

TenisBOT: Implementação de uma Aplicação para Recomendação de Calçados Esportivos Baseada em Modelos de Similaridade de Perfil

Marcos Antonio Speca Junior
Universidade Presbiteriana Mackenzie
São Paulo - SP, Brasil
marcos@speca.com.br

Fabio Silva Lopes
Universidade Presbiteriana Mackenzie
São Paulo - SP, Brasil
<https://orcid.org/0000-0001-8274-7682>

Abstract—Choosing the ideal footwear is crucial for safe and effective running. However, this decision often happens empirically, without the support of technologies to make the process more accurate. These factors led to the development of TennisBot, a recommendation system that blends a user profile similarity-based recommendation model with a conversational interface. The product development used Design Science Research (DSR) to build the open application architecture. Its implementation involved ten distinct technologies, and an initial dataset of 278 evaluations was assembled to generate recommendations. TennisBot was made available via WhatsApp, leveraging a Large Language Model (LLM). Evaluations with users showed that the recommendations were relevant and effective, validating the application's approach.

Keywords—Recommendation by similarity; Large Language Models; Conversational interface; Open application architecture.

Resumo—Definir o calçado ideal é importante para a prática segura e eficaz da corrida, contudo, a escolha ocorre de modo empírico e sem o suporte de tecnologias capazes de tornar este processo mais acurado. Estes fatores motivaram o desenvolvimento do TennisBot, um sistema de recomendação que integra um modelo de recomendação baseado em similaridade de perfil de usuários a uma interface conversacional. O desenvolvimento do produto empregou a metodologia Design Science Research (DSR) como base para desenvolver a arquitetura de aplicação aberta. A implementação envolveu dez tecnologias distintas e um dataset inicial com 278 avaliações foi composto para gerar as recomendações. A aplicação foi disponibilizada via WhatsApp, utilizando um Large Language Model (LLM) na versão gratuita. Uma avaliação com usuários foi realizada e demonstrou que as recomendações geradas foram relevantes e eficazes, de modo a validar a proposta da aplicação.

Palavras-chave—Recomendação por similaridade; Interface conversacional; Arquitetura de aplicação aberta.

I. INTRODUÇÃO

A corrida de rua destaca-se como uma das modalidades esportivas mais populares globalmente, ocupando a terceira

posição em prática regular, atrás apenas da caminhada e do futebol [1]. Apesar de sua acessibilidade, a escolha do calçado adequado é um desafio complexo para os corredores. Fatores como conforto, melhoria de desempenho e prevenção de lesões são essenciais, e a decisão é ainda mais complicada pela vasta gama de opções disponíveis no mercado, cada uma com características técnicas distintas [2]–[4].

O problema de sobrecarga de informações na escolha de tênis de corrida para atletas amadores pode ser abordado sob uma perspectiva analítica. Em um cenário onde a quantidade de opções supera a capacidade de análise individual do usuário, a aplicação de Sistemas de Recomendação (SR) pode gerar soluções interessantes. Esses sistemas são projetados para mitigar a sobrecarga de informações, conectando usuários a itens de maior relevância com base em suas preferências e nas características dos produtos. Sendo assim, pode-se observar este problema na perspectiva analítica e inferir que a tecnologia pode auxiliar os corredores (usuários) a escolherem o seu tênis (item) dentre as diversas opções, através de um Sistema de Recomendação (SR).

De modo complementar, uma arquitetura aberta permite que o sistema seja construído com componentes de diferentes fontes e tecnologias, utilizando padrões abertos e amplamente aceitos. No caso específico, este cenário poderia colaborar com a promoção da inovação em uma aplicação experimental capaz de atender a requisitos de modularidade, flexibilidade e escalabilidade, alinhada com princípios da pesquisa científica.

Assim, o problema de pesquisa central deste trabalho pode ser formalizado como: Sistemas de Recomendação baseados em Machine Learning podem auxiliar atletas amadores a selecionar o tênis de corrida mais adequado, considerando suas preferências pessoais e as características técnicas dos calçados?

