`HURRICANE`: um Serviço para Gerência de Dados de Aplicações de Cidades Inteligentes*

Maicon Banni[1], Isabel Rosseti[1], Daniel de Oliveira[1]

[1]Instituto de Computação, Universidade Federal Fluminense (UFF), Niterói, RJ, Brazil

maiconbanni@id.uff.br, {rosseti,danielcmo}@ic.uff.br

Resumo. *O conceito de Cidades Inteligentes ganhou relevância, em especial na última década, por conta da proliferação de dados de cidades. O objetivo do uso desses dados é melhorar os serviços oferecidos à população, por meio do desenvolvimento de aplicações que consomem dados espaço-temporais. Esses dados trafegam por um fluxo que vai desde a coleta, pré-processamento até a visualização. Muitas das soluções existentes para Gerência de Dados neste contexto, ou são específicas para uma aplicação/domínio, ou não consideram todo o ciclo de vida do dado. Nesse artigo, apresentamos o* `Hurricane`*, um serviço configurável e extensível, avaliado com uma aplicação na área de segurança pública, que tem como objetivo permitir que os diferentes usuários envolvidos realizem a gerência e a análise dos dados no contexto de aplicações de Cidades Inteligentes de forma integrada e eficiente durante todo o ciclo de vida do dado.*

Abstract. *The concept of Smart Cities has gained relevance, especially in the last decade, due to the availability of city data. The purpose of using this data is to improve the services offered to the population, through the development of applications that manipulate Spatio-temporal data. These data are processed in a dataflow that starts with the collection, pre-processing, and ends with visualization. Many of the existing solutions for Data Management in this context, are either specific to a particular application/domain or do not consider the entire data life cycle. In this paper, we present* `Hurricane`*, a configurable and extensible service, which was evaluated with an application in the area of public security, to allow the different users involved to manage and analyze data in the context of applications Smart Cities in an integrated and efficient way throughout the data lifecycle.*

1. Introdução

O conceito de *Cidades Inteligentes* ganhou muita relevância na última década [Bilal et al. 2020]. Diversas iniciativas têm sido propostas com os mais variados objetivos, *e.g.*, segurança pública [Chen et al. 2017, Lourenço et al. 2018], saúde [Caban and Gotz 2015], *etc*. Independentemente do objetivo, o foco de uma Cidade Inteligente é ser capaz de gerenciar e utilizar de forma eficiente e eficaz a infraestrutura e os serviços da cidade para oferecer bem-estar aos seus cidadãos [Bilal et al. 2020]. Uma Cidade Inteligente depende fortemente do uso de dados para planejar políticas públicas.

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. A pesquisa foi também apoiada parcialmente por CNPq e FAPERJ.

De fato, nos últimos anos, a infraestrutura das Cidades Inteligentes tem sido capaz de disponibilizar dados em seu grão mais fino e em escalas espaço-temporais sem precedentes (*e.g.*, dados de sensores, telefones, redes sociais, *etc.*) [Nandury and Begum 2016]. Assim, a existência de uma Cidade Inteligente depende de quão bem as organizações que a compõem são capazes de analisar e extrair conhecimento útil dos dados que estão sendo gerados/obtidos. No entanto, o acesso, o processamento e a extração de conhecimento a partir desses dados pode não ser trivial, pois se faz necessário integrar dados de diferentes formatos, granularidades e volumes.

Apesar de diversos portais de dados abertos no contexto de Cidades Inteligentes já terem sido disponibilizados nos últimos anos [Pisco and Marques-Neto 2021], muitas aplicações de Cidades Inteligentes precisam consumir dados previamente integrados em algum nível, e nem sempre os dados disponibilizados nos portais já passaram por uma integração prévia. Por exemplo, no cenário de análise de criminalidade em centros urbanos, diversos tipos de dados podem ser utilizados para uma análise integrada: (i) dados de boletins de ocorrência (ii) distribuição espacial da população de rua, (iii) atendimentos por uso de drogas, *etc*. Enquanto que os dados de boletins de ocorrência podem ser obtidos no portal de secretarias de segurança (*e.g.*, SSP-SP (`https://www.ssp.sp.gov.br`), os indicadores populações de rua e usuários de drogas podem ser obtidos no portal do IPEA (`https://www.ipea.gov.br`). Integrar tais dados, organizá-los e disponibilizá-los para terceiros requer um esforço manual por parte do usuário que não pode ser negligenciado. De fato, esse é um problema em aberto e um grande desafio ainda no contexto de Cidades Inteligentes [Raghavan et al. 2019]. Assim, soluções que sejam capazes de auxiliar na gerência integrada de tais dados de forma automática se fazem necessárias.

Existem diversos trabalhos na literatura que tem como objetivo apoiar a gerência de dados no contexto de Cidades Inteligentes. Entretanto, as soluções existentes ou são específicas para um determinado domínio de aplicação [Zhou et al. 2021, Bellini et al. 2021, Garcia-Font 2020] ou focadas em algum tópico como transferência de dados [Nandury and Begum 2016] ou consultas semânticas [Silva et al. 2021]. Mesmo as soluções que se apresentam como arcabouços genéricos de Gerência de Dados [Liu et al. 2017, Jindal et al. 2020], não consideram a importância da integração dos dados durante todo seu ciclo de vida, *i.e.*, não é possível saber qual dado foi derivado a partir de um determinado dado bruto de entrada e qual etapa do processo realizou tais transformações (e quem as executou), *i.e.*, a sua proveniência [Freire et al. 2008, de Oliveira et al. 2018]. De fato, [Ribeiro and Braghetto 2021] discutem a importância dessa questão e definem uma arquitetura conceitual que engloba todos os serviços necessários para gerência de dados no contexto de Cidades Inteligentes. De acordo com [Ribeiro and Braghetto 2021], uma arquitetura para gerência de dados em Cidades Inteligentes deve considerar as etapas de: (i) ingestão, (ii) gerência de metadados e proveniência, (iii) processamento de dados e (iv) consulta aos dados [Ribeiro et al. 2020].

