

CIDADES INTELIGENTES - CIÊNCIA DE DADOS APLICADOS A PREVISÃO DE ALAGAMENTOS PARA PREFEITURA DE SANTO ANDRÉ E CONSÓRCIO

Gabriel Marques Rangel da Silva ¹; Tiago Sanches da Silva ²

¹ Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

² Professor da Escola de Engenharia Mauá (EEM/CEUN-IMT).

Resumo. *Atualmente a região metropolitana de São Paulo apresenta problemas com alagamentos, e o monitoramento feito, especificamente no município de Santo André, através de estações meteorológicas, gera dados que podem ser utilizados em estudos para previsão de alagamentos a fim de criar medidas para amenizar os danos causados. Porém, a obtenção dos dados ocorre de forma burocrática, através de pedidos utilizando formulários e exportações em planilha Excel, podendo atrasar grupos de pesquisa e empresas contratadas para realizar trabalhos que precisam utilizar esses dados. Através da criação de uma API, relatórios de estações meteorológicas podem ser enviadas para ser salvas dentro do Banco de Dados de um servidor. O mesmo pode ser disponibilizado de forma simples através de um endereço virtual, ocorrendo uma agilização na liberação e atualização dos dados. Existindo um Ecossistema com alta disponibilidade de dados, a aplicação de técnicas e Inteligência Artificial e Ciência de Dados é facilitada e serve como catalizador de inovação tecnológica para o município, sendo assim foi realizado um estudo especificação de uma proposta server-side de API REST com a capacidade de receber dados dos sensores e estações em tempo real, bem como fornecer acesso a esses dados de forma eficiente e com menos burocracia.*

Introdução

No Brasil, mais de 80% dos brasileiros vivem em áreas urbanas e o acelerado crescimento urbano tem criado espaços fragmentados com ampla segregação espacial, agravando a desigualdade social e a degradação ambiental (VALVERDE et al., 2018). Projeções indicam que, caso o padrão de expansão da Região Metropolitana de São Paulo seja mantido conforme registros históricos, a mancha urbana será aproximadamente 38% maior que a atual em 2030, aumentando os riscos de inundações e deslizamentos, atingindo cada vez mais a população, sobretudo, os mais pobres (NOBRE et al., 2011). Para que se tenha um panorama dos alagamentos, é possível aplicar o conceito de Cidades Inteligentes.

Embora não tenha uma definição consensual e amplamente aceita, Cidades Inteligentes tem como objetivo final promover o melhor uso de recursos públicos, aumentando a qualidade dos serviços oferecidos aos cidadãos, enquanto reduz os custos operacionais da administração. Uma aliada no desenvolvimento de cidades inteligentes é a Internet das Coisas urbanas (IoT). As IOTs urbanas são projetadas para apoiar a visão *Smart City*, que visa explorar as mais avançadas tecnologias de comunicação para apoiar serviços de valor agregado para a administração da cidade e para os cidadãos (ZANELLA et al., 2014).

Hoje, é possível observar a dificuldade para que grupos de pesquisas tenham acesso a dados públicos de maneira rápida e eficiente, o que é agravado quando o acesso é a informações de sensores e outros dispositivos IOTs da cidade, isso dificulta pesquisa e projetos que podem beneficiar a população. Um dos casos na prefeitura de Santo André é em relação aos relatórios de estações meteorológicas.

Com estações meteorológicas disponibilizados pela prefeitura de Santo André, foi possível especificar e criar uma API (Interface de Programação de Aplicações em português) de acesso a base de dados do servidor de IoT, esta API fornece suporte à envio de dados por parte dos dispositivos e possibilita a aquisição dos dados por meio de requisições HTTPs respeitando as regras do REST

("Representational State Transfer"), dessa forma pessoas e grupos de pesquisas poderão ter acesso fácil às informações contidas nos dados das estações meteorológicas.

Tendo isso em vista, este trabalho tem como objetivo criar uma API para facilitar o acesso aos dados, fazendo que o desenvolvimento de projetos seja mais eficaz. Além de mostrar conceitos de ciência de dados e modelos de aprendizado de máquina aplicáveis em previsão alagamentos.

