

Booklizer - Um sistema de recomendação literária personalizada com o uso de GNN

Gabriella Silveira Braz¹, Giovana Liao¹, Maria Julia De Pádua¹, Ivan Carlos Alcantara De Oliveira¹

Faculdade de Computação e Informática (FCI)
Universidade Presbiteriana Mackenzie – São Paulo, SP – Brasil

<10402554,10402264,10400630@mackenzista.com.br>

<ivan.carlos@mackenzie.br>

2025

Resumo

O presente trabalho descreve o desenvolvimento do Booklizer, um sistema de recomendação literária personalizada baseado em Graph Neural Networks (GNN). O projeto visa propor uma solução inovadora frente às limitações de plataformas de leitura atuais, como o Goodreads, que não contemplam recomendações personalizadas de acordo com os interesses de cada leitor. A principal motivação consiste em proporcionar uma ferramenta moderna, intuitiva e centrada no usuário, capaz de aprimorar a experiência literária por meio de recomendações mais assertivas. A metodologia adotada envolveu o uso do framework Flutter para o desenvolvimento do aplicativo, Dart para o backend e FastAPI para a comunicação com o modelo de recomendação. O modelo foi estruturado sobre um grafo heterogêneo composto por nós que representam livros, gêneros e autores, sendo treinado com camadas HeteroConv e SageConv para capturar as relações e similaridades entre esses elementos. No aplicativo, o usuário interage com livros apresentados individualmente, sinalizando interesse ou desinteresse, o que aciona o modelo para gerar novas recomendações personalizadas. Os resultados obtidos indicam boas recomendações, evidenciando uma solução bastante satisfatória. O Booklizer contribui para o avanço dos sistemas de recomendação literária ao oferecer uma solução diferenciada e personalizada ao perfil de cada leitor.

Palavras-chave: sistema de recomendação; livros; inteligência artificial; GNN; Flutter.

Abstract

The present work describes the development of Booklizer, a personalized literary recommendation system based on Graph Neural Networks (GNNs). The project aims to propose an innovative solution to the limitations of current reading platforms, such as Goodreads, which do not offer recommendations tailored to each reader's interests. The primary motivation is to provide a modern, intuitive, and user-centered tool capable of enhancing the literary experience through more accurate recommendations. The

methodology adopted involved using the Flutter framework for app development, Dart for the backend and FastAPI for communication with the recommendation model. The model was structured as a heterogeneous graph composed of nodes representing books, genres, and authors, trained with HeteroConv and SAGEConv layers to capture relationships and similarities among these elements. Within the app, users interact with books presented individually, indicating interest or disinterest, which triggers the model to generate new personalized recommendations. The results obtained indicate good recommendation accuracy, demonstrating the effectiveness of the proposed solution. Booklizer contributes to the advancement of literary recommendation systems by offering an differentiated and personalized solution tailored to each reader's profile.

Keywords: recommendation system; books; artificial intelligence; GNN; Flutter.

1. Introdução

O avanço da tecnologia transformou a forma como os leitores interagem com obras literárias. Atualmente, aplicativos como o Goodreads e Skoob, são amplamente utilizados para realizar a gestão de leituras. No entanto, tais sistemas apresentam limitações significativas, como interfaces desatualizadas e a ausência de mecanismos de recomendação personalizadas e assertivas. Em consequência, muitos leitores recorrem a redes sociais, como: TikTok, YouTube, Twitter e Instagram, em busca de indicações literárias, o que colabora na fragmentação da experiência dos usuários, desde a descoberta até o acompanhamento de leitura.

Diante disso, o problema de pesquisa identificado consiste na ausência de personalização nas recomendações de leitura baseadas em preferências reais dos usuários. Essa limitação evidencia uma lacuna tecnológica e funcional nas plataformas atuais, que não são capazes de compreender os interesses da comunidade literária. O Booklizer surge como uma proposta para auxiliar nessa dor, oferecendo uma aplicação capaz de gerar recomendações mais assertivas, intuitivas e dinâmicas, além de centralizar diversas funcionalidades, permitindo a pesquisa de livros e a gestão de atividades de leitura. Para tal, o desenvolvimento do projeto fundamenta-se em conceitos de sistemas de recomendação, desenvolvimento de software, análise de dados, grafos e redes neurais em grafos, além do uso de tecnologias modernas como Flutter, Dart, FastAPI e MySQL. Esses conceitos possibilitaram a construção de um modelo de recomendação baseado em um grafo heterogêneo, que é capaz de explorar as relações entre livros, gêneros e autores, proporcionando uma experiência literária mais personalizada, intuitiva, centralizada e assertiva.

A motivação para o projeto está relacionada à necessidade de oferecer uma ferramenta que centralize funcionalidades de recomendação personalizada e gestão de leitura. A proposta do Booklizer é aprimorar a experiência de descoberta de livros e fomentar o interesse pela leitura, promovendo um ambiente digital mais interativo e integrado aos hábitos dos leitores contemporâneos.

O objetivo central consiste em desenvolver um sistema de recomendação literária personalizada com o uso de *Graph Neural Networks* (GNN), tornando a experiência de leitura mais assertiva e intuitiva para os usuários. Para alcançar esse propósito, foram definidas a arquitetura do sistema e a obtenção dos dados necessários, inicialmente por meio das APIs,

interfaces de programação que permitem a comunicação e a troca de informações entre diferentes sistemas, da OpenLibrary e Google Books, que apresentaram inconsistências nos gêneros, sendo posteriormente adotada uma base de dados do Kaggle, plataforma online que oferece conjunto de dados públicos, contendo informações de 10 mil livros, incluindo título, autores, ano de publicação e lista de gêneros para cada obra, e a criação de um grafo conceitual representando as relações entre livros, autores e gêneros. Além disso, houve a implementação do modelo de GNN para explorar essas relações e gerar recomendações personalizadas e a avaliação da acurácia das recomendações.

A relevância da pesquisa está em sua contribuição para o avanço dos sistemas de recomendação aplicados ao contexto literário, promovendo a personalização da experiência de leitura e incentivando o hábito da leitura por meio da tecnologia.

O documento está organizado em seções que apresentam, de forma sequencial, o desenvolvimento da pesquisa. Nesta seção, é apresentada a introdução que contextualiza o tema, define o problema de pesquisa, os objetivos e a justificativa do estudo. Em seguida, na Seção 2, é exposto o referencial teórico, abordando os principais conceitos relacionados a sistemas de recomendação e redes neurais em grafos. Na sequência, descreve-se a metodologia utilizada para o desenvolvimento do projeto, detalhando as etapas técnicas e as ferramentas empregadas na Seção 3. Posteriormente, na Seção 4, são apresentados os resultados obtidos e a discussão sobre a eficácia do modelo proposto. Por fim, na Seção 5, são expostas as conclusões, juntamente com as considerações finais e sugestões para trabalhos futuros.

2. Referencial Teórico

Nesta seção são apresentados os conceitos relacionados à comunidades literária, sistemas de recomendação literária personalizados, grafos, GNNs, que servem de base para o desenvolvimento deste trabalho.

2.1 Comunidade literária e aplicativos modernos

O surgimento de plataformas digitais expandiu as formas de interação entre leitores, autores e obras, promovendo o desenvolvimento de comunidades literárias digitais. Esses espaços transformaram a leitura em uma atividade compartilhada, mediada por redes sociais e práticas colaborativas que aumentam o engajamento e a troca de experiências entre os usuários. Entre essas comunidades, o BookTok, presente na rede social TikTok, destaca-se como um dos movimentos mais influentes na leitura moderna. Segundo Jerasy e Boffone (2021), o BookTok representa uma nova forma de letramento digital em que os leitores utilizam vídeos curtos para recomendar livros, expressar opiniões e construir vínculos emocionais em torno da leitura. Esse formato, caracterizado pela espontaneidade e alto potencial de engajamento, aumenta o interesse por obras literárias e transforma as formas de circulação cultural no ambiente digital.

