

WIKO: Sistema Inteligente de Monitoramento da Carga de Treinamento, Desempenho e Risco de Lesão em Atletas

Kauan Koech Schmidt¹, Nathan Will Martins¹, Edinilson da Silva Vida¹

¹Instituto Federal de Santa Catarina (IFSC)
Rua Heitor Villa Lobos, 225, 88506-400 – Lages/SC – Brasil

{kauan.k,nathan.wm}@aluno.ifsc.edu.br

edinilson.vida@ifsc.edu.br

Abstract. *This work presents WIKO, an intelligent system for monitoring training load, performance, and injury risk in athletes. Developed with React and Java + Spring Boot, it integrates validated metrics with Large Language Models (LLMs) to generate personalized preventive recommendations. The platform automates load calculations (PlayerLoad, ACWR, monotony, strain) and interprets results via Groq and Spring AI. System Usability Scale (SUS) evaluation demonstrated excellent usability (83.6), confirming its effectiveness as an accessible and reliable tool for supporting decision-making in sports training. Keywords: sports performance, training load, injury prevention, large language models, usability.*

Resumo. *Este trabalho apresenta o WIKO, sistema inteligente para monitoramento da carga de treinamento, desempenho e risco de lesão em atletas. Desenvolvido com React e Java + Spring Boot, integra métricas validadas com Modelos de Linguagem de Grande Escala (LLMs) para gerar recomendações preventivas personalizadas. A plataforma automatiza cálculos de carga (PlayerLoad, ACWR, monotony, strain) e interpreta resultados via Groq e Spring AI. Avaliação com System Usability Scale (SUS) atestou excelente usabilidade (83,6), confirmando sua eficácia como ferramenta acessível e confiável para apoio à decisão em treinamento esportivo. Implementação Palavras-chave: desempenho esportivo; carga de treinamento; prevenção de lesões; modelos de linguagem; usabilidade.*

1. Introdução

Segundo Okazaki et al. (2012), a tecnologia e o alto rendimento no esporte caminham lado a lado, sendo essenciais para maximizar o desempenho dos atletas. O uso de ferramentas tecnológicas no ambiente esportivo tornou-se uma exigência para a obtenção de vantagem competitiva, uma vez que possibilita o controle, monitoramento e análise precisa de variáveis fisiológicas, biomecânicas e técnicas relacionadas ao desempenho atlético. Como destacado por Okazaki et al. (2012), a tecnologia contribui não apenas para a melhoria do rendimento físico, mas também para o planejamento estratégico, a prevenção de lesões e o aumento da longevidade esportiva dos atletas. Esse cenário é especialmente relevante em modalidades de alto rendimento, nas quais pequenos ganhos de desempenho podem ser determinantes entre a vitória e a derrota.

No entanto, conforme apontado por Ráthonyi et al. (2018), o alto custo das tecnologias digitais e a desigualdade no acesso a esses recursos entre equipes de diferentes portes representam desafios significativos para a democratização da inovação no esporte. Embora tecnologias estejam amplamente disponíveis para clubes de elite, equipes amadoras e de menor expressão enfrentam barreiras econômicas e estruturais para incorporar essas ferramentas ao seu cotidiano. Isso resulta em um cenário de assimetria tecnológica, no qual apenas uma parcela dos atletas tem acesso às inovações que poderiam potencializar seu desempenho e reduzir riscos de lesão.

Neste contexto, surge a seguinte pergunta de pesquisa que norteia este estudo: como desenvolver um sistema inteligente de baixo custo para monitoramento da carga de treinamento, desempenho e risco de lesão em atletas que auxiliem treinadores em decisões baseadas em evidências?

Frente a essa realidade, revela-se a necessidade de um sistema inteligente capaz de processar e analisar automaticamente os dados de desempenho de atletas, permitindo avaliações objetivas e baseadas em dados. Esse sistema deve integrar tecnologias para fornecer informações detalhadas sobre o condicionamento físico e os fatores de risco para lesões, auxiliando treinadores e atletas na tomada de decisões mais assertivas.

Diante desse cenário, o objetivo deste trabalho é desenvolver o WIKO – Sistema Inteligente de Monitoramento da Carga de Treinamento, Desempenho e Risco de Lesão em Atletas, capaz de registrar dados esportivos, calcular métricas de carga de treino e interpretar esses resultados de forma acessível para treinadores e atletas. O diferencial da proposta é que, além de sinalizar risco de lesão e fadiga por meio de indicadores, o sistema utiliza Modelos de Linguagem de Grande Escala (*Large Language Models* – LLMs) para traduzir métricas complexas em recomendações práticas, explicativas e fundamentadas em evidências científicas (Banister et al., 1975; Foster, 1998; Boyd et al., 2011; Gabbett et al., 2016; Afonso et al., 2021). Essa abordagem amplia a acessibilidade das análises, permitindo que até equipes de menor porte tenham acesso a orientações claras e contextualizadas, antes restritas a departamentos de alto desempenho. Para alcançar esse objetivo, foram definidos os seguintes objetivos específicos:

- Criar uma interface gráfica intuitiva, que facilite a visualização e interpretação dos dados;
- Pesquisar na literatura científica e implementar o cálculo padronizado de métricas de carga de treino (*PlayerLoad*¹, *sRPE*², *ACWR*³, *monotony*⁴ e *strain*⁵), consolidando indicadores de desempenho e risco;
- Integrar um modelo de LLMs para gerar recomendações personalizadas baseadas em evidências científicas;
- Garantir acessibilidade e escalabilidade, viabilizando o uso por equipes de diferentes portes;
- Avaliar a eficácia e usabilidade do sistema por meio de testes com usuários.

¹Índice de carga externa baseado em acelerações e esforços locomotores.

²Carga interna obtida a partir da percepção subjetiva de esforço e duração da sessão.

³Relação entre a carga aguda recente e o condicionamento crônico.

⁴Indicador da variação ou repetitividade das cargas na semana.

⁵Estresse de treino semanal, produto do volume pela *monotony*.

A metodologia será composta por cinco etapas, visando alcançar os objetivos propostos. Na etapa 1 será realizada uma revisão bibliográfica sobre técnicas de Inteligência Artificial aplicadas ao esporte, com ênfase em LLMs, utilizando livros e artigos especializados como fontes principais. Na etapa 2 será implementada uma interface gráfica utilizando *React*, junto com a integração do *back-end* que será feito com *Java* com o ecossistema *Spring*. Na etapa 3 será desenvolvido o módulo responsável pelo cálculo das métricas de carga de trabalho e riscos de lesão e fadiga. Esses resultados servirão como insumo para a camada de inteligência do sistema. A etapa 4 assegurará a acessibilidade e escalabilidade do sistema, permitindo seu uso por equipes de diferentes tamanhos e contextos, garantindo que a solução seja eficaz tanto para pequenas quanto para grandes equipes. Na etapa 5, a eficácia e a usabilidade do sistema serão avaliadas por meio de testes com usuários reais, com o objetivo de validar a funcionalidade, a experiência de uso e a capacidade do sistema de atender às necessidades dos times.

Do ponto de vista metodológico, este estudo adota uma abordagem quantitativa, por meio da aplicação do teste *System Usability Scale* (SUS), visando obter dados objetivos sobre a usabilidade do sistema. Quanto à natureza, trata-se de uma pesquisa aplicada, pois busca desenvolver uma solução prática voltada ao monitoramento de desempenho esportivo com apoio de tecnologias inteligentes. Seu objetivo é exploratório, uma vez que pretende compreender o problema em profundidade e levantar subsídios para aprimorar a solução proposta, enquanto a avaliação com o teste SUS valida a solução. Os procedimentos metodológicos incluem pesquisa bibliográfica, que servem de base teórica e técnica para o desenvolvimento do sistema e para a compreensão do contexto esportivo analisado (Wazlawick, 2009).

Este artigo está organizado em cinco seções. A primeira seção introduz o tema central, destacando sua importância e justificando a relevância do projeto. Na segunda seção, é apresentado o embasamento teórico que sustenta a pesquisa. A terceira seção aborda os requisitos do sistema e detalha a modelagem da plataforma. Já a quarta seção apresenta o desenvolvimento e os resultados obtidos através da validação e teste SUS. Por fim, na quinta seção, são apresentadas as considerações finais, sintetizando as principais conclusões, apontando as limitações do estudo e sugerindo direções para pesquisas futuras.

2. Fundamentação Teórica

A fundamentação teórica deste trabalho apoia-se em um conjunto de conceitos, teorias e estudos que justificam o uso de tecnologias digitais no monitoramento e análise do desempenho esportivo. Essa base sustenta o desenvolvimento do WIKO e organiza-se em eixos complementares: os fundamentos científicos e tecnológicos do desempenho esportivo; os procedimentos de engenharia de *features* e análise de correlação; as métricas de monitoramento da carga de treino; os recursos de Inteligência Artificial, API REST e usabilidade que viabilizam a aplicação prática do sistema; e a análise de trabalhos correlatos que orientou as decisões de projeto e inovações da plataforma.

2.1. Fundamentos Científicos e Tecnológicos do Desempenho Esportivo

Esta seção aprofunda a compreensão sobre o desempenho esportivo, que envolve uma abordagem multidisciplinar e integra conhecimentos científicos e avanços tecnológicos. O

desempenho resulta da combinação de fatores fisiológicos, tecnológicos e metodológicos que precisam ser analisados de forma integrada. Lesão, fadiga e desempenho são conceitos centrais para o monitoramento esportivo e estão intimamente relacionados: a fadiga, quando mal gerenciada, pode aumentar o risco de lesão, enquanto ambos impactam diretamente o desempenho do atleta. A compreensão desses elementos fornece a base para o uso de métricas de carga e para o papel interpretativo dos LLMs no WIKO, permitindo transformar dados objetivos em recomendações práticas para treinadores e atletas. Para otimizar o desempenho dos atletas, é necessário considerar os mecanismos biológicos que regulam o esforço e a fadiga, as tecnologias aplicadas ao monitoramento de treinos e competições e as práticas de análise sistemática de dados, que transformam informações em subsídios para a tomada de decisão de treinadores e comissões técnicas.

2.1.1. Fisiologia do Exercício e Fadiga Muscular

A fisiologia do exercício investiga como o organismo humano responde aos estímulos do treinamento, analisando tanto respostas agudas (como aumento da frequência cardíaca, ventilação pulmonar e recrutamento muscular imediato) quanto adaptações crônicas (hipertrofia, maior capacidade oxidativa e eficiência cardiovascular). Esses processos são ilustrados na Figura 1, que apresenta as principais respostas fisiológicas imediatas e as adaptações estruturais de longo prazo provocadas pelo exercício, além dos mecanismos associados à fadiga muscular, como a redução de glicogênio e o acúmulo de íons hidrogênio. O entendimento desses fenômenos é essencial para compreender como o corpo mantém o equilíbrio interno diante de diferentes intensidades e durações de esforço (Machado e Marques, 2013).



Figura 1. Aspectos fisiológicos do exercício e efeitos sobre a fadiga muscular.

A fadiga muscular, do ponto de vista fisiológico, pode ser entendida como a incapacidade de manter o rendimento durante o exercício físico moderado e prolongado. Trata-se de um fenômeno multifatorial que pode ter origem central, quando há redução da condução dos impulsos nervosos do sistema nervoso central até a musculatura, ou periférica, quando ocorrem alterações no próprio músculo, como a depleção de glicogênio, a redução na liberação de cálcio e o acúmulo de íons hidrogênio, que afetam o pH intracelular e comprometem o processo de contração muscular (Santos e Blanco, 2009).

A compreensão desses mecanismos conecta-se diretamente ao conceito de carga de treinamento, frequentemente descrita em termos de carga aguda (carga acumulada em aproximadamente sete dias) e carga crônica (média da carga aguda ao longo de três

a seis semanas). Esse equilíbrio entre estresse imediato e adaptação de longo prazo é considerado central no monitoramento esportivo, por estar associado tanto à melhoria do desempenho quanto à prevenção de lesões (Banister et al., 1975; Gabbett et al., 2016).

2.1.2. Tecnologia no Esporte de Alto Rendimento

A aplicação da tecnologia no esporte de alto rendimento tornou-se estratégica, especialmente para a coleta, organização e análise de dados de desempenho. Entre as inovações mais impactantes, destacam-se os Sistemas de Posicionamento Global (GPS) e os dispositivos vestíveis (*wearables*), que se consolidaram como principais ferramentas para monitorar a carga externa em esportes coletivos. Esses recursos permitem registrar métricas como distância percorrida, velocidade máxima, acelerações, desacelerações e mudanças de direção, fornecendo indicadores objetivos da exigência física suportada pelos atletas (Cummins et al., 2013), conforme ilustrado na Figura 2.



Figura 2. Tecnologia aplicada ao esporte de alto rendimento: equipamentos para captação de dados.

Embora outros equipamentos, como cronômetros com fotocélulas e medidores de potência, continuem relevantes em contextos específicos, os dados de GPS e sensores inerciais embutidos em coletes tornaram-se a principal fonte de informação para clubes e pesquisadores. Essa predominância deve-se à capacidade de combinar grande volume de dados com praticidade na coleta em campo, permitindo análises em tempo real e comparações entre diferentes sessões de treino e jogos.

De acordo com o Apêndice A.1, diversas métricas derivadas desses dispositivos são essenciais para o monitoramento esportivo. A métrica “Distância total” representa o volume de deslocamento do atleta, enquanto a “Velocidade máxima” indica o pico atingido. Outras métricas, como a “Distância em alta intensidade” (faixas de velocidade acima de 25–30 km/h), capturam momentos de maior exigência física. Complementar-

mente, índices compostos como o *PlayerLoad* reúnem múltiplas dimensões do esforço em um único valor, auxiliando na avaliação da intensidade da sessão.

No contexto do WIKO, esses dados não são coletados diretamente, mas através da importação de arquivos em formato CSV, não envolvendo coleta direta na plataforma. Uma vez recebidos, o sistema aplica fórmulas obtidas na literatura esportiva, transformando dados brutos em métricas estruturadas. Esses resultados são disponibilizados aos treinadores de forma clara, acompanhados de alertas automáticos quando os valores indicam risco potencial de sobrecarga ou lesão. Além disso, o sistema integra LLMs, que fornecem recomendações baseadas em evidências científicas para auxiliar na interpretação das métricas e apoiar decisões práticas sobre ajustes de treino e estratégias de recuperação. Essa abordagem democratiza o acesso à análise de desempenho, reduzindo a dependência de *softwares* proprietários e de alto custo, frequentemente inacessíveis a equipes de menor porte.

Além do aspecto técnico, é fundamental considerar os princípios éticos e legais associados ao uso da tecnologia no esporte. Conforme destaca Drumond (2011), qualquer recurso tecnológico deve respeitar a integridade física e moral do atleta, promovendo justiça, equidade e benefícios reais para todos os envolvidos. Da mesma forma, a coleta e o tratamento de dados precisam estar em conformidade com a Lei Geral de Proteção de Dados Pessoais (LGPD), assegurando consentimento informado, transparência e medidas de segurança para proteger informações sensíveis (Brasil, 2018).

2.1.3. Análise de Desempenho Esportivo

De acordo com (Cotta, 2018, p. 14), a análise de desempenho esportivo pode ser definida como:

”a ciência que estuda a performance de um indivíduo inserido em atividade competitiva, fornecendo argumentos baseados em dados que racionalizam a tomada de decisão e auxiliam profissionais em sua avaliação.”

Essa análise vai além da simples observação do comportamento técnico ou tático dos atletas, fundamentando-se na coleta sistemática de dados, que são organizados, interpretados e utilizados para embasar decisões de treinamento, estratégias de jogo e prevenção de lesões. Dentre os dados mais relevantes nesse contexto, destacam-se marcadores fisiológicos como a variabilidade da frequência cardíaca, limiar de lactato, potência, volume de treino e carga interna. Esses indicadores são amplamente discutidos na literatura de fisiologia do exercício por sua relevância no monitoramento da intensidade do esforço, na identificação precoce da fadiga e na prevenção de lesões, sendo fundamentais para ajustes individualizados no planejamento esportivo (Machado e Marques, 2013).

Conforme Shamah (2021), os principais campos de investigação na análise do desempenho esportivo são: (i) protocolos para evolução do rendimento físico; (ii) estudo de lesões e processos geradores de fadiga; (iii) propostas metodológicas aplicadas à análise de jogo, relacionadas a aspectos físicos, táticos e técnicos; (iv) detecção do talento esportivo; e (v) a influência de diferentes variáveis situacionais nas partidas e no desempenho, e o resultado conquistado pelas equipes.

A atuação do analista depende, portanto, não apenas da observação, mas da capa-

cidade de filtrar e interpretar dados de forma que possam ser compreendidos e aplicados por técnicos, preparadores físicos e atletas. Nesse processo, os indicadores quantitativos assumem papel central: índices como o *PlayerLoad* (Boyd et al., 2011), a percepção subjetiva de esforço (sRPE) (Foster, 1998), e métricas derivadas de séries temporais, como a razão aguda:crônica (ACWR) (Gabbett et al., 2016), *monotony* e *strain* (Foster, 1998; Afonso et al., 2021), oferecem parâmetros objetivos para mensurar a carga de trabalho e antecipar riscos.

2.2. Engenharia de *Features*

Antes que as métricas de carga de treino sejam apresentadas em detalhe, é necessário compreender como os dados brutos são tratados para se tornarem variáveis úteis. Essa etapa, denominada engenharia de *features*, consiste no processo de seleção, transformação e criação de variáveis a partir dos registros primários, com o objetivo de destacar padrões relevantes e aumentar a eficácia dos métodos de análise (Zheng e Casari, 2018; Chicco, 2021).

A literatura de ciência de dados e aprendizado de máquina reconhece a engenharia de *features* como um dos fatores mais determinantes para o desempenho de modelos e métodos analíticos, pois garante que as informações capturadas traduzam relações significativas presentes nos dados. No contexto esportivo, essa prática é essencial para converter registros primários, como distância percorrida, acelerações ou valores de percepção subjetiva de esforço (sRPE), em métricas derivadas que representam de forma mais fidedigna a carga e o impacto do treinamento, tais como o *PlayerLoad*, a *monotony* e o *strain* (Cummins et al., 2013; Gabbett et al., 2016).

2.3. Métricas de Monitoramento da Carga de Treino

No contexto do WIKO, as métricas de carga fundamentam-se em fórmulas conceituadas na literatura, como *PlayerLoad*, sRPE, ACWR, *monotony* e *strain*. Esses indicadores permitem quantificar a carga de treinamento e o estado de preparação dos atletas. A partir deles, o sistema utiliza LLMs para gerar recomendações personalizadas, oferecendo orientações práticas para prevenção de lesões, otimização do desempenho e manejo da fadiga.

O monitoramento da carga permite quantificar o estresse imposto pelo treinamento e relacioná-lo às respostas de fadiga e às adaptações do atleta. Neste trabalho, empregamos métricas obtidas na literatura para estimar carga externa e interna, bem como índices derivados que auxiliam na gestão de risco e no ajuste da programação de treinos.

2.3.1. *PlayerLoad* (Carga Externa)

Tradicionalmente, o *PlayerLoad* é obtido a partir de acelerômetros triaxiais, como a raiz quadrada da soma dos quadrados das variações de aceleração nos eixos anteroposterior, mediolateral e vertical, sendo amplamente utilizado como índice robusto de carga externa (Boyd et al., 2011). No contexto do WIKO, quando não há dados de acelerometria direta, adota-se uma versão adaptada que integra variáveis derivadas de GPS e eventos de alta exigência, preservando a lógica de combinar múltiplas dimensões do esforço em um único índice:

$$PL_{sim} = \alpha \times dist + \beta \times (acc_{expl} + dec_{expl}) + \gamma \times sprints,$$

em que α , β e γ são pesos calibráveis ao contexto (modalidade, nível competitivo e perfil do elenco). Essa aproximação é documentada e utilizada como *fallback* sempre que não há sRPE disponível ou para complementar a análise de carga externa.

2.3.2. sRPE (Carga Interna)

A carga interna é estimada pela *session-RPE* (sRPE), proposta por Foster, calculada como o produto entre a percepção subjetiva de esforço da sessão e sua duração (Foster, 1998):

$$\text{sRPE} = \text{RPE}_{\text{sess}} \times \text{duration (min)}.$$

A sRPE captura a resposta psicofisiológica global ao treinamento e é especialmente útil por sua simplicidade, baixo custo e boa sensibilidade a variações de intensidade.

2.3.3. ACWR (Razão Carga Aguda:Crônica)

A razão aguda:crônica (ACWR) compara a carga de curto prazo (fadiga recente) com a carga de longo prazo (condicionamento) (Gabbett et al., 2016). Denotando CA como a soma (ou média ponderada) dos últimos 7 dias e CC como a média móvel de 3 a 6 semanas, tem-se:

$$\text{ACWR} = \frac{\text{CA}_{7d}}{\text{CC}_{3-6w}}.$$

Valores próximos de um indicam equilíbrio entre fadiga e preparo; desvios marcantes podem sinalizar subcarga (descondicionamento) ou sobrecarga (risco elevado).

2.3.4. Variação Semanal da Carga (%Up)

A variação semanal (%Up) quantifica a progressão da carga entre semanas consecutivas:

$$\%Up = \frac{\text{CA}_{\text{semana atual}} - \text{CA}_{\text{semana anterior}}}{\text{CA}_{\text{semana anterior}}} \times 100.$$

Progressões graduais são desejáveis; aumentos abruptos tendem a elevar o risco de fadiga excessiva e lesões por sobrecarga (Gabbett et al., 2016).

2.3.5. Monotony e Strain

A *monotony* reflete a variação (ou repetitividade) das cargas diárias em uma semana (Foster, 1998; Afonso et al., 2021):

$$\text{monotony} = \frac{\text{carga diária}}{\text{DP(cargas diárias)}}$$

Valores elevados indicam baixa variação entre as sessões (estímulo repetitivo). Já o *strain* combina volume semanal com a *monotony*, resultando em um indicador sensível do estresse cumulativo:

$$\text{strain} = \text{carga semanal} \times \text{monotony}.$$

Picos de *strain* têm sido associados a maior probabilidade de eventos adversos e servem como alerta para intervenções de recuperação e ajuste de distribuição de cargas (Foster, 1998; Afonso et al., 2021).