Em resposta a essa questão, o objetivo principal deste estudo foi desenvolver e implementar uma arquitetura aberta

para um Sistema de Recomendação de tênis de corrida para atletas amadores. O sistema será concebido para fornecer recomendações personalizadas e otimizadas, baseadas em dados de preferências e características individuais dos atletas.

Este artigo está dividido da seguinte forma: Na seção II estão descritos os fundamentos teóricos e a seção II-B, o Mapeamento Sistemático da Literatura (MSL) realizado. Na sequenciam, a seção III descreve a metodologia utilizada no projeto e a seção IV apresenta as etapas executadas para desenvolver a arquitetura do sistema de recomendação e a aplicação. Os resultados são apresentados na seção VI e, por fim, as conclusões são apresentadas na seção VII.

II. REVISÃO DA LITERATURA

A. Conceitos Envolvidos

Podemos classificar os Sistemas de Informação (SI) que utilizam recomendação em dois grupos: Os SI que utilizam SR como meio para apoio às atividades do sistema, e os SI que utilizam SR como fim, com aderência ao perfil do usuário que o utiliza [5]. Neste estudo, optamos por seguir com a aplicação classificada na segunda categoria, considerando que o sistema deve identificar similaridade a um perfil de usuário para recomendar o tênis mais adequado.

Exemplos destes sistemas podem ser encontrados em aplicações conhecidas, como o algoritmo de recomendação de programas de TV da Netflix, sugestão de produtos da Amazon, sugestões de amigos do Facebook, e recomendação de músicas do Spotify [6], [7]. Estes são apenas alguns exemplos, pois este tipo de algoritmo está presente em diversas outras aplicações.

Não obstante, os SR podem ser classificados de acordo com suas fontes de dados e estratégias de recomendação: Filtragem Colaborativa (*Collaborative Filtering*), Baseados em Conteúdo (*Content-Based*), Baseados em Conhecimento (*Knowledge-based*), Raciocínio baseado em casos (*Case-based Reasoning*) e Híbridos [6]–[9].

A definição de qual tipo de algoritmo é mais adequado ao problema, é orientada pela existência prévia de um *dataset* de avaliação, o tipo de avaliação disponível (se é explícita ou implícita) e a disponibilidade dos dados característicos dos usuários e dos itens [8].

Considerando os pontos supracitados, optou-se por seguir com um Mapeamento Sistemático da Literatura para reunir subsídios atrelados ao estado da arte da pesquisa sobre SR.

B. Mapeamento Sistemático de Literatura

Mapeamentos sistemáticos de literatura (MSL) tem como objetivo dar uma visão ampla sobre o estado da arte de um tema selecionado, e diferente da revisão sistemática de literatura, o mapeamento não se propõe a realizar uma análise crítica dos

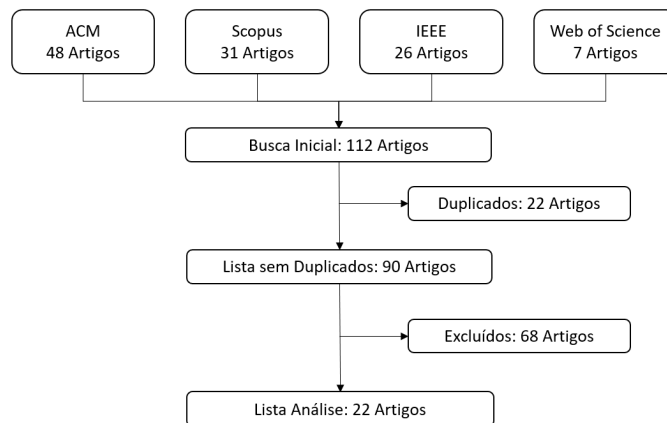


Fig. 1. Síntese do Mapeamento Sistemático da Literatura.

