

Módulo de Recomendação para Padel (REPADEL): Classificação de Jogadores via Modelos de Predição

Gabriel Castagna Henrique, Alexandre de Oliveira Zamberlan

Curso de Sistemas de Informação

UFN - Universidade Franciscana

Santa Maria - RS

gabriel.castagna@ufn.edu.br; alexz@ufn.edu.br

Resumo—Este trabalho situa-se na área de Sistemas de Informação, com ênfase no desenvolvimento de Sistemas de Recomendação baseados em técnicas de aprendizado de máquina para categorização ou classificação. A pesquisa é motivada pelo crescimento do Padel em Santa Maria/RS e pela inexistência de soluções estruturadas para classificar de forma adequada o nível de proficiência dos atletas. Nesse contexto, propõe-se o REPADEL que coleta e processa dados de desempenho dos atletas, recomendando o nível de proficiência do atleta. Como principais resultados, destacam-se: a implementação, o treinamento e os testes do módulo REPADEL no ambiente Google Colab, por meio da comparação de desempenho e acurácia de algoritmos de predição na biblioteca PyCaret; e o projeto, desenvolvimento e integração do módulo REPADEL permitindo a gestão de avaliação de atletas.

Palavras-chave: Inteligência Artificial; Sistema de Recomendação; Mineração de Dados; Reconhecimento de Padrões.

I. INTRODUÇÃO

Sistemas de Recomendação são ferramentas computacionais que filtram e sugerem elementos de interesse para os usuários com base em seus perfis, preferências ou comportamentos anteriores. Esses sistemas utilizam técnicas de aprendizado de máquina, baseadas em redes neurais, recuperação da informação e mineração de dados para processar volumes de dados e gerar recomendações personalizadas (ou categorizações) com relevância [1].

Dutra [2], em reportagem, destaca que o Padel, esporte de raquete, tem ganhado popularidade mundial, incluindo no Rio Grande do Sul, especialmente na cidade de Santa Maria. O Padel combina elementos do Tênis e do Squash, sendo jogado em uma quadra menor e com regras próprias. Em Santa Maria, o esporte é praticado desde a década de 1990, mas teve um crescimento significativo nos últimos cinco anos [2]. Por ser acessível e divertido, é adequado para jogadores de todas as idades e níveis de habilidade. No entanto, reconhecer o nível de um atleta (sua proficiência) é essencial, especialmente para a participação em torneios e desafios. Esse reconhecimento deve ocorrer tanto pela interação entre os jogadores quanto pela avaliação contínua do professor ou treinador durante os treinos.

A. Justificativa e Objetivos

A definição adequada do nível de proficiência dos atletas de Padel é essencial para garantir competições equilibradas, treinos coerentes com o nível técnico e a formação de duplas compatíveis. A ausência de critérios objetivos de categorização pode gerar diversos problemas, como confrontos desbalanceados, progressão esportiva prejudicada, treinos ineficazes e desmotivação devido a desafios inadequados. Também dificulta a organização de torneios e compromete a evolução consistente dos praticantes[3]. Nesse contexto, torna-se fundamental desenvolver um módulo capaz de coletar e processar dados de atletas, analisando suas habilidades e indicando o nível de proficiência mais adequado. Esse tipo de ferramenta contribui diretamente para competições mais justas, treinos mais eficientes e maior engajamento dos jogadores.

Para a realização desta pesquisa, o módulo proposto foi fundamentado e integrado à estrutura e à dinâmica do Sistema Longevos¹, plataforma utilizada para a gestão do Circuito Longevos de Padel, destinado a atletas veteranos acima de 40 anos e organizado em etapas com sistema de ranking, de modo a incentivar a prática esportiva contínua nessa faixa etária. Esse sistema é mantido pelo Laboratório de Práticas da Computação² da instituição e encontra-se formalmente registrado no Instituto Nacional da Propriedade Industrial (INPI). Em operação desde 2023, o Sistema Longevos consolidou-se por meio da realização de diversos torneios, abrangendo um número significativo de jogadores e categorias distintas. Ressalta-se, ainda, a atuação da empresa parceira ER Clinic, que contribui de forma ativa para o desenvolvimento, especialmente na disponibilização segura dos dados na Internet e na aplicação de boas práticas de engenharia de software.

O objetivo deste trabalho é projetar, implementar e disponibilizar um módulo, integrado ao Sistema Longevos, capaz de coletar e processar dados de atletas, avaliar suas habilidades no jogo de Padel e recomendar o nível de proficiência mais adequado.

¹<https://longevos.lapinf.ufn.edu.br/>

²<https://lapinf.ufn.edu.br>

Como objetivos específicos, assume-se: i) realizar uma revisão bibliográfica sobre Sistemas de Recomendação; ii) pesquisar e testar bibliotecas do ecossistema Python aplicadas a Sistemas de Recomendação; iii) identificar e analisar os critérios para a definição da proficiência no Padel; iv) projetar, implementar, integrar e avaliar um módulo para recomendação no contexto do Padel.

II. REFERENCIAL TEÓRICO

Nesta seção, destacam-se os tópicos centrais relacionados à proposta, bem como as estratégias empregadas durante a construção do projeto. Englobando áreas como esporte Padel, Sistemas de Recomendação e suas variações, Inteligência Artificial e as ferramentas necessárias para o desenvolvimento deste trabalho. Além disso, são discutidos estudos e trabalhos relacionados, que contribuem para o embasamento teórico da pesquisa.

A. Esporte Padel e Treinamento

O Padel é um esporte de raquete praticado em duplas, disputado em uma quadra de 20×10 metros cercada por paredes de vidro ou muros, que podem ser utilizadas como parte das jogadas. Sua configuração moderna foi estabelecida em 1969 por Enrique Corcuera, no México, e atualmente o esporte é regulamentado pela Federação Internacional de Padel. Durante as partidas, aproximadamente um quarto das ações envolve o uso das paredes, os jogos apresentam duração média de 55 minutos e as trocas de bola duram, em média, 9 segundos [4, 5]. A prática do esporte é organizada em níveis de categorização, definidos a partir do desempenho técnico e tático dos jogadores. As classificações mais comuns incluem as categorias iniciante, intermediária e avançada, além de subdivisões por faixa etária e gênero. Em formatos mais específicos, empregam-se classificações numeradas, como 1^a, 2^a, 3^a, 4^a, 5^a e 6^a categorias, sendo a 1^a categoria composta por atletas com maior nível de proficiência. Entretanto, a categorização no Padel apresenta desafios relevantes. Estudos apontam que a ausência de critérios objetivos e a predominância de avaliações subjetivas podem gerar distorções nas classificações, prejudicando a equidade competitiva e dificultando a organização de confrontos equilibrados [3]. Essas imprecisões podem resultar em discrepâncias entre a categoria atribuída ao atleta e seu nível real de habilidade.

B. Sistemas de Recomendação

Sistemas de recomendação são ferramentas que auxiliam os usuários na descoberta de itens de interesse, reduzindo a sobrecarga de informação ao sugerir conteúdos personalizados com base em seus perfis e preferências [1]. Esses sistemas utilizam técnicas de aprendizado de máquina e recuperação de informações para gerar recomendações personalizadas, seja por meio da análise de classificações, interações ou preferências dos usuários.

Conforme Domingos [6], aprendizado de máquina é uma subárea da Inteligência Artificial (IA) que desenvolve algoritmos e modelos capazes de aprender a partir de dados, ou seja, identificar padrões e tomar decisões sem serem explicitamente programados para cada tarefa específica. Pode-se citar alguns exemplos de aplicações: reconhecimento de imagem e voz (como em sistemas de vigilância ou assistentes virtuais); sistemas de recomendação (Netflix, Spotify, Amazon); diagnóstico médico assistido por IA; detecção de fraudes bancárias; previsão de demanda e comportamento do consumidor; chatbots e assistentes virtuais.

Para contextualizar, é possível enquadrar a IA e o aprendizado de máquina da seguinte forma [6]: a Ciência da Computação contém a Inteligência Artificial, que, por sua vez, possui Redes Neurais e que, finalmente, contém as técnicas de aprendizados de máquina (supervisionado e não supervisionado).

Já a expressão mineração de dados é o processo de explorar grandes volumes de dados para descobrir padrões, correlações, tendências e informações úteis (*insights*) que não são imediatamente visíveis. Esse processo envolve a aplicação de algoritmos e técnicas estatísticas para extrair conhecimento a partir dos dados brutos [7]. A mineração de dados pode ser utilizada para: análise de comportamento de clientes (recomendação de produtos com base em histórico de compras); detecção de fraudes em sistemas bancários; segmentação de mercado e perfis de usuários; previsão de tendências (como vendas, epidemias, consumo de energia).

