

**UNIVERSIDADE FEDERAL DO MARANHÃO**  
**BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA**  
**ENGENHARIA DA COMPUTAÇÃO**  
**PROJETO E DESENVOLVIMENTO DE SOFTWARE**

**GRUPO 1 – FITAI**

São Luís

2025

**ANDRÉ LUIS AGUIAR DO NASCIMENTO (2020034117)**  
**HUGO SAMUEL DE LIMA OLIVEIRA (2023030425)**  
**LEONARDO SAMPAIO SERRA (2022002963)**  
**LUCAS EMANUEL GOMES AMARAL (2021041340)**  
**WESLEY DOS SANTOS GATINHO (2020051056)**

## **DOCUMENTAÇÃO - FITAI**

Documento apresentado como requisito parcial de avaliação da disciplina Projeto e Desenvolvimento de Software - Turma 01, no curso Bacharelado Interdisciplinar em Ciência e Tecnologia da Universidade Federal do Maranhão.

Prof. Dr. Thales Levi Azevedo Valente.

## RESUMO

Este documento detalha o desenvolvimento do aplicativo **FitAI**, focado em elevar a experiência de treino através da **inteligência artificial**. Ele proporcionará uma interface amigável e intuitiva para a criação de planos de treino personalizados, monitoramento de exercícios em tempo real com feedback preciso, e acompanhamento detalhado do progresso, garantindo a segurança e a eficácia de cada movimento e a possibilidade de visualização de progresso através de gráficos.

**Palavras-chave:** inteligência artificial, monitoramento, exercícios, progresso, gráficos.

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>5</b>
<b>1.1 Visão Geral do Documento .....</b>	<b>5</b>
<b>1.2 Contexto e Escopo do Projeto.....</b>	<b>5</b>
<b>1.2.1 O Problema.....</b>	<b>5</b>
<b>1.2.2 Justificativa e Oportunidade.....</b>	<b>5</b>
<b>1.3.1 Análise de Mercado.....</b>	<b>6</b>
<b>2 OBJETIVOS .....</b>	<b>6</b>
<b>2.1 Objetivo Geral.....</b>	<b>6</b>
<b>2.2 Objetivos Específicos.....</b>	<b>6</b>
<b>3 ELICITAÇÃO DE REQUISITOS.....</b>	<b>6</b>
<b>3.2 Requisitos Funcionais (RF).....</b>	<b>6</b>
<b>3.2 Requisitos Não Funcionais (RNF).....</b>	<b>7</b>
<b>4 DIAGRAMAS DE CASO DE USO .....</b>	<b>9</b>
<b>6 DIAGRAMA DE SEQUÊNCIA.....</b>	<b>44</b>
<b>7 DIAGRAMA DE ATIVIDADE.....</b>	<b>63</b>
<b>8 DIAGRAMA DE ESTADO .....</b>	<b>67</b>
<b>REFERÊNCIAS .....</b>	<b>72</b>

# 1 INTRODUÇÃO

## 1.1 Visão Geral do Documento

Este documento serve como a fonte central de referência para o projeto FitAI. Seu propósito é descrever em detalhes a concepção do sistema, seus objetivos, requisitos funcionais e não funcionais, e a arquitetura técnica por meio de um conjunto de modelos e diagramas UML. Ele se destina a guiar a equipe de desenvolvimento durante a implementação e a servir como registro formal do projeto para fins acadêmicos e de manutenção futura.

## 1.2 Contexto e Escopo do Projeto

### 1.2.1 O Problema

O mercado de fitness, embora em plena expansão, apresenta uma barreira significativa para muitos praticantes: o acesso a uma orientação personalizada e de alta qualidade. A execução correta de exercícios, especialmente a manutenção da postura adequada, é um fator crítico para garantir não apenas a eficácia do treino, mas principalmente a segurança do praticante, minimizando o risco de lesões.

Historicamente, a solução para essa demanda se concentra em duas alternativas principais: a contratação de Personal Trainers, que representa uma solução de alto custo e inacessível para uma grande parcela do público, e o uso de wearables específicos para monitoramento, que podem impor barreiras de adoção e conveniência. Essa conjuntura cria uma lacuna de mercado para quem busca excelência no treino sem grandes investimentos ou complexidades.

### 1.2.2 Justificativa e Oportunidade

O desenvolvimento do FitAI é justificado por uma confluência de fatores tecnológicos e de mercado. A proposta de valor do projeto é revolucionar o cenário do fitness utilizando a câmera do smartphone e Inteligência Artificial avançada para democratizar o acesso a treinos de maior qualidade, garantindo eficácia e segurança.

O projeto se posiciona em um momento oportuno, alinhado com a transformação digital do setor de saúde e bem-estar. A IA surge como um pilar tecnológico capaz de viabilizar o feedback postural preciso e em tempo real através da visão computacional, utilizando um dispositivo onipresente: o smartphone. O lançamento de um Mínimo Produto Viável (MVP) em curto prazo permitirá validar a demanda real por essa tecnologia, coletar feedback essencial e estabelecer uma vantagem competitiva para o FitAI como líder em tecnologia fitness.

1.3 Concepção do Sistema

1.3.1 Análise de Mercado

O setor global de saúde e bem-estar alcançou US\$ 1,8 trilhões em 2024, impulsionado pela transformação digital. Empresas que integram IA na gestão podem aumentar a eficiência operacional em até 40%. A forte tendência de modelos híbridos é evidenciada pelo fato de que 80% dos praticantes buscam opções digitais para complementar treinos presenciais, indicando uma clara necessidade de flexibilidade. Adicionalmente, o uso estratégico de dados para tomada de decisões se mostra um grande diferencial competitivo.

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver um protótipo funcional e estável de um aplicativo móvel, o FitAI, que emprega Inteligência Artificial para fornecer feedback postural em tempo real, demonstrando a viabilidade técnica e o potencial de mercado de uma solução inovadora para treinos mais seguros e eficazes.

2.2 Objetivos Específicos

- Demonstrar a Inovação da IA no Feedback Postural em Tempo Real.
- Entregar um Protótipo Funcional e Estável.
- Validar o Potencial de Mercado e a Experiência do Usuário.
- Demonstrar Capacidade de Planejamento e Execução Ágil.

3 ELICITAÇÃO DE REQUISITOS

3.2 Requisitos Funcionais (RF)

RF1		
Autenticação e Sessão de Usuário		
RF1.1	Autenticação e Sessão de Usuário	O sistema deve permitir a autenticação do usuário exclusivamente via Google.
RF1.2	Autenticação e Sessão de Usuário	O sistema deve permitir ao usuário realizar logout da sua sessão.
RF1.3	Autenticação e Sessão de Usuário	O sistema deve permitir ao usuário revogar o acesso do aplicativo à sua conta Google.
RF2		
Monitoramento de Exercícios Guiado por IA		
RF2.1	Monitoramento de Exercícios Guiado por IA	O sistema deve permitir ao usuário selecionar um exercício para monitoramento ao vivo.

RF2.2	Monitoramento de Exercícios Guiado por IA	O sistema deve exibir o número de repetições e o tempo decorrido do exercício em tempo real.
RF2.3	Monitoramento de Exercícios Guiado por IA	O sistema deve fornecer feedback da IA sobre a execução do exercício em tempo real.
RF2.4	Monitoramento de Exercícios Guiado por IA	O sistema deve fornecer instruções detalhadas para a execução de cada exercício.
RF2.5	Monitoramento de Exercícios Guiado por IA	O sistema deve permitir ao usuário iniciar e parar o monitoramento do treino.
RF2.6	Monitoramento de Exercícios Guiado por IA	O sistema deve armazenar os resultados do monitoramento (duração, repetições, etc.) no banco de dados.
<b>RF3</b> <b>Registro e Análise de Dados</b>		
RF3.1	Registro e Análise de Dados	O sistema deve permitir o registro de dados de cardio, peso e medidas corporais.
RF3.2	Registro e Análise de Dados	O sistema deve permitir o registro de dados via imagem (OCR simulado) e também de forma manual.
RF3.3	Registro e Análise de Dados	O sistema deve permitir adicionar registros de cardio (data, tipo, distância, tempo).
RF3.4	Registro e Análise de Dados	O sistema deve permitir adicionar registros de peso (data, valor em kg).
RF3.5	Registro e Análise de Dados	O sistema deve permitir adicionar registros de medidas corporais (data, tipo, valor em cm).
RF3.6	Registro e Análise de Dados	O sistema deve visualizar o progresso dos dados registrados em gráficos de evolução.
<b>RF4</b> <b>Geração de Ideias e Dicas Fitness com IA</b>		
RF4.1	Geração de Ideias e Dicas Fitness com IA	O sistema deve permitir ao usuário obter uma "Dica Fitness do Dia" gerada por IA.
RF4.2	Geração de Ideias e Dicas Fitness com IA	O sistema deve permitir ao usuário gerar ideias de treino personalizadas com base em um prompt.
RF4.3	Geração de Ideias e Dicas Fitness com IA	O sistema deve armazenar as sugestões geradas para consulta e histórico.

### 3.2 Requisitos Não Funcionais (RNF)

<b>RNF1</b> <b>Performance</b>		
RNF1.1	Performance	O tempo de processamento de imagem (OCR) deve ser inferior a 45 segundos.

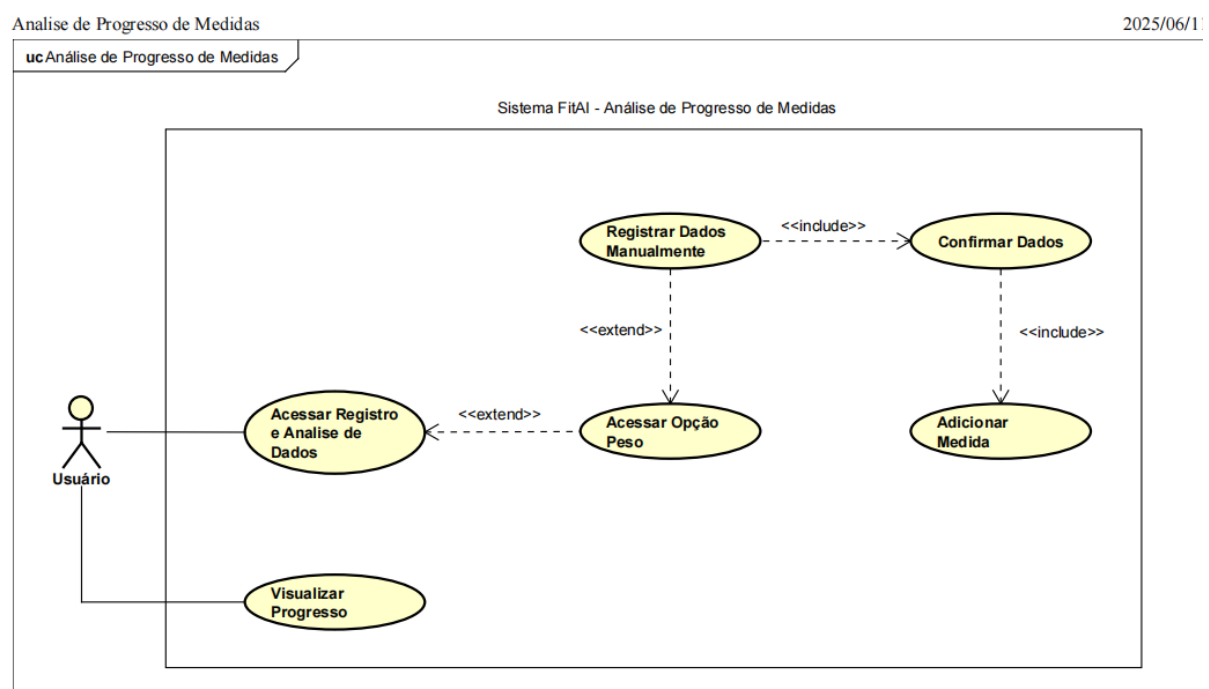
RNF1.2	Performance	A geração de dicas e treinos por IA deve ser inferior a 20 segundos.
RNF1.3	Performance	O carregamento de históricos e gráficos de progresso deve ser ágil (inferior a 20 segundos).
<b>RNF2</b> <b>Segurança</b>		
RNF2.1	Segurança	Os dados do usuário devem ser armazenados e transmitidos de forma segura para proteger a privacidade.
<b>RNF3</b> <b>Usabilidade</b>		
RNF3.1	Usabilidade	A interface deve ser intuitiva, fácil de usar e com feedback visual claro para as ações do usuário.
RNF3.2	Usabilidade	As instruções e o feedback da IA devem ser claros, objetivos e compreensíveis.
RNF3.3	Usabilidade	A inserção manual de dados deve ser simples e direta.
<b>RNF4</b> <b>Confiabilidade</b>		
RNF4.1	Confiabilidade	O sistema deve garantir a persistência e a integridade dos dados registrados pelo usuário, mesmo em caso de falhas.
RNF4.2	Confiabilidade	A funcionalidade de OCR deve ter uma alta taxa de acerto na extração de dados.
<b>RNF5</b> <b>Escalabilidade</b>		
RNF5.1	Escalabilidade	O sistema deve ser capaz de suportar um número crescente de usuários e dados sem degradação de performance.
<b>RNF6</b> <b>Compatibilidade</b>		
RNF6.1	Compatibilidade	A interface deve ser responsiva e se adaptar a diferentes tamanhos de tela (desktop, tablet, smartphone).
<b>RNF7</b> <b>Manutenibilidade</b>		
RNF7.1	Manutenibilidade	O código-fonte deve ser modular, bem documentado e seguir boas práticas de engenharia de software.



## 4 DIAGRAMAS DE CASO DE USO

Diagrama de Casos de Uso é uma ferramenta essencial na modelagem de sistemas, especialmente no contexto de desenvolvimento de software. Ele desempenha um papel crucial no processo de análise e design, pois oferece uma visão clara e concisa de como o sistema interage com seus usuários ou outros sistemas. Sua principal função é descrever as funcionalidades que o sistema deve fornecer a seus usuários, ou "atores", e como essas interações ocorrem. Nessa seção, serão abordados os Diagramas de Caso de Uso para o sistema FITAI, bem como a descrição detalhada de cada um.

Figura 01: Diagrama de Casos de Uso de Análise de Progresso de Medidas



Fonte: Autores, 2025.

### 4.1. Descrição Geral Análise de Progresso de Medidas

O diagrama representa o módulo "Análise de Progresso de Medidas" do sistema FitAI. Ele detalha a interação do ator **Usuário** com os casos de uso para acessar, registrar e visualizar a evolução de suas medidas corporais. O objetivo do módulo é fornecer uma ferramenta para que o usuário monitore seu progresso de forma simples e eficaz.

#### 4.1.1. Atores

- **Usuário:** Pessoa que utiliza o sistema para registrar e acompanhar suas medidas corporais, visualizando o progresso ao longo do tempo.

#### 4.1.2. Casos de Uso

#### 4.1.3. Acessar Registro e Análise de Dados

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
- **Fluxo principal:**
  1. O usuário seleciona a opção para acessar a área de registro e análise.
  2. O sistema exibe a tela principal desta funcionalidade, onde outras ações podem ser iniciadas.
- **Pós-condições:**
  - A tela de registro e análise é exibida, pronta para a interação do usuário.
- **Observação:** Este caso de uso é o ponto de partida para o registro de novas medidas e pode ser estendido por outros casos de uso, como Acessar Opção Peso.

#### 4.1.4. Visualizar Progresso

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
  - Deve haver pelo menos uma medida registrada no histórico do usuário.
- **Fluxo principal:**
  1. O usuário seleciona a opção para visualizar seu progresso.
  2. O sistema recupera o histórico de medidas do usuário.
  3. O sistema apresenta os dados de progresso ao usuário (ex: em gráficos ou tabelas).
- **Pós-condições:**
  - Os dados de progresso do usuário são exibidos na tela.

#### 4.1.5. Acessar Opção Peso

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar na tela do caso de uso Acessar Registro e Análise de Dados.
- **Fluxo principal:**
  1. O usuário seleciona a opção específica para gerenciar a medida "Peso".

2. O sistema exibe a interface relacionada ao registro de peso.

- **Pós-condições:**

- O usuário tem acesso às opções de registro manual para a medida de peso.

- **Observação:** Este caso de uso **estende** Acessar Registro e Análise de Dados, ou seja, é uma funcionalidade opcional que parte dele.

#### 4.1.6. Registrar Dados Manualmente

- **Atores envolvidos:** Usuário.

- **Pré-condições:**

- O usuário deve ter executado o caso de uso Acessar Opção Peso.

- **Fluxo principal:**

1. O usuário opta por registrar um novo dado de peso manualmente.
2. O sistema apresenta os campos para a inserção dos dados.
3. O usuário preenche as informações (ex: valor do peso).
4. Ao submeter, o fluxo do caso de uso Confirmar Dados é obrigatoriamente iniciado.

- **Pós-condições:**

- Os dados inseridos pelo usuário são enviados para validação.

- **Observação:** Este caso de uso **estende** Acessar Opção Peso e **inclui** obrigatoriamente o caso de uso Confirmar Dados.

#### 4.1.7. Confirmar Dados

- **Atores envolvidos:** Usuário, Sistema.

- **Pré-condições:**

- O caso de uso Registrar Dados Manualmente foi iniciado e os dados foram submetidos.

- **Fluxo principal:**

1. O sistema recebe os dados inseridos.
2. O sistema valida as informações (ex: verifica se o valor é numérico e válido).
3. O sistema pode solicitar uma confirmação final do usuário.
4. Após a confirmação, o fluxo do caso de uso Adicionar Medida é obrigatoriamente iniciado.

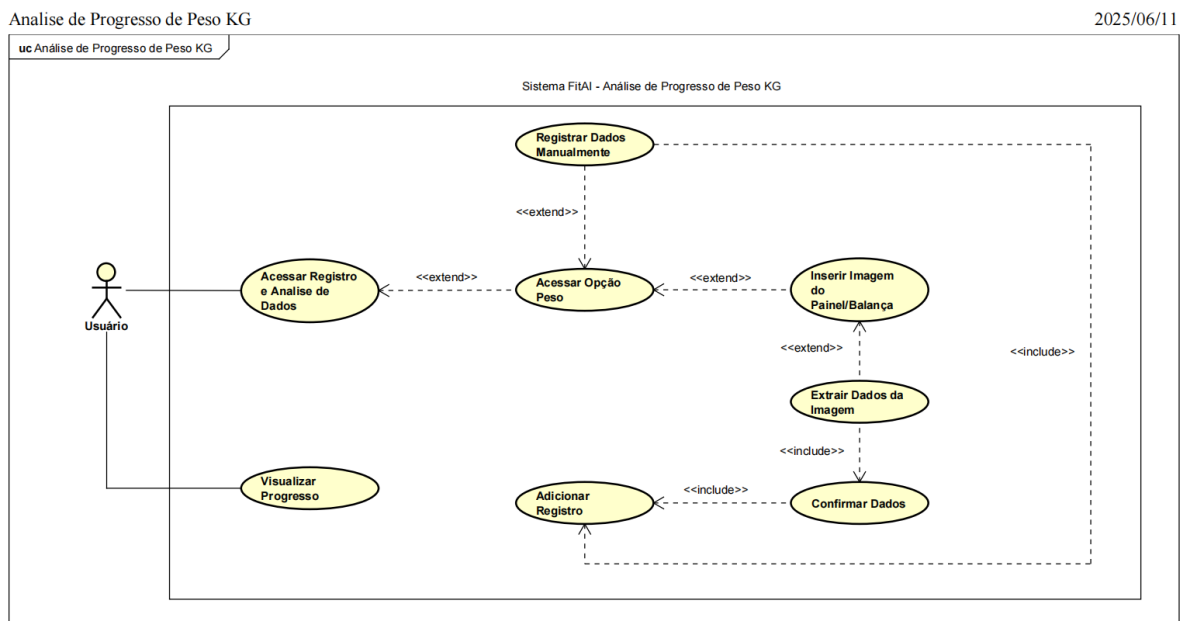
- **Pós-condições:**

- Os dados são validados e confirmados, prontos para serem salvos.
- **Observação:** Este é um caso de uso **incluído**. Ele é uma parte obrigatória do Registrar Dados Manualmente e, por sua vez, **inclui** Adicionar Medida.

#### 4.1.8. Adicionar Medida

- **Atores envolvidos:** Sistema.
- **Pré-condições:**
  - O caso de uso Confirmar Dados foi concluído com sucesso.
- **Fluxo principal:**
  1. O sistema registra a nova medida no banco de dados, associando-a ao perfil do usuário e à data correspondente.
- **Pós-condições:**
  - A nova medida está salva permanentemente no sistema.
  - O sistema informa ao usuário que o registro foi bem-sucedido.
- **Observação:** Este é um caso de uso **incluído**, representando o passo final e obrigatório que ocorre após a execução de Confirmar Dados.

Figura 02: Diagrama de Casos de Uso de Análise de Progresso de Peso



Fonte: Autores, 2025.

## 4.2. Descrição Geral Análise de Progresso de Peso

O diagrama representa a nova versão do módulo "Análise de Progresso de Peso" do sistema FitAI. Esta atualização expande as funcionalidades de registro, permitindo que o **Usuário** insira seu peso de duas formas: manualmente ou através da extração de dados de uma imagem do painel de uma balança. O objetivo é oferecer mais conveniência e precisão no acompanhamento do progresso.

### 4.2.1. Atores

- **Usuário:** Pessoa que utiliza o sistema para registrar e acompanhar seu peso, seja por inserção manual ou por meio de fotos, e para visualizar seu progresso.

