

# Sistema de recomendação para o aprendizado de lógica no ambiente escolar

Camilla Pozer de Matos  
dept. de Ciência da Computação  
Instituto Federal Catarinense  
Videira, Brasil  
camillapozerm@gmail.com

**Resumo**—O presente trabalho irá destacar a importância do desenvolvimento lógico na infância, bem como explorar a ideia da criação de uma plataforma direcionada ao ensino da lógica nas escolas. O objetivo da plataforma é oferecer questões lógicas subdivididas em fases e categorias, as quais serão apresentadas aos alunos e respondidas de forma objetiva. Para a seleção de questões para determinado aluno, será utilizado um sistema de recomendação com inteligência artificial, que apresenta-se como foco deste trabalho.

**Palavras-chave**—lógica, sistema, recomendação, algoritmo.

## I. INTRODUÇÃO

Todo ser humano possui a capacidade de raciocinar, sendo que essa habilidade é aperfeiçoada conforme recebe estímulos. Apesar disso acontecer com maior intensidade na fase adulta, é na infância que o cérebro possui uma maior capacidade de absorver e aprender com esses estímulos, visto que se encontra em fase de desenvolvimento [1].

Baseado na premissa de que é na infância que devemos iniciar uma rotina de pensar logicamente, diversos autores estudaram e afirmam que a melhor maneira de introduzir a lógica na vida de uma criança é através da escola, visto que é um local de aprendizado e o aluno já está acostumado a desenvolver o máximo de suas capacidades cognitivas [2][3][4].

Seguindo as propostas abordadas pelos autores citados, a forma mais eficaz de aperfeiçoar o raciocínio é aliar a lógica ao ensino de programação, sendo que este último traz inúmeras vantagens aos estudantes. Para isto, aulas de lógica de programação foram aplicadas em escolas e comprovaram sua eficácia com ótimos resultados [2].

Apesar dos bons resultados, existem muitos impedimentos para trazer essas aulas de lógica a todas as escolas, por isso o objetivo é criar uma plataforma exclusiva para o ambiente escolar, com o intuito de eliminar a necessidade de um professor especializado e ampliar a margem de horários que os alunos terão para as aulas. A plataforma será composta por questões que serão recomendadas aos alunos através de uma inteligência artificial.

No presente artigo, busca-se trazer os melhores métodos de implementar um sistema de recomendação, bem como sua metodologia, tipos de sistema e os resultados obtidos com a execução do algoritmo.

## II. REFERENCIAL TEÓRICO

Atualmente, a nova geração de crianças e adolescentes que frequentam a escola vivem numa realidade chamada *cybercultura*, ou seja, estão imersos na tecnologia em um nível tão elevado, que os mesmos possuem dificuldades de direcionar a atenção ao ensino, assim como fixar um novo aprendizado. Por isso, [5] ressalta a importância de aliar a tecnologia ao ensino, visto que os dois fazem parte do cotidiano dos alunos e juntando-os pode-se conseguir o máximo potencial dos discentes.

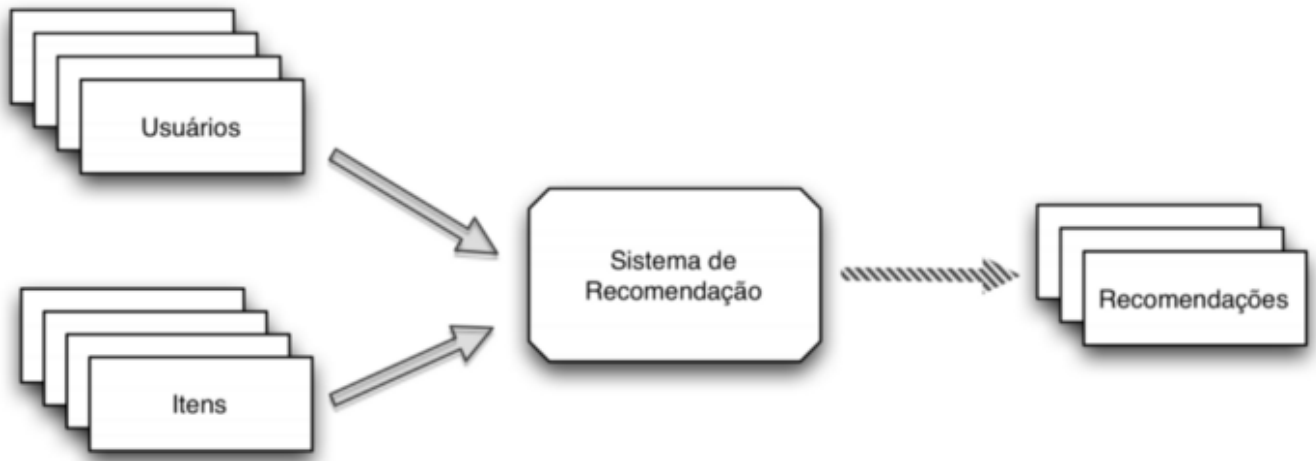
Uma forma de trazer a tecnologia, e, principalmente, a lógica para o âmbito educacional é através da programação, visto que facilita o ensino e proporciona muitos benefícios. Além de melhorar a criatividade, organização, desempenho pessoal e escolar e facilitar o aprendizado, a programação simplifica o processo de ensino de lógica, visto que torna os problemas mais tangíveis e fáceis de solucionar. Dessa forma, o aluno possui ganhos bilaterais, seja no aperfeiçoamento de seu raciocínio, assim como o aprendizado de uma linguagem de programação, que pode acarretar em uma futura profissão [6].

