FACULDADE DE TECNOLOGIA DE PRESIDENTE PRUDENTE
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DANIEL AMERICO DA SILVA TRENTIN
IGOR ARIMATHEIA CONTRI

GO HEALTH

Presidente Prudente – SP

2026

Fatec
Presidente
Prudente

FACULDADE DE TECNOLOGIA DE PRESIDENTE PRUDENTE
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DANIEL AMERICO DA SILVA TRENTIN
IGOR ARIMATHEIA CONTRI

GO HEALTH

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Presidente Prudente, como requisito parcial para obtenção do diploma de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador(a): Prof. Álvaro Ferraz D'Arce

Presidente Prudente – SP

2026

DANIEL AMERICO DA SILVA TRENTIN
IGOR ARIMATHEIA CONTRI

GO HEALTH

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Presidente Prudente, como requisito parcial para obtenção do diploma de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em: 01 de junho de 2026.

BANCA EXAMINADORA

_____
Orientador: Me. Prof. Álvaro Ferraz D'Arce
FATEC – Presidente Prudente
Presidente Prudente

_____
Profa. Dra. Giovana Angelica Ros
FATEC – Presidente Prudente
Presidente Prudente

_____
Prof. Me. Rodrigo Vilela da Rocha
FATEC – Presidente Prudente
Presidente Prudente

RESUMO

CONTRI, Igor; AMERICO, Daniel. **GoHealth: Especificação e desenvolvimento de um aplicativo de gamificação e accountability para a saúde física**. Orientador: Álvaro Ferraz D'Arce. 2026. xx f. Trabalho de Conclusão de Curso (Análise e Desenvolvimento de Sistemas) - Faculdade de Tecnologia de Presidente Prudente, Presidente Prudente, SP, 2026.

O presente trabalho apresentou a Especificação de Requisitos de Software (ERS) e o desenvolvimento do aplicativo móvel Go Health, uma plataforma digital focada no incentivo à prática regular de atividades físicas por meio de técnicas de gamificação e validação social interativa (accountability). O problema central da pesquisa abordou a frequente dificuldade que os indivíduos enfrentaram para manter a consistência de longo prazo em suas rotinas de exercícios físicos e hábitos saudáveis. Para mitigar esta taxa de abandono, o projeto concebeu e implementou uma solução arquitetural baseada no padrão Model-View-Controller (MVC) com uma robusta camada de Serviços, utilizando o framework Laravel no back-end e React Native no front-end para garantir uma experiência multiplataforma fluida. A metodologia de engenharia de requisitos detalhou fluxos de interação complexos, englobando a modelagem de casos de uso essenciais como o registro de check-ins diários, o cálculo inteligente de sequências individuais (streaks) que respeitou jornadas de descanso personalizadas, e a gestão de grupos de apoio. O sistema processou a validação de regras específicas de cada comunidade de forma estritamente transacional, o que assegurou a integridade do banco de dados relacional e impediu a contabilização de atividades fora dos padrões exigidos pelos administradores dos grupos. Além disso, a aplicação automatizou a consolidação de rankings periódicos e o envio de notificações, estimulando o engajamento contínuo. A validação arquitetural confirmou que a segregação das lógicas de validação e gamificação em serviços isolados promoveu um baixo acoplamento e alta coesão no código-fonte. O trabalho concluiu que a união estruturada de mecanismos psicológicos de recompensa contínua com a pressão social positiva de comunidades privadas resultou em uma ferramenta tecnológica eficaz, escalável e capaz de apoiar ativamente os usuários na manutenção de uma vida fisicamente ativa.

Palavras-chave: Gamificação; Accountability; Engenharia de Software; Hábitos Saudáveis; Desenvolvimento Mobile.

ABSTRACT

CONTRI, Igor; AMERICO, Daniel. **Go Health: Specification and development of a gamification and accountability application for physical health**. Advisor: Álvaro Ferraz D'Arce. 2026. xx f. Course Conclusion Work (Systems Analysis and Development) - University of Technology de Presidente Prudente, Presidente Prudente, SP, 2026.

This work presented the Software Requirements Specification (SRS) and the development of the Go Health mobile application; a digital platform focused on encouraging the regular practice of physical activities through gamification techniques and interactive social validation (accountability). The core research problem addressed the frequent difficulty individuals faced in maintaining long-term consistency in their physical exercise routines and healthy habits. To mitigate this dropout rate, the project designed and implemented an architectural solution based on the Model-View-Controller (MVC) pattern with a robust Service layer, utilizing the Laravel framework on the back-end and React Native on the front-end to ensure a fluid cross-platform experience. The requirements engineering methodology detailed complex interaction flows, encompassing the modeling of essential use cases such as recording daily check-ins, the intelligent calculation of individual streaks that respected personalized rest journeys, and the management of support groups. The system processed the validation of specific rules for each community in a strictly transactional manner, which ensured the integrity of the relational database and prevented the accounting of activities outside the standards required by group administrators. Furthermore, the application automated the consolidation of periodic rankings and the delivery of notifications, stimulating continuous engagement. The architectural validation confirmed that segregating validation and gamification logic into isolated services promoted low coupling and high cohesion within the source code. The work concluded that the structured combination of psychological mechanisms for continuous reward with the positive social pressure of private communities resulted in an effective and scalable technological tool capable of actively supporting users in maintaining a physically active life.

Keywords: Gamification; Accountability; Software Engineering; Healthy Habits; Mobile Development.

SUMÁRIO

1. INTRODUÇÃO

1.1 OBJETIVO

Este documento consiste em uma *ERS* (Especificação de Requisitos de Software) baseada na norma *IEEE* 830/1998 (Institute of Electrical and Eletronics Engineers) e tem como objetivo especificar os requisitos do software em desenvolvimento, inteirando o cliente e os desenvolvedores sobre o desenvolvimento e a utilização do software.

1.2 ESCOPO

O sistema permitirá o gerenciamento completo do ciclo de vida dos usuários, incluindo a capacidade de criar, autenticar e administrar contas individuais, bem como gerenciar perfis com informações pertinentes. A plataforma proverá mecanismos seguros para acesso e recuperação de credenciais, garantindo a integridade da identidade do usuário. Além da gestão individual, o sistema oferecerá funcionalidades para a criação e administração de grupos, permitindo que os usuários controlem a composição e a participação de seus membros.

A interação principal ocorrerá por meio do registro de check-ins, que poderão ser acompanhados de mídias associadas para validação e contextualização. O sistema será responsável por gerenciar e armazenar permanentemente esses registros. Cada check-in submetido passará por um processo de validação para garantir sua conformidade com regras predefinidas. Com base nos registros validados, o sistema calculará continuamente as sequências de atividades individuais e determinará as sequências e pontuações consolidadas para os grupos.

Para fomentar o engajamento e a competitividade, o sistema estabelecerá classificações dinâmicas, comparando o progresso entre usuários e entre grupos com base nas métricas de desempenho calculadas. Para garantir a consistência da participação, o sistema também emitirá lembretes periódicos aos usuários.

Adicionalmente, a plataforma será capaz de gerar relatórios detalhados sobre

o progresso individual e o desempenho de grupos, além de fornecer resumos periódicos que consolidam as informações de atividade em intervalos de tempo específicos.

1.3 DEFINIÇÕES, SIGLAS E ABREVIAÇÕES

- F_B: Funções básicas.
- F_F: Funções Fundamentais.
- F_S: Funções de Saida.
- CRUD: Create, Retrieve, Update, Delete. Representa as operações básicas de um cadastro (Incluir, Pesquisar, Atualizar, Excluir).

1.4 REFERÊNCIAS

Não há.

1.5 VISÃO GERAL

Esta *ERS* está organizada em capítulos. O *Capítulo 2* fornece uma descrição geral do software em desenvolvimento, contendo uma perspectiva do produto, suas funções, perfil dos usuários do software, características do desenvolvimento e requisitos adiados. O *Capítulo 3* detalha os requisitos do software. O *Capítulo 4* fornece detalhes do projeto do software.

2. DESCRIÇÃO GERAL DO PRODUTO

2.1 ESTUDO DE VIABILIDADE

O sistema proposto apresenta viabilidade técnica, econômica e operacional a partir da análise de seus requisitos e objetivos estratégicos. Do ponto de vista técnico, a solução depende de recursos amplamente disponíveis, como mecanismos de autenticação de usuários, persistência de dados, validação de registros e cálculo de métricas de desempenho. As funcionalidades descritas, incluindo o gerenciamento de contas, perfis e grupos, bem como o controle de check-ins e mídias, podem ser implementadas de forma consistente com tecnologias de mercado, considerando práticas consolidadas de segurança e escalabilidade. A escolha do PHP como tecnologia central tanto para o back-end quanto para o front-end reforça a viabilidade, uma vez que se trata de uma linguagem amplamente utilizada, com vasto ecossistema de bibliotecas e frameworks, além de oferecer suporte consolidado para aplicações dinâmicas. A complexidade principal reside na integração entre os módulos de validação, cálculo de sequências e pontuação, mas não há impeditivos técnicos relevantes para sua execução em um ambiente moderno de desenvolvimento e operação.

2.2 FUNÇÕES DO SISTEMA

O Quadro 2 apresenta as funções básicas do sistema, ou seja, as operações CRUD.

Quadro 2 – Funções Básicas.

