

Carlos Eduardo Rosendo Basseto | RA: 10409941
Daniel Rubio Camargo | RA: 10408823
Gustavo Kenzo Nakazato Sleiman | RA: 10408988
João Pedro Mascaro Baccelli | RA: 10224004
Matheus Santiago de Brito | RA: 10408953

Musculum: Projeto do Laboratório Engenharia de Software

São Paulo

2025

Carlos Eduardo Rosendo Basseto | RA: 10409941
Daniel Rubio Camargo | RA: 10408823
Gustavo Kenzo Nakazato Sleiman | RA: 10408988
João Pedro Mascaro Baccelli | RA: 10224004
Matheus Santiago de Brito | RA: 10408953

Musculum: Projeto do Laboratório Engenharia de Software

Modelo de documentação acadêmica apresentado aos alunos da Universidade Presbiteriana Mackenzie.

Universidade Presbiteriana Mackenzie
Faculdade de Computação e Informática

Orientador: Prof. Dr. Gustavo Scalabrini Sampaio

São Paulo
2025

Sumário

1	INTRODUÇÃO	4
2	DEFINIÇÃO DA DEMANDA	5
2.1	Cenário de negócio / Concepção:	5
2.1.1	O problema ou oportunidade percebida;	5
2.1.2	A razão ou justificativa para esta demanda;	5
2.1.3	A descrição sucinta do produto de software que será produzido;	5
2.1.4	Identifique os clientes, usuários e demais envolvidos/impactados com o produto;	5
2.1.5	Identifique os principais critérios de qualidade para o produto.	6
2.1.6	Principais etapas necessárias para construir este produto.	6
3	REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	7
3.1	Tabela de Requisitos	7
4	FIGMA: PROTÓTIPOS	10
4.1	Link para o figma: Acesse aqui	10
5	MODELAGEM DO SISTEMA	15
5.1	Casos de Uso	15
5.1.1	Diagrama	15
5.1.2	Casos de uso resumidos	16
5.1.2.1	Realizar Login	16
5.1.2.2	Cadastrar Usuário	16
5.1.2.3	Criar Slot de Treino	16
5.1.2.4	Acessar Slot de Treino	16
5.1.2.5	Gerar Treino	16
5.1.2.6	Avaliar Treino	16
5.1.2.7	Remover Slot de Treino	16
5.1.3	Caso de Uso Crítico: UC03 - Criar Slot de Treino	17
5.2	Histórias de Usuário	20
5.2.1	Criar lista	20
5.2.1.1	Critérios de Aceitação - Criar lista:	20
5.2.2	Adicionar produto	20
5.2.2.1	Critérios de Aceitação - Adicionar produto:	20
5.2.3	Gerar Treino	20
5.2.3.1	Critérios de Aceitação - Gerar Treino:	20

5.2.4	Acessar Slot de Treino	21
5.2.4.1	Critérios de Aceitação - Acessar Slot de Treino:	21
5.2.5	Remover Slot de Treino	21
5.2.5.1	Critérios de Aceitação - Remover Slot de Treino:	21
5.2.6	Avaliar Treino	21
5.2.6.1	Critérios de Aceitação - Avaliar Treino:	22
5.3	Diagrama de Classes	23
5.4	Diagrama de Sequência	23
6	ARQUITETURA DO SISTEMA E FERRAMENTAS USADAS . . .	25
6.1	Arquitetura lógica	25
6.2	Tecnologias	26
6.2.1	Frontend	26
6.2.2	Backend	26
6.2.3	Banco de dados	27
6.2.4	Cloud	27
7	DESENVOLVIMENTO	28
8	RESULTADOS	30
8.1	Tela inicial	30
8.2	Criação de slots de treino	30
8.2.1	Visualização dos slots criados	31
8.3	Geração de treino com IA	32
8.4	Gerando treino	33
8.5	avaliar treino	34
8.6	Análise crítica do sistema desenvolvido	35
8.6.1	Também foram identificadas algumas limitações:	35
9	CONCLUSÃO	37
	REFERÊNCIAS	38

1 Introdução

Musculum é um aplicativo que auxilia na criação de um cronograma de treinos na academia, recebendo informações sobre idade, peso, altura, disponibilidade, experiência com os treinos, etc., para criar uma rotina ideal a ser seguida. A rotina será gerada por uma IA chamada Ollama, que fornecerá um treino específico para o objetivo desejado. Os treinos serão armazenados em Slots, que o usuário poderá excluir quando quiser. Será necessário o cadastro e login para acessar o site, a fim de armazenar os dados do usuário e os Slots dos treinos no banco de dados.

2 Definição da Demanda

2.1 Cenário de negócio / Concepção:

2.1.1 O problema ou oportunidade percebida;

A dificuldade de criar e manter um cronograma de treinos personalizado, adequado aos objetivos, condições físicas e disponibilidade do usuário, o que pode levar a resultados insatisfatórios ou ao abandono da prática.

2.1.2 A razão ou justificativa para esta demanda;

A demanda surge pela dificuldade das pessoas em criar treinos personalizados e eficazes, adequados às suas necessidades e objetivos. A falta de conhecimento e a falta de uma orientação contínua dificultam o progresso e aumentam o risco de desmotivação.

2.1.3 A descrição sucinta do produto de software que será produzido;

aplicativo, que gera cronogramas de treino personalizados, baseados em informações como idade, peso, altura, experiência e disponibilidade do usuário. Utilizando a IA Ollama, o sistema cria rotinas de treino específicas para os objetivos de cada pessoa, com a possibilidade de armazenar, excluir ou ajustar os treinos em Slots.

2.1.4 Identifique os clientes, usuários e demais envolvidos/impactados com o produto;

É destinado a pessoas de todas as idades, raças e gêneros, que buscam melhorar sua saúde e atingir objetivos específicos de treino, como emagrecimento, ganho de massa muscular ou condicionamento físico. A plataforma é inclusiva e personalizada, atendendo a diferentes necessidades e condições físicas de cada usuário.

2.1.5 Identifique os principais critérios de qualidade para o produto.

- Navegação Intuitiva
- Desempenho
- Segurança
- Funcionalidade
- Manutenção
- Compatibilidade
- Confiabilidade

2.1.6 Principais etapas necessárias para construir este produto.

1. Planejamento e Levantamento de Requisitos: Definir os objetivos do sistema, requisitos e tecnologias a serem utilizadas.
2. Desenho e Arquitetura: Planejar a arquitetura do sistema, o banco de dados e a interface de usuário.
3. Desenvolvimento: Implementar o front-end, back-end e integrar a IA Ollama para gerar treinos personalizados.
4. Integração e Testes: Integrar os componentes e realizar testes de funcionalidade e usabilidade.
5. Deploy e Lançamento: Preparar o ambiente de produção e realizar o deploy do sistema.
6. Manutenção e Atualizações: Corrigir erros, otimizar a IA e adicionar novas funcionalidades com base no feedback dos usuários.

3 Requisitos Funcionais e Não Funcionais

3.1 Tabela de Requisitos

Abaixo está a tabela com os requisitos funcionais e não funcionais do projeto.

Tabela 1 – Tabela de requisitos funcionais e não funcionais ordenados por prioridade.

Prioridade	Descrição do Requisito	Identificador
1	Cadastro de usuário (Nome, Data de nascimento, Email, Senha)	RF01
1	Realizar login com validação das credenciais	RF02
1	Criar lista (nome da lista)	RF03
1	Adicionar produto em uma lista (nome, descrição, preço, prioridade, categoria, links)	RF04
1	Remover produto de uma lista	RF05
2	Editar informações de produto já adicionado (nome, descrição, preço, prioridade, categoria, links)	RF06
2	Abrir perfil para visualizar informações detalhadas	RF07
2	Abrir lista para visualizar seu conteúdo	RF08
2	Editar nome ou nome de uma lista existente	RF09
2	Remover uma lista criada	RF10
3	Ordenar produtos de uma lista conforme preferências do usuário	RF11
3	Abrir perfil para visualizar informações do usuário	RF12
3	Editar informações do perfil do usuário	RF13
3	Interface limpa, visualmente agradável	RNF01
3	Sistema intuitivo e fácil de usar	RNF02
3	Fluidez e boa performance na integração	RNF03
3	Estabilidade no funcionamento do sistema	RNF04

Nota: RF = requisitos funcionais e RNF = requisitos não funcionais

RF01: O usuário pode se cadastrar no sistema informando nome, data de nascimento, e-mail e senha.

O sistema deve armazenar essas informações com segurança e criar um novo perfil de usuário.

RF02: O usuário pode realizar login no sistema utilizando suas credenciais.

O sistema deve validar as credenciais fornecidas e permitir o acesso caso estejam corretas.

RF03: O usuário pode criar uma nova lista informando apenas o nome da lista.
O sistema deve registrar essa lista associando-a ao perfil do usuário.

RF04: O usuário pode adicionar produtos em uma lista informando nome, descrição, preço, prioridade, categoria e links.
O sistema deve registrar cada produto na lista especificada, com todas as informações fornecidas.

RF05: O usuário pode remover um produto previamente adicionado de uma lista.
O sistema deve excluir esse produto da lista correspondente.

RF06: O usuário pode editar as informações de um produto já adicionado à lista.
O sistema deve atualizar os dados do produto conforme as alterações feitas (nome, descrição, preço, prioridade, categoria e links).

