

**INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS  
GERAIS - *CAMPUS* AVANÇADO PIUMHI  
BACHARELADO EM ENGENHARIA CIVIL**

Rafael Leonel de Castro

**SOLUÇÃO COMPUTACIONAL DIDÁTICA PARA CÁLCULO DE DRENAGEM  
URBANA EM PYTHON: UM ESTUDO DE CASO EM PIUMHI - MG**

Piumhi - MG

2025

Rafael Leonel de Castro

**SOLUÇÃO COMPUTACIONAL DIDÁTICA PARA CÁLCULO DE DRENAGEM  
URBANA EM PYTHON: UM ESTUDO DE CASO EM PIUMHI - MG**

Trabalho de conclusão de curso de graduação apresentado ao Instituto Federal de Ciência e Tecnologia de Minas Gerais como requisito parcial para a obtenção do título de Bacharel em Engenharia Civil.

Orientador: Prof. Dr. Germano de Oliveira Mattosinho

Coorientador: Me. Mauro Henrique Silva

Piumhi - MG

2025

## FICHA CATALOGRÁFICA

---

C355s Castro, Rafael Leonel de.

Solução computacional didática para cálculo de drenagem urbana em Python: um estudo de caso em Piumhi-MG [manuscrito] / Rafael Leonel de Castro. – 2025.  
106 f. : il. color.

Orientador: Germano de Oliveira Mattosinho.

Coorientador: Mauro Henrique Silva.

Trabalho de Conclusão de Curso (bacharelado) – Instituto Federal Minas Gerais. *Campus Avançado Piumhi*, 2025.

1. Urbanização. 2. Drenagem urbana. 3. Modelagem da informação da cidade (CIM). 4. Python (linguagem de programação de computador). 5. Gestão inteligente. I. Mattosinho, Germano de Oliveira. II. Silva, Mauro Henrique. III. Instituto Federal de Minas Gerais. *Campus Avançado Piumhi*. IV. Título.

CDD 627.4

---

Catálogo: Andreia Cristina Damasceno - CRB-6/1974


Rafael Leonel de Castro

## SOLUÇÃO COMPUTACIONAL DIDÁTICA PARA CÁLCULO DE DRENAGEM URBANA EM PYTHON: UM ESTUDO DE CASO EM PIUMHI - MG

Trabalho de conclusão de curso de graduação apresentado ao Instituto Federal de Ciência e Tecnologia de Minas Gerais como requisito parcial para a obtenção do título de Bacharel em Engenharia Civil.


Aprovado em: 08 / 01 /2025 pela banca examinadora:

### BANCA EXAMINADORA

Documento assinado digitalmente  
 GERMANO DE OLIVEIRA MATTOSINHO  
Data: 13/01/2025 20:51:48-0300  
Verifique em <https://validar.it.gov.br>

Prof. Dr. Germano de Oliveira Mattosinho

Docente Orientador(a)

Documento assinado digitalmente  
 HUMBERTO COELHO DE MELO  
Data: 14/01/2025 15:26:41-0300  
Verifique em <https://validar.it.gov.br>


Prof. M.e Humberto Coelho de Melo

Membro da Banca Examinadora – Docente da área

Documento assinado digitalmente  
 STELLA MARIA GOMES  
Data: 14/01/2025 10:38:51-0300  
Verifique em <https://validar.it.gov.br>


Prof. M.e. Stella Maria Gomes

Membro da Banca Examinadora – Docente de outra área

Documento assinado digitalmente  
 JOAO BATISTA PEREIRA  
Data: 13/01/2025 21:23:15-0300  
Verifique em <https://validar.it.gov.br>

Eng. M.e. João Batista Pereira

Membro da Banca Examinadora – Membro IV

Documento assinado digitalmente  
 YURI TAGLIERI SAO  
Data: 14/01/2025 09:28:54-0300  
Verifique em <https://validar.it.gov.br>

Eng. M.e. Yuri Taglieri São

Membro da Banca Examinadora – Membro V

## AGRADECIMENTOS

A Deus, pela força, sabedoria e proteção em todos os momentos dessa jornada. Sem a Sua presença, este trabalho e todas as conquistas que o antecederam não seriam possíveis.

Aos meus pais, pelo amor incondicional, incentivo e dedicação ao longo de minha vida. Vocês são a base de tudo e os maiores responsáveis por eu estar onde estou hoje.

À minha namorada, pelo carinho, paciência e por estar ao meu lado nos momentos de dificuldade e de alegria, sempre me motivando a seguir em frente.

Aos professores que, com dedicação e conhecimento, contribuíram para a minha formação acadêmica e para o desenvolvimento deste trabalho. Suas orientações foram essenciais para que este projeto tomasse forma.

Aos meus amigos de faculdade, pelo companheirismo, troca de experiências e por fazerem desta etapa da minha vida uma experiência mais leve e enriquecedora.

Ao meu orientador, pela prontidão em me ajudar sempre que necessário, compartilhando seu conhecimento e experiência de forma generosa.

Ao meu coorientador, que mesmo pertencendo a outro *campus*, nunca mediu esforços para contribuir com este trabalho, sendo um apoio fundamental para o seu desenvolvimento.

A cada um de vocês, meu mais profundo agradecimento.

## RESUMO

O acelerado processo de urbanização tem imposto desafios complexos aos sistemas de drenagem urbana, especialmente devido à crescente impermeabilização do solo, o que destaca a urgência na implementação de soluções tecnológicas para mitigar e prevenir tais problemas. Este estudo propõe o desenvolvimento de uma **Solução Computacional Didática para Cálculo de Drenagem Urbana**, utilizando a linguagem de programação Python, com ênfase no dimensionamento de sarjetas e na quantificação de bocas de lobo. O principal objetivo foi otimizar o processo de dimensionamento, oferecendo uma ferramenta que facilite a tomada de decisões no planejamento e gestão da infraestrutura urbana. A metodologia adotada baseia-se em **Programação Orientada a Objetos (POO)**, estruturando a solução em oito classes específicas, cada uma dedicada a um estágio distinto do dimensionamento. A ferramenta foi aplicada em um estudo de caso na cidade de Piumhi - MG, onde problemas recorrentes de drenagem causavam sérios transtornos à população. Os resultados obtidos demonstraram a eficácia do modelo, proporcionando cálculos mais rápidos e precisos, além de minimizar os erros frequentemente associados ao processo manual. Além disso, a solução oferece a capacidade de simular cenários futuros, considerando a expansão urbana e o aumento dos volumes de água a serem transportados e coletados, com base na projeção de crescimento populacional. A ferramenta também contribui para o aprimoramento dos sistemas de drenagem existentes, apoiando a tomada de decisões mais embasadas e eficientes. Conclui-se que a incorporação de tecnologias avançadas no campo da engenharia urbana é crucial para a obtenção de resultados mais confiáveis, eficientes e sustentáveis, promovendo soluções integradoras que atendem tanto às necessidades da população quanto aos desafios do planejamento e gestão urbanos.

Palavras-chave: urbanização; drenagem urbana; modelagem da informação da cidade (CIM); Python (linguagem de programação de computador); gestão inteligente.

## ABSTRACT

The accelerated process of urbanization has posed complex challenges to urban drainage systems, particularly due to the increasing soil impermeabilization, which highlights the urgency of implementing technological solutions to mitigate and prevent such issues. This study proposes the development of a Didactic Computational Solution for Urban Drainage Calculation, using the Python programming language, with a focus on the dimensioning of curbs and the quantification of stormwater inlets. The main objective was to optimize the dimensioning process, providing a tool that facilitates decision-making in urban planning and infrastructure management. The methodology adopted is based on Object-Oriented Programming (OOP), structuring the solution into eight specific classes, each dedicated to a distinct stage of the dimensioning process. The tool was applied in a case study in the city of Piumhi, MG, where recurring drainage problems caused significant disruptions to the population. The results obtained demonstrated the effectiveness of the model, providing faster and more accurate calculations, while minimizing the errors often associated with manual processes. Moreover, the solution offers the capability to simulate future scenarios, considering urban expansion and increased volumes of water to be transported and collected, based on projected population growth. The tool also contributes to the enhancement of existing drainage systems, supporting more informed and efficient decision-making. To conclude, the integration of advanced technologies in the field of urban engineering is essential for achieving more reliable, efficient, and sustainable results, promoting integrative solutions that meet both the needs of the population and the challenges of urban planning and management.

**Keywords:** urbanization; urban drainage; City Information Modeling (CIM); Python (computer programming language); smart management.

## LISTA DE FIGURAS

Figura 1 – Alagamento na Rua Amazonas em Piumhi - MG.....	15
Figura 2 – Ponte alagada na Rua Amazonas em Piumhi - MG.....	15
Figura 3 – Boca de lobo em Piumhi - MG .....	16
Figura 4 - Taxa de urbanização em 2022: população urbana de Piumhi - MG comparada à população média de MG e Brasil .....	16
Figura 5 – População urbana e rural de Piumhi - MG em 2022.....	17
Figura 6 – População urbana e rural de Piumhi - MG nos anos de 1991, 2000, 2010 e 2022 .	17
Figura 7 - Impacto da urbanização no balanço hídrico, escoamento e áreas de inundação .....	21
Figura 8 - Estrutura do plano de águas pluviais .....	26
Figura 9 - Perfil transversal da sarjeta.....	33
Figura 10 – Comparativo de bocas de lobo em cruzamentos de vias.....	34
Figura 11 – Bocas de lobo em cruzamentos de vias.....	35
Figura 12 – Tipos de boca de lobo .....	36
Figura 13 – Bocas de lobo tipo grelha.....	37
Figura 14 – PV.....	38
Figura 15 - Código na linguagem PHP.....	43
Figura 16 - Código na linguagem C++.....	44
Figura 17 - Código na linguagem Java.....	44
Figura 18 - Código na linguagem Python.....	44
Figura 19 - Linguagens de programação preferidas das pessoas que estão aprendendo a programar.....	46
Figura 20 – Exemplo de comentários em Python.....	53
Figura 21 – Declaração de método em Python.....	55
Figura 22 - Hierarquia de classes .....	58
Figura 23 – Atributos e métodos da calculadora .....	59
Figura 24 - Fluxograma sintético de execução da solução computacional. ....	65
Figura 25a - Resultados da solução computacional .....	83
Figura 25b - Resultados da solução computacional .....	83
Figura 25c - Resultados da solução computacional .....	84
Figura 25d - Resultados da solução computacional .....	84
Figura 25e - Resultados da solução computacional .....	85
Figura 26a - Resultados da planilha .....	85

Figura 26b - Resultados da planilha .....	86
---	----

## LISTA DE QUADROS

Quadro 1 - Períodos de retorno para diferentes ocupações .....	28
Quadro 2 – Período de recorrência.....	28
Quadro 3 - Coeficiente de Escoamento Superficial / <i>Run-Off</i> de acordo como tipo de ocupação e de bacia. ....	30
Quadro 4 - Coeficiente de Escoamento Superficial / <i>Run-Off</i> por tipo de superfície. ....	30
Quadro 5 – Coeficiente de rugosidade (Manning) .....	32
Quadro 6 - Fatores de redução de escoamento das sarjetas.....	33
Quadro 7 – Fator de redução das bocas de lobo .....	37
Quadro 8 – <i>DataFrame</i> de informações pessoais.....	50
Quadro 9 - Tipos de Variáveis em Python .....	51
Quadro 10 - Operadores Matemáticos em Python.....	52
Quadro 11 - Operadores de Comparação em Python .....	52
Quadro 12 – Exemplo de código com <i>Docstrings</i> em Python .....	54
Quadro 13 – Exemplo de funções em Python .....	54
Quadro 14 – Exemplo de aplicação do método <i>format</i> em Python.....	56
Quadro 15 – Exemplo de aplicação do método <i>read</i> em Python .....	56
Quadro 16 – Exemplo de aplicação do método <i>capitalize</i> em Python .....	56
Quadro 17 – Exemplo de aplicação do método <i>upper</i> em Python .....	56
Quadro 18 – Exemplo de aplicação do método <i>replace</i> em Python.....	57
Quadro 19 – Exemplo de aplicação do método <i>find</i> em Python .....	57
Quadro 20 - Exemplos de códigos em Python .....	61
Quadro 21 - Execução dos códigos ( <i>main</i> ) em Python .....	61

## LISTA DE TABELAS

Tabela 1 – Valores dos coeficientes de Manning ( $n$ ).....	32
Tabela 2 – Espaçamentos entre os poços de visita .....	38

## LISTA DE SIGLAS

API	Application Programming Interface
BIM	Building Information Modeling
CIM	City Information Modeling
DNIT	Departamento Nacional de Infraestrutura de Transportes
GIS	Geographic Information System
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Integrated Development Environment
PDDrU	Plano Diretor de Drenagem Urbana
POO	Programação Orientada a Objetos
QGIS	Quantum Geographic Information System
TICs	Tecnologias de Informação e Comunicação

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>14</b>
<b>1.1</b>	<b>Justificativa</b> .....	<b>18</b>
<b>1.2</b>	<b>Objetivos</b> .....	<b>19</b>
<b>1.2.1</b>	Geral.....	19
<b>1.2.2</b>	Específicos .....	19
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b> .....	<b>20</b>
<b>2.1</b>	<b>O Processo de Urbanização</b> .....	<b>20</b>
<b>2.2</b>	<b>Drenagem Urbana</b> .....	<b>23</b>
2.2.1	Área de estudo.....	27
2.2.2	Intensidade máxima de precipitação .....	27
2.2.3	Vazão de chuva .....	29
2.2.4	Meio Fio.....	31
2.2.5	Sarjeta.....	31
2.2.6	Bocas de Lobo.....	34
2.2.7	Poços de Visita (PV's) .....	38
<b>2.3</b>	<b>Sistema de Informações Geográficas (SIG) ou Geographic Information System (GIS)</b> .....	<b>38</b>
<b>2.4</b>	<b>Tecnologias de Informação e Comunicação (TICs)</b> .....	<b>41</b>
<b>2.5</b>	<b>Python</b> .....	<b>42</b>
2.5.1	Bibliotecas.....	48
2.5.1.1	<i>DataFrames</i> .....	49
2.5.2	Variáveis .....	50
2.5.3	Operadores .....	51
2.5.4	Comentários em Python.....	53
2.5.5	Funções .....	53
2.5.6	Métodos.....	55
<b>2.6</b>	<b>Programação Orientada a Objetos (POO)</b> .....	<b>57</b>
2.6.1	Métodos de Classe .....	60
2.6.2	Outras Palavras Reservadas do Python.....	60
2.6.3	Exemplos de códigos .....	60
<b>2.7</b>	<b>O Projeto de Pesquisa</b> .....	<b>62</b>
<b>3</b>	<b>METODOLOGIA</b> .....	<b>64</b>

<b>3.1</b>	<b>Área de Estudo .....</b>	<b>66</b>
<b>3.2</b>	<b>O Projeto de Programação.....</b>	<b>66</b>
3.2.1	Classes.....	67
3.2.1.1	<i>Classe LerTabela .....</i>	<i>67</i>
3.2.1.2	<i>Classe IntensidadePluviometrica.....</i>	<i>68</i>
3.2.1.3	<i>Classe Localidade .....</i>	<i>69</i>
3.2.1.4	<i>Classe Trecho.....</i>	<i>70</i>
3.2.1.5	<i>Classe Sarjeta .....</i>	<i>71</i>
3.2.1.6	<i>Classe BocaDeLobo.....</i>	<i>73</i>
3.2.1.7	<i>Classe ProfissionalEngenharia.....</i>	<i>76</i>
3.2.1.8	<i>Classe Main.....</i>	<i>79</i>
<b>4</b>	<b>RESULTADOS E DISCUSSÃO .....</b>	<b>81</b>
<b>4.1</b>	<b>Comparação entre cálculos e intervenções realizadas em Piumhi - MG .....</b>	<b>87</b>
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>89</b>
	<b>REFERÊNCIAS .....</b>	<b>90</b>
	<b>APÊNDICE A – CADASTRAMENTO DOS DADOS E CÁLCULOS REALIZADOS PELA SOLUÇÃO COMPUTACIONAL.....</b>	<b>99</b>
	<b>APÊNDICE B – CADASTRAMENTO DOS DADOS E CÁLCULOS REALIZADOS NA PLANILHA.....</b>	<b>104</b>

## 1 INTRODUÇÃO

O processo de urbanização tem aumentado a cada dia e com isso surge a necessidade de utilizar áreas naturais para expansão das cidades, além das áreas que já eram ocupadas, as quais se tornam cada vez mais saturadas e impermeabilizadas, conseqüentemente, aumentando o escoamento superficial.

A alta impermeabilização do solo é uma das conseqüências graves desse processo de urbanização, sendo que antes, as águas oriundas de precipitação e de uso do homem, em grande parte eram absorvidas pelo solo e agora, com a redução dessa área de infiltração, ocorre conseqüentemente o aumento do escoamento superficial.

Esses fatores acabam sobrecarregando os sistemas de drenagem, pois a princípio, quando os mesmos foram projetados (mesmo considerando o crescimento populacional), o volume de escoamento provavelmente era inferior e possivelmente não se calculou e adotou a taxa de crescimento adequada, visto que há problemas de alagamentos (Figuras 1 e 2). Assim, mesmo que agora o sistema de drenagem esteja sendo eficiente, em grande parte do município, ele poderá apresentar problemas futuros, causando transtornos como: alagamento das vias, deterioração de sarjetas e vias, disseminação de doenças, entre outros.

Nos grandes centros urbanos os problemas tendem a apresentar maiores proporções devido a vários fatores supracitados, mas a realidade é que independentemente do local e da densidade populacional os transtornos causados por um sistema de drenagem inexistente ou ineficiente será visível principalmente em dias mais chuvosos. Fato este que pode ser comprovado no município de Piumhi - MG e que foi registrado pela própria população nas datas de 15/10/2018 e 19/02/2019, conforme Figuras 1 e 2. Vale ressaltar que tais problemas já foram sanados pelo município com a atualização do sistema de drenagem urbana, p.e. quantidade de bocas de lobo, na própria Rua Amazonas e na Rua Severo Veloso, que possuía parte da sua vazão precipitada desaguando na Rua Amazonas.

Figura 1 – Alagamento na Rua Amazonas em Piumhi - MG



Fonte: Facebook, 2019.

Figura 2 – Ponte alagada na Rua Amazonas em Piumhi - MG



Fonte: Facebook, 2019.

Uma das causas para estes alagamentos pode ser pela grande quantidade de lixo jogado nas ruas e/ou outros rejeitos da construção civil, bem como folhas, que são conduzidos para as sarjetas até chegar às bocas de lobo e que acabam obstruindo parte da passagem de água nas mesmas, fator este que pôde ser observado com frequência em alguns bairros mais novos na cidade de Piumhi - MG, conforme mostrado na Figura 3. Esta questão também precisa ser analisada em relação a gestão pública da limpeza urbana, o que não é escopo deste trabalho.

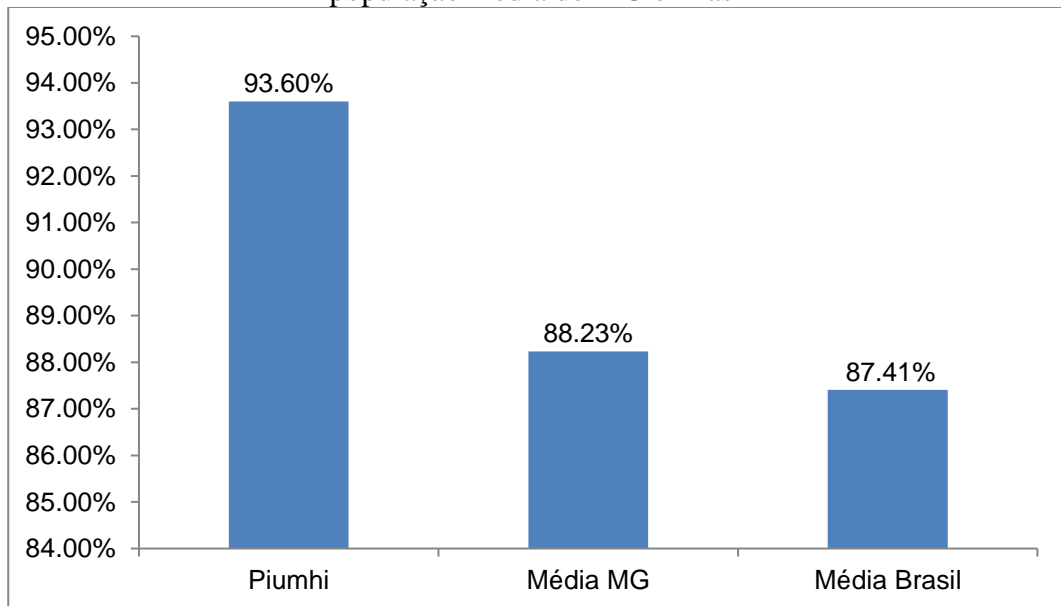
Figura 3 – Boca de lobo em Piumhi - MG



Fonte: Arquivo pessoal, 2019.

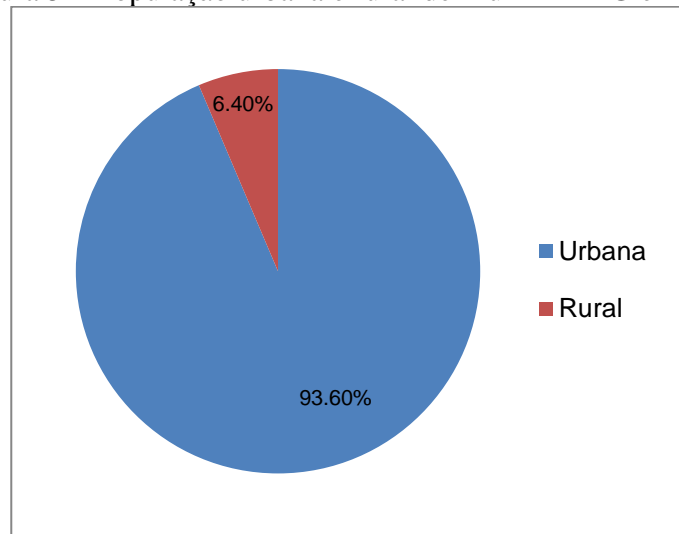
Por meio da Figura 4, é possível observar que em 2022, o município de Piumhi - MG, apresentava uma taxa de urbanização acima da média das taxas de Minas Gerais e do Brasil, fato este que corrobora a relevância de análises acerca do tema proposto. Em complemento, a Figura 5 ilustra o comparativo entre a população urbana e rural no ano de 2022, do mesmo município citado anteriormente.

Figura 4 - Taxa de urbanização em 2022: população urbana de Piumhi - MG comparada à população média de MG e Brasil



Fonte: IBGE, 2022, adaptado pelo autor, 2025.

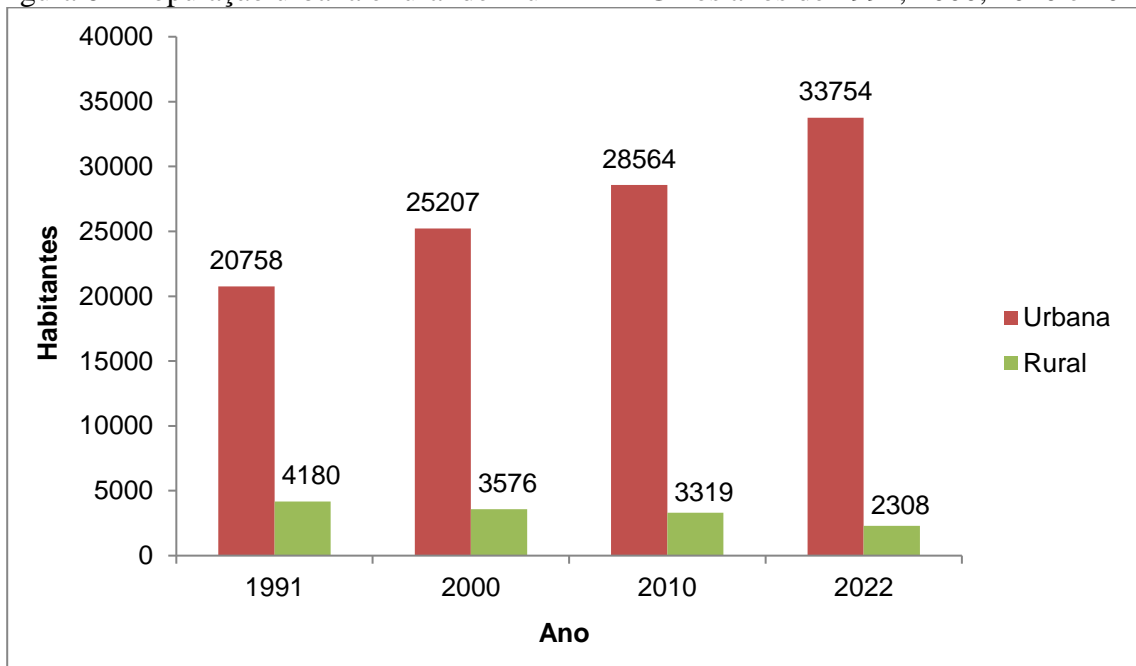
Figura 5 – População urbana e rural de Piumhi - MG em 2022



Fonte: IBGE, 2022, adaptado pelo autor, 2025.

A análise dos dados apresentados na Figura 6 revela o crescimento populacional de Piumhi - MG, evidenciado pelo aumento significativo no número de habitantes. De acordo com o IBGE, em 1991 o município tinha uma população urbana e rural de 20.758 e 4.180 habitantes, respectivamente, já em 2022 foram registrados 33.754 habitantes na zona urbana e 2.308 habitantes na zona rural (Instituto Brasileiro de Geografia e Estatística – IBGE, 1991; 2000; 2010; 2022).

Figura 6 – População urbana e rural de Piumhi - MG nos anos de 1991, 2000, 2010 e 2022



Fonte: IBGE, 1991; 2000; 2010; 2022, adaptado pelo autor, 2025.

Nascimento (2011) relata que dentre todos os municípios, os que mais sofrem com problemas de drenagem urbana são os que possuem menos de 45 mil habitantes. As causas podem ser, falta de recursos para investimentos nessa área, regiões com climas e relevos bem distintos onde tem menores precipitações e conseqüentemente menos investimentos na área de drenagem, outra causa também, poderia ser que, os municípios menos populosos tendem a apresentar níveis educacionais e socioeconômicos mais baixos, não conseguindo requerer de forma correta serviços de boa qualidade.

Cabe aos profissionais da área de engenharia utilizarem os avanços tecnológicos para estudar e desenvolver soluções capazes de auxiliar e otimizar a elaboração de projetos, de forma a obter resultados mais acurados e conseqüentemente solucionar problemas como os citados anteriormente.

Conforme observado no estado da arte (Capítulo 2) e pela escassez de sistemas de drenagem urbana em alguns municípios relatado pelo Sistema Nacional de Informações sobre Saneamento (SNIS, 2022), de todos os municípios do Brasil, 43,6% possuem sistema exclusivo para drenagem das águas pluviais urbanas, que é a solução correta e adequada para evitar problemas como alagamentos e contaminação do esgoto, 10,9% possuem sistema unitário (misto com esgotamento sanitário), 26,3% possuem sistema combinado e 4,1% possuem algum tipo de tratamento para águas pluviais. Em contrapartida, 15,1% dos municípios não possuem nenhum tipo de sistema de drenagem, o que reflete uma grave deficiência na infraestrutura urbana e no planejamento de saneamento (SNIS, 2022). Nesse contexto, o presente trabalho preenche parte desta lacuna de otimização gratuita para auxiliar no dimensionamento de microdrenagem urbana.

## **1.1 Justificativa**

A escolha do tema deste trabalho foi motivada pela experiência adquirida em um projeto de pesquisa financiado por bolsa no IFMG *Campus* Piumhi - MG. Durante o desenvolvimento deste projeto, identificou-se a necessidade de soluções para locais com irregularidades no sistema de drenagem urbana, que tem gerado problemas de inundações em bairros próximos ao campus (Silva *et al.*, 2019b e 2020; Silva e Mattosinho, 2021). Essa problemática evidenciou a relevância de se propor ferramentas computacionais que auxiliem no dimensionamento e melhoria desses sistemas.

Outro fator que reforça a importância deste estudo é o crescimento populacional acelerado e o aumento da migração de habitantes da zona rural para a zona urbana, conforme apontado nas Figuras 4, 5 e 6. Esse fenômeno resulta em maior impermeabilização do solo e

pressiona as infraestruturas existentes, agravando problemas relacionados à drenagem pluvial. Assim, o trabalho busca contribuir com soluções práticas e didáticas, aplicáveis ao contexto urbano de Piumhi - MG.

## **1.2 Objetivos**

### **1.2.1 Geral**

O presente estudo caracteriza-se pela elaboração de um modelo computacional, através da linguagem de programação Python, que auxiliará no dimensionamento de sarjetas e bocas de lobo, que fazem parte do sistema de drenagem urbana, a fim de automatizar e otimizar o método de dimensionamento já existente.

### **1.2.2 Específicos**

- Aprofundar os conhecimentos e métodos projetivos, obtidos na área de drenagem urbana, através da disciplina de Saneamento cursada na graduação;
- Definir o método de dimensionamento para microdrenagem urbana;
- Aplicar a metodologia de cálculo para dimensionar parte de um sistema de microdrenagem urbana;
- Elaborar código em Python com ênfase na Programação Orientada a Objetos (POO) para compor a modelagem de dimensionamento;
- Aplicar os conceitos da área de programação no desenvolvimento de blocos de códigos, em sintaxe clara e objetiva;
- Obter como produto final um modelo numérico simplificado para auxiliar no dimensionamento de elementos de microdrenagem urbana;
- Gerar para o usuário uma planilha resumo com as quantidades de bocas de lobo necessárias em cada trecho de acordo com os cálculos realizados;
- Validar o produto final aplicando-o ao município de Piumhi - MG. Para isso, será utilizada uma planilha de cálculo previamente desenvolvida e validada em um projeto de pesquisa anterior, com base em dados reais coletados no mesmo estudo.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 O Processo de Urbanização

De acordo com Righetto (2009), há anos atrás, as cidades eram sendo desenvolvidas próximas aos leitos dos rios e córregos com o intuito de facilitar o acesso a água, mas com o passar dos anos foram surgindo novas tecnologias que auxiliavam no desenvolvimento e as cidades foram se expandindo, exigindo então obras de pavimentação, construção de pontes e desvio dos cursos d'água, conseqüentemente foram ocorrendo a ocupação total ou parcial de áreas que eram comumente alagadas naturalmente e de alguns cursos d'água secundários.