### 4.2.2. Casos de Uso

#### 4.2.3. Acessar Registro e Análise de Dados

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
- **Fluxo principal:**
  1. O usuário seleciona a opção para acessar a área de registro e análise.
  2. O sistema exibe a tela principal desta funcionalidade, que serve como ponto de partida para outras ações.
- **Pós-condições:**
  - A tela de registro e análise é exibida, pronta para a interação do usuário.

#### 4.2.4. Visualizar Progresso

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
  - Deve haver pelo menos um registro de peso no histórico do usuário.
- **Fluxo principal:**
  1. O usuário seleciona a opção para visualizar seu progresso.
  2. O sistema recupera e exibe o histórico de peso do usuário (ex: em um gráfico de linha).
- **Pós-condições:**
  - Os dados de progresso do usuário são exibidos na tela.

#### 4.2.5. Acessar Opção Peso

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar na tela do caso de uso Acessar Registro e Análise de Dados.
- **Fluxo principal:**
  1. Na tela de análise, o usuário seleciona a opção específica para gerenciar "Peso".
  2. O sistema exibe a interface de gerenciamento de peso, oferecendo as opções de registro.
- **Pós-condições:**
  - O usuário tem acesso às opções para Registrar Dados Manualmente ou Inserir Imagem do Painel/Balança.
- **Observação:** Este caso de uso **estende** Acessar Registro e Análise de Dados, sendo uma funcionalidade opcional que parte dele.

#### 4.2.6. Registrar Dados Manualmente

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve ter executado o caso de uso Acessar Opção Peso.
- **Fluxo principal:**
  1. O usuário seleciona a opção para registrar o peso manualmente.
  2. O sistema apresenta um campo para a inserção do valor do peso.
  3. O usuário digita o valor e submete o dado.
  4. Ao submeter, o fluxo do caso de uso Confirmar Dados é obrigatoriamente iniciado.
- **Pós-condições:**
  - O dado inserido manualmente é enviado para o fluxo de confirmação.
- **Observação:** Este caso de uso **estende** Acessar Opção Peso. A seta tracejada para Confirmar Dados indica que a confirmação é o próximo passo lógico e obrigatório.

#### 4.2.7. Inserir Imagem do Painel/Balança

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve ter executado o caso de uso Acessar Opção Peso.

- **Fluxo principal:**
  1. O usuário seleciona a opção para registrar o peso a partir de uma imagem.
  2. O sistema aciona a câmera do dispositivo ou abre a galeria de fotos.
  3. O usuário tira uma foto do painel da balança ou seleciona uma imagem existente.
- **Pós-condições:**
  - Uma imagem é carregada no sistema, pronta para a extração de dados.
- **Observação:** Este caso de uso **estende** Acessar Opção Peso.

#### 4.2.8. Extrair Dados da Imagem

- **Atores envolvidos:** Usuário, Sistema.
- **Pré-condições:**
  - O caso de uso Inserir Imagem do Painel/Balança foi concluído.
- **Fluxo principal:**
  1. O usuário aciona a função para ler o peso da imagem carregada.
  2. O sistema processa a imagem (usando OCR - Reconhecimento Óptico de Caracteres) para identificar o valor do peso.
  3. O sistema preenche o campo de peso com o valor extraído.
  4. O fluxo do caso de uso Confirmar Dados é obrigatoriamente iniciado.
- **Pós-condições:**
  - O dado extraído da imagem é enviado para o fluxo de confirmação.
- **Observação:** Este caso de uso **estende** Inserir Imagem do Painel/Balança. A seta para Confirmar Dados indica que a confirmação é o próximo passo obrigatório.

#### 4.2.9. Confirmar Dados

- **Atores envolvidos:** Usuário, Sistema.
- **Pré-condições:**
  - O fluxo foi iniciado a partir de Registrar Dados Manualmente ou Extrair Dados da Imagem.
- **Fluxo principal:**
  1. O sistema exibe o valor do peso (digitado ou extraído) para o usuário.
  2. O usuário verifica se o valor está correto e pode editá-lo se necessário.
  3. O usuário clica no botão para confirmar o valor.

4. Após a confirmação, o fluxo do caso de uso Adicionar Registro é obrigatoriamente iniciado.

- **Pós-condições:**

- O valor do peso é validado e confirmado pelo usuário.

- **Observação:** Este é um caso de uso **incluído** (reutilizado) pelos fluxos de registro manual e por imagem. Ele, por sua vez, **inclui** Adicionar Registro.

#### 4.2.10. Adicionar Registro

- **Atores envolvidos:** Sistema.

- **Pré-condições:**

- O caso de uso Confirmar Dados foi concluído com sucesso.

- **Fluxo principal:**

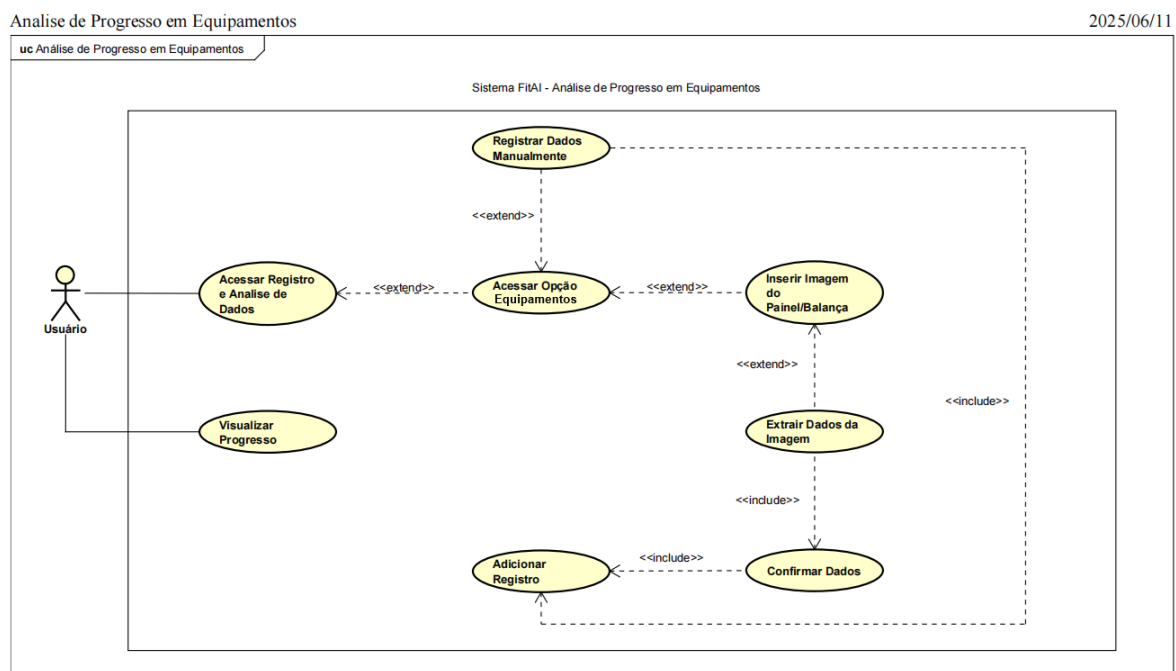
1. O sistema salva o valor do peso confirmado no banco de dados, associando-o ao perfil do usuário e à data do registro.

- **Pós-condições:**

- Um novo registro de peso é criado e salvo permanentemente no sistema.
  - O sistema exibe uma mensagem de sucesso ao usuário.

- **Observação:** Este é um caso de uso **incluído**, representando o passo final e obrigatório do processo de registro, iniciado após Confirmar Dados.

Figura 03: Diagrama de Casos de Uso de Análise de Progresso em Equipamentos.



Fonte: Autores, 2025



### 4.3. Descrição Geral Análise de Progresso em Equipamentos.

O diagrama representa o módulo "Análise de Progresso em Equipamentos" do sistema FitAI. O objetivo é permitir que o **Usuário** monitore sua evolução de desempenho em diversos equipamentos de ginástica. O sistema oferece duas formas distintas para o registro de dados: um registro manual rápido e um registro por imagem, que permite extrair dados diretamente do painel de um equipamento (como uma esteira de cardio ou o seletor de peso de uma máquina de musculação) e confirmá-los antes de salvar.

#### 4.3.1. Atores

- **Usuário:** Pessoa que utiliza o sistema para registrar e acompanhar seu progresso de desempenho em equipamentos, inserindo dados manualmente ou por meio de fotos, e visualizando seu histórico.

#### 4.3.2. Casos de Uso

##### 4.3.3. Acessar Registro e Análise de Dados

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
- **Fluxo principal:**
  1. O usuário seleciona a opção para acessar a área de registro e análise.
  2. O sistema exibe a tela principal desta funcionalidade, que serve como ponto de partida para outras ações.
- **Pós-condições:**
  - A tela de registro e análise é exibida, pronta para a interação do usuário.

##### 4.3.4. Visualizar Progresso

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
  - Deve haver pelo menos um registro salvo no histórico do usuário.
- **Fluxo principal:**

1. O usuário seleciona a opção para visualizar seu progresso.
  2. O sistema recupera e exibe o histórico de registros do usuário (ex: em um gráfico).
- **Pós-condições:**
    - Os dados de progresso do usuário são exibidos na tela.

#### 4.3.5. Acessar Opção Equipamentos

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar na tela do caso de uso Acessar Registro e Análise de Dados.
- **Fluxo principal:**
  1. Na tela de análise, o usuário seleciona a opção específica para gerenciar o progresso em "Equipamentos".
  2. O sistema exibe a interface de gerenciamento, oferecendo as opções de registro.
- **Pós-condições:**
  - O usuário tem acesso às opções para Registrar Dados Manualmente ou Inserir Imagem do Painel/Balança.
- **Observação:** Este caso de uso **estende** Acessar Registro e Análise de Dados, sendo uma funcionalidade opcional que parte dele.

#### 4.3.6. Registrar Dados Manualmente

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve ter executado o caso de uso Acessar Opção Equipamentos.
- **Fluxo principal:**
  1. O usuário seleciona a opção para registrar os dados manualmente.
  2. O sistema apresenta os campos necessários para a inserção dos dados (ex: nome do equipamento, peso, repetições).
  3. O usuário digita os valores e aciona a opção para salvar.

4. O fluxo do caso de uso Adicionar Registro é iniciado diretamente.

- **Pós-condições:**

- O dado inserido manualmente é enviado diretamente para ser salvo no histórico.

- **Observação:** Este caso de uso **estende** Acessar Opção Equipamentos e representa um fluxo de "adição rápida", pulando a etapa de confirmação.

#### 4.3.7. Inserir Imagem do Pannel/Balança

- **Atores envolvidos:** Usuário.

- **Pré-condições:**

- O usuário deve ter executado o caso de uso Acessar Opção Equipamentos.

- **Fluxo principal:**

1. O usuário seleciona a opção para registrar a partir de uma imagem.
2. O sistema aciona a câmera do dispositivo ou abre a galeria de fotos.
3. O usuário tira ou seleciona uma foto do painel do equipamento.

- **Pós-condições:**

- Uma imagem é carregada no sistema, pronta para a extração de dados.

- **Observação:** Este caso de uso **estende** Acessar Opção Equipamentos.

#### 4.3.8. Extrair Dados da Imagem

- **Atores envolvidos:** Usuário, Sistema.

- **Pré-condições:**

- O caso de uso Inserir Imagem do Pannel/Balança foi concluído.

- **Fluxo principal:**

1. O usuário aciona a função para ler os dados da imagem carregada.
2. O sistema processa a imagem para identificar os valores relevantes.
3. O sistema preenche os campos de registro com os valores extraídos.
4. O fluxo do caso de uso Confirmar Dados é obrigatoriamente iniciado.

- **Pós-condições:**
  - O dado extraído da imagem é enviado para o fluxo de confirmação.
- **Observação:** Este caso de uso **estende** Inserir Imagem do Pannel/Balança e **inclui** Confirmar Dados.

#### 4.3.9. Confirmar Dados

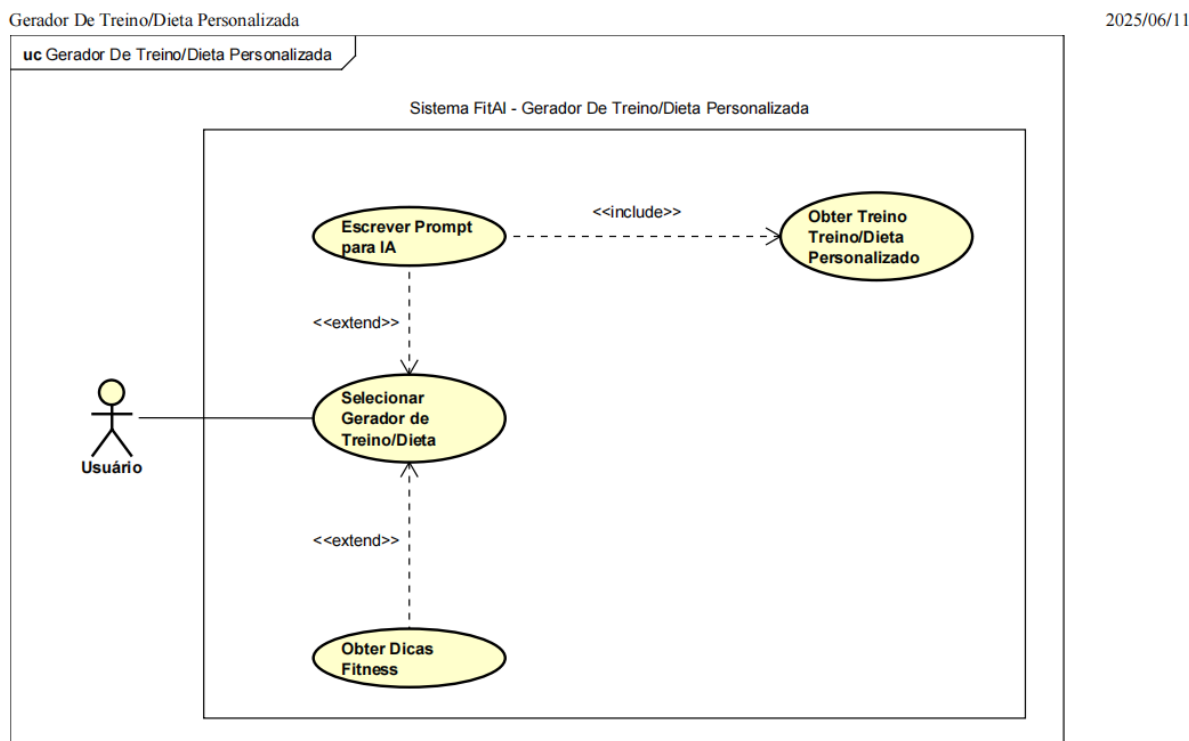
- **Atores envolvidos:** Usuário, Sistema.
- **Pré-condições:**
  - O fluxo foi iniciado a partir de Extrair Dados da Imagem.
- **Fluxo principal:**
  1. O sistema exibe os dados extraídos da imagem para o usuário.
  2. O usuário verifica se os valores estão corretos, podendo editá-los.
  3. O usuário clica no botão para confirmar os valores.
  4. Após a confirmação, o fluxo do caso de uso Adicionar Registro é obrigatoriamente iniciado.
- **Pós-condições:**
  - Os valores são validados e confirmados pelo usuário.
- **Observação:** Este caso de uso faz parte apenas do fluxo de registro por imagem e **inclui** Adicionar Registro.

#### 4.3.10. Adicionar Registro

- **Atores envolvidos:** Sistema.
- **Pré-condições:**
  - O fluxo foi iniciado diretamente por Registrar Dados Manualmente ou após a conclusão de Confirmar Dados.
- **Fluxo principal:**
  1. O sistema salva os dados no banco de dados, associando-os ao perfil do usuário.
- **Pós-condições:**
  - Um novo registro de progresso é criado e salvo permanentemente no sistema.

- O sistema exibe uma mensagem de sucesso ao usuário.
- **Observação:** Este é o caso de uso final, responsável por persistir os dados, e pode ser acionado por ambos os fluxos de registro.

Figura 04: Diagrama de Casos de Uso de Gerador de Treino/Dieta Personalizada.



Fonte: Autores, 2025.

#### 4.4. Descrição Geral Gerador De Treino/Dieta Personalizada

O diagrama representa o módulo "Gerador De Treino/Dieta Personalizada" do sistema FitAI. Esta funcionalidade utiliza Inteligência Artificial para prover ao **Usuário** planos de treino e dieta customizados com base em suas solicitações (prompts). Além disso, o módulo oferece a opção de obter dicas de fitness gerais, funcionando como um assistente virtual para os objetivos de saúde do usuário.

##### 4.4.1. Atores

- **Usuário:** Pessoa que interage com o sistema para solicitar e receber planos de treino/dieta gerados por IA ou para obter dicas de fitness.

##### 4.4.2. Casos de Uso

##### 4.4.3. Selecionar Gerador de Treino/Dieta

- **Atores envolvidos:** Usuário.

- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
- **Fluxo principal:**
  1. O usuário navega no aplicativo e seleciona a funcionalidade do gerador de treino/dieta.
  2. O sistema exibe a tela principal do gerador, apresentando as opções disponíveis.
- **Pós-condições:**
  - A interface do gerador está pronta para receber os comandos do usuário.
- **Observação:** Este é o caso de uso base, que serve como ponto de partida e pode ser estendido pelas funcionalidades Escrever Prompt para IA e Obter Dicas Fitness.

#### 4.4.4. Escrever Prompt para IA

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar na tela do caso de uso Selecionar Gerador de Treino/Dieta.
- **Fluxo principal:**
  1. O usuário escolhe a opção para criar um plano personalizado.
  2. O sistema apresenta uma área de texto para o usuário descrever sua solicitação (prompt), detalhando seus objetivos, restrições, preferências, etc.
  3. O usuário escreve e envia o prompt.
  4. Ao enviar, o fluxo do caso de uso Obter Treino/Dieta Personalizado é obrigatoriamente iniciado.
- **Pós-condições:**
  - O prompt do usuário é enviado ao sistema para processamento pela IA.
- **Observação:** Este caso de uso **estende** Selecionar Gerador de Treino/Dieta e **inclui** obrigatoriamente o caso de uso Obter Treino/Dieta Personalizado.

#### 4.4.5. Obter Treino/Dieta Personalizado

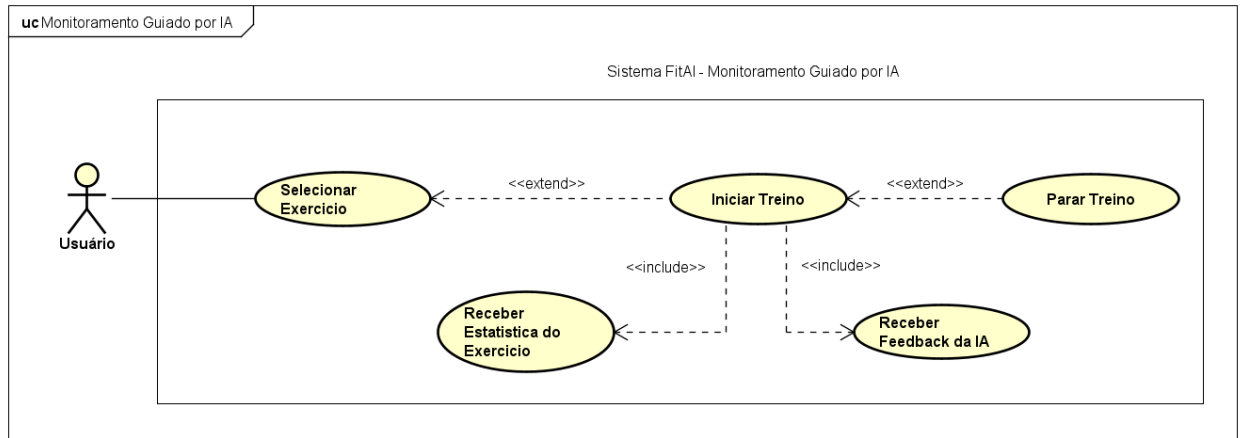
- **Atores envolvidos:** Sistema (FitAI).
- **Pré-condições:**
  - O caso de uso Escrever Prompt para IA foi concluído pelo usuário.

- **Fluxo principal:**
  1. O sistema recebe o prompt do usuário.
  2. O sistema processa a solicitação utilizando seu modelo de Inteligência Artificial.
  3. O sistema gera um plano de treino ou dieta de acordo com o prompt.
  4. O plano gerado é formatado e exibido na tela para o usuário.
- **Pós-condições:**
  - Um plano de treino ou dieta personalizado é apresentado ao usuário.
- **Observação:** Este é um caso de uso **incluído**, representando a resposta automática e obrigatória do sistema à submissão de um prompt. Ele não é iniciado diretamente pelo ator.

#### 4.4.6. Obter Dicas Fitness

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar na tela do caso de uso Selecionar Gerador de Treino/Dieta.
- **Fluxo principal:**
  1. O usuário seleciona a opção para receber dicas de fitness.
  2. O sistema pode solicitar um tópico de interesse ou gerar uma dica aleatória.
  3. O sistema consulta a IA para obter uma dica relevante.
  4. A dica é exibida na tela para o usuário.
- **Pós-condições:**
  - Uma dica de fitness é apresentada ao usuário.
- **Observação:** Este caso de uso **estende** Selecionar Gerador de Treino/Dieta, funcionando como uma funcionalidade alternativa à geração de um plano completo.

