RESIGeT On-Line
Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica
Vol. 14 Nro. 1 2024
Uni-FACEF
Centro Universitário Municipal de Franca

Canine Match - Produto Mínimo Viável

Arthur Henrique de Paula
Graduando em Engenharia de Software – Uni-FACEF
arthurhenriquep2@gmail.com

Prof. Me. Carlos Eduardo de França Roland
Docente do Departamento de Computação – Uni-FACEF
roland@facef.br

Resumo

Entre diversas raças caninas, cada uma apresenta características únicas, comportamentos distintos e necessidades específicas, o que torna a seleção do animal de estimação ideal uma tarefa complexa e muitas vezes confusa para a adoção. O aplicativo Canine Match é uma resposta à necessidade de oferecer suporte eficiente e personalizado aos potenciais tutores de cães. Por meio de algoritmos de IA e interface amigável, o aplicativo busca direcionar os usuários para a raça mais compatível com seu estilo de vida, preferências e circunstâncias específicas. A relevância desta iniciativa reside na importância de garantir uma convivência saudável entre os tutores e seus animais de estimação. Ao oferecer orientações personalizadas, o Canine Match não apenas simplifica o processo de escolha, mas também contribui para a prevenção de problemas comportamentais e o abandono de animais, promovendo o bem-estar tanto dos cães quanto de seus cuidadores. Além disso, o aplicativo visa otimizar o tempo e os recursos dos usuários, fornecendo recomendações precisas de forma rápida e eficiente. Ao facilitar a seleção consciente e adequada da raça, o Canine Match fortalece os laços afetivos entre humanos e animais, contribuindo para o estabelecimento de relações mais saudáveis e duradouras. Foi aplicada a metodologia ágil nos processos do Ciclo de Vida do Desenvolvimento de Software para a implementação da solução, com a adoção de Python e a biblioteca gráfica TKInter para acesso ao sistema de recomendação do Google baseado em conteúdo, pela API GEMINI. Dessa forma, o desenvolvimento do protótipo funcional representa uma solução inovadora para atender às necessidades de um público cada vez mais comprometido com o cuidado responsável e o bem-estar dos animais de estimação.

Palavras-chave: Canine Match; cuidado animal responsável; escolha de animal de estimação; inteligência artificial; raças caninas.

Abstract

Among various canine breeds, each presents unique characteristics, distinct behaviors, and specific needs, making the selection of the ideal pet a complex and often confusing task for adoption. The Canine Match app is a response to the need to offer efficient and personalized support to potential dog owners. Through AI algorithms and a user-friendly interface, the app aims to guide users to the breed most compatible with their lifestyle, preferences, and specific circumstances. The relevance of this initiative lies in the importance of ensuring a healthy coexistence between owners and their pets. By offering personalized guidance, Canine Match not only simplifies the selection process but also contributes to the prevention of behavioral problems and

animal abandonment, promoting the well-being of both dogs and their caregivers. Additionally, the app aims to optimize users' time and resources by providing accurate recommendations quickly and efficiently. By facilitating the conscious and appropriate selection of a breed, Canine Match strengthens the emotional bonds between humans and animals, contributing to the establishment of healthier and more lasting relationships. The agile methodology was applied in the Software Development Life Cycle processes for the implementation of the solution, with the adoption of Python and the TKInter graphical library for access to Google's content-based recommendation system via the GEMINI API. In this way, the development of the functional prototype represents an innovative solution to meet the needs of an increasingly committed audience to responsible care and the well-being of pets.

***Keywords**: animal responsible care; artificial intelligence; Canine Match; choice of pet; dog breeds.*

1 Introdução

A escolha de um animal de estimação, especialmente um cão, envolve diversos fatores que vão além da simples preferência estética. Cada raça canina possui características únicas, comportamentos específicos e necessidades distintas, o que pode tornar o processo de seleção do animal de estimação ideal uma tarefa complexa e, muitas vezes, confusa para os futuros tutores. Esse desafio é amplificado pela vasta diversidade de raças disponíveis e pela necessidade de encontrar um companheiro que se ajuste ao estilo de vida, preferências pessoais e circunstâncias específicas de cada indivíduo.

Em resposta a essa demanda crescente por orientações precisas e personalizadas, o Canine Match foi criado com o objetivo de ser um aplicativo inovador que utiliza algoritmos de Inteligência Artificial (IA) para auxiliar potenciais tutores de cães na identificação da raça mais compatível com suas características e condições de vida. Com uma interface amigável e intuitiva, o Canine Match oferece uma solução prática e eficiente para um problema comum, mas muitas vezes negligenciado, no processo de adoção de animais de estimação.

A relevância desta iniciativa reside na promoção de uma convivência harmoniosa e saudável entre tutores e seus animais de estimação. Ao fornecer orientações personalizadas, o aplicativo não apenas simplifica o processo de escolha, mas também contribui para a prevenção de problemas comportamentais e o abandono de animais, situações frequentemente decorrentes de escolhas inadequadas. Dessa forma, o aplicativo promove o bem-estar tanto dos cães quanto de seus tutores, ao mesmo tempo em que fortalece os laços afetivos e estabelece relações mais duradouras e satisfatórias entre eles.

Além disso, o protótipo otimiza o tempo e os recursos dos usuários, oferecendo recomendações precisas de maneira rápida e eficiente. Ao facilitar uma adoção consciente e informada, o Canine Match não só melhora a experiência dos tutores de cães, mas também incentiva um cuidado responsável e comprometido com o bem-estar dos animais de estimação.

Portanto, o desenvolvimento do projeto representa uma solução inovadora na aplicação de tecnologias recentes, e necessária para um público cada vez mais atento às responsabilidades envolvidas na adoção de um animal de estimação. Este trabalho apresenta o impacto e a eficácia do Canine Match na escolha adequada de raças caninas, podendo ser uma ferramenta de avaliação do potencial

de melhorar a convivência entre humanos e animais de estimação, contribuindo para uma sociedade mais consciente e comprometida com o bem-estar animal.

2 Bases Teóricas

Nessa seção são apresentadas as bases teóricas que nortearam a definição do tema e problema de pesquisa, bem como da hipótese de solução e as tecnologias utilizadas para implementação e testes da solução.

