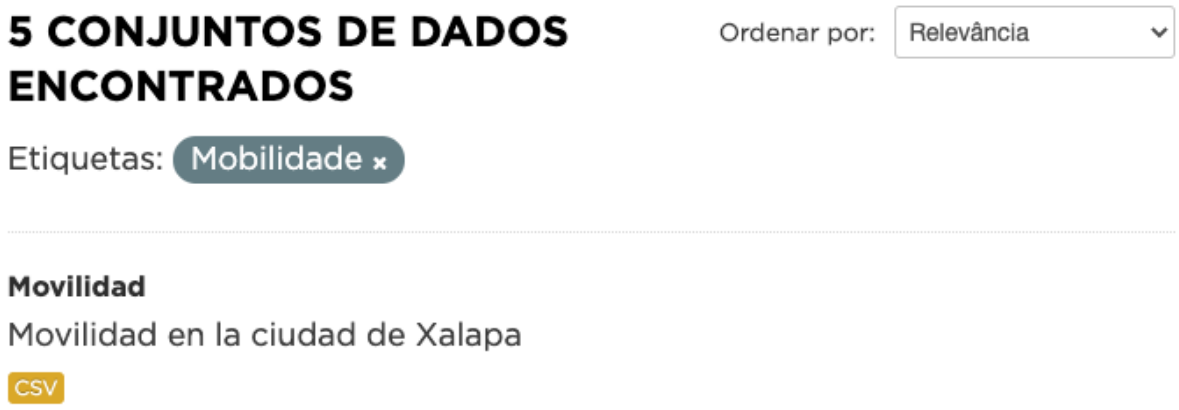
Acesse o site https://dataurbe.appcivico.com/.

Figura 4.1.3.1: Painel inicial do DataUrbe



Escolha uma categoria. Estão disponíveis dados separados por cidade como também por área. Para esta demonstração, é acessado o item “Mobilidade”.

Figura 4.1.3.2: Busca de conjuntos de dados



Escolha algum conjunto de dados disponível.

Quando disponível, aparecerá uma opção de “Pré-visualização”.

Figura 4.1.3.3: Funcionalidade de pré-visualização dos dados

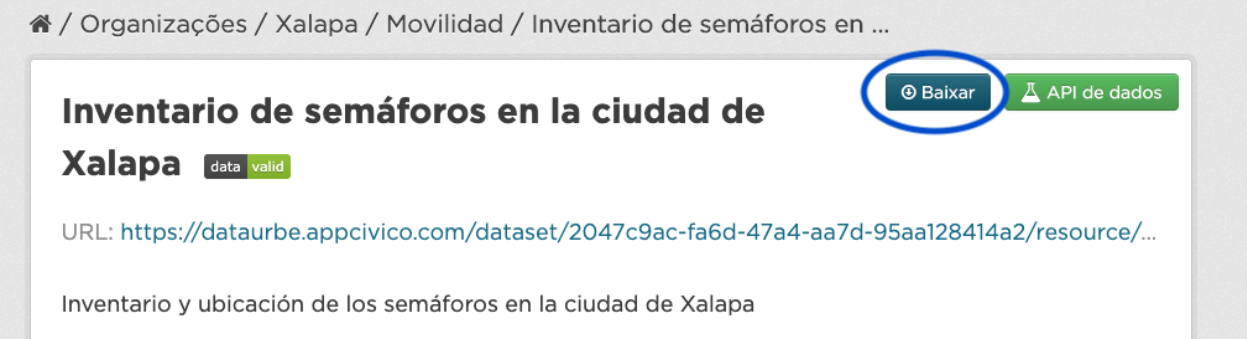
Ao clicar na opção “Pré-visualização” você será redirecionado para uma página customizada pelo autor do conjunto de dados com alguma visão dos dados. No seguinte exemplo vemos uma visão em tabela.

Figura 4.1.3.4: Visualização dos dados

_id	No	UBICAC...	BANDE...	BANDE...	LATIGO ...	LATIGO ...	PEDEST...	2 LUCES	3 LUCES	4 LUCES
1	1	LAZARO...	6				1		15	
2	2	LAZARO...	2				2	2	2	2
3	3	LAZARO...	2				2			8
4	4	LAZARO...	2	1					2	3
5	5	LAZARO...								
6	6	LAZARO...	1	3					12	
7	7	LAZARO...	2				2	2	2	2
8	8	LAZARO...	2						12	
9	9	RUIZ CO...	3						7	
10	10	RUIZ CO...	4						6	2
11	11	RUIZ CO...	2						2	4
12	12	RUIZ CO...	4							8
13	13	RUIZ CO...			2		1	2	1	1

Para ter acesso direto aos dados clique em “Baixar”

Figura 4.1.3.5: Funcionalidade que permite o download dos dados



Publicando Visualizações no Dataurbe

Esta seção tem como finalidade instruir os gestores das cidades em como criar visualizações a partir dos dados anexados no Dataurbe e como torná-las acessíveis publicamente para os cidadãos.

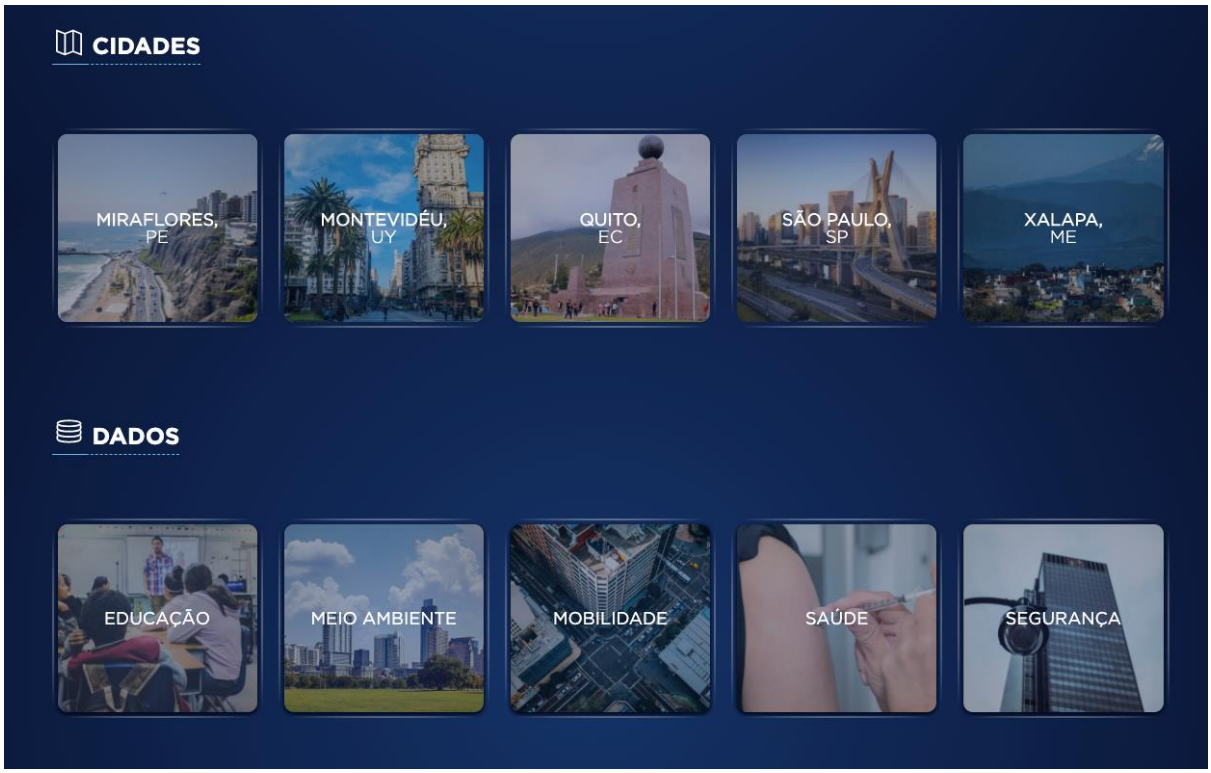
Para isso, as próximas seções deste documento serão apresentadas na forma de um manual de instruções passo a passo caso o. Este manual também está disponível no formato de vídeo e o link de acesso está disponível na seção de anexos.

Passo 1 - Buscando uma Base de Dados

O primeiro passo é encontrar uma base de dados que se queira visualizar. Para isso, acesse o site https://dataurbe.appcivico.com/.

Na página, você poderá buscar por dados de cada uma das cidades, ou pelas categorias de Educação, Meio Ambiente, Mobilidade e etc.

Figura 4.1.3.6: Painel inicial do DataUrbe



Você também pode pesquisar por conjunto de dados digitando na barra de busca.

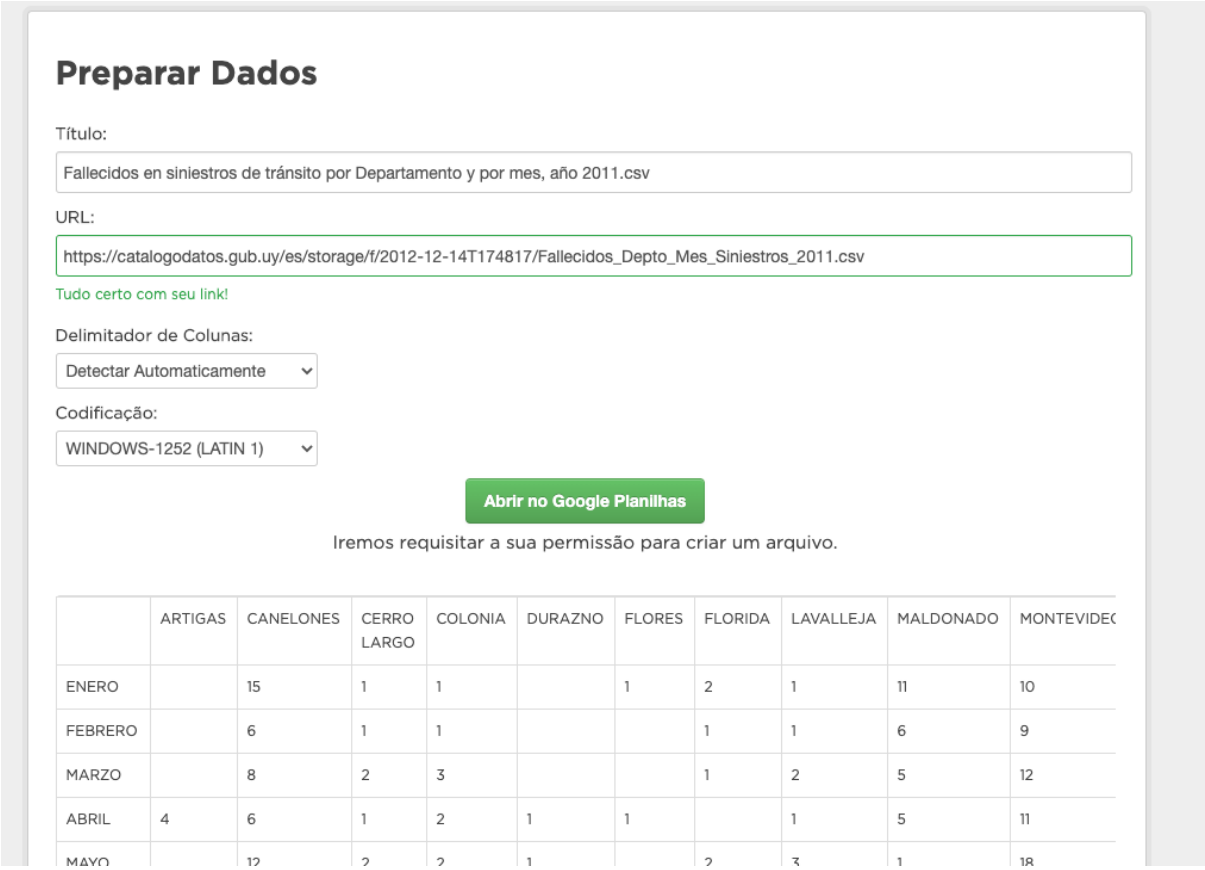
Passo 2 - Preparando os Dados

Nos arquivos suportados, aparecerá a opção: “Abrir com Google Planilhas”.

Figura 4.1.3.7: Integração com o Google Planilhas

Você será levado para uma página já com alguns parâmetros preenchidos. É possível alterar o título da planilha entre outras opções disponíveis. Quando estiver satisfeito com as configurações, aperte em “Abrir no Google Planilhas”.

Figura 4.1.3.8: Painel para preparação dos dados

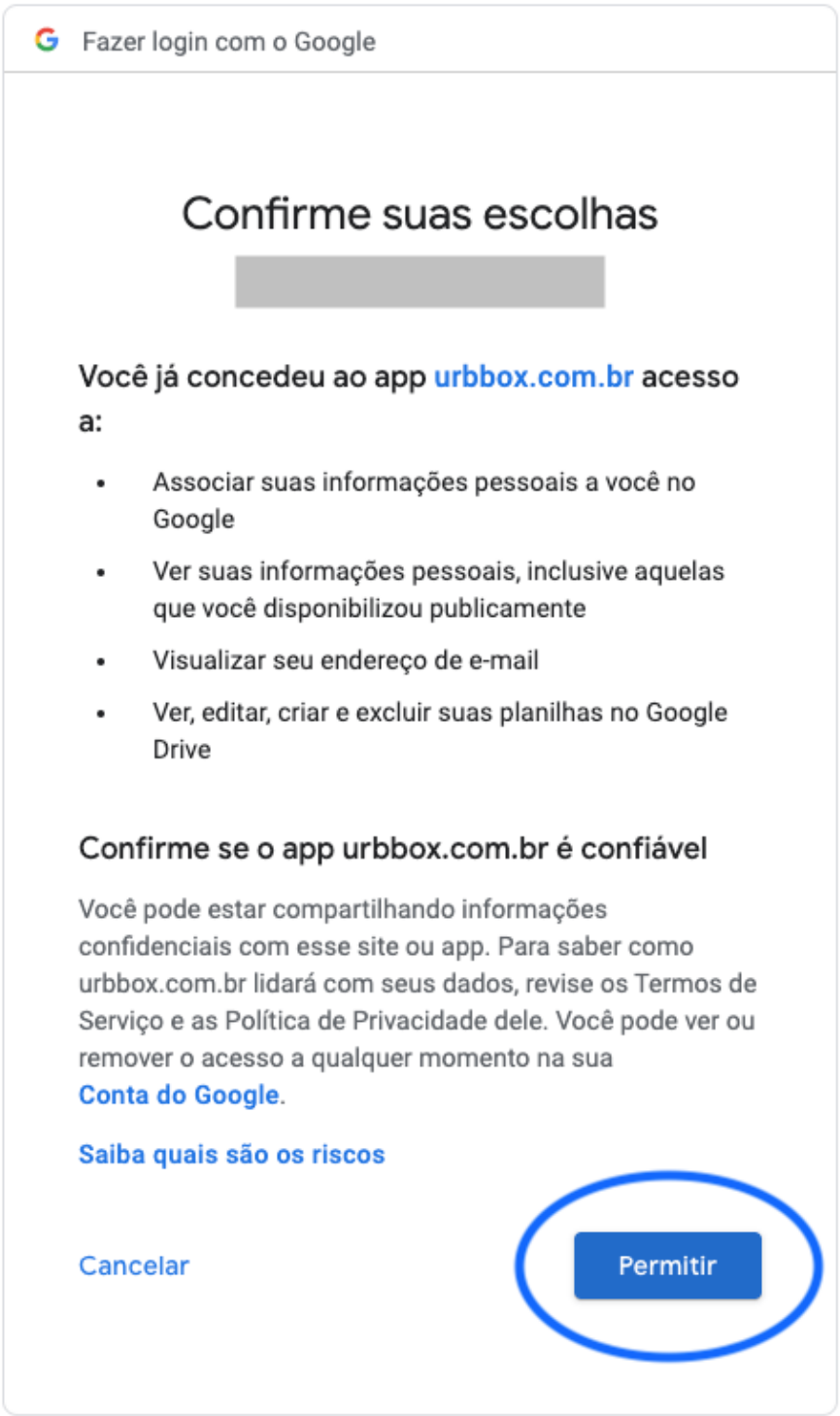


	ARTIGAS	CANELONES	CERRO LARGO	COLONIA	DURAZNO	FLORES	FLORIDA	LAVALLEJA	MALDONADO	MONTEVIDEO
ENERO		15	1	1		1	2	1	11	10
FEBRERO		6	1	1			1	1	6	9
MARZO		8	2	3			1	2	5	12
ABRIL	4	6	1	2	1	1		1	5	11
MAYO		12	2	2	1		2	3	1	18

Passo 3 - Autorizando a Criação da Planilha

Para que o sistema consiga criar uma planilha no Google Planilhas, é necessário que o usuário possua uma conta “@gmail.com” e autorize o acesso a ela.

Figura 4.1.3.9: Autorização de acesso Gmail

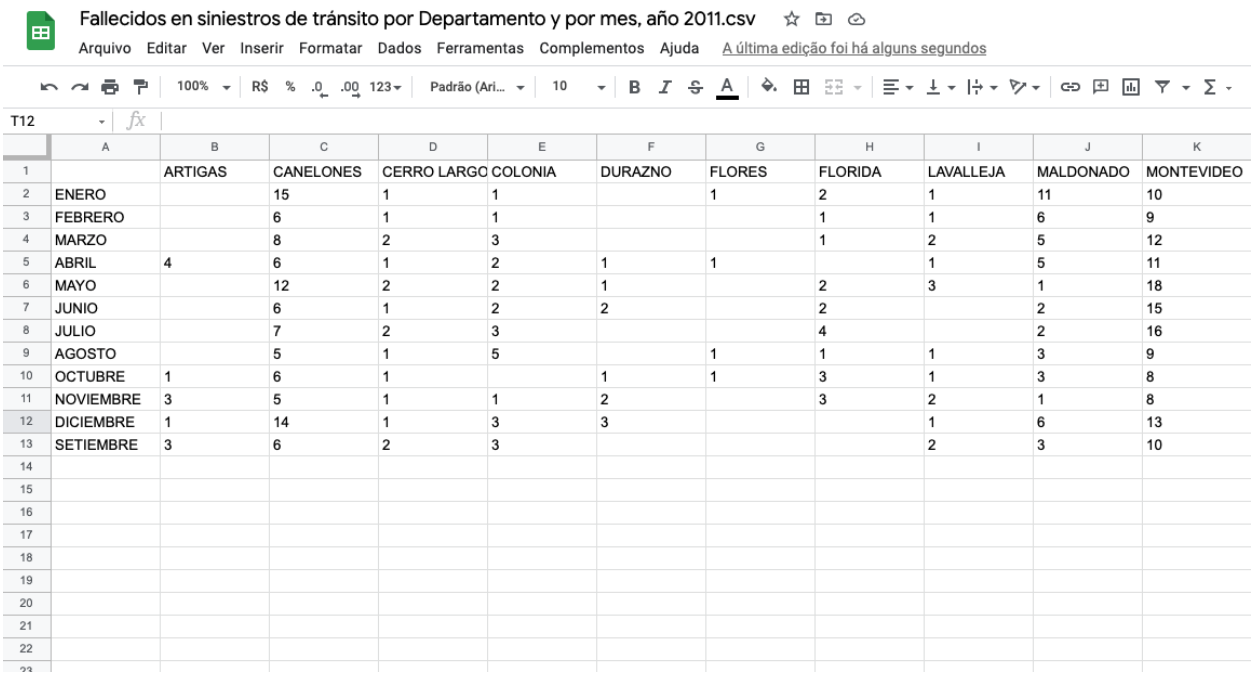


Para isso, clique em “Permitir”.

Passo 4 – Criando uma Visualização da Planilha

Pronto! Agora você já tem acesso a uma cópia dos dados disponibilizados em uma planilha. Aqui pode-se prosseguir com as análises da mesma forma que se trabalha no conhecido "Excel".

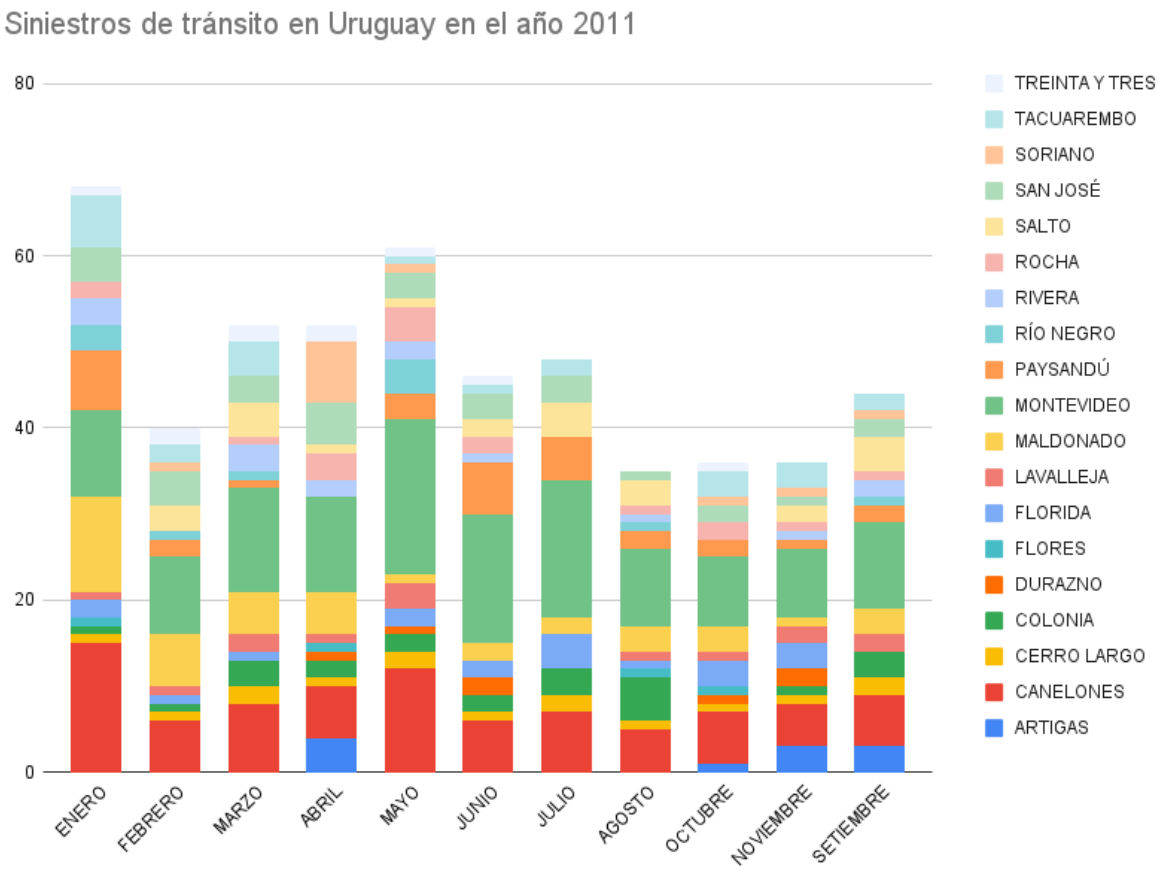
Figura 4.1.3.10: Trabalhando com a planilha de dados



	A	B	C	D	E	F	G	H	I	J	K
1		ARTIGAS	CANELONES	CERRO LARGO	COLONIA	DURAZNO	FLORES	FLORIDA	LAVALLEJA	MALDONADO	MONTEVIDEO
2	ENERO		15	1	1		1	2	1	11	10
3	FEBRERO		6	1	1			1	1	6	9
4	MARZO		8	2	3			1	2	5	12
5	ABRIL	4	6	1	2	1	1		1	5	11
6	MAYO		12	2	2	1		2	3	1	18
7	JUNIO		6	1	2	2		2		2	15
8	JULIO		7	2	3			4		2	16
9	AGOSTO		5	1	5		1	1	1	3	9
10	OCTUBRE	1	6	1		1	1	3	1	3	8
11	NOVIEMBRE	3	5	1	1	2		3	2	1	8
12	DICIEMBRE	1	14	1	3	3			1	6	13
13	SETIEMBRE	3	6	2	3				2	3	10
14											
15											
16											
17											
18											
19											
20											
21											
22											

Serão selecionadas as células da planilha e criado um gráfico a partir delas.

Figura 4.1.3.11: Gráficos gerados a partir dos dados



Passo 5 – Publicando um Gráfico

Selecione a opção “Publicar gráfico”.

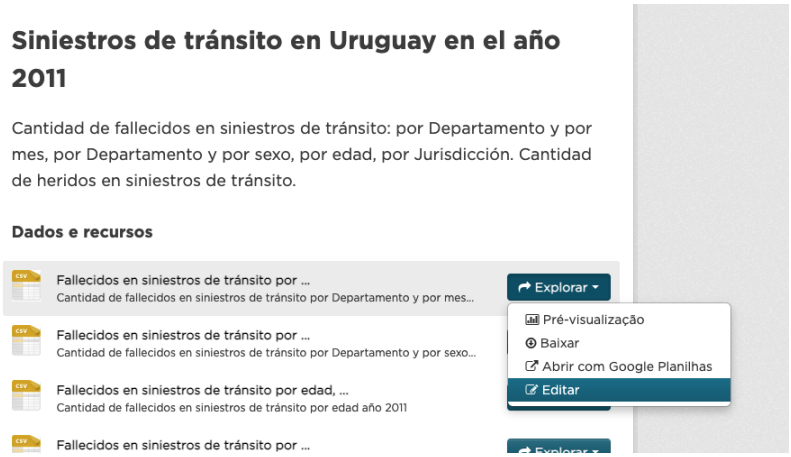
Figura 4.1.3.12: Publicando as figuras e/ou tabelas geradas

Após publicar, copie o link exibido pela ferramenta do Google Sheets. Este link deverá ser usado para compartilhar esta visualização com o Dataurbe.

Passo 6 – Criando uma Nova Visão para o Recurso

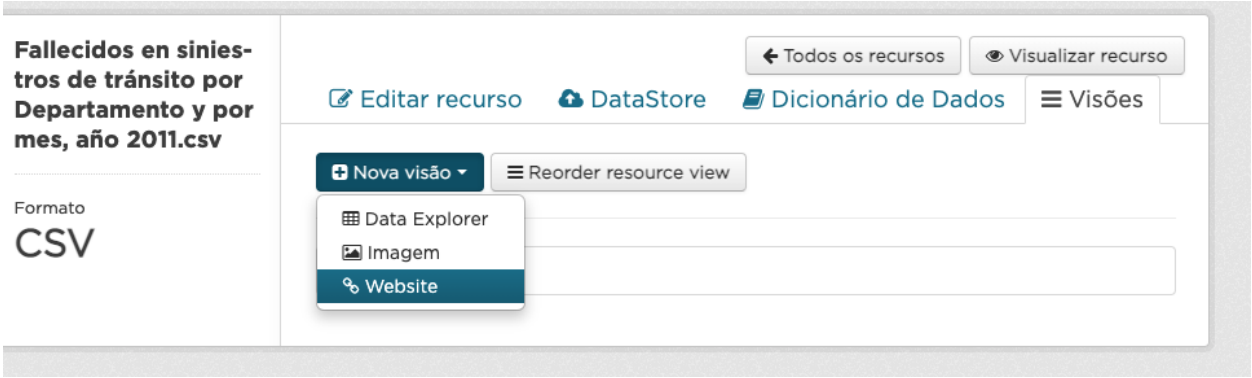
As cidades que possuírem acesso autorizado ao conjunto de dados poderão visualizar a opção “Editar” exibida na seguinte imagem.

Figura 4.1.3.13: Funcionalidade de edição dos dados



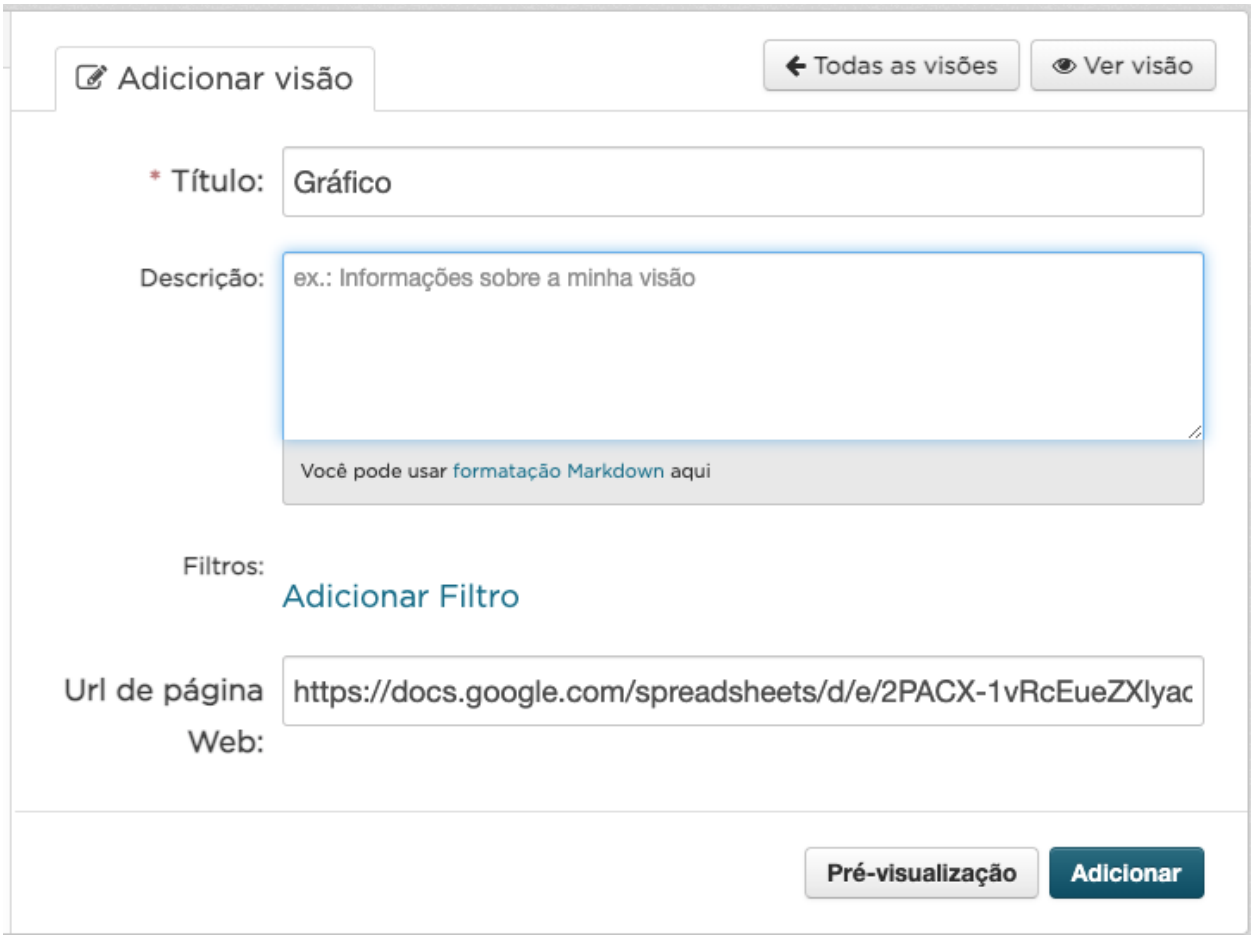
Selecione a aba “Visão”, clique em “Nova visão” e por fim clique em “Website”.

Figura 4.1.3.14: Funcionalidades de visualização disponíveis



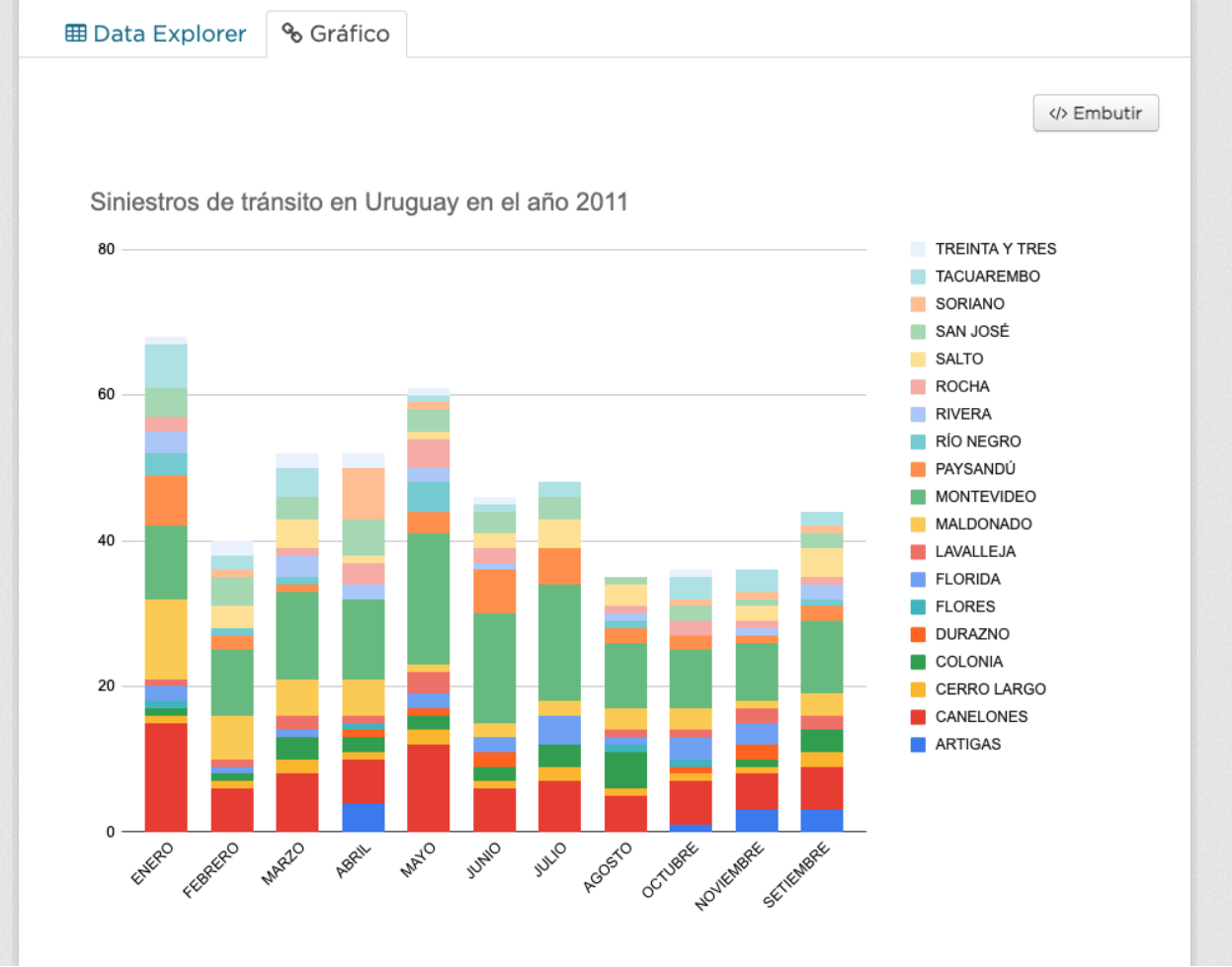
Adicione um título e no campo “Url de página Web” cole o link do gráfico.

Figura 4.1.3.15: Disponibilizando as figuras e/ou tabelas ao público



Clique em “Adicionar” e pronto. Agora os cidadãos poderão visualizar os dados de uma forma mais clara e interativa.

Figura 4.1.3.16: Aba contendo a figura e/ou tabela gerada



Manual - Procedimento Avançado

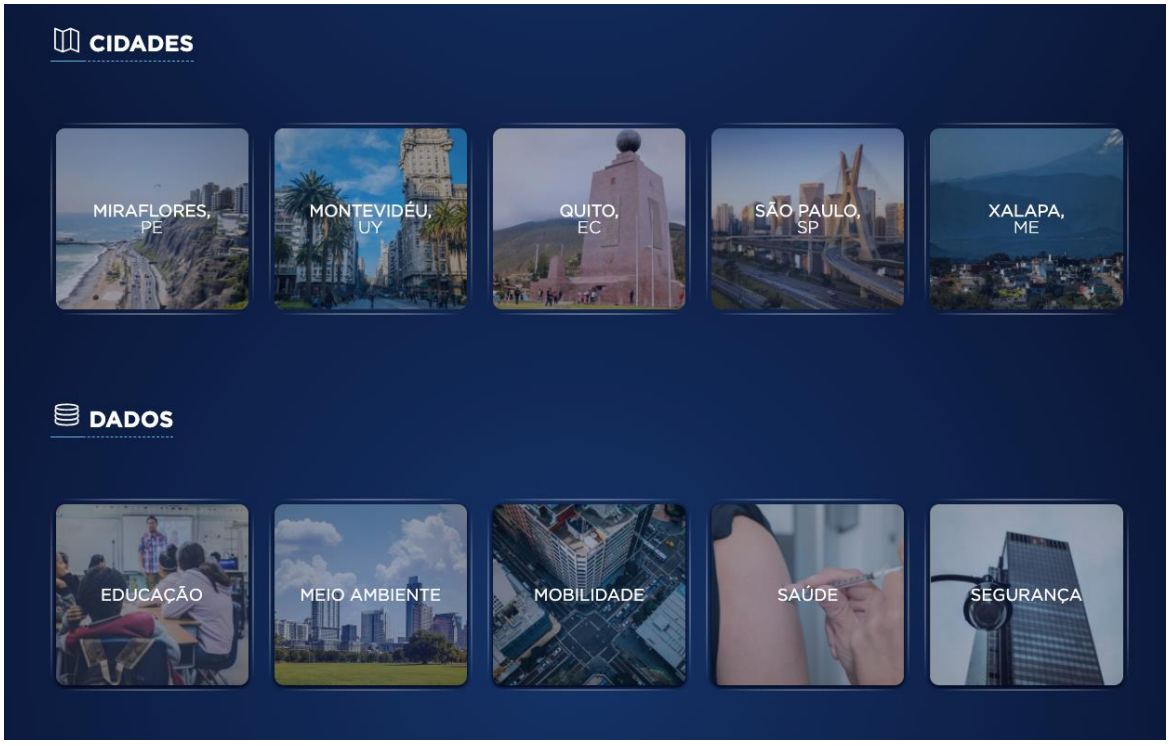
O seguinte processo de visualização de dados é semelhante em todos os casos até o **Passo 3**. Após essa etapa, é opção do leitor escolher a ferramenta de *Business Intelligence* ideal para visualização da tabela de dados. Neste tutorial, explica-se como funciona o processo utilizando o *AWS Quick Sight.*

Passo 1 - Buscando uma Base de Dados

O primeiro passo é encontrar uma base de dados que queira visualizar. Para isso, acesse o site https://dataurbe.appcivico.com/.

Na página você poderá buscar por dados de cada uma das cidades, ou pelas categorias de Educação, Meio Ambiente, Mobilidade e etc.

Figura 4.1.3.17: Painel inicial do DataUrbe



Você também pode pesquisar por conjunto de dados digitando na barra de busca.

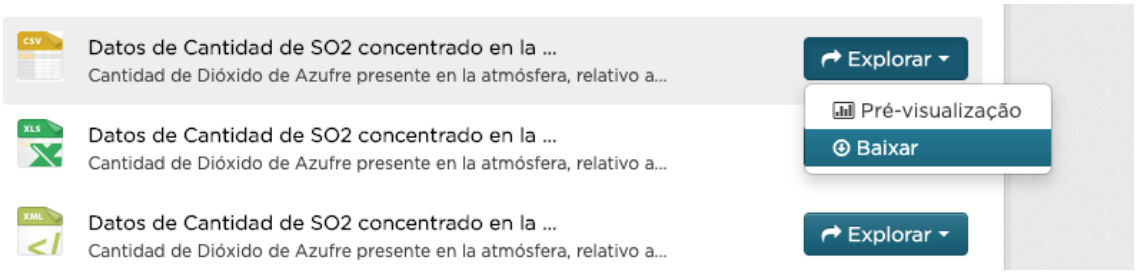
Figura 4.1.3.18: Busca de conjuntos de dados

Passo 2 - Baixando a Base de Dados

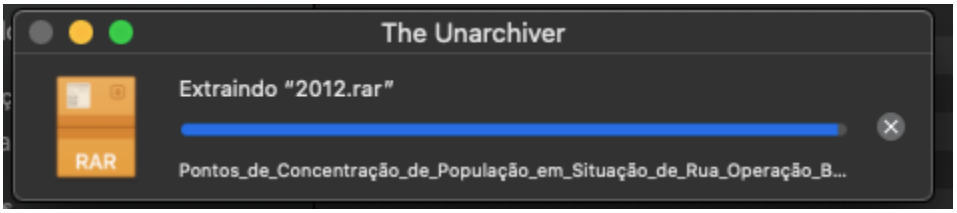
Ao escolher uma base de dados, você precisará baixar os dados para seu computador.

De preferência, escolha uma representação em CSV[11].

Figura 4.1.3.19: Download do conjunto de dados



Às vezes os arquivos estão compactados. Nesses casos você precisará descompactá-los utilizando ferramentas como WinRAR (no Windows) ou The Unarchiver (no MacOS).



[11] O padrão CSV é ideal para inúmeros casos pelo seu formato enxuto baseado em texto não formatado. Ele é simples de entender, editar, escrever e é suportado por todas as ferramentas de visualização.

Passo 3 - Preparando Base para Visualização

O *AWS Quick Sight* e a maioria das ferramentas de BI no mercado podem ler arquivos do tipo:

- CSV: Comma-separated values
- TSV: Tab-separated values
- CLF: Common Log Format
- ELF: European Location Framework
- XLSX: Microsoft Excel
- JSON: JavaScript Object Notation

Se após descompactar os dados, a base não estiver em algum desses formatos, precisaremos utilizar algumas ferramentas de transformação.

Nos passos 3.1 e 3.2 explicam-se dois casos possíveis de tratamento de dados.

Passo 3.1 - Transformando Arquivos Shapefile em CSV

Um caso comum de transformação é quando os arquivos se encontram no padrão Shapefile. Geralmente são casos que envolvem dados de geolocalização.