A mineração de dados é uma etapa essencial dentro de um campo mais amplo [7], que é a Ciência de Dados como parte estatística da Ciência da Computação. Os principais tipos ou categorias de algoritmos de mineração são a classificação, agrupamento, regras de associação e séries temporais. Como exemplos, pode-se citar a empresa Google que utiliza mineração de dados para oferecer anúncios personalizados. O Netflix usa padrões de visualização para recomendar novos títulos. Supermercados analisam o comportamento de compra para reorganizar prateleiras ou criar promoções direcionadas. Hospitais podem usar mineração de dados para prever complicações em pacientes com base em históricos médicos.

Retomando o tópico de Sistema de Recomendação, há alguns métodos conhecidos, como Filtragem Colaborativa; a Filtragem Baseada em Conteúdo; e a Filtragem Híbrida.

1) *Filtragem Colaborativa*: Os sistemas de recomendação baseados em Filtragem Colaborativa agregam avaliações e identificam semelhanças entre usuários com base em suas classificações, gerando assim novas recomendações [1]. Nesses sistemas a previsão do interesse de um usuário em determinado item é realizada por meio da análise de correlações entre itens e suas avaliações. Esse processo inicia-se com o cálculo da similaridade entre usuários e itens, permitindo a identificação de padrões de preferência. Dessa forma, a recomendação de um item para um usuário-alvo

ocorre a partir da escolha de itens similares aos selecionados por outros usuários.

Por exemplo, considere dois usuários da Netflix: o usuário João assistiu e avaliou positivamente as séries Stranger Things, The Witcher e Dark, enquanto a usuária Ana também demonstrou interesse em Stranger Things e The Witcher. Com base nessa similaridade de preferências, o sistema de recomendação pode sugerir à Ana a série Dark, mesmo que ela ainda não tenha assistido, assumindo que seu gosto é semelhante ao de João.

A Filtragem Colaborativa pode ser subclassificada em duas abordagens principais: Filtragem Colaborativa baseada em Modelo e Filtragem Colaborativa baseada em Memória [1]. Na abordagem baseada em modelo, são aplicadas técnicas de aprendizado de máquina (redes neurais) e mineração de dados para processar as avaliações dos usuários sobre os itens. Com isso, é possível treinar um modelo capaz de realizar previsões mais precisas, aprimorando o desempenho do sistema de recomendação [8]. Já na abordagem baseada em memória, a recomendação é realizada a partir da similaridade entre as escolhas dos usuários ou das características dos próprios itens. Para essa abordagem, um dos algoritmos mais utilizados é o *k-Nearest Neighbors* (*kNN*), que opera sobre a base de usuários ou itens, classificando-os com base em sua similaridade.

2) *Filtragem baseada em Conteúdo*: A filtragem baseada em conteúdo é uma técnica amplamente utilizada em Sistemas de Recomendação, especialmente quando o objetivo é analisar o comportamento de cada usuário de forma individual, sem depender de informações de outros perfis. Essa abordagem funciona identificando as características dos itens que o usuário já consumiu ou avaliou positivamente, como gênero, autor, categoria ou palavras-chave e, com base nisso, recomenda novos itens semelhantes [9].

A construção do perfil do usuário pode ocorrer de duas formas principais: por meio de informações declaradas (explícitas) quando o próprio usuário informa suas preferências diretamente, como preenchimento de formulários ou avaliações diretas; e por meio de dados implícitos quando o sistema observa o comportamento do usuário, como histórico de navegação, tempo de permanência em itens e padrões de clique [10]. Essas informações são organizadas em um vetor de preferências, que representa o gosto do usuário. Esse vetor é então comparado com vetores que descrevem os itens disponíveis, permitindo ao sistema identificar quais são mais parecidos com o que o usuário já gostou.

Seguindo o exemplo citado na Filtragem Colaborativa, a filtragem baseada em conteúdo também está presente em plataformas como a Netflix. Nesse caso, em vez de considerar o comportamento de outros usuários, o sistema analisa exclusivamente o histórico individual. Por exemplo, se um usuário costuma assistir a filmes de ação dirigidos por determinados autores, o sistema identifica essas preferências e sugere títulos com características semelhantes, como gênero,

direção ou enredo. Em *sites de e-commerce*, essa abordagem funciona de forma semelhante: os produtos sugeridos são baseados nas visualizações e compras anteriores do próprio usuário, levando em conta atributos como marca, categoria ou faixa de preço.

3) *Filtragem Híbrida*: A filtragem híbrida combina múltiplas técnicas de recomendação, geralmente integrando a Filtragem Colaborativa e a filtragem baseada em conteúdo, com o objetivo de superar limitações específicas de cada abordagem isolada. Essa integração é particularmente útil para contornar problemas como o início frio (*cold start*), a escassez de dados (*sparsity*) e a falta de diversidade nas recomendações [11].

A filtragem híbrida tem sido amplamente adotada em ambientes reais devido à sua robustez e capacidade de adaptação. Em bibliotecas digitais, por exemplo, modelos híbridos permitem recomendar livros combinando perfis de interesse e metadados dos documentos, aumentando a precisão das sugestões. Apesar da maior complexidade computacional e de implementação, os ganhos em cobertura e relevância justificam sua aplicação [12].

4) *Padrão Ouro*: Uma das ferramentas de *Machine Learning* é o padrão ouro (*gold standard*) [13] para auxiliar na validação e no treinamento dos modelos do sistema. Esse conceito refere-se a um conjunto de dados confiáveis e representativos, selecionados manualmente por profissionais da área, que serve como base de comparação para avaliar a qualidade das previsões realizadas por modelos de aprendizado de máquina.

Neste estudo, foi aplicado um questionário³ a atletas de Padel da cidade de Santa Maria/RS, com o objetivo de verificar suas habilidades técnicas e táticas no jogo. Em seguida, o treinador ou professor responsável pôde selecionar as respostas dos atletas que realmente considera pertencentes a uma determinada categoria (padrão ouro). Essas respostas foram armazenadas em uma base composta por 34 perguntas e aproximadamente 200 respostas. O treinador ficou responsável por revisar essas respostas e indicar os jogadores que melhor representam os diferentes níveis de proficiência.

As respostas desses jogadores formaram, então, o conjunto de referência utilizado para treinar e avaliar o sistema. Dessa forma, o uso do padrão ouro pode contribuir para aprimorar a precisão do modelo e garantir que as recomendações estivessem alinhadas à realidade observada nas quadras.

5) *Acurácia, Precisão, Recall, AUC, Kappa e MCC*: Para mensurar a qualidade dos modelos de classificação, foram utilizadas métricas reconhecidas na literatura de aprendizado de máquina: acurácia, precisão e *recall*. Essas métricas são calculadas a partir da matriz de confusão, que compara as previsões do modelo com os valores reais, classificando os resultados em quatro categorias [14]: Verdadeiro Positivo

³O questionário encontra-se no Apêndice A

(VP) são casos corretamente classificados como positivos; Verdadeiro Negativo (VN) casos corretamente classificados como negativos; Falso Positivo (FP) casos incorretamente classificados como positivos; e Falso Negativo (FN) casos incorretamente classificados como negativos.

A partir desses valores, calculam-se as métricas [15]: i) acurácia mede a proporção de previsões corretas em relação ao total de amostras; ii) precisão indica a proporção de previsões positivas que realmente são positivas; iii) *recall* representa a capacidade do modelo em identificar corretamente os casos positivos; iv) AUC (*Area Under the Curve*), que mede a área sob a curva ROC e expressa a capacidade do modelo em distinguir entre as classes positivas e negativas, sendo valores próximos de 1 indicativos de excelente desempenho; v) F1-Score, que corresponde à média harmônica entre a precisão e o *recall*, representando um equilíbrio entre ambas as métricas, especialmente útil em casos de desbalanceamento de classes; vi) Kappa de Cohen, que avalia o grau de concordância entre as previsões do modelo e os valores reais, descontando a concordância que ocorreria por acaso, variando de -1 (discordância total) a 1 (concordância perfeita); vii) Coeficiente de Correlação de Matthews (MCC), que mede a qualidade geral das previsões, considerando todas as combinações da matriz de confusão (VP, VN, FP e FN), sendo 1 o desempenho perfeito, 0 o equivalente ao acaso e -1 o desempenho totalmente incorreto.

