

UNIVERSIDADE FEDERAL DO PARANÁ

ALISSON GABRIEL DOS SANTOS

GABRIEL ALAMARTINI TRONI

LEONARDO FELIPE SALGADO

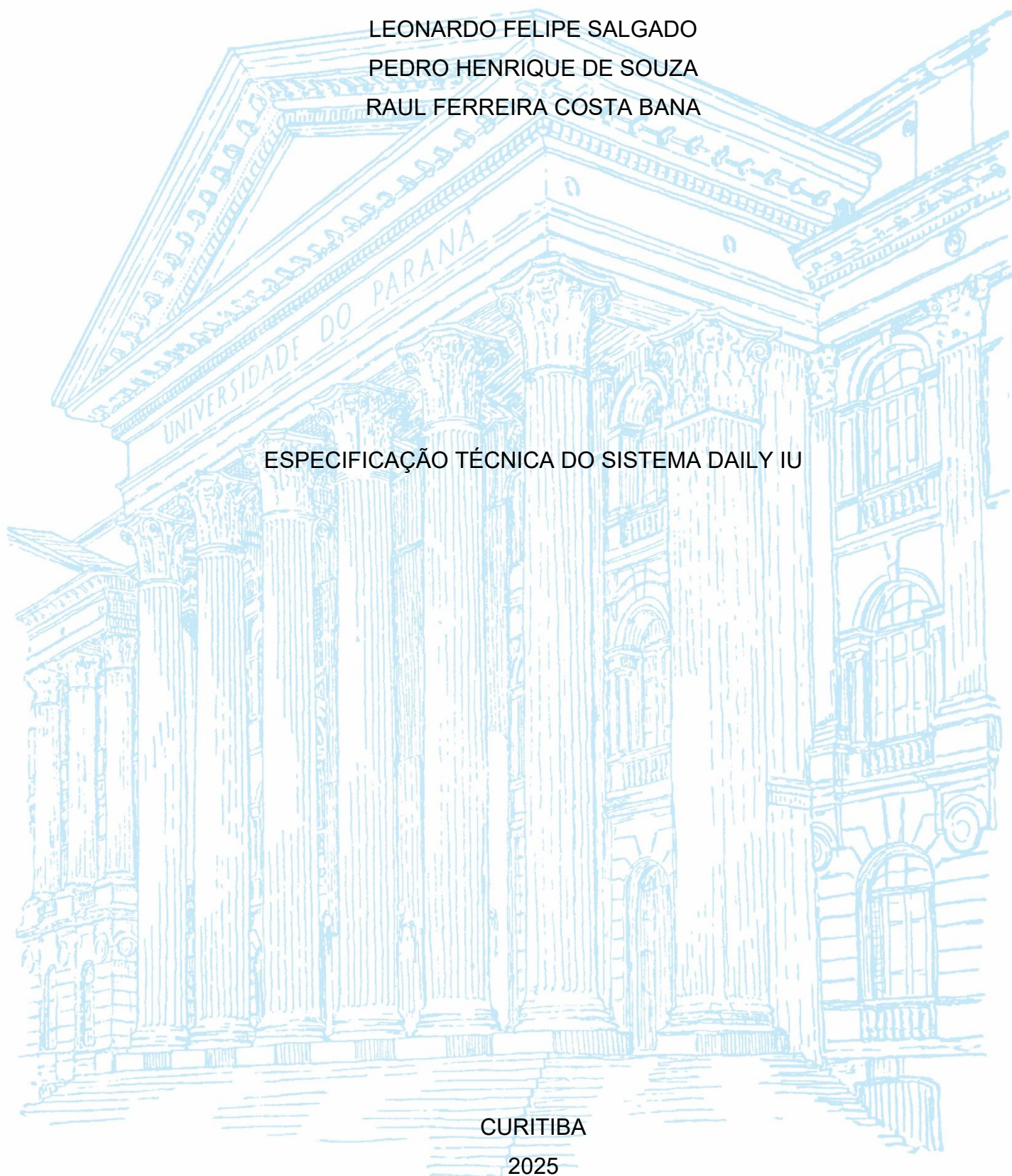
PEDRO HENRIQUE DE SOUZA

RAUL FERREIRA COSTA BANA

ESPECIFICAÇÃO TÉCNICA DO SISTEMA DAILY IU

CURITIBA

2025



ALISSON GABRIEL DOS SANTOS  
GABRIEL ALAMARTINI TRONI  
LEONARDO FELIPE SALGADO  
PEDRO HENRIQUE DE SOUZA  
RAUL FERREIRA COSTA BANA

## ESPECIFICAÇÃO TÉCNICA DO SISTEMA DAILY IU

Especificação de sistema apresentada ao curso de Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador(a): Prof. Dr. Alexander Robert Kutzke

CURITIBA

2025



UNIVERSIDADE FEDERAL DO PARANÁ  
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
Rua Alcides Vieira Arcoverde 1225, - - Bairro Jardim das Américas, Curitiba/PR, CEP 81520-260  
Telefone: 3360-5000 - <https://ufpr.br/>

Ata de Reunião

### TERMO DE APROVAÇÃO

ALISSON GABRIEL SANTOS  
RAUL FERREIRA COSTA BANA  
GABRIEL ALAMARTINI TRONI  
LEONARDO FELIPE SALGADO  
PEDRO HENRIQUE DE SOUZA

### DAILY IU: SOFTWARE PARA CONTROLE E MITIGAÇÃO DE INCONTINÊNCIA URINÁRIA EM MULHERES

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. Alexander Robert Kutzke  
Orientador – SEPT/UFPR

Prof. Razer Anthom Nizer Rojas Montaña  
SEPT/UFPR

Curitiba, 08 de dezembro de 2025.



Documento assinado eletronicamente por **RAZER ANTHOM NIZER ROJAS MONTANO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 08/12/2025, às 20:31, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **ALEXANDER ROBERT KUTZKE, PROFESSOR DO MAGISTERIO SUPERIOR**, em 09/12/2025, às 13:36, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **8452566** e o código CRC **46AA834A**.

## RESUMO

Este trabalho apresenta a documentação e os artefatos produzidos durante o desenvolvimento do software DailyIU, uma aplicação web e mobile voltada ao auxílio no controle da incontinência urinária por meio de exercícios, registro de informações e compartilhamento de conteúdos. A modelagem do sistema foi estruturada com base na UML, incluindo diagramas de casos de uso, classes e sequência, além do modelo físico do banco de dados, garantindo a representação clara da arquitetura e do comportamento do sistema. As Histórias de Usuário complementam essa modelagem ao descrever de forma objetiva as necessidades e expectativas dos usuários. A especificação das APIs, documentada com Swagger conforme o padrão OpenAPI, assegura padronização, clareza e consistência na comunicação entre serviços. A arquitetura integra frontend, backend e banco de dados, utilizando tecnologias como React Native, React.js, Next.js, Node.js, Spring Kotlin e SQL Server. Em conjunto, esses elementos demonstram a estrutura organizacional do DailyIU e sua capacidade de oferecer uma solução completa, funcional e bem documentada para o apoio à saúde e ao bem-estar do usuário.

**Palavras-chave:** Incontinência Urinária; Diário Miccional; Aplicativos Móveis de Saúde; Conteúdo Educativo; Acessibilidade;

## ABSTRACT

This work presents the documentation and artifacts produced during the development of the DailyIU software, a web and mobile application designed to support the management of urinary incontinence through exercises, information tracking, and content sharing. The system modeling was structured using UML, including use case, class, and sequence diagrams, as well as the physical database model, ensuring a clear representation of the system's architecture and behavior. User Stories complement this modeling by objectively describing user needs and expectations. The API specification, documented with Swagger following the OpenAPI standard, ensures standardization, clarity, and consistency in service communication. The system architecture integrates frontend, backend, and database layers, employing technologies such as React Native, React.js, Next.js, Node.js, Spring Kotlin, and SQL Server. Together, these elements demonstrate the organizational structure of DailyIU and its capacity to provide a complete, functional, and well-documented solution to support user health and well-being.

**Keywords:** Urinary Incontinence; Urination Diary; Mobile Health; Educational Content; Accessibility;

## LISTA DE FIGURAS

Figura 1 - Esquematização utilizada para a aplicação DailyIU .....	11
Figura 2 - Diagrama de Caso de Uso .....	13
Figura 3 - Diagrama de Classes .....	14
Figura 4 - Tela de Questionário Inicial.....	15
Figura 5 - Diagrama de Sequência - Responder Questionário.....	17
Figura 6 - Tela de Realizar Autocadastro .....	18
Figura 7 - Diagrama de Sequência - Realizar Autocadastro .....	19
Figura 8 - Tela de Realizar Login .....	20
Figura 9 - Diagrama de Sequência - Realizar Login.....	21
Figura 10 - Tela de Registrar Diário Miccional .....	22
Figura 11 - Diagrama de Sequência - Registrar Diário Miccional.....	23
Figura 12 - Tela de Consultar Histórico.....	24
Figura 13 - Diagrama de Sequência - Consultar Histórico .....	25
Figura 14 - Tela de Realizar Treino.....	26
Figura 15 - Diagrama de Sequência - Realizar Treino .....	28
Figura 16 - Tela de Avaliar Treino .....	29
Figura 17 - Diagrama de Sequência - Avaliar Treino .....	30
Figura 18 - Diagrama de Sequência - Repetir Questionário.....	31
Figura 19 - Tela de Interagir com Publicações .....	32
Figura 20 - Diagrama de Sequência - Interagir com Publicação .....	34
Figura 21 - Tela de Manter Perfil.....	35
Figura 22 - Diagrama de Sequência - Manter Perfil .....	36
Figura 23 - Tela de Definir Acessibilidade.....	37
Figura 24 - Diagrama de Sequência - Definir Acessibilidade .....	38
Figura 25 - Tela de Gerenciar Notificações .....	39
Figura 26 - Diagrama de Sequência - Gerenciar Notificações .....	40
Figura 27 - Tela de "Sobre o Sistema" .....	41
Figura 28 - Diagrama de Sequência - Acessar "Sobre o Sistema".....	42
Figura 29 - Tela "Fale Conosco" .....	43
Figura 30 - Diagrama de Sequência - Acessar "Fale Conosco" .....	44
Figura 31 - Tela de Recuperação de Senha .....	45
Figura 32 - Diagrama de Sequência - Recuperar Senha .....	46
Figura 33 - Tela de Realizar Postagem.....	47

Figura 34 - Diagrama de Sequência - Realizar Postagem .....	48
Figura 35 - Tela de Denunciar Postagem.....	49
Figura 36 - Diagrama de Sequência - Denunciar Postagem .....	51
Figura 37 - Tela de Manter Usuários .....	52
Figura 38 - Diagrama de Sequência - Manter Usuários .....	53
Figura 39 - Tela de Conceder Perfil de "Profissional da Saúde" .....	54
Figura 40 - Diagrama de Sequência - Conceder Perfil de "Profissional da Saúde" ...	55
Figura 41 - Tela de Gerenciar Denúncias .....	56
Figura 42 - Diagrama de Sequência - Gerenciar Denúncias .....	58
Figura 43 - Tela de Manter Exercícios .....	59
Figura 44 - Diagrama de Sequência - Manter Exercícios .....	61
Figura 45 - Tela de Manter Treino.....	62
Figura 46 - Diagrama de Sequência - Manter Treino .....	64
Figura 47 - Tela de Manter Plano de Treino.....	65
Figura 48 - Diagrama de Sequência - Manter Plano de Treino .....	67
Figura 49 - Tela de Manter Atributo de Exercício .....	68
Figura 50 - Diagrama de Sequência - Manter Atributo de Exercício .....	70
Figura 51 - Tela de Manter Categoria de Exercício .....	71
Figura 52 - Diagrama de Sequência - Manter Categoria de Exercício .....	73
Figura 53 - Tela de Manter Categoria de Publicação .....	74
Figura 54 - Diagrama de Sequência - Manter Categoria de Publicação .....	76
Figura 55 - Modelo Lógico do Banco de Dados .....	77
Figura 56 - Diagrama da Arquitetura dos Recursos de Infraestrutura Hospedados na Azure.....	78

## SUMÁRIO

<b>CONTEXTUALIZAÇÃO .....</b>	<b>9</b>
<b>REFERÊNCIAS.....</b>	<b>12</b>
<b>APÊNDICE A - DIAGRAMA DE CASOS DE USO .....</b>	<b>13</b>
<b>APÊNDICE B - DIAGRAMA DE CLASSES.....</b>	<b>14</b>
<b>APÊNDICE C - HISTÓRIAS DE USUÁRIO, PROTÓTIPOS DE TELA E DIAGRAMAS DE SEQUÊNCIA.....</b>	<b>15</b>
<b>APÊNDICE D - DIAGRAMA LÓGICO DO BANCO DE DADOS .....</b>	<b>77</b>
<b>APÊNDICE E - ARQUITETURA DA INFRAESTRUTURA.....</b>	<b>78</b>
<b>APÊNDICE F - ESPECIFICAÇÃO DE APPLICATION PROGRAMMING INTERFACES (APIS).....</b>	<b>79</b>

## CONTEXTUALIZAÇÃO

Este trabalho reúne toda a documentação e os diagramas elaborados para o desenvolvimento do software DailyIU, cujo propósito é auxiliar o usuário no controle da incontinência urinária por meio da prática de exercícios, bem como criar uma rede de informação.

A aplicação DailyIU, voltada a auxiliar o usuário no controle da incontinência urinária, permite ao usuário acompanhar seu diário miccional, treinos para a melhora da condição, além de visualizar conteúdos sobre o tema, como relatos de outros usuários e dicas de saúde e bem-estar. Também possibilita a criação de conteúdos próprios para compartilhar com outros usuários, além da configuração de perfil e notificações, bem como definições de acessibilidade.

O sistema oferece ainda ferramentas para a administração, que pode administrar exercícios, treinos e planos de treinos para os usuários, além de gerenciar perfis de usuários e publicações denunciadas.

Sommerville (2018, p. 85) destaca que a modelagem de sistemas envolve a criação de modelos abstratos que representam diferentes visões de um sistema, geralmente por meio de notações gráficas baseadas em diagramas da UML (*Unified Modeling Language*).

Booch, Rumbaugh e Jacobson (2012, p. 45) afirmam:

A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais, tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis.

Conforme os autores Booch, Rumbaugh e Jacobson (2012), a UML fornece uma linguagem padronizada para representar a estrutura e o comportamento de sistemas de *software*, tornando possível visualizar, especificar e documentar seus diversos aspectos. No contexto deste projeto, a modelagem foi organizada por meio de diagramas que representam as partes do sistema e suas interações.

O diagrama de Casos de Uso (Apêndice A) apresenta as interações entre usuários (atores) e o sistema, destacando suas principais funcionalidades e a visão geral do uso da aplicação. O diagrama de Classes (Apêndice B) descreve a estrutura

estática da aplicação, incluindo classes, atributos, métodos e relacionamentos, servindo como base lógica da arquitetura interna do *software*.

Segundo Cohn (2004), as Histórias de Usuário contribuem para o desenvolvimento ao expressar necessidades e expectativas dos usuários de forma clara e orientada ao valor, funcionando como um elo entre a perspectiva técnica e o ponto de vista do usuário final.

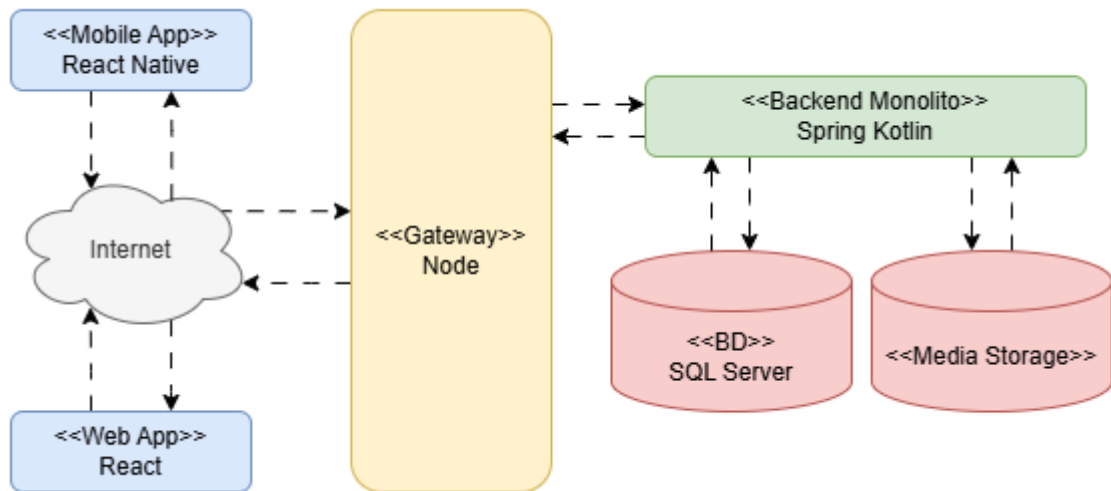
Os diagramas de Sequência (Apêndice C) demonstram o fluxo de mensagens entre objetos e usuários ao longo do tempo. Cada diagrama foi disposto logo abaixo de sua respectiva História de Usuário, garantindo correspondência direta entre a funcionalidade descrita e sua dinâmica de execução.

O modelo físico do Banco de Dados (Apêndice D) organiza tabelas, campos e relacionamentos, definindo a estrutura de armazenamento de dados. Ele traduz o modelo conceitual para uma implementação concreta, assegurando integridade e coerência das informações.

A especificação das *Application Programming Interfaces* (Apêndice F) apresenta a documentação estruturada dos serviços disponibilizados pelo sistema, descrevendo seus *endpoints*, métodos, parâmetros e respostas. Essa especificação é implementada por meio do *Swagger*, ferramenta que adota o padrão *OpenAPI* para padronizar e detalhar o comportamento das APIs (SWAGGER, 2025). Com isso, torna-se possível visualizar e testar interativamente cada funcionalidade, garantindo maior clareza na comunicação entre desenvolvedores, facilitando integrações e assegurando que o consumo dos serviços ocorra de forma consistente e conforme o projeto.

A arquitetura da aplicação, conforme Figura 1, integra *frontend*, *backend* e banco de dados, utilizando React Native para o aplicativo *Mobile* (REACT NATIVE, 2025), React.js (REACT, 2025), Next.js (NEXT, 2025) e Tailwind CSS (TAILWINDCSS, 2025) para o *Frontend Web*, Node.js (NODE, 2025) e Express.js (EXPRESS, 2025) na construção do API Gateway, Spring Kotlin (Kotlin, 2025) na criação do *Backend* e SQL Server como Banco de Dados (SQL, 2025). Essa estrutura garantiu bom desempenho, organização entre camadas e fluxo consistente de dados. O *frontend* concentra a interação com o usuário, o *backend* aplica as regras de negócio e o banco de dados armazena informações essenciais ao funcionamento do sistema.

