

FELIPE AMORIM E MURILO VIDAL

**SISTEMA DE INTERAÇÕES SOCIAIS BASEADAS EM SIMILARIDADE**

*(versão pré-defesa, compilada em 9 de junho de 2019)*

Trabalho de conclusão de curso apresentado como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. Alexander Kutzke.

CURITIBA PR  
2019

# Resumo

O resumo deve conter no máximo 500 palavras, devendo ser justificado na largura da página e escrito em um único parágrafo<sup>1</sup> com um afastamento de 1,27 cm na primeira linha. O espaçamento entre linhas deve ser de 1,5 linhas. O resumo deve ser informativo, ou seja, é a condensação do conteúdo e expõe finalidades, metodologia, resultados e conclusões.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

**Palavras-chave:** palavra-chave 1, palavra-chave 2, palavra-chave 3.

---

<sup>1</sup>E também não deve ter notas de rodapé; em outras palavras, não siga este exemplo... ;-)

# Abstract

The abstract should be the English translation of the “resumo”, no more, no less.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

**Keywords:** keyword 1, keyword 2, keyword 3.

# Lista de Figuras

2.1	Um grafo com 5 nós e 6 arestas. . . . .	16
2.2	Duas representações de um grafo não direcionado. (a) Um grafo $G$ com cinco nós e sete arestas. (b) A representação de uma lista de adjacência de $G$ . (c) A representação de uma matriz de adjacência. Fonte: CORMEN et al. (2009), tradução dos autores. . . . .	17
2.3	Duas representações de um grafo direcionado. (a) Um grafo $G$ com seis nós e oito arestas. (b) A representação de uma lista de adjacência de $G$ . (c) A representação de uma matriz de adjacência. Fonte: CORMEN et al. (2009), tradução dos autores.	17
3.1	Tela de criação de perguntas do MVP. Fonte: os autores. . . . .	19
3.2	Tela de visualização de perguntas do MVP. Nesta tela é possível escolher uma pergunta para ser respondida. Fonte: os autores. . . . .	20
3.3	Tela de visualização de respostas recebidas do MVP. Nesta tela é possível marcar perguntas favoritas. Fonte: os autores. . . . .	20
3.4	Visualização do grafo que representa as ligações entre usuários da rede social no MVP. Fonte: os autores. . . . .	21
3.5	Arquitetura do software. Fonte: os autores. . . . .	22
3.6	Diagrama de caso de uso nível 1. Fonte: os autores. . . . .	23
3.7	Diagrama de caso de uso nível 2. Fonte: os autores. . . . .	24
3.8	Diagrama de caso de uso UC001 - Visualizar Questão. Fonte: os autores. . . . .	24
3.9	Diagrama de caso de uso UC002 - Responder Questão. Fonte: os autores. . . . .	25
3.10	Diagrama de caso de uso UC003 - Visualizar Respostas. Fonte: os autores. . . . .	25
3.11	Diagrama de caso de uso UC004 - Criar Questão. Fonte: os autores. . . . .	26
3.12	Diagrama de caso de uso UC005 - Visualizar Combinações. Fonte: os autores. . . . .	26
3.13	Diagrama de classes. Fonte: os autores. . . . .	27
3.14	Evolução de $k$ se todas as perguntas postadas por $A$ forem respondidas por $B$ . Fonte: os autores. . . . .	29
3.15	Evolução de $k$ se $A$ postou 50 perguntas e apenas 25 respostas de $B$ foram curtidas. Fonte: os autores. . . . .	30
3.16	Evolução de $k$ em função das respostas curtidas se $A$ e $B$ têm poucas perguntas postadas. Ex.: menos que 10 perguntas. Fonte: os autores. . . . .	31

3.17	Evolução de $k$ em função das respostas curtidas se $A$ e $B$ têm muitas perguntas postadas. Ex.: mais que 50 perguntas. Fonte: os autores. . . . .	31
------	---	----

# Lista de Tabelas

3.1	Formulário inicial . . . . .	28
-----	------------------------------	----

# Lista de Abreviações

TADS	Tecnologia em Análise e Desenvolvimento de Sistemas
SDK	<i>Software development kit</i>
VM	<i>Virtual Machine</i>
MVP	<i>Minimum viable product</i>
UFPR	Universidade Federal do Paraná

# Lista de Símbolos

$\psi$

psi, vigésima terceira letra do alfabeto grego



# Sumário

<b>1</b>	<b>Introdução . . . . .</b>	<b>10</b>
<b>2</b>	<b>Fundamentação Teórica. . . . .</b>	<b>13</b>
2.1	Redes Sociais . . . . .	13
2.2	Algoritmos de Sugestão. . . . .	14
2.3	Grafo . . . . .	14
<b>3</b>	<b>Metodologia . . . . .</b>	<b>19</b>
3.1	Método de desenvolvimento . . . . .	19
3.2	Arquitetura . . . . .	22
3.3	Tecnologia aplicada . . . . .	23
3.4	Análise do sistema . . . . .	23
3.5	Cálculo de do peso da aresta . . . . .	28
<b>4</b>	<b>Produto . . . . .</b>	<b>33</b>
4.1	Perguntas e respostas . . . . .	33
4.2	Mensageiro instantâneo . . . . .	33
<b>5</b>	<b>Conclusão . . . . .</b>	<b>34</b>
	<b>Referências . . . . .</b>	<b>35</b>

# 1 Introdução

A atividade humana em sociedade demanda interação entre os indivíduos. As interações e o conjunto de participantes que dividem conhecimento e interesses formam as redes sociais, (MARTELETO, 2001). Essa interação social tem, de mais a mais, acontecido no ambiente virtual. No Brasil, estima-se que existam 130 milhões de usuários de redes sociais na internet, de acordo com KEMP (2018). Os brasileiros dispõem, em média, 3h39min por dia em redes sociais (KEMP, 2018).

Dentro desta dinâmica de interações cada indivíduo busca relacionamentos baseados em seus interesses. Por isso existem muitas redes sociais especializadas em relacionamentos profissionais, artísticos, casuais e também íntimos, como LinkedIn, Pinterest, Facebook e Tinder, respectivamente.

As interações dentro das redes sociais podem ser indesejadas e, muitas vezes, desagradáveis, pois, assim como nos relacionamentos presenciais, as pessoas discordam umas das outras, não tem os mesmos interesses ou simplesmente não têm afinidade alguma. Portanto, sugerir contatos ou conteúdos para os usuários é uma maneira de manter as redes interessantes e agradáveis.

Este trabalho busca desenvolver uma rede social cujas interações iniciais são trocas de perguntas e respostas de maneira anônima entre os usuários. É tomado em conta o volume das perguntas e respostas trocadas e quantas delas foram apontadas como agradáveis e, a partir disso, calcula-se um nível de afinidade entre os atores. No momento que se considera que duas pessoas são suficientemente ligadas por seus interesses e interações, o software põe em contato essas pessoas com a habilitação de um mensageiro instantâneo.

Um grafo, não direcionado e com peso nas arestas é usado como estrutura de dados para a representação dos usuários e a similaridade entre eles. O cálculo do peso é a variável que justifica o caráter científico deste trabalho. O grafo é uma estrutura de dados na qual as unidades elementares, que são chamadas de nós, estão ligadas entre si por arestas, cujo valor e orientação representam a conexão entre os elementos.

Neste trabalho, os usuários são representados pelos nós do grafo e sua similaridade é representada pelas arestas. Essa estrutura pode ter uma vantagem sobre as outras estruturas de dados pois o cálculo uma aresta para cada vizinho para representar de similaridade tem pouca carga de processamento.

A parte funcional da rede social aqui proposta, diferencia-se das redes sociais direcionadas à criação de novos relacionamentos, como o Tinder e o Happn, na qual o interesse entre os usuários é despertado, sobretudo, pela aparência, uma vez que a foto é obrigatória. O sistema aqui proposto dispensa o uso de fotos, e se posta como uma alternativa à avaliação puramente pela aparência dos participantes, como acontece nos sistemas citados inicialmente.

A proposta desta rede social é criar novos relacionamentos entre os usuários a partir da afinidade entre eles, permitindo que usuários se conheçam a partir de interesses comuns.

A expressão anônima dos interesses do usuário pelo uso das perguntas possibilita uma interação franca e direcionada à interação. Perguntas elaboradas sem critério ou sensibilidade terão poucas respostas e o usuário não será bem sucedido na busca por um novo contato. Como toda rede social, o histórico das interações pode gerar dados importantes para pesquisa sociológica. A mecânica singular e simplificada das interações geradas com o uso software ??? pode dar luz à aspectos diferenciados da natureza humana.

A eficiência do cálculo de vizinho mais próximo a partir de grafos já foi citada por PATERSON e YAO (1992) nas aplicações de geometria computacional e na simulação física. MISHRA (2019), propõe um algoritmo para calcular o vizinho mais próximo em um grafo sem visitar todos os nós, corroborando a versatilidade dessa estrutura de dados quando o volume de dados é consideravelmente grande.

Classificação de dados e algoritmos de sugestão são amplamente usados em plataformas de e-commerce, notícias e em redes sociais para aproximar o conteúdo disponível em uma página ao interesse do usuário. A partir da experiência frutífera de um usuário em um site de e-commerce, os algoritmos sugerem produtos que alguém vai gostar porque outra pessoa com experiência ou interesses similares efetuou uma compra.

SHAMS e HARATIZADEH (2016) afirmam que existem métodos de classificação orientados a avaliação ou orientados a classificação. Este trabalho busca um método eficiente de sugerir contatos dentro da rede social. É uma rede social baseada em similaridade entre as pessoas. Essa similaridade entre usuários é obtida a partir de um questionário inicial e da troca de perguntas e respostas anonimamente entre os usuários, portanto, um método orientado à classificação, segundo a definição de SHAMS e HARATIZADEH (2016).

O software foi desenvolvido utilizando o Modelo em Cascata com a produção de um produto mínimo viável durante o levantamento dos requisitos. O desenvolvimento foi planejado levando em conta as habilidades dos desenvolvedores e o tempo necessário para adquirir competência nas ferramentas e linguagens de programação envolvidas. O *front-end* foi desenvolvido primeiro por representar o maior volume de trabalho comparativamente às competências dos desenvolvedores.

Foi utilizado o *framework* Ionic para desenvolvimento de aplicativo. O apelo do Ionic reside no fato de ser uma SDK de código aberto com capacidade multiplataforma. O software escrito com base no *framework* Ionic tem capacidade de ser utilizado em vários navegadores, no sistema operacional *Android* e no *IOS*.

Para a avaliação do cálculo da aresta do grafo, foram carregados valores fictícios e a função foi testada analiticamente. Simulações de volume de perguntas e respostas são comparadas ao número de arestas do grafo e os seus respectivos pesos. A média do peso das arestas dentro da simulação foi considerado para obter um valor mínimo adequado para criar as conexões entre os usuários.

Este trabalho foi estruturado em três parte:.

