

RepublicSearch, um site para busca e cadastro de repúblicas para estudantes

RepublicSearch, a site for searching and registering student housing

DOI:10.34117/bjdv9n1-243

Recebimento dos originais: 16/12/2022

Aceitação para publicação: 16/01/2023

Daniel Rovetta Passos

Graduando em Análise e Desenvolvimento de Sistemas

Instituição: Instituto Federal do espírito Santo – Campus de Alegre

Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000

E-mail: daniel.rovettpassos@gmail.com

Adriano Fernando Bezerra Supeleto

Graduando em Análise e Desenvolvimento de Sistemas

Instituição: Instituto Federal do espírito Santo – Campus de Alegre

Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000

E-mail: adriano.supeleto.17@gmail.com

Wiris Rafael Januário Wernek

Graduando em Análise e Desenvolvimento de Sistemas

Instituição: Instituto Federal do espírito Santo – Campus de Alegre

Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000

E-mail: rafaeljwerneck43@gmail.com

Áquila Jerad Moulin Ditzz

Mestre em Ensino

Instituição: Instituto Federal do espírito Santo – Campus de Alegre

Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000

E-mail: ajmdietz@gmail.com

Cleziel Franzoni da Costa

Mestre em Informática

Instituição: Instituto Federal do espírito Santo – Campus de Alegre

Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000

E-mail: cleziel Franzoni@gmail.com

Flávio Pavese Simão

Doutor em Produção Vegetal

Instituição: Instituto Federal do espírito Santo – Campus de Alegre
Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000
E-mail: fpavese.simao@gmail.com

Carlos Alexandre Siqueira da Silva

Doutor em Informática

Instituição: Instituto Federal do espírito Santo – Campus de Alegre
Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000
E-mail: carlosalexandress@gmail.com

Pedro David Netto Silveira

Doutor em Informática

Instituição: Instituto Federal do espírito Santo – Campus de Alegre
Endereço: Rodovia ES-482, Cachoeiro-Alegre, Km 47, Rive, Alegre - ES,
CEP: 29500-000
E-mail: pedro.dns@gmail.com

RESUMO

Todo ano diversos alunos ingressam em escolas e faculdades, muitas vezes em cidades diferentes das quais residem. A necessidade da procura de uma república se torna trabalhosa devido ao fato de não se conhecer as pessoas que oferecem esse tipo de serviço e a falta do conhecimento da cidade em que procura residir. Este trabalho pretende facilitar a busca do público estudantil, por repúblicas que estejam disponíveis em uma determinada região, com o desenvolvimento de um site, que execute recomendações e que permita pesquisas com nível de detalhamento sobre características de cada local.

Palavras-chave: repúblicas, site, faculdades.

ABSTRACT

Every year several students enroll in schools and colleges, often in cities different from where they reside. The need to look for a republic becomes laborious due to the fact that one does not know the people who offer this kind of service and the lack of knowledge of the city in which one is seeking. This work intends to facilitate the student public's search for republics that are available in a given region, with the development of a website, which executes recommendations and allows research with a level of detail about the characteristics of each location.

Keywords: republics, site, colleges.

1 INTRODUÇÃO

Todo ano diversos alunos ingressam em escolas e faculdades, muitas vezes em cidades diferentes das quais residem. Por isso pode ser gerado um movimento pendular dos estudantes ou a busca por uma moradia próximo ao local de estudo.

Segundo o site Guia do Estudante, “com a criação e expansão de programas como o Prouni, o Fies e principalmente o Sisu, que oferece vagas em instituições públicas de Ensino Superior, o número de pessoas que entram na faculdade cresceu.”. Com isso, aumentou a procura dos estudantes que desejam encontrar uma moradia compartilhada, comumente chamadas de repúblicas.

A necessidade da procura de uma república se torna trabalhosa devido ao fato de não conhecer as pessoas que oferecem esse tipo de serviço e a falta do conhecimento da cidade em que procura residir e isso dificulta encontrar um local que seja desejado.

Além disso, existe um outro problema muito sério. O tipo de moradia durante a faculdade, seja em residência estudantil, casa dos pais ou repúblicas universitárias, têm relação e efeito significativo sobre o desempenho acadêmico dos estudantes universitários ingressantes na educação superior e tem relação direta com a permanência do estudante e conclusão de seu curso (LACERDA e VALENTINI, 2018).

Uma solução para esse problema seria a criação de uma plataforma web onde seria feito o cadastro e a catalogação dessas repúblicas, assim facilitando a busca por um local que esteja de acordo com as necessidades de cada estudante.