RF07: O usuário pode acessar seu perfil para visualizar informações detalhadas.
O sistema deve apresentar os dados completos do perfil, como nome, e-mail, listas criadas, entre outros.

RF08: O usuário pode abrir uma lista para visualizar todo o seu conteúdo.
O sistema deve exibir todos os produtos contidos na lista selecionada.

RF09: O usuário pode editar o nome de uma lista existente.
O sistema deve permitir a alteração do nome e atualizar o dado no banco de dados.

RF10: O usuário pode remover uma lista que tenha sido previamente criada.
O sistema deve excluir a lista e todos os seus conteúdos associados.

RF11: O usuário pode ordenar os produtos de uma lista conforme suas preferências (ex: por preço, prioridade ou categoria).
O sistema deve aplicar o critério de ordenação selecionado e reorganizar a exibição dos itens.

RF12: O usuário pode visualizar novamente seu perfil com foco em informações pessoais.
O sistema deve fornecer acesso ao perfil com informações resumidas de conta e preferências.

RF13: O usuário pode editar as informações do seu próprio perfil.
O sistema deve permitir a atualização de dados como nome, e-mail e senha.

RNF01: O sistema deve apresentar uma interface limpa e visualmente agradável para facilitar a navegação.

RNF02: O sistema deve ser intuitivo e fácil de usar, mesmo para usuários com pouca experiência técnica.

RNF03: O sistema deve garantir fluidez e boa performance na integração das

funcionalidades.

RNF04: O sistema deve manter estabilidade durante seu funcionamento, evitando travamentos e falhas inesperadas.

4 Figma: Protótipos

4.1 Link para o figma: Acesse aqui

Na Figura 1, é apresentada a tela inicial do sistema, responsável por oferecer ao usuário as opções de login ou registro. Seu principal objetivo é permitir o acesso seguro à plataforma, direcionando o usuário conforme seu status: se já possui uma conta, pode clicar em "Sign in" para realizar o login; caso contrário, pode clicar em "Register" para iniciar o processo de cadastro. Essa etapa é fundamental para garantir que apenas usuários autenticados possam utilizar os recursos do sistema. A interface foi projetada de forma simples e objetiva, visando uma experiência de uso clara e acessível desde o primeiro contato.

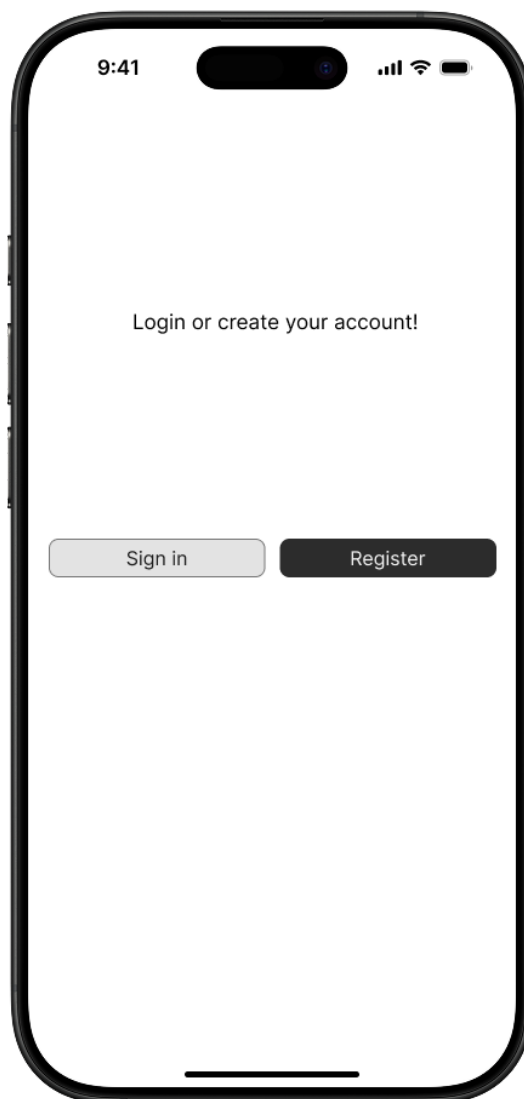


Figura 1 – Tela de login/registro do usuário.

Na Figura 2, é exibida a tela inicial do aplicativo após o usuário realizar o login. Essa interface tem como objetivo principal servir de ponto de partida para a utilização das funcionalidades centrais do sistema. No centro da tela, há um botão de destaque com a opção "Make a WorkOut", que leva o usuário ao processo de geração de treino. Esse botão representa o principal fluxo de uso da aplicação, permitindo que o usuário inicie rapidamente a criação de um plano de treino personalizado. A interface é minimalista, focando na simplicidade e na clareza da ação esperada, garantindo uma navegação direta e intuitiva.

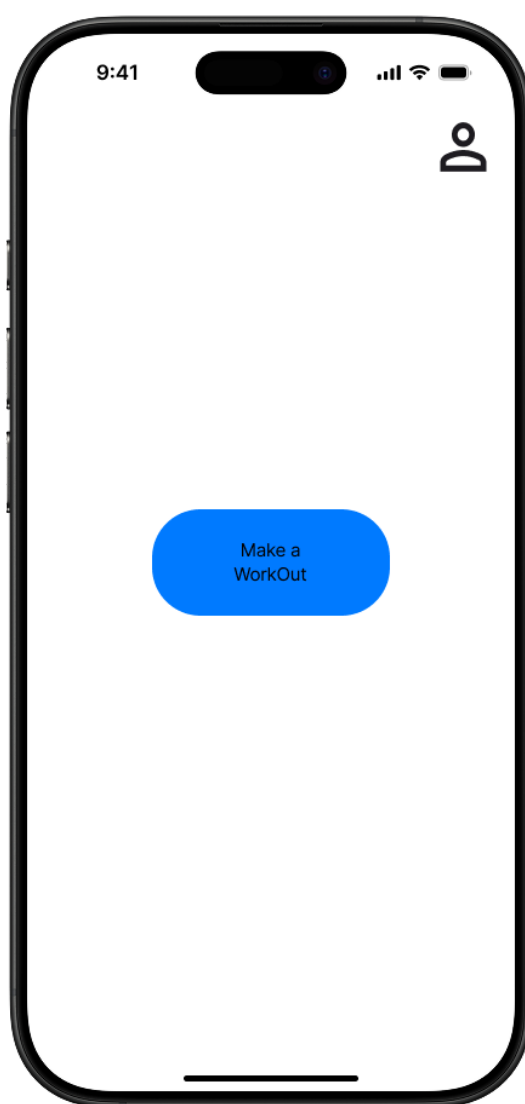


Figura 2 – Tela inicial do app.

Na Figura 3, é apresentada a tela de preenchimento de dados pessoais, necessária para a geração de um treino personalizado. Essa interface tem como objetivo coletar informações relevantes sobre o usuário, como sexo, idade, altura, peso, objetivo com os treinos, frequência semanal, experiência anterior, possíveis restrições médicas e outros dados adicionais. Todas essas informações serão utilizadas pela inteligência artificial do

sistema para montar um plano de treino alinhado ao perfil e às necessidades do usuário. A tela é estruturada de forma clara e orientada ao preenchimento, facilitando a usabilidade e garantindo a coleta completa dos dados essenciais para a recomendação eficiente dos exercícios.

9:41

Complete below to generate your personalized workout!

Sex: Value

Age: Value

Height: Value

Weight: Value

What is your goal?: Value

How many days a week will you train?: Value

Have you trained before?: Value

Do you have medical restrictions?: Value

Any additional information?: Value

Figura 3 – Registro de informações de usuários para criação de treinos.

Na Figura 4, observa-se a tela onde a inteligência artificial do sistema apresenta ao usuário um treino personalizado, com base nas informações fornecidas anteriormente. A IA lista os exercícios sugeridos, incluindo nome, número de séries e repetições. Abaixo da sugestão, há uma pergunta automatizada solicitando o feedback do usuário: “Are you satisfied with this workout? If not, what changes would you like to make?”, permitindo que o usuário confirme sua satisfação ou solicite ajustes. Essa etapa torna o processo interativo e adaptativo, garantindo que o treino esteja alinhado às expectativas e preferências do usuário antes de ser armazenado no sistema.

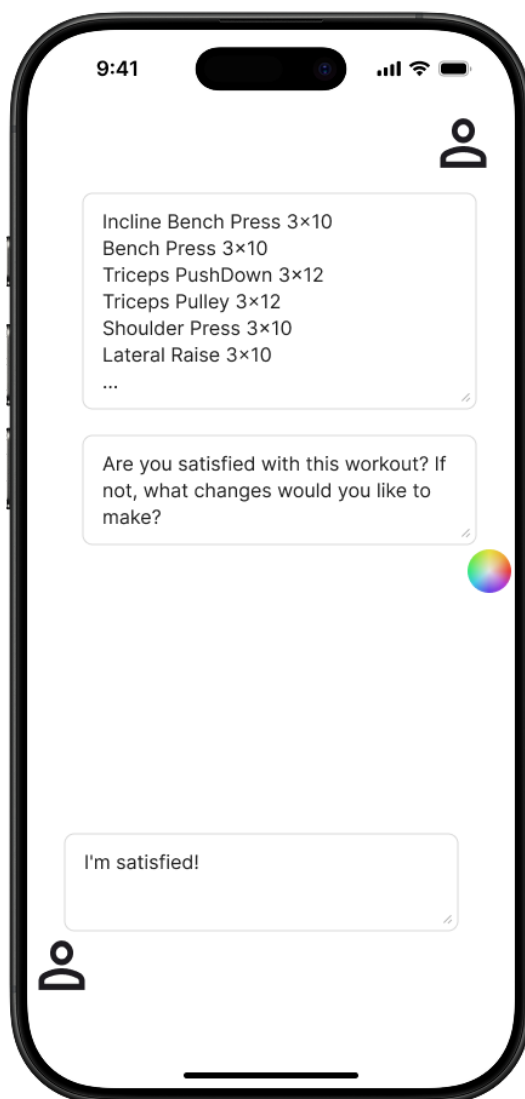


Figura 4 – Ia gerando treino para o usuário.