Figura 1 - Esquematização utilizada para a aplicação DailyIU



Fonte: Os Autores (2025)

Por fim, os conteúdos seguintes apresentam, na ordem, os Apêndices A até F — Diagrama de Casos de Uso, Diagrama de Classes, Diagramas de Sequência e Modelo Físico do Banco de Dados, Arquitetura de Infraestrutura e Especificação de *Application Programming Interfaces* (API) — reunindo os principais artefatos gerados durante a modelagem e o desenvolvimento do sistema DailyIU. Esses materiais evidenciam a organização estrutural e relacional do sistema, complementando a fundamentação teórica e oferecendo uma visão abrangente da aplicação desenvolvida.

## REFERÊNCIAS

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: Guia do usuário. Tradução de Fábio Freitas da Silva e Cristina de Amorim Machado. Rio de Janeiro: Elsevier, 2012.

COHN, Mike. User Stories: Aplicando Histórias de Usuário. São Paulo: Casa do Código, 2015.

EXPRESS - Node.js web application framework. Disponível em: <https://expressjs.com/>. Acesso em 10 dez. 2025.

KOTLIN documentation. Disponível em: <https://kotlinlang.org/docs/home.html>. Acesso em: 10 dez. 2025.

NEXT.js documentation. Disponível em: <https://nextjs.org/docs>. Acesso em 10 dez. 2025.

NODE.js documentation. Disponível em: <https://nodejs.org/pt-br>. Acesso em 10 dez. 2025.

REACT documentation. Disponível em: <https://react.dev/>. Acesso em 10 dez. 2025

REACT NATIVE documentation. Disponível em: <https://reactnative.dev/docs/getting-started>. Acesso em 10 dez. 2025.

SOMMERVILLE, Ian. Engenharia de Software. 10. ed. São Paulo: Pearson Education do Brasil, 2018. ISBN 978-85-430-2497-4.NA ABNT.

SQL server technical documentation. <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver17>. Acesso em > 10 dez. 2025.

SWAGGER api documentation & Design Tools for Teams. Disponível em: <https://swagger.io/>. Acesso em 10 dez. 2025.

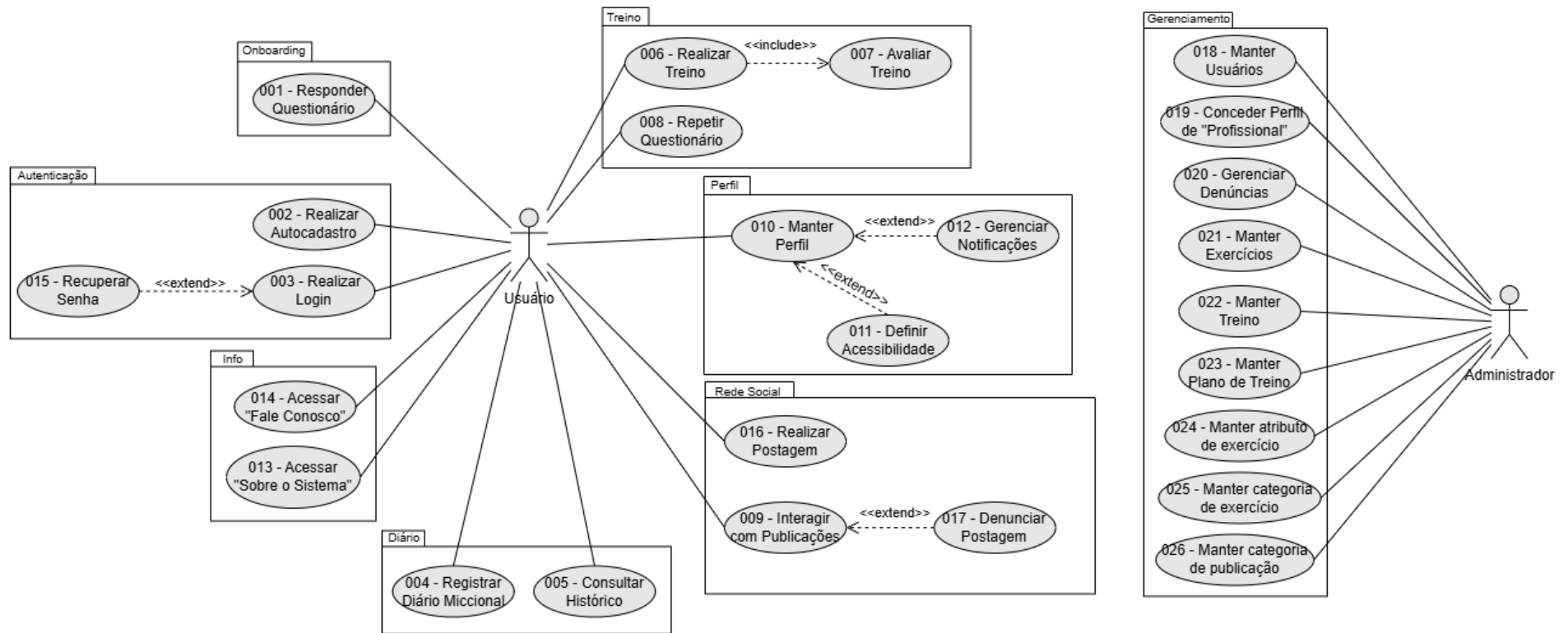
TAILWINDCSS documentation. Disponível em: <https://v2.tailwindcss.com/docs>. Acesso em: 10 dez. 2025.

## APÊNDICE A - DIAGRAMA DE CASOS DE USO

O diagrama de caso de uso a seguir representa os casos de uso relacionados à aplicação desenvolvida.

Figura 2 - Diagrama de Caso de Uso

### Diagrama de Caso de Uso - Nível 2





## APÊNDICE C - HISTÓRIAS DE USUÁRIO, PROTÓTIPOS DE TELA E DIAGRAMAS DE SEQUÊNCIA

### HU 1. HISTÓRIA DE USUÁRIO 001 – RESPONDER QUESTIONÁRIO

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Responder um questionário de triagem
<b>Para</b>	Que seja gerado um plano de treinos personalizado de acordo com a minha condição e intensidade dos casos de incontinência

Figura 4 - Tela de Questionário Inicial

23:59 23°C 23°C ... 5G 55%

< Voltar

Etapa 4 / 6

Gostaríamos de saber a quantidade de urina que você pensa que perde (Assinale a alternativa que melhor representa a sua percepção)

Nenhuma

Uma pequena quantidade

Uma quantidade moderada

Uma quantidade elevada

Continuar

### Critérios de Aceitação

<b>Critério 1</b>
Deve montar perfil de paciente do usuário e gerar o plano de treinos.
<b>Regra</b>
Quando o usuário responder a última pergunta do questionário, o sistema deve armazenar as informações no banco de dados, classificar o tipo do usuário e montar o plano de treinos.

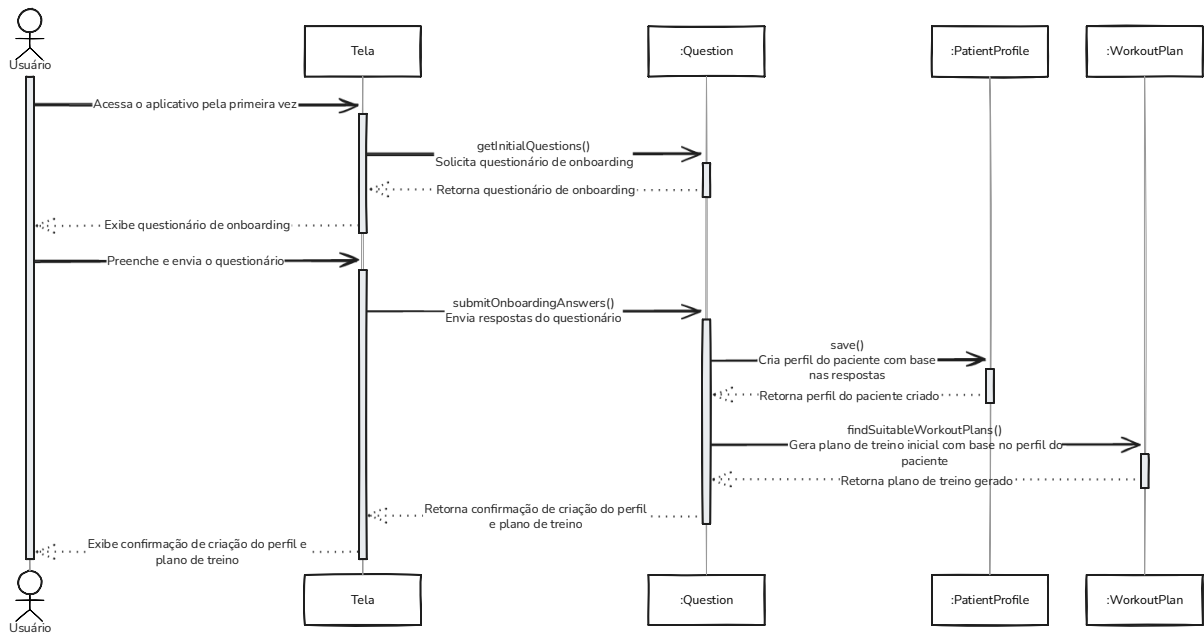
<b>Critério 2</b>
Não deve continuar caso haja campos inválidos.
<b>Regra</b>
Caso o usuário tente prosseguir com campos obrigatórios vazios, o sistema deve exibir erro. O sistema não deve permitir tipos de dados inválidos, como caracteres em campos numéricos. Quando isso ocorrer, o caractere inválido é apagado automaticamente.

<b>Critério 3</b>
Deve categorizar o nível de incontinência baseado nas respostas do formulário inicial.
<b>Regra</b>
A soma dos pontos das questões 3, 4 e 5 define o nível: 0-5 (leve), 6-12 (moderada), 13-18 (grave), 19-21 (muito grave).

<b>Critério 4</b>
Deve bloquear treinos para casos graves e muito graves.
<b>Regra</b>
Usuários classificados como grave ou muito grave não podem acessar os treinos e recebem mensagem orientando procurar um profissional.

## Diagrama de Sequência 001 – Responder Questionário

Figura 5 - Diagrama de Sequência - Responder Questionário




## HU 2. HISTÓRIA DE USUÁRIO 002 – REALIZAR AUTOCADASTRO

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Fazer meu cadastro no sistema
<b>Para</b>	Ter uma conta

Figura 6 - Tela de Realizar Autocadastro

0:00 5G 64

  
DailyIU  
Criar sua conta

Nome

E-mail

Senha

Confirmar senha

Aceito os termos de uso

**Criar Conta**

ou

Já tem conta? [Entrar](#)

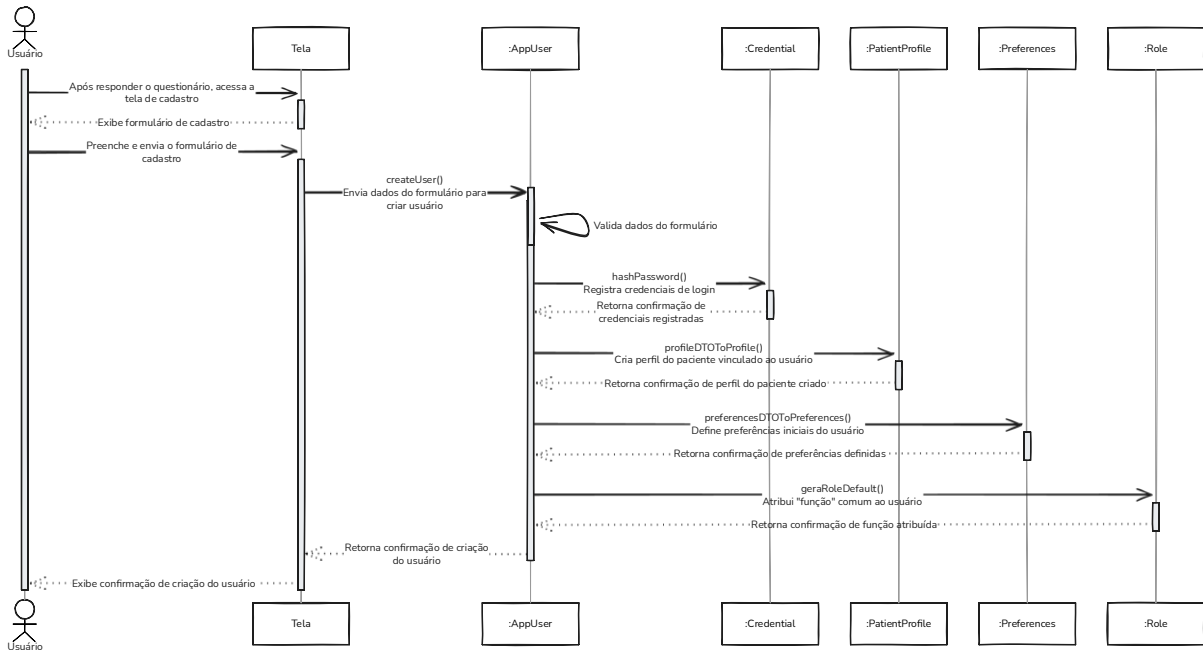
### Critérios de Aceitação

<b>Critério 1</b>
Deve ser possível criar uma conta através de uma tela em que o usuário informa: <ul style="list-style-type: none"> <li>a) Nome</li> <li>b) E-mail</li> <li>c) Senha</li> </ul>
<b>Regra</b>
O E-mail deve ser único

<b>Critério 2</b>
Não deve continuar caso haja campos inválidos.
<b>Regra</b>
Caso o usuário tente prosseguir com campos obrigatórios vazios, o sistema deve exibir erro. O sistema não deve permitir tipos de dados inválidos, como caracteres em campos numéricos. Quando isso ocorrer, o caractere inválido é apagado automaticamente.

### Diagrama de Sequência 002 – Realizar Autocadastro

Figura 7 - Diagrama de Sequência - Realizar Autocadastro



## HU 3. HISTÓRIA DE USUÁRIO 003 – REALIZAR LOGIN

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Fazer meu login no sistema
<b>Para</b>	Poder acessar minha conta e ter acesso às funcionalidades do sistema

Figura 8 - Tela de Realizar Login

0:08 5G 64

DailyIU

Entrar na sua conta

E-mail

luiza@g.com

Senha \*

Lembrar de mim

Entrar

Ainda não tem conta? [Criar Conta](#)

[Esqueceu a senha?](#)

1 2 3 4 5 6 7 8 9 0  
q w e r t y u i o p  
a s d f g h j k l  
↑ z x c v b n m ↵  
?123 @ PT · EN . ✓

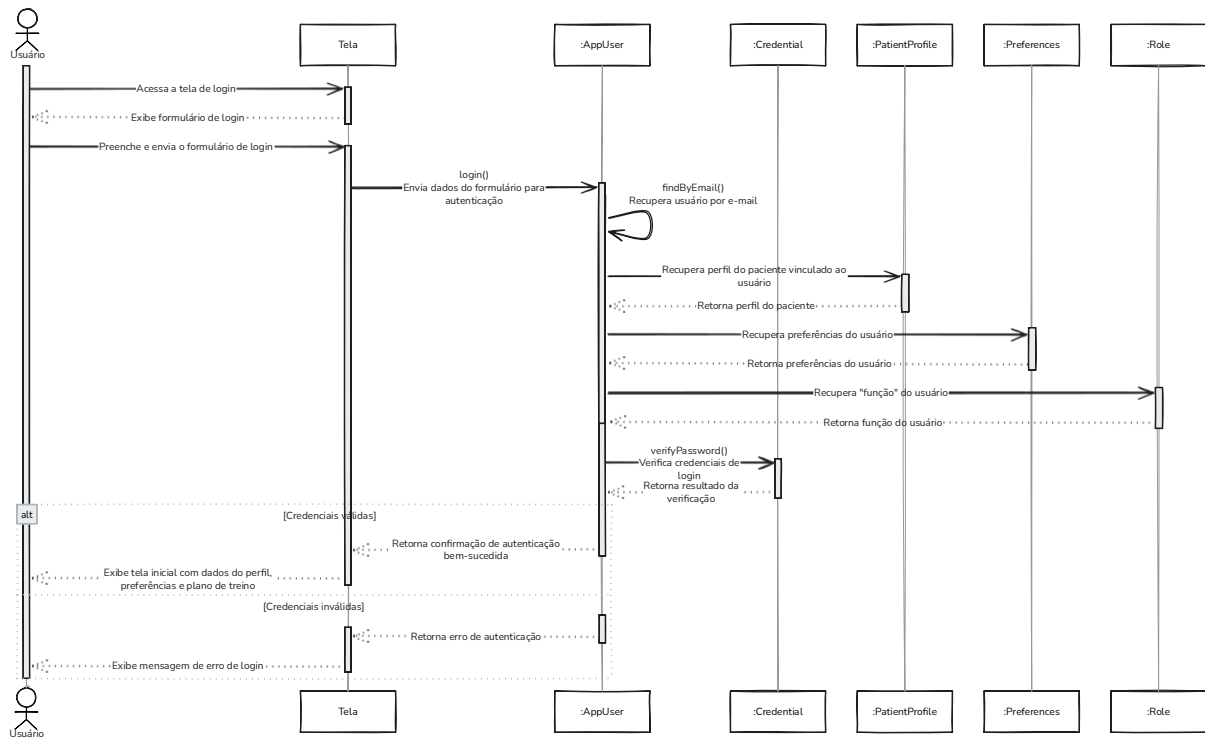
### Critérios de Aceitação

Critério 1
Deve permitir o usuário preencher seu e-mail e senha para fazer login

Critério 2
Caso o e-mail ou senha estejam incorretos, deve retornar a mensagem de “Credenciais inválidas”

### Diagrama de Sequência 003 – Realizar Login

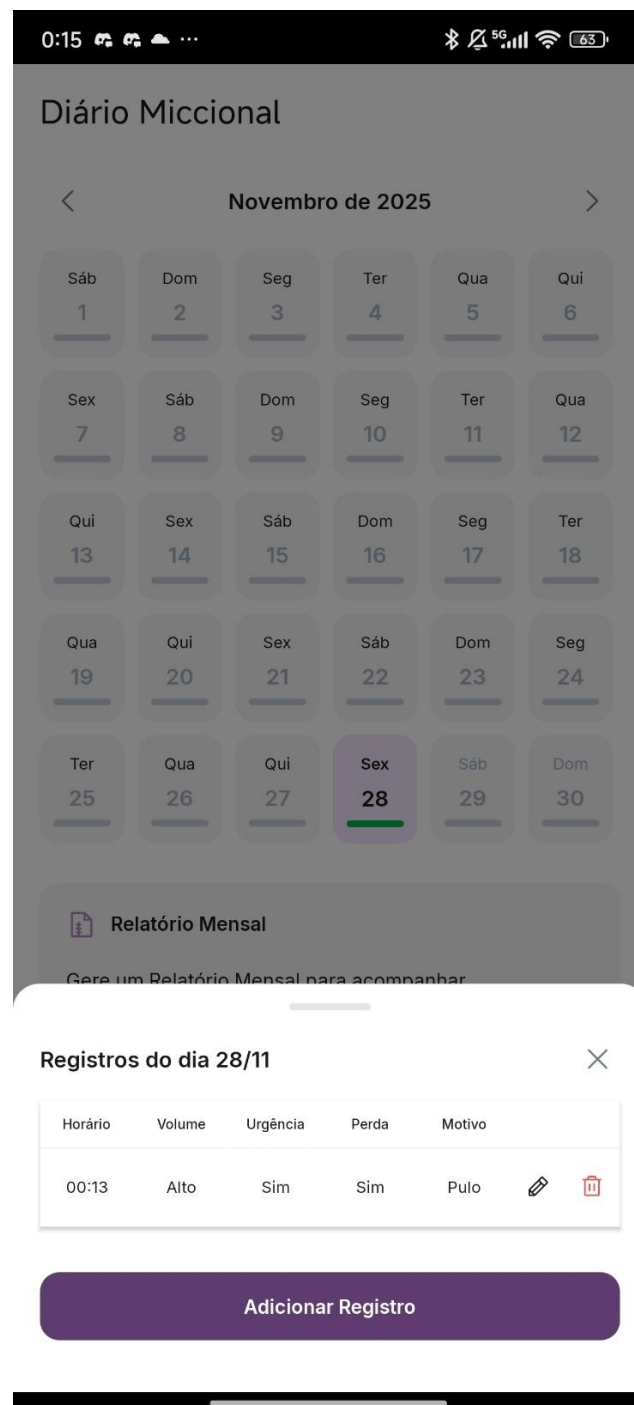
Figura 9 - Diagrama de Sequência - Realizar Login