Complementando essa perspectiva, Maddox e Gill (2023) analisam o BookTok como uma ecologia cultural na qual o design da interface e o funcionamento do algoritmo do TikTok moldam o surgimento de subcomunidades, as chamadas páginas do TikTok. Essa estrutura permite que leitores com interesses semelhantes se conectem, formando grupos específicos com base em gêneros literários, autores ou estilos narrativos. Assim, o BookTok se configura como um ambiente híbrido no qual interações interpessoais e processos automatizados se combinam para definir tendências e fortalecer laços sociais em torno da leitura.

Além do TikTok, outras plataformas também desempenham um papel fundamental na construção dessas comunidades. Por exemplo, o Goodreads se destaca como uma rede social para leitores, aproveitando recursos compartilhados para avaliar, classificar e recomendar livros. De acordo com Amaral e Salvador (2018), a dinâmica de interação no Goodreads é aprimorada pelo uso da folksonomia, a marcação coletiva de livros utilizando tags criadas pelos próprios usuários. Esse processo de categorização contribui para a formação de conhecimento social compartilhado, tornando o Goodreads não apenas um espaço para expressar opiniões, mas também uma ferramenta para organizar colaborativamente informações literárias.

Dessa forma, as comunidades literárias digitais desempenham um papel importante na socialização da leitura, fomentando o engajamento, o senso de pertencimento e novas maneiras de descobrir obras. Essa dinâmica serve como ponto central para o Booklizer, que busca integrar os princípios de personalização e interação social presentes em plataformas como TikTok e Goodreads. Compreender essas comunidades é fundamental para o desenvolvimento de sistemas que conectem a tecnologia à experiência coletiva, aproximando os leitores dos livros de uma forma mais inteligente e integrada.

2.2 Sistemas de recomendação

Os sistemas de recomendação surgiram como resposta ao problema da sobrecarga de informações, com o objetivo de auxiliar os usuários na descoberta de itens de interesse em meio a grandes volumes de dados. Segundo Adomavicius e Tuzhilin (2005), esses sistemas têm como função principal filtrar informações relevantes e oferecer sugestões personalizadas com base nas preferências e comportamentos dos usuários.

De acordo com Batmaz et al. (2019), o funcionamento de um sistema de recomendação é fundamentado na relação entre dois conjuntos principais: usuários e itens. A interação entre esses elementos é representada por uma matriz que contém avaliações explícitas, como notas, ou implícitas, como histórico de cliques, tempo de leitura, entre outros. O desafio do sistema é prever as preferências desconhecidas e gerar recomendações adequadas, mesmo com dados incompletos. Nesse contexto, a eficiência do sistema depende diretamente da quantidade e da qualidade das interações registradas entre usuários e itens, sendo essencial lidar com limitações estruturais da matriz de avaliações. A literatura recente aponta que, embora amplamente utilizados, os sistemas de recomendação ainda enfrentam desafios como o *cold start*, ou seja, ausência de dados iniciais sobre novos usuários ou itens, a esparsidade da matriz de avaliações, além de questões relacionadas à diversidade e equidade das sugestões geradas (KOUKI et al., 2023). Tais aspectos são fundamentais para garantir recomendações mais justas, transparentes e relevantes ao perfil de cada usuário.

De acordo com Adomavicius e Tuzhilin (2005), os sistemas de recomendação evoluíram significativamente ao longo dos anos, consolidando-se em três principais abordagens: filtragem baseada em conteúdo, filtragem colaborativa e métodos híbridos. A filtragem baseada em conteúdo (*Content-Based Filtering*) consiste em recomendar ao usuário itens semelhantes àqueles que ele demonstrou preferência no passado. Já a filtragem colaborativa (*Collaborative Filtering*) baseia-se no comportamento coletivo dos usuários, identificando padrões de similaridade entre perfis e sugerindo itens que pessoas com gostos semelhantes apreciaram anteriormente. Essa abordagem parte do princípio de que indivíduos com históricos de consumo parecidos tendem a gostar de itens semelhantes no futuro. Por sua vez, os métodos híbridos (*Hybrid Approaches*) surgem como uma solução para mitigar as limitações de cada técnica isoladamente, como o problema do *cold start*, a

esparsidade de dados ou a falta de diversidade nas recomendações, combinando elementos das duas abordagens anteriores para produzir resultados mais robustos.

No contexto do Booklizer, o sistema de recomendação foi desenvolvido para sugerir livros personalizados com base nas preferências literárias do usuário, considerando suas escolhas de gêneros, autores e estilos de leitura. Essa abordagem visa proporcionar uma experiência interativa e adaptativa, reforçando o papel dos sistemas de recomendação como instrumentos de apoio à descoberta de novos conteúdos e à valorização da experiência do leitor.

2.3 Grafos

Um grafo é uma estrutura matemática utilizada para representar relações entre elementos de um conjunto. Segundo Costa (2011), um grafo finito G é formado por um par $(V(G), A(G))$, em que $V(G)$ representa um conjunto finito e não vazio de nós, e $A(G)$ corresponde a uma família de pares não ordenados de elementos de $V(G)$. De acordo com Pavlopoulos (2011), essas redes possibilitam a visualização e o processamento de relações complexas, tornando-se essenciais em aplicações que envolvem conexões múltiplas entre entidades, como: redes sociais, sistemas de transporte, biologia computacional e sistemas de recomendação.

Ainda de acordo com Costa (2011), quando as arestas conectam pares distintos de vértices e não há repetição de conexões ou laços, o grafo é denominado grafo simples. Esse tipo de estrutura é amplamente empregado em modelagens computacionais por sua clareza e eficiência na representação de dados relacionais, como é o caso do grafo utilizado no Booklizer.

Além dos grafos simples, é possível classificar grafos quanto à homogeneidade de seus nós. Grafos homogêneos são aqueles onde seus vértices representam elementos de um mesmo tipo, enquanto grafos heterogêneos incluem diferentes tipos de entidades, permitindo que nós de categorias distintas se conectam por meio de arestas. Chairatanakul (2025) afirma que os grafos heterogêneos podem representar os dados de aplicações do mundo real de forma mais fiel e precisa do que os grafos homogêneos. Por conta disso, no desenvolvimento do Booklizer, optou-se pelo uso de um grafo heterogêneo, pois ele possibilitou representar de forma mais completa as relações entre os livros, autores e seus gêneros.

2.4 Redes neurais em grafos (GNN)

2.4.1 Conceitos e princípios

A partir da estrutura de grafos surgem as Redes Neurais em Grafos, modelos de aprendizado profundo desenvolvidos para lidar com dados organizados em grafo. De acordo com WU et al. (2021), enquanto redes neurais tradicionais operam sobre dados em estruturas fixas, como vetores ou matrizes, as GNNs são capazes de processar e aprender representações a partir das relações entre os nós de um grafo.

Segundo Khemani et al. (2024), as GNNs possuem a capacidade de capturar padrões complexos contidos nas conexões entre entidades, tornando-se adequadas para quando os dados são interligados e não euclidianos, ou seja, dados que não podem ser representados em um plano cartesiano. O princípio central dessas redes é o mecanismo de passagem de mensagens, no qual cada nó troca informações com seus vizinhos, agregando-as iterativamente para atualizar sua própria representação, como pode ser observado na Figura 1. Esse processo permite que o modelo

aprenda não apenas as características individuais de cada nó, mas também o contexto relacional em que ele está inserido.

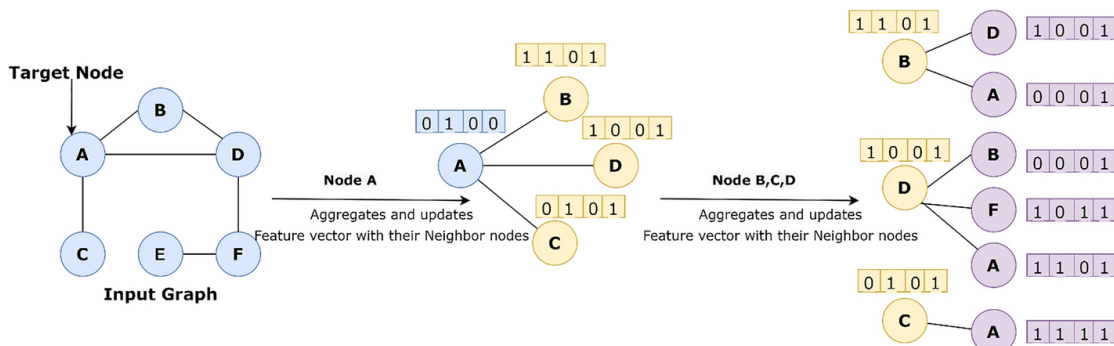


Figura 1 - Esquema do fluxo de mensagens entre nós (message passing).