C. Ferramentas: Bibliotecas, Frameworks e Application Programming Interface (API)

Para a construção do módulo de recomendação proposto, utiliza-se o ecossistema Python, que dispõe de um conjunto abrangente de bibliotecas e frameworks voltados ao processamento de dados, ao desenvolvimento de modelos de recomendação e à integração com aplicações Web. Entre as principais ferramentas empregadas estão: Pandas [16], NumPy [17], PyCaret [18], Django [19] e Bootstrap [20].

A biblioteca Pandas é responsável pela leitura e manipulação da base de dados, que inicialmente foi estruturada na planilha de perguntas e respostas do questionário sobre questões técnicas e táticas de alguns jogadores. Com o uso dessa ferramenta, é possível realizar tarefas como remoção de inconsistências, tratamento de valores ausentes, transformação de colunas e codificação de dados categóricos. O Pandas fornece estruturas de dados de alto desempenho e fáceis de usar, como *DataFrames*, que são essenciais para a análise de dados em Python.

Em conjunto com o Pandas, foi utilizada a biblioteca NumPy, que oferece suporte para operações vetoriais e matriciais de alto desempenho. Essa etapa é fundamental para garantir que os dados estejam prontos para o treinamento dos modelos de recomendação. O NumPy é reconhecido por ser a principal biblioteca para computação científica em

Python, fornecendo um objeto de matriz multidimensional e ferramentas para trabalhar com essas matrizes.

O PyCaret é uma biblioteca de código aberto (*open-source*) desenvolvida em Python, voltada à automação de tarefas de aprendizado de máquina, abrangendo desde o pré-processamento dos dados até a comparação e o ajuste de modelos. Atua como um *wrapper* ou um sistema ou API que encapsula e integra serviços ou outras bibliotecas, unificando diversos frameworks consolidados, como Scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt e Ray. Essa integração simplifica a criação de *pipelines* (fluxos estruturados) de aprendizado de máquina, automatizando etapas como limpeza de dados, transformação de variáveis, normalização, tratamento de valores ausentes e otimização de hiperparâmetros, com suporte a tarefas de classificação, regressão, clusterização, séries temporais, entre outras aplicações.

Para a interface da aplicação, é utilizado o framework Web Django, que permite a criação de uma aplicação responsiva, capaz de receber dados de entrada do usuário e retornar recomendações de forma dinâmica. A integração entre Django e Python possibilita que o modelo de recomendação seja incorporado diretamente à lógica da aplicação, garantindo rapidez no processamento e facilidade de manutenção. Além disso, o framework Bootstrap também é utilizado para compor a interface, uma vez que possui componentes para sites e aplicações Web, usando HTML, CSS e JavaScript, garantindo responsividade e boa experiência ao usuário.

Por fim, registra-se que um sistema de recomendação completo, normalmente, envolve: coleta e pré-processamento de dados (Pandas e NumPy); modelagem e treinamento (PyCaret); avaliação de desempenho (métricas como acurácia, precisão e *recall*); geração de recomendações (funções que retornam os *top-N* itens por usuário); exposição via API (Django e Bootstrap).

D. Trabalhos relacionados

Nesta seção, são discutidos trabalhos no contexto de Sistemas de Recomendação, contribuindo para o contexto do estudo realizado.

O trabalho de Douglas Diehl Lutz [21] propôs o desenvolvimento de uma aplicação Web voltada à recomendação de *issues* de projetos *open-source* hospedados no GitHub⁴, com o objetivo de facilitar a contribuição de desenvolvedores para essas iniciativas. A solução adota algoritmos híbridos de recomendação, combinando técnicas baseadas em conhecimento e filtragem de informações, além de utilizar a API do GitHub para a extração dos dados necessários.

João Pedro Raskopf Denardin Saldanha [22] apresentou um Sistema de Recomendação voltado à colaboração entre pesquisadores, utilizando uma rede de confiança construída a partir de publicações científicas em coautoria. O sistema

⁴<https://github.com>

propõe uma arquitetura híbrida, combinando recomendação baseada em conteúdo e métricas de confiança como centralidade e distância entre nós da rede, a fim de melhorar a qualidade das sugestões. A implementação foi realizada em Python, com o uso das bibliotecas Pandas, Scikit-learn e NetworkX, e utilizou dados pré-processados da plataforma Kennis, que consolida informações da base Lattes⁵.

Leonardo Furlan [23] desenvolveu um sistema Web de recomendação de artigos científicos com base nas áreas de interesses especificadas no Currículo Lattes dos usuários. O sistema integra técnicas de recomendação colaborativa, por meio do algoritmo *kNN* e da Correlação de Pearson. Além disso, combinou a recomendação baseada em conhecimento, a partir da análise do perfil acadêmico. A arquitetura foi construída pelo ecossistema PHP com banco de dados MySQL, utilizando *Web Crawlers* para extrair automaticamente artigos do Google Acadêmico.

Os trabalhos analisados abordam o desenvolvimento de Sistemas de Recomendação voltados a diferentes necessidades, mas apresentam características metodológicas em comum. Em todos os casos, foram adotadas abordagens híbridas, combinando mais de uma técnica de recomendação para melhorar a precisão dos resultados. O sistema de recomendação de *issues* para projetos *open-source* combinou filtragem de informações e técnicas baseadas em conhecimento para apoiar desenvolvedores na escolha de tarefas. O sistema voltado à colaboração em redes científicas utilizou uma arquitetura modular que integra métodos baseados em conteúdo e em confiança para sugerir novos parceiros de pesquisa. Já o estudo sobre bibliotecas digitais propôs o uso conjunto de técnicas de recomendação baseada em conhecimento e Filtragem Colaborativa para personalizar a busca de artigos científicos.

O presente trabalho propõe a criação de um módulo de recomendação baseado em técnicas de aprendizado supervisionado, com o objetivo de prever a categoria mais adequada para atletas de Padel. O sistema utiliza um modelo de classificação treinado a partir das respostas fornecidas pelos atletas em um questionário online, contendo informações sobre aspectos técnicos, táticos, físicos e comportamentais. A partir dessas respostas e de um modelo previamente treinado, o sistema é capaz de recomendar a categoria que melhor representa o nível de proficiência do atleta.

III. METODOLOGIA

Para o gerenciamento do projeto, desenvolvimento, disponibilização e integração do módulo REPADEL, adotou-se a metodologia ágil Scrum, aliada à técnica Kanban para organização e controle das atividades. As ferramentas e recursos utilizados incluem: Trello como quadro Kanban para controle das tarefas; GitHub como plataforma de versionamento e gerenciamento do código-fonte; Astah para a

modelagem de aspectos funcionais e estruturais do sistema; linguagem Python; bibliotecas Pandas, numpy e PyCaret; frameworks Bootstrap e Django. O Sistema Longevos, onde o módulo está inserido, está hospedado em servidor institucional do Laboratório de Práticas da Computação da UFN, obedecendo a boas práticas de segurança, políticas de *backup*, diretrizes legais para o tratamento de dados sensíveis, e regras de manutenção e acesso.

A. Aplicação da Metodologia Scrum

No contexto do Scrum, são destacados seus principais elementos: papéis, eventos, artefatos e as regras que os conectam. Entre os papéis, destacam-se: *Product Owner* que é responsável pela gestão e priorização do *Product Backlog*, que consiste em uma lista organizada de funcionalidades e requisitos do produto. Neste projeto, o treinador de Padel Lucas de Souza atua como *Product Owner*, sendo responsável por definir prioridades e garantir que as entregas agreguem valor ao sistema, especialmente no que diz respeito à manutenção e à evolução contínua; *Scrum Master* que atua como facilitador do processo, promovendo o uso adequado da metodologia, removendo impedimentos e incentivando a adoção das boas práticas ágeis. Essa função é desempenhada pelo orientador Alexandre de Oliveira Zamberlan; *Development Team* (Dev Team) que é composto pelo aluno Gabriel Castagna e o professor Alexandre de Oliveira Zamberlan. O time é responsável pela implementação dos itens definidos no *Sprint Backlog*, com foco na entrega de incrementos funcionais do sistema.