## HU 4. HISTÓRIA DE USUÁRIO 004 – REGISTRAR DIÁRIO MICCIONAL

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Registrar diariamente a ocorrência de casos de incontinência e como me sinto em relação a isso
<b>Para</b>	Manter um histórico dessas informações no sistema

Figura 10 - Tela de Registrar Diário Miccional



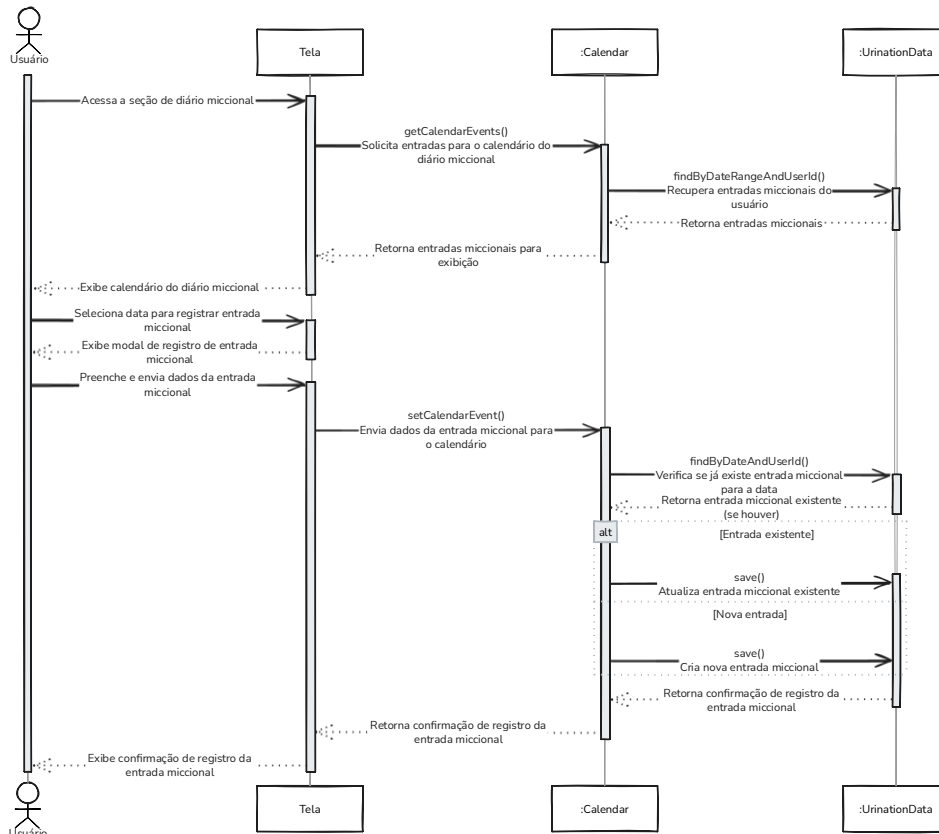
### Critérios de Aceitação

<b>Critério 1</b>
Deve registrar as informações miccionais do dia.
<b>Regra</b>
Quando o usuário clicar em “Adicionar registro” o sistema deve abrir um modal para perguntar os dados miccionais, horário, volume, urgência, peso e motivo. Depois que o usuário inserir as informações e clicar em “Confirmar”, o sistema deve registrar a informação no diário.

<b>Critério 2</b>
Deve alterar o registro.
<b>Regra</b>
Quando o usuário clicar no ícone de edição de um dos registros, o sistema deve abrir um modal com os dados que haviam sido salvos e permitir a edição.

### Diagrama de Sequência 004 – Registrar diário miccional

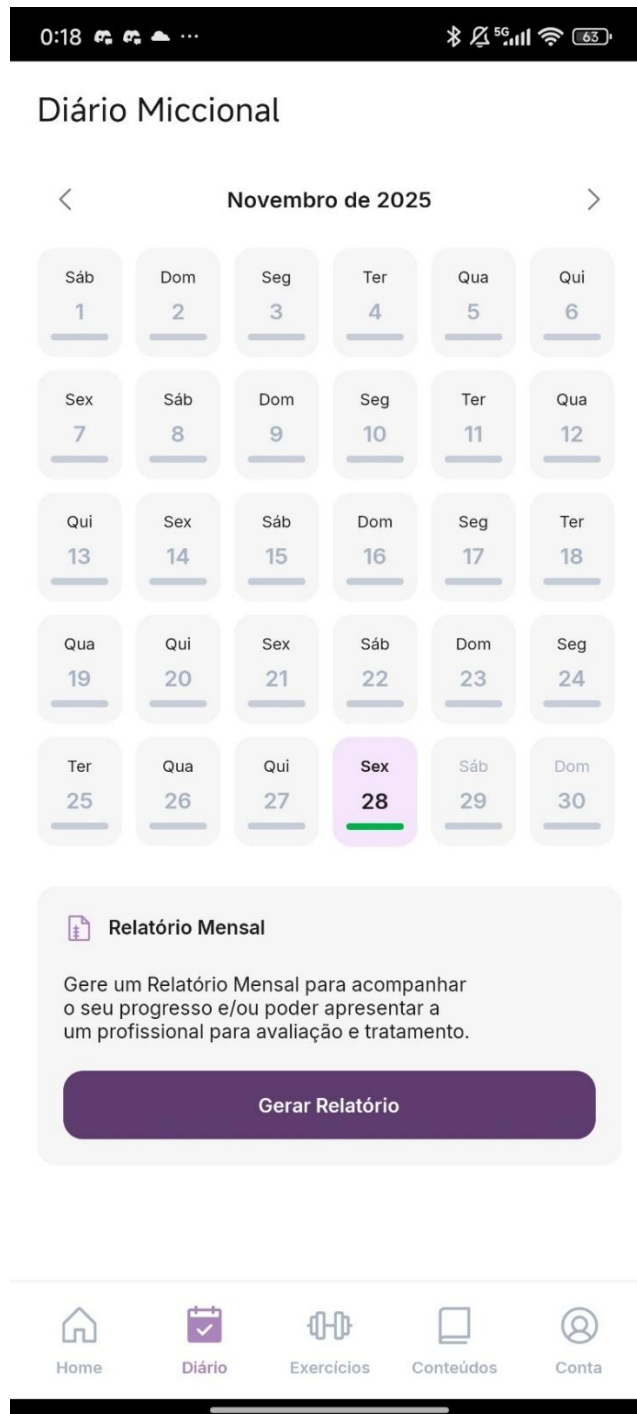
Figura 11 - Diagrama de Sequência - Registrar Diário Miccional



## HU 5. HISTÓRIA DE USUÁRIO 005 – CONSULTAR HISTÓRICO

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Consultar o histórico dos meus treinos e diário miccional
<b>Para</b>	Poder visualizar minha rotina e evolução ao longo do tempo

Figura 12 - Tela de Consultar Histórico



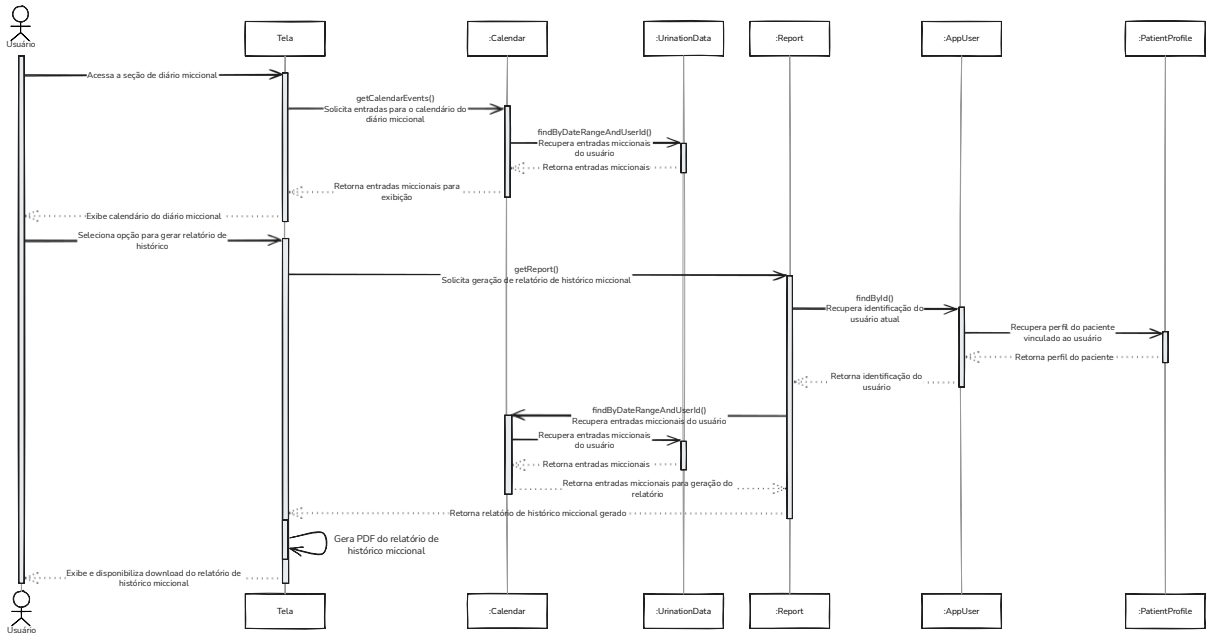
### Critérios de Aceitação

<b>Critério 1</b>
Deve mostrar o histórico de treinos e diário.
<b>Regra</b>
O sistema deve apresentar um calendário em que os dias treinados estarão marcados em destaque. Ao clicar em um dia, o treino realizado no dia é apresentado. Os dias com registro miccional também estarão marcados em destaque. Ao clicar em um dia, os registros do dia são apresentados.

<b>Critério 2</b>
Deve gerar e exportar relatório.
<b>Regra</b>
Quando o usuário clicar no botão “Gerar relatório”, o sistema deve gerar o relatório com o histórico. Quando o botão “Exportar relatório” for clicado, o sistema faz download do relatório.

### Diagrama de Sequência 005 – Consultar histórico

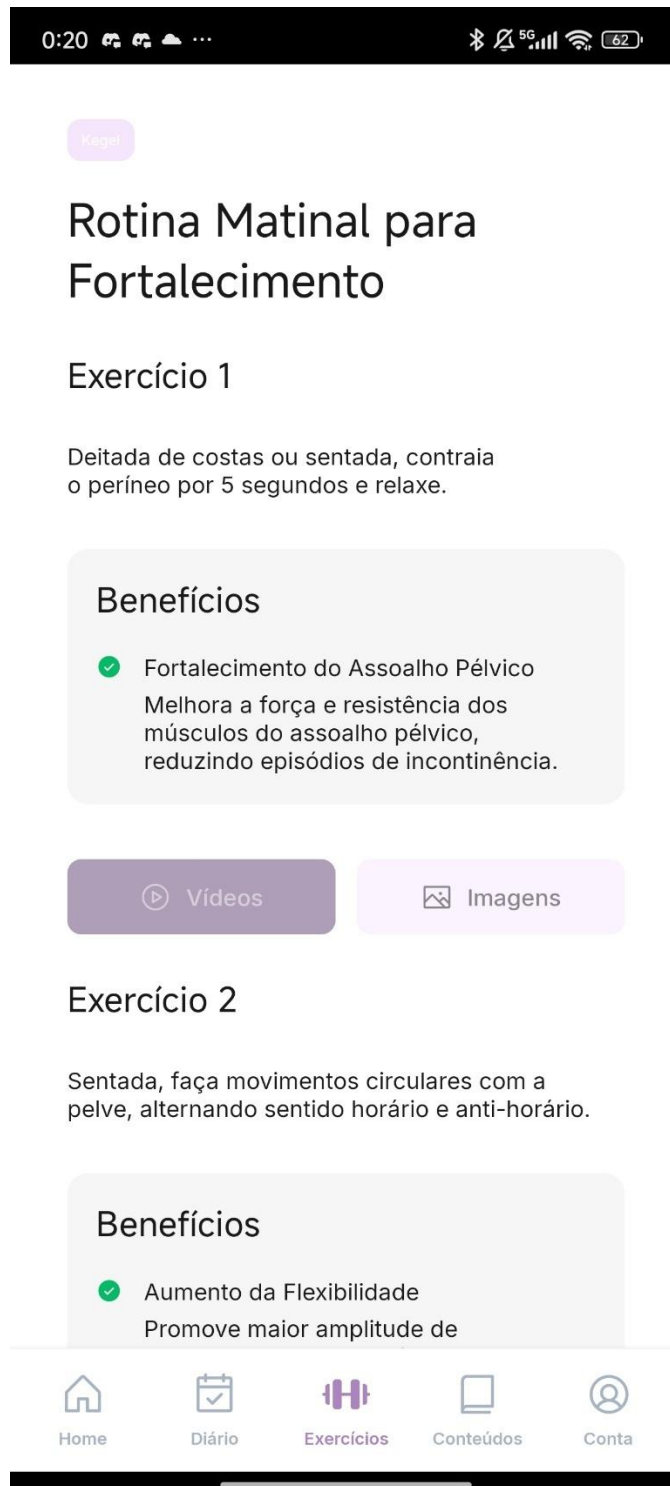
Figura 13 - Diagrama de Sequência - Consultar Histórico



## HU 6. HISTÓRIA DE USUÁRIO 006 – REALIZAR TREINO

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Acessar um treino com ilustrações e vídeos instrutivos
<b>Para</b>	Realizar os exercícios corretamente em casa

Figura 14 - Tela de Realizar Treino



### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir o início do treino.
<b>Regra</b>
Quando o usuário clicar no botão “Iniciar Treino”, deve ser apresentada uma tela com as ilustrações e/ou vídeos instrutivos e progresso (exercício x de y).

<b>Critério 2</b>
Deve permitir alternar as instruções entre vídeo e ilustração.
<b>Regra</b>
Durante qualquer momento do treino, o pode trocar a maneira que as instruções são apresentadas, alternando entre vídeo e ilustração.

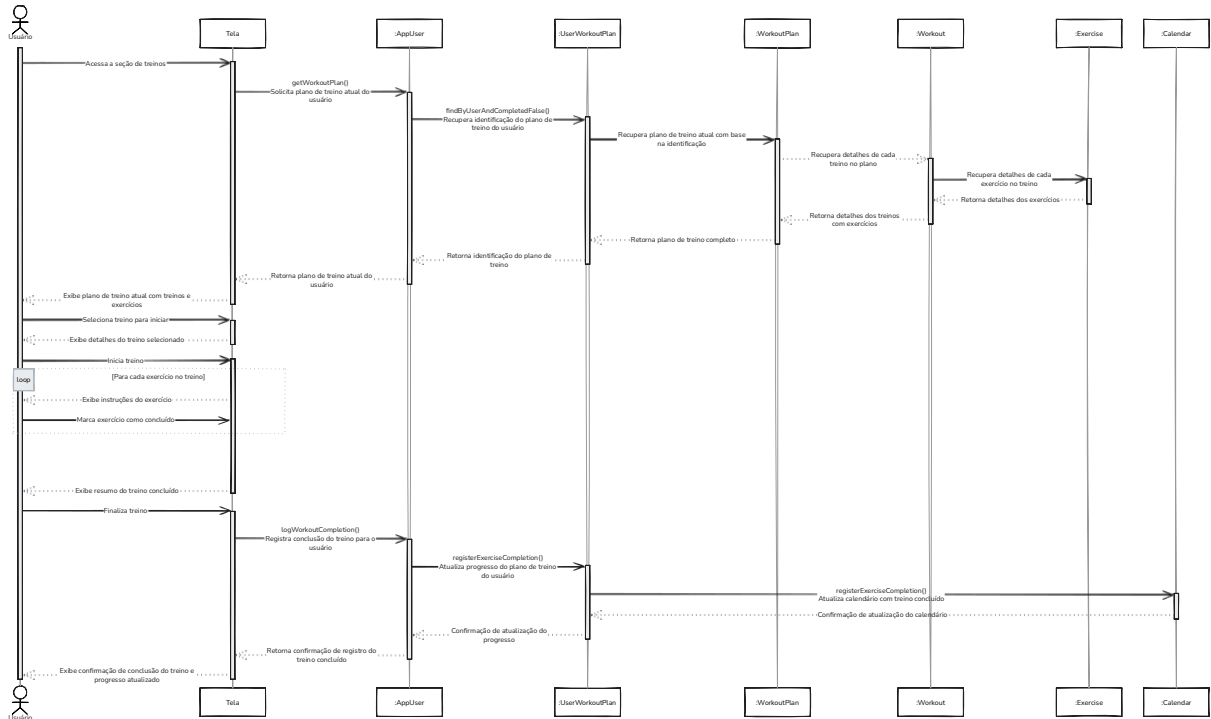
<b>Critério 3</b>
Deve armazenar a preferência do usuário sobre o formato de instrução.
<b>Regra</b>
Quando a opção de tipo de instrução for alterada (de Vídeo para Ilustração e vice-versa) o sistema deve guardar essa informação para exercícios e treinos futuros.

<b>Critério 4</b>
Deve permitir pausar e sair do treino.
<b>Regra</b>
Quando o usuário clicar no botão de pausar ou de sair do treino, o sistema deve responder de acordo.

<b>Critério 5</b>
Deve redirecionar para a tela de avaliação do treino ao concluir o treino.
<b>Regra</b>
Quando um treino for realizado até o fim, o sistema deve automaticamente redirecionar o usuário para a tela de avaliação do treino.