Figura 4.1.3.20: Exemplo de arquivos em formato Shapefile

Para isso, cria-se um pequeno script **Python 3**, que pode ser adaptado para transformar o seu arquivo:

```
import geopandas as gpd
from csv import QUOTE_ALL

# Efetua a leitura do arquivo
df = gpd.read_file('CAMINHO/PARA/ARQUIVO.shp')

# Cria novo arquivo com a extensão .CSV
df.to_csv('ARQUIVO_TRANSFORMADO.csv', sep=',', index=False, header=True,
quoting=QUOTE_ALL)
```

Passo 3.2 - Convertendo para UTF-8

Em alguns casos, são baixados datasets no formato CSV, mas o encoding do arquivo está em outro formato. Na maioria das vezes estão em LATIN 1 (ISO/IEC 8859-1).

Existem várias formas para converter enconding de arquivos. Esse exemplo usando uma ferramenta chamada **iconv.** Ela já vem instalada em sistemas operacionais Unix como MacOS ou Ubuntu.

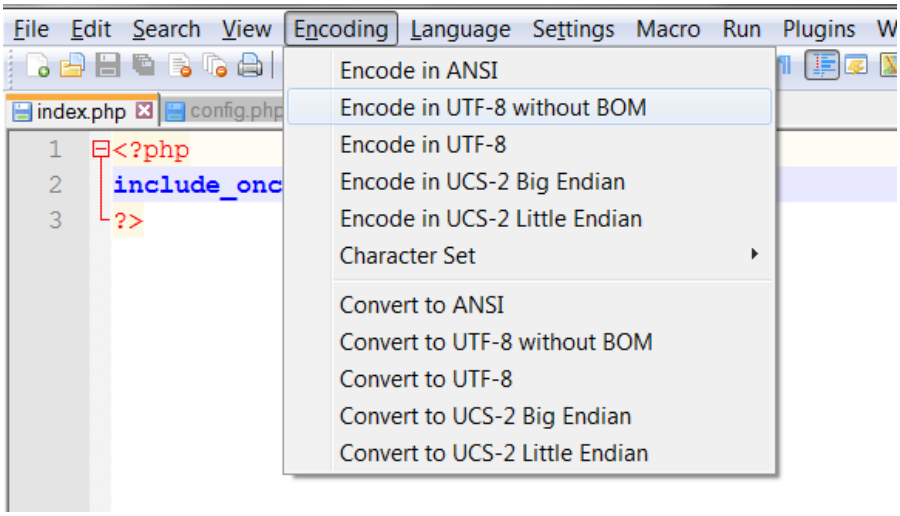
Na linha de comando (terminal), digite:

```
iconv -f LATIN1 -t UTF8 caminho/para/arquivo.csv >
destino_convertido_para_utf8.csv
```

No Windows, você pode utilizar um software chamado Notepad++ (https://notepad-plus-plus.org/)

Vá em “Encoding”, depois em “Encode in UTF-8 without BOM” (Converter para UTF-8 sem BOM). Depois vá em “File” (Arquivo) e depois em “Save” (Salvar).

Figura 4.1.3.21: Alterando o 'Encoding' dos dados

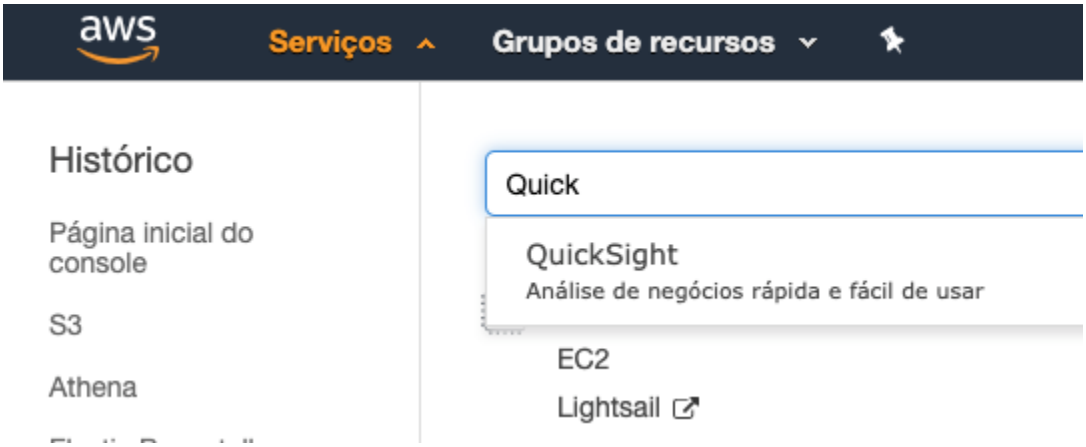


Passo 4 - Criando uma Análise no Quick Sight

Acesse sua conta na AWS Management Console (https://console.aws.amazon.com/).

Vá no menu superior chamado "Serviços" e pesquise por Quick Sight. Escolha a resposta que aparecer.

Figura 4.1.3.22: Acessando o QuickSight



Com o *QuickSight* aberto, à direita clique em "Nova análise".

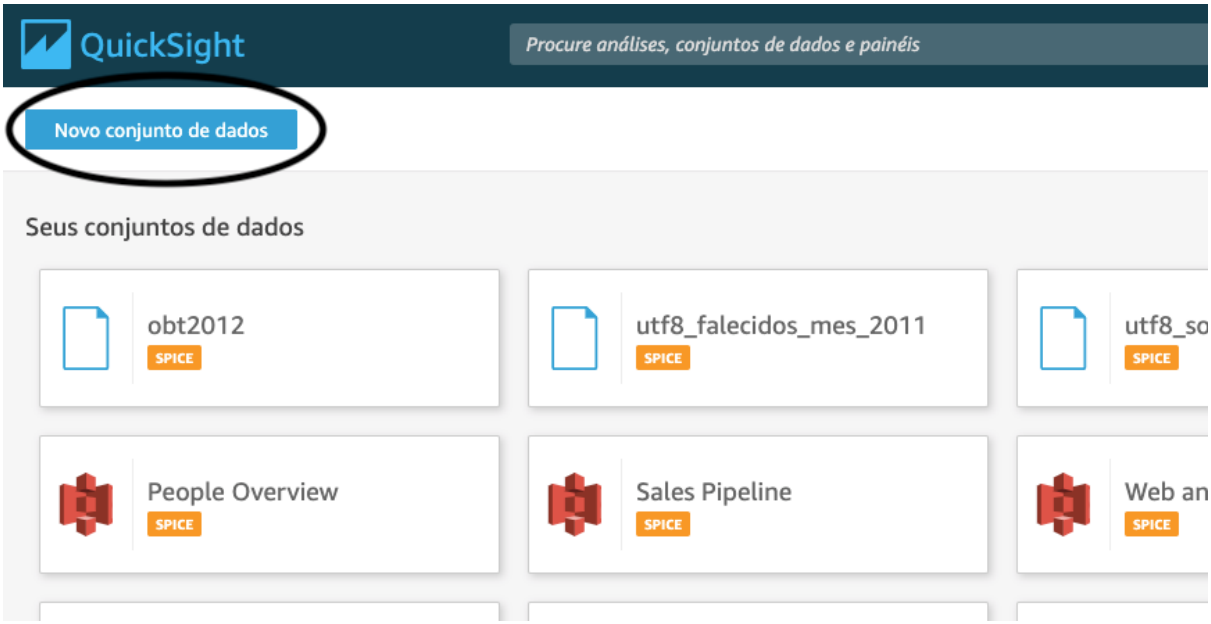
Figura 4.1.3.23: Gerando uma nova análise

Você pode criar várias análises diferentes para um mesmo conjunto de dados.

Passo 4.1 - Criando um Conjunto de Dados

Cria-se um novo conjunto de dados para a nossa base.

Figura 4.1.3.24: Adicionando um conjunto de dados



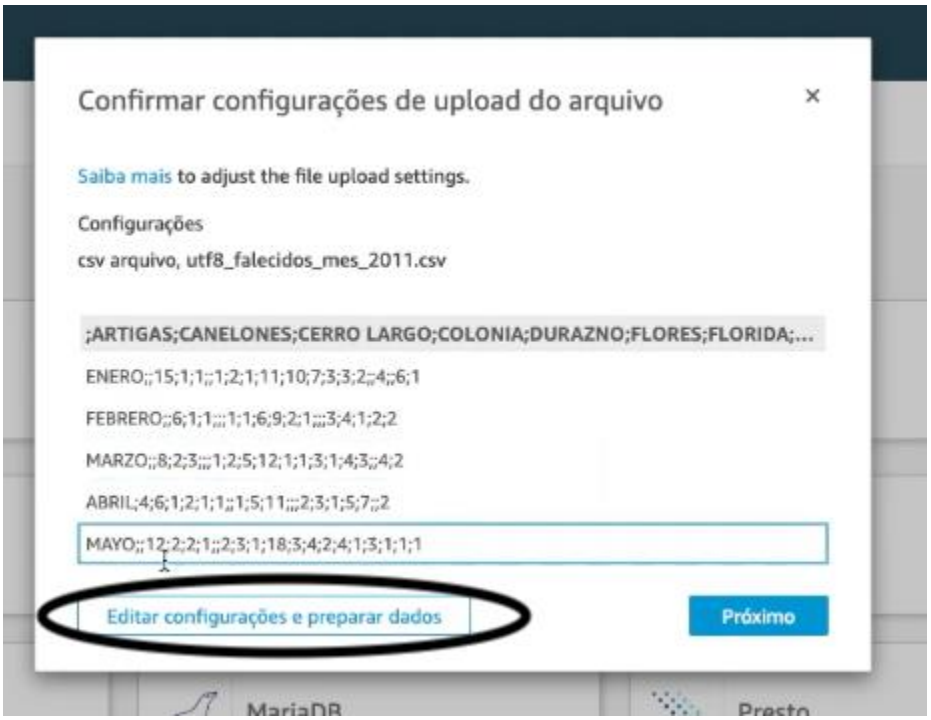
Selecione a opção “Fazer upload de um arquivo”:

Passo 4.2 - Corrigindo alguns Possíveis Problemas

Às vezes o arquivo está separado por “;” (ponto-e-vírgula) ao invés de vírgula. Ou você precisa renomear alguma coluna.

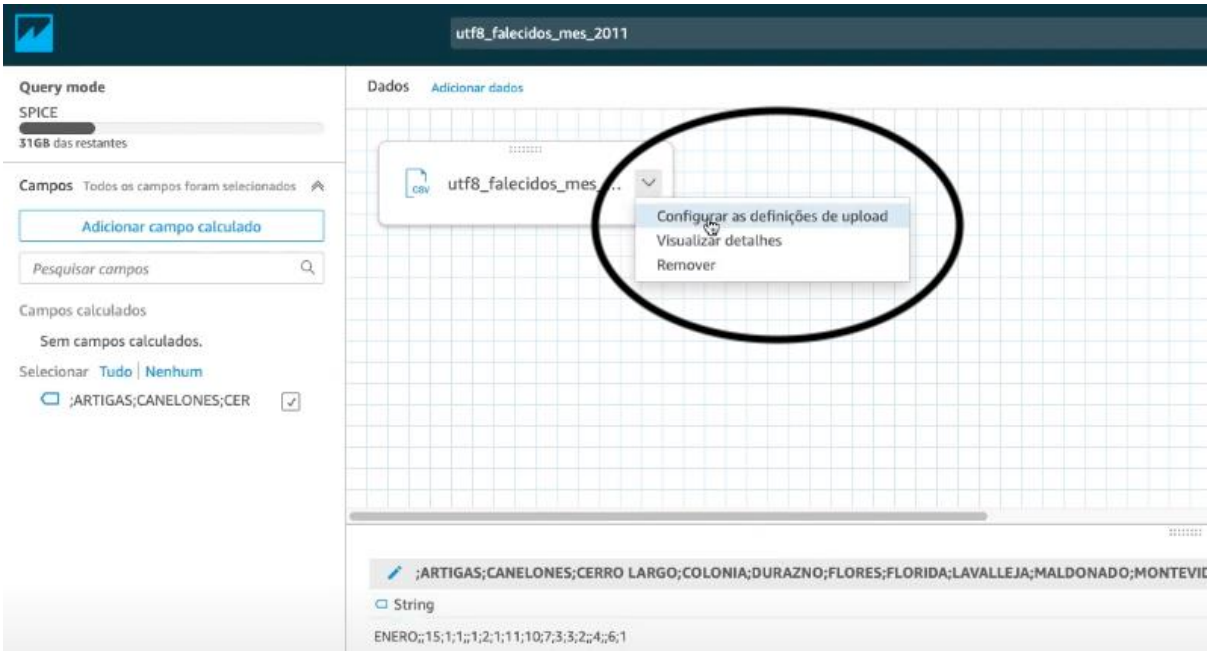
Nesse caso, clique em “Editar configurações e preparar dados”

Figura 4.1.3.25: Configurando os dados

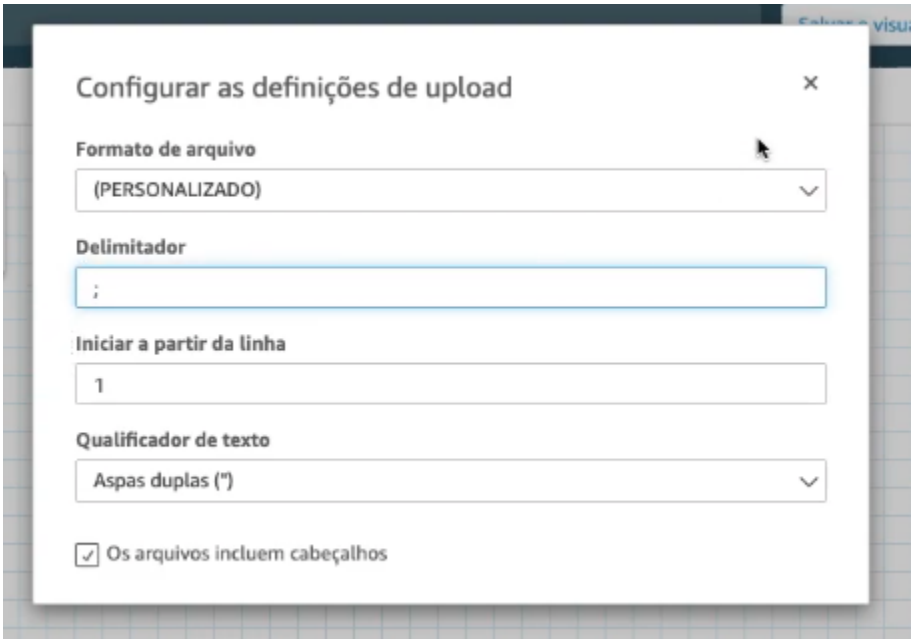


Você será levado para uma tela intermediária onde será possível editar informações do arquivo enviado.

Figura 4.1.3.26: Definindo especificidades para upload

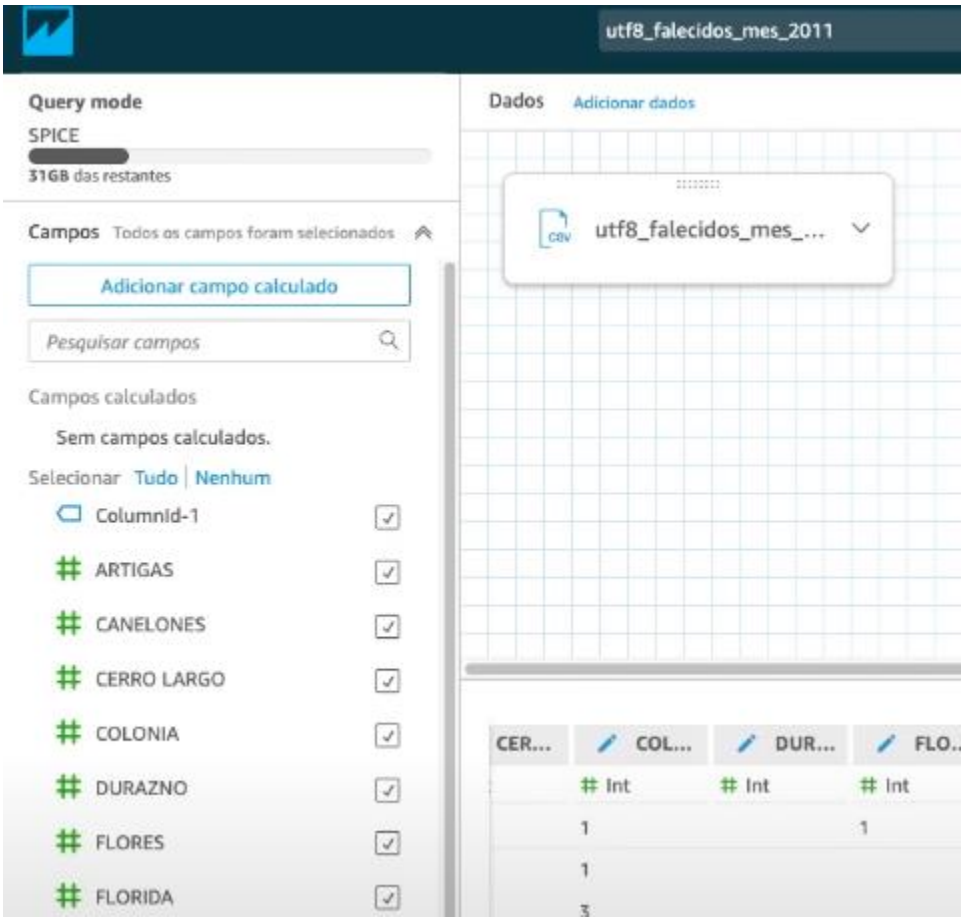


Nessa página você pode consertar essas opções:



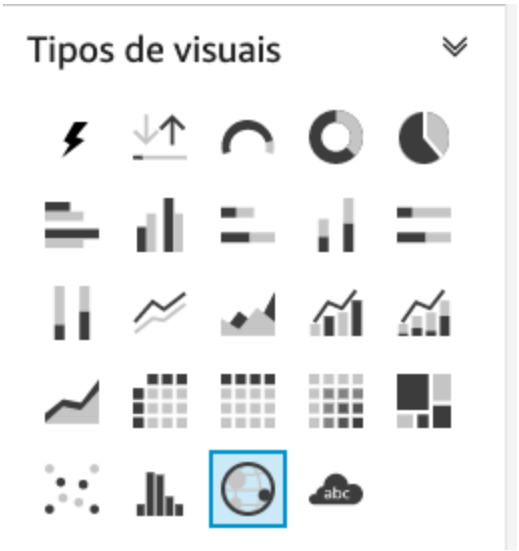
Feito isso, as colunas deverão ser corretamente reconhecidas.

Para prosseguir, clique em “Salvar e Visualizar”



Passo 5 - Selecionando a Visualização Desejada

Dependendo dos seus dados, você pode escolher uma visualização específica.

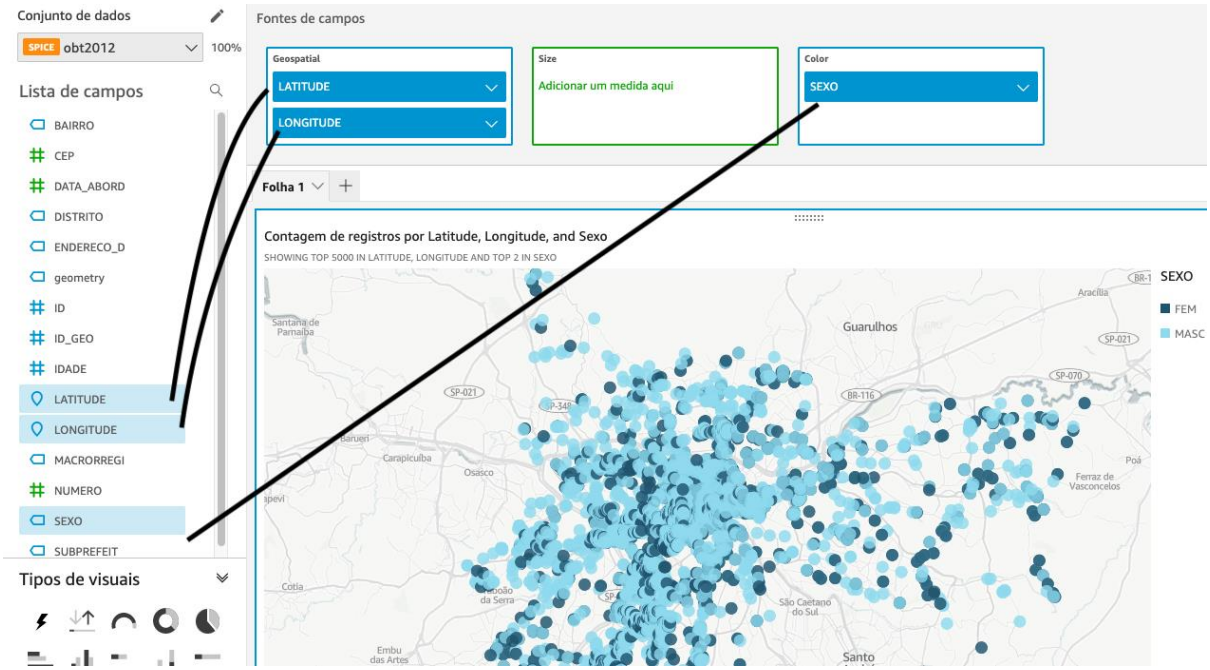


Ao selecionar um tipo de visualização, aparecerá um menu para organizar quais colunas farão parte do gráfico desejado.

No exemplo abaixo têm-se colunas representando as coordenadas geográficas e algum parâmetro para definir a cor dos pontos.

Arraste as colunas para as fontes de campos respectivas.

Figura 4.1.3.27: Painel para criação de gráficos

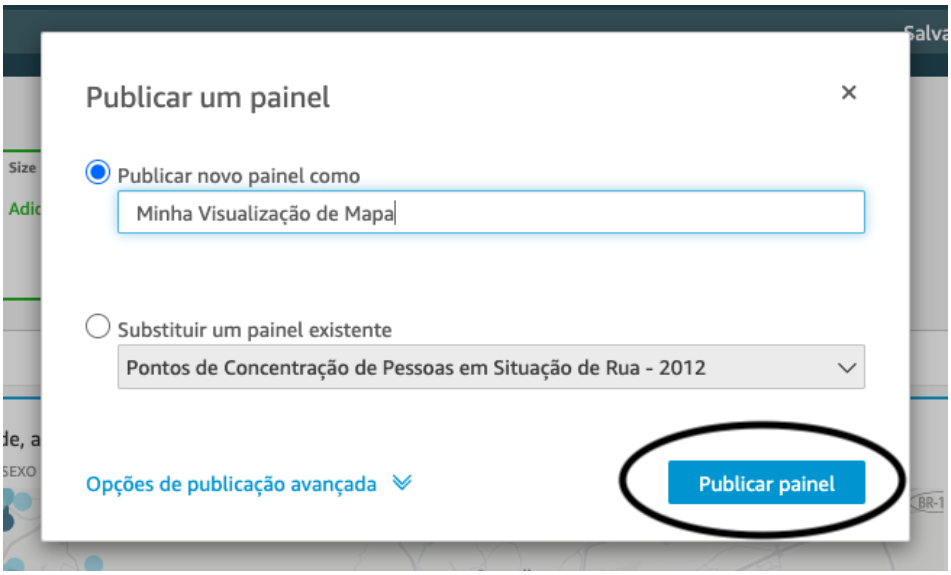


Passo 6 - Compartilhando o Dashboard

Para dar acesso aos outros usuários cadastrados deve-se publicar um painel. Para isso, basta clicar em “Compartilhar” e em seguida “Publicar painel”.

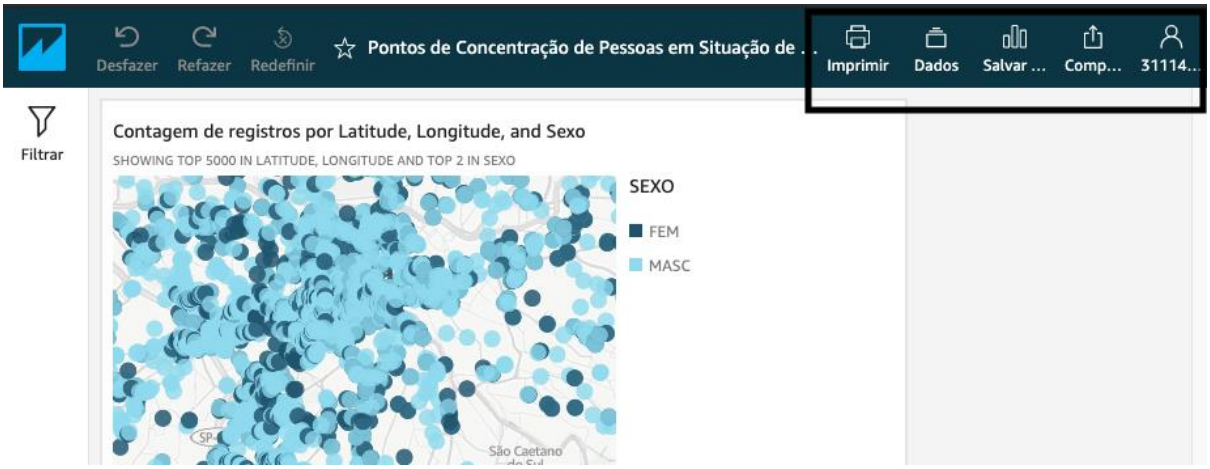
Dê um nome para o seu painel e clique em “Publicar painel”.

Figura 4.1.3.28: Publicando os dados



Após a publicação você poderá configurar o acesso ao usuário, imprimir ou enviar o relatório por e-mail.

Figura 4.1.3.29: Visualização e configuração dos gráficos e/ou tabelas criadas



4.2 Estimativa de poluição de ar

Os exercícios de *Design Thinking* realizados em 2019 revelaram preocupação das cidades com as questões ambientais, especificamente, àquelas relacionadas à poluição do ar, problema comum nos centros urbanos com efeitos adversos diretos na saúde e na mortalidade da população. Tendo em vista a importância do tema e o alinhamento geral do projeto em incentivar o uso de dados massivos para apoiar o planejamento urbano e a formulação e implementação de políticas públicas baseadas em evidências, a presente seção apresenta o desenvolvimento de um estudo de estimação de poluição de ar fazendo uso dos dados do Waze.

A poluição atmosférica é uma mistura de partículas sólidas e gases despejados no ar, que são nocivos tanto à saúde humana quanto ao meio ambiente. De acordo com a Organização Mundial da Saúde (OMS, 2018), estima-se que 4,2 milhões de mortes ocorrem todo ano devido à exposição à poluição atmosférica e 91% da população mundial vive em lugares onde a qualidade do ar está abaixo dos limites definidos. A exposição a poluentes – gases poluentes e material particulado – afetam especialmente a saúde de crianças, idosos e dos que padecem de doenças respiratórias ou cardiovasculares crônicas. Além de impactar a morbidade respiratória, é causador de risco de outros problemas de saúde, como baixo peso ao nascer, incidência e mortalidade por câncer, partos prematuros, entre outros (OMS 2003, Dapper, Nikoli and Zanini 2016, Quito s.d.)

Assim, a poluição atmosférica tornou-se uma questão importante por afetar a população mundial em grande escala e diversos esforços têm sido tomados a fim de reduzir a emissão de poluentes e predizer a qualidade do ar de diversos locais no mundo.

Grande parte da poluição atmosférica vem do uso e produção de energia. Em grandes centros urbanos, os automóveis movidos a gasolina e diesel são os principais emissores de gases poluentes. Além do aumento da quantidade de automóveis em circulação nos últimos anos, a poluição é agravada pelo tempo gasto nos congestionamentos que aumenta a queima desses combustíveis.

Especificamente, o processo de combustão do motor tende a emitir dióxido de nitrogênio (NO_2) e óxido nítrico (NO). Embora a emissão de NO seja baixa comparada à de NO_2, ao entrar em contato com o oxigênio do ar, o NO se converte em NO_2. O NO_2 aumenta a sensibilidade a doenças respiratórias como bronquite e asma; além disso, alguns estudos têm relacionado a

exposição a NO_2 com a incidência de doenças neurológicas como doença de Alzheimer, doença de Parkinson e autismo. O NO_2 também contribui para a formação de outros poluentes como ozônio (O_3) e material particulado fino ($PM_{2.5}$).

Dentre as cidades do projeto, os dados de São Paulo, Quito e Lima chamam atenção. Na Região Metropolitana de São Paulo, os veículos foram responsáveis em 2019 por 96,4% de Monóxido de Carbono (CO), 72,8% de hidrocarbonetos (HC), 64,9% de Óxidos de Nitrogênio (NO_X), 40% de Material Particulado menor que 10 µm (PM_{10}) e 11,5% de Óxidos de Enxofre (SO_X), segundo estimativas da CETESB (2020). A contribuição das fontes móveis também é alta em Quito. De acordo com o relatório anual do Distrito Metropolitano de Quito (2014) referente à 2011, veículos foram responsáveis por 98,5% na emissão de CO, 24,5% em SO2, 69,2% em NO_X, 33,2% em PM10 e 31,1% compostos orgânicos voláteis não metânicos (COVNM). Por fim, segundo a Comissão Multisetorial para a Gestão da Iniciativa de Ar Limpo para Lima e Callao (2019), as fontes móveis foram responsáveis por 58% das emissões de $PM_{2.5}$ da região em 2016.

Políticas públicas têm sido arquitetadas e implementadas para enfrentar este problema premente. O conhecimento detalhado dos efeitos dos poluentes do ar é um pré-requisito para o desenvolvimento de políticas eficazes para reduzir o impacto adverso da poluição do ar (WHO, 2003).

Contudo, muitas cidades carecem de instrumentos de monitoramento de qualidade de ar, insumos básicos para a elaboração de políticas mais eficientes. A maioria das cidades não possui um bom sistema de monitoramento de ar, sendo Ásia e África os continentes com medição mais precária e, mesmo países com renda mais alta como os Estados Unidos a densidade de monitoramento é baixa (Carabetta 2019).

A distribuição das estações de monitoramento nas áreas urbanas que teria o potencial de permitir a avaliação de uma concentração regional dos poluentes não é densa o suficiente para detectar a heterogeneidade espacial da dispersão dos poluentes. Além disso, é baixa a capacidade de captação das variações regionais.

Muitos trabalhos visando principalmente a saúde pública, fazem uso de métodos indiretos de medição considerando, por exemplo, o tráfego veicular como método de avaliação da exposição à poluição (Habermann, Medeiros and Gouveia 2011)

Os métodos de estimação de poluição mais conhecidos são Location Based, Interpolação, Dispersão, Land Use Regression (LUR) que na literatura são usados isoladamente ou combinados entre si (Carabetta 2019, Habermann, Medeiros and Gouveia 2011). Muitos deles usam como proxy dados estáticos como censo, matrizes origem-destino, uso do solo ou congestionamento de tráfego. Contudo, além de terem um menor dinamismo temporal, os dados necessários podem não estar disponíveis em países/cidades menos desenvolvidos. Alguns modelos têm maior complexidade, o que envolve um processamento maior de dados e esforço computacional.

O presente trabalho pretende colaborar com a criação de um sistema capaz de viabilizar a medição indireta da qualidade de ar, que possa ser alternativa para locais em que não se afere a contaminação de ar por meio de sensores. Seguindo a linha do projeto “Big Data para Desenvolvimento Urbano Sustentável” e com base em uma referência inovadora de estimação de poluição utilizando dados massivos do aplicativo Waze utilizado por Carabetta (2019), replicamos a mesma metodologia nas cidades participantes do projeto.

Naquele trabalho, o autor propõe uma metodologia aberta, *crowdsourced* e escalável para modelar a poluição do ar espacial em cidades em todo o mundo. Foram usados os dados do Waze e do Open Street Maps para construir um conjunto de recursos destinados a modelar as emissões dos carros em (grandes) cidades para treinar um modelo usando dados captados de uma região de 30 km2 de Oakland, Califórnia, nos Estados Unidos da América.

O objetivo é mostrar, a partir de um estudo acadêmico inovador, como é possível fazer utilização de dados massivos produzidos pelo aplicativo Waze para ajudar a produzir informação relevante para o uso de gestores públicos. O sistema desenvolvido traz uma contribuição inicial a partir dos casos das cidades envolvidas neste projeto e tem como principal objetivo incentivar gestores a fazer uso de dados disponíveis como os do Waze que combinados com outros métodos testados na literatura podem servir como insumos relevantes para o planejamento urbano-ambiental.

4.2.1 Metodologia

A disponibilidade de registros históricos sobre a qualidade do ar e os dados sobre a presença de emissores de poluentes tem feito com que muitos trabalhos tenham sido propostos nessa área. Alguns desses trabalhos utilizam modelos de aprendizado de máquina para predizer a qualidade

do ar em alguns lugares do mundo, como por exemplo: Califórnia (Castelli, et al. 2020), Taiwan (Fi-John, et al. 2020), Bangladesh (Shahriar, et al. 2020) e Shangai (Ma, et al. 2020). Em aprendizado de máquina, um modelo é ajustado através de dados históricos para os preditores (por exemplo, número de carros e fábricas em uma região) e dados para a variável alvo (por exemplo, concentração de NO_2 na região), após essa primeira fase chamada de treinamento, novos dados são apresentados para os preditores com o intuito de estimar os novos valores para a variável alvo.

Neste trabalho foram utilizados dados do aplicativo Waze para os preditores e como variável alvo utilizamos registros do nível de NO_2 obtidos por centros de monitoramentos para 4 diferentes cidades (Tabela 4.2.1.1), ambos considerando o período de abril de 2019. Os modelos de aprendizado de máquina utilizados foram previamente treinados com dados para a cidade de Oakland. As predições foram ajustadas para as cidades de interesse e visualizadas através de uma interface gráfica que mostra o nível predito para o poluente sobre um mapa da região analisada.

Tabela 4.2.1.1: Número de Estações de Monitoramento com Dados Disponíveis e Fonte de Dados por Cidade

Cidade	Número de estações	Fonte de dados
Lima	8	Servicio Nacional de Meteorología e Hidrología del Perú – SENAMHI https://www.senamhi.gob.pe/?p=calidad_del_aire-estadistica&e=112265
Montevidéu	4	Red de Monitoreo de la Calidad del Aire de Montevideo https://montevideo.gub.uy/areas-tematicas/ambiente/calidad-del-aire/informes-semanales-2019
Quito	7	Secretaría de Ambiente del Municipio del Distrito Metropolitano Quito Em http://www.quitoambiente.gob.ec vá em: Políticas y Planeación Ambiental / Red de Monitoreo / Descarga datos historicos

São Paulo	9	QUALAR - Sistema de Informações da Qualidade do Ar da CETESB (Companhia Ambiental do Estado de São Paulo). https://qualar.cetesb.sp.gov.br/qualar/exportaDadosAvanc.do?method=filtrarParametros

Fonte: Elaboração Própria
* Dados coletados em 04/2019

4.2.2 API Waze

O Waze é um aplicativo utilizado em celulares e *tablets* que permite a navegação por GPS. Durante a utilização do Waze é possível obter informações em tempo real, como por exemplo, a situação das rodovias, o tempo de trajeto, o mapa de trânsito e alertas de incidentes. O usuário do Waze também pode incluir dados atualizando as informações sobre as rodovias. Os principais tipos de dados no Waze são os dados de alerta e os dados de congestionamento. Os dados de alerta consistem em um relatório do usuário para uma latitude e longitude específicas. Os dados de congestionamento identificam rodovias congestionadas através da comparação dos dados históricos de posição e velocidade obtidos via GPS dos usuários (Carabetta 2019).

O acesso aos dados se dá através de uma API disponibilizada pelo Waze que possibilita a consulta em tempo real, dado um polígono correspondente a região de interesse. A FGV estabelece o retorno da API através de um banco de dados distribuído na infraestrutura Amazon AWS desde outubro de 2018. Através desse banco de dados é possível obter dados de alerta e congestionamentos atualizados para diferentes cidades da América Latina.

Neste trabalho foram utilizados dados de congestionamento do Waze para estimar o nível de poluição no ar. Os dados de congestionamento, ao contrário dos dados de alerta, são independentes do engajamento ativo dos usuários, o que torna esses dados mais confiáveis para a estimativa do modelo. Além disso, as informações contidas nos dados de congestionamento são possivelmente mais relevantes para a estimativa do nível de poluição em uma determinada região.

4.2.3 Modelo de Oakland

Como parte das nossas análises foi utilizado o modelo proposto por Carabetta (2019) para predizer o nível de poluição na cidade de Oakland na Califórnia. Para a construção desse modelo

foram utilizados dados de trânsito obtidos através do API do Waze e dados do nível de três poluentes na atmosfera: Carbono negro (BC), óxido nítrico (NO) e dióxido de nitrogenio (NO_2). A fim de obter um modelo com boa performance foram avaliados diferentes algoritmos de regressão e, além disso, foi feita uma busca de parâmetros para maximizar a performance do modelo.

Inicialmente além dos dados de trânsito obtidos pelo API do Waze também foram considerados os dados do Open Street Map (OSM). No entanto, nos testes finais dos algoritmos considerados, foram utilizadas apenas as variáveis correspondentes aos dados do API do Waze. Dentre as 22 variáveis disponibilizadas pela API do Waze foram utilizadas apenas as 12 que estão mais relacionadas às condições do tráfego na região avaliada e não são altamente correlacionadas. Na Tabela 4.2.3.1, mostramos as variáveis utilizadas e a descrição de cada uma delas. Com o auxílio da biblioteca H3[12] fornecida pelo Uber, as regiões analisadas da cidade de Oakland foram divididas em hexágonos e utilizou-se os valores das variáveis para esses hexágonos.

Tabela 4.2.3.1: Variáveis Utilizadas para a Predição do Nível de Poluição em Oakland.

Variável	Descrição
avg_congested_prop	Média do comprimento de um congestionamento
max_length	Comprimento máximo de um congestionamento
avg_speed	Média das velocidades médias nos congestionamentos
min_median_level	Mínima mediana do nível de congestionamento
bool_highway	Indicador de uma rodovia
bool_ramps	Indicador de uma rampa
count_highway	Número de congestionamentos nas rodovias no período
count_streets	Número de congestionamentos nas ruas no período
count_primary	Número de congestionamentos em vias primárias no período
count_secondary	Número de congestionamentos em vias secundárias no

[12] https://eng.uber.com/h3/

	período
count_primary_street	Número de congestionamentos em ruas primárias no período
count_primary_ramps	Número de congestionamentos em rampas primárias no período

Fonte: Elaboração Própria

As variáveis alvo, ou seja, os níveis dos diferentes poluentes, foram obtidas através dos dados de um estudo conduzido pela Aclima Inc.[13] e Google. No estudo, as empresas equiparam dois carros com medidores de poluição atmosférica que circularam por 1 ano nas regiões residenciais, comerciais e industriais da cidade de Oakland na Califórnia. Após os registros, foi feito um processo de redução dos dados e finalmente foi calculada a mediana diária para o ano (2015-2016) e o desvio padrão para 21000 segmentos de rodovias em Oakland. Para as variáveis alvo, as regiões analisadas de Oakland também foram divididas em hexágonos e o valor para cada hexágono corresponde à mediana dos valores para os trechos de rodovia dentro do limite do hexágono considerado.

O conjunto final de dados foi composto por 283 instâncias (283 hexágonos) onde aproximadamente 90% das instâncias (254 hexágonos) foram utilizadas para treino e 10% (29 hexágonos) foram utilizadas para validação. Quatro algoritmos de aprendizagem de máquina foram propostos inicialmente para estimar o nível dos poluentes: regressão linear simples, XGBoost[14], Floresta aleatória e regressão linear múltipla. Para o caso da regressão linear simples, a única variável utilizada foi a presença de autoestrada, que assume valor 1 se o hexágono intersecciona uma autoestrada e 0, caso contrário. Para os outros algoritmos foram utilizadas todas as 12 variáveis (Tabela 4.2.3.1). Para o caso da Floresta aleatória e do XGBoost foram ajustados os parâmetros através de uma busca aleatória por parâmetros dentro de uma faixa definida (Random Search). Para avaliar a performance dos diferentes algoritmos foi utilizado o erro quadrático médio (MSE) nos dados de validação. Os resultados são mostrados na Tabela 4.2.3.2.

Tabela 4.2.3.2: Erro Quadrático Médio (MSE) de cada Algoritmo por Poluente

[13] https://www.aclima.io

[14] Chen, Tianqi, e Guestrin, Carlos. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). 2016.

Algoritmos	BC	NO	NO_2
Regressão linear simples	0,086	95,8	24,0
Regressão linear múltipla	0,089	78,0	25,6
Floresta aleatória	0,058	49,6	18,6
XGBoost	0,047	39,5	16,0

Fonte: Elaboração Própria

É possível verificar que o algoritmo que apresenta a melhor performance é o XGBoost. Sendo assim, esse algoritmo foi escolhido para estimar os níveis dos poluentes no modelo criado para a cidade de Oakland. Os parâmetros utilizados para o modelo XGBoost são mostrados na Tabela 4.2.3.3.