O objetivo desse projeto é o desenvolvimento de um sistema *web* voltado para estudantes a fim de auxiliá-los na procura e na oferta de moradia estudantil, onde seja possível, inclusive detalhar características desejadas de cada local a ser alugado, e com o apoio de uma inteligência artificial, receber sugestões de opções de aluguel disponíveis.

2 REFERENCIAL TEÓRICO

Esta seção apresenta os conceitos necessários para entendimento da proposta, e isto inclui, (i) Repúblicas estudantis e (ii) Sistemas de recomendação e recuperação inteligente da informação. Além disso, nesta seção também são apresentados os trabalhos correlatos.

2.1 REPÚBLICAS ESTUDANTIS

Uma república estudantil destinada a albergar estudantes. Essas moradias são montadas e divididas em uma casa ou em um apartamento que suporte um número maior

de pessoas. Geralmente, os moradores são de um mesmo curso ou então de uma mesma instituição de ensino, mas acontece de variar bastante.

O objetivo principal dos universitários que alugam apartamentos, casas e até mesmo quitinetes para dividir com outras pessoas é economizar nas despesas diárias, no aluguel que será dividido igualmente entre os moradores e evitar custos adicionais com possíveis transportes diários.

2.2 SISTEMAS DE RECOMENDAÇÃO

Os Sistemas de Recomendação fazem uso de informações sobre as preferências de seus usuários dentro de um determinado contexto, como filmes, músicas, livros, destinos de viagens etc. e assim produzem sugestões de conteúdos. As informações podem ser adquiridas a partir de avaliações feitas pelos próprios usuários ou sob demanda, com o monitoramento do comportamento dos usuários, como por exemplo: músicas ouvidas, aplicativos baixados, sites visitados etc. (BOBADILLA et al., 2013):

Esses sistemas são geralmente usados no domínio em plataformas de comércio, mas atualmente têm sido demandados em diversas áreas, inclusive no oferecimento de produtos e serviços. Os algoritmos destes sistemas normalmente são baseados em técnicas de recuperação inteligente da informação (RIVERA; TAPIA-LEON; LUJAN-MORA, 2018).

2.3 TRABALHOS CORRELATOS

Foi efetuada uma pesquisa a fim de encontrar outras aplicações semelhantes com o sistema proposto. Foram encontradas diversas aplicações, a maioria de cunho comercial. Dentre os trabalhos encontrados, foram destacados dois que mais se assemelham ao RepublicSearch.

Desde já destacamos que o grande diferencial entre o RepublicSearch e todas as propostas encontradas é a funcionalidade de recomendação inteligente de opções para os usuários.

2.3.1 Republika

O primeiro trabalho trata do site Republika, que fornece o cadastro e exibição das repúblicas destinadas aos estudantes. Tal plataforma se apresenta com uma interface colorida, filtros de busca por cidade, preço e gênero.

Aparentemente o site se encontra em uma fase de desuso/desenvolvimento, o que pode acarretar na dificuldade de navegação, cadastro de repúblicas, procura pelos anúncios, formas de acesso a plataforma, entre outros.

2.3.2 República Fácil

Em Mendes (2018) foi apresentado um projeto cujo objetivo é desenvolver uma plataforma web que sirva como espaço de divulgação e procura de vagas em repúblicas, além de estabelecer um canal de comunicação eficiente entre quem oferece e quem procura uma república.

O “República Fácil” foi criado com o intuito de auxiliar a divulgação e a busca por vagas de república, e segundo os autores, o diferencial do sistema é que ele conta com área pessoal do usuário e permite que cada usuário gerencie as vagas por ele anunciadas e a comunicação entre ele e quem se interessa por elas.

2.3.3 Domuz

O site Domuz apresenta possibilidades de filtros que facilitam a busca por vagas de república. Um pormenor, em relação a utilização do sistema, é que ele depende de um moderador para gerenciamento das vagas. Na plataforma ainda existe uma área de login que não permite interação entre os usuários.

Por se tratar de uma iniciativa comercial, o site apresenta alguns banners com notícias e anúncios, principalmente por não se tratar de uma plataforma específica para consulta de repúblicas. Além dessa funcionalidade, que se assemelha à deste trabalho, o site também provê funcionalidades de blog, notícias, dicas, tutoriais etc.

3 METODOLOGIA

Para o desenvolvimento deste trabalho, nos baseamos na metodologia de pesquisa experimental apresentada por Wazlawick (2009), seguindo o estilo de pesquisa “apresentação de um produto”, muito comum para a área de computação.