## Diagrama de Sequência 006 – Realizar treino

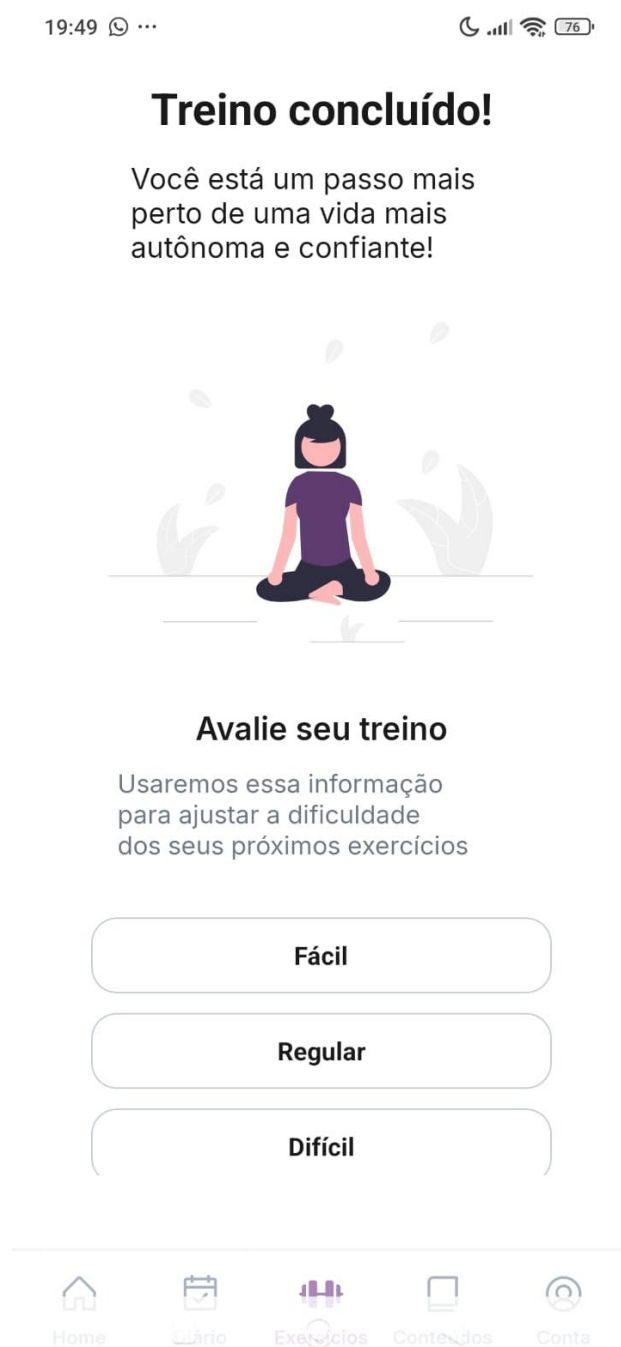
Figura 15 - Diagrama de Sequência - Realizar Treino



## HU 7. HISTÓRIA DE USUÁRIO 007 – AVALIAR TREINO

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Avaliar o treino realizado de acordo com a minha percepção de dificuldade dele
<b>Para</b>	O sistema ajustar minha rotina e fornecer treinos condizentes com a minha condição atual

Figura 16 - Tela de Avaliar Treino

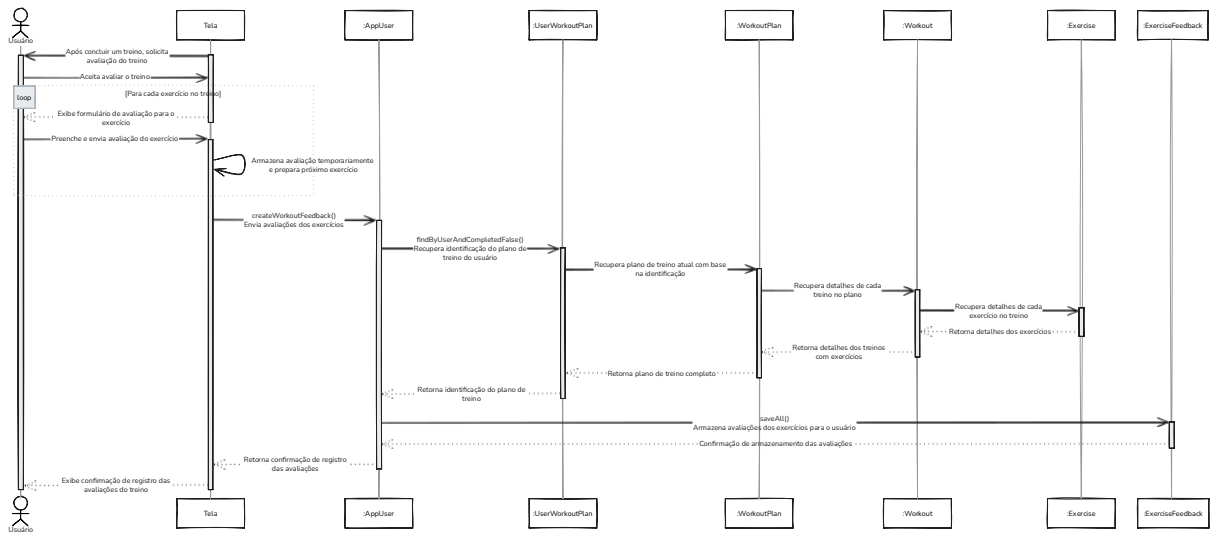


### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir a avaliação do treino realizado.
<b>Regra</b>
Quando o usuário selecionar uma das opções de avaliação de treino (“Fácil”, “Regular” e “Difícil”), o sistema deve armazenar a informação no banco de dados e recalcular a rotina de treinos do usuário. Depois, deve redirecionar para a tela inicial.

### Diagrama de Sequência 007 – Avaliar treino

Figura 17 - Diagrama de Sequência - Avaliar Treino



## HU 8. HISTÓRIA DE USUÁRIO 008 – REPETIR QUESTIONÁRIO

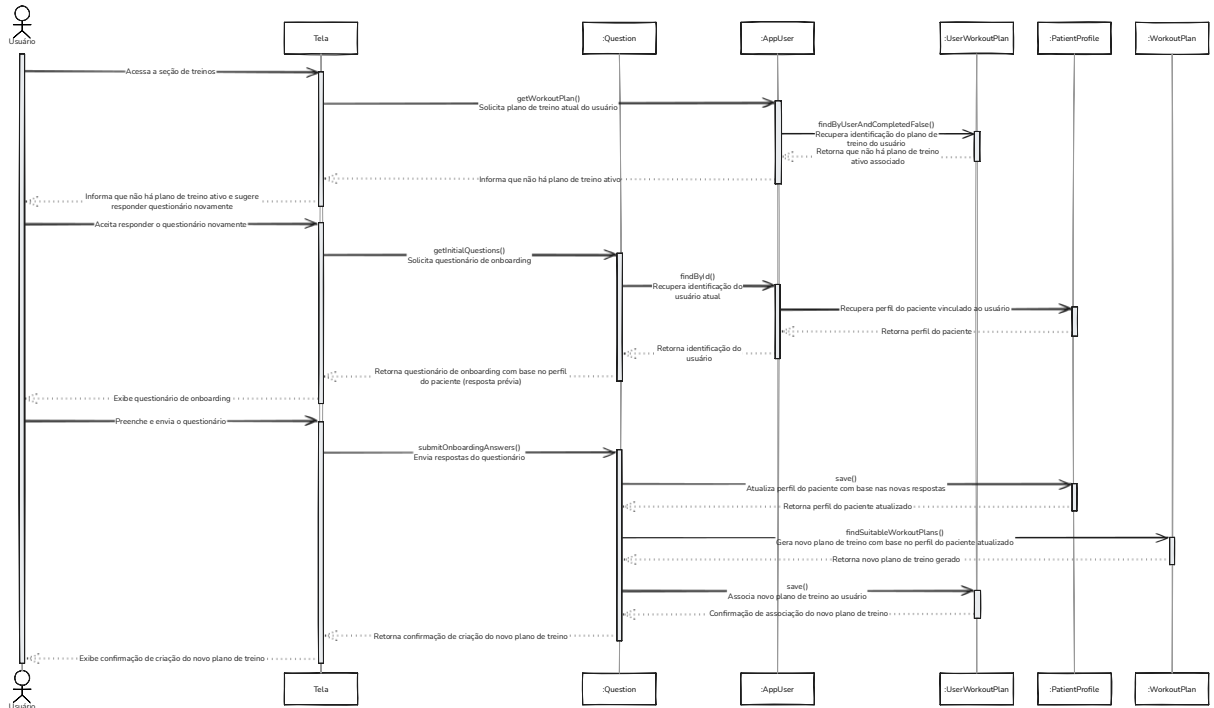
<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Responder periodicamente o formulário
<b>Para</b>	Acompanhar minha evolução e recategorização de treino

### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir que o usuário responda novamente

### Diagrama de Sequência 008 – Repetir questionário

Figura 18 - Diagrama de Sequência - Repetir Questionário



## HU 9. HISTÓRIA DE USUÁRIO 009 – INTERAGIR COM PUBLICAÇÕES

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Interagir com outras publicações
<b>Para</b>	Visualizá-las, comentá-las, curtí-las e/ou salvá-las

Figura 19 - Tela de Interagir com Publicações



### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir curtir e descurtir publicações.
<b>Regra</b>
Quando o usuário clicar no ícone de “curtir”, o sistema deve registrar a interação no banco de dados e alterar o ícone para o estado “curtido”. Se o usuário clicar novamente, o sistema deve remover o registro da curtida e retornar o ícone ao estado padrão.

<b>Critério 2</b>
Deve permitir comentar publicações.
<b>Regra</b>
Quando o usuário clicar em “Comentar”, o sistema deve abrir um campo de texto para inserção do comentário. Ao enviar, o sistema salva o comentário no banco de dados, vinculado à publicação e ao usuário autor.

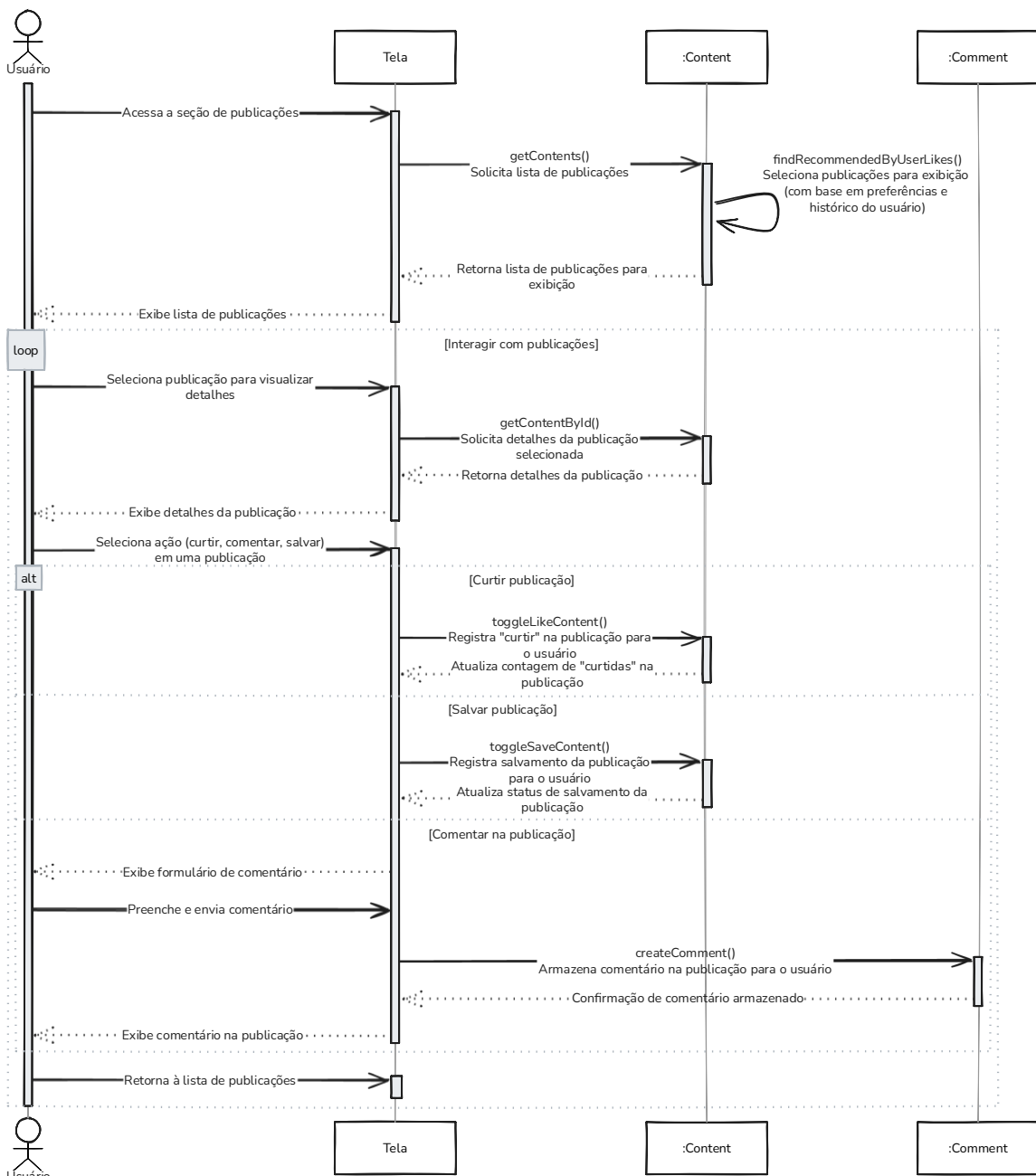
<b>Critério 3</b>
Deve permitir favoritar publicações.
<b>Regra</b>
Ao clicar no ícone de “favorito”, o sistema deve adicionar a publicação à lista de favoritos do usuário. Caso o ícone seja clicado novamente, a publicação deve ser removida da lista.

<b>Critério 4</b>
Deve permitir remover comentários próprios.
<b>Regra</b>
O usuário pode excluir apenas comentários feitos por ele. Ao clicar em “Excluir comentário”, o sistema deve abrir um modal de confirmação. Após a confirmação, o comentário é removido.

<b>Critério 5</b>
Deve permitir responder outros comentários nas publicações.
<b>Regra</b>
Quando o usuário clicar em “Responder”, o sistema deve abrir um campo de texto para inserção da Resposta. Ao enviar, o sistema salva a resposta no banco de dados.

### Diagrama de Sequência 009 – Interagir com publicação

Figura 20 - Diagrama de Sequência - Interagir com Publicação



## HU 10. HISTÓRIA DE USUÁRIO 010 – MANTER PERFIL

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Visualizar e editar meu perfil
<b>Para</b>	Gerenciar minhas informações e interações no sistema

Figura 21 - Tela de Manter Perfil

< Perfil



 Alterar Foto

 Remover

### Informações Pessoais

Nome completo \*

Leonardo

Email \*

usuario2@example.com

Gênero

Masculino

Feminino

Outro

Cancelar

Salvar

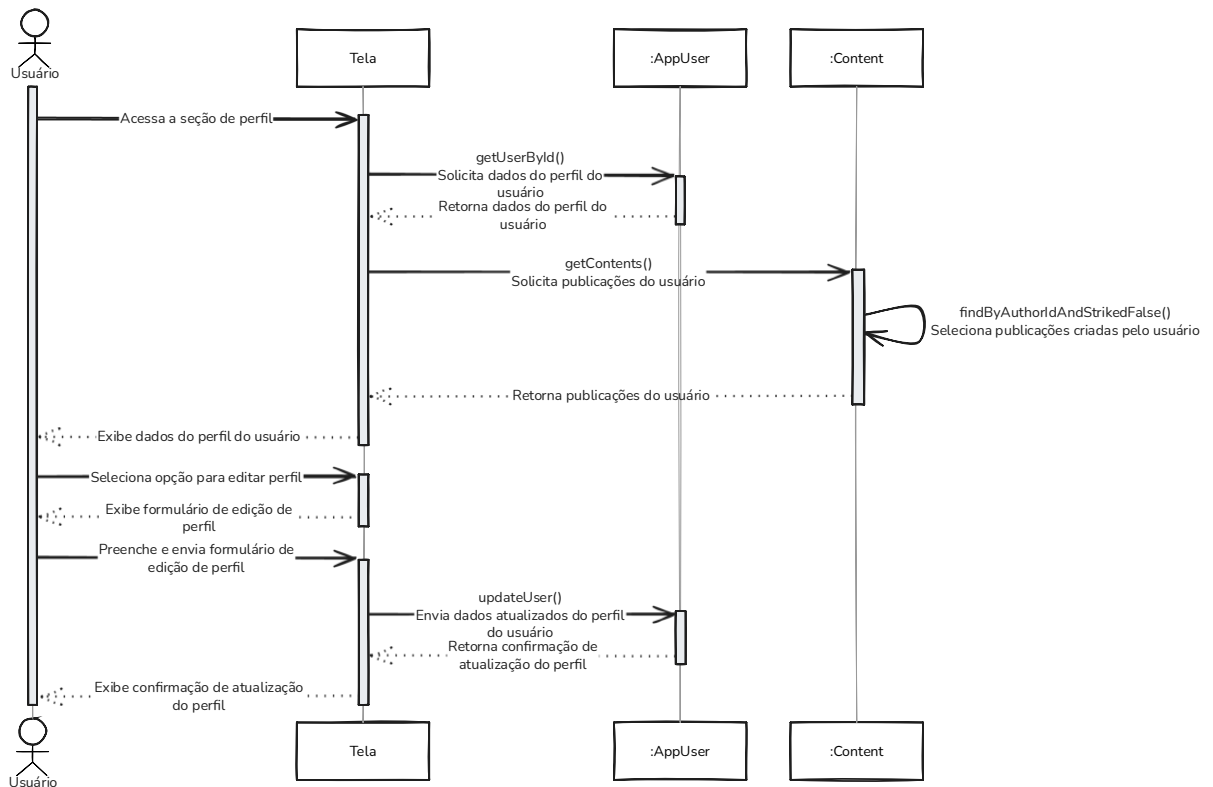
### Critérios de Aceitação

<b>Critério 1</b>
Deve exibir o perfil do usuário logado com informações.
<b>Regra</b>
A tela mostra nome, foto, lista de publicações e salvos.

<b>Critério 2</b>
Deve permitir editar informações pessoais e foto.
<b>Regra</b>
Ao clicar em “Editar Perfil”, o sistema permite alterar foto e texto de biografia.