- **Fundamentação Teórica:** Nesta seção, visitamos os principais autores e documentos relacionados à redes sociais, algoritmos de classificação e de sugestão e grafos.
- **Metodologia de Desenvolvimento:** Nesta seção, é exposto o método de desenvolvimento do software e a análise feita para otimizar a função que calcula a aresta do grafo que representa as conexões entre os usuários.
- **Apresentação dos resultados:** Telas e funcionalidades do produto final.

## 2 Fundamentação Teórica

### 2.1 Redes Sociais

- Histórico de redes sociais
- Tipos de redes sociais
- Redes parecidas com nosso trabalho
- Análises de tempo despendido em redes sociais.

Desde a invenção da internet em 1991, mais e mais aplicações vem sendo criadas para facilitar e estimular os relacionamentos virtuais. As redes sociais virtuais tiveram início em 1997 com o site *Six Degrees*, (TERRELL, 2017). *Six Degrees* é nomeado em referência ao estudo de MILGRAM (1967), que teorizava que são necessários seis laços de amizade para que quaisquer pessoas estejam conectadas. *Six Degrees* teve seu fim em 2001, mas iniciou o processo de popularização das redes sociais e é considerado a primeira rede social pois permitia que as pessoas criassem seus perfis individuais e adicionar outros contatos à sua rede. Teve 3,5 milhões de usuários no seu auge.

Entre as precursoras está também a rede especializada em relacionamentos profissionais, *LinkedIn*. Foi criada em no final de 2002 e tem, desde então, seu objetivo principal em criar conexões entre profissionais, estudantes e corporações. O *LinkedIn* é, ainda hoje, a rede mais popular neste nicho com mais de 500 milhões de usuários.

A rede mais popular atualmente, segundo LUA (2019), é o Facebook com 2,23 milhões de usuários ativos mensalmente. O Facebook foi criado em 2004 como uma rede específica para estudantes da Universidade de Harvard. Em 2006 foi aberta ao público e em 2008 já era a rede social mais visitada do mundo.

Há também redes especializadas em outros tipos de conexão ou outras maneiras de publicar conteúdo. O YouTube é uma rede social especializada na divulgação de vídeos criados pelos usuários. O Twitter, distingue-se das outras redes por iniciar suas atividades permitindo apenas a publicação de textos com, no máximo, 140 caracteres. Este limite foi dobrado posteriormente, mas o foco ainda mantém-se em pequenas postagens.

Por fim, há as redes especializadas em relacionamentos românticos. O Tinder, o Happn e o OKCupid são exemplos de redes sociais direcionadas à criação de relacionamentos íntimos

entre os usuários. O sistema desenvolvido neste trabalho enquadra-se nesta categoria pois tem o objetivo de colocar usuários em contato com pessoas desconhecidas que têm potencial para formarem amizades ou até mesmo casais.

No Brasil, 66% da população tem acesso à internet segundo KEMP (2018). Dentre os usuários da internet 93% são ativos em alguma rede social. Segundo KEMP (2018), o brasileiro despende, em média, 3h39min por dia em alguma rede social. Este tempo é passado em contato com informações e pessoas de várias culturas diferentes. Muitas vezes as interações despertam sentimentos indesejados e a informação divulgada não é completamente conexa à realidade.

Por essa razão, mais e mais redes sociais oferecem a oportunidade, muitas vezes compulsoriamente, do usuário ser exposto somente à pessoas e conteúdos que tenham afinidade com seu perfil. Na busca por uma experiência agradável, as sugestões de conteúdo e contatos vão ao encontro da necessidade do ser humano de se socializar. E participar de uma rede social, seja ela física ou virtual, faz parte das necessidades básicas apontadas por MASLOW (1943), ao definir uma hierarquia para as necessidades básicas do ser humano.

De uma maneira ou de outra, todas as redes sociais tem seu foco no conteúdo gerado pelos usuários. Porém, grande parte da monetização empregada refere-se à divulgação de peças publicitárias pagas por empresas particulares e públicas. Dentro deste escopo, são utilizados algoritmos que tomam em conta os interesses dos usuários para fazer sugestões de conteúdo publicitário. Há vários métodos para relacionar os dados produzidos pelos usuários para gerar sugestões, sejam elas de conteúdo, publicidade ou novos contatos.

## 2.2 Algoritmos de Sugestão

- Como sugerir novos contatos.
- Como as outras redes fazem isso.

Os algoritmos de sugestão são utilizados para tomar em conta os dados disponíveis dos usuários, sejam eles dados pessoais ou dados gerados a partir da utilização do sistema, e gerar sugestões de conteúdos, publicidade, notícias ou contatos para cada usuário. O valor do bom algoritmo de sugestão reside em sugerir oportunamente conteúdos pertinentes que podem ser apreciados pelos usuários.

## 2.3 Grafo

- O que é grafo - *Fonte básica de definição de grafo LIVRO??*
- Quantos tipos de grafo existem.
- Aplicações comuns de grafo

- Algoritmos que usam grafos

O conceito matemático de grafo parte de abstrações de situações da vida real onde objetos ou pessoas representadas por pontos e suas conexões e interações são representadas por linhas BONDY e MURTY (1976). Numa rede social, as pessoas são esses pontos, que convencionalmente são chamados de nós, e suas relações com outras pessoas são representadas por linhas, chamadas de arestas. Um grafo é uma tripla ordenada  $G = (N(G), A(G), \psi_G)$ , que consiste de um conjunto não vazio  $N(G)$  de nós, um conjunto  $A(G)$ , diferente de  $N(G)$ , de arestas, e uma *função de incidência*  $\psi_G$  que associa a cada nó de  $G$  um par não ordenado de nós de  $G$ . Se  $e$  é uma aresta de  $G$  e  $u$  e  $v$  são arestas de tal modo que  $\psi_G(e) = uv$ , então  $e$  é dito que *une*  $u$  e  $v$  BONDY e MURTY (1976).

*Exemplo*

$$G = (V(G), E(G), \psi_G) \quad (2.1)$$

onde

$$V(G) = v_1, v_2, v_3, v_4, v_5 \quad (2.2)$$

e  $\psi_G$  é definido por

$$\begin{aligned} \psi_G(e_1) &= v_1v_2, \psi_G(e_2) = v_2v_3, \psi_G(e_3) = v_3v_3, \psi_G(e_4) = v_2v_4 \\ \psi_G(e_5) &= v_2v_4, \psi_G(e_6) = v_4v_5, \psi_G(e_8) = v_2v_5, \psi_G(e_4) = v_2v_4 \end{aligned} \quad (2.3)$$

Grafos têm este nome pois podem ser representadas graficamente, segundo (BONDY e MURTY, 1976). A figura 2.1, é um diagrama que representa um grafo com 5 nós e 6 arestas. Nesta figura, os nós têm identificação - letras - e as arestas tem um *peso*. O peso das arestas pode ser usado para representar a força da ligação entre os nós ou até mesmo a distância entre os nós. A convenção tomada neste caso depende do contexto e da aplicação do grafo.

Quando a relação entre dois nós é simétrica, ou seja, a relação entre o nó  $A$  e o nó  $B$  é idêntica, diz-se que o grafo é *não orientado*. Quando a relação entre dois nós é assimétrica, portanto,  $A$  tem uma relação com  $B$ , mas essa relação não equivale a  $B$  para  $A$ , tem-se um *dígrafo* ou um *grafo orientado*.

Este trabalho utiliza o grafo não orientado pois este bem representa as relações de amizade recíprocas. Tal decisão é tomada pois, quando da sugestão de um novo contato, ambos vão ser apresentados reciprocamente, estabelecendo, portanto, uma relação mútua de interação.

Diversas áreas da ciência, desde a biologia à linguística, utilizam grafos para representar dados. A representação de objetos e suas relações é uma ferramenta frequente em várias pesquisas.

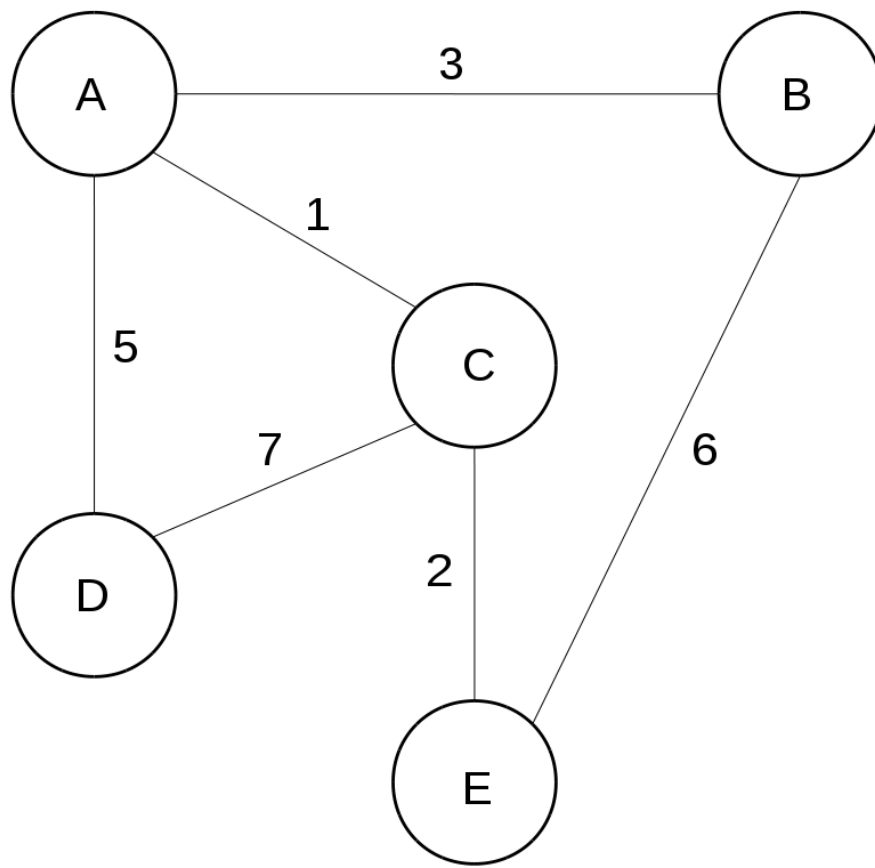


Figura 2.1: Um grafo com 5 nós e 6 arestas.

Na ciência da computação, o grafo é a uma estrutura de dados amplamente frequente e existem centenas de problemas computacionais definidos com o seu uso CORMEN et al. (2009). Um caso tradicional com solução utilizando grafos é a do caixeiro viajante. Neste problema, os nós do grafo são as localidades e as arestas têm o peso equivalente à distância entre as cidades. (DIJKSTRA, 1959), propôs o primeiro algoritmo para encontrar o caminho mais curto entre dois nós de um grafo. O algoritmo encontra um caminho curto, mas não o ideal. Vários trabalhos são realizados ainda hoje na busca pelo caminho mais curto ideal em um grafo. Muitos desses trabalhos são baseados no algoritmo de Dijkstra. Esse algoritmo será apresentado ainda neste capítulo.