Como destaca Felizardo (2016), as manifestações da urbanização não se referem apenas ao aumento da população, mas também à expansão territorial, com o crescimento no número de ruas, residências, indústrias, entre outros. Além disso, Cabral e Cândido (2019) evidenciam que a urbanização também está atrelada ao desenvolvimento econômico, social e ambiental, ou seja, esse processo de urbanização quando ocorrido de forma desorganizada pode causar prejuízos ao homem e ao meio ambiente.

Os primeiros impactos causados pela expansão ou criação de uma nova cidade se manifestam já nas fases iniciais de execução, principalmente quando envolvem grande desmatamento. Segundo Tucci e Mendes (2006), a vegetação tem um papel crucial na regulação do fluxo de água, uma vez que retém parte da precipitação, especialmente nas áreas de folhagem mais densa. Assim, a remoção da cobertura vegetal pode desequilibrar os ciclos hídricos locais, tornando a área mais vulnerável a problemas como erosão e aumento do escoamento superficial.

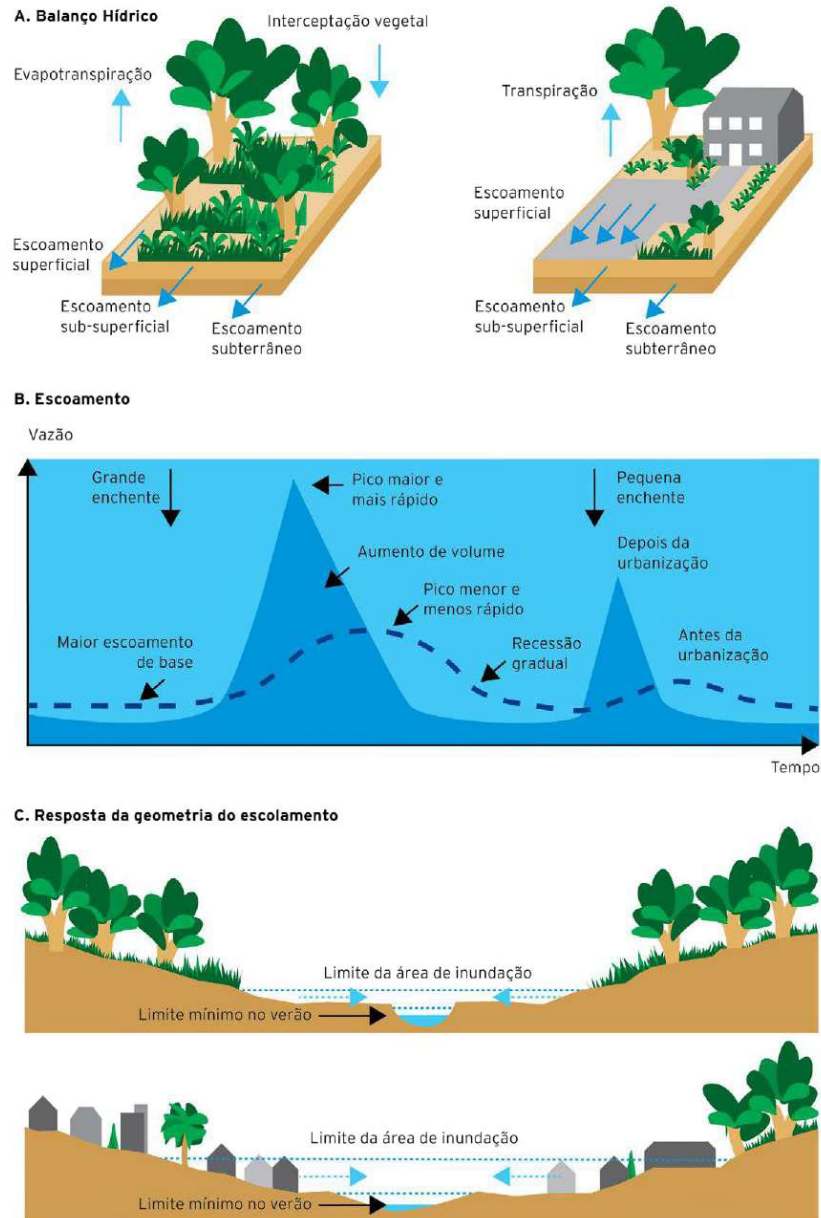
Silva *et al.*, (2019a) retratam que essa impermeabilização do solo é resultado de tentativas do homem de obter-se um maior aproveitamento do solo para atender as necessidades que o crescimento urbano exige.

A falta de área para infiltração e um sistema de drenagem ineficiente aumentam os riscos de alagamentos em dias de chuvas intensas. De acordo com Reis (2015) para que problemas como este sejam evitados deve se projetar uma cidade de acordo com as características naturais daquela área e realizar uma ocupação do solo de forma consciente.

A Figura 7 ilustra o aumento das inundações de acordo com três aspectos, que são: Balanço Hídrico, mostrando o aumento do escoamento superficial ocasionado pela diminuição da área de infiltração; Escoamento, onde o hidrograma registra o aumento da vazão em um intervalo de tempo menor (menor tempo de concentração), quando se comparado a antes do processo de urbanização, ou seja, a água atinge os cursos d'água naturais ou canais de drenagem, em maior volume e com mais velocidade, elevando o risco de enchentes; Resposta da

Geometria do Escoamento, representando a ocupação das várzeas e consequentemente inundações dessas áreas em vazões de cheias (Distrito Federal, 2023).

Figura 7 - Impacto da urbanização no balanço hídrico, escoamento e áreas de inundação



Fonte: Schuler, 1987 *apud* SUDERHSA; CH2M Hill, 2002 *apud* Distrito Federal, 2023.

Segundo Botelho (2017), há três formas em que o processo de urbanização pode acontecer, sendo elas: Focar em uma urbanização que preserve as características naturais da área, minimizando os danos com pequenas intervenções; Desconsiderar as características naturais, mas incluir grandes obras de proteção, minimizando danos, mas elevando custos; Projetar sem se preocupar com as condições naturais do terreno e sem a realização de obras de drenagem, o que pode gerar problemas sérios além de necessitar futuramente de medidas corretivas com valores exorbitantes ou quase impraticáveis.

A necessidade de grandes intervenções urbanas, particularmente em grandes centros, muitas vezes justifica a adoção da terceira alternativa mencionada por Botelho (2017). Essa escolha, embora muitas vezes realizada por razões práticas, como a falta de espaço e a urgência de soluções, geralmente desconsidera fatores ambientais e sustentáveis.

De acordo com Felizardo (2016), a urbanização no Brasil ocorre de forma espontânea, na qual parte da cidade é construída dentro das normas e planos diretores vigentes e outra parte vem originando-se de forma desordenada, como favelas, loteamentos, entre outros. Com isso Cabral e Cândido (2019), evidenciam que o crescimento urbano não ocorre de forma sustentável justamente por essa desorganização, originando vários problemas e afetando o planejamento urbano.

Belo Horizonte (2022), explica que, é importante que faça parte de uma proposta de urbanização de qualquer área, uma avaliação dos lançamentos de águas pluviais e seus efeitos consequentes. Portanto, é necessário conduzir estudos que analisem a viabilidade de implementação de técnicas compensatórias de drenagem para mitigar os impactos causados.

Os efeitos negativos decorrentes do processo de urbanização, quando ocorrido de forma desordenada, são diversos, além das inundações como já citado anteriormente, até a disseminação de doenças devido a poluição das águas nos córregos de abastecimento e/ou falta de saneamento básico.

De acordo com Tucci (2016), é preciso atuar de forma preventiva para que esses problemas sejam extintos ou pelo menos amenizados, sendo assim as medidas de controle já devem ser aplicadas às novas edificações que surgem, ainda segundo ele algumas das alternativas existentes são: limitar a quantidade de água que é direcionado da edificação para a rede pública de drenagem, reter parte dos sedimentos que seriam também destinados à rede pública para que assim reduza a poluição das águas e incentivar a adoção de áreas permeáveis em cada edificação para diminuir o volume de escoamento.

Tucci (2000) analisa a influência da urbanização nas bacias urbanas e conclui que:

Um habitante introduz cerca de 49 m<sup>2</sup> de área impermeável numa bacia, para cada 10% de aumento de área impermeável ocorre cerca de 100% de aumento no coeficiente de escoamento de cheia e no volume de escoamento superficial, apenas o arruamento produz aumento do volume e do coeficiente de escoamento de 260% e para cada 13% de ocupação com área impermeável no lote ocorre aumento de 115% no coeficiente de escoamento. (TUCCI, 2000, p. 67)

Essas estimativas obtidas por Tucci (2000) são preocupantes quando ocorridas sem a infraestrutura necessária para conseguir drenar todo volume de água que é escoado, tendo em vista a grande porcentagem de aumento no escoamento devido ao processo de impermeabilização do solo. Outro fator interessante que pode ser comprovado é o fato de que

apenas a criação das ruas já influencia no aumento do escoamento superficial, assim percebe-se que os loteamentos ainda que em fase inicial de elaboração, já começam a contribuir para o aumento do escoamento em dias de precipitação.

Com todos os problemas decorrentes de um processo de urbanização assim expostos, se faz necessário a utilização de um sistema de drenagem que leve em conta a previsão de população futura de forma assertiva, plano diretor, zoneamento e ter atenção especial para o tempo de retorno e de concentração dos eventos climáticos a partir dos estudos hidrológicos, que apontam que os eventos são cíclicos.

## **2.2 Drenagem Urbana**

Tucci (2005) define drenagem urbana como um conjunto de dispositivos responsáveis por coletar água originada pela precipitação em meios urbanos, realizar seu tratamento e proporcionar meios para o retorno aos rios.

Righetto (2009) explica que, de início o fluxo de água originado da precipitação é escoado de acordo com a topografia do terreno, mas após a criação ou expansão de cidades o caminho percorrido é definido pelo traçado das ruas, podendo ser bem distinto do seu comportamento original.

Juntamente com a delimitação das ruas e toda formação do planejamento urbano, deve ser pensado o sistema de drenagem como ferramenta indispensável para se evitar problemas como inundações e alagamentos. Segundo Silva (2018), quando o sistema de drenagem não é projetado de forma adequada ainda na fase inicial do planejamento urbano, o mesmo certamente terá que ser implantado posteriormente e com um custo bem maior, gerando despesas e transtornos ao município que poderiam ter sido evitadas.

Belo Horizonte (2022) destaca que, o gerenciamento das águas pluviais e fluviais deve ser concebido como um sistema integrado, no qual não há uma solução única prioritária, mas sim uma variedade de medidas e ações em diversas escalas, no qual, as técnicas convencionais desempenham um papel importante e podem até mesmo ser combinadas com abordagens de controle na fonte e intervenções em nível regional.

No planejamento de sistemas voltados para uma drenagem mais sustentável, é comum utilizar dispositivos convencionais de captação e condução, que visam tanto alimentar quanto complementar as técnicas de controle na fonte. O sistema de drenagem é composto por obras de micro e macrodrenagem, e é importante saber suas diferenças para que suas aplicações sejam bem sucedidas.

De acordo com São Paulo (2012), entende-se que fazem parte do sistema de microdrenagem os dispositivos responsáveis por coletar e direcionar a água precipitada, como as bocas de lobo, galerias, vias, sarjetas, incluindo-se também alguns canais desde, que de pequeno porte.

Na macrodrenagem estão compreendidos os canais e/ou estruturas de grande porte. Tucci (2005), afirma que o sistema de microdrenagem visa atender a volumes de precipitação inferiores ao que o sistema de macrodrenagem é projetado para receber.

Medidas adotadas que visam intervenções localizadas e que desconsideram a dinâmica da bacia hidrográfica podem resolver problemas específicos em um determinado ponto, mas têm o potencial de gerar impactos significativos nas áreas seguintes (Distrito Federal, 2023). Essa atitude de acordo com Canholi (2014) está errada, pois assim o problema está apenas sendo transferido para jusante, locais que não apresentavam problemas em relação a drenagem agora certamente sofrerão com inundações.

Reda *et al.*, (2016) ainda cita que quando os transtornos com inundações começam a ser gerados, é comum a adoção de condutos que conseguem coletar e conduzir grandes volumes de água para jusante, mas essa medida tende a aumentar o fluxo d'água, afetando mais ainda os próximos trechos. Outrossim, deve-se atentar ao despejo deste efluente em estruturas adequadas.

É nítido que mudanças, principalmente nos grandes centros urbanos, devem ser aplicadas, se não for possível obter uma medida para eliminar os transtornos, que pelo menos sejam minimizados. Dentre as soluções que podem ser empregadas na drenagem urbana, têm-se duas divisões características de acordo com sua natureza, as medidas estruturais e as não estruturais.

Para Canholi (2014) como medidas estruturais são adotadas obras físicas de engenharia com o objetivo de prevenir e/ou corrigir problemas com inundações. Já as medidas não estruturais podem ser executadas sem a necessidade de grandes obras e sim a adoção de leis, manuais, programas de conscientização da população, soluções de bioengenharia, entre outros.

Obras para captação, armazenamento e transporte de águas precipitadas fazem partes de medidas estruturais que podem ser adotadas, já as medidas não estruturais vão além do objetivo de apenas correção de problemas, elas visam também conscientizar para um melhor uso do ambiente urbano, A fim de proporcionar a população mais qualidade de vida, bem estar, entre outros (Righetto, 2009).

A atitude tomada pelo poder público frente aos problemas e desafios é extremamente importante, mas a população também deve exercer seus deveres para poderem reivindicar pelos

seus direitos. Nesse sentido, Impulcetto (2017) destaca que é de extrema importância a união entre os gestores e a população para que as medidas não estruturais sejam executadas com sucesso e sem transtornos.

É fundamental que o poder público conte com uma equipe técnica qualificada para serem responsáveis pelos projetos, execução e fiscalização das obras de drenagem, a fim de evitar gastos desnecessários, retrabalho e garantir que o sistema projetado funcione perfeitamente. A adoção da metodologia BIM (*Building Information Modeling*) e o conceito CIM (*City Information Modeling*) são grandes aliados para mitigação destes fatos, especialmente com o suporte de regulamentações recentes. O Decreto nº 11.888, de 22 de janeiro de 2024, que revogou os decretos anteriores nº 9.983 de 2019 e nº 9.377 de 2018, tem por objetivo criar condições favoráveis para o investimento e a ampla adoção do BIM no País (Brasil, 2024a).

Ademais, o Ministério do Desenvolvimento, Indústria, Comércio e Serviços lançou o curso "Democratizando BIM: agentes públicos", com foco na capacitação de servidores públicos para a adoção do BIM em projetos e obras públicas. Essa ação busca fortalecer o conhecimento técnico e ampliar a utilização do BIM, alinhando-se às metas de modernização e eficiência no setor público (Brasil, 2024b).

As iniciativas mencionadas acima têm grande potencial para mitigar um problema identificado por Cruz, Souza e Tucci (2007), que destaca a falta de um setor dedicado exclusivamente à gestão do sistema de drenagem urbana. Frequentemente, essa é atribuída ao setor de obras do município, o que resulta em planejamento inadequado, já que a coordenação dessas atividades, na maioria das vezes, fica a cargo de técnicos que não possuem formação específica na área de drenagem.

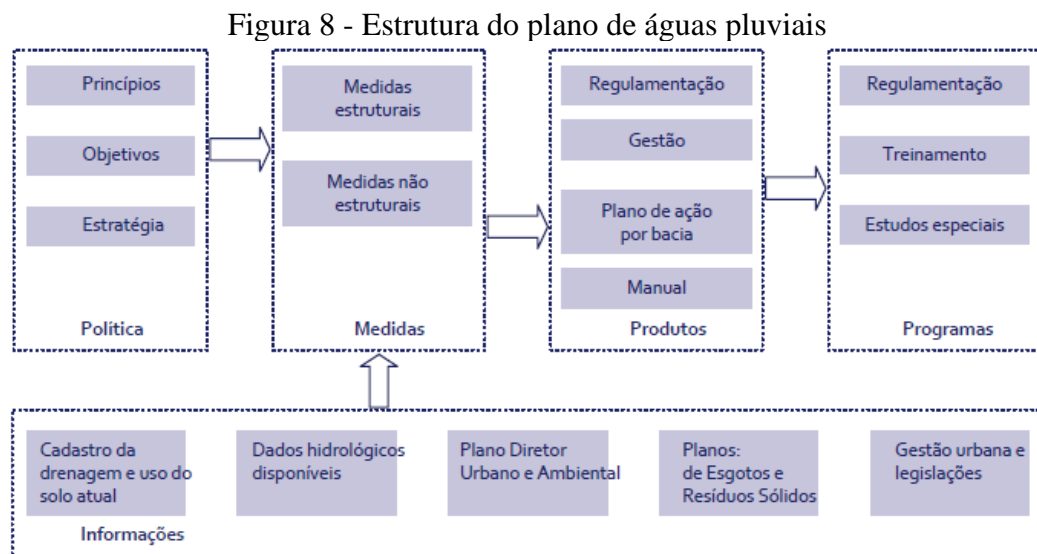
Um instrumento que auxilia os gestores e conduz a sociedade ao uso correto do espaço urbano é o Plano Diretor que de acordo com a Lei 10.257 de 10 de julho de 2001 estabelece como principal critério para implantação de um Plano Diretor todo município que tiver mais de vinte mil habitantes (Brasil, 2001).

Segundo Martins e Fornari (2019), o Plano Diretor de um município é composto por ações que ajudam no desenvolvimento urbano, a fim de solucionar ou evitar os problemas decorrentes de uma urbanização irregular. Pedrosa *et al.*, (2021) reitera que o Plano estabelece orientações para o uso, a ocupação e o parcelamento do solo, buscando garantir uma distribuição equilibrada dos equipamentos urbanos e da infraestrutura essencial, com especial atenção para os sistemas de drenagem.

Em Piumhi - MG a Lei complementar n° 67, de 25 de setembro de 2019 estabelece normas e condições para parcelamento, ocupação e uso do solo urbano no município. Dentro dessas normas, a Taxa de Permeabilidade é um conceito importante que se refere à área de um terreno que é permeável, geralmente coberto por vegetação (Piumhi - MG, 2019).

Essa lei permite que, as edificações em áreas urbanas com exceção das localizadas em zonas de proteção ambiental, substituam até 50% da área permeável por sistemas de captação e drenagem, desde que esses sistemas sejam capazes de reter o volume equivalente de água que seria absorvido pela área permeável remanescente. As taxas de permeabilidade variam de acordo com as zonas pré-definidas, dentre elas destaca-se a taxa mínima de 20% para zonas de adensamento e 90% para zonas de proteção ambiental (Piumhi - MG, 2019).

Como o processo de urbanização está relacionado com a expansão da rede de drenagem urbana, outra ferramenta que segundo Tucci (2012) deve ser criada juntamente com o Plano Diretor e o de saneamento, é o Plano Diretor de Drenagem Urbana (PDDrU) ou Plano Diretor de Manejo de Águas Pluviais. A estrutura desse plano está contida na Figura 8.



Fonte: Tucci, 2012, adaptado de Tucci, 2007.

O Plano Diretor de Drenagem Urbana é de extrema importância para gerenciar a distribuição do escoamento superficial gerado pelas águas pluviais, sempre acompanhando o crescimento urbano e o sistema de drenagem urbana, a fim de combater ou pelo menos minimizar os problemas gerados por inundações, que afetam a economia e o meio ambiente (Righetto, 2009).

Outro fator que torna o Plano Diretor de Drenagem Urbana ainda mais essencial é que segundo Brasil (2023), o município que solicitar recursos da União para obras ou atividades de

melhorias no sistema de drenagem urbana, deve apresentar o seu plano e leis municipais relacionadas, a fim de garantir que as ações a serem aplicadas tenham sido planejadas.

Rocha (2014), evidencia que cada município tem suas necessidades, podendo ser diferentes ou não, sendo assim cada um destes devem elaborar o seu próprio PDDrU de acordo com as características da região.

### 2.2.1 Área de estudo

Segundo Belo Horizonte (2022), no processo de delimitação da bacia, deve ser respeitado as características físicas e geográficas do solo, além da configuração das vias, que influenciam o fluxo da água. Também é crucial considerar as áreas de contribuição, que mesmo estando fora da delimitação, possam vir a contribuir com o aumento do escoamento futuramente. Para a definição dos divisores de bacias, é necessário ter conhecimento sobre as direções do escoamento e a disposição das quadras, edificações e vias, tanto as existentes quanto as planejadas.

### 2.2.2 Intensidade máxima de precipitação

De acordo com Tucci (2001), precipitação máxima é um acontecimento extremo e que é caracterizado por três grandezas (curvas IDF), que são: intensidade, duração e frequência, sendo de extrema importância seu estudo para conhecer a vazão de enchente em uma bacia.

Petrucci e Oliveira (2019) ainda destacam que quando estes altos volumes de precipitação não são considerados em projetos de drenagem urbana, por exemplo, podem ocorrer vários problemas.

Dias e Antunes (2010) destacam que, caso o local onde será projetado o sistema de drenagem urbana não possua dados pluviométricos disponíveis, pode-se utilizar informações de outros locais desde que possuam características meteorológicas semelhantes.

Para o cálculo da intensidade pluviométrica são necessários alguns dados que podem ser obtidos por fórmulas matemáticas e empíricas ou consultados em tabelas, sendo eles, o tempo/período de retorno ou tempo/período de recorrência, tempo de concentração e demais parâmetros.

Para Tucci (2001) a variável tempo de retorno (T), está associado com a possibilidade da ocorrência de problemas que a obra pode gerar mesmo depois de pronta e que serão sentidos pelos usuários. Ou seja, quanto maior for o tempo de retorno determinado, maior a segurança

por um período de tempo mais longo. Os valores podem ser acessados através dos Quadros 1 e 2.

Quadro 1 - Períodos de retorno para diferentes ocupações

<b>Tipo de ocupação da área</b>	<b>Período de retorno (anos)</b>
Residencial	2
Áreas comerciais	5
Áreas com edifícios público	5
Aeroportos	2 - 5
Áreas comerciais altamente valorizadas e terminais aeroportuários	5 - 10

Fonte: CETESB, 1986.

Quadro 2 – Período de recorrência.

<b>Espécie</b>	<b>Período de recorrência (anos)</b>
Drenagem superficial	5 a 10
Drenagem subsuperficial	10
Bueiros Tubulares	15 (como canal)
	25 (como orifício)
Bueiro Celular	25 (como canal)
	50 (como orifício)
Pontilhão	50
Ponte	100

Fonte: DNIT, 2006a.

O tempo de concentração é definido como o intervalo necessário para que a água da chuva percorra a distância entre o ponto mais distante da bacia hidrográfica e o local de interesse. Também pode ser entendido como o intervalo de tempo entre início da precipitação e o momento em que toda a área da bacia já esteja contribuindo para o escoamento (DNIT, 2005). Em bacias urbanas ele é definido por meio da somatória dos tempos de concentração dos diversos trechos existentes, sendo que, nos trechos iniciais geralmente é adotado como 10 min. Porém, esse valor pode ser superestimado para bacias muito impermeáveis ou com grande declividade, sendo recomendado o planejamento mais detalhado nessas situações (Tucci, 2001). Já para DNIT (2006a), pode ser usado um tempo de concentração constante igual a 5 min para as obras de drenagem superficial.

Segundo Pinto e Junior (2017) os parâmetros “k”, “a”, “b” e “c”, variam de acordo com o regime pluviométrico local e podem ser determinados tanto por estações pluviométricas da região quanto por bancos de dados existentes. O software PLUVIO 2.1, criado pelo Grupo de Pesquisas em Recursos Hídricos (GPRH) da Universidade Federal de Viçosa, disponibiliza esses parâmetros para diversas regiões do Brasil (Fiorio, 2012).

Os parâmetros **k**, **a**, **b** e **c** citados são coeficientes empíricos utilizados em modelos matemáticos para descrever o comportamento do regime pluviométrico de uma região. Esses

coeficientes são ajustados com base em séries históricas de dados pluviométricos e refletem características específicas das chuvas, como intensidade, duração e frequência.

- **k (constante de escala):** Relaciona-se com a intensidade média das chuvas, ajustando o modelo às condições climáticas gerais da região.
- **a, b, e c (parâmetros de forma):** Ajustam a curvatura e o comportamento da relação intensidade-duração-frequência (IDF) da precipitação.
  - **a** pode estar ligado à inclinação inicial da curva IDF.
  - **b** ajusta a forma da curva, refletindo como a intensidade varia com a duração.
  - **c** é frequentemente um parâmetro de correção ou deslocamento para ajustar discrepâncias menores.

### 2.2.3 Vazão de chuva

O uso do Método Racional em bacias com áreas de até 2 km<sup>2</sup>, 3 km<sup>2</sup> ou até 5 km<sup>2</sup> varia conforme os critérios e objetivos dos autores, sendo influenciado pelas características hidrológicas e geomorfológicas da região em análise. Para áreas menores, como até 2 km<sup>2</sup>, o método Racional é mais adequado, pois pressupõe uniformidade espacial e temporal da precipitação, condições frequentemente verificadas em bacias pequenas e urbanizadas, onde há alta impermeabilização (Christofidis, 2010). Já para áreas um pouco maiores, como entre 3 e 5 km<sup>2</sup>, sua aplicação é justificada em contextos específicos, onde outros métodos podem ser menos práticos ou os dados disponíveis são insuficientes para abordagens mais complexas. No entanto, em bacias maiores, as limitações do método tornam-se mais evidentes, dado que fatores como a variabilidade espacial da precipitação e os efeitos de armazenamento na bacia não são devidamente considerados (Benites *et al.*, 2017; São Paulo, 1999).

Portanto, a escolha de limites para a aplicação do Método Racional reflete um compromisso entre a simplicidade do modelo, as condições locais da bacia e a disponibilidade de dados, sendo que para bacias maiores ou com características mais complexas, recomenda-se o uso de métodos alternativos, como o Hidrograma Unitário Sintético ou os Métodos Estatísticos descritos pelo DNIT (2005).

Uma das variáveis utilizadas nesse método é o coeficiente de escoamento superficial, que, segundo Tucci (2000), representa a relação entre o volume de água precipitada e o volume escoado superficialmente, sendo que seu valor geralmente é determinado por consultas em tabelas. Sendo assim esse coeficiente está diretamente ligado a taxa de urbanização e o tipo de cobertura do solo na área em estudo, pois quanto mais urbanizado for o local, a tendência é que

o volume escoado seja maior devido a baixa capacidade de infiltração e conseqüentemente maior será o coeficiente de escoamento. Os Quadros 3 e 4 apresentam esses coeficientes para tempos de retorno variando de cinco a 10 anos (DNIT, 2005).

Quadro 3 - Coeficiente de Escoamento Superficial / *Run-Off* de acordo como tipo de ocupação e de bacia.

DESCRIÇÃO DAS ÁREAS DAS BACIAS TRIBUTÁRIAS	COEFICIENTE DE DEFLÚVIO "C"
Comércio	
Áreas Centrais	0,70 a 0,95
Áreas da periferia do centro	0,50 a 0,70
Residencial	
Áreas de uma única família	0,30 a 0,50
Multi-unidades, isoladas	0,40 a 0,60
Multi-unidades, ligadas	0,60 a 0,75
Residencial (suburbana)	0,25 a 0,40
Área de apartamentos	0,50 a 0,70
Industrial	
Áreas leves	0,50 a 0,80
Áreas densas	0,60 a 0,90
Parques, cemitérios	0,10 a 0,25
Playgrounds	0,20 a 0,35
Pátio e espaço de serviços de estrada de ferro	0,20 a 0,40
Terrenos baldios	0,10 a 0,30

Fonte: DNIT, 2005.

Quadro 4 - Coeficiente de Escoamento Superficial / *Run-Off* por tipo de superfície.

TIPO DE SUPERFÍCIE	COEFICIENTE DE DEFLÚVIO "C"
Ruas	
Asfalto	0,70 a 0,95
Concreto	0,80 a 0,95
Tijolos	0,70 a 0,85
Trajeto de acesso a calçadas	0,75 a 0,85
Telhados	0,75 a 0,95
Gramados; solos arenosos	
Plano, 2%	0,05 a 0,10
Médio, 2 a 7%	0,10 a 0,15
Íngreme, 7%	0,15 a 0,20
Gramados; solo compacto	
Plano, 2%	0,13 a 0,17
Médio, 2 a 7%	0,18 a 0,22
Íngreme, 7%	0,15 a 0,35

Fonte: DNIT, 2005.

De acordo com São Paulo (1999), ao se aderir a um coeficiente de escoamento constante é necessário estar atento a alguns fatores, que são:

- Levar em consideração um futuro aumento populacional e conseqüentemente maior taxa de urbanização, o que pode alterar as condições de cobertura do solo;

- Quando a bacia apresentar áreas com diversos tipos de cobertura do solo distintos, é recomendado calcular o coeficiente através de uma média ponderada, segundo a Equação 1;

$$C = \frac{1}{A} \times \sum C_i \times A_i \quad (1)$$

Onde  $C_i$ : coeficiente de escoamento superficial correspondente à ocupação  $i$  e  $A_i$ : área da bacia correspondente à ocupação  $i$ .

- Para tempos de retorno maiores que 10 anos, é importante realizar um ajuste no valor do coeficiente, através da Equação 2.

$$C_T = 0,8 \times T^{0,1} \times C_{10} \quad (2)$$

Onde  $C_T$ : coeficiente de escoamento superficial para tempo de retorno  $T$  (anos) e  $C_{10}$ : coeficiente de escoamento superficial para tempo de retorno de 10 anos.

Tempos de retorno maiores que 10 anos podem ser utilizados em obras de bueiros (10 a 20 anos) e pontes (50 a 100 anos) (DNIT, 2005).

