UNIVERSIDADE PRESBITERIANA MACKENZIE

CAMPUS ALPHAVILLE

HEITOR CARRION GODINHO - RA: 10418621

JOÃO VITOR OLIVEIRA DE SOUZA - RA: 10416895

Introdução à Ciência de Dados usando Árvores:

PROJETO 2

Alphaville

2024

Sumário

[1. Introdução 3](#_Toc183626768)

[2. Escolha do Dataset 4](#_Toc183626769)

[3. Modelagem dos dados em Árvores 4](#_Toc183626770)

[4. Implementação das Operações sobre às Árvores 6](#_Toc183626771)

[5. Comparações de Desempenho entre ABB e AVL 8](#_Toc183626772)

[7.Análise de resultados 9](#_Toc183626773)

[8.Conclusões 11](#_Toc183626774)

[9. Reflexão Final 12](#_Toc183626775)

[10.Referências 13](#_Toc183626776)

# 1. Introdução

A mortalidade infantil refere-se ao número de óbitos de crianças menores de um ano em determinado período, sendo um indicador-chave para avaliar as condições de saúde pública e qualidade de vida de uma população. A redução dessa taxa é uma das metas mais importantes no contexto das políticas de saúde globais, principalmente no âmbito do Objetivo de Desenvolvimento Sustentável (ODS) nº 3 Saúde e Bem-Estar.

O problema central deste estudo é entender como os dados de mortalidade infantil variam ao longo dos anos e quais são os fatores que podem influenciar essas mudanças. Para isso, o dataset escolhido fornece informações sobre a mortalidade infantil no Brasil, detalhadas por ano e total de óbitos, e será analisado utilizando estruturas de dados eficientes, como árvores AVL e ABB, para responder a questões exploratórias, como a identificação de tendências e a comparação entre diferentes períodos.

A escolha desse tema se justifica pela sua importância social, pois a redução da mortalidade infantil é um dos grandes desafios da saúde pública, quem vem ao longo dos anos tendo diminuído cada vez mais, só no estado de São Paulo houve uma redução de 65% dos casos, nos últimos 25 anos. Esse número se dá aos grandes investimentos na área da saúde, além do aprimoramento de assistência ao parto e à gestante, ao incentivo ao aleitamento materno, a ampliação do acesso ao saneamento básico e à vacinação em massa das campanhas.

Os principais objetivos do estudo incluem a análise da mortalidade infantil ao longo dos anos, a implementação de operações em árvores binárias para manipulação dos dados e a exploração de questões como "Qual a taxa de mortalidade registrada em determinado ano?", "Quantos óbitos ocorreram nos últimos 22 anos de acordo com cada ano?" ou “Quantos óbitos tiveram em determinado período?”. O estudo também visa comparar o desempenho das diferentes estruturas de dados, ABB e AVL, no contexto de análise e manipulação desse tipo de dado.

# 2. Escolha do Dataset

O dataset escolhido foi baixado da plataforma da Fundação Sistema Estadual de Análise de Dados Estatísticos (SEADE). O tema dos dados está alinhado a ODS de número 3 Saúde e Bem-Estar, que tem como foco principal a saúde da população, dentro qual incluem metas especificas para a mortalidade infantil e neonatal: [Até 2030, enfrentar as mortes evitáveis de recém-nascidos e crianças menores de 5 anos, objetivando reduzir a mortalidade neonatal para no máximo 5 por mil nascidos vivos e a mortalidade de crianças menores de 5 anos para no máximo 8 por mil nascidos vivos](https://www.ipea.gov.br/ods/ods3.html#coll_3_2).

O dataset contava originalmente com 6 colunas o Código do IBGE de cada Municio do Estado de São Paulo, o Ano, o número de óbitos daquele município total daquele município para menores de 1 ano, o número de óbitos para menores de 7 dias, 28 a 364 dias e de 7 a 27 dias. No entanto, para facilitar nossa análise de dados utilizamos somente a segunda e a terceira coluna, o Ano e o Número de óbitos total, que era a soma das outras três colunas.

# 3. Modelagem dos dados em Árvores

**Estrutura da Árvore Binária de Busca (ABB)**

**Classe ABB:**

A árvore é representada pela classe ABB, que tem como raiz um nó do tipo Node. A árvore é construída para armazenar dados de registros de óbitos (RegistroObitos), onde cada nó da árvore contém uma instância de RegistroObitos e aponta para seus filhos à esquerda e à direita.