Como estrutura de dados computacionais, existem várias modelagens conhecidas para representação de um grafo. (CORMEN et al., 2009), cita duas formas fundamentais:

- A lista de adjacência
- A matriz de adjacência

A figura 2.2, ilustra um grafo  $G$  como um diagrama, como uma lista e como uma matriz. A implementação computacional das estruturas lista e matriz

The adjacency-list representation of a graph  $G = (V, E)$  consists of an array  $Adj$  of  $|V|$  lists, one for each vertex in  $V$ . For each  $u \in V$ , the adjacency list  $Adj[u]$  contains all the vertices  $v$



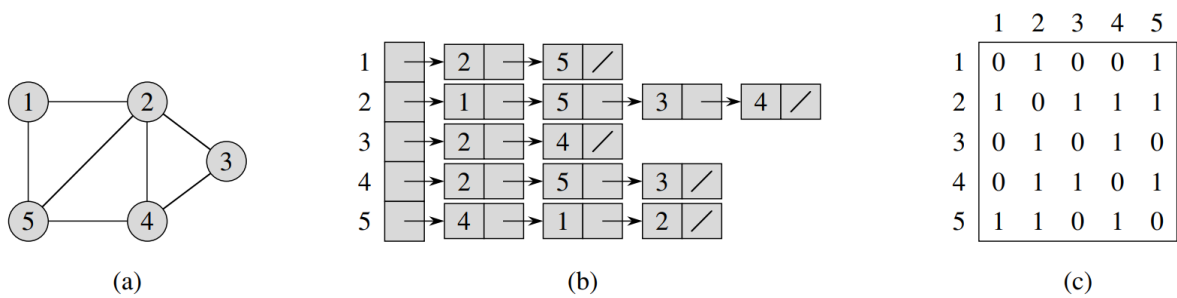


Figura 2.2: Duas representações de um grafo não direcionado. (a) Um grafo  $G$  com cinco nós e sete arestas. (b) A representação de uma lista de adjacência de  $G$ . (c) A representação de uma matriz de adjacência. Fonte: CORMEN et al. (2009), tradução dos autores.

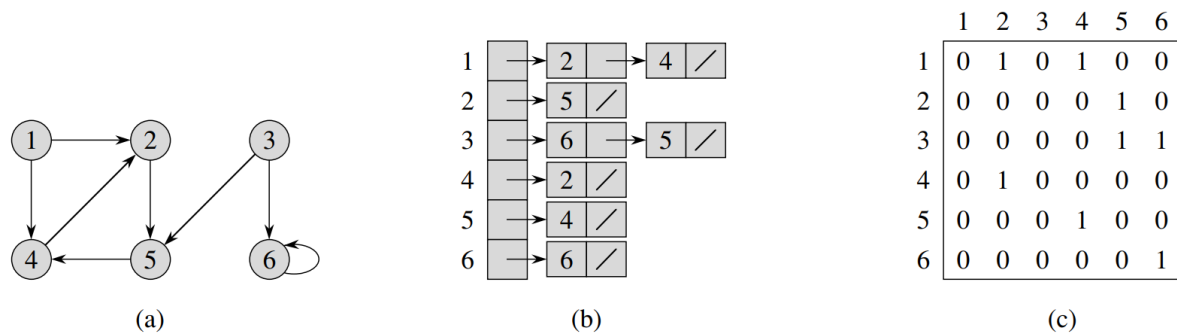


Figura 2.3: Duas representações de um grafo direcionado. (a) Um grafo  $G$  com seis nós e oito arestas. (b) A representação de uma lista de adjacência de  $G$ . (c) A representação de uma matriz de adjacência. Fonte: CORMEN et al. (2009), tradução dos autores.

such that there is an edge  $(u, v) \in E$ . That is,  $\text{Adj}[u]$  consists of all the vertices adjacent to  $u$  in  $G$ . (Alternatively, it may contain pointers to these vertices.) The vertices in each adjacency list are typically stored in an arbitrary order. Figure 22.1(b) is an adjacency-list representation of the undirected graph in Figure 22.1(a). Similarly, Figure 22.2(b) is an adjacency-list representation of the directed graph in Figure 22.2(a). If  $G$  is a directed graph, the sum of the lengths of all the adjacency lists is  $|E|$ , since an edge of the form  $(u, v)$  is represented by having  $v$  appear in  $\text{Adj}[u]$ . If  $G$  is an undirected graph, the sum of the lengths of all the adjacency lists is  $2|E|$ , since if  $(u, v)$  is an undirected edge, then  $u$  appears in  $v$ 's adjacency list and vice versa.

**22.1 Representations of graphs**

For both directed and undirected graphs, the adjacency-list representation has the desirable property that the amount of memory it requires is  $(V + E)$ . Adjacency lists can readily be adapted to represent weighted graphs, that is, graphs for which each edge has an associated weight, typically given by a weight function  $w : E \rightarrow \mathbb{R}$ . For example, let  $G = (V, E)$  be a weighted graph with weight function  $w$ . The weight  $w(u, v)$  of the edge  $(u, v) \in E$  is simply stored with vertex  $v$  in  $u$ 's adjacency list. The adjacency-list representation is quite robust in that it can be modified to support many other graph variants. A potential disadvantage of the adjacency-list representation is that there is no quicker way to determine if a given edge  $(u, v)$  is present in the graph than to search for  $v$  in the adjacency list  $\text{Adj}[u]$ . This disadvantage can be remedied by an adjacency-matrix representation of the graph, at the cost of using asymptotically more memory. (See Exercise 22.1-8 for suggestions of variations on adjacency lists that permit

faster edge lookup.) For the adjacency-matrix representation of a graph  $G = (V, E)$ , we assume that the vertices are numbered  $1, 2, \dots, |V|$  in some arbitrary manner. Then the adjacency-matrix representation of a graph  $G$  consists of a  $|V| \times |V|$  matrix  $A = (a_{ij})$  such that  $a_{ij} = 1$  if  $(i, j) \in E$ , 0 otherwise. Figures 22.1(c) and 22.2(c) are the adjacency matrices of the undirected and directed graphs in Figures 22.1(a) and 22.2(a), respectively. The adjacency matrix of a graph requires  $(V^2)$  memory, independent of the number of edges in the graph. Observe the symmetry along the main diagonal of the adjacency matrix in Figure 22.1(c). We define the transpose of a matrix  $A = (a_{ij})$  to be the matrix  $A^T = (a^T_{ij})$  given by  $a^T_{ij} = a_{ji}$ . Since in an undirected graph,  $(u, v)$  and  $(v, u)$  represent the same edge, the adjacency matrix  $A$  of an undirected graph is its own transpose:  $A = A^T$ . In some applications, it pays to store only the entries on and above the diagonal of the adjacency matrix, thereby cutting the memory needed to store the graph almost in half. Like the adjacency-list representation of a graph, the adjacency-matrix representation can be used for weighted graphs. For example, if  $G = (V, E)$  is a weighted graph with edge-weight function  $w$ , the weight  $w(u, v)$  of the edge  $(u, v) \in E$  is simply stored as the entry in row  $u$  and column  $v$  of the adjacency matrix. If an edge does not exist, a NIL value can be stored as its corresponding matrix entry, though for many problems it is convenient to use a value such as 0. Although the adjacency-list representation is asymptotically at least as efficient as the adjacency-matrix representation, the simplicity of an adjacency matrix may make it preferable when graphs are reasonably small. Moreover, if the graph is unweighted, there is an additional advantage in storage for the adjacency-matrix representation. Rather than using one word of computer memory for each matrix entry, the adjacency matrix uses only one bit per entry.

## 3 Metodologia

### 3.1 Método de desenvolvimento

O produto foi desenvolvido pelo modelo de cascata. A priorização das atividades foi baseada na competência da equipe em cada área de desenvolvimento. Dessa maneira, o *front end* foi considerado o foco e a prioridade no desenvolvimento, tendo em vista que o nível de conhecimento da equipe na área de design e usabilidade era perceptivelmente menor do que a habilidade para o desenvolvimento do *back end*.

A análise de requisitos. . .

Um MVP, foi desenvolvido durante a fase de projeto. A figura 3.1 apresenta a tela de criação de perguntas do MVP. Neste produto, é possível criar perguntas (3.1), responder perguntas postadas por outros usuários (3.2) e marcar essas respostas (3.3). Assim, o produto já tem informações suficientes para criar um grafo com os usuários no qual o peso das arestas é o nível de afinidade entre os eles baseado na quantidade de respostas marcadas.

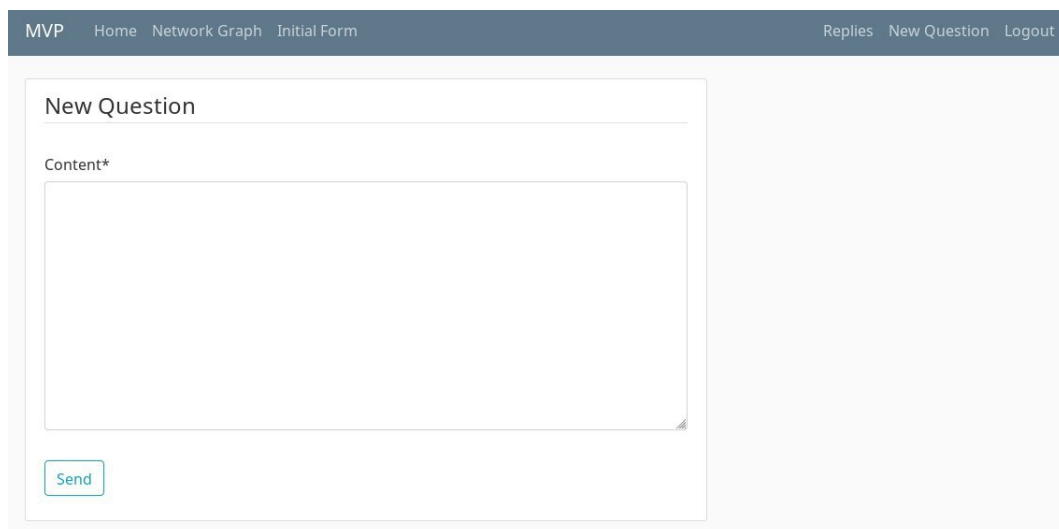
The image shows a web application interface for creating a new question. At the top, there is a dark blue navigation bar with links: MVP, Home, Network Graph, Initial Form, Replies, New Question, and Logout. The main content area is light gray. A white box with a thin border contains the 'New Question' form. Inside the box, the title 'New Question' is at the top. Below it is a label 'Content\*' followed by a large, empty text area for input. At the bottom left of the text area is a small 'Send' button.

Figura 3.1: Tela de criação de perguntas do MVP. Fonte: os autores.

