

Trabalho prático: Rede Social Alunos USP

1. Introdução

Nas atuais redes sociais, a amizade é um conceito bastante amplo: na lista de contatos de cada usuário, há amigos próximos, colegas e apenas conhecidos. Com esse embasamento, desenvolveu-se então um aplicativo que contemple facilitar o relacionamento entre amizades verdadeiras.

Pode-se dizer que “amizade verdadeira” é aquela que há muita afinidade entre as pessoas que compõem esse relacionamento. Para isso, fora desenvolvido um sistema capaz de receber dados a partir de um arquivo de dados, que o usuário informa ao iniciar o sistema, e daí em diante fazer interações sociais, tais como: adicionar amigos, buscar sugestão de amizade verdadeira, verificar amigos de baixa afinidade etc.

O sistema desenvolvido consiste em uma rede social de alunos USP, em que os alunos podem se relacionar. O sistema possui um parâmetro de afinidade, que é estabelecido a partir de informações pessoais de cada usuário da aplicação, assim possibilita o cálculo da porcentagem que uma amizade possa ser verdadeira entre os alunos.

2. Implementação

2.1. Usuário

A implementação do usuário no sistema é feita por meio de um Tipo Abstrato de Dado (TAD). Desta forma, garantimos as propriedades definidas pelo conceito de TAD sobre o usuário, que são: reutilização da estrutura (biblioteca) e impedir que o usuário seja alterado no sistema após sua criação.

Cada usuário possui 9 atributos que o identificam e definem a sua personalidade no sistema.

São eles:

- Id: é o índice do usuário na lista de vértices;
- Username: o nome de usuário é o identificador do usuário no sistema. O tamanho máximo é de 19 caracteres;
- Gender: representa o gênero do usuário. O tamanho máximo é de 9 caracteres;
- Age: representa a idade do usuário. É representado por um inteiro;

- Movie: é um atributo que define a personalidade do usuário, ou seja, é considerado no cálculo da afinidade. O tamanho máximo é de 63 caracteres;
- Place: é um atributo que define a personalidade do usuário, ou seja, é considerado no cálculo da afinidade. O tamanho máximo é de 63 caracteres;
- Book: é um atributo que define a personalidade do usuário, ou seja, é considerado no cálculo da afinidade. O tamanho máximo é de 63 caracteres;
- Hobby: é um atributo que define a personalidade do usuário, ou seja, é considerado no cálculo da afinidade. O tamanho máximo é de 63 caracteres;
- Sport: é um atributo que define a personalidade do usuário, ou seja, é considerado no cálculo da afinidade. O tamanho máximo é de 63 caracteres;

Com base nos atributos Movie, Place, Book, Hobby e Sport é feita o cálculo da porcentagem de afinidade. A afinidade é calculada com base na quantidade de atributos que dois usuários possuem em comum pela quantidade de atributos comparados (5).

Ou seja,

$$Afinidade = \frac{qtd. \text{ atributos em comum } (usuário1, usuário2)}{qtd. \text{ total de atributos}} \cdot 100$$

Define-se como amizade verdadeira usuários que possuem afinidade $\geq 60\%$.

Por fim, a implementação do usuário no projeto:

```
struct user_ {
    int id;
    char username[MAX_SIZE_USERNAME];
    char gender[MAX_SIZE_GENDER];
    int age;
    /*!< Opções favoritas */
    char movie[MAX_SIZE_WORD];
    char place[MAX_SIZE_WORD];
    char book[MAX_SIZE_WORD];
    char hobby[MAX_SIZE_WORD];
    char sport[MAX_SIZE_WORD];
};
```

Figura 1 - Estrutura usuário em código

2.2. Grafo

A rede social no sistema é implementada como um grafo ponderado não direcionado. Cada usuário da plataforma é um vértice e quando os usuários se tornam amigos eles passam a ter uma aresta que os conectam.

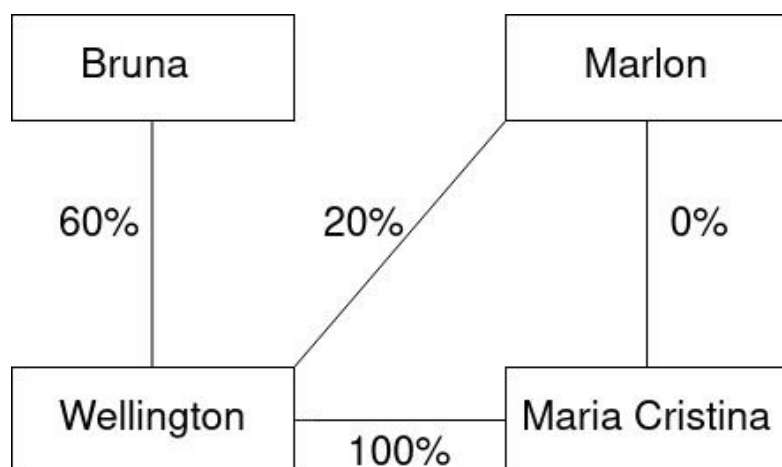


Figura 2 - Representação em grafo de amizades de usuários

Na imagem acima:

- São amigos: Bruna e Wellington, Marlon e Wellington, Marlon e Maria Cristina, Wellington e Maria Cristina
- Não são amigos: Bruna e Maria Cristina, Marlon e Bruna.