B. Eventos do Scrum

Um dos eventos centrais da metodologia [24] é o *Sprint*, ciclo de desenvolvimento de curta duração, que neste projeto teve periodicidade semanal. Durante o *Sprint*, a equipe se dedica à entrega de funcionalidades incrementais, completas e passíveis de validação. Ao final de cada ciclo, ocorre a *Sprint Review*, momento em que os itens desenvolvidos são apresentados ao cliente para validação — função assumida pelo treinador Lucas de Souza e o professor Alexandre de Oliveira Zamberlan. Esse retorno é essencial para o direcionamento das próximas entregas. Antes do início de cada *Sprint*, realiza-se o *Sprint Planning*, no qual são definidos os itens a serem trabalhados com base nas prioridades do *Product Backlog* e na aprovação do *Product Owner*. Complementarmente, a equipe realiza retrospectivas ao final de cada *Sprint* com o objetivo de identificar oportunidades de melhoria contínua no processo de desenvolvimento, aumentando a eficiência e a qualidade das entregas futuras.

C. Modelagem do REPADEL

A Figura 1 ilustra todos os casos de uso propostos para o sistema⁶. Esses casos de uso podem ser agrupados em

⁵<https://lattes.cnpq.br>

⁶Os principais descritivos de casos de uso podem ser encontrados no Apêndice B.

duas categorias principais: i) gestão básica (usuários, clubes, categorias de perguntas, perguntas e enquetes); ii) gestão da recomendação (respostas dos atletas, definição de atletas de referência, treinamento e aplicação do modelo, geração e visualização da recomendação).

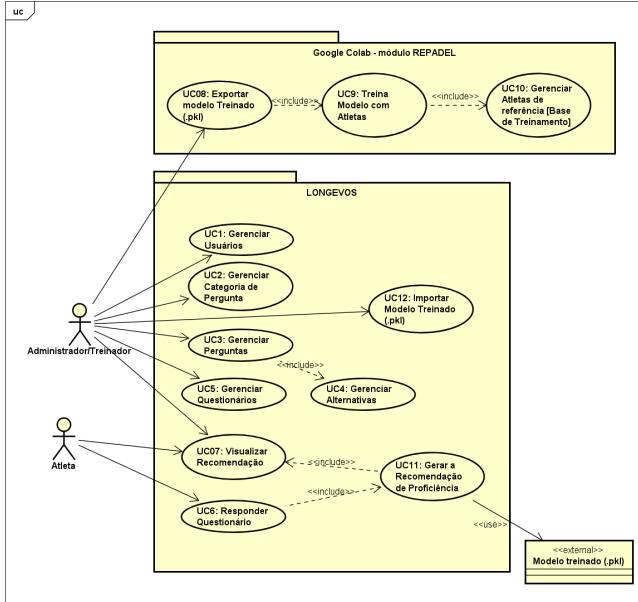


Figura 1. Diagrama de Casos de Uso do módulo REPADEL no Sistema Longevos.

D. Arquitetura do REPADEL

O módulo de recomendação REPADEL foi desenvolvido com uma arquitetura composta por duas etapas principais, concebidas para otimizar o processo de treinamento, validação e aplicação de modelos preditivos.

Na primeira etapa, a execução ocorre no ambiente Google Colab, onde as bibliotecas Pandas e PyCaret são instaladas e configuradas para o processamento e aprendizado de máquina. Nesse ambiente, utiliza-se a base de dados proveniente do questionário respondido pelos 178 atletas, armazenada em formato XLSX. Essa base é empregada como conjunto de treinamento, sendo submetida a procedimentos de pré-processamento e limpeza conduzidos automaticamente pelo PyCaret.

Em seguida, são executadas rotinas do PyCaret responsáveis pela comparação de diferentes algoritmos de classificação, incluindo a avaliação de métricas de desempenho e a seleção automática do modelo mais adequado para o contexto da base de dados. Após a seleção, o modelo escolhido é submetido ao processo de treinamento supervisionado, resultando em um artefato final (arquivo executável ou binário) representando o modelo treinado e otimizado.

A segunda etapa do módulo está integrada ao Sistema Longevos, especificamente no menu REPADEL. Nessa fase,

o artefato gerado é incorporado ao aplicativo de questionários do sistema, permitindo a interação direta com os usuários (atletas cadastrados). A partir das respostas fornecidas em novos questionários, o módulo realiza a inferência utilizando o modelo treinado, gerando recomendações automáticas sobre as categorias mais adequadas a cada atleta.

Essa arquitetura híbrida, combinando o ambiente de desenvolvimento em nuvem (Google Colab) e a integração com o sistema Web (Longevos), permite a atualização contínua do modelo, garantindo escalabilidade, reprodutibilidade e eficiência no processo de recomendação.

E. Estudo estatístico realizado: processo de avaliação da proposta

Entre novembro e dezembro de 2024, sob a coordenação do professor e treinador Lucas de Souza, foi disponibilizado um questionário de autoavaliação⁷, com o objetivo de coletar dados de jogadores de Padel, visando à identificação de sua categoria e nível de proficiência técnica. Com base nessas informações, torna-se possível desenvolver um módulo de recomendação, uma vez que a base de dados resultante fornece elementos iniciais de referência e treinamento para modelos de aprendizado de máquina. Ao todo, foram registradas 178 respostas de atletas praticantes da cidade de Santa Maria. O questionário é composto por 34 questões, que abrangem aspectos como dados pessoais (sexo, faixa etária, tempo de prática no Padel), nível de experiência com o esporte, participação em torneios, habilidades técnicas relacionadas aos fundamentos do jogo (por exemplo, saque, voleio, smash), estilo e ritmo de jogo, objetivos com a prática esportiva, além da percepção individual do atleta quanto à sua própria categoria no Padel, entre outros fatores relevantes. Registra-se, por fim, que a última questão do questionário trata da autorização para uso e compartilhamento dos dados fornecidos pelo atleta, assegurando a legitimidade no acesso e processamento das informações. Para o desenvolvimento do módulo de recomendação, como já mencionado, foi utilizada a biblioteca PyCaret. Foram testadas duas planilhas distintas para o treinamento do modelo: i) a primeira contendo os dados brutos, exatamente como coletados no questionário, sem qualquer limpeza ou transformação; ii) e a segunda, com os dados tratados pelo Pandas e NumPy, em que as respostas foram convertidas para uma escala Likert de 1 a 5 e categorizadas conforme o tipo de variável.

O PyCaret realiza automaticamente as etapas de pré-processamento, como o tratamento de valores ausentes, codificação de variáveis categóricas, normalização, padronização e remoção de *outliers*⁸. Em razão disso, observou-se que o desempenho entre as duas abordagens foi semelhante,

⁷Google Forms. Disponível nos Apêndices.

⁸Em mineração de dados, um *outlier* é uma observação que se desvia significativamente do padrão geral dos dados, ou seja, é um valor muito diferente dos demais e que não segue a tendência esperada do conjunto de dados [7].

apresentando acurácia de 0,7486 para a planilha tratada e 0,7262 para a planilha não tratada.

Tabela I
RESULTADO DO MELHOR MODELO REALIZADO PELO PYCARET.

Model	Accuracy	AUC	Recall	Prev.	F1	Kappa	MCC	TT (Sec)	
xgboost	Extreme Gradient Boosting	0.7262	0.0000	0.7262	0.7285	0.6997	0.6641	0.6783	0.0800
et	Extra Trees Classifier	0.7248	0.0000	0.7248	0.7280	0.6920	0.6600	0.6795	0.0660
rf	Random Forest Classifier	0.7181	0.0000	0.7181	0.6987	0.6785	0.6506	0.6674	0.0660
catboost	CatBoost Classifier	0.7171	0.0000	0.7171	0.7021	0.6801	0.6503	0.6669	1.3680
gbc	Gradient Boosting Classifier	0.6910	0.0000	0.6910	0.6862	0.6654	0.6209	0.6344	0.1440
ridge	Ridge Classifier	0.6767	0.0000	0.6767	0.6861	0.6579	0.6033	0.6149	0.0460
dt	Decision Tree Classifier	0.6271	0.0000	0.6271	0.6129	0.5947	0.5424	0.5567	0.0440
lr	Logistic Regression	0.6267	0.0000	0.6267	0.6547	0.6080	0.5441	0.5587	0.4290
svm	SVM - Linear Kernel	0.5910	0.0000	0.5910	0.6156	0.5685	0.5038	0.5172	0.0470
lightgbm	Light Gradient Boosting Machine	0.5267	0.0000	0.5267	0.5312	0.5054	0.4199	0.4310	0.1830
knn	K Neighbors Classifier	0.4848	0.0000	0.4848	0.4214	0.4231	0.3422	0.3692	0.2410
ada	Ada Boost Classifier	0.4800	0.0000	0.4800	0.3520	0.3744	0.3508	0.4244	0.0550
nb	Naive Bayes	0.4376	0.0000	0.4376	0.4426	0.4095	0.3109	0.3249	0.0450
dummy	Dummy Classifier	0.2748	0.0000	0.2748	0.0760	0.1189	0.0000	0.0000	0.0470
lda	Linear Discriminant Analysis	0.2267	0.0000	0.2267	0.1678	0.1717	0.1318	0.1756	0.0500
qda	Quadratic Discriminant Analysis	0.1133	0.0000	0.1133	0.1757	0.1252	-0.0152	-0.0186	0.0480