### Diagrama de Sequência 010 – Manter Perfil

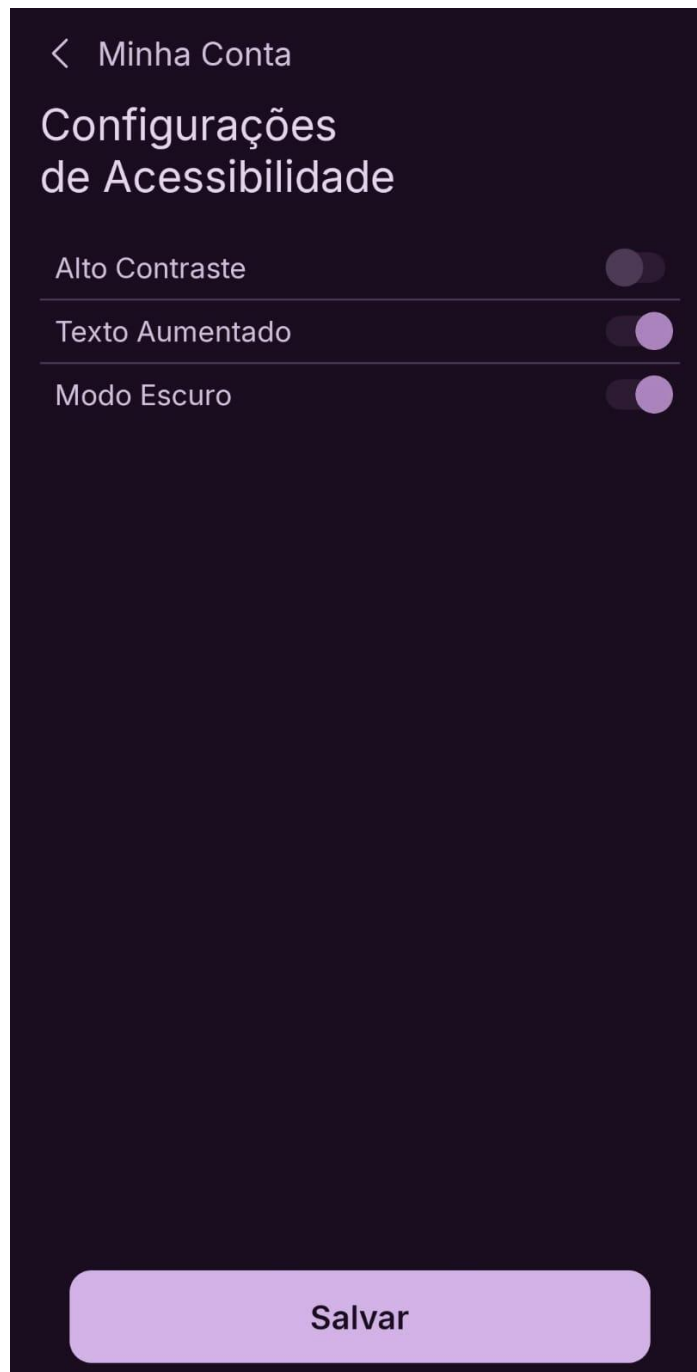
Figura 22 - Diagrama de Sequência - Manter Perfil



## HU 11. HISTÓRIA DE USUÁRIO 011 – DEFINIR ACESSIBILIDADE

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Customizar as opções de acessibilidade, como tamanho ampliado de fontes, alto contraste e modo noturno
<b>Para</b>	Conseguir usar o sistema com autonomia e conforto

Figura 23 - Tela de Definir Acessibilidade

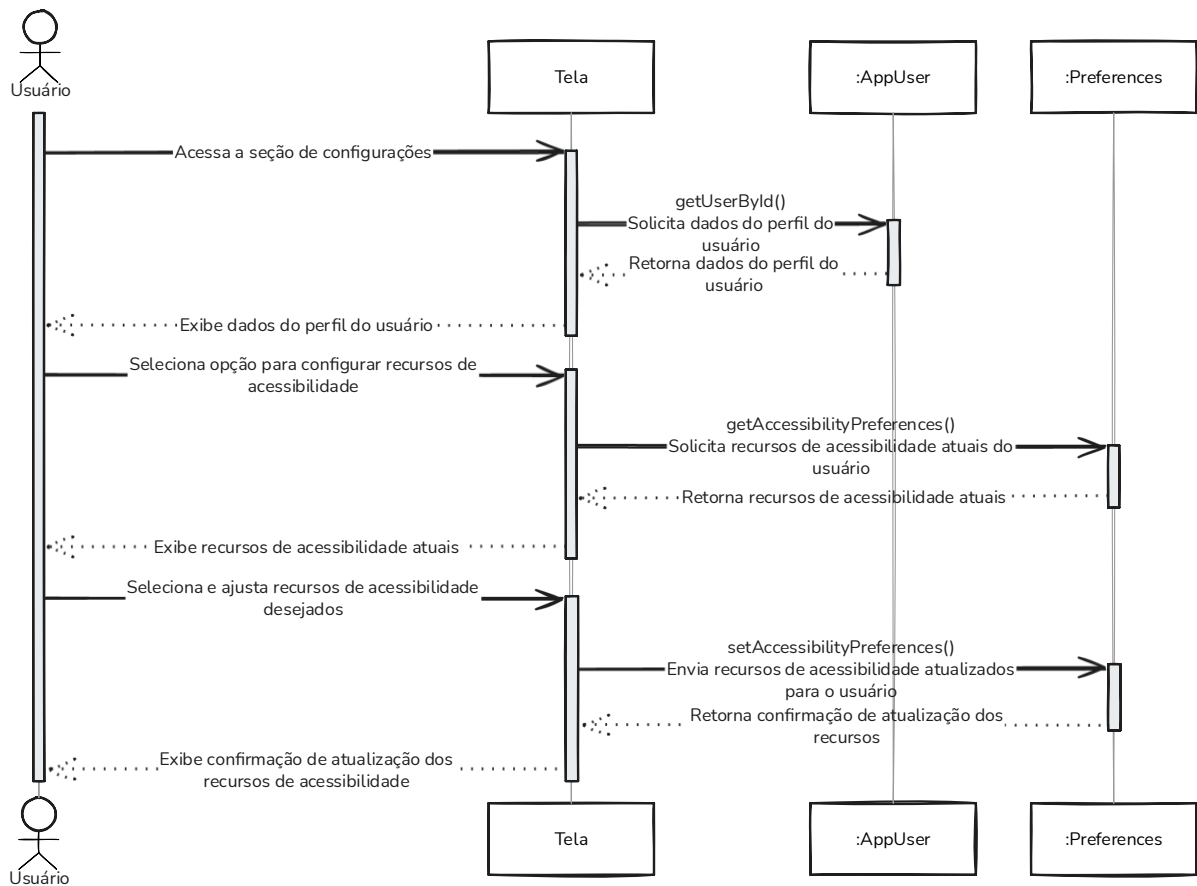


### Critérios de Aceitação

Critério 1
Deve ativar e desativar recursos de acessibilidade.
Regra
Os recursos de acessibilidade Fonte Ampliada, Alto Contraste e Modo Noturno, cada um, um input do tipo toggle. Ao ter o valor desse input alterado, o sistema deve aplicar as alterações necessárias.

### Diagrama de Sequência 011 – Definir Acessibilidade

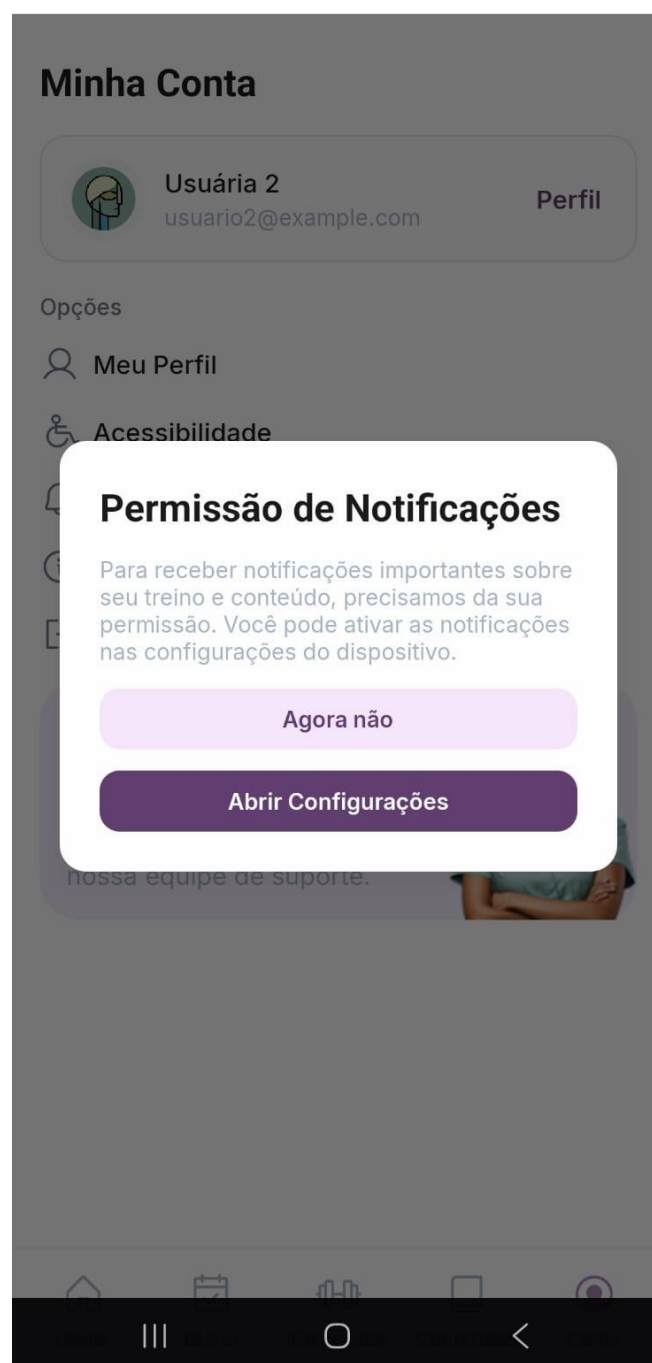
Figura 24 - Diagrama de Sequência - Definir Acessibilidade



## HU 12. HISTÓRIA DE USUÁRIO 012 – GERENCIAR NOTIFICAÇÕES

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Ativar ou desativar as notificações
<b>Para</b>	ter controle sobre os alertas que recebo e personalizar minha experiência no app

Figura 25 - Tela de Gerenciar Notificações

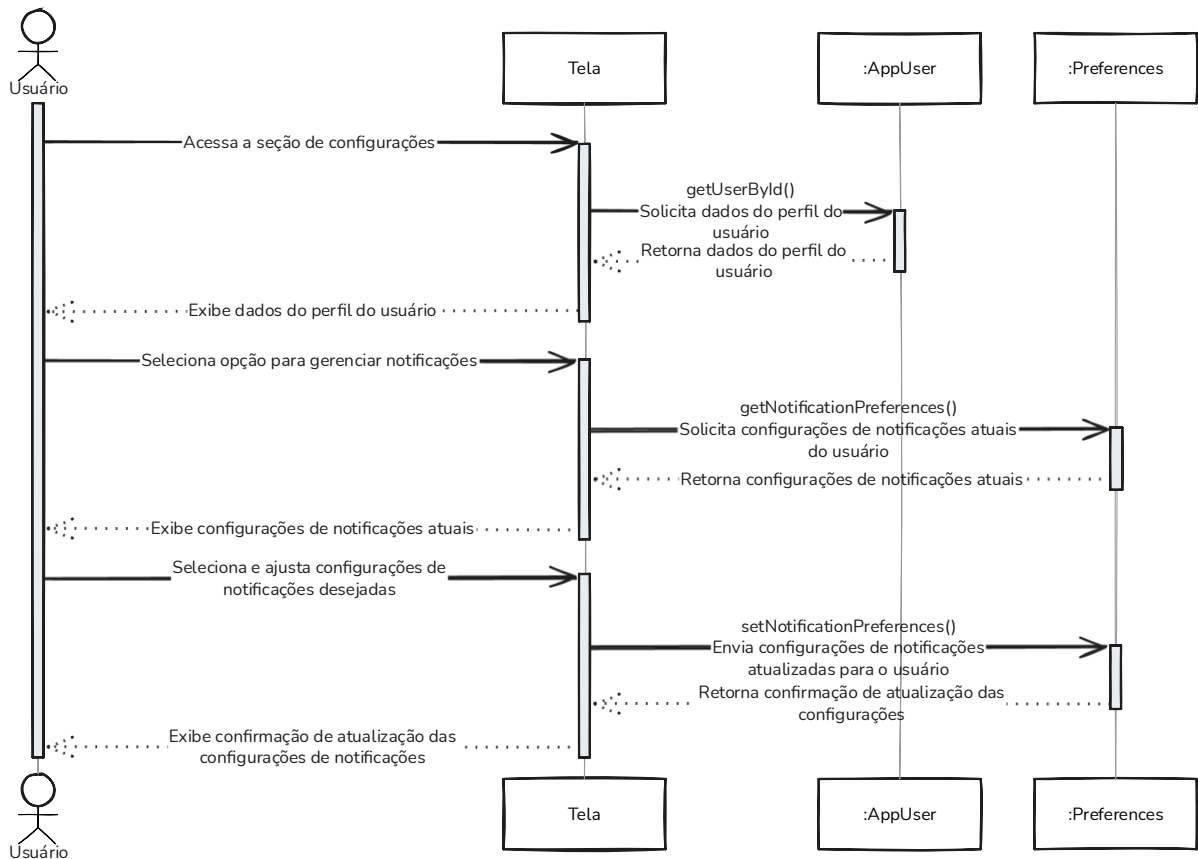


### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir que o usuário ative ou desative notificações e salvar automaticamente sua escolha.
<b>Regra</b>
A alteração feita pelo usuário (ativar ou desativar notificações) deve ser aplicada imediatamente e persistir, mesmo após fechar e reabrir o aplicativo.

### Diagrama de Sequência 012 – Gerenciar Notificações

Figura 26 - Diagrama de Sequência - Gerenciar Notificações



## HU 13. HISTÓRIA DE USUÁRIO 013 – ACESSAR “SOBRE O SISTEMA”

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Visualizar informações do sistema
<b>Para</b>	Saber mais sobre o sistema

Figura 27 - Tela de "Sobre o Sistema"

< Minha Conta

## Sobre o Aplicativo

### DailyIU

O DailyIU é um aplicativo desenvolvido para auxiliar no acompanhamento e tratamento da incontinência urinária. Através de ferramentas como diário miccional, exercícios de fisioterapia pélvica e conteúdo educativo, buscamos proporcionar uma melhor qualidade de vida aos usuários.

### Funcionalidades

- Diário Miccional: Registre e acompanhe seus episódios de incontinência
- Exercícios de Fisioterapia Pélvica: Pratique exercícios guiados para fortalecimento do assoalho pélvico
- Conteúdo Educativo: Acesse informações e dicas sobre incontinência urinária
- Relatórios: Gere relatórios detalhados do seu acompanhamento

### Versão

1.0.0

### Importante

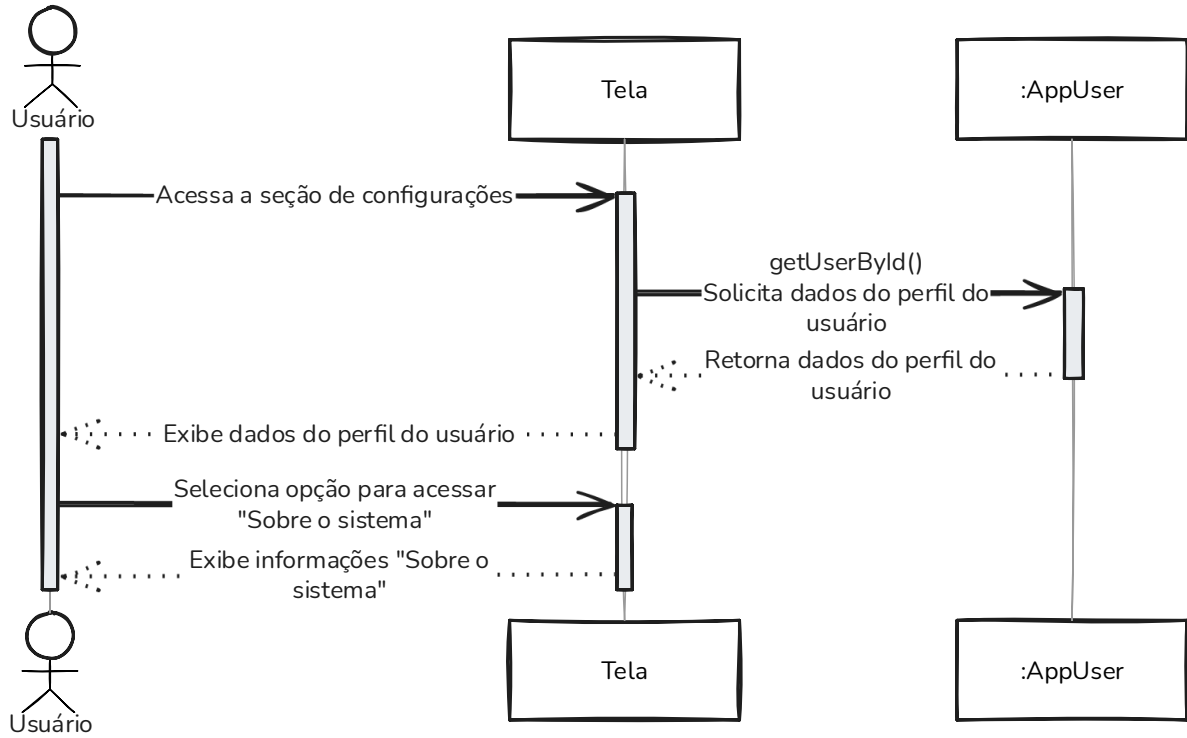
Este aplicativo é uma ferramenta de apoio e não substitui o acompanhamento médico. Sempre consulte um profissional de saúde para diagnóstico e tratamento adequados.



<b>Critério 1</b>
Deve permitir visualizar informações e a versão do sistema.

