

GITHUB: <https://github.com/ArthurBusquet/trabalho-teoria-dos-grafos>

Arthur: responsável por criar o método `deleta_no` na classe `grafoLista` que recalcula automaticamente os valores da ordem e grau do grafo e dos vértices. A `deleta_no`, assim como os métodos da `grafoLista`: `deleta_aresta`, `set_no` e `set_aresta` utilizam os métodos auxiliares `set_vertice`, `set_aresta`, `get_vertice`, `get_aresta` e `removeConexao` que criei anteriormente para realizar suas operações de forma coesa. Também realizei correções de erros no funcionamento do método de busca em profundidade e consequentemente o método `n_conexo`. Apoiei meus colegas no debug de problemas envolvendo a realocação dinâmica da matriz.

Por fim, gerenciei o novo repositório no github, resolvendo conflitos em merges, avaliando pull requests e apoiando os colegas em eventuais dúvidas sobre git.

Bernardo: Responsável pela implementação da remoção de arestas e suas funções auxiliares no grafo em lista encadeada. Contribuiu para o gerenciamento de branches do repositório, incluindo auxílio em merges entre branches. Implementou a funcionalidade completa para remover arestas em um grafo representado por lista encadeada, incluindo a criação de funções auxiliares necessárias para garantir a integridade da estrutura de dados. Participou ativamente no gerenciamento de branches do repositório, contribuindo para a organização do código e garantindo a rastreabilidade das alterações. Colaborou com outros desenvolvedores, auxiliando na resolução de conflitos e realizando merges entre branches de forma eficiente para integrar as contribuições de todos os membros da equipe.

Geovanni: Responsável pela implementação das Funções `Deleta_no - Matriz`, `Deleta_aresta_nao_direcionada - Matriz`, `Deleta_aresta_direcionada - Matriz`, `reorganiza_vetor_pesos - matriz`, `reorganiza_matriz - matriz`.

Pedro: Responsável pela elaboração da documentação da Parte 2 do trabalho, garantindo a clareza e organização das informações técnicas do projeto. Além disso, desenvolveu as funções `novo_no` e `nova_aresta` para ambas as implementações do grafo, tanto na lista encadeada quanto na matriz de adjacência. Essas funções foram fundamentais para permitir a inserção dinâmica de novos vértices e arestas, assegurando a correta manipulação da estrutura do grafo e sua escalabilidade.

Jorge: criei a função `Maior menor distância()`, ajudei na integração dos códigos e no ajuste na matriz, para redimensioná-la quando passasse do limite, era necessário ao final do trabalho uma atenção maior nessa questão, mas por conta do tempo não conseguimos editar, além da integração dos códigos dos outros membros com o projeto, participei ativamente na sincronização dos trabalhos e das edições finais e organização estrutural, dentre outras.