2.1 Tecnologias Web e Desenvolvimento de Aplicações Interativas

Python, uma linguagem de programação de alto nível e de propósito geral, tem se destacado no desenvolvimento de aplicações devido à sua simplicidade, legibilidade e vasta gama de *frameworks* disponíveis. Para agilizar o desenvolvimento do aplicativo foi utilizada a biblioteca Tkinter, baseada em Tcl/Tk, que facilita a implementação de interfaces gráficas com Python numa base sólida para o desenvolvimento rápido e eficiente de aplicações.

Python é uma linguagem de fácil aprendizado, por sua sintaxe simples, ao mesmo tempo que oferece poderosas funcionalidades de processamento. As principais características da linguagem, segundo (Carvalho, 2021) são descritas no Quadro 1.

Quadro 1 - Características da linguagem Python

Sintaxe simples e legível	Uma das características distintivas do Python é sua sintaxe simples e legível, que se assemelha à linguagem humana. Isso facilita a escrita e compreensão do código, tornando Python uma escolha ideal para iniciantes e experientes programadores.
Amplo conjunto de bibliotecas padrão	Python possui uma vasta oferta de bibliotecas padrão, que oferecem uma ampla gama de módulos e funcionalidades para diversas tarefas, desde manipulação de arquivos até desenvolvimento web e científico. Essas bibliotecas abrangentes economizam tempo e esforço do desenvolvedor, permitindo a implementação rápida de soluções.
Portabilidade e multiplataforma	Python é uma linguagem multiplataforma, o que significa que os programas escritos em Python podem ser executados em diversas plataformas, incluindo Windows, MacOS e Linux, sem a necessidade de modificação do código-fonte. Isso oferece uma grande flexibilidade para desenvolvedores e usuários.
Facilidade de aprendizado e comunidade ativa	Python é conhecido por ser uma das linguagens mais fáceis de aprender, graças à sua sintaxe intuitiva e abordagem orientada a objetos. Além disso, Python possui uma comunidade ativa e acolhedora, que oferece amplo suporte, recursos educacionais e bibliotecas de código aberto para

	ajudar os desenvolvedores a resolverem problemas e expandirem seus conhecimentos.
Flexibilidade e versatilidade	Python é uma linguagem altamente flexível e versátil, adequada para uma ampla variedade de aplicações, desde desenvolvimento web e desktop até análise de dados, aprendizado de máquina e automação de tarefas. Essa versatilidade torna Python uma escolha popular entre os desenvolvedores que buscam uma linguagem única para atender a diferentes necessidades.
Desenvolvimento rápido e produtividade elevada	Devido à sua sintaxe concisa e à ampla gama de bibliotecas disponíveis, Python permite o desenvolvimento rápido de protótipos e a implementação eficiente de soluções. Isso resulta em uma produtividade elevada para os desenvolvedores, que podem focar mais na lógica do problema em vez de se preocuparem com detalhes de baixo nível.

Fonte: Carvalho, 2021

2.3 APIs RESTful e Comunicação entre Serviços

As APIs RESTful desempenham um papel essencial na arquitetura moderna de aplicações web, atuando como o principal meio de comunicação entre diferentes componentes e serviços. Seguindo os princípios da arquitetura REST (*Representational State Transfer*), essas APIs adotam um conjunto de restrições que favorecem a criação de serviços web escaláveis e performáticos. Elas permitem que os dados sejam compartilhados entre sistemas de maneira eficiente e consistente, utilizando operações HTTP padrão, como *GET*, *POST*, *PUT* e *DELETE*, para executar ações como recuperação, criação, atualização e exclusão de recursos (Aws, 2024).

De Louzada (2024) destaca-se que um dos maiores benefícios das APIs RESTful é a sua capacidade de facilitar a integração entre sistemas heterogêneos. Independentemente das tecnologias utilizadas (linguagens de programação, *frameworks* ou plataformas), as APIs RESTful oferecem um meio universal para que diferentes serviços possam se comunicar. Isso é viável porque essas APIs utilizam formatos de dados amplamente aceitos, como JSON e XML, que podem ser facilmente interpretados por uma vasta gama de sistemas.

Além disso Aws (2024) afirma que as APIs RESTful promovem interoperabilidade, permitindo que componentes de software, desenvolvidos de forma independente, trabalhem em conjunto. Esse aspecto é especialmente valioso em ambientes corporativos, onde diferentes departamentos podem criar soluções distintas, mas ainda assim precisam garantir que essas soluções se comuniquem de maneira eficaz.

Outro aspecto é a escalabilidade. Como as APIs RESTful são projetadas para serem *stateless* (sem estado), cada solicitação é independente, permitindo que os sistemas sejam escalados horizontalmente com facilidade. Isso significa que múltiplas instâncias de um serviço podem ser adicionadas ou removidas conforme necessário, sem comprometer a integridade das interações entre os componentes. (Louzada, 2024).

3 Personalização de Serviços

A personalização de serviços envolve a adaptação de produtos ou serviços de acordo com as preferências e necessidades individuais dos usuários. Isso é alcançado por meio de técnicas como recomendações baseadas em conteúdo, filtragem colaborativa e análise preditiva de comportamento do usuário (Antonio, 2008).

Nesse contexto, a experiência de usuário personalizada é essencial para aumentar a satisfação e a fidelidade do cliente. Isso inclui a personalização da interface do usuário, o conteúdo apresentado e as funcionalidades oferecidas, de acordo com as preferências e o histórico de interações do usuário.

A coleta e o uso de dados pessoais para personalização de serviços, segundo Antonio (2008), levantam questões importantes de privacidade e segurança. É fundamental garantir que os dados dos usuários sejam protegidos e que o uso desses dados seja transparente e ético, em conformidade com regulamentações como LGPD.

Cada raça de cachorro possui características únicas, incluindo tamanho, temperamento, nível de atividade e necessidades de cuidados. Compreender essas características é fundamental para ajudar os tutores a escolher a raça mais adequada às suas necessidades e estilo de vida (Antonio, 2008).

Segundo o autor, o estilo de vida do tutor, incluindo fatores como espaço disponível, tempo dedicado ao exercício e presença de crianças ou outros animais de estimação, também influencia na escolha da raça mais adequada. Levar em consideração esses fatores é essencial para garantir uma correspondência personalizada bem-sucedida entre o tutor e o animal de estimação.

4 Inteligência Artificial e Sistemas de Recomendação

A Inteligência Artificial (IA) desempenha papel central na personalização de serviços, especialmente por meio de técnicas de Aprendizado de Máquina. Aprendizado supervisionado, não supervisionado e por reforço são fundamentais para o desenvolvimento de sistemas de recomendação eficazes, que conseguem uma alta assertividade (Marques, 2022).

