



## Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

### Exercício sobre Árvore AVL

**Objetivo:** Praticar a implementação de árvore AVL.

O endereço do sistema de submissão é o <https://les.dsc.ufcg.edu.br:8443/EasyLabCorrection>.

Relembre o conceito de árvore AVL visto em sala de aula.

Atividades necessárias antes de iniciar o exercício:

1. Crie um projeto no Eclipse chamado LEDA, por exemplo (pode ser qualquer outro nome que lhe convier);
2. Descompacte o arquivo baixado (exceto o PDF) na pasta dos fontes (normalmente **src**) do seu projeto LEDA criado no seu workspace. O arquivo baixado tem a seguinte estrutura:
  - adt
  - avltree
  - AVLTree.java (INTERFACE DE UMA AVL)
  - AVLTreeImpl.java (IMPLEMENTAÇÃO PARCIAL DE UMA AVL)
  - bst
  - BSTNode.java
  - BST.java
  - BSTImpl.java (IMPLEMENTAÇÃO PRONTA DO ALUNO DE UMA BST)
  - bt
  - BTNode.java (CLASSE REPRESENTANDO UM NÓ DE UMA ÁRVORE BINÁRIA GÊNICA)
  - BT.java (INTERFACE DE UMA ÁRVORE BINÁRIA GÊNICA)
3. Para se concentrar na implementação de árvore AVL, você precisa utilizar sua implementação de árvore BST do roteiro anterior. **Portanto, é essencial que você TESTE sua BST para garantir que os métodos dela estejam funcionando corretamente.**
4. No Eclipse, selecione a pasta dos fontes no projeto LEDA e faça um refresh (apertar F5). Note que deve aparecer um pacote `adt.heap` contendo os arquivos mencionados acima.

Agora você está pronto para começar a trabalhar nas seguintes atividades:

1. Observe a interface `AVLTree.java`. Ela descreve os serviços de uma árvore AVL genérica.
2. Observe também a existência de implementação incompleta `AVLTreeImpl`. Você precisa implementar os métodos incompletos e sobrescrever os necessários.

### Instruções para o envio

Ao terminar o exercício, faça os seguintes passos:

1. Compacte a pasta **adt** que existe nos fontes de seu projeto LEDA (**src**) e retire suas classes de teste desse arquivo compactado. A compactação DEVE ser feita a partir do diretório raiz de seus fontes de forma a

preservar a estrutura de pastas que refletem a estrutura dos pacotes (package) Java. Por exemplo, voce deve ter um arquivo compactado NOME\_COMPLETO\_DO\_ALUNO.ZIP com a seguinte estrutura:

```
- adt
-- avltree
--- AVLTreeImpl.java (IMPLEMENTACAO PARCIAL DE UMA AVL)
-- bst
--- BSTImpl.java (IMPLEMENTACAO PRONTA DO ALUNO DE UMA BST)
```

2. Envie esse arquivo com sua solução para o sistema de submissão e verifique que o contador de submissões será alterado.

#### Observações finais:

- **A interpretação do exercício faz parte da atividade.**
- **A atividade é individual. A conversa entre alunos é proibida.**
- **É proibido coletar códigos prontos e adaptar. Implemente as questões. Isso é para seu aprendizado.**
- **Caso voce observe qualquer problema no sistema de submissão, contacte o professor imediatamente.**
- **Se voce nao compactar o arquivo seguindo a estrutura de diretórios a compilação não terá sucesso e o sistema mostrará isso. Erro de compactação serão de responsabilidade do aluno. O professor não ajudará o aluno nesse item. É só seguir as instruções deste arquivo.**