No desenvolvimento da aplicação proposta neste trabalho, foi realizada uma pesquisa bibliográfica, aplicada e descritiva, com os dados obtidos a partir de uma busca por aplicações já existentes. Os elementos obtidos foram analisados qualitativamente quanto às suas funcionalidades com o objetivo de gerar soluções para facilitar o acesso de informação aos estudantes sobre repúblicas instaladas em uma cidade.

Após a pesquisa bibliográfica, iniciou-se o desenvolvimento do sistema seguindo o ciclo de vida iterativo e incremental (PRESSMAN e MAXIN, 2021) passando pelas fases de levantamento de requisitos, análise e projeto do sistema quando os modelos de software e dados foram produzidos.

Em cada ciclo durante o desenvolvimento da aplicação, após o desenho dos modelos na fase de projeto, inicia-se a fase de programação do sistema gerencial, que foi feita com a linguagem C# e por último, a fase de testes, que foram conduzidos com as técnicas de caixa preta e caixa branca (PRESSMAN e MAXIN, 2021).

O agente inteligente, que usa técnicas de recuperação inteligente da informação para promover as recomendações do sistema, foi implementado com a linguagem Python e com as bibliotecas de processamento de linguagem natural *nlTK* e *Spacy*.

Foi utilizado o *Google Scholar* e o *SciELO* para a pesquisa acadêmica e o *Google Docs* para a realização da documentação. Para a modelagem do sistema foi utilizado o software ASTAH. Para a implementação do software estão sendo utilizados *SQL Server* como sistema gerenciador de bancos de dados e a tecnologia ASP.NET sobre Visual Studio para programação.

Será posteriormente estruturado e aplicado um questionário com algumas perguntas com a finalidade de apurar o grau de conhecimento sobre a experiência dos usuários ao utilizar o sistema proposto.

O objetivo deste questionário é de avaliar qualitativamente e quantitativamente a proposta, valendo-se de um instrumento de coleta de dados constituído por uma série de perguntas, em que os estudantes deverão responder por meio de um formulário web.

4 O PROJETO REPUBLICSEARCH

Nesta seção será apresentado o projeto de software da plataforma, que foi nomeada como *Republic Search* no seu estágio atual. Serão demonstrados os modelos do projeto e uma explicação mais detalhada sobre o funcionamento da funcionalidade de recomendação inteligente.

4.1 REQUISITOS FUNCIONAIS

De acordo com Sommerville (2011), requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Cada requisito

funcional representa uma ação do software. Na Tabela 1 são apresentados os requisitos funcionais do sistema deste trabalho.

Tabela 1 - Requisitos Funcionais

Sigla	Nome	Descrição	Dependência
RF01	Realizar Login	O usuário deve ser capaz de realizar login no sistema.	
RF02	Gerenciar Contas de Usuários	O usuário deve ser capaz de efetuar o gerenciamento (cadastrar, visualizar e alterar) de sua conta.	RF01
RF03	Gerenciar Anúncios	O usuário deve ser capaz de efetuar o gerenciamento (Cadastrar, visualizar, alterar e excluir) de seus anúncios publicados.	RF01
RF04	Buscar Anúncios	O sistema deve possibilitar ao usuário opções de buscas e filtros aos anúncios.	
RF05	Recomendação de anúncios	O sistema deve utilizar uma IA que sugere determinados anúncios ao usuário de acordo com seu histórico de buscas.	RF04

Fonte: O Autor (2022)

4.2 REQUISITOS NÃO FUNCIONAIS

De acordo com Sommerville (2011), requisitos não funcionais são restrições aos serviços ou funções oferecidos pelo sistema. Os requisitos não funcionais são as consistências e validações que devem ser aplicadas sobre os requisitos não funcionais. No quadro 2 são apresentados os requisitos não funcionais identificados no sistema.