4.1 GEMINI

A inteligência artificial da Google, desenvolvida pela Google AI, destaca-se como uma das plataformas mais avançadas do mundo. A trajetória da empresa no campo da inteligência artificial começou no início dos anos 2000, culminando no lançamento do Google Brain em 2011, um projeto pioneiro no uso de redes neurais profundas. Esse marco foi fundamental para avanços em áreas como visão computacional e processamento de linguagem natural (PLN) (Google AI, 2024).

Ao longo dos anos continuou a evoluir alcançando grandes conquistas como o desenvolvimento do AlphaGo em 2016, um algoritmo de aprendizado por reforço que derrotou campeões humanos no jogo de Go. Em 2018, a empresa lançou o BERT (*Bidirectional Encoder Representations from Transformers*), que transformou a forma como as máquinas entendem a linguagem humana, sendo amplamente utilizado em pesquisas online. Essas inovações refletem o impacto profundo da Google AI em setores como saúde, automação e serviços digitais (Google AI, 2024).

Atualmente, a Google AI mantém sua liderança no setor de inovação tecnológica, oferecendo soluções empresariais por meio do Google Cloud AI, que

permite o uso de *machine learning* em diversas aplicações. Além disso, a empresa continua a aperfeiçoar seus modelos de linguagem, incluindo os modelos *transformer*, que serviram de base para diversas outras inteligências artificiais avançadas, como o GPT.

O estado atual da Google AI reflete seu contínuo compromisso com o avanço da inteligência artificial em diversas frentes, desde a geração automática de texto e código, até a criação de *scripts*, peças musicais e ferramentas de comunicação automatizada, como e-mails e cartas. Em 2023, a Google introduziu o Bard, seu *chatbot* de inteligência artificial, projetado para proporcionar interações mais naturais e contextualizadas aos usuários, em uma tentativa de se manter à frente da concorrência no campo da IA conversacional (Google AI, 2024).

Sua plataforma de IA, segundo a empresa, treinada em vastos conjuntos de dados de texto e código, representa uma ferramenta poderosa para a criação de conteúdo criativo e técnico, consolidando sua presença em diversas indústrias e no cotidiano dos usuários. Ao mesmo tempo, a empresa se compromete a continuar investindo em IA ética e responsável, moldando o futuro da tecnologia com inovação e cuidado.

4.2 Algoritmos de Recomendação

Os algoritmos de recomendação desempenham um papel essencial na análise de grandes volumes de dados, tanto de usuários quanto de itens, para prever preferências e fornecer recomendações personalizadas. Esses algoritmos são amplamente utilizados em plataformas de e-commerce, *streaming* de mídia e redes sociais, entre outras. Existem diferentes abordagens para a construção de sistemas de recomendação, cada uma com suas particularidades, conforme detalhado no Quadro 2 segundo (Sacramento, 2022).

Quadro 2 - Sistemas de recomendação

Filtragem Colaborativa	Esse método baseia-se nas interações e preferências de usuários semelhantes para fazer recomendações. A filtragem colaborativa pode ser dividida em dois tipos: baseada em usuários e baseada em itens. Na abordagem baseada em usuários, o sistema recomenda itens que usuários com preferências semelhantes já gostaram. Na abordagem baseada em itens, a recomendação é feita com base na similaridade entre os itens consumidos. Um dos desafios desse método é o problema do "cold start", quando não há dados suficientes sobre um novo usuário ou item.
Filtragem Baseada em Conteúdo	Esse algoritmo utiliza as características dos itens, como palavras-chave ou descritores, para recomendar produtos semelhantes àqueles que o usuário já demonstrou interesse. Ao contrário da filtragem colaborativa, a filtragem baseada em conteúdo não depende de outros usuários, mas sim das preferências explícitas do próprio usuário. No entanto, esse método pode limitar a diversidade das recomendações, uma vez que tende a sugerir itens que são muito semelhantes aos consumidos anteriormente.

Redes Neurais e Deep Learning	Com o avanço do aprendizado profundo, as redes neurais começaram a ser aplicadas em sistemas de recomendação. Redes neurais podem capturar padrões complexos e não lineares nos dados, superando as limitações dos métodos tradicionais. Modelos como autoencoders, redes recorrentes (RNN) e redes neurais convolucionais (CNN) têm sido empregados para personalizar recomendações, utilizando grandes quantidades de dados. Um exemplo é o uso de embeddings para representar usuários e itens em espaços vetoriais, permitindo recomendações mais precisas e diversificadas.
Sistemas Híbridos	Muitos sistemas de recomendação modernos combinam diferentes abordagens para melhorar a precisão. Um sistema híbrido pode, por exemplo, integrar a filtragem colaborativa com a filtragem baseada em conteúdo, ou ainda incorporar redes neurais para melhorar o desempenho em dados não estruturados. A combinação dessas técnicas visa atenuar as limitações de cada método isolado e garantir recomendações mais robustas e relevantes.

Fonte: adaptado de Sacramento (2022)

4.3 Avaliação de Sistemas de Recomendação

A avaliação dos sistemas de recomendação é um aspecto crítico para garantir que as sugestões feitas aos usuários sejam eficazes e relevantes. Existem várias métricas usadas para medir o desempenho dos sistemas de recomendação, cada uma fornecendo uma perspectiva diferente sobre a qualidade das recomendações, como apresentado no Quadro 3 de acordo com Lunardi (2021).