Identificação	Descrição
F_B01 Gerenciar Conta de Usuário	Permite criar, editar, excluir um usuário; redefinir senha; redefinir e-mail; verificar e-mail, desativação de conta.
F_B02 Gerenciar Grupos	Permite criar, adicionar, editar, excluir um grupo.
F_B03 Gerenciar Perfil	Permite o usuário manter um perfil com informações complementares, como nome, e-mail, biografia pessoal, imagem de exibição e preferencias de notificação.
F_B04 Gerenciar membros do	Permite o gerenciamento dos membros do grupo, incluindo envio de convites, entrada voluntaria, saída

grupo	voluntaria e remoção forçada.
F_B05 Gerenciar Check-ins	Permite o usuário fazer check-ins diário para comprovar a prática de atividade física, permitindo criar, adicionar, alterar, excluir o check-ins.
F_B06 Gerenciar Mídias	Permite o envio de mídia vinculadas a check-ins, como imagens ou vídeos.
F_B07 Logar no Sistema	Permite o acesso ao sistema a partir das credenciais do usuário login e senha.
F_B08 Recuperar Senha	Permite que o usuário, a partir de um e-mail cadastrado, recupere sua senha.

Fonte: Elaborado pelo próprio autor.

O Quadro 3 apresenta as funções fundamentais do sistema, ou seja, as implementações das regras de negócio.

Quadro 3 – Funções Fundamentais.

Identificação	**Descrição**
F_F01: Registrar Check-in	O sistema aceita 1 check-in válido por dia/usuário e valida o formato (dados, foto, vídeo)
F_F02: Calcular sequência individual	Incrementa se houve check-in no “dia” corrente e zera a sequência se faltar 2 dias seguidos além dos especificados como “dias de descanso”.
F_F03: Calcular Check-in em Grupos	Incrementa conforme as regras de restrições de check-in do grupo e contabiliza independente das pontuações pessoais e de outros grupos.
F_F04: Estruturar Dinâmica de Grupo	Garantir que a comunidade tenha regras claras de cobrança para os check-ins, estabelecendo o formato exigido (texto, mídia ou ambos) para comprovação dos treinos e configurando os dados de apresentação do grupo.
F_F05: Planejar Jornada de Descanso	O sistema permite configurar os dias da semana em que o usuário não realizará treinos (descanso), sem que resete a sequência
F_F06: Registrar Check-in via Smartwatch	O sistema utilizaria metadados provenientes dos sensores do dispositivo (frequência cardíaca, acelerômetro e GPS) para confirmar que o usuário está em atividade física no momento do check-in.

Fonte: Elaborado pelo próprio autor.

O Quadro 4 apresenta as funções de saída do sistema, ou seja, relatórios, gráficos e listagens.

Quadro 4 – Funções de Saída.

Identificação	**Descrição**
F_S01 Relatório de progresso individual	Emite um relatório com histórico de melhor sequência, dias ativos na semana/mês e desempenho semanal/mensal em relação ao mesmo período anterior.
F_S02 Relatório de grupo	Emite um relatório de desempenho em relação aos competidores do grupo, ranking atual e histórico.
F_S03 Resumos periódicos	Notificações no e-mail para metrificar e compartilhar a progressão.

Fonte: Elaborado pelo próprio autor.

2.3 CARACTERÍSTICAS DO USUÁRIO

Os futuros usuários do sistema apresentam perfis variados, com predominância de nível educacional médio e superior, refletindo um público capaz de compreender instruções básicas e interpretar relatórios de desempenho. A experiência prévia com informática é heterogênea, abrangendo desde usuários com domínio avançado de recursos digitais até aqueles com conhecimento limitado, restrito ao uso de aplicativos comuns, como navegadores e ferramentas de comunicação.

Esse cenário indica que a maioria dos usuários terá condições de interagir com o sistema sem dificuldades significativas, desde que a solução adote fluxos intuitivos e linguagem clara.

No entanto, para grupos com menor familiaridade tecnológica, pode ser necessário oferecer treinamento pontual e materiais de apoio, assegurando a plena utilização das funcionalidades e evitando barreiras de adoção. Dessa forma, o sistema deverá conciliar robustez técnica com simplicidade de uso, garantindo acessibilidade e engajamento de diferentes perfis de usuário.

2.4 LIMITES, DEPENDÊNCIAS E SUPOSIÇÕES

O desenvolvimento e a implantação do sistema dependem de infraestrutura compatível com PHP, servidores web atualizados e banco de dados relacional. A responsabilidade por backups será do cliente, devendo ser executados em intervalos regulares para garantir integridade e recuperação de dados. A utilização do sistema

pressupõe que os usuários tenham acesso a navegadores atualizados e conexão estável à internet, condição essencial para registro de check-ins e geração de relatórios. As notificações exigem permissões adequadas do usuário e correta configuração do ambiente. Também se pressupõe conformidade com normas de segurança e privacidade vigentes, sem as quais a implantação pode ser comprometida. Por fim, considera-se que o cliente disponibilizará suporte básico aos usuários, garantindo adoção plena das funcionalidades.

2.5 REQUISITOS ADIADOS

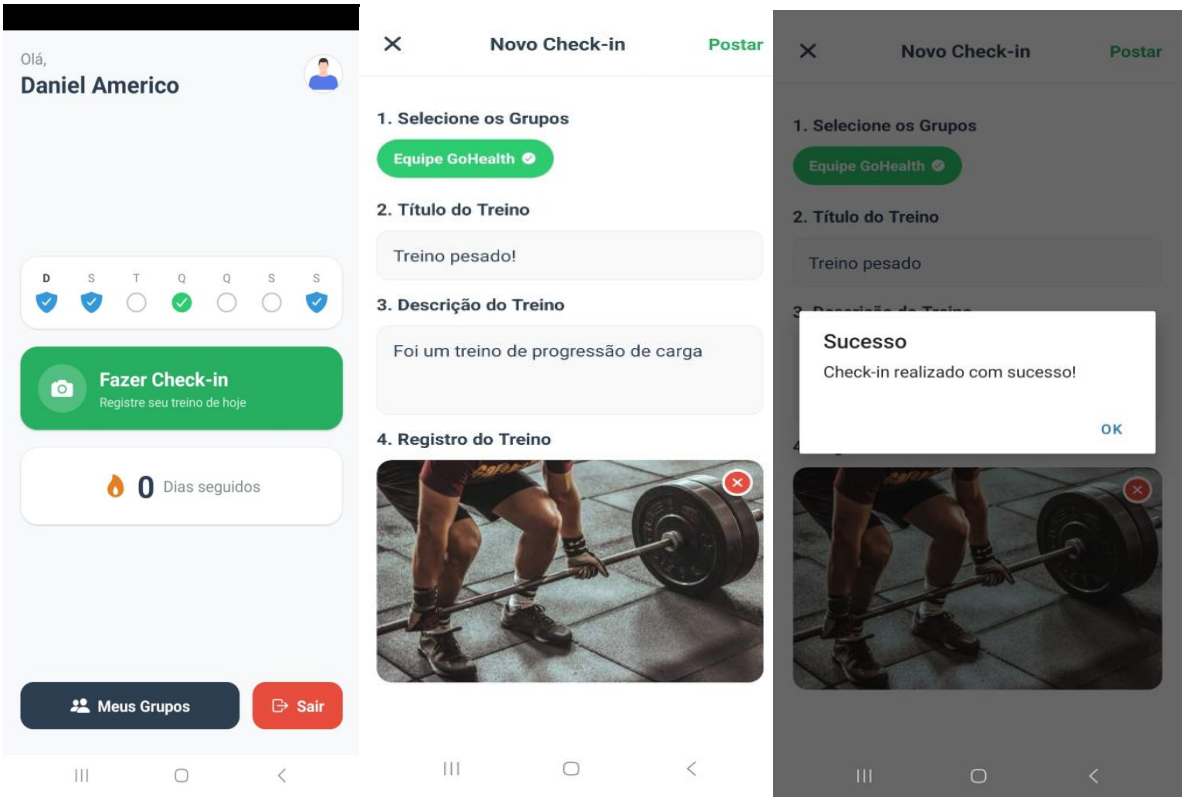
F_F06: Registrar Check-in via Smartwatch	O sistema utilizaria metadados provenientes dos sensores do dispositivo (frequência cardíaca, acelerômetro e GPS) para confirmar que o usuário está em atividade física no momento do check-in.
F_S01: Relatório de progresso individual	Emite um relatório com histórico de melhor sequência, dias ativos na semana/mês e desempenho semanal/mensal em relação ao mesmo período anterior.
F_S02 Relatório de grupo	Emite um relatório de desempenho em relação aos competidores do grupo, ranking atual e histórico.

