

CARLOS FELIPE AMORIM E MURILO VIDAL

**SISTEMA DE INTERAÇÕES SOCIAIS BASEADAS EM SIMILARIDADE: UMA
ANÁLISE DETERMINÍSTICA PARA O CÁLCULO DO PESO DAS ARESTAS EM
UM GRAFO**

(versão pré-defesa, compilada em 17 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

Escrever resumo

Palavras-chave: grafo, recomendação, rede social.

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.	17
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.	18
3.1	Fases empregadas do modelo em cascata. Fonte: os autores.	19
3.2	Tela de criação de perguntas do MVP. Fonte: os autores.	20
3.3	Tela de visualização de perguntas do MVP. Nesta tela é possível escolher uma pergunta para ser respondida. Fonte: os autores.	20
3.4	Tela de visualização de respostas recebidas do MVP. Nesta tela é possível marcar perguntas favoritas. Fonte: os autores.	21
3.5	Visualização do grafo que representa as ligações entre usuários da rede social no MVP. Fonte: os autores.	21
3.6	Protótipo da tela de visualização de perguntas. Fonte: os autores.	22
3.7	Protótipo da tela de criação de perguntas. Fonte: os autores.	22
3.8	Protótipo da tela de criação de respostas para as perguntas apresentadas. Fonte: os autores.	23
3.9	Protótipo da tela de visualização das respostas dadas às perguntas produzidas pelo usuário. Fonte: os autores.	23
3.10	Protótipo da tela de visualização dos contatos. Fonte: os autores.	23
3.11	Protótipo da tela de mensagens instantâneas. Fonte: os autores.	23
3.12	Diagrama de implementação. Fonte: os autores.	25
3.13	Diagrama de classes. Fonte: os autores.	27
3.14	Diagrama de banco de dados. Fonte: os autores.	28
3.15	Evolução de k se todas as perguntas postadas por A forem respondidas por B . Fonte: os autores.	30
3.16	Evolução de k se A postou 50 perguntas e houve apenas 25 respostas de B . Fonte: os autores.	31

3.17	Evolução de k em função das respostas curtidas se A e B têm poucas perguntas postadas. Ex.: menos que 10 perguntas. Fonte: os autores.	32
3.18	Evolução de k em função das respostas se A e B têm muitas perguntas postadas. Ex.: mais que 50 perguntas. Fonte: os autores.	32
3.19	Valor de k em função do número de perguntas respondidas e de respostas curtidas entre dois usuários. Fonte: os autores.	34
3.20	A área em vermelho escuro compreende a relação entre respostas e respostas apreciadas entre dois usuários onde k é maior que 1,5 para 25 perguntas respondidas. Fonte: os autores.	35
3.21	A área em vermelho escuro compreende a relação entre respostas e respostas apreciadas entre dois usuários onde k é maior que 1,5 para 200 perguntas respondidas. Fonte: os autores.. . . .	35
3.22	Crescimento de k de acordo com a função 3.7 onde $w = 5$ e com um limite de x estabelecido em 5. Fonte: os autores.. . . .	36
4.1	Grafo representando os usuários do sistema e suas relações. Fonte: os autores.. .	38
4.2	Tela de login. Fonte: os autores.	40
4.3	Tela de registro. Fonte: os autores.	40
4.4	Tela de visualização de questões. Fonte: os autores.	41
4.5	Tela de criação de questões. Fonte: os autores.. . . .	41
4.6	Tela de resposta para questões. Fonte: os autores.	42
4.7	Tela de visualização de respostas. Fonte: os autores.	42
4.8	Tela de visualização dos contatos criados. Fonte: os autores.	43
4.9	Tela de visualização das mensagens recebidas no mensageiro instantâneo. Fonte: os autores.. . . .	44
A.1	Diagrama de caso de uso nível 1. Fonte: os autores.	48
A.2	Diagrama de caso de uso nível 2. Fonte: os autores.	49
A.3	Diagrama de caso de uso UC001 - Visualizar Questão. Fonte: os autores. . . .	50
A.4	Diagrama de caso de uso UC002 - Responder Questão. Fonte: os autores. . . .	50
A.5	Diagrama de caso de uso UC003 - Visualizar Respostas. Fonte: os autores. . . .	51
A.6	Diagrama de caso de uso UC004 - Criar Questão. Fonte: os autores.	51
A.7	Diagrama de caso de uso UC005 - Visualizar Combinações. Fonte: os autores. .	52

Lista de Tabelas

3.1	Formulário inicial	29
-----	------------------------------	----

Lista de Abreviações

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

Lista de Símbolos

ψ

psi, vigésima terceira letra do alfabeto grego

Sumário

1	Introdução	10
2	Fundamentação Teórica.	13
2.1	Redes Sociais	13
2.2	Algoritmos de Recomendação	14
2.3	Grafo	15
3	Materiais e Métodos	19
3.1	Método de desenvolvimento	19
3.1.1	Análise de requisitos	19
3.2	Arquitetura	24
3.3	Tecnologia aplicada	25
3.4	Análise do sistema	26
3.5	Cálculo do peso da aresta	29
4	Apresentação do Sistema	38
4.1	Perguntas e respostas	39
4.2	Mensageiro instantâneo	43
5	Considerações Finais	45
	Referências	46
	Apêndice A: Casos de uso	48
A.1	Diagramas de Caso de uso	48
A.2	Diagramas de sequência	49
A.3	Especificação de Casos de Uso	53

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 partes:

- **Fundamentação Teórica:** Nesta seção, visitamos os principais autores e documentos relacionados à redes sociais, algoritmos de classificação e de sugestão e grafos.
- **Materiais e Métodos:** Nesta seção, são expostos 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 do Sistema:** Telas e funcionalidades do produto final.

2 Fundamentação Teórica

2.1 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

despense, 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 Recomendação

Quando um usuário acessa um sistema qualquer que lhe oferece itens, por exemplo, livros, um sistema de recomendação pode lhe proporcionar a conveniência de sugerir-lhe um livro mais adequado para o seu gosto quando a quantidade de opções disponíveis é opressivamente vasta, diz RICCI et al. (2010).

Esses itens, são quaisquer coisas que podem ser sugeridas ao usuário, sejam produtos, serviços, notícias, publicidade, ou, no caso deste trabalho, outros usuários com potencial para tornarem-se amigos.

Para RICCI et al. (2010), a forma mais simples de um sistema de recomendação é uma lista com itens ordenados pela similaridade com o usuário que receberá a recomendação. Os sistemas de sugestão analisam vários tipos de dados por diversos métodos diferentes para encontrar os itens mais adequados para cada usuário. De acordo com RICCI et al. (2010), os primeiros algoritmos de recomendação utilizavam dados gerados por recomendações feitas pela comunidade de usuários. Esse método comparava gostos similares entre os usuários e recomendava itens ainda não consumidos por alguns usuários. A este método dá-se o nome de Filtragem Colaborativa (GOLDBERG et al., 1992).

Desde os anos 1990, a perseguição por sistemas de recomendações cada vez mais eficientes impulsionou o surgimento do Prêmio Netflix em 2006, (Netflix, 2006). Neste concurso, a empresa Netflix propôs pagar um prêmio de US\$1.000.000 para quem elaborasse um sistema que pudesse melhorar em, pelo menos, 10% a acurácia do sistema que estava em uso à época. O time *BellKor's Pragmatic Chaos*, venceu a disputa empregando diversos algoritmos e métodos de classificação mesclados em um mesmo sistema, (BELL et al., 2007).

Entre as possibilidades de métodos e algoritmos que podem ser empregados em um sistema de recomendação, podemos ressaltar, baseados no texto de Felfernig et al. (2014), as seguintes:

- Filtragem colaborativa
- Filtragem baseada em conteúdo
- Recomendação baseada em conhecimento

A filtragem colaborativa, como já mencionado, leva em conta as preferências já informadas pelos usuários para compará-las e sugerir novas possibilidades de itens. Tem a tendência de sugerir itens que já são apreciados por usuários que, de alguma maneira, já se relacionam, seja por interesses definidos, grupos específicos ou até mesmo por parentesco.

A filtragem baseada em conteúdo toma por verdadeira a afirmação de que os usuários que já demonstraram interesse por um tópico somente serão agradados por tópicos semelhantes e não podem ser instigados por assuntos fora do escopo do que já foi definido como interesse inicialmente. Dessa maneira, o sistema considera de que um usuário que declarou interesse por filmes deve receber sugestões de itens relacionados a este tema. É claro que este método tem a fraqueza de não sugerir novos assuntos que, potencialmente, despertariam o interesse do usuário.

Finalmente, a recomendação baseada em conhecimento, que toma em conta regras de restrições de interesse definidas no sistema e os aplica à lista de itens disponíveis. Desse modo, os itens são sugeridos a partir do resultado da filtragem realizada a partir das regras definidas. A definição de tais regras passa por análises semânticas do conteúdo disponível no sistema que diga respeito ao usuário. O sistema desenvolvido neste trabalho assemelha-se a uma recomendação baseada em conhecimento no sentido de que a regra definida para relacionar os itens será uma equação determinística que calcula um valor de proximidade entre dois itens.

Os vários métodos de recomendação tem sua acurácia diretamente influenciada pela quantidade de informação disponível sobre os usuários ou gerada pelos usuários. Quanto maior a quantidade de informação disponível, maior a acurácia e, em linhas gerais, menor a performance do sistema, uma vez que o tempo de processamento da informação é incrementado. Por essa razão, boas soluções são baseadas em aprendizado de máquina e inteligência artificial. O emprego dessas tecnologias tem o objetivo de melhorar a performance e a acurácia das sugestões.

2.3 Grafo

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* ψ_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

$$G = (N(G), A(G), \psi_G) \quad (2.1)$$

onde

$$N(G) = n_1, n_2, n_3, n_4, n_5 \quad (2.2)$$

e ψ_G é definido por

$$\begin{aligned} \psi_G(a_1) &= n_1n_2, \psi_G(a_2) = n_2n_3, \psi_G(a_3) = n_3n_4, \psi_G(a_4) = n_2n_4 \\ \psi_G(a_5) &= n_2n_4, \psi_G(a_6) = n_4n_5, \psi_G(a_7) = n_2n_5, \psi_G(a_8) = n_2n_4 \end{aligned} \quad (2.3)$$

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.

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

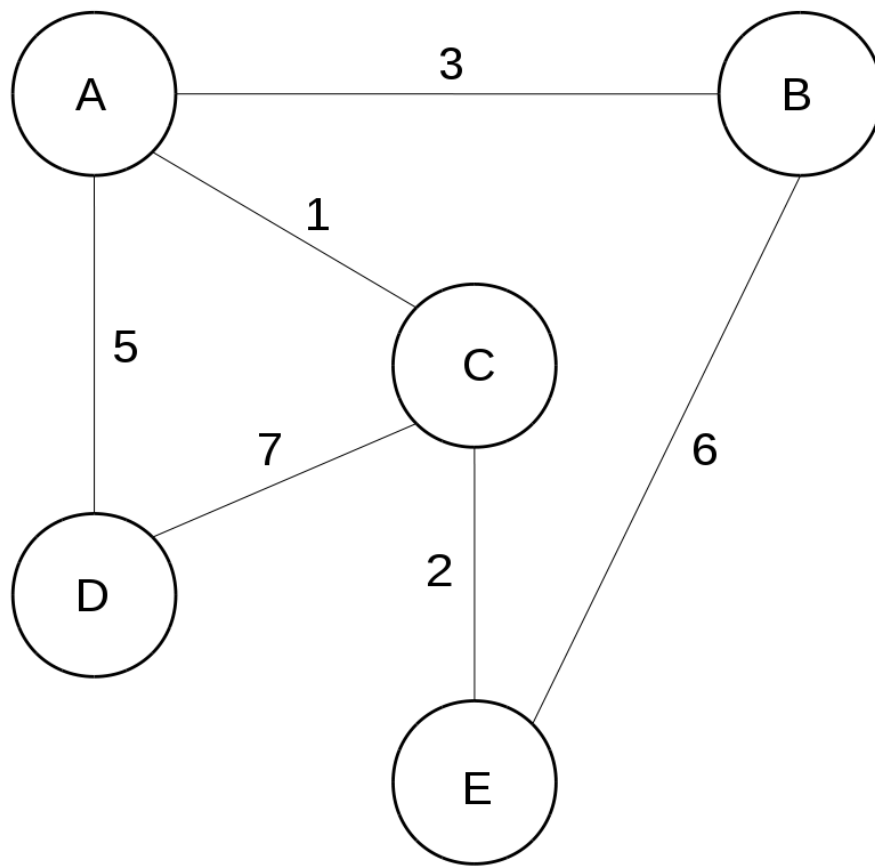


Figura 2.1: Um grafo com 5 nós e 6 arestas.

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

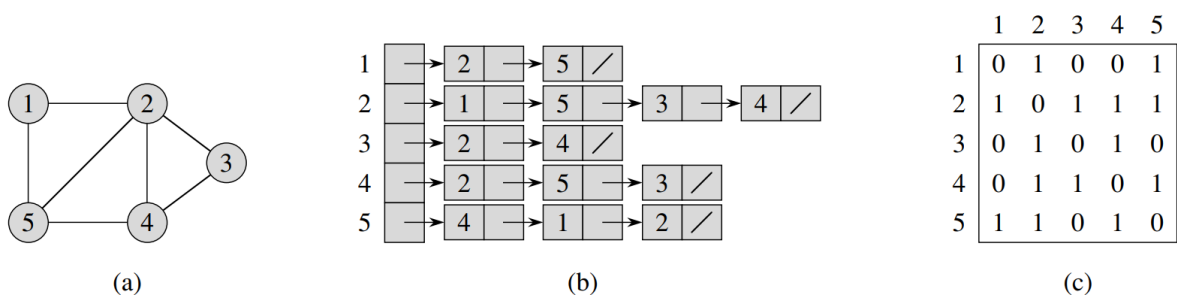


Figura 2.2: Duas representações de um grafo não direcionado. (a) Um grafo G com cinco nós e sete arestas. (b) A representação de uma lista de adjacência de G . (c) A representação de uma matriz de adjacência. Fonte: CORMEN et al. (2009), tradução dos autores.

