



## **XVII Semana Nacional de Ciência e Tecnologia**

**Biomias do Brasil: Diversidade, Saberes e Tecnologias Sociais**

02 a 06 de Dezembro de 2024

ISSN 2594-8237

# **UM SISTEMA DE GERENCIAMENTO DE DESASTRES AMBIENTAIS UTILIZANDO ÁRVORE B: UMA ANÁLISE COMPARATIVA DE DESEMPENHO FRENTE A OUTRAS ESTRUTURAS DE DADOS**

Gregory Gabriel Ozaki Coelho e Rallyson Dos Santos Ferreira

Instituto de Ciências Exatas e Tecnologia – Universidade Federal do Amazonas  
Rua Nossa Senhora do Rosário, 3683 – Tiradentes – Itacoatiara/AM

*gregory.coelho@ufam.edu.br, rallysonferreira@gmail.com*

**Resumo:** Este artigo aborda a crescente frequência e intensidade dos desastres ambientais, destacando a necessidade de sistemas eficazes para seu gerenciamento. O objetivo é analisar o desempenho da Árvore B em operações de busca, inserção e remoção, comparando-a com outras estruturas de dados. Foram utilizados scripts em Python para medir o tempo de execução das operações, permitindo uma avaliação da eficiência da Árvore B em diferentes volumes de dados. Os resultados demonstraram que a Árvore B apresenta um desempenho superior em comparação com árvores AVL e listas encadeadas em cenários de grande volume de dados, evidenciando sua eficácia no gerenciamento de informações.

**Palavras-Chave:** Árvore B. Desempenho. Estrutura de Dados. Meio-Ambiente.

## **1. INTRODUÇÃO**

Os desastres ambientais, como inundações, incêndios florestais e secas prolongadas, têm se tornado cada vez mais frequentes e intensos, impulsionados por fatores como as mudanças climáticas e as ações humanas (Chame & Sianto, 2021). As consequências desses eventos são devastadoras, impactando ecossistemas, infraestruturas e a vida das populações. Nessa perspectiva, muitas cidades pequenas enfrentam sérias dificuldades na gestão de desastres ambientais, muitas vezes devido à falta de sistemas adequados para monitorar, registrar e responder a essas ocorrências. A ausência de uma infraestrutura tecnológica eficiente agrava a vulnerabilidade dessas localidades, uma vez que a falta de dados organizados dificulta tanto a prevenção quanto a recuperação de desastres (Tsai & Chang, 2023).

A falta de sistemas integrados para o gerenciamento de desastres ambientais em cidades menores pode resultar na perda de informações valiosas, como padrões históricos de desastres, o que impede a análise de tendências e a implementação de estratégias eficazes de mitigação (Tsai & Chang, 2023). Além disso, a dificuldade na busca e no registro de novos eventos, bem como a inexistência de relatórios públicos, limita a capacidade de decisão e o engajamento das comunidades locais. Com o agravamento das questões ambientais, especialmente devido ao aquecimento global,



# **XVII Semana Nacional de Ciência e Tecnologia**

**Biomass do Brasil: Diversidade, Saberes e Tecnologias Sociais**

02 a 06 de Dezembro de 2024

ISSN 2594-8237

torna-se ainda mais urgente desenvolver soluções tecnológicas que possam otimizar a gestão desses desastres e promover a resiliência nas regiões mais vulneráveis.

Este trabalho propõe a implementação de Árvores B como uma estrutura de dados eficiente para o desenvolvimento de um sistema de gerenciamento de desastres ambientais. O sistema tem como objetivo abordar os desafios de armazenamento, organização e busca de dados críticos relacionados a desastres, visando aprimorar tanto a resposta quanto a prevenção. Além de facilitar o acesso a informações históricas e permitir um gerenciamento mais ágil e preciso das ocorrências ambientais, o trabalho também busca demonstrar que o uso de Árvores B para gerenciar desastres ambientais pode ser mais eficiente do que outras estruturas de dados. Com isso, pretende-se promover uma maior conscientização pública e otimizar os recursos disponíveis para a resposta a emergências.

O restante do artigo está organizado da seguinte maneira. A Seção 2 apresenta alguns conceitos básicos e discute trabalhos relacionados. A Seção 3 apresenta a metodologia utilizada enquanto a Seção 4 mostra os resultados e as discussões. A Seção 5 apresenta as considerações finais e os trabalhos futuros.

## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Desastres Ambientais**

Para Cavalcante e Temoteo (2018) Os desastres ambientais são eventos de grande impacto, resultantes de fatores naturais e ações humanas, que afetam ecossistemas e comunidades. Dividem-se em desastres naturais e antropogênicos, sendo agravados pela urbanização desordenada e mudanças climáticas, especialmente no Brasil, onde áreas frágeis sofrem com décadas de crescimento sem planejamento.

#### **2.1.1 Desastres Naturais**

Desastres naturais, como terremotos, inundações e secas, resultam de processos ambientais intensificados por mudanças climáticas e ações humanas. No Brasil, a rápida urbanização aumenta a vulnerabilidade, especialmente em áreas periféricas com pouca infraestrutura (Chame & Sianto, 2021; Cavalcante & Temoteo, 2018).

#### **2.1.2 Desastres Antropológicos**

De acordo com Abrantes e Neto (2021) os desastres ambientais de origem antropogênica resultam de interações complexas entre atividades humanas e o meio ambiente, ocasionando impactos severos nos ecossistemas, nas comunidades e na biodiversidade. Esses eventos muitas vezes têm como causa práticas inconvenientes de gestão ambiental, como o descarte incorreto de resíduos, que é uma das principais origens de desastres urbanos.



# **XVII Semana Nacional de Ciência e Tecnologia**

**Biomias do Brasil: Diversidade, Saberes e Tecnologias Sociais**

02 a 06 de Dezembro de 2024

ISSN 2594-8237

## **2.2 Desafios na Gestão de Desastres e o Papel da Tecnologia em Regiões Pequenas**

Muitas cidades pequenas enfrentam grandes desafios na gestão de desastres devido à ausência de tecnologias como o sensoriamento remoto e os Sistemas de Informação Geográfica (SIG). Essas ferramentas são fundamentais para identificar áreas vulneráveis, acompanhar mudanças no ambiente e desenvolver estratégias eficazes de mitigação (Dutra et al., 2021). Sem elas, a capacidade de prever e reagir a desastres torna-se significativamente prejudicada, aumentando a exposição das populações locais. Para Gonzalez (2023) as tecnologias digitais têm se mostrado cruciais para enfrentar desafios ambientais e sociais, como a gestão de dados e a identificação de áreas de risco, além de favorecer a conscientização e a comunicação eficiente em momentos de crise.

## **2.3 Estruturas de Dados e Linguagem de Programação**

Estruturas de dados organizam, armazenam e manipulam informações em programas, sendo essenciais para a eficiência de algoritmos. A escolha adequada de uma estrutura pode impactar significativamente o desempenho de um sistema (Souza & Coelho, 2015).

### **2.3.1 Árvores**

Dentro das estruturas de dados, as árvores são estruturas hierárquicas compostas por nós, amplamente usadas em bancos de dados, sistemas de arquivos e algoritmos de busca. Cada nó pode ter zero ou mais filhos, sendo organizados em termos de grau, nível e altura, o que influencia a eficiência das operações (Ziviani, 2007).

Segundo Ziviani (2007), o grau de uma árvore refere-se ao número máximo de filhos que um nó pode ter (como em árvores binárias, onde cada nó possui no máximo dois filhos), enquanto o nível indica a distância de um nó até a raiz, e a altura representa o nível máximo da árvore. Os nós folhas são aqueles sem filhos, localizados na extremidade da estrutura, enquanto a raiz é o nó superior, sem pai. Os galhos conectam os nós, formando uma estrutura, onde cada nó pai pode ter um ou mais filhos, mas cada nó filho está ligado a apenas um pai.

### **2.3.2 Árvore B**

Entre as diversas estruturas de árvores, a árvore B é amplamente usada em bancos de dados e sistemas de arquivos devido à eficiência em operações como busca e inserção, realizadas em tempo logarítmico. Ela é balanceada, garantindo consistência no tempo de acesso aos dados, e organizada por um parâmetro  $m$  (ordem ou grau), que define, no mínimo  $m$  chaves por nó e no máximo  $2m-1$ , com exceção da raiz que pode ter no mínimo  $m$ . Enquanto o número filhos por nó, pode ser no mínimo  $m+1$  e no máximo  $2m+1$ . Essa estrutura é ideal para aplicações com grande volume de dados, devido à sua capacidade de otimizar o uso do espaço e tempo de acesso (Ziviani, 2007).