Fonte: Dos autores

resultados de estudos primários, mas sim traçar um panorama geral do estado da arte sobre o tema, que pode ser um ponto de partida para pesquisas futuras [10].

Neste MSL, foram consideradas as seguintes bases de dados indexadas: ACM Digital Library¹, IEEE Xplore², SCOPUS³, e Web of Science⁴. Foram realizadas buscas nas bases em português SOL⁵ e Scielo⁶, porém nenhum resultado foi encontrado dentro dos mesmos critérios de busca em português.

A Figura 1 apresenta a síntese do Mapeamento sistemático da Literatura. No final do fluxo, foram selecionados 22 artigos. Dentre os quais, apenas 2 se propuseram a recomendar o calçado, e mesmo assim a recomendação ocorre em nível de categoria de calçado e não de um modelo específico, sendo que os itens mais comuns em recomendação estão relacionados com treinamento e ritmos de prova.

Como critério de seleção, foram definidas 6 questões de pesquisa para avaliar aspectos relevantes dos artigos analisados, as perguntas e seus objetivos estão descritas no Quadro 1.

¹<https://dl.acm.org/>

²<https://ieeexplore.ieee.org/>

³<https://www.scopus.com>

⁴<https://webofscience.com/wos>

⁵<https://sol.sbc.org.br/>

⁶<https://www.scielo.br/>

Pergunta	Objetivo
P01. Qual o objetivo principal do artigo?	Esta questão tem a intenção de mapear qual era o objetivo principal do artigo a fim de entender quais tipos de problema estão sendo tratados nos artigos relacionados.
P02. Foram aplicados algoritmos de recomendação? Se sim, quais?	Esta questão visa mapear quais algoritmos de recomendação estão sendo aplicados, dentre as categorias discutidas na seção II deste artigo.
P03. Como as bases de dados de usuários e itens foram construídas ou coletadas?	Esta questão visa entender como foi realizada a coleta dos dados que é um passo importante para o sucesso dos sistemas de recomendação, tanto dados dos usuários quanto dos itens recomendados.
P04. Quais tipos de itens foram recomendados?	Esta questão mapeia quais itens foram recomendados aos corredores e suas características, pois é relevante para o mapeamento entender se estão sendo recomendados itens além dos calçados esportivos.
P05. Como o sistema de recomendação foi avaliado?	Esta questão visa entender como os sistemas de recomendação foram testados e avaliados. Dentre as opções tradicionais existem três abordagens para a validação: Estudos com Usuários, métodos on-line e métodos off-line. [8].
P06. Houve algum desenvolvimento de interface com usuário?	Outro aspecto importante a se mapear é como houve a interação com os usuários, se houve algum desenvolvimento de interface do usuário para coleta dos dados e/ou recebimento das recomendações.

Quadro 1 - Perguntas do Mapeamento Sistemático

Entre os artigos selecionados, os modelos e algoritmos de recomendação mais utilizados foram das categorias *content-based* e *case-based reasoning*, o que pode ser explicado pela falta de dados de avaliações dos usuários prévia e, portanto, os algoritmos focaram em utilizar os dados dos próprios usuários e itens para encontrar similaridades e realizar a recomendação.

De modo complementar, observou-se o ecossistema de Open-Source software (OSS) para as opções de integração e desenvolvimento da aplicação.

III. METODOLOGIA

Este estudo se pautou no Design Science Research (DSR) [11] como base para alcançar os objetivos propostos. Além

do rigor científico que o método propicia, a generalização da solução para a mesma classe de problemas faz parte das etapas deste método, podendo assim contribuir ainda mais para o conhecimento científico, além da solução específica ao problema em questão. A Figura 2 apresenta as etapas e atividades do DSR adaptados a este estudo:

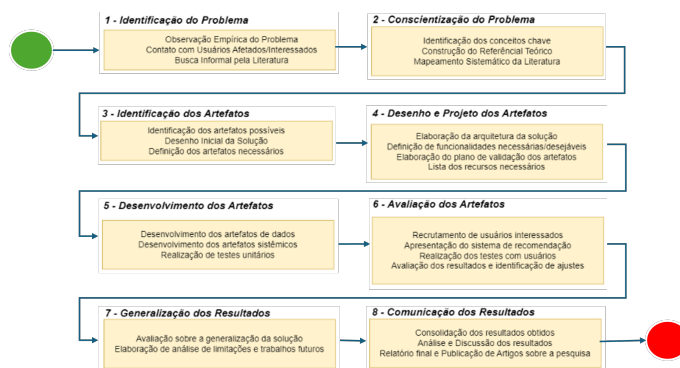


Fig. 2. DSR adaptada para o desenvolvimento do TenisBot

A Etapa 1 foi iniciada com a observação empírica do contexto de corrida de rua e a formulação do problema de pesquisa. Em seguida, uma busca informal na literatura foi realizada a fim de encontrar possíveis soluções sistêmicas e algorítmicas para este problema, e foi neste ponto que foram identificados os algoritmos de recomendação, que serviram de subsídio para a Etapa 2, onde executou-se o MSL.

Na Etapa 3 estruturou-se a arquitetura da aplicação, a partir da identificação de todos os artefatos sistêmicos da solução, ou seja, bases de dados, algoritmos, interfaces e funções que compõem a solução proposta, apresentados na Figura 3.

As Etapas 4 e 5 seguiram os procedimentos tradicionais de desenvolvimento de software. Na Etapa 6 aplicou-se um questionário de avaliação do aplicativo e uma avaliação SUS para avaliar a interface.

Após a avaliação da solução desenvolvida uma análise sobre a generalização para a classe dos problemas foi realizada na Etapa 7. Por fim, a Etapa 8 foi consolidada neste artigo.

IV. ARQUITETURA DO SISTEMA TÊNISBOT

Para que o sistema de recomendação seja acessado por diversos usuários, optou-se pela criação de uma aplicação "chat" disponibilizada via plataforma de mensagens WhatsApp, portanto, a arquitetura do sistema de recomendação possui três principais camadas: A camada conversacional, com a qual o usuário fará a interação com o sistema de recomendação, para informar suas características e receber recomendações; A camada da aplicação, que compreende a lógica e os algoritmos de busca de informações, atualização de perfil e algoritmo

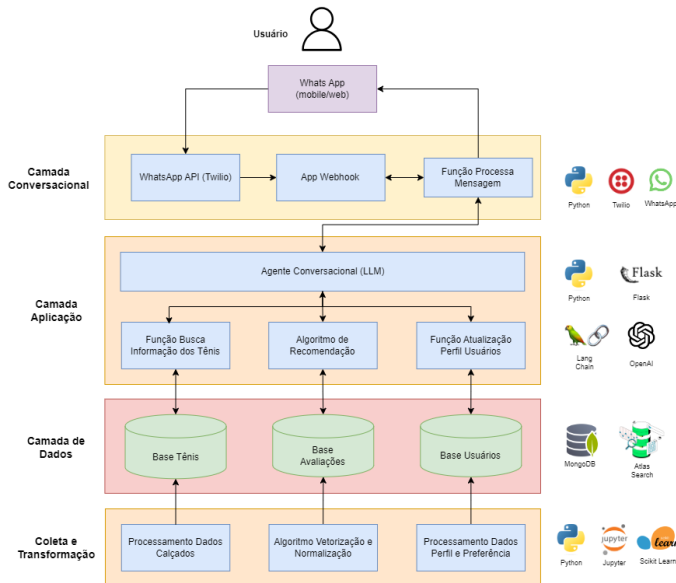


Fig. 3. Camadas e Tecnologias utilizadas na Aplicação TennisBot.
Fonte: Dos autores

de recomendação; E a camada de dados, que compreende as bases de dados referentes aos perfis de usuários, avaliações e características dos tênis de corrida.