Além disso, o peso da aresta indica a porcentagem de afinidade entre os alunos. Por exemplo: Wellington e Maria Cristina possuem uma afinidade de 100% (todos os gostos são iguais) e Marlon e Maria Cristina possuem uma afinidade de 0% (não possuem gostos em comuns).

A implementação do grafo no projeto foi feita utilizando o conceito de lista de adjacências, com a lista que armazena os vértices e a lista de adjacências dinâmicas.



Figura 3 - Representação idealizada do grafo do sistema

A primeira ideia do grafo foi armazenar na aresta o índice de quem a pessoa estava relacionada. Entretanto, ao fazer uma análise em uma pessoa em específico (um vértice da lista), se quisesse obter informações sobre a pessoa com quem está conectado, seria necessário percorrer a lista de usuários novamente (tendo em vista que é uma lista dinâmica) para encontrar as informações da pessoa associada. Por isso, decidimos optar por em cada aresta, armazenar o endereço da pessoa com quem está se relacionando.

As estruturas que compõe o grafo na linguagem C ficaram da seguinte forma:

```
typedef struct node_ NODE;  
struct node_ {  
    USER* user;  
    LIST* adjacency_list;  
    NODE* next;  
};
```

Figura 4 - Estrutura do nó do grafo em código (vértices)

```
struct graph_ {  
    int number_of_vertices;  
    NODE* head;  
    NODE* foot;  
};
```

Figura 5 - Estrutura do grafo em código

```
struct node_ {  
    USER* user;  
    float weight; /*!< Representa a afinidade em porcentagem */  
    NODE* next;  
};
```

Figura 6 - Estrutura do nó inserido na lista de adjacência de um vértice

```
struct list_ {  
    int size;  
    NODE* head;  
    NODE* foot;  
};
```

Figura 7 - Estrutura da lista de adjacência

Por fim, a escolha de uma lista de adjacência ao invés de uma matriz de adjacência foi devido aos seguintes fatores:

- Não sabia-se a priori quantos usuários a rede social possuiria, e a matriz de adjacência ocupa $O(|V|^2)$ de espaço, enquanto a lista de adjacência ocupa apenas $O(|V|+|A|)$;
- A lista de adjacência possui complexidade $O(|V|+|A|)$ para processar todo o grafo, enquanto a matriz de adjacência possui complexidade $O(|V|^2)$. Desta forma, a lista de adjacência é melhor para o processo "identificar as pessoas com perfil 'extrovertido, ..." pois neste caso é preciso analisar todo o grafo;
- Para o tópico "detectar pessoas na lista de contatos..." a lista de adjacências também é uma melhor escolha, pois ela apenas olha a lista de adjacência do grafo $O(\text{grau do vértice})$, enquanto na matriz de adjacência deveria ser olhado todos os $|V|$ vértices na matriz relacionados com o vértice, ou seja $O(|V|)$.

3. Execução

3.1. Arquivo com dados dos usuários

Para executar a aplicação, é necessário um arquivo com a definição do perfil dos usuários que irão fazer parte da rede social.

```
nome do usuário: beltrano
gênero: masculino
idade: 19
filme predileto: vingadores
local predileto: bar do zé
livro: 1984
hobby: cinema
esporte: corrida

nome do usuário:ciclano
gênero: feminino
idade: 21
filme predileto: prometheus
local predileto: praça xv
livro: feliz ano velho
hobby: leitura
esporte: ping pong

nome do usuário:magrini
gênero: feminino
idade: 19
filme predileto: prometheus
local predileto: bar do zé
livro: guia do mochileiro
hobby: correr
esporte: natacao
```

Figura 8 - Exemplo de como deve estar o arquivo de dados para entrada no sistema

Obrigatoriamente, cada usuário deve conter 8 atributos: nome de usuário, gênero, idade, filme predileto, local predileto, livro, hobby, esporte; e eles devem estar nessa ordem no arquivo.

O nome do atributo não é obrigatório, mas os dois pontos (:) em cada linha é necessário. Logo, o arquivo poderia ser representado sem o nome dos atributos.

```
: beltrano  
: masculino  
: 19  
: vingadores  
: bar do zé  
: 1984  
: cinema  
: corrida
```

Figura 9 - Exemplo de atributos de um usuário para entrada no sistema

É obrigatório a separação entre cada usuário por um espaço, além disso, a última linha do arquivo deve conter apenas um espaço após a definição do último usuário.