Material e Métodos

Para a construção da API (Application Programming Interface), um conjunto de padrões que permite a construção de aplicativos, onde ele conecta aplicações, podendo ser utilizada nos mais variados tipos de negócios. (GUEDES, 2019) Foi utilizado a linguagem de programação Python e baseado na arquitetura REST ("Representational State Transfer"), desenvolvido por Roy Fielding em 2000. A arquitetura REST é definido pelas seguintes restrições:

- Protocolo cliente/servidor sem estado, isso é tanto o servidor quanto o cliente não precisam armazenar estado da comunicação entre mensagens, todo pedido feito via HTTP (Protocolo de Transferência por Hipertexto) já contém a informação necessária.
- Interface uniforme: uso de conjunto de operações bem definidas que se aplicam a todos os recursos.
- Sintaxe universal para identificar recurso, no caso do REST se utiliza a URI (Identificador de recurso uniforme).
- Uso de hipermídia, como, por exemplo, o HTML (Linguagem de Marcação de Hipertexto) e JSON (Notação de objeto de Javascript), tanto para a informação da aplicação quanto as transições de estado da aplicação.
- Uso de cache em respostas especificadas de requisições, fazendo processos de requisição mais eficientes e a API mais escalável.
- Sistema em camadas onde Dados, API e Autenticação são separados em diferentes servidores ao fazer *deploy* dos mesmos.

Essa arquitetura foi escolhida devido à sua compatibilidade com qualquer outra arquitetura que faz requisição HTTP (*HyperText Transfer Protocol*), é de fácil manutenção, provem escalabilidade, integra melhor plataformas distintas e é segura.

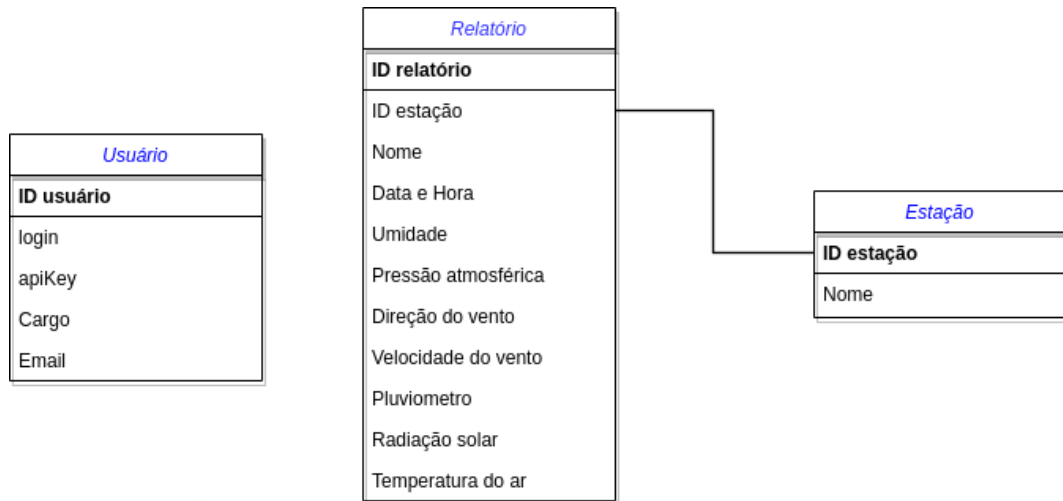
Para desenvolver a API foi utilizado as seguintes bibliotecas:

- Flask
Uma biblioteca Flask que é um *framework*, um conjunto de códigos prontos para desenvolvimento de forma simplificada, eficiente e de qualidade. Com ela é possível fazer facilmente criação de rotas, ou *endpoints*, e criação de recursos para a API na arquitetura REST.
- SQLAlchemy
O pacote SQLAlchemy é também um *framework*, mas de mapeamento objeto-relacional (ORM), uma técnica para aproximar o paradigma de desenvolvimento de aplicações orientadas a objetos ao paradigma do banco de dados relacional (FONSECA, 2020). De modo simples, ela faz a tradução do armazenamento e criação de objetos de um programa orientado a objeto para o banco de dados e vice-versa.