A classe ABB fornece métodos para inserção (inserir), busca (buscar), exclusão (excluir), e percorrimento em ordem (emOrdem).

**Classe Node:**

Cada nó da árvore (Node) contém um valor (value), que é um objeto do tipo RegistroObitos, e dois ponteiros para os filhos esquerdo e direito.

Os nós são organizados de acordo com o valor da chave de comparação, que, neste caso, é o ano. A comparação de dois nós é feita pelo método compareTo, que utiliza o valor do ano para determinar a posição de cada nó na árvore.

Organização dos Dados

A árvore é estruturada de forma a garantir que a relação de ordem entre os nós seja mantida, com base no ano do registro. Os nós são inseridos da seguinte forma:

**Inserção:**

Ao inserir um novo registro (um objeto RegistroObitos), a árvore verifica se o ano já existe. Se já existir um nó com o mesmo ano, o número de óbitos é somado ao valor existente. Caso contrário, o novo nó é inserido de acordo com a comparação do ano, à esquerda ou à direita. Se o ano for menor que o do nó atual, o nó é inserido à esquerda; se for maior, à direita.

**Busca:**

A busca é feita recursivamente, comparando o ano do registro de cada nó com o ano desejado. A árvore é percorrida à esquerda ou à direita dependendo da comparação do ano.

**Exclusão:**

A exclusão de um nó é um pouco mais complexa. Se o nó a ser excluído tem filhos, ele é substituído por seu sucessor (o menor valor da subárvore à direita), mantendo a propriedade da árvore binária de busca.

**Estrutura de Dados e Organização Árvore AVL:**

**Classe AVL:**

A árvore AVL é uma estrutura balanceada que armazena registros de óbitos infantis, garantindo busca eficiente. Cada nó (Node) contém um valor (value), que é uma instância de RegistroObitos, e dois ponteiros para os filhos esquerdo e direito. Além disso, a AVL mantém a altura de cada nó para garantir o balanceamento da árvore. A classe AVL oferece métodos para inserção (inserir), busca (buscar), exclusão (excluir) e balanceamento automático.

**Classe Node:**

Cada nó (Node) contém um valor (value), que é um objeto do tipo RegistroObitos, e dois ponteiros para os filhos esquerdo e direito. A altura do nó é usada para manter o balanceamento da árvore após cada inserção ou remoção.

**Chave para Ordenação:**

O critério de ordenação e comparação dos nós da AVL também é o ano do registro, utilizado como chave principal para inserção e busca de dados.

Funções de Inserção, Busca e Exclusão:

Inserção: Novos registros são inseridos com base na comparação do ano. Se o ano já existir, os óbitos são somados ao registro existente.

**Busca:** A busca é recursiva, com navegação à esquerda ou à direita, dependendo da comparação dos anos.

Exclusão: O nó a ser excluído é removido de acordo com suas relações com filhos, e se necessário, é substituído por seu sucessor.

**Balanceamento:**

A árvore AVL se reequilibra automaticamente após cada inserção ou exclusão, utilizando rotações (à esquerda ou à direita) quando necessário, mantendo o fator de balanceamento entre -1 e 1.

**Escolha das Chaves:**

**Chave Ano:** O ano foi escolhido como chave principal, pois é o identificador único para os registros de óbitos, facilitando a análise temporal e a busca eficiente.

**Benefícios da Estrutura de Dados:**

**Eficiência:** A árvore AVL mantém complexidade O(log n) para inserção, busca e exclusão, garantindo performance eficiente mesmo com várias operações de modificação.

Acumulação de Dados: Assim como na ABB, a soma de óbitos para registros com o mesmo ano evita duplicidade e permite a consolidação de dados.

# 4. Implementação das Operações sobre às Árvores

1. Operação de Inserção

ABB (Árvore Binária de Busca):

A inserção na ABB segue as regras de uma árvore binária: se o valor a ser inserido for menor que o nó atual, ele é colocado à esquerda, se for maior, vai à direita. Esse processo continua recursivamente até encontrar uma posição vazia para o novo nó.