De forma a propor um serviço que ofereça tais funcionalidades descritas por [Ribeiro and Braghetto 2021], este artigo apresenta o `Hurricane`, um serviço configurável e extensível para gerência de dados no contexto de Cidades Inteligentes. O `Hurricane` permite (i) carregar dados heterogêneos de fontes externas, (ii) pré-processar os dados (*e.g.*, remover dados duplicados), (iii) realizar agregações prévias que facilitem a análise e visualização dos dados, uma vez que muitas aplicações de Cidades

Inteligentes realizam múltiplas agregações no espaço e no tempo e devem retornar resultados com o mínimo possível de latência, *e.g.*, abaixo de 0,5 segundos [Liu and Heer 2014], (iv) associar dados entre diferentes conjuntos de dados por meio de mapeamentos definidos pelo usuário, (v) capturar metadados, incluindo dados de proveniência, (vi) anonimizar os dados brutos e pré-processados e (vi) disponibilizá-los por meio de APIs para desenvolvedores/consumidores.

Diferentemente de outras abordagens de gerência de dados para Cidades Inteligentes, o `Hurricane` segue a abstração de *dataflows* [Silva et al. 2017, de Oliveira et al. 2019b] na gerência dos dados, uma vez que um *dataflow* representa uma evolução das transformações de dados e acompanha a propagação dos dados de domínio ao longo das aplicações em uma granularidade fina (*e.g.*, múltiplos arquivos de dados relacionados). Dessa forma, o `Hurricane` não armazena somente os dados já processados, mas também os dados intermediários e os dados brutos que foram carregados das fontes externas em um *Data Lake* [Nargesian et al. 2019]. Isso permite que o *dataflow* seja registrado e análises não consideradas inicialmente possam ser configuradas no serviço e disponibilizadas *a posteriori* a partir dos dados brutos armazenados. Além disso, o `Hurricane` considera questões de privacidade dos dados, já que dados sensíveis de indivíduos podem estar contidos nos dados obtidos em fontes externas (*e.g.*, dados pessoais em um boletim de ocorrência). O `Hurricane` foi avaliado em um estudo de caso com uma aplicação na área de segurança pública e os resultados se mostraram promissores.

O presente artigo se encontra organizado em 4 seções além da Introdução. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta os detalhes do `Hurricane`. A Seção 4 apresenta a avaliação do `Hurricane`, e, finalmente, a Seção 5 conclui esse artigo e apresenta trabalhos futuros.

2. Trabalhos Relacionados

Conforme mencionado anteriormente, existem diversos trabalhos na literatura que propõem abordagens para gerência de dados no contexto de Cidades Inteligentes. Cada uma das abordagens propostas cobre parcialmente a arquitetura de referência definida por [Ribeiro and Braghetto 2021]. A seguir são discutidas algumas dessas abordagens. [Liu et al. 2017] e [Jindal et al. 2020] propõem arcabouços para gerência de dados independente do domínio de aplicação. De fato, os arcabouços propostos consideram diversos dos requisitos elencados por [Ribeiro and Braghetto 2021] como ingestão dos dados, integração e consulta. Entretanto, tais arcabouços não consideram a importância da gerência de metadados durante o ciclo de vida do dado nem questões de privacidade. Ainda, os dois arcabouços não consideram o registro dos dados brutos ao longo do processo de integração, o que pode comprometer auditorias e análises *post-mortem*.

[Consoli et al. 2015] propõem um arcabouço para integração de dados de Cidades Inteligentes por meio do conceito de *Linked Open Data*. Os dados gerenciados pela abordagem de [Consoli et al. 2015] são representados como triplas RDF, e ontologias são associadas a cada *dataset* importado no arcabouço. Questões de ingestão e tratamento dos dados não são consideradas. Similarmente, [Silva et al. 2021] também propõem um arcabouço chamado Aquedücte para integração semântica de dados de Cidades Inteligentes. O Aquedücte fornece a possibilidade de carregar dados a partir de arquivos JSON ou de *Shapefiles*. Uma vantagem da proposta é que ela é baseada no protocolo NGSI-

LD (*Next Generation Service Interfaces - Linked Data*). Porém, além de requisitar o uso do protocolo NGSI-LD (que não é suportado por muitas aplicações), o Aquedücte ainda precisa converter os dados provenientes de fontes externas para o NGSI-LD, seguindo o modelo semântico definido, o que gera um *overhead* adicional.

[Costa and Santos 2017] propõem uma abordagem para implantar um *Data Warehouse* para Cidades Inteligentes, em conjunto com um *storage* que armazena os dados de entrada em seu formato bruto. A abordagem de [Costa and Santos 2017] se baseia em ferramentas e bibliotecas conhecidas como o Talend e o HDFS (*Hadoop Distributed File System*). O *Data Warehouse* gerado pode ser consultado utilizando SQL via Presto. A abordagem proposta por [Mehmood et al. 2019] é bastante similar a de [Costa and Santos 2017], com a diferença que ela disponibiliza uma interface de consulta e visualização própria. Entretanto, tanto a abordagem de [Costa and Santos 2017] quanto a de [Mehmood et al. 2019] não oferecem dados de proveniência e nem capturam outros metadados, além de não oferecer questões de anonimização de dados.