Tabela 4.2.3.3: Valores dos Parâmetros para o XGBoost

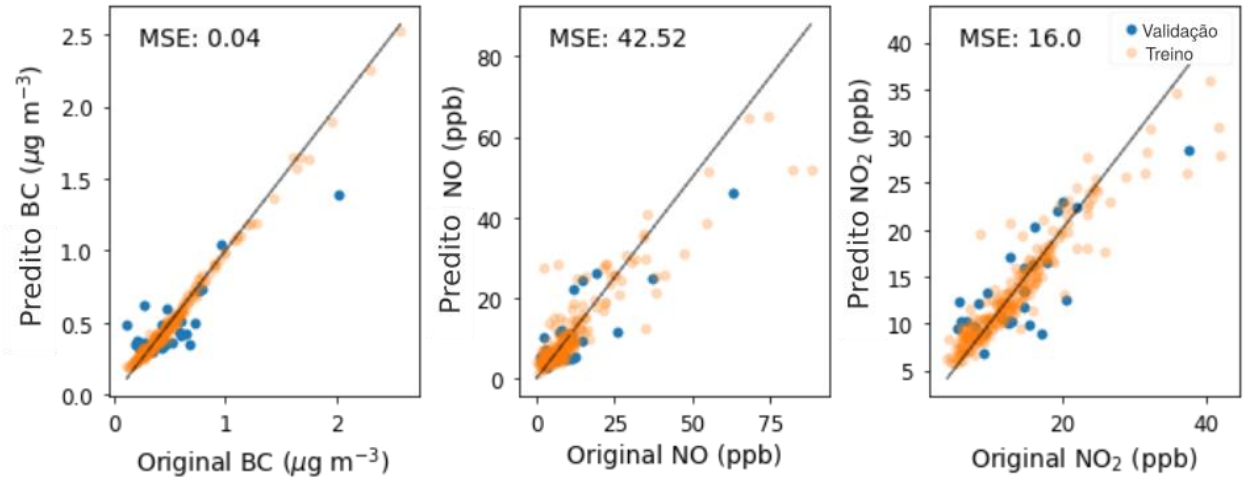
Parâmetro	Valor
colsample_bytree	0,6
eta	0,1
eval_metric	rmse
gamma	0
learning_rate	0,1
max_depth	7
min_child_weight	10
n_estimators	100
reg_alpha	0
reg_lambda	1,0
tree_method	exact

Fonte: Elaboração Própria
* Os outros parâmetros não mostrados na tabela têm valor padrão da biblioteca XGBoost versão 1.4.0 para Python (https://xgboost.readthedocs.io/en/latest/parameter.html)

Na Figura 4.2.3.1, são mostrados gráficos de dispersão dos dados de treino e de validação para as estimativas feitas com o XGBoost. Os códigos para o modelo de Oakland podem ser obtidos

no repositório do github[15]. Os pontos em azul correspondem aos dados de validação e os pontos em laranja correspondem aos dados de treino. Em cada gráfico é mostrado o MSE para a estimativa de cada poluente utilizando o XGBoost.

Figura 4.2.3.1: Gráficos de Dispersão para os Diferentes Poluentes



Fonte: Adaptada da dissertação de Carabetta (2019)

4.2.4 Construção do software

Na construção do software foi utilizado um modelo de aprendizado de máquina desenvolvido anteriormente por Carabetta (2019). Os preditores para este modelo são baseados nos dados de congestionamento do Waze para a cidade de Oakland na Califórnia e a variável alvo é o nível de NO_2 na mesma cidade.

Foi treinado um modelo de regressão XGBoost após realizar uma busca aleatória por parâmetros que maximizam a performance do modelo. O modelo de regressão XGBoost apresentou uma performance melhor que outros modelos com diferentes algoritmos para a regressão. Os códigos para esse modelo estão disponíveis em um repositório no GitHub (https://github.com/JoaoCarabetta/master_thesis).

O objetivo deste trabalho foi estimar o nível de NO_2 em regiões de São Paulo, Quito, Montevidéu e Lima. Sendo assim, os preditores utilizados foram dados do Waze relacionados ao congestionamento para essas cidades. Para obter esses dados, dividiram-se as regiões em

[15] https://github.com/JoaoCarabetta/master_thesis

hexágonos dados pela biblioteca H3 fornecida pelo projeto Uber H3. Em seguida, foram buscados na base de dados os valores para as variáveis independentes relativas aos polígonos correspondentes aos hexágonos definidos para cada região. Além disso, para regiões específicas de cada cidade, foi possível obter o nível de NO_2 registrado por agências locais de monitoramento.

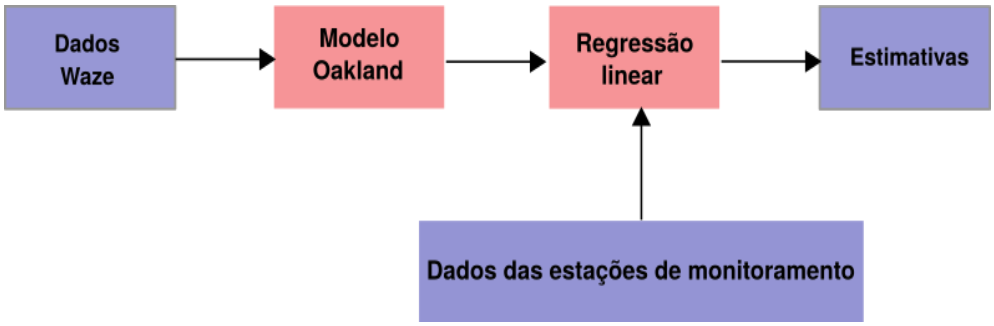
Ajustes realizados para cada cidade

Inicialmente obteve-se uma estimativa do nível de NO_2 para os locais onde havia registros do nível de NO_2 (*i.e.* estações de monitoramento) utilizando o modelo treinado com os dados de Oakland. Verificamos que havia uma grande divergência entre as estimativas e os registros. Para solucionar isso, treinou-se um modelo de regressão logística para cada cidade avaliada onde a variável independente foram as estimativas obtidas pelo modelo XGBoost treinado para Oakland e a variável dependente foram os registros do nível de NO_2 para os locais determinados. Assim, definiu-se o procedimento de estimativa do nível de NO_2 para cada cidade (Figura 4.2.4.1). Isso foi necessário pois cada cidade possui suas próprias particularidades, como características climáticas e geográficas (altimetria, ventos, clima, etc.) e de tipo de vias, ocupação de solos e trânsito (proporção de caminhões/carros, quantidade de edifícios, distribuição de tipos de vias, etc.), o que não é levado em consideração pelo modelo de Oakland.

Além disso, percebeu-se que havia uma melhora considerável no modelo quando levada em consideração a vizinhança de cada hexágono. Em termos simples, o valor passado para o modelo como variáveis independentes de cada hexágono passou a ser a média/máximo daquele hexágono com seus vizinhos. Isso faz com que ocorram mudanças menos bruscas de previsão quando se está na borda de uma área sem vias para carros ou no hexágono imediatamente ao lado de grandes avenidas.

Finalmente, com este modelo é possível estimar o nível de NO_2 para outras regiões da cidade. Para tal, as predições para estas regiões foram geradas e exportadas em um arquivo que serve para alimentar uma interface gráfica (mostrada em uma seção a seguir) que mostra o nível de NO_2 para cada hexágono sobre um mapa do local analisado.

Figura 4.2.4.1: Procedimento para a Obtenção das Estimativas do Nível de NO_2.



Fonte: Elaboração Própria
* Os quadrados em azul representam os dados utilizados e as estimativas, e em vermelho temos os modelos de aprendizado de máquina.

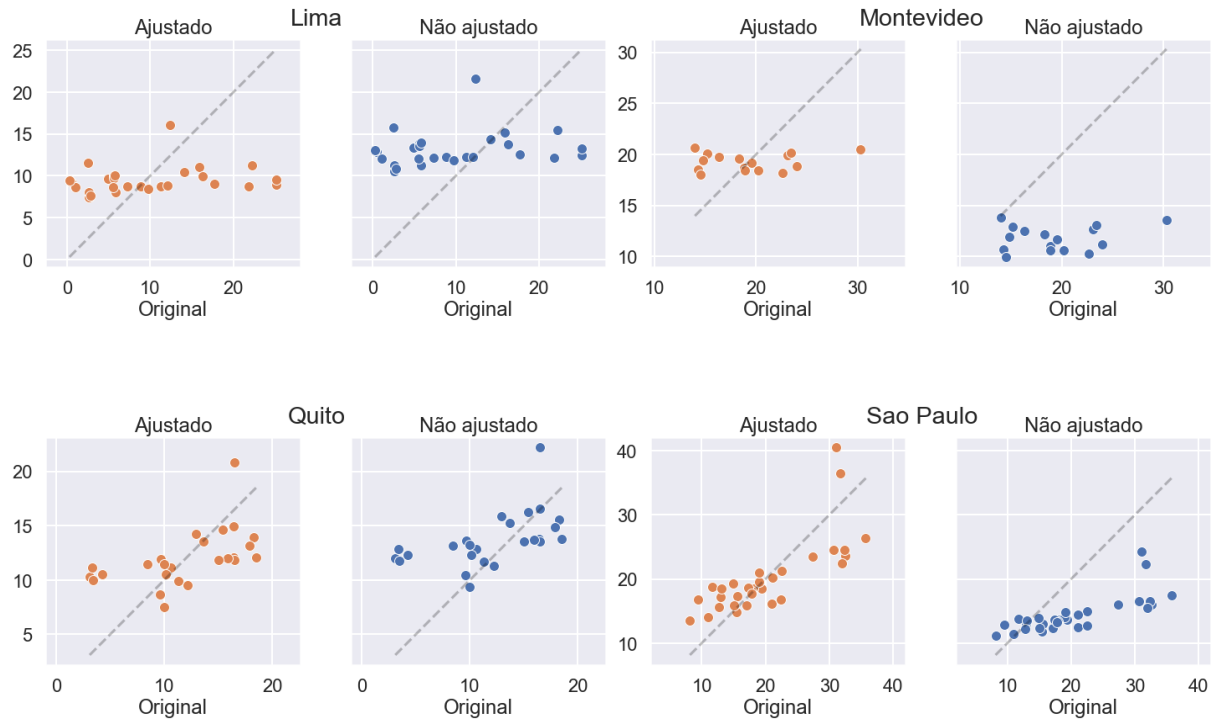
Os códigos foram desenvolvidos em linguagem Python, com auxílio da linguagem SQL para realizar as consultas ao banco de dados. Todos os códigos estão disponíveis no repositório no GitHub https://github.com/mcf1110/smart-cities-pollution.

4.2.5 Descrição dos dados e modelo

No total, foram consideradas 28 estações de monitoramento de poluição (8 para Lima, 4 para Montevidéu,7 para Quito e 9 para São Paulo, Tabela 4.2.1.1), e um raio de 7 hexágonos ao redor de cada estação, o que resulta em 3478 hexágonos distintos. A cobertura de área de 7 hexágonos foi escolhida por ser o valor máximo obtido levando-se em consideração o tempo de um dia para a obtenção de dados a partir da plataforma AWS. Para cada hexágono, foram considerados os valores médios de poluição medidos em cada uma das 4 primeiras semanas de abril de 2019, resultando em 4 agregações: 01 a 07 de Abril, 08 a 14 de Abril, 15 a 21 de Abril e 22 a 28 de Abril. Assim, o resultado foi um total de 13.912 instâncias, uma para cada combinação de hexágono-semana (3478 hexágonos x 4 semanas).

O ajuste linear obtido para cada cidade, considerando as estações de monitoramento, é mostrado abaixo na Figura 4.2.5.1. Em azul, temos os dados conforme preditos pelo modelo de Oakland (Carabetta 2019). Em laranja, temos os dados ajustados linearmente para diminuir o erro quadrático. A linha pontilhada mostra o que seria um resultado ideal, em que todas as predições estão de acordo com o valor real.

Figura 4.2.5.1: Predição do Modelo de Oakland para as Quatro Cidades.



Fonte: Elaboração Própria

Conforme é possível observar, São Paulo e Quito foram as cidades cujas predições ficaram mais próximas do valor real. Quito apresentou resultados em geral satisfatórios para a maioria dos pontos. Contudo, para Lima e Montevidéu o modelo predisse valores praticamente constantes, mesmo com o ajuste linear. Para Lima e Montevidéu podemos dizer que o modelo não funcionou.

O ajuste linear foi escolhido devido à pequena quantidade de estações disponíveis, ao contrário de Oakland, onde havia centenas de hexágonos para treinar o modelo. O modelo ajustado pode então ser utilizado para prever a poluição nos demais hexágonos da cidade. Mas é preciso lembrar as limitações do modelo, pois com poucos dados é difícil fazer um modelo confiável. O ideal é que fossem feitas medidas de qualidade do ar similares às realizadas em Oakland para cada uma das cidades onde se deseja desenvolver um modelo.

Deve-se também notar que devido a possíveis variações sazonais, o ideal é treinar um modelo para cada mês do ano. Por exemplo, durante um verão quente e úmido espera-se que os ajustes sejam diferentes daqueles de um inverno frio e seco.

No repositório mencionado anteriormente, está disponibilizado todo o código que aplica os passos para chegar aos resultados aqui descritos, conforme exposto na seção “Inclusão de Novos Dados” do presente relatório. Também é fornecido o csv resultante da execução deste *notebook*: *“03 Prediction/final_prediction.csv”*, que contém, para todas as instâncias, um id, a data do primeiro dia da semana, a cidade em que se encontra, a taxa de poluição real, a taxa de poluição predita pelo modelo de Oakland, e a taxa de poluição ajustada linearmente. Isso permite a replicabilidade e futura extensão desse trabalho.

4.2.6 Estimativa dos erros para o modelo

Para cada um dos modelos propostos realizou-se a estimativa do erro de predição. Para isso, foi feita uma validação cruzada onde foram consideradas as 4 semanas avaliadas e separadas em 10 grupos. Foram treinados com dados de 9 grupos e validados com os dados de 1 grupo. Assim como no caso do modelo para Oakland, foi utilizado como uma das medidas de performance o MSE. Além disso, utilizou-se também a média percentual absoluta do erro (MAPE). Na Tabela 4.2.6.1, são mostradas as médias do MAPE e MSE para o modelo sem ajuste e para a validação cruzada considerando o modelo com ajuste linear.

Tabela 4.2.6.1: Médias do MAPE e MSE para o Modelo Sem Ajuste e para a Validação Cruzada para cada um dos Modelos Com Ajuste Linear.

Modelos	Com ajuste linear		Sem ajuste linear	
	MAPE	MSE	MAPE	MSE
São Paulo	49,40	34,71	46,29	159,37
Quito	27,93	24,48	46,10	26,93
Montevidéu	71,66	31,87	51,57	125,37
Lima	54,19	66,85	326,38	80,49

Fonte: Elaboração Própria

4.2.7 Interface gráfica

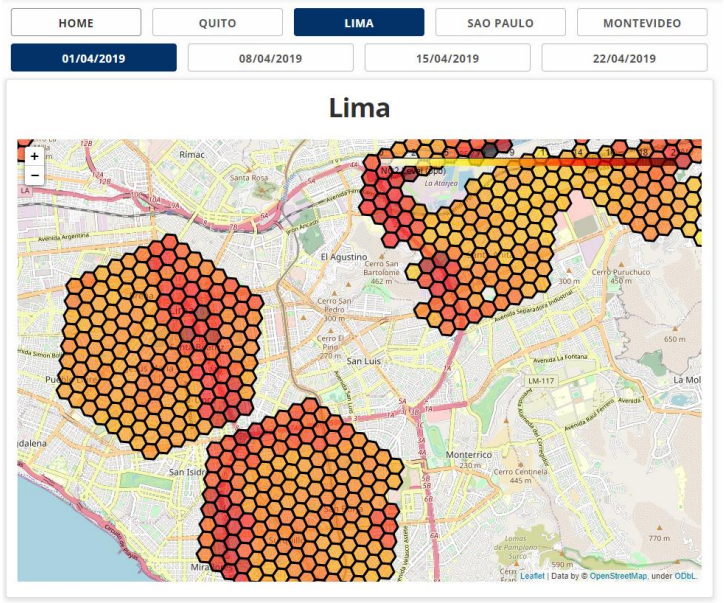
Para permitir que usuários sem conhecimento técnico realizem a visualização dos níveis de poluição preditos, foi desenvolvida uma interface gráfica baseada em mapas. A interface é totalmente baseada em tecnologias web, podendo ser acessada por qualquer usuário que tenha um navegador.

Um jupyter notebook está disponível no repositório que lê o arquivo csv gerado pelos passos anteriores e gera essa interface. Assim, caso no futuro haja a necessidade de atualizar a interface, isso pode ser feito sem muitos problemas. Como a interface é composta totalmente por arquivos HTML e CSS estáticos, ela pode ser hospedada sem a necessidade de uma infraestrutura complexa de servidor. Hoje, ela está hospedada na plataforma Netlify, e pode ser acessada através do link https://smart-cities-pollution.netlify.app/.

Na interface, é possível escolher uma cidade em uma semana para inspeção. Cada mapa apresenta as regiões hexagonais, seguindo o padrão H3 da Uber, em que foi possível realizar a predição naquela cidade. As Figuras 4.2.7.1 a 4.2.7.4 apresentam trechos dos mapas para as quatro cidades.

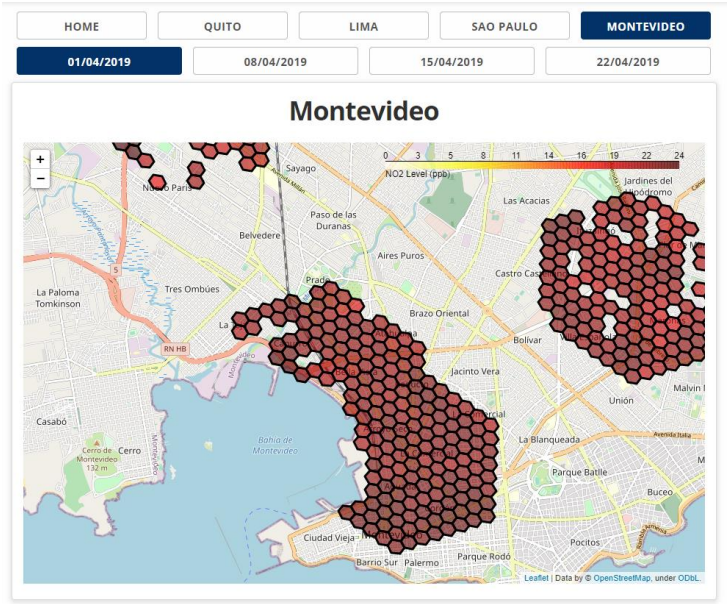
Quanto mais escuro for um determinado hexágono, maior o nível de poluição estimado pelo modelo para aquela região. Isso nos permite inferir visualmente, por exemplo, que a região que acompanha a Avenida Marginal Tietê, em São Paulo, deve possuir um alto nível de poluição devido ao trânsito tradicionalmente intenso na região.

Figura 4.2.7.1: Níveis de Poluição Preditos para algumas Regiões de Lima



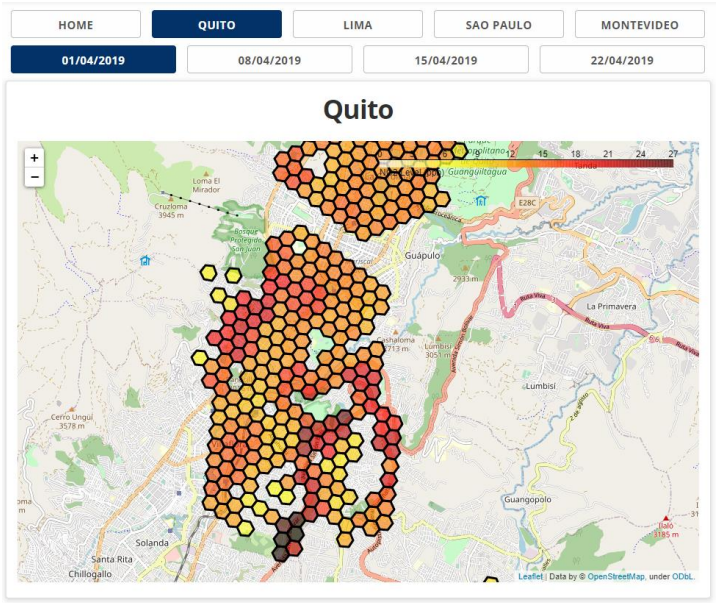
Fonte: Elaboração Própria

Figura 4.2.7.2: Níveis de Poluição Preditos para algumas Regiões de Montevidéu



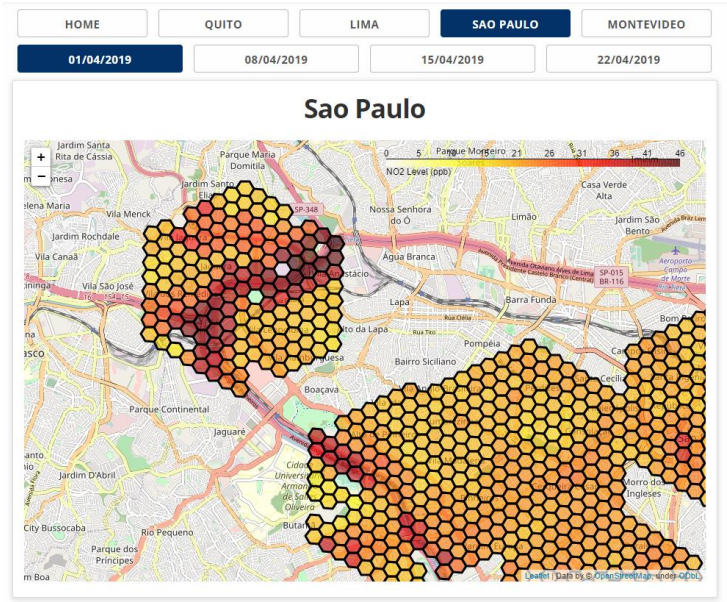
Fonte: Elaboração Própria

Figura 4.2.7.3: Níveis de Poluição Preditos para algumas Regiões de Quito



Fonte: Elaboração Própria

Figura 4.2.7.4: Níveis de Poluição Preditos para algumas Regiões de São Paulo



Fonte: Elaboração Própria

4.2.8 Inclusão de novos dados

Para gerar predições para uma nova cidade e/ou mês, os passos a se tomar são os seguintes:

1. **Obter os dados reais de poluição da cidade naquele mês em específico.** Naturalmente, esse passo varia de caso para caso, pois cada cidade costuma ter N estações de monitoramento localizadas em pontos diferentes, e não existe maneira uniforme de obtê-los, pois cada agência governamental pode disponibilizá-los (ou não) da maneira que for conveniente. Depois, basta colocar os dados obtidos na pasta do GitHub "01 Pollution Data/{nome_cidade}/y_data.csv", no mesmo formato das outras cidades. O h3id deve, idealmente, ter a resolução 9. Esse csv será usado para ajustar linearmente o modelo ao final do processo.
2. **Obter os dados do waze para as estações.** Este passo consiste em executar o arquivo do GitHub "02 Waze Data/Generate Datasets", depois, claro, de configurar o ambiente para acessar o Athena, através da AWS. Na segunda célula, é possível configurar o tamanho do raio em volta das estações a se considerar, bem como qual o tamanho máximo que cada consulta SQL pode ter. Ao executar este arquivo, ele irá realizar consultas SQL e irá armazenar os resultados em csvs na máquina local. Cada consulta costuma levar cerca de 6-10min, então o processo todo pode levar algumas horas, a depender da quantidade de cidades e raio considerado.
3. **Executar a predição.** O código completo - disponível no GitHub - é executado em "03 *predictions/03 Prediction.ipynb"*. Ali, pode-se configurar o período que deve ser considerado (se um dia, uma semana, um mês, ou qualquer outro período), além da data máxima (útil, por exemplo, para descartar os dias ao final de um mês que não completam uma semana). O raio ao redor das estações deve ser ajustado para corresponder ao que foi configurado no passo anterior. Ao executar o notebook, serão mostrados os gráficos com os ajustes e será gerado o dataset com as predições finais.
4. **Gerar os mapas,** através do arquivo do GitHub *"04 GUI/Generate GUI Files"*. Esse passo irá ler o csv gerado no passo 3 e gerar os HTMLs correspondentes. Ao fim do processo, basta fazer upload da pasta www para o servidor que desejar, ou abrir os HTMLs diretamente no navegador.

4.3 Apoio ao Transporte Público

O transporte público é um dos temas mais desafiadores para a gestão pública, dado que afeta não apenas o tempo gasto na locomoção diária, mas também impacta significativamente na qualidade de vida e na saúde dos usuários.

Para dar apoio ao transporte público, propomos o uso de dados das secretarias de transporte e mobilidade de modo a apoiar o cidadão nas decisões de viagem levando-se em conta tempos de deslocamento e as opções de modais de transporte público ou particular disponíveis.

Dessa forma, os produtos relatados neste documento consistiram em desenvolver duas ferramentas:

- **GTFS Estático.** O primeiro desenvolvimento apresentado neste capítulo consistiu em criar um GTFS (*General Transit Feed Specification*) para as cidades de São Paulo, Lima, Xalapa e Montevidéu. O GTFS é um conjunto de arquivos padronizados contendo a especificação de uma rede de transportes e podem ser um insumo poderoso para fornecer informação aos usuários de transporte.
- **Comparação de velocidades de carros e ônibus.** O objetivo desta ferramenta é o de fornecer aos cidadãos um comparativo dos tempos de viagem entre dois modos de transporte diferentes: o veículo particular e o transporte coletivo.

Os produtos relatados neste documento foram desenvolvidos de acordo com a disponibilidade de dados e informações de cada cidade do projeto. Assim, foram utilizados tanto dados abertos, quanto informações fornecidas pelos gestores para chegar a diferentes funcionalidades, respeitando o conceito-chave de se apropriar dos avanços tecnológicos para melhorar o uso do transporte público.

Como observado ao longo deste projeto, há uma imensa gama de dados produzidos diariamente por entes públicos e privados nos grandes centros urbanos que tem um alto potencial de auxiliar no planejamento urbano. Contudo, não basta que esses dados existam, eles precisam estar disponíveis e organizados de tal sorte que sejam vistos como insumos que permita que os agentes públicos e privados possam fazer uso deles e desenvolvam tecnologias com vistas a melhorar a vida da população.

No âmbito deste projeto, foi possível desenvolver visualizações e gerar arquivos GTFS que podem apoiar o desenvolvimento de futuras aplicações nas cidades de Lima, Xalapa, São Paulo e Montevidéu.

Outras cidades como São Paulo e Montevidéu possuem dados de ônibus em tempo real, o que permitiu a criação adicional de outra ferramenta de comparação de tempos de viagem entre carros e ônibus, que será relatada na seção 4.3.2.

4.3.1 GTFS Estático

No sentido de melhorar as informações de transporte público de ônibus, o primeiro protótipo apresentado neste capítulo consistiu em criar arquivos na especificação GTFS (General Transit Feed Specification) para as cidades que ainda não possuíam, bem como apresenta um exemplo de uso de GTFS que consiste numa visualização de roteirização do sistema de ônibus coletivo. Para se chegar a essa roteirização foram desenvolvidas as funcionalidades que serão apresentadas na seção seguinte.

A especificação GTFS é um conjunto de arquivos padronizados contendo a especificação de uma rede de transportes, com informações de linhas, rotas, paradas e tempo de partida. Ele permite, por exemplo, que desenvolvedores utilizem esses dados para criar aplicações e visualizações para traçar rotas, estimar o tempo de deslocamento, entre outras coisas. Dessa forma, uma de suas funções primárias é de ser um instrumento muito útil para desenvolvimento de aplicações que fornecem informações aos usuários de transporte público, otimizando a sua viagem em termos de tempo e custos. Mas essa padronização dos dados também pode ser útil para os gestores públicos no âmbito do planejamento urbano (Wong 2013). Combinados com outras fontes de dados ou tecnologias, estes arquivos podem ser usadas no planejamento e monitoramento da operação do transporte, como cumprimentos de metas de partidas, por exemplo ((Monteiro, Pons e Speicys 2015), além de usos em outros temas correlacionados com o do transporte, como o de segurança de pedestres, entre outros (Catala 2011).

A especificação GTFS, apresentada pela primeira vez em 2005, é o resultado de um projeto entre o Google e a TriMet em Portland para criar um planejador de viagens de transporte público usando o aplicativo da web Google Maps. Por causa da abordagem colaborativa para seu desenvolvimento, a especificação foi projetada para ser simples para as agências criarem, fácil para os programadores acessarem e abrangente o suficiente para descrever um sistema de

trânsito intrincado. O GTFS identifica uma série de arquivos separados por vírgulas que, juntos, descrevem as paradas, viagens, rotas e tarifas acerca do serviço de uma agência. O Google abriu o feed para uso geral em meados de 2007 e se propagou amplamente à medida que as agências traduziam suas programações de trânsito para o formato. O feed é o padrão mais usado para troca de dados de trânsito estático nos Estados Unidos hoje. De acordo com dados do GTFS Data Exchange de julho de 2012, pouco mais de 25 por cento das agências nos Estados Unidos publicaram dados de trânsito aberto no formato GTFS (Wong, 2013)

Funcionalidades desenvolvidas

O trabalho desenvolvido e relatado neste documento foca no desenvolvimento da especificação GTFS para as cidades de São Paulo, Montevidéu, Xalapa e Lima. Foram desenvolvidas duas funcionalidades: uma aplicação web () e a geração de arquivos GTFS. Ao final, é mostrado um exemplo de uso dos dados em formato GTFS para uma criar rotas. Serão relatados adiante os aspectos técnicos que envolveram o desenvolvimento, bem como as particularidades e escolhas feitas para cada cidade, juntamente aos procedimentos para a instalação dos programas necessários para a visualização da roteirização.

As principais funcionalidades desenvolvidas são:

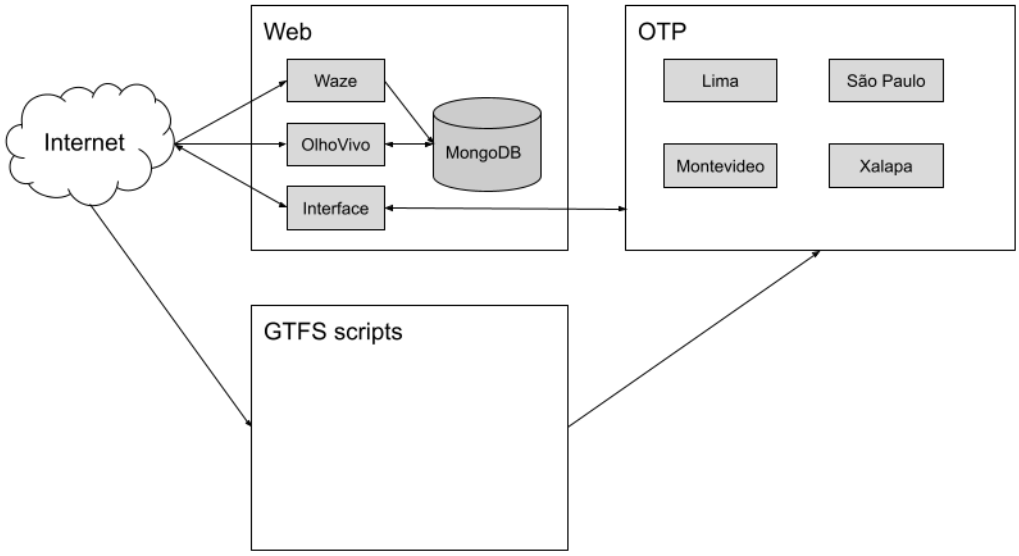
- Aplicação web
    - geração de grafo a partir do GTFS de São Paulo (InterSCityBus)
    - coleta e armazenamento de dados da API OlhoVivo
    - (InterSCityBus)coleta e armazenamento de dados do Waze
    - interface unificada para geração de rotas de ônibus com Open Trip Planner (OTP)
    - instruções e automação dos passos para iniciar todos os softwares
- Geração de arquivos GTFS compatíveis com OTP para São Paulo (Brasil), Xalapa (México), Lima (Peru) e Montevidéu (Uruguai)

Todo o código fonte está hospedado em https://gitlab.com/pragmacode/bid_fgv_interscity/smartcities_bigdata sob a licença de software livre Mozilla Public License 2.0.

4.3.1.1 Arquitetura

O código do sistema é estruturado em duas partes principais: um sistema web e um conjunto de scripts para geração de GTFS. Ambos são escritos na linguagem de programação Python, escolhida pela ampla disponibilidade de ferramentas para esta linguagem. Na figura a seguir é possível ter uma visão geral da arquitetura e integração do sistema.

Figura 4.3.1.1.1: Arquitetura do sistema



Fonte: Elaboração própria

A aplicação web foi criada utilizando o arcabouço Django para melhor organizar o código e deixá-lo padronizado para futuros desenvolvimentos.

Esse sistema tem três partes principais:

A primeira, se integra à API do Waze para armazenar no MongoDB dados de alertas, irregularidades e congestionamentos.

Outra é a integração com o serviço OlhoVivo fornecido pela SPTRANS que também armazena no MongoDB os dados de geolocalização dos ônibus do município de São Paulo. Ambos os

componentes persistem dados no MongoDB, escolhido também por sua flexibilidade e popularidade, para que no futuro outras aplicações possam ser desenvolvidas para consumir esses dados.

A terceira parte do sistema tem a responsabilidade de prover uma interface web para que usuários possam planejar viagens de ônibus. Para possibilitar isso, foi utilizado o Open Trip Planner (OTP) e dados fornecidos pelas cidades sobre os itinerários dos ônibus. O OTP foi escolhido para fazer parte do sistema, pois tem o código aberto, já é utilizado com sucesso em outros projetos e provê, tanto uma interface web pronta para uso, quanto APIs que podem ser úteis para futuras implementações de visualizações avançadas.

Por fim, para utilizar o OTP é exigido que os dados de transporte público estejam formatados em GTFS e possui algumas limitações quanto a ele. Isso nos levou a criar scripts à parte da aplicação web que sejam capazes de ler os dados fornecidos pelas cidades em diferentes formatos e convertê-los para GTFS seguindo os requisitos para uso do OTP. A escolha por criar scripts fora da aplicação web se deu no contexto de buscar pela solução mais simples possível e com a visão de que os dados de ônibus das cidades não mudam com frequência.

Representação de GTFS em grafo

Como mencionado anteriormente, parte do que foi desenvolvido consiste na integração do projeto InterSCityBus. Esse projeto a partir do GTFS estático da cidade de São Paulo constrói um grafo no qual posteriormente mapeia em tempo real os dados obtidos por meio da API OlhoVivo. O objetivo dessa forma de representar os dados é permitir que sejam criadas métricas (por exemplo, velocidades em corredores de ônibus) e ferramentas (por exemplo, estimativa do tempo de chegada do próximo ônibus em um ponto) que sejam úteis para técnicos e para os usuários do sistema.

Como parte da integração com o InterSCityBus, tanto o grafo gerado quanto os dados coletados em tempo real foram modelados para armazenamento no MongoDB. Dessa forma, possibilitamos o acesso em paralelo ao mesmo conjunto de dados por diferentes aplicações, inclusive em contextos de computação distribuída.

No começo do projeto trabalhou-se com a possibilidade de usar dados com as posições em tempo real dos ônibus. Inicialmente foi criada essa infraestrutura para os dados em tempo real,

mas apenas a cidade de São Paulo possuía esta informação. Portanto, o foco foi dado para a parte dos dados estáticos, que a maioria das cidades tinham disponíveis. De toda forma, outras cidades que tenham os dados em tempo real poderiam trabalhar com essa infraestrutura se assim desejarem.

Apenas com os dados estáticos, por exemplo, não é possível prever quando um ônibus irá chegar em um ponto, dado que é necessário saber a posição dos ônibus em cada instante. Então, utilizar os dados em tempo real seria um outro caminho, mas primeiro é preciso ter o GTFS estático para depois ter o dinâmico.

Coleta de dados para séries históricas

Ainda relacionadas aos dados em tempo real, foram realizadas coletas dos dados fornecidos pelo Waze. A intenção foi dar uma infraestrutura para que as cidades possam ter esse armazenamento na própria infraestrutura delas, se assim desejarem. E da mesma forma esse armazenamento é interessante para criação de séries históricas. Esses dados podem ser usados para outros fins, como criar aplicações das mais diversas áreas.

Escalabilidade do OTP

OpenTripPlanner (OTP) é uma família de projetos de software de código aberto que fornece informações de passageiros e serviços de análise de rede de transporte. O principal componente Java do lado do servidor encontra itinerários combinando segmentos de trânsito, pedestres, bicicletas e carros por meio de redes criadas a partir de dados OpenStreetMap e GTFS amplamente disponíveis e de padrão aberto. Este serviço pode ser acessado diretamente por meio de sua API da web ou por meio de uma variedade de bibliotecas cliente Javascript, incluindo modernos componentes modulares reativos voltados para plataformas móveis.

Lançado em 2009, o projeto atraiu uma comunidade próspera de usuários e desenvolvedores, recebendo apoio de órgãos públicos, startups e consultorias de transporte semelhantes. O OTP capacita serviços de planejamento de viagens regionais e nacionais em todo o mundo, bem como inúmeros aplicativos móveis populares para várias cidades.

A documentação do OTP menciona que a versão 2, usada no projeto, foi criada tendo como um de seus objetivos a escalabilidade. Porém, não foram encontrados estudos e resultados que comprovem e que forneçam referências sobre como adequar os recursos computacionais às expectativas de demanda.

Por esse motivo e dado que a aplicação de roteirização tem potencial para ter alta demanda do público das cidades, recomendamos que as mesmas realizem os testes apropriados após a instalação, de forma a simular cenários pertinentes às suas realidades de demanda.

Pontos de atenção da arquitetura

Durante o período de duração do projeto, o foco foi criar provas de conceito para as cidades sobre o que é possível realizar com os dados de GTFS e apontar quais informações são importantes para atingir os resultados. O protótipo desenvolvido pode ser utilizado internamente pelas administrações, mas não está finalizado para uso do público em geral.

Uma parte dos resultados apresentados, por falta de dados ou por formatação incompatível com as ferramentas, se baseia em suposições e aproximações, pois também não seria possível esperar que as cidades produzissem os dados necessários no limite de tempo do projeto.

4.3.1.2 Instalação

Esta seção apresenta o processo de instalação dos sistemas e configuração dos ambientes. O objetivo foi o de realizar a instalação de modo que ela fosse facilmente replicável pelas cidades envolvidas no projeto e de forma independente do sistema operacional.

O passo-a-passo é demonstrado no texto abaixo e também no link do vídeo[16]: https://gitlab.com/pragmacode/bid_fgv_interscity/smartcities_bigdata/-/raw/master/deploy/demo_ubuntu.mp4

O repositório do sistema contém o arquivo “HACKING.md” que fornece instruções detalhadas para a operação do sistema. A seguir, serão repetidas as instruções para instalação utilizando Docker Compose, pois a consideramos a mais simples para usuários novos. Entretanto, mesmo

[16] O vídeo não possui áudio e propõem-se a apenas demonstrar visualmente o processo de instalação.

sendo mais simples, ainda é recomendável que elas sejam executadas por um profissional capacitado para operar sistemas de informação.

Requisitos

- Python 3.9
- Poetry
- Git
- Docker
- Docker Compose

Configuração

O primeiro passo é, usando o Git, clonar o repositório do projeto em https://gitlab.com/pragmacode/bid_fgv_interscity/smartcities_bigdata. Dentro da pasta criada, que chamaremos de raiz do projeto, é preciso instalar as dependências do projeto executando:

```
poetry install
```

Que fará o download dos pacotes necessários e suas dependências. A título de completude, listamos aqui os pacotes base que usamos:

- ☐ `behave-django`
- ☐ `chromedriver`
- ☐ `django`
- ☐ `django-bootstrap4`
- ☐ `django-selenium`
- ☐ `djongo`
- ☐ `geopandas`
- ☐ `geopy`
- ☐ `networkx`
- ☐ `pandas`
- ☐ `pykml`
- ☐ `pylama`
- ☐ `pyproj`

- [ ] `pytest-django`
- [ ] `pytest-mock`
- [ ] `requests`
- [ ] `selenium`
- [ ] `sentry-sdk`
- [ ] `shapely`

Feito isso, será preciso criar a seguinte estrutura de pastas com os dados arquivos que serão utilizados de base para o sistema:

- `data/`
  - `sao_paulo/`
    - `otp/`
      - `buslines.gtfs.zip`
      - `map.pbf`
    - `gtfs/`
  - `xalapa/`
    - `otp/`
      - `buslines.gtfs.zip`
      - `map.pbf`
  - `lima/`
    - `otp/`
      - `buslines.gtfs.zip`
      - `map.pbf`
  - `montevideo/`
    - `otp/`
      - `buslines.gtfs.zip`
      - `map.pbf`
  - `mongodb/`

Nessa estrutura de pastas, os arquivos `map.pbf` são extrações do Open Street Map para cada uma das cidades. Já a pasta `mongodb/` inicialmente é um diretório vazio que, posteriormente, será utilizado pela aplicação para persistir dados. A seguir, para cada cidade, será detalhado o que é esperado nos demais subdiretórios e arquivos.

Na pasta `sao_paulo/`, estão os dados da cidade de **São Paulo**. No subdiretório `gtfs/`, devem estar os arquivos não compactados do GTFS da cidade. No subdiretório `otp/`, o arquivo `buslines.gtfs.zip` contém praticamente o mesmo conteúdo do subdiretório `gtfs/`, porém os arquivos `stop_times.txt` e `trips.txt` precisam ser ajustados seguindo os passos explicitados na seção 4.3.1.4.

Os dados de **Xalapa** devem estar no diretório `xalapa/`. Como não foi fornecido um GTFS oficial para ela, criamos um script que o gera a partir dos dados produzidos no *Mapatón Xalapa 2016*. A partir da raiz do projeto, há o arquivo `samples/gtfs/xalapa.gtfs.zip` que deixamos já gerado para facilitar esse processo. Você pode copiá-lo para dentro do subdiretório `otp/`.

Os dados de **Lima** devem estar no diretório `lima/`. Como não foi fornecido um GTFS oficial para ela, criamos um script que o gera a partir dos dados fornecidos pelo contato da cidade. A partir da raiz do projeto, há o arquivo `samples/gtfs/lima.gtfs.zip` que deixamos já gerado para facilitar esse processo. Você pode copiá-lo para dentro do subdiretório `otp/`.

Os dados de **Montevidéu** devem estar no diretório `montevideo/`. Como não foi fornecido um GTFS oficial para ela, criamos um script que o gera a partir dos dados disponíveis em https://catalogodatos.gub.uy/organization/d9026405-35be-410e-b251-492fba99c57d?groups=transporte. A partir da raiz do projeto, há o arquivo `samples/gtfs/montevideo.gtfs.zip` que deixamos já gerado para facilitar esse processo. Você pode copiá-lo para dentro do subdiretório `otp/`.