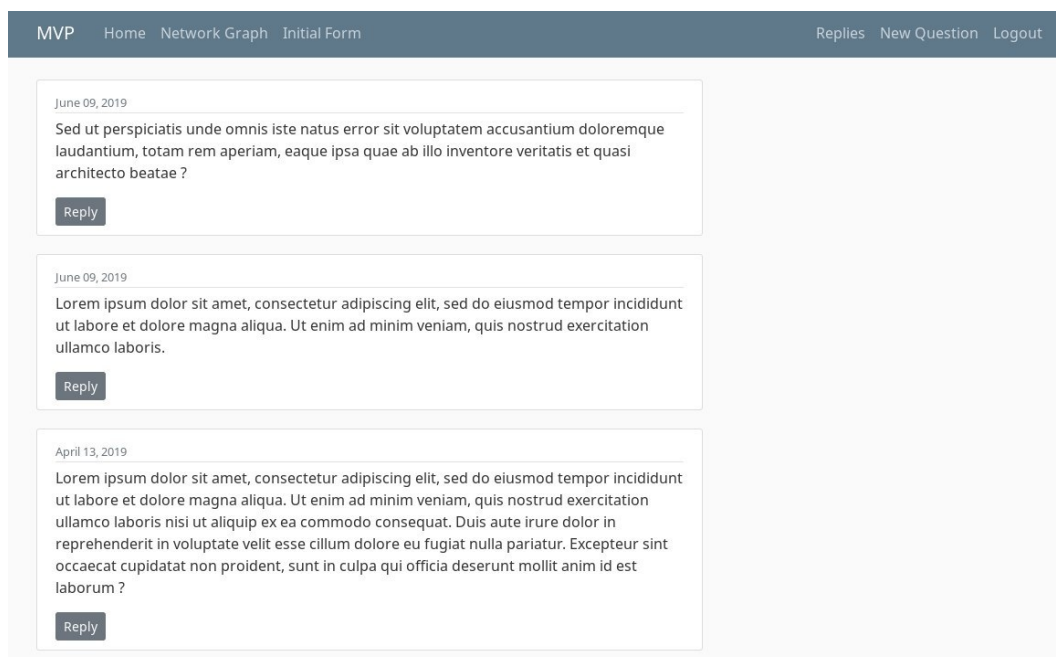


Figura 3.2: Tela de visualização de perguntas do MVP. Nesta tela é possível escolher uma pergunta para ser respondida. Fonte: os autores.

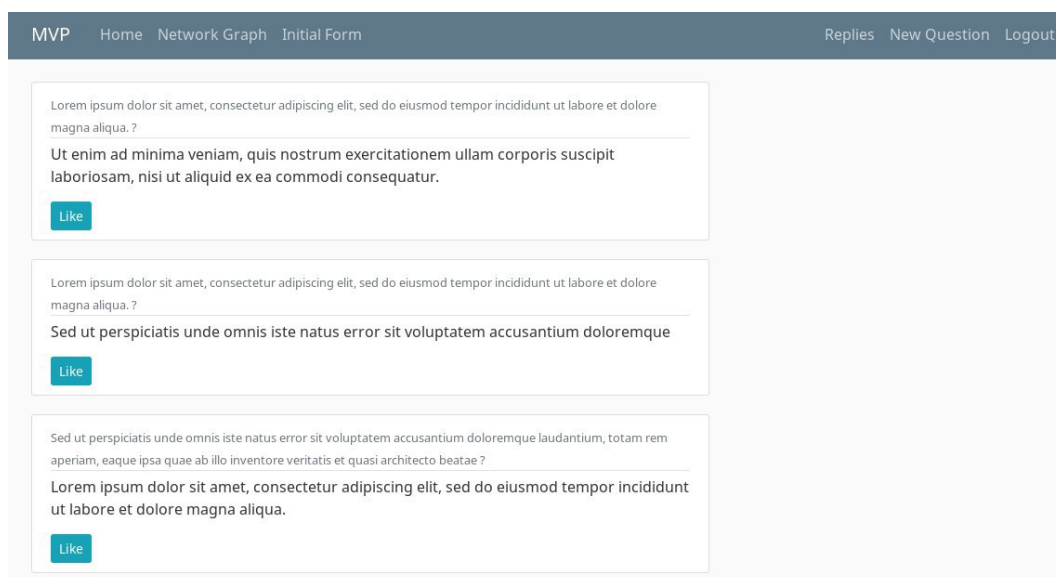


Figura 3.3: Tela de visualização de respostas recebidas do MVP. Nesta tela é possível marcar perguntas favoritas. Fonte: os autores.

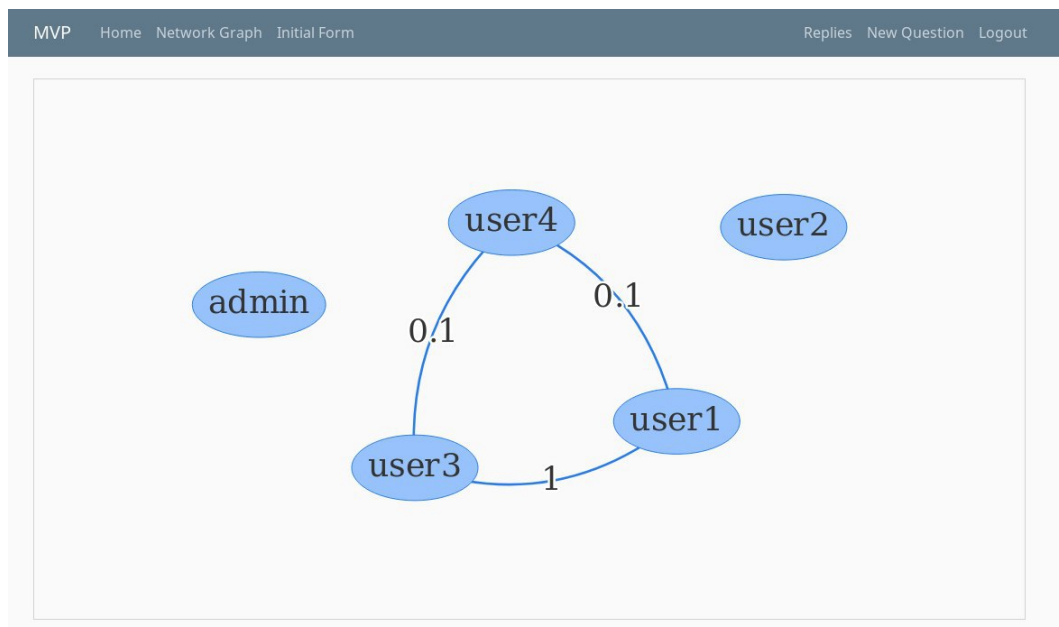


Figura 3.4: Visualização do grafo que representa as ligações entre usuários da rede social no MVP. Fonte: os autores.

A figura 3.4 é a tela do MVP que representa os usuários e suas conexões criadas na rede social por meio de um grafo.

## 3.2 Arquitetura

- Qual a tecnologia
- Custo
- Arquitetura

Desenho da arquitetura

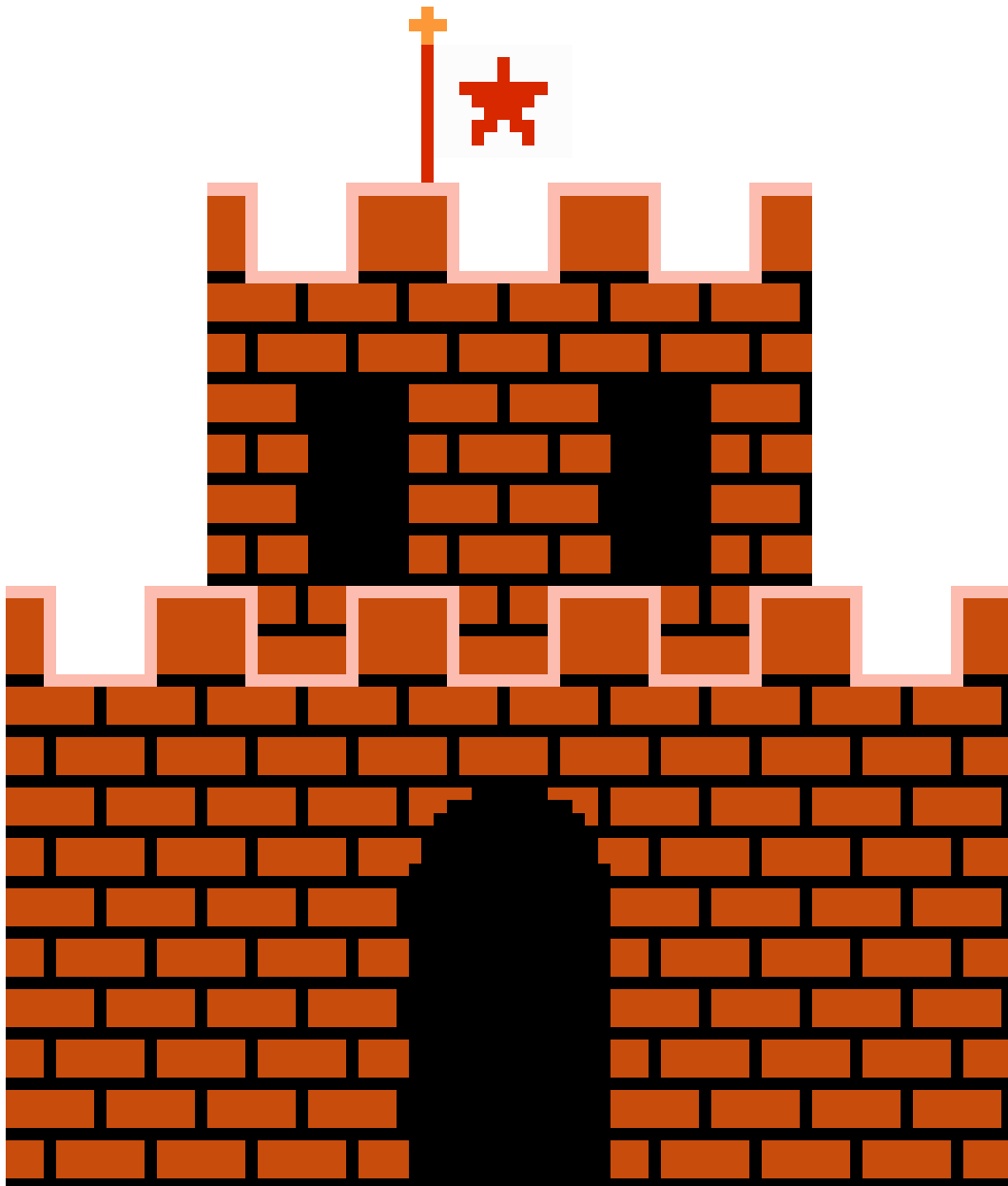


Figura 3.5: Arquitetura do software. Fonte: os autores.

### 3.3 Tecnologia aplicada

- Linguagens de programação
- Frameworks

Frameworks: IONIC django  
angular e github.

Linguagens: python typescript javascript html/css Hardware: PC Notebook thinkpad x201, SSD 120GB e 8GB de RAM primeira geracao do i5 e arch linux, server usa debian stretch Infraestrutura: Docker VPS: CPU: Intel (Haswell, noTSX) (1)@2.3GHz GPU: CirrusLogicGD5446 Memory: 583MB / 1956MB.

