

Algoritmo e Estrutura de Dados 2

Exercício de Programação 2:

Construindo um grafo de encontros presenciais da RMSP

Relatório 2 - Histograma dos graus dos nós para o cenário 1

Paulo Rogério Andrade de Oliveira - 9365783

Cenário 1 - *“tudo está funcionando normalmente: neste cenário são consideradas todas as origens e todos os destinos e as arestas são criadas entre quaisquer pessoas que frequentem os mesmos lugares (origem e/ou destinos coincidentes)”*.

A abordagem no segundo programa foi similar ao primeiro, foi usada uma interface Map afim de distribuir os vértices (frequentadores), agora pelo grau do nó.

A classe *GrafoService* traz consigo alguns métodos de tratamento do grafo por meio de uma lista de adjacência que se inicializada e incrementada na classe *EstruturaNo*.

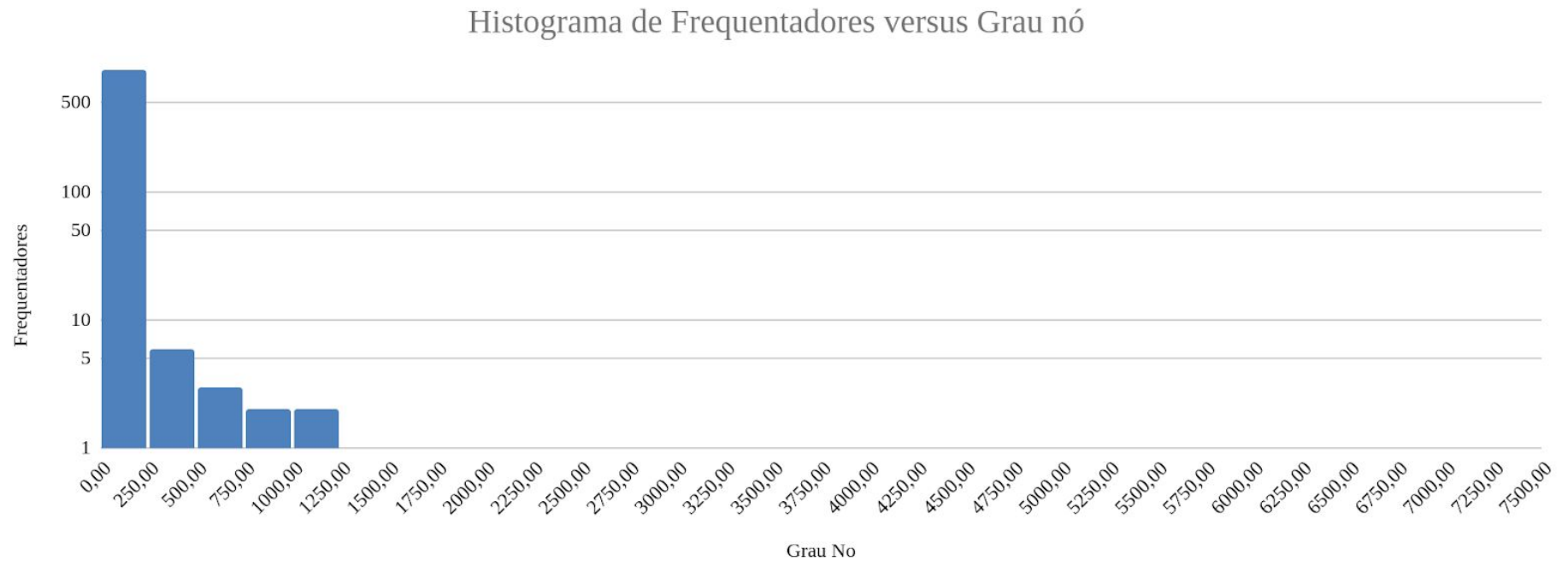
O *mapaGrafo* do tipo `Map<Integer, List<Integer>>` é inicializado na classe *GrafoService* e será incrementado pelo método *processarGrafo*, que recebe `List<String[]> dados`, adiciona os vértices e retonar *mapaGrafo* já populado.

O método *adicionarVertices* é acionado pelo método supracitado, recebe `int no`, `int no_adj` e caso `no` estiver contido no *mapaGrafo*, `no_adj` é adicionado na lista, caso contrário, uma nova lista é criada com os nós.

O método *encontrarDistribuicaoGrauNos*, presente na classe *GrafoService* que recebe o grafo (do tipo `Map<Integer, List<Integer>>`), incrementa a distribuição do nó que está sendo consultado no momento.

A variável *listaAdjacencia* é uma lista de inteiros que é inicializada e incrementada na classe *EstruturaNo*, recebendo um novo nó adjacente caso o nó em análise estiver nela contido.

Histograma:



Projeto no Github: <https://github.com/pauloanalytics/AED/tree/master/EP2>