Figura 05: Diagrama de Casos de Uso de Monitoramento Guiado por IA



Fonte: Autores, 2025.

#### 4.5. Descrição Geral Monitoramento Guiado por IA

O diagrama representa o módulo "Monitoramento Guiado por IA" do sistema FitAI. O propósito desta funcionalidade é oferecer ao **Usuário** uma experiência de treino interativa e inteligente. O sistema utiliza a IA para monitorar a execução de um exercício em tempo real, fornecendo estatísticas de desempenho e feedback corretivo para melhorar a forma, a eficácia e a segurança do treino.

##### 4.5.1. Atores

- **Usuário:** Pessoa que realiza um exercício enquanto é monitorada e guiada pelo sistema de Inteligência Artificial.

##### 4.5.2. Casos de Uso

##### 4.5.3. Selecionar Exercício

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve estar autenticado no sistema.
- **Fluxo principal:**
  1. O usuário acessa a lista de exercícios disponíveis para monitoramento guiado.
  2. O usuário escolhe um exercício que deseja realizar.
  3. O sistema carrega as informações e as instruções para o exercício selecionado.
- **Pós-condições:**
  - O exercício está selecionado e o sistema está pronto para iniciar o monitoramento do treino.



- **Observação:** Este é o caso de uso inicial, que serve como ponto de partida para a funcionalidade de Iniciar Treino.

#### 4.5.4. Iniciar Treino

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O usuário deve ter concluído o caso de uso Selecionar Exercício.
- **Fluxo principal:**
  1. O usuário aciona o comando para iniciar a sessão de treino do exercício selecionado.
  2. O sistema ativa o modo de monitoramento (ex: utilizando a câmera do dispositivo).
  3. Durante a execução, os fluxos dos casos de uso Receber Estatística do Exercício e Receber Feedback da IA são executados continuamente.
- **Pós-condições:**
  - A sessão de treino guiado por IA está ativa e monitorando o usuário.
- **Observação:** Este caso de uso **estende** Selecionar Exercício. Ele **inclui** obrigatoriamente as funcionalidades Receber Estatística do Exercício e Receber Feedback da IA, e pode ser **estendido** pela ação Parar Treino.

#### 4.5.5. Receber Estatística do Exercício

- **Atores envolvidos:** Sistema (FitAI).
- **Pré-condições:**
  - O caso de uso Iniciar Treino deve estar ativo.
- **Fluxo principal:**
  1. Enquanto o usuário se exercita, o sistema rastreia e exibe métricas de desempenho em tempo real.
  2. As métricas podem incluir contagem de repetições, tempo de execução, amplitude do movimento, entre outras.
- **Pós-condições:**
  - O usuário visualiza seus dados de performance continuamente durante o exercício.

- **Observação:** Este é um caso de uso **incluído**, representando uma parte obrigatória e contínua da funcionalidade Iniciar Treino. Ele não é acionado diretamente pelo usuário.

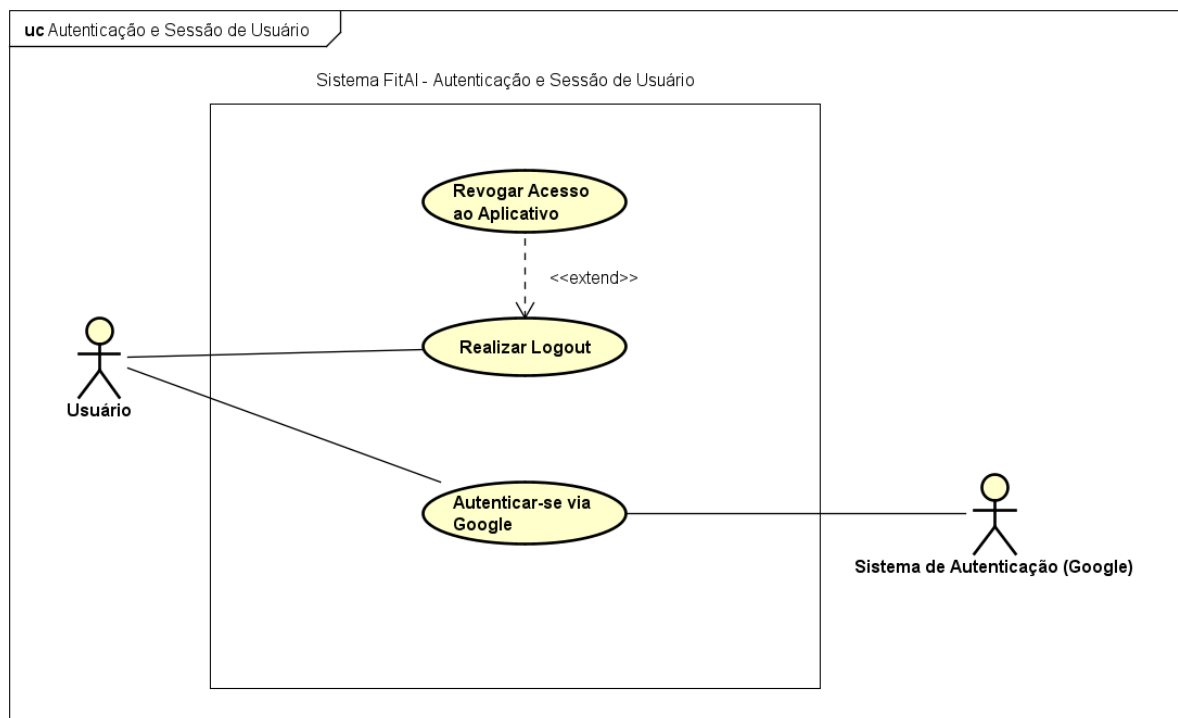
#### 4.5.6. Receber Feedback da IA

- **Atores envolvidos:** Sistema (FitAI).
- **Pré-condições:**
  - O caso de uso Iniciar Treino deve estar ativo.
- **Fluxo principal:**
  1. A IA do sistema analisa a postura e a execução do usuário em tempo real.
  2. Ao identificar uma oportunidade de melhoria ou um erro (ex: postura incorreta, velocidade inadequada), o sistema fornece uma instrução corretiva ao usuário.
  3. O feedback pode ser sonoro (ex: "Mantenha as costas retas") ou visual (uma mensagem na tela).
- **Pós-condições:**
  - O usuário recebe orientação em tempo real para executar o exercício com melhor forma e mais segurança.
- **Observação:** Este é um caso de uso **incluído** e representa o núcleo da funcionalidade de "monitoramento guiado", sendo parte obrigatória do Iniciar Treino.

#### 4.5.7. Parar Treino

- **Atores envolvidos:** Usuário.
- **Pré-condições:**
  - O caso de uso Iniciar Treino deve estar ativo.
- **Fluxo principal:**
  1. O usuário decide encerrar a sessão de treino do exercício atual.
  2. O usuário aciona o comando para parar.
  3. O sistema interrompe o monitoramento e finaliza a sessão de treino.
- **Pós-condições:**
  - A sessão de treino guiado é encerrada. O sistema pode apresentar um resumo do desempenho do usuário.
- **Observação:** Este caso de uso **estende** Iniciar Treino, representando a ação opcional do usuário para finalizar o treino a qualquer momento.

Figura 06: Diagrama de Autenticação e Sessão do usuário.



Fonte: Autores, 2025.

#### 4.6. Descrição Geral Autenticação e Sessão de Usuário

O diagrama representa o módulo de "Autenticação e Sessão de Usuário" do sistema FitAI. Ele descreve como um **Usuário** acessa o sistema de forma segura utilizando um provedor de identidade externo (neste caso, o Google). O módulo também cobre as ações de finalização de sessão (logout) e a revogação completa do acesso do aplicativo à conta do usuário.

##### 4.6.1. Atores

- **Usuário:** Pessoa que deseja acessar o sistema FitAI e gerenciar sua sessão.
- **Sistema de Autenticação (Google):** Ator externo que representa o serviço de identidade do Google. É responsável por validar as credenciais do usuário e autorizar o acesso do FitAI à conta.

##### 4.6.2. Casos de Uso

##### 4.6.3. Autenticar-se via Google

- **Atores envolvidos:** Usuário, Sistema de Autenticação (Google).
- **Pré-condições:**
  - O usuário não possui uma sessão ativa no aplicativo FitAI.
- **Fluxo principal:**

1. O Usuário seleciona a opção "Entrar com Google" na tela de login do FitAI.
2. O sistema FitAI redireciona o Usuário para a interface do Sistema de Autenticação (Google).
3. O Usuário insere suas credenciais (email/senha) na interface do Google e autoriza o acesso do FitAI.
4. O Sistema de Autenticação (Google) valida as credenciais, confirma a autorização e envia um token de autenticação para o sistema FitAI.
5. O sistema FitAI recebe e valida o token, estabelecendo uma sessão segura para o Usuário.

- **Pós-condições:**

- O Usuário está autenticado com sucesso no sistema FitAI e pode acessar as funcionalidades protegidas.

#### 4.6.4. Realizar Logout

- **Atores envolvidos:** Usuário.

- **Pré-condições:**

- O usuário deve possuir uma sessão ativa no aplicativo FitAI.

- **Fluxo principal:**

1. O Usuário seleciona a opção "Logout" ou "Sair" dentro do aplicativo.
2. O sistema FitAI invalida a sessão atual do Usuário no lado do servidor.
3. O sistema redireciona o Usuário para a tela de login.

- **Pós-condições:**

- A sessão do usuário no FitAI é encerrada. O acesso às funcionalidades protegidas exigirá uma nova autenticação.

- **Observação:** Este caso de uso pode ser **estendido** pela funcionalidade Revogar Acesso ao Aplicativo.

#### 4.6.5. Revogar Acesso ao Aplicativo

- **Atores envolvidos:** Usuário.

- **Pré-condições:**

- O usuário está no processo de Realizar Logout ou em uma tela de gerenciamento de conta que ofereça esta opção.

- **Fluxo principal:**

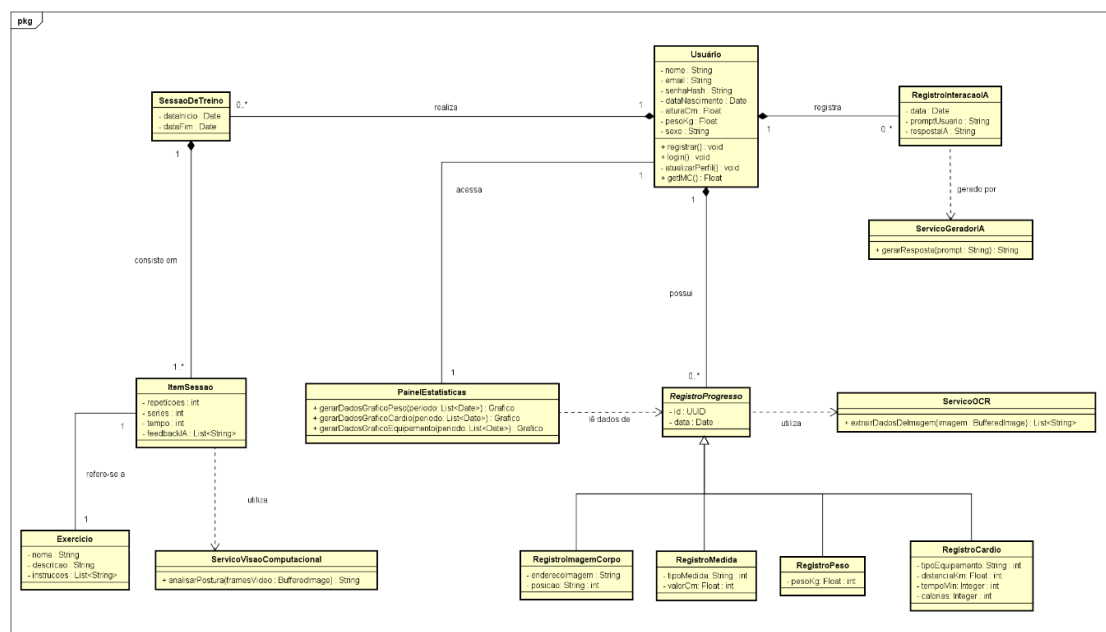
1. O Usuário escolhe a opção explícita para revogar as permissões de acesso do FitAI à sua conta Google (geralmente uma caixa de seleção ou botão adicional na tela de logout).
  2. O sistema FitAI, além de realizar o logout local, envia uma requisição ao Sistema de Autenticação (Google) para invalidar a autorização concedida anteriormente.
- **Pós-condições:**
    - A sessão do usuário é encerrada e as permissões que o FitAI tinha sobre a conta Google do usuário são removidas. Para um novo acesso, o usuário precisará autorizar o aplicativo novamente.
  - **Observação:** Este caso de uso **estende** Realizar Logout, representando uma desconexão mais profunda e segura do que um simples logout.

## 5 DIAGRAMA DE CLASSE

O diagrama de classes detalha a estrutura estática do sistema. A classe Usuário é central e se o diagrama representa a arquitetura estática do software, delineando as principais entidades do sistema, seus atributos, seus comportamentos e os relacionamentos que existem entre elas. O objetivo é servir como uma referência técnica para a equipe de desenvolvimento, garantindo uma compreensão clara da estrutura do código e das interações entre seus componentes.

### 5.1 Diagrama de Classes Completo

Figura 07: Diagrama de Classes Completo



Fonte: Autores, 2025.

### 5.1.1. Descrição Geral

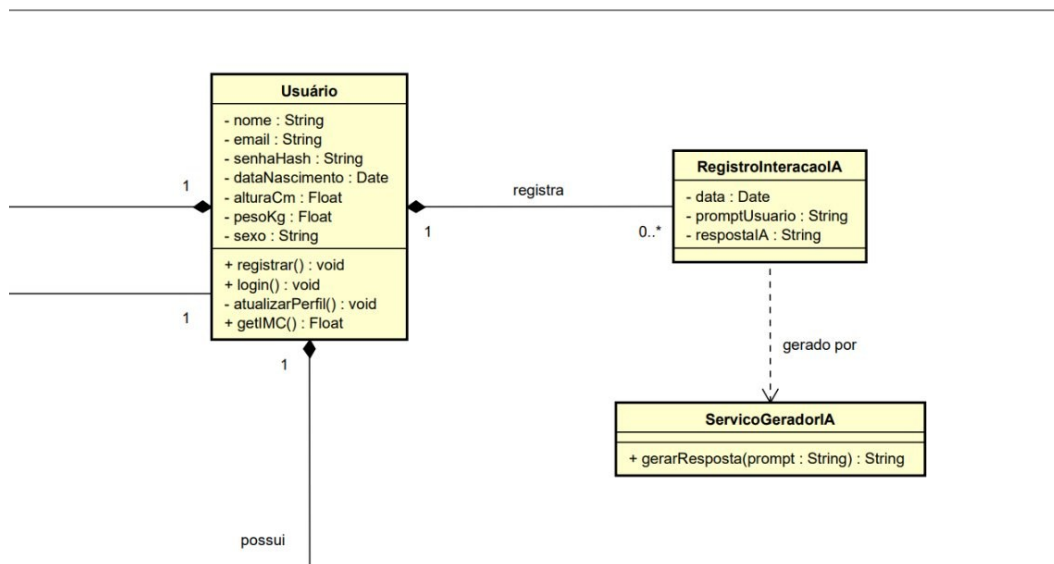
O diagrama de classe ilustra a arquitetura de um **sistema de software para gerenciamento de atividades físicas e acompanhamento de saúde**, projetado como uma plataforma de fitness pessoal. O propósito principal do sistema é permitir que os usuários (Usuario) gerenciem e registrem suas rotinas de treino, monitorem seu progresso físico ao longo do tempo e interajam com recursos inteligentes baseados em Inteligência Artificial.

O sistema é centrado no Usuario, que pode executar SessoesDeTreino compostas por Exercicios específicos. Todo o progresso, seja ele registrado manualmente (como peso e medidas) ou coletado por serviços (como análise de postura por visão computacional), é armazenado em diferentes tipos de RegistroProgresso.

Um diferencial notável é a incorporação de serviços de IA, como um ServicoVisaoComputacional para dar feedback sobre exercícios e um ServicoGeradorIA para interações de suporte (chatbot). Os dados coletados alimentam um PainelEstatisticas, que fornece ao usuário uma visão clara de sua evolução.

## 5.2 Diagrama de Classes Canto Superior Direito

Figura 08: Diagrama de Classes Canto Superior Direito



Fonte: Autores, 2025.

### 5.2.1. Descrição Geral Canto Superior Direito

Este segmento do diagrama de classes modela a funcionalidade de **interação do usuário com um assistente de Inteligência Artificial (IA)** dentro do sistema de fitness. Ele detalha como as informações do usuário são gerenciadas, como suas interações com a IA são registradas e qual serviço é responsável por gerar as respostas da IA. O fluxo principal representado é: um Usuario faz uma pergunta (prompt), o ServicoGeradorIA processa essa pergunta para criar uma resposta, e a conversa completa é armazenada como um RegistroInteracaoIA.

### 5.2.2. Classes

A seguir, a descrição detalhada de cada classe presente no diagrama.

#### Classe: Usuario

- **Descrição:** Representa o usuário final da aplicação. Esta classe é central para o sistema, armazenando dados pessoais, credenciais de acesso e informações físicas relevantes. Ela também define as operações básicas que um usuário pode realizar em seu perfil.
- **Atributos:**
  - - nome: String: (Privado) Nome completo do usuário.
  - - email: String: (Privado) Endereço de e-mail do usuário, provavelmente usado para login.
  - - senhaHash: String: (Privado) Hash da senha do usuário para armazenamento seguro.
  - - dataNascimento: Date: (Privado) Data de nascimento do usuário.
  - - alturaCm: Float: (Privado) Altura do usuário em centímetros.
  - - pesoKg: Float: (Privado) Peso do usuário em quilogramas.
  - - sexo: String: (Privado) Sexo/gênero do usuário.
- **Métodos (Operações):**
  - + registrar(): void: (Público) Realiza o cadastro de um novo usuário no sistema.
  - + login(): void: (Público) Autentica o usuário no sistema.
  - + atualizarPerfil(): void: (Público) Permite que o usuário modifique suas informações de perfil.
  - + getIMC(): Float: (Público) Calcula e retorna o Índice de Massa Corporal (IMC) do usuário.

#### Classe: RegistroInteracaoIA

- **Descrição:** Modela um único registro de interação entre o usuário e o assistente de IA. Sua responsabilidade é armazenar o histórico de conversas, guardando a pergunta do usuário, a resposta da IA e a data da interação.
- **Atributos:**
  - - data: Date: (Privado) Data e hora em que a interação ocorreu.
  - - promptUsuario: String: (Privado) A pergunta ou comando enviado pelo usuário.

- - respostaIA: String: (Privado) A resposta gerada pela IA.
- **Métodos (Operações):**
  - Nenhum método é especificado, indicando que esta é primariamente uma classe de dados (entidade ou DTO).

### Classe: ServicoGeradorIA

- **Descrição:** É uma classe de serviço responsável pela lógica de negócio da Inteligência Artificial. Sua principal função é receber um prompt do usuário e invocar um modelo de IA para gerar uma resposta textual coerente.
- **Atributos:**
  - Nenhum atributo é especificado.
- **Métodos (Operações):**
  - + gerarResposta(prompt: String): String: (Público) Recebe uma string (prompt) como parâmetro e retorna a resposta (String) gerada pela IA.

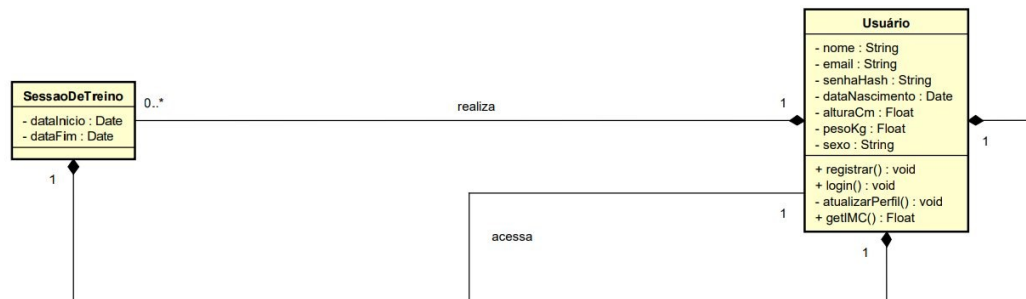
### 5.2.3. Relacionamentos

- **Usuario e RegistroInteracaoIA (Associação)**
  - **Natureza:** Um Usuario **registra** uma ou mais interações. É uma associação direta onde um objeto Usuario está ligado a um conjunto de objetos RegistroInteracaoIA.
  - **Multiplicidade:** 1 para 0..\*. Isso significa que um Usuario pode ter zero ou muitas interações registradas, mas cada RegistroInteracaoIA pertence a exatamente um Usuario.
- **RegistroInteracaoIA e ServicoGeradorIA (Dependência)**
  - **Natureza:** O relacionamento "gerado por" é uma **dependência**, indicada pela linha tracejada. Isso significa que a classe RegistroInteracaoIA (ou a lógica que a cria) depende do ServicoGeradorIA para ser devidamente instanciada. Especificamente, o atributo respostaIA de um RegistroInteracaoIA é o resultado da chamada ao método gerarResposta do ServicoGeradorIA.



### 5.3 Diagrama de Classes Centro Superior

Figura 09: Diagrama de Classes Centro Superior



Fonte: Autores, 2025.