#### 2.2.4 Meio Fio

De acordo com Guedes (2015), meio fio é um componente do sistema de drenagem, podendo ser de concreto ou pedra e localizados entre a via e a calçada, sendo que sua superfície superior deve estar no mesmo nível da calçada.

Em áreas urbanas o meio fio precisa ter uma altura mínima de 10 cm e máxima de 15 cm, sendo que em alguns casos encontra-se com altura de 20 cm, mas não é indicado, pois pode acarretar problema ao abrir a porta do carro e trazer incômodo aos pedestres (São Paulo, 2012).

#### 2.2.5 Sarjeta

As sarjetas são, tradicionalmente, estruturas triangulares localizadas ao lado do meio-fio, na borda da via, tendo como função coletar e direcionar as águas pluviais em direção às bocas de lobo. Sua declividade comumente é adotada igual à declividade longitudinal da via (Belo Horizonte, 2022).

Tucci (2001) destaca que é extremamente importante definir a vazão máxima que a sarjeta é capaz de suportar a fim de se evitar problemas que possam causar transtornos a população, como, alagamentos de vias e calçadas, erosões, caso a velocidade do fluxo de água exceda os valores aceitáveis, entre outros.

Segundo Guedes (2015), é importante analisar e comparar a capacidade de esgotamento da sarjeta com a vazão máxima de escoamento, sendo que, caso a vazão seja superior a sua capacidade é necessário à instalação de bocas de lobo para drenar a quantidade de água excedida.

Há duas possibilidades de se calcular a capacidade da sarjeta, a primeira seria considerar que o fluxo de água seja conduzido apenas através da sarjeta com a altura da lâmina d'água atingindo até 10 cm e a segunda hipótese considera que o fluxo de água escoe também pela calha da rua adotando que a altura da lâmina d'água não supere 15 cm, em ambas as hipóteses se considera que a declividade da seção transversal da via seja de 3% (Tucci, 2001).

Ao calcular a capacidade de escoamento da sarjeta é importante realizar a escolha correta do coeficiente de rugosidade ou de Manning, de acordo com o tipo de material da sarjeta. Alguns desses coeficientes estão presentes na Tabela 1 e no Quadro 5.

Tabela 1 – Valores dos coeficientes de Manning ( $n$ )

Cobertura da bacia	$n$
Asfalto suave	0,012
Asfalto ou concreto	0,014
Argila compactada	0,030
Pouca vegetação	0,020
Vegetação densa	0,350
Vegetação densa e floresta	0,400

Fonte: TUCCI, 2001.

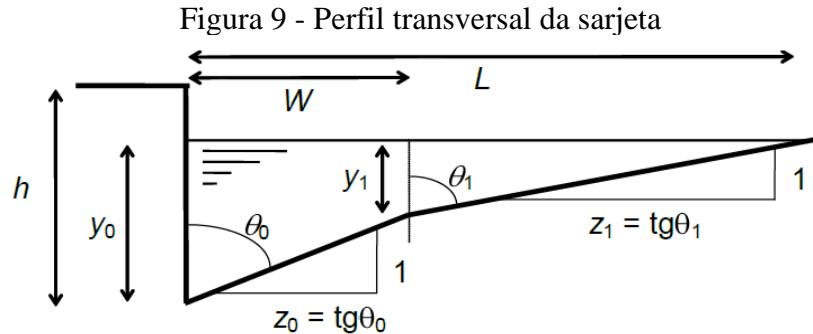
Quadro 5 – Coeficiente de rugosidade (Manning)

Natureza do revestimento do canal	Coeficientes				Velocidades limites (m/s)
	Ótimo	Bom	Regular	Mau	
Concreto de Cimento e de Alvenaria	0,017	0,020	0,025	0,030	4,0
Pedras grandes e lisas	0,025	0,030	0,033	0,035	2,5
Pedras irregulares	0,025	0,030	0,033	0,035	2,5
Terra enleivada (com vegetação)	0,025	0,030	0,033	0,035	1,2 – 1,8
Terra emperrada	0,028	0,030	0,033	0,035	0,5 – 1,8

Fonte: DNIT, 2006b.

A Figura 9 ilustra o perfil transversal de uma sarjeta, destacando as variáveis geométricas e parâmetros relacionados ao cálculo da capacidade de escoamento da água. Nessa representação, observa-se a altura da lâmina d'água indicada por  $y_0$ , bem como os ângulos de inclinação da sarjeta ( $\theta_0$ ) e da calha ( $\theta_1$ ), fundamentais para determinar a declividade e o comportamento do escoamento ao longo da superfície. A largura da sarjeta ( $W$ ) e o comprimento da calha ( $L$ ) são considerados para analisar a capacidade de drenagem. Esta

estrutura servirá de base para aplicar as fórmulas de capacidade em ambas as hipóteses apresentadas.



Fonte: Lucas, 2019 adaptado de Barreiro, 1997.

Conforme exposto por Lucas (2019), é importante fazer um ajuste na máxima vazão que pode ser escoada pela sarjeta, multiplicando seu resultado por um fator de redução da sua capacidade, a fim de garantir uma maior segurança no dimensionamento devido a alguns fatores que podem ocorrer após finalização como, desgaste da sarjeta devido alta velocidade da água em locais com elevada declividade, acúmulo de sedimentos e/ou lixo em trechos de baixa declividade, entre outros. Os fatores estão presentes no Quadro 6.

Quadro 6 - Fatores de redução de escoamento das sarjetas

Declividade da sarjeta (%)	Fator de redução
0,4	0,50
1 - 3	0,80
5	0,50
6	0,40
8	0,27
10	0,20

Fonte: São Paulo, 2012, *apud* adaptado de DAEE CETESB, 1980.

Outro fator importante a se analisar é a velocidade de escoamento nas sarjetas. Macedo (2020) destaca que existem limites recomendáveis a serem respeitados, porém verifica-se que esses valores possuem variações de acordo com o manual de drenagem de cada estado, dentre os manuais analisados o limite mínimo encontrado foi de 0,75 m/s e o máximo de 4,0 m/s. Já Belo Horizonte (2022), não definiu uma velocidade mínima, mas determina uma máxima de 3,0 m/s. Enquanto DNIT (2006b), estabelece algumas velocidades máximas de acordo com o material da sarjeta, podendo ser consultadas no Quadro 5.

Deve-se ficar atento quanto aos benefícios e malefícios que podem ser gerados dependendo dos limites utilizados. Cauduro, Müller e Rosa (2019) explicam que velocidades menores aumentam a durabilidade da sarjeta, além de diminuir a possibilidade de ocorrência de inundações e consequentemente uma velocidade muito alta podem causar efeito contrário.

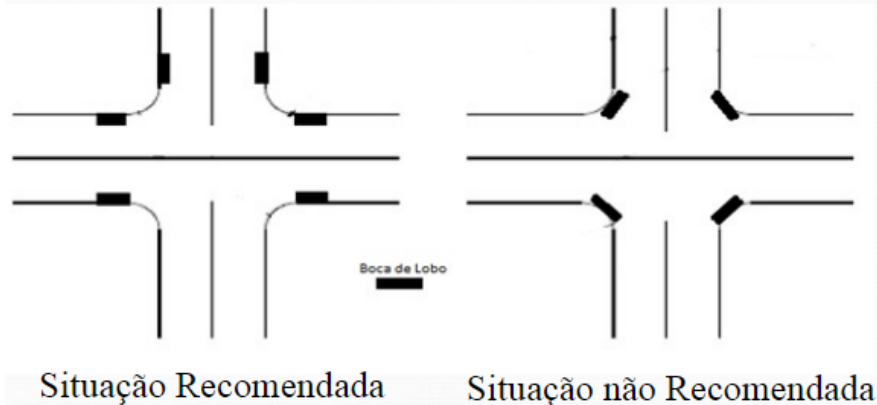
### 2.2.6 Bocas de Lobo

São Paulo (2012), defini as bocas de lobo como, dispositivos instalados nas sarjetas, com a função de coletar as águas que passam pela mesma e direcionar esse fluxo até as galerias ou tubulações subterrâneas que a transportam até os rios.

Tucci (2001) destaca algumas orientações sobre a localização das bocas de lobo: a) devem ser instaladas em ambos os lados da via quando a sarjeta estiver saturada ou exceder sua capacidade de drenagem; b) devem ser posicionadas nos pontos mais baixos da área; c) caso não seja avaliada a capacidade suportada pela sarjeta é recomendado manter um espaçamento máximo de 60 metros entre as bocas de lobo; d) a instalação ideal das bocas de lobo é próxima aos cruzamentos frequentados por pedestres, nas esquinas e ligeiramente montante da faixa de travessia; e) não é aconselhável colocá-las nos vértices de interseção das sarjetas devido aos seguintes motivos: os pedestres teriam dificuldade para atravessar a rua, tendo que enfrentar a água em alta velocidade e o fluxo convergente das sarjetas resultaria em um escoamento contrário ao da entrada na boca de lobo.

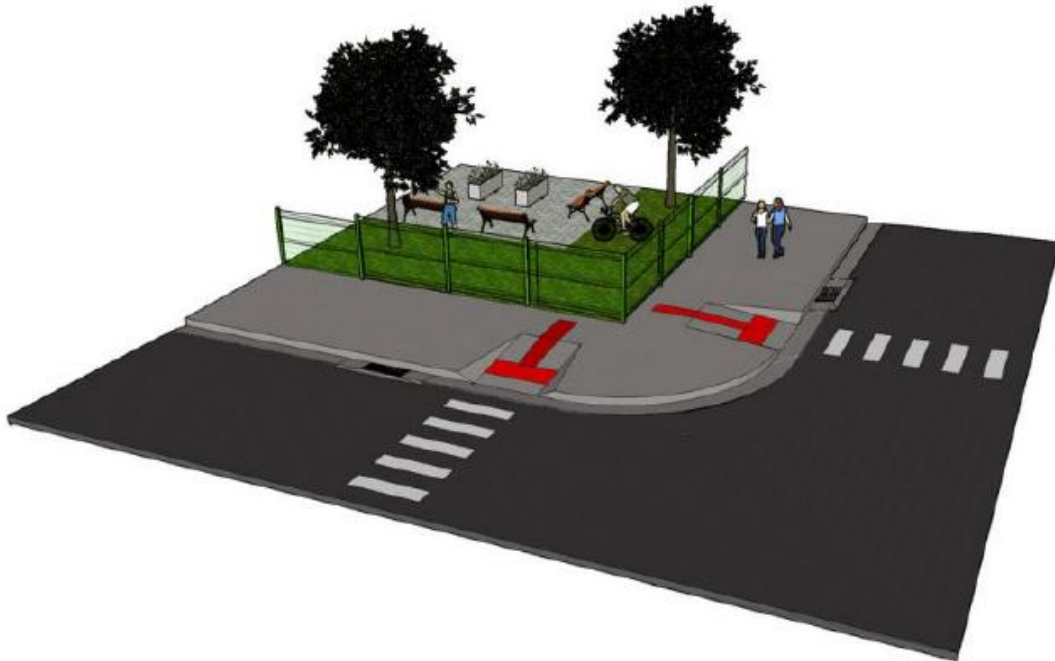
As Figuras 10 e 11 ilustram a locação correta das bocas de lobo em cruzamentos.

Figura 10 – Comparativo de bocas de lobo em cruzamentos de vias



Fonte: Guedes, 2015.

Figura 11 – Bocas de lobo em cruzamentos de vias



Fonte: Belo Horizonte, 2022.

De acordo com o Manual de Drenagem Urbana de São Paulo (2012), o posicionamento das bocas de lobo deve ser feito de forma a direcionar eficientemente as águas superficiais para as galerias, além disso, nos pontos mais baixos da via, é essencial que as bocas de lobo instaladas tenham acesso para manutenção, a fim de prevenir a formação de áreas com alagamentos e água parada. Ainda sobre a locação das bocas de lobo, Distrito Federal (2023), descreve que, as distâncias entre elas, deve ser determinada considerando a vazão da área de contribuição, a largura máxima de alagamento permitida na via e a altura do meio fio.

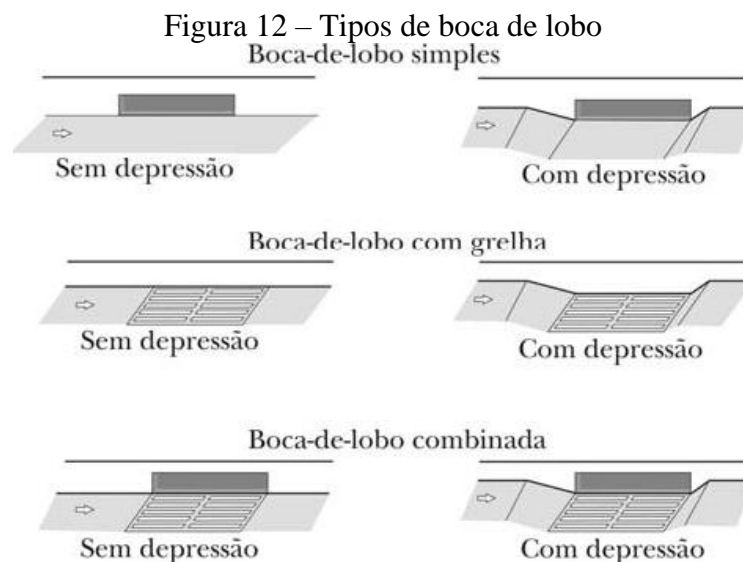
Segundo Guedes (2015), as bocas de lobo podem estar localizadas em dois pontos das sarjetas. Ponto intermediário, quando o trecho for contínuo e possuir declividade constante, o fluxo de água entra apenas por um lado da boca de lobo. Ponto baixo, utilizado quando ocorre uma mudança da declividade da via ou próximo de cruzamentos, neste caso ocorre entrada de água por dois lados da boca de lobo.

A CETESB (1986), definiu quatro tipos de boca de lobo, sendo elas: Boca de lobo simples ou de guia, boca de lobo com grelha, boca de lobo combinada e boca de lobo múltipla. A seguir serão detalhadas e ilustradas pela Figura 12 as três primeiras, que são utilizadas neste trabalho.

1. Boca de lobo simples ou de guia, aplicada principalmente em pontos baixos das sarjetas, possui uma abertura vertical na guia, chamada guia chapéu, permitindo a entrada da água escoada. Sua capacidade de esgotamento é influenciada pela rapidez com que a água muda de direção na sarjeta, o que depende da declividade transversal da sarjeta, ou seja, criar uma depressão junto à face da guia aumentaria essa declividade e conseqüentemente a

capacidade de escoamento. Pode-se destacar como vantagem, a diminuição de obstrução por detritos e sua desvantagem está relacionada à baixa capacidade de esgotamento quando utilizada em sarjetas que possuem grande declividade longitudinal.

2. Boca de lobo com grelha, sua abertura é constituída de barras metálicas perpendiculares (grelhas transversais) ou paralelas (grelhas longitudinais) em relação ao escoamento. Sua capacidade de esgotamento sobre aumento quando utilizado com depressão, porém não é viável, pois pode afetar o tráfego de veículos, já utilizá-las com grelhas longitudinais também pode aumentar sua capacidade e diminuir o acúmulo de resíduos sobre as grelhas. Mas se comparada ao tipo de boca de lobo anterior, o uso das grelhas também gera uma desvantagem, pois, aumenta a área de contato disponível para o acúmulo de resíduos, diminuindo então sua eficiência.
3. Boca de lobo combinada é um conjunto de bocas de lobo, constituído de uma simples e uma de grelha, todas as características mencionadas anteriormente são válidas para este tipo. No seu funcionamento em conjunto, a grelha é a maior responsável pela capacidade de drenagem quando não estiver obstruída, porém se houver obstrução, a abertura na guia se torna indispensável para o correto funcionamento, outra maneira que pode tornar mais eficiente, é locando a grelha a jusante da guia chapéu.

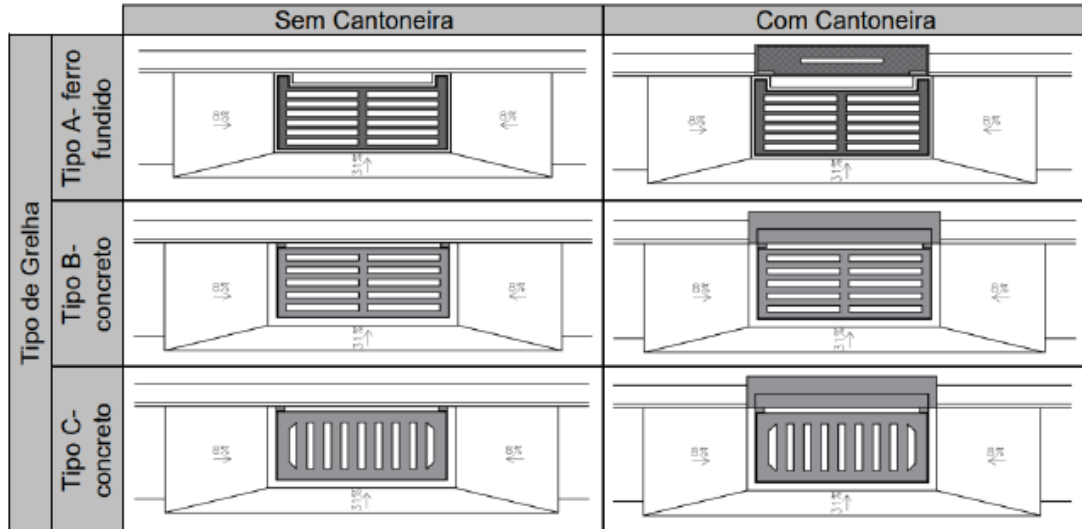


Fonte: Coelho e Lima, 2011, adaptado de CETESB, 1980.

A Figura 13 ilustra três tipos de bocas de lobo com grelha, onde elas se diferem pelo material das grades ou sentido das mesmas, além da sua utilização com ou sem cantoneira, que é a abertura na guia. As bocas de lobo tipo grelha com barras longitudinais serão sempre mais indicadas, com exceção de quando trazer perigo, como por exemplo, em ciclovias. Embora

existam bocas de lobo sem cantoneiras, seu uso é indicado, pois a obstrução por detritos pode reduzir a capacidade de escoamento das grades (Belo Horizonte, 2022).

Figura 13 – Bocas de lobo tipo grelha



Fonte: Belo Horizonte, 2022.

*Nota:* Inclinações das depressões paralelas e perpendiculares ao meio fio iguais a 8% e 31% respectivamente.

Para definir a capacidade real das bocas de lobo é necessário utilizar um fator de redução que depende da localização e do tipo da mesma, conforme mostra o Quadro 7. São Paulo (2012) explica que essa redução se faz necessário, tendo em vista que pode ocorrer, obstrução na entrada das mesmas por acúmulo de resíduos, defeitos ou imperfeições na via e/ou na sarjeta, diferenças entre o projeto e a execução. Todos esses detalhes podem afetar a capacidade de esgotamento das bocas de lobo.

Quadro 7 – Fator de redução das bocas de lobo

Localização nas sarjetas	Tipo de Boca de Lobo	Fator de redução
<b>Ponto baixo</b>	Simple	0,80
	Com grelha	0,50
	Combinada	0,65
<b>Ponto intermediário</b>	Simple	0,80
	Com grelha na horizontal	0,60
	Grelha transversal ou longitudinal com barras transversais	0,50
	Combinadas (grelha com barras longitudinais)	0,66
	Combinadas (grelha com barras transversais)	0,55

Fonte: CETESB, 1980, adaptado pelo autor, 2024.

### 2.2.7 Poços de Visita (PV's)

PV's são caixas subterrâneas, cuja tampa deve ser removível e alinhada com a superfície da via ou calçada, de modo a permitir o acesso para limpeza e/ou manutenção. São instalados em locais em que há mudança de diâmetro e/ou de direção da rede coletora ou degraus (Figura 14), além de pontos onde ocorre união da tubulação de dois ou mais trechos e/ou ramais de ligação (Distrito Federal, 2023).

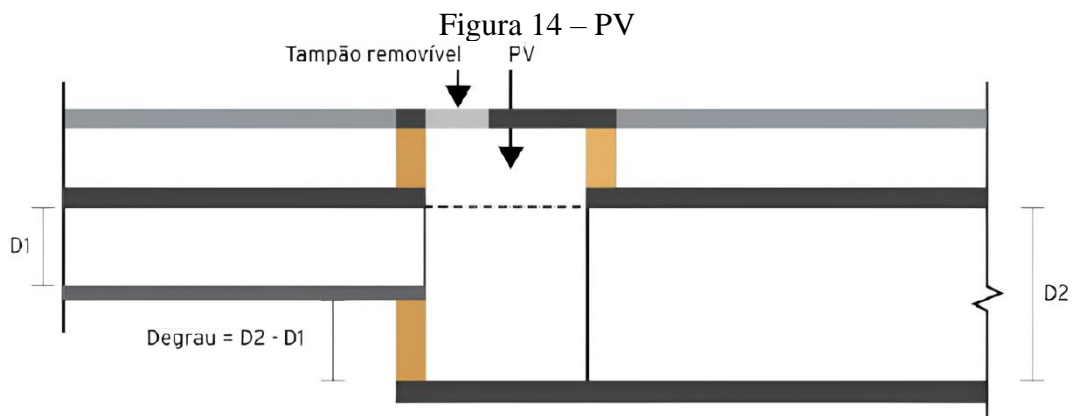
Tucci (2001), também cita o uso de poços de visita para locais que possuem variações de declividade, além disso, deve-se considerar um espaçamento máximo entre eles, representado na Tabela 2. Outro fator importante também citado é que, quando a diferença de nível entre as tubulações de entrada e saída ultrapassa 0,70 m, o poço passa a ser denominado "poço de quebra".

Tabela 2 – Espaçamentos entre os poços de visita

Diâmetro ou altura do conduto (m)	Espaçamento (m)
0,30	120
0,50 - 0,90	150
1,00 ou mais	180

Fonte: Tucci, 2001, adaptado de CETESB, 1980.

A Figura 14 ilustra um PV instalado em um ponto onde ocorre a mudança de diâmetro da rede.



Fonte: Distrito Federal, 2023.

## 2.3 Sistema de Informações Geográficas (SIG) ou Geographic Information System (GIS)

Tao (2013) explica que de maneira geral o GIS é uma ferramenta muito importante porque além de receber os dados também é possível realizar a manipulação dos mesmos,

facilitando a descoberta dos pontos problemáticos e conseqüentemente a identificação das soluções apropriadas.

Fitz (2008), defini SIG como:

Um sistema constituído por um conjunto de programas computacionais, o qual integra dados, equipamentos e pessoas com o objetivo de coletar, armazenar, recuperar, manipular, visualizar e analisar dados espacialmente referenciados a um sistema de coordenadas conhecido. (Fitz, 2008, p. 23)

Segundo Macedo, Mendes e Costa (2018), o SIG foi projetado para funcionar com a função de auxiliar no gerenciamento de informações, onde o usuário, ao entender suas funcionalidades, pode utilizá-lo para o levantamento de hipóteses e após análises, essas poderão ser descartadas ou confirmadas como relevantes para o estudo em questão.

Os Sistemas de Informação Geográfica (SIG's) permitem coletar, armazenar, analisar e visualizar dados georreferenciados, ajudando a compreender padrões e tendências ao vincular informações descritivas a locais específicos (Marchiori, Novaes e Silva, 2021).

Uma das principais vantagens de implementar a ferramenta SIG destacada por Fernandes (2016), é que ela possibilita aos gestores meios para agirem de maneira preventiva e realizarem o controle da gestão, de maneira com que o usuário que manuseia a ferramenta consiga captar o que está para acontecer, permitindo, se necessário, antecipar esses acontecimentos no ambiente que ocorre a análise. Corrêa (2013) ainda complementa que, com os crescentes avanços na tecnologia SIG, juntamente com a implantação da sua metodologia em projetos de drenagem, têm contribuído para a integração entre os sistemas de prevenção contra inundações e os estudos relacionados a problemática.

Este recurso é de extrema importância para aplicação no contexto em que está sendo estudado, pois a utilização de sistemas como estes que permitem monitorar os volumes de precipitações antes mesmo de ocorrerem, passando mais segurança à população de forma com que as pessoas consigam agir de forma rápida diante dos problemas e também para que os gestores consigam monitorar as áreas de maiores riscos, atuando de maneira pontual.

Dentre tantos recursos que são oferecidos pelo GIS, um que é muito utilizado em várias áreas de estudos são os mapas. Esses oferecem a possibilidade de criar diferentes formas de visualizá-los e dentro deles realizar alterações, análises e obter informações. Como os mapas são interativos, os usuários podem usá-los para definir qualquer informação que auxiliará na execução de tarefas.

Como exemplo de aplicação vale citar um exemplo descrito por Shahrour *et. al.* (2017), onde foi analisado o desempenho do SIG aplicado em uma Cidade Inteligente (*Smart City*). O

emprego do SIG exemplificado seguiu as seguintes etapas, concluído o lançamento dos dados coletados no Software Arc GIS, foi feito um monitoramento da rede de água e esgoto a fim analisar o controle de qualidade e promover melhorias delas, além do monitoramento da rede elétrica que mostrou parâmetros de corrente, tensão e frequência.

Após estes processos os usuários têm acesso em tempo real, aos parâmetros obtidos em formas de gráficos e estatísticas, com o auxílio de um aplicativo. Os autores concluíram que o experimento foi de muito sucesso e que puderam visualizar o quão poderoso e útil é a ferramenta SIG.

A ferramenta SIG sem dúvidas surgiu para auxiliar em processos que antes demandavam muito tempo para ser tomado uma decisão por parte do gestor, mas mesmo com esse auxílio ainda é preciso a presença de um técnico capacitado na área de estudo para saber decidir pela melhor solução final e para coletar os dados necessários para o SIG.

A cidade de Curitiba, reconhecida como a mais inteligente e conectada do Brasil no *Ranking Connected Smart Cities 2022*, é um exemplo de como a implementação de tecnologias avançadas, como o BIM (*Building Information Modeling*), pode transformar o planejamento urbano. Curitiba está aprimorando seu sistema de mobilidade e expandindo o uso de tecnologias digitais no desenvolvimento urbano. Desde 1980, a cidade passou a adotar sistemas mais modernos e, em 2018, iniciou um projeto de implementação do BIM, com investimentos em treinamentos e aquisição de novos equipamentos e softwares, com o objetivo de propagar o BIM nas diversas áreas da gestão municipal (Fiorin, 2023).

Costa *et al.* (2018) ainda cita que uma grande vantagem de utilização do SIG são os monitoramentos que podem ser feitos à distância, não obrigando a presença de um técnico em campo para realizar tal função, conseqüentemente contribui para amenizar erros que poderiam surgir devido a falta de imagens atuais ou com pouca quantidade de detalhes sendo possível extrair poucas informações do local em estudo.

O SIG tem sido utilizado em diferentes áreas para suprir diversas necessidades, dentre elas pode-se citar o monitoramento de cheias em microbacias urbanas. A ferramenta foi aplicada no município de São Carlos - SP onde foi possível simular vários perfis de alagamentos. Felizardo (2016) que descreveu todo processo de simulação, concluiu que, com todas as informações extraídas através do auxílio da ferramenta SIG, é possível estudar melhor quais medidas preventivas podem ser aplicadas para evitar ou pelo menos minimizar os danos causados pelos eventos de cheias, os dados obtidos também permitem que os gestores atuem de maneira antecipada, sendo que em bacias que faltam esses dados é preciso esperar a ocorrência de algum desses eventos para utilizar como modelo e só depois propor as intervenções.

Além dos sistemas de geoinformação para análise de dados espaciais existe uma evolução que visa integrar esses dados ao nível de cidade, que é o sistema de modelagem da informação CIM (*City Information Modeling*). Esse sistema juntamente com o BIM, promovem uma convergência tecnológica que possibilita o monitoramento e o planejamento urbano de forma integrada, atendendo às demandas das cidades inteligentes e da gestão urbana eficiente (Knopp e Paraizo, 2021).

Teixeira (2021) ainda destaca que o CIM oferece uma abordagem para a criação de modelos digitais urbanos que vão além de representações em 3D, permitindo também a integração de informações apresentadas. CIM combina os conceitos de SIG e BIM, promovendo uma visão abrangente que auxilia no planejamento e gestão das cidades, ao integrar dados geoespaciais com informações sobre edificações e infraestrutura.

O CIM supera todas as combinações dos modelos BIM, ao representar um nível mais avançado na gestão de redes de infraestrutura, administração e atividades humanas. Ele facilita a visualização, análise e monitoramento do ambiente urbano, apoiando o planejamento e a execução de projetos desde uma perspectiva local até regional, unificando de forma multidisciplinar todos os modelos de dados espaciais (Dantas, Sousa e Melo, 2019).

## **2.4 Tecnologias de Informação e Comunicação (TICs)**

De acordo com Santos (2014), as TICs são dispositivos tecnológicos desenvolvidos com o intuito de receber, armazenar e organizar os dados neles inseridos, esses dispositivos também conseguem comunicar-se entre eles auxiliando que as informações geradas possam ser compartilhadas.

Os avanços tecnológicos têm auxiliado profissionais em várias áreas, dentre elas o setor da construção civil, onde novas formas de projetar e construir vêm surgindo a cada dia, sempre buscando maiores níveis de perfeição e otimização do tempo.

Leão (2014) afirma que nos últimos anos muitos aspectos têm mudado na indústria da construção civil e um deles é o aumento da competitividade entre empresas, isso tem refletido no processo de gestão das mesmas, onde está sendo necessário buscar inovações para obter um produto final de maior qualidade e com um custo de produção menor.

A adoção de tecnologias inovadoras e métodos modernos de gestão é essencial para otimizar processos, minimizar desperdícios, garantir a qualidade, atender às necessidades dos usuários e contribuir para a melhoria contínua nas construções (Böes e Patzlaff, 2016).

É possível observar que ainda existe uma resistência à adesão de novas tecnologias por alguns profissionais, outro problema que foi destacado por Oliveira e Serra (2017) é o desafio ainda existente de tornar as ferramentas TICs acessíveis aos trabalhadores.

Veloso (2011) evidencia que o uso das TICs mesmo que seja importante e gere oportunidades para novos profissionais, o seu uso não substitui alguns outros processos e/ou profissionais da área, ou seja, a presença de um técnico responsável pelo serviço ainda é essencial em muitos casos para realizar funções que as TICs ainda não são capazes de operar.