Quadro 3 - Métricas de desempenho

Precisão (Accuracy)	Mede a proporção de itens recomendados que o usuário realmente gostou. A precisão é importante para garantir que o sistema ofereça recomendações que sejam úteis, mas sozinha pode não fornecer uma visão completa da qualidade do sistema.
Revocação (Recall)	Avalia a capacidade do sistema de recomendar todos os itens relevantes. Um sistema com alta revocação deve ser capaz de sugerir a maior quantidade possível de itens que atendam às preferências do usuário. No entanto, pode ocorrer que um sistema com alta revocação sacrifique a precisão.
Cobertura	Refere-se à proporção de todos os itens disponíveis que são recomendados pelo sistema. Um sistema com boa cobertura garante que uma ampla variedade de itens seja sugerida, evitando recomendações muito repetitivas ou previsíveis.
Diversidade	A diversidade mede a variedade dos itens recomendados. Um sistema de recomendação deve ser capaz de sugerir não apenas os itens mais populares ou previsíveis, mas também uma gama diversificada de opções. Um sistema com alta diversidade pode

	aumentar a satisfação do usuário ao apresentar novos itens ou categorias que ele não conhecia, evitando a monotonia nas recomendações.
Personalização	Essa métrica avalia o quão bem o sistema adapta suas recomendações às preferências individuais de cada usuário. Sistemas de recomendação eficientes devem ser capazes de aprender com as interações passadas do usuário e fazer sugestões que reflitam seus interesses específicos.
Serendipidade	Mede o quão inesperadas, mas relevantes, são as recomendações feitas ao usuário. Um sistema serendipitoso é capaz de surpreender o usuário com sugestões que ele não esperava, mas que são alinhadas aos seus gostos, aumentando a satisfação geral com a plataforma.

Fonte: adaptado de Lunardi (2021)

A combinação dessas métricas é fundamental para garantir que os sistemas de recomendação não apenas ofereçam sugestões precisas, mas também promovam uma experiência diversificada, útil e personalizada ao usuário.

5 Métodos e Ferramentas

Nessa seção são apresentados os métodos aplicados e as ferramentas utilizadas para a implementação do protótipo funcional para testes da hipótese de solução do projeto.

5.1 Python e TKInter

Python é amplamente utilizada em desenvolvimento web, ciência de dados, aprendizado de máquina e automação, tendo sido utilizada no *backend* do Canine Match para implementar a lógica de negócios, o sistema de recomendação e a integração com a API GEMIN.

Tkinter é a biblioteca padrão do Python para a criação de interfaces gráficas de usuário (GUIs). Ela oferece um conjunto de ferramentas para construir aplicações desktop com elementos interativos. Tkinter foi utilizado para desenvolver a interface do Canine Match, permitindo uma interação amigável e intuitiva entre os usuários e o sistema. A interface permite que os usuários insiram suas preferências e receba recomendações personalizadas de raças de cães. A ferramenta é uma escolha popular para a criação de GUIs em Python devido à sua simplicidade, flexibilidade e integração direta com a linguagem. Isso facilita o desenvolvimento e manutenção da aplicação, permitindo a criação rápida de protótipos e a fácil implementação de funcionalidades interativas (Python, 2021).

5.2 API do GEMINI

GEMINI é a inteligência artificial da Google AI, treinada em um enorme conjunto de dados de texto e código. É capaz de fornecer informações detalhadas e atualizadas sobre diversas raças de cães, incluindo características físicas, temperamento e necessidades específicas. A API foi utilizada para acesso à base de conhecimento da GEMINI para identificar características de raças de cães,

fornecendo dados em tempo real para o sistema de recomendação do Canine Match. A aplicação faz requisições à API GEMINI, documentadas em Gemini (2024), para obter informações atualizadas sempre que necessário.

A utilização da API GEMINI garante que as informações sobre as raças sejam precisas, atualizadas e abrangentes. Isso melhora a precisão das recomendações do aplicativo e reduz a necessidade de manutenção manual da base de dados, permitindo que a equipe se concentre em melhorar outras funcionalidades do aplicativo.

5.3 Metodologia Ágil

A metodologia ágil é um conjunto de práticas de gerenciamento de projetos que enfatiza a entrega incremental, colaboração constante e adaptação rápida às mudanças. Metodologias ágeis como Scrum e Kanban são comumente utilizadas para organizar o trabalho em ciclos curtos e iterativos. Este projeto utilizou a metodologia Kanban para organizar o desenvolvimento da aplicação em *sprints*. Cada *sprint* envolveu o planejamento, desenvolvimento, revisão e teste de funcionalidades específicas do Canine Match. A metodologia ágil permite uma adaptação rápida às necessidades dos usuários e mudanças nos requisitos, garantindo um desenvolvimento mais eficiente e alinhado com as expectativas dos *stakeholders*.

5.4 Ciclo de Vida do Desenvolvimento de Software (CVDS)

O CVDS é estruturado em fases, desde o planejamento da arquitetura do aplicativo, levantamento e análise dos requisitos, projeto, implementação, testes e atualizações. Cada fase é iterativa, permitindo melhorias contínuas ao longo do projeto. Durante a fase de planejamento, levantamento e análise, os requisitos foram coletados e priorizados. Na fase de projeto, a arquitetura da aplicação e a interface do usuário foram definidas. A fase de implementação envolveu a codificação e integração dos componentes do sistema. A fase seguinte incluiu testes de integração e testes de aceitação do usuário. Finalmente o protótipo funcional foi concluído para uso. A abordagem iterativa garante que cada componente da aplicação seja revisado e aprimorado regularmente. Isso resulta em um produto final de alta qualidade, com menor risco de falhas e maior alinhamento com os requisitos dos usuários.

5.5 Sistema de Recomendação Baseado em Conteúdo

O sistema de recomendação utiliza algoritmos baseados em conteúdo para comparar as características fornecidas pelos usuários com as características das raças de cachorro obtidas da API GEMINI. Algoritmos de recomendação baseados em conteúdo analisam as preferências dos usuários e as características dos itens, neste caso as raças de cães, para fornecer sugestões personalizadas. Foi implementado um algoritmo de recomendação que analisa os dados do questionário do usuário, como estilo de vida, preferências de atividades, e outros critérios relevantes. Esses dados são comparados com as informações das raças de cães obtidas da API GEMINI para gerar recomendações personalizadas. Este método permite gerar recomendações personalizadas e precisas, aumentando a satisfação do usuário com as sugestões fornecidas. A utilização de um sistema baseado em