Fonte: KHEMANI et al. (2024)

2.4.2 Camadas da GNN

Para o desenvolvimento da GNN, é necessário definir os tipos de camadas que compõem o modelo, pois elas determinam como as informações serão agregadas e propagadas entre os nós. No Booklizer, foram utilizadas as camadas HeteroConv e SAGEConv, que permitem relacionar informações entre entidades de naturezas distintas, capturando padrões de similaridade entre obras e as preferências dos usuários. A camada HeteroConv é projetada especificamente para grafos com múltiplos tipos de nós e arestas, permitindo que a rede agregue informações de diferentes relações sem confundir entidades distintas. Já a SAGEConv realiza uma agregação amostral que aprende representações de nós combinando suas próprias características com as de seus vizinhos, utilizando um agregador baseado na média convolucional, que atua como uma aproximação linear de convoluções (YANG et al., 2020). A integração dessas camadas permite que a GNN capture padrões, tanto específicos de cada tipo de relação quanto estruturais do grafo como um todo, preparando os dados para a geração de representações vetoriais de cada nó. Esse agrupamento de informações pode ser visto na Figura 2 a seguir.

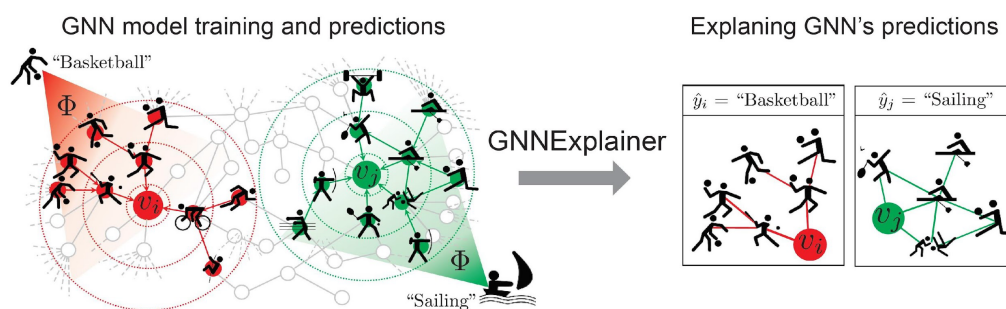


Figura 2 - Esquema do fluxo de agregação de informações na GNN

Fonte: YING et al. (2019)

2.4.3 Embeddings

Essas representações vetoriais, conhecidas como *embeddings*, são vetores numéricos que codificam as propriedades e o contexto relacional de cada nó no grafo, permitindo que o modelo

capture relações em um espaço. De acordo com Deng (2024), essa técnica tem como objetivo projetar estruturas não euclidianas em um espaço vetorial, preservando as características semânticas das relações originais. No Booklizer, os *embeddings* possibilitam que a GNN compreenda as conexões não explicitamente visíveis entre os elementos da rede, o que amplia a capacidade de aprendizado do modelo ao explorar relações entre diferentes entidades.

2.4.4 Otimizador

Para o treinamento do modelo, foi utilizado o otimizador Adam (*Adaptive Moment Estimation*), proposto por Kingma e Ba (2015). Esse método é amplamente empregado em modelos de aprendizado profundo por oferecer um equilíbrio entre eficiência computacional, estabilidade e rapidez do treinamento. Ele é responsável por ajustar automaticamente a intensidade com que a rede aprende, controlando o tamanho dos passos que o modelo dá em direção à melhor solução durante o processo de aprendizado.

De acordo com Kingma e Ba (2015), esse otimizador combina o melhor de dois algoritmos anteriores: o AdaGrad, que se adapta bem quando há muitos parâmetros com pouca variação e o RMSProp, que lida melhor com dados que mudam ao longo do tempo. Essa combinação torna o Adam altamente adaptável, o que é especialmente importante em modelos complexos como as Redes Neurais em Grafos (GNNs), onde há muitas interdependências entre os nós e as relações evoluem dinamicamente durante o treinamento.

2.4.5 Função de perda

A função de perda *Triplet Margin Loss* é uma abordagem utilizada para aprendizado de representações em espaços vetoriais. De acordo com Abdelrahman et al. (2021), as *triplet networks* foram desenvolvidas para ensinar modelos a compreender relações de similaridade e distinção entre amostras, mapeando distâncias no espaço de características de modo que itens semelhantes permaneçam próximos e itens diferentes sejam separados no espaço. Essa função permite um aprendizado mais refinado de relações de similaridade, aplicável a tarefas que envolvem julgamentos subjetivos, como classificação de imagens, reconhecimento facial e ranqueamento de similaridade visual.

Nesse tipo de arquitetura, são utilizados três exemplos por iteração: um nó âncora, um nó positivo, semelhante ao âncora, e um nó negativo, diferente do âncora. O objetivo do treinamento é reduzir a distância entre o âncora e o positivo, enquanto aumenta a distância entre o âncora e o negativo. É possível observar essa dinâmica a partir da Figura 3, onde o nó negativo acaba se afastando do nó âncora após o processo de aprendizado.

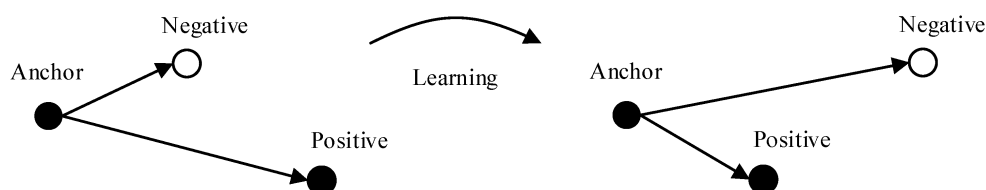


Figura 3 – Ilustração do funcionamento da função de perda Triplet Margin Loss
Fonte: YUAN et al. (2019)

No contexto do *Booklizer*, essa função é empregada de modo similar: cada *triplet* é formado por um livro âncora, um livro positivo (com similaridade de gênero, autor) e um livro negativo (sem relação direta). A GNN aprende a projetar esses elementos em um espaço vetorial, onde obras semelhantes se localizam próximas e obras distintas se afastam. Essa organização do espaço permite que o sistema identifique afinidades literárias, aprimorando a personalização das recomendações.

2.4.6 Métricas de avaliação

As métricas de avaliação são fundamentais para mensurar o desempenho e a precisão de modelos de recomendação. Entre as mais utilizadas estão Precision@K , Recall@K e NDCG@K , que avaliam a qualidade das recomendações geradas a partir de diferentes perspectivas. A Precision@K mede a proporção de itens realmente relevantes entre os K principais recomendados, enquanto a Recall@K avalia a proporção de itens relevantes que foram efetivamente recomendados pelo modelo, em relação ao total de itens relevantes existentes. Já a NDCG@K analisa a qualidade do ranqueamento, indicando o quão próxima está a ordem das recomendações da lista ideal de preferência do usuário, penalizando erros nas primeiras posições da lista. Essas métricas são amplamente aplicadas em sistemas de recomendação baseados em redes neurais gráficas, como descrito por He et al. (2023), no estudo “Graph neural network recommendation algorithm based on improved dual tower model”, em que os autores utilizam tais indicadores para avaliar a precisão e relevância das recomendações geradas por um modelo de recomendação com uma arquitetura semelhante.

2.5 Trabalhos relacionados

Um trabalho relacionado é o de Souza e Souza (2021), que investiga uma estratégia moderna de recomendação apoiada em aprendizado profundo, propondo uma arquitetura híbrida de Redes Neurais Profundas (DNN) capaz de integrar informações colaborativas e de conteúdo. Diferentemente dessa proposta, o presente projeto adota uma abordagem baseada em Redes Neurais Gráficas (GNN), focando na representação semântica das conexões entre livros e gêneros literários. Para isso, é construído um grafo bipartido, e o modelo é treinado com a função de perda *Triplet Margin Loss*, resultando em vetores de representação mais refinados que capturam nuances de similaridade entre obras conectadas por até dois nós vizinhos. Embora ambos os trabalhos busquem a personalização de recomendações, o uso de GNNs permite explorar a topologia do grafo, favorecendo a propagação de informações semânticas entre os nós e resultando em uma abordagem mais estruturada e flexível para o tratamento desses dados.