Após abordar os desafios e a relevância das TIC's no setor de construção, pode-se destacar a conexão com a Indústria 4.0, que representa uma evolução significativa na maneira como a construção civil opera, especialmente com a integração de tecnologias como BIM. Um exemplo claro é a Plataforma BIM BR, lançada pelo governo federal e que tem promovido a utilização de metodologias modernas para aumentar a eficiência e precisão dos projetos (ABDI, 2024).

## 2.5 Python

Uma das razões para a escolha da linguagem Python é sua sintaxe simples e robusta, que facilita a compreensão de conceitos básicos de algoritmos, como sequência, condição e repetição, especialmente para iniciantes com pouca ou nenhuma experiência em programação (Galvão *et al.*, 2022; Mattosinho *et al.*, 2023).

De acordo com Borges (2014), a linguagem de programação em Python foi criada por Guido van Rossum na década de 1990, com base em outra linguagem existente na época, chamada ABC e com o objetivo de atender físicos e engenheiros. Ferreira (2023) destaca que Python continua sendo muito utilizado, podendo se destacar algumas grandes empresas que fazem seu uso, sendo elas: Google, Yahoo e Spotify. Segundo Verbeurgt *et al.* (2020), Python pode ser utilizado de diversos modos, como para coordenação de sistemas e infraestrutura, elaboração de redes, gerenciamento de dados, entre outras funcionalidades, ou seja, sua aplicabilidade não se restringe apenas em processar e visualizar dados.

O Python é uma linguagem interpretada, assim, não necessita de compilação. Nesta linguagem, a indentação é fundamental, o código não irá funcionar se não estiver devidamente indentado, pois é este artifício que indica quais estruturas estão subordinadas, diferente de C++ que utiliza chaves para isso (Nunes e Amaral, 2020, p. 14).

Garlet, Bigolin e Silveira (2018), dizem que seria de extrema importância se as pessoas comessem a se interessar e utilizar da área de programação para que soubessem produzir novas tecnologias e não apenas consumir as mesmas. Para isso Carvalho e Musci (2024)

ressaltam que seria importante a inserção da linguagem de programação já nos currículos educacionais, a fim de se obter profissionais mais capacitados e preparados para o mercado de trabalho.

Existem diversas linguagens de programação, cada uma com um propósito e característica distinta (Jiménez, 2021). Algumas linguagens de programação existentes são: Java, Java Script, Python, C, C++, PHP.

As Figuras 15, 16, 17 e 18 mostram a implementação da mesma semântica em quatro linguagens de programação específicas: PHP, C++, Java e Python, sendo que o objetivo em todas as linguagens é criar uma classe chamada “Faladora” que contém um método “dizerOi”. Esse método recebe um nome como parâmetro e exibe a mensagem “Olá” seguida do nome informado. Apesar das diferenças de sintaxe entre as linguagens, o propósito de todos os códigos é apresentar a mensagem Olá nome. As Figuras supracitadas também evidenciam como cada linguagem lida com criação de classes, instanciação de objetos, definição de métodos e exibição de resultados. Sendo perceptível observar as vantagens do Python mencionadas no início do tópico.

Figura 15 - Código na linguagem PHP

```
PHP
1 <?php
2
3 class Faladora {
4     public function dizerOi($nome){
5         echo "Olá $nome! <br />";
6     }
7 }
8
9 $falador = new Faladora();
10 $falador.dizerOi("GOBS")
11 ?>
```

Fonte: Silva, 2017.

Figura 16 - Código na linguagem C++

```

C++
1 class Faladora {
2     public:
3         void dizerOi (String nome)
4 };
5
6 void Faladora:dizerOi (String nome) {
7     cout << "Olá" << nome->c_atr{} << endl;
8 }
9
10 int main (void) {
11     Faladora falador = new Faladora ();
12     faladora->dizerOi (new String ("GOBS"))
13 }

```

Fonte: Silva, 2017.

Figura 17 - Código na linguagem Java

```

Java
1 package teste;
2
3
4 public class Faladora {
5     public void dizerOi (String nome) {
6         System.out.print ("Olá"+nome);
7     }
8
9
10 public static void main (String[] args) {
11     Faladora falador = new Faladora ();
12     falador.dizerOi ("GOBS");
13 }
14 }
15

```

Fonte: Silva, 2017.

Figura 18 - Código na linguagem Python

```

Python
1 class Faladora():
2     def dizerOi (self, nome):
3         print("Olá %s" %nome)
4
5 falador = Faladora()
6 falador.dizerOi ("GOBS")
7

```

Fonte: Silva, 2017.

Assim como pode ser apresentado através da Figura 18, Matthes (2023) evidencia que, ao programar com Python é possível escrever um código de forma mais simples e com menos linhas se comparado a outras linguagens executando a mesma função, no qual sua sintaxe também contribui para desenvolver um código mais “limpo”, com maior facilidade de ser lido, além da simplicidade em encontrar erros e mais favorável para promover mudanças no código.

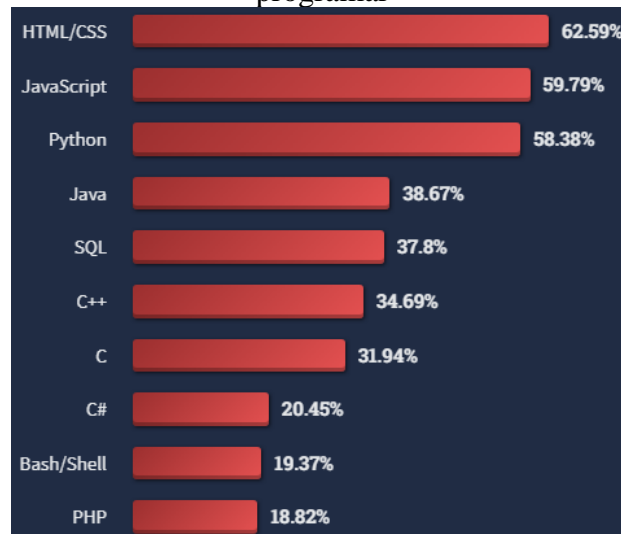
De acordo com Paiva *et al.* (2020), uma grande vantagem de utilizar Python e não C, C++ ou Java, é que essas linguagens necessitam que sempre sejam declarados os tipos de dados que suas variáveis irão armazenar. Diferentemente de Python que possui tipagem dinâmica.

A comunidade *Stack Overflow* (2015) explica que a principal diferença entre as tipagens dinâmica e estática está no processo de verificação dos tipos de dados e variáveis que são utilizados. Na tipagem dinâmica essa verificação é realizada em tempo de execução, já na tipagem estática é feito durante o processo de compilação.

Segundo Buriol, Marco e Argenta (2009) Python possui tipagem dinâmica, ou seja, a variável só tem seu tipo definido quando está sendo executada, mais precisamente no momento em que um valor é atribuído a variável. Sua tipagem é forte, ou seja, ele verifica e só valida a operação se as variáveis forem do mesmo tipo.

No ano de 2022, foi realizada uma pesquisa com alguns desenvolvedores, sobre as linguagens mais utilizadas em várias categorias e ficou constatado que Python foi a terceira linguagem mais usada por pessoas que estão aprendendo a codificar, ficando atrás apenas de HTML/CSS e Java Script respectivamente, conforme pode ser observado na Figura 19. Mas vale ressaltar que HTML/CSS não são consideradas linguagens de programação (*Stack Overflow*, 2022).

Figura 19 - Linguagens de programação preferidas das pessoas que estão aprendendo a programar



Fonte: *Stack Overflow*, 2022.

Segundo Leandro (2018), a linguagem de programação em Python, consegue integrar grande número de profissionais da área de computação, além de ser de fácil entendimento, pois utiliza códigos presentes na realidade dos usuários, facilitando o raciocínio lógico.

De acordo com Menezes (2010), outra grande vantagem de usar Python é que ele se encontra gratuitamente para download e utilização. Além disso, ele é compatível com vários sistemas operacionais como, Linux, FreeBSD, Microsoft Windows e macOS X, o que dispensa acesso a internet para fazer seu uso. A *Python Software Foundation* (2024a) menciona que muitos sistemas compatíveis com Unix, incluindo macOS e algumas distribuições Linux, vêm com o Python pré-instalado como parte da base de instalação do sistema, além disso, o código-fonte do Python pode ser baixado e compilado para Windows e macOS.

Mas como toda linguagem apresenta algumas desvantagens, com Python não é diferente e alguns cuidados sempre devem ser tomados. Por isso Srinath (2017) destaca que é preciso ter atenção quando os códigos começam a ficar muito extensos, pois na ocorrência de algum erro o mesmo poderá exigir mais tempo para ser detectado. A POO (Programação Orientada a Objetos), entretanto, pode mitigar esses problemas, dividindo o código em classes e objetos menores e específicos, cada um com suas funções.

Após esta reflexão sobre um possível desafio no uso de Python, é importante mencionar o conceito de "*deprecated*". No desenvolvimento de software, esse termo refere-se a métodos, classes ou funções que foram marcadas como obsoletos, indicando que, embora ainda funcionem, não devem mais ser usados. Isso geralmente ocorre porque existe uma alternativa

mais moderna e eficiente ou porque o componente apresenta problemas que não serão corrigidos (Cruz, 2024).

Na engenharia é comum quando o profissional se depara com algum problema que aparentemente seja de difícil solução e ali se perde horas tentando encontrar a solução. Segundo Buriol, Marco e Argenta (2009), tais problemas geralmente são analisados através de métodos numéricos computacionais, facilitados pela POO.

Segundo Coelho (2007), com a programação em Python é possível obter um produto final com mais qualidade, elaborado em menor tempo e com a facilidade em realizar manutenção.

De acordo com Srinath (2017), Python está sempre em constante evolução, pois como se trata de uma tecnologia de código aberto sem limitações, os usuários podem criar softwares e ainda compartilhar os mesmos, além de poderem aplicar melhorias em seu código-fonte, para isso a linguagem conta com grandes comunidades dedicadas a sempre propor melhorias.

Maldaner (2019) enfatiza que existem duas etapas a serem seguidas para programar. Primeiramente é necessário pensar na solução do problema que deseja solucionar e desenvolvê-la. Em seguida projetar uma interface para o seu programa, de maneira que fique mais intuitiva possível para o usuário.

Nguyen (2019) enfatiza que é possível escrever códigos até mesmo em um simples editor de texto, mas ao utilizar um IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado) para Python, o programador tem acesso a mais funcionalidades como, realizar testes, controle de versão, gestão de ambientes e pacotes, entre outras, simplificando as tarefas e aumentando o ganho de produtividade. Existem diversos IDE's que são utilizados para programar, como Spyder, PyCharm, VSCode, IDLE.

Gräbner (2020) explica que Spyder é um ambiente de desenvolvimento integrado com muitas ferramentas disponíveis para executar várias tarefas, como, edição de arquivo de texto, edição de códigos em um *Shell* (interpretador que executa os programas), visualização de imagem geradas pelo código, entre outras.

Nguyen (2019) relata que, com a utilização do PyCharm é possível aumentar o ganho de produtividade, pois permite terminar com processos repetitivos, possui detecção de erros em tempo real, permite também navegar pelo código com uma pesquisa inteligente que pode ser útil em códigos mais extensos, além de suporte para projetos de desenvolvimento web.

O VSCode é um editor de código gratuito desenvolvido pela Microsoft e disponível para Windows, Linux e macOS. Suporta linguagens como HTML, CSS, JavaScript, entre outras linguagens de programação, possui uma ampla biblioteca de extensões, além de integração com

o *GitHub* e possibilita trabalho remoto colaborativo. Entre seus diferenciais, destaca-se o terminal embutido, que elimina a necessidade de abrir ferramentas externas como "*Prompt* de Comando" ou "*PowerShell*", e a função de depurador, que facilita a busca de erros ao executar o código linha por linha (Evaldo, 2021).

Segundo Paiva *et al.*, (2020), apesar do IDLE não ser o melhor instrumento para se programar, ele é o mais indicado para quem está iniciando o aprendizado na área da programação. Levada (2021) destaca que o editor IDLE é um dos mais recomendados, pois possui um ambiente muito mais simples e compacto para se programar, além de ser bem leve, o que não necessita de uma máquina com processador especial.

Branchier (2017) explica que a necessidade por obter informações mais rapidamente tem feito com que a tecnologia da informação se torne cada vez mais indispensável em diversas áreas, refletindo em um crescimento acelerado na busca por profissionais qualificados de informação e conhecimento.

Furgeri (2018) relata que existe uma carência muito grande em programadores no Brasil, pois percebe-se que a maioria dos alunos que estão cursando na área acabam não terminando ou não exercendo a profissão depois de formado e alguns após finalizarem não estão preparados para enfrentar um mercado de trabalho que exige um conhecimento muito acurado na área. Segundo Lima e Menezes (2024), parte dessa dificuldade pode estar associada a obstáculos comuns no processo de aprendizagem, como a compreensão de conceitos básicos, aplicação do raciocínio lógico e uso adequado da sintaxe das linguagens. Os estudantes também enfrentam desafios na abstração de problemas e na transformação dessas abstrações em código funcional.

Diante de inúmeras qualidades citadas sobre Python, pode se destacar sua sintaxe simples e clara, o que facilita sua aprendizagem para os iniciantes. Além de possuir uma grande comunidade ativa, facilitando as buscas por respostas sobre as dúvidas e também a existência de várias bibliotecas que contribui para codificar com mais agilidade. Por isso o presente trabalho opta por fazer o uso dessa linguagem de programação para implementação do projeto.

### 2.5.1 Bibliotecas

Srinath (2017) destaca que uma característica importante presente em Python é a existência de bibliotecas variadas, que ao serem utilizadas oferecem recursos, minimizando a necessidade de escrever todos os códigos que seriam necessários sem o uso das mesmas. Segundo Menezes (2010), além de proporcionar ganho de tempo, o uso das bibliotecas também

minimiza a quantidade de erros, pois tudo que nela está presente já foi testado antes por outro programador.

Segundo Borges (2014), para a instalação das bibliotecas no computador, é comum a utilização do PIP. Caso tenha acesso à internet as mesmas já são baixadas nas suas versões mais atuais. Para realizar as instalações, basta digitar no *Prompt* de comando do Windows *pip install nome\_do\_pacote*. Neste trabalho foram utilizadas as bibliotecas Pandas e Tabulate. Logo, para instalação das mesmas foi feito *pip install pandas* e *pip install tabulate*, respectivamente.

Existem várias bibliotecas, cada uma com sua particularidade e características distintas. McKinney (2018) explica que a biblioteca NumPy é uma das mais importantes quando se deseja processamento numérico. Oferece vários recursos como, *arrays* multidimensionais eficientes, funções matemáticas rápidas, ferramentas para leitura e escrita de dados, recursos de álgebra linear, e uma API (*Application Programming Interface*) para integração com bibliotecas em C, C++ e FORTRAN.

De acordo com Lopes *et al.* (2019), o Matplotlib é uma biblioteca de plotagem 2D em Python que gera gráficos de alta qualidade para impressão e ambientes interativos.

Chen (2018) descreve que o Pandas é uma ferramenta de código aberto em Python para análise de dados. Ele permite trabalhar com dados de planilha, facilitando o carregamento, manipulação e combinação de informações. O Pandas introduz dois tipos de dados principais: *Series*, que representa uma única coluna e o *DataFrame*, que representa dados de planilha completos e pode ser visto como um dicionário de objetos *Series*.

#### 2.5.1.1 DataFrames

De acordo com Lopes *et al.* (2019), *DataFrame* é uma estrutura de dados bidimensional, tabular e modificável em seu tamanho, podendo conter dados heterogêneos, organizada em linhas e colunas rotuladas. Como por exemplo, para criar um *DataFrame* a partir de estruturas nativas do Python, como dicionários de listas, é possível usar o parâmetro *columns* para definir a ordem das colunas caso ele não seja utilizado, o construtor organizará as colunas automaticamente por ordem alfabética. O Quadro 8 ilustra um exemplo de utilização do *DataFrame*.

Quadro 8 – *DataFrame* de informações pessoais

```

# Importa a biblioteca pandas para manipulação de dados
import pandas as pd

""" Cria um dicionário com os dados. As chaves representam os nomes das colunas
e os valores são as listas de dados. """
dados = {
    "Nome": ["Ana", "Bruno", "Carla"], # Coluna de nomes
    "Idade": [25, 30, 22], # Coluna de idades
    "Cidade": ["São Paulo", "Rio de Janeiro", "Belo Horizonte"] # Coluna de cidades
}

# Converte o dicionário em um DataFrame
df = pd.DataFrame(dados)

# Exibe o DataFrame completo
print("DataFrame completo:")
print(df)

# Exibe apenas os nomes das colunas
print("\nNomes das colunas:")
print(df.columns)

# Exibe os dados da coluna 'Nome'
print("\nDados da coluna 'Nome':")
print(df["Nome"])

# Exibe as duas primeiras linhas do DataFrame
print("\nAs duas primeiras linhas do DataFrame:")
print(df.head(2))

# Exibe informações resumidas sobre o DataFrame
print("\nInformações sobre o DataFrame:")
print(df.info())

```

Fonte: Autoria própria, 2024.

### 2.5.2 Variáveis

Segundo Souza (2011), as variáveis são essenciais para armazenar dados, sendo necessário apenas determinar seu nome e quais serão estes dados que as mesmas terão que arquivar, ressaltando que Python reconhece caracteres maiúsculos e minúsculos, logo, uma

variável pode conter o mesmo nome de outra mudando apenas o caractere, que ela será tratada diferente da outra.

O Quadro 9 ilustra alguns tipos de variáveis e uma breve explicação de como atuam.

Quadro 9 - Tipos de Variáveis em Python

Exemplo	Explicação
$a = 2$	Variável "a" armazena um número inteiro, logo ela é do tipo <i>int</i> ;
$b = 2.1$	Variável "b" armazena um número que contém casas decimais, logo ela é do tipo <i>float</i> . (Obs.: Em Python, os números decimais devem ser escritos utilizando ponto e não vírgula);
$c = \text{"palavra"}$	Variável "c" armazena uma palavra, logo ela é do tipo <i>string</i> ;
$d = True$	Variável "d" assume um valor verdadeiro ( <i>True</i> ), logo ela é do tipo <i>bool</i> ;
$e = False$	Variável "e" assume um valor falso ( <i>False</i> ), logo ela é do tipo <i>bool</i> (( <i>booleano</i> / lógico);
$nomes = [\text{'José'}, \text{'João'}, \text{'Maria'}]$	Variável "nomes" é uma lista que armazena vários nomes, logo ela é do tipo <i>list</i> ;
$numeros = [1, 2, 5]$	Variável "numeros" é uma lista que armazena vários números, logo ela é do tipo <i>list</i> ;
$numeros2 = (1, 2, 5)$	Variável "numeros2" é uma <i>tupla</i> que armazena vários números, logo ela é do tipo <i>tuple</i> ;
$dados = \{\text{'Nome': 'João', 'Idade': 50}\}$	Variável "dados" é um dicionário que agrupa vários elementos, logo ela é do tipo <i>dict</i> ;
$type(v)$	Comando para mostrar o tipo da variável "v";
$int(v)$	Comando para converter a variável "v" para <i>int</i> ;
$float(v)$	Comando para converter a variável "v" para <i>float</i> ;
$str(v)$	Comando para converter a variável "v" para <i>string</i> .

Fonte: Autoria própria, 2024.

### 2.5.3 Operadores

De acordo com Souza (2023), com a utilização dos operadores é possível reproduzir argumentos lógicos em forma de códigos, unindo a linguagem matemática com a programação.

Montoro (2012) destaca que é preciso tomar cuidado quando é necessário fazer a utilização de vários operadores em uma expressão, devido à prioridade que alguns operadores têm em relação a outros, a utilização de parênteses é uma solução para determinar a ordem de execução dos operadores dentro da expressão.

No Quadro 10 estão descritos os principais operadores matemáticos utilizados em Python.

Quadro 10 - Operadores Matemáticos em Python

Operador	Nome	Exemplo	Resultados dos exemplos
+	Soma	3+3	6
-	Subtração	3-3	0
*	Multiplicação	3*3	9
/	Divisão	3/3	1
%	Módulo	7%2	1
**	Potenciação	3**3	27
//	Divisão inteira	5//2	2

Fonte: Autoria própria, 2024.

Paiva *et al.*, (2020) destaca que os operadores de soma e multiplicação também podem ser utilizados com *strings* (palavras ou letras), por exemplo, se deseja imprimir a mensagem “Olá mundo!”, uma alternativa é concatenar essas duas palavras com o operador de soma, ficando da seguinte forma: `print(“Olá, ” + “mundo!”)`. Ainda segundo o autor, o operador de multiplicação pode ser usado para repetir uma palavra ou letra quantas vezes desejar, por exemplo, para repetir a *string* ABC duas vezes: `print(2 * “ABC”)`, o resultado retornará ABCABC.

Segundo Masiero, Ferreira e Junior (2012) existem também os operadores de comparação, empregues quando há necessidade de analisar algum tipo de condição. Esses estão representados no Quadro 11.

Quadro 11 - Operadores de Comparação em Python

Operador	Descrição	Exemplos	Resultados dos exemplos
==	Verifica se o valor de um elemento é igual ou não ao valor do outro	a=1 e b=2 a==b	<i>False</i>
!=	Verifica se o valor de um elemento é diferente ou não ao valor do outro	a=1 e b=2 a!=b	<i>True</i>
<>	Semelhante ao operador !=	a=2 e b=2 a<>b	<i>False</i>
>	Verifica se o valor do elemento à esquerda é maior do que o valor do elemento à direita	a=1 e b=2 a>b	<i>False</i>
<	Verifica se o valor do elemento à esquerda é menor do que o valor do elemento à direita	a=1 e b=2 a<b	<i>True</i>
>=	Verifica se o valor do elemento à esquerda é maior ou igual do que o valor do elemento à direita	a=1 e b=2 a>=b	<i>False</i>
<=	Verifica se o valor do elemento à esquerda é menor ou igual do que o valor do elemento à direita	a=1 e b=2 a<=b	<i>True</i>

Fonte: Masiero, Ferreira e Junior, 2012, adaptado pelo autor, 2024.

#### 2.5.4 Comentários em Python

Souza (2011) descreve que os comentários são utilizados para registrar de forma mais detalhada o que o código está realizando, além de auxiliar na leitura e conseqüentemente no entendimento do mesmo.

Paiva *et al.*, (2020) também revela que os comentários são muito úteis de serem empregados em trechos de códigos que ainda não devem ser executados por algum motivo, dessa maneira apenas essa parte que possui o # não irá rodar no programa, mas continuará lá para alterações futuras.

Na Figura 20 é possível observar dois tipos de comentário, o primeiro que é um comentário descrito em apenas uma linha, em que o mesmo sempre precisa ser iniciado com o caractere # e o segundo que é utilizado para expressar comentários mais extensos que serão compostos em mais de uma linha, este é iniciado e finalizado com três aspas.

Figura 20 – Exemplo de comentários em Python

```
#Este é um comentário de uma linha

'''Este é um comentário multilinhas,
ou seja, assim é possível fazer
um comentário com mais de uma linha'''
```

Fonte: Autoria própria, 2024.

#### 2.5.5 Funções

Conforme mencionado por Borges (2014), as funções em Python são blocos de código que podem retornar objetos, aceitar parâmetros opcionais e serem documentadas com *Docstrings*. Elas possuem seu próprio escopo local, o que permite evitar conflitos com variáveis de escopo global, e ainda permitem que seus comportamentos sejam alterados por decoradores.

*Python Software Foundation* (2024b) explica que as *Docstrings* são usadas para descrever a funcionalidade de módulos, classes, métodos e funções, tornando o código mais compreensível. O Quadro 12 exemplifica essa função.

Quadro 12 – Exemplo de código com *Docstrings* em Python

```
def calcular_media(a, b, c):
    """
    Calcula a média aritmética de três números.
    Parâmetros:
    a (float): O primeiro número.
    b (float): O segundo número.
    c (float): O terceiro número.

    Retorna:
    float: A média dos três números.
    """
    return (a + b + c) / 3

media = calcular_media(7, 8, 9)
print(f"A média é: {media}")
```

Fonte: Autoria própria, 2024.

Alguns teóricos das linguagens de programação apontam que, em Python, as funções são classificadas como objetos de primeira classe. Isso significa que elas podem ser desenvolvidas durante a execução do programa, atribuídas a variáveis ou armazenadas em estruturas de dados, além de serem apresentadas como argumentos para outras funções e retornadas como resultados, o que garante maior flexibilidade no desenvolvimento de código (Ramalho, 2015).

Quadro 13 – Exemplo de funções em Python

```
#Definindo uma função que aceita outra função como argumento
def aplicar_funcao(funcao, valor):
    return funcao(valor)

#Função que será passada como argumento
def dobrar_valor(x):
    return x * 2

#Usando a função aplicar_funcao
resultado = aplicar_funcao(dobrar_valor, 5)
print(resultado) # Apresenta na tela: 10
```

Fonte: Autoria própria, 2024.

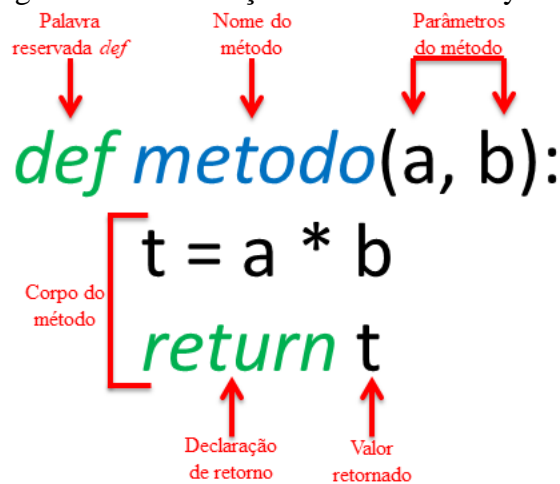
No Quadro 13, a função **aplicar\_funcao** recebe outra função (**dobrar\_valor**) e o número cinco, aplicando a função ao valor fornecido, demonstrando então como as funções em

Python podem ser passadas como argumentos, ilustrando a ideia de funções como objetos de primeira classe.

### 2.5.6 Métodos

“Um método é uma função que “pertence” a um objeto instância” (*Python Software Foundation, 2024c*). Os métodos se diferem das funções por duas diferenças sintáticas básicas, a sintaxe para chamar ambos são diferentes e na determinação do método que é feita dentro da definição de classe (Downey, 2016). A Figura 21 ilustra como realizar a declaração de um método.

Figura 21 – Declaração de método em Python



Fonte: Autoria própria, 2024.

Carvalho (2016) explica que, na maioria das vezes um método deve possuir um retorno, pois se algo é solicitado então pressupõe-se que seja retornado alguma resposta, para isso utiliza-se o *return*.

De acordo com Montoro (2013), Python possui uma série de métodos nativos para serem utilizados a fim de executar funções básicas. Alguns métodos de objetos tipo *string* serão exemplificados a seguir.

- Método *format()*: utilizado para substituir conteúdo de uma frase por objetos passados por esse método;

Quadro 14 – Exemplo de aplicação do método *format* em Python

```

nome = "Rafael"
idade = 28
cidade = "Vargem Bonita"

mensagem = "Olá, meu nome é {}. Tenho {} anos e moro em {}.".format(nome, idade, cidade)
print(mensagem) # Apresenta na tela: Olá, meu nome é Rafael. Tenho 28 anos e moro em Vargem Bonita.

```

Fonte: Autoria própria, 2024.

- Método *read()*: utilizado para ler o conteúdo de um arquivo;

Quadro 15 – Exemplo de aplicação do método *read* em Python

```

def ler_arquivo():
    arquivo = open("meuarquivo.txt", 'r')
    conteudo = arquivo.read()
    arquivo.close()
    return conteudo

conteudo = ler_arquivo()
print(conteudo) # Apresenta na tela: o texto armazenado no arquivo chamado meuarquivo

```

Fonte: Autoria própria, 2024.

- Método *capitalize()*: utilizado para retornar uma *string* com a primeira letra maiúscula e o restante minúsculo;

Quadro 16 – Exemplo de aplicação do método *capitalize* em Python

```

texto = "python é incrível!"
texto_corrigido = texto.capitalize()

print(texto_corrigido) # Apresenta na tela: Python é incrível!

```

Fonte: Autoria própria, 2024.

- Método *upper()*: utilizado para retornar uma *string* com todos os seus caracteres em maiúsculos;

Quadro 17 – Exemplo de aplicação do método *upper* em Python

```

texto = "programação é muito bom!"
texto_corrigido = texto.upper()

print(texto_corrigido) # Apresenta na tela: PROGRAMAÇÃO É MUITO BOM!

```

Fonte: Autoria própria, 2024.

- Método *replace()*: utilizado para substituir trechos em uma *string*;

Quadro 18 – Exemplo de aplicação do método *replace* em Python

```
texto = "Programação é muito bom!"
texto_corrigido = texto.replace("bom", "difícil")

print(texto_corrigido) # Apresenta na tela: Programação é muito difícil!
```

Fonte: Autoria própria, 2024.

- Método *find()*: indica a posição de uma *substring* dentro de uma *string*.

Quadro 19 – Exemplo de aplicação do método *find* em Python

```
texto = "Aprender Python é muito gratificante."
indice = texto.find("Python")

print(indice) # Apresenta na tela: valor 9, que é a posição inicial da palavra "Python" na string
```

Fonte: Autoria própria, 2024.

## 2.6 Programação Orientada a Objetos (POO)

De acordo com Oberleitner e Masieiro (2021) a POO originou-se da necessidade de encontrar uma linguagem de programação em que as linhas de código fossem representadas o mais próximo da realidade possível, no qual o objeto é a peça principal e que simboliza qualquer entidade.