Na Figura 5, é apresentada a tela que exibe o treino já criado e salvo para o usuário. Nela, é possível visualizar os exercícios prescritos pela IA de forma organizada por dias — no exemplo, o treino corresponde ao Dia 1. Cada exercício é acompanhado de uma ilustração anatômica e sua respectiva descrição com quantidade de séries e repetições, como “Incline bench press 3x10” e “Bench press 3x10”. Na parte superior, o usuário pode alternar entre os modos “WorkOuts Slots” e “IA Trainer”, enquanto na parte inferior há uma navegação por slots numerados, indicando diferentes treinos salvos. Essa tela oferece uma visão clara e prática do plano de treino, permitindo ao usuário acompanhar e seguir suas rotinas de forma estruturada.

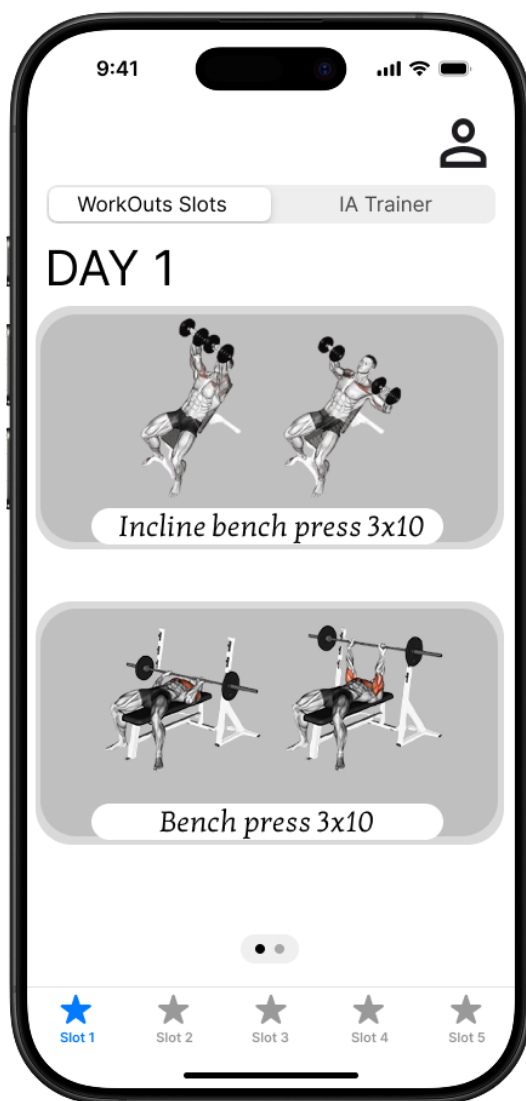


Figura 5 – Tela do treino criado.

5 Modelagem do sistema

5.1 Casos de Uso

5.1.1 Diagrama

O diagrama abaixo, representa as principais funcionalidades do Sistema de Recomendação de Treinos. O ator Usuário pode se cadastrar, realizar login, criar e acessar slots de treino. Ao criar um slot, o sistema inclui automaticamente a geração de treino pela IA, responsável por personalizar os treinos. Ao acessar um slot, o usuário pode opcionalmente avaliá-lo ou removê-lo. A interação entre usuário e sistema é complementada por relacionamentos de inclusão («include») e extensão («extend»), representando ações obrigatórias e opcionais no fluxo de uso.

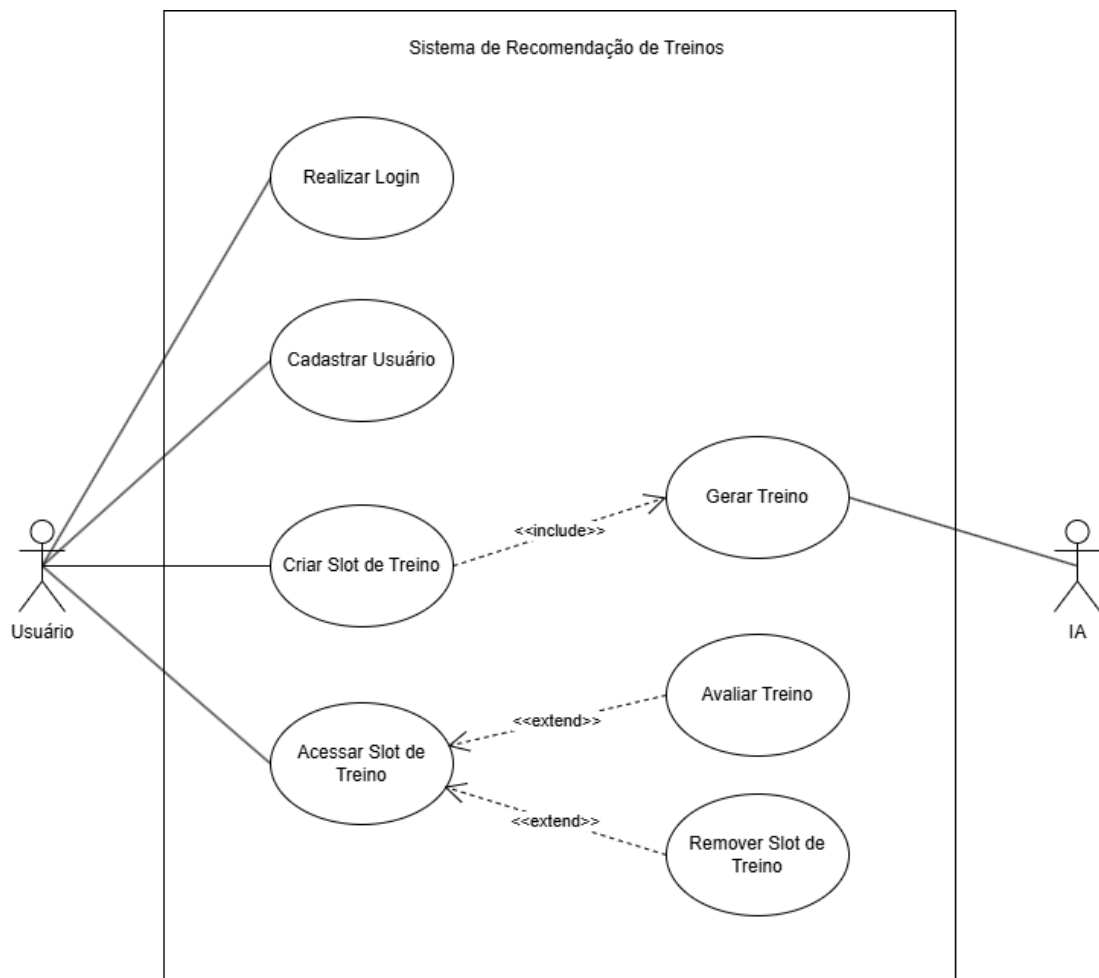


Figura 6 – Diagrama de Casos de Uso.

5.1.2 Casos de uso resumidos

5.1.2.1 Realizar Login

Descrição: O **usuário** fornece suas credenciais (e-mail e senha). O **sistema** valida as informações e, se estiverem corretas, autentica o usuário e libera o acesso à plataforma.

5.1.2.2 Cadastrar Usuário

Descrição: O **usuário** fornece informações pessoais como nome, e-mail, data de nascimento e senha. O **sistema** valida os dados e cria uma nova conta de usuário.

5.1.2.3 Criar Slot de Treino

Descrição: O **usuário** seleciona um horário e define os parâmetros de um novo slot de treino. O **sistema** registra o slot e associa-o ao perfil do usuário.

5.1.2.4 Acessar Slot de Treino

Descrição: O **usuário** acessa os slots de treino previamente criados. O **sistema** exibe os detalhes do treino, como horário, exercícios e ajustes possíveis.

5.1.2.5 Gerar Treino

Descrição: O **usuário** informa dados pessoais (idade, peso, altura, disponibilidade, objetivos etc.). O **sistema**, por meio da IA Ollama, gera um treino personalizado com base nas informações fornecidas.

5.1.2.6 Avaliar Treino

Descrição: O **usuário** avalia o treino recebido, informando sua experiência. O **sistema** registra a avaliação e pode utilizá-la para ajustar treinos futuros.

5.1.2.7 Remover Slot de Treino

Descrição: O **usuário** seleciona um slot de treino para excluir. O **sistema** remove o slot da agenda e atualiza as informações do usuário.

