

Trabalho Final

Linguagem de Programação II e Estruturas de Dados II

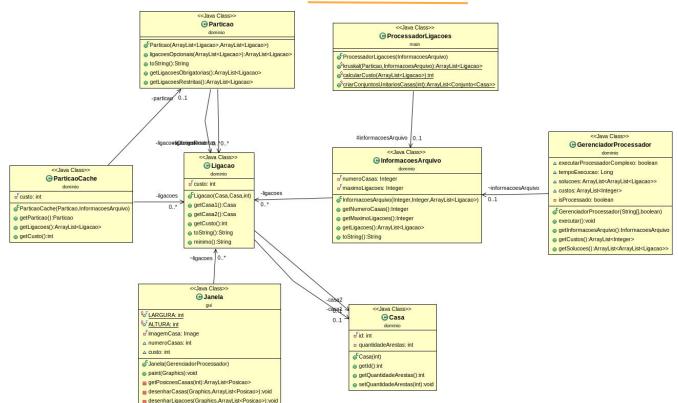
Grupo:

- → Deyvid William Silva de Medeiros
- → Guilherme Euller Dantas Silva
- → Nathãn Vitor de Lima





👺 Diagrama de Classes



desenharCustoTotal(Graphics):void





👺 Diagrama de Classes

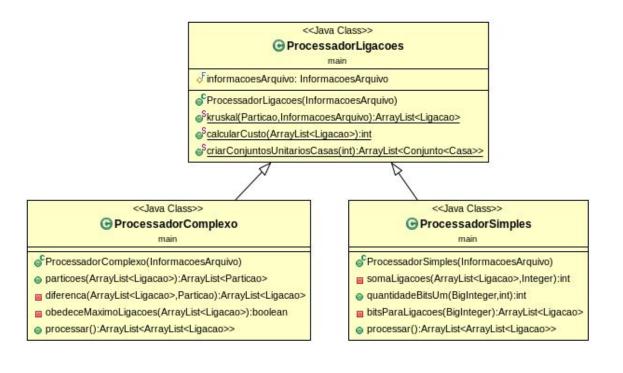






Diagrama de Classes

<< lava Class>> (Main main

Main() Smain(String[]):void

<<Java Class>> Posicao Fx: int oFy: int Posicao(int,int) getX():int

a getY():int

SparseArgs(String[]):String

<< lava Class>> LeitorArquivos

LeitorArquivos() SlerArquivo(String):InformacoesArquivo <<Java Class>> PrintFile

PrintFile()

SgenerateFile(GerenciadorProcessador):void

SqetBiggerSize(ArrayList<ArrayList<Ligacao>>):int[] sgetCaracter(int,char):String

SqetBiggerCost(ArrayList<Integer>):int

<< Java Class>> Arquivolnvalido excecao

Arquivolnvalido(String)





Etapa 1





Kruskal









K-best





```
List = \{A\}
Calculate_MST (A)
while MST ≠ Ø do
     Get partition P_s \in \text{List} that contains the smallest spanning tree
     Write MST of P_s to Output File
     Remove Ps from List
                                               PROCEDURE PARTITION (P)
     Partition(P_s).
                                               P_1 = P_2 = P;
                                               for each edge i in P do
                                               if i not included in P and not excluded from P then
                                                     make i excluded from P_i;
                                                     make i included in P_2;
                                                     Calculate MST (P1);
                                                     if Connected (P_I) then
                                                        add P1 to List;
                                                    P_1 = P_2;
```





Etapa 2





Obrigado!