## **XVII Semana Nacional de Ciência e Tecnologia**

**Biomias do Brasil: Diversidade, Saberes e Tecnologias Sociais**

02 a 06 de Dezembro de 2024

ISSN 2594-8237

### **2.3.3 Implementação de Árvore B em Python**

Python é uma linguagem de programação de alto nível, reconhecida por sua simplicidade e versatilidade. Criada por Guido van Rossum em 1991, é interpretada e suporta paradigmas como programação imperativa, funcional e orientada a objetos. Sua sintaxe clara permite que os desenvolvedores foquem na lógica do problema, facilitando seu uso em diversas áreas (Biscaro, 2023).

Lima (2014) discute que a implementação de Árvores B em Python é estratégica para manipular grandes volumes de dados com eficiência, devido ao tempo logarítmico em operações de busca, inserção e exclusão, essencial em bancos de dados e sistemas de arquivos.

## **3. MATERIAL E MÉTODO**

### **3.1 Tipo de Pesquisa**

Esta pesquisa é de natureza aplicada, pois busca gerar conhecimento prático para o desenvolvimento e implementação de soluções voltadas ao gerenciamento de desastres ambientais. A abordagem adotada é predominantemente quantitativa, com o objetivo de estruturar e analisar dados para a construção de um modelo eficiente. Além disso, caracteriza-se como um estudo exploratório, uma vez que investiga novos métodos e ferramentas, como a utilização de árvores B, para aprimorar processos de tomada de decisão e mitigação de impactos em cenários de desastres ambientais.

### **3.2. Ferramentas Utilizadas**

Para o desenvolvimento e implementação do modelo de gerenciamento de desastres ambientais utilizando árvores B, foram empregadas as seguintes ferramentas e tecnologias:

- **Linguagens de Programação:** Python, devido à sua versatilidade e amplo suporte para bibliotecas voltadas ao processamento de dados e algoritmos estruturais.
- **Softwares e Plataformas:** Foram utilizados softwares para o desenvolvimento de código estruturado e manutenção de scripts, como o VS Code.
- **Ambiente de Desenvolvimento:** O trabalho foi realizado em um ambiente operacional Windows, utilizando tanto um desktop quanto um notebook para a execução das atividades.

### **3.3 Estruturação da Árvore B**

O objetivo geral do código é implementar a inserção, busca, exibição e remoção de registros de desastres utilizando uma Árvore B, que organiza os dados de forma eficiente. Dentro do algoritmo, há uma função `gerar_id` responsável por criar identificadores únicos para os desastres, considerando o tipo (A para antropológicos e N



# XVII Semana Nacional de Ciência e Tecnologia

Biomass do Brasil: Diversidade, Saberes e Tecnologias Sociais

02 a 06 de Dezembro de 2024

ISSN 2594-8237

para naturais) e a data fornecida, além de incrementar contadores globais conforme o tipo. A seguir há uma representação de um ID gerado:

DAM202401N: Desastres ambiental natural do ano 2024, o primeiro registrado.

DAM202401A: Desastre ambiental antropológico do ano 2024.

A função `coletar_dados_usuario` na figura 1 abaixo solicita ao usuário a data do registro e o tipo do desastre, chamando a função `gerar_id` para gerar o identificador correspondente. A figura a seguir, apresenta um pseudocódigo das funções citadas.

Figura 1 — Coletar e gerar ID

```
INICIAR contadores: antropológicos e naturais em 0

FUNÇÃO gerar_id(data, tipo):
    SE tipo == "A":
        incrementar contador antropológico
        numero_desastre = contador antropológico
    SENÃO:
        incrementar contador natural
        numero_desastre = contador natural
    ano = extrair ano de data
    RETORNAR "DAM" + ano + numero_desastre.formatado + tipo

FUNÇÃO coletar_dados_usuario():
    data = solicitar entrada
    tipo = "A" SE entrada == "1" SENÃO "N"
    RETORNAR gerar_id(data, tipo)
```

Fonte: Figura do autor (2024).