### 3.4 Análise do sistema

Diagramas de caso de USO

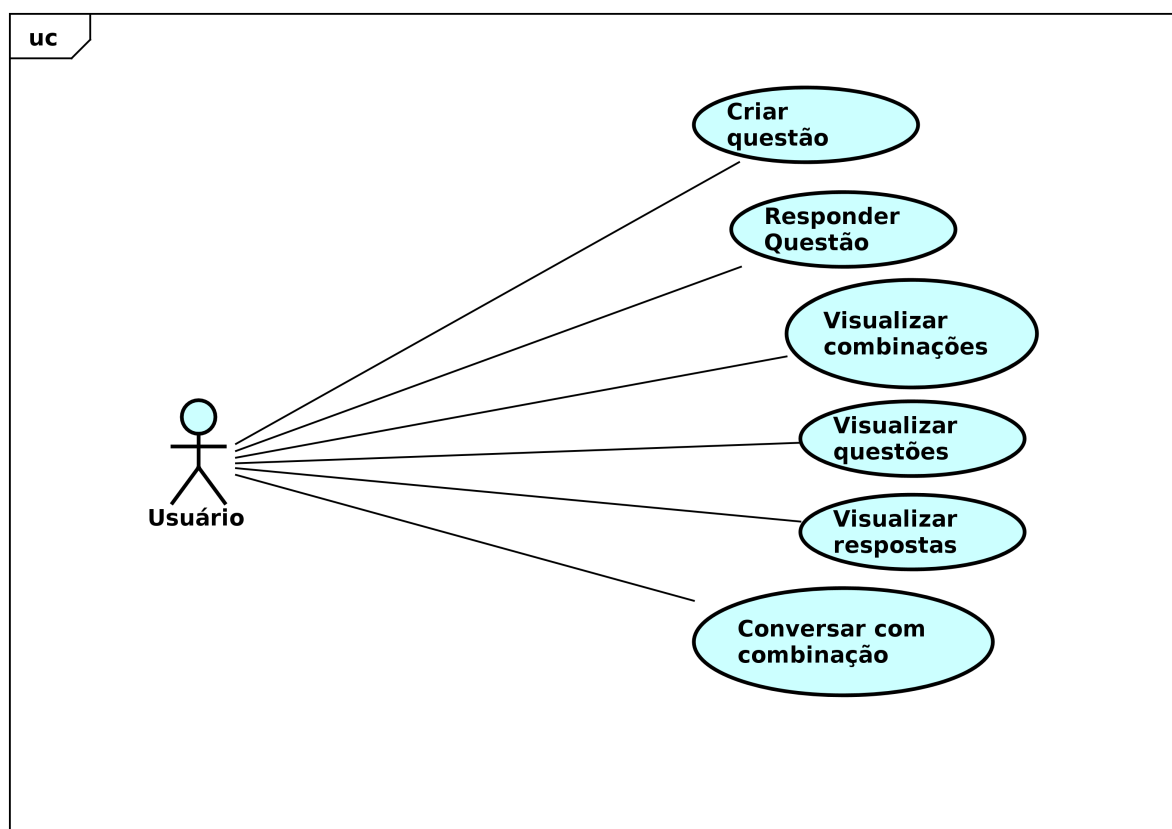


Figura 3.6: Diagrama de caso de uso nível 1. Fonte: os autores.

Diagramas de sequência

Diagrama de Classes

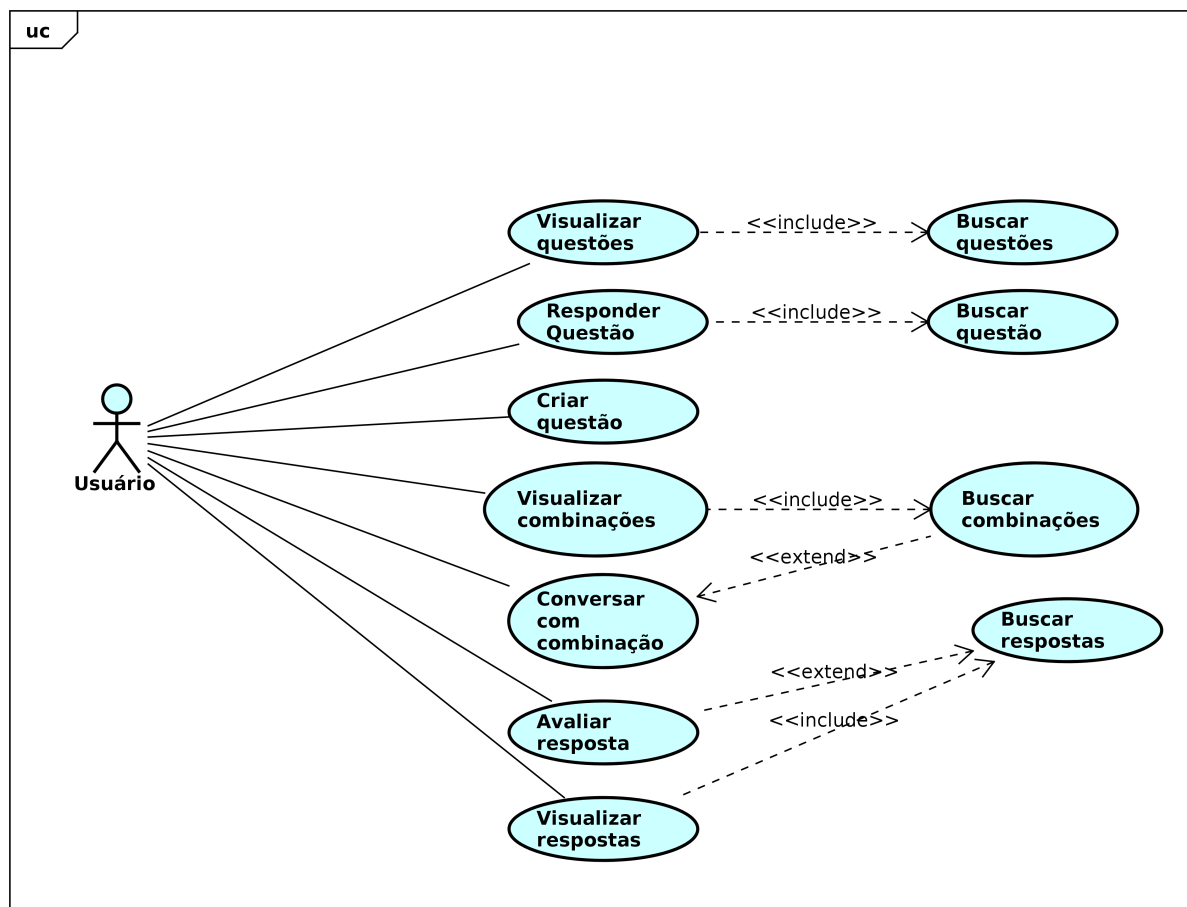


Figura 3.7: Diagrama de caso de uso nível 2. Fonte: os autores.

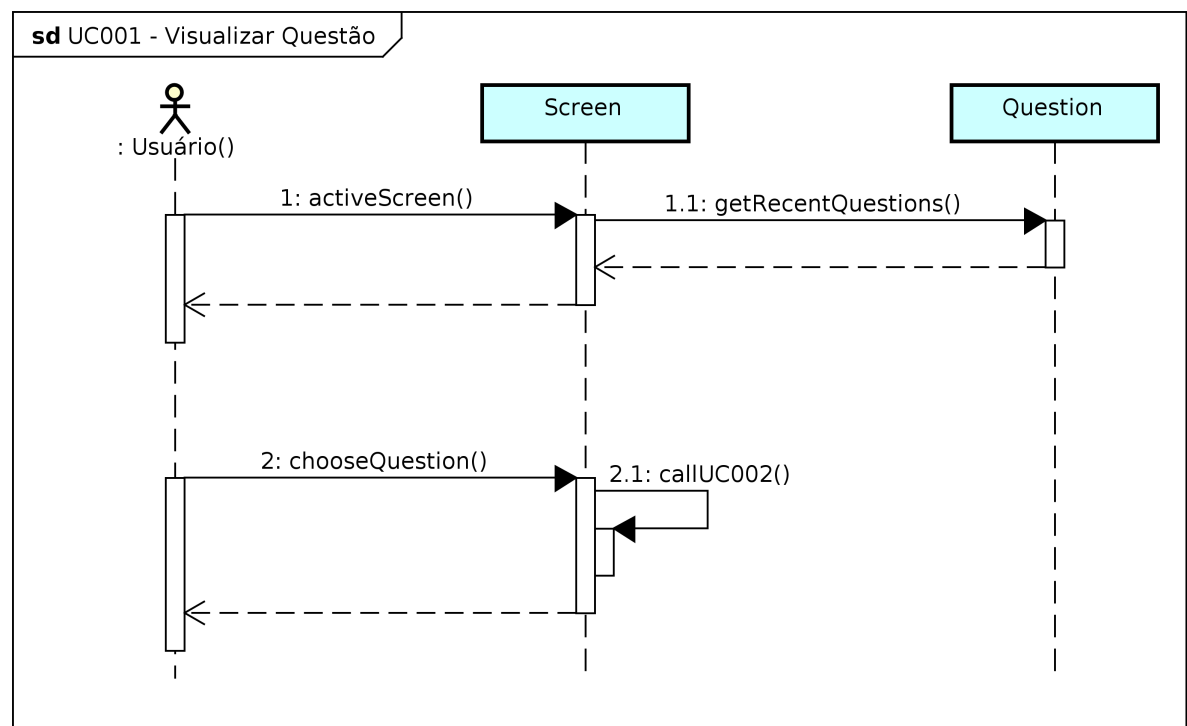


Figura 3.8: Diagrama de caso de uso UC001 - Visualizar Questão. Fonte: os autores.