Por último, edite o arquivo `docker-compose.yml` presente na raiz do projeto.

- `SECRET_KEY`: chave aleatória secreta cuja função é validar requisições e que você deve gerar como preferir
- `OLHO_VIVO_TOKEN`: obtido no portal para desenvolvedores da SPTRANS
- `SENTRY_DSN`: chave do serviço de notificação de erros que deve ser obtido por meio da criação de uma conta
- `WAZE_BASE_URL`: obtida por meio do cadastro no programa da Waze para cidades

Execução

A partir da raiz do projeto, execute:

```
docker-compose up
```

Com esse comando, todos os processos necessários devem ser iniciados. Os serviços estarão disponíveis nas seguintes URLs:

- localhost:8080 - São Paulo OTP instance
- localhost:8081 - Xalapa OTP instance
- localhost:8082 - Lima OTP instance
- localhost:8083 - Montevideo OTP instance
- localhost:8000 – Webserver

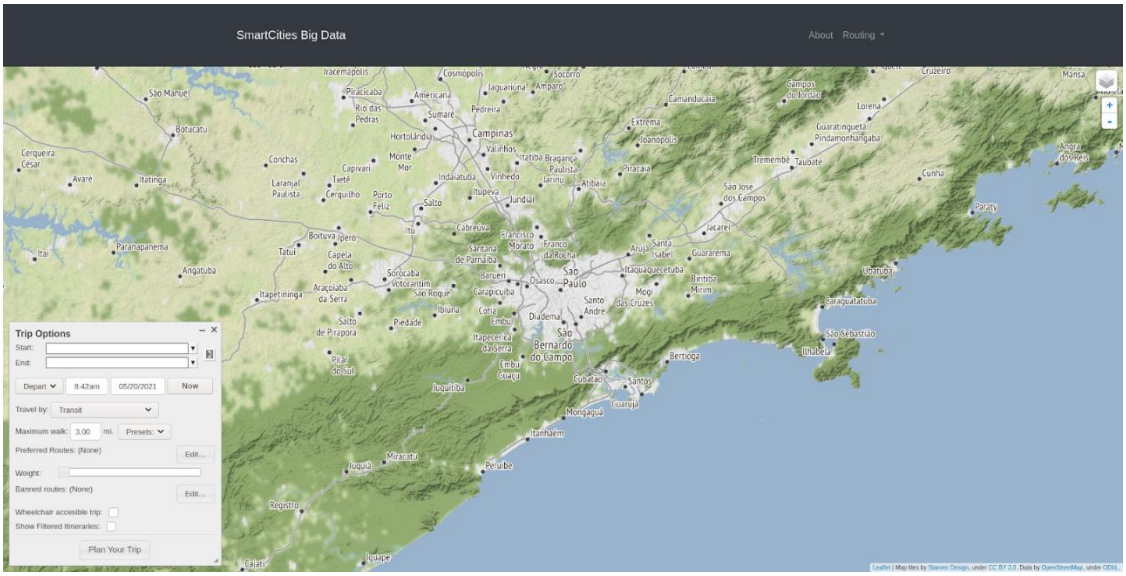
4.3.1.3 Roteirização

Um dos usos mais comuns dos dados do GTFS é a possibilidade de traçar rotas do sistema de transporte. Neste projeto, para fazer essa roteirização foi utilizado o software livre chamado Open Trip Planner (OTP). Ele junta os dados do Open Street Map e os dados do GTFS estático e usa isso para fornecer roteiros para os usuários do sistema de transporte. Uma vez instalado, para utilizar basta definir origem e destino, dia, horário de partida e chegada e o sistema toma conta de construir as rotas e mostrar os itinerários tanto em texto quanto no mapa.

Instruções de uso

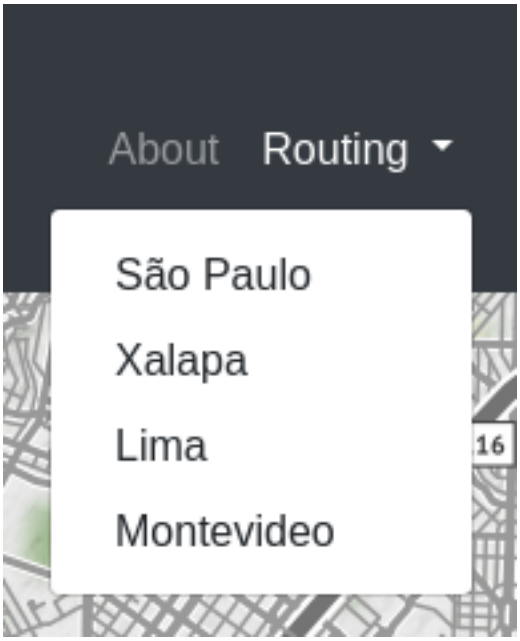
Após a instalação, para iniciar a operação do sistema acesse http://localhost:8000/otp_interface/sao_paulo.

Figura 4.3.1.3.1: Interface do produto



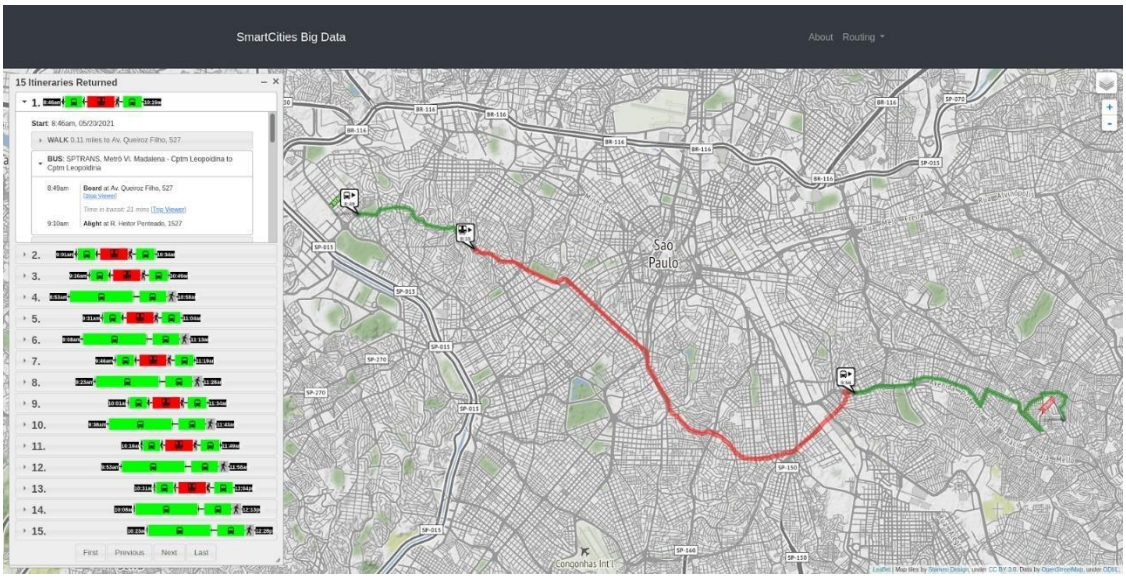
No canto superior direito existe o menu “Routing” que permitirá escolher a cidade para a qual deseja gerar rotas.

Figura 4.3.1.3.2: Cidades disponíveis para análise



Para calcular uma rota, clique no mapa uma vez de modo a especificar o ponto de partida e uma segunda vez para o destino. Após o segundo clique serão exibidas todas as opções de rota encontradas.

Figura 4.3.1.3.3: Visualização das rotas encontradas



Acessando dados armazenados

Para adicionar uma nova cidade e ter acesso aos dados da API do Waze já coletados:

1. edite `smartcities_bigdata/settings.py`
2. adicione uma nova entrada no dicionário `WAZE_POLYGONS` com os dados da cidade

Para se conectar ao MongoDB e consumir os dados que são coletados do Waze e OlhoVivo faça:

```
docker-compose run mongo mongo
```

Esse comando deve abrir o CLI do MongoDB para que você possa realizar uma consulta. Para informações sobre a estrutura de armazenamento, verifique os arquivos do projeto `interscitybus/models.py` e `waze/models.py`.

4.3.1.4 Geração de GTFS estático

O *General Transit Feed Specification* (GTFS) é uma especificação de dados que permite agências públicas de trânsito publicar seus dados de trânsito em um formato que pode ser consumido por uma vasta gama de softwares. Ele é composto por duas partes: a estática, que contém dados sobre agenda, tarifa e componentes geográficos; e a em tempo real que contém previsões de chegada, posição dos veículos e avisos sobre os serviços.

Nesse projeto, o foco foi a geração da parte estática com base em informações fornecidas pelas cidades a respeito do transporte público com ônibus. O objetivo final é produzir GTFS compatíveis com o OTP.

A cidade de **São Paulo** já disponibiliza um GTFS atualizado através do portal da SPTRANS para desenvolvedores, porém, o mesmo é disponibilizado em um formato não suportado pelo OTP. Para ajustá-lo de acordo com os requisitos do OTP foi criado um script que pode ser utilizado seguindo os passos:

1. certifique-se de que no GTFS de São Paulo existe um arquivo `frequencies.txt` e de que ele esteja no diretório correto
2. a partir da raiz do projeto, entre no diretório `gtfs_generators`
3. execute `poetry run python sp_use_frequencies.py`
4. isso gerará arquivos `new_stop_times.txt` e `new_trips.txt`
5. esses arquivos gerados devem ser copiados dentro do `.gtfs.zip`, no lugar dos `trips.txt` e `stop_times.txt` originais (lembre-se de remover o prefixo new_)
6. por último remova o arquivo `frequencies.txt`

Xalapa não possui um GTFS estático pronto, mas há informações sobre as rotas de ônibus em https://datos.gob.mx/busca/dataset/rutas-tranporte-publico. Esses dados estão no formato GeoJSON e foi criado um script para gerar GTFS a partir deles que pode ser usado da seguinte forma:

```
poetry run python gtfs_generators/esri_shapefile_to_gtfs.py <ROOT DIR>
```

Onde `<ROOT_DIR>` é a pasta contendo os dados ESRI Shapefile.

Recebemos diretamente dos representantes de **Miraflores** um arquivo no formato KML contendo todas as rotas de Lima. Para este também foi criado um script que gera o GTFS. Para executá-lo faça:

```
poetry run python gtfs_generators/kml_to_gtfs.py <ROOT_DIR>
```

Onde `<ROOT_DIR>` é a pasta contendo os dados KML.

Para **Montevidéu** foram encontrados dados no endereço https://catalogodatos.gub.uy/organization/d9026405-35be-410e-b251-492fba99c57d?groups=transporte. Nesse, faça download dos seguintes arquivos:

- `v_uptu_lsv_destinos.zip`
- `uptu_variante_no_maximal.zip`
- `v_uptu_variantes_circulares.zip`
- `v_uptu_paradas.zip`
- `HORARIOS_OMNIBUS datos.zip`
- `uptu_pasada_variante.zip`
- `uptu_pasada_circular.zip`

Depois execute o script:

```
poetry run python gtfs_generators/montevideo_data_to_gtfs.py <ROOT_DIR>
```

Onde `<ROOT_DIR>` é a pasta contendo os arquivos listados acima.

4.3.1.5 Geração de GTFS para outras cidades

Com os scripts desenvolvidos, podemos gerar GTFS para outras cidades que forneçam seus dados no formato **ESRI Shapefile** ou **KMZ**.

- ☐ ESRI Shapefile:
- ☐ Crie um diretório que contenha uma pasta para cada linha de ônibus, dentro delas, é obrigatório ter um arquivo chamado `route.zip`, que contenha um shapefile com a rota em uma estrutura de Linestring; além disso, é possível que também esteja disponível

um arquivo opcional chamado `stops.zip`, que contenha um shapefile com as paradas da rota, fornecidas em estrutura de lista.

- `poetry run python gtfs_generators/esri_shapefile_to_gtfs.py <ROOT DIR>`
- Onde `<ROOT_DIR>` é a pasta contendo os dados ESRI Shapefile.

- KMZ:

- Basta extrair o arquivo .kml de dentro do .kmz que contém as rotas de ônibus e rodar o mesmo script que usamos para a cidade de Lima, apontando para a localização do arquivo .kml, isto é:
- `poetry run python gtfs_generators/kml_to_gtfs.py <ROOT_DIR>`
- Onde `<ROOT_DIR>` é a pasta contendo o arquivo .kml extraído.

4.3.1.6 Sugestões às cidades

O formato GTFS tem uma especificação relativamente aberta, permitindo que os mesmos dados possam ser representados de formas distintas. Aplicações que usam esse arquivo, como o OTP, geralmente não suportam todas essas variações e definem um subconjunto suportado.

Nesse contexto, a sugestão para **São Paulo**, é de gerar o GTFS sem usar o `frequencies.txt`, como forma de oferecer uma versão de maior compatibilidade com as aplicações que consomem GTFS. Entendemos que a versão atual é interessante do ponto de vista de ser mais enxuta, portanto, uma alternativa seria disponibilizar para download as duas opções.

Montevidéu não fornece o GTFS diretamente, mas disponibiliza todas as informações necessárias para gerá-lo exigindo poucas suposições por parte do programador. Essa quantidade grande de dados trouxe desafios com respeito à organização, consistência e acurácia dos dicionários de dados, conforme listados a seguir:

- A documentação lista 3 tipos de serviço (dias úteis, sábados e domingos), porém, havia um quarto tipo nos dados para os quais convencionamos que representam linhas que operam todos os dias;

- Alguns arquivos têm informações conflitantes entre si, por exemplo, identificadores repetidos de linhas diferentes, casos nos quais adicionamos sufixos para diferenciar as linhas.

Já para **Xalapa** e **Lima** havia menos informação disponível do que o necessário para um GTFS mínimo, o que nos forçou, para gerar um GTFS, a fazer suposições mais fortes:

- os arquivos `calendar.txt` e `agency.txt` foram gerados com informações fictícias
- linhas sem informações sobre paradas tiveram paradas criadas a cada 500m;
- para todas as linhas foi suposto que a primeira viagem parte às 6h00 e a última às 21h00, todos os dias;
- viagens partem a cada 15 minutos;
- o tempo de viagem entre paradas é fixo em 2 minutos.

Portanto, para gerar GTFS mais precisos para essas duas cidades os seguintes dados são importantes:

- dias de operação e as linhas categorizadas de acordo;
- informações das empresas que operam o sistema e as linhas categorizadas de acordo;
- informações completas sobre todas as paradas de ônibus e quais linhas passam por elas em cada horário específico;
- horário de partida de cada viagem de cada linha;
- tempo gasto para o veículo percorrer o trajeto entre cada parada.

4.3.1.7 Próximos passos

O projeto apresentado é suficiente para demonstrar aos gestores das cidades possíveis aplicações que podem ser produzidas com dados públicos sobre transporte, porém, para ser utilizado pelo público geral são necessárias evoluções.

Como primeiro passo sugerimos investir na qualidade geral do sistema, atendendo aos apontamentos feitos previamente nesse relatório na seção “Pontos de atenção da arquitetura”. Feito isso, a seguir são destacadas as questões relevantes para roteirização, geração de GTFS e processo de instalação.

Roteirização

A versão entregue é esperada que seja totalmente funcional. Para disponibilizá-la ao público, recomendamos ao gestor ter atenção aos seguintes fatores:

- ☐ buscar apoio de profissionais de tecnologia qualificados para operar as que o projeto aplica e suportar o sistema;
- ☐ quantificar a demanda esperada do público em acessos por segundo;
- ☐ realizar teste de carga e dimensionar os recursos de infraestrutura de acordo.

Conforme os cidadãos utilizarem a plataforma outras demandas surgirão, mas no momento que escrevemos esse relatório as que nos parecem mais relevantes são:

1. criar uma interface gráfica mais simples do que a fornecida pelo OTP,

- ☐ dado que a interface usada atualmente é parte do OTP e a utilizamos no projeto para atender às demandas dentro do prazo,

- ☐ contudo, há o entendimento de que ela é mais complexa do que o necessário para o público geral e pode trazer dificuldades no uso,

- ☐ outro ponto que uma nova interface gráfica precisará melhorar com relação à do OTP é quanto à internacionalização, visto que em nosso estudo não encontramos formas de traduzi-la para outros idiomas;

2. criar interface gráfica para o gestor fazer upload de dados GTFS e do OSM atualizados;
3. quantificar a demanda esperada do público em acessos por segundo;

4. realizar testes de escalabilidade sobre o OTP para definir os requisitos de recursos computacionais necessários e provisioná-los de acordo com a demanda esperada;

Geração de GTFS

O principal foco é facilitar o uso pelas cidades sempre que houver dados mais recentes de forma que essas possam atualizar sem precisar de auxílio de uma equipe técnica:

1. melhorar os scripts para tratar erros de formato e exibir mensagens significativas que permitam diagnóstico e correção mais rápidos;

2. melhorar os scripts definindo um formato de entrada que as cidades possam utilizar para fornecer os dados faltantes mencionados anteriormente de forma que o GTFS gerado seja mais completo e sem dados fictícios;

3. criar uma interface gráfica que permita ao gestor público fazer o upload dos dados brutos da cidade para processamento e depois o download do GTFS.

Instalação

O processo de instalação das ferramentas descrito no relatório é dependente de técnicos capacitados para realizá-lo. Essa dependência foi necessária para podermos entregar processos agnósticos em termos de provedor de infraestrutura e sistema operacional. Ou seja, garantir que os sistemas possam ser instalados pelo maior número de cidades.

Por outro lado, nem todas as cidades podem ter técnicos capacitados disponíveis, mas podem ter recursos para contratar serviços de infraestrutura terceirizados. Nesse contexto, é interessante ter também os processos de instalação preparados para pelo menos um grande provedor de infraestrutura, como a AWS. Dessa forma, o gestor poderá seguir instruções para contratar os serviços desse provedor e instalar a plataforma. Feito isso, os próprios serviços do provedor devem se assegurar automaticamente quanto à escalabilidade e disponibilidade eliminando grande parte da dependência de técnicos especializados para operar a plataforma.

4.3.2 Comparação de velocidades de carros e ônibus

A falta de planejamento urbano e de uso do solo em grandes centros urbanos somado à expansão da frota de veículos particulares nas últimas décadas tem afetado significativamente a vida dos cidadãos, principalmente, por provocar altos níveis de congestionamentos. Além do elevado tempo gasto com locomoções diárias, aumenta-se o nível de exposição à poluição do ar e sonora, prejudiciais à saúde.

Incentivar o uso do transporte público e diminuir o uso de automóveis particulares seria uma forma de contribuir para a diminuição dos congestionamentos urbanos. Neste sentido, esta seção apresenta uma aplicação desenvolvida com uso de dados em tempo real com intuito de oferecer ao cidadão mais informação para a escolha de seu tipo de viagem. Em adendo, também visa oferecer aos gestores uma ferramenta de monitoramento de políticas de transporte e trânsito e de produção de evidências para formulação de políticas de priorização dos serviços de transporte público coletivo, dado que, além de serem mais sustentáveis, são mais eficientes. A importância dessa ferramenta consiste na sensibilização dos usuários acerca das vantagens dos corredores de ônibus, especialmente, nos horários de pico. Seu objetivo principal é incentivar a migração modal. Trata-se de uma aplicação que pretende mudar o comportamento dos indivíduos em direção a uma cidade mais sustentável por meio de produtos tecnológicos.

4.3.2.1 Ferramenta

O objetivo desta ferramenta é o de fornecer aos cidadãos um comparativo dos tempos de viagem entre dois modos de transporte diferentes: o veículo particular e o transporte coletivo. Em posse dessas informações, o cidadão pode decidir com mais segurança qual a melhor forma de deslocamento para uma dada viagem, considerando além dos aspectos financeiros (custo da passagem de transporte público, da gasolina, do estacionamento, entre outros custos relacionados) e de conforto, o tempo de deslocamento de acordo com o modo de transporte escolhido.

Dada a disponibilidade de dados do ônibus em tempo real, foi possível desenvolver a ferramenta para as cidades de São Paulo e de Montevidéu. Contudo, a solução é pensada para ser extensível a quaisquer cidades que se interessem por gerar esse tipo de evidência para futuro planejamento urbano.

Para isso, a ferramenta contou com a integração de três bases de dados diferentes, sendo duas para o cálculo de velocidades dos ônibus e uma com os dados de velocidade dos veículos particulares. É importante notar que em geral a velocidade do ônibus e dos veículos particulares em uma mesma via é diferente: em vias onde o tráfego é compartilhado os ônibus tendem a desempenhar uma velocidade menor, uma vez que devem parar frequentemente para que os passageiros embarquem e desembarquem. Entretanto, em vias que contam com priorização de tráfego para o transporte coletivo, é esperado que os ônibus se desloquem com velocidade

superior à dos veículos particulares. Por essa razão, as velocidades dos ônibus e dos veículos particulares são calculadas a partir de fontes de dados diferentes.

Para o cálculo das velocidades dos ônibus, a ferramenta combinou duas fontes de dados: o arquivo GTFS contendo a descrição da rede de transportes da cidade, e os dados de GPS dos ônibus em tempo real. O arquivo GTFS contém informações como a posição dos pontos de ônibus, a frequência de partida das linhas e o trajeto realizado em cada rota de ônibus. Os dados de GPS consistem, essencialmente, em coordenadas geográficas associadas ao instante em que foi capturada, ao veículo que a transmitiu e à viagem que está sendo realizada. Como o dado de GPS é naturalmente impreciso com erros da ordem de dezenas a centenas de metros, são feitas associações geométricas entre o dado bruto e o traçado esperado para a viagem. Sendo assim, os veículos precisam ser mapeados constantemente e os respectivos deslocamentos computados para que, então, seja calculada a velocidade de cada um.

Paralelamente, são utilizados os dados provenientes da plataforma Waze, que utiliza como base a coleta de posições e demais informações de deslocamento de dispositivos celulares dos motoristas que trafegam utilizando o aplicativo. A partir de então, tem-se acesso à ocorrência de congestionamentos na cidade, que são expressos por meio de uma série de coordenadas geográficas, compondo uma linha, e uma medida de severidade associada à extensão e lentidão. O cálculo da velocidade associada aos veículos usuários do Waze é feito tomando a localização dos trechos de interesse, o que permite consultar em tempo real o tempo médio que os veículos circulando no trecho levam para percorrê-lo.

Os dados coletados, considerando todas as fontes descritas, são armazenados em um banco de dados e, dessa forma, ficam acessíveis para consultas futuras. Na proposta atual, no entanto, o uso das informações de tráfego é utilizado para compor o painel que apresenta de forma simultânea a situação do tráfego no transporte público e privado.

São mostradas ao usuário informações relativas aos principais eixos de transporte da cidade, comparando-se tanto a velocidade (em km/h) e o tempo (em minutos) para observar a diferença no comportamento de ônibus e carros ao longo dessas vias nas duas cidades. No caso de São Paulo, as vias em observação consistem em corredores, onde o tráfego dos dois modos de transporte é completamente segregado, resultando em aproximadamente 150 km mapeados. Para Montevideo, foram selecionados cerca de 100 km de vias em que se observa concentração do fluxo de veículos, embora corredores não estejam presentes.

- Vias Monitoradas

Lista de vias de Montevidéu nas direções Centro-Bairro e Bairro-Centro:

- ☐ Av. Agraciada
- ☐ Bv. España
- ☐ Av. Comercio
- ☐ Av. Luiz Alberto de Herrera
- ☐ Bv. General Artigas
- ☐ Av. Italia
- ☐ Av. 18 De Julio

Lista de vias de São Paulo nas direções Centro-Bairro e Bairro-Centro:

Zona Norte

- ☐ Inajar de Souza - parte 1
- ☐ Inajar de Souza - parte 2
- ☐ FAIXA - Engenheiro Caetano Alvares

Zona Sul

- ☐ Santo Amaro/9 de julho
- ☐ Parelheiros - parte Rio Bonito
- ☐ Ibirapuera

Zona Leste

- ☐ Paes de Barros
- ☐ FAIXA - Luiz Inácio Anhaia Melo
- ☐ Expresso Tiradentes

Zona Oeste

- ☐ Rebouças
- ☐ Pirituba - parte Lapa
- ☐ Pirituba - parte São João

4.3.2.2 Metodologia

Primeiramente, utiliza-se como referência a coleção de arquivos no padrão *General Transit Feed Specification*, desenvolvido pelo Google. A partir de então, são conhecidas informações sobre a malha viária na região e as características de sua operação. Em especial, são utilizados os dados espaciais que descrevem rotas e pontos de parada de veículos para construir uma representação geométrica da rede, que consiste em um conjunto de nós (pontos de parada) e segmentos de conexão (ruas e avenidas). A partir de então, as vias de interesse são selecionadas na base de dados para posterior monitoramento e comparação de operação considerando transporte público e privado.

Transporte privado

A análise do tráfego de veículos privados é feita graças ao apoio fornecido pelo Waze no presente projeto. Acessando dados em tempo real por meio de APIs, tem-se acesso aos dados coletados a partir do aplicativo, o que configura um processo de *crowdsourcing.*

Ressalta-se que a parceria com o Waze dá acesso a dados que requerem menos processamento, por serem disponibilizados em um formato mais próximo do que é requerido para uso. Verifica-se, então, o tempo de percurso nas vias de interesse no instante da consulta, bem como a ocorrência de eventos (congestionamentos ou acidentes).

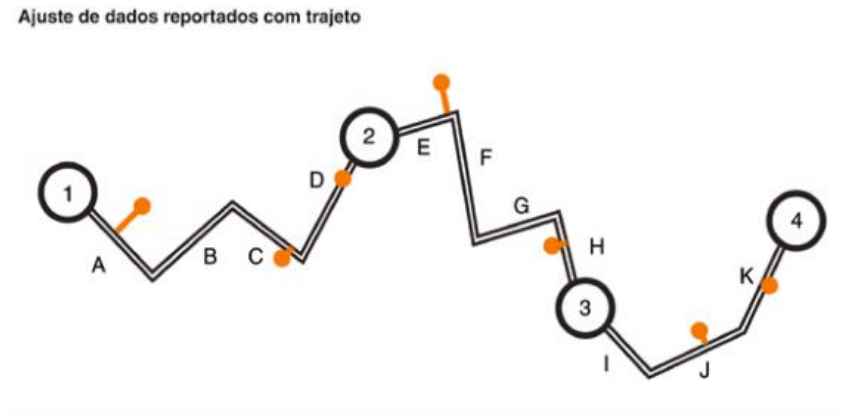
Transporte público

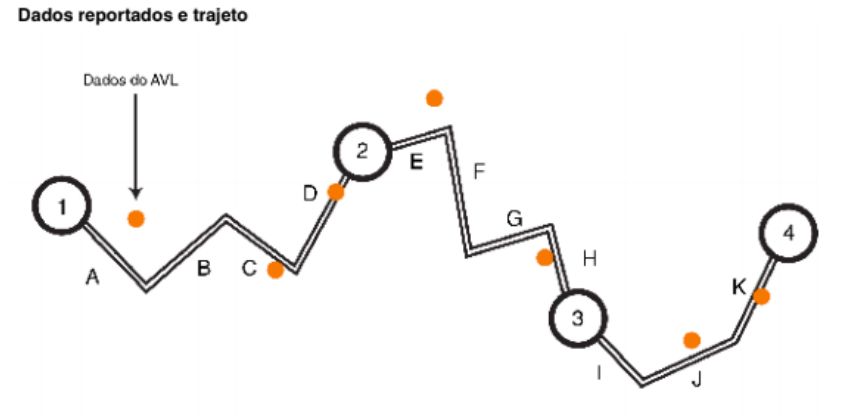
No caso do transporte público, as posições dos veículos em circulação, na forma de coordenadas geográficas, são recebidas em tempo real. Para o projeto em questão, foram acessadas APIs disponibilizadas nos endereços abaixo:

- São Paulo: http://olhovivo.sptrans.com.br/
- Montevideo: https://api.montevideo.gub.uy/

Os dados, essencialmente, são compostos por um identificador do veículo que o transmitiu, o instante da transmissão, a respectiva posição (no formato de latitude e longitude) e a rota que está sendo realizada. No entanto, esta informação não se relaciona diretamente com o traçado referente à viagem realizada pelo veículo.

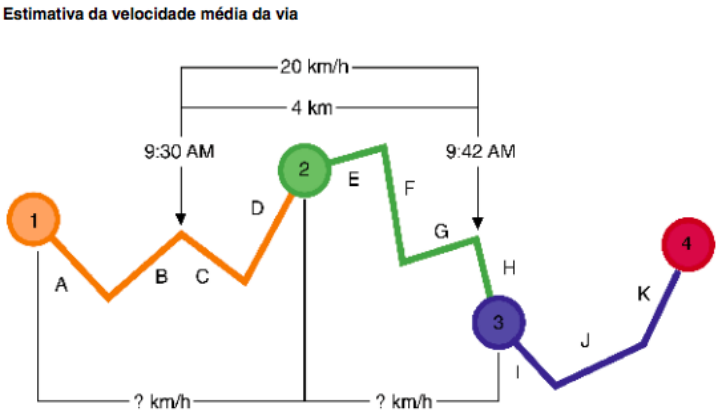
Sendo assim, é necessário posicionar adequadamente o ponto recebido em relação à geometria que descreve o trajeto realizado pelo veículo, aproximando analiticamente a coordenada geográfica ao segmento correspondente da rota a partir da distância ortogonal entre estes dois elementos.





O processamento descrito permite que se verifique a qualidade do dado a partir de diversos parâmetros geométricos. Este ajuste atesta, portanto, se o dado recebido é válido para fins de monitoramento da rede, o que elimina o uso de coordenadas contendo erros significativos durante a medição. O descarte é feito também nos casos em que há grande latência na transmissão dos dados.

A velocidade de um veículo é calculada quando ocorrem duas (ou mais) transmissões sucessivas de dados, conforme apresenta a figura abaixo.



Tendo em vista que a transmissão de dados é aleatória e, por isso, não ocorre sobre os pontos (nós) tidos como referência na rede, a velocidade calculada inicialmente não corresponde à velocidade de operação em um dado circuito.

No exemplo anteriormente apresentado, tem-se uma posição transmitida entre os pontos 1 e 2, enquanto o dado imediatamente posterior está entre os pontos 2 e 3. Portanto, a velocidade calculada (no caso, 20 km/h) não corresponde à velocidade de operação no trecho 1-2, nem no trecho 2-3. Visando a padronização no processamento de dados provenientes de diversos carros, são feitas interpolações em relação aos nós (extremidades dos circuitos) da via para adequar o cálculo da velocidade aos trechos previamente definidos.

Assim, a velocidade em um trecho passa a ser computada a partir dos diversos veículos que circulam nele, melhorando a amostragem e, consequentemente, a confiança sobre sua condição de tráfego.

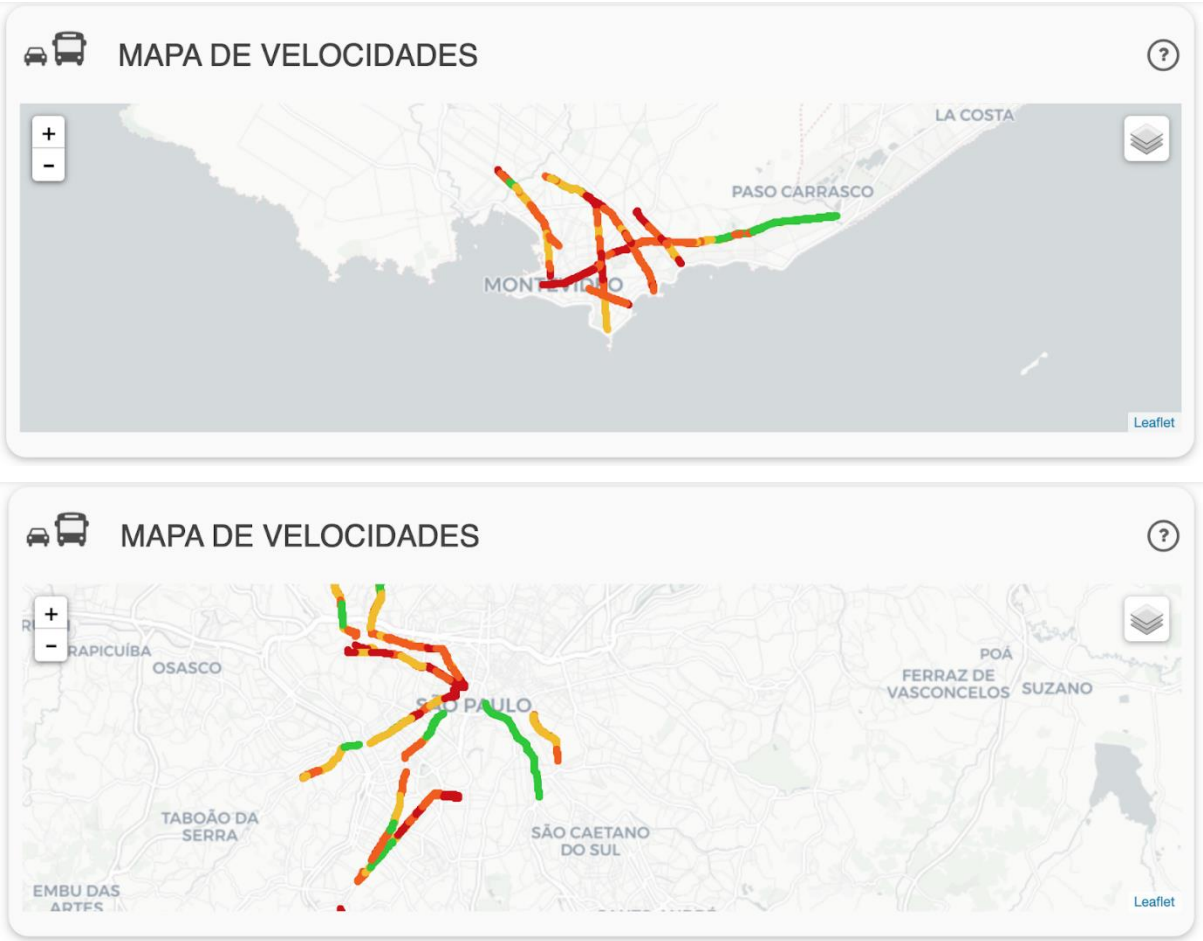
4.3.2.3 Visualização

O acesso ao painel de cada cidade pode ser feito pelos endereços a seguir:

- São Paulo: http://smt.scipopulis.com:5007/comparativo/saopaulo
- Montevideo: http://smt.scipopulis.com:5007/comparativo/montevideo

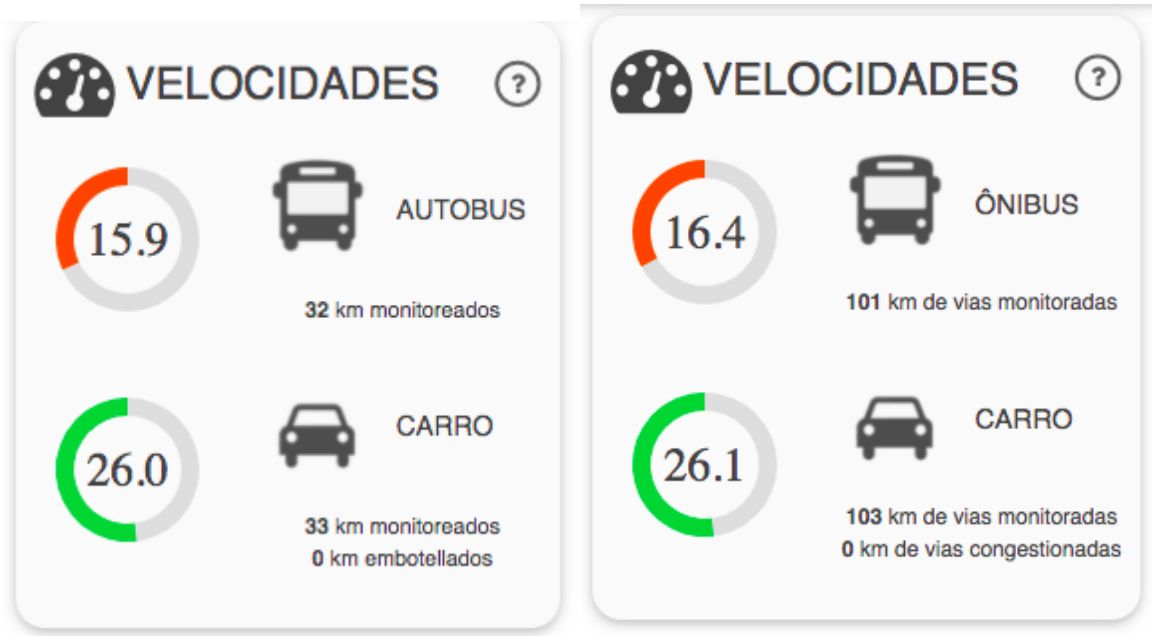
O painel é composto por diversos cartões, que apresentam em tempo real os dados anteriormente mencionados. No topo, o mapa mostra as vias selecionadas para monitoramento sobre o viário da cidade. Conforme demonstra a escala, a cor associada ao traçado refere-se ao intervalo que contém o valor de velocidade no instante da observação.

Figura 4.3.2.3.1: Interface do aplicativo



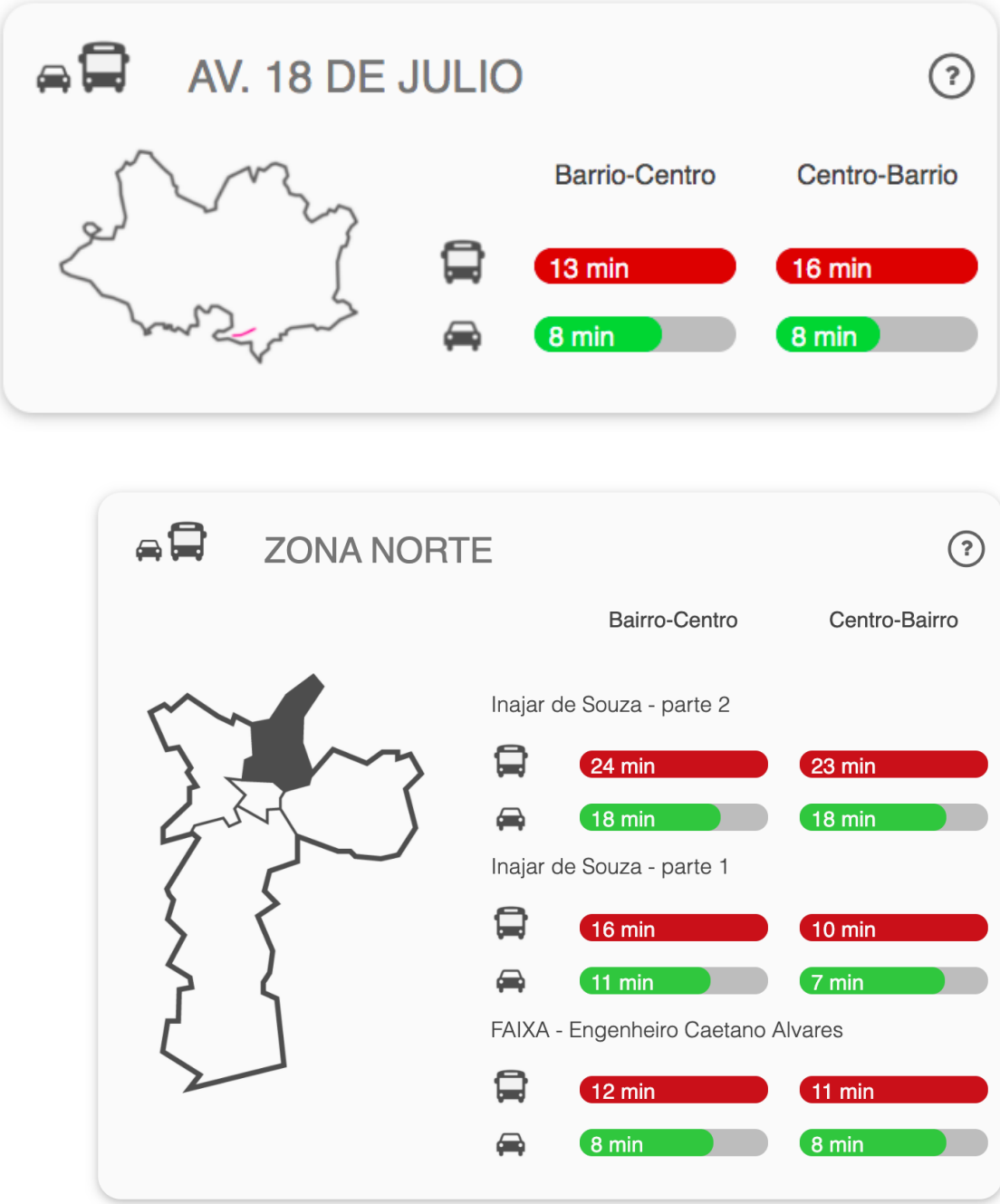
Ao lado do mapa, ainda considerando todas as vias sob monitoramento, tem-se uma comparação de forma global das velocidades calculadas pelos dados do Waze e da agência de transporte público municipal. Além disso, é possível também verificar a quilometragem monitorada em tempo real na cidade, bem como a ocorrência de congestionamentos nos trechos.

Figura 4.3.2.3.2: Painel de velocidades



Por fim, os demais cartões apresentam de forma comparativa o tempo de percurso em cada uma das vias mapeadas, considerando os sentidos de tráfego e o uso de transporte público (ônibus) e individual. As cores entre as barras indicam quão significativa é a diferença entre o tempo que as duas classes de veículo tomam para completar o trecho. Isto é, barras em amarelo indicam que o desempenho é similar, enquanto barras que se contrapõem (vermelho e verde) demonstram que o modo em verde percorre em tempo expressivamente menor o segmento.