Desenvolvimento da API

Para versionamento do projeto foi utilizado a ferramenta git e a plataforma Github, para adicionar de forma gradual novos recursos. Foi diagramado quais modelos e respectivos atributos seriam necessários implementar na API que servirá de interface para aplicativos que usuários desenvolverão eventualmente para consumir a base de dados.

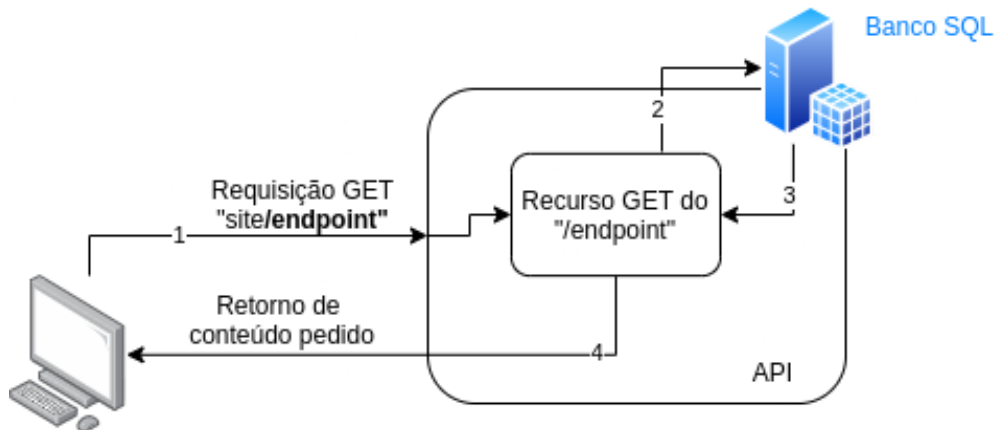
Figura 1: Diagrama de Classes com atributos



Fonte: O autor (2021)

A criação do banco de dados com tabelas e modelos foi feito de acordo com a Figura 1, onde é possível observar um relacionamento entre atributos, foi utilizado a biblioteca SQL Alchemy. Para utilizar esses modelos, criou-se recursos com a biblioteca Flask. O acesso dos modelos se dá por métodos de requisição dentro dos recursos.

Figura 2: Diagrama com um exemplo de requisição



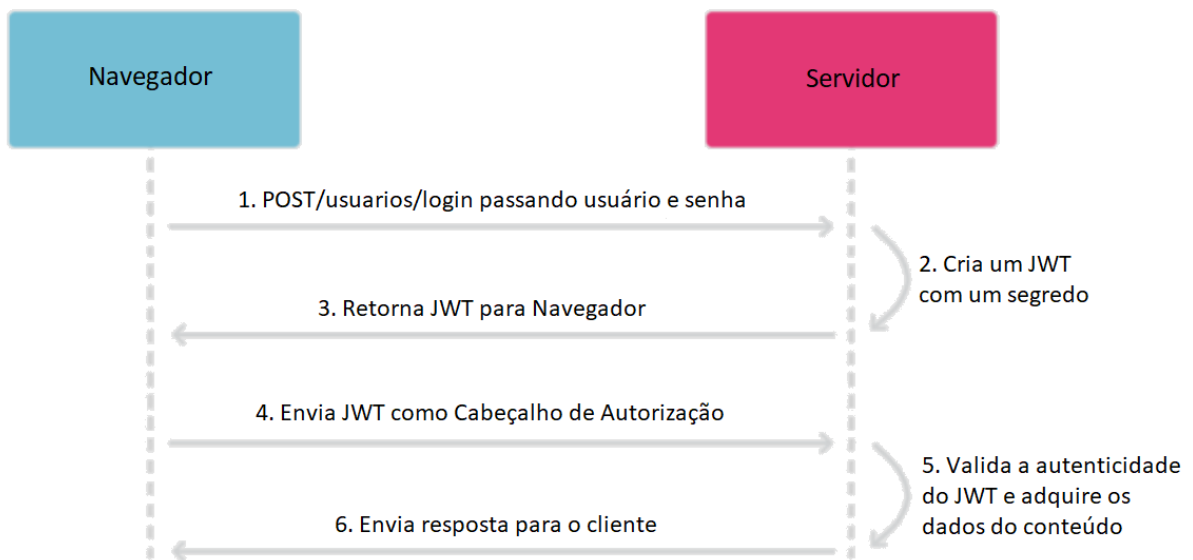
Fonte: O autor (2021)