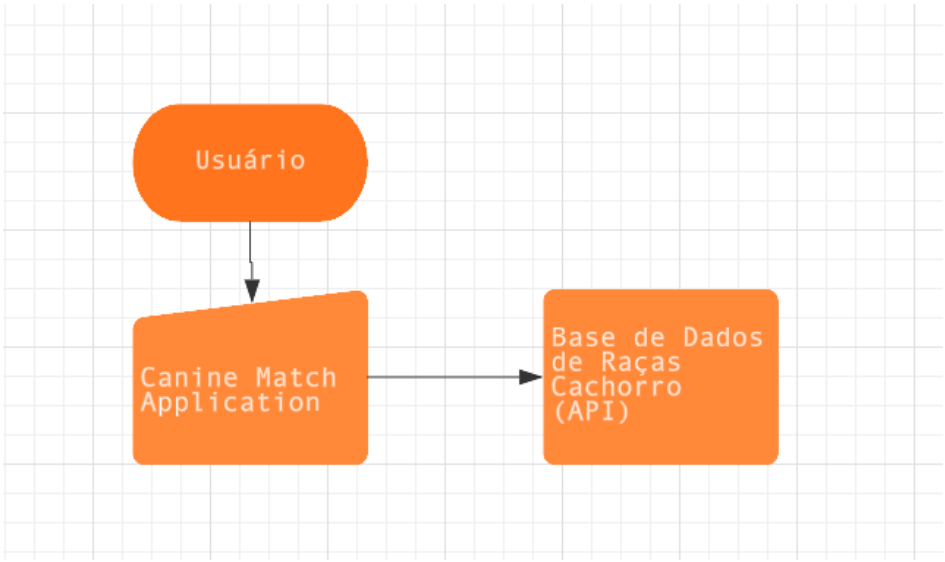
conteúdo garante que as recomendações sejam relevantes para os interesses e necessidades individuais dos usuários.

6 Desenvolvimento da Aplicação Proposta

A aplicação proposta seguiu uma arquitetura na qual o cliente interage com a interface da aplicação feita com *CustomTkinter*. Após a coleta dos dados por meios de um formulário, é realizada uma consulta na API do Gemini, a fim de coletar as melhores recomendações de raças de cachorro baseadas nas respostas do usuário.

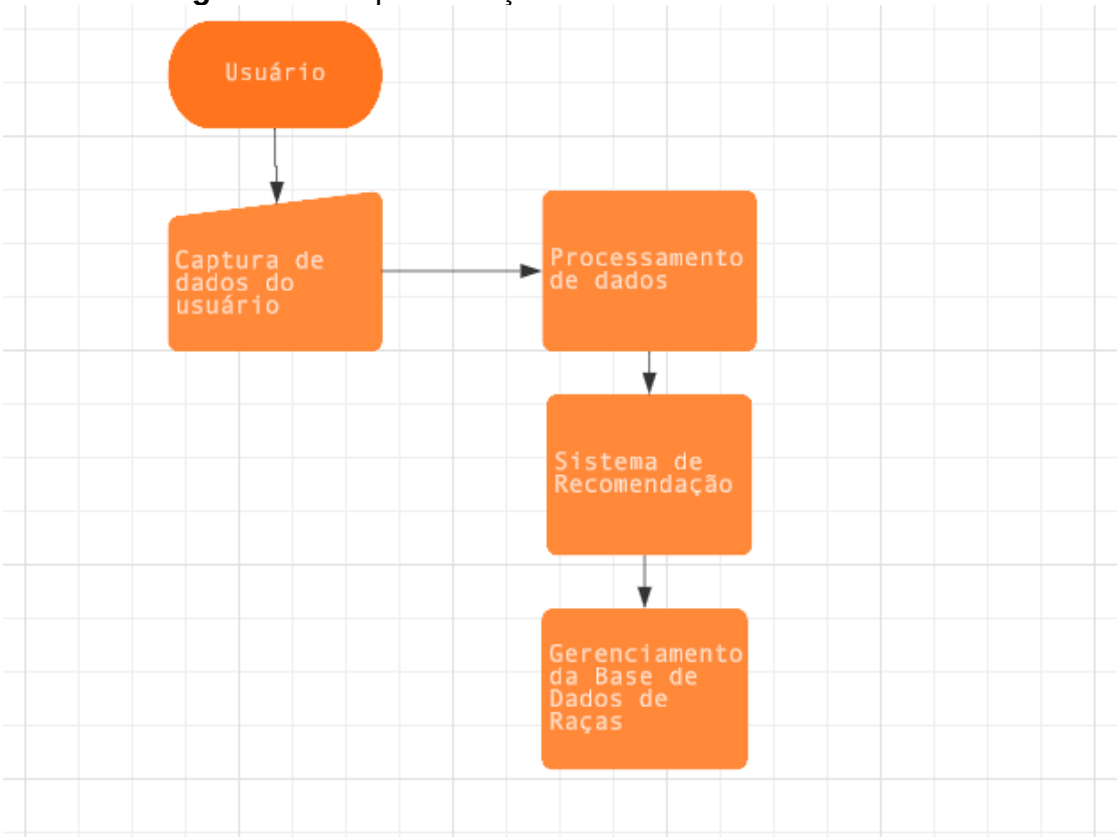
Foi implementado um sistema de recomendação baseado em conteúdo, que analisa as características fornecidas pelo usuário e as compara com as características das raças de cachorro para gerar recomendações personalizadas. As Figuras 1 e 2 representam os diagramas que embasaram a implementação do aplicativo com o objetivo de garantir que as recomendações sejam precisas e relevantes para as necessidades e preferências individuais dos usuários.

Figura 1 - Representação Gráfica do DFD de Nível 0



Fonte: autoria própria

Figura 2 - Representação Gráfica do DFD de Nível 1



Fonte: autoria própria

7 Coleta e Análise de Dados

Um questionário abrangente foi desenvolvido para coletar informações detalhadas sobre o estilo de vida, preferências e necessidades específicas dos usuários. O questionário inclui perguntas sobre o espaço disponível para o cachorro, nível de atividade do usuário, entre outros fatores. É utilizado no processo de *onboarding* do usuário para obter dados relevantes para o sistema de recomendação. Os dados coletados são armazenados em uma base de dados e utilizados para personalizar as recomendações. Coletar dados detalhados permite ao sistema de recomendação gerar sugestões mais precisas e personalizadas. Isso garante que as recomendações sejam adequadas ao estilo de vida e preferências dos usuários, aumentando a probabilidade de uma adoção bem-sucedida e satisfatória.

A base de conhecimento de raças de cães é acessada através da API GEMINI, que retorna informações detalhadas sobre diversas raças, incluindo características físicas, temperamento, necessidades de cuidados e níveis de atividade. A API GEMINI é atualizada regularmente, garantindo que as informações sejam sempre precisas e relevantes.