3. REQUISITOS ESPECÍFICOS

3.1 REQUISITOS DE INTERFACE EXTERNA

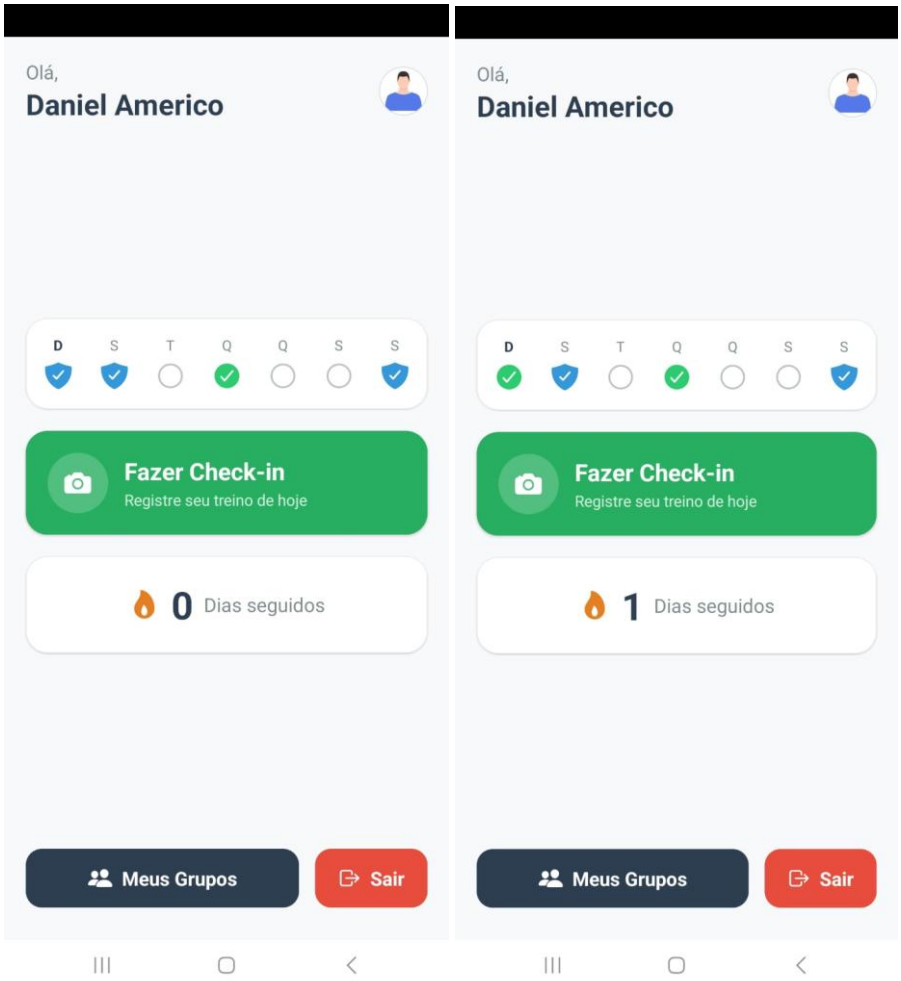
3.1.1 INTERFACES DO USÁRIOS DOS CASOS DE USO

Figura 1 – Caso de Uso Realizar Check-in



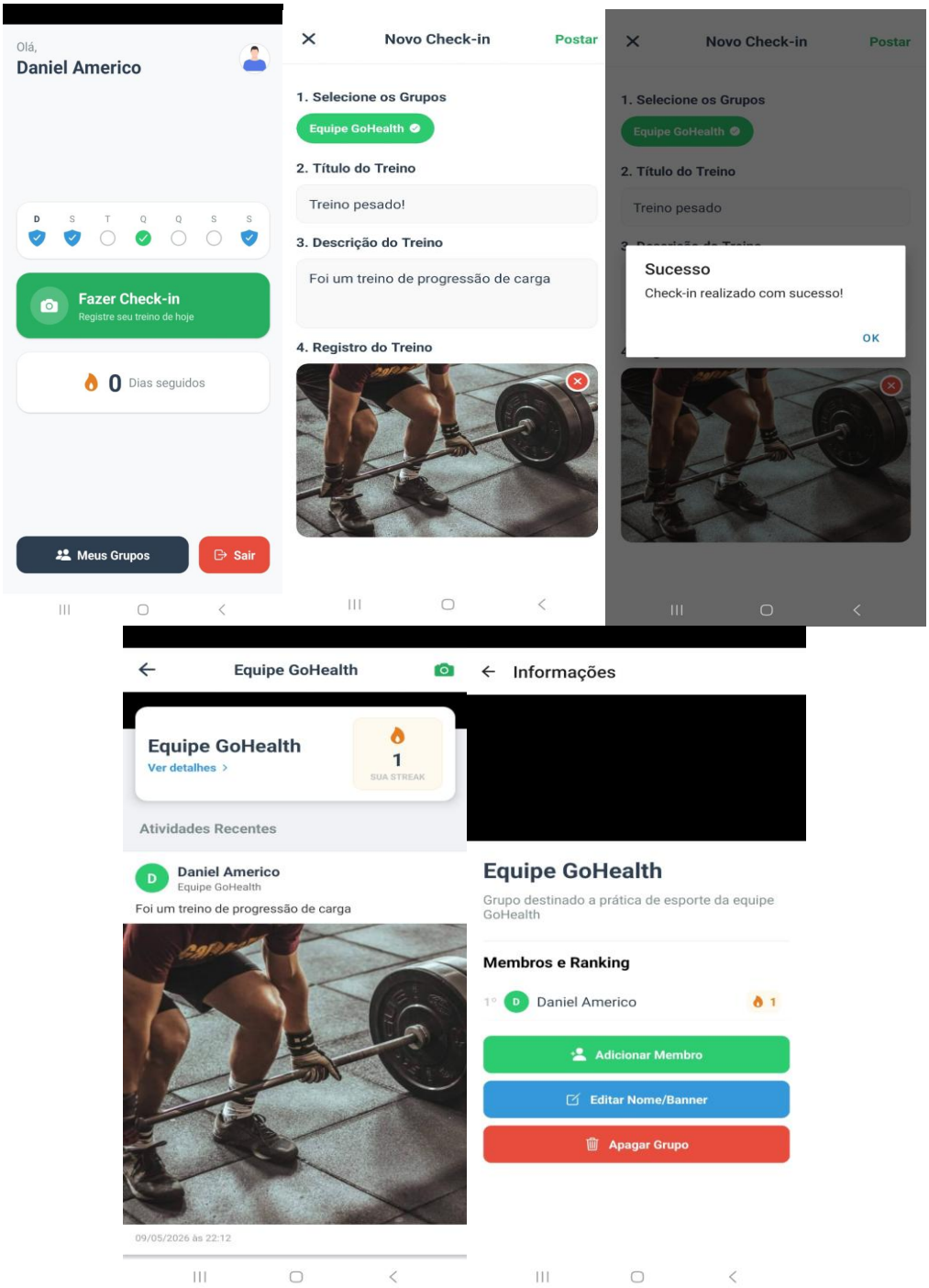
Fonte: Elaborado pelo próprio autor.

Figura 2 – Caso de Uso Calcular Sequência Individual



Fonte: Elaborado pelo próprio autor.

Figura 3 – Caso de Uso Calcular Check-in em Grupos

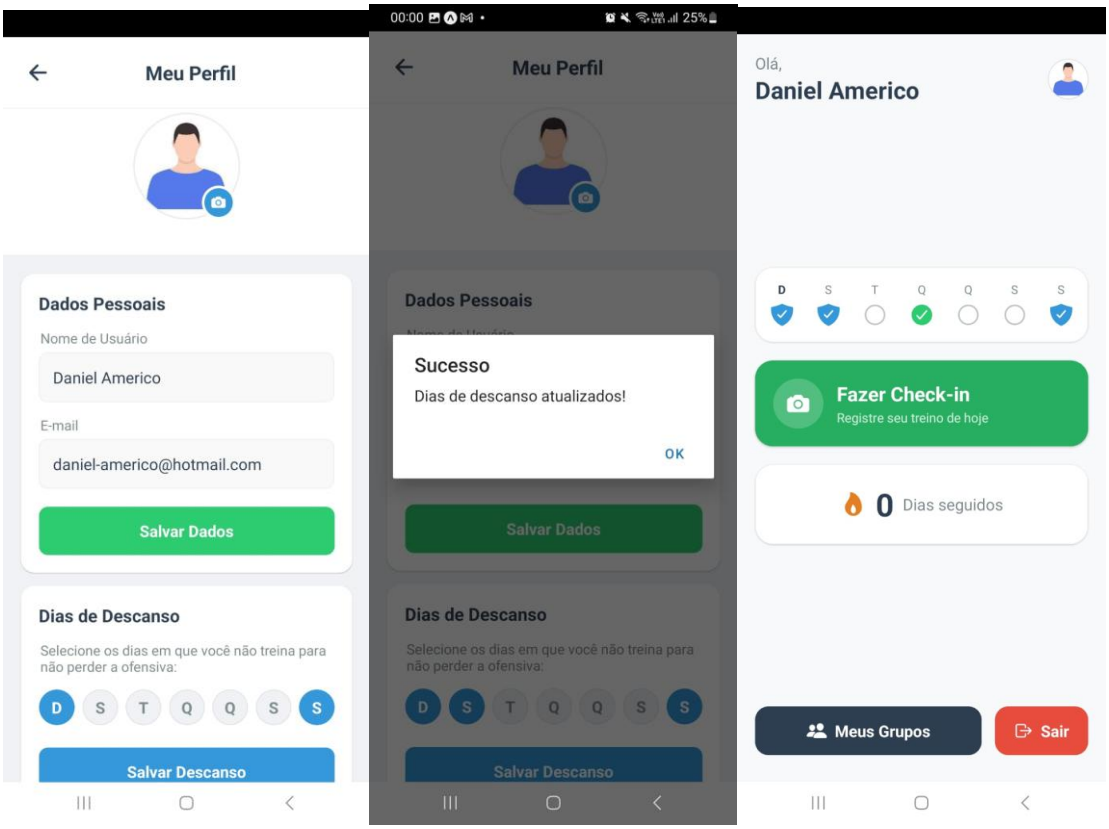


Fonte: Elaborado pelo próprio autor.