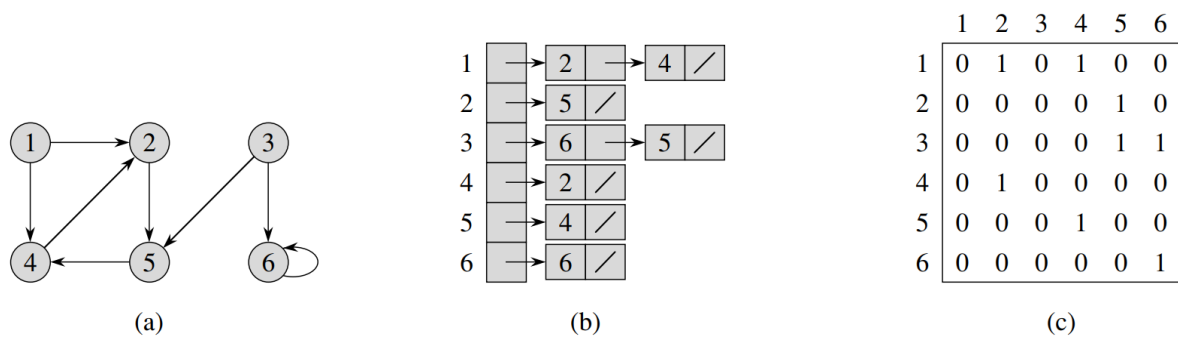


Figura 2.3: Duas representações de um grafo direcionado. **(a)** Um grafo G com seis nós e oito arestas. **(b)** A representação de uma lista de adjacência de G . **(c)** A representação de uma matriz de adjacência. Fonte: CORMEN et al. (2009), tradução dos autores.

A figura 2.2, ilustra um grafo G como um diagrama, como uma lista de adjacência e como uma matriz de adjacência. A implementação computacional de um grafo pode ser a partir de uma lista de adjacência. Desse modo Um grafo $G = (N, E)$ consiste de uma lista de listas de nós, uma lista para cada nó em N . Para cada nó em G , tem-se uma lista dos nós vizinhos. Essa lista também pode conter somente um ponteiro para os nós vizinhos. Normalmente é suficiente armazenar os nós vizinhos em ordem arbitrária pois será o peso da aresta que determinará uma hierarquia entre os nós vizinhos, não a ordem pela qual eles foram adicionados nem qualquer outro atributo.

3 Materiais e Métodos

3.1 Método de desenvolvimento

O produto foi desenvolvido pelo modelo gestão de projetos por 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*.

Na figura 3.1, estão as fases implementadas do modelo em cascata as quais foram aplicadas no desenvolvimento do sistema. Note-se iteração das fases de desenvolvimento e verificação, que representa a recorrência da atividade de implementar uma funcionalidade e imediatamente testá-la. Apesar de haver iteração não foi empregado método ágil de desenvolvimento uma vez que não se observou entrega de valor parcial.

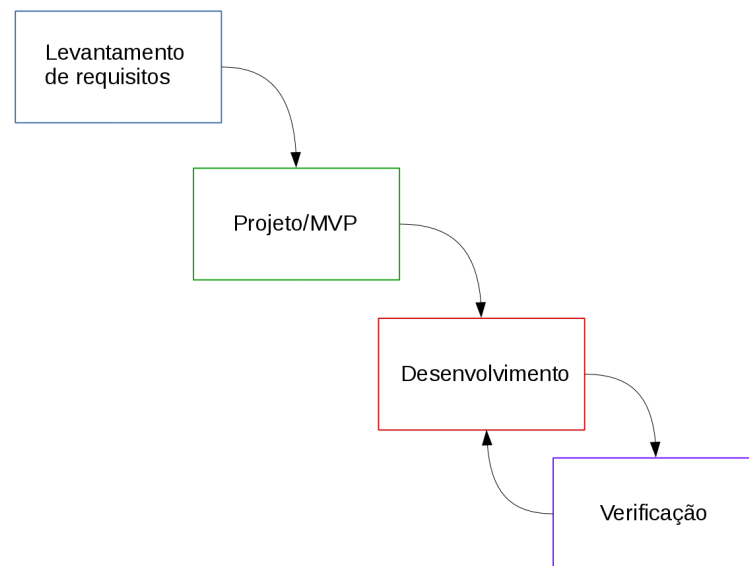


Figura 3.1: Fases empregadas do modelo em cascata. Fonte: os autores.

3.1.1 Análise de requisitos

Um MVP, foi desenvolvido durante a fase de levantamento de requisitos. A figura 3.2 apresenta a tela de criação de perguntas do MVP. Neste produto, é possível criar perguntas (3.2),

responder perguntas postadas por outros usuários (3.3) e marcar essas respostas (3.4). 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.

Figura 3.2: Tela de criação de perguntas do MVP. Fonte: os autores.

Figura 3.3: Tela de visualização de perguntas do MVP. Nesta tela é possível escolher uma pergunta para ser respondida. Fonte: os autores.

As figuras 3.2, 3.3 e 3.4 são as telas implementadas com as funções de criar questão, ver questão e ver respostas, respectivamente. Neste MVP o valor do K foi calculado com a seguinte equação:

$$k_{ab} = \frac{R_{ab} + R_{ba}}{P_a + P_b} \quad (3.1)$$

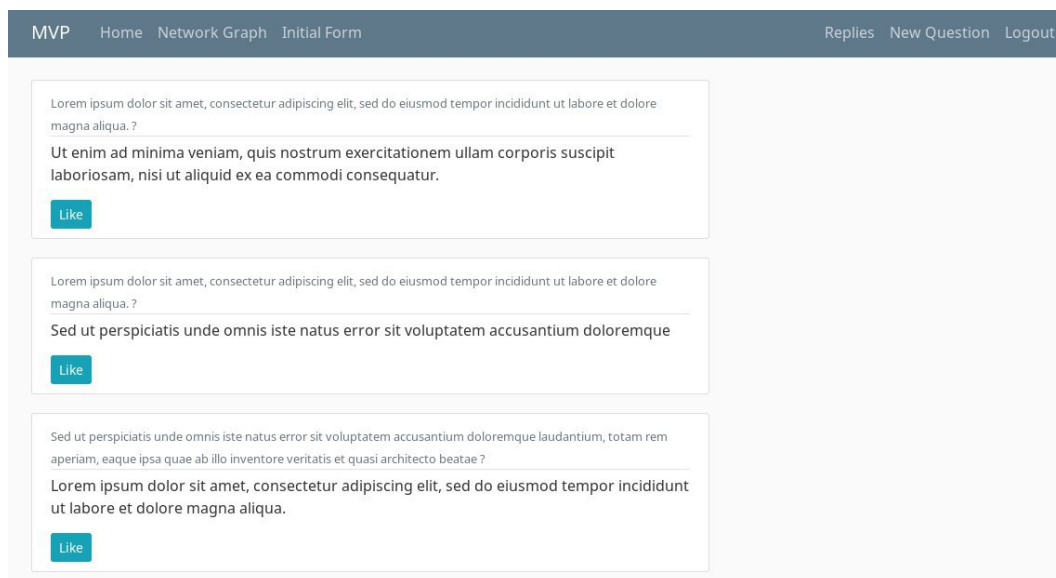


Figura 3.4: Tela de visualização de respostas recebidas do MVP. Nesta tela é possível marcar perguntas favoritas. Fonte: os autores.

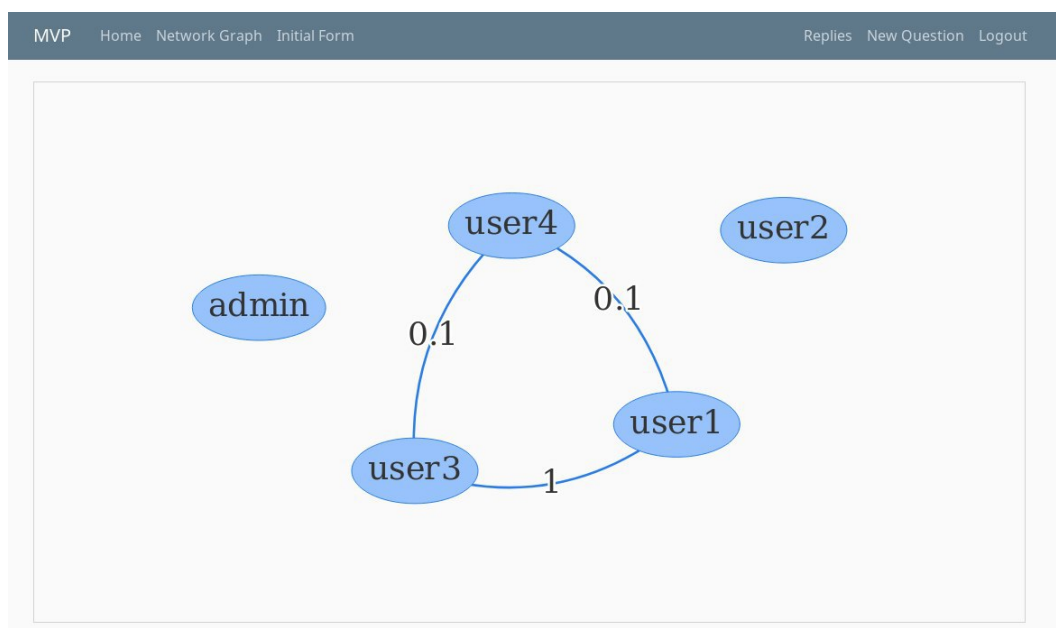


Figura 3.5: Visualização do grafo que representa as ligações entre usuários da rede social no MVP. Fonte: os autores.

Onde R_{ab} e R_{ba} são o número de respostas dadas pelo usuário A e pelo usuário B , respectivamente; e P_a e P_b são o número de perguntas criadas pelo usuário A e B , respectivamente.

A figura 3.5 é a tela do MVP que representa os usuários e suas conexões criadas na rede social por meio de um grafo. Esta tela mostra todos os usuários cadastrados na rede social como os nós do grafo e as respectivas conexões como as arestas. O valor mostrado nas arestas é nível de afinidade entre os usuários.

Depois do MVP, as telas principais foram prototipadas com o uso do software Pencil. Todas as telas que seriam implementadas foram desenhadas na fase de levantamento de requisitos. Neste momento, foi decidido que a aplicação teria 5 telas principais, sejam elas:



Figura 3.6: Protótipo da tela de visualização de perguntas. Fonte: os autores.



Figura 3.7: Protótipo da tela de criação de perguntas. Fonte: os autores.

- Visualizar perguntas. Figura 3.6
- Escrever perguntas. Figura 3.7
- Escrever respostas. Figura 3.8
- Visualizar respostas. Figura 3.9
- Visualizar contatos. Figura 3.10



Figura 3.8: Protótipo da tela de criação de respostas para as perguntas apresentadas. Fonte: os autores.



Figura 3.9: Protótipo da tela de visualização das respostas dadas às perguntas produzidas pelo usuário. Fonte: os autores.

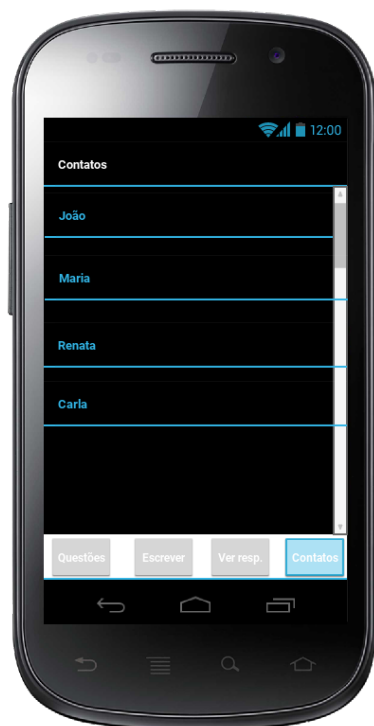


Figura 3.10: Protótipo da tela de visualização dos contatos. Fonte: os autores.



Figura 3.11: Protótipo da tela de mensagens instantâneas. Fonte: os autores.

Foi definida, a partir da análise dos requisitos, a seguinte ordem de desenvolvimento:

1. Desenhar telas
 - (a) Login
 - (b) Registro
 - (c) Criação de perguntas
 - (d) Visualização de perguntas
 - (e) Criação de respostas
 - (f) Visualização de respostas
2. Desenvolvimento do *back end*
 - (a) Criação do banco de dados
 - (b) Criação de perguntas
 - (c) Visualização de perguntas
 - (d) Criação de respostas
 - (e) Visualização de respostas

3.2 Arquitetura

O software foi concebido para ter capacidade multi-plataforma. Desta maneira, ele pode ser acessado via *browser*, aplicação Android ou IOS. A aplicação, ou página web, faz requisições para o servidor via protocolo http.

Na figura 3.12, há a arquitetura geral da aplicação e o diagrama de pacotes na mesma ilustração.

Neste conceito, o cliente faz uma requisição ao servidor e fornece o seu *token* de identificação. Os dados inseridos pelo usuário são, a partir da interface, transportados pelas classes do pacote Models para o pacote Page respectivo, dependendo do tipo de requisição. O pacote Pages chama uma classe do pacote Services que fará a requisição ao servidor no modelo que é compreendido pela API Controller. Este pacote usa Models do lado do servidor para requisitar do banco de dados as informações solicitadas inicialmente pelo cliente. Os dados são transformados pelo pacote Serializer que os transforma em um objeto JSON que é enviado novamente ao cliente.