#### 5.3.1. Descrição Geral Centro Superior

Este trecho do diagrama de classes representa a **funcionalidade central de registro de treinos** do sistema. Ele estabelece a relação fundamental entre o **Usuario** da aplicação e as **SessoesDeTreino** que ele realiza. O propósito é modelar como as atividades de treino são associadas a um usuário específico, permitindo o rastreamento de seu histórico de exercícios. Cada **SessaoDeTreino** funciona como um evento temporal, marcado por um início e um fim.

#### 5.3.2. Classes

A seguir, a descrição detalhada de cada classe presente no diagrama.

##### Classe: Usuario

- **Descrição:** Representa o usuário final da aplicação. Esta classe é central para o sistema, armazenando dados pessoais, credenciais de acesso e informações físicas relevantes. Ela também define as operações básicas que um usuário pode realizar em seu perfil.
- **Atributos:**
  - - nome: String: (Privado) Nome completo do usuário.
  - - email: String: (Privado) Endereço de e-mail do usuário, provavelmente usado para login.
  - - senhaHash: String: (Privado) Hash da senha do usuário para armazenamento seguro.
  - - dataNascimento: Date: (Privado) Data de nascimento do usuário.
  - - alturaCm: Float: (Privado) Altura do usuário em centímetros.

- - pesoKg: Float: (Privado) Peso do usuário em quilogramas.
- - sexo: String: (Privado) Sexo/gênero do usuário.
- **Métodos (Operações):**
  - + registrar(): void: (Público) Realiza o cadastro de um novo usuário no sistema.
  - + login(): void: (Público) Autentica o usuário no sistema.
  - + atualizarPerfil(): void: (Público) Permite que o usuário modifique suas informações de perfil.
  - + getIMC(): Float: (Público) Calcula e retorna o Índice de Massa Corporal (IMC) do usuário.

#### **Classe: SessaoDeTreino**

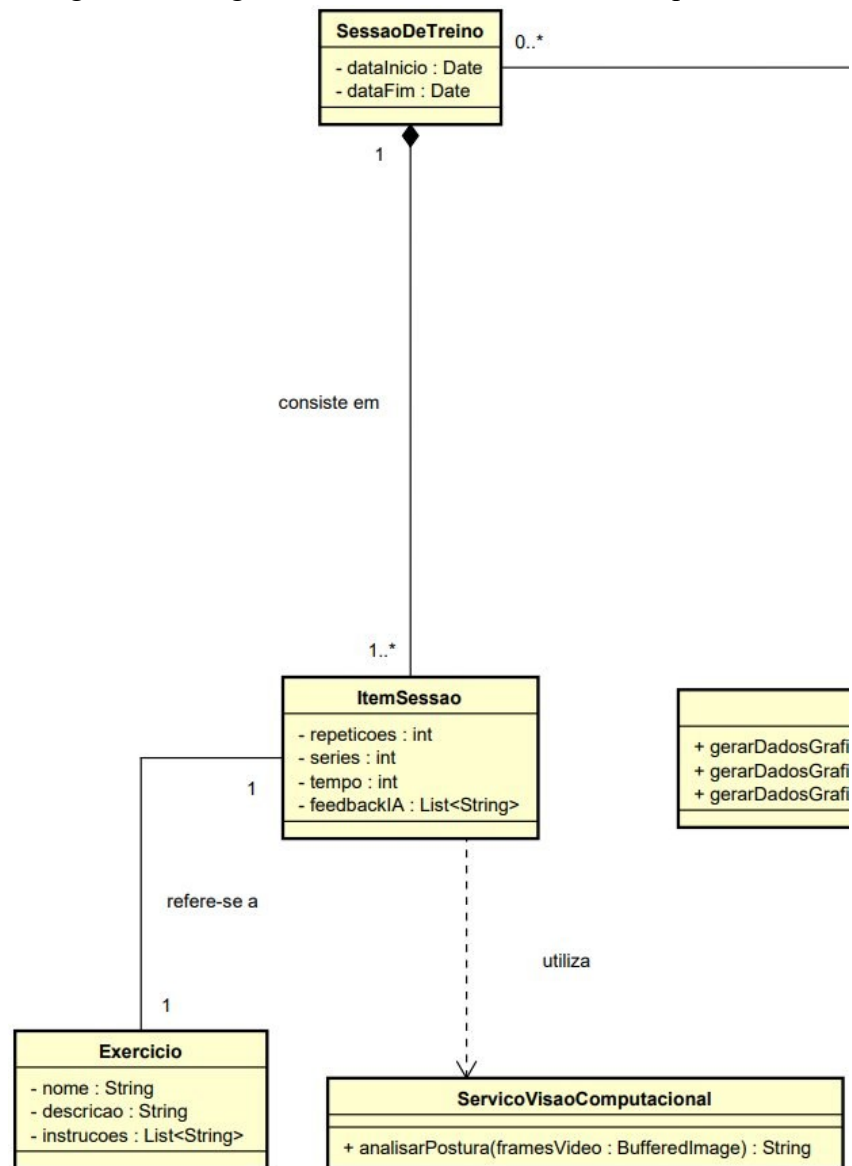
- **Descrição:** Modela uma única sessão de treinamento realizada pelo usuário. Sua principal responsabilidade é registrar a ocorrência de um treino como um evento com duração definida, armazenando quando ele começou e quando terminou.
- **Atributos:**
  - - dataInicio: Date: (Privado) A data e hora de início da sessão de treino.
  - - dataFim: Date: (Privado) A data e hora de término da sessão de treino.
- **Métodos (Operações):**
  - Nenhum método é especificado, indicando que esta é primariamente uma classe para armazenamento de dados do evento de treino.

#### **5.3.3. Relacionamentos**

- **Usuario e SessaoDeTreino (Associação)**
  - **Natureza:** A associação "**realiza**" descreve a ação de um usuário executar uma sessão de treino.
  - **Multiplicidade:** 1 para 0..\*. Isso significa que um Usuario pode realizar **zero ou muitas** SessoesDeTreino ao longo do tempo. Por outro lado, cada SessaoDeTreino deve ser realizada por **exatamente um** Usuario.
- **Outros Relacionamentos (Parciais)**
  - O diagrama mostra um relacionamento "**acessa**" partindo da classe Usuario e outro relacionamento partindo da classe SessaoDeTreino (com multiplicidade 1), mas as classes de destino de ambos não estão visíveis neste recorte.

## 5.4 Diagrama de Classes Extremidade Esquerda

Figura 10: Diagrama de Classes Extremidade Esquerda



Fonte: Autores, 2025.

### 5.4.1. Descrição Geral Extremidade Esquerda

Este segmento do diagrama detalha a **estrutura interna de uma sessão de treino**, focando em como os exercícios são organizados e executados. Ele descreve uma hierarquia clara: uma **SessaoDeTreino** é composta por um ou mais **ItemSessao**. Cada **ItemSessao** representa a execução de um **Exercicio** específico, com metas definidas (como repetições e séries).

Um aspecto central desta parte do sistema é a integração com um serviço de Inteligência Artificial, o **ServicoVisaoComputacional**. Este serviço é utilizado durante a execução de um item de sessão para analisar a postura do usuário e fornecer feedback, tornando o treino mais interativo e seguro.

### 5.4.2. Classes

A seguir, a descrição detalhada de cada classe presente no diagrama.

#### Classe: SessaoDeTreino

- **Descrição:** Modela uma única sessão de treinamento realizada pelo usuário. Sua principal responsabilidade é registrar a ocorrência de um treino como um evento com duração definida, armazenando quando ele começou e quando terminou. Atua como o contêiner principal para todos os exercícios realizados.
- **Atributos:**
  - - dataInicio: Date: (Privado) A data e hora de início da sessão de treino.
  - - dataFim: Date: (Privado) A data e hora de término da sessão de treino.
- **Métodos (Operações):**
  - Nenhum método é especificado.

#### Classe: ItemSessao

- **Descrição:** Representa um único bloco de exercício dentro de uma SessaoDeTreino. Ele detalha os parâmetros de execução para um exercício específico, como o número de repetições, séries e tempo, além de armazenar o feedback gerado pela IA.
- **Atributos:**
  - - repeticoes: int: (Privado) O número de repetições a serem executadas.
  - - series: int: (Privado) O número de séries a serem executadas.
  - - tempo: int: (Privado) O tempo de execução ou de descanso, dependendo do exercício.
  - - feedbackIA: List<String>: (Privado) Uma lista de mensagens de feedback geradas pelo serviço de visão computacional durante a execução.
- **Métodos (Operações):**
  - Nenhum método é especificado.

#### Classe: Exercicio

- **Descrição:** Funciona como uma classe de catálogo ou definição. Ela armazena as informações descritivas e instrutivas sobre um tipo de exercício físico (ex: "Supino", "Agachamento"), sem se preocupar com quantas vezes ele foi executado.
- **Atributos:**
  - - nome: String: (Privado) O nome do exercício.
  - - descricao: String: (Privado) Uma breve descrição sobre o exercício.

- - instrucoes: List<String>: (Privado) Uma lista com o passo a passo de como realizar o exercício corretamente.
- **Métodos (Operações):**
  - Nenhum método é especificado.

#### Classe: ServicoVisaoComputacional

- **Descrição:** É uma classe de serviço especializada, cuja responsabilidade é processar imagens de vídeo para realizar a análise de postura do usuário. Ela oferece uma funcionalidade específica que pode ser consumida por outras partes do sistema.
- **Atributos:**
  - Nenhum atributo é especificado.
- **Métodos (Operações):**
  - + analisarPostura(framesVideo: BufferedImage): String: (Público) Recebe um frame de vídeo (BufferedImage) e retorna uma String contendo o feedback da análise de postura.

#### 5.4.3. Relacionamentos

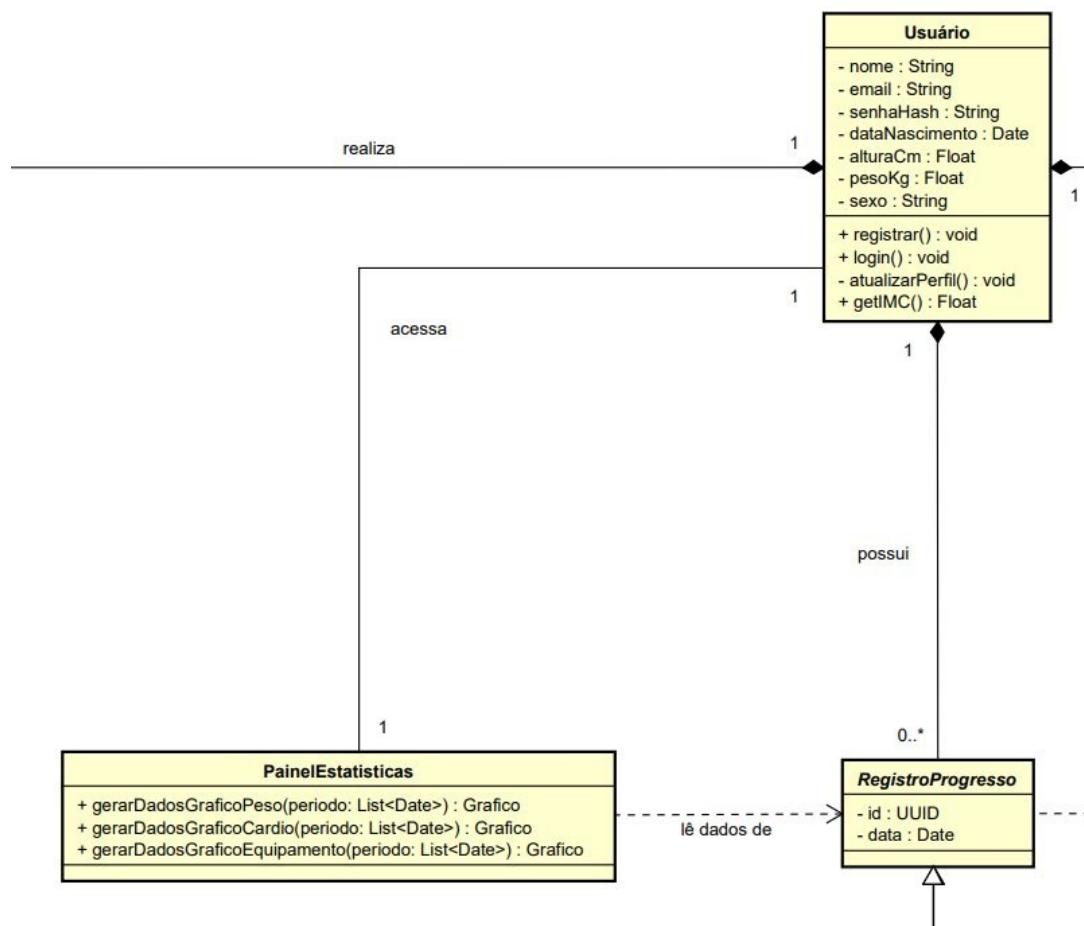
- **SessaoDeTreino e ItemSessao (Composição)**
  - **Natureza:** Uma SessaoDeTreino "**consiste em**" um ou mais ItemSessao. O losango preenchido na extremidade de SessaoDeTreino indica uma **Composição**, que é uma forte relação "todo-parte".
  - **Descrição:** Isso significa que um ItemSessao não pode existir de forma independente; seu ciclo de vida está estritamente ligado ao da SessaoDeTreino. Se a sessão for excluída, todos os seus itens também serão.
  - **Multiplicidade:** 1 para 1..\*. Uma SessaoDeTreino deve conter **pelo menos um** ItemSessao, e cada ItemSessao pertence a **exatamente uma** SessaoDeTreino.
- **ItemSessao e Exercicio (Associação)**
  - **Natureza:** Um ItemSessao "**refere-se a**" um Exercicio. Trata-se de uma associação padrão.
  - **Descrição:** Cada item de uma sessão é a aplicação prática de um exercício definido no catálogo. O ItemSessao representa a execução ("fiz 10 agachamentos"), enquanto o Exercicio representa a definição ("o que é um agachamento").
  - **Multiplicidade:** 1 para 1. Cada ItemSessao refere-se a **exatamente um** Exercicio, e um Exercicio pode ser referenciado por muitos ItemSessao (em diferentes sessões de treino).

- **ItemSessao e ServicoVisaoComputacional (Dependência)**

- **Natureza:** Um ItemSessao "utiliza" o ServicoVisaoComputacional. A linha tracejada indica uma relação de **Dependência**.
- **Descrição:** A classe ItemSessao (ou a lógica que a opera) depende do serviço para executar uma de suas funções, que é a análise de postura. O resultado do método analisarPostura pode ser usado para popular o atributo feedbackIA.

## 5.5 Diagrama de Classes Centro

Figura 11: Diagrama de Classes Centro



Fonte: Autores, 2025.

### 5.5.1. Descrição Geral Centro

Este trecho central do diagrama de classes foca no **acompanhamento e visualização do progresso do usuário**. Ele conecta o Usuario aos seus registros históricos (RegistroProgresso) e à interface onde esses dados são exibidos de forma consolidada (Painelestatisticas).

O fluxo principal é o seguinte: o Usuario possui uma coleção de RegistroProgresso, que são criados ao longo do tempo para marcar diferentes tipos de evolução. Para visualizar esse avanço, o Usuario acessa o PainelEstatisticas, que por sua vez lê os dados dos registros de progresso para gerar gráficos e relatórios.

### 5.5.2. Classes

A seguir, a descrição detalhada das classes introduzidas neste diagrama.

#### Classe: PainelEstatisticas

- **Descrição:** Esta classe atua como um componente de visualização de dados ou um painel de controle (dashboard). Sua responsabilidade é agregar os dados de progresso do usuário e transformá-los em representações gráficas, facilitando a análise de desempenho em diferentes áreas (peso, cardio, etc.).
- **Atributos:**
  - Nenhum atributo é especificado.
- **Métodos (Operações):**
  - + gerarDadosGraficoPeso(periodo: List<Date>): Grafico: (Público) Gera um gráfico da evolução do peso do usuário dentro de um período de tempo especificado. Retorna um objeto do tipo Grafico.
  - + gerarDadosGraficoCardio(periodo: List<Date>): Grafico: (Público) Gera um gráfico do desempenho cardiovascular do usuário em um determinado período. Retorna um objeto do tipo Grafico.
  - + gerarDadosGraficoEquipamento(periodo: List<Date>): Grafico: (Público) Gera um gráfico relacionado ao uso de equipamentos (possivelmente cargas ou tempo) em um período. Retorna um objeto do tipo Grafico.

#### Classe: RegistroProgresso

- **Descrição:** É uma classe fundamental para o rastreamento da evolução do usuário. O nome em itálico sugere que se trata de uma **classe abstrata**, servindo como um modelo base para todos os tipos específicos de registros de progresso (como peso, medidas, etc.). Ela garante que todo registro tenha um identificador único e uma data.
- **Atributos:**
  - - id: UUID: (Privado) Um Identificador Único Universal para garantir que cada registro seja único.
  - - data: Date: (Privado) A data em que o registro de progresso foi feito.
- **Métodos (Operações):**

- Nenhum método é especificado. As operações concretas seriam definidas em suas classes filhas (indicadas pela seta de herança que aponta para baixo, fora do recorte do diagrama).

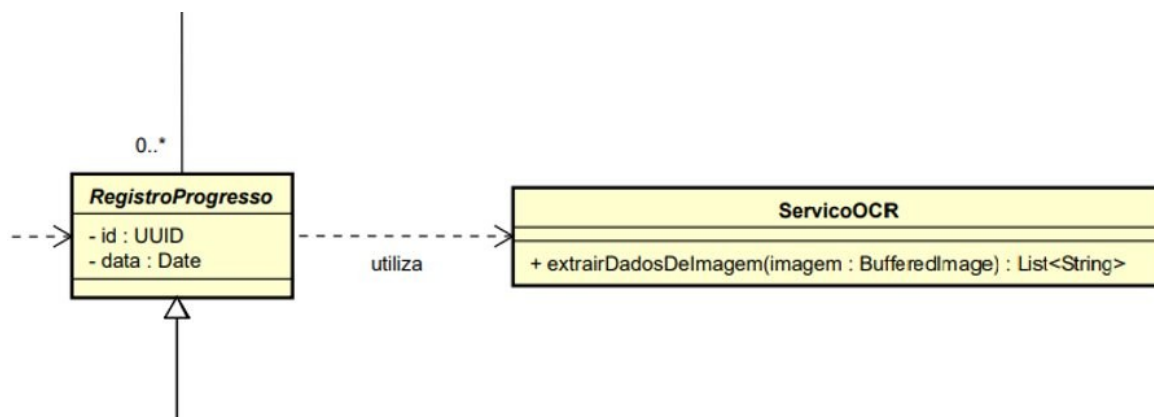
### 5.5.3. Relacionamentos

- **Usuario e PainelEstatisticas (Associação)**
  - **Natureza:** O Usuario (classe já detalhada) "**acessa**" o PainelEstatisticas.
  - **Descrição:** Este relacionamento indica que o usuário interage com o painel para visualizar seus dados. A multiplicidade de 1 para 1 sugere que, no contexto de uma sessão ou da visão de um usuário, ele acessa uma instância de painel que exibe seus dados específicos.
- **Usuario e RegistroProgresso (Associação)**
  - **Natureza:** O Usuario "**possui**" registros de progresso.
  - **Descrição:** É uma relação de posse, onde cada RegistroProgresso pertence a um usuário.
  - **Multiplicidade:** 1 para 0..\*. Um Usuario pode ter **zero ou muitos** RegistroProgresso associados a ele.
- **PainelEstatisticas e RegistroProgresso (Dependência)**
  - **Natureza:** O PainelEstatisticas "**lê dados de**" RegistroProgresso. A linha tracejada indica uma dependência.
  - **Descrição:** Para poder gerar os gráficos, a classe PainelEstatisticas precisa acessar e processar a informação contida nos objetos RegistroProgresso. Se a estrutura de RegistroProgresso mudasse, PainelEstatisticas provavelmente precisaria ser atualizado.



## 5.6 Diagrama de Classes Extremidade Direita

Figura 12: Diagrama de Classes Extremidade Direita



Fonte: Autores, 2025.

### 5.6.1. Descrição Geral Extremidade Direita

Este segmento do diagrama de classes demonstra a integração de uma funcionalidade de automação para a entrada de dados. Ele ilustra como o sistema utiliza um serviço de **Reconhecimento Óptico de Caracteres (OCR)** para popular os registros de progresso.

O fluxo de trabalho sugerido é que, ao criar um novo RegistroProgresso, o sistema pode opcionalmente utilizar o ServicoOCR para extrair informações textuais de uma imagem (por exemplo, uma foto do painel de uma esteira ou de uma balança digital), simplificando a entrada de dados para o usuário.

### 5.6.2. Classes

A seguir, a descrição detalhada da classe apresentada neste diagrama.

#### Classe: ServicoOCR

- **Descrição:** É uma classe de serviço especializada, cuja única responsabilidade é extrair texto de arquivos de imagem. Ela encapsula a lógica complexa do Reconhecimento Óptico de Caracteres, oferecendo uma interface simples para ser consumida por outras partes do sistema.
- **Atributos:**
  - Nenhum atributo é especificado.
- **Métodos (Operações):**
  - + extrairDadosDeImagem(imagem: BufferedImage): List<String>: (Público)  
Recebe um objeto de imagem (BufferedImage) como parâmetro e retorna uma lista de String contendo o texto identificado na imagem.