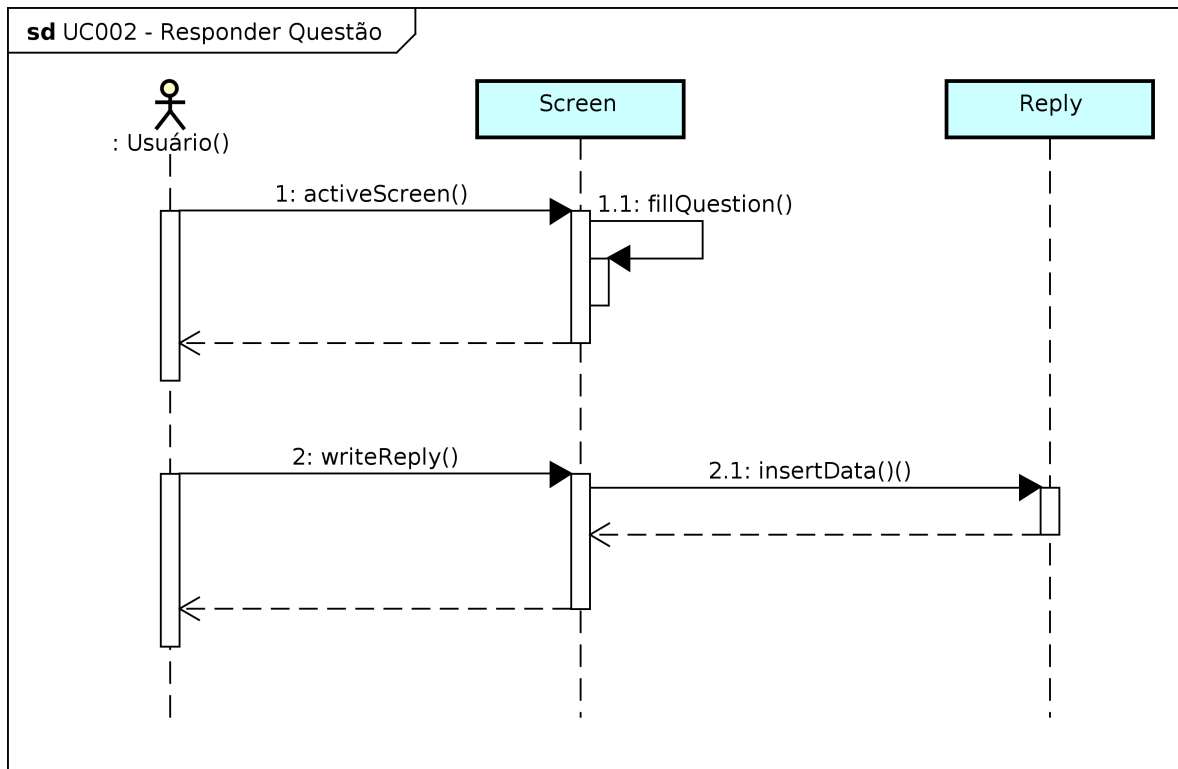


Figura 3.9: Diagrama de caso de uso UC002 - Responder Questão. Fonte: os autores.

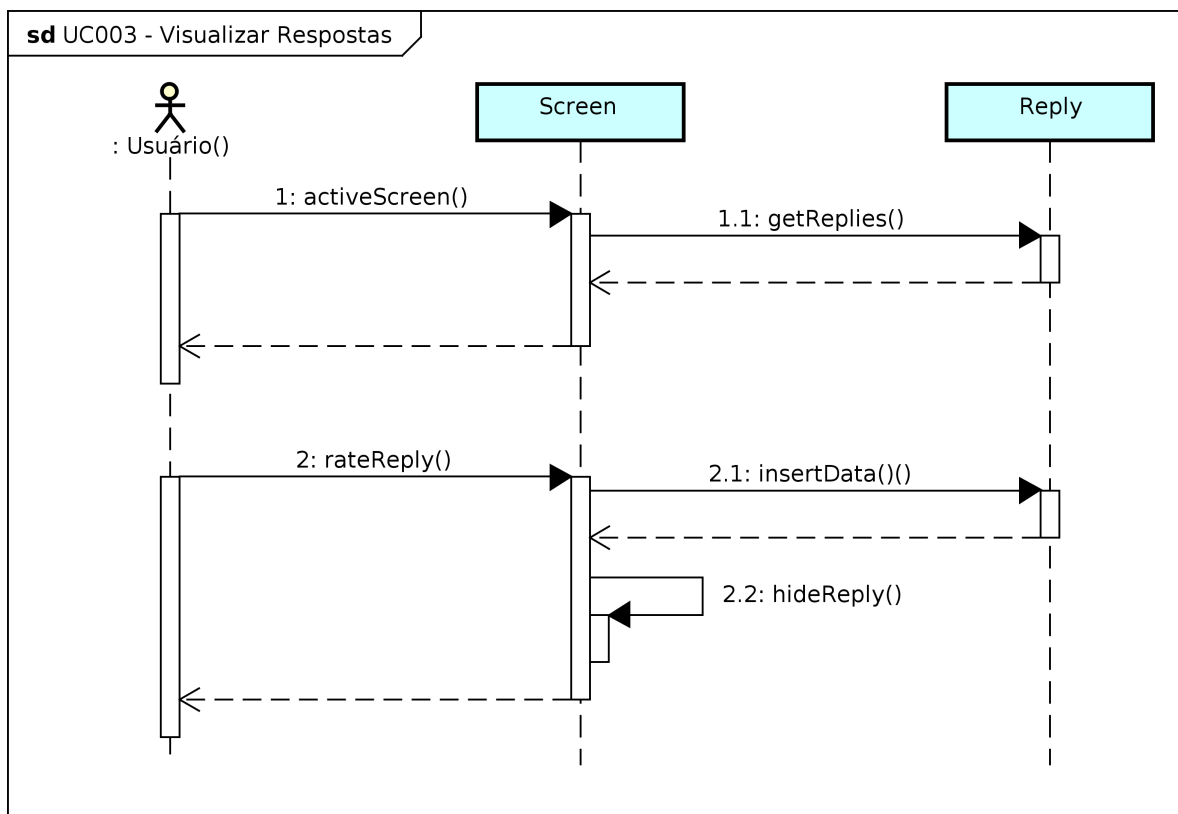


Figura 3.10: Diagrama de caso de uso UC003 - Visualizar Respostas. Fonte: os autores.

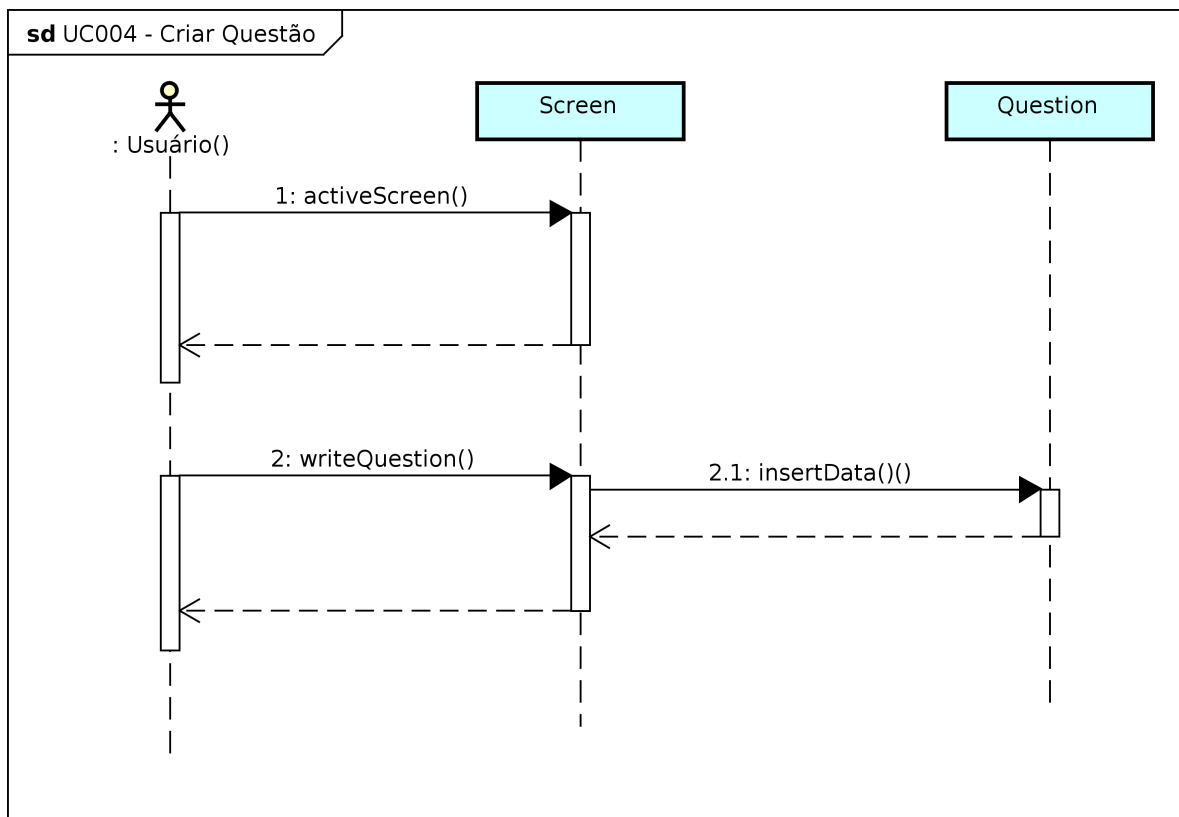


Figura 3.11: Diagrama de caso de uso UC004 - Criar Questão. Fonte: os autores.

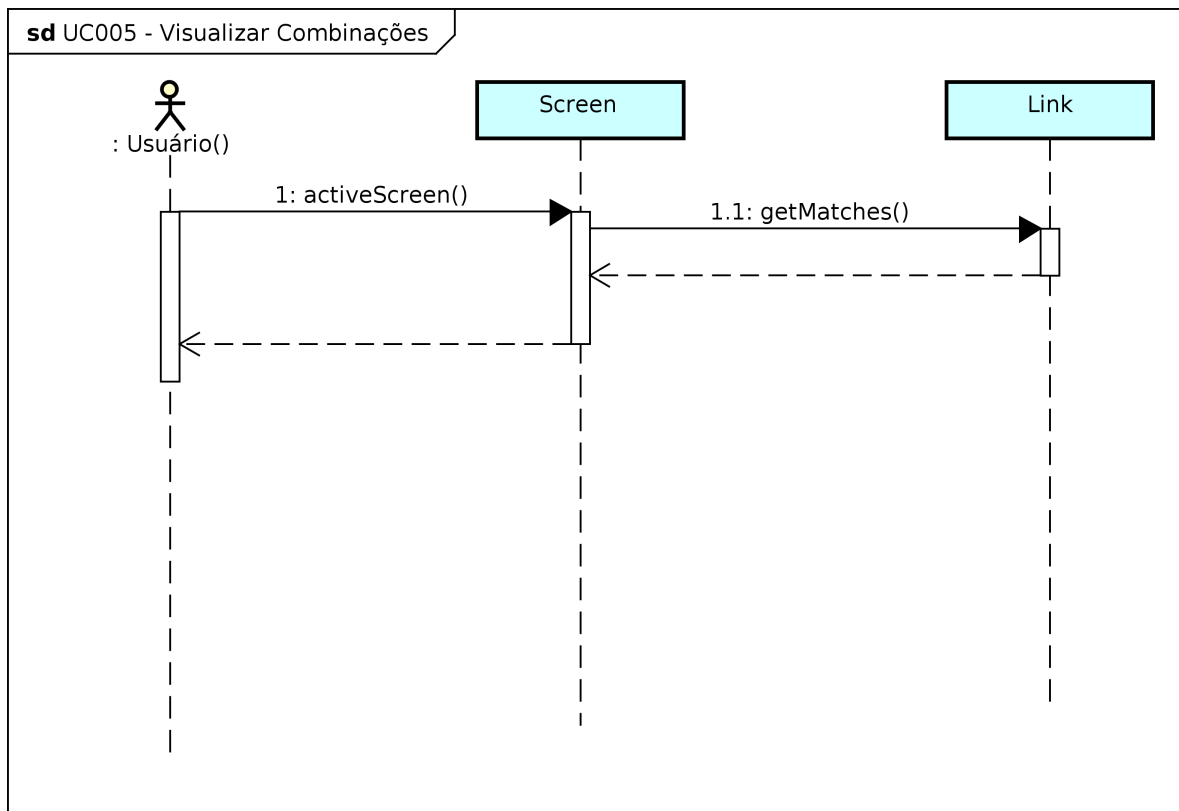


Figura 3.12: Diagrama de caso de uso UC005 - Visualizar Combinações. Fonte: os autores.