3. Metodologia

A metodologia adotada neste trabalho caracteriza-se como uma pesquisa de caráter exploratório e descritivo. Trata-se de uma abordagem voltada ao desenvolvimento de uma solução tecnológica capaz de personalizar a experiência de leitura por meio de sistemas de recomendação baseados em grafos e redes neurais. O desenvolvimento desta solução foi estruturado em etapas, que conduziram o projeto desde o planejamento conceitual e levantamento de requisitos até a implementação do sistema.

3.1. Revisão da Literatura

Com a finalidade de identificar textos norteadores e aderentes ao proposto neste trabalho, ao longo de sua execução, foi realizada uma revisão da literatura, em que foram selecionadas 18 referências, como artigos, dissertações e teses, pelo título e resumo aderentes ao conteúdo desta proposta, fazendo uso da base ‘*Best Books (10k) Multi-Genre Data*’ (KAGGLE, 2021). Todas essas referências foram lidas na íntegra e algumas delas citadas neste artigo. As *strings* utilizadas nas buscas foram: sistema de recomendação; recomendações de livros, comunidades literárias, graph neural network, triplet loss, grafos heterogêneos e métricas de modelos de dados.

3.2 Levantamento de requisitos e prototipação da interface

O levantamento de requisitos funcionais e não funcionais teve como objetivo mapear de forma clara as necessidades do sistema e de seus usuários. Essa fase foi conduzida a partir de uma análise de mercado, incluindo uma revisão crítica de aplicativos amplamente utilizados, como o Goodreads e o Skoob. Por meio dessa análise, foram identificadas lacunas e oportunidades de inovação que o Booklizer poderia explorar. Como resultado disso, obteve-se como requisitos não funcionais, a interface responsiva e intuitiva; segurança e privacidade dos dados; boa performance; e compatibilidade com diferentes dispositivos móveis. Já os requisitos funcionais propostos foram: cadastro de usuários com informações obrigatórias e opcionais; recomendação de livros com base em interesses e avaliações; organização de títulos em estantes virtuais (“Tenho interesse” e “Já lido”); visualização gráfica do histórico de leitura; e uma ferramenta de busca por livros.

Com base nesses requisitos, foi realizada a prototipação da interface na ferramenta Figma, com o objetivo de definir a arquitetura visual e o fluxo de navegação do sistema. Foram elaborados protótipos de baixa e alta fidelidade, representando a disposição dos elementos gráficos, cores, ícones e componentes interativos. Essa etapa permitiu avaliar a usabilidade e antecipar ajustes de design antes da implementação, reduzindo retrabalhos e garantindo uma experiência de usuário mais fluida e intuitiva. Os protótipos também serviram como referência visual para o desenvolvimento posterior em Flutter, assegurando coerência entre o design e a funcionalidade do sistema.

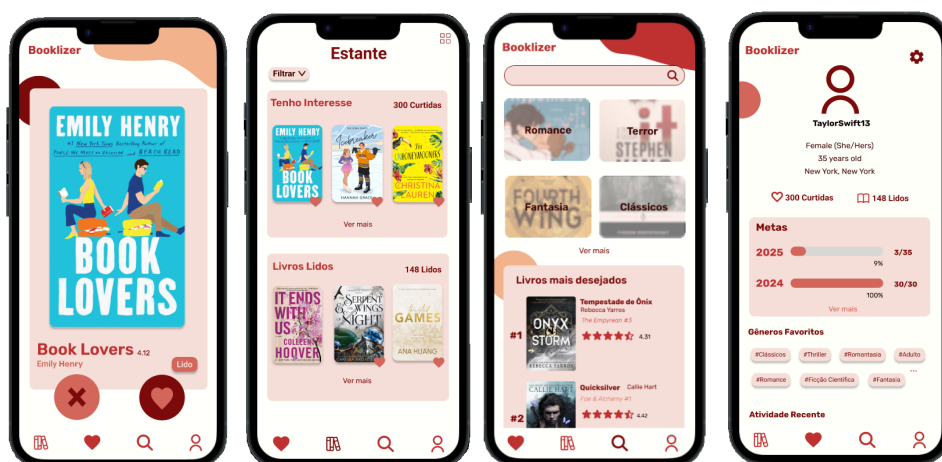


Figura 4 - Protótipos de alta fidelidade para as telas: Recomendação, Estante, Busca e Perfil
Fonte: Autoria própria.

3.3 Obtenção de Dados

Inicialmente, foram consideradas a base do OpenLibrary, a API do Google Books e a API do ISBN, porém todas apresentavam inconsistências na classificação de gêneros, ausência de informações essenciais e dificuldades de uniformização, tornando-as inadequadas para o modelo proposto. Diante disso, optou-se pela base “Best Books (10k) Multi-Genre Data”, disponível no Kaggle (2021), contendo informações de livros recomendados pelo Goodreads.

Cada registro da base inclui título do livro, autor, descrição, lista de gêneros, média de avaliações, número de avaliações e link para a página do Goodreads. Para preparar os dados para o sistema de recomendação, foram aplicados filtros para remover duplicatas de livros, autores anônimos e entradas sem gêneros. Além disso, foram mantidos apenas livros com pelo menos um gênero válido. Após esses procedimentos, a base resultou em 8.915 livros processados.

A exploração inicial dos dados foi realizada utilizando *scripts* em Python. A base total conta com 616 gêneros distintos, destacando-se Fiction (5.559 livros), Nonfiction (2.266), Fantasy (2.219) e Classics (2.089) como os mais recorrentes. Foram produzidas visualizações gráficas para compreender a frequência dos gêneros, identificar autores mais produtivos e analisar a predominância de gêneros por autor. É possível visualizar a partir da Figura 5, o gráfico que ilustra a distribuição dos principais gêneros na base, o qual fornece informações importantes para orientar a modelagem e as decisões de recomendação do sistema.

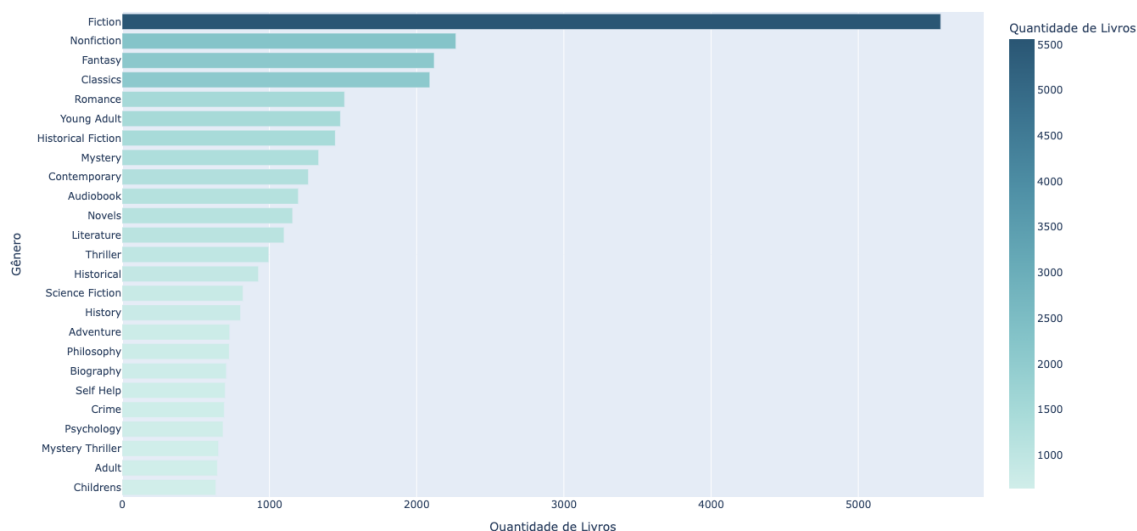


Figura 5 – Top 25 gêneros das obras
Fonte: Autoria própria.