Como exemplificado na Figura 2, os métodos dentro de um recurso são definidos por verbos HTTP e acessados por *endpoints* da API. Assim, é possível alimentar e consumir os dados dos modelos de dentro da API.

Segurança

A camada de segurança com a finalidade de proteger os dados de usuários mal-intencionados, foi feita através de autenticação de login JWT (JSON Web Token), um token assinado que faz a autenticação de uma requisição web como exemplificado na Figura 3.

Figura 3: Diagrama sobre o funcionamento do JWT



Fonte: Adaptado pelo autor de NASCIMENTO; “Entendendo tokens JWT (Json Web Token)” (2018)

Porém, a segurança vem do fato do token ser encriptado através de um algoritmo escolhido, na API desenvolvida foi utilizado o algoritmo RS256 por ser um algoritmo assimétrico e utilizar um par de chaves, tornando-o mais seguro.

A implementação de JWT foi feita a partir da biblioteca Flask. Outro fator de segurança implementado é a utilização de cargos para camadas de utilização. Dessa forma, métodos-base de requisição podem ser utilizados por todos os usuários, e administradores podem fazer todas as funções, com exceção de uma, o método de postagem de relatórios feitos apenas por estações meteorológicas que também terão usuários próprios.

Deploy

Para realizar a instalação na Nuvem e executar a API quase sem interrupções durante o processo de atualização, foi feito um container Docker. Um container Linux otimiza a utilização de hardware e fornece isolamento total entre as aplicações no servidor. Além disso sua replicabilidade é transparente e independente do hardware do servidor, sendo assim fornece alta capacidade de escalabilidade horizontal, então o número de dispositivos pode aumentar exponencialmente que a API irá acompanhar esse crescimento.

Funcionamento

O método utilizado durante o funcionamento, a API funcionará de forma ininterrupta e a prefeitura de Santo André implementará um sistema onde os dados das estações serão enviados para um portal e os mesmos encaminhados para a API.

Ciência de Dados

A Ciência de Dados é uma disciplina de tornar os dados uteis, ou seja, é a ciência de extrair *insights* de dados criados pelas organizações atuais. Para extrair as informações a partir de um conjunto de dados é necessário passar por uma série de etapas (IBM, 2021).

A partir de aulas do Minor de Ciência de Dados (conjunto de eletivas fechadas relacionados a Ciência de Dados) oferecido pelo Instituto Mauá de Tecnologia e leitura de materiais relacionados, foi possível aprender conceitos para executar extração de dados a partir de Análise Descritiva e Modelagem de Dados. Além de conceitos para criar modelos de aprendizado de máquina a partir dessa extração.

Resultados e Discussão

API

Para verificar a integridade e disponibilidade do sistema, testes foram feitos a partir de *scripts* Python para simular um ambiente de funcionamento do sistema passando por todos os recursos disponíveis. Os testes mostraram que o sistema consegue suportar uma quantidade de requisições suficientes para o funcionamento além de não ter apresentado nenhum problema de integridade.

Além dos testes por *scripts*, foram feitos testes manuais para verificar a segurança da API. Essa citada anteriormente se mostrou segura tanto no quesito de uso de tokens JWT quanto nos cargos de usuários no sistema.