Figura 4 – Caso de Uso Estruturar Dinâmica de Grupo

Fonte: Elaborado pelo próprio autor.

Figura 5 – Caso de Uso Planejar Jornada de Descanso



Fonte: Elaborado pelo próprio autor.

3.1.4 INTERFACES DE SOFTWARE

Sistemas Operacionais (Client-side): O aplicativo móvel requer os sistemas operacionais Android (versão 8.0 ou superior) ou iOS (versão 13.0 ou superior) para execução, utilizando o ambiente do Expo e os motores nativos de renderização do dispositivo.

Sistema Operacional (Server-side): O servidor de aplicação backend requer um ambiente baseado em sistema operacional Linux ou Windows com interpretador PHP instalado para a hospedagem e execução dos serviços.

Backend: O servidor utiliza o framework Laravel, que atua como o núcleo lógico para processamento de regras de negócio, autenticação e rotas.

Frontend Mobile: A interface do usuário é construída e gerada através do framework React Native.

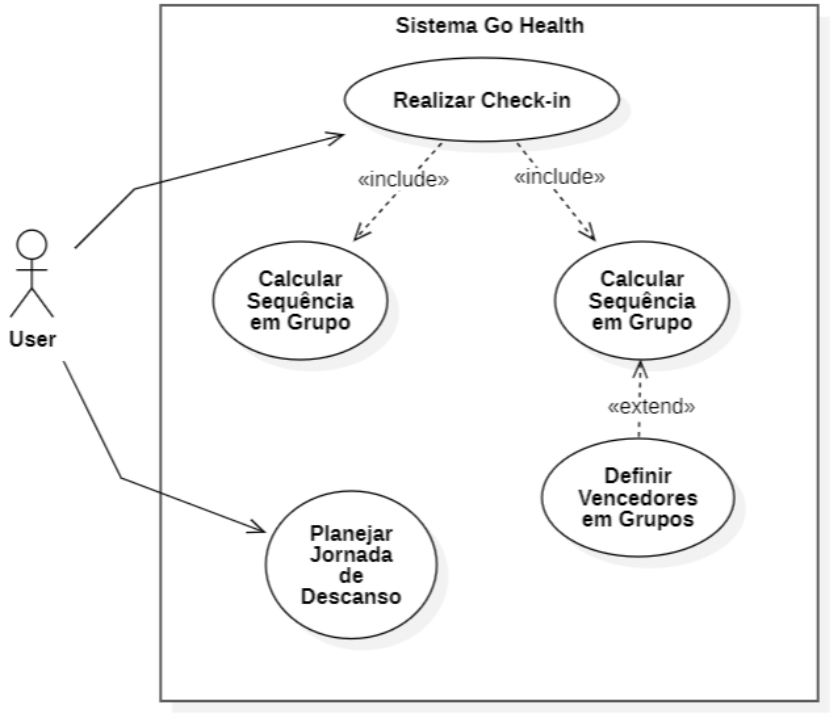
Sistema Gerenciador de Banco de Dados (SGBD): O sistema utiliza um banco de dados relacional (como MySQL ou PostgreSQL) para a persistência das entidades (usuários, grupos, check-ins). A comunicação entre a aplicação e o banco de dados é intermediada de forma segura pelo ORM (Object-Relational Mapping) Eloquent, nativo do Laravel.

Comunicação de Rede (API REST): A interface de comunicação entre o aplicativo móvel (Frontend) e o servidor (Backend) ocorre por meio de uma API RESTful. A troca de informações é realizada sobre o protocolo HTTP/HTTPS, utilizando o padrão de formatação de dados JSON.

Câmera e Galeria: O aplicativo interage com as APIs de software nativas da câmera e do gerenciador de arquivos do dispositivo móvel, através da biblioteca expo-image-picker, para captura e upload de fotos nos check-ins e alteração de avatar.

Armazenamento Local: A aplicação se comunica com o sistema de armazenamento assíncrono nativo do dispositivo móvel (via AsyncStorage) para a persistência local de tokens de sessão (JWT/Sanctum) e controles de estado (como a visualização de modais de boas-vindas).

3.2 DIAGRAMA DE CASOS DE USO



Fonte: Elaborado pelo próprio autor.

3.3 ESPECIFICAÇÕES DOS CASOS DE USO

Caso de Uso: R_F01 – Realizar Check-in

Finalidade: Registrar o check-in diário de um usuário.

Atores: Usuário

Visão Geral: Garantir que o check-in realizado por um usuário seja consistente, autêntico e registrado corretamente, além de atualizar automaticamente as sequências e pontuações associadas.

Ações do Ator	Resposta do Sistema
1- O usuário solicita o registro de um check-in.	
	2- O sistema verifica se o os

	dados de check-in são válidos
	3- O sistema verifica que o check-in do dia atual ainda não foi realizado.
	4- O sistema registra o check-in como válido e o associa ao usuário.
	5- O sistema inclui o caso de uso “Calcular Sequência Individual” para atualizar a sequência do usuário
	6- O sistema inclui o caso de uso “Calcular Sequência e Pontuação em Grupos” para atualizar pontuações dos grupos em que o usuário participa.
	7- O sistema informa ao usuário que o check-in foi aceito

Sequencias alternativas

- Linha 2: Se os dados não atenderem aos critérios, o sistema rejeita o check-in e informa o motivo

- Linha 3: Caso o check-in já tenha sido realizado, o sistema alerta o usuário, não registra o check-in e finaliza o processo.

Caso de Uso: R_F02 Calcular Sequência Individual

Finalidade: Calcular a sequência Individual do Usuário

Atores: Usuário

Visão Geral: Determinar a sequência contínua de check-ins de um usuário, considerando regras de frequência e descanso configuradas.

Ações do Ator	Resposta do Sistema
1- O usuário registra um check-in válido no sistema	
	2- O sistema identifica um novo check-in validado.
	3- O sistema consulta o histórico de check-ins do usuário.
	4- O sistema avalia se o check-in mantém a continuidade da sequência com base nos dias entre o último check-in e o atual.
	5- O sistema incrementa a sequência atual.
	6- O sistema atualiza a nova sequência individual no perfil do usuário.
	7- O sistema disponibiliza a informação para relatórios e feedback ao usuário.

Sequências alternativas

- Linha 4: Se o intervalo entre check-ins for maior que 2 dias além dos definidos pelo usuário como “dias de descanso”, a sequência é reiniciada.

Caso de Uso: R_F03 Calcular Check-in em Grupos

Finalidade: Calcular check-in em grupo.
Atores: Usuário
Visão Geral: Avaliar a sequência coletiva de check-ins e gerar pontuações para grupos de usuários.

Ações do Ator	Resposta do Sistema
1- O usuário registra um check-in válido no sistema.	
	2- O Sistema identifica um novo check-in validado.
	3- O sistema consulta o histórico de check-ins dos respectivos grupos dos quais o usuário é membro.
	4- O sistema verifica se o check-in atende as restrições predefinidas pelo administrador do grupo (metadados, tipo de mídia).
	5- O sistema aplica regras de pontuação baseadas em sequência e quantidade total de check-ins.
	6- O sistema incrementa o total de check-ins do usuário no grupo.
	7- O sistema disponibiliza os resultados para relatórios e

	rankings.

Sequências alternativas

- Linha 4: Se o check-in não for válido para as definições do grupo, o usuário é informado que precisa de um novo check-in, válido para o grupo.

Caso de Uso: R_F04 – Estruturar Dinâmica de Grupo

Finalidade: Permitir que um usuário crie ou edite um grupo, definindo suas regras de validação e identidade visual.
Atores: Dono do Grupo (Usuário)
Visão Geral: Garantir que a comunidade tenha regras claras de cobrança para os check-ins, estabelecendo o formato exigido (texto, mídia ou ambos) para comprovação dos treinos e configurando os dados de apresentação do grupo.

Ações do Ator	Resposta do Sistema
1- O usuário solicita a criação (ou edição) de um grupo, informando os detalhes e a regra de validação.	
	2- O sistema verifica se os dados e arquivos anexados (banner) são válidos e cumprem os requisitos de tamanho/formato
	3- O sistema registra o grupo e consolida a regra de validação escolhida para os futuros check-

	ins.
	4- O sistema associa o usuário criador como dono e como o primeiro membro da comunidade (em caso de criação).
	5- O sistema informa ao usuário que o grupo foi estruturado e salvo com sucesso.

Sequências alternativas

- Linha 2: Se os dados não atenderem aos critérios, o sistema rejeita a estruturação e informa o erro de validação ao usuário.

Caso de Uso: R_F05 – Planejar Jornada de Descanso

Finalidade: Configurar os dias da semana em que o usuário não realizará treinos.
Atores: Usuário
Visão Geral: Personalizar as regras de gamificação do aplicativo de acordo com a rotina real do usuário, garantindo que o sistema de ofensivas (fogo/"streak") não seja zerado injustamente nos dias de repouso programados.

Ações do Ator	Resposta do Sistema
1- O usuário seleciona os dias da semana destinados ao repouso e solicita a gravação da rotina.	
	2- O sistema verifica se o formato dos dias selecionados é válido.
	3- O Sistema verifica se a última alteração de dia de

	descanso é maior do que sete dias.
	4- O sistema atualiza o perfil do usuário, registrando os novos dias como exceções para o cálculo de ofensiva.
	5- O sistema informa ao usuário que a jornada de descanso foi atualizada com sucesso.