A utilização do PyCaret foi fundamental neste processo, uma vez que a biblioteca permite submeter o conjunto de dados e realizar através de seu método *compare_models()*, a comparação entre algoritmos de aprendizado supervisionado, incluindo *Random Forest*, *Decision Tree*, *K-Nearest Neighbors*, *Support Vector Machine*, entre outros. Com base nas métricas de desempenho obtidas, o PyCaret identificou o modelo com melhor resultado para o treinamento. Em ambos os cenários analisados (planilha tratada e planilha não tratada), diferentes algoritmos foram avaliados quanto ao desempenho na tarefa de classificação dos atletas. No cenário com a planilha tratada, o algoritmo que apresentou o melhor desempenho foi o *Extra Trees Classifier*. Entretanto, quando avaliado com a planilha não tratada, o modelo com melhor desempenho foi o *XGBoost*. Apesar da pequena diferença de desempenho entre os modelos optou-se pela utilização do *XGBoost* como modelo final do sistema. Essa decisão foi motivada principalmente pela perspectiva de usabilidade: ao adotar o modelo treinado diretamente com as respostas textuais da planilha bruta, elimina-se a necessidade de o usuário inserir códigos numéricos correspondentes a cada alternativa. Dessa forma, evita-se confusão, erros de digitação e inconsistências no preenchimento, contribuindo para uma experiência mais simples e intuitiva para o usuário.

F. Funcionamento do REPADEL

A construção e o treinamento do modelo para o módulo de recomendação ocorreu no ambiente Google Colab, utilizando, inicialmente, a planilha Excel como base de dados, referentes ao questionário respondido pelos atletas. Essa etapa teve caráter experimental, permitindo avaliar o comportamento dos modelos antes de sua integração ao Sistema Longevos, que utiliza o banco de dados MySQL para armazenamento e consultas simultâneas.

O processo foi implementado em etapas, conforme os códigos que seguem.

```
import pandas as pd
from pycaret.classification import *
# leitura da planilha ou base de dados
df = pd.read_excel("Pastal.xlsx")
df = df.dropna(subset=[ "AUTOAVALIAÇÃO: CATEGORIA" ])
```

As duas primeiras linhas importam as bibliotecas essenciais: Pandas e PyCaret. Em seguida, a base de dados é carregada e as linhas com valores ausentes na variável-alvo ‘AUTOAVALIAÇÃO: CATEGORIA’ são removidas, assegurando que apenas registros completos sejam utilizados no treinamento.

A configuração do ambiente experimental é feita pela função *setup()*, que define parâmetros como a variável-alvo, a semente aleatória (*session_id=123*), a normalização dos dados e a proporção de divisão entre treino e teste.

```
# configuracao da base de treino
exp = setup(
    data=df,
    target="AUTOAVALIAÇÃO: CATEGORIA",
    session_id=123,
    normalize=True,
    train_size=0.8, #80% da base para treino
    ignore_features=[ "DADOS_PESSOAIS: NOME" ] )
```

O parâmetro *ignore_features* exclui a coluna contendo o nome dos participantes, evitando viés no aprendizado. Após a configuração, o PyCaret automaticamente realiza todas as etapas de preparação dos dados. O comando *compare_models()* executa a comparação entre múltiplos algoritmos e retorna o de melhor desempenho, sendo possível visualizar o resultado na Tabela I.

```
# funcao de comparacao dos modelos existentes
best_model = compare_models()
```

Em seguida, o modelo selecionado é ajustado com *tune_model()*.

```
# funcao de ajustes de PyCaret
tuned_model = tune_model(best_model)
```

Para validar a capacidade do modelo em prever corretamente novas amostras, uma segunda planilha (‘NovosAtletas.xlsx’) foi carregada e submetida ao modelo ajustado.

```
# outro ajuste em nova planilha ou base
new_data = pd.read_excel("NovosAtletas.xlsx")
predicoes = predict_model(tuned_model, data=new_data)
predicoes.to_excel("Previsoes_NovosAtletas_texto.xlsx",
index=False)
```

Por fim, o modelo final foi salvo em formato binário por meio da função *save_model()*, permitindo que ele seja reutilizado no Sistema Longevos sem a necessidade de reprocessar ou retreinar toda a base.

```
save_model(tuned_model, "modelo_repadel")
```

Do lado do Sistema Longevos, é possível integrar o modelo executável gerado e selecionado, que é carregado com *load_model()* e utilizado diretamente para gerar recomendações.

```
modelo_carregado = load_model("modelo_repadel")
```

Dessa forma, o módulo pode classificar novos atletas de maneira imediata e consistente, utilizando o conhecimento previamente adquirido, como um serviço dentro do Longevos.

G. Resultados do módulo junto ao Sistema Longevos

O módulo de recomendação projetado, implementado, treinado e avaliado foi integrado ao Sistema Longevos, como já mencionado. A inclusão do módulo de recomendação justifica-se pelo fato de o Sistema Longevos já possuir histórico de uso consolidado e base de dados representativa, oriunda de diferentes competições e perfis de atletas. Esse contexto favorece a validação do módulo em cenários reais, ao mesmo tempo em que enriquece as funcionalidades do sistema, agregando mecanismos de personalização capazes de auxiliar na indicação de categorias adequadas, estratégias de treinamento ou potenciais adversários.

The screenshot shows the Django Admin interface with the 'repadel' model selected. The left sidebar lists various models like 'EXPLORER', 'LONGEVOS', 'projeto', 'repadel', etc. The main area displays the 'repadel' model's fields: 'models.py' (containing code for 'CategoriaPerguntaAtivoManager', 'CategoriaPergunta', 'Alternativa', 'Questionario', 'Resposta'), 'admin.py' (containing code for 'PerguntaAdmin'), 'apps.py', 'models.py', 'tests.py', 'urls.py', and 'views.py'. The code in 'models.py' includes imports from django.apps, django.db, django.urls, and utils.gerador_hash.

Figura 2. Estrutura Model View Template REPADEL.

A Figura 2 mostra a inserção do REPADEL no Longevos. Na imagem, há visão do *Model*, em que as classes referentes ao questionário são listadas.

A Figura 3 e Figura 4 mostram, respectivamente, como o REPADEL se apresenta para o Administrador e para o Atleta. Registra-se que na *home* do atleta, já é possível visualizar dados resultantes do módulo de recomendação.

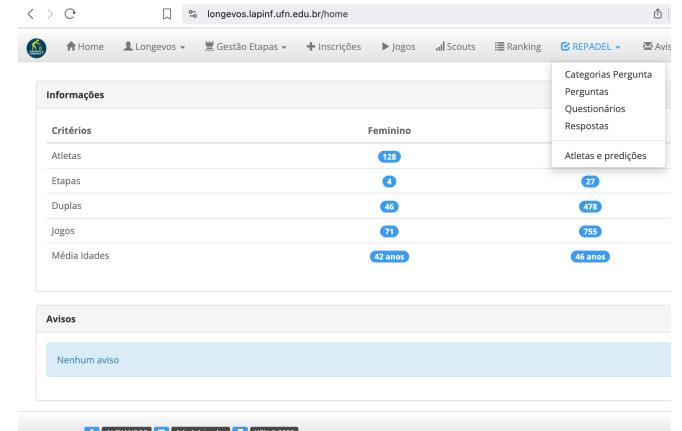


Figura 3. Estrutura Visão do Administrador do REPADEL.