Uma vez que não foi possível integrar o sistema por completo neste momento, não há como afirmarmos o funcionamento real. Porém, as simulações mostraram que é possível o uso do sistema de forma estável com um número considerável de requisições em um intervalo de tempo. Além disso, durante a construção da API foi possível fazer modificações e acréscimos necessários de forma rápida e eficiente graças aos *frameworks* utilizados e à arquitetura REST.

Porém, a arquitetura REST não foi totalmente implementada devido à utilização do sistema de login juntamente com os recursos, caso seja necessário refatorar o código será feita uma nova API para sistemas de cadastro, login e cargos.

Ciência de Dados

Através das aulas do Minor e leituras de materiais foi possível aprender conceitos utilizados na área de Ciência de Dados.

Na disciplina Análise de Dados, ferramentas e softwares, como Power BI e linguagem de programação Python, são utilizados, em cima de um conjunto de dados, para realizar uma Análise Descritiva sendo este um processo em que se explora o conjunto de dados a fim de extrair informações relevantes através destes passos:

- **Pré-processamento:** ocorre a verificação de possível falta de dados e ajuste de variáveis para adequá-las.
- **Análise:** são feitos gráficos e tabelas de correlação a fim de analisar visual e estatisticamente.

Na disciplina Aprendizado de máquina, definido por Arthur Samuel, em 1959 como: “O campo de estudo que dá aos computadores a habilidades de aprender sem ser explicitamente programado.” (GERON, 2019), linguagem Python foi utilizada dentro de *Jupyter Notebooks* (aplicação em que *scripts* podem ser feitos e executados de forma separada, porém em um ambiente único) para utilizar modelos de Aprendizado de Máquina. A linguagem Python é principalmente utilizada por sua vasta variedade de bibliotecas para este fim e com comunidades amplas complementando e criando bibliotecas.

Os modelos de Aprendizado de Máquina são categorizados em diferentes tipos, sendo alguns deles:

- **Supervisionado:**
 - Aprendizado em que os dados de treinamento têm inclusos soluções desejadas, chamados de rótulo. Alguns modelos utilizados para este tipo de aprendizado são:
 - Regressão Linear.

- Regressão Logística.
- Árvores de decisão e Florestas Aleatórias.

Sendo esses também separados em modelos de regressão e de classificação.

- **Não-supervisionado:**

Aprendizado em que não existe rótulos no conjunto de dados de treinamento, então é um modelo “autodidata”. Alguns modelos utilizados para este tipo de aprendizado são:

- *K-Means*.
- Análise de Componentes Principais.
- *Apriori*.

Na disciplina Ciência de dados, também foi utilizado *Jupyter Notebook*, porém neste caso para realizar a Análise Descritiva e a partir dela uma Modelagem de Dados, processo onde gera-se variáveis descritivas a partir dos dados existentes através de normalização e conversão de dados categóricos, por exemplo. Depois de passar por essas duas etapas, aplica-se os conhecimentos da disciplina de Aprendizado de Máquina para processar os dados corrigidos e modelados através de um modelo adequado. O processo de Modelagem de Dados e Aprendizado de Máquina pode-se repetir diversas vezes até que se encontre um modelo final satisfatório.

Em estudos sobre previsões de alagamentos, observou-se a utilização de dados meteorológicos, o que enquadra eles em um tipo de conjunto de dados chamado Série Temporal. Esse citado anteriormente é definido como um conjunto de observações registradas sequencialmente, em intervalos regulares, durante um período de tempo (ANDRADE; IPPOLITO; PEREIRA, 2020).

No mesmo trabalho citado anteriormente, a análise descritiva mostra a relação existente entre a Ponto de Orvalho e Temperatura com Precipitação e utilizaram modelos de aprendizado de máquina supervisionado para prever alagamentos. Já outro, para estudar a influência da variabilidade espacial da chuva para previsão de alagamentos, utilizou dados de acúmulo de precipitação com baldes pluviométricos e correlação espacial (ARNAUD et al., 2002).

Os casos anteriores mostram como dentro da Ciência de Dados há diversas maneiras para chegar em determinados fins.