Sequência alternativa

- Linha 2: Se houver falha na comunicação ou os dados enviados estiverem corrompidos, o sistema rejeita a alteração, mantém a configuração anterior e alerta o usuário sobre o erro.

- Linha 3: Se o usuário já tiver alterado os dias de descanso em um intervalo menor que sete dias, o sistema devolve erro e não altera os dias de descanso.

3.4 DIAGRAMAS DE ATIVIDADES DOS CASOS DE USO

Diagrama de Atividade: RF_F01 - Realizar Check-in

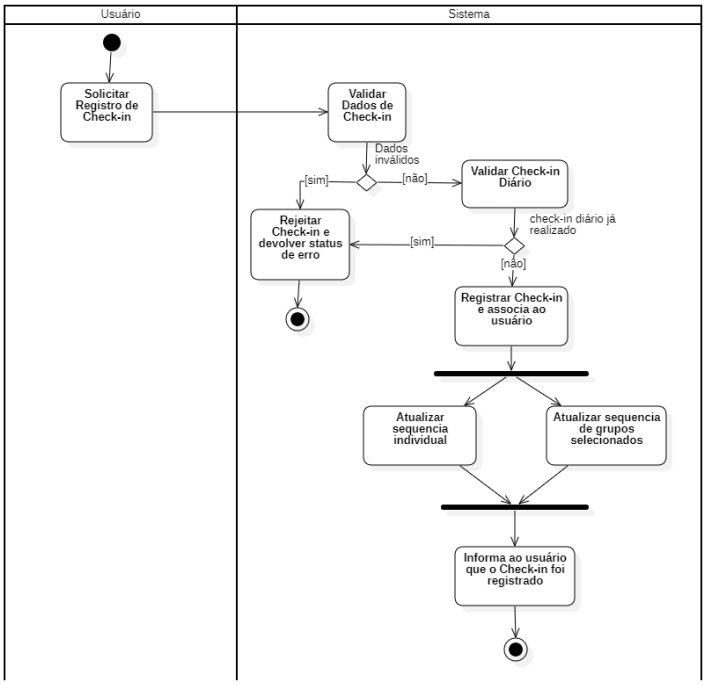


Diagrama de Atividade: RF_F02 - Calcular Sequência Individual

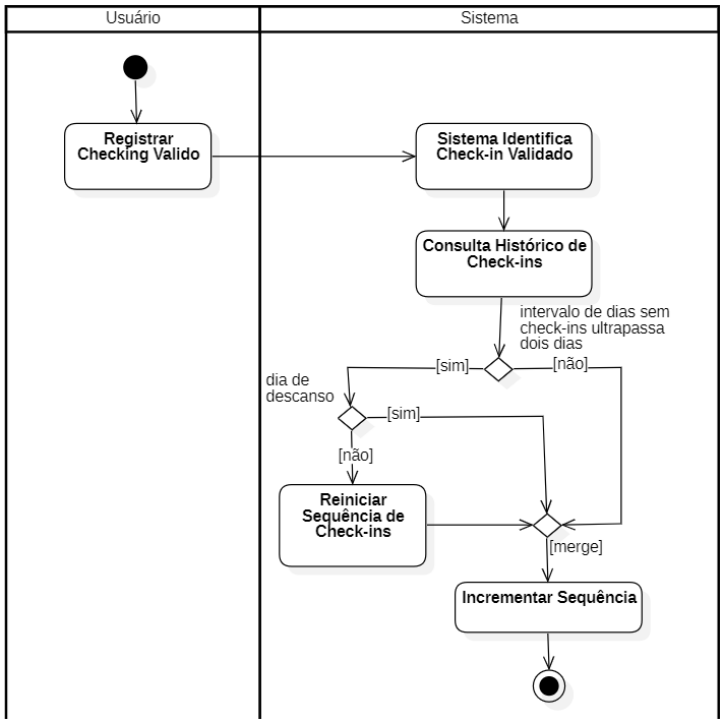


Diagrama de Atividade: RF_F03 - Calcular Check-in em Grupos

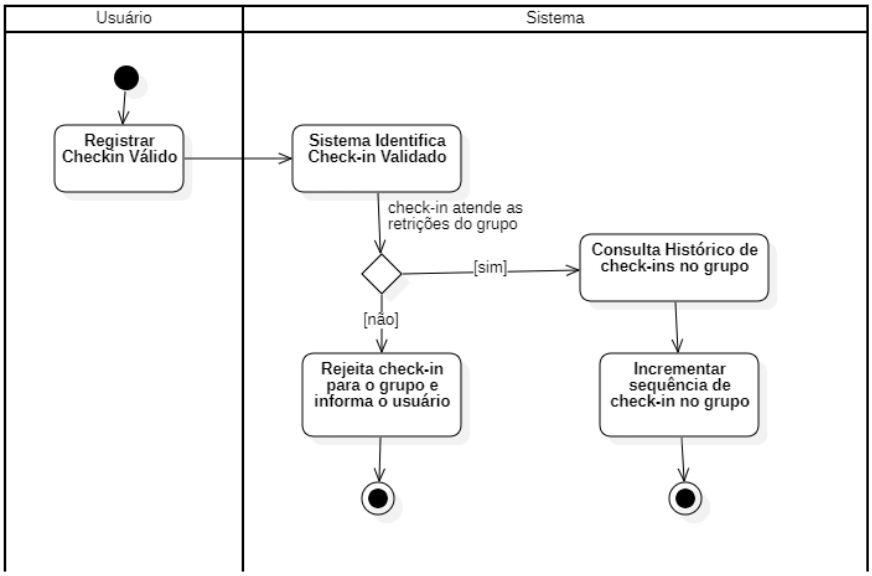


Diagrama de Atividade: RF_F04 - Estruturar Dinâmica de Grupo

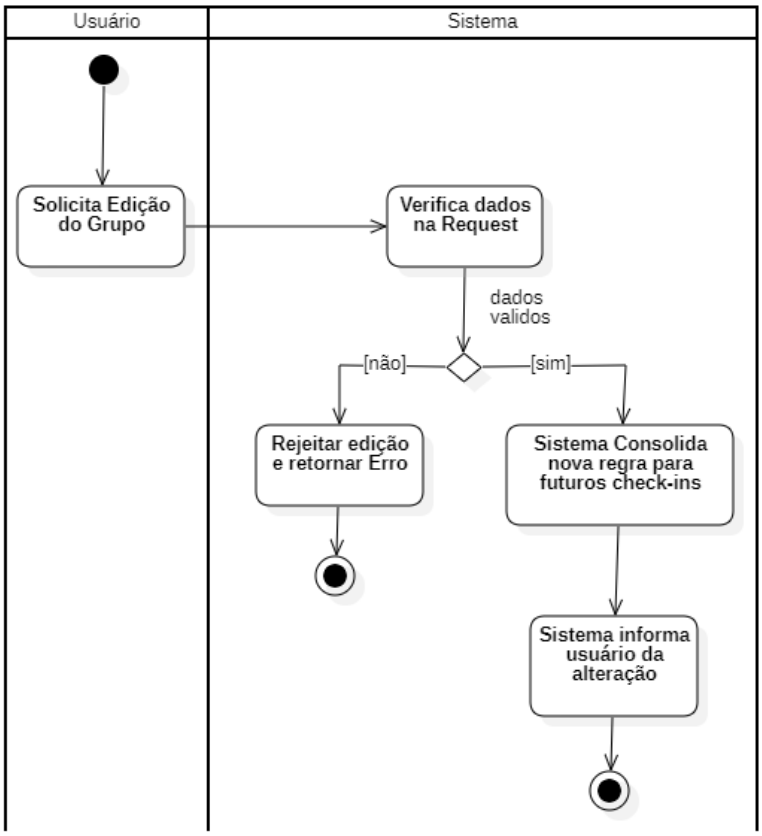
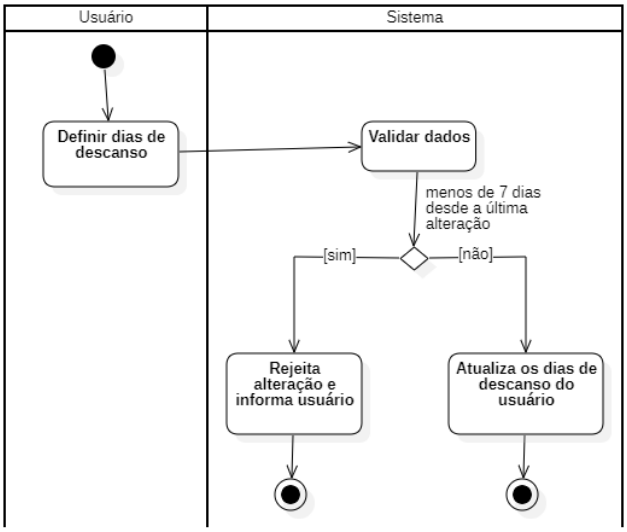
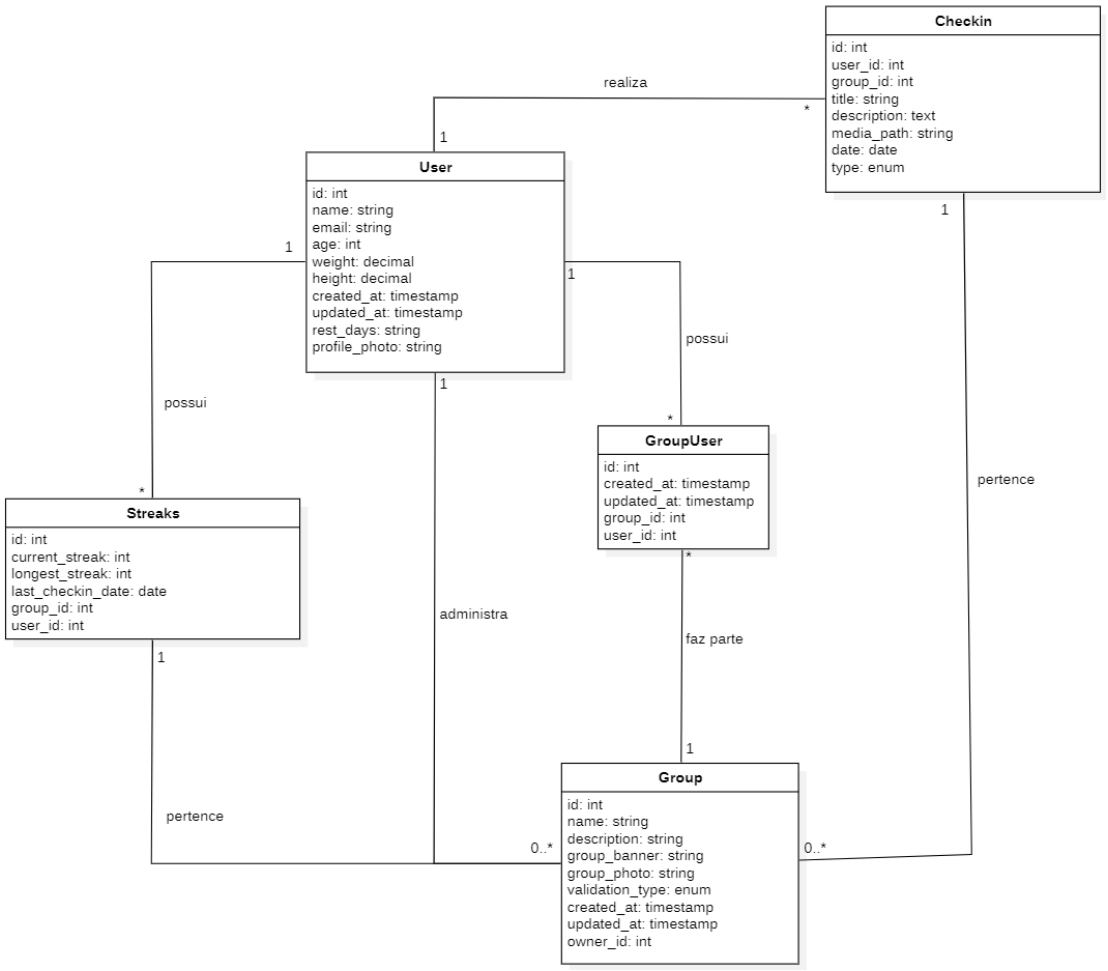