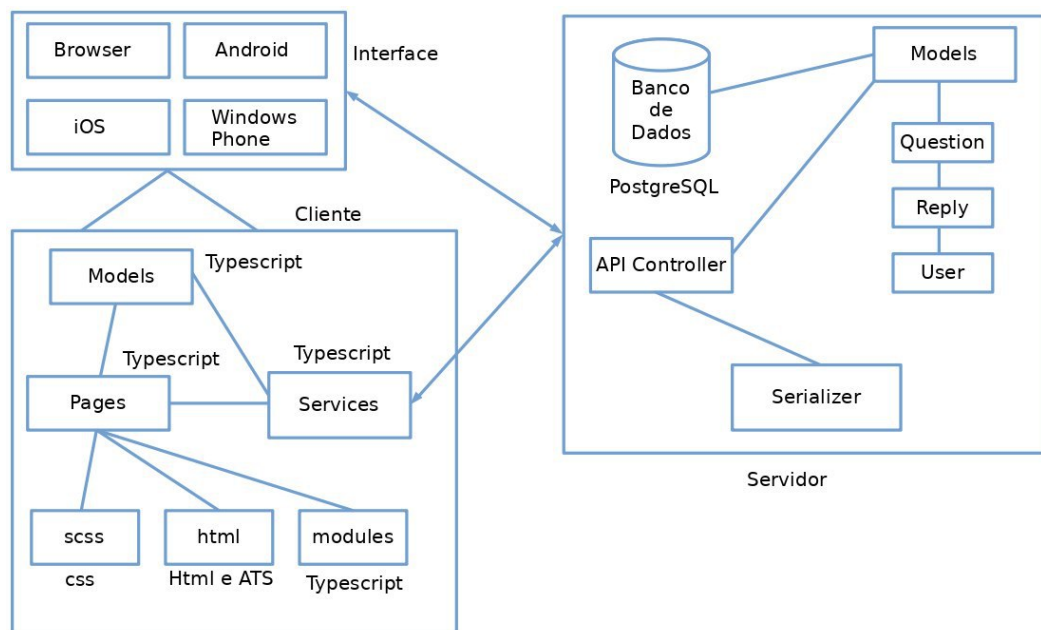


Figura 3.12: Diagrama de implementação. Fonte: os autores.

3.3 Tecnologia aplicada

O software foi implementado usando o *framework* Ionic (LUCAS e WIEGERT, 2019), para o *front end* e o Django (CHENG et al., 2019), para a camada de acesso aos dados. Essa decisão facilitou o desenvolvimento da interface, uma vez que o Ionic tem boas funcionalidades para esse fim, e possibilitou uma rápida integração com o servidor a partir do Django.

Do lado cliente, o *framework* Ionic utiliza as linguagens Typescript, JavaScript, HTML e CSS. No servidor a linguagem utilizada é Python e o sistema gerenciador de banco de dados é o PostgreSQL.

O Ionic é um *framework* de desenvolvimento multi-plataforma, o que possibilita o desenvolvimento de aplicações que podem ser utilizadas em equipamentos com sistema operacional Android, IOS ou os navegadores Chrome, Safari, Edge, Firefox e Internet Explorer 11 ou superior. A escolha por este *framework* para o desenvolvimento do software neste trabalho é baseada, essencialmente, na facilidade de criar uma aplicação multi-plataforma e no fato de ser uma ferramenta *opensource*. Tendo em vista que nenhum dos desenvolvedores do grupo tinha experiência em desenvolvimento *mobile*, a decisão pelo Ionic mostrou-se acertada pois a curva de aprendizado é evidentemente acentuada no início da utilização. Outra vantagem clara comparativamente ao desenvolvimento Android nativo, é que o *framework* requer consideravelmente menos recurso do que um SDK como o Android Studio, por exemplo.

O Django foi utilizado neste trabalho para o desenvolvimento da camada de acesso aos dados. É completamente desenvolvido na linguagem de programação *Python* sob o padrão de arquitetura *Model-View-Controller*. A preferência pelo Django também está relacionada ao fato

deste ter código aberto. Além disso, os integrantes da equipe já tinham experiência no uso do Django e essa facilidade era uma conveniência que não podia ser desprezada.

O versionamento e o compartilhamento do desenvolvimento foi realizado como uso do GitHub.

O desenvolvimento foi realizado em uma estação de trabalho do modelo PC Notebook Thinkpad X201, cujas especificações são as seguintes.

- Disco: SSD 120GB.
- Memória RAM: 8GB
- Processador: Intel I5-540M (2.53GHz, 3MB Cache)
- Sistema Operacional: Arch Linux

O servidor é administrado por um serviço de VPS pago e tem as seguintes especificações:

- Memória RAM: 2GB
- Processador: Intel Haswell 2.3GHz
- Sistema Operacional: Debian Stretch

As máquinas virtuais que rodam a aplicação são administradas com o uso da ferramenta Docker.

3.4 Análise do sistema

O primeiro passo da análise do sistema foi a definição dos casos de uso. No Apêndice A estão os diagramas de caso de uso, os diagramas de sequência e as especificações dos casos de uso.

As classes definidas estão ilustradas na figura 3.13. A estrutura de dados que representa o usuário tem as relações com os seus vizinhos no grafo representadas por um *link* que carrega o nome do vizinho e o valor de k . Dessa maneira, temos classes definidas de modo que *User* está relacionado a *Neighbor* por uma lista de adjacência, uma vez que existe uma relação de muitos para muitos.

A figura 3.14 é o diagrama do Banco de Dados. Sabendo que o *framework* Django foi utilizado para desenvolver as conexões com o banco de dados e este utiliza o arquitetura *Model-View-Controller*, as entidades são muito semelhantes às classes, pois as classes definem os objetos do *model*.

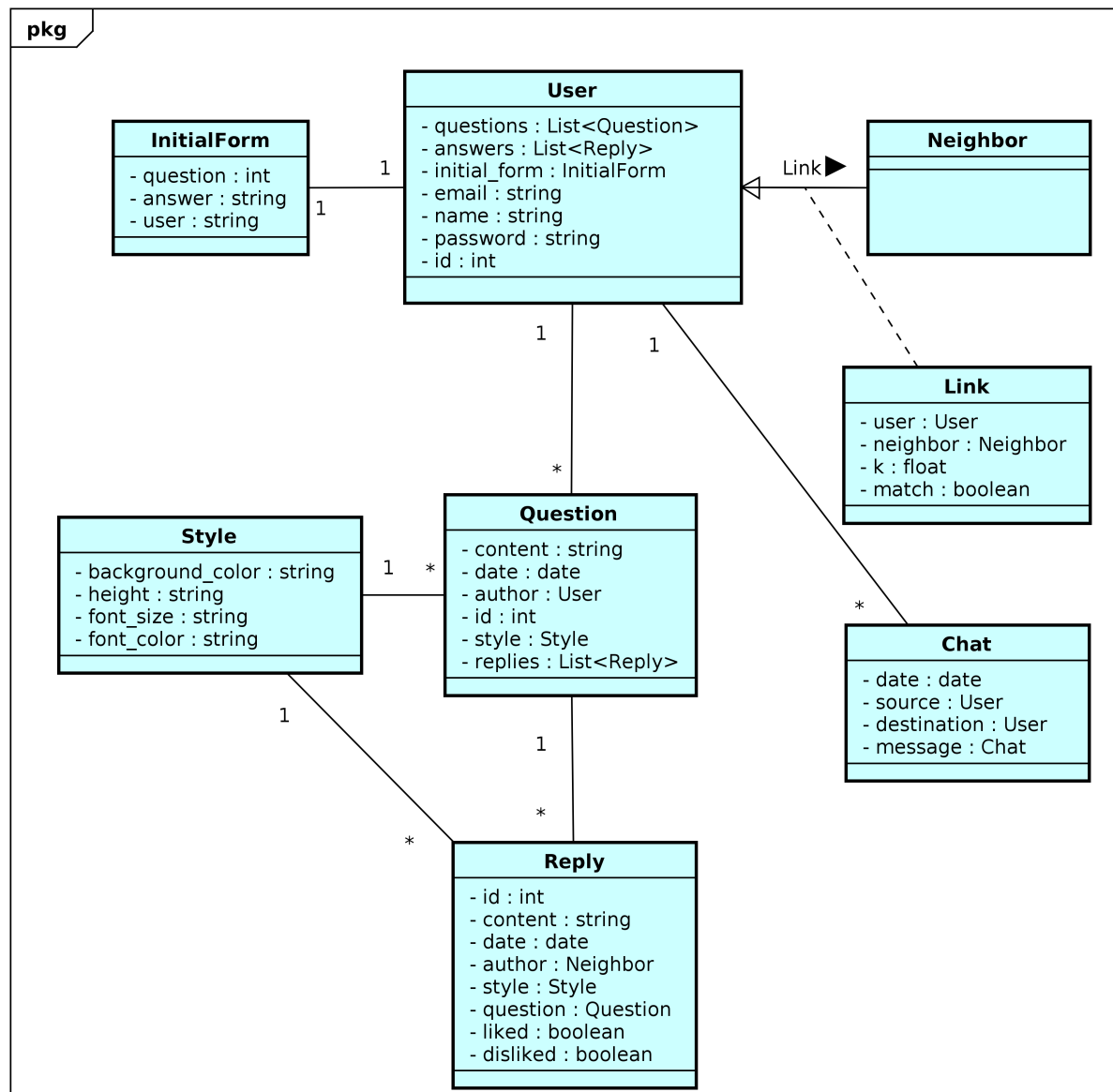


Figura 3.13: Diagrama de classes. Fonte: os autores.

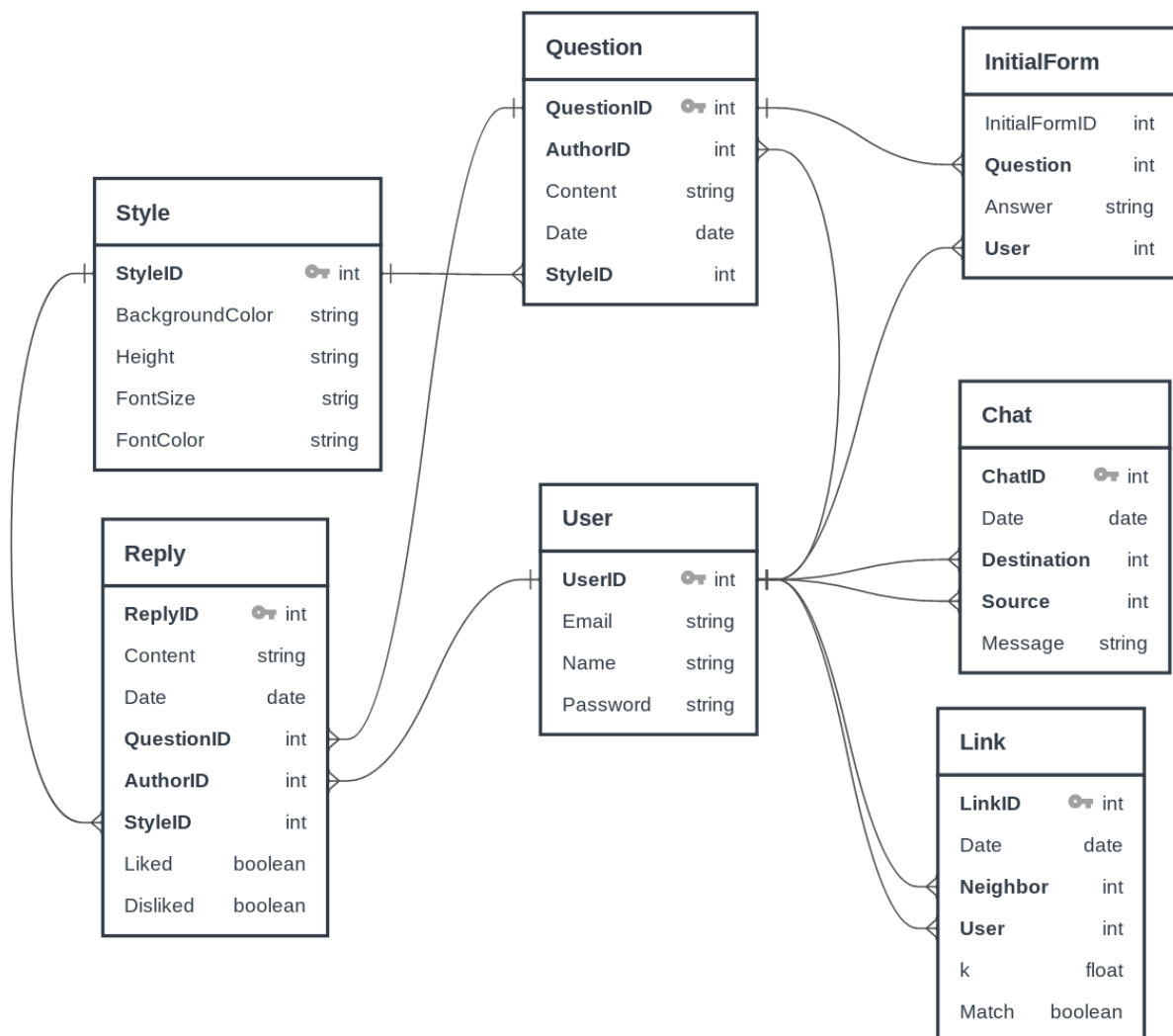


Figura 3.14: Diagrama de banco de dados. Fonte: os autores.

3.5 Cálculo 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?

As perguntas elaboradas para o formulário em 3.1, são simplificadas para não exaurir o usuário em sua primeira utilização do software. Essas perguntas podem ser melhor elaboradas a partir dos resultados obtidos ao longo da vida do software ou podem ser regionalizadas para atender melhor cada nicho de usuários. Como o objetivo deste trabalho não é fazer a análise sociológica dos interesses dos usuários, não foi despendido esforço maior para otimizar essas questões.