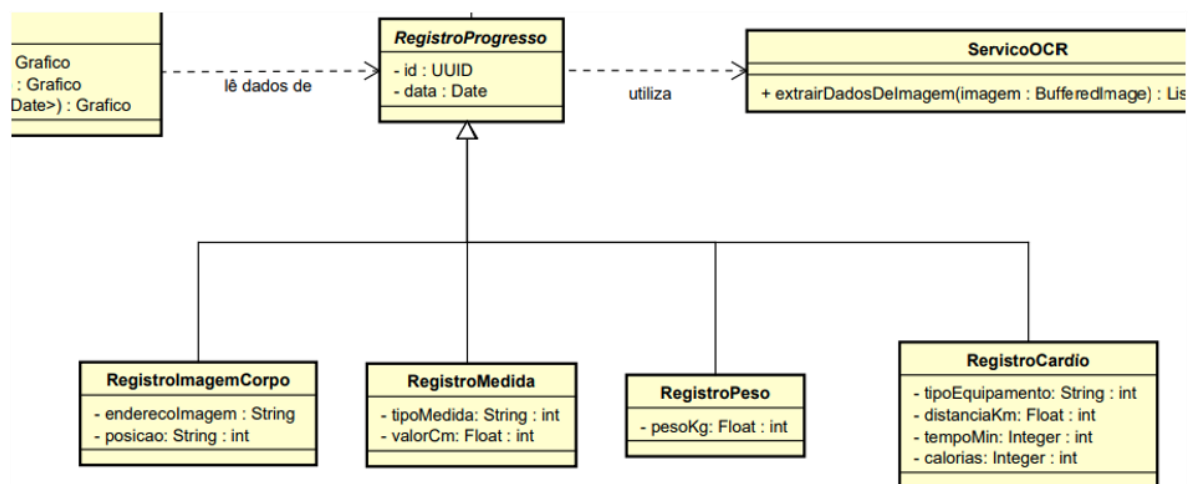
### 5.6.3. Relacionamentos

- **RegistroProgresso e ServicoOCR (Dependência)**

- **Natureza:** A lógica de criação de um RegistroProgresso (classe já detalhada) **"utiliza"** o ServicoOCR. A relação é de **Dependência**, indicada pela linha tracejada.
- **Descrição:** Isso significa que, para executar a funcionalidade de criação de um registro a partir de uma imagem, a aplicação depende do ServicoOCR. O serviço é chamado para processar a imagem e fornecer os dados que serão armazenados no RegistroProgresso.
- 

## 5.7 Diagrama de Classes Canto Inferior Direito

Figura 13: Diagrama de Classes Canto Inferior Direito



Fonte: Autores, 2025.

### 5.7.1. Descrição Geral Canto Inferior Direito

Este segmento final do diagrama detalha a **especialização dos registros de progresso**. Ele revela como a classe abstrata **RegistroProgresso** serve de base para diversas classes concretas, cada uma projetada para armazenar um tipo específico de dado de evolução do usuário.

A estrutura de herança aqui demonstrada é o pilar do sistema de acompanhamento, permitindo que o progresso seja registrado de múltiplas formas — através de fotos (**RegistroImagemCorpo**), medidas corporais (**RegistroMedida**), pesagem (**RegistroPeso**) e dados de atividades cardiovasculares (**RegistroCardio**). Essa abordagem permite que o sistema trate todos os registros de forma genérica quando necessário (por exemplo, ao listá-los por data), mas também utilize os detalhes específicos de cada tipo para análises e gráficos.

### 5.7.2. Classes

A seguir, a descrição detalhada das classes concretas que herdam de RegistroProgresso.

#### Classe: RegistroImagemCorpo

- **Descrição:** Armazena uma referência a uma imagem corporal do usuário, permitindo o acompanhamento visual do progresso.
- **Atributos:**
  - - enderecoImagem: String: (Privado) O caminho ou URL para o arquivo de imagem armazenado.
  - - posicao: String: (Privado) Descreve a pose na foto (ex: "Frente", "Lado", "Costas") para padronização.
- **Atributos Herdados:** Herda id e data de RegistroProgresso.

#### Classe: RegistroMedida

- **Descrição:** Representa o registro de uma medida corporal específica, como cintura, bíceps ou peito.
- **Atributos:**
  - - tipoMedida: String: (Privado) O tipo de medida registrada (ex: "Cintura", "Quadril").
  - - valorCm: Float: (Privado) O valor da medida em centímetros.
- **Atributos Herdados:** Herda id e data de RegistroProgresso.

#### Classe: RegistroPeso

- **Descrição:** Especializada em armazenar um registro de peso do usuário em uma data específica.
- **Atributos:**
  - - pesoKg: Float: (Privado) O valor do peso do usuário em quilogramas.
- **Atributos Herdados:** Herda id e data de RegistroProgresso.

#### Classe: RegistroCardio

- **Descrição:** Modela os dados de uma sessão de exercício cardiovascular.
- **Atributos:**
  - - tipoEquipamento: String: (Privado) O equipamento utilizado (ex: "Esteira", "Bicicleta").
  - - distanciaKm: Float: (Privado) A distância percorrida na atividade, em quilômetros.

- - tempoMin: Integer: (Privado) A duração total da atividade, em minutos.
- - calorias: Integer: (Privado) A estimativa de calorias queimadas durante a atividade.
- **Atributos Herdados:** Herda id e data de RegistroProgresso.

### 5.7.3. Relacionamentos

- **RegistroProgresso e suas Subclasses (Generalização/Herança)**
  - **Natureza: Generalização.** As classes RegistroImagemCorpo, RegistroMedida, RegistroPeso e RegistroCardio são filhas da classe RegistroProgresso. A seta com triângulo vazio apontando para RegistroProgresso indica essa relação de herança.
  - **Descrição:** Cada uma das classes filhas "é um tipo de" RegistroProgresso. Elas herdam todos os atributos e métodos de RegistroProgresso (neste caso, id e data) e adicionam seus próprios atributos específicos. Isso permite que o sistema trate todos esses objetos de forma polimórfica. Por exemplo, o PainelEstatisticas pode ler uma lista de RegistroProgresso sem precisar saber o tipo concreto de cada um, acessando a data de todos para montar uma linha do tempo.

## 6 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência modela a interação entre objetos ao longo do tempo para realizar uma funcionalidade específica. Ele é útil para entender a dinâmica do sistema e garantir que as mensagens trocadas entre os objetos estão corretas. É possível identificar as dependências, as trocas de informações e as operações realizadas ao visualizar e compreender o fluxo de execução do sistema. Essa abordagem facilita a análise, o design, a implementação e a depuração do software.

### Componentes Principais:

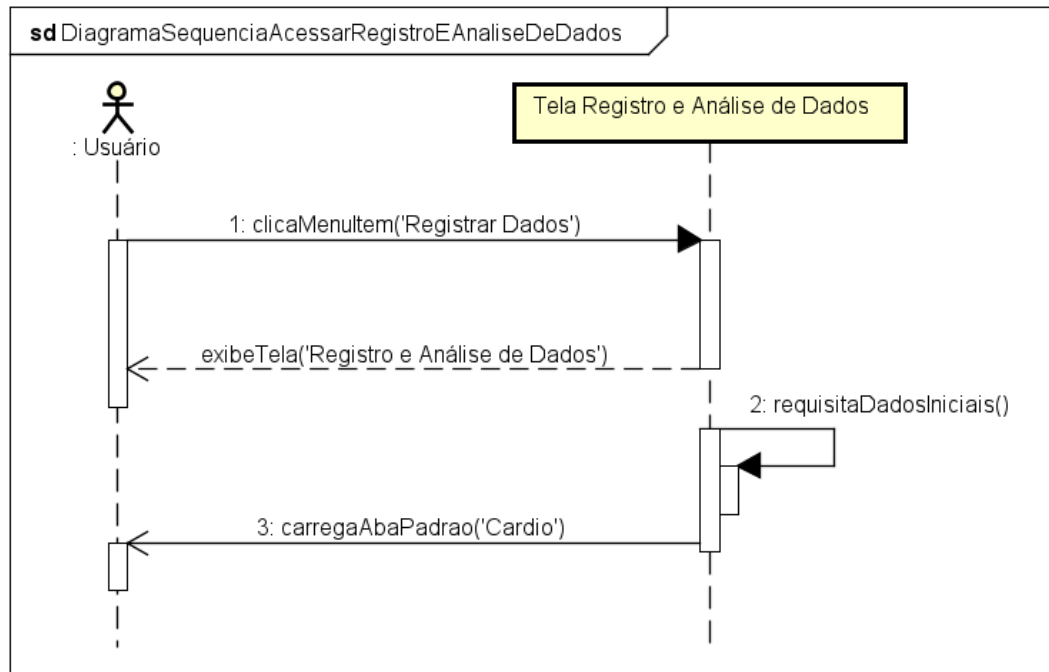
1. **Objetos:** Instâncias das classes participantes da interação.
2. **Lifelines:** Representam a existência dos objetos durante a interação.
3. **Mensagens:** Mostram a comunicação entre objetos na forma de chamadas de métodos ou troca de dados.

Nas figuras a seguir, temos os diagramas de sequência que irão compor o sistema:

### 6.1 Diagrama de Sequência: Acessar Opções em Registrar Dados

Este diagrama de sequência ilustra o cenário de um usuário acessando a funcionalidade de registro e análise de dados em um sistema. A interação começa com a ação do usuário de selecionar a opção correspondente em um menu e culmina com a exibição da tela e o carregamento dos dados iniciais para a análise.

Figura 14: Acessar Opções em Registrar Dados



Fonte: Autores, 2025.

#### Participantes (Lifelines)

- **: Usuário:** O ator que inicia a interação. Representa a entidade humana que está utilizando o sistema.
- **Tela Registro e Análise de Dados:** O objeto que representa a interface de usuário (UI) para a funcionalidade de registro e análise de dados. É responsável por exibir as informações e interagir com outras partes do sistema para obter os dados.

#### Sequência de Mensagens

##### 1. Mensagem: clicaMenuItem('Registrar Dados')

- **Remetente:** : Usuário
- **Destinatário:** Tela Registro e Análise de Dados
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário inicia o processo clicando na opção de menu "Registrar Dados".

##### 2. Mensagem: exibeTela('Registro e Análise de Dados')

- **Remetente:** Tela Registro e Análise de Dados
  - **Destinatário:** : Usuário
  - **Tipo:** Retorno (implícito).
  - **Descrição:** A tela é exibida para o usuário como resposta à sua ação. Este passo é mostrado como uma linha tracejada de retorno, indicando que a tela se torna visível.
3. **Mensagem:** requisitaDadosIniciais()
- **Remetente:** Tela Registro e Análise de Dados
  - **Destinatário:** Tela Registro e Análise de Dados (auto-chamada)
  - **Tipo:** Chamada síncrona.
  - **Descrição:** Após ser exibida, a própria tela invoca um método para buscar os dados que serão exibidos por padrão.
4. **Mensagem:** carregaAbaPadrao('Cardio')
- **Remetente:** Tela Registro e Análise de Dados
  - **Destinatário:** : Usuário
  - **Tipo:** Retorno (implícito).
  - **Descrição:** Após a requisição de dados, a tela carrega a aba padrão "Cardio" e a exibe para o usuário, completando a inicialização da interface.

## 6.2 Diagrama de Sequência: Acessar Registro e Análise de Dados

Este diagrama de sequência ilustra a interação do usuário ao selecionar uma aba na tela "Registro e Análise de Dados". O cenário específico detalhado é o carregamento do conteúdo correspondente à aba de "Peso" após o usuário clicar nela.

### 1. Participantes (Lifelines)

- **: Usuário:** O ator que interage com a interface do sistema.
- **Tela Registro e Análise de Dados:** O objeto que representa a interface gráfica do usuário, responsável por receber a ação do usuário, processar a requisição e exibir o conteúdo atualizado.

### 3. Sequência de Mensagens

2. **Mensagem:** clicaAba(Medida, Peso, Cardio)
- **Remetente:** : Usuário
  - **Destinatário:** Tela Registro e Análise de Dados
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O usuário inicia a interação clicando em uma das abas disponíveis (Medida, Peso, Cardio).

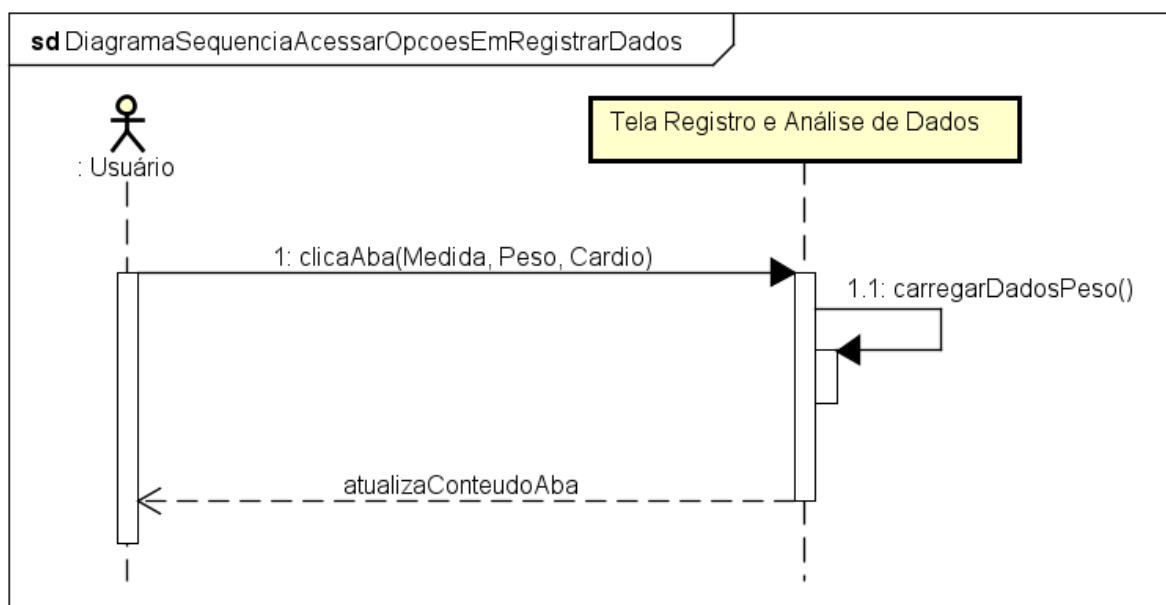
### 3. Mensagem: carregarDadosPeso()

- **Remetente:** Tela Registro e Análise de Dados
- **Destinatário:** Tela Registro e Análise de Dados (auto-chamada)
- **Tipo:** Chamada síncrona.
- **Descrição:** Como resultado da ação do usuário (clique na aba "Peso"), a própria tela invoca um método para carregar os dados específicos relacionados a peso.

### 4. Mensagem: atualizaConteudoAba

- **Remetente:** Tela Registro e Análise de Dados
- **Destinatário:** : Usuário
- **Tipo:** Retorno (implícito).
- **Descrição:** A tela retorna para o usuário a atualização do seu conteúdo, exibindo os dados de peso carregados na aba correspondente.

Figura 15: Acessar Registro e Análise de Dados

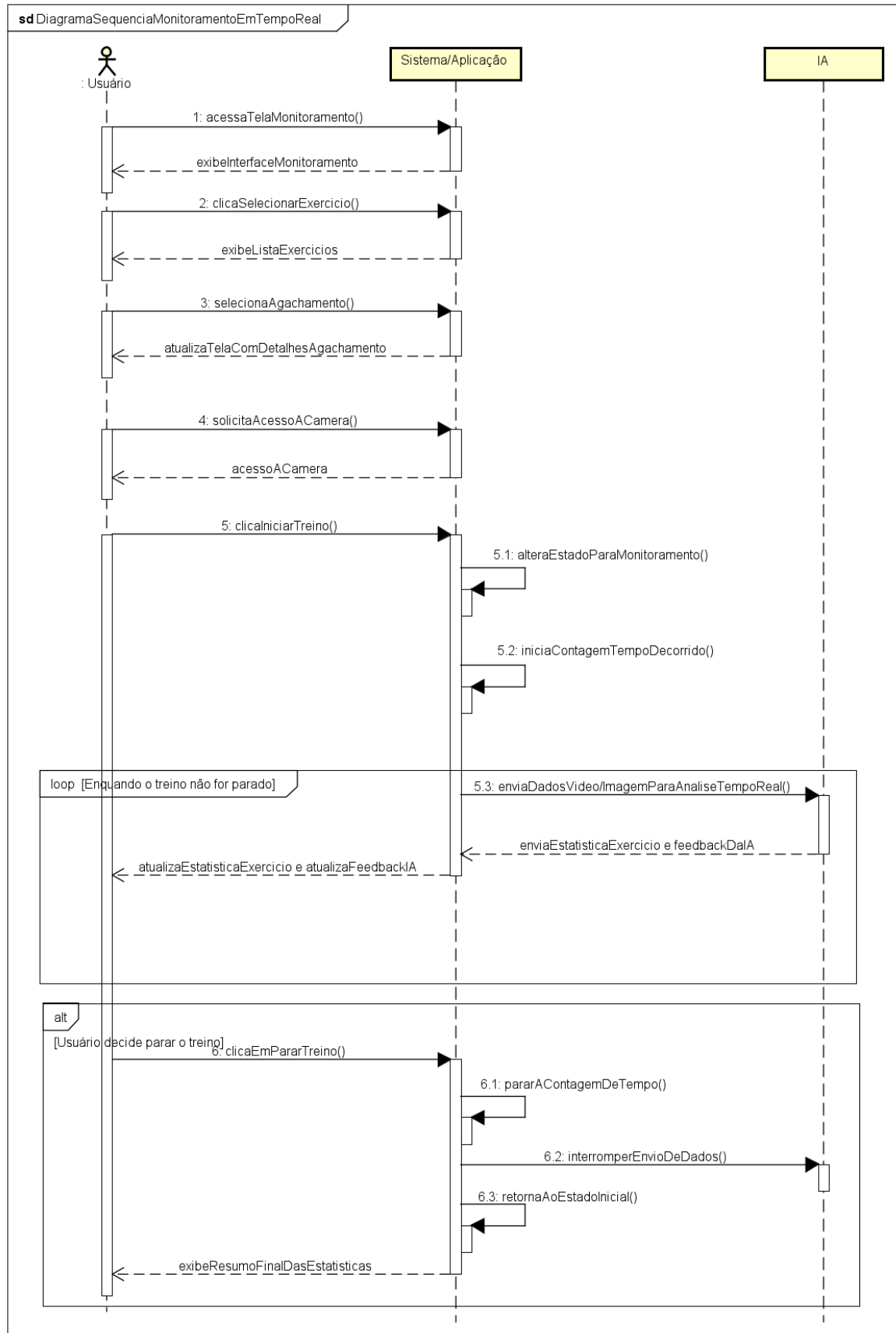


Fonte: Autores, 2025.

## 6.3 Diagrama de Sequência: Monitoramento em Tempo Real

Este diagrama de sequência ilustra o cenário de um usuário iniciando um treino com monitoramento em tempo real em uma aplicação. O processo inclui a navegação inicial, a seleção de um exercício, o acesso à câmera, o início do monitoramento (que envolve um loop de envio e recebimento de dados com uma inteligência artificial) e a opção de finalizar o treino.

Figura 16: Monitoramento em Tempo Real



Fonte: Autores, 2025.



## Participantes (Lifelines)

- **: Usuário:** O ator que interage com a aplicação para iniciar e finalizar o monitoramento do treino.
- **Sistema/Aplicação:** O objeto que representa a interface e a lógica de controle da aplicação. É responsável por gerenciar a interação com o usuário e a comunicação com a IA.
- **IA:** O objeto que representa a Inteligência Artificial, responsável por analisar os dados de vídeo/imagem em tempo real e fornecer feedback estatístico.

## 3. Sequência de Mensagens

### 1. Mensagem: acessaTelaMonitoramento()

- **Remetente:** : Usuário
- **Destinatário:** Sistema/Aplicação
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário inicia o processo acessando a tela de monitoramento.

### 2. Mensagem: exibeInterfaceMonitoramento

- **Remetente:** Sistema/Aplicação
- **Destinatário:** : Usuário
- **Tipo:** Retorno (implícito).
- **Descrição:** O sistema exibe a interface de monitoramento ao usuário.

### 3. Mensagem: clicaSelecionarExercicio()

- **Remetente:** : Usuário
- **Destinatário:** Sistema/Aplicação
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário interage para selecionar um exercício a ser monitorado.

### 4. Mensagem: exibeListaExercicios

- **Remetente:** Sistema/Aplicação
- **Destinatário:** : Usuário
- **Tipo:** Retorno (implícito).
- **Descrição:** O sistema apresenta a lista de exercícios disponíveis.

5. **Mensagem:** selecionaAgachamento()

- **Remetente:** : Usuário
- **Destinatário:** Sistema/Aplicação
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário escolhe o exercício "Agachamento" da lista.

#### 6.4 Diagrama de Sequência: Obter Dicas Fitness

Este diagrama de sequência ilustra a interação de um usuário com o sistema para obter uma "Dica Fitness do Dia". O cenário descreve o processo desde o acesso à tela de dicas até o clique no botão para solicitar uma dica e a subsequente exibição da dica na tela.

##### Participantes (Lifelines)

- **: Usuário:** O ator que interage com a interface do sistema para solicitar e visualizar as dicas fitness.
- **Sistema (Interface):** O objeto que representa a interface de usuário da aplicação. É responsável por receber as ações do usuário, gerenciar a exibição de telas e coordenar a comunicação com o gerador de dicas.
- **Gerador de Dicas Fitness:** O objeto que representa a lógica de negócio ou o serviço responsável por gerar e fornecer as dicas fitness.