Tabela 2 - Requisitos Não Funcionais

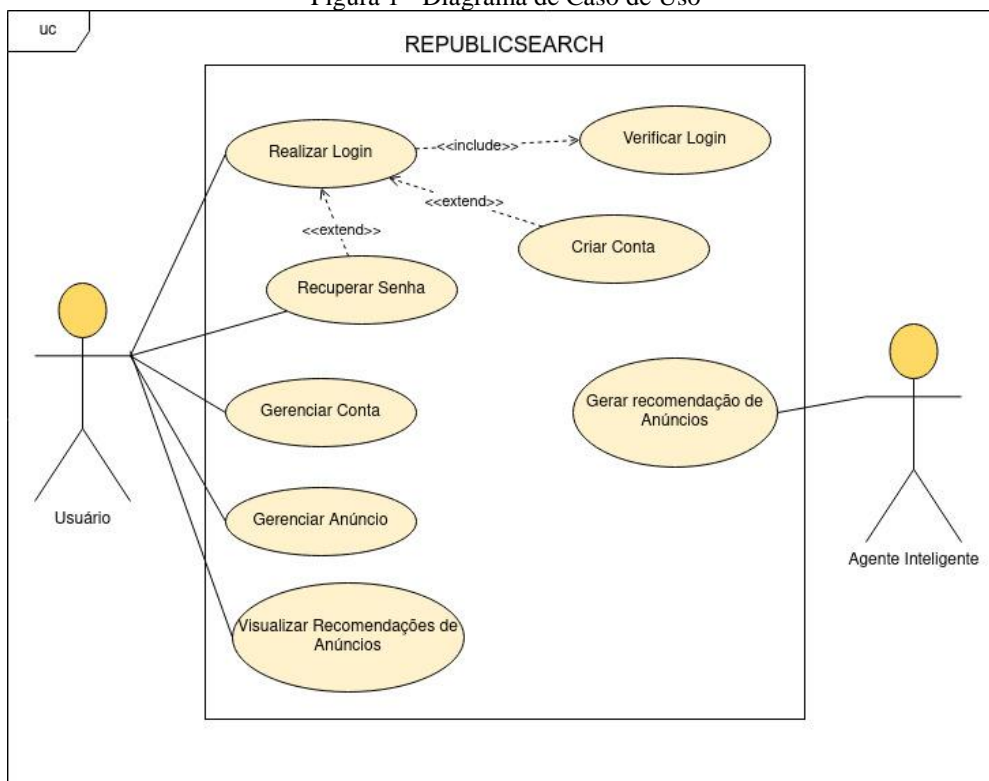
Identificador	Descrição
RNF01	O sistema deve ser desenvolvido multi-plataforma.
RNF02	O sistema deve ter conexão com banco de dados.

Fonte: O Autor (2022)

4.3 DIAGRAMA DE CASO DE USO

De acordo com Guedes (2009), O diagrama de casos de uso tem por objetivo apresentar uma visão externa geral das funcionalidades que o sistema deverá oferecer aos usuários, sem se preocupar em profundidade com a questão de como tais funcionalidades serão implementadas. A seguir, na Figura 1, está apresentado o diagrama de caso de uso do sistema.

Figura 1 - Diagrama de Caso de Uso



Fonte: O Autor (2022)

No diagrama são apresentados 2 atores, o usuário e o Agente Inteligente. O usuário, que representa uma pessoa utilizando o sistema, acessa a página inicial, esta que exibirá todos os anúncios cadastrados, junto com suas informações. Nessa página é possível realizar uma busca mais específica, de acordo com as preferências do usuário. Caso ele deseje criar e manter um anúncio, é necessário que ele acesse o sistema através de seu login, e caso não possua será necessário realizar seu cadastro.

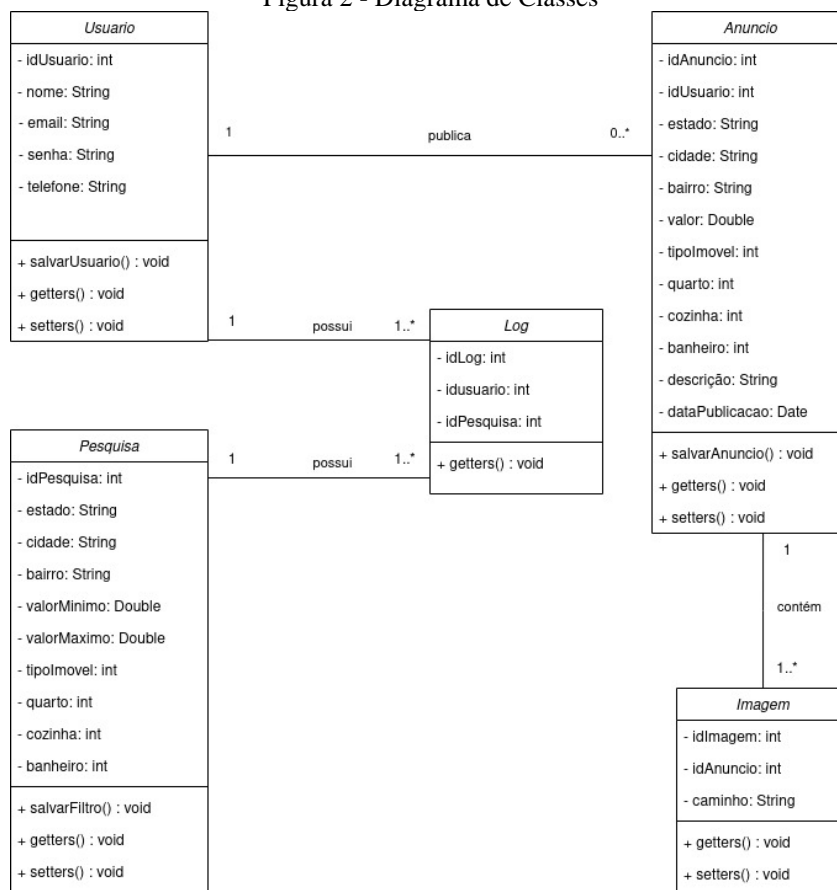
Caso o usuário esqueça sua senha, será possível solicitar a recuperação na página de login do sistema. Ao visualizar e selecionar o anúncio desejado, será possível identificar as informações de contato do anunciante, permitindo assim que o usuário possa ter uma comunicação para esclarecimento de dúvidas ou realizar uma negociação. O agente inteligente representa a parte do sistema que irá sugerir determinados anúncios ao usuário, de acordo com seu histórico de buscas.