Dentro do algoritmo, para criar a estrutura de Árvore B, definimos a classe `No` representando os nós da Árvore B, armazenando chaves (dados) e filhos. Logo após, a classe `ArvoreB` para implementar a lógica principal da estrutura, incluindo operações de inserção, busca, divisão de nós e remoção de elementos. A figura 2 a seguir, apresenta o pseudocódigo da função de busca, enquanto a figura 3 apresenta as funções de inserção, remoção e suas funções auxiliares.

Figura 2 — Função de busca

```
FUNÇÃO busca(id, no):
    SE no é NULL:
        no = raiz
    PARA cada chave EM no.chaves:
        SE id == chave:
            RETORNAR True
        SE id < chave E no NÃO é folha:
            CHAMAR busca(id, no.filhos[i])
    SE não é folha:
        CHAMAR busca(id, no.último_filho)
    RETORNAR False
```

Fonte: Figura do autor (2024).





# XVII Semana Nacional de Ciência e Tecnologia

Biomass do Brasil: Diversidade, Saberes e Tecnologias Sociais

02 a 06 de Dezembro de 2024

ISSN 2594-8237

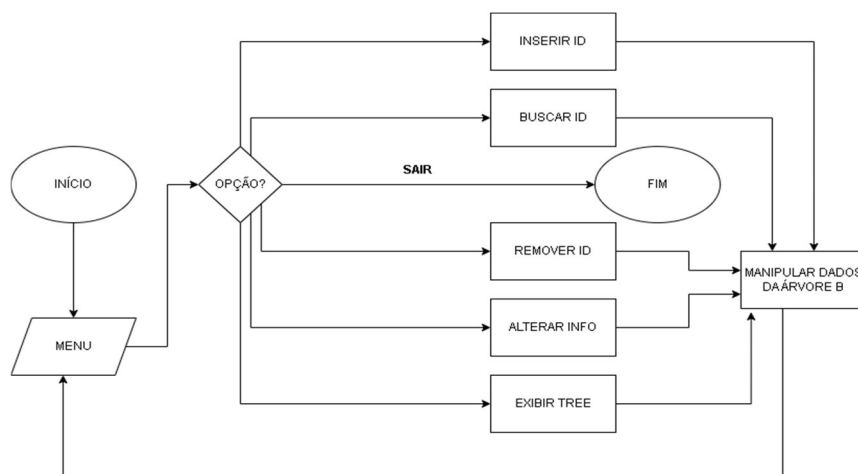
**Figura 3 — Funções de inserção, remoção e auxiliares**

```
FUNÇÃO insercao(id):  
  SE raiz cheia: // Se a raiz já estiver cheia, ela precisa ser dividida  
    criar nova raiz // Criar uma nova raiz para acomodar o nó dividido  
    dividir raiz atual // Divide a raiz atual em dois nós e promove uma chave para a nova raiz  
  inserir_nao_cheio(raiz, id) // Insere o id em um nó apropriado, garantindo que as propriedades da árvore B  
  
FUNÇÃO dividir(no, indice): // Divide o nó cheio em dois e promove uma chave para o nó pai  
  mover chave meio para o pai // A chave mediana é movida para o nó pai, separando as chaves entre dois novos nós  
  dividir filhos entre os dois nós // Os filhos do nó original são redistribuídos entre os dois novos nós resultantes  
  
FUNÇÃO inserir_nao_cheio(no, id): // Insere o id em um nó que não está cheio  
  SE no é folha: // Se o nó atual é uma folha, insere o id diretamente  
    inserir id ordenadamente // Coloca o id no local correto, mantendo as chaves ordenadas  
  SENÃO: // Se o nó não é uma folha, desce até o nó filho correto  
    localizar filho para inserir // Determina qual filho deve receber o id  
    SE filho cheio: // Se o nó filho também está cheio, divide-o antes de continuar  
      dividir filho  
    CHAMAR inserir_nao_cheio // Recursivamente insere o id no filho correto  
  
FUNÇÃO remover(id):  
  CHAMAR remover_recursivo(raiz, id) // Inicia a remoção recursiva começando pela raiz  
  
FUNÇÃO remover_recursivo(no, id): // Realiza a remoção de um id em um nó específico  
  SE no é folha: // Se o nó é uma folha, a remoção é direta  
    remover id diretamente // Apenas remove o id, já que ele não possui filhos  
  SENÃO: // Se não é uma folha, a remoção é mais complexa  
    localizar id // Encontra a posição do id no nó  
    SE encontrado: // Se o id está no nó atual  
      substituir por máximo/mínimo ou mesclar nós // Mantém a ordem  
    SENÃO: // Se o id não está no nó atual  
      balancear filhos e continuar remoção // Garante que os nós filhos sejam balanceados antes de prosseguir
```