Apesar de ser o ideal, aplicar aulas de lógica com professores especializados é uma tarefa complicada, visto que não há professores suficientes na área, além de ser necessário um tempo específico para essas aulas, acarretando na modificação da grade curricular [5]. Como isso não é possível atualmente, propõe-se a criação de uma plataforma web direcionada ao ensino de lógica, sendo que no laboratório de informática das escolas, os alunos poderão acessar a plataforma e responder questões lógicas, tendo a possibilidade de acompanhar sua evolução cognitiva. Ainda, essas questões são subdivididas em fases e categorias, que serão recomendadas através de um sistema que utiliza inteligência artificial.

Segundo [7], a técnica de recomendação facilita a busca por informações relevantes em determinado contexto, principalmente quando a quantidade de dados envolvida é muito extensa. Um algoritmo de recomendação basicamente filtra e seleciona os itens que determinado usuário provavelmente vai gostar, com base em dados já existentes.

Como pode ser visto na figura 1, existem os usuários e os itens, que podem ser produtos em um e-commerce, filmes em um site de streaming, questões em uma plataforma de ensino, entre outros. Esses dois tipos de dados, usuários e itens, são repassados a um sistema de recomendação que gera uma lista personalizada de itens para determinado usuário [8].

**Figura 1.** Funcionamento de um sistema de recomendação.



Fonte: SANTANA (2018).

Dentro da área de inteligência artificial existem algumas formas diferentes de fazer a recomendação, como a recomendação por popularidade, a baseada em conteúdo e a feita através de filtragem colaborativa.

#### RECOMENDAÇÃO POR POPULARIDADE

Esse tipo de recomendação é o mais simples e menos personalizável para os usuários, consistindo apenas na popularidade dos itens. O algoritmo apenas calcula a quantidade de avaliações ou vendas de um determinado item e mostra ao usuário os que obtiveram maior número [9].

Apesar de fácil implementação, não é um método muito assertivo, visto que não leva em conta nenhum parâmetro relacionado às preferências do cliente.

#### RECOMENDAÇÃO BASEADA EM CONTEÚDO

Um dos tipos mais utilizados, a recomendação baseada em conteúdo utiliza como base os itens que o usuário gosta para sugerir outros que tenham características parecidas. Um exemplo visível é o caso da Netflix, que sugere para o usuário filmes e séries parecidos com o que ele já viu [9].

Outro exemplo de aplicação do sistema é abordado na figura 2, onde um usuário leu um artigo e recebe como recomendação um outro artigo semelhante.

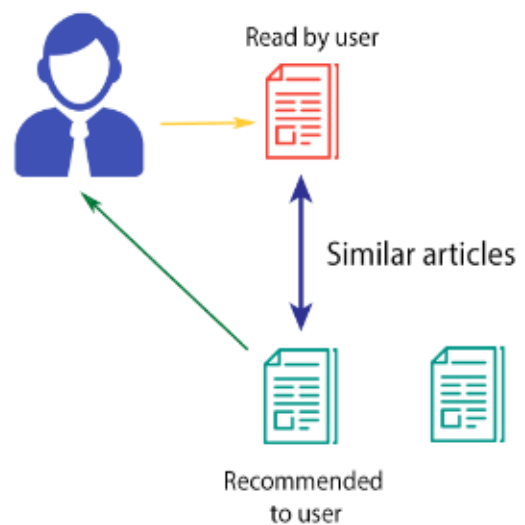
Esse algoritmo de recomendação possui algumas vantagens e desvantagens. A maior vantagem é o fato de não precisar de muitos dados para iniciar a recomendação, sendo que com apenas um item visto pelo usuário já é suficiente para recomendar outros semelhantes [10].