A arquitetura da aplicação é apresentada na Figura 3, onde as camadas de implementação e respectivas tecnologias estão explicitadas.

A. Camada Conversacional

Esta camada é responsável pela integração do sistema de recomendação com as APIs da plataforma WhatsApp. Para a comunicação nesta plataforma foram necessários três componentes:

- **Número de telefone digital** Um número de telefone digital foi necessário para a criação de uma conta real de WhatsApp Business para que pudesse ser acessado por qualquer número de celular via WhatsApp. Um número de telefone digital foi obtido através de fornecedores especializados em VOIP. Foi escolhida a empresa BR DID como fornecedor do número digital.
- **Broker de mensagens** Uma plataforma "Broker" de mensagens foi utilizada para tornar o processo de integração com a API WhatsApp Business mais simples, além de contar com recursos como monitoria das mensagens, log de erros detalhados e fila de mensagens. É possível utilizar a API WhatsApp Business sem a plataforma Broker, no entanto todos os recursos adicionais de monitoria e log de erros teriam de ser implementados no sistema diretamente. Utilizamos a plataforma "Twilio".

- **Webhook da aplicação** Um Webhook é uma técnica utilizada em sistemas Web que permite receber mensagens e informações sem a necessidade de uma nova solicitação. Esta técnica é útil para receber mensagens da API Whatsapp de forma assíncrona, sem a necessidade de manter uma conexão contínua das APIs com o sistema de recomendação.

B. Camada da Aplicação

Esta camada consiste em uma aplicação em Python utilizando o framework Flask, que possui diversos recursos e funções para servir páginas dinamicamente. Dentro desta camada foram desenvolvidos os principais algoritmos da arquitetura, o agente conversacional, algoritmo de recomendação e funções de atualização de dados e busca de informações sobre os tênis.

Agente conversacional: Optou-se por utilizar os grandes modelos de linguagem (Large Language Models - LLMs) para apoio à interface conversacional. Utilizou-se as capacidades de NLP destes modelos para estruturar os inputs dos usuários que buscam recomendações, acionar o algoritmo de recomendação e devolver as recomendações em formato de linguagem natural, sem necessidade de implementação de diversas telas e formulários para coleta e processamento dos dados.

O agente conversacional foi construído utilizando o LangChain. Este framework auxilia no desenvolvimento de aplicações baseadas em LLMs, com diversas capacidades estendidas, como: criação de templates de "prompts", manter uma memória interna durante a execução com os detalhes da conversa, e também a capacidade de executar funções específicas em determinados momentos chave da conversa.

Para a seleção do LLM, optou-se por avaliar os modelos previamente analisados em [12]. Não obstante, o custo e a performance de cada modelo também foi avaliado. Por fim, para esta implementação foram utilizados os modelos GPT-3.5-Turbo e GPT-4-Turbo da OpenAI, porém a arquitetura se manteve agnóstica ao modelo, e pode receber qualquer outro LLM através de APIs.

O agente conversacional desta aplicação possui três funções definidas: Atualização de cadastro do usuário (caso o usuário queira atualizar algum detalhe de seu perfil durante a conversa), Busca de informações detalhadas no *dataset* de tênis de corrida, e o acionamento do algoritmo de recomendação, passando informações do perfil do usuário e recebendo nome de calçados para recomendar ao usuário.