2.3.6. Interpretação Prática e Uso no WIKO

No WIKO, essas métricas são calculadas de forma padronizada a partir dos dados de cada atleta e agregadas em painéis com *labels* de risco e tendências. Em linhas gerais: (i) manter a ACWR próxima de um indica equilíbrio entre preparo e fadiga; (ii) progressões semanais moderadas (%Up) reduzem a chance de picos de estresse; (iii) *monotony* elevada sugere necessidade de variar estímulos; (iv) picos de *strain* pedem redução de carga e ênfase em estratégias de recuperação.

Esses resultados alimentam um módulo de recomendações baseado em LLMs, que gera orientações contextualizadas e referenciadas para treinadores e atletas. Dessa forma, o WIKO não apenas apresenta números, mas traduz as métricas em ações práticas, apoiando decisões sobre volume, intensidade, distribuição semanal e estratégias de recuperação (Foster, 1998; Gabbett et al., 2016; Afonso et al., 2021).

Nesse contexto, o sistema WIKO diferencia-se por integrar os princípios da análise de desempenho esportivo com recursos de Inteligência Artificial, automatizando etapas que tradicionalmente dependeriam exclusivamente do trabalho humano. Esse diferencial democratiza o acesso à análise de desempenho, oferecendo uma ferramenta de apoio à decisão técnico-científica também para equipes de menor porte.

2.4. Inteligência Artificial

Desde o início de sua trajetória, a Inteligência Artificial (IA – *Artificial Intelligence*) tem sido alvo de diferentes definições. Uma das mais conhecidas foi proposta por Rich e Knight em 1991, ao afirmarem que a IA se ocupa do estudo de computadores que fazem coisas que, naquele momento, pessoas faziam melhor. Essa definição, embora ampla, já indicava a busca por emular capacidades humanas em máquinas, seja por meio do raciocínio, aprendizado ou tomada de decisão (Cozman et al., 2021).

Ao longo do tempo, novas concepções surgiram, ampliando o escopo da IA. Russell e Norvig, na década de 1990, categorizaram o campo em abordagens que buscam criar sistemas que pensam como humanos, agem como humanos, pensam racionalmente ou agem racionalmente (Russell e Norvig, 2010). Essa diversidade reflete tanto a multiplicidade de objetivos quanto os desafios técnicos e filosóficos envolvidos.

A IA moderna evoluiu para subáreas especializadas, como o Aprendizado de Máquina (*Machine Learning*) e o Aprendizado Profundo (*Deep Learning*), responsáveis por grande parte dos avanços recentes. O crescimento do poder computacional e a disponibilidade de grandes volumes de dados impulsionaram essas técnicas, permitindo desde a análise de imagens médicas até a detecção de padrões complexos em séries temporais (Aggarwal, 2018). A relação hierárquica entre esses campos é ilustrada na Figura 3, em que os *Large Language Models* (LLMs) aparecem como uma subárea do *Deep Learning*, inserida no escopo mais amplo da Inteligência Artificial.

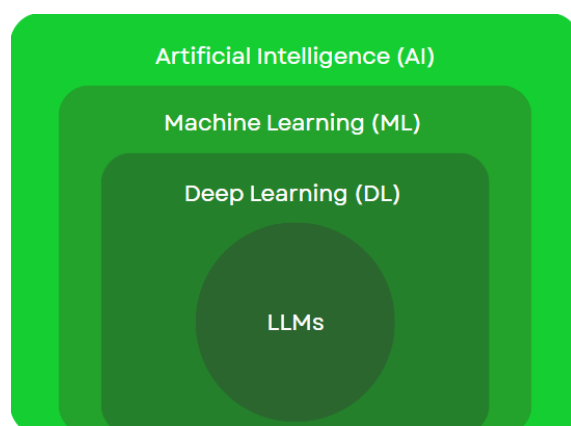


Figura 3. Relação hierárquica entre *Artificial Intelligence*, *Machine Learning*, *Deep Learning* e LLMs.

Atualmente, destaca-se a ascensão dos Modelos de Linguagem de Grande Escala (LLMs – *Large Language Models*), que ampliaram as fronteiras da IA ao processar linguagem natural em alto nível, interpretar contextos complexos e gerar recomendações baseadas em grandes volumes de conhecimento (Bommasani et al., 2021).

2.4.1. Fundamentos dos Modelos de Linguagem de Grande Escala (LLMs)

Os *Large Language Models* (LLMs) pertencem a uma classe de modelos de Inteligência Artificial conhecidos como *foundation models*, caracterizados pelo pré-treinamento em grandes volumes de dados textuais e pela capacidade de adaptação a diferentes tarefas por meio de ajustes mínimos (Bommasani et al., 2021). Entre suas principais bases técnicas está a arquitetura *Transformer*, que utiliza mecanismos de autoatenção para processar sequências de forma paralela e capturar relações de longo alcance entre elementos do texto (Vaswani et al., 2017). Essa estrutura permite que os LLMs representem a linguagem em espaços vetoriais de alta dimensionalidade, viabilizando operações de compreensão semântica, raciocínio e geração de linguagem natural. A Figura 4 ilustra de forma simplificada essa arquitetura, destacando o papel dos módulos de codificação (*encoder*) e decodificação (*decoder*) na transformação de sequências de entrada em saídas contextualizadas.

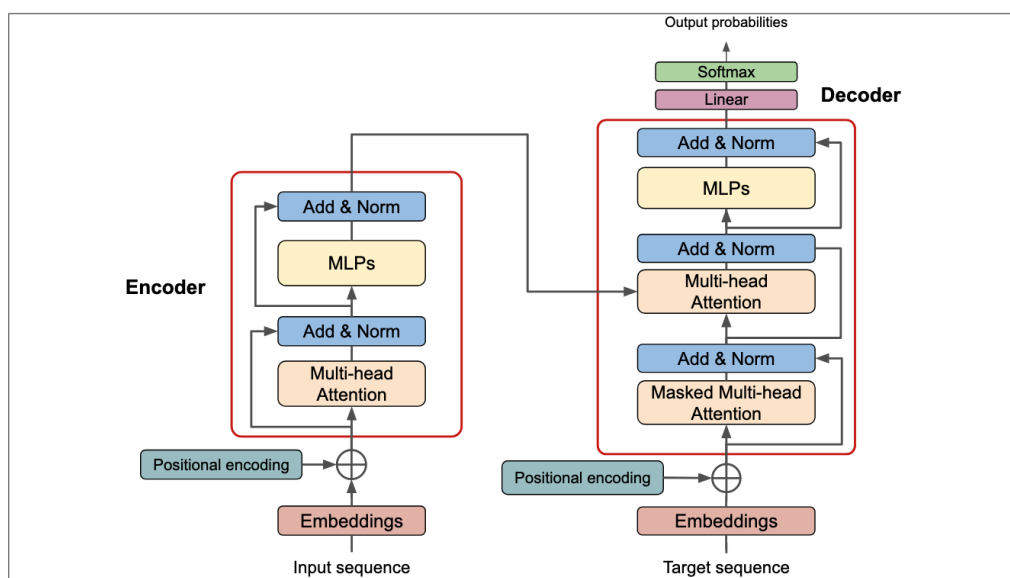


Figura 4. Arquitetura do modelo *Transformer*, base da maioria dos LLMs.
Fonte: Adaptada de Vaswani et al. (2017).

Outra característica central é a capacidade de generalização contextual, que possibilita aos modelos realizar tarefas diversas sem treinamento específico, apenas a partir de instruções fornecidas no momento da inferência (*in-context learning*). Esse comportamento inclui fenômenos como o *few-shot learning*, no qual o modelo ajusta suas respostas a partir de poucos exemplos, e o *zero-shot learning*, no qual executa tarefas apenas a partir de uma descrição textual. Tais propriedades ampliam a aplicabilidade dos LLMs a diferentes domínios, mesmo em cenários de dados limitados.

Além disso, os LLMs podem incorporar novos conhecimentos científicos de forma dinâmica, sem a necessidade de re-treinamento completo. Estratégias como o ajuste de *prompts* e o uso de *Retrieval Augmented Generation* (RAG) permitem integrar evidências atualizadas da literatura, assegurando que o modelo mantenha aderência a práticas contemporâneas e contextuais de cada área de aplicação (Lewis et al., 2020). Essa flexibilidade diferencia os LLMs de modelos mais restritos e amplia seu potencial como ferramentas de suporte à decisão em ambientes em constante evolução.

Esses aspectos técnicos constituem a base para compreender a relevância dos LLMs em cenários que exigem não apenas processamento de dados, mas também interpretação e comunicação de informações em linguagem natural.

2.4.2. LLMs versus *Machine Learning* Tradicional

Modelos supervisionados e não supervisionados de *machine learning* aprendem padrões a partir de *features* tabulares para tarefas como classificação e regressão. No contexto deste trabalho, tais modelos poderiam estimar risco de lesão/fadiga a partir de métricas como ACWR, *monotony* e *strain*. Entretanto, os LLMs pertencem a uma classe distinta: *foundation models* treinados em grandes volumes de dados textuais, capazes de realizar compreensão e geração de linguagem natural, raciocínio e síntese (Bommasani et al.,

2021).

Neste projeto, os LLMs não substituem a coleta e o cálculo das métricas objetivas; ao contrário, atuam como camada interpretativa e recomendativa. As métricas são calculadas no *back-end*, enviadas ao LLM em formato estruturado (*JavaScript Object Notation* – JSON) e, a partir de regras e evidências científicas, o modelo produz recomendações textuais contextualizadas (por exemplo, ajustes de volume/intensidade e estratégias de recuperação). Essa abordagem amplia a acessibilidade das análises, pois traduz indicadores técnicos em orientações práticas, mantendo rastreabilidade entre dado, interpretação e ação. Tal perspectiva está em consonância com os relatórios técnicos recentes, que destacam que modelos, como o *Claude 3*, podem apoiar áreas sensíveis, como saúde e bem-estar, da mesma forma que, no esporte, funcionam como suporte interpretativo e não como substitutos diretos de especialistas (OpenAI, 2023; Anthropic, 2024).

Esse contraste evidencia que os LLMs funcionam como uma camada interpretativa e textual, o que os torna particularmente adequados para domínios em que a tradução de métricas objetivas em recomendações práticas é essencial.

2.4.3. Modelos de Linguagem de Grande Escala (LLMs) no Esporte

Conforme descrito nas subseções anteriores, os LLMs apresentam características técnicas que os diferenciam de abordagens clássicas de *machine learning*, como a capacidade de contextualização, generalização e atualização contínua de conhecimento. Essas propriedades tornam o seu uso especialmente promissor em domínios nos quais a interpretação dos dados é tão relevante quanto sua coleta, como é o caso do esporte.

Na literatura científica esportiva, métricas como o sRPE, o ACWR, a *monotony* e o *strain* são amplamente reconhecidas como indicadores sensíveis do equilíbrio entre carga, fadiga e risco de lesão (Foster, 1998; Gabbett et al., 2016). Contudo, tais indicadores exigem interpretação adequada para que resultem em decisões práticas efetivas, o que frequentemente limita seu uso a equipes com analistas especializados. Nesse contexto, os LLMs funcionam como uma camada interpretativa, traduzindo resultados numéricos em recomendações textuais fundamentadas em evidências científicas, como ajustes de volume/intensidade ou estratégias de recuperação.

No WIKO, esse fluxo ocorre de forma estruturada: as métricas de carga de trabalho são calculadas no *back-end*, enviadas ao LLM em formato JSON e, a partir de *prompts* construídos com cuidado, visando o bem estar do atleta, retornam recomendações contextualizadas ao perfil do atleta e ao objetivo do treinador. Além de ampliar a rastreabilidade entre dado, interpretação e ação, essa integração torna o sistema mais acessível e democrático, permitindo que clubes de menor porte ou mesmo equipes de base usufruam de análises antes restritas a estruturas de alto rendimento.

Dessa forma, os LLMs não apenas expandem a utilidade prática das métricas, mas também promovem a democratização do conhecimento científico aplicado ao esporte, aproximando a análise de carga da realidade de diferentes contextos competitivos. Outro aspecto relevante é que os LLMs permitem a incorporação de novos conhecimentos científicos sem a necessidade de re-treinamento completo do modelo: por meio de ajustes de *prompts* ou do uso de técnicas de *Retrieval Augmented Generation* (RAG), o sistema

pode integrar rapidamente evidências atualizadas da literatura, mantendo-se alinhado às melhores práticas em ciência do esporte.

2.5. API REST

A API REST é um estilo de arquitetura amplamente adotado no desenvolvimento de aplicações *web* modernas, especialmente em contextos onde há separação entre cliente e servidor. O REST é baseado em um conjunto de restrições arquiteturais que promovem a criação de sistemas distribuídos escaláveis, flexíveis e de fácil manutenção, utilizando protocolos padrão da *web*, como o *Hypertext Transfer Protocol* (HTTP) (Masse, 2011).

No paradigma *RESTful*, os dados são organizados em recursos, identificados por URIs (*Uniform Resource Identifiers*). As operações sobre esses recursos são realizadas por meio dos métodos HTTP: GET (recuperar), POST (criar), PUT (atualizar) e DELETE (remover). Cada requisição é autônoma e sem estado (*stateless*), o que significa que o servidor não armazena contexto entre as chamadas, cada interação contém todas as informações necessárias para seu processamento. A Figura 5 ilustra essa estrutura, destacando a interação entre diferentes clientes e a API REST, bem como os fluxos de comunicação com fontes de dados e outros serviços.

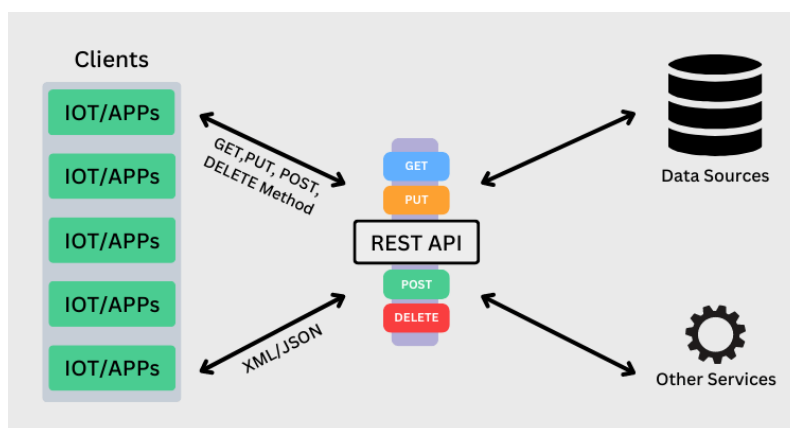


Figura 5. Diagrama representativo da arquitetura REST: comunicação entre clientes, métodos HTTP, e serviços conectados.

Fonte: Adaptada de Software (2021).

Entre as principais vantagens da API REST destacam-se: simplicidade de implementação, compatibilidade com múltiplas linguagens e plataformas, eficiência na comunicação via formatos leves como JSON, e facilidade de integração com aplicações *front-end* modernas. Essas características tornam o REST uma escolha natural para sistemas que exigem modularidade, portabilidade e evolução constante (Masse, 2011).

No contexto do WIKO, a arquitetura REST é utilizada para viabilizar a comunicação entre o *front-end*, desenvolvido com *React*, e o *back-end* em *Spring Boot*. Essa abordagem permite que os dados de desempenho inseridos ou importados pelos usuários sejam enviados ao servidor de forma padronizada e eficiente, recebendo em resposta as análises realizadas pelos modelos de Inteligência Artificial. Além disso, a estrutura REST favorece a escalabilidade do sistema e facilita futuras integrações com outras plataformas esportivas ou dispositivos de coleta de dados.

2.6. Teste SUS de Usabilidade

Conforme destacado por Nielsen(2007, p. 16), a usabilidade:

é um atributo de qualidade relacionado à facilidade do uso de algo. Mais especificamente, refere-se à rapidez com que os usuários podem aprender a usar alguma coisa, à eficiência deles ao usá-la, ao quanto lembram daquilo, ao seu grau de propensão a erros e ao quanto gostam de utilizá-la.

A usabilidade na *web*, segundo Nielsen (2007), deve ser baseada em observações reais do comportamento dos usuários, e não apenas em opiniões ou suposições. Observar como as pessoas interagem com um *site* em tarefas práticas é mais eficaz do que perguntar como elas acham que usariam a interface. Isso permite identificar problemas e melhorar a experiência com base em dados concretos.

Desta forma, o *System Usability Scale* (SUS) é uma metodologia desenvolvida por John Brooke em 1986 para mensurar, de forma rápida e eficaz, a usabilidade percebida de sistemas interativos. Com aplicação simples e de baixo custo, o SUS é amplamente utilizado em avaliações de produtos digitais tanto na indústria quanto na pesquisa acadêmica. A escala baseia-se em um questionário com dez afirmações, respondidas imediatamente após o uso do sistema, por meio de uma escala de *Likert* de cinco pontos, que varia de “discordo completamente” a “concordo completamente” (Brooke, 1986).

A Figura 6 ilustra visualmente a escala de *Likert* utilizada na coleta das respostas, permitindo aos participantes expressar suas percepções de maneira intuitiva e direta. Cada item pode ser respondido com uma pontuação de 1 a 5, onde 1 representa “discordo completamente” e 5 representa “concordo completamente”.

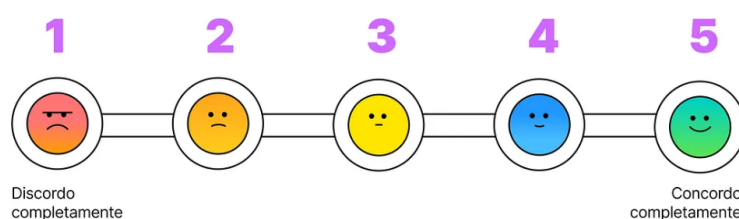


Figura 6. Escala de *Likert* utilizada no SUS.Fonte: Brasil (2024).

As 10 afirmações aplicadas são:

- Eu acho que gostaria de usar esse sistema com frequência.
- Eu acho o sistema desnecessariamente complexo.
- Eu achei o sistema fácil de usar.
- Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
- Eu acho que as várias funções do sistema estão muito bem integradas.
- Eu acho que o sistema apresenta muita inconsistência.
- Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

- Eu achei o sistema atrapalhado de usar.
- Eu me senti confiante ao usar o sistema.
- Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

A pontuação do SUS é calculada da seguinte forma: para os itens ímpares (positivos), subtrai-se 1 da pontuação obtida. Para os itens pares (negativos), subtrai-se a pontuação de 5. A soma total desses valores é então multiplicada por 2,5, resultando em uma pontuação final que varia de 0 a 100. A Figura 7 demonstra a interpretação dessa pontuação, classificando os níveis de aceitabilidade de acordo com a usabilidade percebida.

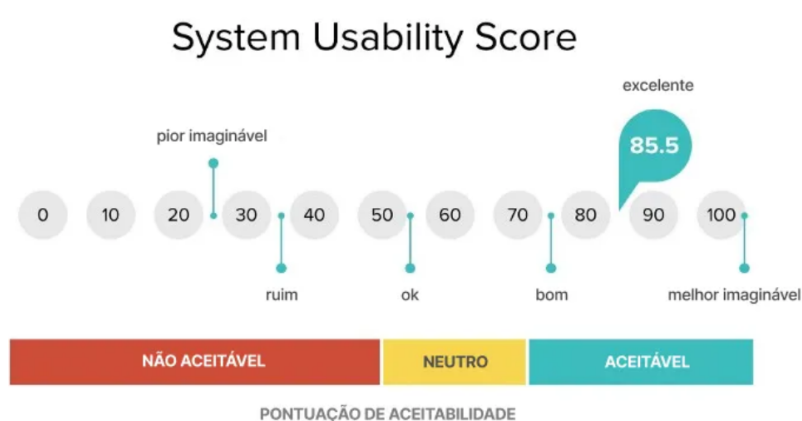


Figura 7. Pontuação final do SUS e classificação de aceitabilidade.
Fonte: Brasil (2024).

2.7. Trabalhos Correlatos

Durante o processo de levantamento e seleção de referências no *Google Acadêmico*, abrangendo o período de 2014 a 2024, foram identificadas cinco publicações relevantes relacionadas ao monitoramento e análise de desempenho esportivo com Inteligência Artificial. Para isso, foram utilizadas expressões como: “sistema de monitoramento esportivo com inteligência artificial”, “análise de desempenho esportivo com métricas de carga de treino”, “aplicação de IA no esporte de alto rendimento”, “sistemas inteligentes para prevenção de lesões em atletas” e “monitoramento de fadiga muscular com inteligência artificial”.

Para avaliar os trabalhos selecionados, elaborou-se uma matriz de pertinência baseada nos critérios propostos por Garrard (2022), considerando aspectos técnicos, funcionais e de usabilidade. Cada estudo foi analisado quanto à presença ou ausência desses critérios, permitindo avaliar sua relevância para o contexto do WIKO. Com base nessa análise, foram destacados os cinco trabalhos mais alinhados aos objetivos do sistema, os quais foram detalhados em dois quadros comparativos que apresentam as características técnicas e funcionais dos sistemas analisados em estudos correlatos.

Ao término da análise, os trabalhos mais alinhados aos objetivos do WIKO foram selecionados como base para a construção da proposta. Diferentemente das soluções revisadas, que em sua maioria se restringem à visualização de dados em *dashboards* ou

relatórios estáticos, o WIKO propõe uma abordagem interpretativa e preditiva, capaz de apoiar diretamente treinadores e atletas.

Nesse contexto, destaca-se o trabalho de McFarland (2014), que propõe uma solução simples para o monitoramento da saúde de atletas, utilizando o *Excel* como interface e banco de dados acessível via *web* e dispositivos móveis. O sistema é centrado no atleta, com registro manual por meio de formulários e planilhas. Sem uso de testes de usabilidade ou inteligência artificial, a abordagem é qualitativa e não opera em tempo real, oferecendo apenas visualizações básicas. Embora prático e acessível, o sistema apresenta limitações em escalabilidade e profundidade analítica.

O trabalho de Conde (2021) utiliza o *Power BI* para visualização e a *Amazon Web Service (AWS)* para armazenamento, focando na geração de relatórios para gestores de Tecnologia da Informação (TI) em ambiente *web*. A usabilidade é avaliada com escala de *Likert*. No entanto, a ausência de Inteligência Artificial reduz o potencial interpretativo da solução, que se limita a visualizações estratégicas, sem interação em tempo real ou com outros perfis de usuários. O sistema não oferece recomendações automáticas, o que restringe sua aplicação em contextos que exigem respostas rápidas e personalizadas.