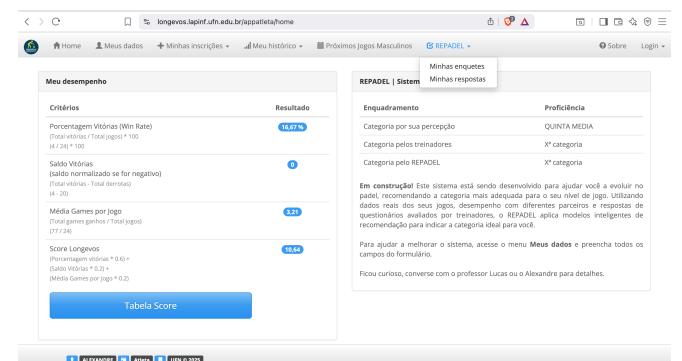


Figura 4. Estrutura Visão do Atleta do REPADEL.

1) Integração do modelo treinado (binário) com o Sistema Longevos: Dentro da estrutura do projeto, especificamente em projeto/repadel/ml_models e projeto/repadel/utils, foi realizada a integração do modelo treinado ao Sistema Longevos. A pasta ml_models passou a armazenar o arquivo binário do modelo treinado (.pkl), enquanto a pasta utils recebeu o componente *PredictionEngine*, responsável por carregar o modelo por meio da função *load_model()* e executar a predição com base no dicionário de respostas (Figura 5) fornecido pelo usuário ao responder um questionário.

O *PredictionEngine* converte o dicionário recebido em um DataFrame Pandas e utiliza a função *predict_model()* para gerar a recomendação final (Figura 6).

IV. CONCLUSÕES E TRABALHOS FUTUROS

A implementação do módulo REPADEL representa um avanço significativo na aplicação de técnicas de Inteligência Artificial no contexto esportivo, especialmente no Padel. A partir de um conjunto estruturado de dados, composto por um questionário de 32 questões respondido por 178 atletas, foi possível explorar processos de Mineração de

The screenshot shows a questionnaire interface with the following sections:

- CATEGORIA: COMPOSICAO_JOGO**
- BANDEJA_RETOMADA_REDE**
 - Minha bandeja é regular e me ajuda a manter o ponto.
 - Consigo acertar a bandeja, mas com pouca profundidade.
 - Tenho dificuldade em executar a bandeja com controle.
 - Uso a bandeja para controlar o ponto e pressionar o adversário.
- CONTROLE_RITMO**
 - Sim
 - Não
 - Não sei responder
- DEFESA_ATAQUE**
 - Atacar
 - Defender
 - Ainda não sei
- DEVOLUCAO_SAQUE**
 - Minhas devoluções são regulares e dificultam o próximo golpe do adversário.
 - Consigo devolver, mas sem colocação ou efeito.
 - Sou capaz de devolver saques com precisão e iniciar o ponto em vantagem.
 - Tenho dificuldade em devolver saques com controle.
- MOVIMENTACAO**
 - Consigo me movimentar razoavelmente, mas perco algumas bolas.
 - Tenho boa movimentação e me posiciono bem na maioria das jogadas.
 - Tenho excelente movimentação e antecipo jogadas do adversário.
 - Tenho dificuldade em me posicionar corretamente.
- POSICIONAMENTO**
 - Trabalho bem em dupla, cobrindo a quadra de forma eficiente.
 - Coordeno razoavelmente, mas ainda deixo buracos na defesa.
 - Tenho excelente entrosamento e crio jogadas estratégicas com o parceiro.
 - Tenho dificuldade em coordenar movimentos com o parceiro.
- SAIDA_PAREDE_ATAQUE**

At the bottom, there are navigation links: ADMINISTRADOR, Administrator, and UFN © 2023.

Figura 5. Algumas perguntas exibidas ao usuário

The screenshot shows the recommendation results for an administrator:

- Atleta:** ADMINISTRADOR
- Sua autoavaliação:** 3^a categoria
- Categoria prevista pelo REPADEL:** 4^a categoria

Figura 6. Recomendação ao final do questionário

Dados, identificar padrões relevantes e aplicar métodos de Sistemas de Recomendação orientados à categorização de atletas. O ecossistema Python, aliado ao uso do PyCaret, permitiu selecionar automaticamente o melhor modelo de predição, conduzir o treinamento, validação e avaliação, além de oferecer métricas para fundamentar a escolha final.

O desenvolvimento no Google Colab possibilitou um ambiente acessível, reproduzível e ágil para todas as etapas de modelagem, enquanto a integração do modelo treinado ao Sistema Longevos garantiu uma aplicação prática e funcional. A incorporação do REPADEL ao aplicativo de gestão de questionários demonstrou que os resultados não apenas atingiram níveis satisfatórios de desempenho. Além

disso, o suporte do PyCaret na exportação do modelo executável ou binário facilitou a implantação dentro do sistema, permitindo que o módulo de recomendação funcionasse de forma transparente e eficiente.

Em síntese, o REPADEL consolida a ponte entre teoria e prática, unindo técnicas modernas de aprendizado de máquina, ferramentas acessíveis e necessidades reais do cenário esportivo. O projeto evidencia o potencial de soluções baseadas em IA para apoiar decisões no treinamento e categorização de atletas, fortalecendo o papel da tecnologia como aliada na evolução do desempenho esportivo.

Como trabalhos futuros, destaca-se a necessidade de realizar uma avaliação sistemática do REPADEL dentro do próprio Sistema Longevos, uma vez que, nesta etapa, apenas a integração técnica foi concluída sem a aplicação de testes com usuários reais ou a coleta de métricas de usabilidade, precisão percebida ou impacto prático no processo de categorização. Outro caminho promissor é a expansão do módulo para suportar novos questionários, permitindo variar tanto a quantidade de perguntas quanto as categorias analisadas, abrindo espaço para estudos longitudinais, personalização por tipo de atleta ou até a criação de modelos especializados para diferentes perfis. Esses desdobramentos reforçam o potencial do REPADEL como ferramenta em constante evolução, alinhada às demandas do esporte e às oportunidades proporcionadas pela Inteligência Artificial.

REFERÊNCIAS

- [1] Francesco Ricci, Lior Rokach e Bracha Shapira. “Recommender systems: Techniques, applications, and challenges”. Em: *Recommender systems handbook* (2021), pp. 1–35.
- [2] Paulo André Dutra. *Liga Santa-Mariense de Padel consolida nova fase do esporte na cidade*. URL: <https://claudemirpereira.com.br/2023/06/esporte-liga-santa-mariense-de-padel-consolida-nova-fase-do-esporte-na-cidade/?mibextid=Zxz2cZ> (acesso em 08/03/2025).
- [3] Francisco Pradas et al. “Gender differences in physical fitness characteristics in professional padel players”. Em: *International Journal of Environmental Research and Public Health* 18.11 (2021), p. 5967.
- [4] Confederação Brasileira de Padel - COBRAPA. URL: <https://www.cobrapa.com.br/pagina/4/historia> (acesso em 02/04/2025).
- [5] Gema Torres-Luque et al. “Match analysis of elite players during paddle tennis competition”. Em: *International Journal of Performance Analysis in Sport* 15.3 (2015), pp. 1135–1144. DOI: 10.1080/24748668.2015.11868857. URL: <https://doi.org/10.1080/24748668.2015.11868857>.
- [6] Pedro Domingos. “A few useful things to know about machine learning”. Em: *Communications of the ACM* 55.10 (2012), pp. 78–87.