Diagrama de Atividade: RF_F05 - Planejar Jornada de Descanso



3.6 MODELO CONCEITUA



4. PROJETO DE SOFTWARE

4.1 DIAGRAMAS DE INTERAÇÃO

Diagrama de Interação: RF_F01 - Realizar Check-in

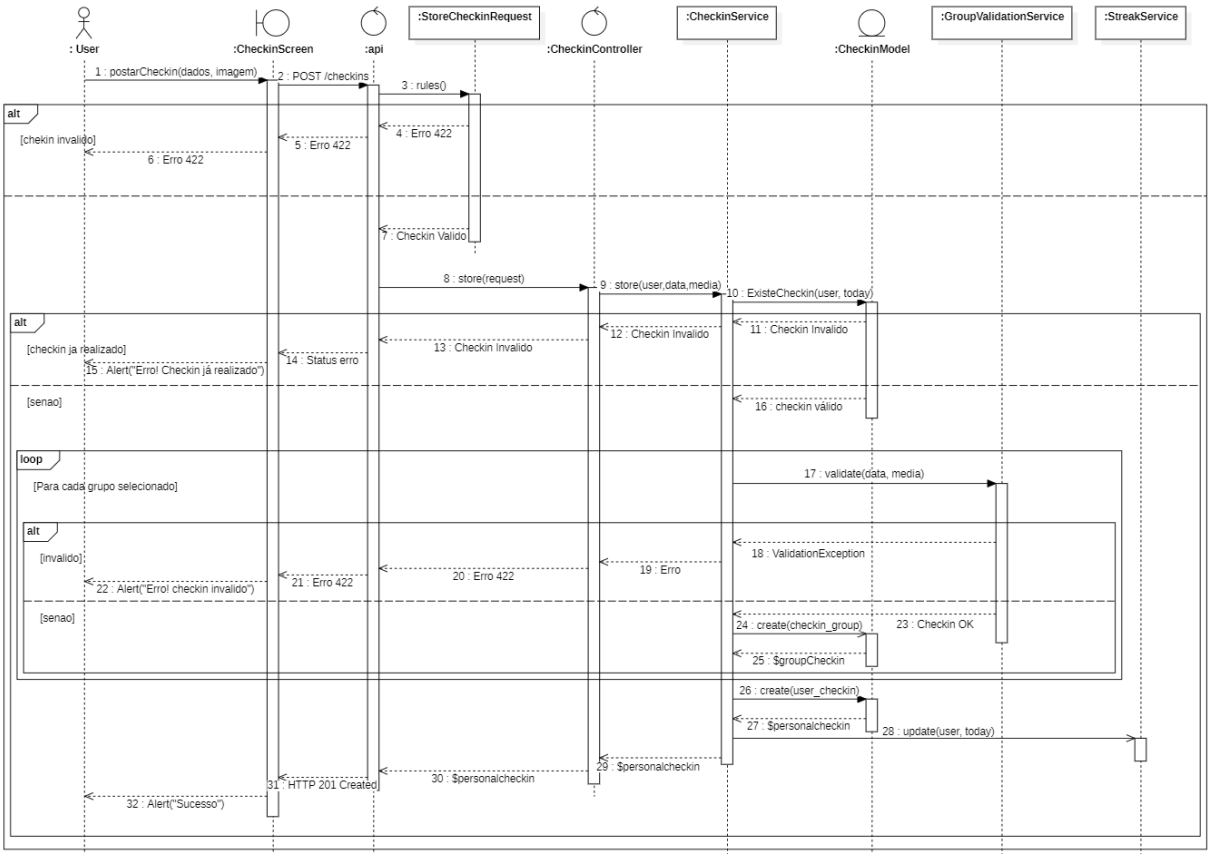


Diagrama de Interação: RF_F02 - Calcular Sequência Individual