Figura 4.3.2.3.3: Comparação do tempo de percurso entre vias

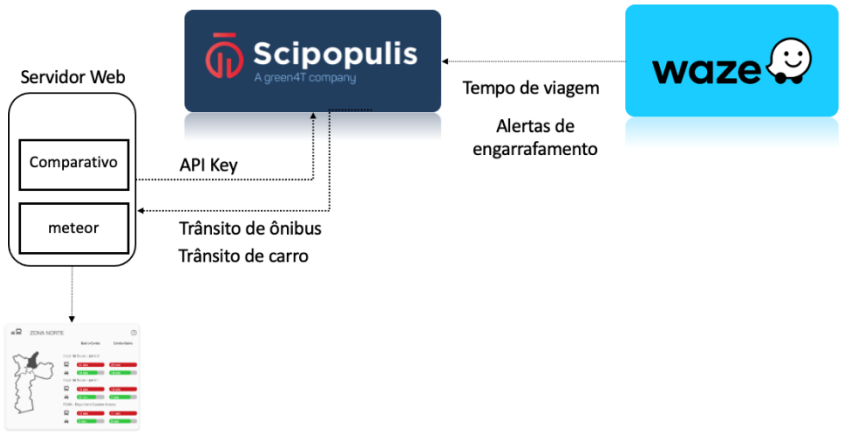


4.2.3.4 Implantação

O painel comparativo entre ônibus e carros foi desenvolvido utilizando o framework javascript Meteor (www.meteor.com), com a biblioteca Materialize para os elementos gráficos

(materializecss.com) e a biblioteca D3 para a visualização dos dados (d3js.org). Ele consome dados de duas APIs: a API de previsão de tempos de viagem de carros fornecida pelo Waze (waze.com) e a API de previsão de tempos de viagem de ônibus fornecida pela Scipopulis (scipopulis.com). O diagrama abaixo descreve a arquitetura básica do sistema.

Figura 4.2.3.4.1: Arquitetura do sistema



O código está disponível no seguinte repositório: bitbucket.org/scipopulis/bid-comparativo. Ele contém um arquivo README com instruções detalhadas para instalação e execução do projeto. São necessárias quatro configurações, conforme lista abaixo.

1. **Configuração das chaves de acesso à API da Scipopulis** a API que alimenta o painel requer uma chave de autenticação para permitir acesso aos dados. Essa chave é específica para cada cliente da API, e deve ser configurada em um arquivo à parte.
2. **Configuração de usuário e senha para acesso ao site** caso se deseje que o acesso ao site seja feito mediante usuário e senha, deve-se configurar o usuário e senha em um arquivo de configuração específico para esse fim, como descrito no README do projeto.
3. **Configuração de variáveis de ambiente para implantação** as configurações de variáveis de ambiente para implantação do sistema devem ser feitas utilizando-se o `pm2-meteor`. Os arquivos de configuração se encontram no diretório `deploy` do projeto. É possível criar diversas configurações para implantação em diferentes ambientes. Por exemplo, se for necessário realizar a implantação em um ambiente de homologação para

validação antes da passagem a um ambiente de produção, é possível criar um diretório `deploy/homolog` com as configurações do ambiente de homologação e um diretório `deploy/prod` com as configurações do ambiente de produção. O projeto contém um arquivo `deploy/prod/pm2-meteor.json` de exemplo.

4. **Configuração do script de deploy** o script `deploy.sh`, localizado no diretório raiz do projeto, é responsável por se conectar ao servidor via ssh e realizar a implantação do projeto. Para configurar o script, é necessário informar nas primeiras linhas o endereço do servidor, o usuário que tem permissão para se conectar via ssh, e o diretório onde se encontra sua chave privada para acesso ao servidor. O script se conecta ao servidor de destino, recupera o código do repositório e executa uma série de comandos para configuração do sistema e implantação do projeto. É possível designar qual o ambiente de implantação passando como parâmetro o nome das configurações do ambiente. No exemplo descrito no item 3, para realizar a implantação no ambiente de produção basta executar a linha `./deploy.sh prod`, enquanto se o objetivo for implantar no ambiente de homologação, basta executar o comando `./deploy.sh homolog`.

Importante notar que, como o script de implantação recupera as versões mais recentes diretamente do repositório de código, é fundamental realizar o `commit` das mudanças e enviar as mudanças ao repositório antes de executar o script de implantação. Caso as mudanças sejam realizadas apenas localmente na máquina do desenvolvedor e não forem enviadas ao repositório, o código copiado para o servidor de implantação não estará atualizado e as mudanças não serão efetuadas.

4.4 Apoio à Gestão do Trânsito

4.4.1 Apoio à Gestão do Trânsito - Índice de Congestionamento

A população urbana aumentou exponencialmente - de 751 milhões em 1950 para 4,2 bilhões em 2018 - e continuará essa tendência (ONU 2008). O processo de urbanização gera impactos no modo como os indivíduos se relacionam com a cidade, sobretudo em como gerenciar de maneira apropriada a escalabilidade na provisão de serviços e circulação de bens para que o desenvolvimento seja sustentável.

O efeito deste aumento implica prestar atenção a muitos aspectos de sustentabilidade urbana, no entanto, enfatizamos o aspecto da mobilidade e a gestão de tráfego.

Um dos fenômenos presentes na maioria das cidades de médio e grande porte em todo o mundo é o congestionamento veicular. Na União Europeia estima-se que o custo do tempo perdido no trânsito foi equivalente a 1,4% do PIB da região (Van Essen, et al. 2019) enquanto os Estados Unidos apontam perdas na ordem de 0,7% do PIB nacional (CEBR 2014). Apesar de existirem poucos estudos de congestionamento urbano na América Latina (AL) avaliando os impactos e os custos sociais, índices como o elaborado pela INRIX, mostrou que em 2020, das dez cidades mais congestionadas no mundo, três são latino-americanas: Bogotá (Colômbia), Quito (Equador) e Cali (Colômbia). O ranking do indicador global de tráfego elaborado pela empresa Tom Tom, elencou em 2020 Bogotá como a terceira cidade mais com maior nível de congestionamento e Lima (Peru) ficou na 15ª posição.

Os determinantes de congestionamento nas cidades latino americanas podem ser de diversas naturezas, sendo no nível macro principalmente afetada pelo aumento da taxa de urbanização, a infraestrutura viária que favorece o transporte individual, a baixa acessibilidade e interoperabilidade do transporte público urbano, além dos desafios em financiar o seu desenvolvimento, (Calatayud, et al. 2021)

O fenômeno de congestionamento veicular pode ser recorrente ou não recorrente, sendo o rendimento medido principalmente pela observação da velocidade da via, fluxo, densidade, comprimento e duração da fila. Geralmente, os resultados de monitoramento são gerados por inspeção visual de Circuito Fechado de TV (CFTV) ou em pontos de observação estratégicos. Com o advento de novas tecnologias digitais de tráfego urbano, as cidades podem contar com um abundante volume de dados em tempo *quasi*-real. Este atual cenário traz novas perspectivas de gestão e principalmente na tomada de decisão, seja no nível operacional, planejamento ou de políticas públicas.

Neste trabalho foi desenvolvido um protótipo de plataforma de dados provenientes da *Application Programming Interface* (API) do Waze para as cidades: São Paulo, Montevidéu, Quito e para o distrito de Miraflores, em Lima. Devido à baixa disponibilidade de dados, não foi desenvolvido protótipo para a cidade de Xalapa. Os dados são provenientes de duas fontes principais: dados armazenados e pré-processados na nuvem da Amazon Web Services (AWS) e diretamente da API do Waze. O primeiro fornece dados para visualização da série histórica e o último em tempo

quasi-real. Foi utilizado como referência o método utilizado pela Companhia de Engenharia de Tráfego de São Paulo (CET) para validação do método proposto. O método utilizado pela CET foi escolhido pela disponibilidade de dados da série histórica do índice de congestionamento de São Paulo. Além disso, o método é invariante ao tipo de cidade, ou seja, a sua aplicabilidade pode ser facilmente estendida e desta forma possibilita a validação dos resultados e ajustes do modelo para as demais cidades.

Esta plataforma não estava contemplada no Termo de Referência 8 (TR8) e foi adicionada ao produto atual após ser identificada a necessidade da criação de uma ferramenta para gestão de trânsito e dos níveis de congestionamento junto com os técnicos e representantes da Prefeitura de São Paulo.

4.4.1.1 Objetivos e Método de Análise

O objetivo geral da atividade é validar se os dados gerados pelo aplicativo de navegação colaborativa Waze são capazes de descrever o congestionamento para as cidades de São Paulo, Montevidéu, Quito e para o distrito de Miraflores em Lima.

Aplicaram-se métodos de análise exploratória de dados e foram observadas a correlação entre a curva real de tráfego e a curva obtida pelos dados do aplicativo. Para a curva real foram utilizados os dados de observação utilizados pela Companhia de Engenharia de Tráfego (CET) de São Paulo. A escolha é devido a maturidade do modelo de gestão de tráfego em São Paulo, a validação com analistas experientes que acompanham o indicador de lentidão, a disponibilidade de dados históricos para comparação, e a familiaridade com os dados do aplicativo Waze já em uso na Companhia em outros serviços de monitoramento. O método da CET é invariante quanto ao tipo de cidade, podendo ser aplicado ou utilizado como validação nas demais cidades de nosso estudo.

No caso dos dados serem minimamente satisfatórios em termos de qualidade e significativa representação da condição de tráfego, a etapa seguinte é o desenvolvimento de um protótipo para gestão de tráfego das cidades. O protótipo possui as seguintes premissas:

- Usabilidade e foco na experiência do usuário;
- Código aberto;

- Interface que proporcione o diagnóstico do tráfego utilizando os dados históricos e tempo real;
- Portável;
- Extensível;
- Adaptável;
- Simples execução e manutenção;
- Escalabilidade da monitoração;
- Baixo custo.

O protótipo não representa um produto final, pois dependerá da infra-estrutura que cada cidade disponibiliza em ambiente de produção e adaptações serão necessárias. As principais adaptações são detalhadas nos capítulos seguintes.

4.4.1.2 O Caso da Cidade de São Paulo - Brasil

A medição do congestionamento no trânsito da cidade de São Paulo teve início em meados de 1980, sendo um importante indicador de desempenho, que registra nos dias úteis, no horário entre 07 e 20 horas, sua extensão nas vias monitoradas pela Companhia de Engenharia de Tráfego de São Paulo (CET-SP). Este indicador é amplamente divulgado por órgãos da imprensa e consolidou-se como uma referência dos padrões da qualidade do trânsito na cidade, sendo utilizado também para a elaboração de diversos estudos técnicos e estatísticos.

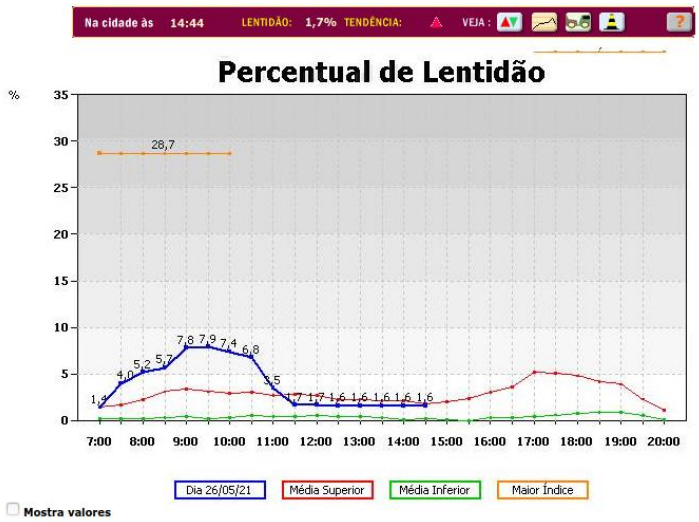
Inicialmente as informações eram coletadas por inspeção visual de funcionários alocados no topo de altos edifícios estrategicamente localizados na cidade, também conhecido por Postos Avançados de Campo (PAC). Atualmente, a coleta é realizada por meio de sistema de câmeras de Circuito Fechado de Televisão (CFTV) e as imagens são transmitidas para as Centrais de Monitoramento, onde são realizadas análises e o envio de orientações aos agentes operacionais.

Até meados de 2007 o índice de congestionamento do trânsito era representado em quilômetros de vias monitoradas, posteriormente a CET passou a apresentar o indicador no formato percentual, representado pela relação entre a extensão em quilômetros de vias congestionadas e o total de quilômetros monitorados que atualmente é de 868 km.

$$\text{Indice de congestionamento} = \frac{\text{Vias congestionadas}}{\text{Vias monitoradas}} 100$$ **Equação 1**

De acordo com a CET-SP, o sistema viário monitorado é contabilizado considerando a dupla mão de circulação, nas vias de sentido único ou pistas segregadas com canteiro no mesmo sentido de circulação, por exemplo, pista local/central/expressa das Marginais, contabiliza-se a extensão total de cada pista.

Figura 4.4.1.2.1: Gráfico do Índice de Congestionamento ou Lentidão (%)

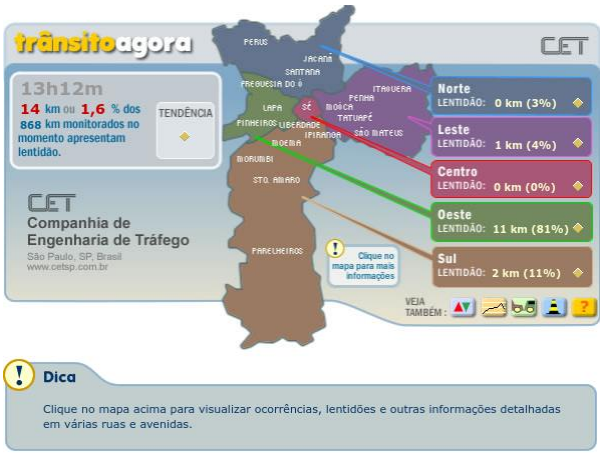


Fonte: CET/SP

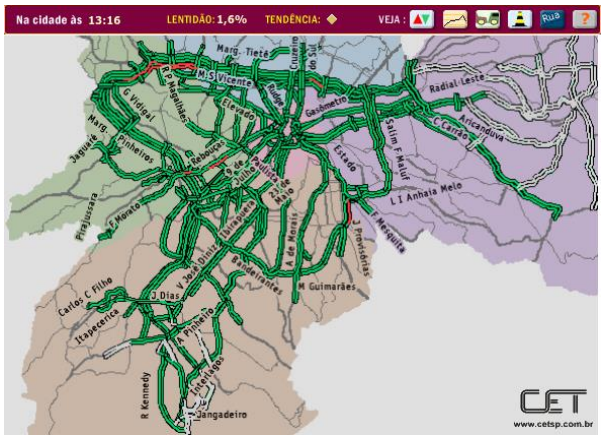
O indicador calculado a cada 30 minutos é comparado com a série histórica do mesmo dia nos 12 meses anteriores, Figura 4.4.1.2.1. No cálculo estão descartados os valores dos meses de janeiro, fevereiro, julho e dezembro, feriados e emendas de feriados e valores cuja diferença em relação ao valor médio ultrapassa 1,5 desvio padrão. Os limites inferior e superior correspondem a um desvio padrão das amostras selecionadas e representa o intervalo de valores do indicador considerados aceitáveis, quando não há ocorrência de grande impacto no trânsito.

A CET-SP também disponibiliza as informações de trânsito em sua *homepage* no formato de mapas, tabelas e gráficos, Figura 4.4.1.2.2. A informação de lentidão do tráfego é detalhada por região da cidade—Norte, Sul, Leste, Oeste e Centro— além de informações de incidentes nas vias.

Figura 4.4.1.2.2: Mapa de Fluidez e Detalhes dos Status das Vias Monitoradas



a. Mapa de fluidez do trânsito



b. Detalhe da Região Oeste

Ordem:
Corredor
Tamanho

Na cidade às 14:47 LENTIDÃO: 1,7% TENDÊNCIA: VEJA :

Lentidão

Corredor	Sentido	Via	Local	Tamanho (m)
Marginal Tietê	Castelo/A.Senna	Loc	De 760 m antes de Freguesia do O até 700 m depois de Nova Fepasa	3390
		Exp	De 640 m antes de Velha Fepasa até Gastao Vidigal	2610
Marginal Tietê - Pista Central	Castelo/A.Senna		De 861 m antes de Freguesia do O até Bandeirantes	2139
Rebouças/ Eusébio Matoso, Av	Bairro/Centro		De Oscar Freire até Antonio Rosa	1550
Juntas Provisórias, R das	Ipiranga/Vila Prudente		De Grande Sao Paulo até Taylor	1500
Marginal Tietê	Castelo/A.Senna	Exp	De 89 m antes de Vila Guilherme até 450 m depois de Cruzeiro do Sul	1061
Max Feffer Túnel (Cidade Jardim)	Bairro/Centro		De Desemboque até Emboque	810
Radial Leste - DEC MO	Centro/Bairro	Exp	De Pires do Rio Término até Alvaro Ramos	800
Radial Leste - DEC BR	Centro/Bairro		De Tuiuti até Pires do Rio	690
Rebouças/ Eusébio Matoso, Av	Centro/Bairro		De Antonio Sabino Pte Eusebio Matoso até Arcoverde	540
Marginal Tietê	Castelo/A.Senna	Exp	De Remedios até 640 m depois de Rod. Castelo Branco	60

c. Detalhamento do trecho congestionado

Lentidão nos Principais Eixos

Eixo	Sentido	Lentidão (Km)	% em Relação à Lentidão da Cidade	Tendência
1 - Marginal Tietê	A.Senna/Castelo	0,0	0,0 %	◆
	Castelo/A.Senna	9,3	64,5 %	◆
2 - Marginal Pinheiros	Castelo/Interlagos	0,0	0,0 %	◆
	Interlagos/Castelo	0,0	0,0 %	◆
3 - Bandeirantes	Imigrantes/Marginal	0,0	0,0 %	◆
	Marginal/Imigrantes	0,0	0,0 %	◆
4 - Eixo Norte-Sul	Aeroporto/Santana	0,0	0,0 %	◆
	Santana/Aeroporto	0,0	0,0 %	◆
5 - Eixo Leste-Oeste	Lapa/Penha	0,7	4,8 %	▲
	Penha/Lapa	0,0	0,0 %	◆

d. Congestionamento por eixo viário, indicadores e tendência do tráfego

Fonte: CET/SP

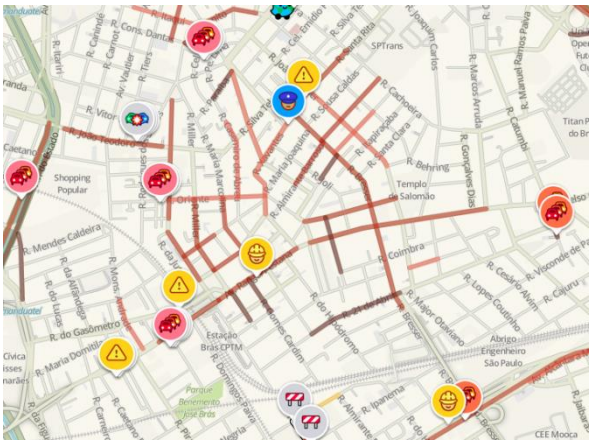
* http://cetsp1.cetsp.com.br/monitransmapa/agora/

Além disso, os dados de congestionamento fornecem a tendência do comportamento do tráfego comparando com o registro imediatamente anterior. O resultado da diferença entre os pares de dados consecutivos fornece a classificação de tendência alta, estável e baixa.

4.4.1.3 Waze API

O Waze é uma empresa de tecnologia que fornece o serviço de navegação colaborativa. Além de expor informações em tempo real das condições de trânsito nas cidades, também fornece informações de status dos trechos de via com a classificação de congestionamento, incidentes, acidentes graves e sugestões de rotas, Figura 4.4.1.3.1.

Figura 4.4.1.3.1: Waze Map e Indicação das Vias Congestionadas, Bloqueios de Vias e Alertas.



Fonte: Waze

Por ser uma ferramenta colaborativa, onde os usuários compartilham seus dados de geolocalização e notificam os alertas de incidentes, a qualidade dos dados está diretamente relacionada ao volume de usuários cadastrados na aplicação. Em 2019, São Paulo possuía aproximadamente 4,5 milhões de usuários ativos. A impressionante cifra foi um dos motivos da escolha da cidade na comparação e validação dos dados fornecidos pela API do Waze.

As informações coletadas são adicionadas nas camadas tendo como base o mapa viário da cidade. Cabe ressaltar que o monitoramento do Waze é feito em todas as vias da cidade. O mapa é representado por um objeto matemático chamado de grafo. O grafo é composto de vértices e arestas que se conectam, simulando o viário da cidade. Na Figura 4.4.1.3.2, utiliza-se uma hipotética avenida para identificar os nós representados pelas circunferências em cinza, e as arestas, designando os segmentos. Os segmentos de uma mesma via podem possuir nomenclaturas diferentes, por exemplo, a Avenida Simón Bolívar, em Quito, possui os seguintes

segmentos: Av. Simón Bolívar (Sur), Av. Simón Bolívar (Norte), Paso Deprimido Av. Simón Bolívar. Além disso, as vias podem ter segmentos paralelos, como é o caso das pistas da Marginal Tietê, em São Paulo.

Figura 4.4.1.3.2: Grafo Representando uma Hipotética Avenida

Fonte: Elaboração Própria

Analisar os segmentos do mapa viário do Waze é fundamental para obter uma tabela de conversão da nomenclatura das vias. Desta forma cada cidade poderá decidir se quer incluir o conjunto dos segmentos ou apenas um trecho específico de via para monitoramento.

Para análise de congestionamento utilizamos apenas os dados armazenados pela Waze *Traffic jams* que consiste nas informações de desaceleração de tráfego com base na localização e velocidade dos usuários. Os campos do arquivo são populados e expostos pela API através de uma chamada utilizando o protocolo HTTP. Os campos utilizados na elaboração da prova de conceito estão identificados na Tabela 4.4.1.3.1.

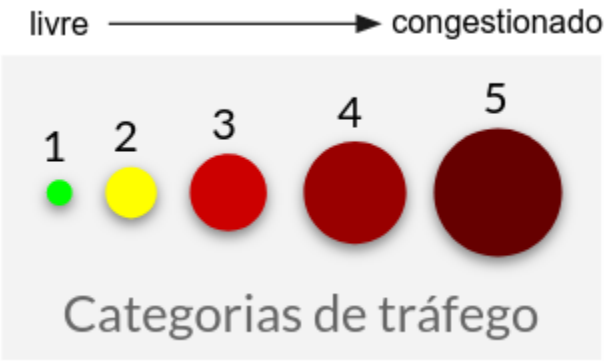
Tabela 4.4.1.3.1: Campos Utilizados na Comparação Entre CET e Waze Extraídos das Informações Geradas pela Waze *Traffic Jams*

Nome	Tipo	Descrição
`pubMillis`	Timestamp	Unix time
`line`	Latitude e Longitude	Linestring
`length`	Integer	Comprimento em metros
`street`	string	Nome da via
`city`	string	Nome da cidade
`country`	string	Nome do país
`level`	Integer	Categoria de congestionamento (0= fluxo livre - 5= Bloqueado)

Fonte: Elaboração Própria

A extração dos dados foi realizada na tabela de dados disponibilizada pela Fundação Getúlio Vargas (FGV) que os armazena em uma nuvem dedicada para o projeto *Big Data* para o Desenvolvimento Urbano Sustentável. Os dados representam uma fotografia do viário minuto a minuto, sendo os dados crus processados antes do armazenamento.

Figura 4.4.1.3.3: A Classificação de Congestionamento do Waze segue Cinco Categorias Distintas Variando do Fluxo Livre ao Congestionado.



O campo `level` é essencial na comparação com os dados da CET, Fig. 3.1.3.3. O Waze distingue o congestionamento em cinco categorias:

- **Categoria 1:** Fluxo livre, os veículos se locomovem no limite da velocidade da via;
- **Categoria 2:** Tráfego leve, pequenas variações de velocidade, mas sem comprometimento do fluxo;
- **Categoria 3:** Tráfego moderado, iminência de congestionamento;
- **Categoria 4:** Tráfego pesado, redução de velocidade e atrasos significativos;
- **Categoria 5:** Tráfego parado, veículos com significativa redução de velocidade ou completamente parados.

Os campos `length, street e line` contribuem para detecção de segmentos únicos no grafo e evitar redundância no somatório do congestionamento.

4.4.1.4 Comparação CET e Waze

Para detectar o efeito do tráfego selecionamos uma sexta-feira, véspera de Carnaval, do ano de 2019 (01/03/2019). O motivo é por ser uma data de grande movimentação nas vias da cidade o que contribui na detecção dos picos de congestionamento e por não ter sido influenciada pela restrição da mobilidade urbana devido a pandemia de COVID-19. Além disso, de acordo com o Instituto Nacional de Meteorologia (INMET) não houve a eventos climáticos extremos, sendo observado apenas uma precipitação máxima de 11mm às 5:00am, temperatura média de 21ºC, e umidade relativa média de 85%.

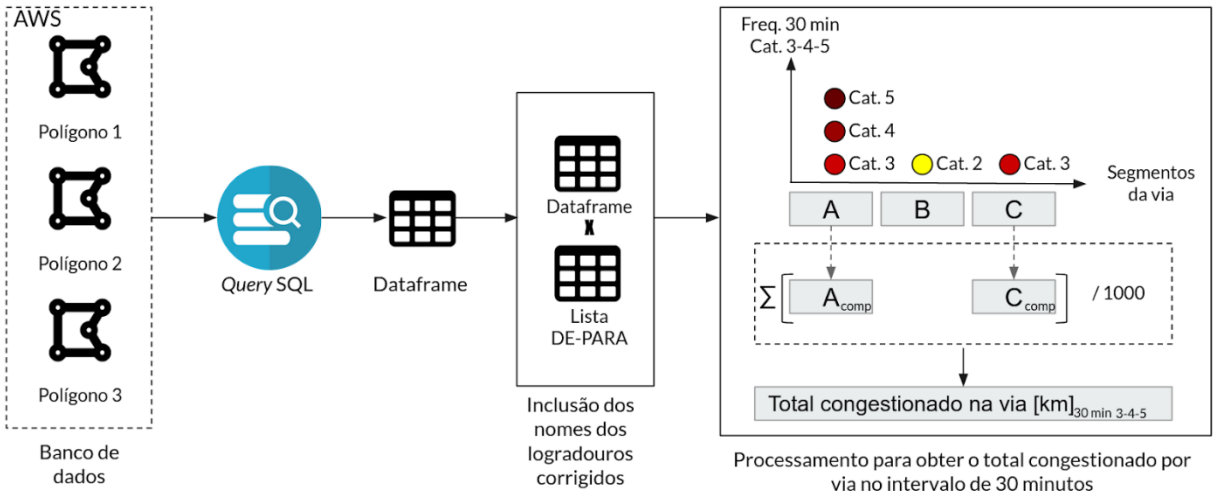
As categorias utilizadas pelo aplicativo Waze para classificação do tráfego foram divididas em dois grupos:

- **Grupo 1**: agrupa as categorias 3, 4 e 5;
- **Grupo 2**: agrupa as categorias 4 e 5 somente.

A categoria 3 na classificação do Waze representa uma zona de transição entre o tráfego livre e o intenso, portanto a inclusão desta categoria é para entender se a inspeção visual realizada pela CET está mais próxima da categoria 3 ou 4 na detecção da iminência de congestionamento.

No exemplo abaixo (Figura 4.4.1.4.1), é mostrado o cálculo para o Grupo 1 (categorias: 3, 4 e 5). O Grupo 2 é obtido bastando excluir os resultados menores que a categoria 4. O total congestionado é convertido em quilômetros que corresponde ao comprimento da fila no intervalo de tempo.

Figura 4.4.1.4.1: Fluxo do Processamento para Obtenção do Total Congestionado por Via na Janela de 30 Minutos.

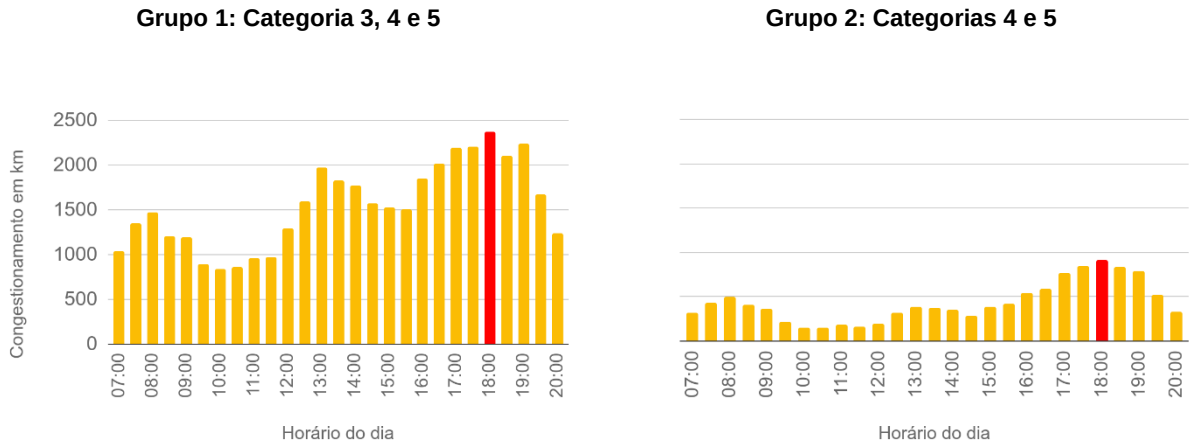


Fonte: Elaboração Própria

Os dados coletados minuto-a-minuto do Waze foram agrupados no intervalo de 30 minutos seguindo o modelo utilizado pela CET, Figura 4.4.1.4.1. As duplicidades de segmentos em cada intervalo foram removidas e obtemos o congestionamento para segmentos únicos para as categorias de interesse (3, 4 e 5). Em seguida, somaram-se os valores do comprimento de congestionamento e obteve-se uma aproximação do valor da fila em metros, posteriormente

convertido em quilômetros, para o intervalo de 30 minutos iniciando às 7 h até às 20 h, conforme Figura 4.4.1.4.2. Na figura, observa-se uma significativa diferença de magnitude entre os dois grupos e similares distribuições. Ambos registraram o pico às 18:00h.

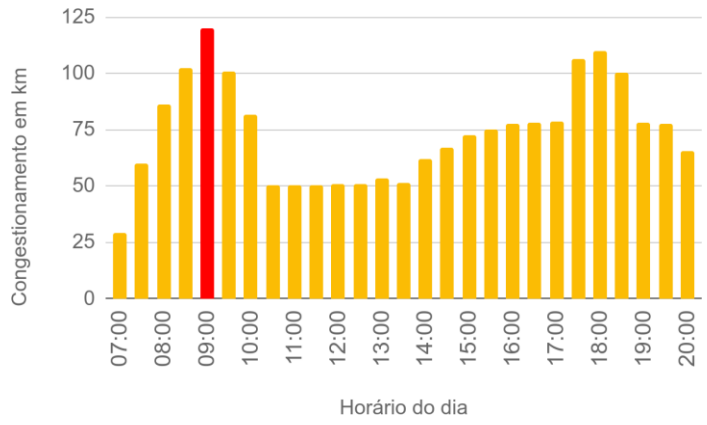
Figura 4.4.1.4.2: Grupo 1 E 2 Referentes ao Agrupamento das Categorias de Condição de Tráfego do Waze.



Fonte: Elaboração Própria

Cabe ressaltar que a categoria 3 pode inserir ruídos nos dados de congestionamento por ser uma fase intermediária entre o fluxo livre/leve e pesado/parado. Muitos segmentos são classificados como 3 pelo algoritmo do aplicativo, porém em poucos instantes mudam para categoria 2 ou 4. Este efeito gera a diferença da magnitude do congestionamento entre os grupos e uma limitação na reconstrução do status de tráfego no passado. No entanto, como estamos interessados na forma da curva e sua tendência, e menos nos valores absolutos, não temos um impacto significativo na análise e validação.

Figura 4.4.1.4.3: O Congestionamento Registrado pela CET apresentou um Pico às 9h e apresentou dois Picos Visíveis na Distribuição.



Fonte: Elaboração Própria

O pico máximo registrado pelos dados coletados pela API do Waze ocorreu às 18h para ambos os grupos (Figura 4.4.1.4.3). A CET registrou um pico máximo às 9h no valor de 120 km de fila e outro às 18h de 110 km, Fig. 3.1.4.3. Observa-se que os Grupos 1 e 2 também detectaram uma flutuação no entre-picos às 13h que foi levemente detectado pela CET. A sensibilidade da detecção de tráfego se dá pela quantidade de segmentos monitorados. A CET seleciona trechos de via ao passo que nos grupos observamos toda sua extensão. O resultado é observável na suavização da curva dos grupos que agrega a média de um conjunto de observações dos segmentos nos intervalos de 30 minutos, enquanto a CET extrai uma observação por segmento monitorado no mesmo intervalo.

Para comparar os picos, foram selecionadas as duas vias mais congestionadas registradas às 9h e às 18h. Na Tabela 4.4.1.4.1, o Grupo 1 capturou as mesmas vias em comparação com a CET às 9h, apenas a ordem está inversa. O Grupo 2 registrou a Marginal Pinheiros e a Av. dos Bandeirantes. A Av. dos Bandeirantes teve 7 km de congestionamento, de acordo com os dados CET. Na Tabela 4.4.1.4.2, observamos o congestionamento às 18h, e notamos que a correspondência foi exata entre os Grupos 1 e 2 com as vias detectadas pela CET.

Tabela 4.4.1.4.1: Duas Primeiras Vias Mais Congestionadas às 9h pelos Grupos 1, Grupo 2 e CET

Grupo 1	Grupo 2	CET
Via	Via	Via
Marg. Pinheiros	Marg. Pinheiros	Marg. Tietê
Marg. Tietê	Av. dos Bandeirantes	Marg. Pinheiros

Fonte: Elaboração Própria

Tabela 4.4.1.4.2: Duas Primeiras Vias Mais Congestionadas às 18h Pelos Grupos 1, Grupo 2 e CET

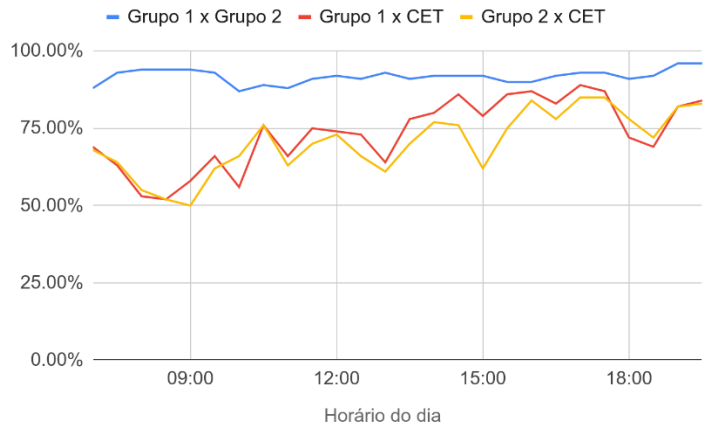
Grupo 1	Grupo 2	CET
Via	Via	Via
Marg. Tietê	Marg. Tietê	Marg. Tietê
Marg. Pinheiros	Marg. Pinheiros	Marg. Pinheiros

Fonte: Elaboração Própria

Como os valores absolutos de congestionamento não são diretamente comparáveis, calculou-se a correlação para obter a relação linear entre as curvas. Foram selecionados 30 dias do ano de 2019 em diferentes meses e obteve-se a média da correlação dos congestionamentos para cada intervalo de 30 minutos do dia das 7h até 20h, Figura 4.4.1.4.4. Em seguida, três comparações foram realizadas: Grupo 1 x CET, Grupo x CET, Grupo 1 x Grupo 2. Pode-se observar que a relação linear entre cada grupo com a CET é positiva, ficando acima de 50%. O Grupo 1, em

geral, teve um desempenho levemente melhor quando comparado ao Grupo 2. Um outro ponto importante é que a relação linear apresentou uma tendência crescente em direção ao final da tarde para as curvas do Grupo 1 x CET e Grupo 2 x CET. Isto pode estar relacionado com a quantidade de usuários conectados ao aplicativo ou na diferença de classificação do congestionamento entre o método do Waze e da CET no horário da manhã. Neste caso, para identificar a causa seria recomendável realizar um estudo mais detalhado comparando a classificação registrada no aplicativo e o que é detectado em campo por inspeção visual. No entanto, mesmo com algumas flutuações observamos que é possível utilizar a classificação do Waze como uma aproximação do congestionamento real. A correlação entre os grupos ficou acima de 88% com uma certa estabilidade ao longo do dia, indicando que o Grupo 1 e o Grupo 2 possuem forte relação linear positiva.

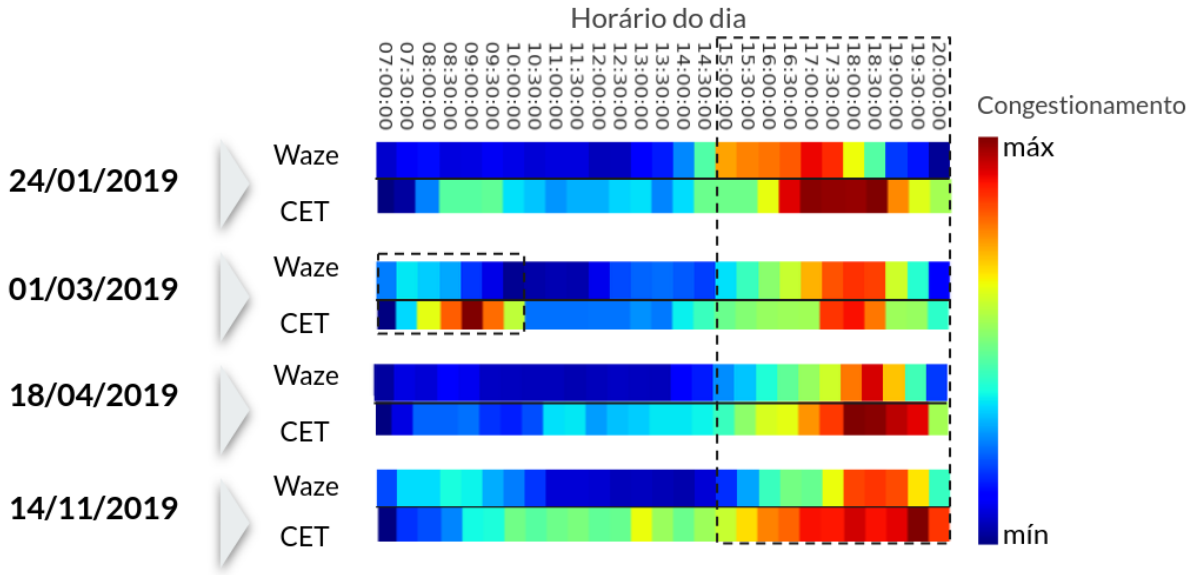
Figura 4.4.1.4.4:Correlação dos Grupos 1, 2 e CET. Os Grupos 1 e 2 Apresentaram Relação Linear Positiva com os Dados da CET.



Fonte: Elaboração Própria

Na Figura 4.4.1.4.5, foram selecionadas quatro vésperas de feriados em dias úteis somente no Grupo 2 que representa tráfego pesado/parado. Foi observado que os intervalos de duração do congestionamento nos dias selecionados estão sobrepostos aos dados coletados da CET. Apenas o dia 01/03/2019, conforme mencionado anteriormente, registrou pico máximo no período da manhã. O mapa de calor reforça a utilização dos dados do Waze como uma fonte confiável de detecção de congestionamento da cidade. Na figura abaixo, pode-se observar que os picos estão sobrepostos no intervalo de duração do congestionamento indicando que o Waze consegue identificar filas nas vias.

Figura 4.4.1.4.5: Mapa de Calor do Congestionamento do Grupo 2 do Waze e CET.



Fonte: Elaboração Própria

Apesar dos resultados positivos no uso dos dados do Waze observados em São Paulo, reforçamos que a qualidade dos dados depende da base de usuários que o aplicativo possui em cada cidade. Obviamente, uma base menor pode ter uma janela de observação maior para que mais dados sejam incluídos em cada subconjunto de observações dos segmentos. Além disso, pode-se agregar os segmentos e obter uma classificação da via em vez de um trecho.

Um esforço adicional por parte dos gestores de tráfego das cidades é a correta associação dos nomes dos logradouros de cada trecho de via no Waze versus o que está registrado nos mapas da cidade. Em geral, os nomes possuem diferenças e a equivalência é fundamental para uma análise correta do tráfego.

Quanto à escolha do Grupo 1 ou 2, para a cidade de São Paulo ambos tiveram desempenho similar. Em diálogo com analistas da CET, um estudo interno na companhia revelou que o Grupo 2 seria mais indicado para uso em produção, já que o número de falsos positivos era relativamente alto quando incluíam a categoria 3 e cruzarem com os dados do CFTV no mesmo instante.