[Garcia-Font 2020] propõe uma arquitetura para gerência de dados centrada no usuário para comunicações no contexto de Cidades Inteligentes. A proposta de [Garcia-Font 2020] define que os dados sejam gerenciados de forma descentralizada para reduzir a dependência de provedores de serviços, entretanto seu foco se dá apenas para aplicações de comunicação. [Zhou et al. 2021] propõem uma arquitetura para gerência de dados de Cidades Inteligentes na área da saúde. A abordagem proposta por [Zhou et al. 2021] considera apenas dados de prontuário e de exames, não sendo extensível para outro domínios. Similarmente, [Bellini et al. 2021] e [Nandury and Begum 2016] também focam em um domínio específico, *i.e.*, transporte público, para propor seu arcabouço. A diferença é que o trabalho de [Nandury and Begum 2016] tem um foco na transferência e carga de dados, não oferecendo soluções para integração de dados e nem metadados. A Tabela 1 sumariza as características dos trabalhos relacionados.

3. O `Hurricane`

O `Hurricane` é um serviço configurável e extensível para gerência de dados para aplicações de Cidades Inteligentes. A Figura 1 apresenta a arquitetura do `Hurricane` (os componentes em cinza são os desenvolvidos pelos autores deste artigo). Conforme mencionado na Seção 1, o `Hurricane` segue a abstração de *dataflow* na gerência e processamento dos dados, *i.e.*, todo o processamento é realizado a partir da instanciação de múltiplos *dataflows*, onde cada etapa do processo é monitorada e tem seus dados e metadados registrados. Cada *dataflow* tem um objetivo específico e é instanciado de forma automática no *Apache Airflow* (`https://airflow.apache.org`) a partir de configurações definidas previamente pelo usuário. O *Airflow* fornece uma série de facilidades como processamento paralelo e distribuído que podem ser utilizadas dependendo do volume de dados a ser processado pelo `Hurricane`.

O dado no `Hurricane` possui um ciclo de vida bem definido, que se inicia com a ingestão dos dados a partir de fontes externas. O processo de ingestão não foi o foco principal desta pesquisa, visto que já existem inúmeras soluções capazes de conectar e extrair dados de uma origem para um destino. O `Hurricane` é capaz de importar dados de diferentes fontes de dados, sejam elas estruturadas ou não. O requisito é que exista

Tabela 1. Comparação dos Trabalhos Relacionados ($\surd$ = Suportado, $\times$ = Não Suportado, $\pm$ = Suportado Parcialmente, $\star$ = Dados não informados).

Abordagens	Características											
	Data Warehouse	*Data Lake*	Apoio Semântico	Proveniência	Anonimização	Específico de Domínio	Orientado a *Dataflow*	Permite execução distribuída	Apoio à Integração	Suporte a Carga	*Open Source*	Prático (não conceitual)
[Consoli et al. 2015]	$\star$	$\star$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\pm$	$\times$	$\surd$	$\surd$
[Nandury and Begum 2016]	$\star$	$\pm$	$\times$	$\times$	$\times$	$\surd$	$\times$	$\star$	$\surd$	$\surd$	$\surd$	$\surd$
[Costa and Santos 2017]	$\surd$	$\surd$	$\times$	$\times$	$\surd$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
[Liu et al. 2017]	$\surd$	$\pm$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
[Mehmood et al. 2019]	$\star$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
[Garcia-Font 2020]	$\star$	$\pm$	$\times$	$\times$	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
[Jindal et al. 2020]	$\surd$	$\pm$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
[Bellini et al. 2021]	$\pm$	$\surd$	$\surd$	$\times$	$\times$	$\surd$	$\times$	$\star$	$\pm$	$\surd$	$\surd$	$\surd$
[Ribeiro and Braghetto 2021]	$\pm$	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$
[Silva et al. 2021]	$\pm$	$\pm$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\pm$	$\surd$	$\surd$	$\surd$	$\surd$
[Zhou et al. 2021]	$\star$	$\pm$	$\surd$	$\times$	$\times$	$\surd$	$\times$	$\star$	$\surd$	$\surd$	$\surd$	$\surd$
`Hurricane`	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

uma API que possa ser consultada para acessar os dados. A única exceção desse requisito se dá na importação dos dados espaciais de cidades (uma vez que praticamente todas as aplicações de Cidades Inteligentes necessitam de tais dados). Os dados espaciais e as topologias das cidades são importadas a partir dos dados do *Open Street Map* (OSM) (`https://www.openstreetmap.org`), *i.e.*, projeto de mapeamento colaborativo para criar um mapa editável do planeta, e um componente específico de ETL (*Extract, Transform, Load*) foi desenvolvido para esse fim (Passo 1 na Figura 1), chamado *ETL[OSM]*. É importante ressaltar que o *ETL[OSM]* estende a biblioteca OSMnx [Boeing 2017], que permite modelar redes de ruas de cidades e quaisquer outras geometrias geoespaciais disponíveis do OSM. O componente realiza o *download* dessas informações em formato JSON e as armazena no *Data Lake* do `Hurricane` (Passo 2). A partir do JSON carregado, é criado um grafo que representa o mapa da cidade. Esse grafo é representado em dois *dataframes*, contendo os nós e arestas. As arestas são particionadas em segmentos de aproximadamente 100 metros de distância. Além do grafo, metadados das vias também são obtidos, *e.g.*, *oneway*, que indica se a via é apenas de mão única e *highway* define o tipo da Rua/Rodovia.