Outro ponto importante a se destacar é a dependência exclusiva apenas do usuário que está sendo avaliado. Nesse caso, não é necessário dados sobre outros usuários, sendo que não há nenhuma ligação entre eles. Além disso, através desse método de recomendação, é possível abranger itens novos e que não são muito populares, visto que a popularidade não é levada em consideração [10].

Porém, há uma desvantagem relacionada à privação do usuário a conteúdos não semelhantes ao que ele consome. Ou seja, o usuário acaba ficando em uma bolha de suas preferências, e não tem a chance de ver coisas diferentes, a menos que pesquise o conteúdo específico. Mesmo sendo ótimo no início, ao passar do tempo pode gerar um desinteresse por parte do usuário, visto que não há diversidade em sua lista de recomendações [8].

Outra questão importante nesse tipo de recomendação é definir a variável de semelhança. Para isso, não há um algoritmo genérico, sendo que cada tipo de conteúdo utiliza de diferentes meios para a classificação. Por exemplo, para recomendação de filmes e livros, pode ser utilizada a semelhança entre gênero ou autores, já em recomendação de produtos de um e-commerce, é necessário ir além e classificar por categorias, marcas, utilidades, entre outras características que podem ser avaliadas [8].

**Figura 2.** Recomendação baseada em conteúdo



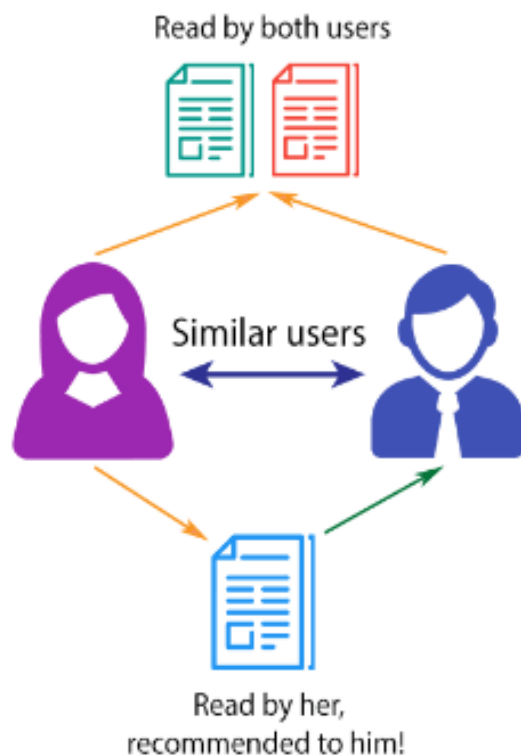
Fonte: SANTANA (2018).

## RECOMENDAÇÃO POR FILTRAGEM COLABORATIVA

Esse método de recomendação é baseado em filtrar itens e usuários com gostos semelhantes, ou seja, se usuários avaliam produtos de forma parecida, significa que possuem preferências parecidas [7]. Como a recomendação é feita com base na interação entre usuários e conteúdo, as características de ambos são ignoradas.

Como pode ser visto na figura 3, se dois usuários têm preferências parecidas, tudo o que um deles consumir será recomendado para o outro.

**Figura 3.** Recomendação por filtragem colaborativa



Fonte: SANTANA (2018).

Esse tipo de recomendação resolve alguns problemas citados na recomendação baseada em conteúdo, como não manter o usuário em uma bolha de preferências e também não necessita classificar o conteúdo de acordo com suas características. Geralmente, a lista de recomendações é gerada através das avaliações que os usuários dão para determinados itens [8].

Apesar de recomendar itens mais assertivos, a filtragem colaborativa enfrenta alguns problemas na questão dos dados. Para criar um sistema realmente eficaz, são necessários dados de todos os usuários e em uma larga escala, sendo que muitas vezes os usuários não avaliam tudo, deixando valores faltando. Além disso, não é possível recomendar itens a usuários novos, visto que o sistema ainda não conhece o padrão de suas preferências [8].

Para resolver o problema de valores faltantes, é possível utilizar machine learning e prever esses valores com base nos que já existem. Os algoritmos

mais utilizados para esse processo são os de fatoração de matrizes.

## III. METODOLOGIA

O intuito do trabalho é criar um algoritmo de recomendação de questões para ser utilizado na plataforma web para ensino de lógica. Após os alunos responderem as questões, poderão dar uma nota de 0 a 10, o que será usado como base para recomendar novas questões.