- [7] Pang-Ning Tan, Michael Steinbach e Vipin Kumar. *Introdução ao datamining: mineração de dados*. São Paulo: Ciência Moderna, 2009.
- [8] John S Breese, David Heckerman e Carl Kadie. “Empirical analysis of predictive algorithms for collaborative filtering”. Em: *arXiv preprint arXiv:1301.7363* (2013).
- [9] João Silva e Maria Oliveira. “Aprendizado de Máquina em Sistemas de Recomendação Baseados em Conteúdo Textual: Uma Revisão Sistemática”. Em: *Academia.edu* (2021). URL: https://www.academia.edu/55116561/Aprendizado_de_M%C3%A1quina_em_Sistemas_de_Recomenda%C3%A7%C3%A3o_Baseados_em_Conte%C3%BAdo_Textual_Uma_Revis%C3%A3o_Sistema%C3%A1tica.
- [10] Ana Costa e Pedro Souza. “Sistemas de Recomendação de Recursos Educacionais: conceitos, técnicas e aplicações”. Em: *Academia.edu* (2021). URL: https://www.academia.edu/58673513/Sistemas_de_Recomenda%C3%A7%C3%A3o_de_Recursos_Educacionais_conceitos_t%C3%A9cnicas_e_aplica%C3%A7%C3%A7%C3%B5es.
- [11] Alesson Bruno Santos Souza. “Uma Abordagem Híbrida para Sistemas de Recomendação Baseados em Filtragem Colaborativa”. Em: (2018). URL: <https://repositorio.ufba.br/bitstream/ri/27622/1/TCC%20-%20Uma%20Abordagem%20H%C3%A3brida%20para%20Sistemas%20de%20Recomenda%C3%A7%C3%A3o%20A7%C3%A3o%20Baseados%20em%20Filtragem%20Colaborativa%20-%20Alesson%20Bruno.pdf>.
- [12] Bruna Beatriz de Moura Vieira, Ketry Gorete Farias dos Passos e Vanessa Marie Salm. “Sistemas de recomendação em bibliotecas: iniciativas e proposta de um modelo teórico híbrido”. Em: *BiblioCanto* 9.1 (2023), pp. 43–66. DOI: 10.21680/2447-7842.2023v9n1ID32504. URL: <https://periodicos.ufrn.br/bibliocanto/article/view/32504>.
- [13] Ian H. Witten et al. *Data Mining: Practical Machine Learning Tools and Techniques*. 5th. Cambridge, MA: Morgan Kaufmann, 2023. Cap. 5, pp. 147–150. ISBN: 978-0-12-822260-6.
- [14] Larissa Alves Trevilato. “Análise Comparativa de Modelos de Representação de Texto e Métodos de Aprendizado de Máquina na Classificação de Notícias Falsas em Português”. Em: (2025).
- [15] Guanis B Vilela Junior et al. “Métricas utilizadas para avaliar a eficiência de classificadores em algoritmos inteligentes”. Em: *Revista CPAQV-Centro de Pesquisas Avançadas em Qualidade de Vidal* Vol 14.2 (2022), p. 2.
- [16] The Pandas Development Team. *Pandas - Python Data Analysis Library*. <https://pandas.pydata.org>. Acesso em: 30 jun. 2025. 2025.
- [17] The NumPy Development Team. *NumPy*. <https://numpy.org>. Acesso em: 30 jun. 2025. 2025.
- [18] Moez Ali. *PyCaret: An open source, low-code machine learning library in Python*. PyCaret version 1.0. Abr. de 2020. URL: <https://www.pycaret.org>.
- [19] The Django Software Foundation. *Django Web Framework*. <https://www.djangoproject.com>. Acesso em: 30 jun. 2025. 2025.
- [20] The Bootstrap Authors. *Bootstrap Framework*. <https://getbootstrap.com/>. Acesso em: 30 jun. 2025. 2025.
- [21] Douglas Diehl Lutz e Gustavo Stangerlin Cantarelli. *Desenvolvimento De Um Sistema De Recomendação De Issues Para Contribuições Em Projetos Open Source*. Santa Maria, RS, Brasil. Disponível em <https://tfgonline.lapinf.ufn.edu.br>: Trabalho de Conclusão de Curso Ciência Da Computação - Universidade Franciscana (UFN), 2020.
- [22] João Pedro Raskopf Denardin Saldanha e Alexandre De Oliveira Zamberlan. *Sistema De Recomendação Baseado Em Confiança Para Promover A Colaboração Em Redes De Pesquisa Científica*. Santa Maria, RS, Brasil. Disponível em <https://tfgonline.lapinf.ufn.edu.br>: Trabalho de Conclusão de Curso Ciência Da Computação - Universidade Franciscana (UFN), 2019.
- [23] Leonardo Antônio da Rosa Furlan et al. “Desenvolvimento de um sistema de recomendação para Bibliotecas Digitais”. Em: *Disciplinarum Scientia | Naturais e Tecnológicas* 19.1 (jan. de 2019), pp. 87–104. URL: <https://periodicos.ufn.edu.br/index.php/disciplinarumNT/article/view/2591>.
- [24] Tomasz Wykowski e Justyna Wykowska. *Lessons learned: Using Scrum in non-technical teams*. Set. de 2019. URL: <https://www.agilealliance.org/resources/experience-reports/lessons-learned-using-scrum-in-non-technical-teams/>.

APÊNDICE

A. Questionário ou Enquete aplicada e utilizada

Avaliação perfil jogador Padel

Atenção PADELISTA!

Questionário de **autoavaliação**, que ajudará cada jogador a **identificar sua categoria e escalão técnico**. A partir dessas informações, poderemos **organizar jogos e torneios mais equilibrados** e dinâmicos!

Responsável técnico: Professor Lucas de Souza - +55 55 98406-5446

* Indicates required question

Por favor, preencha seu nome completo *

Your answer _____

Sexo *

Feminino
 Masculino

Experiência: Há quantos anos pratica o padel? *

Até 2 anos
 Entre 3 e 5 anos
 Entre 6 e 8 anos
 Entre 9 e 11 anos
 Mais de 11 anos

Experiência: Quantas vezes por semana você costuma **TREINAR** com professor * ou técnico?

1 vez por semana
 2 vezes por semana
 3 vezes por semana
 Mais de 3 vezes por semana
 Não treino

Faixa etária *

Até 14 anos
 Entre 15 e 17 anos
 Entre 18 e 21 anos
 Entre 22 e 35 anos
 Entre 36 e 45 anos
 Entre 46 e 55 anos
 Acima de 55 anos

Preencha seu email para receber **feedback** do questionário *

Your answer _____

Qual posição você prefere jogar? *

Esquerda
 Direita
 Tanto faz
 Não sei ainda

Experiência: Quantas vezes por semana você costuma **JOGAR?** *

1 vez por semana
 2 vezes por semana
 3 vezes por semana
 Mais de 3 vezes por semana

Participação em Torneios: Já participou de torneios? *

Sim
 Não

Participação em Torneios: Se sim, em qual categoria vem competindo? *

- Não joga torneios
- Iniciante
- 7ª categoria
- 6ª categoria
- 5ª categoria
- 4ª categoria
- 3ª categoria
- 2ª categoria
- 1ª categoria

Habilidades Técnicas: *

Em uma escala de 1 a 5, em que 1 é Iniciante e 5 é Avançado, como avalia sua **MOVIMENTAÇÃO NA QUADRA?**

A **movimentação** é o saber quando avançar até a rede, quando voltar para o fundo, quando e como cobrir área para ajudar o parceiro.

- 1 - Iniciante
- 2
- 3
- 4
- 5 - Avançado

Habilidades Técnicas: *

Em uma escala de 1 a 5, em que 1 é Iniciante e 5 é Avançado, como avalia seu **SAQUE?**

- 1 - Iniciante
- 2
- 3
- 4
- 5 - Avançado

Composição do Jogo: Você prefere atacar ou defender? *

- Atacar
- Defender
- Ainda não sei

Habilidades Técnicas: *

Em uma escala de 1 a 5, em que 1 é Iniciante e 5 é Avançado, como avalia sua consistência nos **VOLEIOS?**

- 1 - Iniciante
- 2
- 3
- 4
- 5 - Avançado

Composição do Jogo: Você consegue controlar o **ritmo do jogo?** *

Controlar o ritmo de jogo no padel significa assumir o domínio sobre a velocidade, intensidade e cadência com que os pontos (trocadas de bolas) são jogados, obrigando os adversários a jogarem no seu ritmo preferido. Essa habilidade é crucial para impor sua estratégia, minimizar erros e explorar fraquezas dos oponentes.

- Sim
- Não
- Não sei responder

Habilidades Técnicas: *

Em uma escala de 1 a 5, em que 1 é Iniciante e 5 é Avançado, como avalia sua consistência nos **SMASHES?**

- 1 - Iniciante
- 2
- 3
- 4
- 5 - Avançado

Disponibilidade e objetivos: Você prefere jogar em duplas fixas ou rotativas? *

- Fixas
- Rotativas
- Não sei

Disponibilidade e objetivos: Qual é o seu principal objetivo ao participar do grupo? *

- Competir
- Socializar
- Melhorar como jogador
- Não sei
- Other: _____

Autoavaliação de Categoria: Baseando-se nas respostas dadas (reflexão realizada) e nas categorias de torneios locais, como você se enquadra? *

- Iniciante
- 7ª categoria
- 6ª categoria
- 5ª categoria
- 4ª categoria
- 3ª categoria
- 2ª categoria
- 1ª categoria
- Não sei

Golpes

Saque e Retoque: Como avalia sua consistência no saque? *

- Raramente consigo acertar o saque em áreas estratégicas.
- Consigo acertar o saque com regularidade, mas sem muito controle ou potência.
- Meu saque é consistente, consigo variar entre potência e colocação.
- Consigo sacar estrategicamente, pressionando o adversário.