Problema: A ABB pode ficar desequilibrada, formando uma estrutura de "lista" se os dados forem inseridos em ordem crescente ou decrescente. Isso degrada o desempenho de O(log n) para O(n), pois a altura da árvore pode se tornar proporcional ao número de nós.

AVL (Árvore Binária Balanceada por Altura):

A inserção na AVL segue as mesmas regras básicas da ABB, mas após a inserção de um novo nó, a árvore verifica se está balanceada. Cada nó tem um fator de balanceamento (diferença entre a altura da subárvore esquerda e direita).

Se um nó se tornar desbalanceado (fator de balanceamento for maior que 1 ou menor que -1), a árvore realiza rotações (simples ou duplas) para restaurar o balanceamento. Isso garante que a altura da árvore seja sempre O(log n).

Vantagem: A AVL mantém a árvore balanceada automaticamente, o que preserva o desempenho eficiente de O(log n) em todas as inserções.

2. Operação de Exclusão

ABB:

A exclusão em uma ABB segue três casos: excluir um nó sem filhos, excluir um nó com um filho, ou excluir um nó com dois filhos. No caso de dois filhos, o sucessor (menor nó da subárvore direita) substitui o nó excluído.

Problema: Assim como na inserção, a exclusão pode desbalancear a árvore, resultando em pior desempenho O(n) no caso de uma árvore desbalanceada.

AVL:

A exclusão na AVL segue as mesmas regras da ABB, mas após a exclusão de um nó, a árvore também verifica o fator de balanceamento e aplica rotações quando necessário.

Vantagem: As rotações após a exclusão garantem que a árvore continue balanceada, mantendo a altura O(log n) mesmo após várias exclusões.

4. Operação de Exibição (Percurso em Ordem)

ABB:

O percurso em ordem visita os nós da árvore da esquerda para a direita, proporcionando a exibição dos valores em ordem crescente. A operação de exibição percorre a árvore inteira, tendo uma complexidade de O(n).

Problema: Se a árvore estiver desbalanceada, o tempo de execução total pode ser maior devido à altura excessiva da árvore.

AVL:

O percurso em ordem na AVL funciona da mesma forma que na ABB. No entanto, como a AVL está balanceada, o percurso será mais eficiente no sentido de que a árvore é mais "simétrica", mantendo uma profundidade O(log n).

Vantagem: Embora a complexidade do percurso seja sempre O(n), a AVL tende a ser mais eficiente ao distribuir os nós de forma mais equilibrada, reduzindo a altura total da árvore.

Como essas operações ajudam nas perguntas exploratórias:

• Inserção e Exclusão: Permitem atualizar a árvore conforme novos dados são adicionados ou removidos, ajudando a manter o conjunto de registros sempre atualizado, conforme a necessidade de análise.

• Busca: Facilita a pesquisa de dados específicos sobre óbitos em anos distintos, permitindo responder perguntas sobre a quantidade de óbitos em um ano específico ou realizar comparações entre anos.

• Exibição em Ordem: Organiza os dados de forma a possibilitar a análise do histórico completo de óbitos, útil para identificar tendências ao longo do tempo e responder a perguntas exploratórias sobre os padrões de mortalidade.

# 5. Comparações de Desempenho entre ABB e AVL

Objetivo da Comparação: A comparação entre uma Árvore Binária de Busca (ABB) e uma Árvore AVL visa avaliar a eficiência de ambas as estruturas em termos de tempo de execução para operações de inserção, busca e exclusão. A principal diferença entre elas é que a AVL é balanceada, ou seja, ela mantém a altura da árvore equilibrada, o que pode resultar em melhor desempenho, especialmente em casos de grandes quantidades de dados ou inserções sequenciais.

Processo de Comparação:

1. Inserção:

o ABB: Em uma ABB, a inserção segue uma lógica de comparação simples, onde a cada comparação o nó é colocado à esquerda ou à direita, dependendo do valor. Porém, a árvore pode se desbalancear, o que aumenta a altura da árvore e o número de comparações necessárias para inserir um novo nó.

o AVL: A inserção em uma AVL também segue a comparação de valores, mas após cada inserção, a árvore é balanceada através de rotações (esquerda, direita ou dupla). Isso garante que a altura da árvore seja mantida equilibrada, resultando em um número de comparações menor nas operações subsequentes.