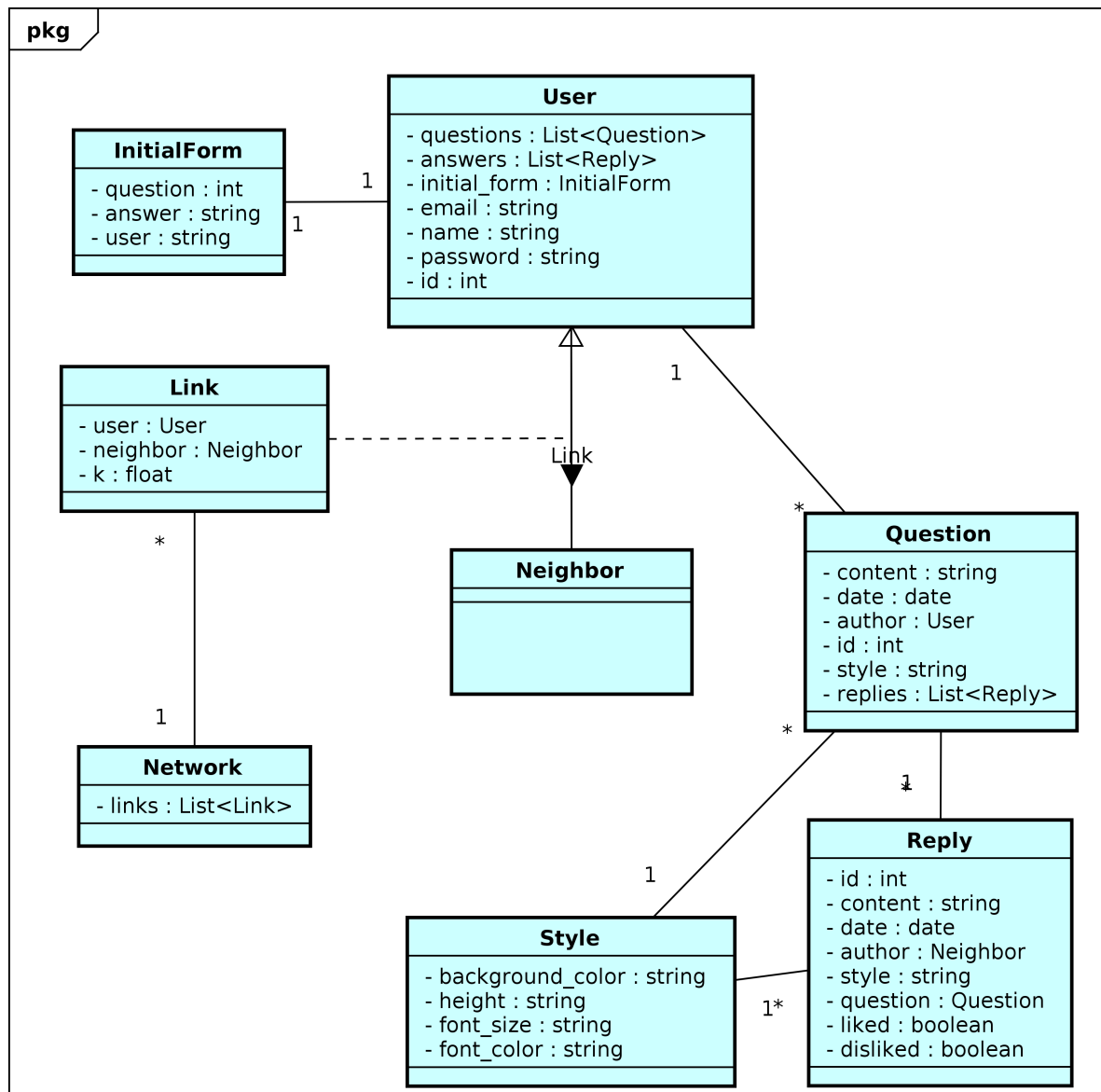


Figura 3.13: Diagrama de classes. Fonte: os autores.

### 3.5 Cálculo de do peso da aresta

O peso de uma aresta do grafo,  $k$ , representa o nível de afinidade entre os dois usuários conectados por esta aresta. Um usuário pode estar conectado a vários outros usuários.

Quando um usuário entra pela primeira vez na rede social, ele é obrigado a preencher um questionário contendo as questões constantes na tabela 3.1. A partir das respostas deste questionário, um valor para  $k$  é calculado levando em conta, tão somente, a similaridade entre as respostas de cada usuário.

Tabela 3.1: Formulário inicial

Pergunta
Você prefere cachorro ou gato?
Você prefere rock ou funk?
Você prefere verão ou inverno?
Você prefere cinema ou teatro?
Você prefere cerveja ou vinho?
Você prefere o dia ou a noite?
Você prefere sair ou ficar em casa?
Você fuma?
Você tem alguma religião?
Você acredita em signos?
Você prefere praia ou campo?

Para a determinação inicial de  $k$ , logo após o preenchimento do formulário, é calculado o total de respostas iguais entre dois usuários e aplicada a seguinte equação:

$$k_{ab} = \frac{R_{ab} * (x - 1)}{N_p} \quad (3.1)$$

Onde  $R_{ab}$  é o total de respostas do usuário A iguais ao usuário B;  $x$  é o valor objetivo para considerar dois usuários similares - este valor será discutido adiante nesta seção - e  $N_p$  é o total de perguntas do questionário inicial.

Na equação 3.1, o numerador é multiplicado por  $x - 1$  para que os usuários que tiveram todas as questões respondidas da mesma maneira no questionário tão somente fiquem muito próximos da margem que define a habilitação do mensageiro instantâneo. O objetivo é tornar obrigatória a interação por meio de perguntas e respostas antes que dois usuários possam ser considerados similares o suficiente para desfrutarem do mensageiro.

Então, conforme as perguntas postadas pelos usuários vão sendo respondidas e apreciadas, valor do peso da aresta, definido como  $k_{ab}$ , que liga os usuários  $A$  e  $B$ , é recalculado, para cada resposta apontada como apreciada, a partir da seguinte definição:

$$k_{ab} = k_{ab} + \frac{(R_{ab} + R_{ba})}{(P_A + P_B)} \quad (3.2)$$

Onde  $R_{ab}$  é o número de respostas que o usuário  $A$  recebeu do usuário  $B$  e  $A$  gostou;  $R_{ba}$  é o número de respostas que o usuário  $B$  recebeu do usuário  $A$  e  $B$  gostou;  $P_A$  é o número de perguntas postadas pelo usuário  $A$  e  $P_B$  o número de respostas postadas pelo usuário  $B$ .

Na figura 3.14, podemos ver a evolução do peso  $k$  para cada resposta recebida pelo usuário  $A$  que ele gostou. Nesta situação o crescimento do valor de  $k$  é linear, pois todas as perguntas postadas pelo usuário  $A$  estão respondidas por  $B$  e as respectivas respostas foram curtidas.

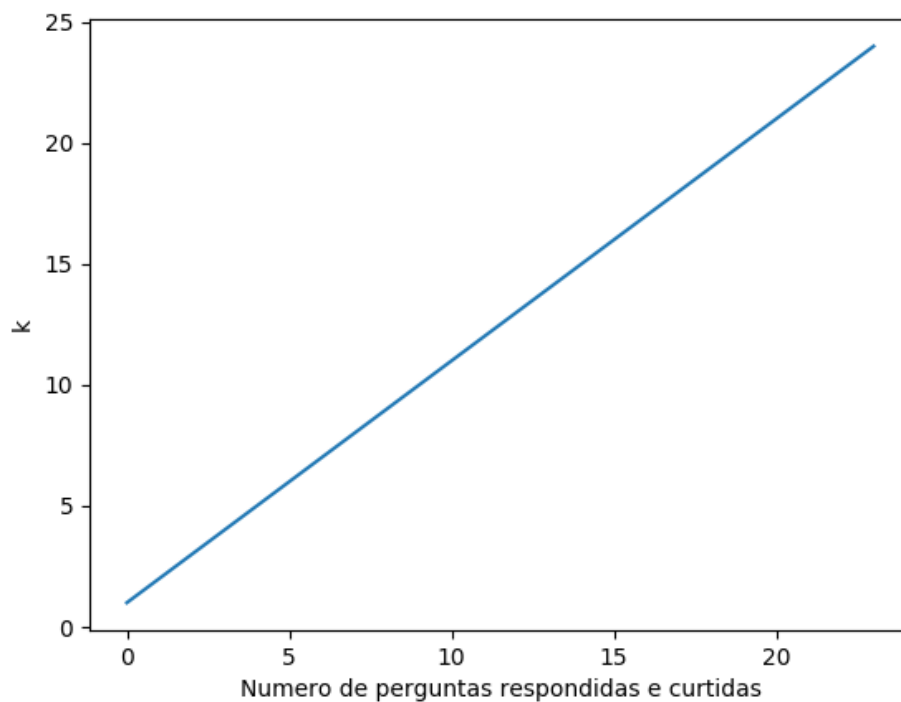


Figura 3.14: Evolução de  $k$  se todas as perguntas postadas por  $A$  forem respondidas por  $B$ . Fonte: os autores.

Como o cálculo leva em conta todas as perguntas postadas pelo usuário  $A$ , no caso de já haver perguntas postadas e não respondidas, o crescimento do valor de  $k$  é afetado num grau inversamente proporcional ao número de perguntas postadas, como pode-se observar na figura 3.15.

É importante observar que o peso do incremento para  $k$  é menor quando maior for o número de perguntas postadas seja por  $A$  ou  $B$ . Dessa maneira, podemos depreender que a interação entre usuários novos, ou com poucas perguntas postadas, terá influência maior no crescimento do valor de  $k$ . Por outro lado, a interação entre usuários com muitas perguntas já postadas, incrementará um valor menor sobre  $k$ .

Tendo calculado um peso para cada aresta que liga os usuários, foi necessário definir um limiar  $x$  que seria o valor mínimo de  $k$  para considerar que dois usuários são similares o suficiente para serem postos em contato sob o mensageiro instantâneo.