5.1.3 Caso de Uso Crítico: UC03 - Criar Slot de Treino

Na Figura 7, O usuário estando logado no sistema, acessa a página de slots de treino e seleciona a opção para criar um novo slot. Esse processo é iniciado com a navegação até a página de slots, onde o sistema exibe a tela com as opções disponíveis.

- UC03

Especificação de caso de uso - CASO DE USO CRÍTICO	
Identificador	UC03
Nome	Criar Slot de Treino
Atores	Usuário
Sumário	O usuário cria um Slot de treino, ficando armazenado em sua conta.
Pré-Condição	O usuário deve estar logado no aplicativo.
Pós-Condição	Um novo Slot de treino é criado e salvo na conta do usuário.
Pontos de Inclusão	-

Pontos de Extensão -

Fluxo Principal (UC03) - CASO DE USO CRÍTICO	
Ações do Ator	Ações do Sistema

Figura 7 – Especificação do Caso de Uso UC03 – Criar Slot de Treino

Na Figura 8, é detalhado o fluxo principal do caso de uso: o sistema solicita informações como idade, peso, altura, disponibilidade e experiência. O usuário preenche os dados, e o sistema valida as informações e as envia à IA.

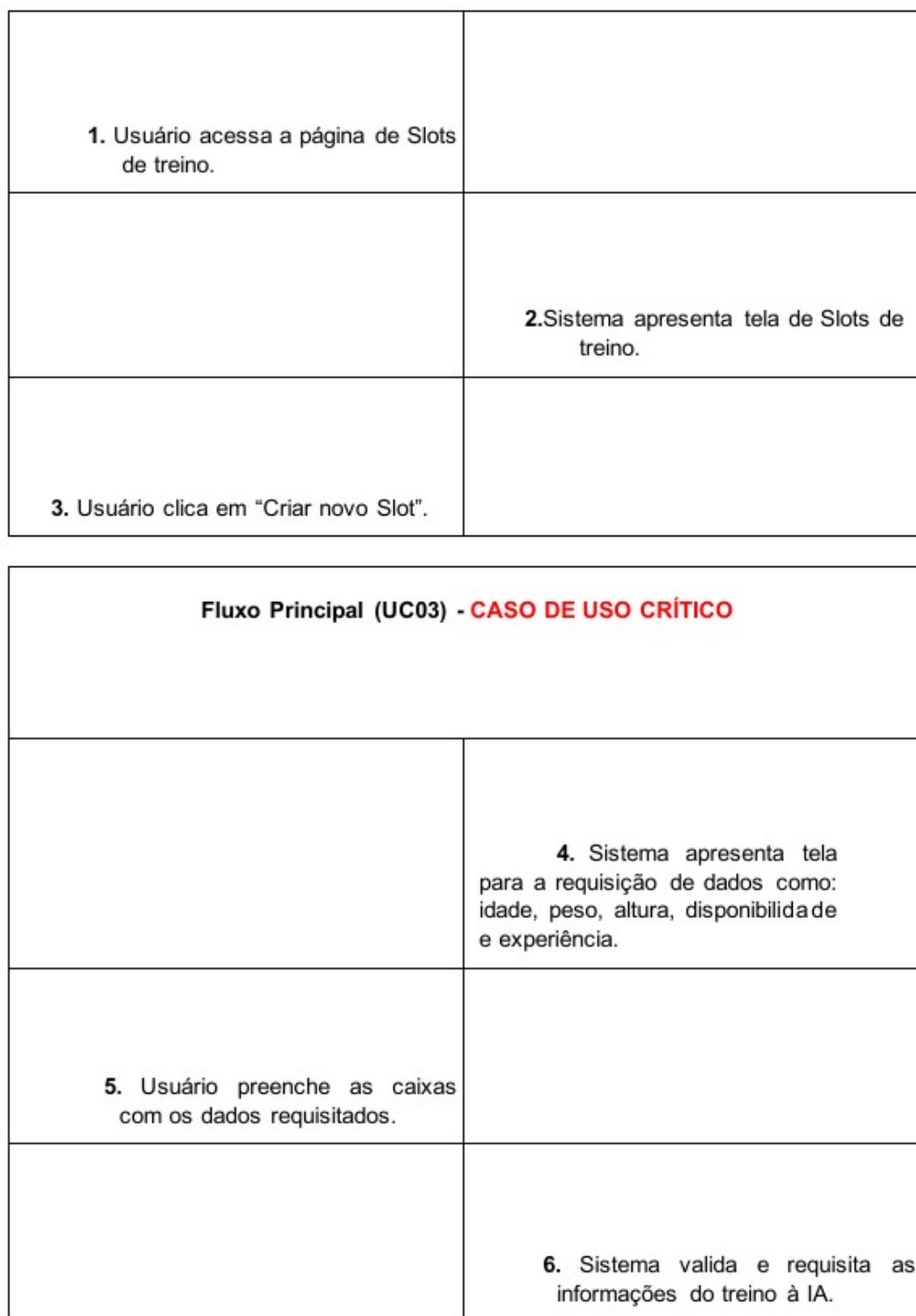


Figura 8 – Fluxo principal do Caso de Uso UC03 – Parte 1

Na Figura 9, se os dados inseridos forem inválidos ou sem sentido, o sistema identifica o erro, notifica o usuário e solicita uma nova tentativa de preenchimento, redirecionando-o

para a etapa correta do fluxo principal.

	7. Recebidas e validadas as informações do treino, o sistema registra novo Slot de treino na conta do usuário.
--	---

Fluxo de Exceção 1 (UC03) - CASO DE USO CRÍTICO	
5.1. Usuário preenche as caixas com dados sem sentido.	
	5.2. Sistema verifica os dados sem sentido.
	5.3. Sistema envia mensagens de erro ao usuário e solicita os dados novamente.
	5.4. Sistema direciona usuário para a etapa 5 do Fluxo Principal.

Figura 9 – Fluxo principal do Caso de Uso UC03 – Parte 2

5.2 Histórias de Usuário

5.2.1 Criar lista

No papel de usuário desejo criar uma nova lista para adicionar produtos.
Eu gostaria de criar uma nova lista para adicionar produtos à minha plataforma.
Assim eu poderia organizar meus itens de forma personalizada.

5.2.1.1 Critérios de Aceitação - Criar lista:

- O sistema exibe uma tela de formulário para o usuário preencher com o seguinte campo: nome da lista.
- O sistema confirma a criação da lista.
- O sistema salva a lista criada e ela fica visível no campo “minhas listas”.

5.2.2 Adicionar produto

No papel de usuário, desejo adicionar um novo produto em uma das minhas listas criadas.
Eu gostaria de adicionar um produto nas listas já criadas na plataforma.
Assim eu poderia organizar meus produtos de forma eficiente e personalizada.

5.2.2.1 Critérios de Aceitação - Adicionar produto:

- O sistema exibe uma tela de formulário para o usuário preencher com os seguintes campos: nome, descrição, preço, prioridade, categoria, links.
- O sistema autentica as informações preenchidas.
- O sistema confirma a adição do produto.

5.2.3 Gerar Treino

No papel de usuário que deseja um treino personalizado.
Eu gostaria que o sistema gerasse um treino baseado nos meus dados e objetivos.
Assim eu poderia seguir um plano eficiente para alcançar minhas metas.

5.2.3.1 Critérios de Aceitação - Gerar Treino:

- O sistema deve enviar os dados fornecidos pelo usuário à IA.
- A IA deve gerar um treino personalizado com base nas informações fornecidas.

- O sistema deve receber e validar as informações do treino.
- O sistema deve armazenar o treino gerado dentro do Slot selecionado pelo usuário.

5.2.4 Acessar Slot de Treino

No papel de usuário que deseja acompanhar seu progresso.

Eu gostaria de acessar um Slot de treino específico.

Assim eu poderia visualizar meu plano de treino personalizado.

5.2.4.1 Critérios de Aceitação - Acessar Slot de Treino:

- O sistema deve apresentar a página de Slots de treino ao usuário.
- O sistema deve listar todos os Slots criados.
- O sistema deve permitir que o usuário clique em um Slot específico.
- O sistema deve redirecionar o usuário para a página do Slot selecionado, exibindo as informações do treino.

5.2.5 Remover Slot de Treino

No papel de usuário que deseja reorganizar seus treinos.

Eu gostaria de remover um Slot de treino que não preciso mais.

Assim eu poderia manter minha lista de treinos organizada e relevante.

5.2.5.1 Critérios de Aceitação - Remover Slot de Treino:

- O sistema deve apresentar a página de Slots de treino ao usuário.
- O sistema deve permitir que o usuário selecione um Slot específico.
- O sistema deve exibir a opção "Deletar Slot" dentro de um menu de opções.
- O sistema deve solicitar confirmação antes de excluir o Slot.
- O sistema deve remover permanentemente o Slot da conta do usuário após a confirmação.

5.2.6 Avaliar Treino

No papel de usuário que realizou um treino.

Eu gostaria de avaliar meu treino após completá-lo.

Assim eu poderia registrar minha opinião e ajudar a melhorar futuros treinos.

5.2.6.1 Critérios de Aceitação - Avaliar Treino:

- O sistema deve apresentar a página de Slots de treino ao usuário.
- O sistema deve permitir que o usuário selecione um Slot específico.
- O sistema deve disponibilizar a opção “Avaliar” dentro de um menu de opções.
- O sistema deve exibir uma página para que o usuário insira sua avaliação.
- O sistema deve registrar e armazenar a avaliação enviada pelo usuário.