No componente *ETL[OSM]*, um *dataflow* é criado dinamicamente para extrair as informações necessárias do OSM para criação da rede de ruas e a topologia da cidade escolhida. O *ETL[OSM]* instancia o *dataflow* a partir de um arquivo de configuração similar ao fragmento apresentado na Listagem 1. A *tag workflow_type* determina o tipo do *dataflow* que deve ser criado dinamicamente, nesse caso, "model", pois gera um modelo da cidade como saída. A *tag datalake_client* define os conectores para o *storage*, no caso *hdfs*, o sistema de arquivos distribuído do *stack* Hadoop. A *tag datalake_workdir* define o diretório de trabalho onde serão gerados todos os dados no *Data Lake*. A tag *retries* define o número de tentativas que deve ser considerado em caso de erro, enquanto que *owner* representa o identificador do usuário responsável pela carga. A tag *metadata_url, schema*

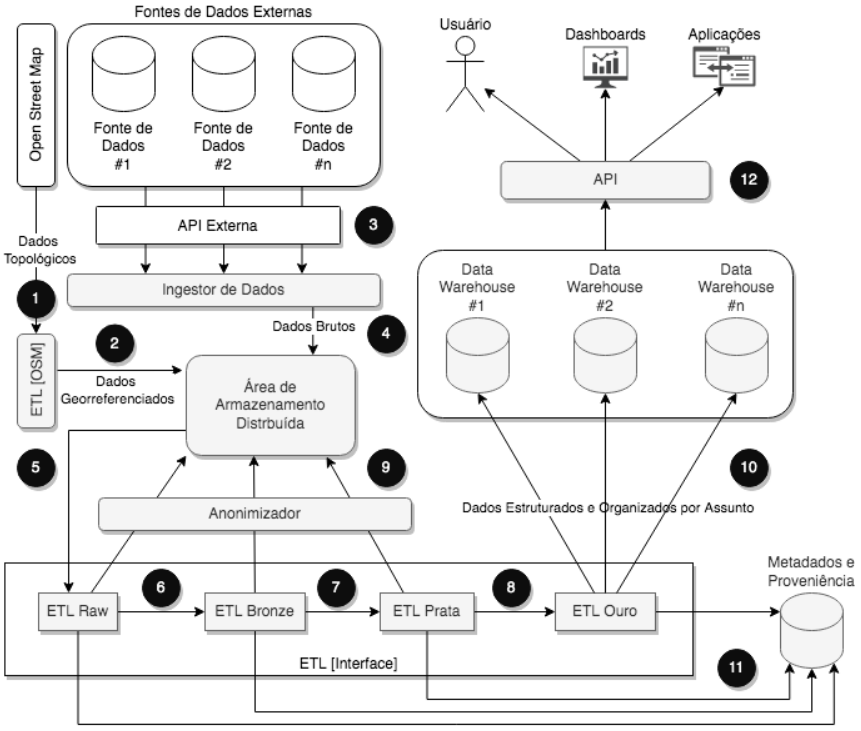


Figura 1. Arquitetura do `Hurricane`.

```
    "workflow_type"     : "model",
    "dag_id"            : "Generate-City-Model",
    "datalake_client"   : "hdfs",
    "datalake_workdir"  : "/datalake/model/SP",
    "retries"           : 5,
    "owner"             : "Anonimizado",
    "metadata_url"      : "postgresql://localhost:5432/cdbase",
    "schema"            : "public",
    "tablespace"        : "pg_default",
    "places"            : [
        {
            "city"      : "Sao Paulo",
            "state"     : "Sao Paulo",
            "country"   : "Brazil"
        }
    ]
```

Listing 1. Fragmento do *Template* de Configuração do *ETL[OSM]*.

e *tablespace* são os parâmetros de conexão com o banco de metadados. Finalmente, a cidade que deve ser considerada para gerar o modelo de redes rodoviárias é definida por meio das tags *city, state* e *country* na *tag places*. Os demais dados de fontes externas são acessados por suas respectivas APIs (Passo 3) pelo componente *Ingestor de Dados* (Passo 4). Esse componente recebe os dados das APIs e os carrega no *Data Lake* na área específica de cada tipo de dado no *storage* distribuído.

Uma vez que os dados brutos e os dados espaciais das cidades já se encontram no *Data Lake*, os *dataflows* de processamento de dados podem ser iniciados (Passo 5). O processamento dos dados no `Hurricane` foi organizado em quatro etapas: (i) *Raw*, (ii) Bronze, (iii) Prata, e (iv) Ouro. Em cada etapa uma série de transformações são executadas nos dados por *dataflows* específicos. Esses *dataflows* são instanciados a partir do arquivo de configuração similar ao fragmento apresentado na Listagem 2. A *tag workflow_type* informa o tipo do *dataflow* que deve ser criado dinamicamente, neste caso, "ge-

```
{
   "workflow_type"     : "general",
   "dag_id"            : "Hurricane-PMERJ-Crimes",
   "related_dag_id"    : "Generate-City-Model",
   "datalake_client"   : "local",
   "datalake_workdir"  : "/datalake/model/SP",
   "schedule_interval" : "0 08 * * 0-6",
   "max_concurrency"   : 4,
   "retries"           : 5,
   "owner"             : "Anonimizado",
   "fact_tablename"    : "crime",
   "aggregation"       : "count",
   "raw_interfaces"    : [
       {
           "name": "vehicles_rob",
           "input_path": "/raw/vehicles_rob/",
           "header": 0,
           "rules_columns": {
               "date"       : "DATA_OCORRENCIA" ,
               "period"     : "PERIODO_OCORRENCIA",
               "latitude"   : "LATITUDE",
               "longitude"  : "LONGITUDE"
           },
           "duplicated_key" : ["ANO_BO", "NUM_BO", "NUMERO_BOLETIM"],
           "att_dimension": [{"name":"TIPO_CRIME", "type":"integer"},...]
       } ...
```

Listing 2. Fragmento do Template de Configuração do ETL *Raw*, ETL Bronze, ETL Prata e ETL Ouro.