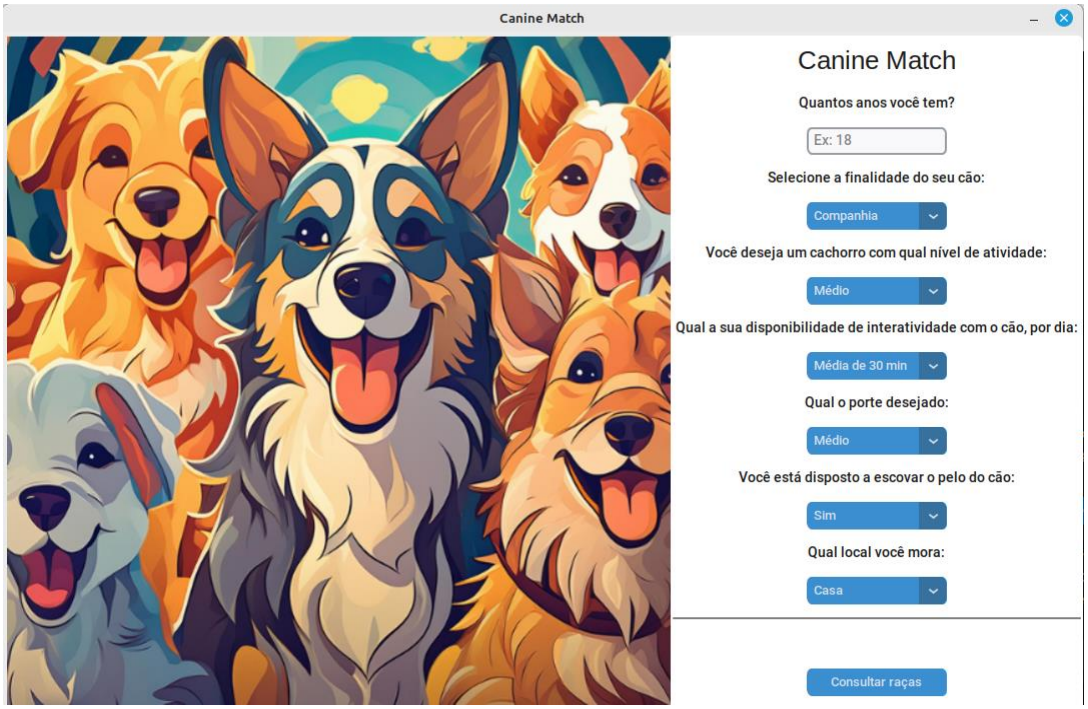
A aplicação faz requisições à API GEMINI para obter informações sobre as raças de cães. Essas informações são utilizadas pelo sistema de recomendação para comparar com as preferências dos usuários e gerar sugestões personalizadas. A utilização da API GEMINI garante acesso a uma base de conhecimento abrangente e sempre atualizada. Isso é essencial para a precisão do sistema de recomendação, permitindo que o Canine Match forneça informações confiáveis e úteis para os usuários.

8 Prototipagem de telas

A interface projetada e implementada é mostrada na Figura 4, na qual o usuário responde ao questionário com suas características para a busca das sugestões de raças, definido com base em Bonnati (2020).

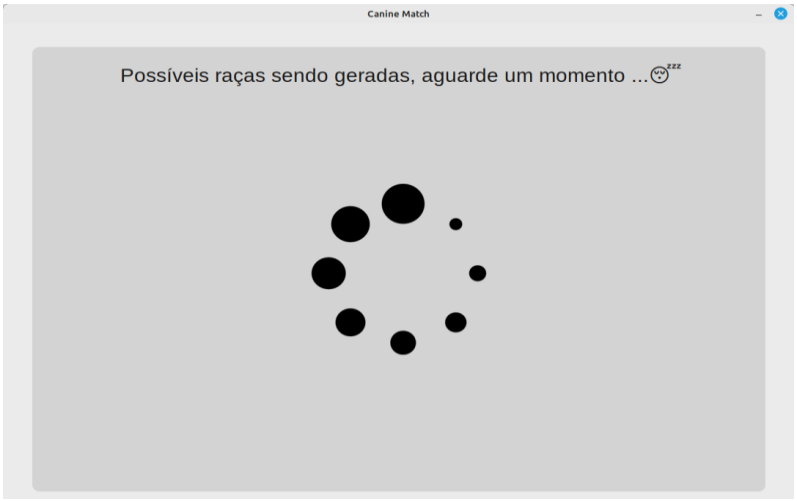
Na Figura 5 é mostrado o indicador de progresso (*spinner*) da execução da requisição da API com os dados coletados, em função do tempo de resposta ser usualmente superior a 1 segundo; e na Figura 6 é mostrada a apresentação do resultado retornado pela plataforma GEMINI.