5.3 Diagrama de Classes

Na Figura 10, o diagrama de classes representa a estrutura principal do sistema. O Usuário cria um ou mais Slots, que armazenam Treinos gerados pela IA. Cada treino está associado a um único slot. O usuário também realiza Avaliações vinculadas aos slots, permitindo analisar os treinos executados. A IA, além de gerar os treinos, também pode atualizá-los. As associações refletem a interação entre os elementos centrais do sistema: usuário, treino, slot, avaliação e IA.

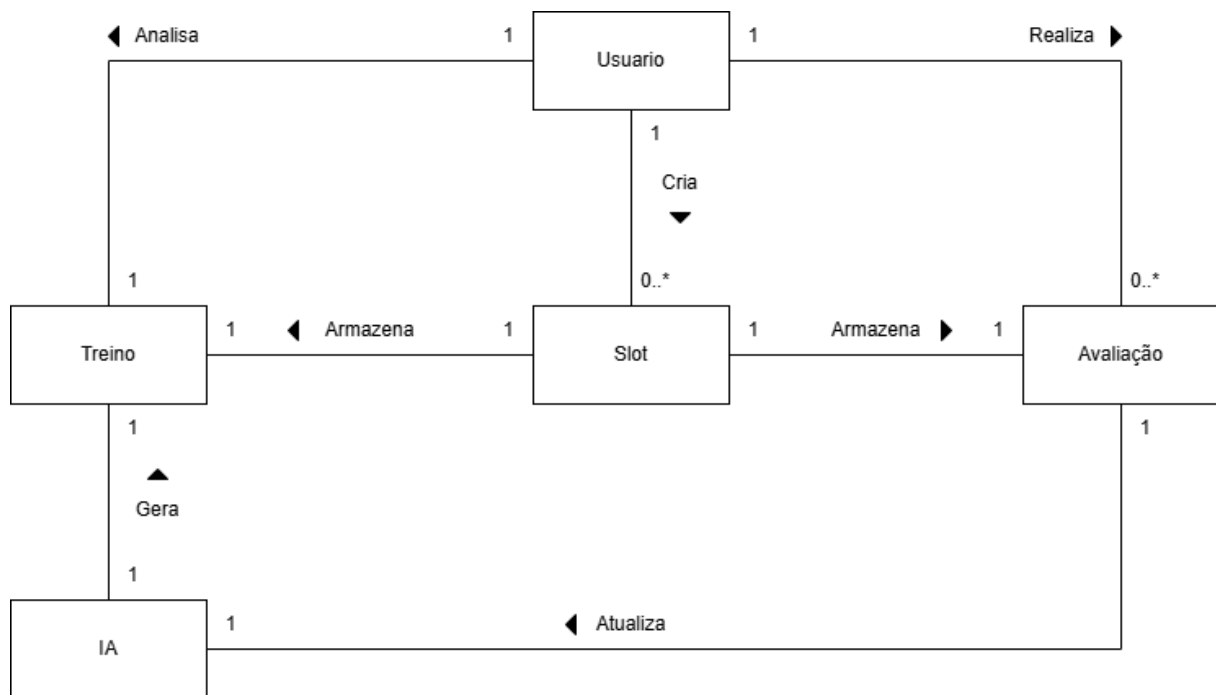


Figura 10 – Diagrama de Classe de Domínio

5.4 Diagrama de Sequência

Na Figura 11, o diagrama de sequência demonstra o fluxo de interação para a criação de um Slot de treino. O processo se inicia quando o usuário aciona a tela para criar o slot, que coleta os dados necessários. Caso os dados estejam inválidos, o sistema solicita uma nova entrada. Em seguida, os dados são enviados à IA, que gera o treino. O treino é então associado ao slot, e o sistema realiza o armazenamento no banco de dados. Por fim, é exibida uma mensagem de sucesso e a tela de slots é atualizada.

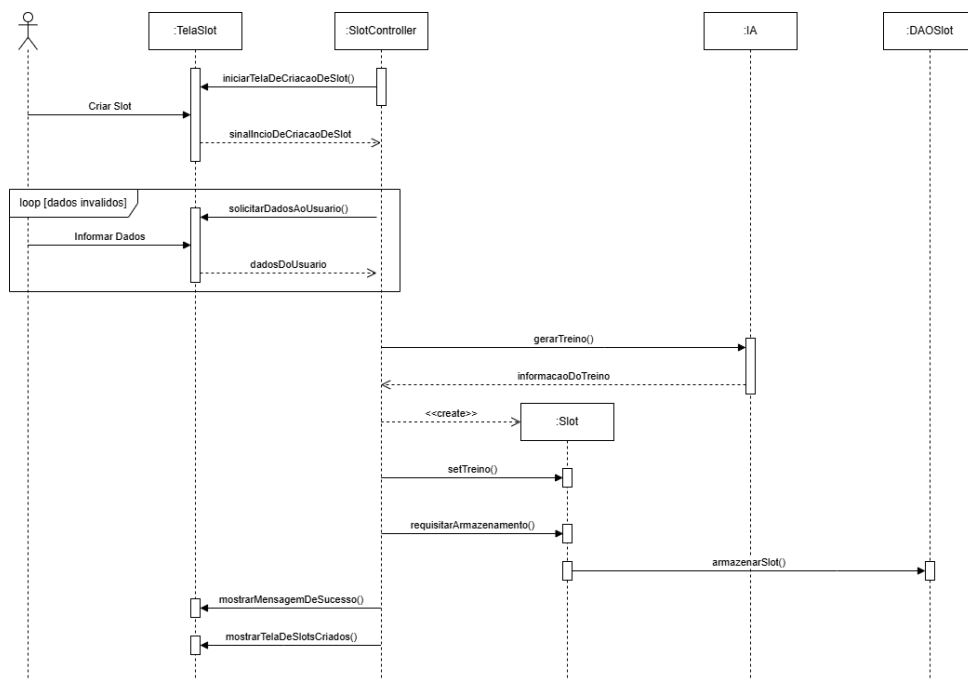


Figura 11 – Diagrama de Sequência

6 Arquitetura do sistema e ferramentas usadas

6.1 Arquitetura lógica

Na Figura 12, o diagrama de pacotes organiza a arquitetura do sistema em quatro camadas principais: Apresentação, Aplicação, Negócio e Persistência. A camada de apresentação contém as interfaces gráficas do sistema, como as telas de login, cronograma e slot de treino. Essas interfaces se comunicam com a camada de aplicação, responsável pelo controle da lógica por meio dos módulos `ControladorDeAutenticação` e `ControleDeTreino`. A camada de negócio centraliza as classes que representam as entidades do sistema: `Usuário`, `Treino` e `Slot`. Por fim, a camada de persistência contém os componentes responsáveis pelo acesso aos dados (`DAOUsuario`, `DAOTreino` e `DAOSlot`), que garantem o armazenamento e a recuperação das informações no banco de dados.

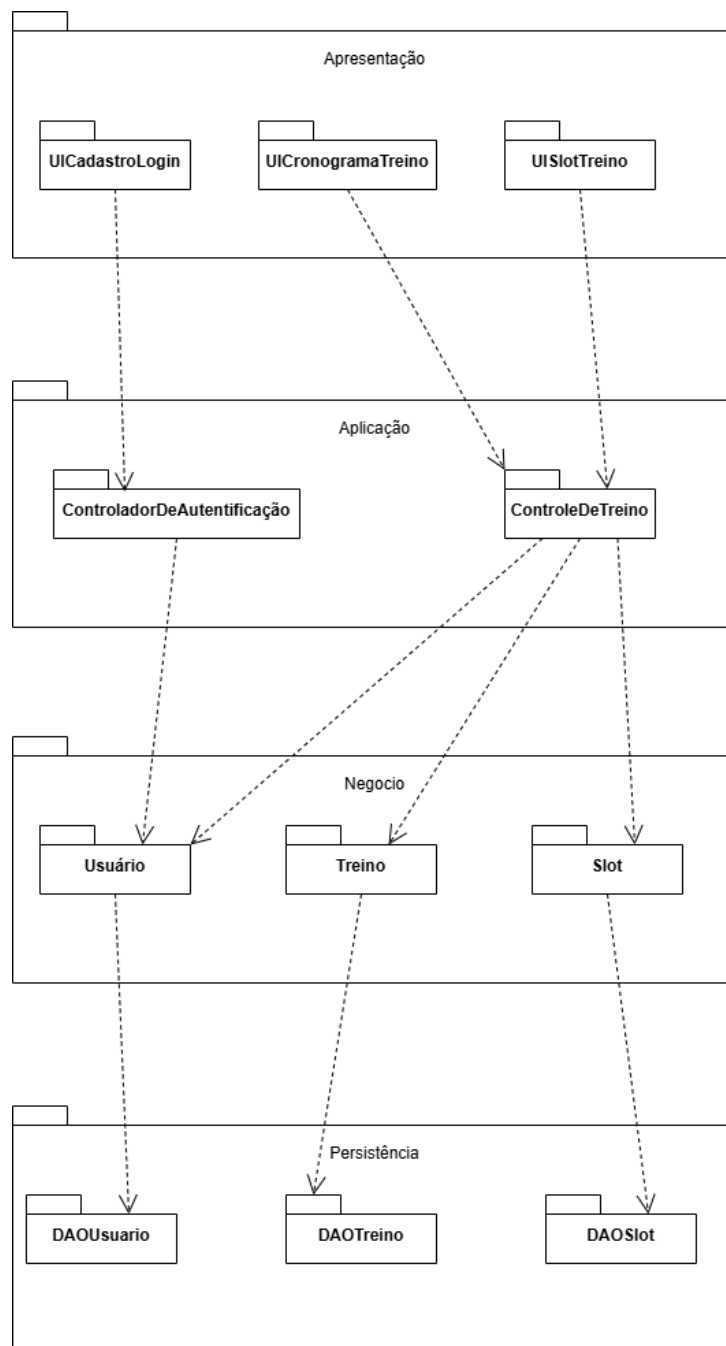


Figura 12 – Diagrama de Pacotes

6.2 Tecnologias

6.2.1 Frontend

- React.js

6.2.2 Backend

- Java Spring Boot

- Java Persistence API

6.2.3 Banco de dados

- MySQL

6.2.4 Cloud

- AWS: Amazon EC2

A implementação do sistema Musculum será realizada com o uso de tecnologias modernas, organizadas em quatro principais camadas: frontend, backend, banco de dados e infraestrutura em nuvem.