neral". A *dag_id* representa o identificador único do *dataflow*. A *tag related_dag_id* representa o *dataflow* responsável por construir o grafo de ruas da cidade e que será integrado aos demais dados baixados. A *tag schedule_interval* é uma configuração usada pelo *scheduler* do *Airflow* para monitorar todas as tarefas dos *dataflows*. A *tag max_concurrency* define a quantidade de *threads* que podem ser executadas em paralelo, respeitando as suas dependências de dados. O subgrupo *raw_interfaces*, define interfaces que realizam o mapeamento dos dados contidos nos arquivos brutos do *Data Lake* com os dados espaciais das cidades obtidos no OSM. Cada interface tem um identificador único, chamado *name*, um diretório onde se encontram os arquivos brutos no *Data Lake* representado na *tag input_path*. Além disso, a *tag rules_columns* define o mapeamento dos dados do arquivo de entrada com os dados extraídos do OSM. Esse mapeamento é realizado por meio das variáveis globais *date*, *period*, *latitude* e *longitude*, que são mapeadas para os atributos contidos no arquivo de dados brutos. No fragmento apresentado na Listagem 2, *date* é mapeado para o atributo `DATA_OCORRENCIA`, *period* para `PERIODO_OCORRENCIA` e *latitude* e *longitude* para atributos de mesmo nome. Esse mapeamento também pode ser realizado para outros tipos de dados que tenham sido importados. Na *tag feature_columns* são definidas características adicionais que deseja-se considerar, *e.g., duplicated_key* que define a chave que deve ser considerada para descartar registros duplicados.

Uma vez que os parâmetros de entrada se encontram definidos, o `Hurricane` inicia a instanciação de cada *dataflow* de processamento de dados. O *dataflow* associado ao componente ETL *Raw* tem como responsabilidade identificar registros duplicados (de acordo com as configurações informadas no arquivo de configuração) e definir uma chave para os dados brutos. Após, o *dataflow* do ETL Bronze é instanciado (Passo 6), e realiza sumarizações e agregações nos dados. Por padrão, essas agregações são realizadas de acordo com valores de latitude/longitude e data (*i.e.*, agregações no espaço e no tempo). É importante ressaltar que o tipo de agregação deve ser configurado na interface (na Listagem 2 a função de `COUNT` foi a definida), e que o usuário é capaz de definir outras dimensões para agregação no arquivo de configuração (*e.g.*, no campo `ATT_DIMENSION`).

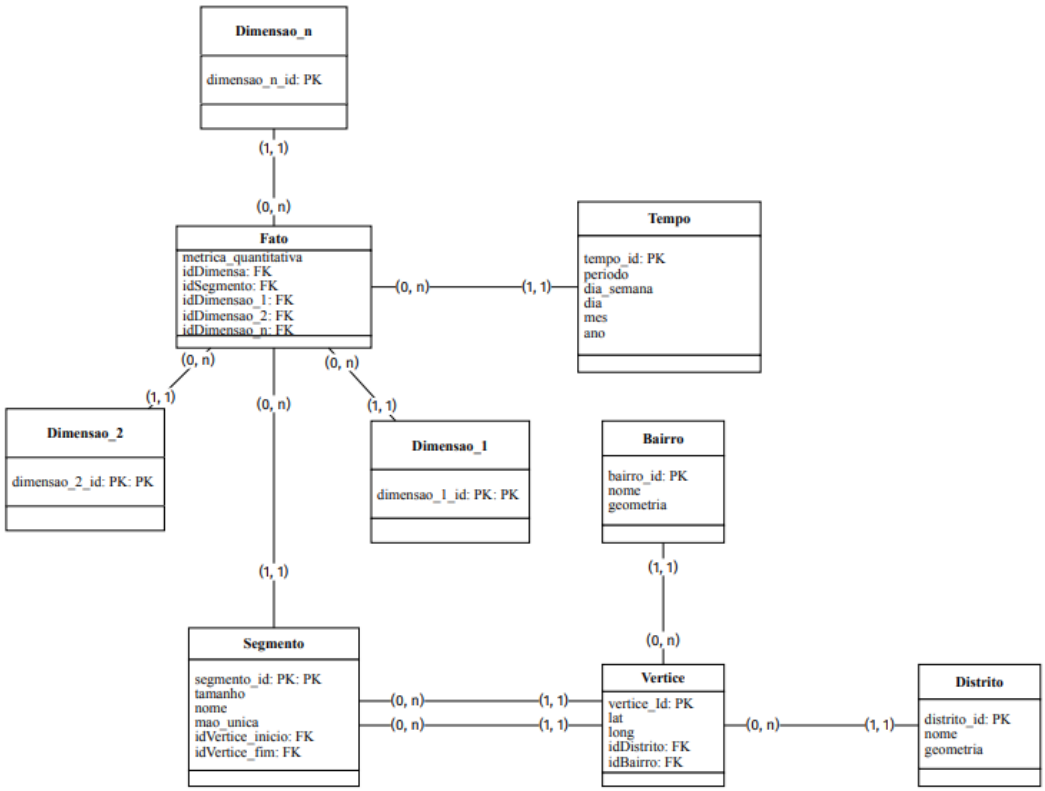


Figura 2. *Template* de Criação do *Data Warehouse no* `Hurricane`.

Assim que as agregações são finalizadas, o *dataflow* do ETL Prata é instanciado (Passo 7). O ETL Prata associa os dados agregados pelo ETL Bronze com os dados extraídos do OSM. É importante ressaltar que nem sempre a latitude e longitude informadas nos dados agregados representam um ponto em um dos segmentos de rua definidos pelo OSM. Assim, o *dataflow* do ETL Prata identifica qual segmento de rua é o mais próximo ao ponto em questão. Finalmente, é instanciado o *dataflow* do ETL Ouro (Passo 8), que cria um *Data Warehouse* (DW) para cada tipo de dado de Cidades Inteligentes. Por padrão são criadas uma tabela fato e tabelas de dimensão relativas à `LOCALIDADE` e ao `TEMPO`, para representar as componentes espacial e temporal. Entretanto, caso outros atributos tenham sido informados em `ATT_DIMENSION`, novas tabelas de dimensão são criadas de acordo com o domínio desses atributos. A Figura 2 apresenta o *template* de criação do *Data Warehouse* com a tabela fato, a dimensão tempo e as tabelas *Segmento*, *Vertice*, *Bairro* e *Distrito* para representar a localidade. As tabelas *Dimensao_1*, *Dimensao_2* e *Dimensao_n* representam as possíveis dimensões que podem ser criadas pelo `Hurricane`.