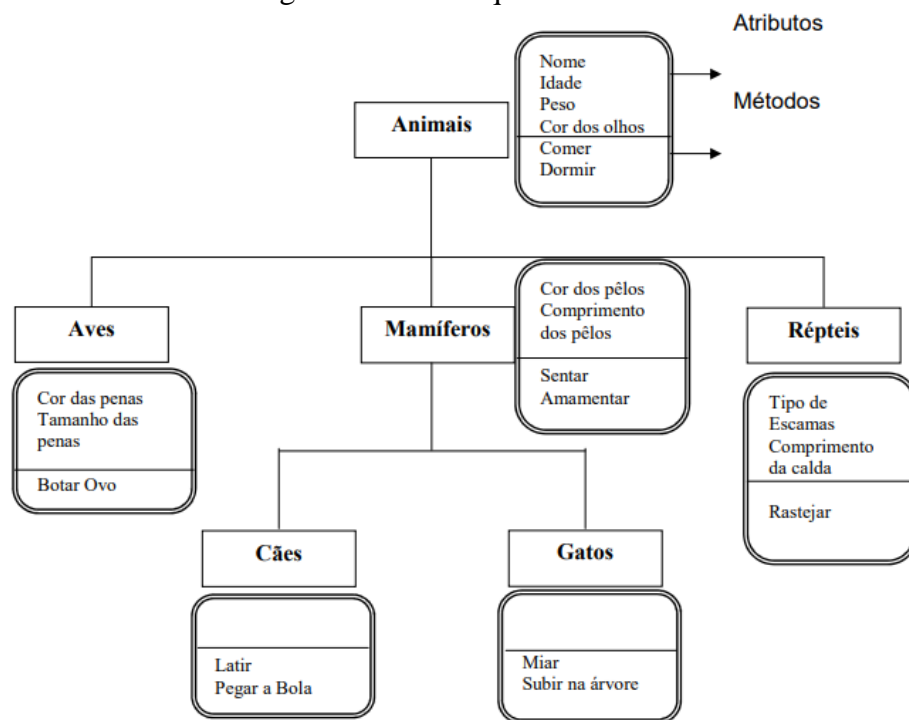
Conforme relatado por Soubhia *et al.* (2019), a programação orientada a objetos é um paradigma de programação muito utilizado e que está presente na estrutura de quase todas as linguagens de programação, em Python ela está evidentemente ativa. Este paradigma possui vantagens como, possibilidade de reutilização do código, ganho de produtividade, sistemas mais flexíveis, códigos mais curtos, fácil manutenção e divisão das responsabilidades.

Segundo Buriol, Marco e Argenta (2009), a programação orientada a objetos permite que o usuário tenha mais domínio sobre seu programa e faça até reutilização do código, além de possibilitar que alterações sejam feitas sem ocorrer mudanças na interface. Martha (2022) destaca que essas características da POO auxiliam para constantes melhorias e atualizações dos softwares.

Furgeri (2018) evidencia que nos últimos anos várias empresas criadoras de *softwares* começaram a modificar seus códigos e estão mudando para linguagens de programação orientadas a objetos. A programação orientada a objetos possui alguns conceitos muito importantes em sua estrutura, como classes, objetos, herança, encapsulamento e abstração.

A Figura 22 exemplifica uma estrutura hierárquica de classes utilizando o conceito de herança, comum na programação orientada a objetos. No diagrama existe uma Superclasse denominada “Animais” e dela se estendem três Subclasses que são “Aves”, “Mamíferos” e “Répteis” cada uma com seus atributos e métodos. A Subclasse “Mamíferos” também se comporta como uma Superclasse em relação as classes “Cães” e “Gatos”, sendo que os atributos da Subclasse “Mamíferos” são os mesmos das suas Subclasses, mas os métodos são diferentes. Outro detalhe importante a ser observado é que os atributos “nome”, “idade”, “peso” e “cor dos olhos” e os métodos “comer” e “dormir” da Superclasse “Animais” são comuns para todas as Subclasses.

Figura 22 - Hierarquia de classes



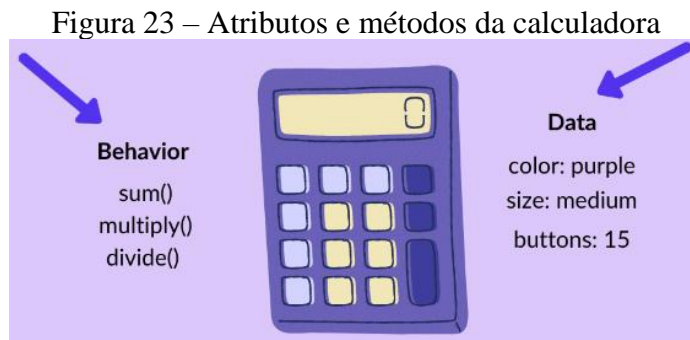
Fonte: Farinelli, 2007.

Conforme explicado por Furgeri (2018) na programação orientada a objetos tudo se inicia a partir da criação das classes e a partir dela podem ser gerados vários objetos. Martha (2022) define as classes como sendo construtoras de objetos, determinando as funções que serão capazes de executar e quais os atributos dos objetos.

Segundo Carvalho e Teixeira (2012) um objeto é uma instância de uma classe e é capacitado para armazenar informações através de seus atributos e agir a mensagens destinadas ao mesmo. Farinelli (2007) explica que todo objeto possui características próprias que identificam o mesmo e essas são chamadas de atributos, ele também pode realizar um conjunto de ações que é identificado como métodos.

Furgeri (2018) também define esse conjunto de características do objeto como “estado”, sendo que desde a origem até o fim deste objeto ele pode apresentar “estados” distintos e essas ações que ele pode executar também podem ser chamadas de “comportamentos”.

Um bom exemplo de objeto pode ser a calculadora, conforme ilustra a Figura 23, a qual possui atributos que são cor, tamanho e quantidade de botões, além dos métodos que são as operações que ela pode realizar como, soma, multiplicação e divisão.



Fonte: Diaz, 2023.

De acordo com Farinelli (2007) é de suma importância que inicialmente seja feito a definição das classes e objetos, pois se for necessário fazer alguma alteração depois de iniciado e mesmo que seja pequena, pode se estender por todas as linhas de código.

Já a herança segundo Germanio (2005) pode ser entendida como a maneira de reutilizar objetos de uma Superclasse para uma nova Subclasse, sendo que todos os atributos dos objetos da Superclasse também podem estar presentes nessa SubClasse, onde os atributos da SuperClasse podem estar disponíveis na SubClasse.

Carvalho e Teixeira (2012) esclarece que o encapsulamento é um método empregado para limitar o acesso aos atributos, métodos e até mesmo as classes, ou seja, o usuário consegue fazer o uso normalmente dos métodos da classe, mas sem ter conhecimento em como eles foram implementados.

Farinelli (2007) destaca que o encapsulamento é uma das grandes vantagens da programação orientada a objetos em relação à programação estruturada, sendo que o objeto estará disponível para uso sem que seja preciso entender como é seu funcionamento e nem como é feito o armazenamento das informações recuperadas, além disso, ele permite que seja feita alterações internas no objeto sem influenciar em outras partes do código que também utilizam esse objeto.

O conceito de abstração é definido por Soubhia *et al.* (2019) como uma das partes mais importantes da orientação a objetos, pois ao trabalhar com objetos reais eles necessitam de abstração, que é saber o que é determinado objeto e o que ele realiza. Para isso ele é dividido

em três partes, a “identidade” trás o nome desse objeto, a “propriedade” que é onde estão as características que ele possui e o “método” que indica a função do objeto.

### 2.6.1 Métodos de Classe

De acordo com a documentação oficial do Python, *def* é uma instrução que geralmente atua como definição de função quando estiver dentro de uma classe (Python Software Foundation, 2024c).

Segundo Carvalho (2016), podem existir diversos métodos em uma classe, sendo que cada um possui uma tarefa que precisa ser realizada dentro da classe.

Existe um método especial chamado `__init__`, o mesmo deve ser escrito com dois caracteres de sublinhado antes e depois da palavra. De acordo com *W3Schools* (2024), ele é utilizado para conceder valores às propriedades do objeto e é requerido sempre que uma classe é instanciada. Furgeri (2021) explica que esse método é o construtor da classe e pode ser definido como *def \_\_init\_\_(self)*, sendo que a quantidade de parâmetros, que fica dentro dos parênteses, varia de acordo com a quantidade de atributos da classe. No Quadro 20 é possível observar a utilização do construtor da classe Calculadora, onde o mesmo recebe os parâmetros “a” e “b”.

### 2.6.2 Outras Palavras Reservadas do Python

*Python Software Foundation* (2024c) descreve que o primeiro parâmetro de um método é constantemente chamado de *self*, mas isso não é uma regra, também podem ser utilizadas outras palavras desde que sempre se utilize a mesma como primeiro parâmetro em qualquer método.

### 2.6.3 Exemplos de códigos

A seguir serão apresentados no Quadro 20 exemplos de códigos utilizando várias variáveis e realizando as funções de soma e comparação. Já o Quadro 21 exhibe onde as funções do programa são executadas de fato, também é aqui que todos os módulos são importados.

Quadro 20 - Exemplos de códigos em Python

```

class Calculadora: # Criando uma classe chamada "Calculadora"
    def __init__(self, a, b): # Construtor da classe Calculadora
        self.a = a # Variável "a" para armazenar o primeiro número
        self.b = b # Variável "b" para armazenar o segundo número

    def soma(self): # Método que calcula a soma de "a" e "b"
        return self.a + self.b # Operação de soma entre "a" e "b"

    def subtracao(self): # Método que calcula a subtração de "a" e "b"
        return self.a - self.b # Operação de subtração entre "a" e "b"

    def multiplicacao(self): # Método que calcula a multiplicação de "a" e "b"
        return self.a * self.b # Operação de multiplicação entre "a" e "b"

    def divisao(self): # Método que calcula a divisão de "a" por "b"
        if self.b != 0: # Condição para evitar divisão por zero
            return self.a / self.b # Operação de divisão entre "a" e "b"
        else: # Caso o divisor seja zero
            return "Erro: Divisão por zero não é permitida" # Mensagem de erro para divisão por zero

```

Fonte: Autoria própria, 2024.

Quadro 21 - Execução dos códigos (*main*) em Python

```

from calculadora import Calculadora # Importando a classe denominada "Calculadora"

def main(): # Função principal para executar o programa
    "Solicitando que o usuário insira um valor para "a" e o mesmo precisa ser do tipo inteiro"
    a = int(input("a: "))
    "Solicitando que o usuário insira um valor para "b" e o mesmo pode ser do tipo decimal ou inteiro"
    b = float(input("b: "))

    calculo = Calculadora(a, b) # Instancia da classe Calculadora

    print("Soma:", calculo.soma()) # Mostra o resultado da soma
    print("Subtração:", calculo.subtracao()) # Mostra o resultado da subtração
    print("Multiplicação:", calculo.multiplicacao()) # Mostra o resultado da multiplicação
    print("Divisão:", calculo.divisao()) # Mostra o resultado da divisão

if __name__ == "__main__":
    main() # Chamando a função principal

```

Fonte: Autoria própria, 2024.

## 2.7 O Projeto de Pesquisa

Conforme supracitado no item 1.1, este trabalho é um aprofundamento do projeto de pesquisa realizado, sendo o mesmo descrito resumidamente a seguir.

A primeira etapa na execução da pesquisa foi analisar através de notícias, relatos dos moradores e até mesmo presencialmente, os locais que estavam inundando com as fortes chuvas. Este processo definiu a área de estudo.

Essa área total foi delimitada e representada no *software* QGIS (*Quantum Geographic Information System*) e com o auxílio do *software* IDE-Sisema (Web GIS), foi possível obter as curvas de nível dela.

Após esse processo foi necessário ir a campo para coletar todas as informações necessárias referentes à boca de lobo, ruas e sarjetas, além disso, também foi necessário analisar o sentido de escoamento do fluxo de água, para realizar uma comparação com as curvas de nível obtidas com o *software* a partir dos *shapefiles* disponíveis no Sisema, e saber se esse sentido iria coincidir.

No total foram coletadas as dimensões e distâncias de 125 bocas de lobo, além das dimensões das sarjetas, largura das ruas e alturas dos meio fios. Para que todas essas informações fossem coletadas, foi necessário percorrer uma extensa distância, passando por todas as ruas que estavam dentro da área de estudo e que estavam concentradas em três bairros.

Através do *software* AutoCAD, foi lançado no mesmo o mapa que continha o traçado das ruas da área delimitada e através do método de Thiessen essa área foi dividida em 219 subáreas e locadas todas as bocas de lobo, uma por uma, de acordo com a sua localização.

Com os dados que foram coletados, a etapa seguinte foi anexar os mesmos em uma planilha de cálculo pré-configurada e disponibilizada pelo professor Dr. Germano de Oliveira Mattosinho durante a disciplina de Saneamento, que já continha todas as fórmulas para realizar todos os cálculos necessários. Algumas adaptações foram executadas na planilha para se obter as informações finais que eram necessárias, como por exemplo, as quantidades mínimas de boca de lobo que deveriam conter em cada trecho/rua.

Ainda em fase de elaboração da pesquisa o grupo de estudantes pôde apresentar e publicar a pesquisa em formato de resumo expandido no VIII Seminário de Iniciação Científica – IFMG que ocorreu em Ribeirão das Neves MG, posteriormente ao término da pesquisa a mesma também foi publicada na revista *ForScience* v. 9, n. 2, ano 2021 (Silva e Mattosinho, 2021).

Durante a realização do projeto de pesquisa, diversas dificuldades foram enfrentadas devido à complexidade e à extensa quantidade de fórmulas envolvidas. Além disso, o projeto abrangeu uma área significativa, demandando o mapeamento, análise e cálculo de elementos como ruas, bocas de lobo e sarjetas. Essas exigências tornaram o processo trabalhoso e suscetível a erros manuais nos cálculos. Com base nessas dificuldades, surgiu a iniciativa de desenvolver uma solução computacional em Python, descrito neste trabalho, com o objetivo de otimizar o tempo dedicado aos cálculos matemáticos e reduzir os erros inerentes ao processo manual.

### **3 METODOLOGIA**

Neste capítulo, serão detalhados todos os procedimentos adotados para o desenvolvimento da ferramenta, seu funcionamento e a classificação da pesquisa realizada. O trabalho em questão caracteriza-se como uma pesquisa aplicada de natureza exploratória e descritiva. Para a realização do estudo, foram utilizados diversos procedimentos técnicos, incluindo a realização de uma revisão bibliográfica sobre trabalhos relacionados ao tema, a pesquisa experimental para o desenvolvimento da ferramenta computacional, a pesquisa de campo para a coleta de dados e a condução de um estudo de caso. Este estudo se justifica pela necessidade de abordar os problemas de drenagem urbana enfrentados por diversos bairros do município de Piumhi - MG, especialmente após a ocorrência de chuvas intensas. A abordagem adotada será de caráter quantitativo, com ênfase na análise e interpretação dos dados coletados.

Figura 24 - Fluxograma sintético de execução da solução computacional.



Fonte: Autoria própria, 2025.

O fluxograma sintético ilustrado na Figura 24 retrata as etapas de inserção de dados, escolha de geometrias e cálculos hidráulicos, conforme explicitado a seguir neste capítulo.

Os dados utilizados para validar a solução computacional foram coletados em campo através do Programa Institucional de Bolsa de Pesquisa – IFMG, Edital 005/2019, onde foi

desenvolvido o projeto **Análise de Drenagem Urbana Implementando Modelagem GIS em Piumhi-MG**, que foi detalhado no Capítulo 2.7.

### **3.1 Área de Estudo**

Antes de iniciar o projeto é fundamental fazer uma análise detalhada da área de estudo.

Belo Horizonte (2022), lista várias informações que devem ser captadas em fase inicial e que devem estar presentes em um relatório de conhecimento, em resumo são: informações da sub-bacia de estudo, estado atual dos terrenos, caracterização urbana, dispositivos de drenagem existentes, além da elaboração de mapa que contemple toda a área de estudo, planta geral da bacia contribuinte, levantamento planialtimétrico do terreno e áreas relevantes do entorno e no caso de novos loteamentos, é essencial realizar um levantamento topográfico inicial e desenvolver uma proposta de urbanização que priorize a mitigação dos impactos no escoamento das águas pluviais.

Para facilitar a utilização da ferramenta é importante contar com o auxílio de um croqui detalhado, que contenha a área total delimitada com as cotas de nível, assim como as áreas de influência de cada trecho, identificação dos trechos sempre nomeados de forma clara e intuitiva e o traçado das ruas.

### **3.2 O Projeto de Programação**

A matriz curricular atual do curso oferece uma base abrangente em diversas áreas essenciais. No entanto, não inclui uma disciplina específica voltada para o aprofundamento em linguagens de programação, o que pode tornar o aprendizado nessa área mais superficial, dificultando o processo de escolha da linguagem ideal e a construção de código. Dessa forma, é necessário um esforço complementar, na busca por esse aprendizado, para desenvolver as habilidades de programação desejadas.

No capítulo 2.5 deste trabalho, destacou-se as vantagens de uso do Python como uma linguagem de programação intuitiva, amplamente utilizada por profissionais e iniciantes devido a sua sintaxe clara e de fácil compreensão. Essa característica é reforçada por Jiménez (2021), que aponta que a gramática de Python torna sua escrita semelhante a um texto tradicional, o que facilita tanto a leitura quanto o aprendizado do código.

Esse panorama serviu de base para a especificação da escolha da linguagem para o desenvolvimento deste trabalho. Após uma análise comparativa entre algumas linguagens de programação, Python se destacou não apenas pela sua facilidade de aprendizado, mas também

pela ampla gama de bibliotecas e ferramentas disponíveis, que atendem às necessidades específicas do projeto (*Stack Overflow*, 2022).

Para garantir um ambiente de escrita acessível e funcional, foi utilizado o editor IDLE, que integra a instalação padrão do Python. Esse Ambiente de Desenvolvimento Integrado oferece simplicidade e eficiência para a construção e execução dos códigos, atendendo perfeitamente ao perfil do projeto, do autor da solução computacional e aos objetivos propostos.

### 3.2.1 Classes

Como dito anteriormente, devido à extensa quantidade de cálculos a serem executados e considerando os conceitos de POO, visando o reaproveitamento do código entendimento, organização e visualização, o programa foi organizado em oito classes. Cada classe desempenha um papel específico no processo de cálculo e simulação, colaborando para determinar a quantidade necessária de bocas de lobo em cada trecho de vias urbanas. A seguir, são descritas essas classes e suas respectivas funcionalidades.

#### 3.2.1.1 Classe *LerTabela*

A classe **LerTabela** é projetada para fazer a leitura e o processamento de uma tabela de constantes IDF (Intensidade-Duração-Frequência) com base no município e na estação pluviométrica selecionada, esses dados são fundamentais para calcular a intensidade das chuvas em diferentes regiões.

O método principal dessa classe, chamado **lendo\_tabela**, é implementado como um método estático, que segundo a comunidade *Stack Overflow* (2017) significa que ele pode ser chamado diretamente sem a necessidade de criar uma instância da classe. Isso é útil para facilitar o acesso rápido aos dados de qualquer município.

O funcionamento dessa classe ocorre da seguinte forma:

- A biblioteca pandas é utilizada para ler um arquivo em formato CSV intitulado "ConstantesIDF.csv", que foi modificado com base em informações apresentadas no artigo de Bello (2018). O referido artigo reúne dados sobre diversas estações pluviométricas do estado de Minas Gerais, incluindo os parâmetros **k**, **a**, **b** e **c**, que são utilizados nas equações de intensidade pluviométrica;
- O arquivo é carregado com codificação ISO-8859-1, adequada para garantir a correta leitura de caracteres especiais em português (*Stack Overflow*, 2014);

- Após a leitura, o método filtra os dados do *DataFrame* com base na cidade e na estação fornecidas como parâmetros, utilizando as colunas "Municípios" e "Estações";
- Por fim, o método **lendo\_tabela**, retorna os valores dos coeficientes **k**, **a**, **b** e **c** correspondentes ao município e a estação pluviométrica especificada pelo usuário, utilizando o método `loc` para filtrar a tabela.

Essa classe garante que os parâmetros IDF necessários para o cálculo da intensidade de chuva sejam corretamente carregados e filtrados, facilitando a integração dos dados com as demais classes do sistema.

Método criado e sua finalidade:

- **lendo\_tabela(cidade)**: Lê o arquivo de constantes IDF e filtra os dados do município e estação pluviométrica solicitada, retornando um *DataFrame* com as colunas de parâmetros IDF para essa localidade.

### 3.2.1.2 Classe *IntensidadePluviometrica*

A classe **IntensidadePluviométrica** é responsável por calcular e fornecer a intensidade da chuva para uma cidade específica, utilizando os parâmetros de intensidade-duração-frequência (IDF).

O funcionamento dessa classe ocorre da seguinte forma:

- Ao ser instanciada, a classe recebe três parâmetros: o nome da cidade (`cidade`), nome da estação (`estacoes`), o período de retorno em anos (`t_anos`), e o tempo de duração da precipitação em minutos (`t_min`);
- A cidade fornecida é utilizada para buscar os coeficientes IDF na classe **LerTabela**. Se os dados da cidade e da estação não forem encontrados, o usuário pode optar por inserir os coeficientes manualmente;

Métodos criados e suas finalidades:

- **set\_parametros\_manual()**: Permite que o usuário insira manualmente os valores dos coeficientes **k**, **a**, **b** e **c**;
- **get\_cidade()**, **get\_t\_anos()**, **get\_t\_min()**: Retornam os valores da cidade, tempo de retorno e duração da chuva, respectivamente;
- **set\_parametro\_k()**, **get\_parametro\_k()**, **set\_parametro\_a()**, **get\_parametro\_a()**, **set\_parametro\_b()**, **get\_parametro\_b()**, **set\_parametro\_c()** e **get\_parametro\_c()**: Atribuem e retornam os valores dos

parâmetros IDF a partir dos dados da tabela, se disponíveis. Esses métodos acessam os valores por meio de índices específicos no DataFrame;

- **set\_valor\_intensidade\_pluviometrica()** e **get\_valor\_intensidade\_pluviometrica()**: Define e retorna o valor da intensidade pluviométrica calculada;

Se no local em estudo não houver registros pluviométricos, uma das maneiras de sanar essa falta é utilizando bases bibliográficas que trazem informações para se chegar ao resultado aproximado, como a equação desenvolvida por Otto Pfafstetter (DNIT, 2005).

$$i = \frac{k \times T^a}{(t+b)^c} \quad (3)$$

$i$  = intensidade média de precipitação (mm/h);

$T$  = tempo de retorno (anos);

$t$  = tempo de concentração (min);

$k, a, b, c$  = constantes de ajuste locais.

### 3.2.1.3 Classe Localidade

Essa classe serve como ponto de entrada para reunir dados específicos de cada localidade e integrar os cálculos gerais do sistema. Por exemplo, uma vazão de um trecho será calculada utilizando o coeficiente de escoamento e os dados de intensidade pluviométrica fornecidos por esta classe.

Métodos criados e suas finalidades:

- **\_\_init\_\_(self, coeficiente\_escoamento, obj\_intensidade\_pluviometrica)**: O construtor inicializa os atributos da instância da classe. Ele requer: Um valor numérico para o coeficiente de escoamento e um objeto de classe relacionado à intensidade pluviométrica;
- **get\_intensidade\_pluviometrica()**: Retorna o objeto de intensidade pluviométrica associado à localidade. Isso permite que outras partes do programa acessem informações pluviométricas para cálculos de vazão;
- **get\_coeficiente\_escoamento()**: Retorna o valor do coeficiente de escoamento, permitindo que ele seja utilizado na função que calcula as vazões;
- **set\_vazoes\_inicial\_trechos()** e **get\_vazoes\_inicial\_trechos()**: Armazenamento e obtenção da lista que armazena a vazão total de cada trecho;
- **set\_vazao()** e **get\_vazao()**: Calcula e retorna a vazão gerada em cada trecho;

A vazão máxima de precipitação nos trechos foi calculada através do Método Racional (Equação 4), que segundo Tucci (1993) é o método mais utilizado em pequenas bacias cujas áreas sejam menores ou iguais a 2 km<sup>2</sup>.

$$Q_i = \frac{C \times i \times A}{360} \quad (4)$$

$Q_i$  = vazão de precipitação (m<sup>3</sup>/s);

$C$  = coeficiente de escoamento superficial;

$A$  = área de contribuição do trecho (ha).

#### 3.2.1.4 Classe Trecho

Cada trecho representa uma conexão entre pontos da rede, caracterizada por atributos como área de contribuição, comprimento e cotas (montante e jusante). Além disso, essa classe gerencia relações hierárquicas entre trechos (montante e jusante), permitindo que o programa construa e analise a rede de drenagem de forma integrada.

Métodos criados e suas finalidades:

- **get\_cadastro()** e **encerrar\_cadastro()**: Esses métodos gerenciam o estado de cadastro dos trechos. Enquanto **get\_cadastro()** verifica se o cadastro está ativo, **encerrar\_cadastro()** altera o estado para indicar que o processo de cadastro foi finalizado;
- **add\_trecho()** e **get\_lista\_trechos()**: Permitem o gerenciamento de uma lista de trechos cadastrados. O método **add\_trecho()** adiciona novos trechos à lista, enquanto **get\_lista\_trechos()** retorna todos os trechos cadastrados;
- **get\_indice\_trecho()** e **trecho\_existente()**: Facilitam a identificação de trechos específicos dentro da lista cadastrada. O método **get\_indice\_trecho()** retorna a posição de um trecho na lista com base no seu identificador, enquanto **trecho\_existente()** verifica se um trecho com determinado identificador já foi registrado;
- **set\_id()** e **get\_id()**: Defini e retorna o id de cada trecho, esse id funciona como um identificador da posição, a fim de ordenar os trechos;
- **add\_trecho\_montante()** e **add\_trecho\_jusante()**: Estabelecem as conexões entre trechos, permitindo que o programa organize a hierarquia entre eles;
- **get\_trechos\_montante()** e **get\_trechos\_jusante()**: Recuperam as listas de trechos conectados a montante e a jusante, possibilitando a análise da rede de drenagem;

- **set\_valor\_area\_contribuicao()**, **get\_valor\_area\_contribuicao()**, **set\_comprimento()**, **get\_comprimento()**, **set\_cota\_montante()**, **get\_cota\_montante()**, **set\_cota\_jusante()**, **get\_cota\_jusante()**: Variáveis que são solicitadas ao usuário e retornadas posteriormente;
- **set\_declividade\_via()** e **get\_declividade\_via()**: Defini e retorna o valor da declividade longitudinal da via;

$$I = \frac{Cota_m - Cota_j}{L_v} \quad (5)$$

$I$  = declividade da via (m/m)

$Cota_m$  = cota montante (m);

$Cota_j$  = cota jusante (m);

$L_v$  = comprimento da via (m).

- **set\_vazao\_futura()**: Possibilita ao usuário inserir uma possível vazão futuro, caso o mesmo julgue necessário;  
Após a inserção da vazão futura (caso haja), teremos então a vazão total em cada trecho, que nada mais é do que o somatório da vazão inicial com a vazão futura.  
$$Q_T = Q_i + Q_F \quad (6)$$
  
 $Q_t$  = vazão total no trecho (m<sup>3</sup>/s);  
 $Q_f$  = vazão futura para o trecho (m<sup>3</sup>/s);
- **has\_data()**: Verifica se o trecho possui informações cadastradas, como área de contribuição, comprimento ou cotas. Esse método é útil para identificar trechos que ainda precisam de dados para os cálculos.

### 3.2.1.5 Classe Sarjeta

A classe Sarjeta tem como objetivo principal modelar e realizar os cálculos relacionados às características e ao comportamento hidráulico das sarjetas em uma rede de drenagem urbana. Ela interage com outras classes, como **Localidade** e **Trecho**, para calcular capacidades, verificações e identificar necessidades de dispositivos como bocas de lobo.

Métodos criados e suas finalidades:

- **get\_largura\_sarjeta()**, **get\_altura\_meio\_fio()**, **set\_altura\_disponivel\_sarjeta()**, **get\_altura\_disponivel\_sarjeta**, **set\_declividade\_sarjeta()** e **get\_declividade\_sarjeta()**: Solicita e retorna os valores da largura da sarjeta, altura do meio fio, altura disponível da sarjeta e declividade da sarjeta;

- **set\_dimensao\_z\_sarjeta()** e **get\_dimensao\_z\_sarjeta()**: Permitem configurar e acessar a dimensão Z da sarjeta, usada para caracterizar sua geometria e, conseqüentemente, influenciar no cálculo da capacidade teórica da sarjeta;

$$Z = \frac{L_s}{y_0} \quad (7)$$

Z = inverso da declividade transversal;

$L_s$  = largura da sarjeta (m);

$y_0$  = altura disponível da sarjeta (m).

- **set\_capacidade\_teorica\_sarjeta()** e **get\_capacidade\_teorica\_sarjeta()**: Determinam a capacidade teórica da sarjeta, ou seja, a máxima vazão que pode ser conduzida considerando condições ideais;

Através da Equação 8 é possível calcular a vazão máxima teórica que a sarjeta tem capacidade de suportar.

$$Q_{ts} = 0,375 \times Z \times \frac{\sqrt{I}}{n} \times y_0^{8/3} \quad (8)$$

$Q_{ts}$  = vazão teórica que pode ser escoada pela sarjeta (m<sup>3</sup>/s);

$n$  = coeficiente de rugosidade (Manning).

- **set\_capacidade\_reduzida\_sarjeta()** e **get\_capacidade\_reduzida\_sarjeta()**: Esses métodos ajustam a capacidade teórica para condições reais, considerando reduções causadas por obstáculos, rugosidade ou outros fatores;

$$Q_{rs} = Q_{ts} \times Coef.redução_{sarjeta} \quad (9)$$

$Q_{rs}$  = vazão reduzida que pode ser escoada pela sarjeta (m<sup>3</sup>/s);

$Coef.redução_{sarjeta}$  = coeficiente de redução da sarjeta.

- **set\_altura\_agua\_sarjeta\_sem\_bl()** e **get\_altura\_agua\_sarjeta\_sem\_bl()**: Definem e retornam a altura da água na sarjeta quando não há bocas de lobo para captação, sendo um indicador da eficácia da sarjeta;

Através da Equação 10 é possível calcular a altura da lâmina d'água na sarjeta sem a boca de lobo.

$$y_{cal} = \left( \frac{Q_T \times n}{0,375 \times \sqrt{I} \times Z} \right)^{3/8} \quad (10)$$

$y_{cal}$  = altura calculada em que a água se encontra na sarjeta sem boca de lobo (m).