Conclusões

Este trabalho mostrou que o armazenamento e distribuição de conjunto de dados pode ser feito de forma rápida e eficiente com utilização de API e que a arquitetura REST permite realizá-lo de forma mais sólida e escalável. É possível simular ambientes para verificar a integridade do sistema através de *scripts* e, assim, evitar que problemas passem despercebidos.

O estudo sobre Ciência de Dados mostrou os processos e métodos se utilizam e algumas ferramentas que auxiliam de forma geral. Especificamente em estudos do uso de Ciência de Dados em previsão de alagamentos, mostrou que há diversas formas para fazê-lo de acordo com os recursos disponíveis.

Referências Bibliográficas

ANDRADE, F. P. S.; IPPOLITO, F. B.; PEREIRA, V. S.; *Sistema Inteligente de Previsão de Alagamentos*. 2020. Disponível no acervo da biblioteca do Instituto Mauá de Tecnologia.

ARNAUD, P.; BOUVIER, C.; CISNEROS, L.; DOMINGUEZ, R.; *Influence of rainfall spatial variability on flood prediction*, 2002. Journal of hydrology, v. 260, p. 216-230. ISSN 0022-1694

FIELDING, R.; *Architectural Styles and the Design of Network-based Software Architectures*, 2000. Dissertação de doutorado. Universidade da Califórnia, Iverne. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>. Acesso em: 17 Nov. 2021.

FONSECA, E.; *O que é ORM?* Disponível em <[https://www.treinaweb.com.br/blog/o-que-e-orm#:~:text=Object%2DRelational%20Mapping%20\(ORM\),do%20banco%20de%20dados%20relacional](https://www.treinaweb.com.br/blog/o-que-e-orm#:~:text=Object%2DRelational%20Mapping%20(ORM),do%20banco%20de%20dados%20relacional)>. Acesso em: 26 Out. 2021.

GERON, A.; *Mãos à Obra Aprendizado de Máquina com Scikit-Learn & TensorFlow: Conceitos, Ferramentas e Técnicas Para Construção de Sistemas Inteligentes*. 2019. ISBN: 978-85-508-0381-4

GUEDES, M.; *O que é uma API?* 2019. Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-uma-api>>. Acesso em: 07 Nov. 2021.

IBM; *Data Science*. Disponível em <<https://www.ibm.com/br-pt/analytics/data-science>>. Acesso em: 10 Nov. 2021.

NASCIMENTO, W. *Entendendo tokens JWT (Json Web Token)*. 2018. Disponível em: <<https://medium.com/tableless/entendendo-tokens-jwt-json-web-token-413c6d1397f6>>. Acesso em: 24 Nov. 2021.

NOBRE, C. A.; YOUNG, A. F.; SALDIVA, P. H. N.; ORSINI, J. A. M.; NOBRE, A. D.; OGURA, A. T.; THOMAZ, O.; VALVERDE, M.; OBREGON, G.; SILVA, G. C. M. D.; SILVEIRA, A., C.; RODRIGUEZ, G. O.; OJIMA, R.; *Vulnerabilidades das megacidades brasileiras às mudanças climáticas: Região Metropolitana de São Paulo*, 2011. Disponível em: <https://www.nepo.unicamp.br/publicacoes/livros/megacidades/megacidades_RMSP.pdf>. Acesso em: 3 Nov. 2021.

VALVERDE, M.; CARDOSO, A.; BRAMBILA, R.; *O padrão de chuvas na região do ABC paulista: os extremos e seus impactos*. *Revista Brasileira de Climatologia*, v. 22, n. 0, 2018. ISSN 2237-8642. Disponível em: <<https://revistas.ufpr.br/revistaabclima/article/view/45929>>.

ZANELLA, A.; BUI, N.; CASTELLANI, A.; VANGELISTA, L.; ZORZI, M.; *Internet of Things for Smart Cities*. Disponível em <<https://ieeexplore.ieee.org/document/6740844>>. Acesso em: 25 Out. 2021.