3.4 Modelagem em grafos

Para estruturar os dados para a GNN, foi construído um grafo heterogêneo representando os relacionamentos entre livros, gêneros e autores. Inicialmente, foi desenvolvido um modelo contendo apenas nós de livros e gêneros, em que as conexões refletiam relações bidirecionais entre essas entidades, permitindo que o modelo aprendesse representações vetoriais dos livros a partir das suas associações com os gêneros. Posteriormente, o grafo foi ampliado para incluir nós de autores, adicionando relações entre livros e seus respectivos autores, enriquecendo a representação

semântica e permitindo capturar afinidades estilísticas e temáticas que não estavam presentes no modelo inicial. Essa modelagem garantiu que cada nó pudesse receber informações de seus vizinhos, permitindo que os *embeddings* aprendidos pela GNN refletissem tanto as associações de gênero quanto de seus autores, tornando as recomendações mais precisas para os usuários.

No desenvolvimento foi utilizado o PyTorch Geometric, que fornece o construtor HeteroData para processar grafos heterogêneos, e Pandas, utilizado para pré-processar os dados, mapear livros, autores e gêneros para índices numéricos e gerar as listas de conexões que formam as arestas do grafo. Para o contexto de testes e modelagem, utilizamos apenas 1000 dados obtidos a partir da tabela original, o qual resultou em 993 obras finais após o processo de limpeza de dados. O grafo foi criado a partir desses livros válidos, mantendo informações de autores e gêneros. Cada nó representava uma entidade do tipo livro (993 nós), autor (671 nós) ou gênero (286 nós), e as arestas refletiam relações entre esses nós: livro-autor e livro-gênero, permitindo agregação bidirecional de informações durante o treinamento da GNN.

Para validar a topologia do grafo e compreender suas conexões, foram realizadas visualizações exploratórias. Na Figura 6, é possível observar o livro “1984” (nó azul claro), seu autor “George Orwell” (nó verde claro) e todos os gêneros associados à obra (nós vermelhos).

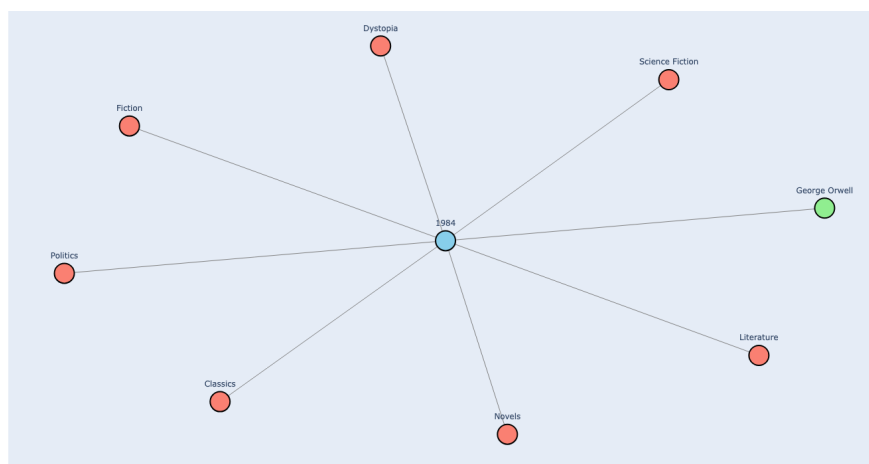


Figura 6 – Conexões do livro ‘1984’
Fonte: Autoria própria.

Os nós do grafo foram inicialmente representados por vetores, nos quais cada posição indica a presença de um nó específico, sem atributos adicionais, permitindo que a GNN aprenda representações vetoriais (*embeddings*) exclusivamente a partir das relações estruturais entre os nós. Essa escolha de representação garante que o modelo foque na topologia do grafo e nas conexões entre entidades, capturando padrões de similaridade entre livros, autores e gêneros.

O grafo final apresentou 1.950 nós e 15.656 arestas, sem nós isolados, garantindo que todos os elementos estivessem conectados. As estatísticas de grau indicaram que livros tinham entre 4 e 16 conexões (média de 15,77), autores entre 2 e 28 (média de 2,96) e gêneros entre 2 e 1.570 (média de 47,80), refletindo a diversidade de relacionamentos na rede. Essa estrutura heterogênea forneceu a base para o treinamento da GNN, permitindo que o modelo aprendesse

representações vetoriais dos livros, considerando tanto suas características individuais quanto o contexto com autores e gêneros.

3.5 Graph Neural Networks (GNN)

3.5.1 Construção do modelo

Para o desenvolvimento do modelo de recomendação, foi utilizada uma Graph Neural Network (GNN) aplicada a um grafo heterogêneo. Essa abordagem possibilitou representar interações entre diferentes entidades do domínio literário, como livros, gêneros e autores, em uma estrutura relacional, permitindo que o modelo aprendesse padrões de similaridade a partir das conexões entre nós.

Inicialmente, foi implementado um modelo composto apenas por nós de livros e gêneros. A arquitetura utilizava duas camadas GCNConv, responsáveis pela propagação de informações entre os nós, com dimensões intermediárias e de saída configuradas em 24 e 16 neurônios, respectivamente. As features de entrada eram representações one-hot de identidade dos nós, e o treinamento foi conduzido utilizando a função de perda *Triplet Margin Loss*. Esse modelo inicial foi treinado por 60 épocas, apresentando uma redução consistente da perda, de 0,7370 na primeira época para 0,2209 na última, o que indica que a rede foi capaz de aprender representações semânticas entre livros e gêneros. O tempo total de treinamento foi de aproximadamente 8 minutos, e as recomendações geradas mostraram coerência temática.

Com base nesses resultados, identificou-se a necessidade de incorporar novos tipos de entidades e relações ao grafo, com o objetivo de enriquecer a representação semântica e tornar as recomendações mais precisas. Assim, foi desenvolvido um modelo aprimorado, que passou a incluir nós de autores. Essa versão empregou uma arquitetura composta por camadas HeteroConv combinadas com SAGEConv, permitindo a agregação de informações entre diferentes tipos de nós. O modelo utilizou dimensão intermediária de 24 e saída de 64, aumentando a capacidade de representar relações complexas entre livros, gêneros e autores, o que resultou em *embeddings* mais expressivos e recomendações mais precisas.

3.5.2 Treinamento do modelo

Como citado anteriormente, ambos os modelos utilizaram *Triplet Margin Loss*, na qual cada iteração empregava três exemplos: um nó âncora (livro), um nó positivo (livro do mesmo autor ou compartilhando algum gênero) e um nó negativo (livro sem relação direta com o nó âncora). Este método permitiu que o modelo aprendesse a projetar livros similares próximos no espaço vetorial e livros distintos mais afastados, aprimorando a capacidade de recomendação personalizada.

Durante o treinamento, a atualização dos *embeddings* de cada nó foi realizada de forma iterativa utilizando o otimizador Adam, que ajusta os pesos da rede de maneira adaptativa, controlando a taxa de aprendizado individual para cada parâmetro. Essa abordagem favorece uma convergência mais rápida e estável, além de lidar melhor com a esparsidade e heterogeneidade do grafo. Após o treinamento, os *embeddings* gerados foram utilizados para calcular similaridades entre livros e gerar recomendações, permitindo identificar obras com afinidades literárias mesmo sem conexões explícitas no grafo. Esse processo garantiu que o sistema fosse capaz de sugerir

livros com base em relações latentes de gênero e autoria, promovendo recomendações mais precisas e personalizadas para os usuários.

3.6 Desenvolvimento da Aplicação

O desenvolvimento do Booklizer foi estruturado segundo a arquitetura cliente-servidor, composta por uma aplicação desenvolvida em Flutter e Dart (*frontend*) e uma API RESTful implementada em Python, utilizando o *framework* FastAPI (*backend*). Essa abordagem permitiu a separação entre as camadas, favorecendo a escalabilidade e a manutenção do sistema. Além disso, o aplicativo também segue o padrão de design conforme os protótipos definidos no Figma, garantindo uma interface responsiva e consistente.