#### 3. Sequência de Mensagens

1. **Mensagem:** acessaTelaIdeiasFit()

- **Remetente:** : Usuário
- **Destinatário:** Sistema (Interface)
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário inicia o processo acessando a tela de dicas fitness.

2. **Mensagem:** exhibeTelaIdeiasFit

- **Remetente:** Sistema (Interface)
- **Destinatário:** : Usuário
- **Tipo:** Retorno (implícito).
- **Descrição:** O sistema exhibe a tela de dicas fitness para o usuário.

3. **Mensagem:** clicaBotaoObterDicaFitnessDia()

- **Remetente:** : Usuário

- **Destinatário:** Sistema (Interface)
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário clica no botão para solicitar uma dica fitness diária.

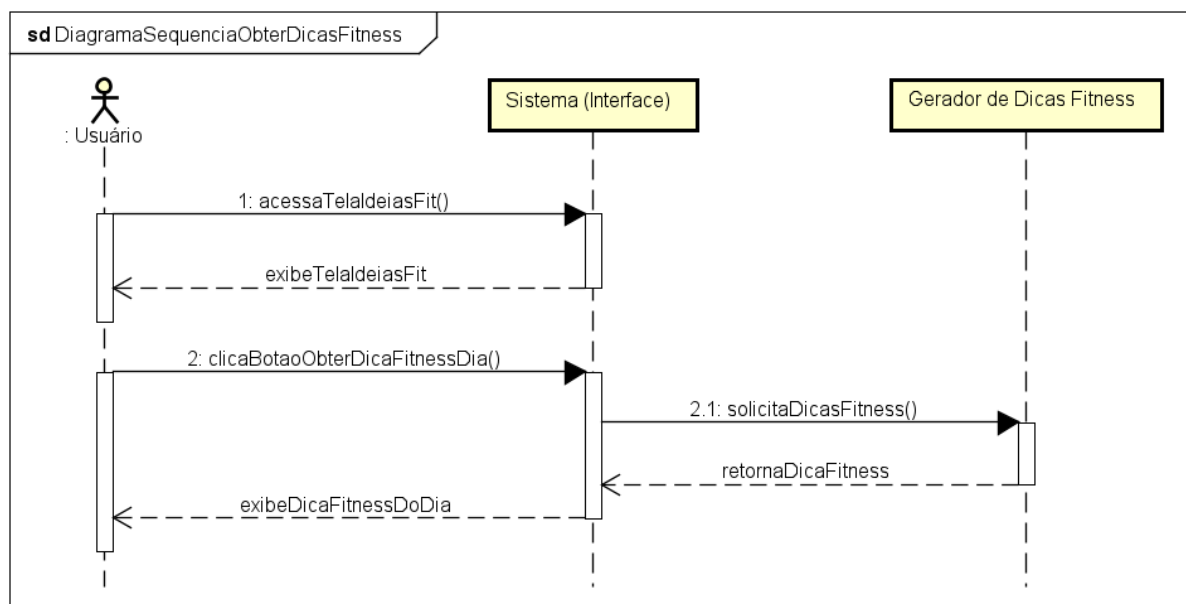
4. **Mensagem:** solicitaDicasFitness()

- **Remetente:** Sistema (Interface)
- **Destinatário:** Gerador de Dicas Fitness
- **Tipo:** Chamada síncrona.
- **Descrição:** A interface do sistema envia uma requisição para o Gerador de Dicas Fitness.

5. **Mensagem:** retornaDicaFitness

- **Remetente:** Gerador de Dicas Fitness
- **Destinatário:** Sistema (Interface)
- **Tipo:** Retorno (implícito).
- **Descrição:** O Gerador de Dicas Fitness envia a dica solicitada de volta para a interface do sistema.

Figura 17: Obter Dicas Fitness



Fonte: Autores, 2025.

## 6.5 Diagrama de Sequência: Obter Treino/Dieta Personalizada

Este diagrama de sequência ilustra o processo de um usuário obtendo um plano de treino ou dieta personalizado. A interação começa com o usuário acessando a tela de ideias, inserindo uma solicitação (prompt) e, em seguida, o sistema coordenando a comunicação com um gerador de treino/dieta, que por sua vez utiliza um módulo de Inteligência Artificial para gerar o plano final.

### Participantes (Lifelines)

- **: Usuário:** O ator que interage com a interface do sistema, fornecendo as informações necessárias para a personalização.
- **Sistema (Interface):** O objeto que representa a interface de usuário da aplicação. Ele é responsável por receber a entrada do usuário e exibir o resultado final.
- **Gerador de Treino/Dieta:** O objeto que representa o serviço de lógica de negócio responsável por intermediar a solicitação do usuário e o módulo de IA, formatando a entrada e a saída de dados.
- **Módulo de IA:** O objeto que representa o serviço de Inteligência Artificial, responsável por processar o "prompt" do usuário e gerar o plano de treino ou dieta personalizado.

### 3. Sequência de Mensagens

1. **Mensagem:** `AcessaTelaIdeiasFit()`
  - **Remetente:** : Usuário
  - **Destinatário:** Sistema (Interface)
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O usuário inicia a interação acessando a tela de ideias fitness.
2. **Mensagem:** `exibeTelaIdeiasFit`
  - **Remetente:** Sistema (Interface)
  - **Destinatário:** : Usuário
  - **Tipo:** Retorno (implícito).
  - **Descrição:** O sistema exibe a interface de "Ideias Fitness" para o usuário.
3. **Mensagem:** `digitaPromptNoCampoDeTexto()`
  - **Remetente:** : Usuário
  - **Destinatário:** Sistema (Interface)
  - **Tipo:** Chamada síncrona.

- **Descrição:** O usuário insere um texto (prompt) no campo de entrada para descrever o tipo de treino ou dieta que deseja.

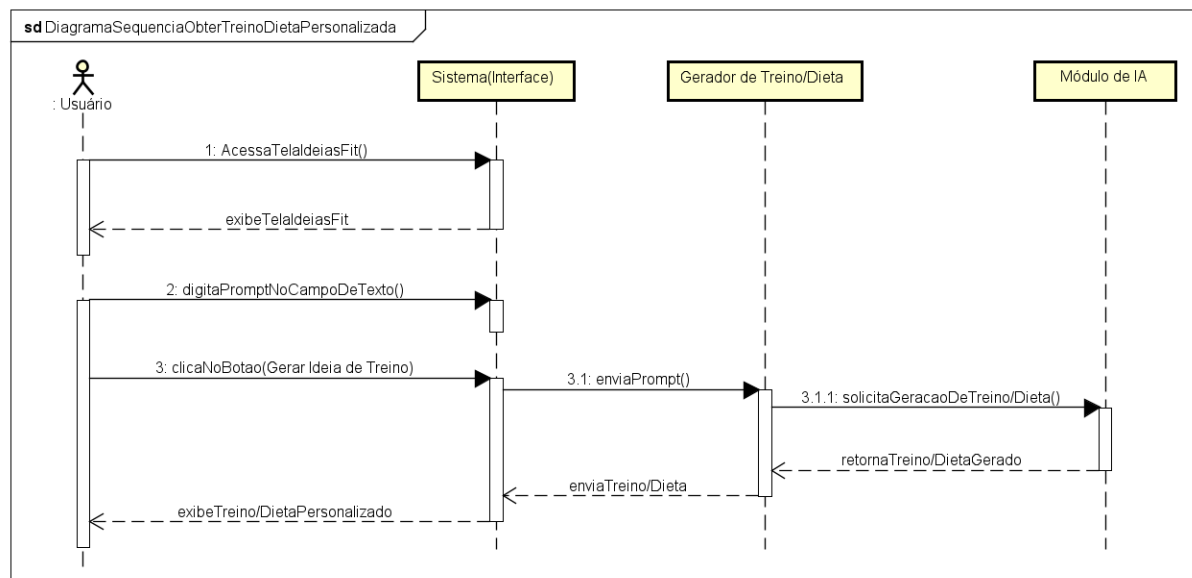
4. **Mensagem:** `clicaNoBotao(Gerar Ideia de Treino)`

- **Remetente:** : Usuário
- **Destinatário:** Sistema (Interface)
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário clica no botão para iniciar o processo de geração do plano personalizado.

5. **Mensagem:** `enviaPrompt()`

- **Remetente:** Sistema (Interface)
- **Destinatário:** Gerador de Treino/Dieta
- **Tipo:** Chamada síncrona.
- **Descrição:** O sistema envia o prompt do usuário para o serviço de geração de treino/dieta.

Figura 18: Obter Treino/Dieta Personalizada



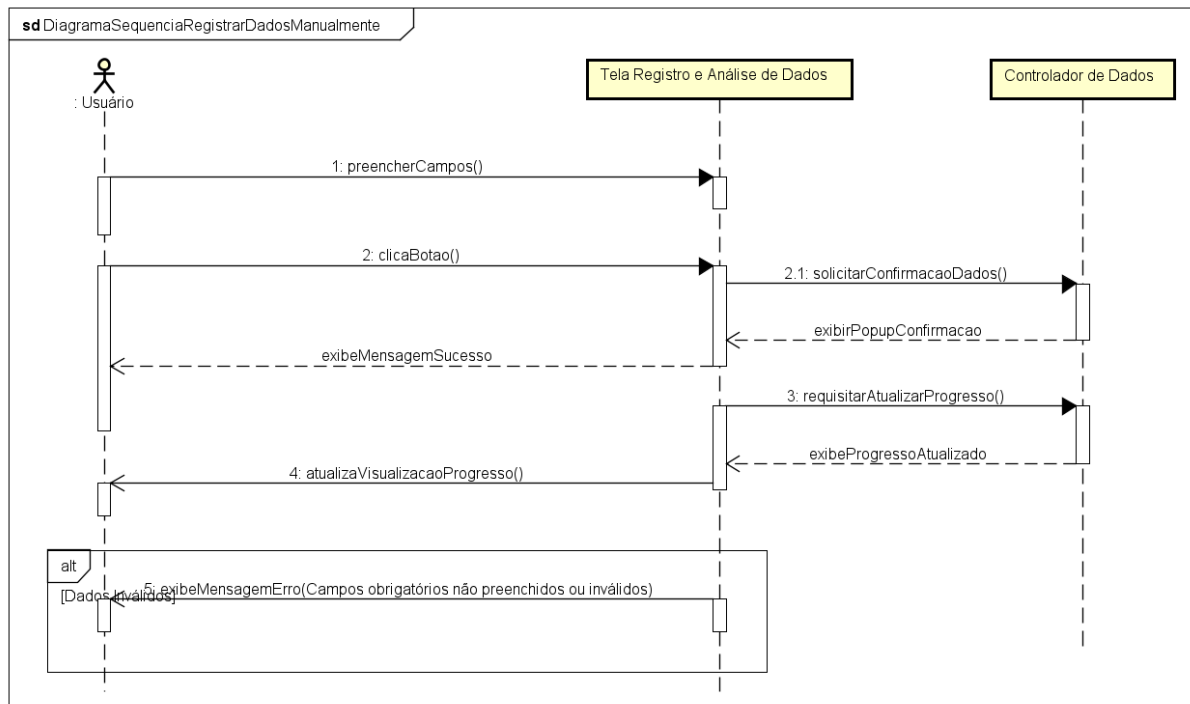
Fonte: Autores, 2025.

## 6.6 Diagrama de Sequência: Registrar Dados Manualmente

Este diagrama de sequência ilustra o processo de um usuário registrando dados manualmente em uma aplicação. O cenário inclui o preenchimento de campos, o clique no botão

de envio, a validação dos dados (que pode ter um resultado de sucesso ou erro), e a subsequente atualização do progresso do usuário.

Figura 19: Registrar Dados Manualmente



Fonte: Autores, 2025.

### Participantes (Lifelines)

- **: Usuário:** O ator que interage com a interface para inserir e registrar os dados.
- **Tela Registro e Análise de Dados:** O objeto que representa a interface gráfica do usuário. Ele recebe a entrada do usuário e coordena a comunicação com o controlador de dados.
- **Controlador de Dados:** O objeto que encapsula a lógica de negócio para validar e persistir os dados. Ele é responsável por gerenciar a confirmação e a atualização do progresso.

### 3. Sequência de Mensagens

#### 1. Mensagem: preencherCampos()

- **Remetente:** : Usuário
- **Destinatário:** Tela Registro e Análise de Dados
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário insere as informações nos campos da tela.

2. **Mensagem:** `clicaBotao()`

- **Remetente:** : Usuário
- **Destinatário:** Tela Registro e Análise de Dados
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário clica no botão de registro para enviar os dados.

3. **Mensagem:** `solicitarConfirmacaoDados()`

- **Remetente:** Tela Registro e Análise de Dados
- **Destinatário:** Controlador de Dados
- **Tipo:** Chamada síncrona.
- **Descrição:** A interface solicita ao controlador de dados que confirme e processe os dados inseridos.

4. **Mensagem:** `exibirPopupConfirmacao`

- **Remetente:** Controlador de Dados
- **Destinatário:** Tela Registro e Análise de Dados
- **Tipo:** Retorno (implícito).
- **Descrição:** O controlador retorna uma mensagem para que a interface exiba um pop-up de confirmação.

5. **Mensagem:** `exibeMensagemSucesso`

- **Remetente:** Tela Registro e Análise de Dados
- **Destinatário:** : Usuário
- **Tipo:** Retorno (implícito).
- **Descrição:** Em caso de sucesso, a tela exibe uma mensagem para o usuário indicando que os dados foram registrados.

## 6.7 Diagrama de Sequência: Visualizar Progresso de Registrar Dados

Este diagrama de sequência ilustra o processo de um usuário visualizando seu progresso de medidas registradas. O cenário abrange a ação do usuário de acessar a aba ou tela de "Medidas", a requisição de dados ao controlador, e a apresentação do resultado na interface, que pode ser um gráfico/tabela (se houver dados) ou uma mensagem de ausência de dados.

### Participantes (Lifelines)

- **: Usuário:** O ator que interage com a interface para visualizar o progresso.
- **Tela Registro e Análise de Dados:** O objeto que representa a interface de usuário. Ele recebe a ação do usuário e coordena a busca e a exibição dos dados.
- **Controlador de Dados:** O objeto que encapsula a lógica de negócio para buscar e fornecer os dados de medidas do usuário.

### 3. Sequência de Mensagens

1. **Mensagem:** `clicaAba("Medidas")` ou `AcessaTela()`
  - **Remetente:** : Usuário
  - **Destinatário:** Tela Registro e Análise de Dados
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O usuário inicia a interação clicando na aba "Medidas" ou acessando a tela correspondente.
2. **Mensagem:** `carregarProgressoMedidas()`
  - **Remetente:** Tela Registro e Análise de Dados
  - **Destinatário:** Controlador de Dados
  - **Tipo:** Chamada síncrona.
  - **Descrição:** A tela solicita ao controlador de dados para carregar o progresso de medidas do usuário.
3. **Mensagem:** `buscarMedidasUsuario()`
  - **Remetente:** Controlador de Dados
  - **Destinatário:** Controlador de Dados (auto-chamada)
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O controlador executa sua lógica interna para buscar as medidas do usuário.
4. **Mensagem:** `retornaMedidas()`
  - **Remetente:** Controlador de Dados
  - **Destinatário:** Tela Registro e Análise de Dados
  - **Tipo:** Retorno (implícito).

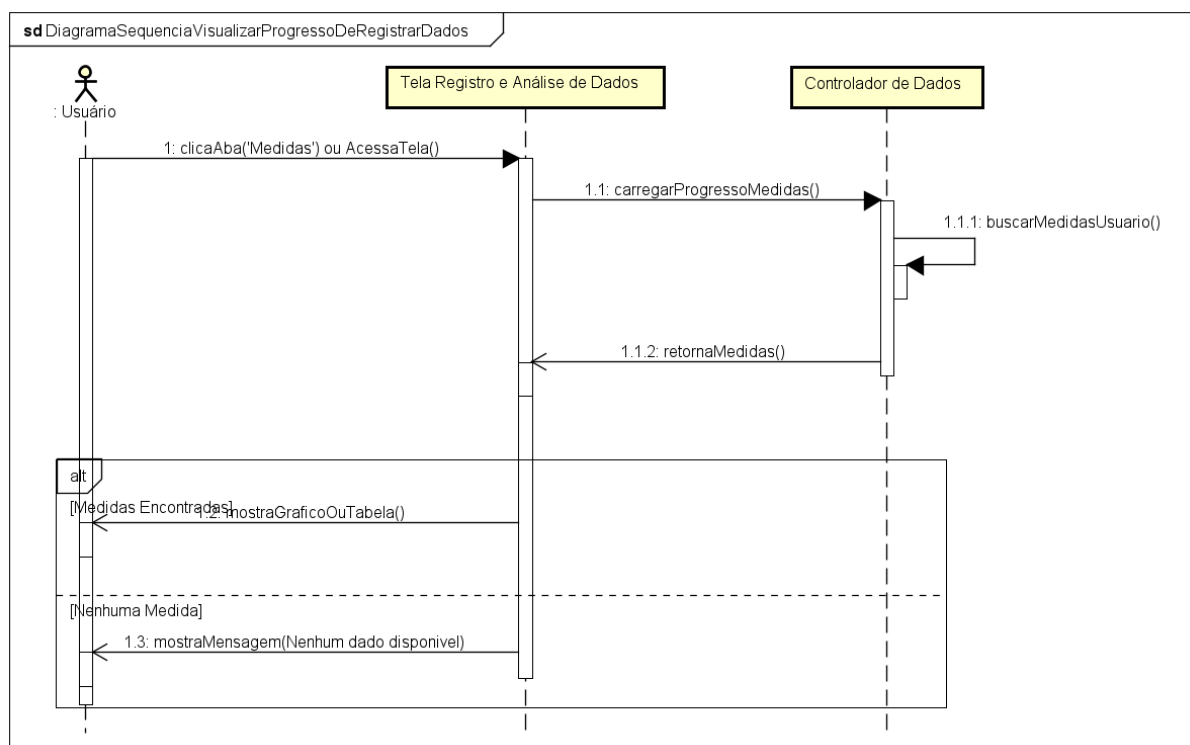


- **Descrição:** O controlador de dados retorna a lista de medidas encontradas para a tela.

5. **Mensagem:** mostraGraficoOuTabela()

- **Remetente:** Tela Registro e Análise de Dados
- **Destinatário:** : Usuário
- **Tipo:** Chamada síncrona (dentro do fragmento alt).
- **Descrição:** Se medidas foram encontradas, a tela exibe os dados em formato de gráfico ou tabela para o usuário.

Figura 20: Visualizar Progresso de Registrar Dados



Fonte: Autores, 2025.

## 6.8 Diagrama de Sequência: Realiza Login com Google

Este diagrama de sequência ilustra o processo de login de um usuário no "FITAI" utilizando a autenticação via Google. O cenário detalha os passos desde o clique inicial do usuário, passando pela interação com o serviço de autenticação do Google e a verificação no banco de dados do FITAI, até os possíveis fluxos de sucesso e falha, como a negação da autorização ou problemas de conexão.

### Participantes (Lifelines)

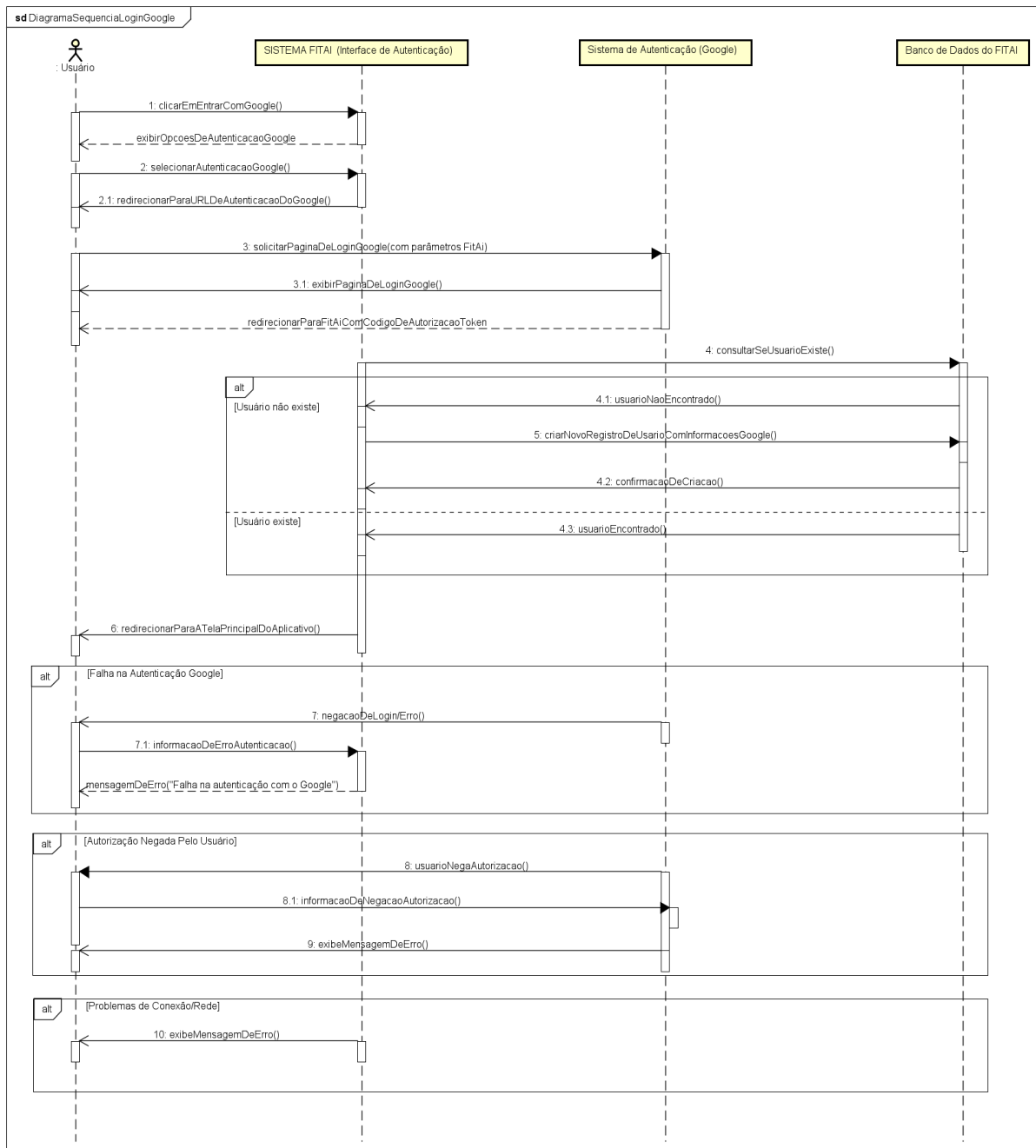
- **: Usuário:** O ator que interage com a interface para iniciar e concluir o processo de login.
- **SISTEMA FITAI (Interface de Autenticação):** O objeto que representa a interface de autenticação do sistema FITAI. É responsável por exibir as opções de login e gerenciar as redireções.
- **Sistema de Autenticação (Google):** O objeto que representa o serviço de autenticação de terceiros (Google) que gerencia o login e a autorização do usuário.
- **Banco de Dados do FITAI:** O objeto que representa o banco de dados do sistema FITAI, onde as informações dos usuários são armazenadas.