Protótipo de plataforma de dados

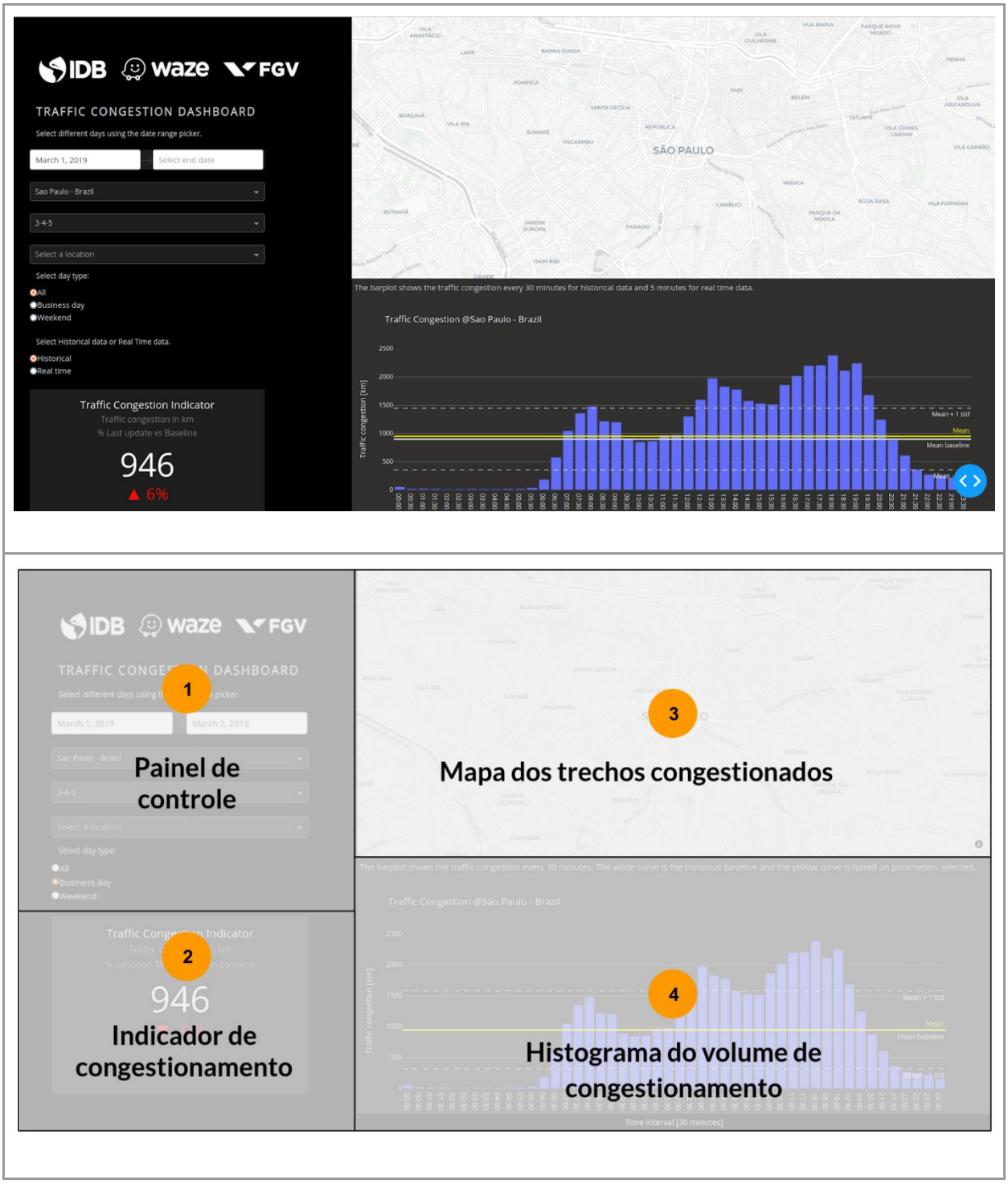
Uma vez validado o potencial de uso dos dados do Waze como gestão de tráfego urbano, o próximo passo foi elaborar uma prova de conceito que servisse de referência inicial para um produto mais robusto no futuro, Fig. 4.4.1.4.6.

O painel possui quatro divisões:

- **Controle** - filtros para definir o intervalo a ser analisado, a cidade, o grupo (1: categorias 3-5 do Waze e 2: categorias 4-5 do Waze), tipo do dia (útil, final de semana, ou ambos), a via monitorada e o tipo de dado (histórico ou tempo real).
- **Indicador de congestionamento** - representado pelo valor do congestionamento em quilômetros e um delta referente a diferença de uma linha de base que pode ser uma média ou uma curva histórica versus a curva real.
- **Mapa dos trechos congestionados** - aponta os segmentos congestionados no mapa para os dados históricos e em tempo real.
- **Histograma do volume de congestionamento** - para os dados históricos a agregação temporal é a cada 30 minutos, enquanto o tempo real é a cada 5 minutos. A curva de referência pode ser calculada seguindo um critério de análise da série histórica, mas para ilustração adotamos a média nos dados históricos e a curva de congestionamento da observação de um único dia.

Na figura abaixo, a visualização junta o trecho congestionado no mapa e um histograma para acompanhamento da flutuação do tráfego ao longo do dia. O indicador reflete a diferença da curva esperada para o dia ou período versus o observado.

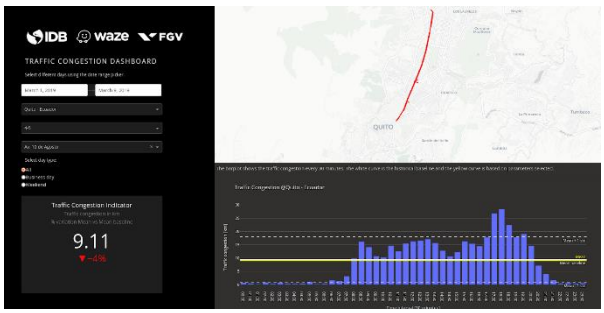
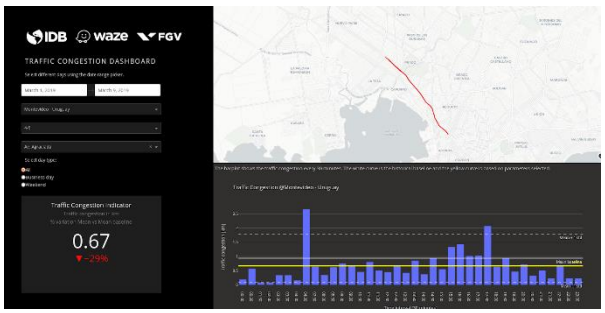
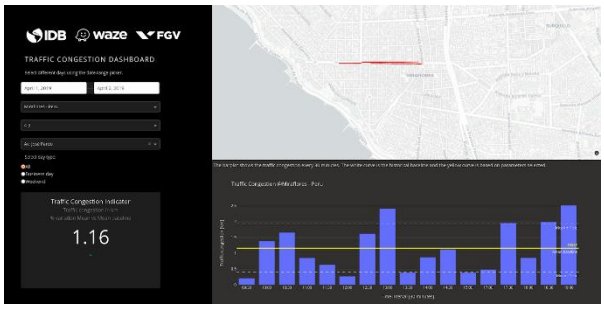
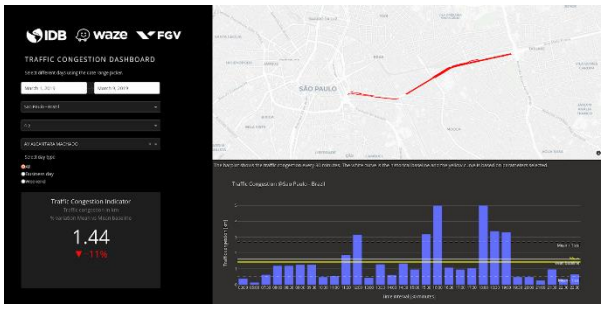
Figura 4.4.1.4.6: Painel de Gestão de Congestionamento.



Fonte: Elaboração Própria

Abaixo (Figura 4.4.1.4.7), exemplo de visualização de painéis construídos para as quatro cidades.

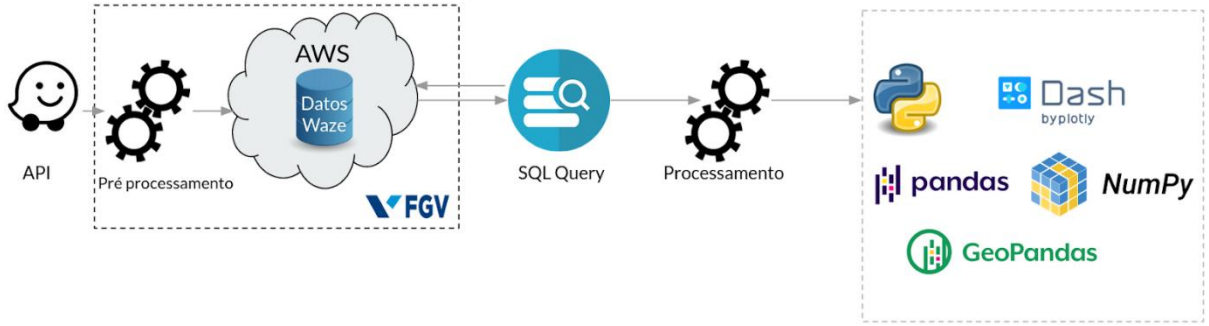
Figura 4.4.1.4.7: Painéis de Gestão para Quito, Montevidéu, Miraflores e São Paulo

	
Quito, Equador	Montevidéu, Uruguai
	
Miraflores, Peru	São Paulo, Brasil

Fonte: Elaboração Própria

Os dados seguem dois fluxos distintos: históricos e tempo real. Para os dados históricos a arquitetura inicia na captura dos dados na API do Waze via solicitação HTTP. Em seguida, é processado e inserido no bucket '`aws-athena-query-results-east-2`' e base de dados '`cities`', ambos na nuvem da FGV configurada na *Amazon Web Service* (AWS). Como utilizamos ferramentas *Open Source* no desenvolvimento do painel utilizamos o Python como linguagem de programação e as ferramentas Dash da Plotly, Pandas, Numpy e Geopandas, conforme ilustrado na Figura 4.4.1.4.8.

Figura 4.4.1.4.8 Arquitetura de Coleta, Processamento dos Dados Históricos e Ambiente de Desenvolvimento para Visualização dos Dados do Painel de Controle de Congestionamento.

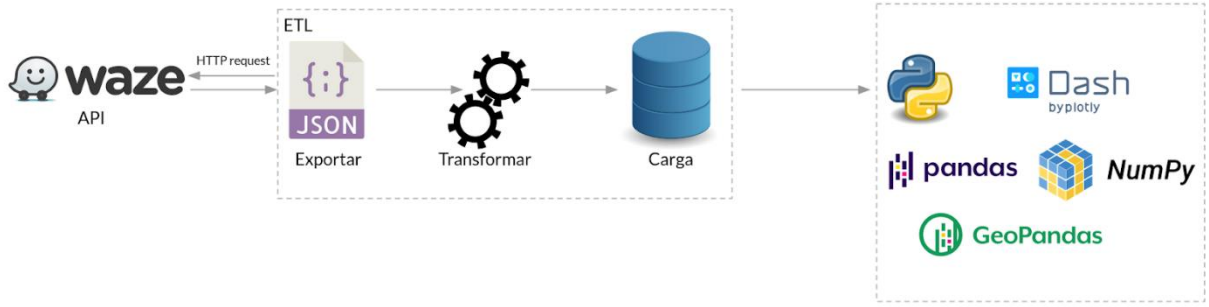


Fonte: Elaboração Própria

***** A lista de bibliotecas está no ANEXO.

Para acesso dos serviços da AWS fora do ambiente da nuvem é necessário instalar duas bibliotecas: Boto3 e AWScli. Boto3 acessa os serviços de nuvem da Amazon direto do IDE, bastando primeiramente configurar as credenciais de acesso. Para isto, usamos o AWScli para configurar os arquivos `~.aws/credentials` e `~.aws/config`. No primeiro arquivo é configurada a chave de usuário e chave de acesso e, no segundo, configurações de acesso à nuvem como região, grupo de trabalho e banco de dados. Após estas configurações é possível realizar *queries* dentro dos blocos dos *scripts* sem a necessidade de acessar a suíte da AWS na área logada via *web browser.*

Figura 4.4.1.4.9: Arquitetura de Coleta, Processamento dos Dados em Tempo Real e Ambiente de Desenvolvimento para Visualização dos Dados do Painel de Controle de Congestionamento.



Fonte: Elaboração Própria

***** A lista de bibliotecas está no ANEXO.

Para os dados em tempo real realizamos chamadas diretas na API do Waze sem passar pela nuvem da FGV. Desta forma, diminuímos a latência e os dados são processados e visualizados de acordo com a configuração do intervalo de requisições, atualmente em 5 minutos. Em seguida, transformaram-se os dados para serem consumidos pelos scripts de visualização. Para realizar chamadas direto na API é necessário ser cadastrado como Partner do Waze e obter a credencial `Clientid`. Além disso, é necessário configurar o polígono da cidade. No caso de Xalapa, Quito, Montevidéu, e distrito de Miraflores, basta um único polígono. Porém, para São Paulo é necessário separar em dois ou mais polígonos e posteriormente concatenar os resultados.

O arquivo de DE-PARA das vias monitoradas é essencial no painel de controle, já que converte os trechos com as denominações do Waze para os nomes dos logradouros que os gestores de tráfego das cidades estão mais acostumados (Tabela 4.4.1.4.3). Para exemplificar, a ‘Av. Galo Plaza Lasso’ pode ser identificada no Waze como: 'Av. Galo Plaza Lasso (Vía Lateral) > (N)', 'Av. Galo Plaza Lasso (Vía Central)', 'Av. Galo Plaza Lasso', 'Salida Hacia Av. Galo Plaza Lasso >

(N)', 'Salida Hacia Av. Galo Plaza Lasso (Vía Lateral) > (N)', e 'Salida Av. Galo Plaza Lasso > (S)'.

Tabela 4.4.1.4.3 Descrição dos Campos do Arquivo de DE-PARA de Vias

Nome	Tipo	Descrição
`new_street`	string	Nomenclatura utilizada pelas equipes de gestão de tráfego das cidades
`street`	string	Nomenclatura utilizada pelo Waze

Fonte: Elaboração Própria

Para a cidade de São Paulo, por exemplo, que já possui os logradouros das vias monitoradas de acordo com o mapa da prefeitura, o arquivo servirá de referência para conversão dos trechos do identificados no Waze para o arquivo original da CET. As demais cidades que não possuem um arquivo de vias monitoradas deverão primeiramente selecioná-las e em seguida identificá-las no mapa do Waze. Vale ressaltar que os trechos com vias nos dois sentidos poderão ter nomenclaturas distintas. A escolha das vias pode obedecer um critério de volume de tráfego, classificação viária, por região de interesse, ou mesmo uma seleção sem um critério pré-determinado.

Para o desenvolvimento deste produto, foram utilizadas listas de vias (Tabela 4.4.1.4.4) indicadas pelos representantes das cidades ou as vias mais importantes de cada localidade considerando o fluxo de veículos.

Tabela 4.4.1.4.4: Lista de Vias Utilizadas nos Protótipos

Cidade	Vias monitoradas
Miraflores, Peru	Alameda Ricardo Palma Bajada de Armendáriz Avenida José A. Larco Avenida Arequipa Calle Lima Alameda de la Avenida Pardo Avenida Óscar R. Benavides Bajada Balta Avenida de la aviacion Avenida Alfredo Benavides Avenida Santa Cruz Avenida Angamos Avenida Reducto

Cidade	Vias monitoradas
Quito, Equador	Av. Mariscal Sucre Av. Pedro Vicente Maldonado Av. Simón Bolívar Av. Galo Plaza Lasso Av. Tnte. Hugo Ortiz Av. 12 de Octubre Av. 10 de Agosto Av. América Av. Amazonas Av. De los Shyris Av. Eloy Alfaro Av. de la Prensa Av. 6 de Diciembre Av. Diego Vásquez de Cepeda E28B Av. Manuel Córdova Galarza Oswaldo Guayasamín Av. Ruta Viva Av. Alonso de Angulo Av. Morán Valverde Av. Ajaví Av. Cardenal de la Torre Av. Rodrigo de Chávez Av. Colón Av. Naciones Unidas Av. Gaspar de Villarroel Av. Río Coca Av. El Inca Av. De los Granados Av. Machala
Montevidéu, Uruguai	Avenida 8 de Octubre Avenida 18 de Julio Avenida 19 de Abril Avenida Bolivia Avenida Brasil Avenida Cataluña Avenida del Libertador Brigadier General Juan Antonio Lavalleja Avenida Doctor Luis Alberto de Herrera Avenida General Fructuoso Rivera Avenida Italia Avenida José Belloni Avenida Carlos Maria de Pena Avenida Agraciada Avenida Luis Batile Berres

	Avenida General Rivera Avenida de las Instrucciones

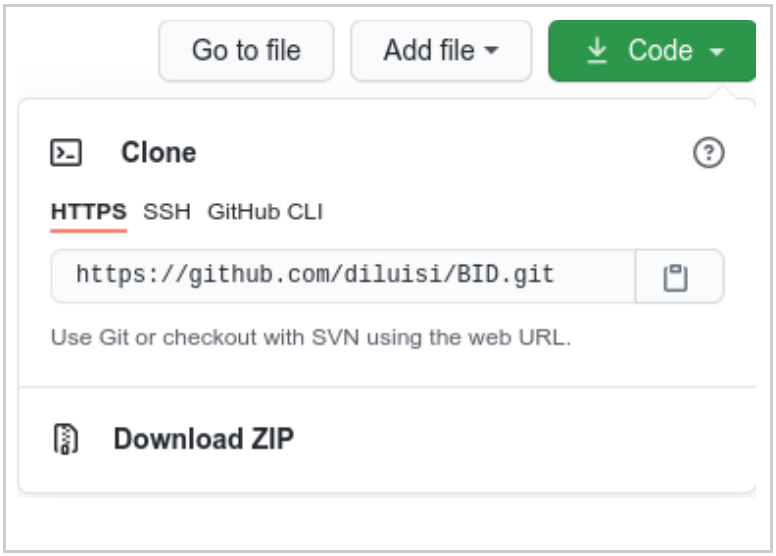
Cidade	Vias monitoradas
São Paulo, Brasil	A lista de vias de São Paulo corresponde a mais de 600 trechos monitorados atualmente pela CET. Para o protótipo utilizamos todos os trechos em arquivo fornecido pela CET.
Xalapa, México	Xalapa não possuía dados no Waze, por esta razão não incluímos no protótipo. No entanto, isto não inviabiliza caso a cidade queira utilizar o modelo.

Fonte: Elaboração Própria

4.4.1.5 Instalação

A instalação do protótipo de visualização de dados de congestionamento é realizada após o *download* do projeto no repositório https://github.com/diluisi/BID, conforme Figura 4.4.1.5.1, e a instalação das bibliotecas python para este projeto, conforme Apêndice A. O arquivo de captura dos dados da API do Waze processa separadamente os dados e salva em arquivo **.csv*. Este processo deverá ser substituído por carga no banco de dados na infra-estrutura de tecnologia de cada cidade.

Figura 4.4.1.5.1: Download dos Arquivos do Protótipo de Visualização de Dados.

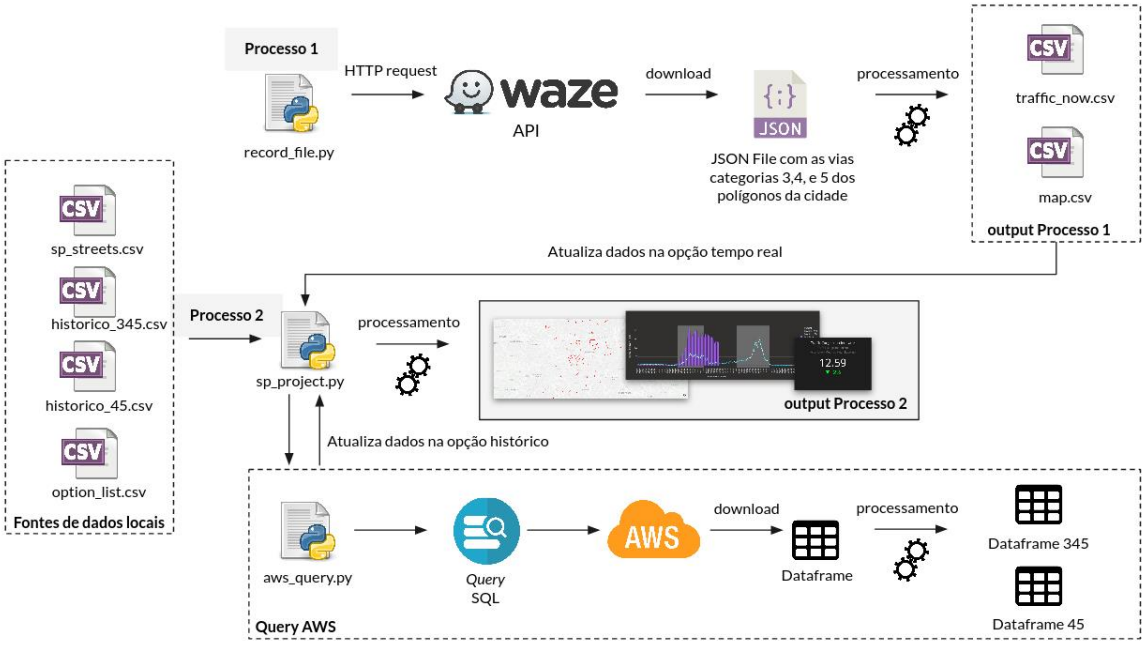


Fonte: Elaboração Própria

4.4.1.6 Mapa da arquitetura de dados

A Figura 4.4.1.6.1 esquematiza a visão geral da arquitetura de dados atual. Para o protótipo não se foca em otimizações de código ou de arquitetura, portanto cabe ressaltar que o resultado final pode haver algumas restrições de performance. Para a instalação em ambiente de produção será necessário ajustar o modelo para diminuir a latência nas buscas de dados da AWS, que atualmente apresenta o maior gargalo no tempo de execução. Caso se opte por fazer o pré-processamento dos dados diretamente da API do Waze, é possível obter uma redução na necessidade de espaço de armazenamento, bem como melhor desempenho na visualização de dados em tempo real.

Figura 4.4.1.6.1: Visão Geral da Arquitetura de Dados Atual



Fonte: Elaboração Própria

Na Tabela 4.4.1.6.1, descrevemos os arquivos e processos da arquitetura atual:

Tabela 4.4.1.6.1: Descrição da Arquitetura de Dados Atual

Nome	Descrição
`Processo 1`	Processo de gravação dos dados em tempo real da API do Waze
`Processo 2`	Processo core de geração da visualização de dados
`sp_streets.csv`	Arquivo de DE-PARA das vias monitoradas
`historico_345.csv`	Curva de referência para o Grupo 1. Utilizamos de um único dia, mas o ideal é que a curva seja o resultado de um modelo de análise histórica com as particularidades de tráfego de cada cidade.
`historico_45.csv`	Curva de referência para o Grupo 2. Utilizamos de um único dia, mas o ideal é que a curva seja o resultado de um modelo de análise histórica com as particularidades de tráfego de cada cidade.

`option_list`	Opções de seleção de vias (útil quando se analisa mais de uma cidade)
`record_file.py`	*Script* de processamento de dados históricos
`Waze API`	Interface de requisição HTTP dos dados do Waze
`JSON`	Formato de saída da requisição dos dados da API do Waze
`traffic_now.csv`	Armazenamento dos dados extraídos da API. Visualização dos dados históricos agrupados a cada 30 minutos
`map.csv`	Armazenamento temporário dos dados de latitude e longitude dos trechos congestionados do Grupo 1 e Grupo 2. Este arquivo é sobrescrito a cada atualização.
`aws_query.py`	Envio de solicitação da query para a AWS de acordo com o intervalo selecionado no painel. Retorna dois *dataframes* para cada grupo.
`sp_project.py`	Script de processamento e geração da visualização de dados

Fonte: Elaboração Própria

4.4.1.7 Configurando de uma nova cidade no painel

Para cada arquivo estão destacados os trechos de códigos ou campos que devem ser modificados para executar o projeto em uma nova cidade.

No arquivo `aws_query.py` é necessário modificar a *query* executada na AWS, conforme trecho de código abaixo:

aws_query.py

```
sql = ('SELECT "pub_utc_date", "polygon_slug", "uuid", "street", "length",'
                  ' "level", "delay", "type", "line_geojson" '
                              ' FROM "cities"."br_saopaulo_waze_jams" '
                              ' WHERE year=2019 AND month = '+str(1)+'
AND     day='+str(1)+ ' AND city=\'São Paulo\'')
```

O arquivo `sp_project.py` contém o código de visualização e neste *script* é necessário modificar o *drop-down* com as opções de cidade.

sp_project.py

```
dcc.Dropdown(
                                                id="region-dropdown",
                                                options=[
                                                      {"label": "Sao Paulo -
Brazil", "value": "Sao Paulo - Brazil"},
                                                      ],
                                                #placeholder="Select a category",
                                                value='Sao Paulo - Brazil',
                                                style={"width": "45rem"},
                                                clearable=False,
                                                )
```

O mapa possui coordenadas para centralizá-lo. A alteração seguinte é das coordenadas da cidade que permite visualizar os seus limites geográficos. O valor da chave `mpbx_center` no dicionário `coords` é passado para a função de atualização do mapa `px.scatter_mapbox`, a alteração é na posição da cidade na lista da chave `region` do dicionário `coords`. No exemplo abaixo cada cidade está em uma posição, São Paulo é “0”, Quito “1”, Miraflores “2”, e Montevideo “3”.

sp_project.py

```
coords = pd.DataFrame({"region": ["Sao Paulo - Brazil", "Quito - Ecuador",
"Miraflores - Peru", "Montevideo - Uruguay"],
                        "city_la":[-23.506226,-0.111780,-12.029391,-
34.782417],
                        "city_lo":[-46.6336332,-78.450622,-77.028059,-
56.222555],
                        "mpbx_center":[-23.550164466,-0.202431,-12.124605,-
34.905283]})
.
.
.



figb                                                                    =
px.scatter_mapbox(lat=[coords[coords['region']==region].city_la.iloc[0]],
lon=[coords[coords['region']==region].city_lo.iloc[0]], zoom=3,height=500)
                    figb.update_layout(mapbox_style="carto-positron",
mapbox_zoom=12,                         mapbox_center_lat                       =
coords[coords['region']==region].mpbx_center.iloc[0],
                                        margin={"r":0,"t":0,"l":0,"b":0})
```

No arquivo `record_file.py` o *script* é executado continuamente para coleta dos dados a cada cinco minutos da API do Waze. Neste caso é preciso modificar a lista do id dos polígonos, o arquivo de DE-PARA e os pontos de latitude e longitude da poligonal.

record_file.py

```
cities = ['Sao Paulo 1','Sao Paulo 2', 'Sao Paulo 3']


de_para_sp = pd.read_csv('/sp_streets.txt')


WAZE_POLYGONS = {
      'Sao Paulo 1': {
      'city': 'São Paulo',
      'points': '-46.89388,-23.65469;-46.8972,-23.79037;-46.32591,-23.791;-
46.32865,-23.64431;-46.89388,-23.65469',
      },
      'Sao Paulo 2': {
      'city': 'São Paulo',
      'points': '-46.9344,-23.4539;-46.91986,-23.55706;-46.27304,-23.52685;-
46.30119,-23.35991;-46.9344,-23.4539',
      },
      'Sao Paulo 3': {
      'city': 'São Paulo',
      'points': '-46.2656,-23.52821;-46.92789,-23.5558;-46.90063,-23.65647;-
46.31814,-23.64391;-46.2656,-23.52821',
      }
}
```

Abaixo estão os polígonos que estão definidos na base de dados AWS.

```
# polígonos de cada cidade
WAZE_POLYGONS = {
      'Xalapa': {
      'city': 'Xalapa',
      'points': '-97.16627,19.797331;-96.57464,19.794427;-
96.57660,19.272967;-97.17500,19.276073;-97.16627,19.797331',
```

```
    },
    'Quito': {
    'city': 'Quito',
    'points': '-78.486,0.133;-78.595,-0.176;-78.648,-0.36;-78.445,-
0.404;-78.25,0.114;-78.486,0.133',
    },
    'Montevideo': {
    'city': 'Montevideo',
    'points': '-56.324,-34.601;-56.45,-34.881;-55.934,-35.003;-
55.39,-34.854;-56.324,-34.601',
    },
    'Sao Paulo 1': {
    'city': 'São Paulo',
    'points': '-46.89388,-23.65469;-46.8972,-23.79037;-46.32591,-
23.791;-46.32865,-23.64431;-46.89388,-23.65469',
    },
    'Sao Paulo 2': {
    'city': 'São Paulo',
    'points': '-46.9344,-23.4539;-46.91986,-23.55706;-46.27304,-
23.52685;-46.30119,-23.35991;-46.9344,-23.4539',
    },
    'Sao Paulo 3': {
    'city': 'São Paulo',
    'points': '-46.2656,-23.52821;-46.92789,-23.5558;-46.90063,-
23.65647;-46.31814,-23.64391;-46.2656,-23.52821',
    },
    'Lima':{
    'city':'Miraflores',
    'points':'-77.086,-11.894;-77.187,-12.048;-77.053,-12.211;-
76.94,-12.258;-76.824,-12.197;-76.846,-11.925;-77.086,-11.894'
    }
}
```

Polígonos de cada cidade

O arquivo `record_file.py` foi desenvolvido para atender a cidade de São Paulo que possui três poligonais que precisam ser concatenadas. Para cidades que possuem apenas uma poligonal o código pode ser simplificado.

Além das modificações mencionadas anteriormente, ainda é necessário ajustar os arquivos de dados locais para que correspondam aos da cidade a ser incluída.

4.4.1.8 Considerações finais

Neste projeto foi avaliado o uso dos dados extraídos do Waze como referência na gestão de congestionamento. Foram avaliados e comparados os dados na cidade de São Paulo, por já possuir um método de inspeção do trânsito e uma base de dados histórica.

O resultado da comparação dos dados do Waze com os da CET foi satisfatório e mostrou que a API conseguiu identificar as flutuações de congestionamento. Não foi identificada uma diferença significativa entre os Grupos 1 e 2, ou seja, ambos tiveram correlações similares quando comparados com os dados da CET.

O passo decorrente da avaliação foi a criação de um protótipo para gestão de tráfego utilizando as premissas:

- Usabilidade e foco na experiência do usuário;
- Código aberto;
- Interface que proporcione o diagnóstico do tráfego utilizando os dados históricos e tempo real;
- Portável;
- Extensível;
- Adaptável;
- Simples execução e manutenção;
- Escalabilidade da monitoração;
- Baixo custo.

Procurou-se simplificar o processo de gestão a partir da extração direta da API do Waze e a visualização dos trechos monitorados no mapa e no histograma. Além disso, foi computado um indicador de congestionamento que é o somatório dos trechos congestionados a cada intervalo e o delta que indica a variação em relação a uma curva de referência.

O painel, por ser desenvolvido em código aberto, pode ser incrementado conforme necessidade de cada cidade, podendo agregar modelos mais robustos e gráficos adicionais.

4.4.2 Apoio à Gestão do Trânsito – Identificação de Acidentes

Entre uma das externalidades negativas provocadas pelo aumento do transporte motorizado está o aumento do número de acidentes de trânsito. Acidentes de trânsito ocorrem de forma inesperada, impactando os envolvidos no acidente e muitas vezes causando distúrbios no tráfego de uma região. Esses acidentes podem ser classificados de diversas formas, indo desde colisões leves até capotamentos. Na cidade de São Paulo, o curso de ação recomendado durante a ocorrência de um acidente está associado à presença de vítimas: caso existam, independente da gravidade, o Serviço de Atendimento Móvel de Urgência (SAMU) deve ser acionado o mais rápido possível, juntamente com chamada à Polícia. Ambos os serviços devem ser acionados através de contato por telefone, o que pode levar algum tempo após a ocorrência do acidente.

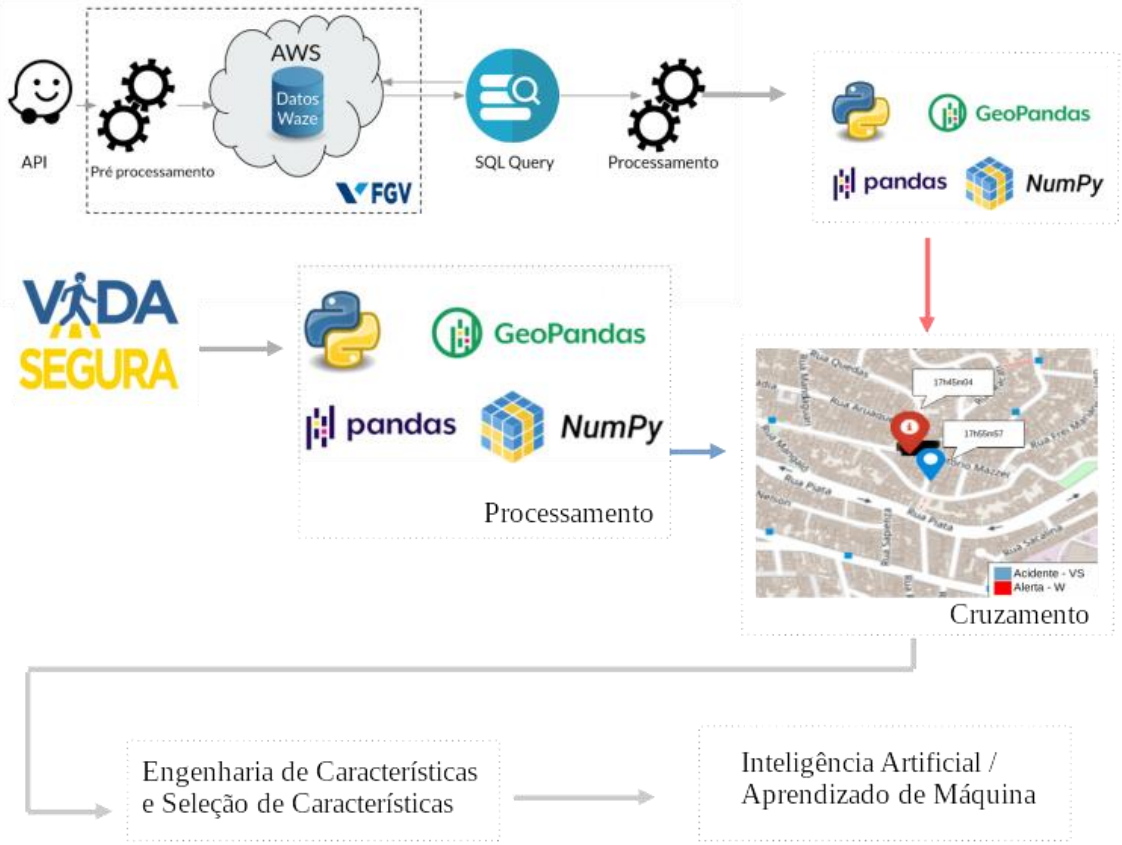
Em um contexto de cidades inteligentes, é desejável que acidentes possam ser detectados de forma rápida e automatizada, especialmente nos casos com vítimas. Todavia, isso iria requerer uma infraestrutura de sensores e IoTs (Internet das Coisas) bem desenvolvida em toda a cidade, o que ainda não temos disponível. Uma possível alternativa é o uso de dados de apps para smartphones, como o Waze.

Assim, nosso objetivo nesse produto é a detecção de acidentes com vítimas usando algoritmos de Inteligência Artificial, os dados de alerta e congestionamentos do Waze e a base de dados Vida Segura, que reporta todos os acidentes com vítimas ocorridos na cidade de São Paulo. No restante do texto apresentamos a base de dados Vida Segura em detalhes e como cruzamos as três bases de dados para gerar uma base única e coerente para uso no aprendizado de máquina. Em seguida, apresentamos o algoritmo e os resultados do treino e teste usando a base de dados gerada. Por fim, apresentamos uma prova de conceito na última seção.

Observa-se na Figura 4.4.2.1, a sumarização do fluxo proposto para detecção de acidentes. Nessa figura temos todos os passos metodológicos executados para o produto 4.2, relacionado a detecção dos acidentes, por meio do uso de dados de API. Em um primeiro estágio temos a coleta dos dados pela API do Waze, seguido por uma etapa de consolidação em um banco de dados. Posteriormente, esses dados consolidados são processados com as bibliotecas de código aberto, e cruzados com a base de dados oficiais de acidentes de referência da cidade de São Paulo. Esse novo conjunto de dados é alimentado em um algoritmo de aprendizado de máquina, que torna possível a generalização da detecção de acidentes.

Figura 4.4.2.1: Fluxograma desse subproduto relacionado a Detecção de Acidentes.

Fonte: Elaboração Própria



4.4.2.1 Base de dados

Para o desenvolvimento deste produto foram utilizados os dados da API do Waze e de acidentes fornecidos pela plataforma VIDA SEGURA - VS da Prefeitura de São Paulo. Empregamos as ocorrências presentes na plataforma VIDA SEGURA - VS, como um referencial dos tipos de acidentes que queremos detectar nos dados provenientes da *Application Programming Interface* – API do Waze. Esse cruzamento está descrito na subseção seguinte.

Plataforma Vida Segura

Instituído pelo Decreto 58.717, de 2019, o Plano de Segurança Viária estabelece um projeto para a cidade de São Paulo para redução de ocorrências graves e mortes no trânsito. Este Plano se norteia em tornar o tráfego mais seguro na cidade, estabelecendo e fortalecendo políticas públicas atualmente em vigência. Chamado de Vida Segura, a estratégia reúne ações de curto,

médio e longo prazo com o objetivo de reduzir à metade as mortes ocorridas no trânsito da cidade até 2028.

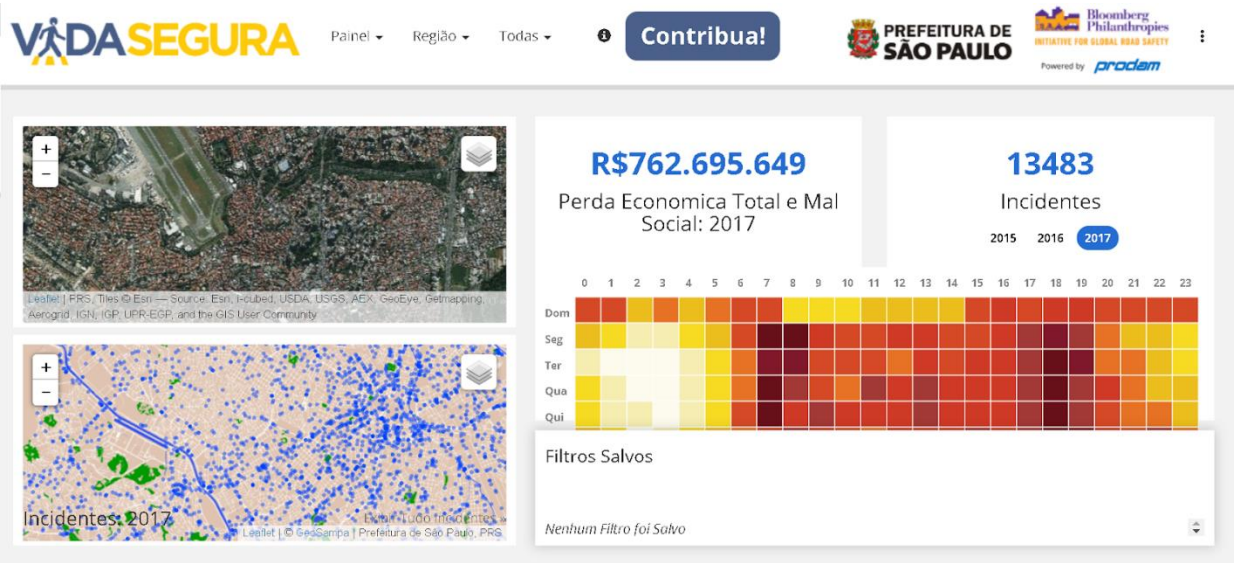
Tomando como referências a metodologia de Visão Zero, o entendimento do programa é que nenhuma morte é aceitável, e compete aos projetistas do sistema uma responsabilidade proativa e integrada para evitar as mortes na via (Belin et al., 2012, WELLE B., et al. 2015, SWOV, 2013). Para tanto, se faz necessário uma integração das múltiplas camadas de projetistas de modo a garantir e entender os detalhes e padrões dos acidentes da cidade.

Uma das inovações previstas por São Paulo no plano é promover a abertura dos dados de equipamentos de fiscalização. A cidade se compromete a encontrar uma maneira de disponibilizar esse conhecimento em uma plataforma digital e aberta (em que já estão disponibilizados os dados de acidentes), para poderem ser usados pela sociedade civil e o setor privado. Informações mais qualificadas ajudam a cidade a tomar decisões melhores na hora de desenvolver novas soluções para aumentar a segurança do sistema de mobilidade.

Para esse fim, foi desenvolvida a plataforma VIDA SEGURA - VS, que unifica metodologias entre o Estado e o Município de São Paulo para sistematização de acidentes e mortes no trânsito. Nessa plataforma de dados abertos, cada subprefeitura consolida os acidentes, veículos e vítimas que necessitaram de ações da Companhia de Engenharia de Tráfego – CET da Cidade de São Paulo. Observamos na Figura 4.4.2.1.1, a apresentação da plataforma, com as consolidações disponíveis:

Figura 4.4.2.1: Visualização da Plataforma Vida Segura

Fonte: **WRI**



A origem dos dados para tanto é oriundo de diferentes plataformas institucionais, utilizadas por várias camadas da sociedade, como o Sistema de Registro Digital de Ocorrências — RDO da Polícia Civil — INFOCRIM, os registros do Instituto Médico Legal —– IML, Dados do Sistema Único de Saúde — DATASUS, Serviço de Atendimento Móvel de Urgência — SAMU, entre outros.

Acerca das informações contidas na plataforma, há uma série histórica de dados de 2014 até 2020, sendo que apenas os de 2018 em diante são consolidados. Essa plataforma disponibiliza de forma estática, após processo de consolidação pela CET, as ocorrências do ano anterior. Os atributos presentes na base podem ser categorizados em informações sobre o quando e onde ocorreu, como: geolocalização, subprefeitura responsável, distrito da cidade e data da ocorrência; e informações sobre a ocorrência, como: número de veículos envolvidos, número de mortos, número de vítimas e número de feridos.

Waze

O Waze é uma empresa de tecnologia que fornece o serviço de navegação colaborativa. Além de expor informações em tempo real das condições de trânsito nas cidades, também fornece informações de *status* dos trechos de via com a classificação de congestionamento, incidentes, acidentes graves e sugestões de rotas.

Por ser uma ferramenta colaborativa, onde os usuários compartilham seus dados de geolocalização e notificam os alertas de incidentes, a qualidade dos dados está diretamente relacionada ao volume de usuários cadastrados na aplicação. Em 2019, São Paulo possuía aproximadamente 4,5 milhões de usuários ativos. A impressionante cifra foi um dos motivos da escolha da cidade na comparação e validação dos dados fornecidos pela API do Waze.