Goudsmit et al. (2022) apresenta a plataforma *Sport Data Valley*, voltada ao suporte de treinadores via *web* e integrada a sensores para coleta de dados. A usabilidade foi avaliada com testes qualitativos, e o *dashboard*, focado em *feedback*, foi co-desenhado com especialistas para atender ao público-alvo. No entanto, a ausência de Inteligência Artificial limita a solução à visualização e retorno imediato dos dados, sem recomendações interpretativas ou automatizadas que auxiliem treinadores na tomada de decisão.

O trabalho de Juliano et al. (2023) propõe um sistema de monitoramento de saúde via dispositivos móveis, usando o *framework Shiny* e armazenamento no *Google Drive*. Com usabilidade avaliada por questionários qualitativos, o sistema oferece um *dashboard* interativo voltado a treinadores, preparadores físicos e atletas. Apesar da boa portabilidade e interface acessível, a ausência de Inteligência Artificial restringe a solução à visualização estática, sem recursos de interpretação automatizada ou geração de recomendações que auxiliem treinadores e atletas no processo decisório.

O estudo de Rebbouj e Lotfi (2024) propõe uma solução de análise de desempenho atlético com *Azure ML* e *Power BI*, mas apresenta limitações: a flexibilidade dos modelos é restrita pela plataforma fechada, e a interação é limitada a relatórios estáticos, sem adaptação em tempo real. O sistema é voltado apenas para treinadores, sem considerar a interação com atletas, e não há validações de usabilidade. Além disso, o foco permanece na elaboração de relatórios para planejamento de treinos, sem contemplar a prevenção de lesões, a emissão de alertas automáticos ou a oferta de recomendações interpretativas baseadas em evidências.

Para proporcionar uma visão mais clara das diferenças e semelhanças entre os trabalhos correlatos, são apresentados, a seguir, quadros comparativos. Neles, pode-se observar de forma objetiva os principais pontos abordados nos estudos revisados, como metodologias, resultados e tecnologias utilizadas. O Quadro 1 realiza um comparativo técnico entre os sistemas analisados, considerando aspectos como as ferramentas e *frameworks* utilizados, o tipo de armazenamento, a portabilidade, a presença ou ausência de Inteligência Artificial e a avaliação de usabilidade. Nota-se que o sistema proposto se

destaca por adotar tecnologias modernas, ser acessível via *web*, incorporar análise interpretativa com IA por meio de LLMs e apresentar avaliação de usabilidade pelo teste SUS, características que nem sempre estão presentes nos demais sistemas.

Já o Quadro 2 apresenta um comparativo funcional, abordando o foco no usuário, o modelo de interação, o tipo de saída fornecida, a atuação em tempo real e o objetivo principal de cada sistema. No contexto desta análise, “tempo real” refere-se à capacidade do sistema de processar e apresentar dados imediatamente após sua coleta, permitindo respostas rápidas e ajustes ainda durante ou logo após a atividade monitorada. Sistemas que apenas analisam os dados após o encerramento das sessões são classificados como não operando em tempo real. Nessa análise, observa-se que o sistema proposto se diferencia ao atender tanto atletas quanto treinadores, oferecer visualização em tempo real acompanhada de alertas de risco e recomendações interpretativas e ter como objetivo central a otimização do desempenho e a prevenção de lesões. Dessa forma, os quadros reforçam a inovação, aplicabilidade e relevância do WIKO em relação aos estudos correlatos.

Quadro 1. Comparativo técnico entre os sistemas analisados

Autores	Ferramentas/Frameworks	Armazenamento	Portabilidade	IA/Recomendações	Usabilidade
McFarland (2014)	<i>Excel</i>	<i>Excel</i>	<i>Web/Mobile</i>	Não	Não consta
Conde (2021)	<i>Power BI</i>	<i>Amazon Web Service</i>	<i>Web</i>	Não	Escala de <i>Likert</i>
Goudsmit et al. (2022)	<i>Sport Data Valley</i>	Integração com sensores	<i>Web</i>	Não	Testes qualitativos
Juliano et al. (2023)	<i>Shiny</i>	<i>Google Drive</i>	<i>Mobile</i>	Não	Questionários qualitativos
Rebbouj e Lotfi (2024)	<i>Power BI</i>	<i>Azure</i>	<i>Web</i>	Sim (ML)	Não consta
Sistema proposto	<i>Spring Boot e React</i>	<i>MySQL</i>	<i>Web</i>	Sim (LLMs – recomendações)	Teste SUS

Quadro 2. Comparativo funcional dos sistemas em estudos correlatos

Autores	Usuário-Alvo	Modelo de Interação	Tipo de Saída	Tempo Real	Objetivo Principal
McFarland (2014)	Atleta	Formulários e planilhas	Qualitativa	Não	Monitoramento de saúde
Conde (2021)	Gestores de TI	Relatórios e painéis	Visuais	Não	Gestão estratégica
Goudsmit et al. (2022)	Treinador	<i>Dashboard</i> co-desenhado	Visual e <i>feedback</i>	Sim	<i>Feedback</i> para ajustes
Juliano et al. (2023)	Ambos (treinador e atleta)	<i>Dashboard</i> interativo	Visuais	Não	Monitoramento de saúde
Rebbouj e Lotfi (2024)	Treinador	Relatórios e BI	Preditiva	Não	Planejamento de treinos
Sistema proposto	Ambos (treinador e atleta)	<i>Dashboard</i> interativo	Visual, alertas e recomendações	Sim	Otimização do desempenho e prevenção de lesões

3. Especificações do Sistema e Modelagem

Esta seção apresenta as especificações técnicas e a modelagem do WIKO. O objetivo é expor os requisitos funcionais e não funcionais do sistema, detalhar as tecnologias empregadas em seu desenvolvimento e descrever a estrutura arquitetural que sustenta a proposta. Essa etapa estabelece a base para compreender como o sistema foi projetado de forma a atender às necessidades identificadas e implementar as inovações discutidas nas seções anteriores. Além das tecnologias voltadas ao desenvolvimento *web*, esta seção também contempla as bibliotecas e *frameworks* empregados para os módulos de Inteligência Artificial, responsáveis pela análise de dados e geração de recomendações automatizadas.

3.1. Tecnologias Utilizadas e Metodologia de Desenvolvimento

O desenvolvimento do WIKO foi guiado por uma abordagem moderna, baseada na separação de responsabilidades entre as camadas de apresentação, lógica de negócio e persistência de dados. Foram utilizadas tecnologias consolidadas no mercado, compatíveis entre si, permitindo integração entre *front-end*, *back-end* e banco de dados.

A camada de apresentação do sistema foi implementada utilizando o *framework* *React*. A abordagem baseada em componentes favorece a organização do código e o reuso

eficiente de elementos visuais. A interface do WIKO foi projetada para proporcionar uma experiência amigável e intuitiva, com foco em visualização clara de dados e navegação fluida para treinadores e analistas, justificando a adoção desse *framework* (React, 2025).

O *back-end* do sistema foi desenvolvido em *Java*, linguagem orientada a objetos amplamente reconhecida por sua portabilidade, desempenho e segurança. A escolha do *Java* também se deve à sua vasta comunidade, suporte a bibliotecas especializadas e compatibilidade multiplataforma, características que o tornam ideal para sistemas como o WIKO (Oracle Corporation, 2025). Para a construção da aplicação *web* e integração entre os componentes, foi utilizado o *framework Spring Boot*, baseado no ecossistema *Spring*. Essa tecnologia simplifica a configuração e o desenvolvimento de aplicações *Java* ao oferecer uma estrutura padronizada, com recursos como injeção de dependência, inicialização automatizada, segurança embutida e suporte nativo a APIs REST. Além disso, destaca-se pela integração facilitada com bancos de dados e serviços externos, bem como pela arquitetura modular e de fácil manutenção (VMware, Inc., 2025).

Para a persistência dos dados, optou-se pelo Sistema Gerenciador de Banco de Dados (SGBD) relacional *MySQL*. Sua estabilidade, desempenho e integração nativa com o *Spring Boot* foram fatores decisivos na escolha. A estrutura relacional do banco facilita a modelagem das entidades do sistema, como usuários, atletas, sessões de treino e métricas de desempenho.

Durante o desenvolvimento, foram empregadas ferramentas de apoio que contribuíram para a organização e eficiência do processo. O *Trello* foi utilizado como ferramenta de gerenciamento de tarefas no estilo *Kanban*, permitindo o acompanhamento das etapas do projeto segundo a metodologia ágil (Atlassian, 2025). O *Figma* foi adotado para a prototipação das interfaces, priorizando usabilidade e clareza visual, o que possibilitou testes preliminares e refinamento do *design* com base em *feedback* dos usuários (Figma, 2024). Por fim, o *Swagger* foi utilizado para documentar e testar as APIs REST do sistema, facilitando a visualização das rotas e a execução de requisições diretamente pela interface, o que otimizou o processo de integração entre *front-end* e *back-end* (SmartBear Software, 2025).

O *Git* foi utilizado como sistema de controle de versão, permitindo registrar e gerenciar alterações no código-fonte do projeto, além de facilitar o trabalho colaborativo entre desenvolvedores. O *GitHub*, por sua vez, serviu como plataforma de hospedagem dos repositórios *Git*, oferecendo recursos para compartilhamento, revisão e colaboração em equipe, fundamentais para o desenvolvimento organizado e versionado do sistema (Alura, 2020).

A condução do desenvolvimento foi orientada pela metodologia ágil *Scrum*, que favorece entregas incrementais e ciclos de *feedback* rápidos. O projeto foi estruturado em *sprints* de duração definida, cada uma com metas claras de implementação. Reuniões semanais e revisões periódicas possibilitaram ajustes de escopo e priorização de tarefas, garantindo maior controle sobre a evolução do sistema e promovendo adaptação contínua ao contexto acadêmico e às descobertas do processo de pesquisa (Scrum, 2025).

3.2. Bibliotecas e Frameworks para Inteligência Artificial

Na camada de Inteligência Artificial, o WIKO não utiliza modelos preditivos tradicionais de *machine learning*, mas combina o cálculo de métricas de carga validadas na literatura

(*PlayerLoad*, *sRPE*, *ACWR*, *monotony* e *strain*) com a integração a LLMs para geração de recomendações interpretativas.

O processamento inicial é realizado no *back-end* em *Java*, com o apoio de bibliotecas voltadas a operações matemáticas e manipulação de dados. Em seguida, os resultados são enviados ao modelo *Groq*, desenvolvido pela xAI, que atua como camada de interpretação e recomendação. Essa abordagem permite transformar os cálculos em orientações práticas (Tanzu, 2025a).

Essa integração ocorre por meio de chamadas REST, nas quais os resultados das métricas calculadas no *back-end* são estruturados em JSON e transmitidos para a API do *Groq*. O modelo processa essas informações e retorna recomendações textuais interpretativas, que são armazenadas e disponibilizadas no sistema para técnicos e atletas.

A integração entre o *Spring AI* e os modelos no *Groq*, representa o núcleo da camada de Inteligência Artificial do WIKO. Esse mecanismo simplifica a comunicação entre aplicações *Java/Spring Boot* e provedores de LLMs, permitindo que as recomendações sejam geradas de forma automática e contextualizada.

Entre as principais bibliotecas utilizadas no projeto, destacam-se:

- ***Apache Commons Math***: biblioteca de código aberto aplicada nos cálculos estatísticos e no processamento das métricas de carga.
- ***Groq API***: interface utilizada para enviar as métricas esportivas estruturadas em formato JSON, por meio de chamadas REST, retornando recomendações em linguagem natural contextualizadas ao desempenho esportivo.

Dessa forma, a inteligência artificial no WIKO é construída sobre bases científicas sólidas, mas ganha poder interpretativo ao integrar-se ao *Groq*, pois traduz métricas complexas em recomendações explicativas e fundamentadas em literatura científica, ampliando a acessibilidade e a utilidade prática para atletas e treinadores.

3.3. Levantamento de Requisitos

O levantamento de requisitos do sistema WIKO foi realizado com base na análise de práticas já validadas no contexto da tecnologia aplicada ao esporte. Esse processo permitiu identificar e estruturar os requisitos do sistema com base em fundamentos teóricos e aplicações práticas testadas por especialistas da área.

Nesta seção, os requisitos estão organizados em três categorias principais. A primeira categoria contempla os requisitos funcionais, que descrevem os comportamentos e funcionalidades esperadas do sistema. A segunda refere-se aos requisitos não funcionais, que definem as restrições e qualidades técnicas que devem ser observadas no desenvolvimento, como desempenho e segurança. Por fim, são apresentadas as regras de negócio, que representam diretrizes e políticas específicas que regem o funcionamento da plataforma.

3.3.1. Requisitos Funcionais

Esta subseção apresenta os requisitos funcionais do sistema que, segundo Sommerville (2011), definem as funcionalidades essenciais que a plataforma deve oferecer para atender

às necessidades dos usuários. Esses requisitos foram definidos com base nos principais processos relacionados ao gerenciamento de atletas, treinadores e registros de atividades esportivas.

- **RF01 – Cadastro de atletas:** o sistema deve permitir o cadastro de atletas, incluindo dados pessoais (nome, data de nascimento, sexo, modalidade esportiva, clube pertencente etc.) e fisiológicos (altura, peso etc.).
- **RF02 – Cadastro de treinadores:** o sistema deve permitir o cadastro de treinadores, incluindo dados pessoais (nome, data de nascimento, sexo, modalidade esportiva, clube pertencente etc.).
- **RF03 – Registro de dados de treinos/jogo:** o sistema deve permitir o registro de sessões de treino/jogo, com data/hora, tipo de exercício, duração e métricas coletadas (distância percorrida, acelerações, *sprints*, zonas de velocidade etc.). Os dados poderão ser inseridos manualmente ou importados por planilhas eletrônicas/arquivos CSV.
- **RF04 – Identificação de padrões de carga:** o sistema deve analisar os dados e indicar se as cargas estão dentro, acima ou abaixo das faixas de referência bibliográficas, permitindo acompanhar a evolução dos atletas ao longo do tempo.
- **RF05 – Geração de alertas a partir de métricas:** o sistema deve identificar variações críticas em indicadores de carga (exemplo: ACWR elevado, *monotony* alta, *strain* excessivo) e emitir alertas automáticos.
- **RF06 – Interpretação das métricas:** o sistema deve integrar os resultados calculados com Modelos de Linguagem de Grande Escala (LLMs), gerando recomendações interpretativas e contextualizadas para atletas e treinadores.
- **RF07 – Alerta de possível risco de lesão:** o sistema deve alertar automaticamente quando os indicadores de carga (ACWR, *monotony*, *strain*) atingirem zonas de risco reconhecidas na literatura.
- **RF08 – Geração de relatórios de desempenho:** o sistema deve gerar relatórios exportáveis (em PDF), reunindo as métricas calculadas e as recomendações fornecidas pela camada de IA.
- **RF09 – Suporte a múltiplos perfis de usuário (atleta e treinador):** o sistema deve permitir acessos com permissões diferenciadas, onde o treinador pode visualizar e comparar múltiplos atletas, enquanto o atleta acessa apenas seus próprios dados.
- **RF10 – Questionário psicoemocional:** o sistema deve disponibilizar, antes de cada sessão, um questionário para que os atletas informem seu estado emocional, nível de estresse, qualidade do sono, fadiga e humor. As respostas serão utilizadas para análise de risco de lesão e bem-estar geral, sendo integradas ao painel do treinador.
- **RF11 – Gamificação:** o sistema deve implementar um mecanismo de gamificação com base em métricas de desempenho. Cada atleta receberá uma pontuação em relação a seu desempenho em treinos e jogo, que será atualizada periodicamente, permitindo a visualização de um *ranking* com os melhores desempenhos da equipe.

3.3.2. Requisitos não Funcionais

Os requisitos não funcionais, conforme descrito por Sommerville (2011), dizem respeito às qualidades do sistema que não estão diretamente ligadas às suas funcionalidades, mas são fundamentais para garantir seu bom desempenho, segurança, usabilidade, disponibilidade e manutenibilidade. A seguir, são apresentados os principais requisitos não funcionais identificados para o WIKO.

- **RNF01 – Acessibilidade *web*:** o sistema deve ser acessível via navegador *web* compatível com os principais navegadores (*Chrome, Firefox, Safari*).
- **RNF02 – Conformidade com a LGPD:** o sistema deve armazenar dados de forma segura, respeitando a LGPD.
- **RNF03 – Manutenibilidade:** o sistema deve ser estruturado de forma modular, facilitando a manutenção, atualizações futuras e correção de falhas sem a necessidade de grandes reestruturações.
- **RNF04 – Usabilidade:** o sistema deve apresentar uma interface gráfica intuitiva, com boa organização visual e facilidade de navegação, garantindo uma experiência acessível mesmo para usuários sem conhecimentos técnicos.
- **RNF05 – Visualização dos dados em interface gráfica intuitiva:** o sistema deve apresentar os dados e os resultados das análises em *dashboards* acessíveis, com gráficos, indicadores e alertas que facilitem a interpretação por atletas e treinadores.

3.3.3. Regras de Negócio

As regras de negócio, segundo Sommerville (2011), representam as diretrizes que definem como o sistema deve se comportar em determinadas situações, de acordo com as políticas e objetivos específicos da organização. Essas regras são fundamentais para garantir a coerência das operações e o alinhamento com os processos institucionais. Abaixo, são descritas as principais regras de negócio do sistema WIKO.

- **RN01 – Alerta por ACWR:** quando o valor da razão carga aguda:crônica (ACWR) ultrapassar 1,5, o sistema deve emitir alerta de risco de sobrecarga. Valores muito abaixo de 0,8 também devem gerar alerta por possível perda de condição física.
- **RN02 – Alerta por *monotony*:** quando a *monotony* semanal superar o valor de 2,0, o sistema deve emitir alerta de risco de fadiga e recomendar maior variação nos estímulos de treino.
- **RN03 – Alerta por *strain*:** quando o valor de *strain* ultrapassar limiares definidos pelo treinador ou pela literatura de referência, o sistema deve indicar a necessidade de estratégias de recuperação.
- **RN04 – Validação de cadastro:** o sistema deve validar que cada atleta cadastrado possua dados completos (nome, data de nascimento, sexo, modalidade esportiva, clube pertencente etc.) antes de permitir o registro de sessões de treino.
- **RN05 – Relatório para atletas:** o sistema só deverá exportar relatórios para atletas com, no mínimo, cinco sessões registradas, garantindo consistência estatística mínima para análise.

3.4. Modelagem da Solução

Esta seção apresenta a modelagem da solução desenvolvida para o WIKO, com o objetivo de representar de forma visual e estruturada os principais elementos que compõem a plataforma. A modelagem inclui o protótipo da interface gráfica, os Diagramas de Casos de Uso e o Diagrama de Entidade-Relacionamento (DER). Essas representações foram elaboradas com base nos requisitos definidos previamente e visam garantir consistência, usabilidade e coerência técnica na construção do sistema.

3.4.1. Protótipo da Interface da Plataforma

Esta seção apresenta os fluxos de tela do treinador e do atleta, junto com os protótipos das telas da plataforma WIKO, ilustrando as principais funcionalidades e interações para cada tipo de usuário.

No Apêndice A.2, a Figura 13 apresenta os fluxos de navegação para os dois perfis de usuário da aplicação: *Treinador* e *Atleta*. O fluxo do treinador inicia na tela principal, com opções de *login* e cadastro. Após o acesso, o treinador é direcionado para a tela inicial, onde pode editar seu perfil, acessar sessões (com possibilidade de visualização individual), cadastrar novas sessões, e gerenciar atletas (incluindo a visualização e edição de informações específicas de cada um).

O fluxo do atleta é mais simplificado. Após o *login* na tela principal, o usuário é levado à sua página inicial, que permite editar o perfil, visualizar suas sessões (com acesso ao detalhamento de cada uma) e consultar informações sobre seu time. Esse fluxo reflete a diferenciação de permissões entre os perfis, onde o treinador possui acesso ampliado às funcionalidades de gestão e análise, enquanto o atleta tem acesso restrito aos seus próprios dados.

O *dashboard* principal do treinador, ilustrado na Figura 14 do Apêndice A.2, consolida as informações da equipe em um painel unificado. Esta interface apresenta estatísticas gerais, *ranking* de desempenho, alertas inteligentes sobre variações métricas significativas e avaliações psicoemocionais dos atletas, permitindo tomadas de decisão ágeis baseadas em dados centralizados.

A interface de criação de sessões (Figura 15, Apêndice A.2) permite ao treinador cadastrar um novo treino ou jogo. O formulário possibilita definir o tipo da sessão, título, data, número de atletas envolvidos, placar (caso aplicável) e observações. Após o cadastro, a sessão é listada junto das outras sessões, onde poderá ser preenchida com dados manualmente ou por meio de importação via planilha.

Para análise detalhada, a Figura 16 (Apêndice A.2) apresenta a interface de visualização de sessão específica. Ao selecionar uma sessão, são listados todos os atletas participantes no painel lateral. Quando um atleta é escolhido, são exibidos gráficos com métricas individuais de desempenho, como distância percorrida, velocidade máxima e número de acelerações. Além disso, a interface permite aplicar uma análise com IA, exportar o relatório da sessão em PDF ou comparar o desempenho com o de outro atleta, oferecendo suporte à tomada de decisão do treinador.

Voltando ao usuário atleta, a tela inicial representada na Figura 17 (Apêndice A.2)

concentra-se em informações personalizadas. O *layout* exibe dados físicos do usuário, sua pontuação atual e posição no *ranking* da equipe a partir do seu desempenho, além da sua sessão mais recente e o próximo jogo agendado. Um calendário destaca os dias de atividades e permite acesso rápido aos compromissos futuros. A navegação lateral oferece atalhos para sessões passadas, informações do time e configurações da conta.

3.4.2. Diagrama de Casos de Uso

Esta subseção apresenta o Diagrama de Casos de Uso do WIKO, que representa as principais funcionalidades disponibilizadas para os dois perfis de usuários: *Treinador* e *Atleta*. A Figura 8 ilustra essas interações, evidenciando o papel central do treinador na gestão do sistema, enquanto o atleta possui acesso a funcionalidades mais restritas. O diagrama também destaca as dependências entre os casos de uso, permitindo uma visualização clara das interações entre os usuários e o sistema.