O arquivo de dados deve conter pelo menos um usuário definido.

3.2. Aplicação sendo executada

Ao executar a aplicação, será solicitado o nome do arquivo que contém os dados dos usuários.

```
Digite o nome do arquivo que contém os perfis  
dos usuários: users
```

Figura 10 - Primeira entrada no sistema (arquivo de dados)

Em seguida, será apresentado o menu que contém as funcionalidades do sistema.

```
-----  
1 - Enviar convite  
2 - Sugestão de 'amigos de verdade'  
3 - Detectar baixa afinidade  
4 - Identificar perfil dos usuários  
5 - Imprimir rede social  
6 - Imprimir usuários da rede social  
7 - Calcule a distância entre dois usuários na rede social  
0 - Sair  
-----
```

Figura 11 - Menu de opções do sistema

As funcionalidades do sistema são:

1. Cria um relacionamento entre dois usuários, ou seja, é criada uma aresta entre dois vértices.

É apresentado os nomes de usuários de todos os participantes da rede social. Como entrada deve ser fornecido dois nomes de usuários (username) presentes no sistema. O primeiro nome de usuário fornecido será quem estará fazendo a solicitação de amizade.

A afinidade entre os dois usuários será calculada e será apresentada uma mensagem para o segundo usuário informando a afinidade e sugerindo se ele deve ou não aceitar o convite de amizade.

```
1
Nome do usuário: beltrano
Nome do usuário: ciclano
Nome do usuário: magrini
Nome do usuário: marlon
Nome do usuário: feliz
Nome do usuário: wellington
Amizade entre os usuários: beltrano wellington
A chance dessa amizade ser verdadeira é de 0.00% e por isso não r
ecomendamos vocês se tornarem amigos. Você pode verificar as sug
estões de possíveis novos 'amigos de verdade'.
O usuário wellington deseja aceitar a solicitação? (1 - Sim, 0 -
Não): █
```

Figura 12 - Mensagem apresentada quando a amizade solicitada é de baixa afinidade

```
1
Nome do usuário: beltrano
Nome do usuário: ciclano
Nome do usuário: magrini
Nome do usuário: marlon
Nome do usuário: feliz
Nome do usuário: wellington
Amizade entre os usuários: feliz wellington
A chance dessa amizade ser verdadeira é de 100.00% e por isso rec
omendamos vocês se tornarem amigos e se conhecerem melhor!
O usuário wellington deseja aceitar a solicitação? (1 - Sim, 0 -
Não): █
```

Figura 13 - Mensagem apresentada quando a amizade solicitada é de alta afinidade

O usuário pode escolher aceitar a solicitação (1) ou não (0).

2. É sugerido para todos os usuários da rede social amizades verdadeiras (se existir) apresentando a afinidade.


```
2
Sugestão de amizade verdadeira para o usuário beltrano:
  Nenhuma sugestão de amizade verdadeira.

Sugestão de amizade verdadeira para o usuário ciclano:
  marlon, afinidade = 60.00%

Sugestão de amizade verdadeira para o usuário magrini:
  marlon, afinidade = 60.00%

Sugestão de amizade verdadeira para o usuário marlon:
  ciclano, afinidade = 60.00%
  magrini, afinidade = 60.00%

Sugestão de amizade verdadeira para o usuário feliz:
  wellington, afinidade = 100.00%

Sugestão de amizade verdadeira para o usuário wellington:
  feliz, afinidade = 100.00%
```

Figura 14 - Tela apresentada na funcionalidade “Sugestão de ‘amigos de verdade’”

3. É detectado na lista de contato de todos os usuários quais amizades possuem baixa afinidade.

```
3
Amizades de baixa afinidade do usuário beltrano:
  ciclano, afinidade: 40.00%

Amizades de baixa afinidade do usuário ciclano:
  beltrano, afinidade: 40.00%

Amizades de baixa afinidade do usuário magrini:

Amizades de baixa afinidade do usuário marlon:
  wellington, afinidade: 0.00%

Amizades de baixa afinidade do usuário feliz:

Amizades de baixa afinidade do usuário wellington:
  marlon, afinidade: 0.00%
```

Figura 15 - Tela apresentada na funcionalidade “Detectar baixa afinidade”

4. É atribuído um perfil para cada usuário. Caso o usuário tenha, em relação a todas as suas amizades, valor maior ou igual a 50% de amigos verdadeiros, ele é considerado “*extrovertido*”, caso contrário (i.e., com o valor menor que 50%), é considerado “*introvertido*”.


```
4
0 usuário beltrano é introvertido.
0 usuário ciclano é extroverdio.
0 usuário magrini é introvertido.
0 usuário marlon é extroverdio.
0 usuário feliz é introvertido.
0 usuário wellington é introvertido.
```

