**Relatório do Projeto**

Parte 1

|  |  |
| --- | --- |
| Nome do Integrante | RA |
| Lucas Meres | 10395777 |
| Renan Tagliaferro | 10395211 |
| Thiago Leandro Liporace | 10395816 |

**Relatório**

Análise da Rede Social e Dinâmicas de Amizades Online no Instagram

DEFINIÇÃO DO PROBLEMA:

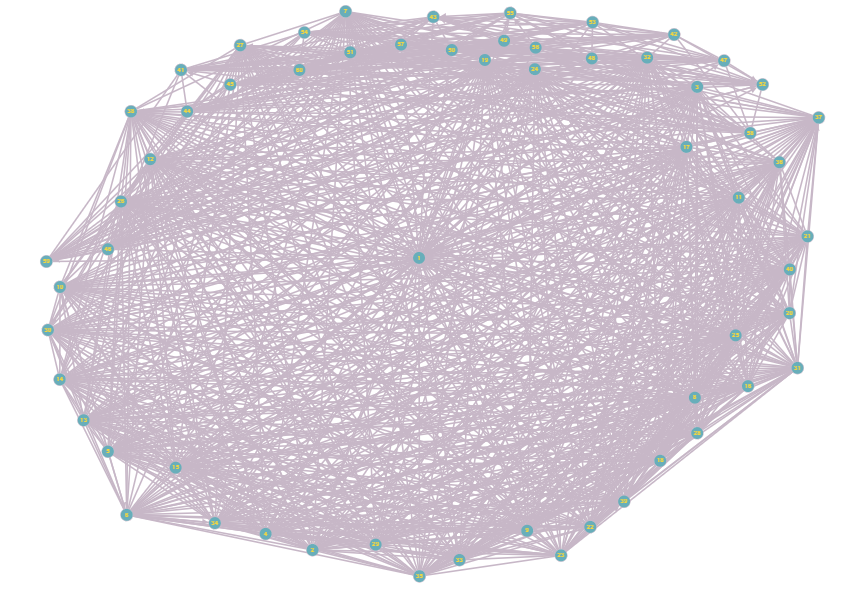
A análise da rede social e das dinâmicas de amizades online no Instagram apresenta uma oportunidade única para explorar as complexidades das interações humanas em um ambiente digital. Esta plataforma, que conta com mais de um bilhão de usuários ativos mensalmente, serve como um vasto campo de estudo para padrões de comportamento, conexões sociais e influência de conteúdo. Cada conta ativa no Instagram pode ser vista como um nó dentro de um extenso grafo, onde as relações de seguir entre os usuários formam as arestas que conectam este intricado tecido social.

O objetivo principal deste estudo é mapear as redes de amizades e analisar as interações de 60 usuários, divididos em 2 grupos sociais diferentes, e analisamos: curtidas, comentários e compartilhamentos. Ao aprofundar na estrutura e dinâmica das redes de amizade online, pretendemos não só mapear as relações existentes, mas também oferecer análises de interações dentro de grupos.

PROJETO:

O Grafo foi modelado no Graph online a partir de uma matriz de adjacência gerada pelo nosso código, o grafo gerado foi o seguinte:

<http://graphonline.ru/pt/?graph=rJjFBFioCkTRjpTF>



Se trata de um grafo extremamente denso, pelo grande número de interações que ocorrem em uma rede social.

A seguir, descreveremos os passos para chegarmos neste grafo:

1º Passo: o webScraping:

Desenvolvemos um código em python, utilizando a biblioteca instaloader, para automatizar a retirada de dados do aplicativo. O código inteiro se encontra em nosso github, na sua versão sem nome de usuários, e senha para podermos logar no aplicativo pelo código. O core do código se encontra na imagem abaixo, e itera sobre os usuários fornecidos em uma lista, que precisam estar abertos ao perfil usado como ponto de partida, e recolha todos os últimos “n” posts dos usuários, sendo “n” também parametrizável, neste caso rastreamos as 200 últimas postagens. Após, o código relaciona o nome dos usuários que realizaram comentários e “deram like” com os nomes que estão na lista de input de usuários. Ao fim, o código joga as informações em um CSV, dando peso 1 para cada like, e peso 2 para cada comentário.