Fonte: Figura do autor (2024).

As funções de busca, remoção e inserção na Árvore B são essenciais para seu funcionamento. A busca percorre as chaves do nó atual e, se necessário, desce aos filhos para encontrar o valor ou determinar sua ausência. A remoção lida com casos como a remoção direta em folhas, substituição por valores adjacentes em nós internos, ou mesclagem de nós, utilizando funções como `get_maximo`, `get_minimo` e `mesclar`. A inserção verifica se o nó está cheio e, se necessário, divide-o para manter as propriedades da árvore, utilizando `inserir_nao_cheio` para localizar a posição correta e inserir de forma ordenada. A Figura 4 a seguir ilustra o fluxograma do sistema.

**Figura 4 – Fluxograma do sistema**



Fonte: Fluxograma do autor (2024).



# XVII Semana Nacional de Ciência e Tecnologia

Biomias do Brasil: Diversidade, Saberes e Tecnologias Sociais

02 a 06 de Dezembro de 2024

ISSN 2594-8237

## 3.4 Métodos de Análise

Os resultados foram analisados com base no desempenho da árvore B em operações como busca, inserção, remoção e de dados, utilizando scripts em Python e a biblioteca *time* para medir os tempos de execução. Esses valores foram comparados com os de uma lista encadeada e uma árvore AVL criada no mesmo ambiente, permitindo avaliar a eficiência da árvore B em termos de escalabilidade e manutenção do balanceamento com diferentes volumes de dados. Simulações com conjuntos de variáveis de dados também foram realizadas para verificar sua eficácia no gerenciamento de informações em cenários de desastres ambientais.

## 4. RESULTADOS E DISCUSSÃO

Nesta seção, apresentamos os resultados obtidos através da execução das operações de inserção, remoção e busca. Os resultados são organizados em tabelas, gráficos e imagens, que ilustram o desempenho do algoritmo em termos de tempo de execução e a estrutura da árvore B ao longo das inserções de dados.

### 4.1 Apresentação dos Resultados

A Tabela 1 abaixo compara o tempo de execução da árvore B com outras estruturas, como a árvore AVL e a lista encadeada. Foram usadas as mesmas entradas para as operações de busca, inserção e remoção. As quantidades de entradas correspondem ao número de IDs inseridos (10, 100 e 1000), enquanto as buscas e remoções ocorreram em 6, 50 e 500 casos, respectivamente. Além disso, calculou-se o tempo total das três funções e sua média. O gráfico a seguir compara o desempenho de execução da árvore B com as outras estruturas.

Tabela 1 — Tempo de execução

Estruturas de Dados	Quant. de Entradas	Busca	Inserção	Remoção	Total	Média
Árvore B	10	0.000025	0.000022	0.000041	0.000088	0.000029
	100	0.000335	0.000261	0.000346	0.000942	0.000314
	1000	0.008891	0.004039	0.003223	0.016153	0.005384
Árvore AVL	10	0.000008	0.000058	0.000036	0.000101	0.000034
	100	0.000081	0.000553	0.000391	0.001025	0.000342
	1000	0.001389	0.010116	0.006978	0.018483	0.006161
Lista Encadeada	10	0.000005	0.000009	0.000007	0.000021	0.000007
	100	0.000329	0.000588	0.000256	0.001173	0.000391
	1000	0.045220	0.035778	0.020714	0.101712	0.033904

Fonte: Tabela do autor (2024).

### 4.2 Discussão dos Resultado

Os resultados obtidos demonstram que, apesar da constante manutenção da árvore B, ela se sobressai quando o assunto são grandes quantidades de informações.