- **set\_verificacao\_capacidade\_sarjeta()** e **get\_verificacao\_capacidade\_sarjeta()**: Verificam se a capacidade da sarjeta é suficiente para atender à vazão calculada;

Se  $y_{cal} < y_0$ , logo, sarjeta eficaz para o trecho;

Se  $y_{cal} > y_0$ , logo, sarjeta não eficaz para o trecho;

- **set\_largura\_agua\_sarjeta()** e **get\_largura\_agua\_sarjeta()**: Configuram e acessam a largura da lâmina d'água que se forma na sarjeta, útil para verificar a velocidade d'água na sarjeta;

$$L_{as} = Z \times y_{cal} \quad (11)$$

- **set\_velocidade\_agua\_sarjeta()** e **get\_velocidade\_agua\_sarjeta()**: Calculam e retornam a velocidade da água na sarjeta;

Através da Equação 12 é possível calcular velocidade da água na sarjeta.

$$V = \frac{2 \times Q_T}{y_{cal} \times L_{as}} \quad (12)$$

$V$  = velocidade de escoamento na sarjeta (m/s)

- **set\_verificacao\_velocidade\_agua\_sarjeta()** e **get\_verificacao\_velocidade\_agua\_sarjeta()**: Validam se a velocidade da água está dentro dos limites normativos já definidos ou aqueles estabelecidos pelo usuário, garantindo a segurança hidráulica;

Condirerando os limites já pré definidos:

Se  $0,6 \leq V \leq 3,0$ , logo, velocidade dentro dos limites permitidos;

Se  $V < 0,6$ , logo, velocidade está muito baixa;

Se  $V > 3,0$ , logo, velocidade está muito alta;

A classe **Sarjeta** é essencial para analisar a eficiência hidráulica das sarjetas na rede de drenagem, assegurando que elas suportem as vazões projetadas e indicando quando e onde são necessárias intervenções, como a instalação de bocas de lobo.

### 3.2.1.6 Classe *BocaDeLobo*

A classe **BocaDeLobo** é projetada para gerenciar as características e os cálculos relacionados às bocas de lobo em sistemas de drenagem urbana. Ela troca informações com as classes **Localidade**, **Sarjeta** e **Trecho** para determinar a quantidade de boca de lobo necessária em cada trecho de acordo com o tipo.

Métodos criados e suas finalidades:

- **set\_tabela\_especifica()** e **get\_tabela\_especifica()**: Estes métodos armazenam e acessam uma tabela associada à um tipo específico de boca de lobo, contendo o nome do trecho, tipo de boca de lobo e sua localização, capacidade de esgotamento e quantidade necessária;

- **set\_tabela\_opcoes()** e **get\_tabela\_opcoes()**: Estes métodos armazenam e acessam uma tabela com as opções disponíveis para diferentes configurações de bocas de lobo. Eles podem ser utilizados no processo de seleção de acordo com o projeto;
- **set\_largura\_da\_abertura\_bl\_guia()**, **get\_largura\_da\_abertura\_bl\_guia()**, **set\_altura\_da\_abertura\_bl\_guia()**, **get\_altura\_da\_abertura\_bl\_guia()**: Definem e retornam a largura da abertura de uma boca de lobo guia, influenciando diretamente na sua capacidade de captação;
- **set\_dimensao\_bl\_grelha\_paralela\_meio\_fio()**, **get\_dimensao\_bl\_grelha\_paralela\_meio\_fio()**, **set\_dimensao\_bl\_grelha\_perpendicular\_meio\_fio()** e **get\_dimensao\_bl\_grelha\_perpendicular\_meio\_fio()**: Defini e acessa as dimensões da boca de lobo com grelha paralelas e perpendiculares ao meio-fio respectivamente;
- **set\_espessura\_das\_grades()**, **get\_espessura\_das\_grades()**, **set\_comprimento\_das\_grades()**, **get\_comprimento\_das\_grades()**, **set\_quantidade\_de\_grades()** e **get\_quantidade\_de\_grades()**: Permitem configurar e acessar as espessuras, comprimentos e quantidades de barras em uma boca de lobo;
- Os métodos de capacidade permitem configurar e acessar as capacidades de diferentes tipos de bocas de lobo, como guia, grelha e combinadas. A capacidade representa a vazão máxima que cada tipo pode captar nas condições ideais;
- Para os cálculos da capacidade da boca de lobo tipo guia, pode haver dois casos distintos.

Se, a altura da lâmina d'água na sarjeta for menor ou igual que duas vezes a altura da abertura da boca de lobo no meio fio:

$$Q_{BL} = 1,7 \times L_{sol} \times y_{cal}^{3/2} \quad (13)$$

$Q_{BL}$  = capacidade da boca de lobo (m<sup>3</sup>/s);

$L_{sol}$  = comprimento da soleira (m);

Se, a altura da lâmina d'água na sarjeta for maior que duas vezes a altura da abertura da boca de lobo no meio fio:

$$Q_{BL} = 3,101 \times L_{sol} \times h^{3/2} \times \sqrt{\frac{2 \times y_{cal} - h}{2 \times h}} \quad (14)$$

$h$  = altura do meio fio (m).

Vale ressaltar que Tucci (2001) definiu que a formulação para o primeiro caso é válida para  $y_{cal}$  menor que a abertura na guia, mas para determinar a capacidade da boca de lobo quando  $y_{cal}$  for maior que uma vez, e menor que duas vezes a abertura na guia, a opção de qual critério utilizar fica a critério do projetista. Sendo assim para este trabalho foi definido que a Equação 13 será utilizada para  $y_{cal}$  menor ou igual a duas vezes a abertura na guia;

- Para os cálculos da capacidade da boca de lobo tipo grelha, também pode haver dois casos distintos.

Se, a altura da lâmina d'água na sarjeta for menor ou igual a 42 cm:

$$Q_{BL} = 1,7 \times P \times y_{cal}^{3/2} \quad (15)$$

$$P = (2 \times A) + (2 \times B) \quad (16)$$

$P$  = perímetro do orifício (m);

$A$  = comprimento do lado perpendicular da boca de lobo em relação à guia (m);

$B$  = comprimento do lado paralelo da boca de lobo em relação à guia (m);

Se, a altura da lâmina d'água na sarjeta for maior que 42 cm:

$$Q_{BL} = 2,91 \times A_L \times \sqrt{y_{cal}} \quad (17)$$

$$A_L = (A \times B) - (L_{BT} \times e_{BT} \times N_{BT}) - (L_{BL} \times e_{BL} \times N_{BL}) \quad (18)$$

$A_L$  = área livre da grade (m<sup>2</sup>);

$L_{BT}$  e  $L_{BL}$  = comprimento das barras transversais e longitudinais, respectivamente (m);

$e_{BT}$  e  $e_{BL}$  = espessura das barras transversais e longitudinais, respectivamente (m);

$N_{BT}$  e  $N_{BL}$  = quantidade de barras transversais e longitudinais, respectivamente;

Ressalta-se que Tucci (2001) estabelece que a formulação para o primeiro caso é válida para  $y_{cal}$  menor ou igual 12 cm, mas para determinar a capacidade da boca de lobo tipo grelha quando  $y_{cal}$  for maior que 12 cm e menor ou igual a 42 cm, a opção de qual critério utilizar fica a critério do projetista. Sendo assim para este trabalho foi definido que a Equação 15 será utilizada para  $y_{cal}$  menor ou igual a 42 cm;

- Já para os cálculos da capacidade da boca de lobo tipo combinada, é feito o somatório capacidade da boca de lobo tipo guia com a capacidade da boca de lobo tipo grelha.

- Os métodos para capacidade reduzida permitem calcular e retornar as capacidades reduzidas de todos os tipos de bocas de lobo. Essa capacidade nada mais é do que capacidade teórica calculada anteriormente multiplicada por um fator de redução presente no Quadro 7.

$$Q_{BLR} = Q_{BLT} \times Coef.redução_{BL} \quad (19)$$

$Q_{BLR}$  = capacidade reduzida da boca de lobo (m<sup>3</sup>/s);

$Q_{BLT}$  = capacidade teórica da boca de lobo (m<sup>3</sup>/s);

$Coef.redução_{BL}$  = coeficiente de redução da capacidade teórica da boca de lobo.

- Os métodos para quantidade permitem calcular e retornar a quantidade de bocas de lobo necessário em cada trecho de acordo com o seu tipo e localização.

$$N_{BL} = \frac{Q_T}{Q_{BLR}} \quad (20)$$

$N_{BL}$  = quantidade de boca de lobo necessária;

$Q_T$  = vazão total no trecho (m<sup>3</sup>/s);

$Q_{BLR}$  = capacidade reduzida da boca de lobo (m<sup>3</sup>/s);

### 3.2.1.7 Classe Profissional Engenharia

A classe **Profissional Engenharia** integra e aplica os cálculos detalhados previamente nas outras classes, utilizando fórmulas, verificações e parâmetros específicos para realizar análises práticas no dimensionamento de microdrenagem urbana. Por conter muitos cálculos e verificações, foram criados vários métodos de forma bem detalhada para facilitar a organização e entendimento do código, a seguir serão explicitados de forma breve o funcionamento dos principais.

Iniciando com o cadastramento dos trechos, o método **executar()** solicita ao usuário que passe algumas informações sobre o trecho, nome do trecho atual, nome da rua que esse trecho se encontra, área de contribuição, comprimento, cota a montante e cota a jusante. Além disso também é necessário informar se deseja considerar uma vazão futura para esse trecho e em caso positivo informar o valor dela. A última informação solicitada é a quantidade de trechos a montante do trecho cadastrado, caso seja diferente de zero, será solicitado o nome desses trechos. Esse método é executado até que o usuário cadastre todos os trechos e para informar o fim do cadastro basta digitar 0 (zero).

Logo após, os trechos são ordenados de acordo com as informações de trechos a montante que foram informados, de modo que se forme a rede de drenagem interligando todos

os trechos. Se algum trecho não tiver tido seus dados informados, então a solução computacional identifica e pergunta se o usuário deseja informar esses dados, em caso negativo, os cálculos não serão feitos para esse trecho por falta de dados, mas em caso positivo os dados serão solicitados um a um na mesma ordem já citada anteriormente e então é mostrado um relatório com todos os trechos cadastrados. Assim já é possível iniciar os processos de cálculos.

O método **calcular\_vazao()** calcula a vazão gerada no próprio trecho e depois soma-se com uma vazão futura se existir, originando então uma vazão total. Essa vazão futura é importante pensando em uma futura expansão da área dimensionada, ou a criação de uma nova área a montante da atual, para isso deve ser estimado o valor de uma vazão gerada por essas. Os resultados são armazenados na lista de vazões iniciais criada na classe localidade, permitindo análises subsequentes.

A próxima etapa importante é calcular a capacidade suportada pela sarjeta, através do método **calcular\_capacidade\_teorica\_sarjeta()** é executada a fórmula para a capacidade teórica, sendo que posteriormente através do método **calcular\_capacidade\_reduzida\_sarjeta()** e com o auxílio de um coeficiente de redução, é gerado a capacidade reduzida. Porém, conforme mostrado no Quadro 5 o mesmo não fornece coeficientes para alguns valores de inclinação transversal da sarjeta, sendo assim, através do método **interpolação()** foi necessário realizar uma interpolação nos valores dos coeficientes, a fim de descobrir os coeficientes que não estão tabelados.

A primeira verificação a ser feita conta com o auxílio do método **verificar\_capacidade\_sarjeta()**, com a finalidade de comparar a altura da lamina d'água calculada com a altura disponível na sarjeta. Se a lamina d'água calculada for menor do que a altura disponível, logo, a sarjeta é eficaz, mas se o resultado for o oposto, significa que a lamina d'água está extrapolando o limite permitido, com isso, será possível alterar algumas variáveis que influenciam diretamente no cálculo da lamina d'água, que poderia ser aumentar a largura da sarjeta, adotar um material para a sarjeta que apresente um coeficiente de rugosidade menor ou instalar uma boca de lobo intermediária. A decisão de qual alteração realizar fica a critério do usuário.

Continuando ainda na etapa das verificações, o método **verificar\_velocidade\_sarjeta()** funciona da seguinte forma, ele informa ao usuário que os limites cadastrados de velocidade da água na sarjeta é de 0,60 m/s e 3,0 m/s para velocidade mínima e máxima respectivamente, porém esse método também permite que o usuário cadastre novos limites. Após isso é feito a comparação do valor da velocidade da água na sarjeta com os limites definidos, caso esteja dentro do aceitável, os cálculos seguem para as próximas etapas, mas caso esteja muito alta ou

muito baixa (fora dos limites), é permitido que o usuário realize alterações em algumas variáveis, como a largura da sarjeta, alteração do material utilizado na sarjeta ou inserção de boca de lobo intermediária.

Estando tudo certo com as verificações da sarjeta, inicia-se a execução dos métodos para os cálculos necessários sobre as bocas de lobo. Uma tabela com as opções possíveis de bocas de lobo é mostrada ao usuário, sendo que para cada tipo de boca de lobo de acordo com a sua posição é atribuído um número que varia de um a oito. Com base nessa tabela e a execução do método **solicitar\_tipo\_bl()**, o usuário tem duas opções de escolha, acionar a tecla “s” se ele já sabe qual boca de lobo utilizar ou “n” caso ele ainda não tenha definido. Para a primeira opção, basta ele inserir o número correspondente ao tipo desejado e assim todos os cálculos referentes a essa boca de lobo serão realizados até definir a quantidade de boca de lobo necessária em cada trecho. Mas para o caso em que ele ainda não definiu qual boca de lobo utilizar, a ferramenta aciona todos os métodos referentes a todos os tipos bocas de lobo, com o objetivo de obter a quantidade das mesmas em cada trecho, desse modo o usuário pode analisar qual será a quantidade necessária de acordo com o tipo e posição da boca de lobo.

Para finalizar, é gerado um arquivo em formato *.xlsx*, com essas quantidades. Os métodos **salvar\_tabela\_especifica\_xlsx()** e **salvar\_tabela\_opcoes\_xlsx()**, salvam os valores com os nomes **TabelaEspecific** e **TabelaOpcoes**, para os casos citados anteriormente em que o usuário inseriu “s” ou “n”, respectivamente. As tabelas contêm cinco colunas para melhor detalhamento dos resultados, sendo elas, Trecho: contendo os nomes dos trechos, Tipo: que se refere ao tipo de boca de lobo, Localização: a posição dessa boca de lobo no trecho, Capacidade de esgotamento: valor da vazão suportada pela boca de lobo e Quantidade necessária: que é a quantidade de boca de lobo necessária naquele trecho de acordo com o seu tipo e localização.

Embora o método de cálculo da vazão tenha sido implementado, não foi possível incluir o cálculo da vazão excedente em cada trecho. Essa limitação deve-se ao nível básico de aprendizado em programação, aliada ao tempo restrito disponível para o desenvolvimento deste trabalho. No entanto, para mitigar essa ausência, o programa foi projetado de modo que a vazão de cada trecho não seja excedida. Isso foi garantido ao estipular que as bocas de lobo instaladas em cada trecho sejam capazes de suportar toda a vazão captada, sem permitir excedentes para o trecho seguinte. Além disso, o dimensionamento garante que, no mínimo, uma boca de lobo seja instalada em cada trecho, elevando a segurança do dimensionamento e o conforto dos usuários da via, evitando acúmulos indesejados de vazão.

### 3.2.1.8 Classe *Main*

Juntamente com a classe **ProfissionalEngenharia**, elas formam as duas principais classes do trabalho.

A classe **Main** foi desenvolvida para centralizar e coordenar a execução do programa de dimensionamento de microdrenagem urbana. Ela é responsável por integrar as demais classes do sistema, organizando o fluxo de dados e cálculos necessários para realizar as análises de retorno, desde a entrada de informações iniciais até a finalização dos cálculos. Essa classe funciona como o ponto principal de interação do usuário com o programa, solicitando entradas, gerenciando objetos de outras classes e exibindo os resultados.

A execução começa com a coleta de informações fundamentais, como o nome da cidade para onde será executado e da estação pluviométrica de onde serão extraídas algumas informações pluviométricas, tempo de retorno, tempo de concentração e coeficiente de escoamento. Esses dados são utilizados para inicializar os objetos das classes auxiliares, como **IntensidadePluviometrica**, Localidade e Trecho, que contêm as características e parâmetros necessários para os cálculos subsequentes. A intensidade pluviométrica, por exemplo, é calculada utilizando um método da classe **ProfissionalEngenharia**, e o valor obtido é armazenado no objeto correspondente.

Com o objeto da classe Trecho inicializado, a classe **Main** coordena os cálculos de vazão e declividade da via, aproveitando métodos especializados da classe **ProfissionalEngenharia**.

A parte do programa que trata da sarjeta é altamente interativa, permitindo ao usuário inserir parâmetros como largura da sarjeta, coeficiente de rugosidade, altura do meio fio, altura disponível da sarjeta e declividade transversal da sarjeta. Esses valores são usados para criar um objeto da classe Sarjeta e realizar cálculos como a capacidade teórica e reduzida, além de verificar as condições hidráulicas, incluindo a velocidade da água na sarjeta e análise da sua eficácia. Caso as condições hidráulicas não sejam atendidas, o programa sugere alterações na sua geometria, material ou no tipo de boca de lobo utilizada, permitindo iterações até alcançar uma configuração adequada.

A seleção do tipo de boca de lobo é realizada com base em uma tabela de códigos exibida ao usuário, que descreve os diferentes tipos disponíveis. Após a escolha, o programa utiliza a classe **BocaDeLobo** para calcular as capacidades e quantidades necessárias em cada trecho.

Por fim, a classe conclui a execução exibindo os resultados finais e as configurações exigidas, consolidando todo o processo de dimensionamento em uma única execução. Essa

estrutura modular e interativa permite que a classe *Main* funcione como um núcleo que conecta todas as partes do sistema, garantindo robustez e flexibilidade ao programa.

## 4 RESULTADOS E DISCUSSÃO

Para a elaboração e execução da solução computacional foi utilizado a versão Python 3.8.1 (lançado em 18 de dezembro de 2019) e para garantir o seu funcionamento adequado é necessário instalar as bibliotecas externas Pandas (versão 2.0.3) e *Tabulate* (versão 0.9.0). Essas bibliotecas desempenham papéis fundamentais, como a manipulação eficiente de tabelas e a apresentação de dados de maneira estruturada. Contudo, vale ressaltar, que o usuário pode adequar o código às novas bibliotecas e/ou novas versões das mesmas que necessariamente precisam ser utilizadas.

O código fonte estará disponível na plataforma *Github* pelo link: [https://github.com/rafa120296/TCC\\_Rafael-Solucao\\_Computacional\\_Piumhi.git](https://github.com/rafa120296/TCC_Rafael-Solucao_Computacional_Piumhi.git)

A sequência de execução do modelo é descrita a seguir e suas respectivas imagens didáticas (*prints*) podem ser verificados nos Apêndices A e B.

A execução da solução computacional pode ser realizada de duas maneiras principais:

1. Através do IDLE do Python:

- Abra o IDLE do Python.
- No menu superior, clique em *File* e em seguida em *Open*.
- Localize a pasta que contém os arquivos da solução computacional e selecione o arquivo da classe *Main*.
- Após abrir o arquivo, vá em *Run* no menu superior e clique em *Run Module* (ou pressione a tecla F5). A solução computacional será executado e o usuário poderá interagir com as opções disponíveis diretamente na interface do IDLE.

2. Através do *Prompt* de Comando do Windows:

- Abra o *Prompt* de Comando do Windows.
- Navegue até a pasta que contém os arquivos da solução computacional utilizando o comando `cd caminho_da_pasta` (substitua `caminho_da_pasta` pelo diretório onde as classes estão salvas).
- Execute o comando `python main.py` isso iniciará a execução da solução computacional diretamente no terminal.

Essas duas opções oferecem flexibilidade ao usuário, permitindo que escolha a interface mais conveniente para executar a solução computacional. A fim de comprovar a veracidade da solução computacional, a mesma será executada através do IDLE utilizando dados reais coletados em campo durante o projeto de pesquisa já mencionado. Os cálculos serão

comparados com os obtidos por meio da planilha de dimensionamento fornecida pelo professor Dr. Germano de Oliveira Mattosinho.

Em campo, durante o projeto de pesquisa foi feito um trabalho mais detalhado para analisar todas as geometrias e isto foi apresentado no relatório final da pesquisa e publicação (Silva *et al.*, 2019b e 2020; Silva e Mattosinho, 2021). Neste trabalho serão consideradas geometrias padronizadas e uniformes, como se fosse o estudo e dimensionamento de um loteamento novo.

A primeira etapa do processo consiste no preenchimento das informações da região em estudo. Em seguida é necessário realizar o cadastro dos trechos, juntamente com as respectivas informações solicitadas pela solução computacional. De forma didática, para este trabalho foram registrados, e apresentados nos Apêndices, 12 trechos. Ressalta-se que o usuário necessita incluir os dados de maneira adequada no momento de preenchimento de dados e de se utilizar “ponto” como separador decimal.

Após o cadastro, a solução computacional gera um relatório detalhado que inclui a quantidade de trechos a montante e a jusante de cada trecho, bem como a vazão total correspondente a cada um deles. Durante a coleta de dados em campo, observou-se uma variação nas dimensões das sarjetas ao longo dos trechos, com alturas de meio-fio variando entre 12 cm e 15 cm e larguras de sarjetas entre 21 cm e 30 cm.

No entanto, como a solução computacional considera valores constantes para essas variações, introduziu-se 15 cm para a altura do meio-fio e 30 cm para a largura da sarjeta, padronizando os cálculos e garantindo consistência nos resultados.

Concluído o cálculo da velocidade da água na sarjeta e verificando sua eficácia, prossegue-se para a etapa de inserção dos dados das bocas de lobo. Neste trabalho, optou por realizar os cálculos considerando os oito tipos de bocas de lobo disponíveis no programa, ampliando as opções de análise.

Por fim, a solução computacional gera o resultado final, que determina a quantidade mínima de bocas de lobo permitidas em cada trecho. Como os cálculos foram realizados para todos os tipos de bocas de lobo, o usuário dispõe de uma análise comparativa, podendo decidir qual tipo será mais adequado e viável para cada situação específica. Os resultados obtidos através da solução computacional foram desmembrados para melhor visualização e estão disponíveis nas Figuras 25a, 25b, 25c, 25d e 25e, além disso nas Figuras 26a e 26b é possível observar os resultados obtidos por meio da planilha, que também foi desmembrada.

Figura 25a - Resultados da solução computacional

Trecho	Tipo	Localização	Capacidade de esgotamento [l/s]	Quantidade necessária
A109	Guia	Ponto baixo	36.08	2
A110	Guia	Ponto baixo	49.91	1
A116	Guia	Ponto baixo	27.69	4
A117	Guia	Ponto baixo	13.13	2
A118	Guia	Ponto baixo	18.01	2
A119	Guia	Ponto baixo	38.98	2
A120	Guia	Ponto baixo	43.97	3
A121	Guia	Ponto baixo	13.29	2
E15	Guia	Ponto baixo	29.23	1
E16	Guia	Ponto baixo	28.67	1
E17	Guia	Ponto baixo	48.4	3
E18	Guia	Ponto baixo	24.63	3
A109	Guia	Ponto intermediário	36.08	2
A110	Guia	Ponto intermediário	49.91	1
A116	Guia	Ponto intermediário	27.69	4
A117	Guia	Ponto intermediário	13.13	2
A118	Guia	Ponto intermediário	18.01	2
A119	Guia	Ponto intermediário	38.98	2

Fonte: Autoria própria, 2025.

Figura 25b - Resultados da solução computacional

A120	Guia	Ponto intermediário	43.97	3
A121	Guia	Ponto intermediário	13.29	2
E15	Guia	Ponto intermediário	29.23	1
E16	Guia	Ponto intermediário	28.67	1
E17	Guia	Ponto intermediário	48.4	3
E18	Guia	Ponto intermediário	24.63	3
A109	Grelha	Ponto baixo	62.01	1
A110	Grelha	Ponto baixo	85.78	1
A116	Grelha	Ponto baixo	47.6	2
A117	Grelha	Ponto baixo	22.56	1
A118	Grelha	Ponto baixo	30.96	2
A119	Grelha	Ponto baixo	67.01	1
A120	Grelha	Ponto baixo	75.58	2
A121	Grelha	Ponto baixo	22.84	1
E15	Grelha	Ponto baixo	50.24	1
E16	Grelha	Ponto baixo	49.28	1
E17	Grelha	Ponto baixo	83.19	2
E18	Grelha	Ponto baixo	42.34	2
A109	Grelha longitudinal	Ponto intermediário	74.41	1

Fonte: Autoria própria, 2025.

Figura 25c - Resultados da solução computacional

A110	Grelha longitudinal	Ponto intermediário	102.94	1
A116	Grelha longitudinal	Ponto intermediário	57.12	2
A117	Grelha longitudinal	Ponto intermediário	27.07	1
A118	Grelha longitudinal	Ponto intermediário	37.15	1
A119	Grelha longitudinal	Ponto intermediário	80.41	1
A120	Grelha longitudinal	Ponto intermediário	90.7	1
A121	Grelha longitudinal	Ponto intermediário	27.41	1
E15	Grelha longitudinal	Ponto intermediário	60.29	1
E16	Grelha longitudinal	Ponto intermediário	59.14	1
E17	Grelha longitudinal	Ponto intermediário	99.83	2
E18	Grelha longitudinal	Ponto intermediário	50.81	2
A109	Grelha transversal	Ponto intermediário	62.01	1
A110	Grelha transversal	Ponto intermediário	85.78	1
A116	Grelha transversal	Ponto intermediário	47.6	2
A117	Grelha transversal	Ponto intermediário	22.56	1
A118	Grelha transversal	Ponto intermediário	30.96	2
A119	Grelha transversal	Ponto intermediário	67.01	1
A120	Grelha transversal	Ponto intermediário	75.58	2
A121	Grelha transversal	Ponto intermediário	22.84	1

Fonte: Autoria própria, 2025.

Figura 25d - Resultados da solução computacional

E15	Grelha transversal	Ponto intermediário	50.24	1
E16	Grelha transversal	Ponto intermediário	49.28	1
E17	Grelha transversal	Ponto intermediário	83.19	2
E18	Grelha transversal	Ponto intermediário	42.34	2
A109	Combinada	Ponto baixo	109.92	1
A110	Combinada	Ponto baixo	152.07	1
A116	Combinada	Ponto baixo	84.38	2
A117	Combinada	Ponto baixo	40	1
A118	Combinada	Ponto baixo	54.89	1
A119	Combinada	Ponto baixo	118.78	1
A120	Combinada	Ponto baixo	133.98	1
A121	Combinada	Ponto baixo	40.49	1
E15	Combinada	Ponto baixo	89.07	1
E16	Combinada	Ponto baixo	87.36	1
E17	Combinada	Ponto baixo	147.48	1
E18	Combinada	Ponto baixo	75.06	1
A109	Combinada longitudinal	Ponto intermediário	111.61	1
A110	Combinada longitudinal	Ponto intermediário	154.41	1
A116	Combinada longitudinal	Ponto intermediário	85.68	2

Fonte: Autoria própria, 2025.

Figura 25e - Resultados da solução computacional

A117	Combinada longitudinal	Ponto intermediário	40.61	1
A118	Combinada longitudinal	Ponto intermediário	55.73	1
A119	Combinada longitudinal	Ponto intermediário	120.61	1
A120	Combinada longitudinal	Ponto intermediário	136.05	1
A121	Combinada longitudinal	Ponto intermediário	41.11	1
E15	Combinada longitudinal	Ponto intermediário	90.44	1
E16	Combinada longitudinal	Ponto intermediário	88.71	1
E17	Combinada longitudinal	Ponto intermediário	149.74	1
E18	Combinada longitudinal	Ponto intermediário	76.21	1
A109	Combinada transversal	Ponto intermediário	93.01	1
A110	Combinada transversal	Ponto intermediário	128.68	1
A116	Combinada transversal	Ponto intermediário	71.4	2
A117	Combinada transversal	Ponto intermediário	33.84	1
A118	Combinada transversal	Ponto intermediário	46.44	1
A119	Combinada transversal	Ponto intermediário	100.51	1
A120	Combinada transversal	Ponto intermediário	113.37	1
A121	Combinada transversal	Ponto intermediário	34.26	1
E15	Combinada transversal	Ponto intermediário	75.36	1
E16	Combinada transversal	Ponto intermediário	73.92	1
E17	Combinada transversal	Ponto intermediário	124.79	1
E18	Combinada transversal	Ponto intermediário	63.51	1

Fonte: Autoria própria, 2025.

Figura 26a - Resultados da planilha

Trecho	Capacidades reduzidas para todos os tipos de boca de lobo de acordo com a sua posição no trecho (l/s)							
	Ponto baixo			Ponto Intermediário				
	Simplex	Grelha	Combinada	Simplex	Grelha longitudinal	Grelha na transversal ou longitudinal com barras transversais	Combinada com barras longitudinais	Combinada com barras transversais
A109	36,08	62,01	109,92	36,08	74,41	62,01	111,61	93,01
A110	49,91	85,78	152,07	49,91	102,94	85,78	154,41	128,68
A116	27,69	47,60	84,38	27,69	57,12	47,60	85,68	71,40
A117	13,13	22,56	40,00	13,13	27,07	22,56	40,61	33,84
A118	18,01	30,96	54,89	18,01	37,15	30,96	55,73	46,44
A119	38,98	67,01	118,78	38,98	80,41	67,01	120,61	100,51
A120	43,97	75,58	133,98	43,97	90,70	75,58	136,05	113,37
A121	13,29	22,84	40,49	13,29	27,41	22,84	41,11	34,26
E15	29,23	50,24	89,07	29,23	60,29	50,24	90,44	75,36
E16	28,67	49,28	87,36	28,67	59,14	49,28	88,71	73,92
E17	48,40	83,19	147,48	48,40	99,83	83,19	149,74	124,79
E18	24,63	42,34	75,06	24,63	50,81	42,34	76,21	63,51

Fonte: Autoria própria, 2025.