A API, é responsável por realizar a comunicação entre o aplicativo e o modelo de recomendação, como pode ser visto na Figura 7. Sua principal função consiste em consultar o modelo de recomendação, processar as solicitações enviadas pelo *front-end* e retornar as respostas no formato JSON, usado para armazenar e transmitir informações entre sistemas de forma estruturada, contendo não apenas os livros recomendados, mas também informações completas como título, autor, descrição e gêneros.

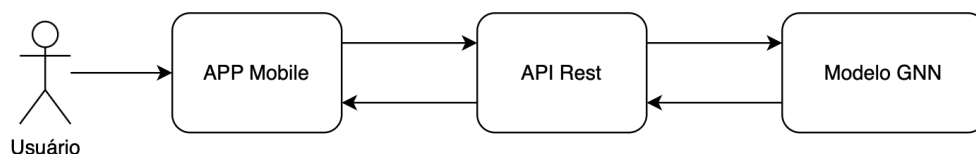


Figura 7 – Arquitetura da aplicação
Fonte: Autoria própria.

O modelo de recomendação é executado no *backend* em python, porém as recomendações foram pré-computadas e armazenadas em um arquivo no formato .pkl, integrado ao aplicativo, de modo a otimizar o desempenho e reduzir o tempo de resposta durante a execução. Essa estratégia foi adotada para viabilizar o funcionamento do sistema mesmo de forma rápida, preservando a experiência do usuário, uma vez que a compilação do modelo parcial, demora aproximadamente 10 minutos.

A comunicação entre o aplicativo e a API ocorre por meio de requisições HTTP assíncronas no formato JSON, o que garante fácil integração entre as camadas. No Flutter, o consumo da API é realizado por meio da biblioteca http, que permite o envio e o recebimento eficientes de dados, assegurando a responsividade e a confiabilidade nas interações cliente-servidor.

3.7 Desenvolvimento da API de recomendação

Após o treinamento do modelo GNN, foi necessário estabelecer a conexão entre o *front-end* e os dados de livros e recomendação. Para isso, foi desenvolvida uma API para disponibilizar o consumo dessas informações pela aplicação. A API possui dois *endpoints*, ou seja, pontos de acesso que recebem e respondem a requisições do cliente, correspondentes às etapas de filtragem inicial e geração de recomendações personalizadas.

a) /livros/filtrar – Filtragem e listagem de obras

Esse *endpoint* fornece ao aplicativo o conjunto inicial de livros de acordo com os gêneros selecionados pelo usuário no primeiro acesso. O corpo da requisição inclui parâmetros como a lista de gêneros, o autor, o modo de correspondência e um parâmetro que realiza a busca de um termo, utilizado para filtrar livros que contenham o título informado, permitindo consultas diretas por nome da obra. A resposta é retornada em JSON, contendo o número total de obras, as páginas disponíveis e uma lista de livros com atributos normalizados, como título, autor, descrição, gêneros, avaliação e ano de publicação.

Durante o processamento, são aplicadas rotinas de limpeza e padronização de dados, removendo duplicatas, convertendo listas de gêneros para o formato correto e substituindo valores ausentes, garantindo consistência entre os registros. Essa normalização assegura que os dados retornados estejam adequados para o modelo de recomendação e prontos para exibição no aplicativo.

b) /recomendar – Geração de recomendações personalizadas

Esse *endpoint* retorna recomendações literárias com base em um livro âncora informado pelo usuário. O corpo da requisição contém o nome do livro e a quantidade de recomendações desejadas. Internamente, a API utiliza a matriz de similaridade gerada a partir dos *embeddings* produzidos pela GNN, identificando as obras mais próximas do livro âncora no espaço vetorial. As obras com maior similaridade são ordenadas e retornadas com metadados completos, incluindo título, autor, descrição, gêneros e pontuação de similaridade convertida em uma escala de 0 a 1.

3.8 Fluxo lógico do funcionamento da recomendação

O fluxo lógico de funcionamento do modelo foi estruturado de forma a integrar as etapas de interação do usuário com o sistema e o processamento das recomendações realizadas pela aplicação. No primeiro acesso, o usuário é recebido por uma tela de *onboarding* contendo dez gêneros literários em destaque. Ele pode selecionar os gêneros de interesse e, caso deseje, visualizar mais opções por meio do botão “Mais gêneros”. Ao confirmar a seleção, o aplicativo envia essas preferências ao *endpoint* /livros/filtrar, que retorna uma lista inicial de livros compatíveis com os gêneros escolhidos. As preferências são armazenadas localmente, garantindo que as escolhas do usuário sejam preservadas durante a sessão e utilizadas em requisições subsequentes.

Em seguida, os livros são exibidos em um carrossel interativo (*card swiper*), apresentando capa, título, autor, sinopse e gêneros associados. Quando uma obra não possui capa registrada no banco de dados, o aplicativo realiza uma requisição adicional à Bookcover API (LONGITOOD, 2024). A partir desse ponto, quando o usuário demonstra interesse por uma obra ao deslizar o *card* para a direita, o livro selecionado é definido como livro âncora. O aplicativo então envia uma nova requisição POST ao *endpoint* /recomendar, que realiza uma consulta ao modelo de recomendação GNN. Esse modelo pré-computado e armazenado em um arquivo de extensão PKL (pickle), permite que as recomendações sejam acessadas de forma rápida e eficiente, sem necessidade de processamento em tempo real. A resposta do *endpoint* contém uma lista de obras semelhantes, que substituem imediatamente as recomendações anteriores, atualizando o carrossel.

Como as recomendações são estáticas, o sistema não realiza aprendizado incremental a partir das novas interações do usuário, mas sim consultas pontuais baseadas no livro âncora

selecionado. Esse fluxo assegura uma experiência de uso contínua e responsiva, combinando leveza no processamento com resultados coerentes ao perfil literário definido durante o *onboarding*.

4. Resultados e Discussão

Nesta seção, são apresentados os resultados obtidos da análise exploratória, modelos e suas métricas, bem como a aplicação e experimentos.

4.1 Desempenho do modelo de recomendação

A análise exploratória do grafo heterogêneo empregado neste estudo revelou uma rede composta por 1.950 nós, distribuídos entre livros, autores e gêneros, sendo 993 livros, 671 autores e 286 gêneros. As conexões entre esses nós totalizaram 15.656 arestas, evidenciando as relações estabelecidas entre livros, gêneros e autores. Não foram identificados nós isolados, o que indica que todas as entidades possuíam, no mínimo, uma conexão registrada.

A comparação entre os modelos desenvolvidos possibilitou avaliar a evolução do desempenho do sistema de recomendação a partir do aprimoramento da estrutura do grafo. O modelo inicial, composto exclusivamente por nós de livros e gêneros, apresentou métricas satisfatórias para uma etapa preliminar, embora tenha se mostrado limitado pela ausência de relações autorais. Por sua vez, o modelo aprimorado, que incorporou também os nós de autores, demonstrou desempenho superior, ao ampliar a conectividade entre as entidades e capturar padrões semânticos mais complexos, resultando em recomendações mais precisas e contextualizadas. A Tabela 1 apresenta o comparativo das principais métricas obtidas em ambos os modelos.

Métrica	Modelo Inicial (Livros + Gêneros)	Modelo Atual (Livros + Gêneros + Autores)
Épocas de Treinamento	60	30
Função de Perda Inicial	0,7370	0,9971
Função de Perda Final	0,2209	0,2522
Precision@10	0,6795	0,994
Recall@10	1,0636	0,016
HitRate@10	1,0000	1,000
MAP@10	0,8655	0,993
NDCG@10	0,9310	0,995

Tabela 1 - Comparativos entre as métricas obtidas a partir dos modelos
Fonte: Autoria própria.

Observa-se que, embora o recall tenha diminuído devido à maior especificidade das recomendações, o modelo atual apresentou ganhos expressivos em precisão, MAP e NDCG, confirmando que a inclusão das relações de autores enriqueceu as representações vetoriais aprendidas pela GNN.

Em suma, a análise dos resultados indica que a abordagem baseada em grafos heterogêneos é eficaz para capturar relações e produzir recomendações. O alto grau de certos gêneros contribui para recomendações centralizadas em categorias populares, enquanto autores com poucas obras apresentam menor representatividade. O modelo demonstrou boa capacidade de fornecer sugestões consistentes, embora a baixa cobertura de itens menos representados aponte para uma oportunidade de melhoria. Ajustes na estratégia de amostragem de vizinhos, na incorporação de informações adicionais de usuários ou no balanceamento do grafo poderiam aumentar o *recall* e a diversidade das recomendações.