Imagem: “core” do código pyhton.

Texto

Descrição gerada automaticamente

O arquivo de output então, por linha, contém o id do usuário, seguido da soma dos likes e comentários dos outros usuários na ordem que eles se encontram nas linhas do CSV.

Por exemplo:

Usuário\_x 0 1 2

Usuário\_y 10 1

Usuário\_z 2 2 0

Neste exemplo, o usuário\_x obteve (entre comentários e likes) peso 0 de si mesmo (excluímos comentários nos próprios posts), peso 1 do Usuário\_y e 2 do usuário\_z, e assim por diante.

Por fim, utilizamos o notepad++, para trocar cada ocorrência de um nome de usuário no arquivo, por um número que os identificaria, que no caso também corresponderia a seu índice na matrix + 1, e obtivemos um arquivo como o demonstrado na imagem abaixo:

Imagem: como ficaram as primeiras 15 linhas do arquivo resultante:

Texto, Carta

Descrição gerada automaticamente com confiança média

Note que como o usuário 1 foi o ponto de partida, ele possui relações com quase todos os usuários. E por pegarmos 40 pessoas de um grupo e 20 de outro, nas demais linhas, muito raramente ocorrem relações entre os dois grupos.

O próximo passo foi transformar este arquivo, em um arquivo na formatação igual ao grafo.txt pedido no enunciado do projeto.

Fizemos um código em C++ que transforma um arquivo como o descrito acima no formato desejado, o código também se encontra no github, mas por motivos de brevidade não será incluso screenshots deste código. Após transformarmos o txt no formato de entrada desejado, utilizamos o método desenvolvido para o projeto chamado FileToGraph(), que transforma o grafo.txt em uma matriz de adjacência. O código está na imagem abaixo:

Imagem: código que transforma o grafo.txt em matriz de adjacência.  
Texto

Descrição gerada automaticamente

Após este passo, com a matriz de adjacência formada, basta imprimi-la no formato aceito pelo site GraphOnline, onde independente do peso, atribuímos 1 para onde há aresta e zero para onde não existam arestas. Foi desenvolvido o código abaixo:

Texto

Descrição gerada automaticamente

O resultado do print da matriz de adjacência foi o seguinte:

Tela de computador com texto preto sobre fundo azul

Descrição gerada automaticamente com confiança média

Depois apenas utilizamos a função “matriz de adjacência” do graphonline, colamos a matriz acima, e ele gerou o grafo demonstrado no começo do arquivo.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

E Assim geramos nosso grafo estudado.

Objetivos da ODS contemplados:

A análise da rede social Instagram, embora pareça distante dos objetivos tradicionalmente associados a desafios globais como pobreza, saúde e educação, a influência das redes sociais no comportamento humano e nas decisões coletivas pode ser significativa, como detalhado abaixo.

ODS 3: Saúde e Bem-estar

Objetivo: Garantir uma vida saudável e promover o bem-estar para todos, em todas as idades.

Justificativa: A análise do comportamento dos usuários no Instagram pode oferecer insights sobre o impacto das redes sociais na saúde mental e emocional das pessoas. Ao identificar padrões de uso que contribuem para o estresse, a ansiedade e a depressão, o projeto pode propor recomendações para promover um ambiente digital mais saudável, alinhando-se com o objetivo de melhorar o bem-estar geral.

ODS 9: Indústria, Inovação e Infraestrutura

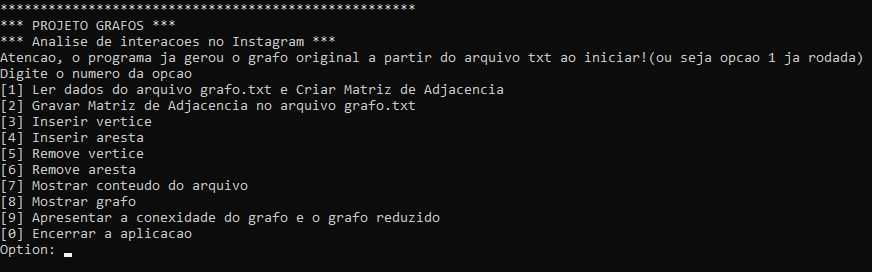
Objetivo: Construir infraestruturas resilientes, promover a industrialização inclusiva e sustentável e fomentar a inovação.