### 3. Sequência de Mensagens

1. **Mensagem:** clicarEmEntrarComGoogle()
  - **Remetente:** : Usuário
  - **Destinatário:** SISTEMA FITAI (Interface de Autenticação)
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O usuário inicia o processo clicando na opção de "Entrar com Google".
2. **Mensagem:** exibeOOpcoesDeAutenticacaoGoogle
  - **Remetente:** SISTEMA FITAI (Interface de Autenticação)
  - **Destinatário:** : Usuário
  - **Tipo:** Retorno (implícito).
  - **Descrição:** A interface de autenticação exibe as opções de login do Google.
3. **Mensagem:** selecionarAutenticacaoGoogle()
  - **Remetente:** : Usuário
  - **Destinatário:** SISTEMA FITAI (Interface de Autenticação)

- **Tipo:** Chamada síncrona.
  - **Descrição:** O usuário seleciona a autenticação com o Google.
4. **Mensagem:** `redirecionaParaURLDeAutenticacaoDoGoogle()`
- **Remetente:** SISTEMA FITAI (Interface de Autenticação)
  - **Destinatário:** : Usuário
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O sistema redireciona o navegador do usuário para a URL de autenticação do Google.
5. **Mensagem:** `solicitarPaginaDeLoginGoogle(com parâmetros FITAI)`
- **Remetente:** : Usuário
  - **Destinatário:** Sistema de Autenticação (Google)
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O navegador do usuário solicita a página de login do Google, incluindo os parâmetros do sistema FITAI.

Figura 21: Realiza Login com Google



Fonte: Autores, 2025.

## 6.9 Diagrama de Sequência: Realiza Sair/Logout do Sistema

Este diagrama de sequência ilustra o processo de logout de um usuário no sistema FITAI, que utiliza autenticação via Google. O cenário principal envolve a invalidação do token de sessão no sistema. Um fluxo alternativo e estendido é apresentado para a revogação do acesso do aplicativo no serviço do Google, o que seria uma opção mais completa de "logout".

### Participantes (Lifelines)

- **: Usuário:** O ator que interage com a interface para iniciar o processo de logout.
- **Sistema FITAI:** O objeto que representa a lógica de controle da aplicação, responsável por gerenciar a sessão do usuário e coordenar a revogação de acesso.
- **Sistema de Autenticação (Google):** O objeto que representa o serviço de autenticação de terceiros (Google), responsável por gerenciar a revogação de tokens de acesso.
- **Banco de Dados do FITAI:** O objeto que representa o banco de dados do sistema FITAI, onde as referências de autenticação do usuário são armazenadas.

### 3. Sequência de Mensagens

1. **Mensagem:** clicarEmSair/logout()
  - **Remetente:** : Usuário
  - **Destinatário:** Sistema FITAI
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O usuário inicia o processo de logout clicando no botão de "Sair" ou "Logout".
2. **Mensagem:** invalidarTokenDeSessao()
  - **Remetente:** Sistema FITAI
  - **Destinatário:** Sistema FITAI (auto-chamada)
  - **Tipo:** Chamada síncrona.
  - **Descrição:** O sistema invalida o token de sessão do usuário, encerrando o acesso imediato.
3. **Mensagem:** confirmacaoDeInvalidacao
  - **Remetente:** Sistema FITAI
  - **Destinatário:** Sistema FITAI (auto-chamada)
  - **Tipo:** Retorno (implícito).
  - **Descrição:** O sistema confirma que a invalidação do token foi bem-sucedida.

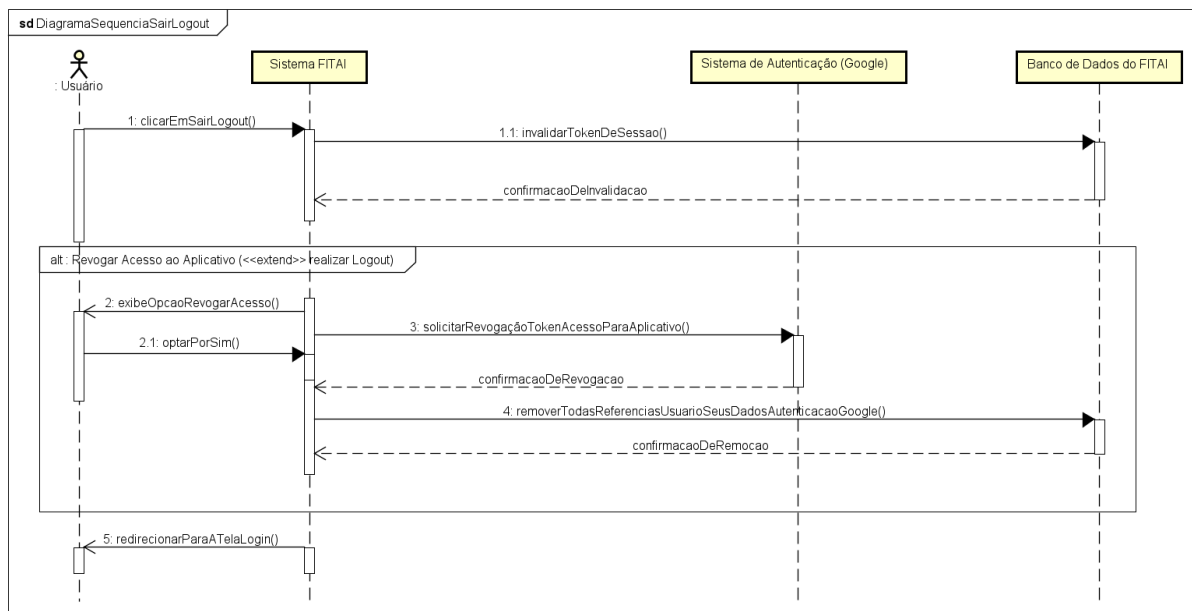
4. **Mensagem:** `exibeOpcaoRevogarAcesso()`

- **Remetente:** Sistema FITAI
- **Destinatário:** : Usuário
- **Tipo:** Chamada síncrona.
- **Descrição:** O sistema exibe uma opção para o usuário revogar o acesso do aplicativo em sua conta do Google.

5. **Mensagem:** `optarPorSim()`

- **Remetente:** : Usuário
- **Destinatário:** Sistema FITAI
- **Tipo:** Chamada síncrona.
- **Descrição:** O usuário decide revogar o acesso, optando por "Sim".

Figura 22: Realiza Sair



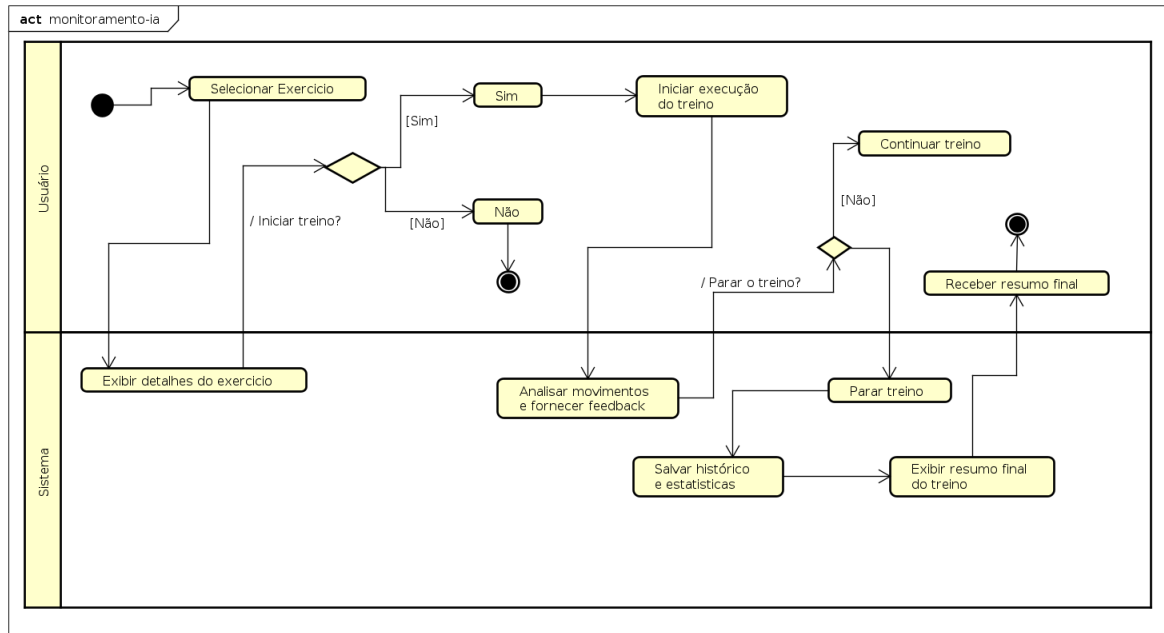
Fonte: Autores, 2025.

## 7 DIAGRAMA DE ATIVIDADE

### 7.1. Diagrama de atividade: Monitoramento por IA.

Este Diagrama ilustra a atividade entre usuário e sistema na funcionalidade de monitoramento de exercícios feito pela IA.

Figura 23: Atividade de Monitoramento de exercício

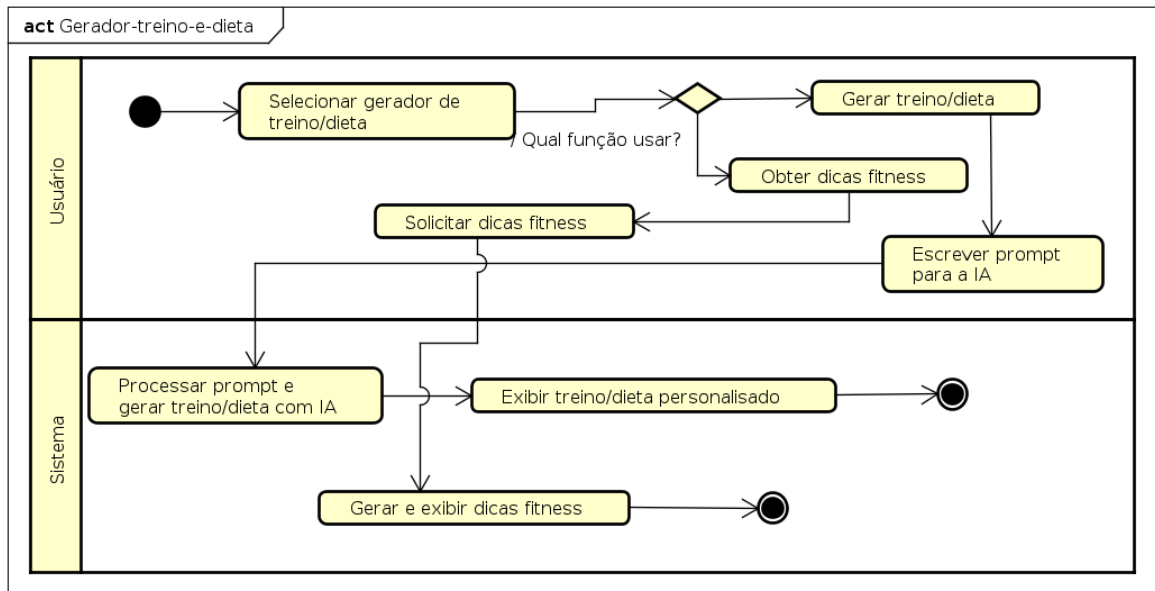


Fonte: Autores, 2025.

### 7.2. Diagrama de atividade: Gerador de treino e dieta.

Este diagrama representa a atividade entre usuário e sistema para a funcionalidade de gerar treinos ou dietas específicas com auxílio de IA.

Figura 24: Atividade de geração de treinos e dietas com IA

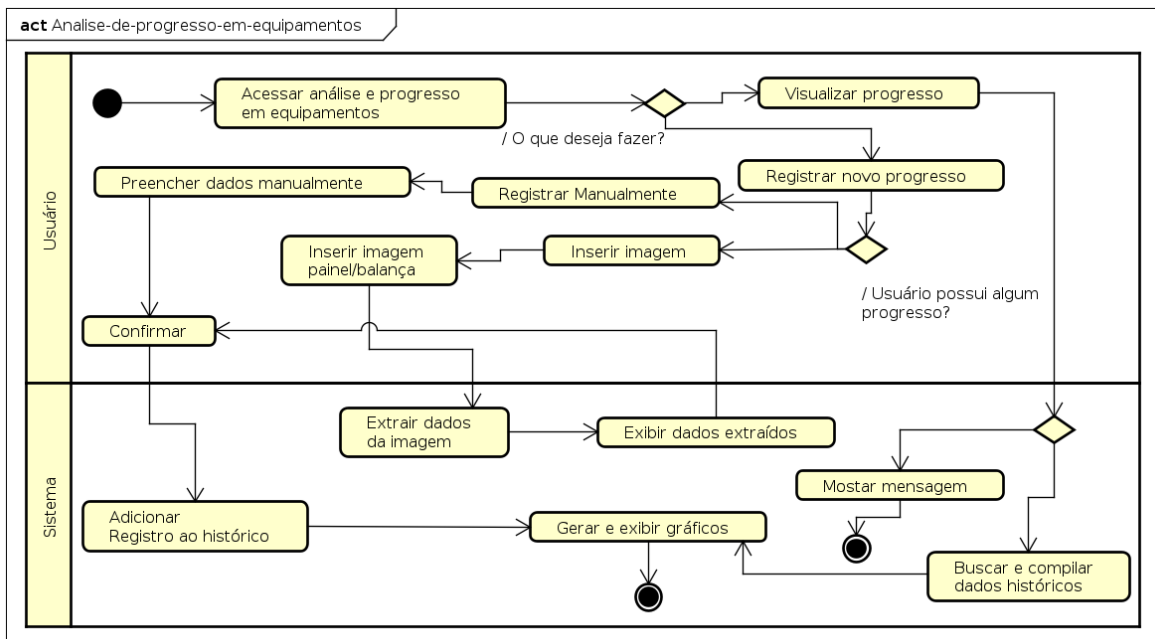


Fonte: Autores, 2025.

### 7.3. Diagrama de atividade: Análise de progresso em equipamentos.

Este diagrama ilustra a atividade entre usuário e sistema para registro, análise e progresso em equipamentos, com geração de gráficos para visualização de progresso.

Figura 25: Atividade de análise e progresso em equipamentos



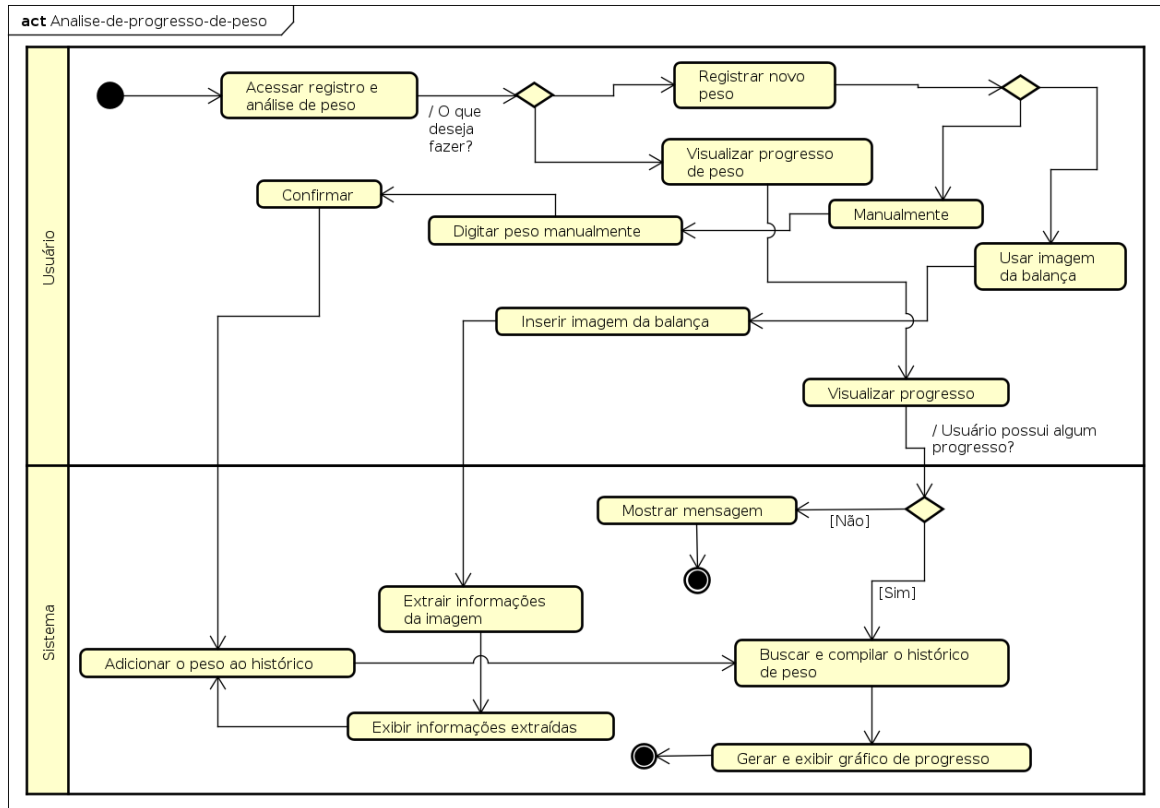
Fonte: Autores, 2025.



#### 7.4. Diagrama de atividade: Análise de progresso de peso.

Este diagrama ilustra a atividade entre usuário e sistema para registro, análise e progresso de peso (kg) do usuário, com geração de gráficos para análise de progresso.

Figura 26: Atividade de análise e progresso de peso (kg)

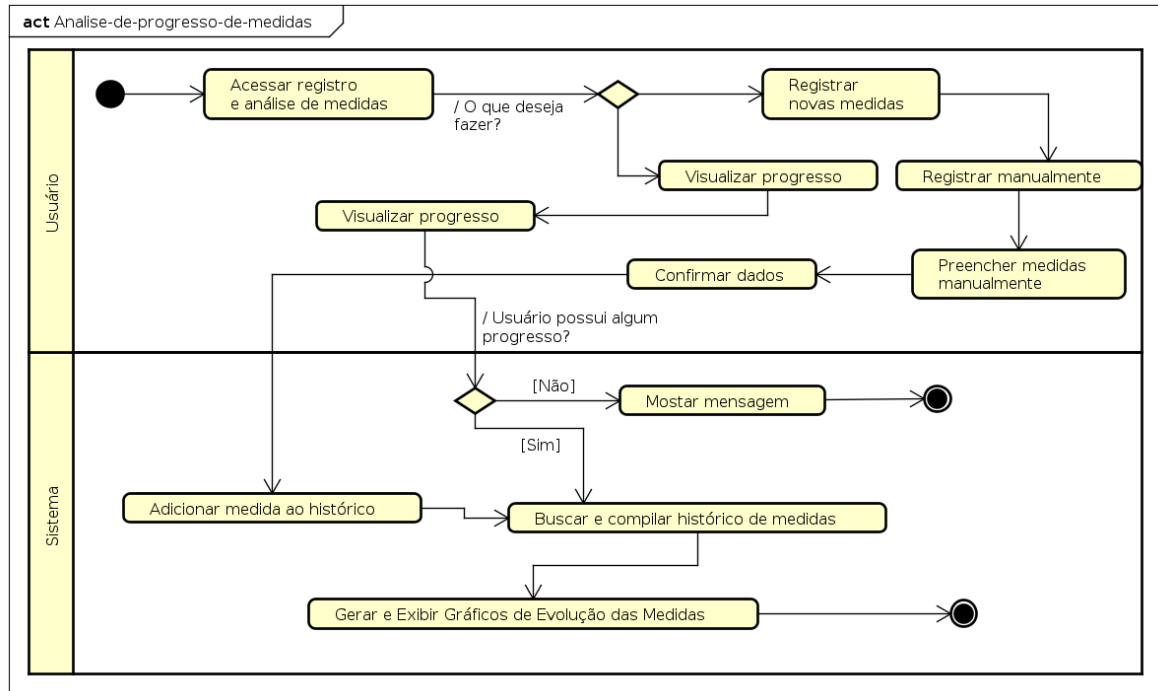


Fonte: Autores, 2025.