4.4 DIAGRAMA DE CLASSES

Segundo Guedes (2009), o diagrama de classes é um dos mais importantes e utilizados da UML. Seu principal enfoque está em permitir a visualização das classes que compõem o sistema com seus respectivos atributos e métodos, bem como demonstrar

como as classes do diagrama que se relacionam, complementam e transmitem informações entre si. A Figura 2 apresenta o modelo de diagrama de classes do sistema.

Figura 2 - Diagrama de Classes

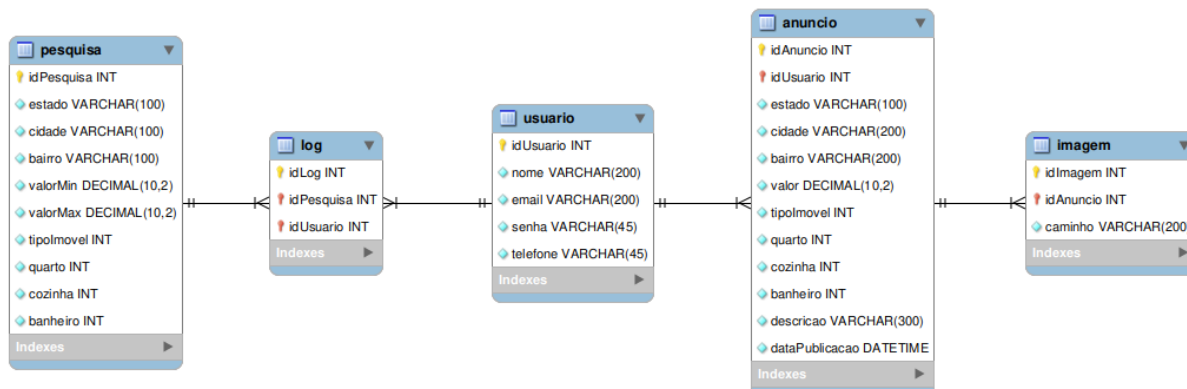


Fonte: O Autor (2022)

4.5 MODELO LÓGICO DO BANCO DE DADOS

Segundo Machado (2020), o modelo lógico de um banco de dados descreve em formato visual as estruturas que estarão presentes no banco de dados de acordo com as possibilidades permitidas pelo modelo conceitual, mas sem considerar características específicas de um Sistema Gerenciador de Banco de Dados (SGBD), assim resultando um esquema lógico de dados sob a ótica do modelo conceitual. A Figura 3 apresenta o diagrama do modelo lógico de dados do sistema.

Figura 3 - Modelo Lógico do Banco de Dados



Fonte: O Autor (2022)

4.6 FUNCIONALIDADE DE RECOMENDAÇÃO

O objetivo das recomendações é tornar a experiência do usuário mais customizada e mais intuitiva a cada utilização. Trazer essas sugestões de anúncios baseadas nas pesquisas mais recentes de cada usuário irá incentivá-los a continuar sua navegação na plataforma.

Essa funcionalidade tem por intenção utilizar de um agente inteligente para interpretação dos dados gerados pelas pesquisas dos usuários e em seguida uma classificação baseada em padrões e compatibilidade de anúncios que podem ser relevantes ao usuário, esse processo se vale de um algoritmo de similaridade.