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:

$$k_{ab} = \frac{R_{ab} * (x - 1)}{N_p} \quad (3.2)$$

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 3.2, 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, o valor do peso da aresta, definido como k_{ab} , que liga os usuários A e B , é recalculado, para cada resposta dada, a partir da seguinte definição:

$$k_{ab} = k_{ab} + \frac{R_{ab} + R_{ba}}{P_A + P_B} \quad (3.3)$$

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

Na figura 3.15, podemos ver a evolução do peso k para cada resposta recebida pelo usuário A . Nesta situação o crescimento do valor de k é linear, pois todas as perguntas postadas pelo usuário A estão respondidas por B .

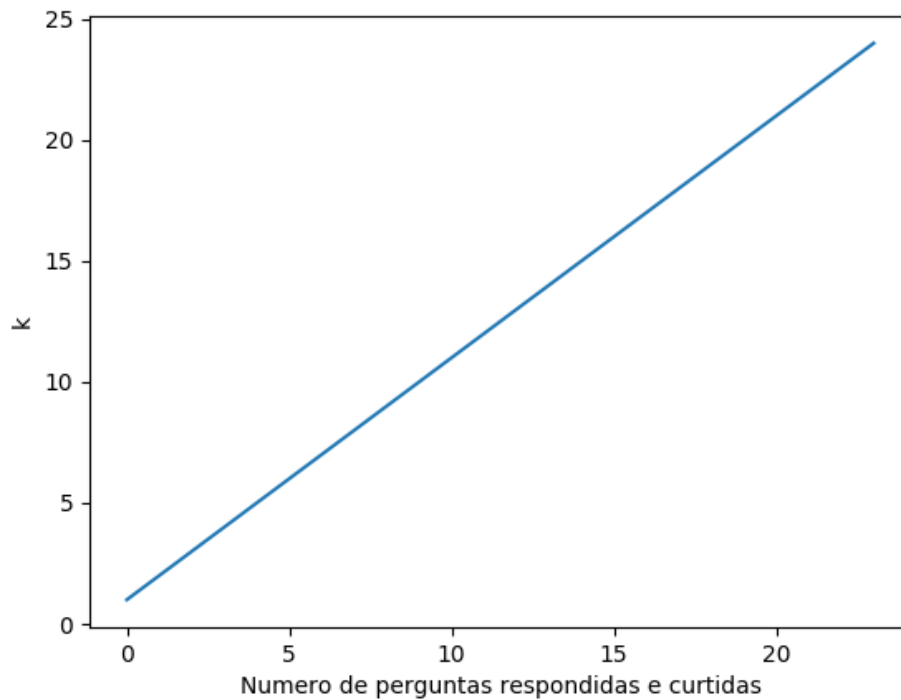


Figura 3.15: Evolução de k se todas as perguntas postadas por A forem respondidas por B . Fonte: os autores.

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 3.16.

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

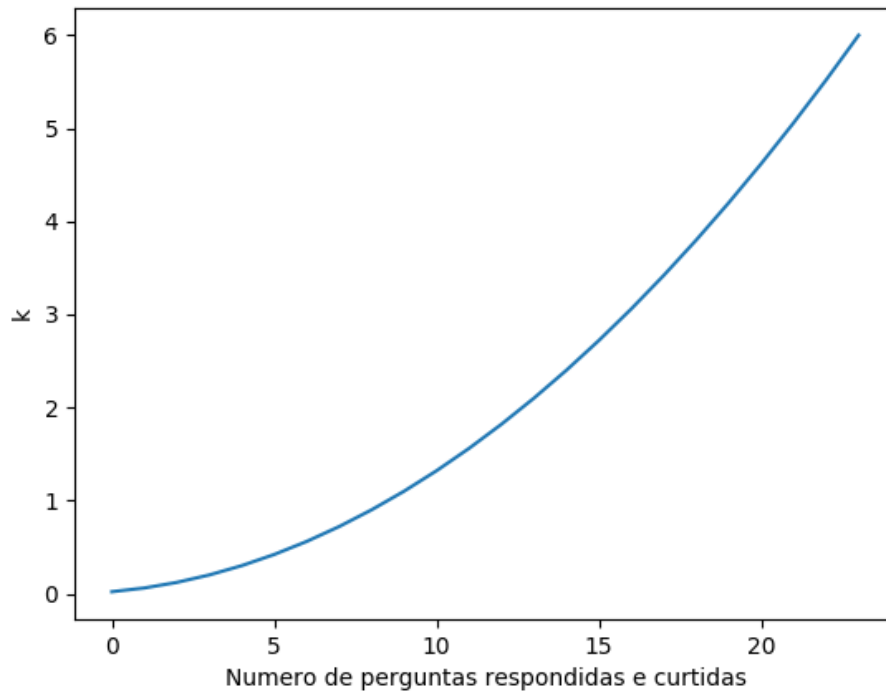


Figura 3.16: Evolução de k se A postou 50 perguntas e houve apenas 25 respostas de B. Fonte: os autores.

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.

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 3.17 mostra que a convergência para $x = 1$ para o caso de de dois usuários que postaram poucas perguntas é rápida, pois o incremento de k é maior. No caso simulado na figura 3.17, pouco mais de cinco respostas foram o suficiente para atingir o x .

A figura 3.18 demonstra que a convergência de k e x para usuários que já postaram muitas perguntas demanda muito mais respostas, neste caso mais que vinte respostas. Isto deve-se ao fato de ter sido decidido que a equação 3.3 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 trocadas entre dois usuários. Assim, x seria o número mínimo de respostas que teriam que ser respondidas entre os usuários para que fossem considerados similares o suficiente para contatarem-se pelo mensageiro instantâneo. Porém, dessa maneira ignora-se que, se houverem

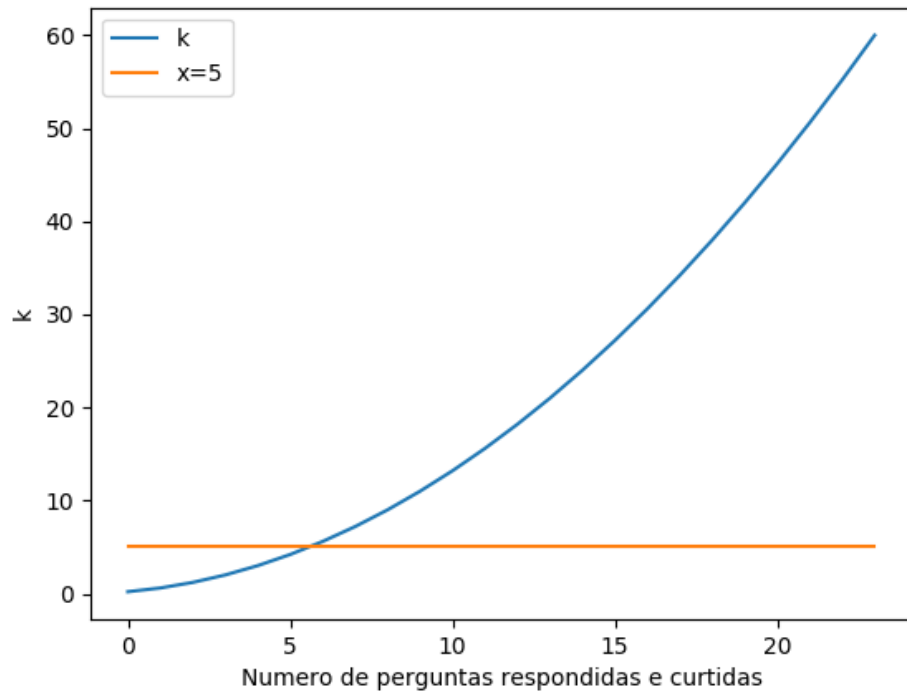


Figura 3.17: Evolução de k em função das respostas curtidas se A e B têm poucas perguntas postadas. Ex.: menos que 10 perguntas. Fonte: os autores.

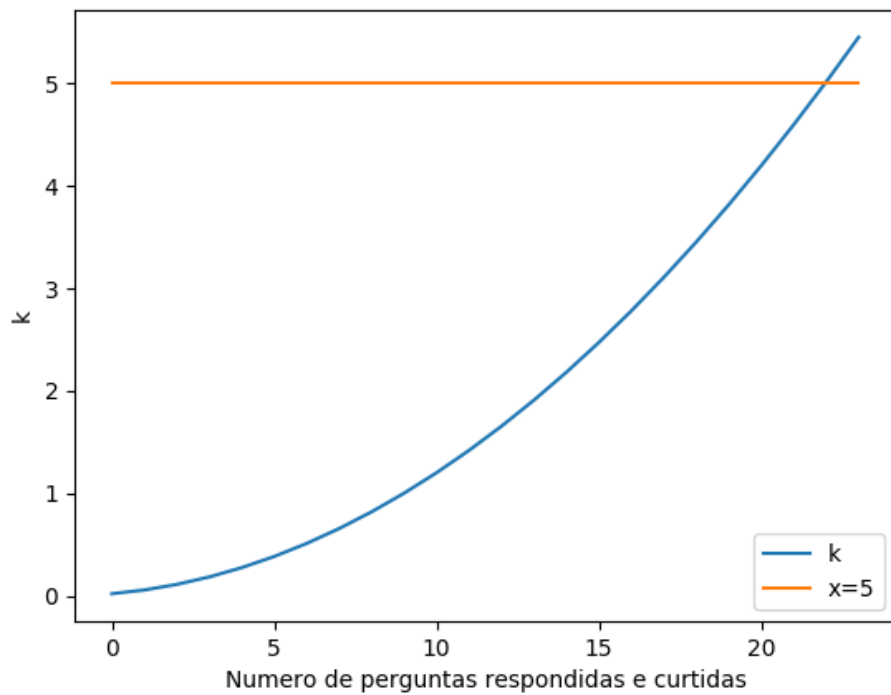


Figura 3.18: Evolução de k em função das respostas se A e B têm muitas perguntas postadas. Ex.: mais que 50 perguntas. Fonte: os autores.

muitas perguntas postadas por A sem resposta de B é possível que hajam mais diferenças do que similaridades entre os usuários, haja vista que muitas perguntas são ignoradas.

Portanto, é imprescindível considerar o volume de perguntas postadas e também contabilizar o número de respostas que foram apreciadas pelo usuário que postou a pergunta. Dessa maneira, propomos uma evolução que adequasse o software para contabilizar o número de perguntas postadas por A e vistas por B , bem como quais respostas cada parte gostou de receber. Assim, poderíamos adequar o cálculo de k como segue:

$$k = k + \frac{R_{ab} + R_{ba}}{P_{AvB} + P_{BvA}} \quad (3.4)$$

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 3.4, 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, neste caso, 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 cronograma do desenvolvimento.

Como medida para contornar a dificuldade de implementar um contador para as perguntas que foram vistas por usuários específicos, foi tomada uma aproximação relacionada somente às perguntas respondidas e às respostas apreciadas. A função a seguir define o cálculo de k com essa premissa:

$$k_{ab} = \frac{R_{abA} + R_{baA}}{R_{ab} + R_{ba}} + \frac{R_{ab} + R_{ba}}{P_a + P_b} \quad (3.5)$$

Onde R_{abA} e R_{baA} são o número de respostas apreciadas pelo usuário A e B , respectivamente; R_{ab} é o número de respostas que A enviou a B ; R_{ba} é o número de respostas que B enviou a A e, finalmente, P_a e P_b o número total de perguntas enviadas por A e B .

Podemos observar na figura 3.19 que, aplicando a função 3.5, temos uma correlação entre o número de perguntas respondidas e o número de perguntas apreciadas pelo usuário o que representa uma linearidade desejável no incremento do valor de k proporcionalmente ao número de perguntas que cada usuário respondeu e cada resposta recebida que ele apreciou.

Dessa maneira, o valor de k cresce tanto para perguntas respondidas quanto para respostas apreciadas. Espera-se um incremento maior para cada resposta apreciada pois o divisor

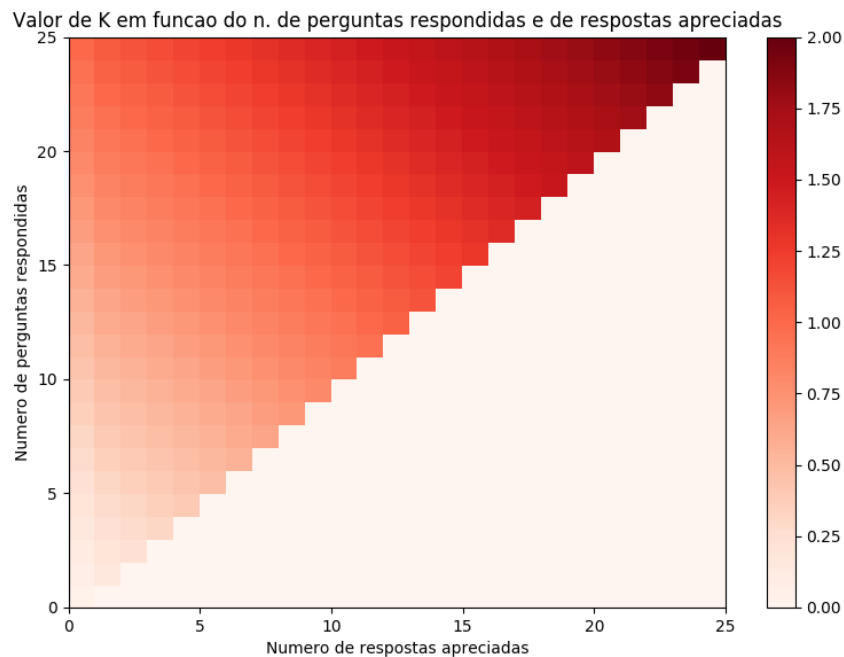


Figura 3.19: Valor de k em função do número de perguntas respondidas e de respostas curtidas entre dois usuários.
Fonte: os autores.

deste elemento é o número total de respostas, enquanto para cada resposta sem apreço o divisor é o número total de perguntas.