4.2 Aplicação e suas interfaces

A implementação das interfaces foi realizada utilizando o *framework* Flutter, enfrentando-se uma curva de aprendizado significativa relacionada à linguagem Dart e à criação de interfaces responsivas. O desenvolvimento priorizou a usabilidade, a organização das informações e a coerência visual, resultando em um conjunto de telas funcionais que refletem a identidade do *Booklizer*.

A Tela de Boas-vindas (Figura 8) é o primeiro contato do usuário com o aplicativo e tem como objetivo causar uma impressão positiva, apresentando de forma clara a proposta da plataforma. Ela introduz o sistema e oferece as opções de *login* e *cadastro*, conduzindo naturalmente à Tela de Login e Cadastro (Figura 9 e 10). Nessa etapa, o usuário pode autenticar-se ou criar uma nova conta, contando com validações básicas, como a verificação do formato de e-mail, e *feedback* visual em caso de erro. Embora ainda não esteja conectada ao banco de dados, essa interface já demonstra o fluxo de acesso e estabelece uma experiência de uso consistente.

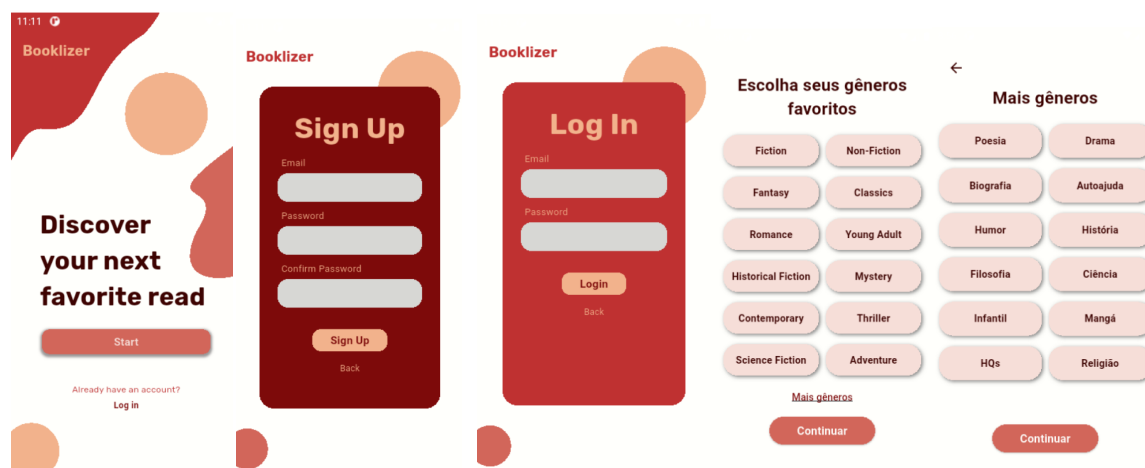
Após o cadastro, o usuário é direcionado à Tela de Escolha de Gêneros (Figura 11 e 12), onde seleciona seus gêneros literários favoritos. A opção “Mais gêneros” amplia as possibilidades, permitindo uma personalização mais detalhada das preferências. Essas escolhas alimentam diretamente o algoritmo de recomendação, que adapta as sugestões exibidas na Tela Principal de Recomendação de Livros (Figura 13 e 14). Nessa tela, o usuário visualiza *cards* deslizáveis com capa, título, autor e nota média, podendo indicar interesse ou desinteresse de forma intuitiva ao deslizar os livros. Essa experiência dinâmica e personalizada representa o núcleo funcional do sistema, tornando o processo de descoberta de novas leituras mais envolvente e relevante.

A experiência continua na Tela de Estantes (Figura 15), que organiza as leituras do usuário em duas categorias, “Tenho Interesse” e “Lido”. Essa estrutura facilita o acompanhamento do progresso e o planejamento de futuras leituras, ainda que, no momento, funcione apenas como uma representação visual. De forma complementar, a Tela de Busca (Figura 16) possibilita encontrar livros por título, autor ou palavra-chave, com sugestões automáticas e filtros por gênero. Assim como a tela de estantes, ela se encontra em estágio de protótipo, ilustrando o fluxo de navegação e a disposição dos componentes.

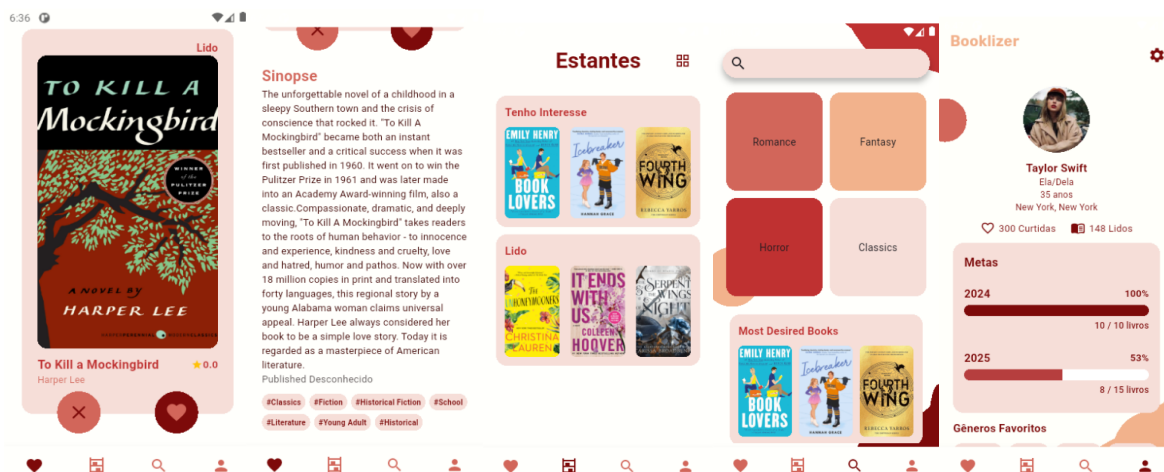
Por fim, a Tela de Perfil do Usuário (Figura 17) centraliza as informações pessoais e literárias, como nome, foto, localização, metas e histórico de leitura. Embora ainda seja uma

estrutura estática, sem funcionalidades de edição ou atualização, ela estabelece as bases para futuras melhorias voltadas à personalização da experiência do leitor dentro do aplicativo.

A implementação da interface do Booklizer consolidou uma base sólida para o sistema, promovendo a integração entre *design* e funcionalidade. As telas de Escolha de Gêneros e Recomendação de Livros já estão conectadas à API e ao sistema de recomendação, enquanto as demais servem como base para futuras expansões. Essa etapa garantiu coerência entre design, funcionalidade e experiência do usuário, fortalecendo o avanço do projeto e sua viabilidade prática.



Figuras (8, 9, 10, 11, 12) - Tela de Boas-Vindas (8), Telas de Login e Cadastro (9 e 10), Telas de Escolha de Gênero (11 e 12)
Fonte: Autoria própria.



Figuras (13, 14, 15, 16, 17) - Tela de Recomendação de Livros (13 e 14), Tela de Estantes (15), Tela de Busca (16), Tela de Perfil do Usuário (17)
Fonte: Autoria própria.

4.3 Experimentos e Discussões

Com o objetivo de avaliar o funcionamento geral do sistema, foram realizados experimentos práticos que contemplaram o fluxo completo de navegação e interação do usuário

dentro do Booklizer. Esses testes buscaram verificar a integração entre as telas, a estabilidade do aplicativo desenvolvido e a eficácia do modelo de recomendação.

Um dos experimentos consistiu em testar a navegação entre as diferentes telas do Booklizer, a fim de avaliar o fluxo e a resposta do sistema durante o uso. Foram simuladas as transições desde a tela de Boas-vindas até as telas de Login, Escolha de Gêneros, Recomendações, Estantes e Perfil, observando-se o comportamento dos botões, menus e rotas configuradas no aplicativo. O teste demonstrou que as transições ocorrem de forma fluida, garantindo que o usuário compreenda facilmente o caminho até as principais funcionalidades. Esse experimento confirmou a consistência da navegação e reforçou a usabilidade do sistema, assegurando uma experiência intuitiva ao longo de todo o fluxo de uso.