Figura 26b - Resultados da planilha

Trecho	Quantidade de boca de lobo necessária de acordo com o e posição no trecho							
	Ponto baixo			Poi				
	Simple	Grelha	Combinada	Simple	Grelha longitudinal	Grelha na transversal ou longitudinal com barras transversais	Combinada com barras longitudinais	Combinada com barras transversais
A109	2	1	1	2	1	1	1	1
A110	1	1	1	1	1	1	1	1
A116	4	2	2	4	2	2	2	2
A117	2	1	1	2	1	1	1	1
A118	2	2	1	2	1	2	1	1
A119	2	1	1	2	1	1	1	1
A120	3	2	1	3	1	2	1	1
A121	2	1	1	2	1	1	1	1
E15	1	1	1	1	1	1	1	1
E16	1	1	1	1	1	1	1	1
E17	3	2	1	3	2	2	1	1
E18	3	2	1	3	2	2	1	1

Fonte: Autoria própria, 2025.

O programa demonstrou sua eficácia ao ser testado no município de Piumhi - MG e seus resultados comparados com a planilha, permitindo o dimensionamento das sarjetas e a quantificação das bocas de lobo em áreas críticas. A solução desenvolvida apresenta uma abordagem intuitiva, reduz o tempo necessário para o dimensionamento e diminui a probabilidade de erros manuais durante o processo. Essa solução computacional pode ser adaptada e aplicada a diferentes localidades, auxiliando gestores e engenheiros na tomada de decisões.

Com base no artigo de Melo *et al.* (2020), que apresenta o desenvolvimento de um plug-in em Python para facilitar a entrada de dados georreferenciados no software QGIS, destaca-se a relevância dessa ferramenta no contexto da engenharia urbana. O plug-in foi concebido para permitir que operadores, mesmo sem conhecimentos técnicos avançados, possam inserir informações coletadas em campo diretamente no sistema de informação geográfica, simplificando o processo de atualização de redes de infraestrutura, como esgoto e drenagem. Essa solução não apenas otimiza o gerenciamento de dados críticos de infraestrutura urbana, mas também é especialmente vantajosa para cidades com recursos limitados. No presente trabalho, a proposta de Melo *et al.* (2020) encontra aplicação direta no contexto da drenagem urbana, reforçando como a automação e a integração de tecnologias podem aprimorar processos de engenharia e melhorar a qualidade dos serviços urbanos.

Braga, Muzzi e Viana (2022), através de um estudo de caso desenvolveram um *software* na linguagem de programação C# para cálculo da vazão de projeto e utilizaram o *software* ArcGIS para delimitação da bacia hidrográfica. Os cálculos consideraram parâmetros extraídos

do ArcGIS, como área da bacia, tipo e uso do solo, cotas altimétricas, comprimento do talvegue e dados hidrológicos regionais obtidos pelo Plúvio 2.1. Os autores concluíram que, a criação do modelo computacional prova a relevância de aplicar conhecimentos técnicos para melhorar o dimensionamento de obras de engenharia, tornando o processamento de dados mais eficiente e possibilitando simulações mais precisas.

Saboia e Lopardo (2015) apresentam o software “Regionaliza 2014”, desenvolvido em Python 2.7, como uma ferramenta avançada para a obtenção de parâmetros hidrológicos em qualquer região do estado do Paraná.

Desenvolvido para aprimorar projetos anteriores realizados pelos Institutos Lactec, tem como objetivo principal obter o contorno da bacia a montante e resultados, como curvas de permanência, vazões máximas e mínimas, número de dias sem chuva, curva de regularização intra-anual, hidrograma de cheias e tempo de recessão, sendo que o usuário precisa informar apenas a localização da exutória da bacia de interesse para obter tais resultados. Com um design de uso simples, o *software* pode ser executado em diversos sistemas operacionais, desde que os compiladores e pacotes necessários do Python estejam devidamente instalados.

Os estudos supracitados, juntamente com o modelo proposto neste estudo, corroboram a necessidade de incorporar ferramentas tecnológicas para lidar com os desafios do planejamento urbano e de engenharia. Além de otimizar processos e aumentar a precisão, essas tecnologias promovem uma abordagem mais eficiente e segura, atendendo às demandas crescentes de projetos complexos. O modelo desenvolvido neste trabalho também apresenta um significativo benefício didático, destacando-se como uma ferramenta acessível para o aprendizado e aplicação prática de conceitos relacionados à drenagem urbana. Por sua facilidade de uso e capacidade de simulação, pode ser integrado em disciplinas de drenagem urbana e áreas afins, auxiliando alunos e profissionais a compreenderem, de forma prática e interativa, os fundamentos e as implicações do dimensionamento de sistemas de drenagem. Dessa forma, reforça-se não apenas a relevância técnica do programa, mas também seu potencial educativo para formação e capacitação no campo da engenharia.

#### **4.1 Comparação entre cálculos e intervenções realizadas em Piumhi - MG**

Em 2019, a Rua Amazonas, localizada em Piumhi - MG, enfrentava sérios problemas de alagamento, conforme registrado nas imagens da época e descrito no Capítulo 1. Esses eventos destacaram a necessidade urgente de intervenções no sistema de drenagem urbana, evidenciando a insuficiência da infraestrutura existente para lidar com os volumes de precipitação.

Entre 2020 e 2022, foram realizadas intervenções significativas na área, incluindo a instalação de novas bocas de lobo e melhorias na infraestrutura já existente. Essas ações visaram mitigar os problemas de alagamento tanto na Rua Amazonas quanto na Rua Severo Veloso, que apresentava vários trechos sem bocas de lobo. A ausência desses dispositivos gerava acúmulo excessivo de água pluvial, parte da qual era direcionada para a Rua Amazonas, agravando os problemas de drenagem na região.

Com base nos dados obtidos durante a pesquisa, foi possível comparar a quantidade de bocas de lobo implantadas nessas intervenções com a quantidade calculada por meio da planilha desenvolvida no projeto de pesquisa. Os resultados indicaram que o número atual de bocas de lobo atende à vazão de projeto, sem ocorrência de escoamento excedente, *a priori*, evidenciando a eficácia das intervenções realizadas.

## 5 CONCLUSÕES

O desenvolvimento deste programa em Python para o dimensionamento de elementos de microdrenagem urbana destaca a essencialidade da integração entre os conhecimentos de engenharia e as ferramentas computacionais modernas, ampliando as capacidades dos profissionais da área. Ao adotar uma abordagem estruturada e modular, baseada em classes, foi possível organizar de forma eficiente e precisa cálculos complexos, como os relacionados à vazão, capacidade das sarjetas e escolha das bocas de lobo, estabelecendo um sistema robusto e de fácil utilização.

A flexibilidade do programa é uma de suas principais qualidades, permitindo simular cenários variados de drenagem urbana ao considerar parâmetros locais cruciais, como a intensidade pluviométrica, coeficientes de escoamento e características geométricas das áreas estudadas. Essa adaptabilidade confere ao programa um enorme potencial de aplicação em diferentes contextos, oferecendo uma solução prática e acessível para engenheiros civis, e outros profissionais, que enfrentam os desafios crescentes de enchentes e alagamentos em ambientes urbanos.

A centralização das operações na classe *Main* e a utilização de métodos de dimensionamento proporcionaram uma execução clara e linear do programa, facilitando a interação com o sistema. Esse modelo de implementação torna o uso da ferramenta intuitivo e eficiente, permitindo que até profissionais menos familiarizados com programação possam utilizá-la de forma prática e sem complexidade excessiva. Além de fornecer resultados precisos e consistentes, o programa auxilia na tomada de decisões, sugerindo ajustes necessários e destacando as variações em cada etapa do processo de dimensionamento, o que facilita a análise e o planejamento das soluções.

Este trabalho enfatiza o papel fundamental da programação na engenharia civil, não apenas como uma ferramenta complementar, mas como um recurso essencial para a resolução de problemas urbanos e o planejamento de soluções mais eficientes e sustentáveis. Embora o programa tenha alcançado seus objetivos iniciais, ele também abre portas para aprimoramentos futuros, como a inclusão de análises custo-benefício, a integração com bases de dados climáticos, a automação no ajuste de parâmetros e dimensionamento completo do sistema de drenagem urbana, por exemplo com foco em loteamentos. Dessa forma, a solução computacional se apresenta como uma contribuição significativa para o avanço das tecnologias aplicadas à engenharia civil, à drenagem urbana e ao planejamento sustentável das cidades.

## REFERÊNCIAS

**104 FM | Piumhi MG.** Disponível em: <https://www.facebook.com/104fmoficial>. 2019. Acesso em: 20 out. 2020.

**ABDI. Nova Estratégia BIM BR: Caminhos para a Construção 4.0 Anuncia Plano de Trabalho e Curso para Agentes Públicos.** 2024. Disponível em: <https://www.abdi.com.br/nova-estrategia-bim-br-caminhos-para-a-construcao-4-0-anuncia-plano-de-trabalho-e-curso-para-agentes-publicos/>. Acesso em: 1 dez. 2024.

**BELLO, I. P. Regionalização de Chuva Intensa para o Estado de Minas Gerais.** Dissertação de Pós-Graduação—Universidade Federal de Lavras, 2018.

**BELO HORIZONTE. Instrução Técnica Para Elaboração De Estudos E Projetos De Drenagem.** 2022. Disponível em: <https://prefeitura.pbh.gov.br/obras-e-infraestrutura/informacoes/publicacoes/instrucao-estudos-e-projetos-de-drenagem>. Acesso em: 3 jun. 2024.

**BENITES, I. et al. Análise Do Sistema De Drenagem Urbana: Estudo De Caso Das Galerias De Águas Pluviais Da Intersecção Da Rua Governador Nei Braga Com A Avenida Brasil Na Cidade De Umuarama – PR.** 2017.

**BÖES, J. S.; PATZLAFF, J. O. Tecnologia da Informação e Comunicação (TIC) Aplicada ao Controle de Qualidade de Obras – Estudo de Caso.** Revista de Arquitetura IMED, v. 5, jun. 2016.

**BORGES, L. E. Python Para Desenvolvedores.** São Paulo: Novatec, 2014.

**BOTELHO, M. H. C. Águas De Chuva: Engenharia Das Águas Pluviais Nas Cidades.** 2017. Disponível em: <https://books.google.com.br/books?id=7ytdDwAAQBAJ&printsec=frontcover&dq=%C3%81guas+de+chuvas:+engenharia+das+%C3%A1guas+pluviais+nas+cidades&hl=pt-BR&sa=X&ved=2ahUKEwiF0NOioLfrAhVcGbkGHagoBoQQ6AEwAHoECAIQAg#v=onepage&q&f=false>. Acesso em: 20 ago. 2023.

**BRAGA, L. DA S.; MUZZI, A. L.; VIANA, T. M. Análise De Parâmetros Hidráulicos E Metodologias De Dimensionamento Para Canais De Drenagem: Estudo De Caso - Itabira/MG.** Revista Engenharia de Interesse Social, v. 7, n. 9, p. 37–65, 28 set. 2022.

**BRANCHIER, H. S. Contribuições Dos Softwares Na Aprendizagem De Análise E Cálculo De Elementos Estruturais.** Trabalho de Conclusão de Curso—Universidade do Vale do Taquari, 2017.

**BRASIL. Lei 10.257, de 10 de Julho de 2001. Planalto.Gov.Br,** 2001. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/leis/leis\\_2001/110257.htm](http://www.planalto.gov.br/ccivil_03/leis/leis_2001/110257.htm). Acesso em: 12 out. 2024.

**SNIS. Drenagem E Manejo Das Águas Pluviais Urbanas.** 2022. Disponível em: <https://www.gov.br/cidades/pt-br/aceso-a-informacao/acoes-e-programas/saneamento/snis/painel/ap>. Acesso em: 10 set. 2024.

BRASIL. **Manual Para Apresentação De Propostas Para Sistemas De Drenagem Urbana Sustentável E De Manejo De Águas Pluviais**. Secretaria Nacional De Saneamento Ambiental, 2023.

BRASIL. **D11888. Planalto.Gov.Br**, 22 jan. 2024a. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/\\_ato2023-2026/2024/Decreto/D11888.htm](https://www.planalto.gov.br/ccivil_03/_ato2023-2026/2024/Decreto/D11888.htm). Acesso em: 6 set. 2024.

BRASIL. **MDIC Apresenta Plano De Trabalho Da Nova Estratégia BIM BR Para 2025-2027**. 2024b. Disponível em: <https://www.gov.br/mdic/pt-br/assuntos/noticias/2024/novembro/mdic-apresenta-plano-de-trabalho-para-2025-2027-da-nova-estrategia-bim-br>. Acesso em: 1 dez. 2024.

BURIOL, T. M.; MARCO, B.; ARGENTA, M. A. **Acelerando O Desenvolvimento Eo Processamento De Análises Numéricas Computacionais Utilizando Python E Cuda**. ResearchGate, 2009.

CABRAL, L. DO N.; CÂNDIDO, G. A. **Urbanização, Vulnerabilidade, Resiliência: Relações Conceituais E Compreensões De Causa E Efeito**. Urbe. Revista Brasileira de Gestão Urbana, v. 11, 2019.

CANHOLI, A. **Drenagem Urbana E Controle De Enchentes**. 2. ed. São Paulo: Oficina de Textos, 2014.

CARVALHO, C. V. DE A.; MUSCI, M. **Utilizando Python No Ensino Da Construção Naval: Experiências De Um Curso De Extensão**. Revista Naval E Oceânica, 2024.

CARVALHO, T. L. E. **Orientação A Objetos: Aprenda Seus Conceitos E Suas Aplicabilidades De Forma Efetiva**. [s.l.] Alura, 2016.

CARVALHO, V. A. D.; TEIXEIRA, G. F. **Programação Orientada A Objetos: Curso Técnico Em Informática**. Colatina: IFES, p. 134, 2012.

CAUDURO, F.; MÜLLER, C. R.; ROSA, G. G. **Proposta De Drenagem Sustentável Para Uma Municipalidade De Pequeno Porte – Estudo De Caso**. Águas Subterrâneas, v. 33, n. 2, 28 maio 2019.

CETESB. **Drenagem Urbana: Manual De Projeto**. 3. ed. São Paulo: Companhia de Tecnologia de Saneamento Ambiental, 1986. p. 464.

CHEN, D. Y. **Análise De Dados Com Python E Pandas**. 1. ed. [s.l.] Novatec, 2018.

CHRISTOFIDIS, H. DO V. **Drenagem Urbana Sustentável : Análise Do Uso Do Retrofit**. Dissertação de Mestrado—Universidade de Brasília, 2010.

COELHO, F. C. **Computação Científica Com Python - Uma Introdução À Programação Para Cientistas**. 2007.

COELHO, M. M. L. P.; LIMA, J. G. DE A. **Eficiência Hidráulica De Bocas De Lobo Situadas Em Sarjetas De Greide Contínuo**. Revista Brasileira de Recursos Hídricos, v. 16, n. 2, jun. 2011.

CORRÊA, M. M. R. **Contribuição Para Avaliação E Gestão De Riscos De Inundações. Caso De Estudo: Bacia Hidrográfica Do Rio Nabão.** Dissertação de Mestrado—Universidade Nova de Lisboa. 2013.

COSTA, D. A. T. *et al.* **Qualidade Ambiental De Sub-Bacia Hidrográfica Urbana: Um Estudo De Caso De Alfenas-Mg.** *Ciência e Natura*, v. 40, n. 20, 27 mar. 2018.

CRUZ, G. D. **Entendendo Métodos Deprecated: O Que São E Como Lidar Com Eles.** 2024. Disponível em: <https://giovanidacruz.com.br/entendendo-metodos-deprecated-o-que-sao-e-como-lidar-com-eles/>.

CRUZ, M. A. S.; SOUZA, C. F.; TUCCI, C. E. M. **Controle Da Drenagem Urbana No Brasil: Avanços E Mecanismos Para Sua Sustentabilidade.** *Abrhidro.org.br*, 2007.

DANTAS, H. S.; SOUSA, J. M. M. S.; MELO, H. C. **The Importance of City Information Modeling (CIM) for Cities' Sustainability.** *IOP Conference Series: Earth and Environmental Science*, v. 225, p. 012074, 24 fev. 2019.

DIAS, F. S.; ANTUNES, P. T. DA S. C. **Estudo Comparativo De Projeto De Drenagem Convencional E Sustentável Para Controle De escoamento Superficial Em Ambientes Urbanos.** Projeto de Graduação—Universidade Federal do Rio de Janeiro, 2010.

DIAZ, D. **Um Guia Para Iniciantes Em Programação Orientada A Objetos (POO) Python.** 2023.

DISTRITO FEDERAL. **Manual De Drenagem E Manejo De Águas Pluviais Urbanas Do Distrito Federal.** 2023. Disponível em: [https://www.adasa.df.gov.br/images/storage/area\\_de\\_atuacao/drenagem\\_urbana/regula%C3%A7%C3%A3o/Manual\\_de\\_drenagem\\_e\\_manejo\\_de\\_aguas\\_urbanas/Manual%20Drenagem%20Adasa\\_Digital\\_rev\\_compactado-2\\_compressed.pdf](https://www.adasa.df.gov.br/images/storage/area_de_atuacao/drenagem_urbana/regula%C3%A7%C3%A3o/Manual_de_drenagem_e_manejo_de_aguas_urbanas/Manual%20Drenagem%20Adasa_Digital_rev_compactado-2_compressed.pdf). Acesso em: 26 set. 2024.

DNIT. **Manual De Hidrologia Básica Para Estruturas De Drenagem.** 2. ed. Rio de Janeiro: Departamento Nacional de Infra-Estrutura de Transportes, 2005. p. 133.

DNIT. **Diretrizes Básicas Para Elaboração De Estudos E Projetos Rodoviários Escopos Básicos/Instruções De Serviço.** 3. ed. Rio de Janeiro: Departamento Nacional de Infra-Estrutura de Transportes, 2006a. p. 484.

DNIT. **Manual De Pavimentação.** 3. ed. Rio de Janeiro: Departamento Nacional de Infra-Estrutura de Transportes, 2006b.

DOWNEY, A. B. **Pense Em Python: Pense Como Um Cientista Da Computação.** 1. ed. [s.l.] Novatec, 2016.

EVALDO, A. **Sublime, Atom Ou Vscod: Qual Melhor Editor De Código?** 2021. Disponível em: [https://www.alura.com.br/artigos/sublime-atom-vscode-qual-melhor-editor-de-codigo?srsId=AfmBOopbljzoTTO\\_jEyWwnHw0kyzQCs4P4X5fUyeflHXQhWiPveYmnC4](https://www.alura.com.br/artigos/sublime-atom-vscode-qual-melhor-editor-de-codigo?srsId=AfmBOopbljzoTTO_jEyWwnHw0kyzQCs4P4X5fUyeflHXQhWiPveYmnC4). Acesso em: 21 nov. 2024.

FARINELLI, F. **Conceitos Básicos De Programação Orientada A Objetos.** *ResearchGate*, 2007.

FELIZARDO, L. M. **Aplicação De Sistema De Informações Geográficas (SIG) Para Modelagem De Eventos Críticos De Vazão Em Uma Microbacia Urbana**. Unesp.br, 2016.

FERNANDES, D. **Implementação De Um Sistema De Gestão Municipal Com Recurso A Sistemas De Informação Geográfica**. Dissertação de Mestrado—Instituto Superior De Engenharia De Lisboa, 2016.

FERREIRA, L. DE A. L. **Programação Visual Em Desenvolvimento E Design De Jogos Digitais Estudo E Análise Da Programação Visual Como Método Alternativo Para Lógica Em Jogos Digitais**. Dissertação de Mestrado-Universidade Beira Interior, 2023.

FIORIN, P. **Curitiba Aplica BIM Autodesk Para Projetos E Obras De Mobilidade Urbana**. 2023. Disponível em: <https://blogs.autodesk.com/por-dentro-da-autodesk-brasil/2023/01/30/curitiba-aplica-bim-autodesk-para-projetos-e-obras-de-mobilidade-urbana/>. Acesso em: 6 set. 2024.

FIORIO, P. R. *et al.* **Comparação De Equações De Chuvas Intensas Para Localidades Do Estado De São Paulo**. Scientific Electronic Library Online, v. 32, n. 6, dez. 2012.

FITZ, P. R. **Geoprocessamento Sem Complicação**. São Paulo: Oficina de Textos, 2008.

FURGERI, S. **Programação Orientada A Objetos – Conceitos E Técnicas**. [s.l.] Saraiva, 2018.

FURGERI, S. **Introdução À Programação Em Python**. [s.l.] Senac, 2021.

GALVÃO, L. Q. *et al.* **O Uso Do Python Na Construção De Simuladores Computacionais: Proposições E Potencialidades Para O Ensino De Física**. Portal de Periódicos UFSC, v. 39, n. 1, p. 204–237, abr. 2022.

GARLET, D.; BIGOLIN, N. M.; SILVEIRA, S. R. **Ensino De Programação De Computadores Na Educação Básica: Um Estudo De Caso**. Revista Eletronica de Sistemas de Informação e Gestão Tecnológica, v. 8, n. 2, 2018.

GERMANIO, L. **Implementação Orientada A Objetos Da Solução De Problemas Estruturais Dinâmicos Via Método Dos Elementos Finitos**. Dissertação de Pós-Graduação-Universidade Federal De Minas Gerais, 2005.

**GPRH - Grupo De Pesquisa Em Recursos Hídricos - Universidade Federal De Viçosa**. Disponível em: <https://www.gprh.ufv.br/?area=softwares>.

GRÄBNER, C. **Getting Started With The Spyder IDE**. 24 jan. 2020.

GUEDES, H. A. S. **Microdrenagem**. 2015. Disponível em: <https://slideplayer.com.br/slide/4874736/>. Acesso em: 13 nov. 2024.

IMPULCETTO, B. **Controle De Cheias Através De Medidas Estruturais**. Trabalho de Conclusão de Curso-Anhanguera, 2017.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **IBGE Censo 2010**. Disponível em: <https://censo2010.ibge.gov.br/sinopse/index.php?uf=31&dados=29>. Acesso em: 29 nov. 2024.

JIMÉNEZ, Ó. R. **Python A Fundo**. 1. ed. [s.l.] Marcombo, 2021.

KNOPP, L.; PARAIZO, R. C. **Plano Diretor Como Instrumento Da Gestão Universitária: Uma Abordagem Sob A Ótica Da Modelagem Da Informação Da Cidade (CIM)**. III SIMPÓSIO NACIONAL DE GESTÃO E ENGENHARIA URBANA [SINGEURB 2021], p. 612–618, 2021.

LEANDRO, G. C. **Desenvolvimento De Software Acadêmico Para Engenharia Química, Utilizando Linguagem Fortran E Python - Cálculo De Reatores**. Trabalho de Conclusão de Curso-Universidade Tecnológica Federal do Paraná, 2028.

LEÃO, C. F. **Proposta De Modelo Para Controle Integrado Da Produção E Da Qualidade Utilizando Tecnologia De Informação**. Dissertação de Mestrado-Universidade Federal do Rio Grande do Sul, 2014.

LEVADA, A. L. M. **Programação Científica Com Python**. 22 jul. 2021.

LIMA, J. R. DE; MENEZES, C. S. D. **As Dificuldades Enfrentadas Pelos Estudantes Na Aprendizagem De Programação De Computadores: Uma Revisão Sistemática Da Literatura**. RNOTE, v. 22, n. 1, p. 130–140, 31 jul. 2024.

LOPES, G. R. *et al.* **Introdução À Análise Exploratória De Dados Com Python**. Escola Regional de Computação Aplicada à Saúde (ERCAS), 24 out. 2019.

LUCAS, B. T. **Proposta De Metodologia Para Cálculo Da Microdrenagem Urbana A Partir De Análise Discretizada Do Escoamento Superficial Da Água De Chuva**. Dissertação de Mestrado-Universidade Federal do Triângulo Mineiro, 2019.

MACEDO, E. P. **Análise Comparativa Dos Manuais E Orientações Técnicas Relativas Ao Dimensionamento De Sarjetas E Bocas De Lobo Das Capitais Brasileiras**. Associação Brasileira de Recursos Hídricos, out. 2020.

MALDANER, L. **Desenvolvimento De Software Acadêmico Para Engenharia Química Utilizando Linguagem Python: Separador De Mistura Binária**. Trabalho de Conclusão de Curso-Universidade Tecnológica Federal do Paraná, 2019.

MARCHIORI, N. C.; NOVAES ENGENHEIRO-AGRÔNOMO, MSC., A. L. T.; SILVA, B. C. DA. **Desenvolvimento De Um Sistema Piloto De Informações Geográficas Para A Gestão Da Piscicultura Continental Catarinense**. Agropecuária Catarinense, v. 34, n. 2, 24 ago. 2021.

MARTHA, L. F. **Introdução À Programação Orientada A Objetos**. [s.l.] PUC Rio, 2022.

MARTINS, E.; FORNARI, M. S. B. **Plano Diretor: Uma Análise Sob A Perspectiva Ambiental**. Revista Eletrônica Calafiori, v. 3, n. 2, nov. 2019.

MASIERO, A.; FERREIRA, L. A.; JÚNIOR, P. T. A. **Algoritmos De Clusterização E Python Científico Apoiando Modelagem De Usuário**. Brazilian Symposium on Human Factors in Computing Systems, v. 11, 1 nov. 2012.

MATTHES, E. **Python Crash Course**. 3. ed. [s.l.] 2023.

MATTOSINHO, G. D. O. *et al.* **Modelagem De Ondas Em Reservatórios: Inovações Para Otimização Dos Usos Múltiplos Do Reservatório De Ilha Solteira.** Peer W, [S. l.], v. 5, n. 13, p. 271–290, 2023. Disponível em: <https://peerw.org/index.php/journals/article/view/617>. Acesso em: 05 mai. 2024.

MCKINNEY, W. **Python Para Análise De Dados.** 1. ed. [s.l.] Novatec, 2018.

MELO, H. C. *et al.* **City Information Modeling (CIM) concepts applied to the management of the sewage network.** IOP Conference Series: Earth and Environmental Science, v. 588, p. 042026, 21 nov. 2020.

MENEZES, N. N. C. **Introdução À Programação Com Python - Algoritmos E Lógica De Programação Para Iniciantes.** Novatec, 2010.

MONTORO, A. F. **Python Ai Descubierto.** [s.l.] RC Libros, 2012.

MONTORO, A. F. **Python 3 Ai Descubierto.** 2. ed. [s.l.] RC Libros, 2013.

NASCIMENTO, J. A. S. DO. **Manejo De Águas Pluviais.** [s.l: s.n.]. Disponível em: [https://biblioteca.ibge.gov.br/visualizacao/livros/liv53096\\_cap10.pdf](https://biblioteca.ibge.gov.br/visualizacao/livros/liv53096_cap10.pdf). Acesso em: 20 set. 2021. Acesso em: 26 out. 2024.

NGUYEN, Q. **Hands-On Application Development With Pycharm.** [s.l.] 2019.

NUNES, A. H. D.; AMARAL, I. F. S. **Learning.Py Uma Apostila De Introdução À Programação Em Python.** [s.l.] Universidade Federal de Minas Gerais, 2020.

OBERLEITNER, A.; MASIERO, A. A. **Programação Orientada A Objetos.** [s.l.] Senac, 2021.

OLIVEIRA, V. H. M. DE; SERRA, S. M. B. **Controle De Obras Por RFID: Sistema De Monitoramento E Controle Para Equipamentos De Segurança No Canteiro De Obras.** Ambiente Construído, v. 17, n. 4, p. 61–77, dez. 2017.

PAIVA, F. *et al.* **Introdução A Python Com Aplicações De Sistemas Operacionais.** IFRN, p. 240, 2020.

PEDROSA, R. A. *et al.* **Plano Diretor De Drenagem Urbana No Contexto Do Planejamento Urbano.** Revista da Academia de Ciências do Piauí, v. 2, n. 2, 2021.

PETRUCCI, E.; OLIVEIRA, L. A. **Relações Entre Intensidade, Duração E Frequência Das Precipitações Máximas De 24 Horas E Equação De Chuvas Intensas Para A Cidade De Uberlândia-Mg.** Revista Brasileira de Climatologia, v. 25, dez. 2019.

PINTO, J. A.; JUNIOR, L. U. R. **Contribuição Para O Dimensionamento De Sistemas De Microdrenagem Urbana.** Revista Científic@ Universitas, v. 4, n. 2, 2017.

PIUMHI. **Lei Complementar N° 67, De 25 De Setembro De 2019.** , 25 set. 2019.

Disponível em:

[https://sapl.piumhi.mg.leg.br/media/sapl/public/normajuridica/2019/2300/lei\\_complementar\\_no\\_67\\_de\\_25\\_de\\_setembro\\_de\\_2019.pdf](https://sapl.piumhi.mg.leg.br/media/sapl/public/normajuridica/2019/2300/lei_complementar_no_67_de_25_de_setembro_de_2019.pdf). Acesso em: 1 nov. 2024.