# XVII Semana Nacional de Ciência e Tecnologia

Biomass do Brasil: Diversidade, Saberes e Tecnologias Sociais

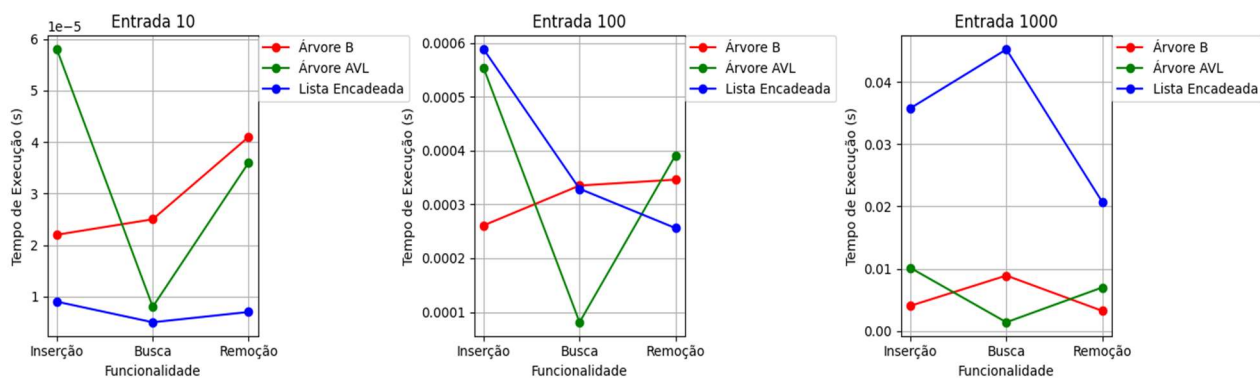
02 a 06 de Dezembro de 2024

ISSN 2594-8237

Como evidenciado na tabela 1, a árvore B apresenta um desempenho superior em cenários que envolvem um grande volume de dados, mesmo que sua performance em operações de busca, inserção e remoção seja inferior à de uma lista encadeada ou de uma árvore AVL em situações com poucas entradas.

Por exemplo, enquanto uma lista encadeada pode realizar operações de busca, inserção e remoção de forma mais rápida com um número reduzido de entradas, essa vantagem se torna irrelevante quando se trata de grandes quantidades de dados. A árvore B, por outro lado, mantém uma estrutura balanceada que permite um acesso mais eficiente a informações críticas, o que é essencial para o gerenciamento de desastres ambientais em cidades pequenas e áreas com alta vulnerabilidade. Os gráficos na figura 5 apresentam isso.

Figura 5 — Gráficos do tempo de execução

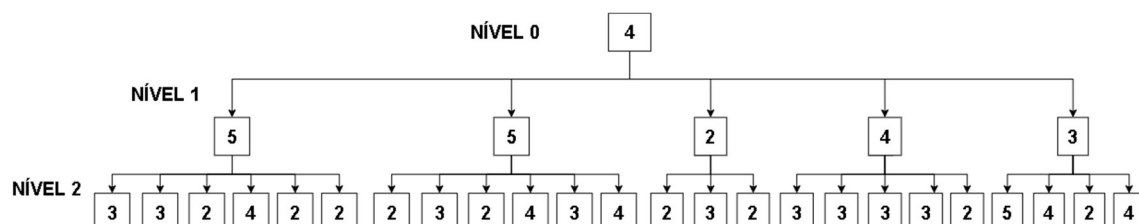


Fonte: Gráfico do autor (2024).

É crucial ter cuidado ao estruturar a árvore, pois problemas na remoção de registros podem afetar significativamente sua eficiência. A manutenção do balanceamento da árvore B, mesmo após a inserção e remoção de IDs, é um fator determinante para garantir um desempenho consistente.

As figuras 6 e 7 ilustram a estrutura da árvore B durante as inserções de dados, mostrando como ela se mantém balanceada após operações de remoção e inserção. Essa capacidade de organização é um dos principais motivos para a recomendação da árvore B em sistemas com grandes volumes de dados. Na figura 6, os números nos nós indicam a quantidade de chaves, enquanto a figura 7 mostra os IDs na árvore B.

Figura 6 — Árvore B balanceada



Fonte: Figura do autor (2024).