Nos demais experimentos, simulou-se a jornada do usuário desde o primeiro acesso até a geração de recomendações personalizadas. Após o cadastro, o sistema direcionou o usuário à tela de escolha de gêneros, onde foram selecionados os temas “Fiction” e “Drama”. Ao confirmar a seleção, a aplicação comunicou-se corretamente com a API de recomendação, exibindo na tela principal obras compatíveis, como *To Kill a Mockingbird* e *The Great Gatsby*. Em uma segunda simulação, o usuário retornou à tela de gêneros para alterar suas preferências, adicionando categorias como “Science Fiction” e “Mystery”. Para que as novas configurações fossem aplicadas, foi necessário reiniciar a aplicação, repetindo o processo anterior. Após o novo carregamento, o sistema apresentou recomendações atualizadas, como *Dune* e *The Hound of the Baskervilles*.

De modo geral, os experimentos confirmaram que o Booklizer atinge seus objetivos de usabilidade. A integração entre design e funcionalidade proporcionou uma experiência fluida e agradável, enquanto o modelo (GNN) demonstrou eficiência na geração de recomendações relevantes. Estes resultados consolidam a viabilidade técnica e conceitual do Booklizer, reforçando o seu potencial.

5. Conclusão

O trabalho teve como propósito o desenvolvimento do Booklizer, um sistema de recomendação literária personalizada baseado em Redes Neurais Gráficas (GNNs). A pesquisa partiu do problema de identificar como técnicas de aprendizado profundo, aplicadas em estruturas de grafos heterogêneos, podem aprimorar a recomendação de livros, buscando preencher uma lacuna observada nos aplicativos de leitura disponíveis atualmente, que carecem de personalização mais precisa.

Para alcançar esse objetivo, foi projetada uma aplicação interativa que integra um modelo de recomendação desenvolvido com PyTorch Geometric e uma interface construída em Flutter e Dart, comunicando-se por meio de uma API implementada em FastAPI. O processo metodológico envolveu a modelagem de grafos heterogêneos com diferentes tipos de nós, que incluem: livros, gêneros e autores.

Os resultados obtidos demonstraram que a inclusão de nós representando autores proporcionou uma melhora significativa na qualidade das recomendações. A comparação entre o modelo inicial, composto apenas por livros e gêneros, e o modelo atual, com a adição dos autores, evidenciou ganhos relevantes nas métricas de desempenho. Essa evolução reforça o potencial das

GNNs para capturar relações complexas entre elementos literários, resultando em recomendações mais contextualizadas.

Como contribuição, o Booklizer demonstra que o uso de grafos heterogêneos é uma abordagem promissora para sistemas de recomendação literária, oferecendo maior capacidade de personalização e uma experiência de descoberta de leitura mais inteligente. O impacto e a relevância acadêmica do projeto foram reconhecidos internacionalmente, uma vez que o artigo foi apresentado no ISLA 2025 (*Information Systems in Latin America Conference*), evento associado ao AMCIS 2025 e organizado pela LACAIS (*Latin American and Caribbean Chapter of the Association for Information Systems - AIS*). O trabalho foi agraciado com o prêmio *Best Research-in-Progress Paper Award*.

Entretanto, o trabalho apresenta limitações relacionadas principalmente ao tamanho do conjunto de dados e à necessidade de ajustes mais refinados nos hiperparâmetros do modelo, fatores que podem influenciar a performance final do sistema.

Para trabalhos futuros, sugere-se a ampliação da base de dados utilizada, a exploração de arquiteturas mais complexas de GNN, técnicas de otimização para aprimorar a eficiência do modelo, bem como o desenvolvimento de novas páginas interativas, que complemente o fluxo do usuário.

6. Referências bibliográficas

ABDELRAHMAN, M. et al. *Triplet loss for learning fine-grained image similarity*. arXiv preprint arXiv:2107.06187 [cs.CV], 2021. Disponível em: <https://doi.org/10.48550/arXiv.2107.06187>. Acesso em: 30 out. 2025.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, v. 17, n. 6, p. 734–749, 2005.

ALBRECHTSLUND, A. M. B. Amazon, Kindle, and Goodreads: implications for literary consumption in the digital age. *Consumption Markets & Culture*, v. 23, n. 6, p. 553–568, 2019. DOI: 10.1080/10253866.2019.1640216.

AMARAL, Adriana; SALVADOR, Tarciso. Folksonomia em sites de redes sociais segmentadas (SRSS) em livros: um estudo exploratório da interface do Goodreads. *RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação*, Campinas, SP, v. 16, n. 2, p. 397–413, 2018. DOI: 10.20396/rdbci.v16i2.8650424.

BATMAZ, Z.; KALELI, C.; DOĞRU, İ. A.; YILDIRIM, Y. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, v. 52, n. 3, p. 1871–1939, 2019.

CHAIRATANAKUL, N.; LIU, X.; HOANG, N. T. et al. Heterogeneous graph embedding with single-level aggregation and infomax encoding. *Machine Learning*, v. 112, p. 4227–4256, 2023. DOI: 10.1007/s10994-022-06160-5.

COSTA, Polyanna Possani da. *Teoria dos grafos e suas aplicações*. 2011. 77 f. Dissertação (Mestrado em Matemática) — Universidade Estadual Paulista, Instituto de Geociências e Ciências Exatas, Rio Claro, 2011.

JERASA, Shannon; BOFFONE, Trevor. BookTok 101: TikTok, Digital Literacies, and Out-of-School Reading Practices. *Journal of Adolescent & Adult Literacy*, v. 65, n. 3, p. 219–226, 2021. DOI: 10.1002/jaal.1199.

KHEMANI, B.; PATIL, S.; KOTTECHA, K. et al. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, v. 11, p. 18, 2024. DOI: 10.1186/s40537-023-00876-4.

KINGMA, D. P.; BA, J. *Adam: A method for stochastic optimization*. In: 3rd International Conference for Learning Representations (ICLR), San Diego, 2015. Disponível em: <https://doi.org/10.48550/arXiv.1412.6980>. Acesso em: 30 out. 2025.

KOUKI, P.; SINHA, R.; EKSTRAND, M. D. Challenges in responsible recommender systems. *arXiv preprint arXiv:2306.12680*, 2023.

MADDOX, Jenny; GILL, Fiona. Assembling “Sides” of TikTok: Examining Community, Culture, and Interface through a BookTok Case Study. *Social Media + Society*, v. 9, n. 4, 2023. DOI: 10.1177/20563051231213565.

MURRAY, Simone. Secret agents: algorithmic culture, Goodreads and datafication of the contemporary book world. *European Journal of Cultural Studies*, v. 24, n. 4, p. 970–989, 2021. DOI: 10.1177/1367549419886026.

PAVLOPOULOS, G. A. et al. Using graph theory to analyze biological networks. *BioData Mining*, v. 4, n. 10, 2011. DOI: 10.1186/1756-0381-4-10.

THELWALL, Mike; KOUSHA, Kayvan. Goodreads: a social network site for book readers. *Journal of the Association for Information Science and Technology*, v. 68, n. 4, p. 972–983, 2017. DOI: 10.1002/asi.23733.

W3SLLEY. *Bookcover API*. 2024. Disponível em: <https://github.com/w3slley/bookcover-api>

YING, R.; BOURGEOIS, D.; YOU, J.; ZITNIK, M.; LESKOVEC, J. *GNNExplainer: Generating Explanations for Graph Neural Networks*. In: *Advances in Neural Information Processing Systems* (NeurIPS), v. 32, p. 9240-9251, Vancouver, Canada, 8-14 Dec. 2019.

YUAN, X.; LIU, Q.; LONG, J.; HU, L.; WANG, Y. Deep image similarity measurement based on the improved triplet network with spatial pyramid pooling. *Information*, v. 10, n. 4, p. 129, 2019. DOI: <https://doi.org/10.3390/info10040129>

WU, Z.; PAN, S.; CHEN, F.; LONG, G.; ZHANG, C.; YU, P. S. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2021. DOI: 10.48550/arXiv.1901.00596.