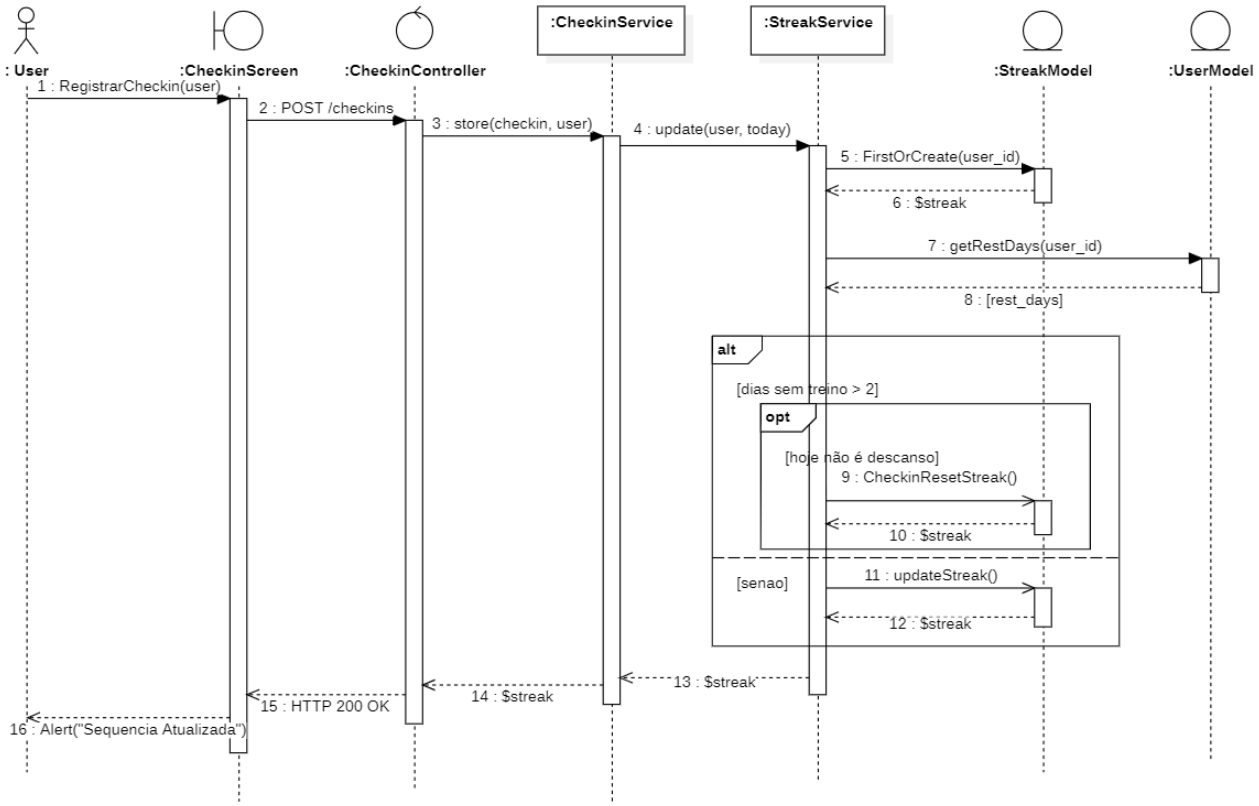


Diagrama de Interação: RF_F03 - Calcular Check-in em Grupos

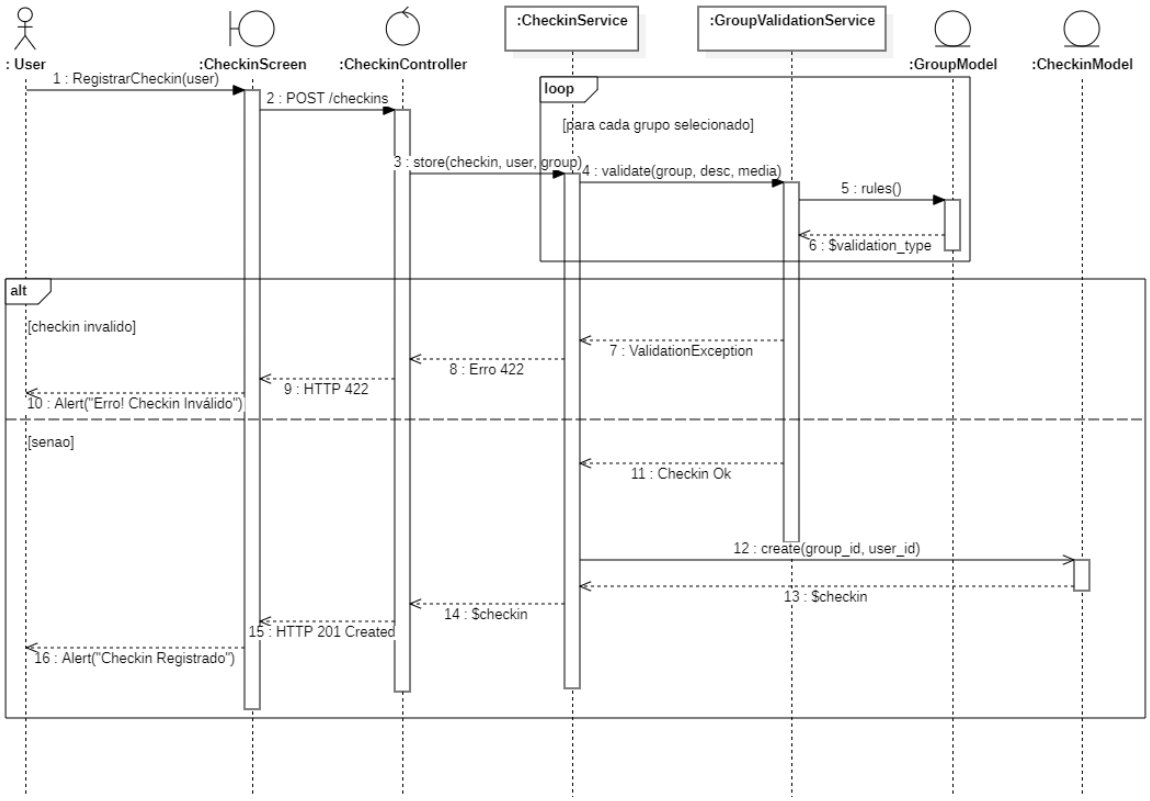


Diagrama de Interação: RF_F04 – Estruturar Dinâmica Grupo

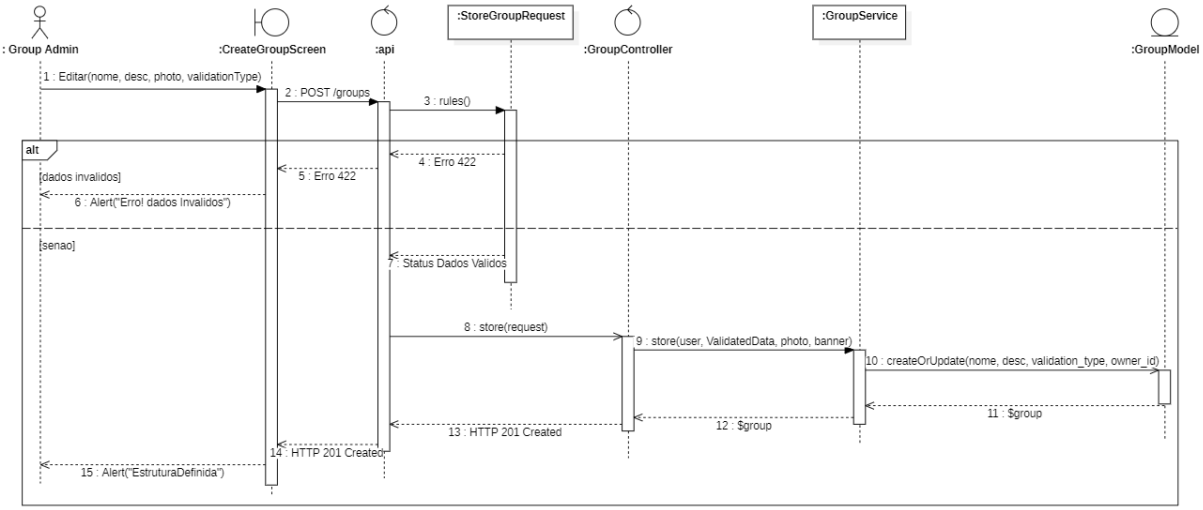
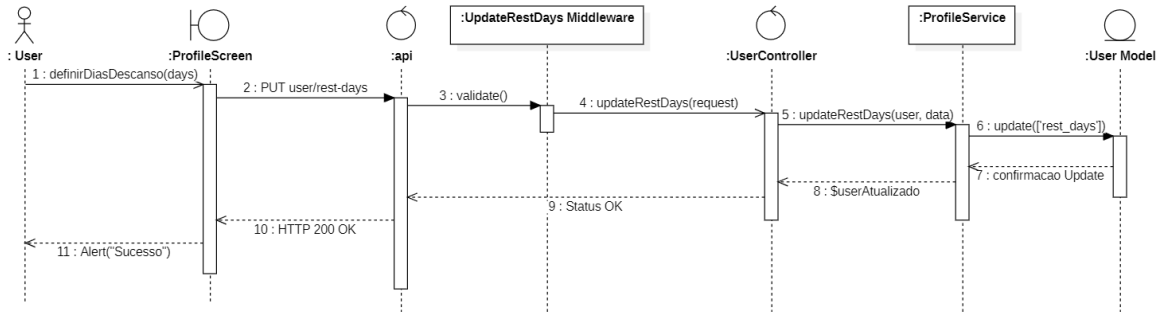
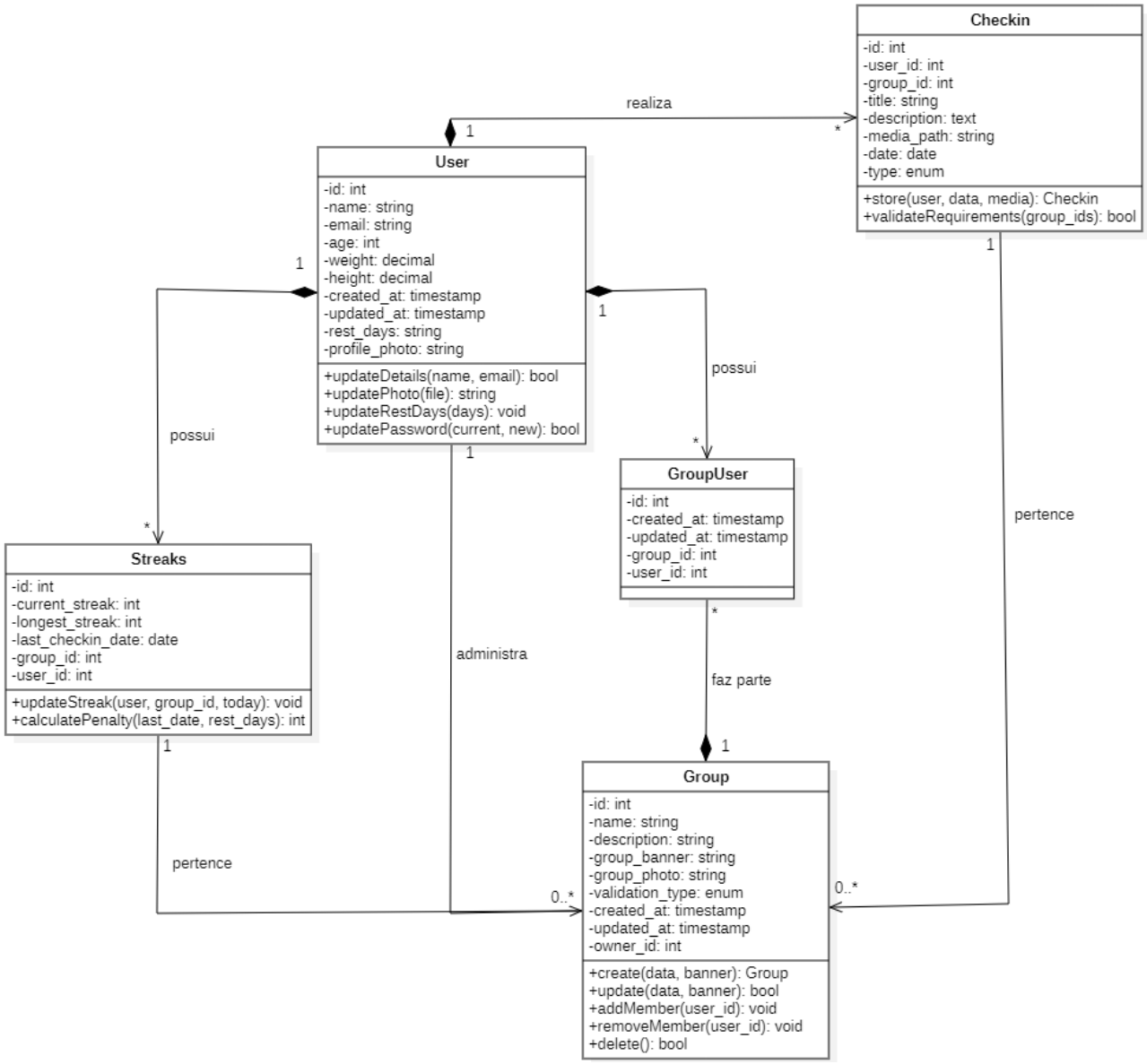