Figura 4 – Tela inicial da aplicação



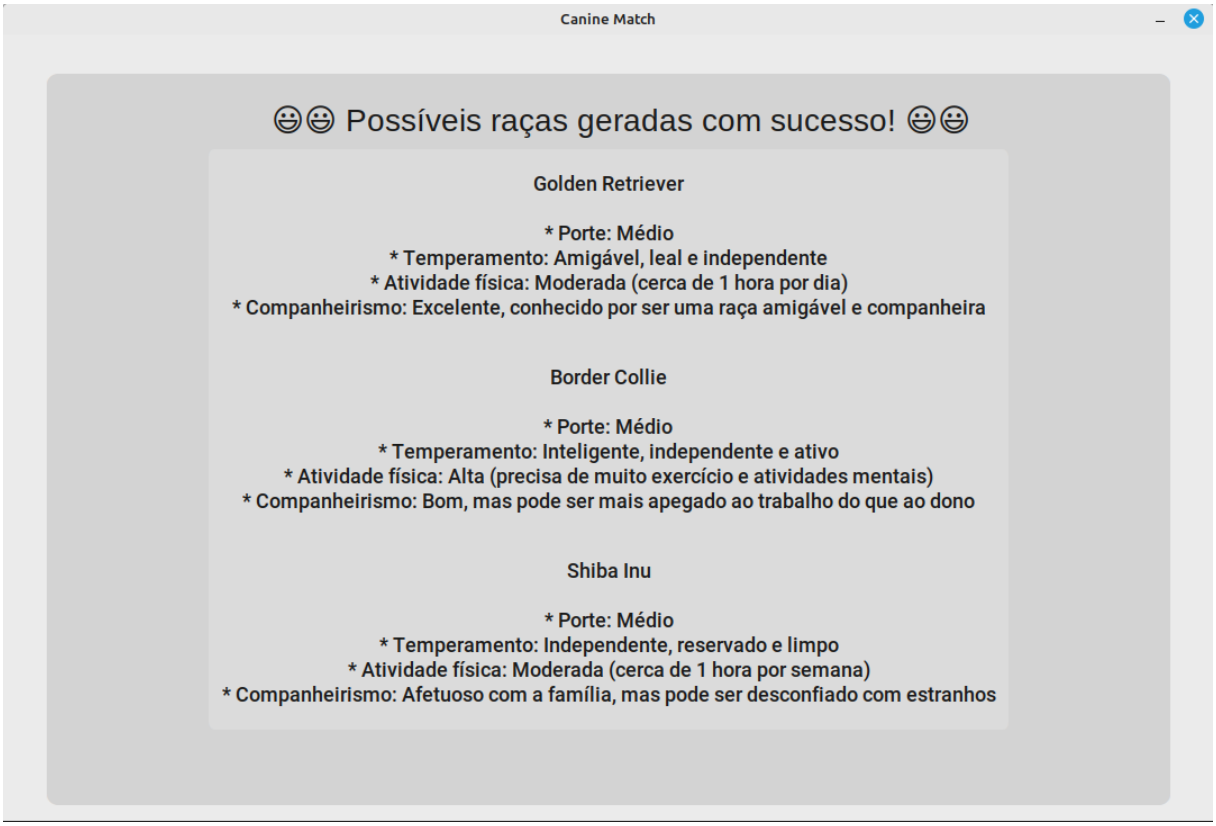
Fonte: autoria própria

Figura 5 – Tela de carregamento (Consultando API Gemini)



Fonte: Autoria própria

Figura 6 – Tela de sucesso com as raças sugeridas



Fonte: Autoria própria

9 Código implementado

Para desenvolvimento da aplicação foi necessário construir um código conciso (Figuras 7) que realiza a apresentação do formulário para coleta das características do usuário que são utilizadas para realizar o consumo da API do GEMINI.

A aplicação tem como principal ator a plataforma GEMINI, que tem sua API consumida através de um *prompt* (Figura 8) pré estabelecido porém personalizado com os dados inseridos no formulário pelo usuário, que ao serem analisados, são direcionados a uma resposta (Figura 9) contendo as raças de cães que mais se adéquam ao perfil do usuário.

Figura 7 – Trecho de código que constrói o formulário de questões ao usuário

```
title = customtkinter.CTkLabel(master=frame, text="Canine Match", font=("Arial", 26))
age_label = customtkinter.CTkLabel(master=frame, text="Quantos anos você tem?", font=("Roboto", 14))
age = customtkinter.CTkEntry(master=frame, placeholder_text="Ex: 18", font=("Roboto", 14))
purpose = customtkinter.CTkLabel(master=frame,text="Selecione a finalidade do seu cão:",font=("Roboto",14))
option_purpose = customtkinter.CTkOptionMenu(master=frame, values=["Companhia", "Guarda", "Apoio ao
Tutor"])
option_purpose.set("Companhia")
stroll = customtkinter.CTkLabel(master=frame, text="Você deseja um cachorro com qual nível de atividade:",
font=("Roboto", 14))
option_stroll = customtkinter.CTkOptionMenu(master=frame, values=["Baixo", "Médio", "Alto"])
option_stroll.set("Médio")
size = customtkinter.CTkLabel(master=frame, text="Qual o porte desejado:", font=("Roboto", 14))
option_size = customtkinter.CTkOptionMenu(master=frame, values=["Pequeno", "Médio", "Grande"])
option_size.set("Médio")
availability = customtkinter.CTkLabel(master=frame, text="Qual a sua disponibilidade de interatividade com
o cão, por dia:", font=("Roboto", 14))
option_availability = customtkinter.CTkOptionMenu(master=frame, values=["Média de 30 min a 1 hr", "Mais de
1 hr"])
option_availability.set("Média de 30 min")
brushing = customtkinter.CTkLabel(master=frame, text="Você está disposto a escovar o pelo do cão:", font=
("Roboto", 14))
option_brushing = customtkinter.CTkOptionMenu(master=frame, values=["Sim", "Não"])
option_brushing.set("Sim")
place = customtkinter.CTkLabel(master=frame, text="Qual local você mora:", font=("Roboto", 14))
option_place = customtkinter.CTkOptionMenu(master=frame, values=["Apartamento", "Casa", "Chacára"])
option_place.set("Casa")
divider = customtkinter.CTkFrame(master=frame, height=2, fg_color="gray")
```

Fonte: autoria própria

Figura 8 – Trecho de código que realiza a requisição para API do GEMINI

```
def return_ia():
    API_KEY = 'AIzaSyBeuFrnVhutue0YT8Ju39l1z-qCyw-wYnc'
    genai.configure(api_key=API_KEY)
    model = genai.GenerativeModel('gemini-pro')
    availability = ''
    brushing = ''

    if option_availability.get() == 'Média de 30 min':
        availability = ' O cão deve ser mais independente do seu dono; '
    if option_brushing.get() == 'Não':
        brushing = ' O cão deve ser de pelagem curta;'

    prompt = " Cite 3 raças de cachorro que se adequam ao perfil de um dono com as seguintes
    características: O dono tem:" + age.get() + " anos de idade; " +  " O cão deve exercer a finalidade
    de: " + option_purpose.get() + '; O cão deve ter necessidade ' + option_stroll.get() + 'de atividade
    física;' + availability + 'O cão deve ser de porte: ' + option_size.get() + '; ' + brushing + ' O cão
    deve se sentir bem em: ' + option_place.get() + '; Me apresente uma resposta direta, com as
    características detalhadas de cada raça citada, RETORNE NO MÁXIMO 750 CARACTERES, utilize essa outra
    resposta sua como exemplo do padrão de resposta: "**1. Beagle**\n\n* Porte: M\u00e9dio\n*
    Temperamento: Amig\u00e1vel, curioso e independente\n* Atividade f\u00edsica: Moderada (1 hora por dia)
    \n* Companheirismo: Excelente, afetuoso e leal\n\n**2. Shiba Inu**\n\n* Porte: M\u00e9dio\n*
    Temperamento: Independente, inteligente e reservado\n* Atividade f\u00edsica: Moderada (1-2 horas por
    semana)\n* Companheirismo: Afetuoso com a fam\u00edlia, mas pode ser desconfiado com estranhos\n\n**3.
    Akita**\n\n* Porte: Grande (masculino at\u00e9 65 cm, feminino at\u00e9 61 cm)\n* Temperamento:
    Independente, leal e protetor\n* Atividade f\u00edsica: Moderada (1 hora por dia)\n* Companheirismo:
    Guardi\u00e3es devotados e companheiros amorosos"'
    response = model.generate_content(prompt)
```