No frontend, será utilizado o framework React.js, que permitirá o desenvolvimento de uma interface interativa, responsiva e de fácil uso para o usuário final.

Para o backend, será adotado o Java Spring Boot, proporcionando uma estrutura robusta para o desenvolvimento das regras de negócio e das APIs da aplicação. A Java Persistence API (JPA) será utilizada para facilitar a comunicação entre as classes de negócio e o banco de dados.

O banco de dados utilizado será o MySQL, que oferecerá uma solução relacional confiável e amplamente compatível com as tecnologias propostas.

A aplicação será hospedada na nuvem da AWS (Amazon Web Services), utilizando o serviço EC2 para garantir escalabilidade, disponibilidade e facilidade no gerenciamento da infraestrutura do sistema.

7 Desenvolvimento

O desenvolvimento do sistema Musculum foi conduzido com o objetivo de oferecer uma plataforma acessível, intuitiva e personalizada para a criação de treinos físicos com suporte de inteligência artificial. Para isso, adotou-se uma arquitetura baseada no modelo cliente-servidor, com o uso de tecnologias modernas tanto no front-end quanto no back-end. A solução também conta com persistência em banco de dados relacional e geração de treinos via um modelo de linguagem local utilizando a ferramenta Ollama.

No front-end, utilizou-se a biblioteca React.js, que permite o desenvolvimento de interfaces dinâmicas e reutilizáveis por meio da composição de componentes. A ferramenta Vite foi empregada como bundler, por oferecer tempos de compilação rápidos e suporte nativo ao React. A navegação entre as páginas foi implementada com a biblioteca React Router DOM, e a estilização foi feita com CSS modularizado, onde cada página possui seu próprio arquivo de estilos, promovendo a organização e a separação de responsabilidades.

O back-end foi construído com Node.js, utilizando o framework Express para criação de rotas e manipulação de requisições HTTP. A comunicação entre o front-end e o back-end foi habilitada com o middleware CORS, permitindo que a aplicação cliente (rodando na porta 5173) realizasse chamadas para o servidor (porta 3000). O servidor também se comunica com a ferramenta Ollama local, que executa o modelo Mistral para gerar treinos personalizados com base nas informações fornecidas pelo usuário. As chamadas à IA são realizadas via requisições HTTP utilizando a biblioteca Axios.

O sistema foi estruturado de forma modular. No front-end, as páginas principais foram divididas da seguinte forma: Home.jsx representa a tela inicial, com botões de navegação; Slots.jsx gerencia os treinos criados, permitindo sua visualização, avaliação ou exclusão; e EntrarSlots.jsx oferece um formulário para geração personalizada de treino. Cada uma dessas páginas está acompanhada por seu respectivo arquivo de estilo (.css), mantendo o projeto organizado e facilitando futuras manutenções.

No back-end, o arquivo principal server.js concentra as rotas de API. Dentre as rotas implementadas, destacam-se: POST /criar-slot, responsável por gerar um novo treino baseado nos dados enviados pelo front-end; GET /slot/:id, que retorna o treino associado ao ID fornecido; e GET /api/testar-banco, que testa a conexão com o banco de dados MySQL. A função gerarTreinoOllama(prompt) encapsula a lógica de chamada à IA, centralizando o comportamento e permitindo reutilização. O armazenamento dos treinos atualmente é feito em memória, através de um objeto chamado slots, mas a estrutura do banco de dados já está preparada para suportar persistência.

O banco de dados foi modelado com duas tabelas principais. A tabela Usuario

armazena informações como nome, idade, peso, altura, disponibilidade e histórico de treino. Já a tabela Slot registra os treinos criados, com campos como descrição do treino, objetivo, data de criação e avaliação, além de um relacionamento com a tabela de usuários. Esse relacionamento é feito por meio de chave estrangeira e está configurado com deleção em cascata.

As variáveis sensíveis, como as credenciais do banco de dados, foram armazenadas em um arquivo .env, garantindo maior segurança e flexibilidade no gerenciamento de ambiente. A conexão com o banco é feita por meio do pacote mysql2/promise, o que permite uso de async/await para consultas assíncronas.

Caso sejam necessárias futuras melhorias ou expansões no sistema, a documentação poderá ser modificada. Isso inclui cenários como a substituição da IA local por uma API externa, inclusão de autenticação de usuários, histórico de treinos, ou persistência de avaliações. Além disso, a adoção de testes automatizados também exigirá a documentação de casos de teste e cobertura.

Em resumo, o desenvolvimento do Musculum foi realizado com foco na escalabilidade, modularidade e acessibilidade, utilizando tecnologias modernas e bem estabelecidas no ecossistema de desenvolvimento web.

8 Resultados

A aplicação *Musculum* foi desenvolvida com o objetivo de proporcionar aos usuários uma ferramenta prática e personalizada para a geração de treinos físicos. Por meio da integração entre front-end, back-end e inteligência artificial, a aplicação permite que o usuário crie, visualize e avalie seus próprios treinos com base em dados individuais.

O projeto completo está disponível no seguinte repositório:

Link do GitHub: <<https://github.com/dcamargoo/musculum.git>>

8.1 Tela inicial

Na figura 13, ao acessar a aplicação, o usuário é direcionado à tela inicial, que apresenta o logotipo do sistema e quatro botões: “Acessar Slot de treino”, “Configurações”, “Perfil” e “Sair”. Essa interface serve como ponto de partida para todas as demais funcionalidades da plataforma.

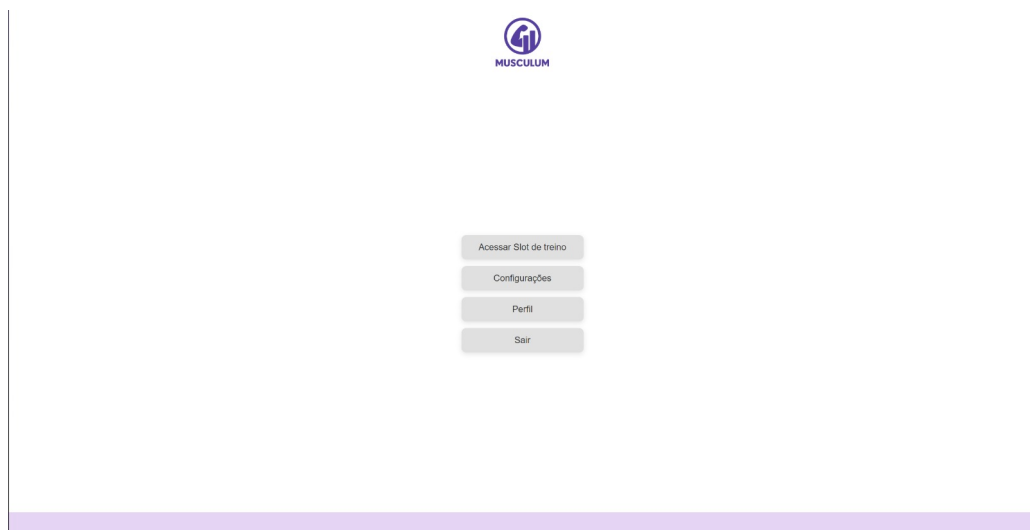


Figura 13 – Tela inicial da aplicação com botões de navegação.

8.2 Criação de slots de treino

Na figura 14, ao clicar em “Acessar Slot de treino”, o usuário é levado a uma nova tela onde pode criar seu primeiro slot. Essa área permite organizar os treinos por grupos ou tipos (como “Peito”, “Pernas” etc.).