É importante ressaltar que durante o processamento do componente ETL Ouro, todas as integrações entre os dados carregados de fontes externas e os dados geográficos já foram previamente efetuadas pelo componente ETL Prata. Logo, a única responsabilidade do ETL Ouro é consumir os dados no *Data Lake* para gerar a visão final dos dados que serão consumidos pelo usuário. Os dados intermediários produzidos por cada *dataflow* são armazenados no *Data Lake*, de forma que possam ser utilizados no futuro (Passo 9) e em seguida sincronizados por meio de um *full overwrite* no *Data Warehouse*

modelado no PostgreSQL (Passo 10). Esses dados opcionalmente podem sofrer um processo de anonimização tanto por pseudonimização quanto usando a técnica de privacidade diferencial [Dwork and Lei 2009, de Oliveira et al. 2019a]. Além disso, todos os metadados de proveniência são carregados em um banco de dados específico (Passo 11) que contém o histórico de todas as transformações realizadas nos dados. Finalmente, o usuário final ou uma aplicação podem consumir os dados integrados e agregados via API do `Hurricane` (Passo 12). O código-fonte do `Hurricane` se encontra disponível em `https://github.com/UFFeScience/Hurricane`.

4. Avaliação do `Hurricane`

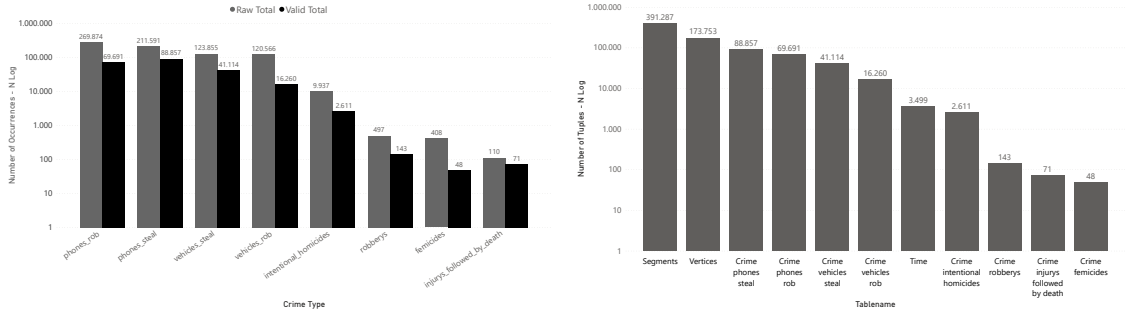
Nessa seção apresentamos a avaliação do `Hurricane` com um estudo de viabilidade, com o objetivo de avaliar o seu uso efetivo e seu desempenho. Para tanto, a seguinte questão de pesquisa foi definida: (QP1) "Os dados integrados entregues pelo `Hurricane` apoiam os usuários e desenvolvedores de aplicações de Cidades Inteligentes?" Como domínio de aplicação foi escolhida a área de Segurança Pública com foco em Policiamento Preditivo. A ideia do policiamento preditivo é tornar o trabalho da força policial evidente para a população, seja pela presença de oficiais de polícia em pontos estratégicos da cidade ou com patrulhas policiais ostensivas. No contexto do policiamento preditivo, existem tarefas bastante complexas de serem executadas como a análise de manchas criminais e a definição de rotas de patrulhamento policial. Tanto a análise de manchas criminais quanto a geração de rotas precisam receber como entrada os dados dos chamados *Hot Spots*, *i.e.*, áreas da cidade onde o índice de criminalidade é alto.

Para ilustrar a ocorrência de *Hot Spots*, tomemos duas regiões da cidade de São Paulo: Alto da Mooca e Itaim Paulista. A Figura 3 apresenta um mapa de calor sobre as ruas para apresentar as áreas de *Hot Spots*, onde a escala varia de amarelo a vermelho intenso, quanto mais escura maior é o índice de criminalidade. Na Figura 3(a), a maioria das ruas do Alto da Mooca não apresenta nenhuma ocorrência de crime para o período selecionado (Novembro/2017). Por outro lado, na Figura 3(b), a maioria das ruas da região do Itaim Paulista apresenta altos índices de criminalidade para o mesmo período, caracterizando vários focos de criminalidade. Para se identificar os *Hot Spots*, os dados de ocorrências de crimes devem se encontrar agregados por segmento de via no mapa e no tempo. Essa é uma tarefa não trivial de ser desempenhada manualmente, e para isso foi usado o `Hurricane` para realizar a gerência e integração desses dados.

Para o estudo, foram carregados no `Hurricane` os dados de crimes disponibilizados no portal da SSP-SP da cidade de São Paulo no ano de 2019. É importante ressaltar que nem todas as ocorrências de todos os tipos de crimes que podem ser obtidas no portal foram consideradas. Todavia, nesse estudo de viabilidade, foram selecionados oito tipos de crimes, *i.e.*, feminicídio, roubo de veículo, furto de veículo, roubo de celular, furto de celular, latrocínio e homicídio. Além disso, existem ocorrências que não apresentam informações de localização (*i.e.*, latitude e longitude), de forma que possibilite a associação da ocorrência do crime a um segmento do mapa. Assim, a Figura 4(a) apresenta o número total de ocorrências nos dados brutos (em cinza) e o número total de ocorrências validadas após o processamento pelos *dataflows* do `Hurricane` (em preto) e a Figura 4(b) apresenta o total de tuplas para cada tipo de dado que se encontra no *Data Warehouse* (*e.g.*, segmentos, vértices, ocorrências de crime, etc).

(a) Região do Alto da Mooca. **(b) Região do Itaim Paulista.**

Figura 3. Ocorrências de Crimes no Alto da Mooca e Itaim Paulista em Nov/2017.