As informações coletadas são adicionadas nas camadas tendo como base o mapa viário da cidade. Cabe ressaltar que o monitoramento do Waze é feito em todas as vias da cidade. O mapa é representado por um objeto matemático chamado de grafo. O grafo é composto de vértices e arestas que se conectam, simulando o viário da cidade.

Para construção da aplicação de detecção de acidentes, utilizamos os dados disponibilizados pela Waze *Traffic Jams* (WJ) e *Alerts* (WA). Esses dados consistem de informações de desaceleração de tráfego com base na localização e velocidade dos usuários, e os alertas disponibilizados pelos usuários, com informações diversas sobre o estado da pista (se ocorreu um acidente ou não). Os campos do arquivo são populados e expostos pela API através de uma chamada utilizando o protocolo HTTP. Os campos utilizados na elaboração da prova de conceito estão identificados na Tabela 4.4.2.1.1 e 4.4.2.1.2.

Tabela 4.4.2.1.1: Campos utilizados da base de dados Waze *Traffic Jams* (WJ).

Nome	Tipo	Descrição
`pubMillis`	Timestamp	Unix time
`line`	Latitude e Longitude	Linestring
`length`	Integer	Comprimento em metros
`street`	string	Nome da via
`city`	string	Nome da cidade
`country`	string	Nome do país
`level`	Integer	Categoria de congestionamento (0= fluxo livre - 5= Bloqueado)

Tabela 4.4.2.1.2: Campos utilizados da base de dados Waze *Alerts* (WA)

Nome	Tipo	Descrição
`pubMillis`	Timestamp	Unix time
`location`	Latitude e Longitude	Latitude e longitude do alerta.
`type`	string	Tipo de alerta, na base.
`subtype`	string	Sub tipo de alerta.
`street`	string	Nome da via
`city`	string	Nome da cidade
`country`	string	Nome do país

A extração dos dados foi realizada na tabela de dados disponibilizada pela Fundação Getúlio Vargas (FGV) que os armazena em uma nuvem dedicada para o projeto *Big Data* para o Desenvolvimento Urbano Sustentável. Os dados representam uma fotografia do viário minuto a minuto, sendo os dados crus processados antes do armazenamento.

4.4.2.2 Cruzamento dos dados

Para o uso de algoritmos de Aprendizado de Máquina e Inteligência Artificial, necessita-se de uma base de dados única, com informações coerentes e que indiquem se os valores dos atributos da base estão associados a um acidente leve (sem vítimas) ou grave (com vítimas). Assim, o primeiro passo para atingir o objetivo proposto foi cruzar as três bases de dados disponíveis:

- Waze_Alerts (WA), a base de dados com alertas enviados por usuários do Waze;
- Waze_Jams (WJ), a base contendo congestionamento detectados pelo sistema interno do Waze.
- Vida Segura (VS), contendo a relação de acidentes com vítimas descritas anteriormente.

As três bases são independentes e não possuem um campo único capaz de unir as bases de forma inequívoca. Logo, foi necessário avaliar os campos presentes em cada base e formas de as associar. Como as bases WA e WJ possuem a mesma fonte (o sistema Waze), o cruzamento entre essas bases foi mais direto, gerando a base de dados do Waze (W). Em um segundo

momento, foram cruzadas a base W com a VS. Como a fonte de dados difere nesse caso, o cruzamento é mais complexo e propenso a associações erradas. As próximas subseções detalham os dois cruzamentos das bases de dados.

Cruzamento de bases Waze Alerts (WA) e Waze Jams (WJ)

A base WA possui diversos tipos de alertas, sendo eles: Acidentes, Congestionamentos, ClimaPerigo e EstradaFechada. Estes são então subdivididos em subtipos. O primeiro passo no cruzamento das bases foi escolher quais alertas manter e quais excluir.

Duas abordagens foram utilizadas: uma apenas com alertas do tipo Acidentes e outra com alertas to tipo Acidentes e os subtipos PerigoCarroNaPista, PerigoCarroNoAcostamento e PerigoMorteNaPista, do tipo ClimaPerigo. Esses dois subtipos foram mantidos assumindo-se que o usuário poderia se confundir ao sinalizar o alerta. Os demais tipos e subtipos de alertas são específicos o suficiente para não gerar confusão e foram excluídos (por exemplo, BuracoNaPista).

Observando os campos de ambas as bases, nota-se que era possível fazer uma associação direta com os campos DataHora e Endereço. Contudo, em ruas e avenidas muito longas podem existir diferentes pontos de congestionamento, de forma que um alerta poderia ser associado a mais de um congestionamento.

Para evitar essa situação, utilizam-se as coordenadas latitude e longitude da base WA e o atributo Linha da base WJ, possibilitando o cálculo da distância entre o alerta e o congestionamento. Estabelecemos uma distância máxima de 200 metros entre as coordenadas, descartando associações com distâncias maiores. No caso de associações com diversas distâncias menores que 200 metros, escolhemos a associação com menor distância.

Nos casos em que um alerta de WA não encontrou nenhuma associação em WJ (algo esperado), assumiu-se que o acidente ocorreu em um local sem congestionamento e atribuiu-se um valor padrão zero para os campos vindos da base WJ.

Cruzamento de bases WxVS

Inicialmente, houve uma tentativa de fazer o mesmo cruzamento entre as bases W e VS que foi feito entre as bases WA e WJ, mas logo notou-se que não seria possível. Isso ocorreu por dois motivos:

- O atributo DataHora presente na base VS era um valor único, não deixando claro se essa hora se referia ao horário do acidente, da chegada das autoridades ao local ou à finalização da ocorrência;
- O atributo Endereço não era compatível entre as bases, com formatos, abreviações e informação de cruzamento entre ruas na base VS diferentes daqueles encontrados em W.

Para cruzar essas bases, foram usados critérios de intervalo de tempo entre a ocorrência VS e o alerta em W e distância usando coordenadas latitude e longitude de ambas as bases, da seguinte forma:

1. Foram selecionados todos os alertas em W em um intervalo de ± 30, 60, 120 e 180 minutos em relação ao horário de uma ocorrência em VS;
2. Foram calculadas as distâncias entre a coordenada da ocorrência em VS e a coordenada do alerta em W, colocando como distância limite os valores de 200 e 500 metros.

Caso existam alertas de W que atendam o intervalo de tempo e o limite de distância considerados, então esses alertas foram associados à ocorrência do VS e recebem o valor de classe 1, indicando um acidente grave, i.e., um acidente com vítimas. Caso contrário, os alertas recebem a classe 0, indicando um acidente leve/sem vítimas.. No exemplo da Figura 4.2.2.2.1 temos que o Acidente – VS está dentro do raio temporal e no raio espacial para permitir o cruzamento.

Figura 4.4.2.2.1: Cruzamento da Base do Waze – W com Vida Segura – VS.

Fonte: Elaboração Própria

Estatísticas dos Cruzamentos e Critérios Escolhidos

Definidos os critérios de cruzamento, foram avaliados como cada um deles impacta no conjunto de dados final. Foram usados dados de 365 dias do ano 2019 nas três bases utilizadas. Os critérios utilizados podem ser resumidos em:

- A200 e A500: Apenas alertas do tipo Acidente, com distâncias de até 200 e 500 metros entre o alerta e o acidente registrado na base Vida Segura;
- AH200 e AH500: Inclui alertas do tipo Acidente e os subtipos PerigoCarroNaPista, PerigoCarroNoAcostamento e PerigoMorteNaPista, do alerta ClimaPerigo, com as mesmas distâncias acima.

Conforme esperado, verificou-se que uma distância menor no critério reduz a taxa de associação entre as bases (Tabela 4.4.2.2.1). Já adicionar outros tipos de alertas além de acidentes aumenta essa taxa, em troca de ter um cruzamento final menos preciso, uma vez que seriam associados acidentes com vítimas (do Vida Segura) a outros tipos de alerta.

Tabela 4.4.2.2.1 Porcentagem de dados da base Vida Segura associadas aos alertas do Waze, por critério

	A200	A500	AH200	AH500
Não encontrado	75%	64%	69%	52%
Encontrado	25%	36%	31%	48%

Fonte: Elaboração Própria

Utilizando os dados VS corretamente associados, avaliou-se qual a porcentagem de alertas do Waze fica associada a acidentes com vítimas (quando houve associação entre as bases) ou sem vítima (Tabela 4.4.2.2.1). Pode-se observar que o critério AH500, apesar de conseguir uma maior taxa de associação com a base VS (Tabela 4.4.2.2.2), essa associação corresponde a apenas 3% de todos os alertas que esse critério permite. Isto ocorre porque a grande maioria dos alertas são de acidentes leves, que não aparecem na base VS. Mas, de forma geral, todos os critérios geram conjuntos de dados desbalanceados, o que impacta na qualidade do aprendizado de máquina (Seção Treino).

Tabela 4.4.2.2.2: Porcentagem de dados do Waze associados ao Vida Segura, por critério

	A200	A500	AH200	AH500
Sem vítimas	96%	93%	98%	97%
Com vítimas	4%	7%	2%	3%

Fonte: Elaboração Própria

Dados esses valores, selecionou-se o critério A500 para gerar o conjunto de dados final por apresentar uma taxa de associação razoável entre as bases e com o menor desbalanceamento.

Enriquecimento, Seleção de Atributos e Conjunto de Dados Final

Nem todos os atributos presentes do dataset WVS obtido após o cruzamento dos dados de W e VS podem ser utilizados durante a fase de treinamento. Por outro lado, novos atributos podem ser definidos a partir dos atributos existentes, enriquecendo a base de dados. Nesta seção são

descritos quais novos atributos puderam ser criados e qual a lógica por trás deles e quais atributos foram selecionados/excluídos e a motivação dessas escolhas.

Enriquecimento

Para enriquecer a base de dados, foram selecionados os seguintes atributos:

1. Length: o comprimento do congestionamento no instante do primeiro alerta de um novo acidente;
2. Level: O nível do congestionamento, variando entre 0 e 5;
3. Delay: O tempo de atraso no deslocamento pelo setor congestionado

Na ausência de congestionamentos, os atributos recebem o valor zero.

Os novos atributos representam a diferença entre os atributos descritos acima no momento do primeiro alerta de acidente (instante 0) e em instantes no passado próximo, especificamente 1, 5, 10, 20 e 30 minutos antes do primeiro alerta de um novo acidente. Esses atributos podem auxiliar no aprendizado do algoritmo em casos em que um acidente causou um congestionamento ou um aumento dele.

Seleção de Atributos

Com todos os atributos vindos da base de alertas, congestionamentos e atributos do enriquecimento, foi preciso escolher quais atributos utilizar no treinamento do algoritmo de aprendizado de máquina. Os atributos podem ser classificados nos seguintes tipos:

- Atributos tipos de alertas: type, subtype, roadtype
- Atributos de qualidade do alerta: nthumbsup, reliability, reportrating, confidence
- Atributos temporais: hour, day, pubmillis, pub_utc_date
- Atributos espaciais: longitude, latitude, street, city, line
- Atributos de congestionamento: Length, level, delay, enriquecidos
- Atributo classe (não nativo do waze, baseado no Vida Segura)

4.4.2.3 Treinamento e Estatísticas da Detecção de Acidentes Graves

Antes de iniciar o treinamento do algoritmo de um algoritmo de aprendizado de máquina, dois passos importantes são necessários:

- Pré-processamento e separação dos dados
- Escolha do algoritmo

Pré-processamento e separação dos dados

Como o conjunto de dados final WVS é desbalanceado, isto é, com muito mais acidentes sem vítimas, empregamos as seguintes técnicas de balanceamento:

- Random Undersampling: remoção aleatória de exemplos da classe majoritária (i.e., acidentes sem vítimas);
- Random Oversampling: repetição aleatória de exemplos da classe minoritária;
- Overweighting na classe minoritária: aplicação de um peso w na classe minoritária.

Esse balanceamento é necessário, pois, caso contrário, o algoritmo de aprendizado ignora a classe minoritária, classificando todos os exemplos como a classe majoritária.

Separou-se o conjunto de dados em conjunto de treino e teste da seguinte forma:

- Treino: primeiros 20 dias de todos os meses de 2019, totalizando 240 dias para treino (~66,6%);
- Validação: os últimos dez dias de cada mês, totalizando 125 dias para teste (~33,3%).

Escolha do algoritmo

Um passo importante no uso de aprendizado de máquina é a escolha do algoritmo adequado. Existem centenas de algoritmos diferentes disponíveis e cada um apresenta um desempenho melhor em contextos específicos. Para este produto, foi utilizado o algoritmo Catboost, baseado em experiências anteriores com conjuntos de dados similares. Esse algoritmo apresenta bons resultados em conjuntos de dados heterogêneos e não necessita de ajuste de parâmetros específicos.

Resultados do treinamento e validação

Para entender os resultados é necessário entender 3 conceitos:

- Precisão: indica a porcentagem de instâncias classificadas corretamente entre aquelas que o algoritmo indicou serem de uma classe;
- Revocação: indica a porcentagem de instâncias de uma determinada classe que o algoritmo conseguiu identificar;
- F1-Score: leva em consideração a precisão e a revocação para gerar uma medida única.

Geralmente, é desejável que as três medidas se aproximem o máximo possível do valor 1 para todas as classes do conjunto de dados. Neste caso particular, o interesse particular é de que o algoritmo acerte a classe “Com vítimas”, pois seria nesse caso em que as autoridades competentes seriam acionadas para atender o acidente reportado. Entretanto, como a quantidade de acidentes com vítimas (dadas pela base Vida Segura) é muito menor que os acidentes reportados pelo Waze (i.e., muitos acidentes ocorrem, mas sem vítimas), aprender os casos quando ocorrem esses acidentes específicos se torna uma tarefa difícil.

Tabela 4.4.2.3.1: Estatísticas do treino e validação do aprendizado considerando as técnicas de desbalanceamento

Classe	**Treinamento**				**Validação**			
	Precisão	Revocação	f1-score	support	Precisão	Revocação	f1-score	support
	Desbalanceado				Desbalanceado			
Sem vítimas	0.95	1.00	0.97	841080	0.95	1.00	0.97	467238
Com vítimas	0.00	0.00	0.00	47785	0.00	0.00	0.00	24259
	Undersampling				Undersampling			
Sem vítimas	0.64	0.71	0.67	47785	0.97	0.71	0.82	467238
Com vítimas	0.68	0.60	0.63	47785	0.09	**0.56**	**0.16**	24259
	Oversampling				Oversampling			
Sem vítimas	0.77	0.79	0.78	841080	0.96	0.78	0.86	467238
Com vítimas	**0.79**	**0.76**	**0.77**	841080	**0.10**	0.45	0.16	24259
	Overweighting				Overweighting			
Sem vítimas	0.97	0.70	0.81	841080	0.97	0.70	0.81	467238
Com vítimas	0.10	0.58	0.17	47785	0.09	0.56	0.15	24259

Fonte: Elaboração Própria

Pode-se observar na Tabela 4.4.2.3.1 que as técnicas de balanceamento tiveram um impacto positivo no resultado dos treinos comparados ao treinamento desbalanceado, com as 3 técnicas tendo resultados similares no conjunto de validação. A técnica Undersampling teve resultados levemente superiores na validação. É importante notar que o valor da precisão ficou baixo em todos os casos porque o número de acidentes sem vítimas é muito maior que o com vítimas. Isto faz com que a maior parte dos acidentes classificados como com vítimas sejam falsos positivos.

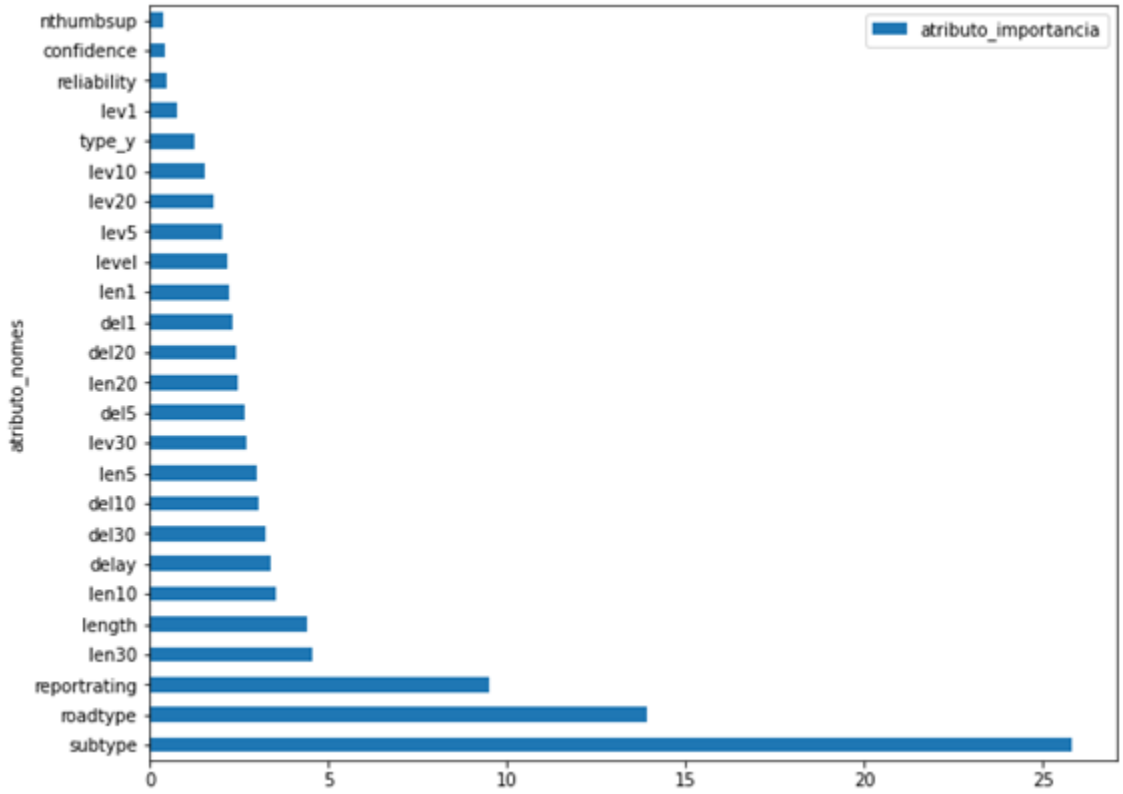
4.4.2.4 Discussão

Na seção anterior observou-se que o algoritmo de aprendizado de máquina não conseguiu obter taxas de precisão e revocação boas o suficiente para a aplicação desejada inicialmente. Nessa seção serão avaliados os possíveis motivos pelos quais o algoritmo não obteve a eficiência desejada e uma alternativa de uso desse modelo, mesmo com suas limitações.

Importância de Atributos e valores

O algoritmo Catboost possui uma função que nos permite analisar a importância de cada atributo no modelo gerado. Assim, nota-se que os atributos subtype, roadtype e reportrating, provenientes da base de Alertas do Waze, dominam a importância na classificação (Figura 3.2.4.1). Por outro lado, os demais atributos dessa mesma base (reliability, confidence e nthumbsup) praticamente não influenciam o modelo. Os atributos da base de congestionamento e os atributos enriquecidos possuem influência intermediária, porém, bem menor que os três mais influentes.

Figure 4.4.2.4.1: Importância de cada atributo na classificação

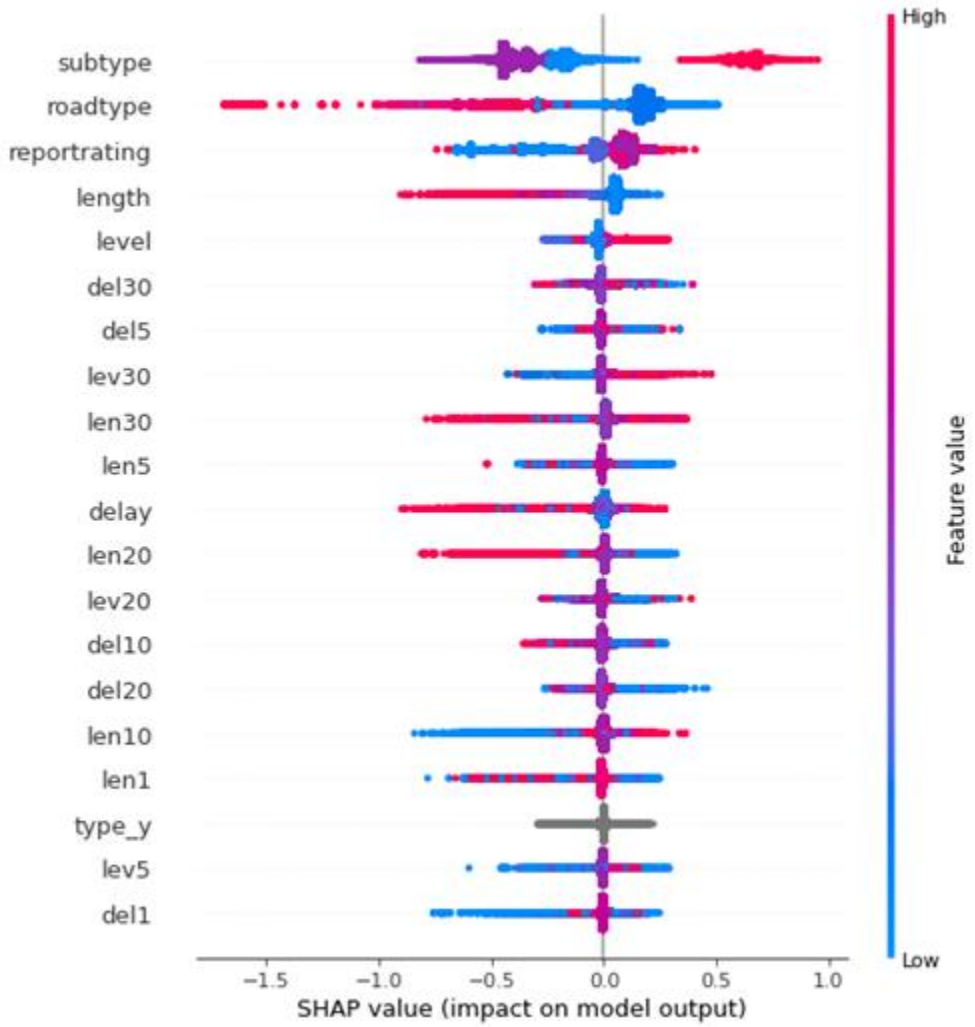


Fonte: Elaboração Própria

Uma outra forma de avaliar a influência de atributos é através dos Valores SHAP (SHapley Additive exPlanations). De forma geral, esses valores indicam não só a influência de cada atributo, mas também a influência dos valores desses atributos na classificação. Observando novamente os atributos mais influentes na Figura 4.4.2.4.1, nota-se que:

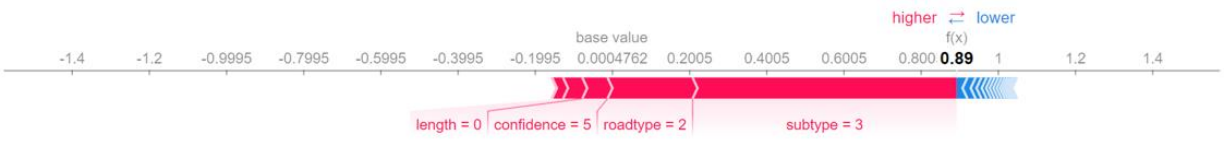
- A categoria “accident_major” do atributo “subtype”, em vermelho no gráfico, contribui para a classe “com vítimas”, a passo que a categoria “accident_minor”, em roxo, contribui para a classe “sem vítimas”. A categoria “not_informed”, em azul, fica em intermediário, tendendo à classe “sem vítimas”;
- As categorias “street” e “primary street” do atributo “roadtype”, em azul, contribuem para a classe “com vítimas” enquanto outros valores contribuem para a classe “sem vítimas”.

Figura 4.4.2.4.2: Valores Shap indicando a influência dos valores de cada atributo na classificação. Valores SHAP maiores que zero indicam contribuição para a classe "com vítimas"



Fonte: Elaboração Própria

Figura **4.4.2.4.2**. Acidente "sem vítima" classificado como "com vítima" incorretamente.



Fonte: Elaboração Própria

Figura 4.4.2.4.3: Acidente “com vítima” classificado corretamente

Fonte: Elaboração Própria

Na ausência de valores significativos nos outros atributos, a classificação é praticamente definida pelos três atributos mais influentes. As Figuras 4.4.2.4.2 e 4.4.2.4.3 mostram dois exemplos de acidentes classificados incorretamente (sem vítima) e corretamente (com vítima). Pode-se observar que o modelo classificou incorretamente o primeiro acidente por causa dos valores dos atributos dominantes e ausência de valores nos demais atributos (acidente sem congestionamento).

Como a ausência de informações de congestionamento deixa o modelo decidir baseado totalmente nos atributos de alerta e como os atributos de alerta sozinhos não são capazes de distinguir adequadamente entre acidentes com ou sem vítimas, percebe-se que os atributos do Waze sozinhos não são suficientes para realizar essa classificação.

Entretanto, apesar da dificuldade nas classificações, uma possibilidade de uso desses dados está relacionada à probabilidade de um acidente ser com ou sem vítima. Se for possível mostrar que uma probabilidade alta gerada pelo modelo está associada à classe correta e que o modelo erra apenas na fronteira entre as classes, então essa probabilidade poderia ser apresentada às autoridades competentes, que poderiam ficar atentas à chegada de alertas de acidentes com altas probabilidades de terem vítimas.

4.4.2.5 Visualização

Para apresentação dos resultados obtidos neste subproduto, uma prova de conceito (*proof of concept* - POC) de um sistema de visualização foi desenvolvido. O sistema foi desenvolvido em linguagem livre, com as bibliotecas de código aberto e pode ser acessado pelo link.

Para implementação da aplicação é necessário a instalação de um conjunto de bibliotecas na linguagem de programação Python. O sistema com a prova de conceito pode ser implementado em qualquer sistema operacional com suporte a linguagem (por exemplo: Windows, Linux, Mac), desde que seja seguido a sequência de passos disponível no repositório para instalação das bibliotecas necessárias. O ambiente necessita de conexão com a *internet*. Não há custos para uso das ferramentas, haja visto o contexto de linguagem em código aberto. Os custos estimados para implantação dessa ferramenta estão contidos na infraestrutura escolhida pela secretária.

Para implantação, recomenda-se um(a) programador(a) com experiência em linguagem Python. Maiores detalhes sobre instalação, dependência e fluxo de arquitetura estão contidos no repositório aberto.

O painel apresenta tanto informações históricas dos acidentes, de 2018 até junho de 2021, quanto informações em tempo real. O Painel possui integração para as cidades de São Paulo, Xalapa, Quito, Montevidéu e Lima, no entanto, o modelo de aprendizado de máquina é aplicado apenas na cidade de São Paulo, pois apenas para esta cidade havia dados suficientes de acidentes com vítimas para fazer a classificação. Nas demais cidades (Xalapa, Quito, Montevidéu e Lima), o sistema apresenta apenas uma sumarização dos dados obtidos pela API do Waze.

Os controles do Painel estão apresentados na Figura 4.4.2.5.1. Nele observa-se a cidade de Xalapa. Sobre as opções de controle disponíveis, há no canto lateral esquerdo as opções de seleção de cidade de interesse, se o processamento é em tempo real ou histórico, seguido do ano, mês, dia da semana, intervalo de tempo e uma contagem dos acidentes reportados pelo Waze. No caso histórico ainda há as opções de controle e apresentação visual dos acidentes por mês. No canto direito observa-se o mapa com georreferenciamento de onde ocorreu o acidente, com histograma de hora, por dia da semana. Seguido de gráfico de barra com as demais distribuições.

Figura 4.4.2.5.1: Painel de visualização dos acidentes.

Fonte: Elaboração Própria

5. Capacitação para utilização e instalação de protótipos

Depois de finalizados, os protótipos foram apresentados para os representantes e técnicos das cidades pela equipe de consultores que os desenvolveram. Todas as instruções necessárias para a utilização e/ou instalação dos protótipos foram apresentados nas respectivas seções do capítulo 4, apresentados nos eventos de capacitação realizados em maio e junho de 2021 e no Produto 5 que essencialmente consolidou apenas as partes instrucionais dos capítulos anteriores.

Neste sentido, dado que o manual se propõe a ser, essencialmente, um consolidado informativo apresentado no Produto 5 as instruções contidas em seu interior são as mesmas já apresentadas nas seções anteriores, optamos por apresentar no texto que se segue apenas as informações referentes aos encontros de capacitação técnica.

5.1. Oficinas de capacitação

Nos dias 25 e 31 de maio e 08 de junho de 2021, realizaram-se encontros virtuais para treinamento e capacitação de representantes das cidades participantes do projeto *Big Data* para o Desenvolvimento Sustentável Urbano.

Cada um dos encontros realizados tinha como objetivo principal capacitar os gestores nos produtos desenvolvidos referentes ao Termo de Referência 8 (TR8). Dentre os protótipos discutidos e disponibilizados para a administração pública desses municípios, temos: i. Estimativa de Poluição do Ar; ii. Painel de comparação de velocidade entre ônibus e carros; o iii. Indicador de congestionamento; a iv. Plataforma de transparência e o iv. GTFS estático.

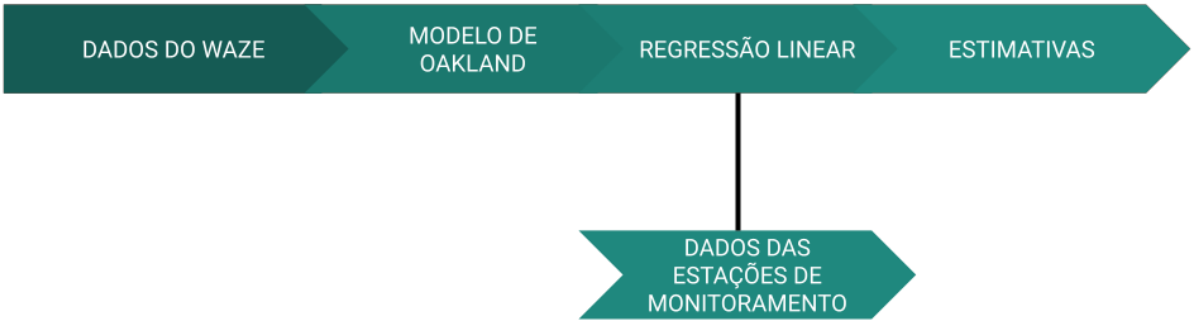
5.1.1. Oficina de Capacitação 1: Estimativa de Poluição do ar e Painel de Comparação de velocidades de carro e ônibus

Estimativa de Poluição do Ar

- O objetivo do protótipo desenvolvido é gerar estimativas para o nível de poluição atmosférica para as cidades de Lima, Montevidéu, Quito e São Paulo.

- O protótipo foi inspirado no modelo de Oakland, objetivando, neste sentido, fornecer predições sobre a poluição das ruas a partir de dados de tipos de vias e trânsito do Waze.
- O modelo de Oakland foi ajustado para abarcar as características de cada cidade participante do projeto. Neste caso, propriamente, as diferenças tais como i. os tipos de vias; ii. o comportamento do trânsito e iii. clima da região necessitam de ajustes.
- Com o modelo de Oakland foi possível desenvolver estas predições para as cidades de Montevidéu, Quito e São Paulo, embora, na prática, o grau de precisão não seja o mesmo do algoritmo treinado para Oakland.
- O processo, neste sentido, se desenvolveu em cinco etapas - simultâneas ou não:

Figura 5.1.1.1: Etapas da construção do protótipo



Fonte: Elaboração própria

- Os modelos foram criados para 4 cidades: i. Lima; ii. Montevidéu; iii. Quito e iv. São Paulo. Para a cidade de Xalapa o modelo não pode ser desenvolvido por dois motivos: i. os dados de apenas uma estação meteorológica estavam disponíveis e ii. os dados do Waze eram limitados em razão do baixo uso do aplicativo.
- Com o modelo criado, uma interface foi desenvolvida para permitir que as predições de poluição sejam visualizadas. A princípio, as predições foram feitas somente para os locais próximos das estações de monitoramento de poluição.

Figura 5.1.1.2: Painel inicial do protótipo

Fonte: Elaboração própria

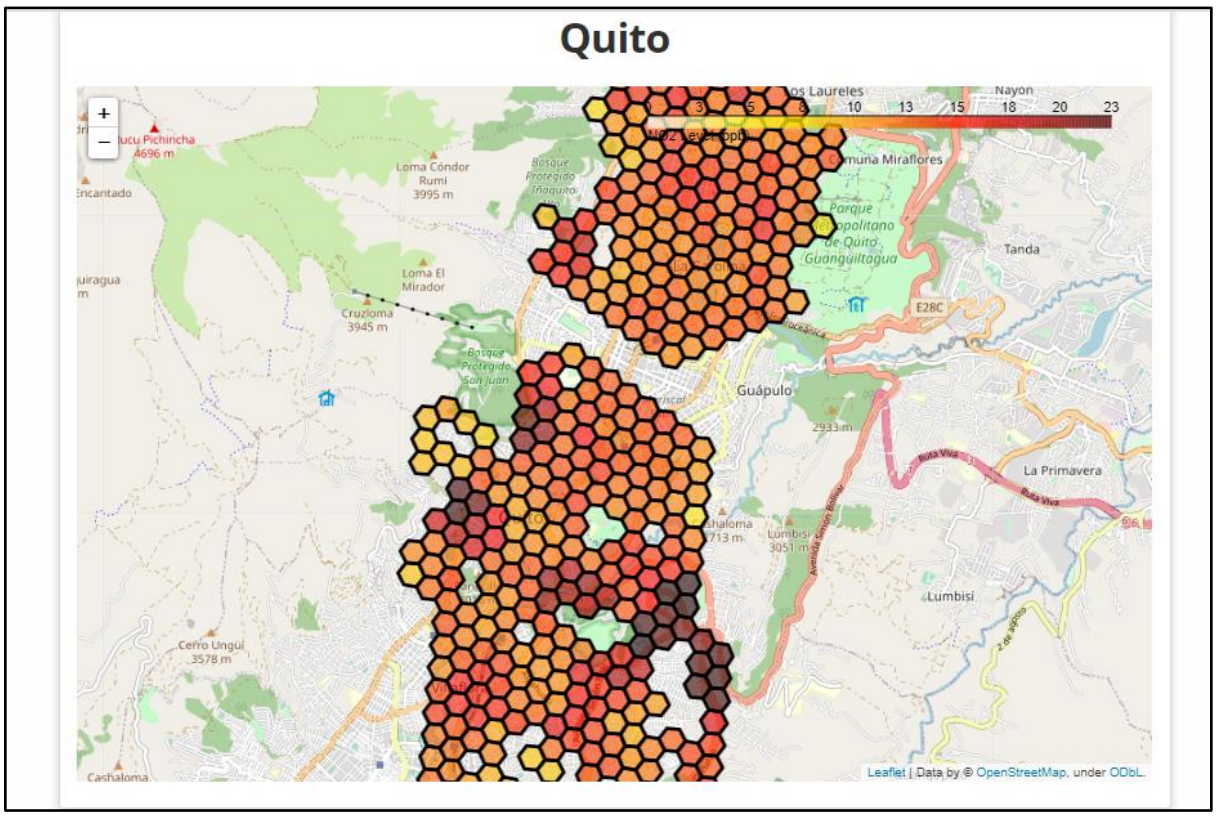
- Dentre as melhorias que podem ser implementadas no modelo utilizado, temos:

  i. criar modelos únicos para cada uma das cidades, de modo que, as características locais sejam consideradas;
  ii. utilizar os dados de outros meses e anos - nesta primeira versão, somente abril de 2019 foi utilizado -, considerando as diferenças climáticas entre os meses.

- A interface criada pode ser acessada em: Smart Cities - Predicting Pollution from Waze Data (smart-cities-pollution.netlify.app).
- O repositório contendo os códigos e dados utilizados na criação da interface pode ser acessado em: mcf1110/smart-cities-pollution: Predicting Pollution from Waze Data (github.com).
- Python foi a linguagem utilizada nos códigos. O programa utilizado foi o Jupyter Notebook. Para maiores informações sobre a linguagem e o programa, acessar: Project Jupyter | Home. Outra maneira de ler os arquivos é por meio da plataforma Google Colaboratory, que pode ser acessada em: Olá, este é o Colaboratory - Colaboratory (google.com).
- No que diz respeito à interpretação das predições exibidas pela interface, é necessário ressaltar que cada hexágono refere-se a ruas e regiões específicas - em acordo com os dados do Waze e Uber. Cada hexágono, neste sentido, possui informações referentes a ele mesmo e aos hexágonos vizinhos. Em relação ao poluente mapeado, propriamente,

NO2 foi o escolhido, considerando que era o único com informações disponíveis para todas as cidades. As cores dos hexágonos representam, neste caso, as gradações e níveis de poluição encontrados naquela localidade.

Figura 5.1.1.3: Predição para Quito

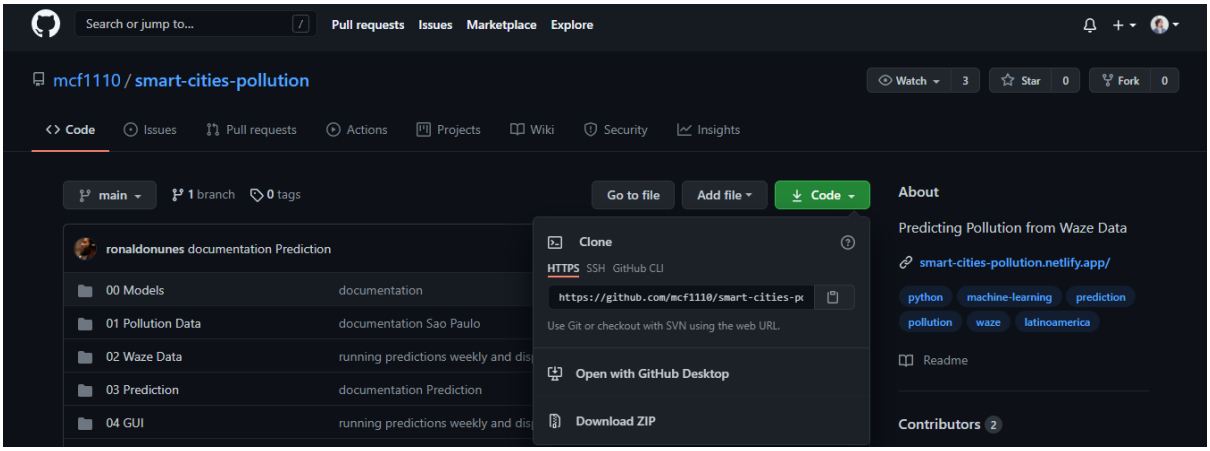


Fonte: Elaboração própria

- Por enquanto, os hexágonos estão disponíveis para apenas alguns pontos das cidades. No entanto, a informação pode ser facilmente expandida no futuro;
- Das cidades analisadas, Montevidéu não possuía dados suficientes e, portanto, as métricas atualmente disponibilizadas não são confiáveis.
- O modelo criado apenas considera as informações de trânsito das cidades. Com ele é possível saber a densidade de trânsito e a densidade de vias, entretanto, as informações referentes a outras fontes de poluição não estão disponíveis porque elas não foram consideradas para este projeto.
- O código pode ser acessado na página do GitHub do projeto - mcf1110/smart-cities-pollution: Predicting Pollution from Waze Data (github.com). É possível fazer

o download dos códigos e dados, clicando-se no botão ‘Code’ e, em seguida, ‘Download ZIP’.

Figura 5.1.1.4: Página inicial do repositório de dados no GitHub



Fonte: Elaboração própria

- Ainda na página do GitHub do projeto, é possível verificar o passo-a-passo para a inclusão de novos dados na guia README.md.
- Atenção, é importante ressaltar que os dados exibidos atualmente na interface referem-se à média de concentração de poluentes em um mês. Por enquanto, os dados não estão em tempo real.

Painel de comparação de velocidade entre ônibus e carros

- O principal objetivo do produto consiste no mapeamento em tempo real da situação do tráfego de ônibus e carros para as cidades de Montevidéu e São Paulo. Por meio de dados de origens diversas, realizou-se o cálculo das velocidades médias de ônibus e carros. Como resultado final, uma interface de monitoramento foi criada, na qual é possível visualizar em tempo real o tempo esperado/necessário para deslocamento, comparando ônibus e carros.

- Para o desenvolvimento do protótipo, a seguinte metodologia foi adotada:

  i. Análise da documentação da rede de transporte público (GTFS);

ii. Seleção das vias para monitoramento;

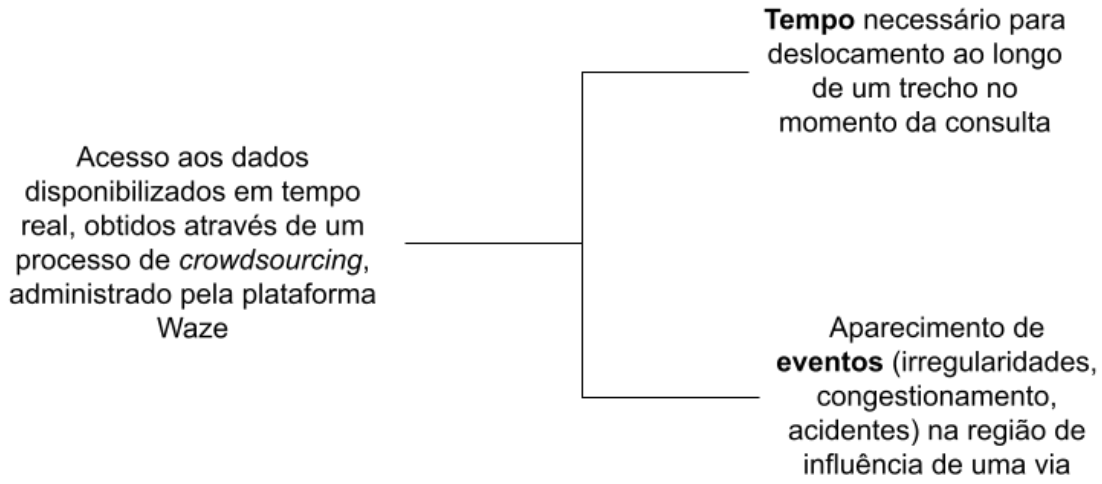
iii. Acesso aos dados de transporte (públicos e privados) via APIs;

iv. Processamento dos dados: cálculo de velocidades e detecção de condições anormais;

v. Disponibilização de métricas em uma interface.