Figura 16 - Tela apresentada na funcionalidade “Identificar perfil dos usuários”

5. Imprime o nome de cada usuário da rede social e a sua lista de amigos.

```
5
Lista de amizade do usuário beltrano:
  Usuário: ciclano com afinidade de 40.00%

Lista de amizade do usuário ciclano:
  Usuário: beltrano com afinidade de 40.00%
  Usuário: marlon com afinidade de 60.00%

Lista de amizade do usuário magrini:
  Nenhuma amizade.

Lista de amizade do usuário marlon:
  Usuário: wellington com afinidade de 0.00%
  Usuário: ciclano com afinidade de 60.00%

Lista de amizade do usuário feliz:
  Nenhuma amizade.

Lista de amizade do usuário wellington:
  Usuário: marlon com afinidade de 0.00%
```

Figura 17 - Tela apresentada na funcionalidade “Imprimir rede social”

6. Imprime o nome e os atributos de cada usuário presente na rede social.

```
6
beltrano
Nome de usuário: beltrano
Gênero: masculino
Idade: 19
Filme predileto: vingadores
Local predileto: bar do zé
Livro: 1984
Hobby: cinema
Esporte: corrida

ciclano
Nome de usuário: ciclano
Gênero: feminino
Idade: 21
Filme predileto: vingadores
Local predileto: bar do zé
Livro: feliz ano velho
Hobby: leitura
Esporte: futebol

magrini
Nome de usuário: magrini
Gênero: feminino
Idade: 19
Filme predileto: prometheus
Local predileto: bar do zé
Livro: guia do mochileiro
Hobby: programar
Esporte: natacao

marlon
Nome de usuário: marlon
Gênero: masculino
Idade: 19
Filme predileto: vingadores
Local predileto: bar do zé
```

Figura 18 - Tela apresentada na funcionalidade “Imprimir usuários da rede social”

7. Calcula a distância entre dois usuários na rede social. Como entrada deve ser fornecido dois nomes de usuários (username) presentes no sistema. É informado a distância entre os usuários ou se não é possível chegar a partir de um vértice no outro.

```
7
Segundo a teoria dos seis graus de separação, no mundo, são necessários no máximo seis laços de amizade para que duas pessoas quaisquer estejam ligadas. Vamos verificar se essa teoria também se aplica a nossa rede social?
Digite o nome de dois usuários para verificar a distância entre eles:
beltrano feliz
A distância entre o usuário beltrano e o usuário feliz é de 4.
```

Figura 19 - Tela apresentada quando há uma distância entre dois usuários

```
7
Segundo a teoria dos seis graus de separação, no mundo, são necessári
os no máximo seis laços de amizade para que duas pessoas quaisquer es
tejam ligadas. Vamos verificar se essa teoria também se aplica a noss
a rede social?
Digite o nome de dois usuários para verificar a distância entre eles:
magrini marlon
Os usuários são amigos!
```

Figura 20 - Tela apresentada quando os usuários já são amigos

```
7
Segundo a teoria dos seis graus de separação, no mundo, são necessári
os no máximo seis laços de amizade para que duas pessoas quaisquer es
tejam ligadas. Vamos verificar se essa teoria também se aplica a noss
a rede social?
Digite o nome de dois usuários para verificar a distância entre eles:
ciclano wellington
O usuário ciclano não possui um laço com o usuário wellington.
```

Figura 21 - Tela apresentada caso os usuários não tenham nenhum laço

0. Finaliza a execução do programa.

3.3. Código

A aplicação completa está disponível no GitHub e pode ser acessada através do link: <https://github.com/magrinibruna/social-network>.

4. Conclusão

Neste relatório foi descrito a elaboração e implementação do sistema “Rede Social Alunos USP”, o qual é responsável por manter relações de amizades possivelmente verdadeiras.

Com esse novo modo de lidar com os relacionamentos dentro da universidade, espera-se que possa facilitar na comunicação com amigos de alta afinidade e também nascer novas amizades com base nas funcionalidades do sistema, que facilita encontrar pessoas que possuem interesses em comum. O sistema é capaz de sugerir opções de pessoas que frequentam lugares, assistem filmes e têm hobbies iguais entre outros interesses.

Destarte, portanto, o projeto alcançou os objetivos inicialmente traçados. Tendo em vista que o sistema fornece todas as funcionalidades básicas para o funcionamento da rede social, além de ser intuitivo e de fácil compreensão ao usuário. Ainda assim, pode ser um projeto contínuo, e receber implementações de funcionalidades, para assim trazer ainda mais conforto ao usuário.