O algoritmo será feito das duas maneiras de recomendação citadas: por popularidade e por filtragem colaborativa. O método de recomendação baseada em conteúdo não será utilizado no momento, visto que ainda não há uma característica de classificação nas questões. O código será desenvolvido em python, através da plataforma Collaboratory do Google, que permite esse tipo de implementação.

A plataforma ainda está em desenvolvimento, por isso não existe uma base de dados dos usuários por enquanto. Até o momento que a plataforma for lançada e a base de dados por populada, será usado um dataset fictício com as mesmas variáveis determinadas: usuário, questão e avaliação (0-10). Posteriormente, o dataset será substituído pelo original, porém o algoritmo será o mesmo, visto que a formatação dos dados é a mesma.

### IMPLEMENTAÇÃO DA RECOMENDAÇÃO BASEADA EM POPULARIDADE

A implementação desse sistema é bem simples. Neste caso, serão aceitas apenas questões que possuem no mínimo 50 avaliações, as quais serão submetidas a um cálculo de média que utiliza o número total de votos somado ao limite estabelecido, sendo que ainda é multiplicado pela média global e média da questão. Dessa forma, será possível listar as questões recomendadas através de um ranking, onde cada uma possui uma nova nota calculada.

### IMPLEMENTAÇÃO DA RECOMENDAÇÃO POR FILTRAGEM COLABORATIVA

Dentro da área de filtragem colaborativa, há ainda outras categorias de recomendação, como filtragem colaborativa baseada em memória, que ainda pode ser dividida entre item-item ou usuário-usuário e filtragem baseada em modelo. Nesse trabalho serão desenvolvidas as recomendações de item-item e a recomendação baseada em modelo [10].

Partindo do pressuposto de que nem todos os alunos irão avaliar todas as questões, neste caso além de fazer um algoritmo para recomendação, será necessário preencher os valores faltantes com base em suposições. Toda a codificação será feita com o auxílio do pacote *surprise*, uma biblioteca python direcionada para sistemas de recomendação.

A formatação dos dados seguirá o padrão proposto pela documentação do *surprise* [11], sendo que o ideal é que eles sigam uma ordem de: usuário, item, avaliação. Com isso pronto, é possível fazer a divisão

dos dados entre amostra para treinamento e amostra para teste, que são de 80% e 20%, respectivamente.

Para realizar o treinamento são usadas as funções do pacote *surprise*, que no final retornam um resultado. Esse resultado pode ser acessado pelo método *get\_neighbord*, que retorna o número de recomendações especificados para determinado usuário ou item. Para isso, são necessários dois parâmetros: o *ID* e o número de recomendações.

Inicialmente, para o método de recomendação item-item, que encontra questões parecidas a partir das avaliações dos usuários, o modelo será treinado com base na função *KNNWithMeans*, que leva em conta a avaliação média de cada usuário [11]. Para sua execução, serão informados dois parâmetros: o número de vizinhos considerados e a medida de similaridade. Com isso, é possível retornar a recomendação para determinada questão.

Na implementação do algoritmo baseado em modelo, é preciso que os dados sejam tratados para poder reduzir as dimensões da matriz e encontrar padrões entre eles. Além disso, através dessa filtragem será possível aprender a preferência de um usuário e criar previsões para ele, caso um dado ainda seja desconhecido [10].

Após organizar a tabela, haverá a redução da matriz através de funções da biblioteca *sklearn*, que realizam esse processo por meio de decomposição de valor singular truncado. Com essa nova matriz é possível realizar a correlação, gerando outra matriz contendo a semelhança entre usuários, a qual será usada como base para a recomendação. Por fim, basta escolher uma questão e encontrá-la na matriz de correlação, sendo que a recomendação está pronta.

#### IV. RESULTADOS

Por ser uma plataforma simples direcionada ao ensino, a recomendação por popularidade já atende aos requisitos mínimos, trazendo as questões preferidas dos alunos. Além disso, qualquer aluno que se cadastre na plataforma terá acesso imediato às questões recomendadas.

Na figura 4 pode ser visto o resultado obtido com a recomendação por popularidade, informando o *ID* da questão, a sua média de nota, a contagem de notas obtidas e um score. Nesse caso, as questões são ordenadas pelo score de forma decrescente, mostrando as mais bem avaliadas primeiro.

As outras recomendações implementadas funcionam muito bem, porém há um consumo maior gasto na sua execução, além de requerer muitos dados para realizar uma recomendação certa. No caso do dataset fictício utilizado, haviam mais de 400.000 dados disponíveis, porém será demorado alcançar um número alto como esse no uso da plataforma.