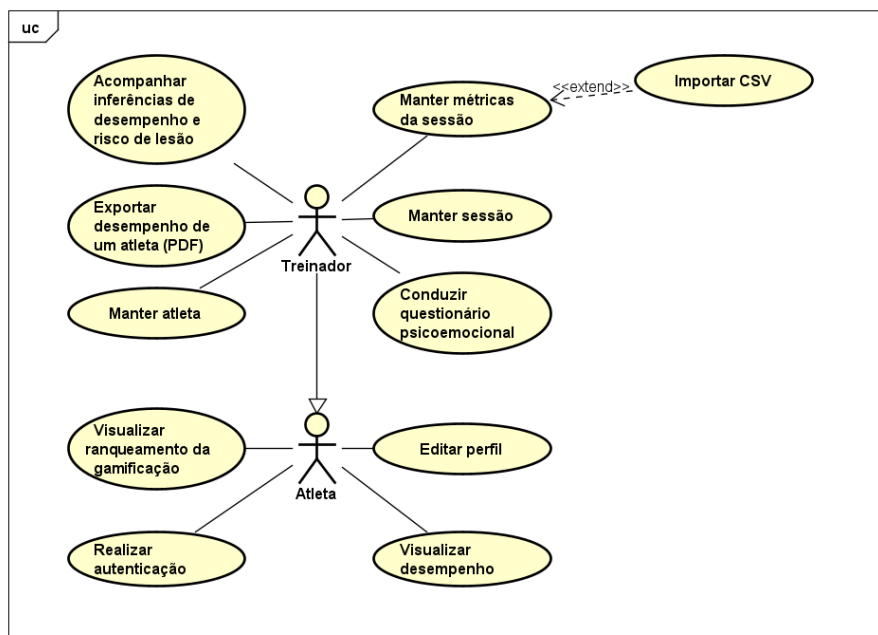


Figura 8. Diagrama de Casos de Uso.

3.4.3. Diagrama de Entidade-Relacionamento

O Diagrama de Entidade-Relacionamento (DER) do sistema WIKO foi elaborado com o objetivo de representar, de forma estruturada e lógica, a organização dos dados e seus relacionamentos no SGBD relacional *MySQL*. A Figura 9 ilustra a modelagem e reflete os requisitos funcionais levantados, buscando garantir a integridade, escalabilidade e flexibilidade da base de dados, de forma a atender às necessidades de monitoramento, análise de desempenho e apoio à tomada de decisão no contexto esportivo.

O modelo contempla múltiplas entidades principais, com destaque para *user*, *coach*, *athlete*, *session* e *metrics*. A entidade *user* centraliza as informações de acesso ao sis-

tema e diferencia os perfis por meio do atributo *user_type*. O *coach* e *athlete* estão vinculados a usuários e organizam a estrutura de treinadores e atletas cadastrados. Informações adicionais de atletas, como altura, peso, composição corporal e estado de lesão, são armazenadas separadamente na entidade *athlete_extra*, garantindo normalização adequada.

A entidade *session* representa treinos ou jogos, vinculados a um treinador, e se relaciona com os atletas por meio de *metrics*, que armazena os dados coletados em campo (distância, velocidade, acelerações, *sprints* etc.). A partir dessas métricas, são gerados registros em *session_loads*, que consolidam o cálculo das cargas (*PlayerLoad*, sRPE, ACWR, *monotony* e *strain*). Situações críticas de carga são registradas em *training_load_alert*, com rótulos interpretativos que permitem sinalizar zonas de risco.

O modelo também contempla o acompanhamento subjetivo por meio da entidade *psycho_form_answer*, que reúne respostas a questionários sobre percepção de esforço, energia, fadiga, sono, humor e dor muscular. Esses dados alimentam a tabela *psycho_risk_score*, que classifica o nível de risco (*low*, *moderate* ou *high*), podendo gerar notificações específicas na entidade *psycho_alert*. A gestão do envio de formulários é controlada pela entidade *psycho_form_invite*.

Por fim, o sistema registra recomendações automáticas em duas entidades complementares: *perf_recommendations*, voltada às recomendações baseadas em métricas de carga, e *psy_recommendations*, destinada às recomendações derivadas de dados psicométricos. Ambas armazenam o texto gerado pelo modelo de linguagem *Groq*, juntamente com os parâmetros de entrada considerados.

Esse modelo evidencia a integração entre dados objetivos de carga e dados subjetivos de bem-estar, estruturando a base necessária para que o WIKO ofereça monitoramento, alertas e recomendações personalizadas a atletas e treinadores.

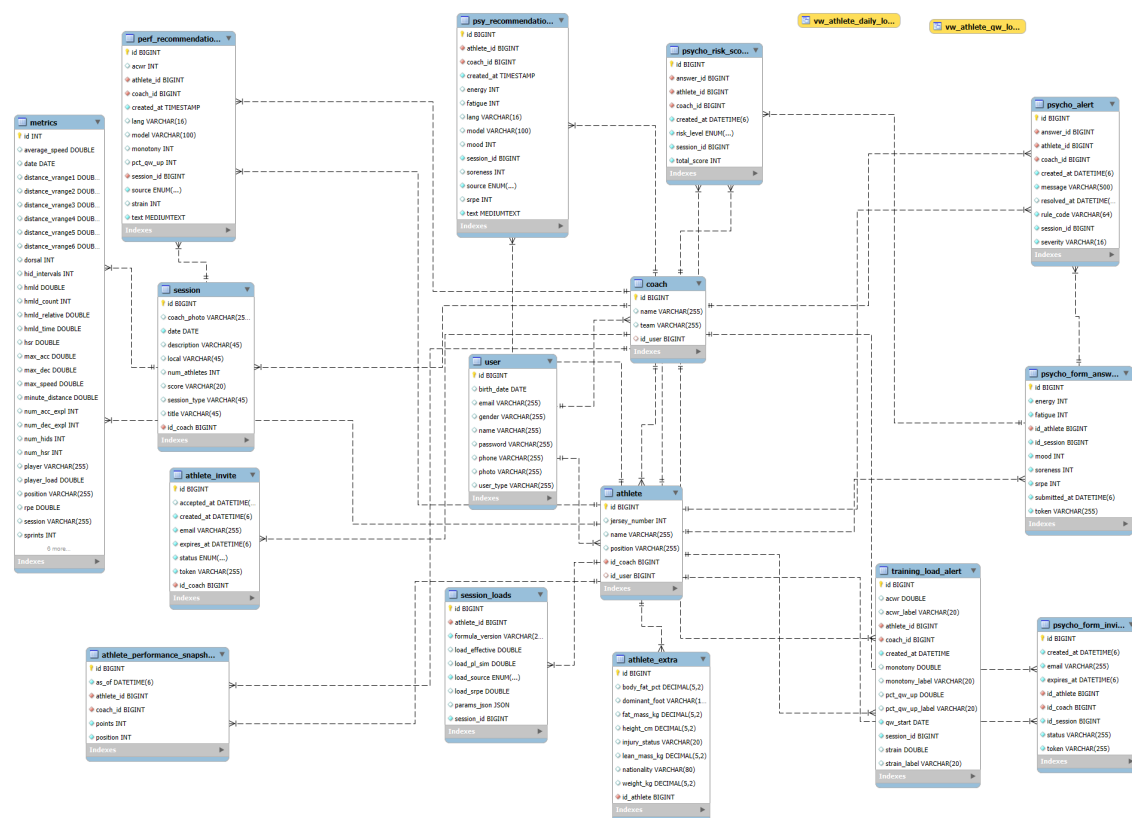


Figura 9. Diagrama de Entidade-Relacionamento.

3.5. Arquitetura do Software

A Figura 10 apresenta a arquitetura da aplicação, que adota uma abordagem *full-stack* composta por *React* no *front-end* e *Spring Boot* no *back-end*. No lado do cliente, a estrutura é organizada em páginas/visões, roteamento (*router*), componentes reutilizáveis, autenticação e um módulo de serviços responsável por realizar requisições HTTP utilizando a biblioteca *Axios*.

No *back-end*, o *Spring Boot* é estruturado em camadas: a camada de controle (*controller layer*), responsável por receber e encaminhar as requisições; a camada de serviços (*service layer*), onde se concentra a lógica de negócio; e a camada de repositórios (*repository layer*), que realiza o acesso ao banco de dados. Além disso, há um módulo responsável pelo cálculo das métricas de carga (*PlayerLoad*, *sRPE*, *ACWR*, *monotony* e *strain*), cujos resultados podem ser enviados ao modelo de linguagem *Groq* (xAI) para geração de recomendações interpretativas. A integração com a LLM do *Groq* é realizada por meio do *Spring AI*, que simplifica a comunicação com modelos de linguagem generativos.

A comunicação entre cliente e servidor ocorre via requisições HTTP, garantindo a separação clara entre interface e processamento, enquanto a camada de integração com LLMs amplia a capacidade analítica do sistema.

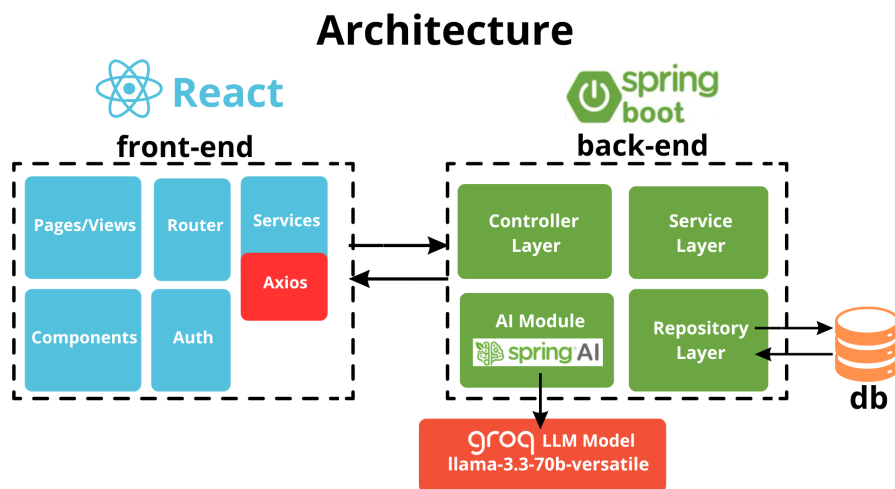


Figura 10. Arquitetura do *software*.

4. Desenvolvimento e Resultados da Plataforma

O desenvolvimento da plataforma WIKO foi conduzido de forma iterativa e incremental, seguindo os princípios do *Scrum* e com o apoio da ferramenta *Trello* para o gerenciamento do *backlog* e organização das entregas. Essa abordagem ágil permitiu priorizar funcionalidades críticas, validar resultados parciais com os usuários-alvo e ajustar o escopo conforme o progresso do projeto. Para garantir acessibilidade e desempenho, a arquitetura de implantação foi distribuída: o *front-end* foi hospedado na plataforma *Vercel*⁶, o *back-end* foi implantado no serviço *Render* e o banco de dados gerenciado através do *Aiven*. Os resultados foram validados tanto por profissionais da área da saúde quanto por usuários finais. Os especialistas aprovaram as saídas do sistema, destacando sua utilidade, fundamentação técnica e aderência aos padrões esperados. Os usuários, por sua vez, corroboraram a praticidade e a usabilidade da ferramenta no contexto de aplicação pretendido.

Por fim, em vez de adotar métricas tradicionais de acurácia de redes neurais, a avaliação do sistema considerou três dimensões centrais: (i) a qualidade das recomendações produzidas pelos modelos de linguagem, em termos de clareza, alinhamento à literatura e relevância para treinadores; (ii) a usabilidade da interface, avaliada por meio do SUS; e (iii) a confiabilidade percebida pelos usuários em relação às recomendações emitidas pela ferramenta.

4.1. Engenharia de *Features* para Cálculo de Cargas

Uma das etapas centrais no desenvolvimento do sistema foi a transformação dos dados brutos das sessões de treino e jogo em métricas quantitativas consistentes com a literatura científica. Para garantir a independência das *features*, realizou-se uma análise de correlação de Pearson entre os atributos numéricos (Apêndice A.1, Figura 12), que confirmou relações esperadas entre métricas correlatas, como distância total e distâncias em zonas de velocidade. Esse processo, conhecido como engenharia de *features*, permitiu

⁶A versão de produção da plataforma está disponível em: <https://simada-frontend.vercel.app/>.

traduzir informações coletadas em indicadores de carga, desempenho e risco de lesão, que servem como base para as demais funcionalidades da plataforma.

4.1.1. Transformação de Dados Brutos em Métricas

As métricas utilizadas no sistema já foram detalhadas na Seção 2 e listadas integralmente no Apêndice A.1. Nesta etapa do desenvolvimento, o foco esteve em implementar computacionalmente essas métricas a partir dos dados brutos coletados nas planilhas de monitoramento.

Para o cálculo das cargas, destacam-se três variáveis principais: a percepção subjetiva de esforço (*Rate of Perceived Exertion* – RPE), a duração da sessão (*time*) e as medidas de distância, acelerações e desacelerações. A partir delas, foram operacionalizadas duas abordagens: (i) a carga interna, obtida pelo método sRPE, e (ii) a carga externa, representada pelo *PlayerLoad* simulado, definido como combinação ponderada de distância, acelerações e *sprints*.

Essas cargas foram então utilizadas como base para o cálculo de métricas já validadas na literatura, como o *Acute:Chronic Workload Ratio* (ACWR), a *monotony* e o *strain*, garantindo que os indicadores implementados no sistema estivessem alinhados com referências científicas consolidadas e pudessem sustentar análises de desempenho e recomendações personalizadas.

4.1.2. Engenharia de *Features*

Após a transformação dos dados brutos em métricas fundamentais de carga, foi realizada a etapa de engenharia de *features*, cujo objetivo foi estruturar essas métricas em indicadores temporais capazes de refletir a dinâmica de esforço e recuperação dos atletas. Esse processo envolveu operações de agregação, cálculo de médias móveis e organização das informações em diferentes janelas de tempo.

Em um primeiro nível, foram obtidas as cargas diárias, calculadas individualmente para cada atleta e sessão. Em seguida, essas cargas foram agregadas em cargas semanais, permitindo avaliar tendências de variação no curto prazo. Para a estimativa de risco de fadiga e lesão, foram aplicadas janelas móveis, de modo a distinguir entre carga aguda (últimos 7 dias) e carga crônica (média de 21 a 28 dias). A partir dessa distinção, foi possível calcular o indicador *Acute:Chronic Workload Ratio* (ACWR).

Além disso, foram implementadas métricas complementares, como a *monotony*, que expressa a variação intra-semanal da carga, e o *strain*, que representa a combinação entre carga total e *monotony*. Essas derivadas permitem interpretar não apenas o volume de treino, mas também sua distribuição e impacto cumulativo.

Esse conjunto de *features* fornece uma visão integrada do estado de cada atleta, servindo como base para a geração de relatórios, identificação de zonas de risco e posterior integração com o módulo de recomendações baseadas em modelos de linguagem.

4.1.3. Interpretação dos Resultados

Os resultados obtidos a partir da engenharia de *features* permitiram transformar informações brutas de monitoramento em indicadores diretamente aplicáveis ao acompanhamento do desempenho esportivo. Na prática, métricas como o ACWR, a *monotony* e o *strain* foram utilizadas para classificar situações de risco e fornecer *labels* interpretativos (por exemplo: equilíbrio, atenção, alto risco), os quais foram exibidos de forma visual no painel do sistema.

Essa interpretação prática é essencial para aproximar os cálculos científicos do cotidiano de treinadores e atletas. Ao integrar esses resultados nos relatórios e *dashboards* da plataforma, o sistema não apenas quantifica o esforço físico, mas também fornece uma base objetiva para decisões sobre ajuste de treinos, prevenção de lesões e manejo da fadiga.

4.2. *Front-end*: Telas e Usabilidade

O *front-end* do WIKO foi desenvolvido como uma aplicação em *React + TypeScript*, com o uso do *Material UI* (MUI) para componentes visuais e *Recharts* para a visualização de dados esportivos. A comunicação com o *back-end* ocorre via serviços REST, encapsulados em um *wrapper Axios* com interceptadores de autenticação *JSON Web Token* (JWT). Esta subseção apresenta as principais telas implementadas, destacando como elas foram projetadas para atender às necessidades dos usuários⁷.

O *Dashboard* do *coach* reúne as informações de todos os atletas sob responsabilidade do treinador, exibindo gráficos de métricas, carga calculada e combinações de indicadores. Essa visão integrada favorece o acompanhamento e a tomada de decisão, conforme ilustrado na Figura 18 do Apêndice A.3.

A Implementação 1 exemplifica a construção do componente responsável por renderizar gráficos de desempenho utilizando a biblioteca *Recharts*. Nesse caso, o gráfico de barras exibe a distância total percorrida por atleta em cada sessão, com suporte a redimensionamento automático, tema dinâmico (claro/escuro) e dicas interativas (*tooltips*). O componente *ChartCard* agrega título, informações descritivas e estilização personalizada via MUI, tornando a visualização responsiva e acessível em diferentes tamanhos de tela. Essa implementação foi essencial para possibilitar a leitura rápida de padrões de carga e identificação de variações atípicas no grupo de atletas.

⁷Código-fonte do *front-end* disponível em: <https://github.com/NathanWMartins/simada-frontend>.

```

1 ChartCard
2   title="Total Distance per Athlete (km) "
3   info="Total distance covered by each athlete in this session. Use it
4     to spot workload outliers and how volume is distributed across
5     the squad."
6
7   Box sx= width: "100%", overflowX: "auto", overflowY: "hidden"
8     Box sx= minWidth: 1000
9       ResponsiveContainer width="100%" height=270
10        BarChart data=data.distanceByAthlete margin= top: 10, right:
11          20, left: 0, bottom: 0
12          CartesianGrid strokeDasharray="3 3" /
13          XAxis dataKey="athlete" interval=0 tick= fontSize: 12
14            tickMargin=10 /
15          YAxis /
16          RTooltip
17            contentStyle=
18              backgroundColor: isDark ? theme.palette.background.paper
19                : "#fff",
20              color: theme.palette.text.primary,
21              border: `1px solid $theme.palette.divider`,
22              borderRadius: 8,
23              boxShadow: isDark ? "0 6px 18px rgba(0,0,0,.5)" : "0 6px
24                18px rgba(0,0,0,.12)",
25
26          labelStyle= color: GREEN
27          itemStyle= color: theme.palette.text.primary
28          formatter=(value: number, name: string) =>
29            [value.toFixed(2), name]
30
31          /
32          Legend /
33          Bar dataKey="km" name="km" fill=GREEN maxBarSize=48 /
34        /BarChart
35      /ResponsiveContainer
36    /Box
37  /Box
38 /ChartCard

```

Implementação 1. *Front-end* — Geração de gráfico (*Recharts*) para distância total por atleta na sessão.

A tela inicial (*Home*) do *coach* apresenta informações gerais como quantidade de sessões, treinos da semana, atletas cadastrados, entre outras. Possui a gamificação com os três melhores ranqueados e prioriza a exibição de dois alertas principais: o de *training load*, relacionado a risco de lesão, fadiga e desempenho, e o psicoemocional. A interface destaca a gravidade de cada alerta, permitindo que o treinador selecione o atleta em questão e solicite à IA uma recomendação sobre como proceder, conforme ilustrado na Figura 11.

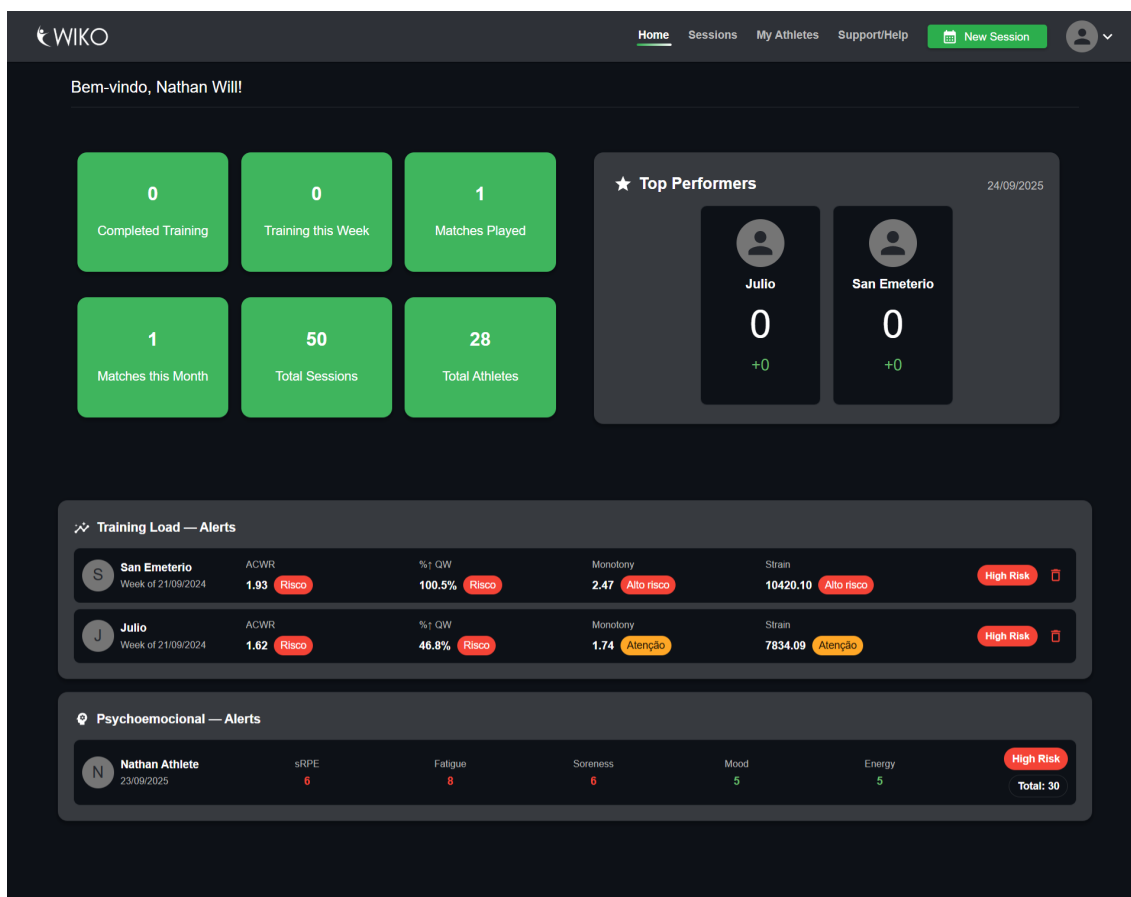


Figura 11. Tela inicial (Home) do coach, com resumo geral, ranqueamento de atletas e alertas principais.