O fluxo da Figura 4 exemplifica como o agente conversacional pode chamar uma função externa, neste exemplo o caso de recomendação de tênis de corrida, no entanto as demais funções são acionadas de forma similar quando

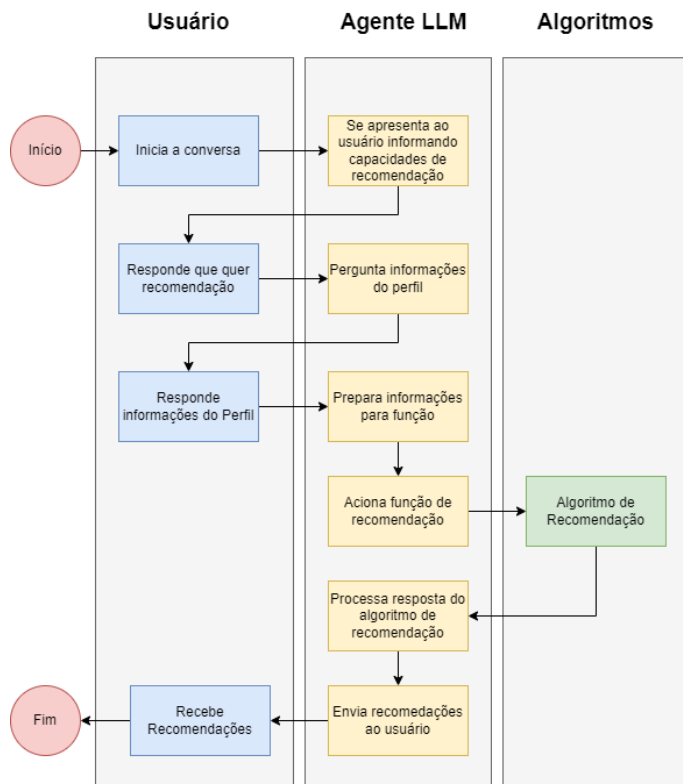


Fig. 4. Fluxo conversacional de recomendação
Fonte: Dos autores

forem necessárias durante o processamento da conversa com o usuário.

C. Camada de Dados

Os *datasets* da aplicação são três: Usuários, Avaliações e Calçados. Este último foi coletado por meio de webcrawler em sites de marcas e e-Commerce). A tecnologia utilizada para os *datasets* foi o MongoDB porque, entre outros qualificadores, permite busca vetorial e atende as necessidade de armazenamento e recuperação de dados.

De modo complementar, vale destacar a etapa de pré-processamento dos dados. Utilizou-se aqui cargas de dados iniciais através de processos "batch" (massivos). Para isso foram desenvolvidos dois scripts em Jupyter Notebooks para depois carregar os *datasets* de calçados e pré-processar os perfis dos usuários, aplicando as regras de pré-processamento e normalização necessárias.

D. Algoritmo de recomendação

Um dos problemas associados aos sistemas de recomendação é denominado início "a frio" (do inglês "*cold-start problem*")

[8]. Basicamente, algoritmos de recomendação precisam de dados iniciais para a modelagem da preferência dos usuários, porém muitas vezes estes dados não existem ou não estão disponíveis de forma adequada para se realizar as primeiras recomendações. Para minimizar este problema, uma base inicial de avaliações foi coletada através de pesquisa com os usuários através de um formulário on-line, contendo perguntas sobre características pessoais dos corredores, hábitos de treino e a avaliação pessoal de calçados específicos.

Os respondentes indicaram: idade, altura, peso, sexo, tipo de pisada (pronada/supinada/neutra), número de treinos por semana e volume de treino semanal em quilômetros. De modo complementar, cada usuário avaliou até três modelos tênis de corrida diferentes, por meio de nota em escala de 1 a 5. Ao todo foram coletados 278 avaliações de 125 usuários diferentes. Este *dataset* inicial foi processado e estruturado em formato vetorial para possibilitar a utilização do algoritmo de similaridade numérica na identificação de usuários similares.

Após a vetorização dos usuários e armazenamento em uma matriz normalizada, o algoritmo de recomendação pode realizar a recomendação. Neste ponto, os experimentos iniciais utilizaram uma versão mais simples de cálculo de similaridade considerando apenas os dois vetores (do usuário *target* e das avaliações).