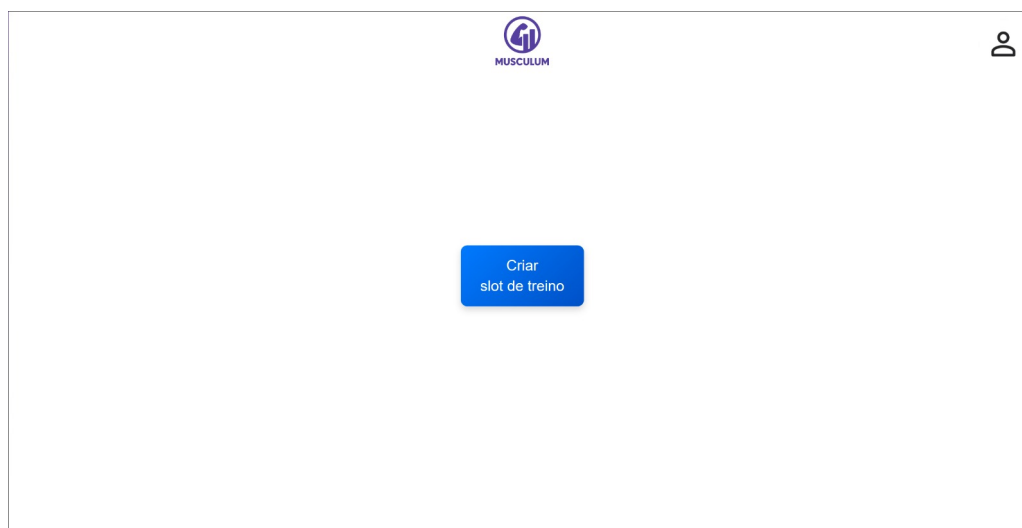


Figura 14 – Página para criar o primeiro slot de treino.

Na figura 15, ao clicar no botão azul, uma janela modal é exibida solicitando que o usuário insira o nome desejado para o novo slot.

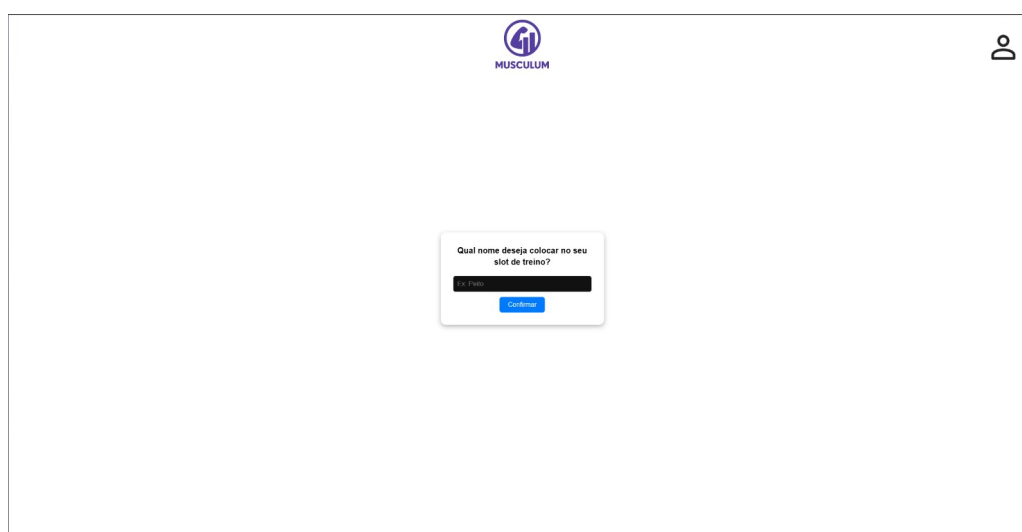


Figura 15 – Modal de criação de slot com campo para nome personalizado.

8.2.1 Visualização dos slots criados

Na figura 16, após nomear o slot, ele aparece em forma de cartão com o nome escolhido e um ícone de lixeira para exclusão. Isso possibilita ao usuário gerenciar seus treinos com facilidade.

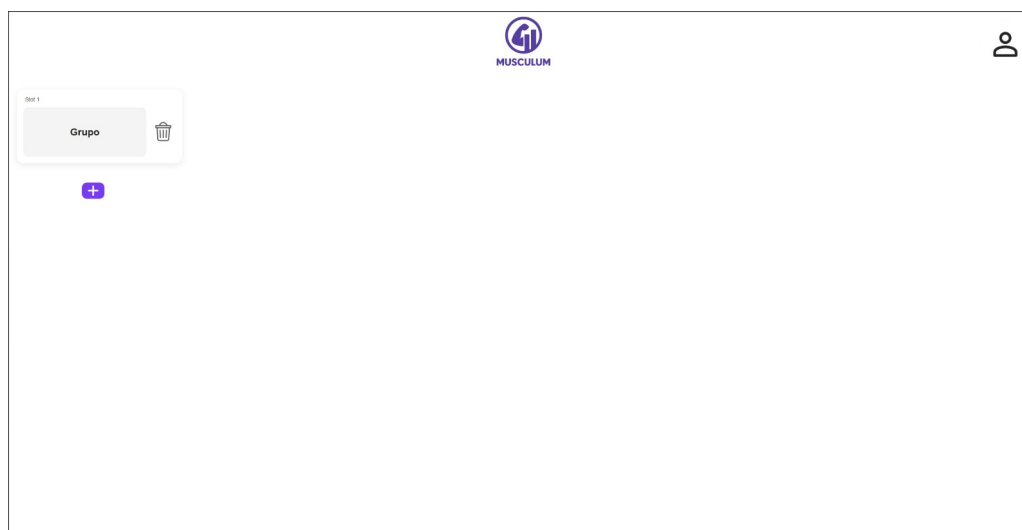


Figura 16 – Slot criado exibido na tela.

Na figura 17, o sistema permite que o usuário crie quantos slots desejar, e eles são exibidos verticalmente com um botão “+” para facilitar a adição de novos grupos de treino.

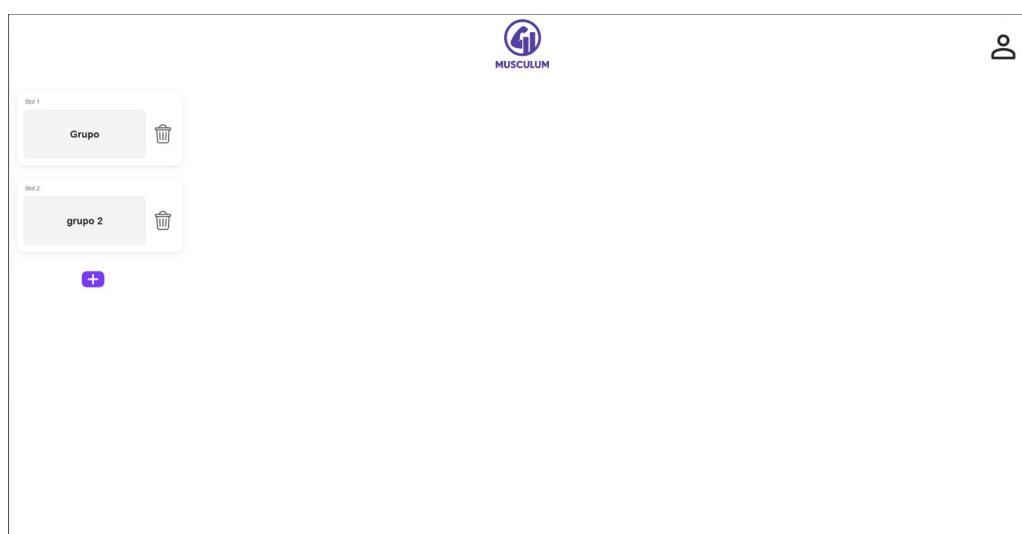


Figura 17 – Exemplo com múltiplos slots criados.

8.3 Geração de treino com IA

Na figura 18, ao acessar um slot que ainda não possui treino, o sistema exibe o botão “Criar treino”. O usuário então é levado a uma tela de formulário para inserir informações necessárias à geração do treino.



Figura 18 – Tela intermediária para iniciar a criação de treino com IA.

Na figura 19, o formulário solicita tipo de treino, idade, altura, peso, objetivo e equipamentos disponíveis. Com essas informações, o sistema constrói um prompt e envia à IA para gerar um treino personalizado.

A interface mostra o logo 'MUSCULUM' e um ícone de perfil no topo. Abaixo, há um formulário intitulado 'Criar novo treino' com campos para: 'Tipo de treino (ex: cardio, força)', 'Idade (anos)', 'Altura (cm)', 'Peso (kg)', 'Objetivo (ex: emagrecer, ganhar massa)', e 'Equipamentos disponíveis'. Abaixo dos campos, há uma caixa de seleção com o texto 'Criar uma rotina diária para todos os dias da semana?' e dois botões: 'Cancelar' e 'Confirmar'.

Figura 19 – Formulário com dados para personalização do treino.

8.4 Gerando treino

Na figura 20, após preencher todos os campos no formulário e clicar no botão “Confirmar”, a aplicação envia os dados para a inteligência artificial processar as informações. Com base nas características fornecidas, a IA gera uma rotina de treino completa, contendo exercícios específicos, quantidade de séries, repetições e eventuais recomendações.

O conteúdo do treino gerado é exibido imediatamente no cartão do slot correspondente, substituindo o texto inicial. Essa funcionalidade oferece ao usuário uma visualização clara e imediata da sua programação de treino personalizada.