Dentro deste caso, um valor fixo para x , determina um número de respostas e respostas apreciadas entre os usuários proporcional ao número total de perguntas. Nas figuras 3.20 e 3.21, percebemos que a região que possui k maior que $x = 1, 5$, por exemplo, tem a mesma área para casos nos quais o número total de perguntas postadas entre dois usuários é 25, que é o caso da figura 3.20, e 200, que é o caso da figura 3.21. Isso implica que o valor número de perguntas necessárias para atingir x é diretamente proporcional ao número total de perguntas postadas pelos dois usuários.

A relação entre o número de perguntas postadas e o peso de cada resposta apreciada no cálculo de k é considerada uma funcionalidade interessante para o caso de usuários que já postaram muitas perguntas e encontram similares com facilidade. Consideramos essa ponderação adequada já que facilita a criação de contatos entre usuários recentes e que ainda postaram poucas perguntas e aumenta o desafio para quem já postou muitas perguntas e, possivelmente, já tem muitos contatos desenvolvidos.

Todavia, levando em conta que isto pode desanimar o usuário mais fiel, decidimos não considerar essa ponderação e aplicar uma função que leve em conta apenas o número total de

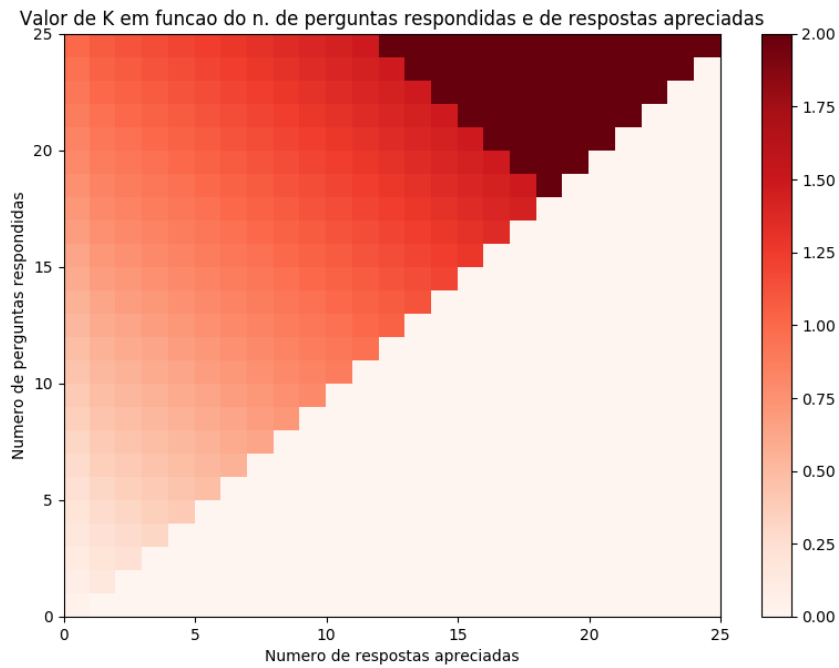


Figura 3.20: A área em vermelho escuro compreende a relação entre respostas e respostas apreciadas entre dois usuários onde k é maior que 1,5 para 25 perguntas respondidas. Fonte: os autores.

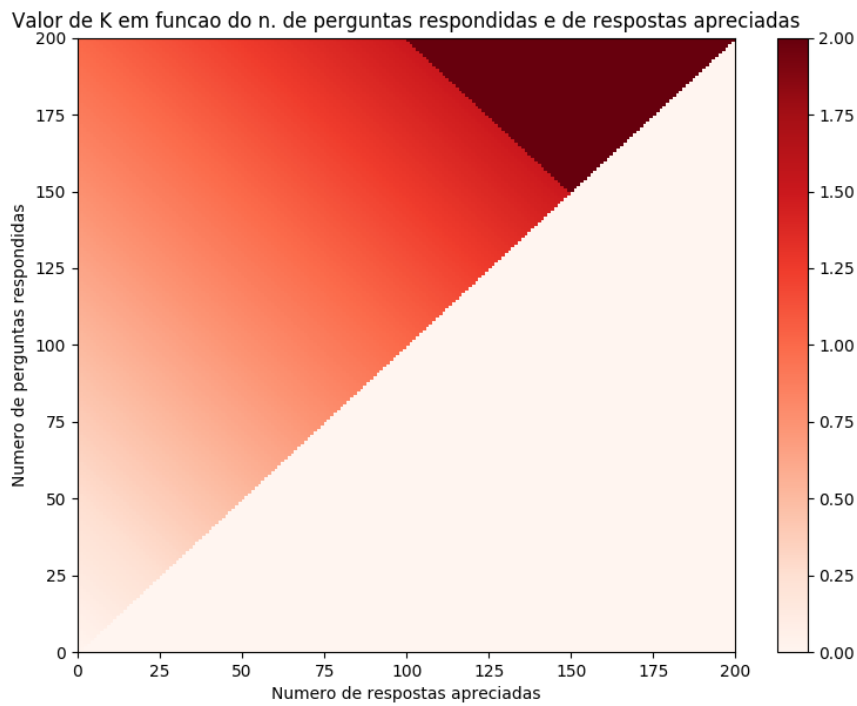


Figura 3.21: A área em vermelho escuro compreende a relação entre respostas e respostas apreciadas entre dois usuários onde k é maior que 1,5 para 200 perguntas respondidas. Fonte: os autores.

respostas e respostas apreciadas. Aplicamos um fator de multiplicação de cinco para aumentar a influência das respostas que foram apreciadas.

$$k_{ab} = (R_{abA} + R_{baA}) * 5 + R_{ab} + R_{ba} \quad (3.6)$$

Porém, considerar as respostas que não foram curtidas pode ser perigoso, uma vez que o usuário que recebe muitas respostas e não aprecia, pode realmente não estar gostando delas e isso culminaria numa proposta de contato desagradável.

Finalmente, foi considerado levar em conta apenas as respostas apreciadas pela razão do número total de respostas. Dessa maneira uma resposta sem apreço é considerada um decremento na força da similaridade calculada pela interação.

$$k_{ab} = (R_{abA} + R_{baA}) - \frac{R_{ab} + R_{ba}}{w} \quad (3.7)$$

Onde w é uma constante que representa o peso dado para o decremento afetado pelo número de perguntas que não foram apreciadas.

Essa função, apesar de simples e elegante, parece ser a mais adequada para a resolução do problema. A equação 3.7 considera coerentemente as informações depreendidas do sistema e estabelece uma curva de crescimento apreciável para k , independentemente de quantas perguntas já foram postadas, como pode-se observar na figura 3.22.

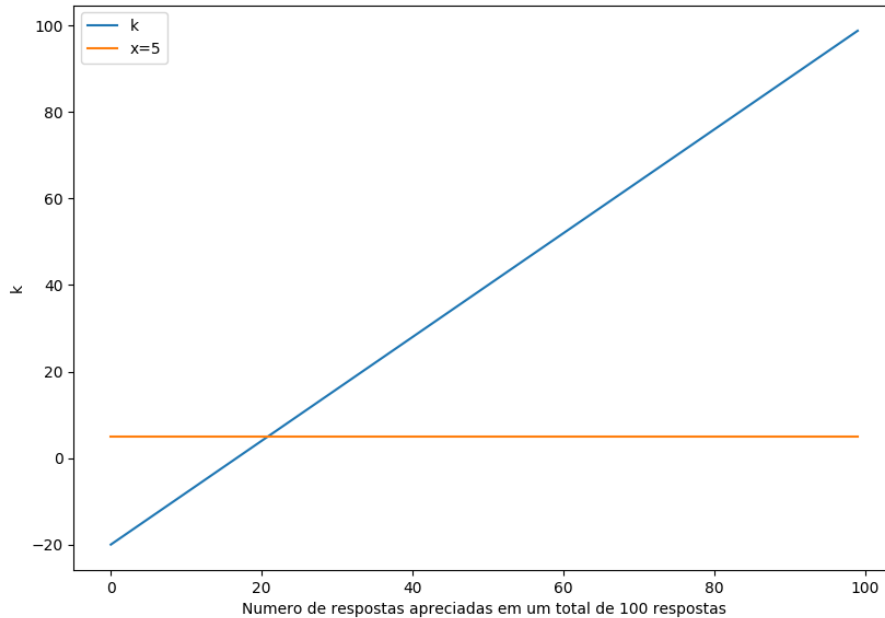


Figura 3.22: Crescimento de k de acordo com a função 3.7 onde $w = 5$ e com um limite de x estabelecido em 5. Fonte: os autores.

Os valores de w e x foram estabelecidos considerando um relacionamento mínimo que deve haver para que não seja banalizada a oferta de contatos e para que o usuário mantenha-se

interessado no sistema. Esses valores podem ser adaptados por região, datas comemorativas ou campanhas específicas de confraternização que podem ser oferecidas para manter a fidelização.

Outras opções de resolução do problema de sugestão de contatos, poderiam envolver inteligência artificial, algoritmos de classificação ou algoritmos de recomendação.

Um refinamento valioso para a acurácia das recomendações de contatos poderiam envolver o uso de inteligência artificial para fazer uma análise semântica das perguntas e respostas. Dessa maneira, poderíamos depreender um volume muito maior de informações, tanto das perguntas quanto das respostas, a partir de palavras-chave, por exemplo, de maneira que um banco de dados muito mais rico definiria cada usuário. A partir desses dados, um algoritmo de recomendação baseado em filtragem de conteúdo poderia aproximar usuários que costumam escrever sobre assuntos correlatos.

A análise do texto poderia propiciar uma classificação dos usuários a partir de seus interesses e aproximá-los de perguntas que seja mais coerentes com seus gostos. Isso otimizaria a dinâmica de respostas e ofereceria uma experiência mais agradável ao usuário, de maneira que ele sentiria-se familiarizado com as perguntas uma vez que elas seriam selecionadas de acordo com seu interesse.

4 Apresentação do Sistema

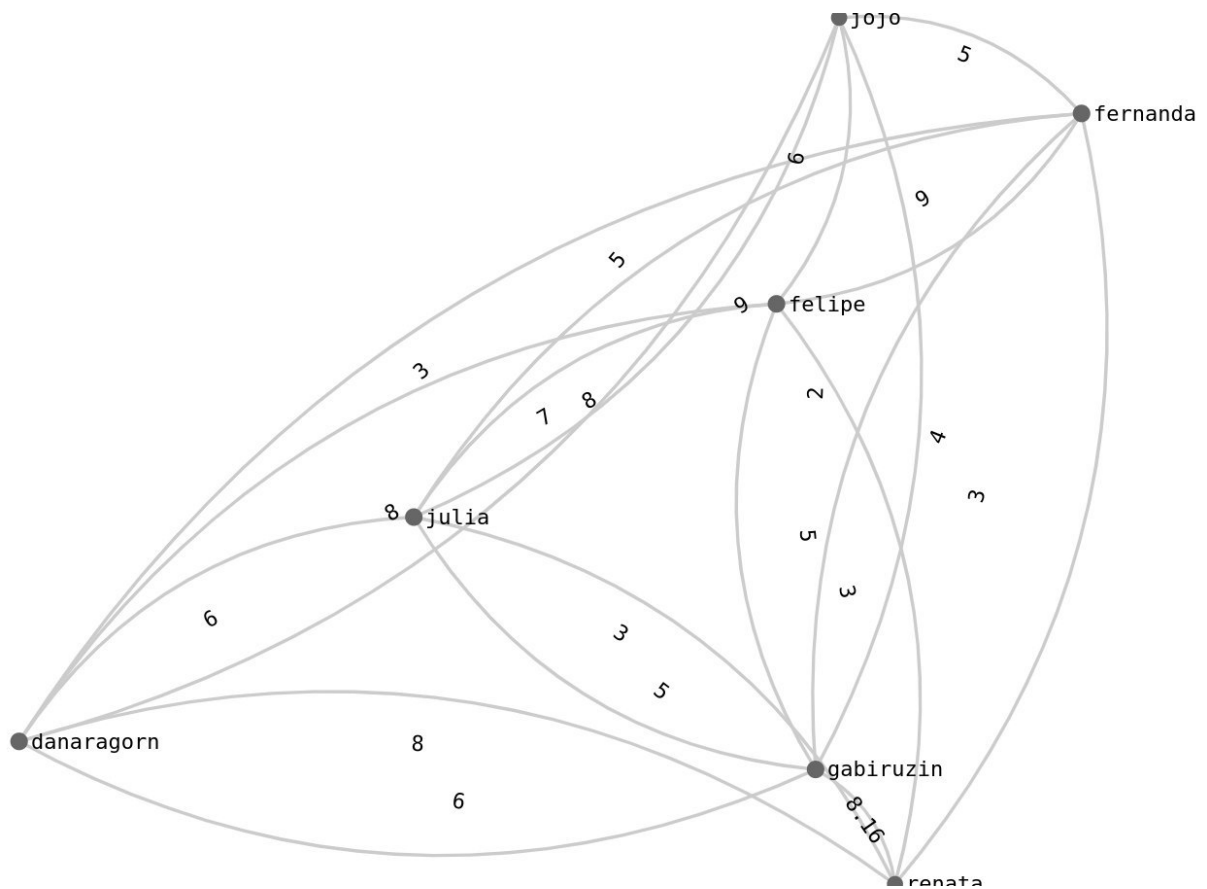


Figura 4.1: Grafo representando os usuários do sistema e suas relações. Fonte: os autores.

O sistema pronto implementou todas as funcionalidades previstas na análise de requisitos. Contou também com a opção de gerar uma imagem do grafo com todos os usuários cadastrados. Na figura 4.1, podemos visualizar um grafo gerado pelo software com os usuários registrados até o momento e as respectivas arestas representando o grau de afinidade entre dois nós. A geração desta imagem não é uma funcionalidade do aplicativo, apenas uma ferramenta desenvolvida para ter funcionalidade administrativa e científica.