# XVII Semana Nacional de Ciência e Tecnologia

Biomass do Brasil: Diversidade, Saberes e Tecnologias Sociais

02 a 06 de Dezembro de 2024

ISSN 2594-8237

**Figura 7 — Árvore B balanceada impressão**

```
Interno (Nível 0): ['DAM198712N', 'DAM201715N', 'DAM203508A', 'DAM207801A']
Interno (Nível 1): ['DAM196404N', 'DAM197310N', 'DAM197802N', 'DAM198502A', 'DAM198701N']
Folha (Nível 2): ['DAM195701N', 'DAM196111N', 'DAM196309A']
Folha (Nível 2): ['DAM196910A', 'DAM197106N', 'DAM197208N']
Folha (Nível 2): ['DAM197412N', 'DAM197501N']
Folha (Nível 2): ['DAM197901A', 'DAM197906N', 'DAM198103N', 'DAM198402N']
Folha (Nível 2): ['DAM198512N', 'DAM198604N']
Folha (Nível 2): ['DAM198811A', 'DAM198910N']
Interno (Nível 1): ['DAM199410A', 'DAM199602N', 'DAM199804A', 'DAM199905N', 'DAM200812N']
Folha (Nível 2): ['DAM199112A', 'DAM199304N']
Folha (Nível 2): ['DAM199507ADAM825190A', 'DAM199601N']
Folha (Nível 2): ['DAM199612A', 'DAM199702A']
Folha (Nível 2): ['DAM199805A', 'DAM199806A', 'DAM199813N', 'DAM199900A']
Folha (Nível 2): ['DAM200302N', 'DAM200404N', 'DAM200702N']
Folha (Nível 2): ['DAM200906N', 'DAM201109A', 'DAM201214A', 'DAM201511A']
Interno (Nível 1): ['DAM202101A', 'DAM202702A']
Folha (Nível 2): ['DAM201803A', 'DAM201902A']
Folha (Nível 2): ['DAM202309A', 'DAM202506A', 'DAM202608A']
Folha (Nível 2): ['DAM203202N', 'DAM203215A']
Interno (Nível 1): ['DAM204801N', 'DAM205905N', 'DAM206601N', 'DAM207408A']
Folha (Nível 2): ['DAM203906A', 'DAM204214A', 'DAM204702A']
Folha (Nível 2): ['DAM205111A', 'DAM205310A', 'DAM205709N']
Folha (Nível 2): ['DAM205908A', 'DAM206108N', 'DAM206309A']
Folha (Nível 2): ['DAM206712N', 'DAM207309N', 'DAM207408A']
Folha (Nível 2): ['DAM207612A', 'DAM207715N']
Interno (Nível 1): ['DAM209715N', 'DAM211112A', 'DAM213514N']
Folha (Nível 2): ['DAM208004N', 'DAM208208A', 'DAM208412N', 'DAM208906A', 'DAM209618N']
Folha (Nível 2): ['DAM210213N', 'DAM210504A', 'DAM210507A', 'DAM211008N']
Folha (Nível 2): ['DAM211307A', 'DAM211711A']
Folha (Nível 2): ['DAM214102N', 'DAM214315N', 'DAM215215N', 'DAM393994A']
```

Fonte: Figura do autor (2024).

Comparando a estudos como os de Cavalcante & Temoteo (2018), que destacam a importância de gerenciar dados de forma eficiente em desastres, fica claro que a escolha da estrutura de dados é crucial. A árvore B se destaca como uma solução eficaz para sistemas de gerenciamento de desastres ambientais, graças à sua eficiência no manejo de grandes volumes de informações.

## 5. CONSIDERAÇÕES FINAIS

Este artigo apresentou o desenvolvimento e a análise de um sistema de gerenciamento de desastres ambientais utilizando a estrutura de dados Árvore B em Python. O objetivo principal foi criar uma solução para o armazenamento e busca de dados relacionados a desastres, visando melhorar a resposta e a prevenção em situações de emergência e analisar o uso da árvore B em comparação a outras estruturas de dados.

Os principais resultados demonstraram que a implementação da Árvore B proporciona um desempenho superior nas operações de inserção, busca e remoção de dados em cenários com grandes volumes de informações. A análise dos dados coletados revelou que a estrutura é capaz de manter a eficiência ao longo do tempo, o que é crucial para a gestão de desastres ambientais.