Fonte: autoria própria

Figura 9 – Trecho responsável por exibir a tela de sucesso

```
def process_return_ia(success_frame, title_success):
    result = return_ia()

    # Removendo mensagem de loading
    title_success.destroy()

    title_frame_success = customtkinter.CTkLabel(master=success_frame,
        text="😃😃 Possíveis raças geradas com sucesso! 😃😃",
        font=("Arial", 28))
    title_frame_success.place(x=200, y=25)

    # Inicia o frame menor com as respostas
    success_in_frame = customtkinter.CTkFrame(master=success_frame, width=750, height=650)
    success_in_frame.place(relx=0.5, rely=0.5, anchor="center")

    text = re.sub(r'\*\*', '', result)

    # Usando re.split() para dividir a string pelos números seguidos de pontos
    partes = re.split(r'\d+\.\s', text)

    # Remover a primeira entrada vazia
    partes = [parte.strip() for parte in partes if parte.strip()]

    # Criação e posicionamento dos labels
    for index, parte in enumerate(partes):
        label = customtkinter.CTkLabel(master=success_in_frame, text=parte, font=("Roboto", 18))
        label.pack(padx=12, pady=20)
```

Fonte: autoria própria

10 Testes

Para avaliar a precisão e eficácia do Canine Match, foram realizados testes em ambiente controlado com a participação de indivíduos reais que possuíam perfis variados, representando diferentes estilos de vida e preferências. Esses testes visam verificar a assertividade do sistema de recomendação do aplicativo, que utiliza a API de IA Gemini para sugerir raças caninas compatíveis com cada usuário.

Os participantes forneceram informações detalhadas por meio de um questionário, incluindo aspectos sobre suas rotinas, níveis de atividade e preferências quanto às características dos cães. A partir dessas respostas, o sistema aplicou algoritmos de recomendação baseados nas características das raças de cães, retornando sugestões personalizadas para cada perfil. A avaliação da assertividade foi realizada ao comparar as raças sugeridas com o feedback dos participantes, que confirmaram se as sugestões correspondiam às suas expectativas e estilo de vida. Os resultados mostraram que o Canine Match, com o suporte da API Gemini, foi capaz de fornecer recomendações precisas, indicando uma alta taxa de sucesso no alinhamento entre os perfis dos usuários e as raças recomendadas. Esses testes validaram a funcionalidade do aplicativo e demonstraram a capacidade do sistema de promover escolhas informadas e compatíveis.

11 Considerações finais

A proposta desse projeto surgiu a partir da percepção de uma necessidade crítica enfrentada por muitos tutores ao escolher uma raça de cão que se alinhe com seu estilo de vida e expectativas. Muitas vezes, a falta de informação precisa leva a decisões que podem resultar em frustração tanto para o tutor quanto para o animal, comprometendo a qualidade de vida e a convivência harmoniosa.

Ao desenvolver este projeto, buscou-se criar uma solução que facilite o processo de escolha, considerando não apenas as preferências do tutor, mas também as necessidades específicas de cada raça, com o principal objetivo de garantir que ambos os lados, tutor e cão possam desfrutar de uma convivência longa, satisfatória e, acima de tudo, harmoniosa.

Com o resultado alcançado, é possível promover o bem-estar animal e fortalecer os laços entre humanos e seus companheiros caninos.

Referências

ANTONIO. Usabilidade de Software. [S. l.]: DEVMEDIA, 2008. Disponível em: https://www.devmedia.com.br/usabilidade-de-software/10246, Acesso em: 3.ago. 2024.

AWS. O que é uma API RESTful? s.d. Disponível em: https://aws.amazon.com/pt/what-is/restful-api/. Acesso em: 27.set.2024.

BONNATI, Camila. Raça de Cachorro: Como escolher o cachorro ideal? O guia completo para te ajudar a tomar uma decisão. [S. l.]: Portal Do Dog, 2020. Disponível em: https://www.portaldodog.com.br/cachorro/racas/raca-de-cachorro-como-escolher/, Acesso em: 22.jul.2024.

CARVALHO, Caroline. O que é Python? — um guia completo para iniciar nessa linguagem de programação. [S. l.]: Alura, 2020. Disponível em: https://www.alura.com.br/artigos/python, Acesso em: 1.ago. 2024.

GEMINI. Introdução de documentação do GEMINI. [S. l.]: Gemini developers, 2024. Disponível em: https://ai.google.dev/gemini-api/docs?hl=pt-br, Acesso em: 29.jul.2024.

GOOGLE AI. Our AI Journey. s.d. Disponível em: https://ai.google/ai-milestones?section=intro. Acesso em: 27.set.2024.

PYTHON. Interfaces Gráficas de usuário com TK. [S. l.]: Python Docs, 2001. Disponível em: https://docs.python.org/pt-br/3/library/tk.html, Acesso em: 29.jul.2024.

LOUZADA, V.; CARVALHO, C.; LARANJA, E. API: o que é, para quê serve e qual é a sua importância. 2024. Disponível em: https://www.alura.com.br/artigos/api. Acesso em: 27.set.2024.

LUNARDI, ARTUR. Entendendo Sistemas de Recomendação. [S. l.]: Arturlunardi medium, 2021. Disponível em: https://arturlunardi.medium.com/entendendo-sistemas-de-recomenda%C3%A7%C3%A3o-c50a20856394, Acesso em: 2.ago.2024.

MARQUES, Ana. Como funcionam os sistemas de recomendação ?. [S. l.]: Tecnoblog, 2022. Disponível em: https://tecnoblog.net/responde/como-funcionam-os-sistemas-de-recomendacao/, Acesso em: 5.ago.2024.

SACRAMENTO, Gabriel. SISTEMAS DE RECOMENDAÇÃO: COMO FUNCIONAM E EXEMPLOS PRÁTICOS. [S. l.]: Blog Somostera, 2022. Disponível em: https://blog.somostera.com/data-science/sistemas-de-recomendacao, Acesso em: 7.ago.2024.