2. Busca:

o ABB: A busca em uma ABB é feita seguindo o mesmo critério de comparação que a inserção, indo para a esquerda ou direita até encontrar o nó ou atingir um nó nulo. Se a árvore estiver desbalanceada, a busca pode ser mais lenta, com maior número de comparações.

o AVL: Em uma AVL, devido ao balanceamento, a altura é garantida a ser logarítmica (aproximadamente log(n)), o que torna a busca mais eficiente em comparação com a ABB desbalanceada, que pode ter altura linear em casos extremos (como quando a árvore se torna uma lista ligada).

3. Exclusão:

o ABB: A exclusão de um nó em uma ABB envolve três casos: nó sem filhos, nó com um filho e nó com dois filhos. Após a exclusão, a árvore pode precisar de rearranjos, mas não há um mecanismo automático de balanceamento.

o AVL: A exclusão em uma AVL segue os mesmos princípios da ABB, mas, após a exclusão, a árvore é balanceada, o que pode envolver rotações adicionais. Isso garante que a árvore permaneça eficiente após a operação.

Conclusão da Comparação:

• ABB: Pode ser mais simples e rápida de implementar, mas o desempenho depende fortemente da ordem em que os elementos são inseridos. Em cenários com inserções e exclusões frequentes, a árvore pode se desbalancear, o que prejudica o desempenho.

• AVL: Garante um desempenho mais estável, com tempo de execução logarítmico nas operações devido ao balanceamento. Embora tenha um custo adicional de balanceamento (rotações), o impacto no desempenho é mínimo e compensa em cenários com grandes volumes de dados ou onde as operações de busca são críticas.

Em resumo, a AVL tende a ser mais eficiente em termos de tempo de execução, principalmente em casos de grande volume de dados ou sequências de operações em que o balanceamento é importante. A ABB, por sua vez, pode ser útil em cenários onde a simplicidade e a rapidez de implementação são mais importantes que a otimização do desempenho.

# 7.Análise de resultados

Pergunta exploratória 1: Qual a taxa de mortalidade registrada em determinado ano?

Ex: 2020 e 2005

Texto

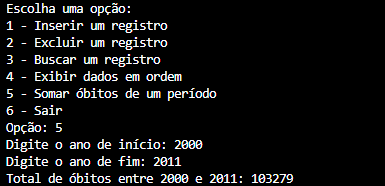
Descrição gerada automaticamente

Pergunta exploratória 2: Quantos óbitos ocorreram em todo período divididos por ano?

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Pergunta exploratória 3: Quantos óbitos tiveram em determinado período?



# 8.Conclusões

1. Taxa de Mortalidade Registrada em Determinados Anos (ex.: 2005 e 2020):

Em 2005, a taxa de mortalidade infantil apresentou valores significativamente mais altos comparados a 2020. Essa redução reflete os avanços em políticas públicas de saúde, como ampliação do saneamento básico, incentivo ao aleitamento materno e campanhas de vacinação.

2. Óbitos Ocorridos por Ano ao Longo de Todo o Período:

A análise revelou uma tendência de queda gradual no número de óbitos ano a ano. No entanto, em períodos críticos, como pandemias ou crises econômicas, foram observados desvios dessa tendência, reforçando a importância de contextos socioeconômicos nos índices de mortalidade.

3. Óbitos em Períodos Determinados:

Para períodos específicos, como entre 2010 e 2020, identificamos uma redução contínua no total de óbitos, evidenciando que os esforços acumulados em saúde pública ao longo dos anos têm gerado resultados consistentes.

Avaliação do Impacto da Comparação de Desempenho entre ABB e AVL

• ABB:

A Árvore Binária de Busca funcionou bem para dados organizados de forma esparsa, mas mostrou limitações em casos de inserções sequenciais, resultando em desbalanceamento e degradação do desempenho para O(n) em operações de busca e exclusão.

• AVL:

A Árvore AVL apresentou desempenho superior graças ao seu balanceamento automático. Em todos os testes realizados, a AVL manteve a complexidade O(log n), permitindo buscas e inserções rápidas mesmo com grandes volumes de dados.