Justificativa: O desenvolvimento de ferramentas analíticas avançadas para estudar o Instagram envolve inovação tecnológica e pode promover melhorias na infraestrutura digital. Este objetivo é atendido pelo projeto ao incentivar o uso de tecnologias inovadoras para análise de dados, contribuindo para a criação de uma indústria digital mais sustentável e inclusiva.

TESTES EXECUÇÃO DO MENU

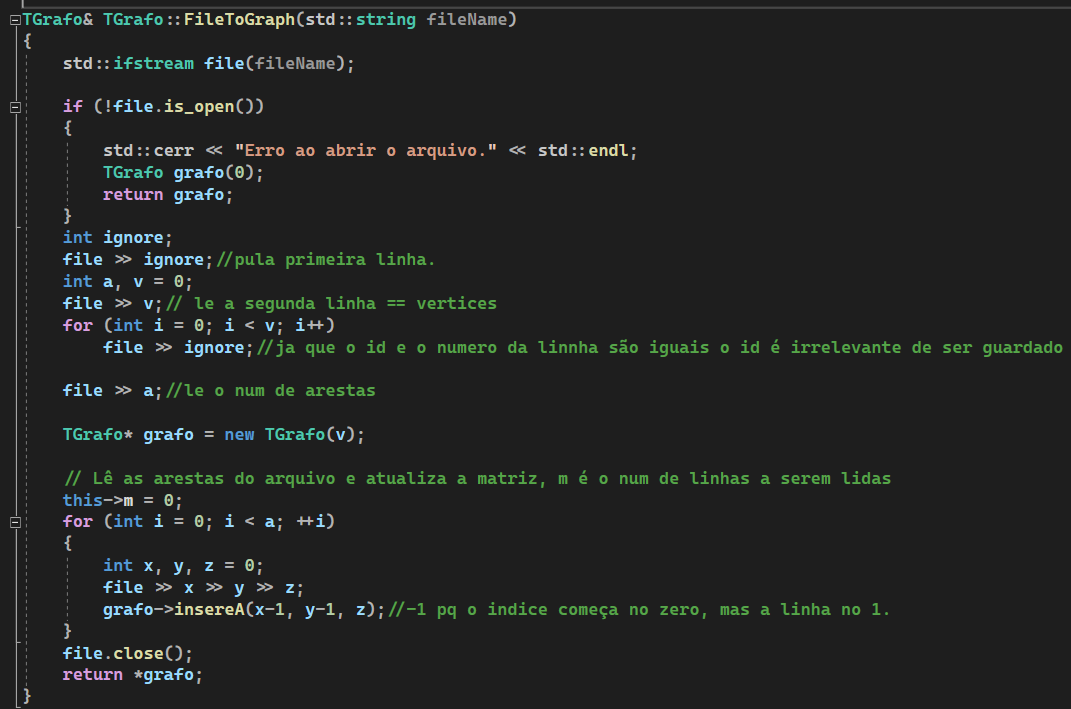
Os testes estão no arquivo Testes.cpp, basta descomentar a linha da main que chama a função “ExecutarTestes”, que os testes serão executados.

Faremos os testes com outros grafos de resultados já conhecidos, pelo grafo do projeto ser muito grande e denso.

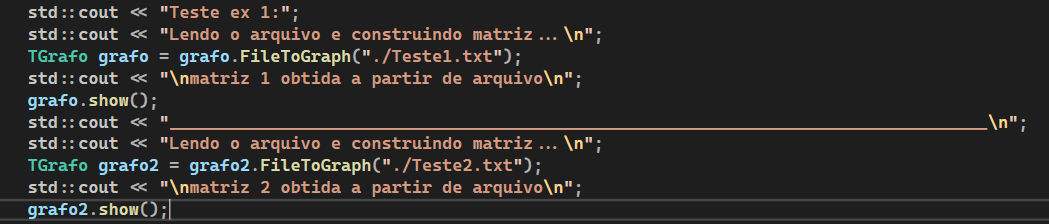
Ao executar o programa, o seguinte menu se abre:  


Por padrão o arquivo com a matriz de adjacência original é carregado lago que o programa é executado, ou seja a opção 1 já é executada.