### Diagrama de Sequência 013 – Acessar “Sobre o Sistema”

Figura 28 - Diagrama de Sequência - Acessar "Sobre o Sistema"



## HU 14. HISTÓRIA DE USUÁRIO 014 – ACESSAR “FALE CONOSCO”

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Entrar em contato com a equipe de suporte
<b>Para</b>	Relatar bugs, pedir orientações específicas e enviar sugestões de melhorias

Figura 29 - Tela "Fale Conosco"



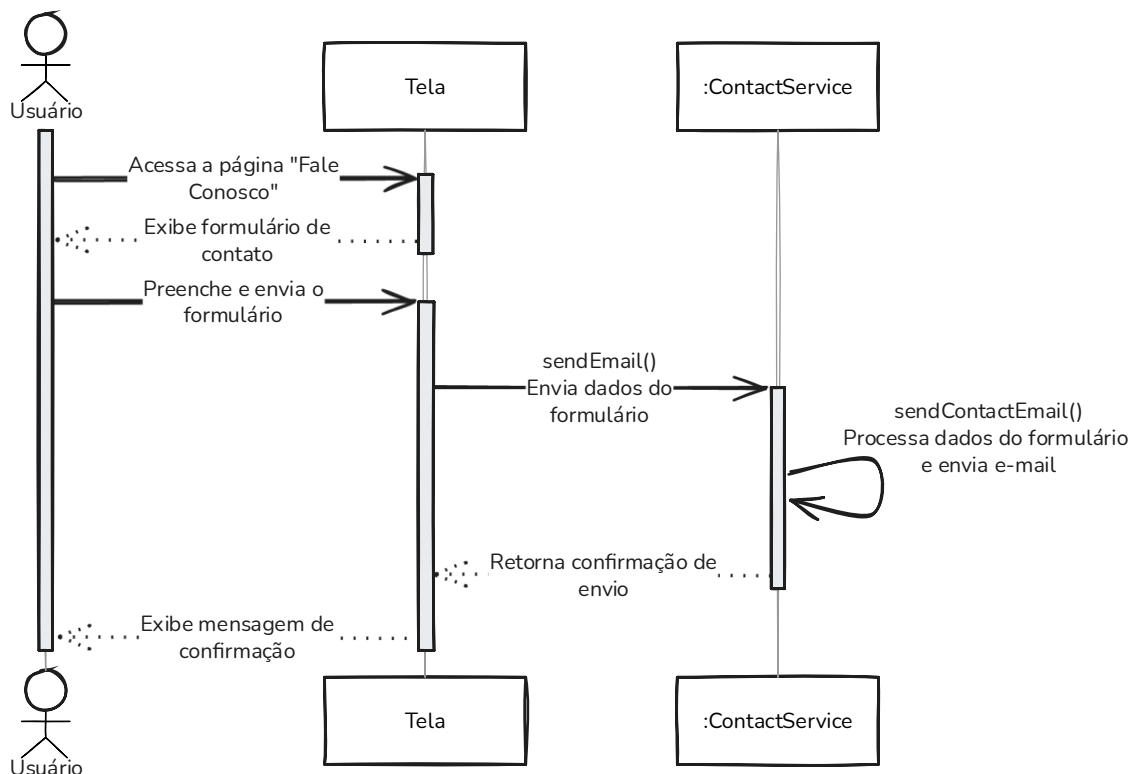
### Critérios de Aceitação

<b>Critério 1</b>
Deve enviar e-mails para o administrador.
<b>Regra</b>
Quando o usuário preencher o formulário de contato e clicar no botão “Enviar”, o sistema deve enviar um e-mail para o administrador com o assunto e mensagem informados. O usuário também deve receber um e-mail com o assunto e mensagem para ter registrado.

<b>Critério 2</b>
Não deve enviar e-mails com campos obrigatórios em branco.
<b>Regra</b>
Quando o usuário clicar no botão “Enviar” e algum dos campos obrigatórios (Assunto e Mensagem) estiverem vazios, o sistema deve exibir uma mensagem de erro logo abaixo do campo em questão

### Diagrama de Sequência 014 – Acessar “Fale Conosco”

Figura 30 - Diagrama de Sequência - Acessar "Fale Conosco"



## HU 15. HISTÓRIA DE USUÁRIO 015 – RECUPERAR SENHA

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Solicitar recuperação de senha
<b>Para</b>	Recuperar meu acesso ao sistema

Figura 31 - Tela de Recuperação de Senha



**DailyIU**

## Esqueci minha senha

E-mail

Código de verificação

Nova senha

Confirmar senha

**Salvar**

**Voltar**



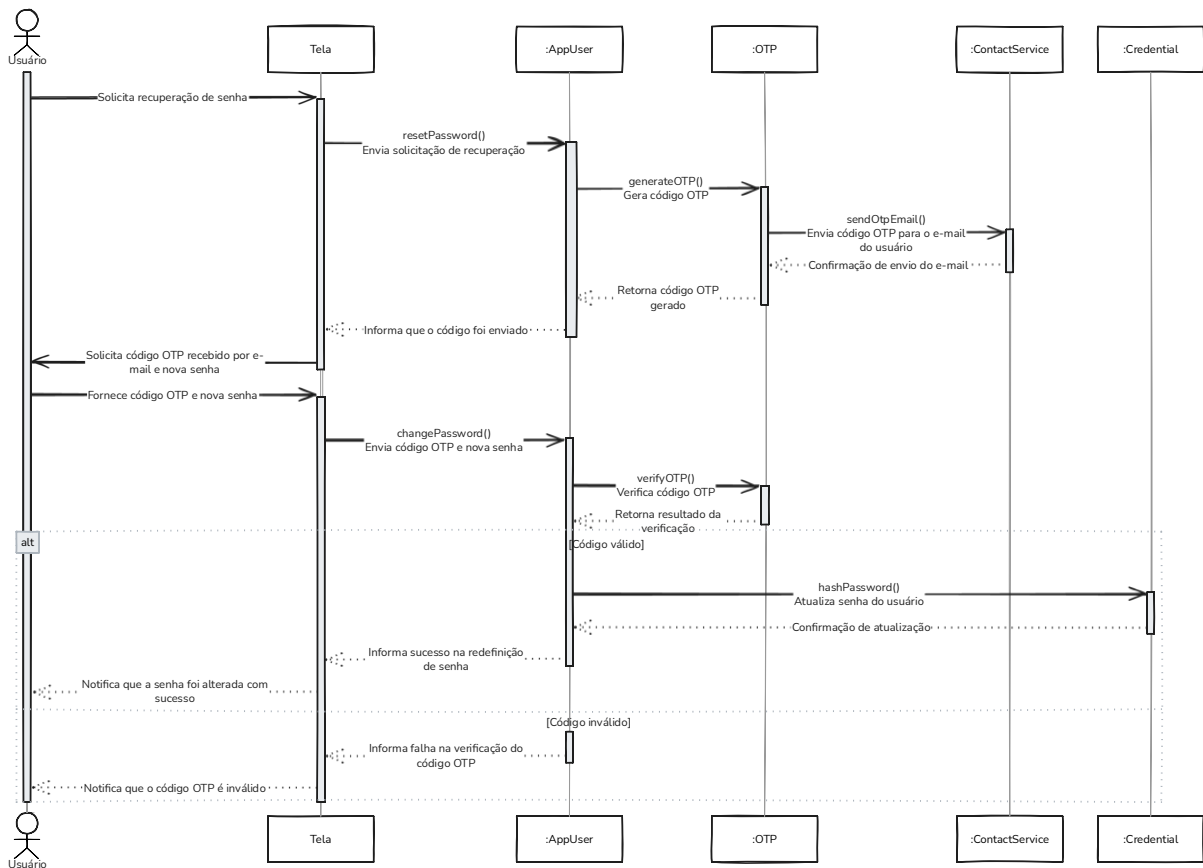
### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir ao usuário recuperar a senha por e-mail.
<b>Regra</b>
Quando o usuário clicar em “Recuperar senha”, é enviado um e-mail com código para redefinição.

<b>Critério 2</b>
Deve permitir redefinir a senha.
<b>Regra</b>
Ao acessar inserir o código enviado, o usuário informa nova senha e confirmação, e o sistema atualiza o banco de dados.

### Diagrama de Sequência 015 – Recuperar Senha

Figura 32 - Diagrama de Sequência - Recuperar Senha



## HU 16. HISTÓRIA DE USUÁRIO 016 – REALIZAR POSTAGEM

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Criar e publicar uma nova postagem
<b>Para</b>	Compartilhar conteúdo educativo, relatos ou opiniões com os demais usuários

Figura 33 - Tela de Realizar Postagem



### Critérios de Aceitação

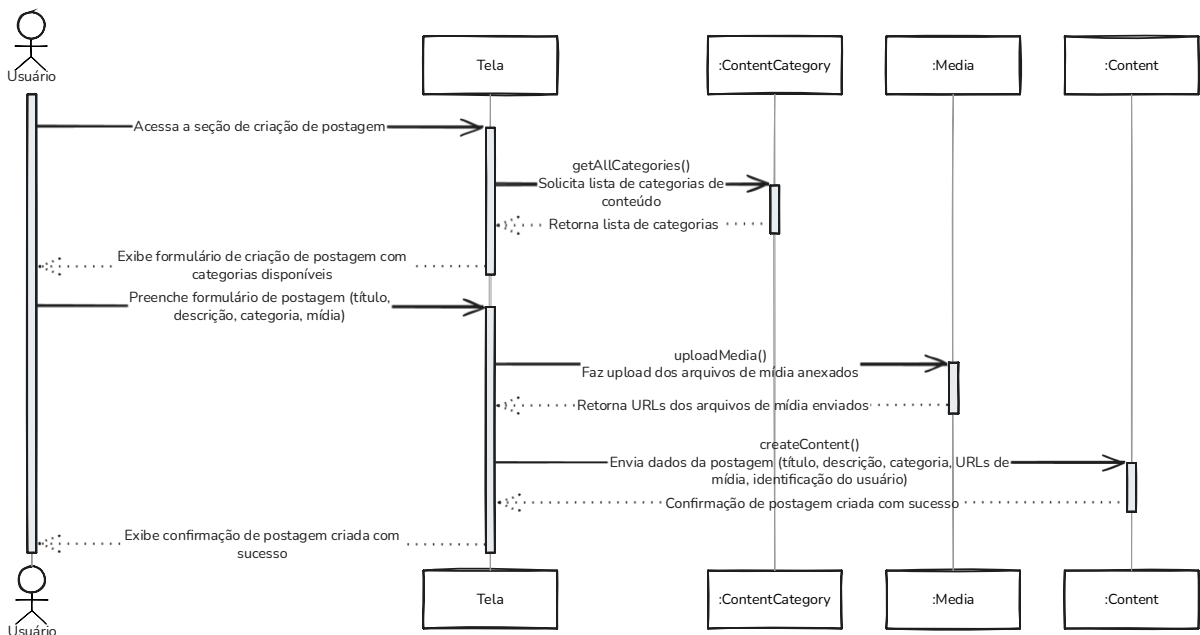
<b>Critério 1</b>
Deve permitir postar texto e/ou mídia.
<b>Regra</b>
Se o usuário tentar publicar sem texto e sem mídia, o sistema deve bloquear e exibir aviso.

<b>Critério 2</b>
Deve confirmar publicação.
<b>Regra</b>
Após salvar a postagem, o sistema deve exibir uma mensagem indicando sucesso.

<b>Critério 3</b>
Não deve aceitar mídia inválida.
<b>Regra</b>
Arquivos fora do formato permitido ou acima do tamanho máximo devem ser rejeitados.

### Diagrama de Sequência 016 – Realizar postagem

Figura 34 - Diagrama de Sequência - Realizar Postagem



## HU 17. HISTÓRIA DE USUÁRIO 017 – DENUNCIAR POSTAGEM

<b>Sendo</b>	Um usuário do sistema
<b>Quero</b>	Denunciar publicações
<b>Para</b>	Avisar a administração sobre publicações que fogem ao escopo do aplicativo ou ferem as diretrizes da plataforma

Figura 35 - Tela de Denunciar Postagem



### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir denunciar uma publicação ao clicar no botão “Denunciar Postagem”.
<b>Regra</b>
Após clicar no botão “Denunciar Postagem”, deve abrir um modal para que o usuário selecione o motivo dentre as opções Spam (conteúdo repetitivo ou irrelevante), Ofensivo (insultos, ódio ou assédio), Falso (informação enganosa ou fake), Inadequado (nudez, violência ou ilegal), Outro.

<b>Critério 2</b>
Deve habilitar o botão de “Confirmar Denúncia” somente após o usuário escolher um motivo.
<b>Regra</b>
Enquanto o usuário não selecionar um motivo, o botão permanece desabilitado.

<b>Critério 3</b>
Deve permitir o usuário cancelar a denúncia.
<b>Regra</b>
Após o usuário clicar em “Cancelar”, o modal fecha sem realizar nenhuma denúncia

<b>Critério 4</b>
Deve enviar a denúncia à página de “Gerenciamento de Denúncias” do administrador do sistema.
<b>Regra</b>
Quando o usuário confirmar a denúncia, a postagem e o motivo devem aparecer na página de gerenciamento de denúncias para avaliação do administrador.

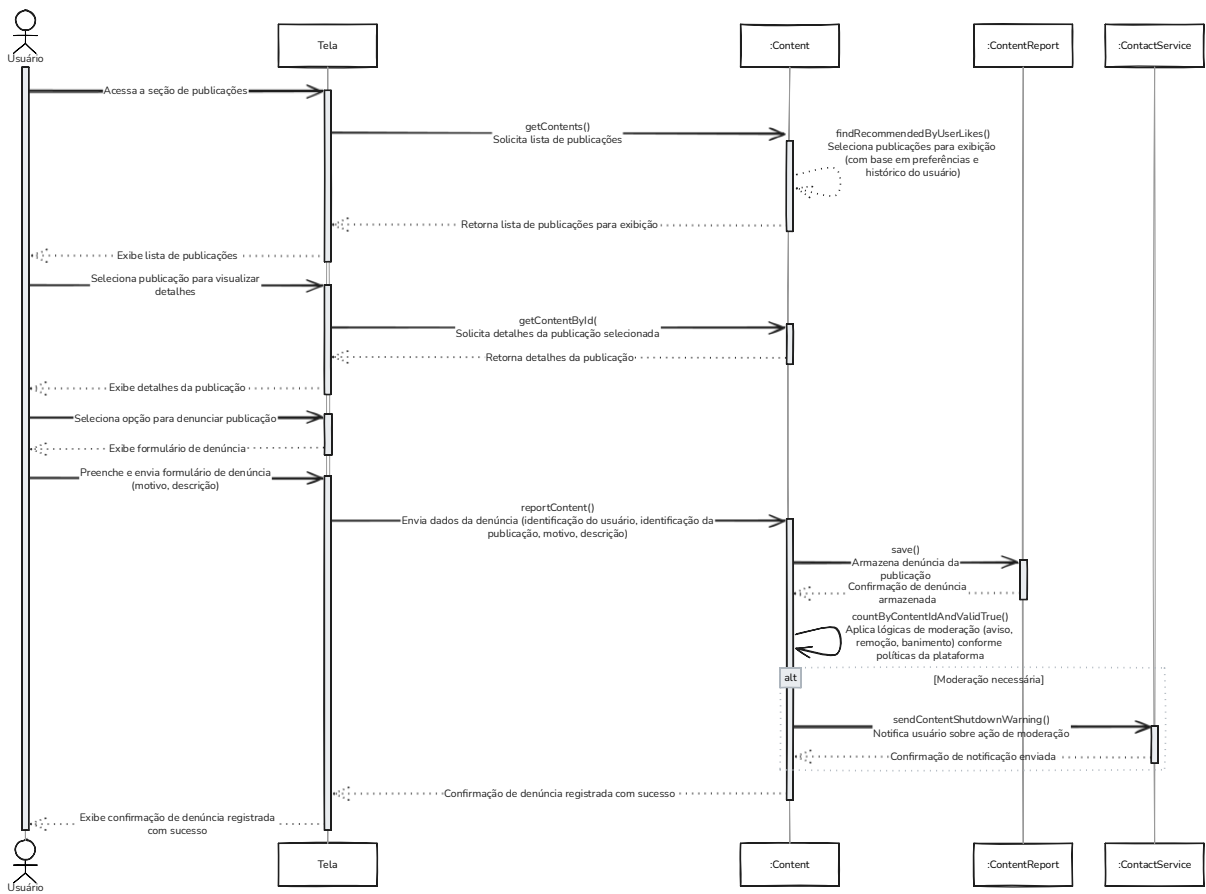
<b>Critério 5</b>
Deve ocultar a publicação dos outros usuários após 5 denúncias.
<b>Regra</b>

Quando 5 usuários diferentes denunciarem a mesma publicação, ela é ocultada dos demais usuários até avaliação dos administradores.

**Critério 6**  
 Deve enviar e-mail ao autor caso sua publicação seja ocultada.

**Diagrama de Sequência 017 – Denunciar postagem**

Figura 36 - Diagrama de Sequência - Denunciar Postagem



## HU 18. HISTÓRIA DE USUÁRIO 018 – MANTER USUÁRIOS

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Realizar operações de alteração e exclusão de usuários
<b>Para</b>	Gerenciar os usuários da aplicação

Figura 37 - Tela de Manter Usuários

The screenshot shows the 'Administração de Usuários' page. At the top, there is a navigation bar with 'Início', 'Diário', 'Exercícios', 'Conteúdos', 'Contato', and 'Administração'. The user is logged in as 'Olá, Usuário!' with a profile icon. The main heading is 'Administração de Usuários' with the subtitle 'Gerencie os usuários do DailyU'. Below this, there is a search bar labeled 'Pesquisar usuários' with a placeholder 'Nome ou e-mail'. To the right of the search bar are filters for 'Perfil' (set to 'Todos') and 'Status' (set to 'Todos'), and a 'Resultados por página' dropdown set to '8'. A 'Limpar filtros' button is located below the search bar. Below the filters is a pagination control showing 'Anterior', '1', '2', and 'Próximo'. The main content is a table with the following columns: 'Nome', 'E-mail', 'Perfil', 'Status', and 'Editar'. The table contains three rows of user data:

Nome	E-mail	Perfil	Status	Editar
Usuário 1	usuario1@example.com	Administrador	Bloqueado	Perfil Desbloquear
Usuária 2	usuario2@example.com	Usuário comum	Ativo	Perfil Bloquear
Amanda	amanda@saudavel.com	Profissional	Ativo	Perfil Bloquear

### Critérios de Aceitação

<b>Critério 1</b>
Deve permitir promover um usuário a administrador.
<b>Regra</b>
Ao clicar em “Tornar Admin”, o sistema atualiza o perfil e concede acesso administrativo.

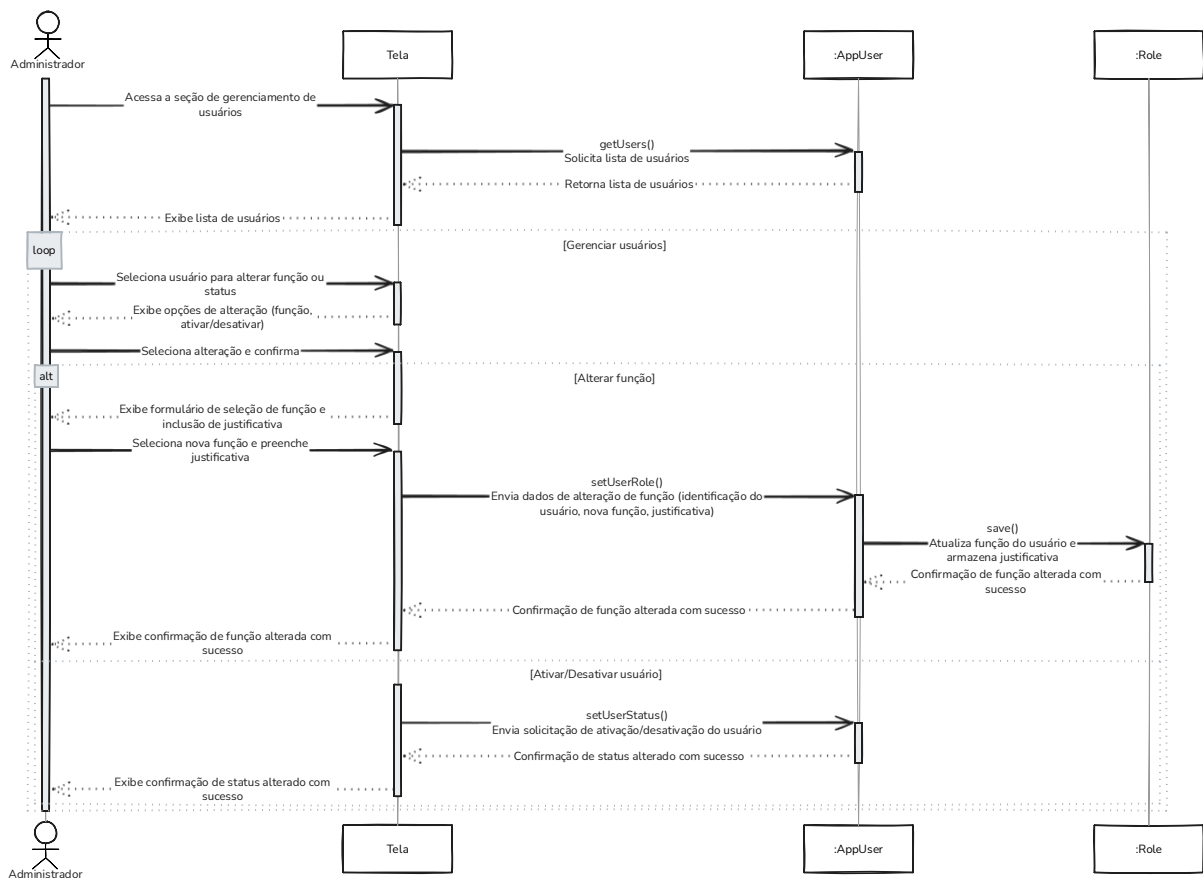
<b>Critério 2</b>
Deve permitir promover o perfil do usuário para profissional da saúde.
<b>Regra</b>
Ao clicar em “Tornar Profissional da Saúde”, o sistema atualiza o perfil e concede acesso de profissional da saúde.