- PYTHON SOFTWARE FOUNDATION. “**Why Is Python Installed On My Computer?**”. 2024a. Disponível em: <https://docs.python.org/3/faq/installed.html>. Acesso em: 23 out. 2024.
- PYTHON SOFTWARE FOUNDATION. **Pep 257 – Docstring Conventions**. 2024b. Disponível em: <https://peps.python.org/pep-0257/>. Acesso em: 17 ago. 2024.
- PYTHON SOFTWARE FOUNDATION. **9. Classes**. 2024c. Disponível em: <https://docs.python.org/pt-br/3/tutorial/classes.html>. Acesso em: 28 jul. 2024.
- RAMALHO, L. **Python Fluente: Programação Clara, Concisa E Eficaz**. 1. ed. [s.l.] Novatec, 2015.
- REDA, A. L. D. L. *et al.* **Urbanização, Mudanças No Clima E Reservatórios Para Amortecimento De Cheias: Comparando Soluções De Micro E Macro Escala**. Health and Environment World Congress, 2016.
- REIS, P. A. D. **Identificação De Áreas Vulneráveis As Enchentes E Inundações Em Áreas Urbanas Através De Modelos Topográficos E Hidráulicos**. Repositorio.ufu.br, 2015.
- RIGHETTO, ANTÔNIO MAROZZI. **Manejo De Águas Pluviais Urbanas**. Mcti.gov.br, 2009.
- ROCHA, C. I. O. **Plano Diretor De Drenagem Urbana Em Cidades Planejadas: Premissa De Zoneamento Baseado No Risco De Contaminação Da Água Subterrânea**. Revista Águas Subterrâneas, v. 28, n. 1, 11 set. 2014.
- SABOIA, J. P. J.; LOPARDO, N. **Software Para Cálculo De Regionalização De Parâmetros Hidrológicos Em Bacias Do Estado Do Paraná**. [s.l.]. 2015.
- SANTOS, C. F. R. DOS. **Tecnologias De Informação E Comunicação**. Unicentro.br, 2014.
- SÃO PAULO. **Diretrizes Básicas Para Projetos De Drenagem Urbana No Município De São Paulo**. [s.l.] Prefeitura do Município de São Paulo, 1999.
- SÃO PAULO. **Manual De Drenagem E Manejo De Águas Pluviais Aspectos Tecnológicos: Diretrizes Para Projetos**. 2012. Disponível em: [https://capital.sp.gov.br/web/licenciamento/w/desenvolvimento\\_urbano/biblioteca\\_digital/manual\\_de\\_drenagem/49018](https://capital.sp.gov.br/web/licenciamento/w/desenvolvimento_urbano/biblioteca_digital/manual_de_drenagem/49018). Acesso em: 12 set. 2023.
- SHAHROUR, I. *et al.* **Lessons From A Large Scale Demonstrator Of The Smart And Sustainable City**. Ecoproduction, p. 193–206, mar. 2017.
- SILVA, B. L. DE A. *et al.* **Conjunto De Drenagem Urbana Nas Cidades E Sua Importância Na Redução De Inundações E Enchentes - Hidrologia**. 2019a.
- SILVA, G. S. D. *et al.* **Diagnóstico Básico Do Sistema De Drenagem Em Piumhi – Mg Com Aplicação De Modelagem Gis**. Atena Editora, 2020.
- SILVA, G. S. D. *et al.* **Estudo De Drenagem Urbana E Infraestrutura Implementando Modelagem Gis Em Piumhi -Mg**. VIII Seminário de Iniciação Científica do IFMG Campus Ribeirão das Neves, 14 ago. 2019b.

SILVA, G. S.; MATTOSINHO, G. DE O. **Diagnóstico Básico Do Sistema De Drenagem Em Piumhi Com Aplicação De Modelagem GIS**. ForScience, v. 9, n. 2, p. e00871, 5 out. 2021.

SILVA, J. A. L. DA. **Tecnologias Alternativas Em Drenagem Urbana Para Os Pontos Críticos De Alagamento Do Município De Goiânia/GO**. 2018.

SILVA, W. **Python - Paradigmas De Linguagem De Programação**. 2017. Disponível em: <https://prezi.com/fb5f97irczyz/python-paradigmas-de-linguagem-de-programacao/>. Acesso em: 7 nov. 2024.

SOUBHIA, A. L. *et al.* **Python 101**. [s.l.] Universidade Federal de Santa Maria, 2019.

SOUZA, J. E. D. **O Uso Da Linguagem De Programação Python Na Resolução De Problemas Matemáticos Do Ensino Médio**. Trabalho de Conclusão de Curso-Universidade Federal de Campina Grande, 2023.

SOUZA, V. A. **Programação Em Python - Aprendendo Na Prática**. [s.l.] Cerne, 2011.

SRINATH, K. R. **Python -The Fastest Growing Programming Language**. International Research Journal of Engineering and Technology (IRJET), v. 4, 2017.

STACK OVERFLOW. **Problemas De Compatibilização De Charset Iso-8859-1 E Utf-8**. 2014. Disponível em: <https://pt.stackoverflow.com/questions/43193/problemas-de-compatibiliza%C3%A7%C3%A3o-de-charset-iso-8859-1-e-utf-8/43205#43205>. Acesso em: 1 dez. 2024.

STACK OVERFLOW. **Qual A Diferença Entre Uma Linguagem De Programação Estática E Dinâmica?** 2015. Disponível em: <https://pt.stackoverflow.com/questions/21508/qual-a-diferen%C3%A7a-entre-uma-linguagem-de-programa%C3%A7%C3%A3o-est%C3%A1tica-e-din%C3%A2mica>. Acesso em: 3 dez. 2024.

STACK OVERFLOW. **Por Que Usar Métodos Estáticos Em Python**. 2017. Disponível em: <https://pt.stackoverflow.com/questions/212312/por-que-usar-m%C3%A9todos-est%C3%A1ticos-em-python>. Acesso em: 21 nov. 2024.

STACK OVERFLOW. **Developer Survey 2022**. 2022. Disponível em: <https://survey.stackoverflow.co/2022/#most-popular-technologies-language-learn>. Acesso em: 7 out. 2024.

TAO, W. **Interdisciplinary Urban GIS For Smart Cities: Advancements And Opportunities**. Geo-spatial Information Science, v. 16, n. 1, p. 25–34, mar. 2013.

TEIXEIRA, B. DE H. M. **Aplicação Da Modelagem Da Informação Da Cidade Para Planejamento Urbano Via Integração Dos Softwares Cityengine E Urban**. Monografia-Universidade Estadual de Campinas, 2021.

TUCCI, C. E. M. **Hidrologia Ciência E Aplicação**. 1. ed. Porto Alegre: Universidade Federal do Rio Grande do Sul, 1993.

TUCCI, C. E. M. **Coefficiente De Escoamento E Vazão Máxima De Bacias Urbanas**. Revista Brasileira de Recursos Hídricos, v. 5, n. 1, mar. 2000.

TUCCI, C. E. M. **Hidrologia Ciência E Aplicação**. 2. ed. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2001.

TUCCI, C. E. M. **Gestão De Águas Pluviais Urbanas**. Global Water Partnership, 2005.

TUCCI, C. E. M. **Gestão Da Drenagem Urbana**. Cepal/IPEA, 2012.

TUCCI, C. E. M. **Regulamentação Da Drenagem Urbana No Brasil**. Revista de Gestão de Água da América Latina, v. 13, n. 1, jun. 2016.

TUCCI, C. E. M.; MENDES, C. A. **Avaliação Ambiental Integrada De Bacia Hidrográfica**. Ministério do Meio Ambiente, 2006.

VELOSO, R. D. S. **Tecnologias Da Informação E Da Comunicação**. [s.l.] Saraiva, 2017.

VERBEURGT, J. *et al.* **R And Python Benchmarking For Geographical Applications**. International Multidisciplinary Scientific GeoConference SGEM, 20 set. 2020.

W3SCHOOLS. **Python \_\_init\_\_() Function**. 2023. Disponível em: [https://www.w3schools.com/python/gloss\\_python\\_class\\_init.asp](https://www.w3schools.com/python/gloss_python_class_init.asp). Acesso em: 18 abr. 2024.

## APÊNDICE A – CADASTRAMENTO DOS DADOS E CÁLCULOS REALIZADOS PELA SOLUÇÃO COMPUTACIONAL.

Informações sobre o local

---

Cidade (insira o nome corretamente, incluindo os caracteres especiais): **piumhi**  
 Estação pluviométrica pertencente a cidade **piumhi** (insira o nome corretamente, incluindo os caracteres especiais): **piumhi**  
 Tempo de retorno [anos]: **2**  
 Tempo de concentração [minutos]: **10**  
 Coeficiente de escoamento: **0.75**  
 Intensidade pluviométrica [mm/h]: **113.28**

Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] **A109**  
 Digite o nome da rua: **Severo Veloso**  
 Informe o valor da área de contribuição do trecho **A109** [ha]: **0.172503**  
 Informe o comprimento do trecho **A109** [m]: **161.6**  
 Informe a cota montante do trecho **A109** [m]: **824**  
 Informe a cota jusante do trecho **A109** [m]: **822.5**  
 Há vazão futura para o trecho **A109** (s/n)?: **n**  
 Informe a quantidade de trecho(s) montante(s) do trecho **A109**: **0**  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] **A110**  
 Digite o nome da rua: **Severo Veloso**  
 Informe o valor da área de contribuição do trecho **A110** [ha]: **0.188543**  
 Informe o comprimento do trecho **A110** [m]: **143**  
 Informe a cota montante do trecho **A110** [m]: **822.5**  
 Informe a cota jusante do trecho **A110** [m]: **822**  
 Há vazão futura para o trecho **A110** (s/n)?: **n**  
 Informe a quantidade de trecho(s) montante(s) do trecho **A110**: **1**  
 Identifique o 1° trecho montante do trecho **A110**. [Pressione zero se não quiser identificá-lo]: **A109**  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] **A116**  
 Digite o nome da rua: **Amazonas**  
 Informe o valor da área de contribuição do trecho **A116** [ha]: **0.384954**  
 Informe o comprimento do trecho **A116** [m]: **84.5**  
 Informe a cota montante do trecho **A116** [m]: **822**  
 Informe a cota jusante do trecho **A116** [m]: **812**  
 Há vazão futura para o trecho **A116** (s/n)?: **n**  
 Informe a quantidade de trecho(s) montante(s) do trecho **A116**: **1**  
 Identifique o 1° trecho montante do trecho **A116**. [Pressione zero se não quiser identificá-lo]: **A110**  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] **A117**  
 Digite o nome da rua: **Lenir Soares**  
 Informe o valor da área de contribuição do trecho **A117** [ha]: **0.075921**  
 Informe o comprimento do trecho **A117** [m]: **38.2**  
 Informe a cota montante do trecho **A117** [m]: **822.5**  
 Informe a cota jusante do trecho **A117** [m]: **820**  
 Há vazão futura para o trecho **A117** (s/n)?: **n**  
 Informe a quantidade de trecho(s) montante(s) do trecho **A117**: **1**  
 Identifique o 1° trecho montante do trecho **A117**. [Pressione zero se não quiser identificá-lo]: **A109**

Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] A118  
 Digite o nome da rua: Lenir Soares  
 Informe o valor da área de contribuição do trecho A118 [ha]: 0.133264  
 Informe o comprimento do trecho A118 [m]: 38.2  
 Informe a cota montante do trecho A118 [m]: 822.5  
 Informe a cota jusante do trecho A118 [m]: 820  
 Há vazão futura para o trecho A118 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho A118: 1  
 Identifique o 1° trecho montante do trecho A118. [Pressione zero se não quiser identificá-lo]: A109  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] A119  
 Digite o nome da rua: Theotil Garcia Ribeiro  
 Informe o valor da área de contribuição do trecho A119 [ha]: 0.205516  
 Informe o comprimento do trecho A119 [m]: 50  
 Informe a cota montante do trecho A119 [m]: 824  
 Informe a cota jusante do trecho A119 [m]: 823.5  
 Há vazão futura para o trecho A119 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho A119: 0  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] A120  
 Digite o nome da rua: Jair Ferreira Leite  
 Informe o valor da área de contribuição do trecho A120 [ha]: 0.373627  
 Informe o comprimento do trecho A120 [m]: 162.5  
 Informe a cota montante do trecho A120 [m]: 823.5  
 Informe a cota jusante do trecho A120 [m]: 820  
 Há vazão futura para o trecho A120 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho A120: 1  
 Identifique o 1° trecho montante do trecho A120. [Pressione zero se não quiser identificá-lo]: A119  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] A121  
 Digite o nome da rua: Jair Ferreira Leite  
 Informe o valor da área de contribuição do trecho A121 [ha]: 0.074725  
 Informe o comprimento do trecho A121 [m]: 131.8  
 Informe a cota montante do trecho A121 [m]: 820  
 Informe a cota jusante do trecho A121 [m]: 812  
 Há vazão futura para o trecho A121 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho A121: 2  
 Identifique o 1° trecho montante do trecho A121. [Pressione zero se não quiser identificá-lo]: A117  
 Identifique o 2° trecho montante do trecho A121. [Pressione zero se não quiser identificá-lo]: A120

Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] E15  
 Digite o nome da rua: Severo Veloso  
 Informe o valor da área de contribuição do trecho E15 [ha]: 0.11868  
 Informe o comprimento do trecho E15 [m]: 161.6  
 Informe a cota montante do trecho E15 [m]: 824  
 Informe a cota jusante do trecho E15 [m]: 822.5  
 Há vazão futura para o trecho E15 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho E15: 0  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] E16  
 Digite o nome da rua: Theotil Garcia Ribeiro  
 Informe o valor da área de contribuição do trecho E16 [ha]: 0.11903  
 Informe o comprimento do trecho E16 [m]: 50  
 Informe a cota montante do trecho E16 [m]: 824  
 Informe a cota jusante do trecho E16 [m]: 823.5  
 Há vazão futura para o trecho E16 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho E16: 0  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] E17  
 Digite o nome da rua: Jair Ferreira Leite  
 Informe o valor da área de contribuição do trecho E17 [ha]: 0.44311  
 Informe o comprimento do trecho E17 [m]: 162.5  
 Informe a cota montante do trecho E17 [m]: 823.5  
 Informe a cota jusante do trecho E17 [m]: 820  
 Há vazão futura para o trecho E17 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho E17: 1  
 Identifique o 1° trecho montante do trecho E17. [Pressione zero se não quiser identificá-lo]: E16  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] E18  
 Digite o nome da rua: Jair Ferreira Leite  
 Informe o valor da área de contribuição do trecho E18 [ha]: 0.223881  
 Informe o comprimento do trecho E18 [m]: 131.8  
 Informe a cota montante do trecho E18 [m]: 820  
 Informe a cota jusante do trecho E18 [m]: 812  
 Há vazão futura para o trecho E18 (s/n)?: n  
 Informe a quantidade de trecho(s) montante(s) do trecho E18: 1  
 Identifique o 1° trecho montante do trecho E18. [Pressione zero se não quiser identificá-lo]: E17  
 Identifique o trecho onde você está: [Pressione zero se não houver mais trechos] 0  
 Cadastro encerrado!

```
====Relatório sintético final=====
Quantidade de trechos identificados: 12
Trecho A109 -> Quantidade de trecho(s) montante: 0
Quantidade de trecho(s) jusante: 3
Trecho A110 -> Quantidade de trecho(s) montante: 1
Quantidade de trecho(s) jusante: 1
Trecho A116 -> Quantidade de trecho(s) montante: 1
Quantidade de trecho(s) jusante: 0
Trecho A117 -> Quantidade de trecho(s) montante: 1
Quantidade de trecho(s) jusante: 1
Trecho A118 -> Quantidade de trecho(s) montante: 1
Quantidade de trecho(s) jusante: 0
Trecho A119 -> Quantidade de trecho(s) montante: 0
Quantidade de trecho(s) jusante: 1
Trecho A120 -> Quantidade de trecho(s) montante: 1
Quantidade de trecho(s) jusante: 1
Trecho A121 -> Quantidade de trecho(s) montante: 2
Quantidade de trecho(s) jusante: 0
Trecho E15 -> Quantidade de trecho(s) montante: 0
Quantidade de trecho(s) jusante: 0
Trecho E16 -> Quantidade de trecho(s) montante: 0
Quantidade de trecho(s) jusante: 1
Trecho E17 -> Quantidade de trecho(s) montante: 1
Quantidade de trecho(s) jusante: 1
Trecho E18 -> Quantidade de trecho(s) montante: 1
Quantidade de trecho(s) jusante: 0
Vazão para o trecho A109 [l/s]: 40.71
Vazão para o trecho A110 [l/s]: 44.50
Vazão para o trecho A116 [l/s]: 90.85
Vazão para o trecho A117 [l/s]: 17.92
Vazão para o trecho A118 [l/s]: 31.45
Vazão para o trecho A119 [l/s]: 48.50
Vazão para o trecho A120 [l/s]: 88.18
Vazão para o trecho A121 [l/s]: 17.64
Vazão para o trecho E15 [l/s]: 28.01
Vazão para o trecho E16 [l/s]: 28.09
Vazão para o trecho E17 [l/s]: 104.58
Vazão para o trecho E18 [l/s]: 52.84
```

```

Largura da sarjeta [m]: 0.3
Digite o coeficiente de rugosidade do material da sarjeta: 0.016
Altura do meio fio [m]: 0.15
Digite a altura disponível da sarjeta [m]: 0.13
Digite a declividade transversal da sarjeta (O valor deve estar entre 0.4 e 10%): 3
Os intervalos cadastrados são: 0.6 m/s <= velocidade água <= 3 m/s
Deseja cadastrar novos intervalos para o cálculo da velocidade da sarjeta? (s/n): n
Velocidade da água na sarjeta do trecho A109 [m/s]: 1.34
Velocidade da água na sarjeta do trecho A110 [m/s]: 0.95
Velocidade da água na sarjeta do trecho A116 [m/s]: 4.26
Velocidade da água na sarjeta do trecho A117 [m/s]: 2.27
Velocidade da água na sarjeta do trecho A118 [m/s]: 2.62
Velocidade da água na sarjeta do trecho A119 [m/s]: 1.44
Velocidade da água na sarjeta do trecho A120 [m/s]: 2.23
Velocidade da água na sarjeta do trecho A121 [m/s]: 2.20
Velocidade da água na sarjeta do trecho E15 [m/s]: 1.22
Velocidade da água na sarjeta do trecho E16 [m/s]: 1.26
Velocidade da água na sarjeta do trecho E17 [m/s]: 2.33
Velocidade da água na sarjeta do trecho E18 [m/s]: 2.90
A velocidade da água na sarjeta do trecho A109, está dentro do permitido
A velocidade da água na sarjeta do trecho A110, está dentro do permitido
A velocidade da água na sarjeta do trecho A116, está muito alta
A velocidade da água na sarjeta do trecho A117, está dentro do permitido
A velocidade da água na sarjeta do trecho A118, está dentro do permitido
A velocidade da água na sarjeta do trecho A119, está dentro do permitido
A velocidade da água na sarjeta do trecho A120, está dentro do permitido
A velocidade da água na sarjeta do trecho A121, está dentro do permitido
A velocidade da água na sarjeta do trecho E15, está dentro do permitido
A velocidade da água na sarjeta do trecho E16, está dentro do permitido
A velocidade da água na sarjeta do trecho E17, está dentro do permitido
A velocidade da água na sarjeta do trecho E18, está dentro do permitido
Deseja realizar alguma alteração? (s/n) s
Sugiro algumas alterações: Modificar Largura da Sarjeta, Coeficiente de rugosidade ou utilizar uma boca delobo intermediária
Sugiro algumas alterações: Modificar Largura da Sarjeta, Coeficiente de rugosidade ou utilizar uma boca delobo intermediária
Largura da sarjeta [m]: 0.3
Digite o coeficiente de rugosidade do material da sarjeta: 0.016
Os intervalos cadastrados são: 0.6 m/s <= velocidade água <= 3 m/s
Deseja cadastrar novos intervalos para o cálculo da velocidade da sarjeta? (s/n): s
Informe o limite superior: 4.3
Informe o limite inferior: 0.6
Velocidade da água na sarjeta do trecho A109 [m/s]: 1.34
Velocidade da água na sarjeta do trecho A110 [m/s]: 0.95
Velocidade da água na sarjeta do trecho A116 [m/s]: 4.26
Velocidade da água na sarjeta do trecho A117 [m/s]: 2.27
Velocidade da água na sarjeta do trecho A118 [m/s]: 2.62
Velocidade da água na sarjeta do trecho A119 [m/s]: 1.44
Velocidade da água na sarjeta do trecho A120 [m/s]: 2.23
Velocidade da água na sarjeta do trecho A121 [m/s]: 2.20
Velocidade da água na sarjeta do trecho E15 [m/s]: 1.22
Velocidade da água na sarjeta do trecho E16 [m/s]: 1.26
Velocidade da água na sarjeta do trecho E17 [m/s]: 2.33
Velocidade da água na sarjeta do trecho E18 [m/s]: 2.90
A velocidade da água na sarjeta do trecho A109, está dentro do permitido
A velocidade da água na sarjeta do trecho A110, está dentro do permitido
A velocidade da água na sarjeta do trecho A116, está dentro do permitido
A velocidade da água na sarjeta do trecho A117, está dentro do permitido
A velocidade da água na sarjeta do trecho A118, está dentro do permitido
A velocidade da água na sarjeta do trecho A119, está dentro do permitido
A velocidade da água na sarjeta do trecho A120, está dentro do permitido
A velocidade da água na sarjeta do trecho A121, está dentro do permitido
A velocidade da água na sarjeta do trecho E15, está dentro do permitido
A velocidade da água na sarjeta do trecho E16, está dentro do permitido
A velocidade da água na sarjeta do trecho E17, está dentro do permitido
A velocidade da água na sarjeta do trecho E18, está dentro do permitido

Sarjeta não eficaz para o trecho A109
Sarjeta não eficaz para o trecho A110
Sarjeta não eficaz para o trecho A116
Sarjeta eficaz para o trecho A117
Sarjeta eficaz para o trecho A118
Sarjeta não eficaz para o trecho A119
Sarjeta não eficaz para o trecho A120
Sarjeta eficaz para o trecho A121
Sarjeta não eficaz para o trecho E15
Sarjeta não eficaz para o trecho E16
Sarjeta não eficaz para o trecho E17
Sarjeta eficaz para o trecho E18
Deseja realizar alguma alteração (s/n)? s

```

Sugiro algumas alterações: Modificar Largura da Sarjeta, Coeficiente de rugosidade ou utilizar uma boca delobo intermediária  
 Largura da sarjeta [m]: 1  
 Digite o coeficiente de rugosidade do material da sarjeta: 0.016  
 Os intervalos cadastrados são: 0.6 m/s <= velocidade água <= 3 m/s  
 Deseja cadastrar novos intervalos para o cálculo da velocidade da sarjeta? (s/n): s  
 Informe o limite superior: 4.3  
 Informe o limite inferior: 0.6  
 Velocidade da água na sarjeta do trecho A109 [m/s]: 0.99  
 Velocidade da água na sarjeta do trecho A110 [m/s]: 0.70  
 Velocidade da água na sarjeta do trecho A116 [m/s]: 3.15  
 Velocidade da água na sarjeta do trecho A117 [m/s]: 1.68  
 Velocidade da água na sarjeta do trecho A118 [m/s]: 1.94  
 Velocidade da água na sarjeta do trecho A119 [m/s]: 1.07  
 Velocidade da água na sarjeta do trecho A120 [m/s]: 1.65  
 Velocidade da água na sarjeta do trecho A121 [m/s]: 1.63  
 Velocidade da água na sarjeta do trecho E15 [m/s]: 0.90  
 Velocidade da água na sarjeta do trecho E16 [m/s]: 0.93  
 Velocidade da água na sarjeta do trecho E17 [m/s]: 1.72  
 Velocidade da água na sarjeta do trecho E18 [m/s]: 2.14  
 A velocidade da água na sarjeta do trecho A109, está dentro do permitido  
 A velocidade da água na sarjeta do trecho A110, está dentro do permitido  
 A velocidade da água na sarjeta do trecho A116, está dentro do permitido  
 A velocidade da água na sarjeta do trecho A117, está dentro do permitido  
 A velocidade da água na sarjeta do trecho A118, está dentro do permitido  
 A velocidade da água na sarjeta do trecho A119, está dentro do permitido  
 A velocidade da água na sarjeta do trecho A120, está dentro do permitido  
 A velocidade da água na sarjeta do trecho A121, está dentro do permitido  
 A velocidade da água na sarjeta do trecho E15, está dentro do permitido  
 A velocidade da água na sarjeta do trecho E16, está dentro do permitido  
 A velocidade da água na sarjeta do trecho E17, está dentro do permitido  
 A velocidade da água na sarjeta do trecho E18, está dentro do permitido

Sarjeta eficaz para o trecho A109  
 Sarjeta eficaz para o trecho A110  
 Sarjeta eficaz para o trecho A116  
 Sarjeta eficaz para o trecho A117  
 Sarjeta eficaz para o trecho A118  
 Sarjeta eficaz para o trecho A119  
 Sarjeta eficaz para o trecho A120  
 Sarjeta eficaz para o trecho A121  
 Sarjeta eficaz para o trecho E15  
 Sarjeta eficaz para o trecho E16  
 Sarjeta eficaz para o trecho E17  
 Sarjeta eficaz para o trecho E18

Informações sobre Boca de Lobo

Opções de boca de lobo:

Tipo de boca de lobo de acordo com a sua localização	Código
Boca de lobo tipo Guia, localizada em ponto baixo	1
Boca de lobo tipo Guia, localizada em ponto intermediário	2
Boca de lobo tipo Grelha, localizada em ponto baixo	3
Boca de lobo tipo Grelha com barras na longitudinal, localizada em ponto intermediário	4
Boca de lobo tipo Grelha com barras na transversal, localizada em ponto intermediário	5
Boca de lobo tipo Combinada, localizada em ponto baixo	6
Boca de lobo tipo Combinada com barras na longitudinal, localizada em ponto intermediário	7
Boca de lobo tipo Combinada com barras na transversal, localizada em ponto intermediário	8

Você já sabe qual boca de lobo deseja utilizar (s/n)? n

Largura da abertura na guia [m]: 0.8

Altura da abertura na guia [m]: 0.1

Dimensão interna da grelha do lado paralelo ao meio fio [m]:

(Se um dos lados for adjacente ao meio fio, insira apenas a metade do valor): 0.8

Dimensão interna do lado perpendicular ao meio fio da boca de lobo tipo grelha [m]: 0.3

Espessura das barras longitudinais [m]: 0.05

Comprimento das barras longitudinais [m]: 0.35

Quantidade de barras longitudinais: 8

Espessura das barras transversais [m]: 0.05

Comprimento das barras transversais [m]: 0.3

Quantidade das barras transversais: 2

**APÊNDICE B – CADASTRAMENTO DOS DADOS E CÁLCULOS REALIZADOS  
NA PLANILHA.**

Intensidade média de precipitação sobre toda área de contribuição		Coeficiente de escoamento superficial		Informações da via	
DADOS PARA PIUMHI - MG		C	0,75	Coefficiente de rugosidade	0,016
T (anos)	2			Declividade transversal (%)	3
t (min)	10			Fator de redução sarjeta	0,8
K	1958,651				
a	0,22				
b	16,239				
c	0,919				
<b>i (mm/h)</b>	<b>113,28</b>				

Trecho	Área de Contribuição (ha)	Q (trecho) (l/s)	Q (futura) (l/s)	Q (total) (l/s)
A109	0,17	40,71		40,71
A110	0,19	44,50		44,50
A116	0,38	90,85		90,85
A117	0,08	17,92		17,92
A118	0,13	31,45		31,45
A119	0,21	48,50		48,50
A120	0,37	88,18		88,18
A121	0,07	17,64		17,64
E15	0,12	28,01		28,01
E16	0,12	28,09		28,09
E17	0,44	104,58		104,58
E18	0,22	52,84		52,84

Trecho de referência	Cota		Comprimento (m)	Declividade (m/m)
	Montante	Jusante		
A109	824	822,5	161,6	0,009
A110	822,5	822	143	0,003
A116	822	812	84,5	0,118
A117	822,5	820	38,2	0,065
A118	822,5	820	38,2	0,065
A119	824	823,5	50	0,010
A120	823,5	820	162,5	0,022
A121	820	812	131,8	0,061
E15	824	822,5	161,6	0,009
E16	824	823,5	50	0,010
E17	823,5	820	162,5	0,022
E18	820	812	131,8	0,061

Trecho	Altura do meio fio (m)	Altura disponível na sarjeta (m)	Largura da Sarjeta (m)	Verificação da capacidade da Sarjeta	Velocidade da água na sarjeta (m/s)	Verificação da Velocidade
A109	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	1,34	OK
A110	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	0,95	OK
A116	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	4,26	Muito alta
A117	0,15	0,13	0,30	Sarjeta eficaz	2,27	OK
A118	0,15	0,13	0,30	Sarjeta eficaz	2,62	OK
A119	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	1,44	OK
A120	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	2,23	OK
A121	0,15	0,13	0,30	Sarjeta eficaz	2,20	OK
E15	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	1,22	OK
E16	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	1,26	OK
E17	0,15	0,13	0,30	Aumentar a sarjeta e/ou colocar BL's	2,33	OK
E18	0,15	0,13	0,30	Sarjeta eficaz	2,90	OK

Trecho	Altura do meio fio (m)	Altura disponível na sarjeta (m)	Largura da Sarjeta (m)	Verificação da capacidade da Sarjeta	Velocidade da água na sarjeta (m/s)	Verificação da Velocidade
A109	0,15	0,13	1,00	Sarjeta eficaz	0,99	OK
A110	0,15	0,13	1,00	Sarjeta eficaz	0,70	OK
A116	0,15	0,13	1,00	Sarjeta eficaz	3,15	OK
A117	0,15	0,13	1,00	Sarjeta eficaz	1,68	OK
A118	0,15	0,13	1,00	Sarjeta eficaz	1,94	OK
A119	0,15	0,13	1,00	Sarjeta eficaz	1,07	OK
A120	0,15	0,13	1,00	Sarjeta eficaz	1,65	OK
A121	0,15	0,13	1,00	Sarjeta eficaz	1,63	OK
E15	0,15	0,13	1,00	Sarjeta eficaz	0,90	OK
E16	0,15	0,13	1,00	Sarjeta eficaz	0,93	OK
E17	0,15	0,13	1,00	Sarjeta eficaz	1,72	OK
E18	0,15	0,13	1,00	Sarjeta eficaz	2,14	OK

<b>Informações das bocas de lobo</b>	
<b>Boca de lobo tipo Guia</b>	
Largura da boca de lobo (m)	0,8
Altura da boca de lobo (m)	0,1
<b>Boca de lobo tipo Grelha</b>	
Lado perpendicular ao meio fio (m)	0,3
Lado paralelo ao meio fio (m)	0,8
<b>Perímetro (m)</b>	<b>2,2</b>
Comprimento barras transversais (m)	0,3
Espessura barras transversais (m)	0,05
Quantidade barras transversais	2
Comprimento barras longitudinais (m)	0,35
Espessura barras longitudinais (m)	0,05
Quantidade barras longitudinais	8
<b>Área livre (m<sup>2</sup>)</b>	<b>0,07</b>