A implementação de todos os algoritmos é relativamente simples, visto que há bibliotecas específicas para isso, como a *surprise*, utilizada neste

trabalho. Os resultados geralmente são assertivos, visto que depende das variáveis utilizadas e de como deseja-se obter a recomendação.

Figura 4. Resultados

	rating	rating_counts	score
question			
059035342X	9.048387	62	8.527574
385504209	8.747664	107	8.471900
446310786	8.949153	59	8.459526
142001740	8.828125	64	8.413055
312195516	8.523256	86	8.287414
679781587	8.446154	65	8.200768
316666343	8.172840	162	8.104190
671027360	8.169492	59	8.037507
743418174	8.157895	57	8.028863
446672211	8.125000	72	8.025314
452282152	8.000000	59	7.945764
044021145X	7.920000	50	7.900883
786868716	7.846154	52	7.863611
345370775	7.781818	55	7.829412
044023722X	7.753846	65	7.809463

Fonte: A autora (2021).

#### V. CONCLUSÃO

Existem diversos tipos de recomendação que atendem diferentes problemas, por isso é essencial entender a real necessidade da recomendação e implementá-la utilizando o algoritmo mais eficiente em cada caso.

Quando se há uma extensa base de dados e busca-se recomendar itens mais assertivos em relação à preferência de cada usuário, o ideal é utilizar algoritmos mais complexos, como a recomendação baseada em conteúdo ou a filtragem colaborativa baseada em modelo, que foi implementada neste trabalho.

No entanto, quando não há muitos dados disponíveis e a recomendação precisa acontecer desde o primeiro acesso do usuário, o ideal é utilizar a recomendação por popularidade. No caso do presente trabalho, a recomendação mais indicada e a qual será utilizada quando a plataforma estiver pronta, será a por popularidade, visando sua simplicidade de implementação e o fato de não precisar dos dados sobre os usuários, recomendando questões para qualquer novo usuário.

#### REFERÊNCIAS

- [1] PONTES, Edel Alexandre Silva et al. Raciocínio lógico matemático no desenvolvimento do intelecto de

- crianças através das operações adição e subtração. *Diversitas Journal*, v. 2, n. 3, p. 469, 2017.
- [2] ZANATTA, Andrei C. et al. Desenvolvimento de Jogos para Estimular a Programação/Lógica em Crianças de 9 a 12 anos. In: I SIMPOSIO IBERO-AMERICANO DE TECNOLOGIAS EDUCACIONAIS. Araranguá, 2017.
- [3] ZANETTI, Humberto A. P. et al. Proposta de ensino de programação para crianças com Scratch e Pensamento Computacional. *Tecnologias Sociedade e Conhecimento*, Campinas, v. 4, n. 1, 2017.
- [4] KROHL, Diego R., DUTRA, Taynara C., MATOS, Camilla P. Programação para alunos do Ensino Fundamental II: os benefícios dos jogos lógicos digitais no aprendizado. *Revista Tecnologias na Educação*, v. 28, n. 10, 2018.
- [5] NICOLETE, Priscila. et al. Integração tecnológica na educação básica pública brasileira: uma análise sobre a evolução temporal e a interdisciplinaridade do tema. *RIAEE*, v. 11, n. 4, p. 2064-2086, 2016.
- [6] REIFF, T. B. Programação de computadores: Uma proposta para o 9º ano do Ensino Fundamental. Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. 2017.
- [7] MENDES, T., *et al.* Algoritmo de recomendação colaborativa aplicado ao contexto educacional. *Braz. Ap. Sci. Rev.*, Curitiba, v. 3, n. 1, p. 703-711, 2019.
- [8] SANTANA, Marlesson. Deep Learning para sistemas de recomendação. Disponível em: <<https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-1-introdu%C3%A7%C3%A3o-b19a896c471e>>. Acesso em: 07 jul. 2021.
- [9] SANTOS, Gustavo. Criando Sistemas de Recomendação em Python. Disponível em: <<https://medium.com/data-hackers/criando-sistemas-de-recomenda%C3%A7%C3%A3o-em-python-ef350f601e3d>>. Acesso em: 07 jul. 2021.
- [10] Como funcionam os sistemas de recomendação? Disponível em: <<https://ilumeo.com.br/todos-posts/2019/08/12/como-funcionam-os-sistemas-de-recomendacao>>. Acesso em: 07 jul. 2021.
- [11] SURPRISE. Documentation. Disponível em: <<https://surprise.readthedocs.io/en/stable/>>. Acesso em 08. jul. 2021.