(a) Total de Ocorrências X Total de Ocorrências Válidas. **(b) Total de Tuplas.**

Figura 4. Estatísticas do Estudo de Viabilidade.

Os participantes do estudo foram selecionados de acordo com sua área de atuação, *i.e.*, especialistas na área de segurança pública. Ao todo, foram selecionados seis participantes especialistas. A ideia é que os participantes avaliassem os dados disponibilizados pelo `Hurricane` tanto para apoiar a análise de manchas criminais quanto para apoiar a geração de rotas policiais. A partir das respostas, pode-se identificar que todos os participantes tem grau de instrução de graduação. A formação dos especialistas abrange as áreas de: Matemática, Estatística e Ciências Sociais. Entre os especialistas, foi identificado que 50% atuam há mais de 10 anos na área e 50% entre 5 e 10 anos.

Na sequência, os especialistas foram treinados para análise dos dados. O treinamento respeitou o mesmo roteiro para todos os participantes, evitando a construção de vieses. Após o uso do `Hurricane`, foi disponibilizado o questionário para a avaliação do serviço. Ao questionar sobre o apoio ao usuário na análise dos dados criminais, em uma escala de 1 a 5 (sendo 1 = pouco e 5 = muito), 66,6% dos especialistas responderam 5, 16,7% responderam 4 e 16,7% responderam 2. Isso mostra que a solução é reconhecida pela maioria dos especialistas como ferramenta que apoia o usuário especialista na integração e gerência dos dados para aplicações de segurança. O usuário que respondeu 2 afirmou que por não ser da área tecnológica não se sentiu confiante no uso do `Hurricane`. Todos os usuários ressaltaram que seria interessante o desenvolvimento de um *dashboard* integrado ao `Hurricane` para visualização parcial dos dados durante o processamento (*i.e., human-in-the-loop* para análise de dados).

Nesse estudo de viabilidade, foram identificadas algumas ameaças à validade. Em primeiro lugar o controle de atualização dos dados. Atualmente não é realizado nenhum controle sobre atualização de dados, *i.e.*, o `Hurricane` não é proativo em sua atualização, dependendo do usuário executá-lo ou agendar sua execução. Além disso, não foi analisada a capacidade de suportar grande quantidade de acessos simultâneos no `Hurricane`. Entretanto, o serviço foi projetado usando ferramentas e bibliotecas (*e.g.*, AirFlow e OSMnx) que permitem a escalabilidade do mesmo.

Além da avaliação qualitativa, foi realizada uma avaliação do desempenho do `Hurricane`. A ideia foi avaliar o tempo necessário para que o serviço executasse cada uma das etapas mencionadas na Seção 3. A Tabela 2 apresenta o tempo médio de processamento (*i.e.*, $\overline{x}$) e o desvio padrão (*i.e.*, σ) de 5 execuções de cada *dataflow* do `Hurricane`, em segundos. É importante ressaltar que esses *dataflows* podem ser executados em paralelo em ambientes distribuídos como *clusters* de máquinas *commodity*, em alguns casos, logo não é correto contabilizar o tempo gasto, como uma simples sumarização de todo o conjunto de tempos. Analisando a Tabela 2 podemos observar que mesmo considerando um *dataset* com apenas um ano de dados (*i.e.*, ano de 2019), os maiores tempos de execução são nas etapas de processamento dos dados da cidade e na etapa de integração e sumarização dos dados. Em especial, o processamento dos dados da cidade, além de receber uma quantidade massiva de segmentos de vias que devem ser processados, ainda precisa obter de fontes externas os metadados de bairros e zonas da cidade, que não são disponibilizados automaticamente pelo OSMnx. Entretanto, tais tempos se mostram aceitáveis considerando o volume de dados envolvido, *i.e.*, cerca de 49 minutos para geração do grafo de vias da cidade (que não precisa ser atualizado sempre) e 32 minutos para integração e sumarização dos dados do ano de 2019.

Tabela 2. Avaliação do Desempenho do `Hurricane` (tempos em segundos).

Dataflow	**Tipo**	$\overline{x}$	σ
ETL[OSM]	Modelo	2.920,53	193,74
ETL Raw	Geral	122,12	3,03
ETL Bronze	Geral	396,22	26,44
ETL Prata	Geral	1.917,01	63,32
ETL Ouro	Geral	77,50	4,45

5. Conclusão

Este artigo apresentou um serviço para gerência e integração de dados no contexto de Cidades Inteligentes chamado `Hurricane`. O objetivo do `Hurricane` é ser configurável e extensível de acordo com a necessidade do usuário analista ou desenvolvedor. O `Hurricane` é capaz de importar dados de múltiplas fontes e integrá-los de acordo com mapeamento realizado pelos usuários. O trabalho apresentado nesse artigo é derivado de uma pesquisa aplicada em projeto multidisciplinar que inclui tanto especialistas em segurança pública da Polícia Militar do Rio de Janeiro quanto especialistas em computação da Universidade Federal Fluminense (UFF).

Um estudo de viabilidade foi conduzido no domínio de segurança pública com os especialistas para analisar se o `Hurricane` de fato oferece apoio na captura, armazenamento, agregação e consulta para aplicações de análise de manchas criminais e geração de rotas policiais. Os participantes do estudo de viabilidade afirmaram a importância de

serviços como o `Hurricane`, que são capazes de obter dados de diversas fontes, tratar os dados e disponibilizá-los de maneira eficiente e ágil. Apesar da disponibilização do `Hurricane` representar um passo importante, os usuários solicitaram a inclusão de novas funcionalidades. Assim, como trabalhos futuros, são planejadas a integração de um *dashboard* no serviço para visualização das informações geradas, e a inclusão de apoio semântico, para que o `Hurricane` seja capaz de identificar novas relações nos dados e conhecimento implícito por meio de inferências. Adicionalmente, avaliações com outros domínios de aplicação também vem sendo planejadas, em especial na análise de dados pluviométricos de grandes centros urbanos. Por fim, outros tipos de mapeamento são possível, porém não foram explorados. Novos formatos serão adicionados para interpretação de arquivos, como conexões *JDBC*, leitura de arquivos *JSON*, *PARQUET* e *ORC*, juntamente com o uso do *Spark* como *framework* de processamento distribuído.