Com a estabilidade da versão simplificada, a versão completa do algoritmo utilizada. Ela consiste em buscar os usuários com características similares através do cálculo de "similaridade por cosseno" e recupera os três usuários com o perfil mais próximo. Após isso, o algoritmo recupera todos os modelos avaliados pelos três usuários similares e calcula as maiores notas, considerando a avaliação média caso os mesmos modelos tenham sido avaliados mais de uma vez com notas diferentes. Por fim, ordena os três tênis mais bem avaliados e retorna estes modelos como recomendações.

Este algoritmo pode ser categorizado como "*demographic-based recommender system*", pois utiliza de dados demográficos do usuário para encontrar outros usuários similares, levando em conta as avaliações apenas na hora de ordenar seus resultados. O Código 1 detalha a estratégia de recomendação adotada.

Para o desenvolvimento deste algoritmo utilizou-se a biblioteca Python Scikit-learn para aplicação do cálculo de similaridade por cosseno e a normalização dos dados para uniformizar as escalas.

V. IMPLEMENTAÇÃO

Através da arquitetura proposta, foi criada a aplicação "TenisBot", um chatbot inteligente que pode recomendar tênis de corrida aos corredores baseado em suas próprias características através de processamento de linguagem natural. Esta

Algoritmo 1: Recomendação de tênis de corrida

- 1: **procedure** RECOMENDARTÊNISCORRIDA
- 2: Pré-processa vetor do usuário;
- 3: Normaliza vetor usuário na mesma escala da base;
- 4: Busca os top 3 vetores dos usuários mais similares;
- 5: Recupera as avaliações dos top 3 usuários similares;
- 6: Ordena calçados por média de avaliações mais alta;
- 7: Retorna as 3 primeiras recomendações;
- 8: **end procedure**

Resultado: Obtém as 3 recomendações de perfis similares ordenados por nota.



Fig. 5. Exemplo de interação efetuada na aplicação

implementação inicial foi realizada a fim de validar a qualidade do sistema de recomendação de forma on-line com os usuários, uma vez que não há dados robustos suficientes para uma validação off-line.

A Figura 5 apresenta um exemplo de interação executada no sistema.

VI. TESTES REALIZADOS

Para os testes iniciais da aplicação e o sistema de recomendação, foi utilizada a plataforma Render, que é um serviço de hosting de aplicações. Optou-se por este provedor

pois ele possui uma oferta inicial gratuita de hosting para carga de processamento reduzido, o que é adequado ao projeto em etapa de testes iniciais, porém permitindo a escala futura.

Foram realizados testes do sistema de forma "on-line". O formulário foi disponibilizado contendo 15 perguntas e um campo de comentário livre, em duas sessões: a primeira permitiu avaliar a qualidade das recomendações com 5 perguntas específicas sobre a recomendação, e a segunda serviu para avaliar a usabilidade do sistema, utilizando o método "System Usability Scale" (SUS) que consiste em 10 perguntas respondidas em escala Likert, de 1 a 5 (discordo totalmente e concordo totalmente, respectivamente) [13].

Mais de 130 corredores amadores receberam o convite para participar dos testes. Trinta deles realizaram trocas de mensagens iniciais com o TenisBot via chat WhatsApp, mas alguns não concluíram o processo de recomendação (três respondentes) e dois deles haviam participado da construção da base inicial. Para este grupo, foi decidido descartar as respostas. Por fim chegou-se ao total de 25 usuários com respostas válidas, que concluíram os testes respondendo o questionário de avaliação completo.

Na avaliação das recomendações, a média das notas obtidas indicam que as recomendações fizeram sentido sendo, considerando que a pergunta chave deste tema obteve média de 4.04 na escalada Likert.

Em relação ao método SUS, aplicou-se o cálculo do score para as 25 respostas obtidas, onde a pergunta P6 foi a 1ª, a P7 a 2ª e assim por diante, obteve-se um score de 84 em média, que indica boa usabilidade dentro da escala proposta pelo método, ficando apenas 1 ponto de obter uma avaliação excelente.