O relatório em PDF é gerado a partir das recomendações produzidas pelo modelo de linguagem (*Groq*, via *Spring AI*). O documento contém informações individuais do atleta, seus *labels* de risco (*ACWR*, *monotony*, *strain* e psicoemocionais), a zona de risco atribuída a cada métrica e uma recomendação textual personalizada. Esse recurso garante a portabilidade e a reprodutibilidade das informações, permitindo que treinadores compartilhem relatórios com a comissão técnica ou arquivem para acompanhamento longitudinal (Implementação 3), conforme ilustrado na Figura 19 do Apêndice A.3.

A Implementação 3 apresenta o trecho de código responsável pela exportação dos relatórios em formato PDF, desenvolvido com as bibliotecas *jsPDF* e *autoTable*. O código organiza as métricas e rótulos de risco em uma tabela formatada e insere, ao final, o texto de recomendações gerado pela IA. O relatório é personalizado com o nome do atleta e salvo automaticamente com formatação padronizada, assegurando clareza e consistência visual nos documentos gerados. Essa automação possibilita que o treinador gere e compartilhe relatórios de maneira rápida, sem depender de ferramentas externas.

A tela de respostas do questionário psicoemocional exibe os dados de uma sessão específica. Nela são mostradas as médias dos indicadores reportados pelos atletas (*sRPE*, fadiga, dor muscular, humor e energia), permitindo ao treinador identificar rapidamente o estado geral da equipe. A interface possibilita ainda o filtro das respostas por posição de atleta e a exportação dos resultados em formato CSV, favorecendo análises comparativas

e armazenamento externo, conforme ilustrado na Figura 20 do Apêndice A.3.

Por fim, a tela principal do atleta foi projetada para engajar o usuário no acompanhamento do próprio desempenho. Nela, o atleta encontra seu perfil, o calendário de sessões passadas e futuras (treinos e jogos), além de sua pontuação na gamificação, um recurso que promove motivação e adesão ao monitoramento. A partir dessa tela, é possível acessar o detalhamento de cada sessão, conforme ilustrado na Figura 21 do Apêndice A.3.

4.3. *Back-end: Endpoints e Serviços*

O *back-end* do WIKO foi desenvolvido em *Java 17* com *Spring Boot*, organizado em camadas para separar responsabilidades e facilitar a manutenção. A estrutura contempla pacotes principais como `controller`, responsável pelos *endpoints* REST; `service`, que concentra as regras de negócio; `repository`, responsável pela persistência de dados; e `model`, que define as entidades do domínio. Essa arquitetura garante clareza no fluxo de dados e escalabilidade para futuras evoluções do sistema⁸.

Entre os *endpoints* implementados, destacam-se os de registro de usuários e os voltados à análise de desempenho e recomendações. O *endpoint* `POST /register/coach` permite cadastrar novos treinadores, enquanto o `POST /register/athlete` realiza o registro de atletas vinculados a um treinador.

Para a camada de recomendações, foram criados dois *endpoints*: o `POST /psy-form/{sessionId}/athletes/{athleteId}/recommendations`, que processa os formulários psicoemocionais, e o `POST /performance/{sessionId}/athletes/{athleteId}/recommendations`, que gera recomendações de desempenho físico. Ambos utilizam a integração com modelos de linguagem para traduzir métricas em orientações práticas.

O *endpoint* `GET /{sessionId}/metrics` disponibiliza as métricas de uma sessão específica, enquanto o `GET /{athleteId}/home/calendar` retorna o calendário de sessões (treinos e jogos) do atleta. Já o `GET /{athleteId}/home/performance` fornece indicadores de gamificação.

Além dos *endpoints* de registro e consulta, destacam-se duas partes centrais do desenvolvimento do *back-end*. A primeira corresponde ao cálculo das métricas derivadas de carga de treino (ACWR, variação semanal, *monotony* e *strain*), fundamentais para a análise do risco de lesão, conforme apresentado na Implementação 2. A segunda é a construção dinâmica dos *prompts* enviados ao modelo de linguagem, que organizam as métricas em um formato compreensível para a IA. As Implementações 4 e 5 do Apêndice A.5 ilustram, respectivamente, o fluxo de geração de recomendações psicoemocionais e de desempenho físico.

```
1 for (Long aid : distinctAthleteIds) {
2     LocalDate latest =
3         weeklyLoadQueryRepository.findLatestQwStart(aid);
4         //... verifica metricas do atleta
5
6     var rows = weeklyLoadQueryRepository.qwWindow(aid, latest, latest);
7     if (rows == null || rows.isEmpty()) continue;
```

⁸Código-fonte do *back-end* disponível em: <https://github.com/KauanKoech/simada-backend>.

```

7
8     var r = rows.get (rows.size () - 1);
9
10    String acwrL = Labels.acwrLabel (toDouble (r.getAcwr ()));
11    String pctQwUpL = Labels.pctQwUpLabel (toDouble (r.getPctQwUp ()));
12    String monoL = Labels.monotonyLabel (toDouble (r.getMonotony ()));
13    String strainL = Labels.strainLabel (toDouble (r.getStrain ()));
14    boolean hasAlert =
15        isAttentionOrRisk (acwrL) ||
16        isAttentionOrRisk (pctQwUpL) ||
17        isAttentionOrRisk (monoL) ||
18        isAttentionOrRisk (strainL);
19
20    if (!hasAlert) continue;
21
22    boolean exists = trainingLoadAlertRepository
23        .findByAthleteIdAndSessionId (aid, sessionIdLong)
24        .isPresent ();
25    if (exists) continue;
26    //... verificacao de treinador e atleta
27    var alert = TrainingLoadAlert.builder ()
28        .athlete (athlete)
29        .coach (coach)
30        .session (session)
31        .qwStart (r.getQwStart ())
32        .acwr (toDouble (r.getAcwr ()))
33        .acwrLabel (acwrL)
34        .pctQwUp (toDouble (r.getPctQwUp ()))
35        .pctQwUpLabel (pctQwUpL)
36        .monotony (toDouble (r.getMonotony ()))
37        .monotonyLabel (monoL)
38        .strain (toDouble (r.getStrain ()))
39        .strainLabel (strainL)
40        .createdAt (Instant.now ())
41        .build ();
42    trainingLoadAlertRepository.save (alert);
43
44    int rawPoints =
45        scoreAcwr (acwrL) + scorePctQwUp (pctQwUpL) +
46        scoreMonotony (monoL) + scoreStrain (strainL);
47    int points = Math.max (0, rawPoints);
48
49    //... verificacao de atleta
50
51    var snap = new AthletePerformanceSnapshot ();
52    snap.setAsOf (Instant.now ());
53    snap.setAthlete (athleteEntity);
54    snap.setCoach (sessionCoach);
55    snap.setPoints (points);
56    snap.setPosition (0);
57    athletePerformanceSnapshotRepository.save (snap);
58 }

```

Implementação 2. Back-end — Cálculo de métricas(ACRW, %QW, monotony e strain).

A partir das métricas geradas diretamente no banco de dados, por meio de consultas SQL e *views* específicas (como *vw_athlete_daily_load* e *vw_athlete_qw_load*), o sistema calcula, de forma automatizada, indicadores consolidados por atleta e sessão. Essa abordagem reduz o custo computacional no lado do servidor *Java*, uma vez que os valores de carga (como *PlayerLoad*, *sRPE*, tempo de treino e intensidade média) já são pré-processados e armazenados na tabela *session_loads*. A camada de serviço, por meio da classe *WeeklyLoadQueryRepository*, realiza a consulta desses dados estruturados e aplica a lógica de classificação de zonas de risco.

A Implementação 4 apresenta o método *psychoCallGroqAndPersist*, responsável por gerar recomendações psicoemocionais personalizadas. O serviço constrói dinamicamente dois blocos de texto, *systemPrompt* e *userPrompt*, que são enviados ao modelo de linguagem. O primeiro define o papel da IA (assistente de desempenho esportivo), o tom empático e os limites de atuação (sem diagnósticos médicos), enquanto o segundo insere o contexto específico da sessão (ID, atleta e pontuações de *sRPE*, fadiga, dor muscular, humor e energia). O modelo retorna uma lista em *markdown* com recomendações objetivas e uma linha final de aviso, reforçando boas práticas de recuperação e bem-estar.

De modo análogo, a Implementação 5 do Apêndice A.5 mostra o método *perfCallGroqAndPersist*, voltado às recomendações de desempenho físico. Nesse caso, o *systemPrompt* instrui o modelo a gerar respostas segmentadas por métrica, *ACWR*, variação semanal de carga (*%QW*), *monotony* e *strain*, cada uma com orientações específicas para ajustes de treino e recuperação. O *userPrompt* inclui tanto os valores numéricos quanto os rótulos de risco associados (por exemplo, *ACWR=attention* ou *Monotony=high_risk*), permitindo que a IA interprete a situação atual do atleta e produza recomendações adaptadas ao contexto.

Por fim, a Implementação 2 detalha a rotina responsável por percorrer os atletas de uma equipe, consultar suas métricas agregadas e identificar situações de atenção ou risco. Para cada atleta, o sistema busca a janela semanal mais recente de dados e calcula os rótulos correspondentes às métricas principais. Caso algum indicador esteja em zona crítica, é gerado um alerta de carga (*TrainingLoadAlert*) e um registro de desempenho (*AthletePerformanceSnapshot*). Esse registro também atualiza a pontuação do atleta na gamificação do sistema, de acordo com a soma ponderada das condições de *ACWR*, *%QW*, *monotony* e *strain*.

4.4. Integração com IA: *Prompt* e Comunicação com o *Groq*

A integração com a camada de inteligência artificial do WIKO foi realizada por meio do *Spring AI*, em conjunto com a plataforma de inferência *Groq* (Tanzu, 2025b). Essa arquitetura permite que o *back-end* formule *prompts* estruturados contendo o contexto esportivo do atleta e, em seguida, envie a requisição ao modelo hospedado na infraestrutura do *Groq*. A resposta em linguagem natural é retornada ao *back-end*, que a disponibiliza imediatamente no *front-end* e também a inclui nos relatórios em PDF exportados pelo sistema.

Para garantir resiliência, foi implementado um mecanismo de *fallback*: caso a comunicação com a IA não esteja disponível, o sistema retorna recomendações padronizadas. Dessa forma, assegura-se que o treinador sempre receba orientações, seja a partir

da geração dinâmica da IA ou de conteúdos previamente validados na literatura.

O fluxo de funcionamento pode ser resumido da seguinte forma:

1. O *front-end* realiza uma chamada para um *endpoint* de recomendações.
2. O *back-end* coleta as métricas e constrói um *prompt* padronizado (*system + user*).
3. O *Spring AI* encaminha a requisição à API do *Groq*.
4. O *Groq* processa o *prompt* em um modelo de linguagem e retorna uma resposta textual estruturada em *markdown*.
5. O *back-end* repassa essa resposta ao *front-end*, que a exibe na interface e a armazena no relatório em PDF.

Foram criados dois conjuntos principais de *prompts*, de acordo com o tipo de recomendação desejada:

- **Recomendações psicoemocionais:** nesse caso, o `systemPrompt` instrui a IA a agir como um assistente de desempenho esportivo, fornecendo recomendações concisas, claras e práticas para bem-estar psicoemocional e recuperação, a partir de escalas de 0 a 10 (sRPE, fadiga, dor muscular, humor e energia). O formato de saída é em *markdown*, com 3 a 7 itens curtos em lista, e sempre finalizado com a mensagem: “*Please seek professional medical support if symptoms persist or worsen*”.
- **Recomendações de desempenho:** para recomendações de carga de treino, o `systemPrompt` define que a saída deve ser organizada por métrica: ACWR, variação semanal de carga (%QW), *monotony* e *strain*. Cada seção deve conter duas a três recomendações práticas em formato de lista, indicando ajustes para a próxima sessão ou semana. Também neste caso, a saída é em *markdown* e finalizada com a mesma mensagem de cautela.

A estrutura do *prompt* utilizado para o envio das informações é dividida em duas partes: o `SystemPrompt`, responsável por definir as regras de estilo, linguagem e formato da saída, e o `UserPrompt`, que fornece o contexto específico do atleta e da sessão (IDs, valores das métricas e respectivos *labels*). Essa abordagem assegura que a IA gere respostas consistentes, objetivas e alinhadas à literatura esportiva, evitando ambiguidades e facilitando sua aplicação prática por treinadores e atletas.

4.5. Documentação da API com *Swagger*

Devido à abrangência das funcionalidades do sistema, o *back-end* conta com diversos controladores REST, cada um responsável por uma parte específica da lógica (registro de usuários, métricas de sessões, recomendações de desempenho e psicoemocionais, relatórios, entre outros). Para evitar que essa fragmentação tornasse a API de difícil compreensão, foi adotada a documentação automática com o *Swagger/OpenAPI* (SmartBear Software, 2025).

O *Swagger* organiza os *endpoints* em categorias baseadas nos controladores da aplicação, utilizando nomes técnicos em inglês e em minúsculas (por exemplo, `athlete-controller`, `session-metrics-controller`, `ingest-controller`). Essa estrutura favorece a navegação mesmo diante do número elevado de serviços, permitindo consultar parâmetros de entrada, códigos de resposta

e exemplos de saída em JSON, além de testar requisições diretamente pelo navegador, conforme apresentado nas Figuras 23, 24 e 25 do Apêndice A.4.

Essa abordagem garante transparência e reduz a curva de aprendizado para novos usuários da API, além de favorecer futuras integrações com outros sistemas ou aplicações móveis.

4.6. Gestão do Projeto com *Scrum* e *Trello*

Para a organização e acompanhamento do desenvolvimento da plataforma, foi adotado o *framework* ágil *Scrum*. O trabalho foi dividido em *sprints* curtas, cada uma com objetivos bem definidos, permitindo que funcionalidades fossem entregues de forma incremental. O *backlog* do produto foi constantemente revisado, priorizando tarefas críticas como ingestão de dados, cálculos de métricas, geração de relatórios e integração com a IA.

Como ferramenta prática para aplicar o *Scrum*, utilizou-se o *Trello*, que possibilitou a criação de quadros organizados em listas (“A fazer”, “Em andamento” e “Concluído”). Essa visualização facilitou a definição de prioridades, a atribuição de responsabilidades e o acompanhamento do progresso do time ao longo das *sprints*, conforme ilustrado na Figura 22 do Apêndice A.4. Essa abordagem combinada garantiu maior transparência no andamento do projeto, além de apoiar a tomada de decisão rápida e a adaptação a mudanças nos requisitos.

4.7. Apresentação dos Resultados do Teste SUS

O Teste SUS foi aplicado a um grupo de nove participantes, todos com histórico de prática esportiva, em especial atletas e ex-atletas de futebol. Essa escolha buscou garantir que os avaliadores tivessem familiaridade com o contexto esportivo e com o tipo de informação apresentada pelo sistema. As respostas completas do formulário aplicado aos atletas encontram-se no Apêndice A.6, nas Figuras 26–35.

Após o cálculo das pontuações, seguindo o método proposto por Brooke (1986), obteve-se uma pontuação média final de 83,6, situando o WIKO na faixa de usabilidade considerada excelente, o que evidencia a percepção positiva dos usuários em relação à plataforma.

Outro ponto relevante é a robustez do Teste SUS quanto à confiabilidade dos resultados. Mesmo que algum participante responda de forma aleatória ou desatenta, a metodologia permite identificar discrepâncias nas respostas, o que possibilita a exclusão dessas respostas da amostra sem comprometer a análise global. Essa característica garante maior consistência e validade estatística ao resultado obtido, fortalecendo a confiabilidade das conclusões sobre a usabilidade do sistema.

4.8. Validação com Profissional da Área Esportiva

Com o objetivo de verificar a aplicabilidade prática e a clareza das informações fornecidas pelo sistema, foi realizada uma etapa de validação junto a profissionais da área esportiva. Para isso, foi disponibilizado um formulário de avaliação voltado a coletar percepções sobre a qualidade das recomendações geradas pela inteligência artificial, a clareza das métricas apresentadas e a utilidade prática do sistema no contexto esportivo.

As respostas obtidas permitiram identificar a receptividade do sistema entre profissionais da área e serviram como base para avaliar seu potencial de aplicação em ambientes reais de monitoramento e análise de desempenho.

4.8.1. Teste de Usabilidade (Profissional—Plataforma)

Além da avaliação de usabilidade voltada aos usuários finais, foi conduzido um segundo teste direcionado especificamente aos profissionais de Educação Física, principais destinatários das análises e recomendações geradas pela plataforma WIKO. O objetivo deste instrumento foi mensurar a clareza, a confiabilidade e a utilidade prática das informações apresentadas pelo sistema no contexto do monitoramento esportivo. O teste contou com a participação de quatro profissionais de Educação Física. As respostas completas desse formulário encontram-se no Apêndice A.6, nas Figuras 36–45.

Assim como o teste SUS tradicional, esta avaliação utilizou uma escala de *Likert* de cinco pontos, variando de “discordo completamente” (1) a “concordo completamente” (5). As perguntas foram adaptadas ao contexto profissional, abordando dimensões como clareza das métricas, confiabilidade das recomendações e aplicabilidade prática dos resultados.

As afirmações apresentadas aos profissionais foram:

- As recomendações fornecidas pelo sistema são confiáveis para apoiar decisões no treinamento.
- As recomendações são apresentadas de forma clara e compreensível.
- As recomendações são práticas e aplicáveis no contexto do treinamento esportivo.
- As recomendações refletem de maneira coerente os indicadores e métricas apresentados pelo sistema.
- O sistema permite aplicar o teste psicoemocional de forma simples e eficiente.
- As recomendações baseadas no teste psicoemocional refletem adequadamente o bem-estar do atleta.
- O questionário psicoemocional é de fácil preenchimento para os atletas.
- O sistema é fácil de utilizar no dia a dia de um treinador.
- O sistema organiza os dados e resultados de forma clara e acessível.
- O uso do sistema contribui para tomar decisões mais seguras sobre carga de treino e prevenção de lesões.

Essas questões permitiram avaliar a percepção dos profissionais quanto à relevância prática da plataforma WIKO como ferramenta de apoio à tomada de decisão esportiva. Os resultados obtidos nessa etapa complementam a análise de usabilidade tradicional, fornecendo uma visão mais abrangente sobre a experiência de uso e a confiança dos especialistas no sistema.

Como destaque, os resultados obtidos nas questões relacionadas à confiabilidade e aplicabilidade das recomendações demonstraram uma percepção extremamente positiva por parte dos profissionais.

Na afirmação “As recomendações fornecidas pelo sistema são confiáveis para apoiar decisões no treinamento”, 75% dos avaliadores indicaram concordância total e

25% concordância parcial, evidenciando um alto nível de confiança nas informações geradas pela plataforma.

Da mesma forma, na afirmação “As recomendações são práticas e aplicáveis no contexto do treinamento esportivo”, observou-se 100% de concordância total, reforçando que as recomendações produzidas pelo sistema são percebidas como úteis e facilmente integráveis ao cotidiano profissional.

Esses resultados confirmam que a plataforma apresenta elevado grau de aceitação entre os profissionais de Educação Física, tanto em termos de confiabilidade quanto de aplicabilidade prática, fortalecendo sua credibilidade como ferramenta de apoio à tomada de decisão esportiva.

5. Considerações Finais

O presente trabalho apresentou o desenvolvimento da plataforma WIKO, cujo objetivo central é apoiar treinadores e atletas na análise da carga de treinamento, prevenção de lesões e otimização do desempenho. A principal contribuição desta pesquisa está na democratização da análise de desempenho esportivo, ao oferecer uma solução acessível, transparente e fundamentada em métricas consagradas da literatura, que já estão disponíveis na comunidade científica, mas cujo uso prático muitas vezes exige *softwares* especializados e de alto custo, tradicionalmente acessíveis apenas a equipes de maior porte.

Outro aspecto inovador consiste no uso de Modelos de Linguagem de Grande Escala (LLMs) para transformar indicadores quantitativos em *insights* acionáveis e personalizados. Essa abordagem amplia a utilidade prática dos cálculos, permitindo que métricas complexas, como ACWR e *monotony*, sejam traduzidas em recomendações compreensíveis, alinhadas ao contexto esportivo.

Além disso, a arquitetura proposta possibilita que o treinador tenha maior controle e clareza na gestão de seus atletas, favorecendo ajustes mais precisos nas cargas de treino e promovendo um manejo mais eficiente do desempenho e da prevenção de lesões.

Por outro lado, reconhecem-se algumas limitações da proposta. A primeira refere-se à dependência de APIs externas, o que pode gerar custos operacionais adicionais, além de levantar preocupações relacionadas à privacidade e à conformidade com a LGPD. Outra limitação está ligada à possibilidade de alucinações por parte dos LLMs, exigindo a presença de supervisão humana para validação das recomendações, de modo a assegurar a confiabilidade das orientações geradas.

Assim, embora desafios ainda existam, os resultados obtidos indicam que a WIKO avança no sentido de tornar a ciência do esporte mais acessível, interpretável e aplicável, abrindo espaço para evoluções futuras, tanto em termos de integração tecnológica quanto de validação em cenários reais de treinamento.

Como continuidade deste trabalho, sugerem-se alguns caminhos de investigação e desenvolvimento. Uma possibilidade é a aplicação de técnicas de *fine-tuning* ou de *Retrieval Augmented Generation* (RAG) sobre bases de conhecimento compostas por literatura científica esportiva. Essa abordagem poderia aumentar a precisão e a confiabilidade das recomendações, reduzindo a dependência de modelos genéricos e aproximando as respostas ainda mais do estado da arte na ciência do esporte.

Outra recomendação para trabalhos futuros consiste em realizar uma comparação sistemática entre as recomendações fornecidas por especialistas humanos e aquelas geradas pelo sistema proposto. Tal análise permitiria avaliar o nível de concordância, identificar pontos de divergência e, conseqüentemente, aprimorar a utilidade prática da plataforma, fortalecendo sua adoção em contextos reais de treinamento.