Vinson (2002) afirma que *“Os algoritmos de similaridade propõem avaliar a semelhança entre informações contidas nas estruturas e expressá-las por meio de valores reais.”* logo a utilização de tal algoritmo se justifica na intenção de obter uma maior e mais assertiva compatibilidade entre os anúncios.

Um exemplo do algoritmo citado anteriormente pode ser visto na Figura 4. Trata-se de uma implementação na linguagem Python que é útil para fazer o cálculo da similaridade dos textos de dois arquivos diferentes. O agente inteligente, parte deste projeto, foi programado com uma adequação do código fonte apresentado.

Figura 4 - Exemplo de um algoritmo de similaridade

```
import nltk
import unicode
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords

def remove_repetidos(lista):
    l = []
    for i in lista:
        if i not in l:
            l.append(i)
    return l

def similaridade(lista1, lista2):
    qtd = 0
    for i in lista1:
        for x in lista2:
            if i[0] == x[0]:
                qtd += 1
    match = qtd * 100/len(lista1)
    return match

file1 = unicode.unicode(open("arquivo.txt", encoding="utf-8").read())
token = RegexpTokenizer(r'[A-z]\w*')
file1 = token.tokenize(file1)
stopwords = nltk.corpus.stopwords.words('portuguese')
file1 = [w.lower() for w in file1 if w not in stopwords]
file1_count = nltk.FreqDist(file1)
print("15 palavras mais frequentes no arquivo 1: \n\n"+str(file1_count.most_common(15))+"\n\n")

file2 = unicode.unicode(open("arquivo2.txt", encoding="utf-8").read())
token = RegexpTokenizer(r'[A-z]\w*')
file2 = token.tokenize(file2)
stopwords = nltk.corpus.stopwords.words('portuguese')
file2 = [w.lower() for w in file2 if w not in stopwords]
file2_count = nltk.FreqDist(file2)
print("15 palavras mais frequentes no arquivo 2: \n\n"+str(file2_count.most_common(15))+"\n\n")

palavras_em_comum = []
for i in range(len(file1)):
    for j in range(len(file2)):
        if file1[i] == file2[j]:
            palavras_em_comum.append(file1[i])

palavras_em_comum=remove_repetidos(palavras_em_comum)
similaridade = similaridade(file1_count.most_common(15) ,file2_count.most_common(15))
print(f"a similaridade dos textos é: {similaridade} %")
```

Fonte: O Autor (2022)

A proposta é que o agente inteligente se baseie nas 10 pesquisas mais recentes do usuário para buscar anúncios que sejam similares e relevantes para ele a partir do algoritmo de similaridade para realização dessa busca. Os anúncios que forem compatíveis serão mostrados ao usuário.

5 A PLATAFORMA REPUBLICSEARCH

Nesta seção será apresentada a plataforma RepublicSerach em seu estágio atual. O usuário ao entrar na plataforma, se depara com a tela de login, apresentada na Figura 5. Nessa tela o usuário fornece seus dados, que serão validados e autenticados. Caso o usuário tente realizar a operação de login na plataforma sem fornecer alguma credencial, uma mensagem de preenchimento obrigatório do campo será exibida.

Caso ele forneça um login e/ou senha que não se encontre na base de dados do sistema, a plataforma retornará uma mensagem de erro, “Nenhuma conta encontrada ou

senha incorreta”. Também é possível encontrar nessa tela, duas opções secundárias para o usuário, sendo: realizar um novo cadastro e recuperar senha.

Figura 5 - Tela de Login



Fonte: O Autor (2022)

Ao clicar na opção de “CADASTRE-SE”, o usuário é redirecionado para a tela de “Criar a sua Conta”, apresentada na Figura 6. A tela apresentará campos cujas informações são necessárias para o usuário criar sua conta na plataforma. Estes campos também apresentam validação das credenciais inseridas, bloqueando o usuário de prosseguir com o cadastro caso ele deixe algum campo em branco ou inserir alguma informação incorretamente ou insuficiente.

Figura 6 - Tela de Criar Conta



Fonte: O Autor (2022)

Se porventura o usuário fornecer um email já existente na base de dados a mensagem “E-mail já cadastrado” será exibida. Caso as credenciais sejam válidas, ao clicar no botão “CADASTRE-SE”, a plataforma irá retornar uma mensagem de “Cadastro

realizado com sucesso”. Após o preenchimento das informações, o usuário será redirecionado para a página de Login da plataforma. Nela, o usuário irá fornecer seus dados, que serão validados e autenticados.