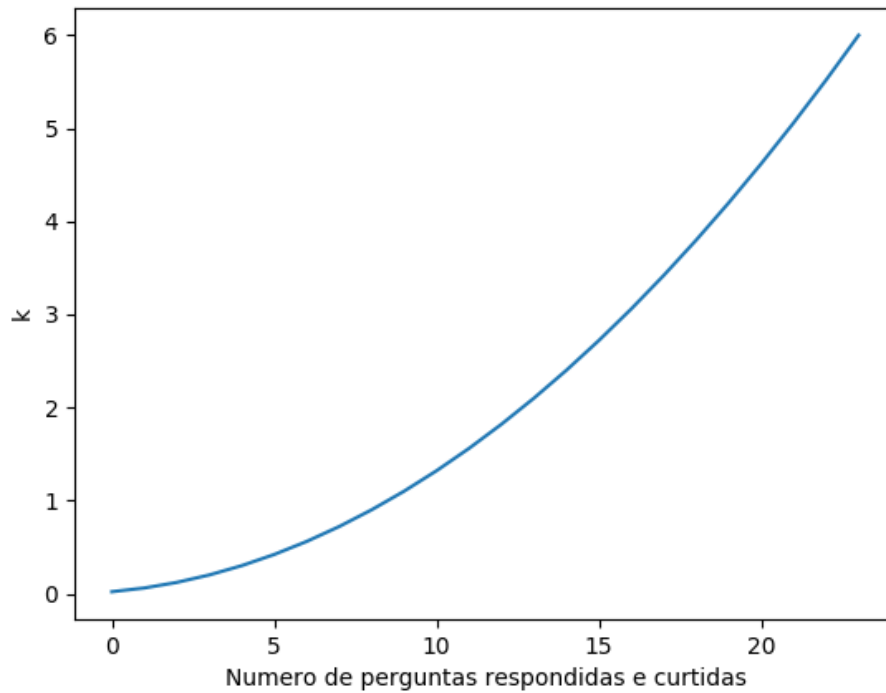


Figura 3.15: Evolução de  $k$  se  $A$  postou 50 perguntas e apenas 25 respostas de  $B$  foram curtidas. Fonte: os autores.

A definição de  $x$  provou-se um desafio, tendo em vista que o comportamento de  $k$  varia em função do número total de perguntas postadas por dois usuários em análise. A figura 3.16 mostra que a convergência para  $x = 1$  para o caso de dois usuários que postaram poucas perguntas é rápida, pois o incremento de  $k$  é maior. No caso simulado na figura ??, pouco mais de cinco respostas curtidas foram o suficiente para atingir o  $x$ .

A figura 3.17 demonstra que a convergência de  $k$  e  $x$  para usuários que já postaram muitas perguntas demanda muito mais respostas curtidas, neste caso mais que vinte respostas. Isto deve-se ao fato de ter sido decidido que a equação 3.2 tem como divisor a soma de todas as perguntas já postadas pelos dois usuários, portanto, o incremento de  $k$  é inversamente proporcional à soma de todas as perguntas postadas pelos usuários.

Uma adequação simplista para este problema seria definir  $k$  como o número total de respostas curtidas entre dois usuários. Assim,  $x$  seria o número mínimo de respostas que teriam que ser curtidas entre os usuários para que fossem considerados similares o suficiente para contatarem-se pelo mensageiro instantâneo. Porém, dessa maneira ignora-se que, se houverem muitas perguntas postadas por  $A$  sem resposta de  $B$  é possível que haja mais diferenças do que similaridades entre os usuários, haja vista que muitas perguntas são ignoradas.

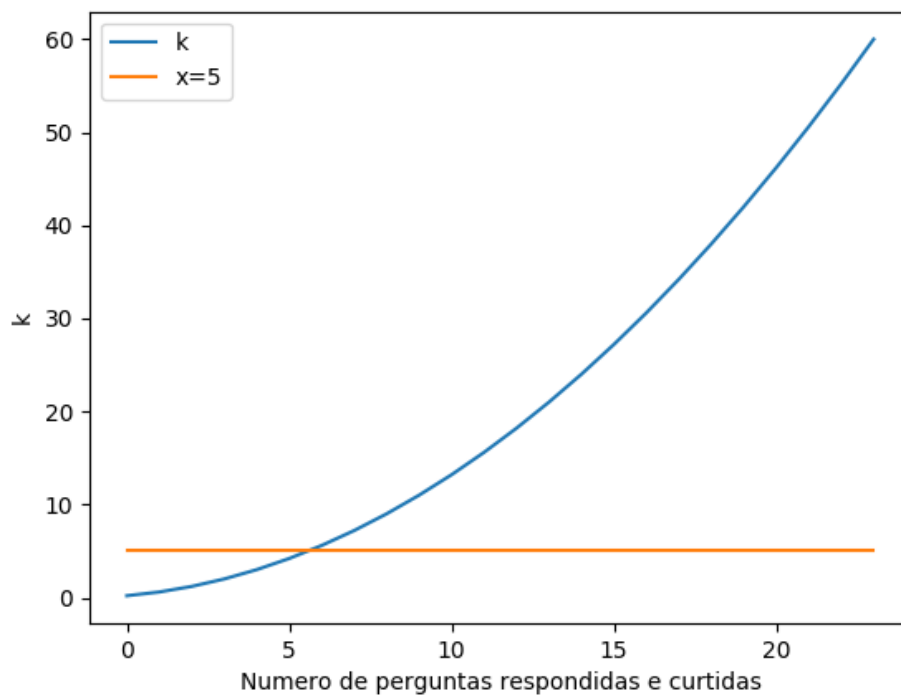


Figura 3.16: Evolução de  $k$  em função das respostas curtidas se  $A$  e  $B$  têm poucas perguntas postadas. Ex.: menos que 10 perguntas. Fonte: os autores.

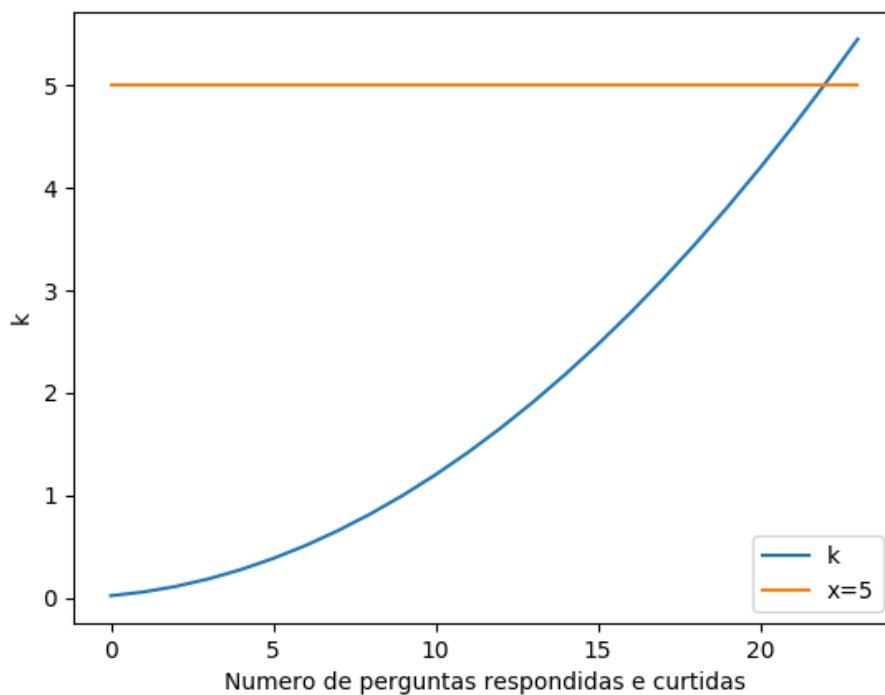


Figura 3.17: Evolução de  $k$  em função das respostas curtidas se  $A$  e  $B$  têm muitas perguntas postadas. Ex.: mais que 50 perguntas. Fonte: os autores.

Portanto, é imprescindível considerar o volume de perguntas postadas. Uma evolução poderia adequar o software para contabilizar o número de perguntas postadas por  $A$  e vistas por  $B$ . Assim, poderíamos adequar o cálculo de  $k$  como segue:

$$k = k + \frac{R_{ab} + R_{ba}}{P_{AvB} + P_{BvA}} \quad (3.3)$$

Onde  $R_{ab}$  é o número de respostas que o usuário  $A$  recebeu do usuário  $B$  e  $A$  gostou;  $R_{ba}$  é o número de respostas que o usuário  $B$  recebeu do usuário  $A$  e a  $B$  gostou;  $P_{AvB}$  é o número de perguntas postadas pelo usuário  $A$  e vistas por  $B$  e  $P_{BvA}$  o número de respostas postadas pelo usuário  $B$  e vistas por  $A$ .

Aplicando a equação 3.3, o incremento do valor de  $k$  para cada resposta curtida seria menor para o caso de muitas perguntas terem sido vistas e poucas respostas terem sido dadas. De maneira inversa, quanto menor o conjunto de perguntas trocadas entre os usuários, maior o peso que uma resposta curtida terá sobre  $k$ .

Seria conveniente também, afetar as perguntas postadas com uma validade por cronologia ou fixar um valor máximo para o total de perguntas postadas pelos usuários  $A$  e  $B$ . Para a implementação do produto neste trabalho, não foi considerada a idade das perguntas nem estabelecido um valor limite do somatório de perguntas postadas por  $A$  e  $B$  pois seria inconveniente para o desenvolvimento.



## 4 Produto

### 4.1 Perguntas e respostas

Telas iniciais, perguntas e respostas.

### 4.2 Mensageiro instantâneo

Mensageiro instantâneo.

## **5 Conclusão**

Na conclusão não devo falar nada que já não tenha sido dito no restante do trabalho.

## Referências

- BONDY, J. e MURTY, U. (1976). *Graph Theory*. Springer Publishing Company, Incorporated, 1st edition.
- CORMEN, T., LEISERSON, C., RIVEST, R. e STEIN, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.
- DIJKSTRA, E. (1959). A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271.
- KEMP, S. (2018). Global digital report 2018. <https://wearesocial.com/blog/2018/01/global-digital-report-2018>. Acessado em 15/05/2019.
- LUA, A. (2019). The 21 most popular social media sites in 2019. <https://buffer.com/library/social-media-sites>. Acessado em 04/06/2019.
- MARTELETO, R. M. (2001). Análise de redes sociais - aplicação nos estudos de transferência da informação. *Ci. Inf.*, 30(1):71–78.
- MASLOW, A. H. (1943). A theory of human motivation. *Psychological Review*, 50(4):370–396.
- MILGRAM, S. (1967). The small-world problem. *Psychology Today*, 1(1).
- MISHRA, A. (2019). Finding nearest neighbors in graphs locally. *arXiv e-prints*.
- PATERSON, M. S. e YAO, F. F. (1992). On nearest-neighbor graphs. *Springer LNCS*, 623(1):416–426.
- SHAMS, B. e HARATIZADEH, S. (2016). Graph-based collaborative ranking. *Expert Systems with Applications*, 67.
- TERRELL, K. (2017). The history of social media. <https://historycooperative.org/the-history-of-social-media/>. Acessado em 04/06/2019.