Referências

- Afonso, J., Nakamura, F. Y., Canário-Lemos, R., Peixoto, R., Fernandes, C., Mota, T., Ferreira, M., Silva, R., Teixeira, A., e Clemente, F. M. (2021). A novel approach to training monotony and acute-chronic workload index: a comparative study in soccer. *Frontiers in Sports and Active Living*, 3:661200.
- Aggarwal, C. C. (2018). *Neural networks and deep learning*. Springer, Cham.
- Alura (2020). O que é Git e GitHub: os primeiros passos nessas ferramentas. Disponível em: <https://www.alura.com.br/artigos/o-que-e-git-github>. Acesso em: 20 jun. 2025.
- Anthropic (2024). Claude 3 system card. Technical Report.
- Atlassian (2025). Trello – organize anything, together. Disponível em: <https://trello.com/>. Acesso em: 12 jun. 2025.
- Banister, E. W., Calvert, T. W., Savage, M. V., e Bach, T. (1975). A systems model of training for athletic performance. *Australian Journal of Sports Medicine*, 7(3):57–61.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Boyd, L. J., Ball, K., e Aughey, R. J. (2011). The reliability of minimaxx accelerometers for measuring physical activity in australian football. *International Journal of Sports Physiology and Performance*, 6(3):311–321.
- Brasil (2018). Lei nº 13.709, de 14 de agosto de 2018 – Lei Geral de Proteção de Dados Pessoais (LGPD). Disponível em: <https://bit.ly/42DAjup>. Acesso em: 23 abr. 2025.
- Brasil, U. D. (2024). Guia atualizado de como utilizar a escala SUS (System Usability Scale) no seu produto. Disponível em: <https://bit.ly/4mqEy5C>. Acesso em: 14 maio 2025.
- Brooke, J. (1986). SUS: a “quick and dirty” usability scale. In Jordan, P. W., Thomas, B., Weerdmeester, B. A., e McClelland, A. L., editors, *Usability Evaluation in Industry*. Taylor and Francis, London. Disponível em: <https://bit.ly/44Jyi2F>. Acesso em: 14 maio 2025.
- Chicco, D. (2021). Ten quick tips for machine learning in computational biology. *BioData Mining*, 14(1):1–23.
- Conde, M. (2021). Especialización en gestión estratégica de la tecnología informática.
- Cotta, R. M. (2018). *Análise de desempenho no futebol*. Appris Editora.
- Cozman, F. G., Plonsky, G. A., e Neri, H. (2021). *Inteligência artificial: avanços e tendências*. Universidade de São Paulo, Instituto de Estudos Avançados, São Paulo.
- Cummins, C., Orr, R., O’Connor, H., e West, C. (2013). Global Positioning Systems (GPS) and microtechnology sensors in team sports: a systematic review. *Sports Medicine*, 43(10):1025–1042.
- Drumond, J. G. F. (2011). Tecnologia e esporte: perspectivas bioéticas. *Revista-Centro Universitário São Camilo*, 5:411–418.

- Figma (2024). Figma. Disponível em: <https://www.figma.com/pt-br/>. Acesso em: 10 nov. 2024.
- Foster, C. (1998). Monitoring training in athletes with reference to overtraining syndrome. *Medicine & Science in Sports & Exercise*, 30(7):1164–1168.
- Gabbett, T. J., Hulin, B. T., Blanch, P., e Whiteley, R. (2016). High training workloads alone do not cause sports injuries: how you get there is the real issue. *British Journal of Sports Medicine*, 50(8):444–445.
- Garrard, J. (2022). *Health sciences literature review made easy: the matrix method*. Jones & Bartlett Learning, Burlington, MA, 6 edition.
- Goudsmit, J., Otter, R. T. A., Stoter, I., van Holland, B., van der Zwaard, S., de Jong, J., e Vos, S. (2022). Co-operative design of a coach dashboard for training monitoring and feedback. *Sensors*, 22.
- Juliano, E., Thakkar, C., Taber, C., Raval, M., Kaya, T., e Senbel, S. (2023). A dynamic online dashboard for tracking the performance of division 1 basketball athletic performance. In *2023 IEEE 28th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 314–318.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. t., Rocktäschel, T., Riedel, S., e Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9459–9474.
- Machado, M. V. e Marques, A. C. (2013). *Fisiologia do exercício*. Fundação Vale, em cooperação com a UNESCO no Brasil. Coordenação: Setor de Ciências Humanas e Sociais da Representação da UNESCO no Brasil.
- Masse, M. (2011). *REST API Design Rulebook*. O’Reilly Media, Sebastopol.
- McFarland, M. (2014). A wellness monitoring tool for youth athletes. *Journal of Australian Strength and Conditioning*, 22:22–26.
- Nielsen, J. (2007). *Usabilidade na web*. Elsevier.
- Okazaki, V. H. A., Dascal, J. B., Okazaki, F. H. A., e Teixeira, L. A. (2012). Ciência e tecnologia aplicada à melhoria do desempenho esportivo. *Revista Mackenzie de Educação Física e Esporte*, 11(1):143–157.
- OpenAI (2023). GPT-4 technical report. arXiv:2303.08774.
- Oracle Corporation (2025). Java platform, standard edition. Disponível em: <https://www.oracle.com/java/technologies/>. Acesso em: 12 jun. 2025.
- Ráthonyi, G., Bácsné Bába, É., Müller, A., e Ráthonyi-Ódor, K. (2018). How digital technologies are changing sport? *Applied Studies in Agribusiness and Commerce*, 12:89–96.
- React (2025). React introduction. Disponível em: <https://www.geeksforgeeks.org/reactjs/reactjs-introduction/>. Acesso em: 12 jun. 2025.
- Rebbouj, M. e Lotfi, S. (2024). Athlete performance prediction using intelligent reporting and regression model analysis: a generative approach for training planning. *Journal of Physical Education and Sport*, 24:1927–1932.
- Russell, S. e Norvig, P. (2010). *Artificial Intelligence: a modern approach*. Prentice Hall, Upper Saddle River, NJ, 3rd edition.
- Santos, E. e Blanco, J. (2009). *Fisiologia da fadiga muscular: quebrando paradigmas*. Phorte Editora, São Paulo.
- Scrum (2025). Scrum. Disponível em: <https://www.scrum.org/>. Acesso em: 12 jun. 2025.

- Shamah, M. E. P. (2021). A prática do analista de desempenho nas categorias de base dos clubes brasileiros da série A.
- SmartBear Software (2025). Swagger - API documentation & design tools for teams. Acessado em: 02 jul. 2025.
- Software, D. (2021). Rest vs GraphQL: which API design style is right for your organization? Disponível em: <http://bit.ly/3ZZyGH6>. Acesso em: 02 jul. 2025.
- Sommerville, I. (2011). *Engenharia de software*. Pearson Prentice Hall, São Paulo, 9ª edição. Tradução de Ivan Bosnic e Kalinka G. de O. Gonçalves. Revisão técnica de Kechi Hirama. 3ª reimpressão, dezembro 2013.
- Tanzu, V. (2025a). Spring AI: Simplifying LLM integration in Java applications.
- Tanzu, V. (2025b). Spring AI with Groq: a blazingly fast ai inference engine. Disponível em: <https://blogs.vmware.com/tanzu/spring-ai-with-groq-a-blazingly-fast-ai-inference-engine/>. Acesso em: 20 set. 2025. Acesso em: 20 set. 2025.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, , e Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30:6000–6010.
- VMware, Inc. (2025). Spring boot project overview. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 12 jun. 2025.
- Wazlawick, R. S. (2009). *Metodologia de pesquisa em ciência da computação*. Elsevier, Rio de Janeiro.
- Zheng, A. e Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Sebastopol, CA.

A. Apêndice

A.1. Métricas da Base de Dados Aplicadas à Geração de Inferência por Inteligência Artificial

Quadro 3. Principais métricas coletadas na base de dados.

Métrica	Descrição
Seção	Treino ou jogo (“1º Tempo”)
Tarefa	Tipo de tarefa realizada na sessão
Data de treino/jogo	Data em que o atleta participou do treino ou partida
Tempo de treino/jogo	Tempo em que o atleta participou do treino ou partida
Posição do jogador	Função em campo (goleiro, zagueiro, atacante etc.)
Número da camisa	Identificador numérico do atleta
Distância total	Soma, em metros, de todo o percurso realizado
Distância por minuto	Distância média percorrida por minuto de atividade
Número de desacelerações explosivas	Desacelerações explosivas $> 2,5 \text{ m/s}^2$
Número de acelerações explosivas	Acelerações explosivas $> 2,5 \text{ m/s}^2$
Distância faixa 1 (0–6 km/h)	Metros percorridos na zona de velocidade 1
Distância faixa 2 (6–12 km/h)	Metros percorridos na zona de velocidade 2
Distância faixa 3 (12–18 km/h)	Metros percorridos na zona de velocidade 3
Distância faixa 4 (18–24 km/h)	Metros percorridos na zona de velocidade 4
Distância faixa 5 (24–30 km/h)	Metros percorridos na zona de velocidade 5
Distância faixa 6 (>30 km/h)	Metros percorridos acima de 30 km/h
Carga do jogador	Soma ponderada das zonas de velocidade (<i>player-load</i>)
Nº de intervalos de alta intensidade	Corridas rápidas ou <i>sprints</i> repetidos

Quadro 4. Métricas específicas de alta velocidade.

Métrica	Descrição
Aceleração máxima	Maior valor de aceleração registrada
Corrida alta intensidade	Distância em corrida de alta intensidade
Desaceleração máxima	Maior valor de desaceleração registrada
Velocidade máxima	Velocidade máxima registrada
Velocidade média	Média das velocidades ao longo da sessão
Total de acelerações	Nº de acelerações > 1,5 m/s ²
Tempo em HMLD	Tempo em zona de alta carga metabólica
Carga metabólica alta	Valor da carga metabólica de alta intensidade
Eventos HMLD	Nº de vezes que o atleta entrou na zona HMLD
Corrida em alta intensidade	Nº de corridas > 20 km/h
Total de desacelerações	Quantidade de eventos de desaceleração
Nº de <i>sprints</i>	>25 km/h por pelo menos 1 s
Corridas em alta velocidade	Eventos > 20 km/h
Tempo na faixa 4 (>30 km/h)	Metros percorridos acima de 30 km/h
Percepção subjetiva de esforço	Esforço percebido durante treino/jogo

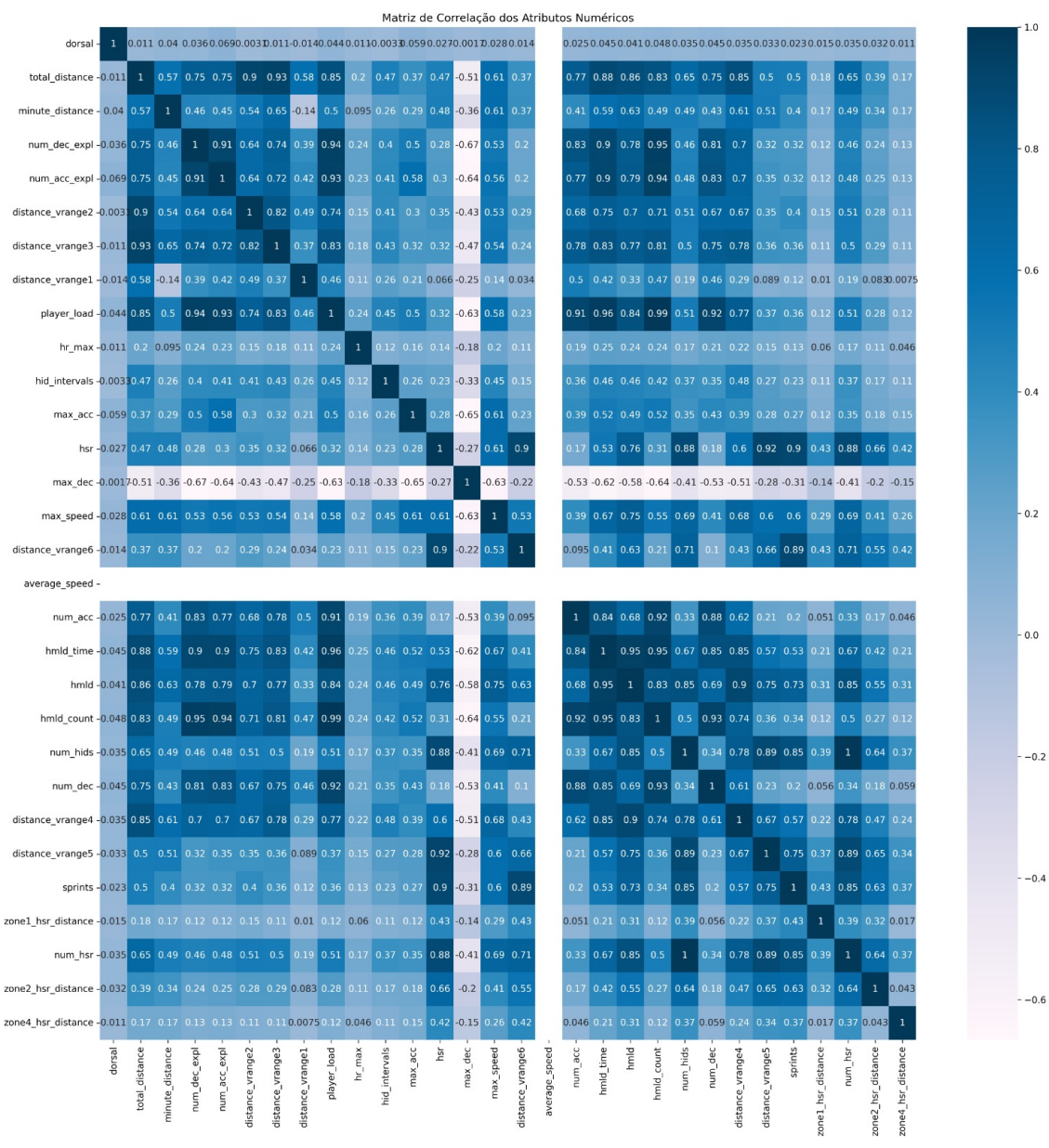


Figura 12. Correlação dos atributos das planilhas dos atletas.

A.2. Protótipos

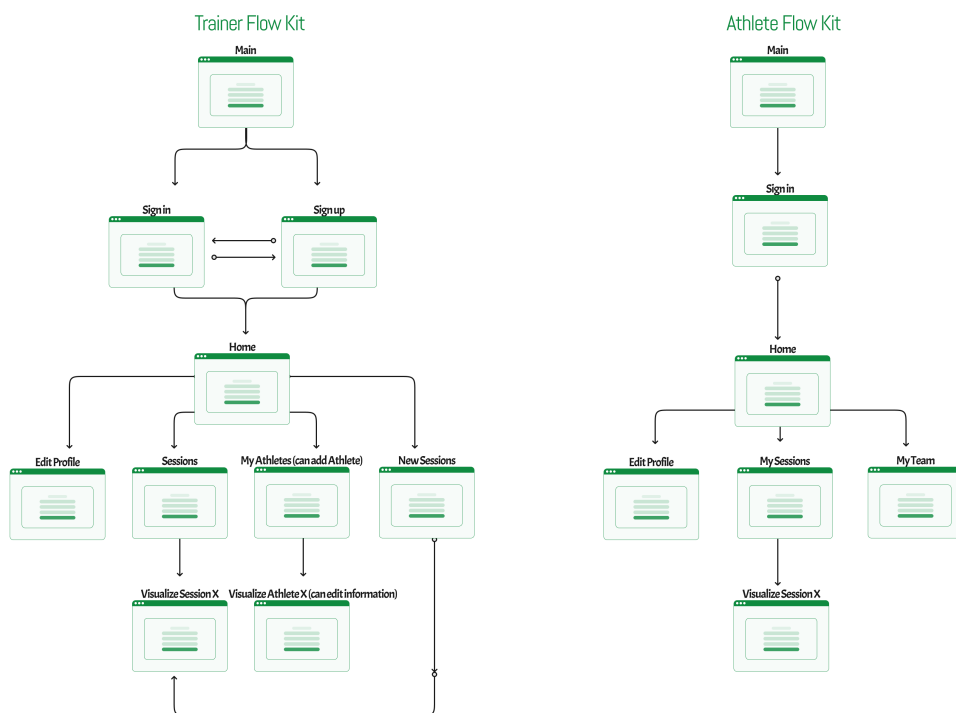


Figura 13. Fluxo de telas dos usuários.

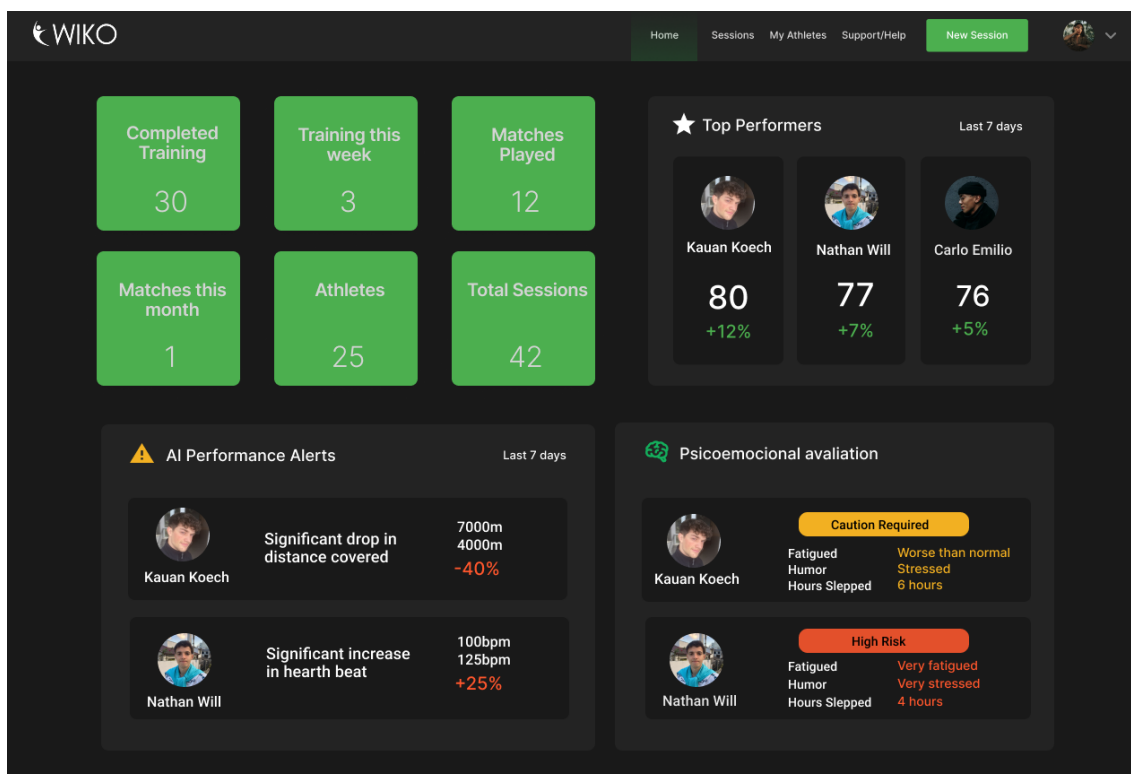


Figura 14. Tela inicial do treinador, com resumo de carga, alertas e pontuação dos atletas.

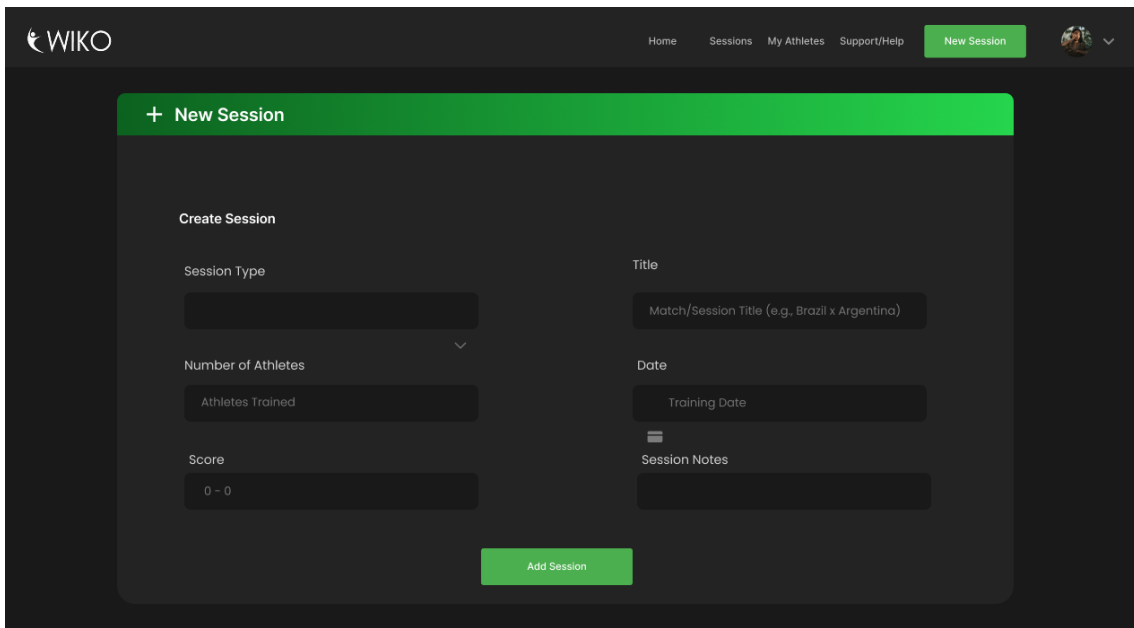


Figura 15. Tela de cadastro de nova sessão de treino ou jogo.