Caso o usuário perca sua senha de login, é possível recuperá-la através da tela de “Esqueceu a Senha?”, apresentada na figura 7. Nela é apresentado um pequeno texto com uma instrução de como recuperar sua senha, por meio do e-mail utilizado para realizar o login no aplicativo. Caso um e-mail inválido seja inserido uma mensagem de “Insira um e-mail válido!” será apresentada, ou se o e-mail informado não constar na base de dados, a mensagem de “Nenhuma conta vinculada a este e-mail” será exibida.

Ao inserir o e-mail vinculado a sua respectiva conta, o usuário deve verificar o e-mail enviado pelo aplicativo para seguir as instruções de recuperação de senha. Ao final da ação, o usuário será redirecionado para a tela de login do aplicativo.

Figura 7 - Tela de Recuperar Conta

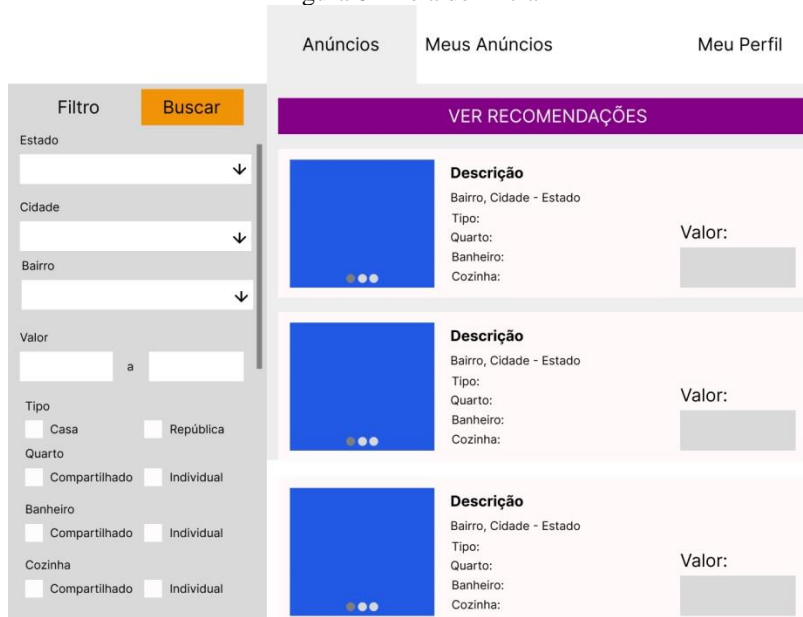


Fonte: O Autor (2022)

Após o sistema autenticar um login válido, o usuário será redirecionado para a tela inicial do aplicativo, apresentada na Figura 8, que possui uma listagem inicial dos anúncios cadastrados no site, junto com a opção de busca por um filtro e um menu de navegação entre páginas no final da tela, com as opções de navegar entre as telas Anúncios, Meus Anúncios e Meu Perfil.

Ao clicar no botão “VER RECOMENDAÇÕES”, o painel principal irá exibir todos os anúncios sugeridos pelo agente inteligente integrado ao sistema.

Figura 8 - Tela de Inicial



Fonte: O Autor (2022)

O usuário pode realizar uma busca mais refinada no campo de Filtro. Após o usuário selecionar o que deseja buscar, basta clicar no botão “Buscar” que a busca será realizada e os anúncios filtrados serão exibidos na página inicial.

Ao encontrar o anúncio desejado e clicar em seu texto ou imagem, o usuário é redirecionado para a página de “Detalhes” (Figura 9). A tela contará com todas as especificações da moradia catalogada. Além de constar essas informações, a página também apresenta ao usuário o nome do anunciante junto com o telefone de contato.

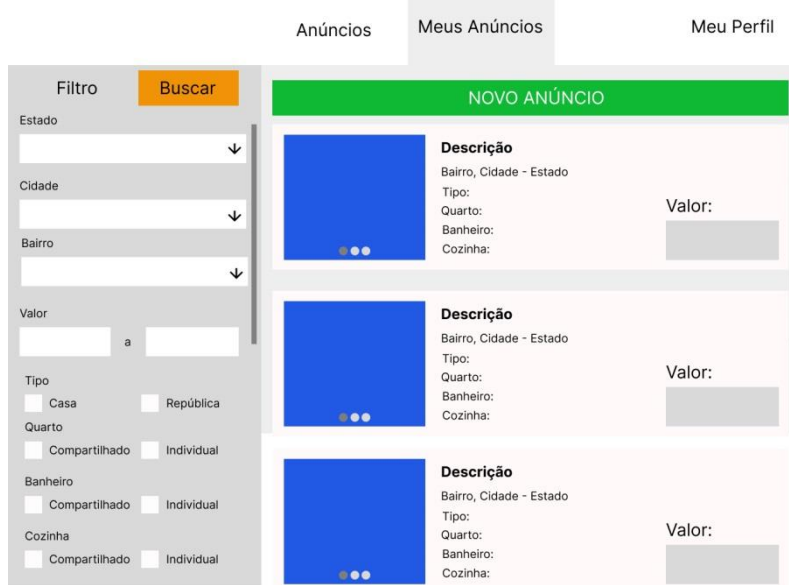
Figura 9 - Tela de Detalhes



Fonte: O Autor (2022)