Na implementação foi aplicada a seguinte função para k :

$$k_{ab} = (R_{abA} + R_{baA}) - \frac{R_{ab} + R_{ba}}{w} \quad (4.1)$$

Onde w foi estabelecido como sendo igual a 5 e x igualmente 5.

A apresentação do sistema foi dividida em duas partes com o intuito de acompanhar o caminho do usuário ao utilizar. Dessa maneira as seções são as seguintes:

- Telas de Perguntas e Respostas
- Tela do mensageiro instantâneo

4.1 Perguntas e respostas

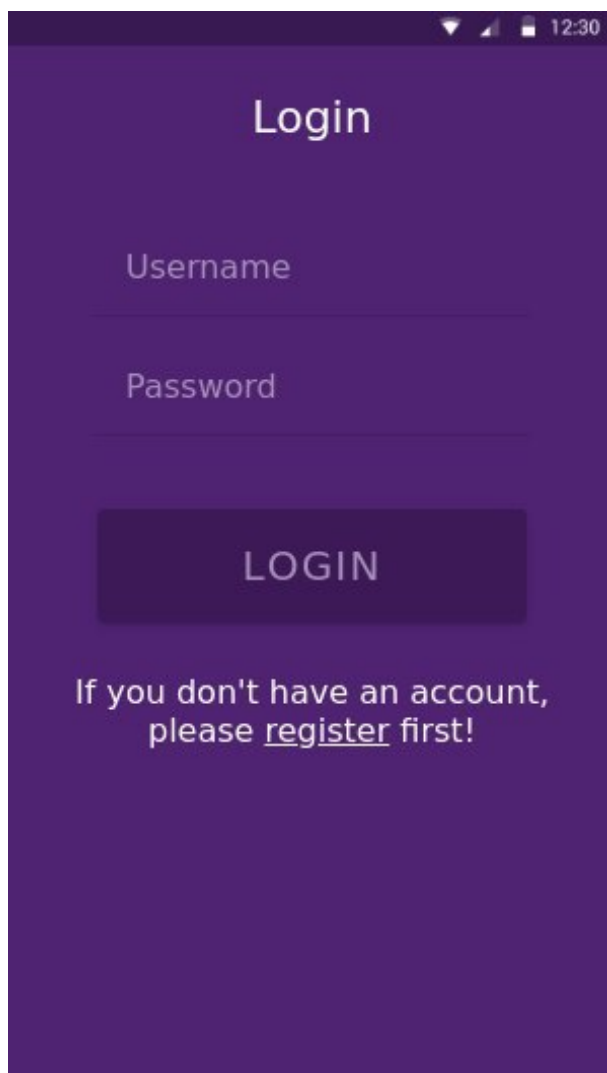
A figura 4.2 é a tela de login do sistema. As cores a partir daqui seguem o mesmo padrão. Foi escolhida uma cor que pudesse ser afirmada como a representação da identidade do sistema. Assim, a cor é memorável por ser incomum. Tendo em vista que o público esperado para o sistema seja majoritariamente jovem, supõe-se que deve ser bem recebida a excentricidade da cor.

A tela de registro pode ser vista na figura 4.3, segue o mesmo padrão da tela de login, portanto, com a mesma identidade visual.

A tela de visualização de questões pode ser vista na figura 4.4. As cores do balão que contém o texto da pergunta podem ser escolhidas pelo usuário que criou a pergunta. Essas cores foram selecionadas para combinarem integralmente com a cor roxa das barras do aplicativo.

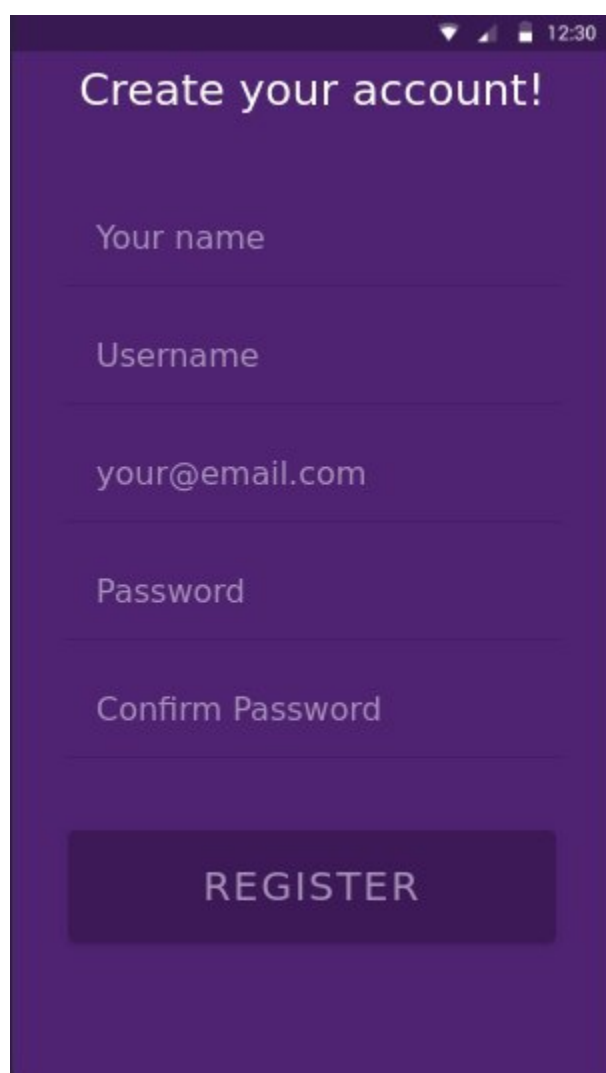
A cor do fundo da questão pode ser escolhida pelo botão no canto inferior direito da tela de criação de questão, como pode ser observado na figura 4.5. Com um toque no botão as cores mudam cíclicamente.

A mesma função de mudar o fundo da tela está presente na tela de criação de respostas, vide figura 4.6. A cor escolhida para a resposta pode ser visualizada pelo usuário que recebeu a resposta, como podemos observar na figura 4.7, que é a tela de visualização de respostas.



A mobile app login screen with a dark purple background. At the top, the status bar shows signal, Wi-Fi, and battery icons along with the time 12:30. The title 'Login' is centered in white. Below it are two white input fields labeled 'Username' and 'Password'. A large, dark purple button with the text 'LOGIN' in white is centered below the fields. At the bottom, white text reads: 'If you don't have an account, please [register](#) first!'.

Figura 4.2: Tela de login. Fonte: os autores.



A mobile app screen for creating an account with a dark purple background. The status bar at the top shows signal, Wi-Fi, and battery icons along with the time 12:30. The title 'Create your account!' is centered in white. Below it are five white input fields labeled 'Your name', 'Username', 'your@email.com', 'Password', and 'Confirm Password'. A large, dark purple button with the text 'REGISTER' in white is centered at the bottom.

Figura 4.3: Tela de registro. Fonte: os autores.



Figura 4.4: Tela de visualização de questões. Fonte: os autores.



Figura 4.5: Tela de criação de questões. Fonte: os autores.

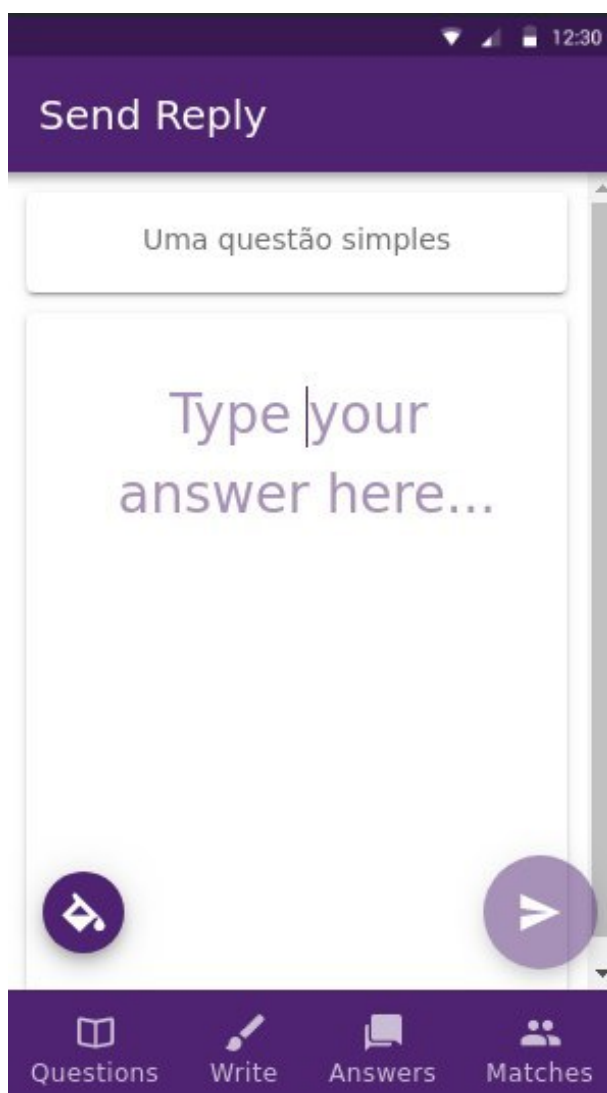


Figura 4.6: Tela de resposta para questões. Fonte: os autores.



Figura 4.7: Tela de visualização de respostas. Fonte: os autores.

4.2 Mensageiro instantâneo

Depois que os usuários interagem suficientemente o sistema decide criar uma relação entre eles, o contato sugerido aparece disponível no botão *matches*, no canto inferior direito da tela, como podemos observar na figura 4.8.

A partir desta tela, quando o usuário seleciona um contato é apresentada a tela de conversa, que podemos visualizar na figura 4.9.



Figura 4.8: Tela de visualização dos contatos criados. Fonte: os autores.

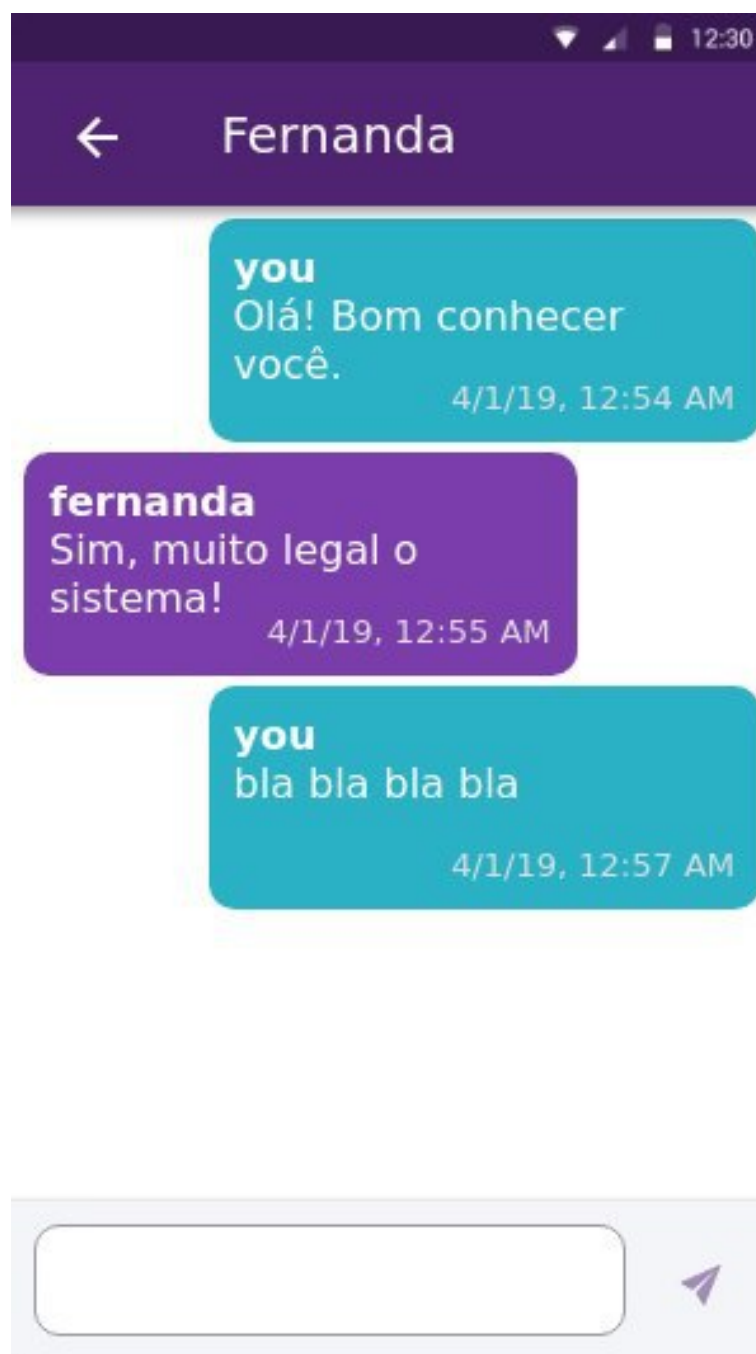


Figura 4.9: Tela de visualização das mensagens recebidas no mensageiro instantâneo. Fonte: os autores.

5 Considerações Finais

As interações sociais em ambiente virtual são parte do dia a dia de muitas pessoas no mundo todo. Muitas vezes as interações em ambiente virtual são tão frequentes ou até mais do que os relacionamentos presenciais. Por isso podemos considerar relevante qualquer esforço de melhoria da experiência dos usuários das redes sociais.

Muitos algoritmos são desenvolvidos para otimizar a experiência do usuário na internet, e as redes sociais estão inclusas nesse esforço pois contém informação, publicidade, serviços e contatos importantes para vida pessoal e profissional dos usuários.