Segue abaixo o código responsável pela função 1



Já demonstramos que ele funciona anteriormente, porém, faremos 2 testes com arquivos novos:

Testes opção 1:  
Foi executado desta maneira:   


Com estes arquivos txt:

Teste 1:  
Uma imagem contendo Texto

Descrição gerada automaticamente

Teste 2:  
Linha do tempo

Descrição gerada automaticamente com confiança média

Resultado:

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Testes Opção 2

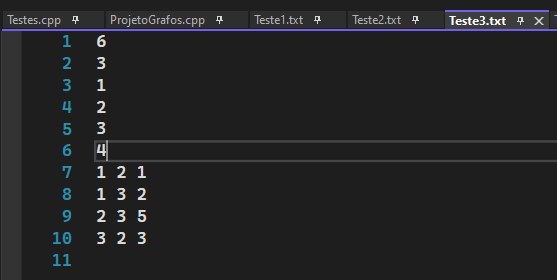
Testes executados desta maneira:

Texto

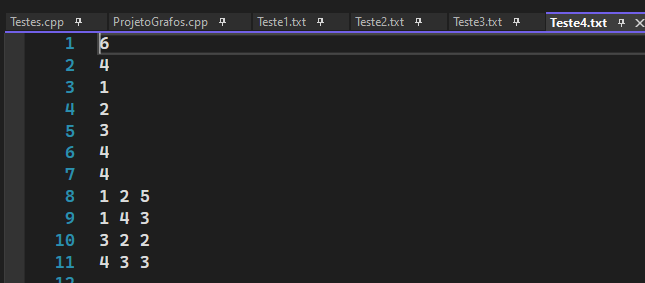
Descrição gerada automaticamente com confiança média

Resultado dos arquivos txt criados:

Teste 3.txt



Teste4.txt



Teste Opção 3:

Código responsável pela opção 3:

Texto

Descrição gerada automaticamente

Foram utilizados os grafos criados pelo teste da opção 1, executados desta maneira:

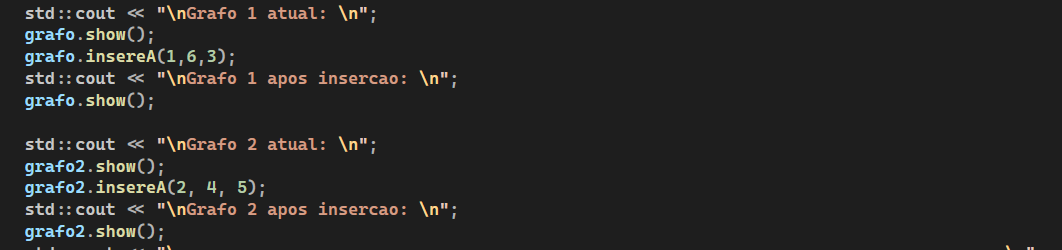
Texto

Descrição gerada automaticamente

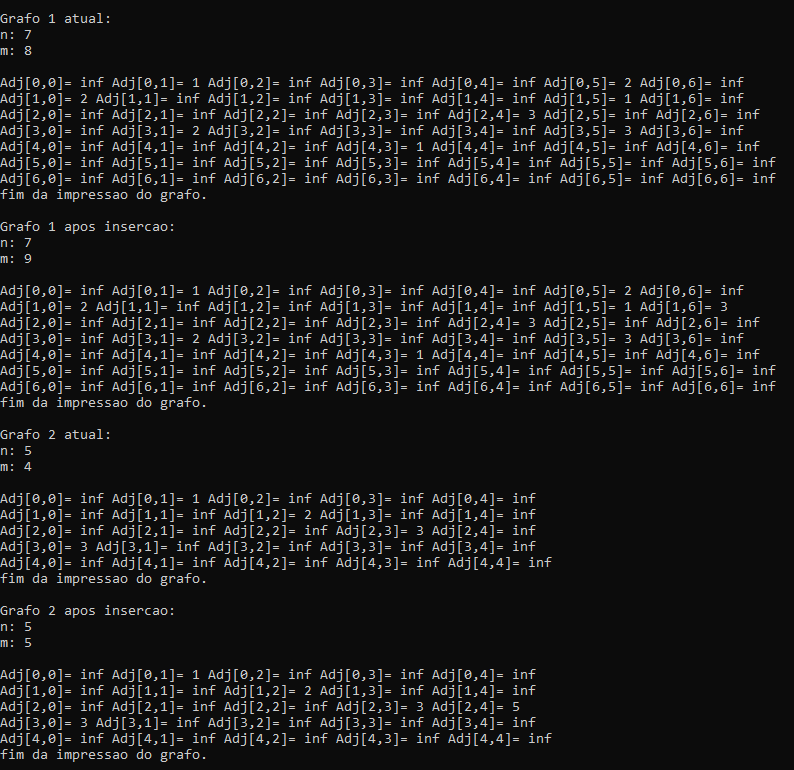
Resultado da execução:  
Uma imagem contendo Texto

Descrição gerada automaticamente

Teste opção 4:

Com base nos grafo aumentado pelo teste da opção 3, vamos inserir arestas da seguinte maneira:  


Resultado:



Teste Opcao 5:

Foi executado desta maneira, com base nos grafos do teste da opção 2:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Este foi o resultado:

Texto

Descrição gerada automaticamente

Teste Opção 6:

Feito em cima dos grafos 1 e 2.

Texto

Descrição gerada automaticamente

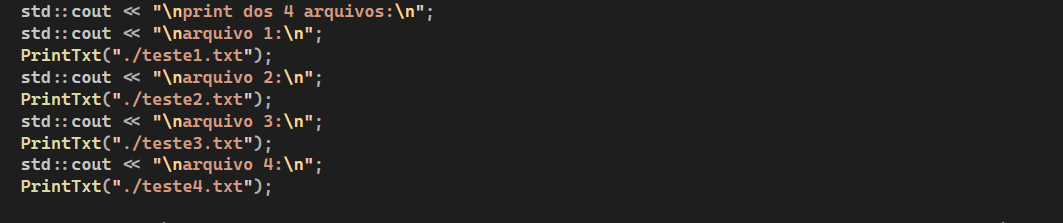
Resultados:

Uma imagem contendo Texto

Descrição gerada automaticamente

Teste Opção 7

Mostraremos o conteúdo dos 4 arquivos de teste: executado desta maneira:



Resultados:

Texto

Descrição gerada automaticamente Texto

Descrição gerada automaticamente

Teste Opção 8:

Foi executado desta maneira:

Texto

Descrição gerada automaticamente

Resultados:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Teste Opção 9:

Executado desta maneira:

Texto

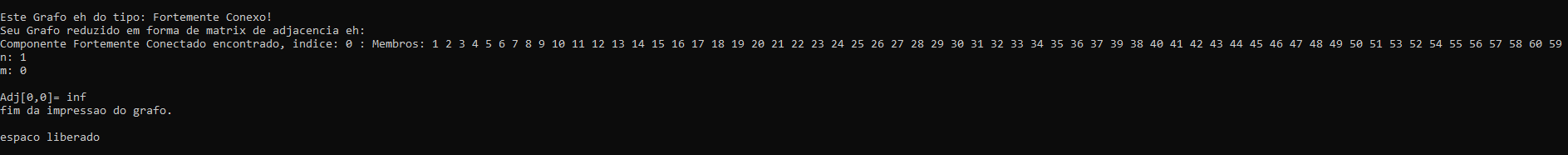
Descrição gerada automaticamente

Resultado:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Agora executando com o nosso grafo encontrado ao analisar instagram:



O interessante é que como todos os contatos partem do primeiro, grafo se torna C3, cujo grafo reduzido é o próprio grafo.

Teste Opção 0:

O programa fecha

Texto

Descrição gerada automaticamente

APENDICE:

LINK GITHUB:

<https://github.com/RenanTagliaferro/Projeto1Grafos>

\*\*FAVOR LEIA O README.