VII. CONCLUSÃO

Este estudo objetivou apresentar uma aplicação de recomendação de calçados esportivos para praticantes de corrida de rua baseada em uma arquitetura aberta que envolveu a integração de um algoritmo de recomendação e uma interface conversacional baseada em LLM. Neste sentido, o estudo apresentou a arquitetura desenvolvida, detalhes do desenvolvimento da aplicação, bem como, do processo de avaliação dos resultados.

A aplicação foi testada do ponto de vista das recomendações que ela gerou, bem como, da usabilidade da camada conversacional. Em ambos os casos as avaliações chegaram a resultados positivos.

Durante a evolução do estudo, observou-se oportunidades para geração de um produto comercial, explorando a possibilidade de recomendações mais adequadas e personalizadas para os usuários em um mercado aquecido. Ainda pode haver impacto social positivo, considerando a facilidade que a aplicação fomenta para a prática esportiva de corrida de rua, promovendo

a saúde e bem estar coletivo, uma vez que este tema é prioridade nos objetivos sustentáveis da ONU.

A título de trabalhos futuros, a pesquisa segue na ampliação da *dataset* de avaliações com o intuito de tornar a aplicação sistema mais robusto. Além disso, pretende-se experimentar a aplicação para recomendação de outros produtos esportivos.

REFERÊNCIAS

- [1] R. M. Hulteen, J. J. Smith, P. J. Morgan, L. M. Barnett, P. C. Hallal, K. Colyvas, and D. R. Lubans, "Global participation in sport and leisure-time physical activities: A systematic review and meta-analysis," vol. 95, pp. 14–25, 2017.
- [2] R. E. Andrade and T. R. T. Santos, "A escolha do calçado por corredores amadores: caracterização e associação com o histórico de lesão auto reportada," vol. 29, no. 4, pp. 386–396, 2022.
- [3] A. Fife, C. Ramsey, J.-F. Esculier, and K. Hébert-Losier, "How do road runners select their shoes? a systematic review," pp. 1–10, 2023.
- [4] A. P. Nguyen, L. Gillain, L. Delieux, C. Detrembleur, P. Mahaudens, and J.-F. Esculier, "Opinions about running shoes in runners and non-runners," vol. 15, no. 1, pp. 43–54, 2023.
- [5] L. Simões, F. Campos, V. Ströele, and R. Braga, "Sistema de recomendação de serviços baseado em uma arquitetura aberta para um ecossistema de software," in *Anais do XIII Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS, Brasil: SBC, 2017, pp. 340–347. [Online]. Available: <https://sol.sbc.org.br/index.php/sbsi/article/view/6061>
- [6] P. Castells and D. Jannach, "Recommender systems: A primer," 2023.
- [7] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent developments in recommender systems: A survey," 2023.
- [8] C. C. Aggarwal, *Recommender Systems*. Springer International Publishing, 2016.
- [9] P. M. Alamdari, N. J. Navimipour, M. Hosseinzadeh, A. A. Safaei, and A. Darwesh, "A systematic study on the recommender systems in the e-commerce," vol. 8, pp. 115 694–115 716, 2020.
- [10] B. Kitchenham and S. M. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Durham, UK: University of Durham, Sep. 2007.
- [11] A. Dresch, D. P. Lacerda, and Junico Antunes, *Design Science Research: Método de Pesquisa para Avanço da Ciência e Tecnologia*. Bookman, 2015.
- [12] L. Albuquerque and R. Gheyi, "Investigating IIm capabilities in the identification of compilation errors in configurable systems," in *Anais Estendidos do XV Congresso Brasileiro de Software: Teoria e Prática*. Porto Alegre, RS, Brasil: SBC, 2024, pp. 39–48. [Online]. Available: https://sol.sbc.org.br/index.php/cbsoft_estendido/article/view/30244
- [13] A. Bangor, P. T. Kortum, and J. T. M. and, "An empirical evaluation of the system usability scale," *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008. [Online]. Available: <https://doi.org/10.1080/10447310802205776>