- Os dados de transporte privado são oriundos da plataforma do Waze. Quanto aos dados de transporte público, o GTFS foi utilizado.

Figura 5.1.1.5: Monitoramento do transporte privado



Fonte: Elaboração própria

- **Quanto ao monitoramento do transporte público**, é necessário apontar que:

i. A rota é representada geometricamente por nós e arestas de acordo com o trajeto, onde os nós representam as paradas e as arestas representam vias;

ii. As posições são recebidas em tempo real e corrigidas de acordo com a trajetória esperada do veículo;

iii. Antes do uso destes dados, verificações temporais e espaciais são realizadas a fim de se mensurar a consistência dos dados. Dados inconsistentes são descartados;

iv. As posições definidas na via são utilizadas para calcular a distância e o tempo entre duas posições enviadas por um ônibus;

v. Conforme o envio ocorre em pontos aleatórios da viagem, as posições são interpoladas linearmente para estimação do horário de passagem pelos pontos de parada.

5.1.2. Oficina de Capacitação 2: Índice de Congestionamento e Plataforma de Transparência

Índice de congestionamento

- O objetivo deste produto foi construir um índice de congestionamento de trânsito para as cidades do projeto a partir dos dados do Waze.
- Foi apresentado estudo de caso acerca do congestionamento em São Paulo. A Companhia de Engenharia de Tráfego (CET) é responsável por monitorar o trânsito do município desde 1980. Atualmente, um total de 868 km de vias são monitoradas pela mesma. O processo de monitoramento se dá, sobretudo, por meio da inspeção visual dos agentes de trânsito e pelo sistema de câmeras do Circuito Fechado de Televisão (CFTV).
- O índice de congestionamento (%) calculado pela CET consiste, basicamente, em **(vias congestionadas/vias monitoradas) * 100**.
- O Waze utiliza a API GeoRSS para recuperar os dados de trânsito internamente. Estes dados podem ser obtidos através de um protocolo HTTP a API do Waze.
- Os mapas do Waze são representados por meio de *grafos,* isto é, por um conjunto de vértices e arestas.
- A sensibilidade dos dados do Waze permite que variações e picos de trânsito sejam identificados.
- No que se refere ao desenvolvimento do protótipo, propriamente, constituiu-se como objetivo usar os dados do Waze para obter um indicador de congestionamento de trânsito. Para isso, os dados foram agregados de 30 em 30 minutos, de modo, a gerar filtros de seleção para a aplicação. Em adendo, uma média foi calculada para cada período disponível, permitindo que o usuário comparasse os resultados obtidos. A aplicação foi feita para os municípios de Miraflores, Montevidéu, São Paulo e Quito.

- Entre as vantagens do indicador de congestionamento de trânsito, temos

i. Baixo custo de implementação;
ii. Maior disponibilidade e confiabilidade dos dados;
iii. Sensibilidade na detecção de flutuações críticas e abruptas no trânsito;
iv. Menor tempo de resposta a eventos críticos ou de alto impacto no fluxo de veículos;
v. É compatível com uma variedade de dispositivos.

- Por outro lado, entre as desvantagens do indicador, pode-se apontar

i. Custo de reeducação e treinamento do algoritmo em uma outra escala de magnitude;
ii. Os dados são mais precisos conforme maior o uso do Waze;
iii. Requer uma equipe com conhecimentos em processamento de grandes volumes de dados;
iv. Pode haver incompatibilidade com os mapas utilizados pelo município, devido a necessidade de fazer um levantamento e renomear as vias que serão monitoradas.

- Dentre os próximos passos no aprimoramento do protótipo criado, temos

i. Conectar a aplicação diretamente na API do Waze, de forma, a obter dados em tempo real;
ii. Testar outras agregações temporais, tais como, 5 minutos, 10 minutos ou 15 minutos;
iii. Calcular a função de trânsito em vez de considerar apenas a média;
iv. Atribuir diferentes gradações de cores aos trechos do mapa em acordo com as classificações de trânsito do Waze;
v. Canal de comunicação direto entre este painel e o painel dos agentes de trânsito e outros gestores municipais;
vi. Criar um portal para que a imprensa e os cidadãos tenham acesso ao Indicador de Congestionamento;
vii. Definir se o indicador deve ser exibido em km e/ou em percentual de vias congestionadas;

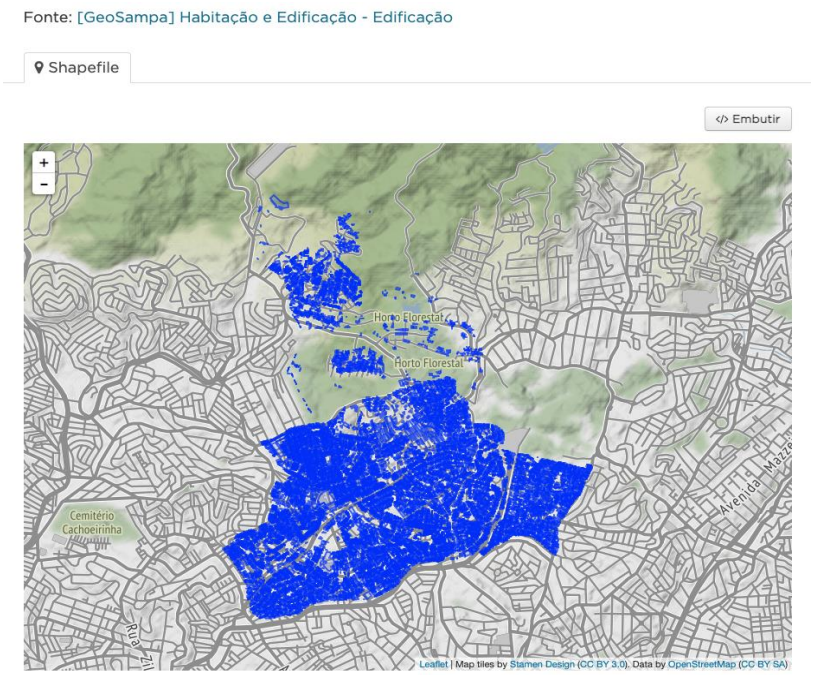
viii. Selecionar e definir as vias que serão monitoradas;

ix. Analisar os dados históricos, de modo a criar uma linha de referência.

Plataforma de Transparência

- A Plataforma de Transparência é o primeiro protótipo desenvolvido no TR8 e possui como objetivo principal o acréscimo de novas usabilidades à plataforma CKAN visando, sobretudo, maior transparência na divulgação de dados e informações que sejam do interesse público.
- Foram apresentadas três novas funcionalidades desenvolvidas dentro da plataforma DataUrbe, entre as quais temos i. criação e visualização de tabelas e gráficos; ii. visualização gráfica de dados oriundos de Shapefiles/GeoJSON e iii. incorporação de projetos em andamento nos municípios em questão.
- No caso do município de São Paulo, a plataforma agora também está integrada com o GeoSampa, permitindo que as informações/dados sejam mais facilmente transferidos. O código está disponível em [luizgabriel/GeoSampa-CKAN: A command line script for importing all GeoSampa data and export automatically to CKAN (github.com)](). Os dados extraídos podem ser visualizados no DataUrbe.

Figura 5.1.2.1: Painel com visualização de dados geocodificados



- No caso dos projetos em andamento, através das credenciais fornecidas a cada um dos municípios participantes, agora os gestores dispõem da opção de acrescentar projetos/planos não iniciados, em andamento ou finalizados na plataforma.

Figura 5.1.2.2: Painel dos conjuntos de dados no DataUrbe

- Para mais informações sobre este processo, conferir o manual de uso dos protótipos (Produto 5) ou as vídeo-aulas disponibilizadas em Termo de Referência 8 (TR8) - YouTube.
- Por fim, o DataUrbe também permite que as cidades disponibilizem gráficos e tabelas já pré-estruturadas aos usuários. Para isso, os gestores podem modificar os conjuntos de dados via Google Planilhas e publicar posteriormente no DataUrbe ou escolher outras ferramentas de *Business Intelligence* - tal como a *AWS Quick Sight* - e compartilhar com o público os resultados e/ou análises feitas. O passo-a-passo deste processo também pode ser encontrado no manual de uso dos protótipos ou na playlist disponibilizada no youtube em Termo de Referência 8 (TR8) - YouTube.

5.1.3. Oficina de Capacitação 3: GTFS Estático

GTFS

- O principal objetivo é demonstrar as possibilidades de uso para dados abertos bem estruturados. Neste caso, especificamente, objetivou-se, sobretudo, realizar um mapeamento da posição dos veículos - do transporte público -, bem como fornecer informações referentes às rotas aos passageiros.
- A arquitetura do protótipo desenvolvido é constituída por 3 partes distintas, entre as quais temos

    i. Web - persiste os dados de interesse através do software MongoDB - neste caso, oriundos do Waze e da plataforma OlhoVivo - e, posteriormente, os integra a interface desenvolvida;
    ii. OTP - software que estabelece uma conexão-tela com a interface desenvolvida;
    iii. GTFS scripts - processo responsável pela geração dos GTFS.

- Os arquivos GTFS consistem em especificações de dados do transporte público - abrangendo ônibus, metrôs e bicicletas. Estas especificações dividem-se em dois grupos distintos, isto é,

    i. GTFS estático - fornece informações a respeito do percurso das linhas, das paradas, horários de chegada e partida, dias de serviço, tarifas e outros.
    ii. GTFS dinâmico - fornece informações referentes à previsão de chegada, a posição dos veículos e aos avisos em tempo real.

- No caso da geração do GTFS estático, é importante ressaltar que cada administração municipal possui um formato diferente de publicação dos dados, sendo, neste caso, necessário scripts adequados para cada das situações encontradas. Entretanto, embora estes scripts tenham sido desenvolvidos em função da realidade encontrada nas cidades participantes do projeto, os códigos foram pensados em termos de fácil replicabilidade por outras gestões.
- Além deste processo, a equipe também desenvolveu representações em grafo do GTFS, tornando possível o acompanhamento em tempo real da posição dos ônibus. Entretanto,

como apenas o município de São Paulo possuía dados dinâmicos, somente a infraestrutura - dos grafos - foi desenvolvida nesse caso.

- Outra possibilidade oferecida pelos dados refere-se a criação de séries históricas, possibilitando posteriormente a integração e a criação de aplicações.
- Para visualizar o tutorial de instalação da aplicação, basta acessar este link: https://gitlab.com/pragmacode/bid_fgv_interscity/smartcities_bigdata/-/raw/master/deploy/demo_ubuntu.mp4
- Open Trip Planner (OTP) é um software livre que agrega os dados do Open Street Map e GTFS e, como resultado, fornece roteiros para o usuário do sistema de transporte. Neste caso, dados os pontos de partida e destino do passageiro, o OTP consulta informações em busca da rota mais rápida e apresenta informações referentes à posição das paradas, os horários de chegada/partida e o tempo de percurso - exibindo as opções de rota encontradas ordenadas de forma decrescente pelo horário de chegada.
- É importante ressaltar que o protótipo foi desenvolvido para as cidades de Lima, Montevidéu, Xalapa e São Paulo.
- Algumas considerações sobre o processo de instalação da aplicação:

  i. O passo inicial consiste na instalação das dependências necessárias para o correto funcionamento da aplicação;
  ii. Em seguida, deve ser realizado, de fato, o download da aplicação no sistema operacional em uso;
  iii. Com isso finalizado, ocorre a instalação de todas as dependências de Python necessárias;
  iv. Com as dependências instaladas, a geração do GTFS para cada uma das cidades ocorre. Caso haja interesse nos arquivos de GTFS, propriamente, eles são facilmente acessíveis neste processo;
  v. Com os GTFS gerados, a aplicação deve ser iniciada. Para isso, uma tecnologia chamada docker[17] deve ser acionada.
  vi. Com este processo finalizado, a página web da aplicação pode ser visualizada.

- Para acessar o repositório do projeto: PragmaCode / bid_fgv_interscity / Smartcities BigData · GitLab

[17] Empowering App Development for Developers | Docker

5.2. Links

As gravações dos treinamentos realizados estão disponíveis na lista de vídeos abaixo:
https://www.youtube.com/playlist?list=PLJHnxoYR52j6gTZV_e4GZsrBkntIdtl8Y

Dia 1 – Estimativa de Poluição:
https://youtu.be/dTOi7DfqKcU

Dia 1: Comparação de velocidades:
https://youtu.be/KSFwodiSFbM

Dia 2 - Índice de congestionamento:
https://www.youtube.com/watch?v=eB0XCI490EI

Dia 2 - Portal da Transparência:
https://youtu.be/s9RWmRRS2Do

Dia 3 - GTFS – Parte 1:
https://www.youtube.com/watch?v=Xc6ngu3ihx4

Dia 3 - GTFS – Parte 2:
https://youtu.be/Xc6ngu3ihx4

As apresentações de slides utilizadas pelos palestrantes estão disponíveis em https://smartcities-bigdata.fgv.br/.

Referências Bibliográficas

ADAMSON, Christopher; Star Schema The Complete Reference.

BELIN, Matts-Åke; TILLGREN, Per; VEDUNG, Evert. Vision Zero–a road safety policy innovation. International journal of injury control and safety promotion, v. 19, n. 2, p. 171-179, 2012.

BROWN, D. E. (5 de junho de 2014). “What’s the difference between Business Intelligence and Big Data?”.

CALATAYUD, A., S.S GONZÁLEZ, MAYA, F.B., F. GIRALDEZ, E J.M. MÁRQUEZ. “Congestión urbana en América Latina y el Caribe: características, costos y mitigación.” Monografía del BID. 2021.

CANALYS (2020). "Cloud market share Q4 2019 and full-year 2019." Canalys Cloud Channel Analysis.

CARABETTA, João. “Mining jams into pollution: how Waze data helps estimating air pollution in large cities.” 2019.

CASTELLI, Mauro, Fabiana Martins Clemente, Aleš Popovič, Sara Silva, e L. Vanneschi. “A machine learning approach to predict air quality in California.” Complexity 2020. 2020.

CATALA, Martin, Samuel Downing, and Donald Hayward. “Expanding the google transit feed specification to support operations and planning.” *Dept. of Transportation. Research Center.* Florida, 2011.

CEBR, Centre for Economics and Business Research. “The future economic and environmental costs of gridlock in 2030.” Report for INRIX. July de 2014.

CETESB, Companhia de Tecnologia de Saneamento Ambiental. “QUALIDADE DO AR NO ESTADO DE SÃO PAULO.” Série de Relatórios Anuais. São Paulo, 2020.

Chen, Tianqi, e Guestrin, Carlos. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). 2016.

COMISIÓN MULTISECTORIAL PARA LA GESTIÓN DE LA INICIATIVA DEL AIRE LIMPIO PARA LIMA. “Diagnóstico de la Gestión de la Calidad Ambiental del Aire de Lima y Callao.” 2019.

DAPPER, Steffani, Caroline Spohr Nikoli, e Roselaine Ruviaro Zanini. “Poluição do ar como fator de risco para a saúde: uma revisão sistemática no estado de São Paulo.” Estudos Avançados. 2016.

DATAURBE. https://dataurbe.appcivico.com/.

DA SILVA FIGUEIREDO, V., & LADEIRA DOS SANTOS, W. (2013). "Transparência e controle social na administração pública." Temas de Administração Pública.

DISTRITO METROPOLITANO DE QUITO, Secretaría de Ambiente. “Informe Final Inventario de

Emisiones de Contaminantes Criterio, DMQ 2011.” 2014.

FGV (2018). Termo de Referência 6. “Desarrollo de la Herramienta Ambiental en Línea para el Registro, Validación y Disposición de Datos (AOCD)”, Big Data para el desarrollo urbano sostenible.

FGV (2020). Termo de Referência 1. “Relatório Técnico com Lista e Diagnóstico dos Dados Públicos e Privados Disponíveis nas Cidades de São Paulo, Montevidéu, Quito, Xalapa e Miraflores”, Big Data para o Desenvolvimento Urbano Sustentável.

FI-JOHN, Changa, Li-Chiu Chang, Che-Chia Kanga, Yi-Shin Wanga, e Angela Huanga. “Explore spatio-temporal PM2.5 features in northern Taiwan using.” Science of the Total Environment. 2020.

GARTNER INSTITUTE (2019). Quadrant for Cloud Infrastructure as a Service, worldwide.

HABERMANN, Mateus, Andrea Paula P. Medeiros, e Nelson Gouveia. “Tráfego veicular como método de avaliação da exposição à poluição atmosférica nas grandes metrópoles.” Revista Brasileira de Epidemiologia. 2011.

LASZEWSKI, Tom e col. “Cloud Native Architectures: Design high-availability and cost-effective applications for the cloud”

LIMBERGER, T. (2007). Transparência administrativa e novas tecnologias: o dever de publicidade, o direito a ser informado e o princípio democrático. Revista de Direito Administrativo.

MA, Jinghui, Zhongqi Yu, Yuanhao Qu, Jianming Xu, e Yu Cao Cao. “Application of the XGBoost Machine Learning Method in PM2.5 Prediction: A Case Study of Shanghai.” Aerosol Air Qual. 2020.

MONTEIRO, J., I. PONS, e R SPEICYS. “Big Data para análise de métricas de qualidade de transporte: metodologia e aplicação.” *Associação Nacional de Transportes Públicos (ANTP)– Série Cadernos Técnicos 20.* 2015.

NEWMAN, Sam; Monolith to Microservices: Sustaining Productivity While Detangling the System.

OMS, Organização Mundial da Saúde. “Health aspects of air pollution with particulate matter, ozone and nitrogen dioxide.” Report on a WHO Working Group. 2003.

OMS, Organização Mundial da Saúde. “Nueve de cada diez personas de todo el mundo respiran aire contaminado” https://www.who.int/es/news/item/02-05-2018-9-out-of-10-people-worldwide-breathe-polluted-air-but-more-countries-are-taking-action. 2018.

ONU. “Las ciudades seguirán creciendo, sobre todo en los países en desarrollo.” 2008.

QUITO, Secretaria de ambiente de. “Los contaminantes communes del aire y sus efectos sobre la salud humana.” Informe. s.d.

RENO, NV, (27 de Abril de 2018). Cloud Growth Rate Increased Again in Q1; Amazon Maintains Market Share Dominance.

SHAHRIAR, Shihab Ahmad, Imrul Kayes, Kamrul Hasan, Mohammed Abdus Salam, e Shawan Chowdhury. “Applicability of machine learning in modeling of atmospheric particle pollution in Bangladesh.” Air Quality, Atmosphere & Health. 2020.

SÃO PAULO. Decreto nº 58.717, de 14 de abril de 2019. Institui o Plano Municipal de Segurança Viária 2019/2028 e o Comitê Permanente de Segurança Viária do Município de São Paulo. Legislação Municipal, São Paulo, Diário Oficial da Cidade de 18/04/2019, p. 1.

SETH, Gilbert e LYNCH, Nancy; Brewer’s Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services.

SHEET, SWOV Fact. Sustainable safety principles, misconceptions, and relations with other visions. SWOV Institute for Road Safety Research: Leidschendam, The Netherlands, 2013.

STALCUP, Katy (5 de fevereiro de 2020). AWS vs. Azure vs. Google Cloud Market Share 2020: What the Latest Data Shows.

SYNERGY RESEARCH GROUP (2018). "Cloud Growth Rate Increased Again in Q1; Amazon Maintains Market Share Dominance."

TAURION, C. (25 de setembro de 2014). Entre os Vs do Big data, velocidade cresce em importância.

TAURION, C. (27 de junho de 2013). Entrevista com Cezar Taurion: O estágio atual do Big Data no Brasil. Disponível em IBM.

VAN ESSEN, H., et al. “Handbook on the external costs of transport, version 2019 (No. 18.4 K83. 131).” 2019.

WAZE, WazeTraffic-Data Specification Document Version 2.8.

WELLE, B., et al., Cities Safer by Design, World Resources Institute, WRI.

WILDER, Bill. “Cloud Architecture Patterns: Using Microsoft Azure”. O'Reilly Media, 2012.

WONG, James C. “Use of the general transit feed specification (GTFS) in transit performance measurement.” *PhD diss., Georgia Institute of Technology.* 2013.

ANEXOS

ANEXO 1: Ata de reunião - Espanhol

ACTA DE REUNIÓN

En el día 18 de febrero de 2020, los representantes de las ciudades participantes del proyecto **Big Data para el Desarrollo Urbano Sostenible** se reunirán en videoconferencia por Skype para tratar de los resultados de las sesiones de Design Thinking celebradas en el II Encuentro Regional del proyecto, en Miraflores en los días 4 y 5 de diciembre.

Estuvieran presentes en la reunión:

Tabla 1: Participantes de la reunión

Integrante	Institución
Marcus	FGV
Pablo	FGV
Bruno	FGV
Poco	FGV
Ciro	FGV
Patricia	FGV
Edgard	APP Cívico
Jimena	Miraflores
Jazmin	Quito
Antonio y Rafael	Xalapa
Leonardo	Montevideo

Los participantes presentaron sus consideraciones y sugerencias a los resultados alcanzados por las sesiones de Design Thinking para que sean evaluadas y empleadas en el desarrollo del TR8.

Recordamos que:

1. ya mapeamos los datos abiertos públicos y privados de las cinco ciudades según las categorías de interés;
2. analizamos en detalle la calidad de los datos disponibles de acuerdo con las normas internacionales;
3. presentamos un conjunto de recomendaciones para la publicación de datos calificados para la generación de valor publico con los datos disponibles;
4. construimos una plataforma donde esos datos están siendo almacenados y estandarizados para ser interoperables; y
5. realizamos aquellas sesiones de Design Thinking para mapear los problemas que pueden ser corregidos por políticas públicas basadas en esos datos. Esta es una metodología muy útil porque permite colocar la persona en el centro del problema y promueve empatía y experimentación, combinando análisis e intuición.

Recordamos también que la pregunta principal que nosotros proponemos en la primera etapa de la sesión fue: ¿cómo los datos mapeados pueden promover desarrollo local a problemas comunes en la cinco ciudades? Y en el segundo día provocamos todos con una pregunta más centrada en las áreas de movilidad y medio ambiente, porque tenemos más datos en relación con estas áreas.

El resultado fueron dos días de dinámica y discusión. Todos los post-its fueron colocados en posiciones estratégicas e después colectado y analizado para construir, basadas en los datos, políticas públicas con lógica.

Para eso intentamos entender:

1. PROBLEMA: ¿Qué problemas comunes se pueden resolver con una política basada en los datos existentes?
2. PÚBLICO: ¿Quién se beneficiaría de esta política?

3. INSUMOS: ¿En qué datos / sistemas se puede trabajar?
4. ACCIONES: ¿Qué interacciones / soluciones tecnológicas son posibles con esos datos y que van a resolver los problemas para el publico?
5. RESULTADOS: ¿Cómo es posible evaluar el impacto de esta política en el futuro?¿Cómo asegurarse de que el problema se soluciono y que el público se benefició?

Y como resultado final obtuvimos cuatro políticas posibles:

Tabla 2: Políticas

POLÍTICA	PÚBLICO	PROBLEMAS	ACCIONES	RESULTADOS
Planeación urbano	Gestor público	Dificultad de planificación y falta de movilidad y incapacidad para tomar decisiones rápidas y efectivas.	Plataforma con información integrada con datos de transporte, incluso camiones, de uso del suelo, de cámaras de estacionamiento y de Waze para planeación de largo plazo.	Mejora en la tomada de decisión, en la gestión de las ciudades e en la movilidad.
			Creación de panel de datos con velocidad excesiva y accidentes en tiempo real con víctimas y sistema de advertencia sobre puntos con mayor riesgo de muerte por exceso de velocidad combinando datos de Waze con datos de las aseguradoras.	Mejora en la planeación e en la seguridad viaria
			User friendly aplicación para visualizar datos de calidad del aire	Calidad de vida
			Aplicativo con información de horario de buses y datos de GPS integrada con datos del Waze para disminuir el tiempo del viaje.	Traslados más rápidos y fácil en la ciudad.

Transparencia	Alcaldes	Dificultad para actuar con transparencia por problemas en el sistema de comunicación.	Creación de plataforma con informaciones sobre desarrollo de proyectos relevantes.	Transparencia en objetivos y mejoras en el servicio público
Participación	Ciudadanos especialmente estudiantes	Manipulación por las redes sociales. Frustración con la política y un sentido de no participación.	Plataformas de participación y monitoreo	Sensación de pertenencia
Capacitación	Empresas y trabajadores	Oportunidades perdidas por falta de competencia tecnológica.	Soluciones de capacitación y calificación	Rentabilidad

Llamamos la primera de política de planeación urbano cuyo público beneficiado sería el gestor público que enfrenta problemas como falta de movilidad e incapacidad para tomar decisiones rápidas y efectivas y dificultad de planificación. La soluciones fueron:

1. Plataforma con información integrada con datos de transporte, incluso camiones, de uso del suelo, de cámaras de estacionamiento y de Waze para planeación de largo plazo cuyo resultado sería la mejora en la tomada de decisión, en la gestión de las ciudades y en la movilidad.
2. Creación de panel de datos con velocidad excesiva y accidentes en tiempo real con víctimas y sistema de advertencia sobre puntos con mayor riesgo de muerte por exceso de velocidad combinando datos de Waze con datos de las aseguradoras para mejorar la seguridad viaria
3. User friendly aplicación para visualizar datos de calidad del aire y que impactaría la Calidad de vida
4. Aplicativo con información de horario de buses y datos de GPS integrada con datos del Waze para disminuir el tiempo del viaje para mejorar traslados con más rapidez y facilidad.

La política de transparencia beneficiaria los alcaldes y gestores públicos que sienten ser difícil actuar con transparencia por problemas en el sistema de comunicación. Para ellos una

plataforma con informaciones sobre desarrollo de proyectos relevantes mejoraría la transparencia en objetivos y en el servicio público.

Otra política que ha aparecido fue la que llamamos de participación para los ciudadanos, especialmente los estudiantes, resolver el problema de manipulación por las redes sociales. Ellos sienten frustración con la política y un sentido de no participación y una política como esa ayudaría con la sensación de pertenencia.

Por último la política de capacitación para empresas y trabajadores para mejorar la competencia tecnológica con soluciones de capacitación y calificación mejorando la rentabilidad.

Después pasamos a decidir cuáles políticas eran posibles de desarrollar considerando el tiempo que tenemos y la capacidad técnica para construir la propuesta para el TR8, cuyo objetivo es desarrollar un entorno de datos para monitorear y calificar los servicios públicos en las cinco ciudades. El objetivo es crear prototipos de un ecosistema tecnológico especializado capaz de tratar y transformar BigData en el desarrollo de políticas públicas urbanas basadas en evidencias.

Los resultados de la discusión en que todos los presentes tuvieran oportunidad de contribuir fue:

1.Política de Transparencia: La mayoría de las ciudades han elegido esta política porque es compatible con las prioridades de la agenda gubernamental.

> Propuesta: Utilizar la propia plataforma CKan que está siendo alimentada por las ciudades como un medio para garantizar el fácil acceso de los ciudadanos a la información producida por las ciudades. Se puede agregar capacidad tecnológica para permitir la verificación cruzada de información, producción de gráficos, visualizaciones y análisis específicos de acuerdo con las necesidades del usuario.

2. Política de planeación urbano: las ciudades han comprendido ser útil un aplicativo con información de horario de buses y datos de GPS para mejorar la movilidad urbana.

> Propuesta: Creación de arquitectura e ingeniería de modelos estandarizados y condensados en paneles para ofrecer subsidios para la planificación y gestión de los transportes públicos. Se explorarán los datos de GTFS de los buses integrados con los

datos generados por el sistema de colaboración del Waze que hoy tenemos acceso y que representa una fuente importante de información e ideal para dialogar con otras bases de datos. Mediante el aprendizaje automático será posible ofrecer insumos para la operación y planificación de la movilidad urbana en cada ciudad.

3. Política de planeación urbano: también se señaló la necesidad de un panel de datos con accidentes en tiempo real y un sistema de advertencia sobre puntos con mayor riesgo de muerte por exceso de velocidad.

Propuesta: Proyección y construcción de una herramienta para ayudar a la gestión pública tomar decisiones operativas en el campo de la movilidad urbana de las ciudades a través de la visualización en tiempo real de cambios repentinos en el grado de congestión de las carreteras, revelando anomalías en el uso de la carretera normalmente causadas por accidentes. Mediante la integración de la información disponible en Waze y otras bases de datos y, en el proceso de aprendizaje automático, será posible visualizar en tiempo real la necesidad de intervenciones específicas para mejorar el flujo de vehículos en las ciudades, generando alertas de riesgo correspondientes cuando necesario.

4. Política de capacitación: Los representantes de las ciudades han comprendido que esta política, por estar incluida en el desarrollo de las tres anteriores, no sería tratada por separado.

Ahora todos se han comprometido a rápidamente validar estas opciones y pensar en otras soluciones posibles para políticas públicas basadas en los datos. Solicitamos que las contribuciones se envíen a más tardar el 18 de marzo para que tengamos tiempo de cerrar la propuesta TR8.

ANEXO 2: Plataforma de Transparência- Links e vídeos

Vídeo Aula – Cadastro de Projetos Públicos
https://youtu.be/Jusm-n8kP3w

Código Fonte – Cadastro de Projetos Públicos
https://github.com/luizgabriel/CKAN.ProjetosPublicos

Código Fonte – Integração GeoSampa CKAN
https://github.com/luizgabriel/GeoSampa-CKAN

Vídeo Aula – Integração GeoSampa CKAN
https://youtu.be/RaGQNkqp9Q8

Vídeo Aula – Tour pelo Dataurbe
https://youtu.be/dhwtMDMXNRM

Vídeo Aula – Visualizando dados do Dataurbe
https://www.youtube.com/watch?v=N1G5hw_dX4Q

Visualização – Siniestros de trânsito em Uruguay em el año 2011
https://dataurbe.appcivico.com/dataset/unasev-siniestros-de-tr-nsito-en-uruguay-en-el-a-o-2011/resource/830b80d0-b201-49a3-9c4d-0d350fd30902?view_id=e9952261-c246-4987-8148-d1d73743077b

Vídeo Aula – Acidentes de trânsito no Uruguai em 2011
https://youtu.be/IjceQabtwek

Visualização Big Data - SO2 Concentrado na Atmosfera de Montevidéu
https://dataurbe.appcivico.com/dataset/mides-indicador-10735/resource/47962698-33be-4a9b-ba43-2a2e7ec16902?view_id=311bde4d-9464-4a5d-b6a3-47b6571259e1

Vídeo Aula – SO2 Concentrado na Atmosfera de Montevidéu
https://youtu.be/WauBa1lC9fU

Vídeo Aula – Pontos de Concentração de População em Situação de Rua em São Paulo
https://youtu.be/gWV8HzCUS3A

Vídeo Aula – Demonstração com Google Planilhas
https://youtu.be/-U1OGc8LflA

ANEXO 3: Estimativa de Poluição do Ar - Links e vídeos

Interface para as cidades do projeto

https://smart-cities-pollution.netlify.app.

Modelo de Oakland (Carabetta 2019)

https://github.com/JoaoCarabetta/master_thesis

Repositório contendo os códigos e dados utilizados na criação da interface

mcf1110/smart-cities-pollution: Predicting Pollution from Waze Data (github.com).

Informações sobre o Jupyter Notebook

https://jupyter.org/

Google Colaboratory

https://colab.research.google.com/notebooks/intro.ipynb

Vídeo da capacitação de estimativa de poluição do ar

CAPACITAÇÃO | DIA 1 | Estimativa de Poluição do Ar - YouTube

ANEXO 4: Apoio ao Transporte Público - Links e vídeos

Comparação de velocidades

Protótipo para a cidade de Montevidéu

smt.scipopulis.com:5007/comparativo/montevideo.

Repositório de códigos e dados

bitbucket.org/scipopulis/bid-comparativo

GTFS

Tutorial de instalação

https://gitlab.com/pragmacode/bid_fgv_interscity/smartcities_bigdata/-/raw/master/deploy/demo_ubuntu.mp4

Capacitação técnica

CAPACITAÇÃO | DIA 3 | GTFS - Parte 1 - YouTube e CAPACITAÇÃO | DIA 3 | GTFS - Parte 2 - YouTube.

Repositório de códigos e dados

PragmaCode / bid_fgv_interscity / Smartcities BigData · GitLab.

ANEXO 5: Apoio a Gestão do Trânsito

Índice de Congestionamento

Descrição dos arquivos .csv gerados

O arquivo *traffic_now.csv* é gerado pela requisição no link do da API do Waze. Os dados gravados no arquivo são como uma fotografia do status do trânsito no intervalo de cinco minutos.

Campos do arquivo *traffic_now.csv*

Campo	Tipo	Descrição
`timestamp`	YYYY-MM-DD HH:MM:SS	Registro da data e hora do evento
`new_street`	char (250)	Nome da rua
`level`	char(3)	Categoria de classificação de lentidão do trânsito do Waze. Varia de 3-5
`length`	int	Comprimento da fila
`5min`	YYYY-MM-DD HH:MM:SS	Timestamp arredondado para o intervalo de cinco minutos
`time_hm`	HH:MM	Hora e minutos do do campo 5min

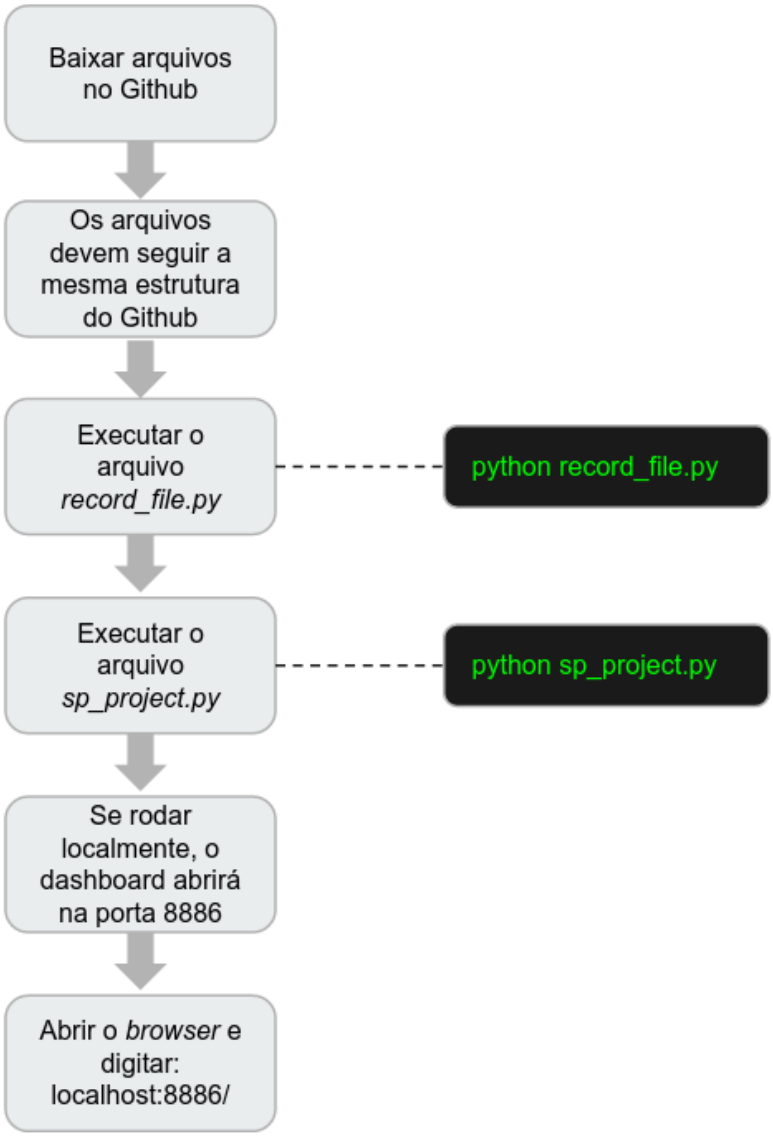
O arquivo *map.csv* é o arquivo sobrescrito a cada cinco minutos e que armazena os pontos que definem um segmento congestionado no mapa *real-time.*

Campos do arquivo *map.csv*

Campo	Tipo	Descrição
`latitude`	char(15)	Latitude
`longitude`	char(15)	Longitude

`level`		Categoria de classificação de lentidão do trânsito do Waze. Varia de 3-5

Passo a passo para executar o painel



Recomendamos que não seja realizada uma busca com períodos maiores que três dias. A busca da base histórica está armazenada na nuvem da Amazon, por esta razão o tempo de download

cresce a uma taxa mais rápida que a execução da *query* para o mesmo intervalo de dias. Para um intervalo de um dia os dados baixados ficam em torno de 500Mb e o tempo de execução da *query* aproximadamente 100s. Para um intervalo de cinco dias os dados baixados vão para quase 6Gb, enquanto o tempo de execução da *query* é de aproximadamente 170s.

Lista das ferramentas utilizadas para o desenvolvimento do painel de congestionamento

Nome	Versão	Descrição
`Python`	`3.7.4`	Linguagem de Programação
`Pandas`	`0.25.1`	Biblioteca Python
`Numpy`	`1.17.2`	Biblioteca Python
`json`	`2.0.9`	Biblioteca Python
`requests`	`2.22.0`	Biblioteca Python
`Dash`	`1.19.0`	Biblioteca Python
`Plotly`	`4.14.3`	Biblioteca Python
`GeoPandas`	`0.6.2`	Biblioteca Python
`shapely`	`1.6.4.post2`	Biblioteca Python
`retry`	`retry-0.9.2`	Biblioteca Python

Para instalação de bibliotecas Python recomendamos os instaladores pip ou conda, então

Pip

https://pypi.org/

Conda

https://www.anaconda.com/products/individual

Repositório de dados e códigos do protótipo

diluisi/BID (github.com).

Capacitação técnica do protótipo

CAPACITAÇÃO | DIA 2 | Índice de Congestionamento - YouTube.

ANEXO 6: Identificação de Accidentes

Repositório de códigos e dados

bruAristimunha/fgv-bid-accident (github.com).

ANEXO 7 - Programação da Capacitação

Oficina de Capacitação 1: Estimativa de Poluição do ar e Painel de Comparação de velocidades de carro e ônibus

Plataforma: Zoom.
Data: Terça-feira, 25 de maio, 2021.
Horário: 16:00 - 18:00 GMT-3

Palestrantes

- ☐ Matheus Fernandes
- ☐ Ronaldo Nunes
- ☐ Raphael Y. de Camargo
- ☐ Roberto Cardoso
- ☐ Camilla Perotto

Duração: 1h24´ZOOM
Total de participantes (incluindo palestrantes e organizadores): 22
Máximo de participantes simultâneos (incluindo palestrantes e organizadores: 13

Tabela: Lista de participantes

Cidade/Organização	Nome do participante
BID	Maurício Bouskela
FGV	Ciro Biderman
FGV	Claudia Oshiro

FGV	Goret Paulo
FGV	Marcus Mendonça
FGV	Patrícia Alencar
Miraflores, Peru	Waldir Tume Ledesma
Montevidéu, Uruguai	Daniel Muniz
Montevidéu, Uruguai	Eduardo Cuitiño
Montevidéu, Uruguai	Florencia Gallo
Montevidéu, Uruguai	Gustavo Arbiza
Quito, Equador	Cristina Cevallos
Quito, Equador	Jazmín Campos
Quito, Equador	Kleber Gualacata
Scipopulis	Roberto Cardoso
Scipopulis	Camilla Perotto
São Paulo, Brasil	Luan Ferraz Chaves
Quito	Pablo Rivera
Quito	Richard Salas
UFABC	Matheus Fernandes
UFABC	Raphael Camargo
UFABC	Ronaldo Nunes
Xalapa, México	Antonio Sobrino

Oficina de Capacitação 2: Índice de Congestionamento e Plataforma de Transparência

Plataforma: Zoom.
Data: Segunda-feira, 31 de maio, 2021.
Horário: 16:00 - 18:00 GMT-3

Palestrantes

- ☐ Diego Silva
- ☐ Raphael Y. de Camargo
- ☐ Luiz Gabriel

Duração: 1h29´ZOOM
Total de participantes (incluindo palestrantes e organizadores): 20
Máximo de participantes simultâneos (incluindo palestrantes e organizadores: 18

Tabela: Lista de participantes

	Nome do participante
	Alvaro Rettich
Xalapa	Antonio Sobrino
São Paulo	Antonio Vieira
	Augustina Tapia
	Boris Goloubintseff
UFABC	Bruno Aristimunha
Quito	Cristina Cevallos
Montevidéu	Daniel Muniz
	Gobierno Abierto
	Gustavo Arbiza
São Paulo	Isis Pereira
Quito	Jazmín Campos
Miraflores	Joel Romani
	Kimberlly Ramon Lossio
São Paulo	Luan Ferraz Chaves
	Miguel Granados
São Paulo	Morino CET

Miraflores	Ottoniel Waldir Tume
Pragma Code	Rafael Reggiani Manzo
	Rodrigo Escalante

Oficina de Capacitação 3: GTFS Estático

Plataforma: Zoom.
Data: Terça-feira, 08 de junho, 2021.
Horário: 16:00 - 18:00 GMT-3

Palestrantes

- ☐ Raphael Y. de Camargo
- ☐ Rafael Reggiani Manzo

Duração: 1h20´ZOOM
Total de participantes (incluindo palestrantes e organizadores): 22
Máximo de participantes simultâneos (incluindo palestrantes e organizadores: 13

Tabela: Lista de participantes

Cidade/Organização	Nome do participante
FGV	Ciro Biderman
FGV	Claudia Oshiro
UFABC	Raphael Camargo
Pragma Code	Rafael Manzo
FGV	Jorge Poco
FGV	Patrícia Mello
São Paulo	Luan Ferraz Chaves
Quito	Kleber Gualacata
Montevidéu	Daniel Muniz

Xalapa	Antonio Sobrino