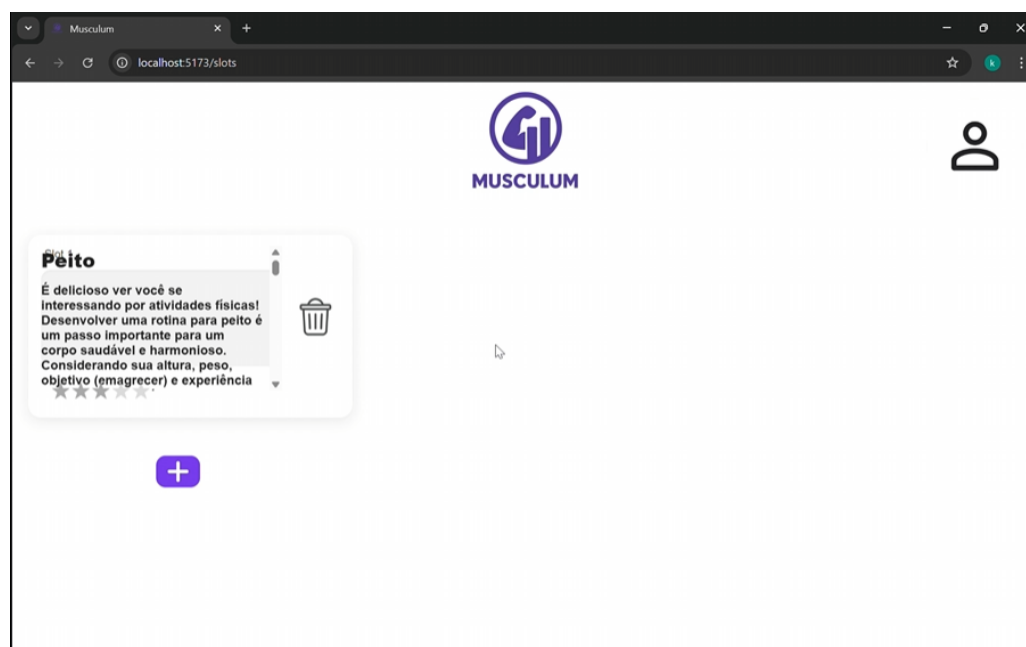


Figura 20 – Treino gerado pela IA exibido no slot selecionado.

8.5 avaliar treino

Na Figura 21, é apresentada a funcionalidade de avaliação do treino gerado. Após visualizar a rotina proposta pela inteligência artificial, o usuário pode atribuir uma nota de 1 a 5 estrelas, de acordo com sua satisfação com o conteúdo recebido.

Esse mecanismo de avaliação não apenas coleta a percepção do usuário sobre o treino, mas também ativa uma lógica automatizada: caso o usuário avalie com 3 estrelas ou menos, o sistema interpreta isso como uma insatisfação e, de forma proativa, realiza uma nova requisição à IA para gerar um treino alternativo. Essa abordagem busca garantir que o usuário receba uma sugestão mais adequada às suas preferências ou expectativas.

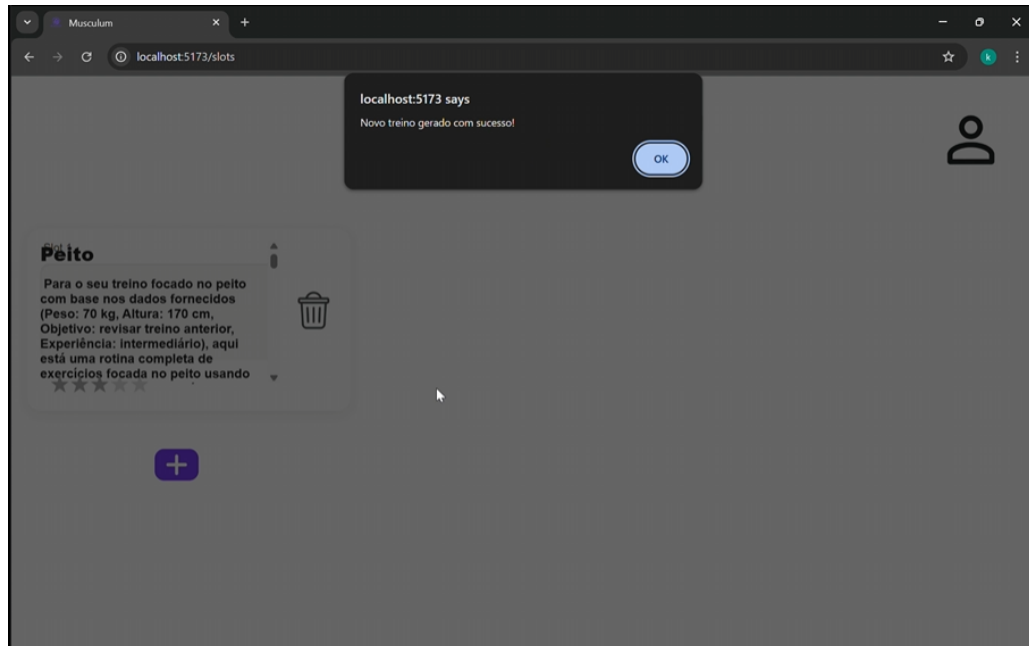


Figura 21 – Treino gerado pela IA exibido no slot selecionado.

8.6 Análise crítica do sistema desenvolvido

O sistema *Musculum* demonstrou-se funcional, intuitivo e com grande potencial de expansão. Entre os pontos fortes destacam-se:

- Interface limpa, acessível e modular.
- Integração eficaz com IA local via Ollama.
- Flexibilidade para criação e avaliação de múltiplos treinos.
- Separação clara entre camadas (front-end, back-end, IA).

8.6.1 Também foram identificadas algumas limitações:

- Ausência de login e autenticação de usuários.
- Armazenamento dos treinos apenas em memória local (não persiste no banco).
- A IA utilizada é limitada por ser gratuita; soluções como a OpenAI poderiam gerar resultados mais precisos.
- O tempo de resposta para geração do treino é elevado, pois o servidor da IA, o front-end e o back-end estão hospedados juntos em uma máquina da AWS com baixo desempenho.

- Ausência de testes automatizados para garantir estabilidade.

9 Conclusão

O desenvolvimento do sistema Musculum representou a construção de uma solução inovadora que combina tecnologia web com inteligência artificial para oferecer aos usuários uma ferramenta personalizada de geração de treinos físicos. O projeto foi concebido com foco na acessibilidade, modularidade e escalabilidade, utilizando tecnologias modernas como React, Vite, Node.js, Express e MySQL, além da integração com o modelo de linguagem Mistral, executado localmente via Ollama.

Durante o desenvolvimento, foram implementadas funcionalidades fundamentais, como a criação e exclusão de slots de treino, avaliação por estrelas, personalização com base nos dados do usuário e armazenamento temporário dos treinos. A interface foi desenhada com foco em simplicidade e usabilidade, e o código-fonte foi modularizado para facilitar futuras manutenções. Além disso, o sistema já conta com estrutura de banco de dados preparada para suportar a persistência de dados e relacionamentos entre usuários e treinos.

A comunicação entre as camadas front-end e back-end foi viabilizada por meio de uma API REST, e a integração com a inteligência artificial permitiu a geração de conteúdos dinâmicos baseados nos dados informados pelo usuário. Esse modelo de construção proporcionou uma aplicação robusta e flexível, que pode ser expandida e aprimorada conforme as necessidades futuras.

Como possíveis melhorias no sistema, destaca-se a implementação de autenticação e login de usuários, possibilitando o controle de sessões e a personalização contínua dos treinos com base em um histórico salvo. Outra melhoria importante seria migrar o armazenamento dos treinos para o banco de dados MySQL, substituindo o armazenamento temporário atual, o que traria maior persistência e integridade das informações. Também seria interessante adicionar recursos visuais mais avançados, como gráficos de progresso, calendário de treinos e notificações automáticas.

Entre os trabalhos futuros que podem ser derivados deste sistema, pode-se destacar a criação de um aplicativo mobile com as mesmas funcionalidades, ampliando o alcance da ferramenta. Outro possível caminho seria a aplicação de técnicas de aprendizado de máquina para recomendar treinos com base no histórico do usuário ou em dados populacionais. Por fim, a integração com dispositivos de monitoramento físico (como smartwatches e pulseiras fitness) pode permitir o ajuste automático dos treinos de acordo com o desempenho real do usuário.

Conclui-se, portanto, que o sistema Musculum atinge seus objetivos iniciais e abre espaço para múltiplas evoluções, servindo como uma base sólida para o desenvolvimento contínuo de tecnologias aplicadas ao bem-estar e à atividade física personalizada.

Referências

CARNEIRO, José Luiz; NUNES, Marcelo C. D. *Desenvolvimento de software ágil: conceitos, métodos e ferramentas*. São Paulo: LTC, 2015.

PACHECO, Jairo. *Engenharia de software: conceitos, técnicas e práticas*. 2. ed. São Paulo: Pearson, 2011.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.

SANTOS, Antonio Carlos de Souza; FONSECA, Aldo L. de Lima. *Introdução à engenharia de software*. 1. ed. São Paulo: Atlas, 2009.

SNYDER, C. *Software development metrics*. 1. ed. New York: McGraw-Hill, 2012.

SOMMERVILLE, I. *Engenharia de software: uma abordagem prática*. 10. ed. São Paulo: Pearson Education, 2015.

SAMPAIO, Gustavo Scalabrini. *Aulas e materiais da disciplina de Engenharia de Software*. Universidade Presbiteriana Mackenzie, São Paulo, 2025. Slides e conteúdos apresentados em aula.