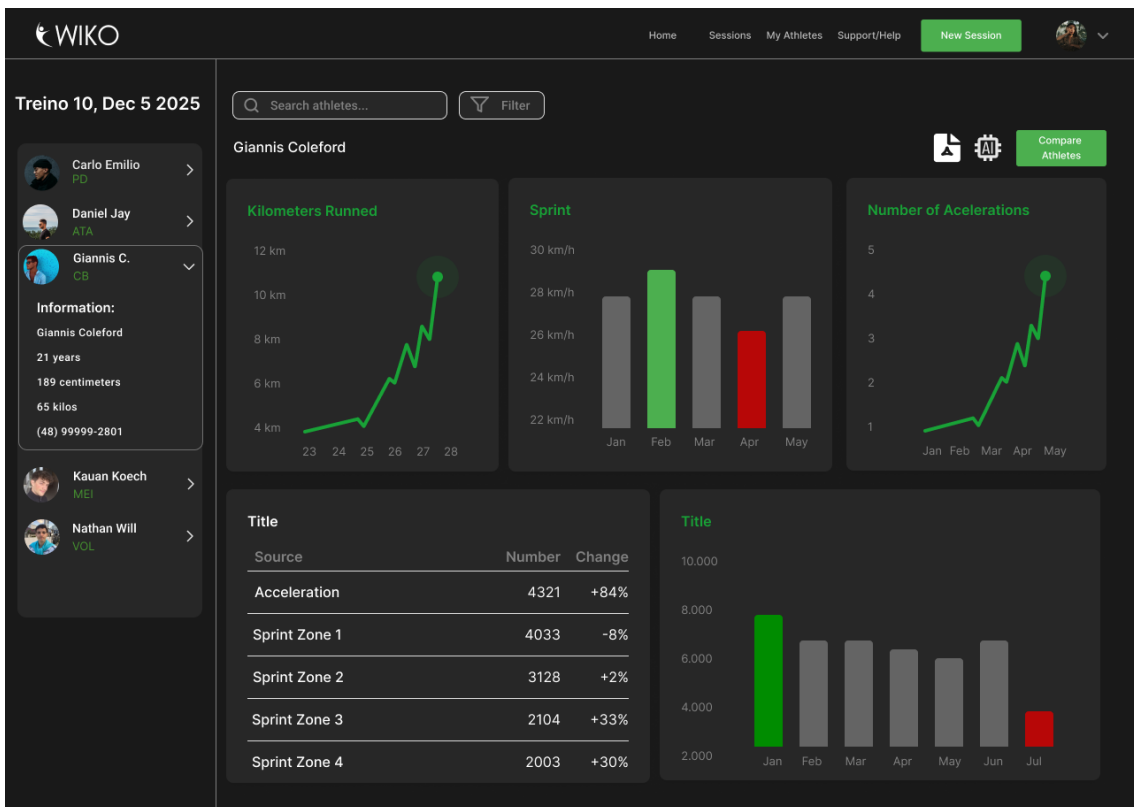


Figura 16. Tela de detalhamento de sessão, com métricas e respostas psicoemocionais.

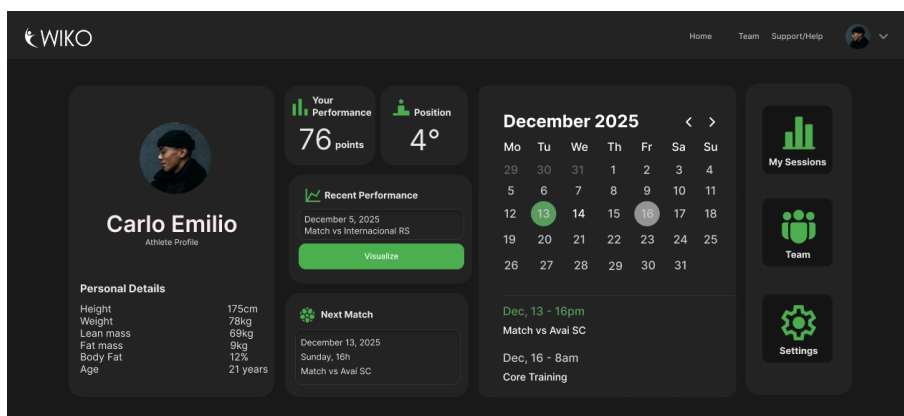


Figura 17. Tela inicial do atleta, com calendário, desempenho e pontuação de gamificação.

A.3. Interfaces do Sistema



Figura 18. Dashboard do coach: visão geral das métricas, cargas e indicadores dos atletas.

Psycho-emotional Report

Generated at 04/10/2025, 15:47:28

Nathan Athlete

nathan.wm@aluno.ifsc.edu.br • Midfielder

Submitted at 02/10/2025, 16:02:07

Metric	Value (0–10)
sRPE	8
Fatigue	8
Soreness	7
Mood	3
Energy	3

AI Recommendations

Recovery and Well-being Recommendations

Given your high sRPE score of 8, it's essential to prioritize recovery to avoid further fatigue and soreness. Focus on low-intensity activities for the next 24-48 hours to allow your body to recuperate. With fatigue and soreness scores of 8 and 7, respectively, ensure you're getting adequate sleep (7-9 hours) and staying hydrated by drinking plenty of water throughout the day. Proper nutrition, including a balanced diet rich in proteins, complex carbohydrates, and healthy fats, is also crucial for recovery. Your mood and energy scores of 3 indicate a need for extra self-care. Engage in relaxation techniques such as meditation, deep breathing, or yoga to help manage stress and improve your mental well-being. Consider reducing your training load temporarily to alleviate pressure. It's vital to monitor your symptoms closely. If you experience persistent or worsening fatigue, soreness, or a low mood, it may be necessary to reassess your training strategy and seek guidance from a sports professional. Remember, taking care of your mental health is as important as your physical recovery. Reach out to friends, family, or a mental health professional if you're struggling with your mood or emotional state. Please seek professional medical support if symptoms persist or worsen.

Figura 19. Relatório em PDF com indicadores individuais, zonas de risco e recomendação textual personalizada.

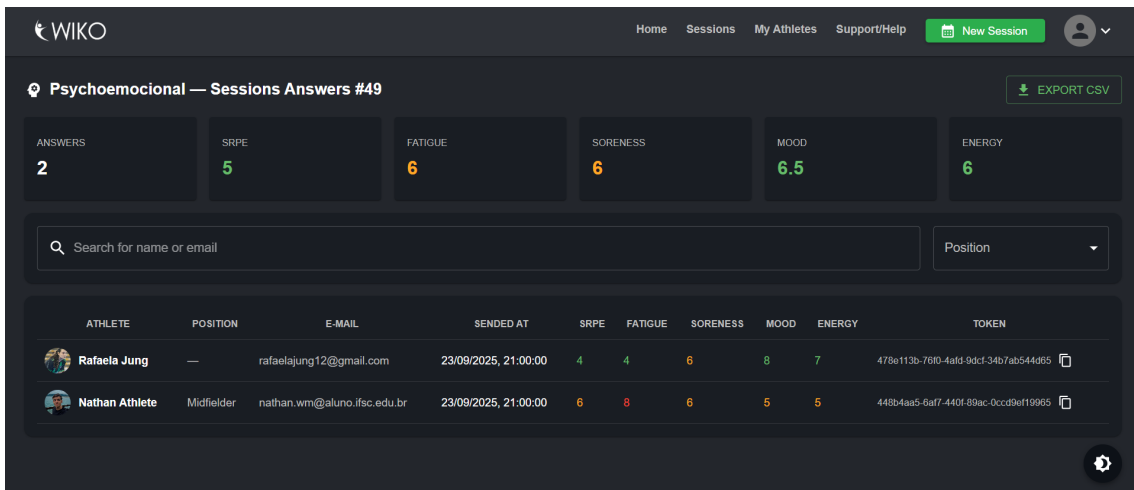


Figura 20. Tela de respostas do questionário psicoemocional, com indicadores médios por sessão, filtros por posição e opção de exportação em CSV.

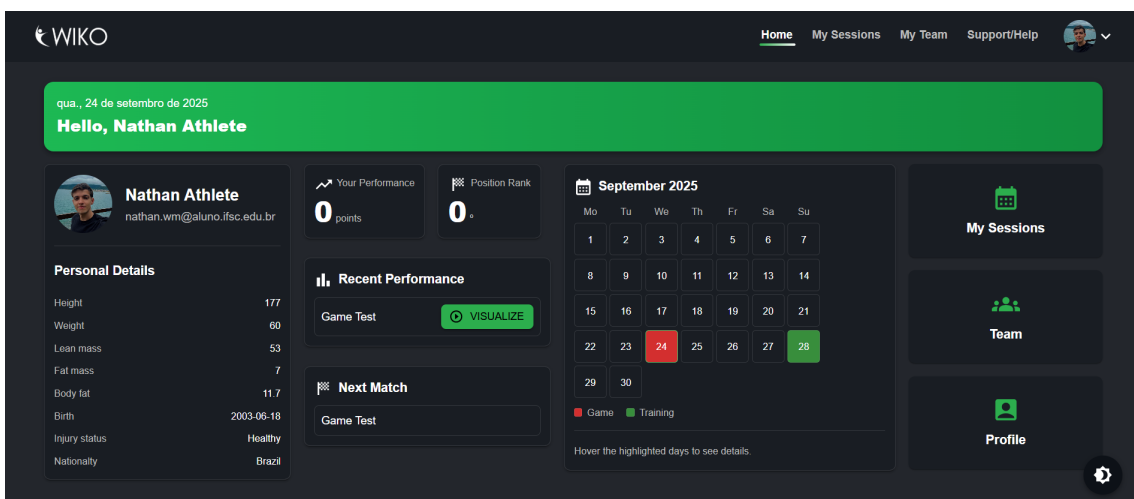


Figura 21. Tela principal do atleta, com perfil, calendário de sessões e gamificação.

A.4. Swagger e Trello

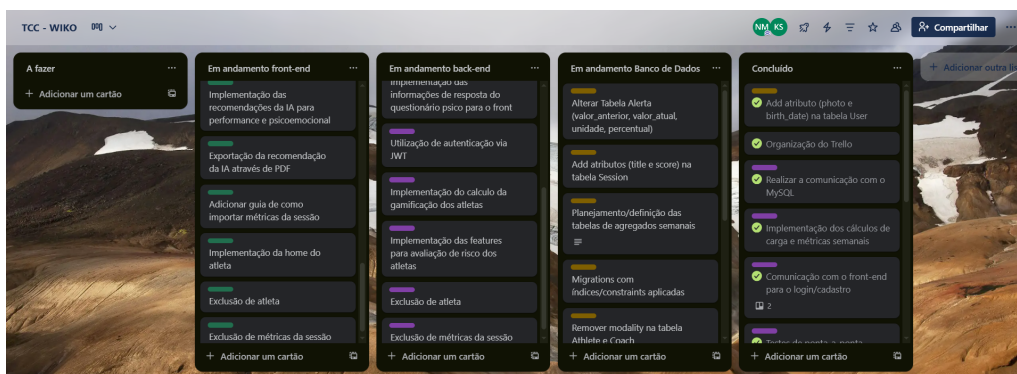


Figura 22. Quadro do Trello utilizado para o gerenciamento das tarefas de desenvolvimento do sistema WIKO.

Swagger powered by SMARTBEAR /v3/api-docs Explore

WIKO API v1.0 OAS 3.1

/v3/api-docs

Intelligent System for Monitoring Training Load, Performance and Injury Risk in Athletes API Documentation

Servers

coach-athlete-controller

- PUT** /api/coach/{coachId}/update/athlete/{athleteId}
- GET** /api/coach/{coachId}/athletes/{athleteId}
- DELETE** /api/coach/athlete/{id}

session-metrics-controller

- PUT** /api/coach/sessions/update/{id}
- POST** /api/coach/sessions/{sessionId}/metrics/import
- PATCH** /api/coach/sessions/update/notes/{id}
- DELETE** /api/coach/sessions/{sessionId}/metrics

coach-profile-controller

- GET** /api/coach/profile/{coachId}
- PUT** /api/coach/profile/{coachId}
- POST** /api/coach/profile/{coachId}/delete-or-transfer
- POST** /api/coach/profile/{coachId}/avatar

athlete-profile-controller

- GET** /api/athletes/{athleteId}/profile
- PUT** /api/athletes/{athleteId}/profile
- POST** /api/athletes/{athleteId}/password
- POST** /api/athletes/{athleteId}/avatar
- DELETE** /api/athletes/delete/{id}

ingest-controller

- POST** /ingest/csv Ingere CSV de métricas para um coach

auth-controller

- POST** /auth/register/coach
- POST** /auth/register/athlete
- POST** /auth/login

session-coach-controller

- POST** /api/session/register
- GET** /api/session/get
- DELETE** /api/session/{id}

Figura 23. Endpoints relacionados aos controladores *coach-athlete*, *session-metrics*, *coach-profile*, *athlete-profile*, *ingest*, *auth* e *session-coach* documentados no Swagger.

psycho-form-controller		^
POST	/api/psycho-form/{token}/submit	∨
POST	/api/psycho-form/create	∨
GET	/api/psycho-form/{token}	∨
GET	/api/psycho-form/answers/sessions/{sessionId}	∨
coach-invite-controller		^
POST	/api/coach/{coachId}/athlete-invitations	∨
POST	/api/auth/complete-invite	∨
GET	/api/athletes/invitations/{token}	∨
Recommendations		^
POST	/api/coach/psy-form/{sessionId}/athletes/{athleteId}/recommendations	∨
POST	/api/coach/performance/{sessionId}/athletes/{athleteId}/recommendations	∨
POST	Generate psycho emotional recommendations for the athletes	
POST	Generate performance recommendations for the athletes	
session-charts-controller		^
GET	/api/sessions/{sessionId}/metrics	∨
GET	/api/sessions/{sessionId}/athletes	∨
coach-controller		^
GET	/api/coach/top-performers	∨
GET	/api/coach/stats	∨
GET	/api/coach/athletes	∨
load-metrics-controller		^
GET	/api/coach/athletes/{athleteId}/risk-calculations	∨
GET	/api/coach/athletes/{athleteId}/export/qw.csv	∨

Figura 24. Endpoints dos controladores *psycho-form-controller*, *session-charts-controller*, *load-metrics-controller*, *recommendations* e *coach-controller* documentados no *Swagger*.

athlete-session-controller		^
GET	/api/athletes/{athleteId}/sessions	∨
peer-athlete-controller		^
GET	/api/athletes/{athleteId}/peers	∨
athlete-controller		^
GET	/api/athlete/{athleteId}/home	∨
GET	/api/athlete/{athleteId}/home/recent	∨
GET	/api/athlete/{athleteId}/home/performance	∨
GET	/api/athlete/{athleteId}/home/next-match	∨
GET	/api/athlete/{athleteId}/home/calendar	∨
performance-alert-controller		^
GET	/api/alerts/training-load	∨
GET	/api/alerts/training-load/athlete/{athleteId}	∨
DELETE	/api/alerts/performance/{alertId}/delete/{coachId}	∨
psycho-alert-controller		^
GET	/api/alerts/sessions/{sessionId}/psy-form/answers/{athleteId}	∨
GET	/api/alerts/psycho-risk/{coachId}	∨

Figura 25. Endpoints dos controladores *athlete-session-controller*, *peer-athlete-controller*, *performance-alert-controller* e *psycho-alert-controller* documentados no Swagger.

A.5. Implementação dos Principais Códigos

```

1  const handleExportPdf = () =>
2    if (!answer) return;
3
4    const fmt = (v?: number) => (v == null ? " " : v.toFixed(2));
5    const fmtPct = (v?: number) => (v == null ? " " :
6      ` ${v.toFixed(1)}% `);
7
8    const doc = new jsPDF( unit: "pt", format: "a4" );
9    const mx = 48; let y = 56;
10
11    doc.setFont("helvetica", "bold"); doc.setFontSize(14);
12    doc.text("Training Load Report", mx, y); y += 18;
13    doc.setFont("helvetica", "normal");
14    doc.text(answer.athleteName ?? `Athlete #${athleteId}`, mx, y); y +=
15      12;
16
17    const rows = [
18      ["ACWR",      fmt(answer.acwr),      human(answer.acwrLabel)],
19      ["%QW",      fmtPct(answer.pctQwUp), human(answer.pctQwUpLabel)],
20      ["Monotony", fmt(answer.monotony),
21        human(answer.monotonyLabel)],
22      ["Strain",    fmt(answer.strain),     human(answer.strainLabel)],
23    ];
24    autoTable(doc, startY: y, head: [{"Metric", "Value", "Label"}], body:
25      rows );

```

```

22 y = (doc as any).lastAutoTable.finalY + 16;
23 if (recoText)
24   doc.setFont("helvetica","bold"); doc.text("AI Recommendations",
25     mx, y); y += 12;
26   doc.setFont("helvetica","normal");
27   doc.text(doc.splitTextToSize(sanitizeRecoText(recoText), 515), mx,
28     y);
29
30 const name = (answer.athleteName ||
31   `athlete_${athleteId}`).replace(/^[a-z0-9]/gi,"_").toLowerCase();
32 doc.save(`report_${name}.pdf`);
33 };

```

Implementação 3. *Front-end* — Exportação PDF mínima com *jsPDF* + *autoTable* (métricas + recomendações).

```

1 private Mono<String> psychoCallGroqAndPersist(Long sessionId, Long
2   athleteId, PsyAlertRecoRequest req) {
3   // ... (busca coachId, validacoes, etc.)
4
5   String systemPrompt = ""
6     You are a sports performance assistant. Generate concise,
7     clear, and actionable recommendations
8     for a HUMAN athlete, focusing on recovery and psycho-emotional
9     well-being, based on scores 0 10 :
10    - sRPE, Fatigue, Soreness, Mood, Energy
11    Rules:
12    - Markdown, 3 7 bullets (1 2 sentences each), pragmatic &
13    empathetic.
14    - No medical diagnoses; suggest load reduction, sleep,
15    hydration, nutrition, recovery.
16    - Always end with: "Please seek professional medical support
17    if symptoms persist or worsen."
18
19    """;
20
21    String userPrompt = ""
22      Session/Athlete context:
23      - sessionId: %d
24      - athleteId: %d
25      Scores (0 10 ): sRPE=%d, Fatigue=%d, Soreness=%d, Mood=%d,
26      Energy=%d
27      Expected: Markdown list (3 7 bullets) + final warning line.
28    """.formatted(sessionId, athleteId,
29      req.srpe(), req.fatigue(), req.soreness(),
30      req.mood(), req.energy());
31
32    // ... (retorno da recomendação da IA)
33  }

```

Implementação 4. *Back-end* — *Prompt* para recomendações psicoemocionais.

```

1 private Mono<String> perfCallGroqAndPersist(Long sessionId, Long
2   athleteId, PerfAlertRecoRequest req) {
3   // ... (busca coachId, validacoes, etc.)
4
5   String acwrLabel = Labels.acwrLabel(d(req.acwr()));
6   String qwLabel = Labels.pctQwUpLabel(d(req.pctQwUp()));

```

```

5 String monoLabel = Labels.monotonyLabel(d(req.monotony()));
6 String strLabel = Labels.strainLabel(d(req.strain()));
7
8 String systemPrompt = """
9     You are a sports performance assistant. Produce training load
10    recommendations BY METRIC:
11    - ACWR, Weekly Load Change (%QW), Monotony (Foster), Strain
12    (Foster)
13    Output:
14    - Markdown with sections: "### ACWR", "### Weekly Load Change
15    (% QW )", "### Monotony", "### Strain"
16    - Each section: 2 3 bullets (1 2 sentences) with
17    next-session & next-week actions.
18    - No medical diagnoses. End with: "Please seek professional
19    medical support if symptoms persist or worsen."
20    Labels guide (do not restate thresholds):
21    ACWR={low|optimal|attention|risk},
22    %QW={sharp_drop|stable|attention|risk},
23    Monotony={healthy|attention|high_risk},
24    Strain={low|attention|high_risk}.
25    """;
26
27 String userPrompt = """
28     Context: sessionId=%d, athleteId=%d
29     Metrics (value + label):
30     - ACWR: %s (%s)
31     - Weekly Load Change (%QW): %s%% (%s)
32     - Monotony: %s (%s)
33     - Strain: %s (%s)
34     Expected: sections for each metric + 2 3 actionable bullets
35     + final warning line.
36     """.formatted(sessionId, athleteId,
37     req.acwr(), acwrLabel,
38     req.pctQwUp(), qwLabel,
39     req.monotony(), monoLabel,
40     req.strain(), strLabel);
41
42 // ... (retorno da recomendacao da IA)
43 }

```

Implementação 5. Back-end — Prompt para recomendações de performance por métrica (ACWR, weekly load variation, monotony e strain).

A.6. Formulários

1) Eu acho que gostaria de usar esse sistema com frequência.

 Copiar gráfico

9 respostas

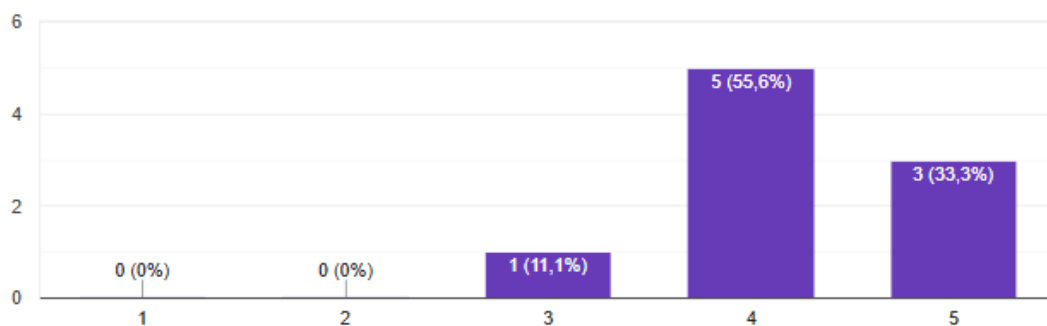


Figura 26. Resposta 1 do formulário dos atletas.

2) Eu acho o sistema desnecessariamente complexo.

 Copiar gráfico

9 respostas

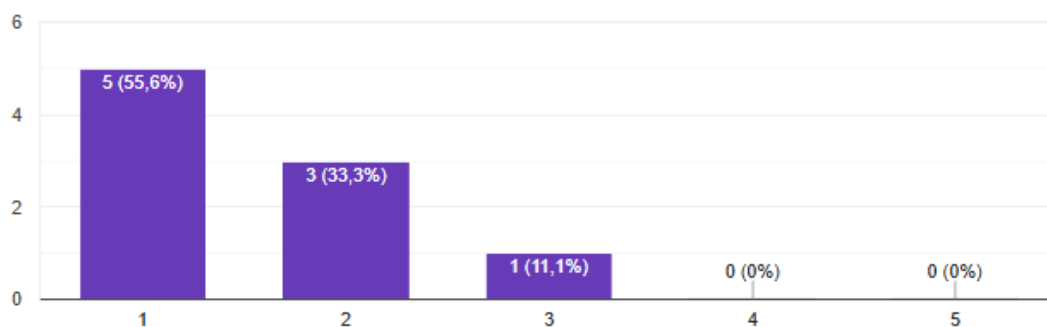


Figura 27. Resposta 2 do formulário dos atletas.

3) Eu achei o sistema fácil de usar.