#### 7.5. Diagrama de atividade: Análise de progresso de medidas.

Este diagrama ilustra a atividade entre usuário e sistema para registro, análise e progresso de medidas do usuário, com geração de gráficos para análise de progresso.

Figura 27: Atividade de análise e progresso de medidas.



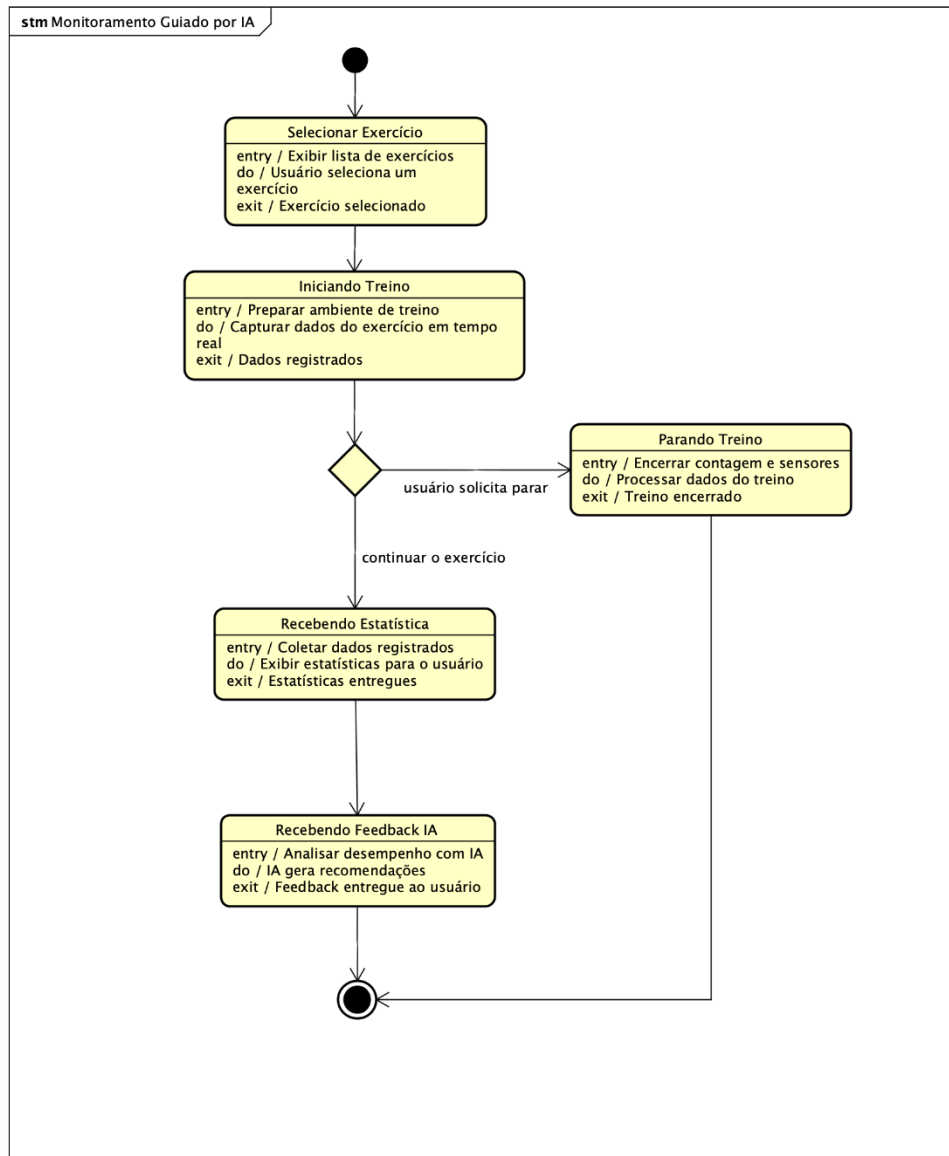
Fonte: Autores, 2025.

## 8 DIAGRAMA DE ESTADO

### 8.1. Diagrama de estado: Monitoramento por IA.

O diagrama de estados representa o fluxo de funcionamento de um sistema de monitoramento de treino físico guiado por IA, desde a seleção do exercício até o fornecimento de estatísticas e feedback ao usuário com base nos dados coletados em tempo real.

Figura 28: Estado de Monitoramento por IA.

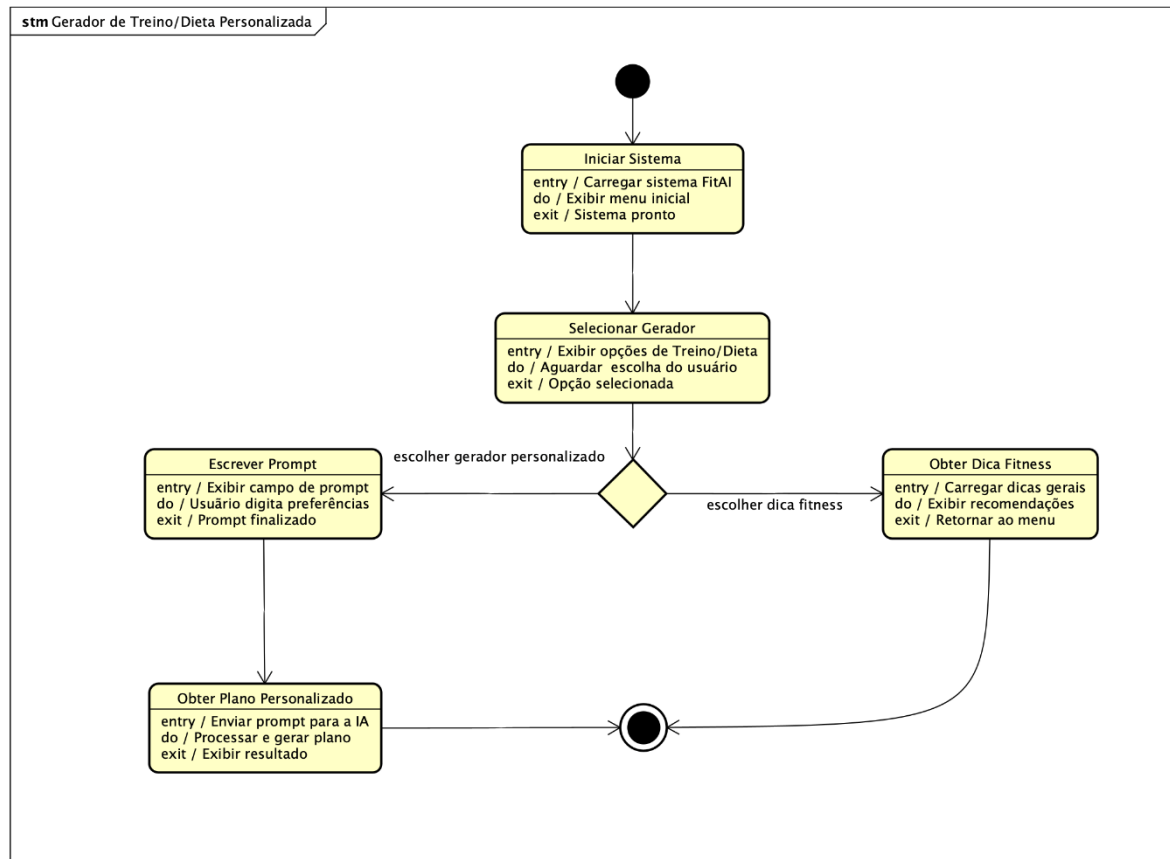


Fonte: Autores, 2025.

### 8.2. Diagrama de estado: Gerador de treino e dieta.

O diagrama de estados descreve o funcionamento de um sistema de geração personalizada de treino ou dieta, no qual o usuário pode escolher entre receber dicas fitness ou criar um plano personalizado por meio de preferências digitadas, com o auxílio de uma IA.

Figura 29: Estado do Gerador de treino e dieta.

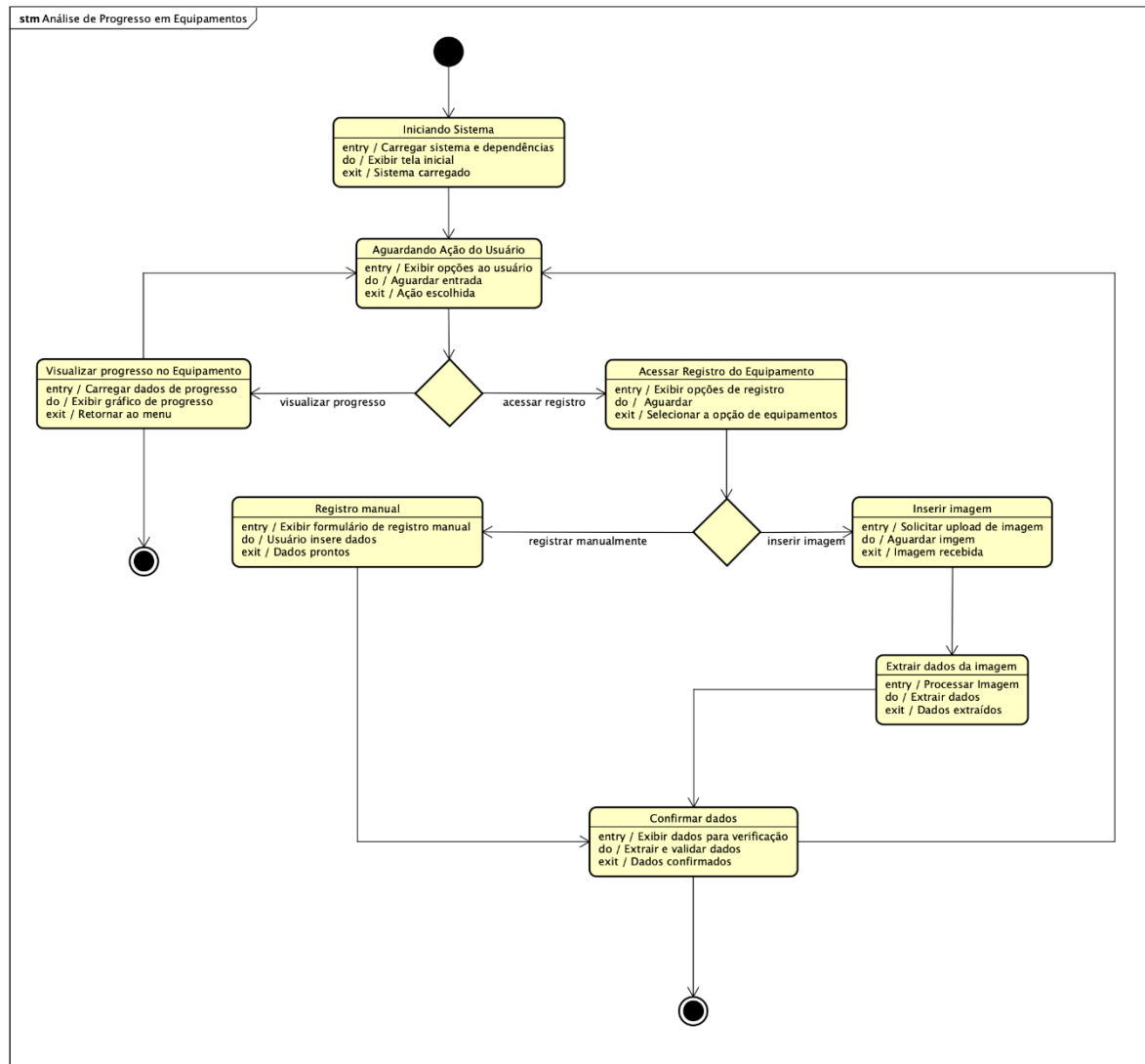


Fonte: Autores, 2025.

### 8.3. Diagrama de estado: Análise de progresso nos equipamentos.

O diagrama de estados representa o funcionamento de um sistema de análise de progresso em equipamentos, permitindo ao usuário visualizar gráficos de evolução e registrar dados de peso manualmente ou por imagem, com validação final antes de concluir o processo.

Figura 30: Estado da Análise de progresso nos equipamentos.

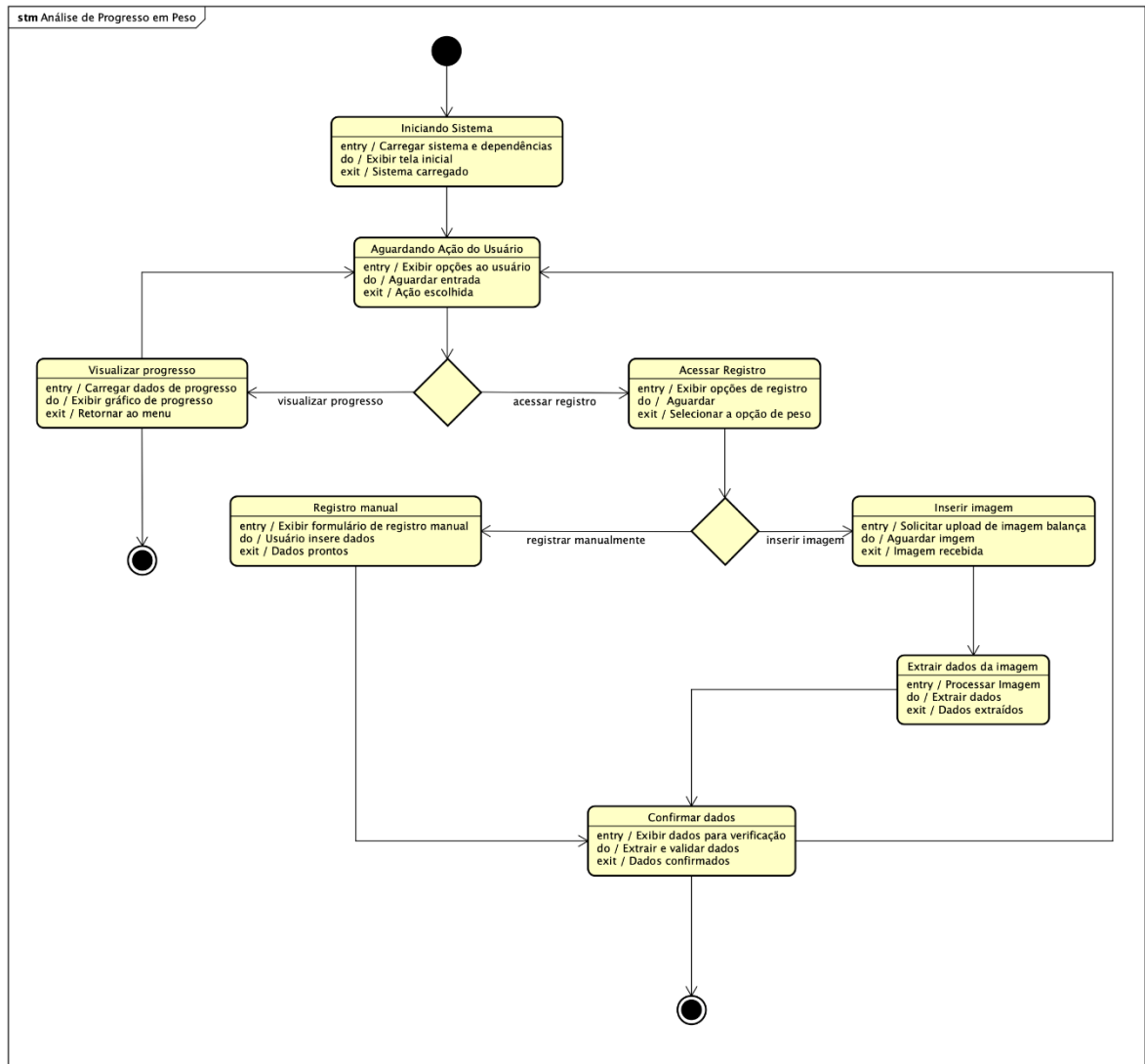


Fonte: Autores, 2025.

#### 8.4. Diagrama de estado: Análise de progresso de peso.

O diagrama de estados descreve o fluxo de um sistema de análise de progresso em peso, permitindo ao usuário visualizar gráficos e registrar medidas manualmente ou por imagem de balança, com extração e confirmação final dos dados coletados.

Figura 31: Estado da Análise de progresso de peso.

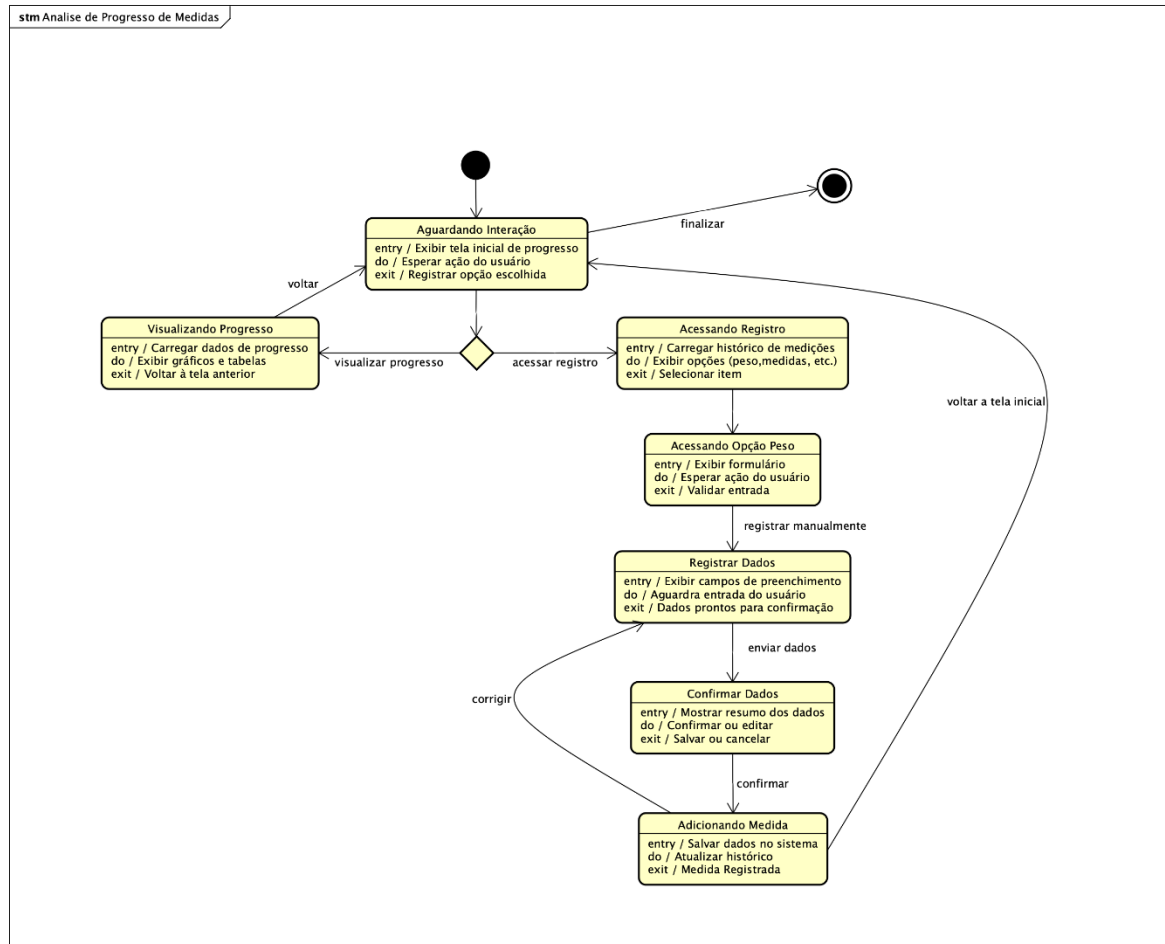


Fonte: Autores, 2025.

### 8.5. Diagrama de estado: Análise de progresso de medidas.

O diagrama de estados representa o funcionamento de um sistema de análise de progresso de medidas, onde o usuário pode visualizar gráficos e registrar manualmente novas medições, com validação e confirmação antes de salvar os dados no sistema.

Figura 32: Estado da Análise de progresso de medidas.



Fonte: Autores, 2025.

## REFERÊNCIAS

ABC EVO. **Tendências do Mercado Fitness para 2025: Como se preparar para um ano de crescimento.** ABC EVO, 2025. Disponível em:

<https://blog.abcevo.com.br/tend%C3%AAsncias-do-mercado-fitness-para-2025-como-se-preparar-para-um-ano-de-crescimento>.

MKT ESPORTIVO. **Top 7 tendências de wellness e fitness para ficar de olho em 2025.**

MKT Esportivo, 2025. Disponível em: <https://www.mktesportivo.com/2025/04/top-7-tendencias-de-wellness-e-fitness-para-ficar-de-olho-em-2025/>

MARCOS. **Tendências no Setor de Fitness para 2025.** Arcielo, 2024. Disponível em:

<https://arcielo.com.br/tendencias-no-setor-de-fitness-para-2025/>.