Este trabalho buscou criar uma rede social na qual os usuários relacionam-se anonimamente por meio de troca de perguntas e respostas até que o sistema decida estabelecer o contato direto entre os usuários através de um mensageiro instantâneo. O desafio foi definir uma estrutura de dados e um método de cálculo de semelhança entre os usuários que fosse acuradamente agradável ao utilizador e demandasse de pouca carga de processamento.

Podemos afirmar que o objetivo foi atingido, uma vez que o software é completamente utilizável e apresenta a performance esperada. Os relacionamentos são criados a partir das interações iniciais e o grafo que representa a rede é atualizado corretamente a cada entrada de novas respostas.

A função que determina o valor da aresta que representa o nível de afinidade entre dois usuários é elegantemente simples e foi determinada como adequada após vários experimentos e descobertas ao longo do desenvolvimento do projeto. A solução simplificada garante bom desempenho e eficácia.

Além da satisfação de encontrar um bom resultado, é igualmente entusiasmante vislumbrar os novos caminhos que este sistema pode tomar para entregar seu valor com mais sofisticação. Podemos observar que há vastas possibilidades de emprego de métodos de extração de dados a partir do léxico dos usuários com aplicação de inteligência artificial e que, a partir destes dados, algoritmos de recomendação ou de classificação mais complexos poderiam ser empregados para melhorar a experiência do usuário e a compreensão das interações estabelecidas nas redes sociais.

Evidente que a ampliação da nossa esfera de conhecimento percebida durante o desenvolvimento deste trabalho nos coloca em contato com o desconhecido. Este contato acelera o aprendizado e lança luz à novas possibilidades que podem ser exploradas em trabalhos futuros com o ímpeto criado por este.

Referências

- BELL, R. M., KOREN, Y. e VOLINSKY, C. (2007). The bellkor solution to the netflix prize.
- BONDY, J. e MURTY, U. (1976). *Graph Theory*. Springer Publishing Company, Incorporated, 1st edition.
- CHENG, C., RAMEZANI, H. e GIBSON, C. (2019). Django project documentation. <https://docs.djangoproject.com/en/2.2/>. Acessado em 08/06/2019.
- 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.
- Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S. e Stettinger, M. (2014). *Basic Approaches in Recommendation Systems*, páginas 15–37.
- GOLDBERG, D., A. NICHOLS, D., M. OKI, B. e TERRY, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35:61–70.
- 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.
- LUCAS, E. e WIEGERT, C. (2019). Ionic - cross-platform mobile app development. <https://ionicframework.com/docs/intro>. Acessado em 08/06/2019.
- 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*.
- Netflix, I. (2006). Netflix prize. <https://www.netflixprize.com/>. Acessado em 15/06/2019.

- PATERSON, M. S. e YAO, F. F. (1992). On nearest-neighbor graphs. *Springer LNCS*, 623(1):416–426.
- RICCI, F., ROKACH, L., SHAPIRA, B. e KANTOR, P. B. (2010). *Recommender Systems Handbook*. Springer-Verlag, Berlin, Heidelberg, 1st edition.
- 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.

Apêndice A: Casos de uso

A.1 Diagramas de Caso de uso

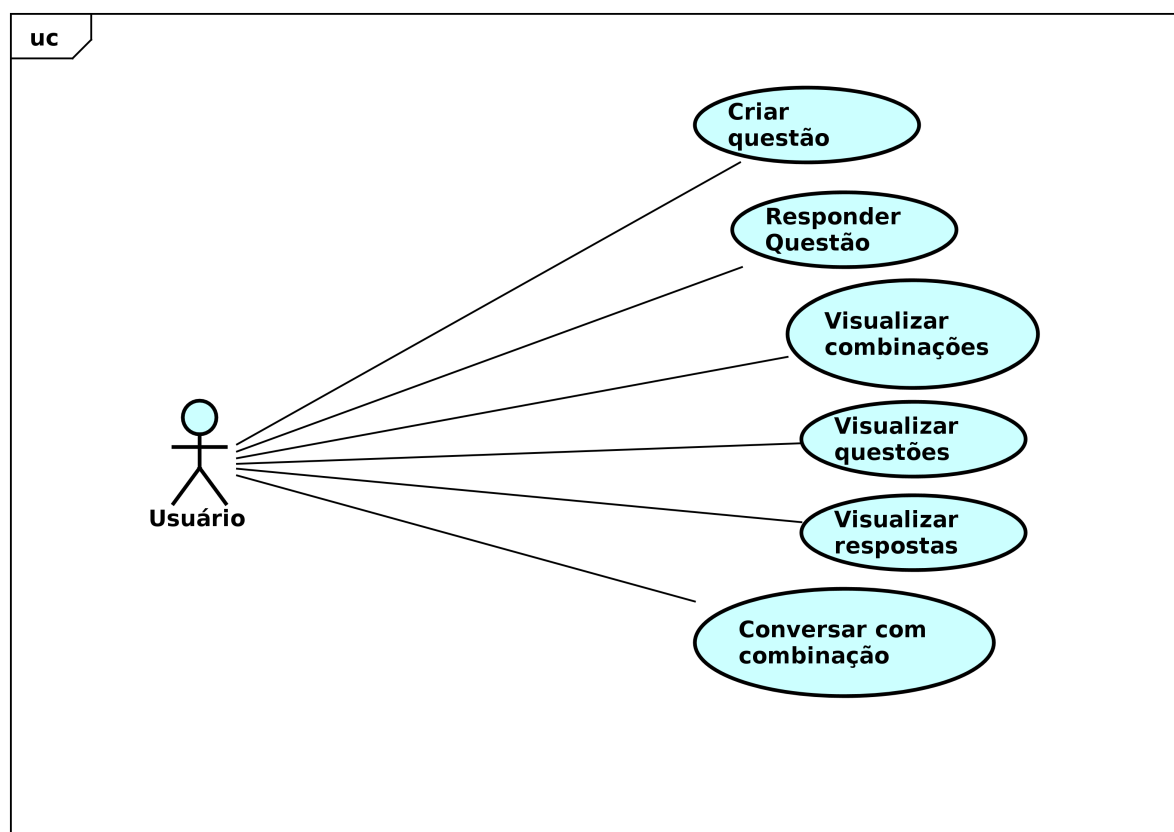


Figura A.1: Diagrama de caso de uso nível 1. Fonte: os autores.

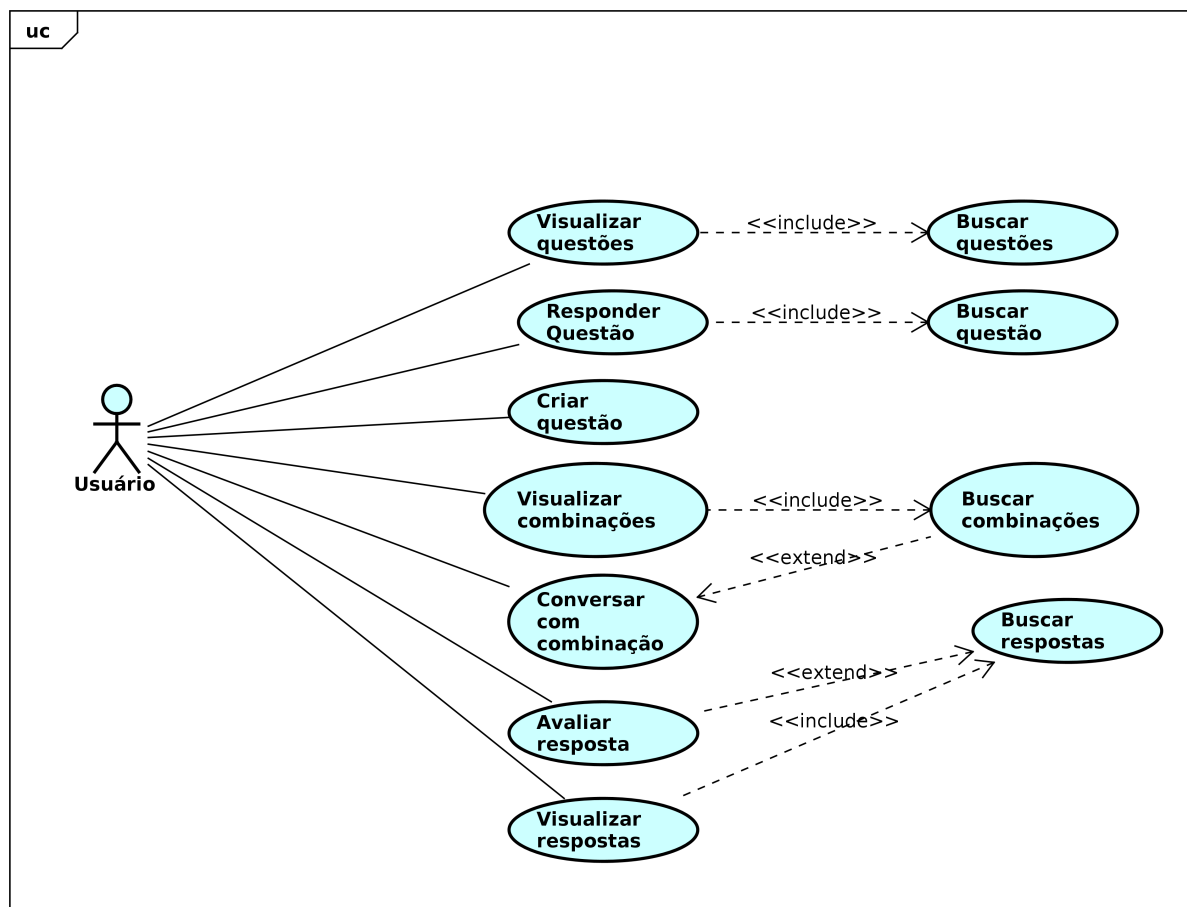


Figura A.2: Diagrama de caso de uso nível 2. Fonte: os autores.

A.2 Diagramas de sequência

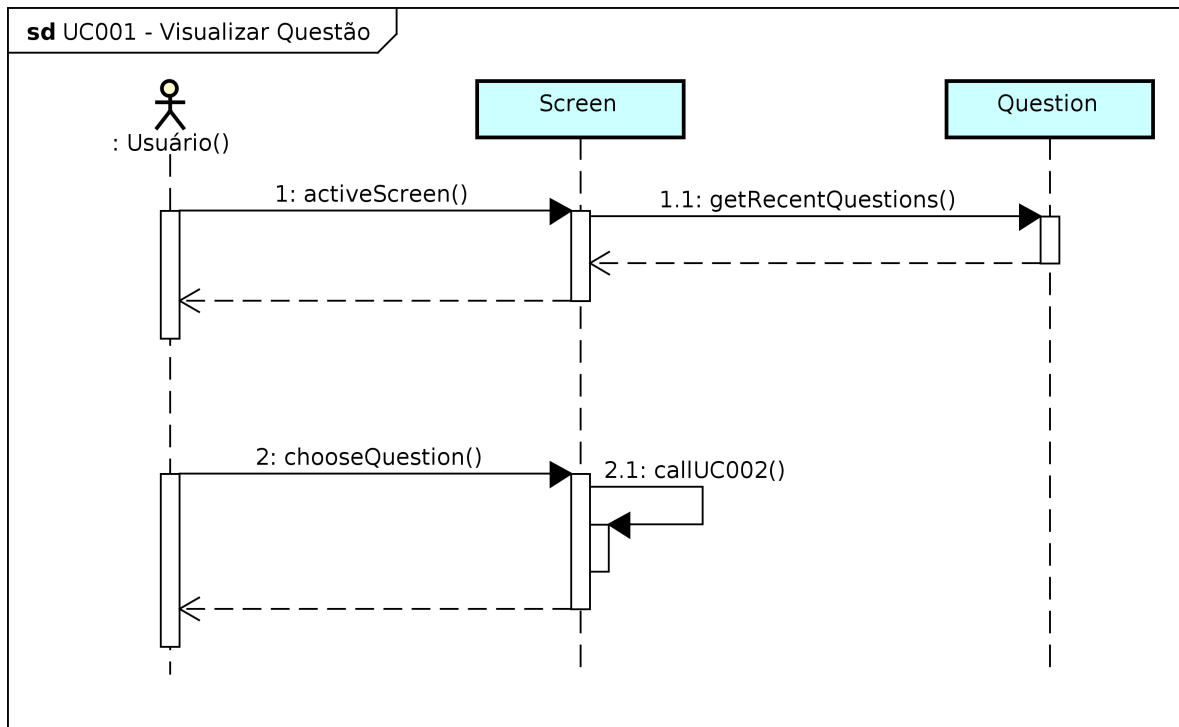


Figura A.3: Diagrama de caso de uso UC001 - Visualizar Questão. Fonte: os autores.