Saque e Retoque: Você utiliza o retoque (saque + segunda bola)? *

- Não.
- Sim, mas com dificuldade em manter a qualidade.
- Sim, e geralmente consigo preparar o ponto com ele.
- Sim, e uso o retoque como parte fundamental da construção do ponto.

Voleios: Como avalia seus voleios de ataque? *

- Muitas vezes eu erro a direção ou intensidade.
- Acerto voleios simples, mas tenho dificuldade em pressionar o adversário.
- Consigo executar voleios sólidos com boa colocação.
- Meus voleios são consistentes e coloco pressão no adversário.

Voleios: Como avalia seus voleios de contenção (defensivos)? *

- Tenho dificuldade em devolver bolas rápidas ou com efeito.
- Acerto voleios defensivos básicos, mas com pouca profundidade ou controle.
- Tenho boa técnica para defender voleios e manter a bola em jogo.
- Sou capaz de transformar a defesa em ataque nos voleios.

Bandeja: Como avalia sua bandeja (golpe defensivo na transição para a rede)? *

- Tenho dificuldade em executar a bandeja com controle.
- Consigo acertar a bandeja, mas com pouca profundidade.
- Minha bandeja é regular e me ajuda a manter o ponto.
- Uso a bandeja para controlar o ponto e pressionar o adversário.

Autoavaliação de Categoria: Dentro da categoria escolhida, você se considera: *

- Fraco
- Médio
- Forte
- Não sei

Smash: Como avalia seu smash? *

- Dificuldade em acertar o smash com potência e controle.
- Consigo acertar smashes simples, mas sem finalizações efetivas.
- Tenho boa potência e controle no smash, especialmente para finalizar pontos.
- Meu smash é consistente e me permite definir pontos regularmente.

Jogadas de Fundo de Quadra: Como avalia sua consistência nos golpes de fundo * FOREHAND?

- Erro frequentemente ou tenho dificuldade em acertar profundidade.
- Consigo manter trocas de bola, mas com pouca precisão.
- Tenho regularidade nos golpes de fundo, com boa direção e profundidade.
- Sou consistente e uso os golpes de fundo para construir jogadas estratégicas.

Jogadas de Fundo de Quadra: Como avalia sua consistência nos golpes de fundo * BACKHAND?

- Erro frequentemente ou tenho dificuldade em acertar profundidade.
- Consigo manter trocas de bola, mas com pouca precisão.
- Tenho regularidade nos golpes de fundo, com boa direção e profundidade.
- Sou consistente e uso os golpes de fundo para construir jogadas estratégicas.

Jogadas de Fundo de Quadra: Como avalia suas devoluções de saque? *

- Tenho dificuldade em devolver saques com controle.
- Consigo devolver, mas sem colocação ou efeito.
- Minhas devoluções são regulares e dificultam o próximo golpe do adversário.
- Sou capaz de devolver saques com precisão e iniciar o ponto em vantagem.

Movimentação: Como avalia sua movimentação na quadra? *

- Tenho dificuldade em me posicionar corretamente.
- Consigo me movimentar razoavelmente, mas perco algumas bolas.
- Tenho boa movimentação e me posiciono bem na maioria das jogadas.
- Tenho excelente movimentação e antecipo jogadas do adversário.

Posicionamento: Como avalia sua capacidade de jogar em dupla? *

- Tenho dificuldade em coordenar movimentos com o parceiro.
- Coordeno razoavelmente, mas ainda deixo buracos na defesa.
- Trabalho bem em dupla, cobrindo a quadra de forma eficiente.
- Tenho excelente entrosamento e crio jogadas estratégicas com o parceiro.

Autorização e compartilhamento: Você autoriza divulgar seu nome e sua autoavaliação de categoria? Qual a categoria que se enquadrou. *

- Sim
- Não

Jogadas de Parede: Como avalia sua saída de parede (defensiva)? *

- Tenho dificuldade em realizar a saída de parede com controle.
- Acerto saídas simples, mas sem muita profundidade.
- Sou consistente nas saídas de parede, com boa direção.
- Transformo saídas de parede em golpes estratégicos para atacar.

Jogadas de Parede: Como avalia suas jogadas de parede ofensivas (*chiquita e bajada de pared*)? *

A **chiquita** é um golpe curto, jogado com pouca força e geralmente direcionado à parte baixa da quadra adversária, próximo à rede ou aos pés dos oponentes. É usada para retirar o adversário da posição ofensiva na rede, obrigando-o a se movimentar para devolver a bola em condições desfavoráveis.

A **bajada de pared** (descida de parede) é um golpe agressivo usado após a bola bater na parede de fundo da sua quadra. É uma resposta ofensiva a um *lob* (bola alta) adversário, permitindo que você aproveite o rebote da parede para atacar.

- Não utilizo essas jogadas no meu jogo.
- Tento usar, mas ainda erro frequentemente.
- Uso com consistência para preparar o ponto.
- Sou capaz de usar essas jogadas para pressionar e finalizar pontos.

B. Descrição casos de uso

ITEM	VALUE
UseCase	UC6: Responder Questionário
Summary	Permitir que o atleta responda o questionário disponibilizado pelo sistema, fornecendo os dados necessários para que posteriormente seja gerada a recomendação de proficiência.
Actor	Atleta
Precondition	O atleta deve estar autenticado no sistema e deve existir um questionário ativo configurado pelo administrador.
Postcondition	As respostas são registradas no sistema e ficam disponíveis para processamento no UC11 – Gerar a Recomendação de Proficiência.
Base Sequence	O atleta acessa o questionário disponibilizado. o sistema apresenta todas as perguntas e alternativas configuradas. o atleta preenche todas as respostas solicitadas. o atleta confirma o envio das respostas. o sistema salva as respostas no banco de dados. o caso de uso é finalizado.
Branch Sequence	Não possui
Exception Sequence	Não possui
Sub UseCase	UC11: Gerar a Recomendação de Proficiência
Note	

ITEM	VALUE
UseCase	UC08: Exportar modelo Treinado (.pkl)
Summary	Permitir que o administrador exporte o modelo do PyCaret treinado no Google Colab, gerando o arquivo .pkl utilizado posteriormente pelo sistema Longevos.
Actor	Administrador/Treinador
Precondition	O modelo deve ter sido previamente treinado com o PyCaret.
Postcondition	O arquivo .pkl contendo o modelo treinado é gerado e disponibilizado para download e armazenamento local.
Base Sequence	O administrador acessa o notebook de treinamento no Google Colab. O ambiente apresenta a função de exportação do modelo treinado. O administrador executa o comando de exportação. O Google Colab gera o arquivo .pkl. O arquivo é disponibilizado para download. O administrador salva o arquivo localmente. O caso de uso é concluído.
Branch Sequence	3a. Caso o comando de exportação exija parâmetros adicionais, o Colab solicita que o administrador os informe antes de prosseguir.
Exception Sequence	Não possui
Sub UseCase	UC10: Gerenciar Atletas de referência [Base de Treinamento], UC9: Treina Modelo com Atletas
Note	Este caso de uso produz o artefato necessário para que o UC12 – Importar Modelo Treinado seja executado no ambiente Longevos.

ITEM	VALUE
UseCase	UC11: Gerar a Recomendação de Proficiência
Summary	Processar as respostas fornecidas pelo atleta e utilizar o modelo treinado para classificar o nível de proficiência, gerando a recomendação final.
Actor	
Precondition	As respostas do questionário devem ter sido registradas (UC06). O modelo treinado deve ter sido importado para o sistema (UC12).
Postcondition	A recomendação de proficiência é gerada e disponibilizada para o atleta visualizar (UC07).
Base Sequence	O sistema inicia o processamento das respostas do atleta. Os dados são preparados e estruturados conforme o padrão esperado pelo modelo. O sistema executa o modelo de machine learning utilizando o arquivo .pkl importado. A classificação de proficiência é gerada. O sistema armazena o resultado no banco de dados. O caso de uso é concluído.
Branch Sequence	Não possui
Exception Sequence	E1: O modelo não está carregado → o sistema impede o processamento e notifica o administrador. E2: Erro durante a execução do modelo → o erro é registrado e uma mensagem de falha é gerada.
Sub UseCase	UC10: Aplicar Modelo as Respostas dos Atletas, UC07: Visualizar Recomendação, UC12: Importar Modelo Treinado (.pkl)
Note	Este caso de uso representa o núcleo do sistema de recomendação, pois produz a classificação final do atleta baseada no modelo de machine learning.