Apesar dos avanços, o sistema apresenta algumas limitações, como a ausência de uma interface gráfica para facilitar a interação com os usuários. Ainda assim, o sistema representa um progresso significativo na gestão de desastres ambientais, com potencial para apoiar órgãos governamentais na melhoria de suas respostas a emergências.

### 5.1 Trabalhos Futuros

Para trabalhos futuros, recomenda-se o desenvolvimento de uma interface gráfica para tornar o sistema mais intuitivo e acessível aos usuários. Além disso, a



## XVII Semana Nacional de Ciência e Tecnologia

Biomias do Brasil: Diversidade, Saberes e Tecnologias Sociais

02 a 06 de Dezembro de 2024

ISSN 2594-8237

implementação de um banco de dados possibilitará o armazenamento persistente das informações, permitindo consultas históricas e enriquecendo os nós da Árvore B com dados adicionais. Essas evoluções tornarão o sistema ainda mais robusto e aplicável em diferentes contextos de gestão de desastres ambientais.

### REFERÊNCIAS

- ABRANTES, MF de; NETO, MDS Desastre natural em Natal-RN: percepções de moradores. *Pesquisa, Sociedade e Desenvolvimento*, v. 16, 2021. DOI: 10.33448/rsd-v10i16.23465. Acesso em: 7 nov. 2024.
- BÍSCARO, N. Cursos de análises de dados públicos com Python. 2023. Disponível em: <https://doi.org/10.55592/icimsteam.2022.5332615>. Acesso em: 14 nov. 2024.
- CAVALCANTE, G.; TEMOTEO, K. Desastre natural associado a evento de chuva intensa no município de Caldas Brandão, Paraíba. *Revista de Geociências do Nordeste*, v. 4, p. 84-98, 2018. Disponível em: <https://doi.org/10.21680/2447-3359.2018v4n0id13315>. Acesso em: 7 nov. 2024.
- CHAME, M.; SIANTO, L. Desastres ecológicos e a saúde: plêiade de ampla magnitude e baixa percepção. *Ciência & Trópico*, v. 45, n. 2, 2021. Disponível em: [https://doi.org/10.33148/cetropicov45n2\(2021\)art2](https://doi.org/10.33148/cetropicov45n2(2021)art2). Acesso em: 7 nov. 2024.
- DUTRA, D.; GENTA, A.; COELHO, C. Modelagem de superfície de custo aplicado ao mapeamento de áreas vulneráveis ao rompimento de barragens, quadrilátero ferífero – MG. *Caderno de Geografia*, v. 31, n. 65, p. 425, 2021. Disponível em: <https://doi.org/10.5752/p.2318-2962.2021v31n65p425>. Acesso em: 7 nov. 2024.
- GONZALEZ, A. Impacto dos desastres naturais em uma população do sul do Brasil e a importância da educação ambiental para redução dos riscos. *Remea – Revista Eletrônica do Mestrado em Educação Ambiental*, v. 40, n. 1, p. 53-73, 2023. Disponível em: <https://doi.org/10.14295/remea.v40i1.13935>. Acesso em: 7 nov. 2024.
- LIMA, R. Ordenação baseada em árvores de fusão. 2014. Disponível em: <https://doi.org/10.48550/arxiv.1407.6753>. Acesso em: 10 nov. 2024.
- SOUZA, J.; COELHO, S. Uma biblioteca gráfica para aprendizagem de algoritmos e estruturas de dados. *Revista Brasileira de Informática na Educação*, v. 23, n. 1, p. 110, 2015. Disponível em: <https://doi.org/10.5753/rbie.2015.23.01.110>. Acesso em: 7 nov. 2024.
- TSAL, M.; CHANG, H. Contribution of accessibility to urban resilience and evacuation planning using spatial analysis. *International Journal of Environmental Research and Public Health*, v. 20, n. 4, p. 2913, 2023. Disponível em: <https://doi.org/10.3390/ijerph20042913>. Acesso em: 14 nov. 2024.
- ZIVIANI, N. *Projeto de algoritmos: com implementações em Java e C++*. São Paulo: Cengage Learning Nacional, 2007. 644 p.