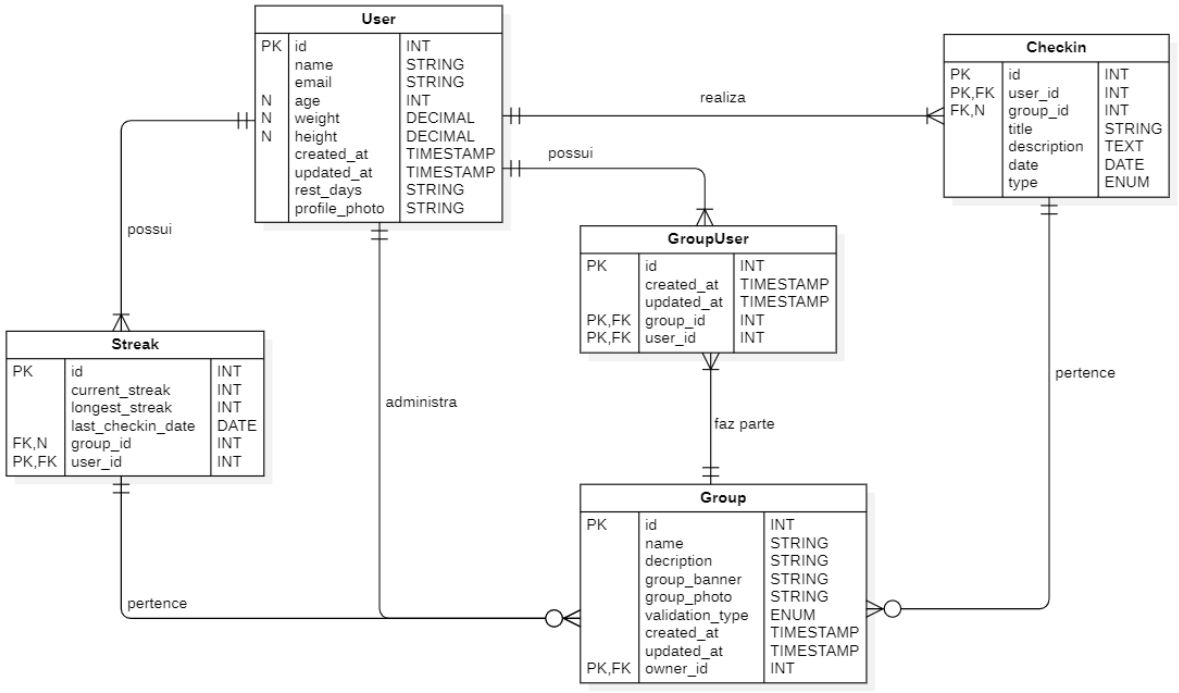
Diagrama de Interação: RF_F05 - Planejar Jornada de Descanso



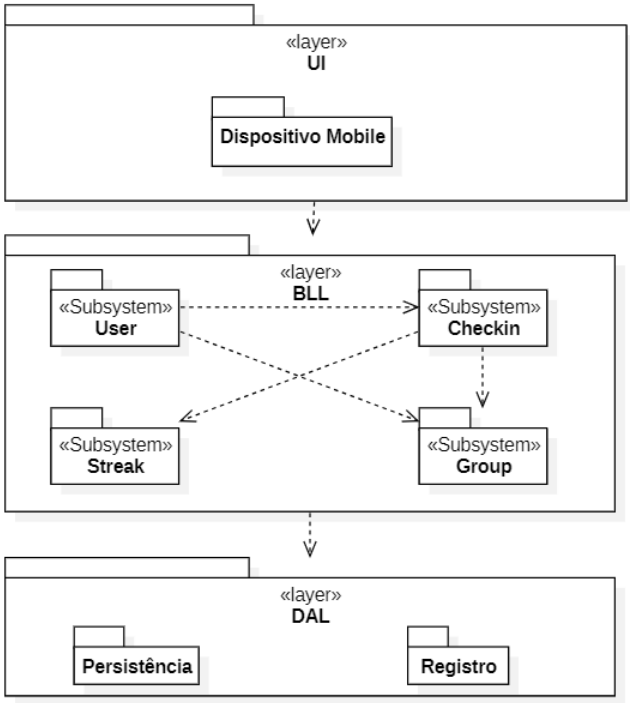
4.2 DIAGRAMA DE CLASSES



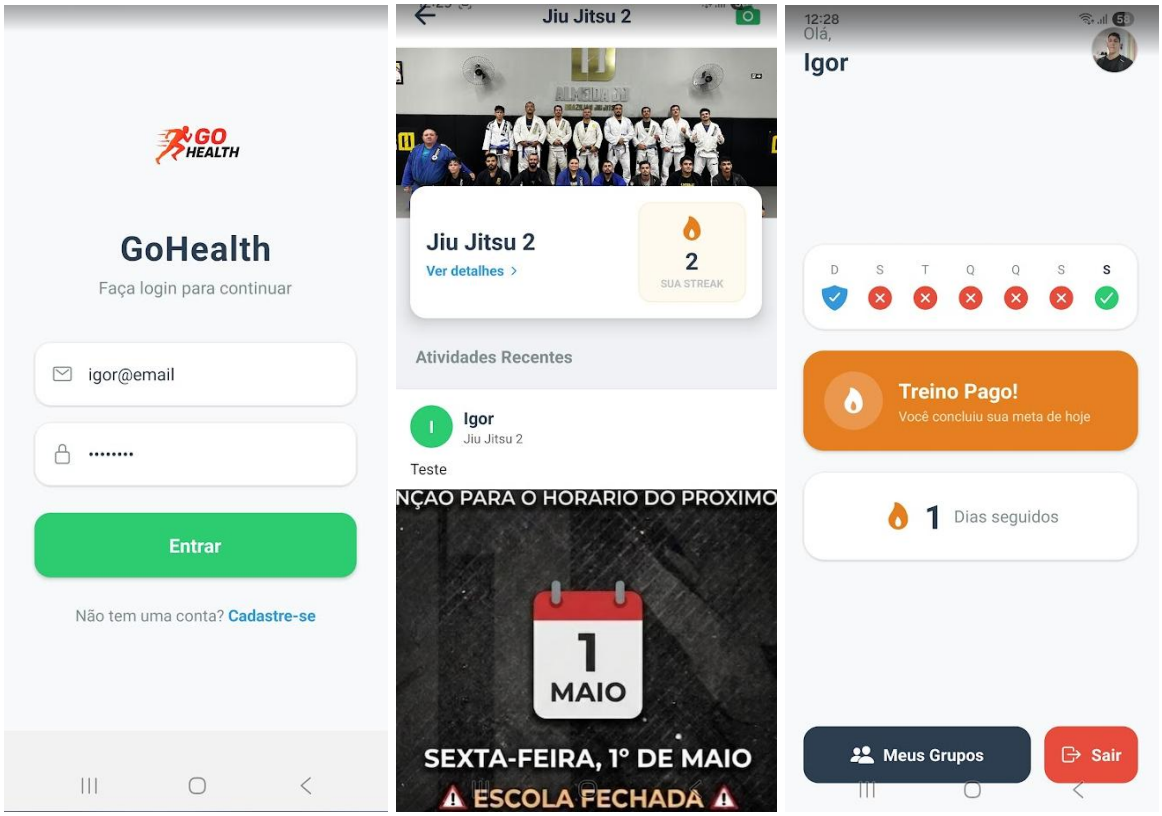
4.3 MODELAGEM DA BASE DE DADOS



4.4 DIAGRAMA DE PACOTES DA ARQUITETURA LÓGICA



4.6 OUTROS LAYOUTS DE TELAS



ANEXO 2 – Manual do Usuário

Manual da utilização do Go Health armazenado na plataforma Youtube,
Disponível em: https://youtu.be/b3ZCZtfCeWA (6 minutos)