

# Relatório Projeto 3.4 AED 2020/2021

Nome: Pedro Afonso Ferreira Lopes Martins

Nº Estudante: 2019216826

TP (inscrição): PL8

Login no *Mooshak*: 2019216826

Nº de horas de trabalho: 04H Aulas Práticas de Laboratório: 02H Fora de Sala de Aula: 02H

(A Preencher pelo Docente) CLASSIFICAÇÃO:

Comentários:

## Estrutura de Dados Principal usada em cada sub-projeto:

PROJ 3.1: Árvore de Merkle (Merkle Tree)

PROJ 3.2: Árvore AVL (AVL Tree)

PROJ 3.3: Árvore de Splay (Splay Tree)

Estruturas de Dados usadas	Merkle Tree	AVL Tree	Splay Tree
VANTAGENS GERAIS (max 3)	<ul style="list-style-type: none"><li>Bastante eficazes para verificação e validação de dados.</li><li>Uso reduzido de memória aquando da sua utilização.</li></ul>	<ul style="list-style-type: none"><li>Ordena as inserções.</li><li>Consultas são bastante rápidas, especialmente em comparação com outras estruturas para um número elevado de casos.</li><li>É balanceada.</li></ul>	<ul style="list-style-type: none"><li>É balanceada</li><li>Consulta é bastante rápida quando temos nós com prioridade (muito mais acedidos que outros.</li></ul>
DESVANTAGENS GERAIS (max 3)	<ul style="list-style-type: none"><li>Uso elevado de CPU.</li><li>Maior uso de RAM aquando da sua execução em comparação com outras estruturas de dados para casos com árvores com bastante profundidade.</li></ul>	<ul style="list-style-type: none"><li>Perda de tempo aquando da ordenação após uma inserção devido a múltiplas rotações.</li></ul>	<ul style="list-style-type: none"><li>Quando a consulta é aleatória pode demorar bastante tempo devido às constantes ordenações.</li><li>Quanto mais nós inseridos, maior o número de rotações necessárias e maior o tempo gasto.</li></ul>
Justificação para a escolha no PROJ 3.1	Escolhi Merkle Tree para este projeto uma vez que é o mais adequado para guardar e verificar hash codes, seguindo a indicação fornecida no enunciado do projeto.		
Justificação para a escolha no PROJ 3.2	Uma vez que nos é comunicado que vão existir muitas mais consultas do que inserções a AVL tree é a estrutura perfeita, uma vez que é a estrutura de dados balanceada cujas consultas são as mais rápidas graças ao ordenamento após inserção, sendo mais equilibrada que a Red-Black Tree. O facto de que as inserções são lentas e nos é dito que o número de inserções é reduzido em comparação com as consultas serve como justificação da escolha efetuada.		
Justificação para a escolha no PROJ 3.3	Splay Tree é a melhor escolha visto que nos é dito no enunciado que os acessos aos clientes é feito com prioridade (uma pequena percentagem de utilizadores é acedido na maioria dos casos), como tal esta estrutura é ideal uma vez que as ordenações após consulta, que demoram bastante tempo de forma aleatória visto que o cliente pode estar num nó com bastante profundidade e serão precisas várias rotações para o trazer para o topo, serão reduzidas uma vez que o cliente tem uma grande probabilidade de estar perto da raíz e, assim, serem necessárias poucas rotações.		