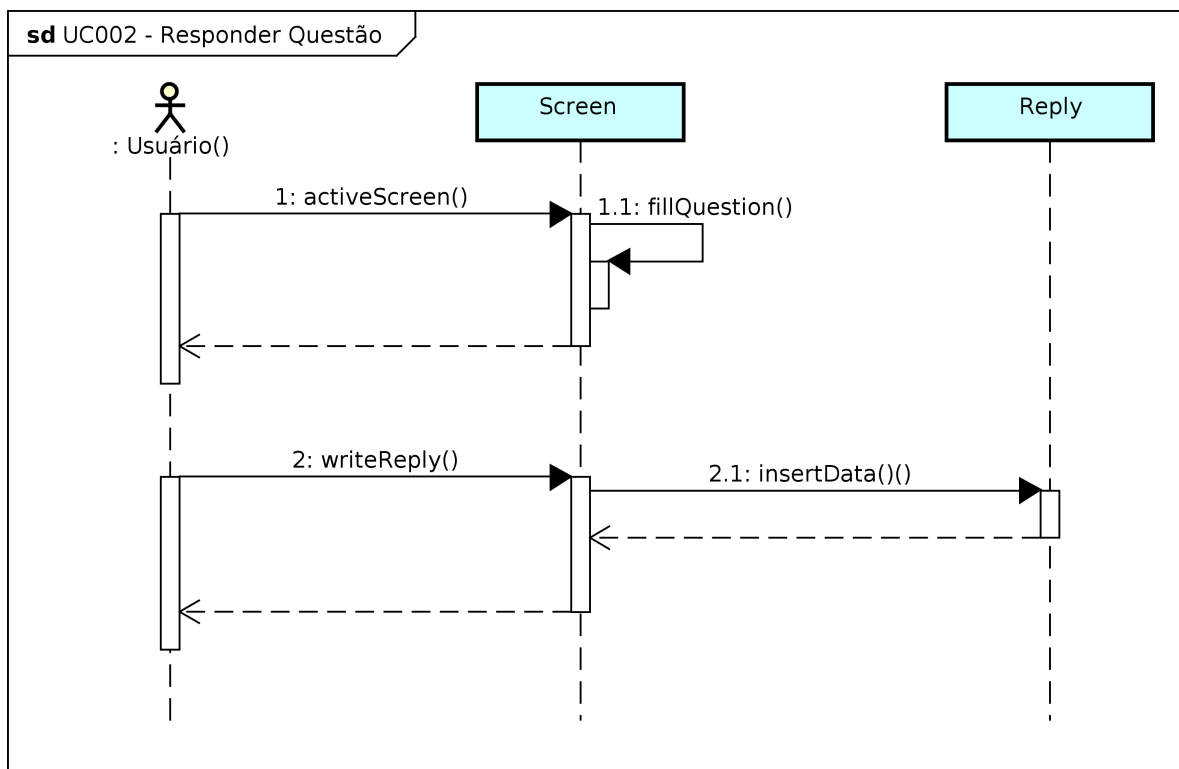


Figura A.4: Diagrama de caso de uso UC002 - Responder Questão. Fonte: os autores.

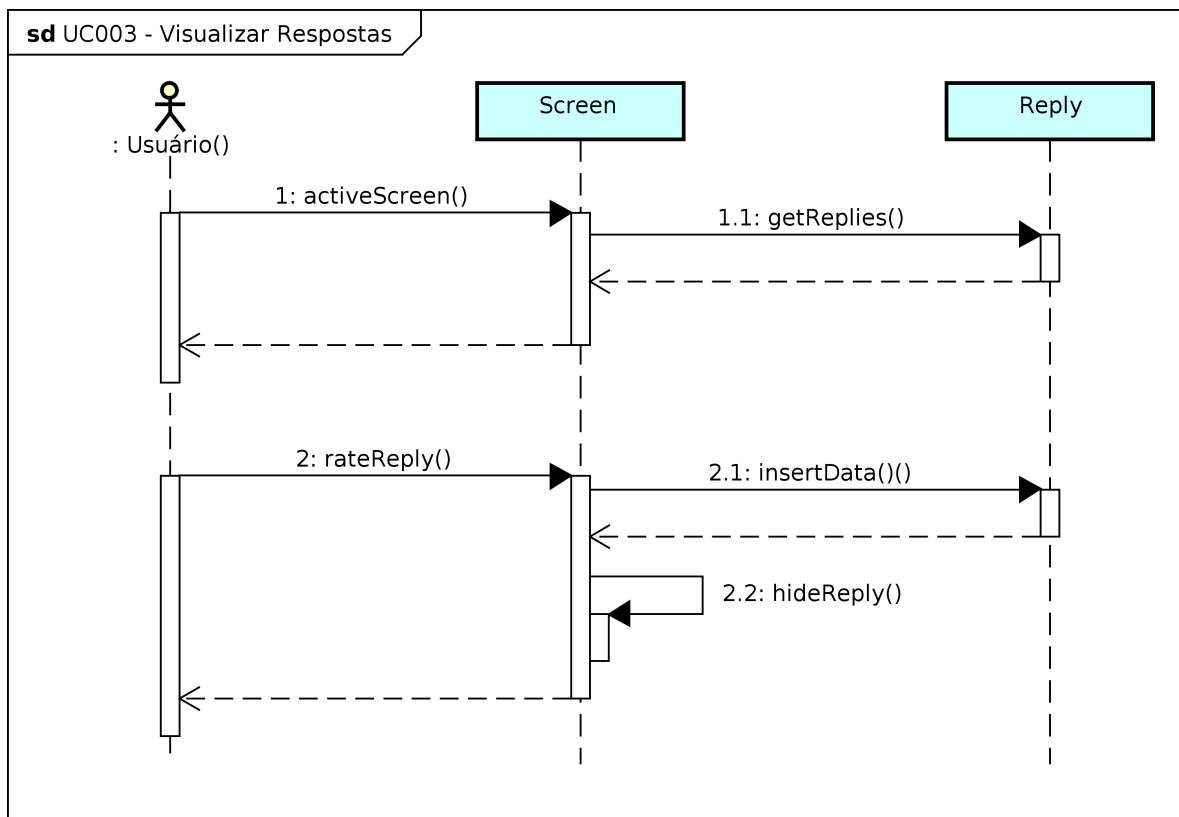


Figura A.5: Diagrama de caso de uso UC003 - Visualizar Respostas. Fonte: os autores.

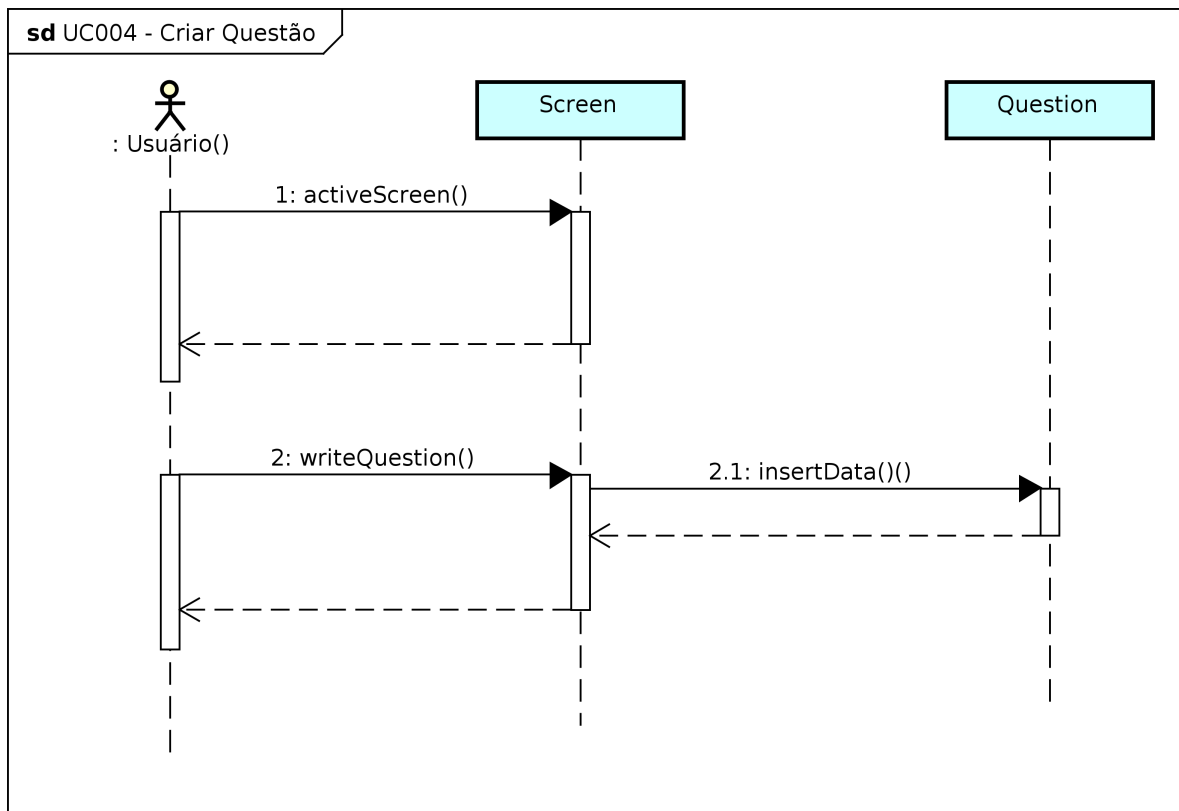


Figura A.6: Diagrama de caso de uso UC004 - Criar Questão. Fonte: os autores.

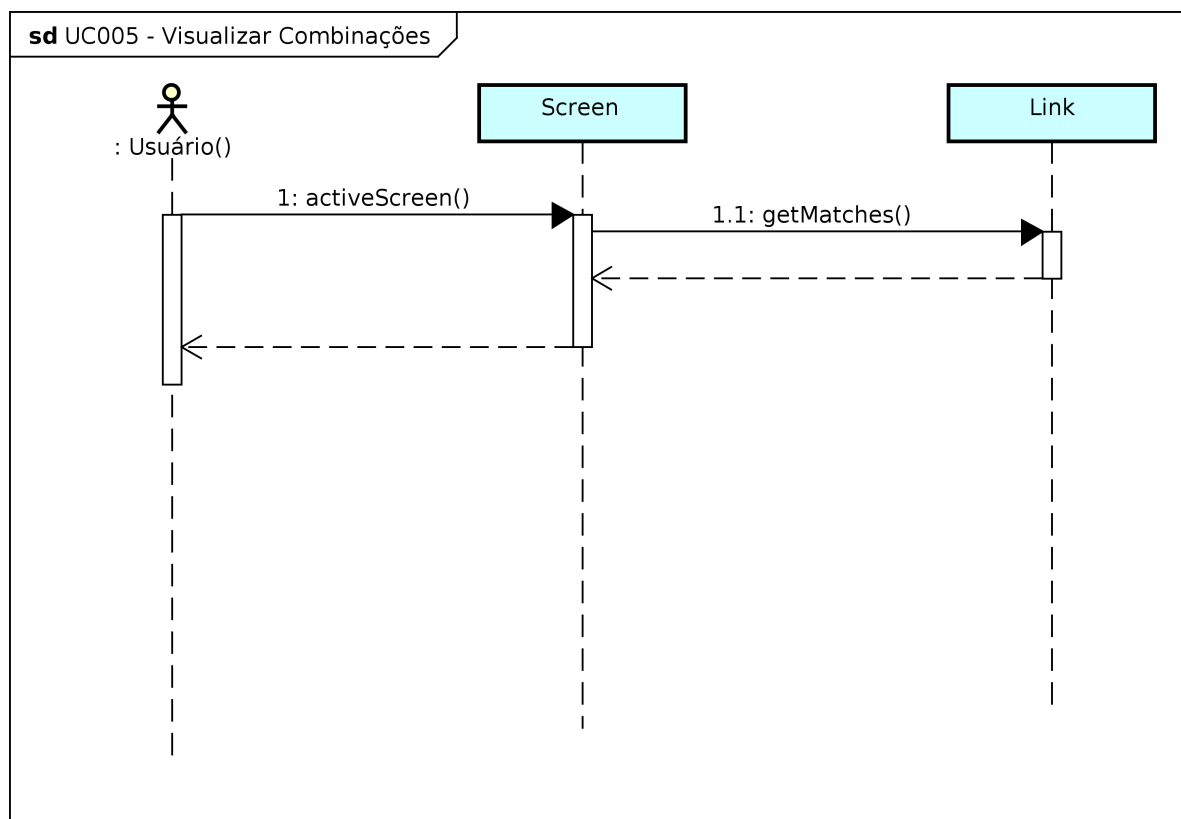


Figura A.7: Diagrama de caso de uso UC005 - Visualizar Combinações. Fonte: os autores.

A.3 Especificação de Casos de Uso

UC001 - Visualizar Questões

Descrição: Este caso de uso tem a função de apresentar as questões para o usuário como em um feed. Data view: Figura 3.6 na página 22

Fluxo de eventos principal:

1. O sistema busca as questões mais recentes para o usuário logado.
2. O sistema carrega as questões buscadas.
3. O sistema apresenta a tela.
4. O usuário escolhe uma questão para responder.
5. O sistema chama UC002.

UC002 - Responder Questão

Descrição: Este caso de uso tem a função de apresentar os recursos envolvidos no processo de responder uma questão.

Data view: Figura 3.8 na página 23

Fluxo de eventos principal:

1. O sistema recebe a questão escolhida.
2. O sistema preenche o card com a questão escolhida.
3. O sistema apresenta a tela.
4. O usuário escreve uma resposta.
5. O usuário clica no botão "Enviar".
6. O sistema registra a resposta.

UC003 - Visualizar Respostas

Descrição: Este caso de uso tem a função de apresentar para o usuário as respostas que ele recebeu para as questões que criou.

Data view: Figura 3.9 na página 23

Fluxo de eventos principal:

1. O sistema busca respostas que ainda não foram avaliadas.

2. O sistema apresenta a tela.
3. O usuário escolhe e avalia uma resposta.
4. o sistema esconde a resposta avaliada.

UC004 - Criar Questão

Descrição: Este caso de uso tem a função de apresentar os recursos necessários para o processo de criação de questões.

Data view: Figura 3.7 na página 22

Fluxo de eventos principal:

1. O sistema apresenta a tela.
2. O usuário rescreve a questão. (E1)
3. O usuário clica no botão enviar.
4. O sistema registra a questão.

Fluxo de eventos de exceção:

E1. A questão deve conter ao menos um caractere.

1. O sistema bloqueia o botão de enviar até que a condição seja satisfeita.

UC005 - Visualizar Combinações

Descrição: Este caso de uso tem a função de apresentar a lista de usuários que foram considerados potenciais amigos do usuário logado.

Data view: Figura 3.10 na página 23

Fluxo de eventos principal:

1. O sistema busca as combinações.
2. O sistema carrega as combinações.
3. o sistema apresenta a tela.
4. O usuário escolhe uma combinação. (E1)
5. Sistema apresenta o mensageiro instantâneo.

Fluxo de eventos de exceção:

E1. O usuário não tem uma combinação.

1. Exibe a tela sem nenhum contato.