Referências

Bellini, E., Bellini, P., Cenni, D., Nesi, P., Pantaleo, G., Paoli, I., and Paolucci, M. (2021). An ioe and big multimedia data approach for urban transport system resilience management in smart cities. *Sensors*, 21:435.

Bilal, M., Usmani, R. S. A., Tayyab, M., Mahmoud, A. A., Abdalla, R. M., Marjani, M., Pillai, T. R., and Targio Hashem, I. A. (2020). *Smart Cities Data: Framework, Applications, and Challenges*, pages 1–29. Springer International Publishing, Cham.

Boeing, G. (2017). Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Comp., Env. and Urban Sys.*, 65:126–139.

Caban, J. J. and Gotz, D. (2015). Visual analytics in healthcare – opportunities and research challenges. *J. of the American Med. Inf. Assoc.*, 22:260–262.

Chen, H., Cheng, T., and Wise, S. (2017). Developing an online cooperative police patrol routing strategy. *Computers, Environment and Urban Systems*, 62:19–29.

Consoli, S., Mongiovì, M., Nuzzolese, A. G., Peroni, S., Presutti, V., Recupero, D. R., and Spampinato, D. (2015). A smart city data model based on semantics best practice and principles. In *WWW 2015*, pages 1395–1400. ACM.

Costa, C. and Santos, M. Y. (2017). The suscity big data warehousing approach for smart cities. IDEAS 2017, page 264–273, New York, NY, USA. ACM.

de Oliveira, D., Rodrigues, E., Costa, S., Amora, P. R. P., Caldas, A., Horta, M., de Fillippis, A. M., Ocaña, K. A. C. S., Vidal, V. M. P., and Machado, J. C. (2019a). Um estudo comparativo de mecanismos de privacidade diferencial sobre um dataset de ocorrências do ZIKV no brasil. In *XXXIV Simpósio Brasileiro de Banco de Dados, SBBD 2019, Fortaleza, CE, Brazil, October 7-10, 2019*, pages 253–258. SBC.

de Oliveira, D. C. M., Liu, J., and Pacitti, E. (2019b). *Data-Intensive Workflow Management: For Clouds and Data-Intensive and Scalable Computing Environments*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

de Oliveira, W. M., de Oliveira, D., and Braganholo, V. (2018). Provenance analytics for workflow-based computational experiments: A survey. *ACM Comput. Surv.*, 51(3):53:1–53:25.

Dwork, C. and Lei, J. (2009). Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380.

Freire, J., Koop, D., Santos, E., and Silva, C. T. (2008). Provenance for Computational Tasks: A Survey. *Computing in Science & Engineering*, pages 20–30.

Garcia-Font, V. (2020). Socialblock: An architecture for decentralized user-centric data management applications for communications in smart cities. *JPDC*, 145:13–23.

Jindal, A., Kumar, N., and Singh, M. (2020). A unified framework for big data acquisition, storage, and analytics for demand response management in smart cities. *FGCS*, 108:921–934.

Liu, X., Heller, A., and Nielsen, P. S. (2017). Citiesdata: a smart city data management framework. *Knowl. Inf. Syst.*, 53:699–722.

Liu, Z. and Heer, J. (2014). The effects of interactive latency on exploratory visual analysis. *IEEE transactions on visualization and computer graphics*, 20:2122–2131.

Lourenço, V., Mann, P., Guimaraes, A., Paes, A., and de Oliveira, D. (2018). Towards safer (smart) cities: Discovering urban crime patterns using logic-based relational machine learning. In *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pages 1–8. IEEE.

Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., Tekes, S., and Riekki, J. (2019). Implementing big data lake for heterogeneous data sources. In *ICDEW 2019*, pages 37–44.

Nandury, S. V. and Begum, B. A. (2016). Strategies to handle big data for traffic management in smart cities. In *ICACCI 2016, India*, pages 356–364. IEEE.

Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q., and Arocena, P. C. (2019). Data lake management: Challenges and opportunities. *Proc. VLDB Endow.*, 12:1986–1989.

Pisco, V. G. and Marques-Neto, H. T. (2021). iwalk: Uma solução para medição e análise da caminhabilidade de cidades com portais de dados abertos. In *Anais do V Workshop de Computação Urbana*, pages 84–97. SBC.

Raghavan, S., Boung Yew, S. L., Lee, Y. L., Tan, W., and Kee, K. K. (2019). *Data Integration for Smart Cities: Opportunities and Challenges*, pages 393–403.

Ribeiro, M. B. and Braghetto, K. R. (2021). A data integration architecture for smart cities. In *SBBD 2021, Rio de Janeiro, Brazil*, pages 205–216. SBC.

Ribeiro, M. W. M., Lima, A. A. B., and de Oliveira, D. (2020). OLAP parallel query processing in clouds with c-pargres. *Concurr. Comput. Pract. Exp.*, 32(7).

Silva, J., Almeida, J. G., Batista, T., and Cavalcante, E. (2021). Aquedücte: A data integration service for smart cities. WebMedia '21, page 177–180, NY, USA. ACM.

Silva, V., Leite, J., Camata, J. J., de Oliveira, D., Coutinho, A. L. G. A., Valduriez, P., and Mattoso, M. (2017). Raw data queries during data-intensive parallel workflow execution. *FGCS*, 75:402–422.

Zhou, R., Zhang, X., Wang, X., Yang, G., Guizani, N., and Du, X. (2021). Efficient and traceable patient health data search system for hospital management in smart cities. *IEEE Internet Things J.*, 8(8):6425–6436.