<b>Critério 3</b>
Deve permitir bloquear ou desbloquear o acesso de um usuário.
<b>Regra</b>
Ao clicar em “Bloquear/Desbloquear”, o sistema remove ou concede o acesso ao perfil do usuário.

<b>Critério 4</b>
Deve permitir voltar o perfil para o de usuário convencional.
<b>Regra</b>
Quando o usuário tiver perfil de Admin ou de Profissional da Saúde, o sistema permite retornar para perfil de usuário convencional.

### Diagrama de Sequência 018 – Manter usuários

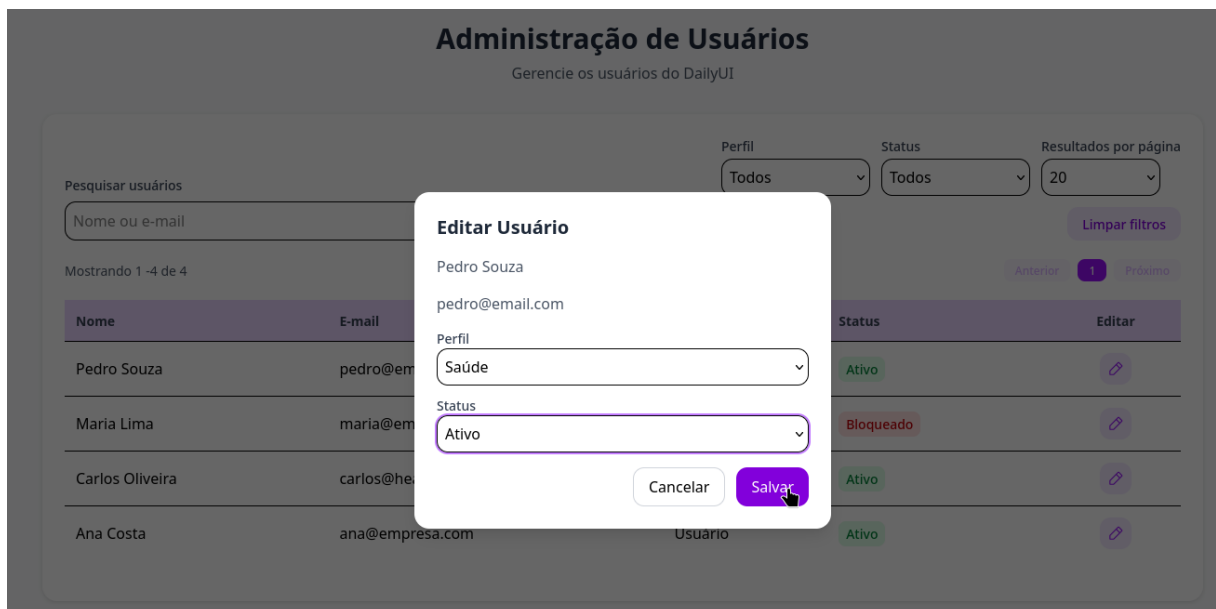
Figura 38 - Diagrama de Sequência - Manter Usuários



HU 19. HISTÓRIA DE USUÁRIO 019 – CONCEDER PERFIL DE  
“PROFISSIONAL DA SAÚDE”

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Conceder o perfil de “Profissional da Saúde” aos usuários que tiveram sua solicitação aprovada
<b>Para</b>	Atribuir o selo de verificação e liberar o acesso aos recursos exclusivos desse tipo de perfil

Figura 39 - Tela de Conceder Perfil de "Profissional da Saúde"



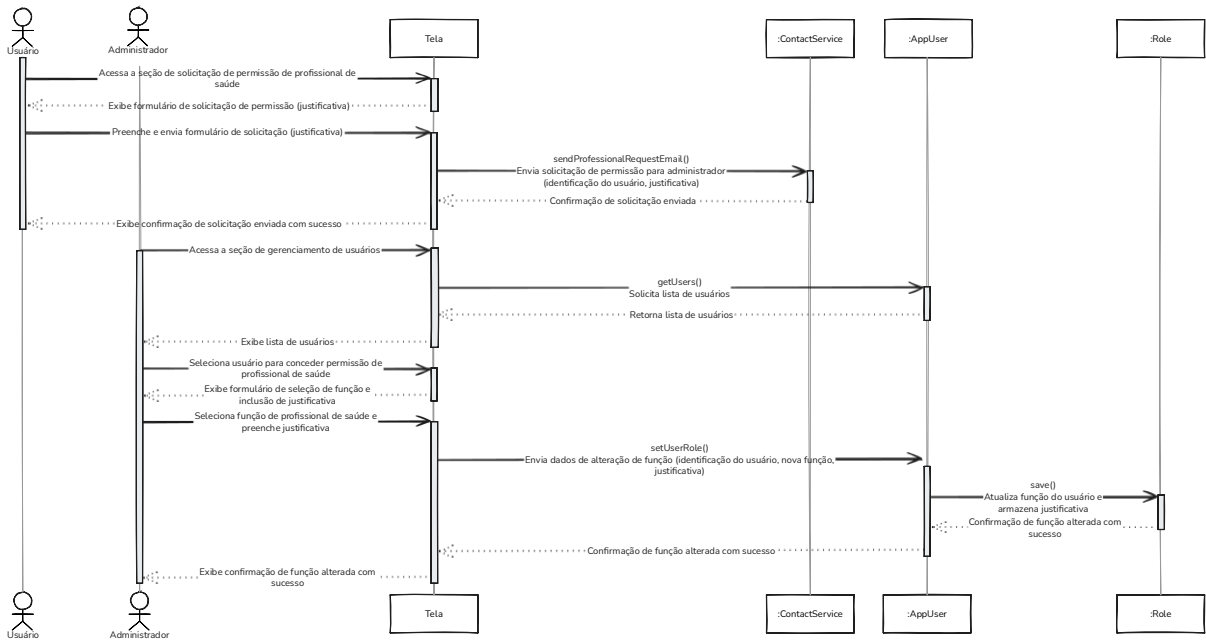
### Critérios de Aceitação

<b>Critério 1</b>
Deve atribuir um selo de “Profissional da Saúde Verificado” aos usuários aprovados.
<b>Regra</b>
Quando o administrador aprovar a solicitação de perfil de profissional da saúde, o sistema deve registrar a aprovação e exibir um selo de verificação visível para todos os usuários, identificando o profissional como verificado.

<b>Critério 2</b>
Deve conceder acesso a recursos exclusivos de profissionais da saúde.
<b>Regra</b>
Quando a solicitação for aprovada, o sistema deve liberar automaticamente o acesso à TAGs de publicações exclusivas de profissionais da saúde.

**Diagrama de Sequência 019 – Conceder perfil de “Profissional da Saúde”**

Figura 40 - Diagrama de Sequência - Conceder Perfil de "Profissional da Saúde"



## HU 20. HISTÓRIA DE USUÁRIO 020 – GERENCIAR DENÚNCIAS

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Gerenciar as publicações denunciadas
<b>Para</b>	Ocultá-las ou mantê-las na linha de tempo das publicações

Figura 41 - Tela de Gerenciar Denúncias



### Critérios de Aceitação

<b>Critério 1</b>
Deve apresentar todas as publicações denunciadas em ordem cronológica de denúncia.
<b>Regra</b>
A lista deve conter todas as publicações denunciadas, ordenadas por status (“Não avaliado”, “Avaliado e permitido”, “Avaliado e oculto”) e data de denúncia decrescente. A publicação denunciada só aparece na lista após 5 denúncias.

<b>Critério 2</b>
Deve permitir alterar o estado da denúncia.
<b>Regra</b>
Ao selecionar uma publicação denunciada, o administrador pode marcá-la como “Avaliado e permitido” ou “Avaliado e oculto”.

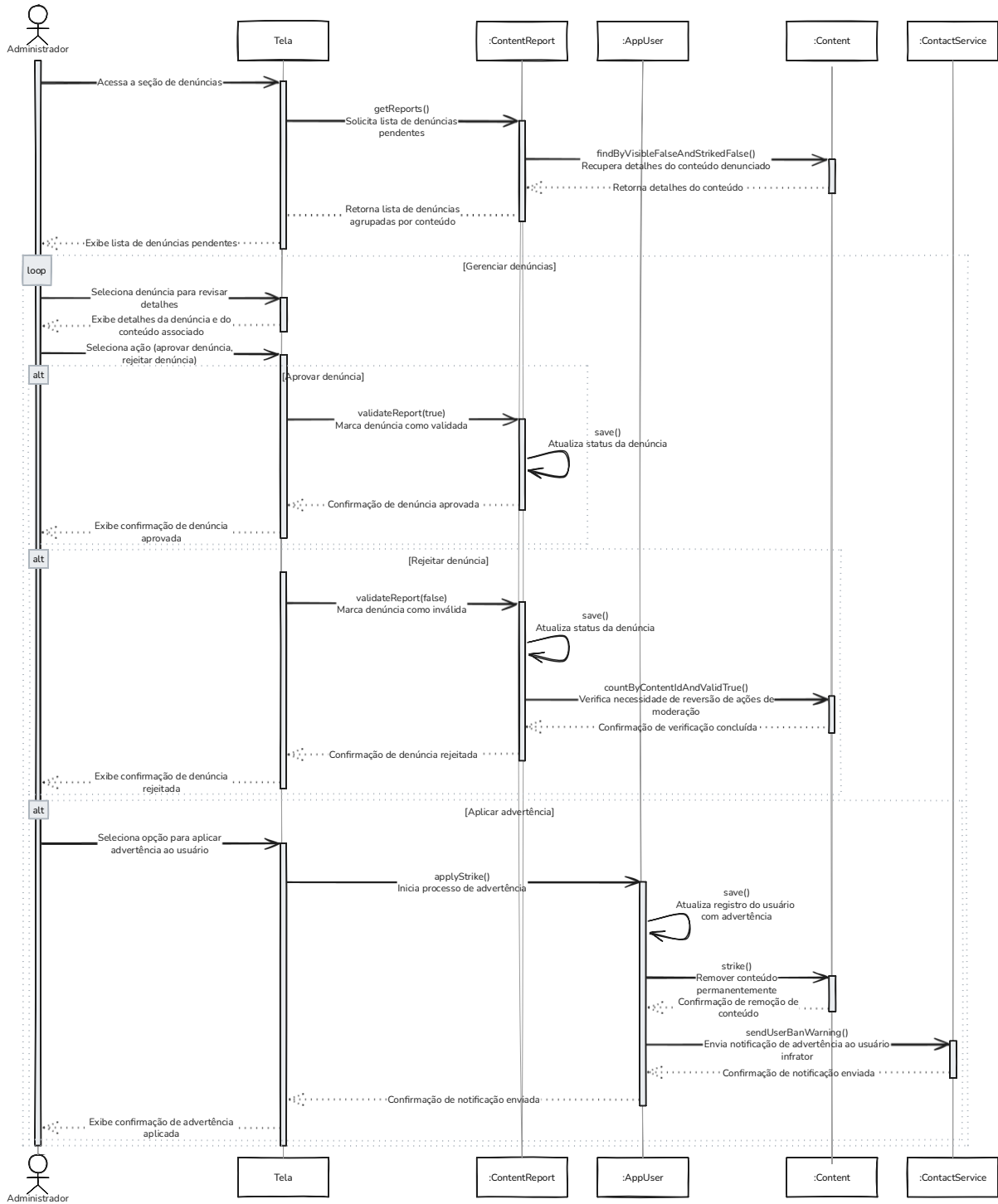
<b>Critério 3</b>
Deve ocultar a publicação quando o administrador selecionar “Ocultar”.
<b>Regra</b>
A publicação é removida da linha do tempo dos usuários e permanece apenas no painel do administrador.

<b>Critério 4</b>
Deve permitir pesquisar e filtrar denúncias.
<b>Regra</b>
O administrador pode pesquisar denúncias por título da publicação ou filtrar por data de denúncia.

<b>Critério 5</b>
Deve bloquear o perfil automaticamente quando ele tiver 3 publicações ocultadas
<b>Regra</b>
Quando o usuário tiver 3 publicações diferentes ocultadas por conta das denúncias, seu perfil tem o acesso bloqueado

## Diagrama de Sequência 020 – Gerenciar denúncias

Figura 42 - Diagrama de Sequência - Gerenciar Denúncias



## HU 21. HISTÓRIA DE USUÁRIO 021 – MANTER EXERCÍCIOS

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Realizar operações de inclusão, alteração e exclusão de imagens e vídeos de exercícios
<b>Para</b>	Manter atualizada a base de dados de exercícios ofertados

Figura 43 - Tela de Manter Exercícios

**Painel de Exercícios**  
Gerencie o catálogo de exercícios, categorias e atributos disponíveis para os planos de treino.

[Novo exercício](#)

Pesquisar exercícios:

Mostrando 1 - 7 de 7

Filtros: Categoria: Todas, Atributo/benefício: Todos, Resultados por página: 8

Título	Categoria	Repetições	Séries	Descanso (s)	Duração (s)	Ações
Exercícios de Kegel Básicos	Kegel	10	3	10	80	<a href="#">Editar</a> <a href="#">Excluir</a>
Exercícios de Ponte	Kegel	10	3	15	90	<a href="#">Editar</a> <a href="#">Excluir</a>

**Critérios de Aceitação****Critério 1**

Não deve permitir o cadastro de exercícios se algum campo obrigatório estiver vazio.

**Regra**

Quando for pressionado o botão “Adicionar” e houver um campo obrigatório vazio, deve ser exibido o erro “Este campo é obrigatório” logo abaixo do campo.

**Critério 2**

Deve cadastrar exercícios.

**Regra**

Quando todos os campos obrigatórios estiverem preenchidos corretamente, o sistema deve salvar no banco de dados o novo exercício.

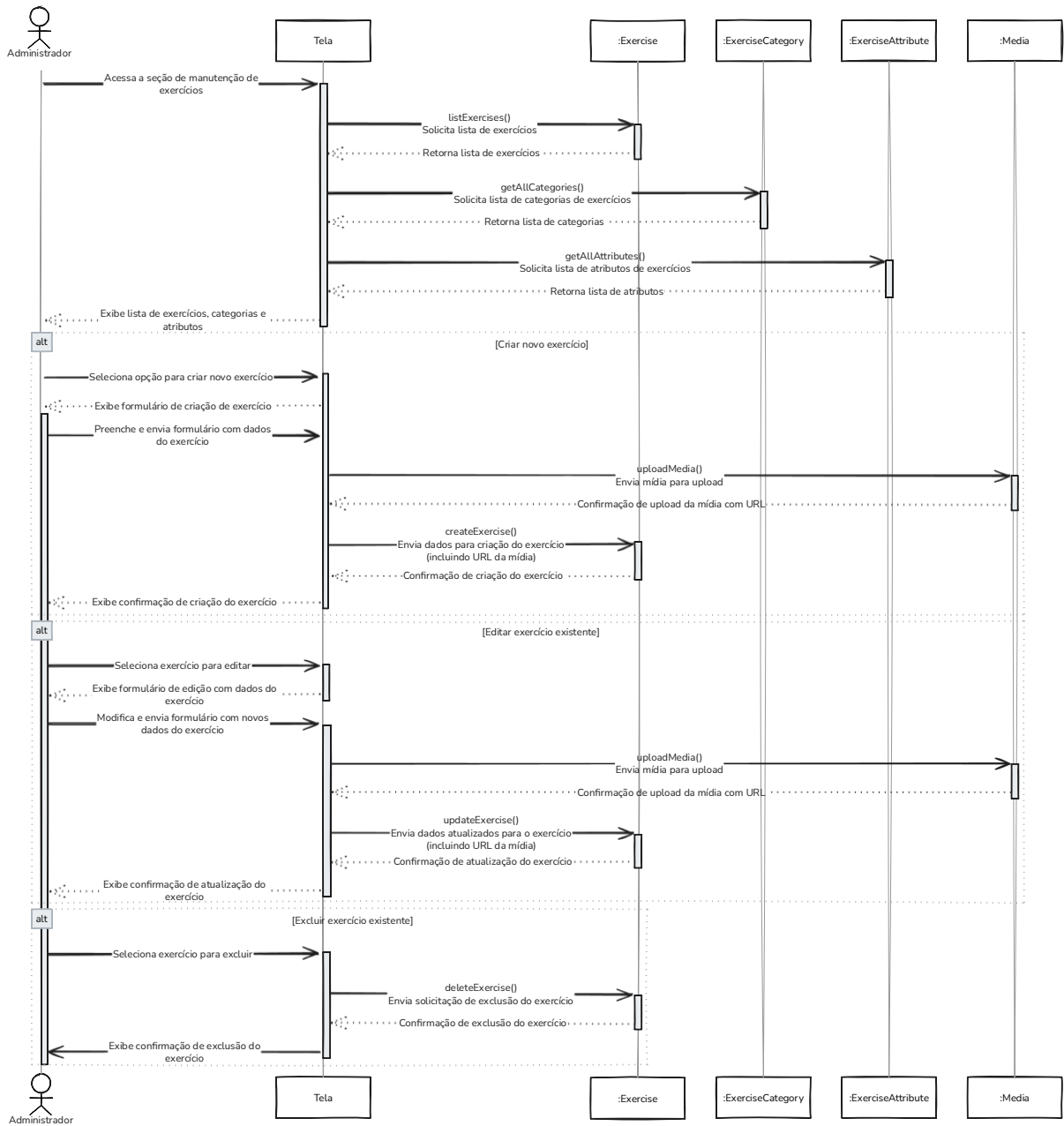
<b>Critério 3</b>
Deve realizar a alteração de exercícios.
<b>Regra</b>
Quando a tela for carregada em modo de alteração, todos os campos devem estar preenchidos com as informações do banco de dados. O campo “Visível” deve ficar disponível para alteração; Após a confirmação, salvar o registro com as alterações feitas, exibir a mensagem “Vídeo alterado com sucesso” e retornar à tela anterior.

