

# Introdução

---

Neste trabalho você e sua equipe irão implementar e comparar dois algoritmos para balanceamento de árvores binárias de busca.

Os algoritmos implementados serão testados usando uma base de dados qualquer (mais detalhes a seguir), e o conjunto de operações típicos envolvendo as árvores: Inserção, Remoção e Busca.

Para cada uma dessas operações você vai precisar registrar as comparações realizadas envolvendo os elementos armazenados na árvore.

Para tanto a sua implementação dos algoritmos precisar ser argumentada com código que realize a contagem das comparações, e permita a apresentação dos valores das contagens. Além disso, a sua implementação deve estar preparada para receber qualquer tipo de dado.

## Argumentações nos algoritmos

---

Será necessário dotar as implementações com código que permita a contagem das comparações realizadas em cada operação (busca, remoção e inserção) realizada na árvore.

Para isso, considere a incorporação de estruturas na definição do tipo árvore, que permita armazenar:

1. O número de operações (busca, inserção e remoção);
2. O número de comparações realizadas por cada tipo de operação;
3. O número de manipulações (rotações) nas operações de inserção e remoção.

## As bases de dados

---

Sua implementação deve estar preparada para manipular qualquer tipo de dado, portanto deve estar neutralizada com relação aos dados armazenados.

Considere que para qualquer base de dados que veja ser utilizada, será necessário desacoplar a solução que usa a árvore, da implementação da árvore.

Para a realização de testes será disponibilizadas uma base contendo as seguintes informações:

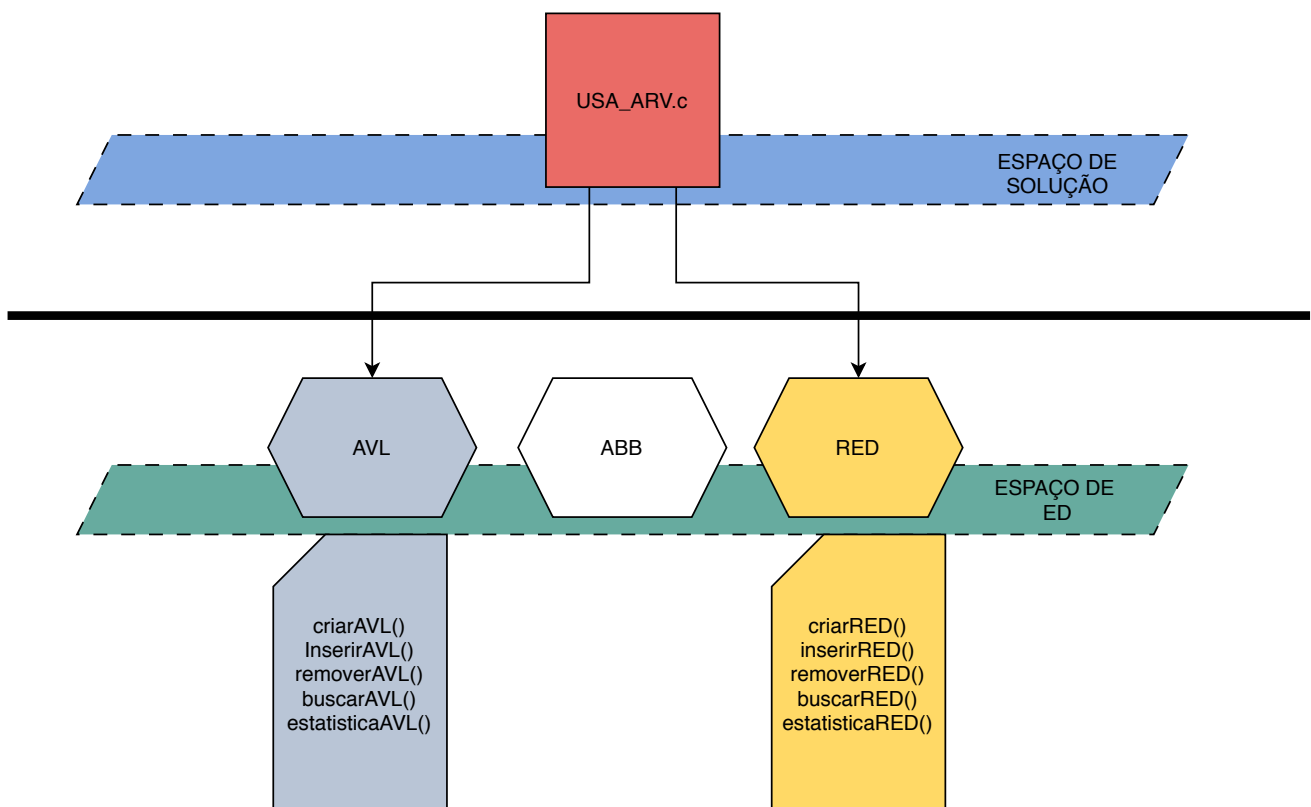
```
{  timestamp:inteiro,
    dia:inteiro,
    mes:inteiro,
    ano:inteiro,
    hora:inteiro,
    minuto:inteiro,
    radiacao_solar:inteiro,
    temperatura:double,
    umidade_relativa: double
}
```

# Sobre a submissão

Você e sua equipe precisarão submeter a implementação da sua solução, seguindo as orientações dadas a seguir:

1. O nome do arquivo de cada implementação `arv_<>.c` `arv_<>.h`
2. Os tipos árvores precisam ser nomeados `t_<*>`, onde `*` deve ser:
  - i) **red** para árvore vermelho e preto;
  - ii) **avl** para árvore AVL.
3. O relatório apresentado:
  - 2.1) Introdução - Apresentação do problema
  - 2.2) Métodos - Algoritmos implementados
  - 2.3) Resultados - As estatísticas coletadas e comentários
  - 2.4) Conclusão - Observações gerais sobre a realização do trabalho.

Na nomeação do arquivo, o (\*) asterísco deve ser substituído pelas (03) três primeiras letras do nome do algoritmo implementado, naquele arquivo. Os arquivos `arv_avl.c` e `arv_avl.h` são os nomes que você precisa usar para implementar a árvore AVL. Nos arquivos `arv_red.c` `arv_red.h` são os nomes que você precisa usar para implementar a árvore RED.



Os algoritmos que podem ser implementados:

- AVL (avl) [visualização apresentação](#)
- Red-Black (red) [visualização apresentação](#)

# Critérios para avaliação

---

Os critérios para avaliar o seu trabalho será como segue:

1. Implementou os dois Algoritmos (nota máxima 10 (dez))
2. Implementou apenas UM algoritmo (nota máxima 06 (seis))

Será ainda considerado:

1. Relatório apresentado;
2. Funcionalidades implementadas;
3. A neutralização das estruturas;

## Sobre as equipes

---

O trabalho deve ser implementado em equipe. As equipes podem ter ATÉ 4(quatro) membros.

No final do relatório, deve-se listar a contribuição de cada membro na elaboração do trabalho. Por exemplo, *Maria Júlia concebeu as estruturas de argumentação (ampliação) de funcionalidade de estatísticas; Pedro Paulo implementou a operação de inserção e remoção em uma AVL.*

Data de entrega: 28-06-2023.