Na aba de menus terá a aba “Meus Anúncios”, onde o usuário será direcionado para uma página, Figura 10, na qual será listada todos os anúncios feitos pelo usuário. Na parte superior do painel terá um botão para realizar o cadastro de novos anúncios.

Figura 10 - Tela de Meus Anúncios



Fonte: O Autor (2022)

Ao selecionar um anúncio, o usuário será redirecionado para a página de “Detalhes”, apresentada na Figura 11. A tela contará com todas as especificações da moradia catalogada, o nome do anunciante, o telefone de contato, e dois botões de ação, para editar ou apagar o anúncio.

Figura 11- Tela Detalhes



Fonte: O Autor (2022)

É importante mencionar que no RepublicSearch existem ainda várias outras telas, como, por exemplo, para controle de anúncios, usuários e perfil, nas quais os usuários podem realizar operações de gerenciamento de seus cadastros.

6 CONSIDERAÇÕES FINAIS

A literatura tem apontado para um grande sucesso no que diz respeito ao suporte computacional como fomentador de sistemas de recomendação para todo tipo de plataforma (ALDRICH, 2011), principalmente com a potencialização fornecida pela inteligência artificial.

A contribuição deste trabalho, considerando o contexto acima, se trata da descrição de uma aplicação para, de um lado, apoiar pessoas que estão oferecendo opções de moradia estudantil, e por outro lado, para ajudar estudantes à procura de repúblicas a encontrar este recurso que é necessário para sua formação. Neste trabalho também disponibilizamos o projeto de software do RepublicaSearch, que pode ser replicado por integrantes da comunidade que sejam interessados na proposta.

Como trabalhos futuros, existe a intenção de estendermos os testes da aplicação produzida com usuários reais, para a geração de dados quantitativos e qualitativos. Além disso, também pretende-se potencializar o sistema produzido com novas funcionalidades inteligentes para integração da plataforma, por meio de robôs de busca, com outros sites da internet que também ofereçam opções de moradia compartilhada.

REFERÊNCIAS

ALDRICH, Susan E. "Recommender systems in commercial use." *AI Magazine*. 28-34 pp. 2011.

BOBADILLA, J. et al. Recommender systems survey. *Knowledge-based systems*. v. 46. Elsevier. 109–132 pp. 2013.

GUEDES, Gilleanes TA. *UML 2: Uma Abordagem Prática*. São Paulo. Novatec. 2009.

LACERDA, Izabella Pirro., Felipe Valentini. "Impacto da moradia estudantil no desempenho acadêmico e na permanência na universidade." *Psicologia Escolar e Educacional* 22. 2018.

MACHADO, Felipe Nery Rodrigues. *Banco de Dados–Projeto e Implementação*. Saraiva Educação SA, 2020.

MENDES, L. A. M., de Lima, L. A. M., Adriano, F. M. A., Melo, J. P. F., & Silva, L. O. **REPÚBLICA FÁCIL: PROTÓTIPO DE UMA PLATAFORMA WEB PARA OFERTA DE VAGAS EM REPÚBLICAS**. In 28ª Mostra Específica de Trabalhos e Aplicações. 2018.

PRESSMAN, Roger S., Bruce R. Maxim. *Engenharia de software-9*. McGraw Hill Brasil, 2021.

RIVERA, A. C., TAPIA-LEON, M., LUJAN-MORA, S. Recommendation systems in education: A systematic mapping study. In: *SPRINGER. International Conference on Information Technology & Systems*. [S.l.]. 937–947 pp. 2018.

SOMMERVILLE, I. *Engenharia de Software*. 09. ed São Paulo. Pearson Prentice Hall.. 2011.

VINSON, Alexander Richard. *PathSim: um algoritmo para calcular a similaridade entre caminhos XML*. 2007.

WAZLAWICK, Raul Sidnei. *Metodologia de pesquisa para ciência da computação*. Vol. 2. Elsevier, 2009.