 Copiar gráfico

9 respostas

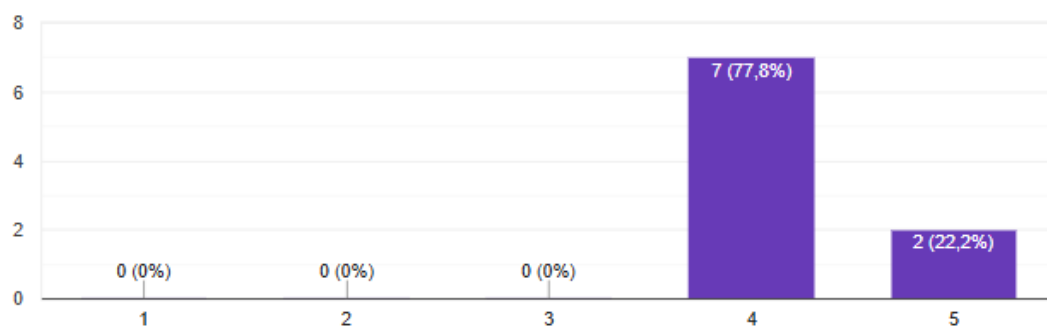


Figura 28. Resposta 3 do formulário dos atletas.

4) Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

 Copiar gráfico

9 respostas

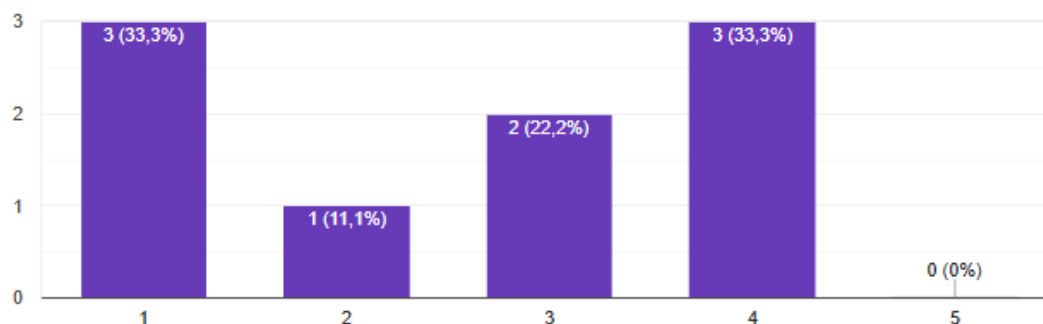


Figura 29. Resposta 4 do formulário dos atletas.

5) Eu acho que as várias funções do sistema estão muito bem integradas.

 Copiar gráfico

9 respostas

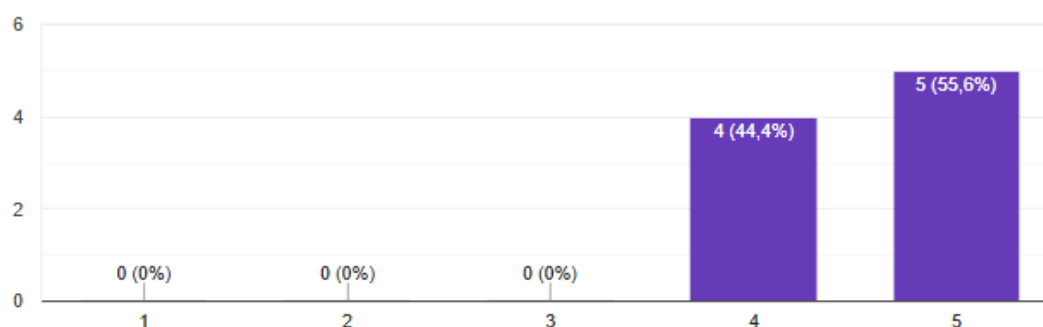


Figura 30. Resposta 5 do formulário dos atletas.

6) Eu acho que o sistema apresenta muita inconsistência.

 Copiar gráfico

9 respostas

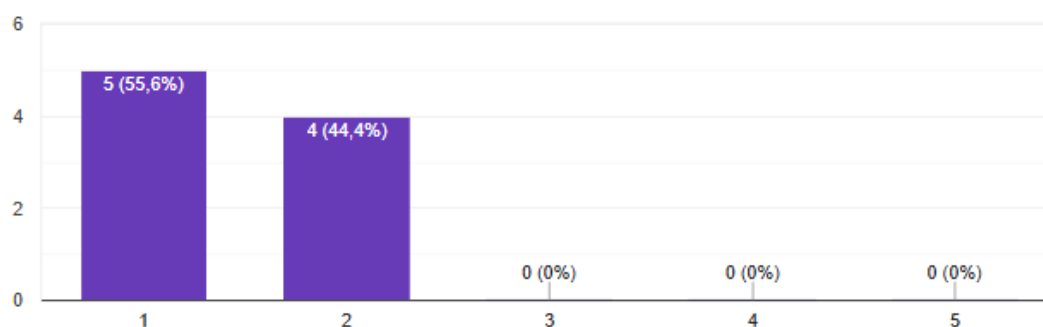



Figura 31. Resposta 6 do formulário dos atletas.

7) Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

 Copiar gráfico

9 respostas

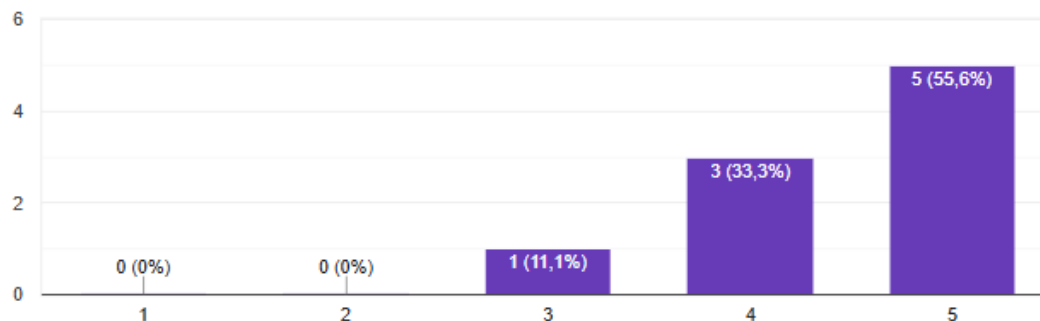


Figura 32. Resposta 7 do formulário dos atletas.

8) Eu achei o sistema atrapalhado de usar.

 Copiar gráfico

9 respostas

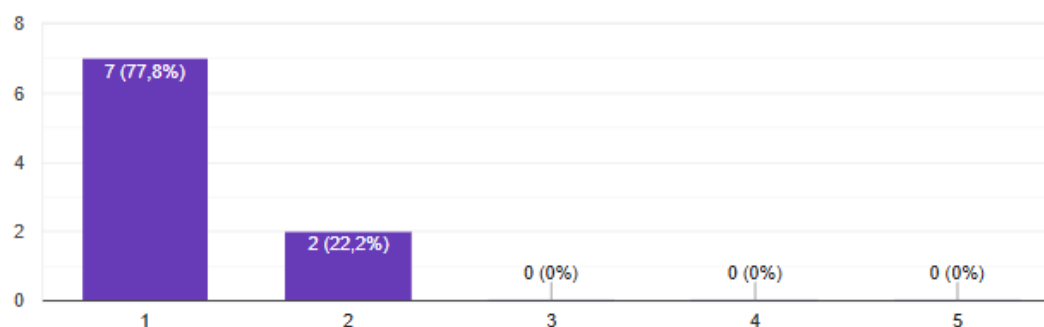


Figura 33. Resposta 8 do formulário dos atletas.

9) Eu me senti confiante ao usar o sistema.

 Copiar gráfico

9 respostas

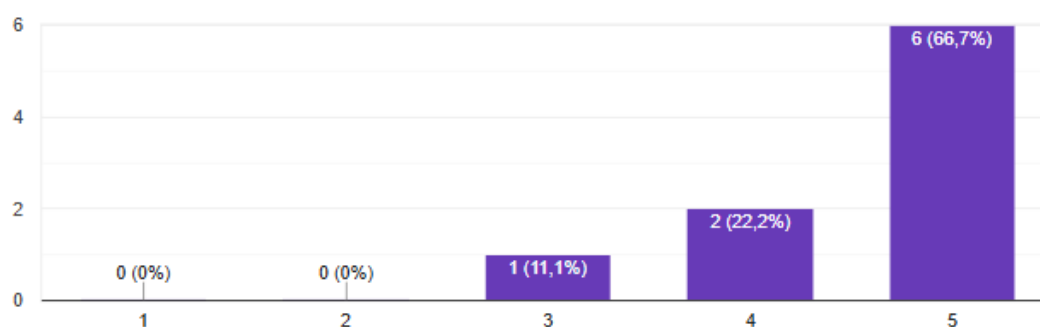


Figura 34. Resposta 9 do formulário dos atletas.

10) Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

[Copiar gráfico](#)

9 respostas

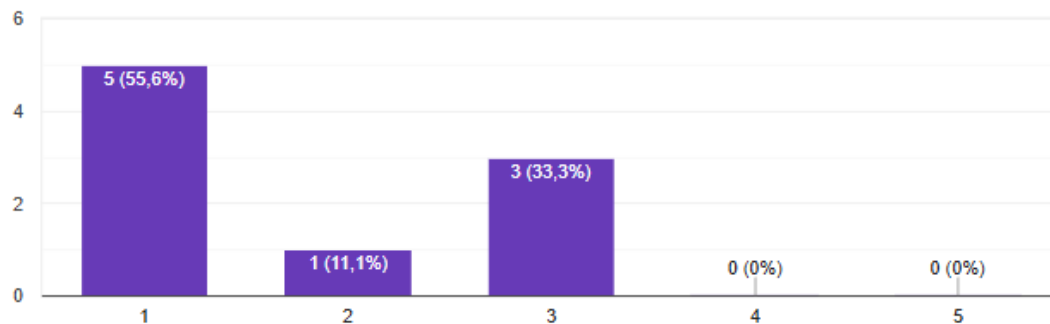


Figura 35. Resposta 10 do formulário dos atletas.

1) As recomendações fornecidas pelo sistema são confiáveis para apoiar decisões no treinamento.

[Copiar gráfico](#)

4 respostas

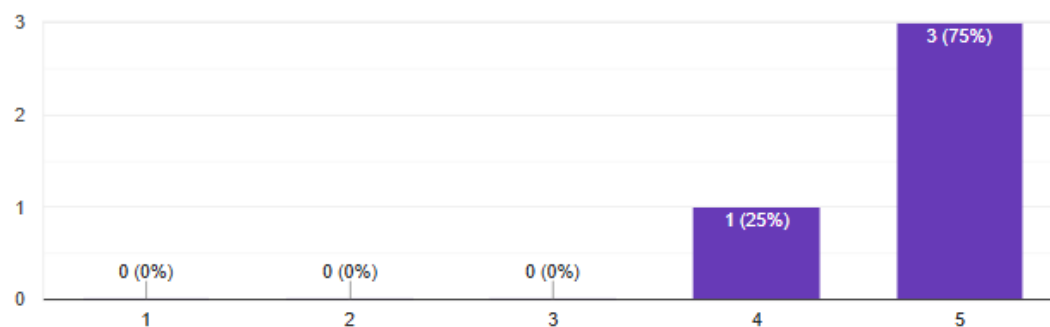


Figura 36. Resposta 1 do formulário dos profissionais.

2) As recomendações são apresentadas de forma clara e compreensível.

[Copiar gráfico](#)

4 respostas

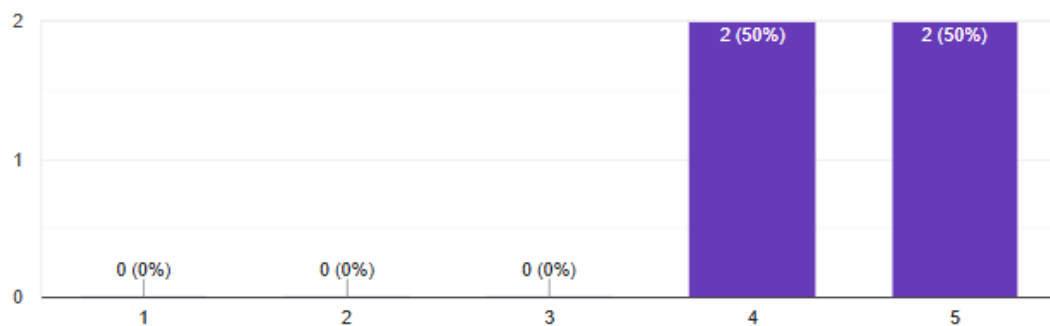


Figura 37. Resposta 2 do formulário dos profissionais.

3) As recomendações são práticas e aplicáveis no contexto do treinamento esportivo.

[Copiar gráfico](#)

4 respostas

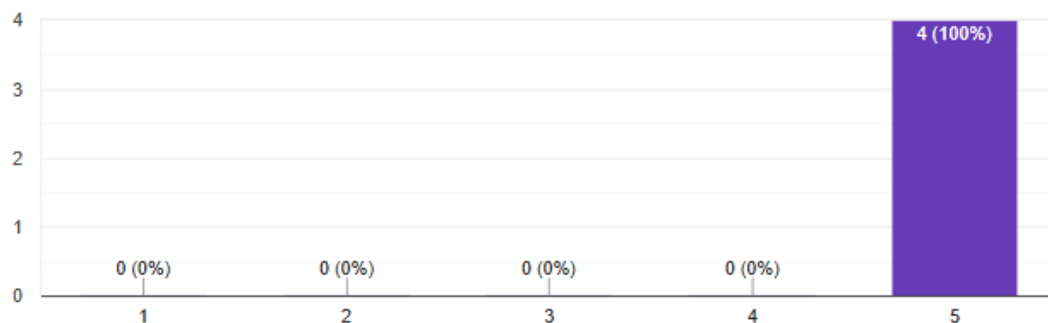


Figura 38. Resposta 3 do formulário dos profissionais.

4) As recomendações refletem de maneira coerente os indicadores e métricas apresentados pelo sistema.

[Copiar gráfico](#)

4 respostas

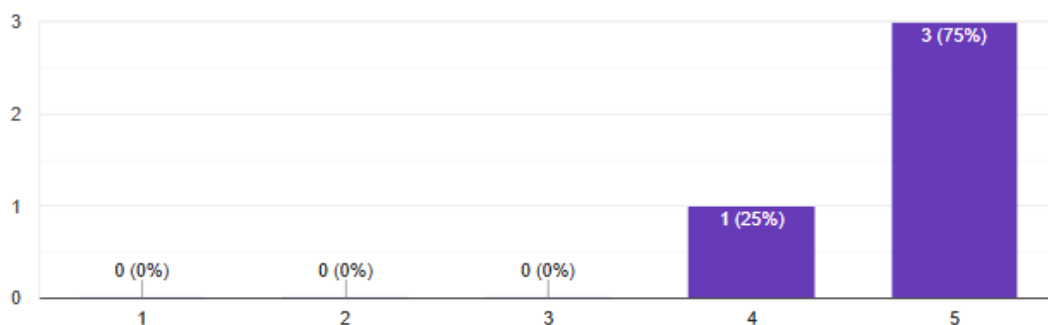


Figura 39. Resposta 4 do formulário dos profissionais.

5) O sistema permite aplicar o teste psicoemocional de forma simples e eficiente.

[Copiar gráfico](#)

4 respostas

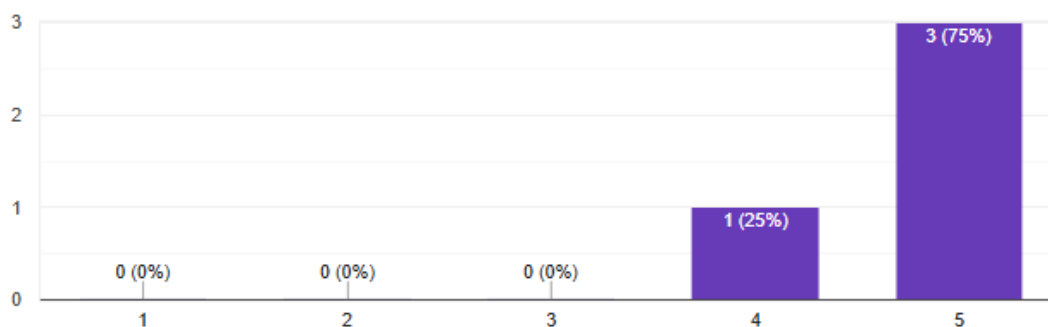


Figura 40. Resposta 5 do formulário dos profissionais.

6) As recomendações baseadas no teste psicoemocional refletem adequadamente o bem-estar do atleta.

 Copiar gráfico

4 respostas

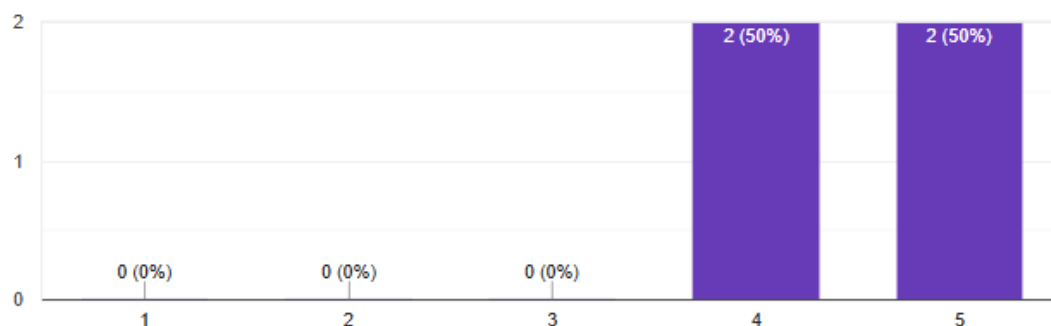


Figura 41. Resposta 6 do formulário dos profissionais.

7) O questionário psicoemocional é de fácil preenchimento para os atletas.

 Copiar gráfico

4 respostas

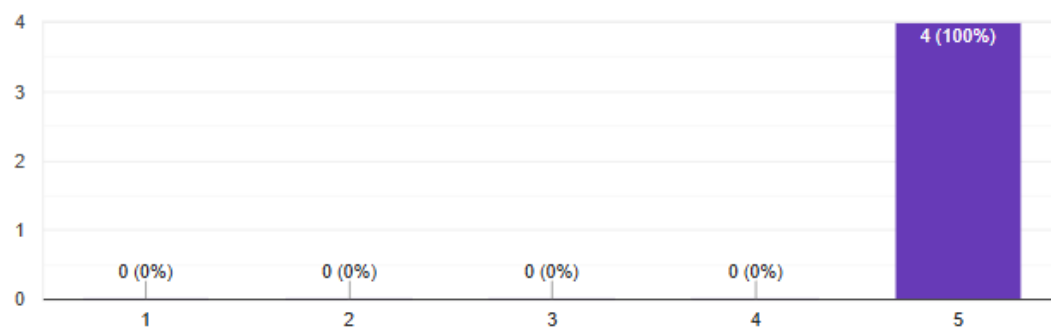


Figura 42. Resposta 7 do formulário dos profissionais.

8) O sistema é fácil de utilizar no dia a dia de um treinador.

 Copiar gráfico

4 respostas

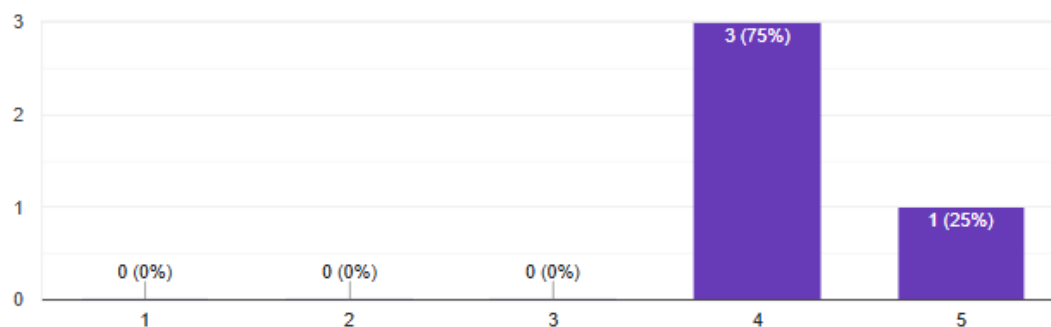


Figura 43. Resposta 8 do formulário dos profissionais.

9) O sistema organiza os dados e resultados de forma clara e acessível.

 Copiar gráfico

4 respostas

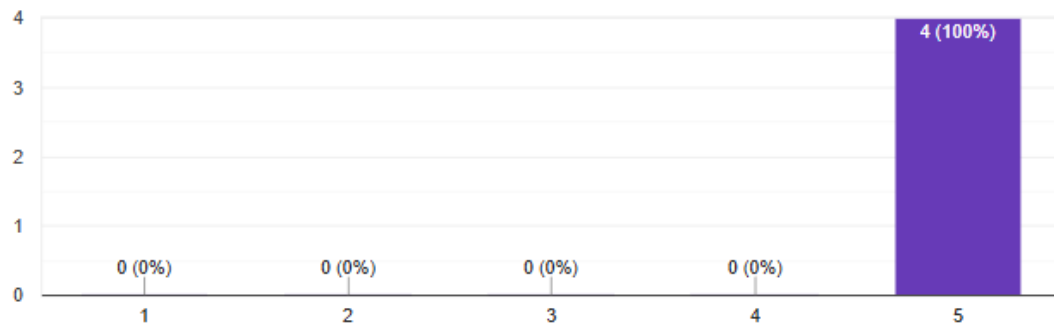


Figura 44. Resposta 9 do formulário dos profissionais.

10) O uso do sistema contribui para tomar decisões mais seguras sobre carga de treino e prevenção de lesões.

 Copiar gráfico

4 respostas

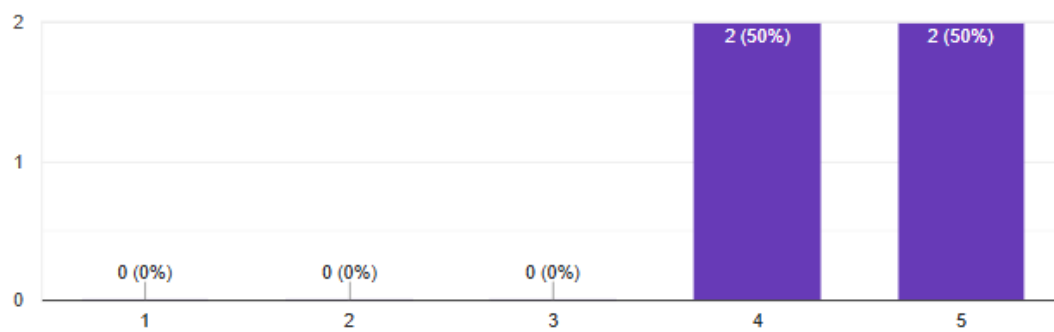


Figura 45. Resposta 10 do formulário dos profissionais.