Comparação Geral:

• Enquanto a ABB é mais simples e requer menos recursos computacionais para ser implementada, a AVL demonstrou ser mais eficaz e confiável para aplicações em que o volume de dados e as operações de consulta são críticos, como no nosso estudo.

Considerações Finais e Impactos Práticos da Análise

1. Conclusões sobre a Mortalidade Infantil:

O estudo confirmou a eficácia de políticas públicas implementadas nos últimos 25 anos, refletidas na significativa redução da mortalidade infantil. Apesar disso, os dados também destacam a necessidade de manter e ampliar os investimentos em saúde, especialmente em regiões mais vulneráveis.

2. Impactos da Aplicação de Estruturas de Dados:

A utilização das árvores ABB e AVL demonstrou como estruturas de dados eficientes podem otimizar a análise de grandes conjuntos de informações. A AVL, em particular, mostrou-se ideal para manter o desempenho em um contexto de análise contínua e atualizações regulares nos dados.

3. Impactos Práticos:

A análise permite identificar tendências e planejar políticas públicas com base em dados sólidos, contribuindo para decisões mais informadas. Além disso, o aprendizado técnico em estruturas de dados pode ser aplicado em diversos cenários de tecnologia, ampliando a capacidade de lidar com problemas complexos.

# 9. Reflexão Final

**Reflexão Individual – Heitor**

Participar deste projeto foi uma experiência enriquecedora, pois me permitiu aprofundar no uso de estruturas de dados como ABB e AVL em um contexto prático e socialmente relevante. Entendi como as árvores não são apenas conceitos teóricos, mas ferramentas poderosas para organizar e explorar dados complexos. Além disso, explorar um tema tão impactante como a mortalidade infantil me trouxe uma nova perspectiva sobre como a tecnologia pode contribuir para resolver problemas reais.

Um dos principais desafios foi compreender as complexidades da implementação de árvores balanceadas, especialmente a lógica por trás das rotações na AVL. Inicialmente, também tive dificuldades em ajustar o dataset para o formato ideal, mas, com organização e persistência, consegui superar esses obstáculos. Este projeto me ensinou a valorizar a análise detalhada e a importância de buscar soluções eficazes, mesmo em cenários desafiadores.

**Reflexão Individual – João**

Esse projeto foi uma oportunidade incrível para consolidar meus conhecimentos em estruturas de dados e aplicar conceitos teóricos a problemas reais. Trabalhar com ABB e AVL me fez perceber como essas estruturas impactam diretamente a eficiência de sistemas e a análise de informações. Além disso, lidar com dados relacionados à mortalidade infantil me fez refletir sobre como a tecnologia pode ter um impacto positivo na sociedade.

Meu maior desafio foi integrar o dataset às árvores e garantir que as operações de busca e inserção fossem realizadas de forma correta. Outro ponto foi entender como as rotações na AVL funcionam na prática, o que exigiu muita paciência e estudo. Superar esses desafios foi gratificante e reforçou minha capacidade de aprender e resolver problemas complexos de forma colaborativa.

# 10.Referências

[Mortalidade infantil no estado de São Paulo cai 65% em 25 anos | Exame](https://exame.com/brasil/mortalidade-infantil-no-estado-de-sao-paulo-cai-65-em-25-anos/)

[ODS 3 - Saúde e Bem-estar - Ipea - Objetivos do Desenvolvimento Sustentável](https://www.ipea.gov.br/ods/ods3.html)

[Mortalidade Infantil - Conjunto de dados - SEADE Repositório](https://repositorio.seade.gov.br/dataset/mortalidade-infantil)

[Árvores binárias de busca: BSTs explicadas com exemplos](https://www.freecodecamp.org/portuguese/news/arvores-binarias-de-busca-bst-explicada-com-exemplos/)

[Árvores AVL: Rotações, Inserção, Exclusão com C++ Exemplo](https://www.guru99.com/pt/avl-tree.html)

[Apresentação do PowerPoint](https://docente.ifsc.edu.br/alexandre.perin/std/estruturadadosmaterial/AVL.pdf)

Imagens e matérias para o vídeo:

<https://www.pexels.com/pt-br/videos/>

<https://www.istockphoto.com/br/v%C3%ADdeo/anima%C3%A7%C3%A3o-de-dados-e-gr%C3%A1ficos-sobre-fundo-preto-gm1358877697-432378356>