<b>Critério 4</b>
Deve excluir o exercício.
<b>Regra</b>
Quando o botão de exclusão for clicado, um modal deve aparecer com os dados do exercício e perguntar se o usuário tem certeza da exclusão. Havendo confirmação, o vídeo deve ser excluído do banco de dados.

<b>Critério 5</b>
Deve retornar à tela inicial.
<b>Regra</b>
Quando for pressionado o botão de voltar, deve retornar à tela inicial.

## Diagrama de Sequência 021 – Manter exercícios

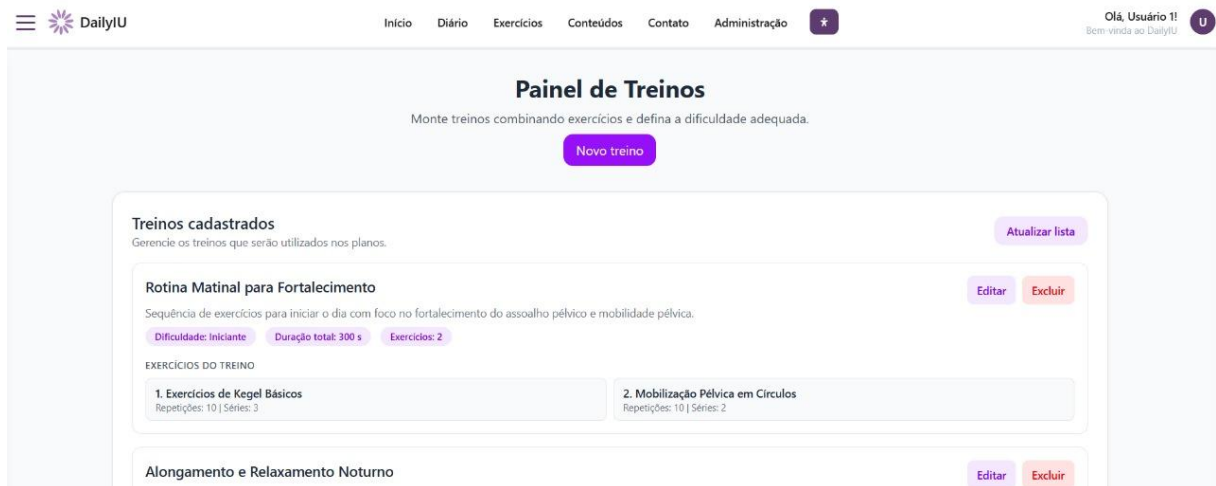
Figura 44 - Diagrama de Sequência - Manter Exercícios



## HU 22. HISTÓRIA DE USUÁRIO 022 – MANTER TREINO

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Realizar operações de inclusão, alteração e exclusão de treinos
<b>Para</b>	Manter atualizada a base de dados de treinos ofertados

Figura 45 - Tela de Manter Treino



### Critérios de Aceitação

#### Critério 1

Deve cadastrar treinos, agrupando exercícios.

#### Critério 2

Deve realizar a alteração de treinos.

#### Regra

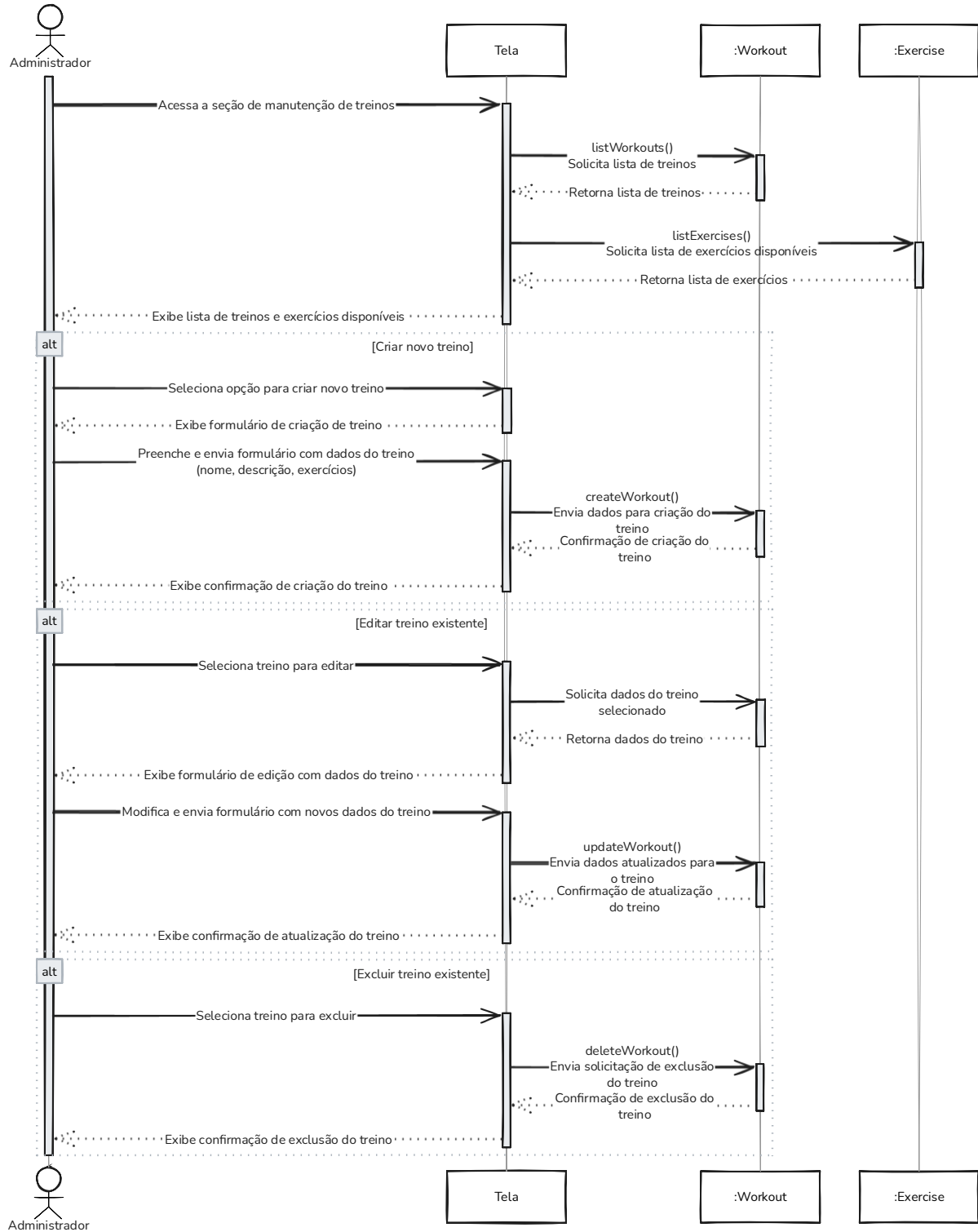
Quando a tela for carregada em modo de alteração, todos os exercícios devem estar preenchidos com as informações do banco de dados. Os exercícios devem ficar disponível para alteração. Após a confirmação, salvar o registro com as alterações feitas, exibir a mensagem “treino alterado com sucesso” e retornar à tela anterior.

<b>Critério 4</b>
Deve excluir o treino.
<b>Regra</b>
Quando o botão de exclusão for clicado, um modal deve aparecer com os dados do treino e perguntar se o usuário tem certeza da exclusão. Havendo confirmação, o treino deve ser excluído do banco de dados.

<b>Critério 5</b>
Deve retornar à tela inicial.
<b>Regra</b>
Quando for pressionado o botão de voltar, deve retornar à tela inicial.

## Diagrama de Sequência 022 – Manter treino

Figura 46 - Diagrama de Sequência - Manter Treino



## HU 23. HISTÓRIA DE USUÁRIO 023 – MANTER PLANO DE TREINO

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Realizar operações de inclusão, alteração e exclusão de planos de treinos
<b>Para</b>	Manter atualizada a base de dados de planos treinos ofertados

Figura 47 - Tela de Manter Plano de Treino

### Critérios de Aceitação

#### Critério 1

Deve cadastrar planos de treinos, agrupando treinos.

#### Critério 2

Deve realizar a alteração de planos de treinos.

#### Regra

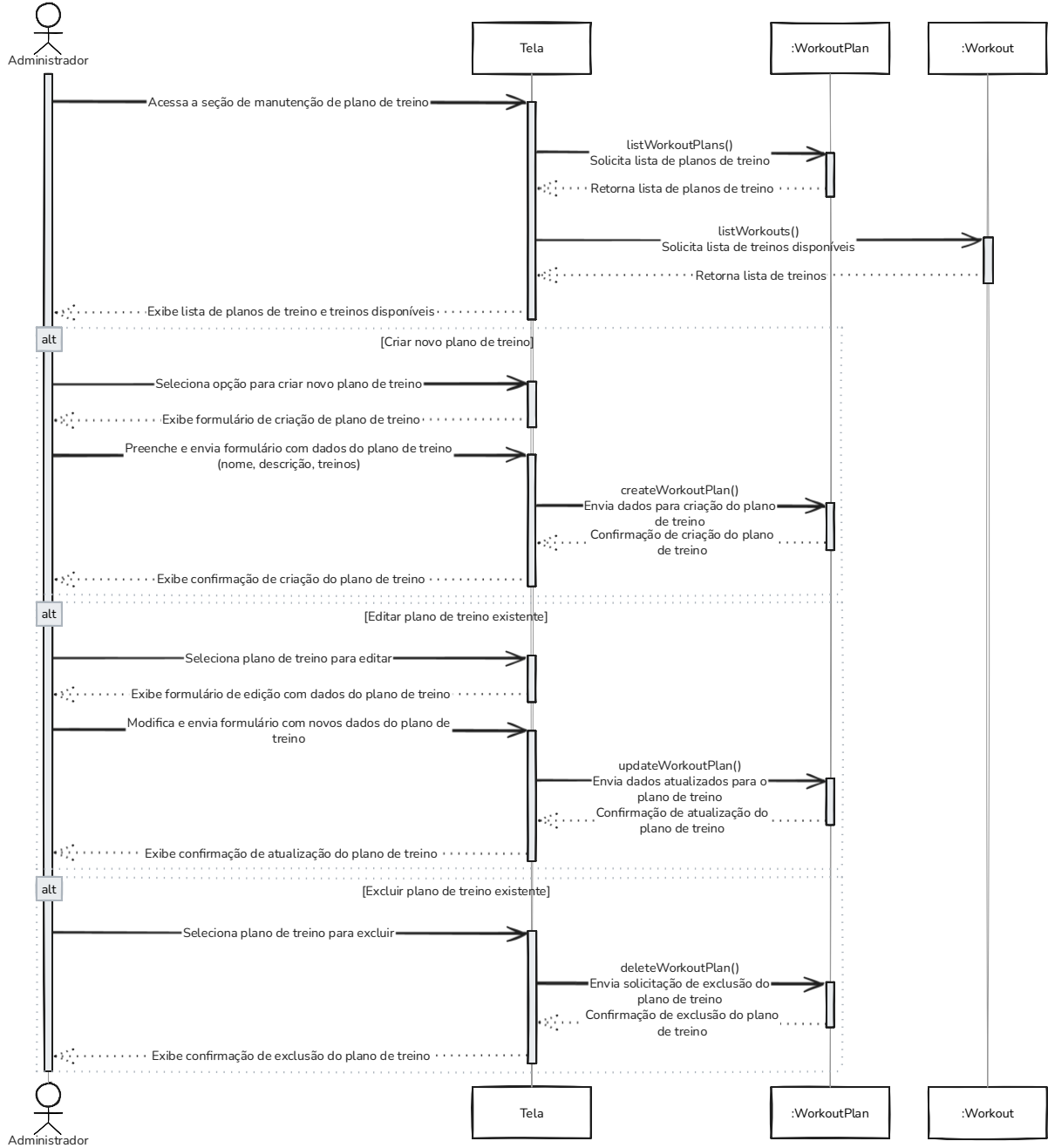
Quando a tela for carregada em modo de alteração, todos os treinos devem estar preenchidos com as informações do banco de dados. Os treinos devem ficar disponível para alteração. Após a confirmação, salvar o registro com as alterações feitas, exibir a mensagem “plano de treinos alterado com sucesso” e retornar à tela anterior.

<b>Critério 4</b>
Deve excluir o plano de treinos.
<b>Regra</b>
Quando o botão de exclusão for clicado, um modal deve aparecer com os dados do plano de treino e perguntar se o usuário tem certeza da exclusão. Havendo confirmação, o treino deve ser excluído do banco de dados.

<b>Critério 5</b>
Deve retornar à tela inicial.
<b>Regra</b>
Quando for pressionado o botão de voltar, deve retornar à tela inicial.

## Diagrama de Sequência 023 – Manter plano de treino

Figura 48 - Diagrama de Sequência - Manter Plano de Treino



## HU 24. HISTÓRIA DE USUÁRIO 024 – MANTER ATRIBUTO DE EXERCÍCIO

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Realizar operações de inclusão, alteração e exclusão de atributo de exercício
<b>Para</b>	Manter atualizada a base de dados de atributos disponíveis

Figura 49 - Tela de Manter Atributo de Exercício

**Critérios de Aceitação**

<b>Critério 1</b>
Não deve permitir o cadastro de atributos de exercícios se algum campo obrigatório estiver vazio.
<b>Regra</b>
Quando for pressionado o botão “Adicionar” e houver um campo obrigatório vazio, deve ser exibido o erro “Este campo é obrigatório” logo abaixo do campo.

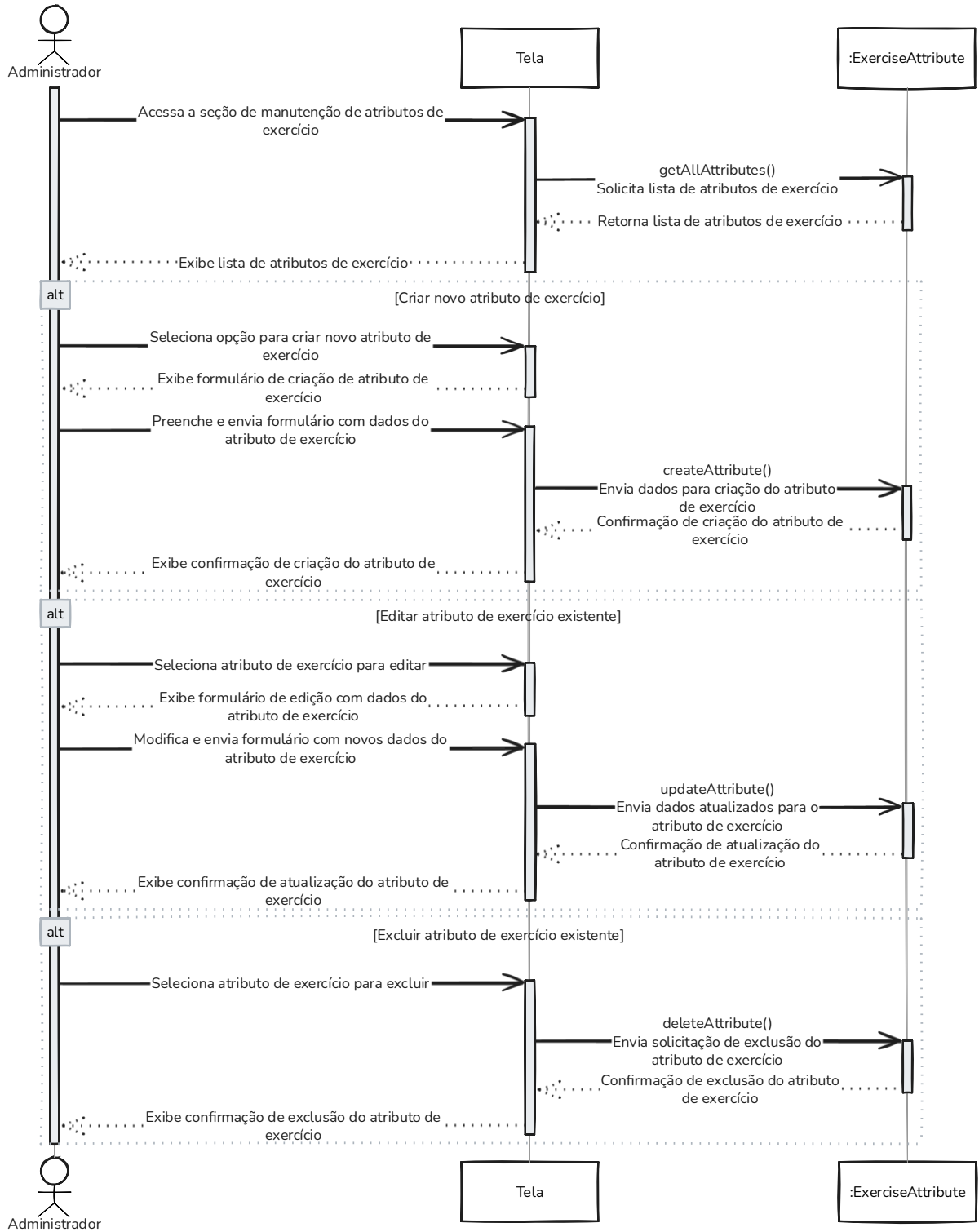
<b>Critério 2</b>
Deve cadastrar atributos de exercícios.
<b>Regra</b>
Quando todos os campos obrigatórios estiverem preenchidos corretamente, o sistema deve salvar no banco de dados o novo atributo.

<b>Critério 3</b>
Deve realizar a alteração de atributos.
<b>Regra</b>
Quando a tela for carregada em modo de alteração, todos os campos devem estar preenchidos com as informações do banco de dados. Após a confirmação, salvar o registro com as alterações feitas, exibir a mensagem de sucesso e retornar à tela anterior.

<b>Critério 4</b>
Deve excluir o atributo de exercício.
<b>Regra</b>
Quando o botão de exclusão for clicado, um modal deve aparecer com os dados do atributo de exercício e perguntar se o usuário tem certeza da exclusão. Havendo confirmação, o atributo deve ser excluído do banco de dados.

### Diagrama de Sequência 024 – Manter atributo de exercício

Figura 50 - Diagrama de Sequência - Manter Atributo de Exercício



## HU 25. HISTÓRIA DE USUÁRIO 025 – MANTER CATEGORIA DE EXERCÍCIO

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Realizar operações de inclusão, alteração e exclusão de categoria de exercício.
<b>Para</b>	Manter atualizada a base de dados de categorias disponíveis

Figura 51 - Tela de Manter Categoria de Exercício

### Critérios de Aceitação

<b>Critério 1</b>
Não deve permitir o cadastro de categorias de exercícios se algum campo obrigatório estiver vazio.
<b>Regra</b>
Quando for pressionado o botão “Adicionar” e houver um campo obrigatório vazio, deve ser exibido o erro “Este campo é obrigatório” logo abaixo do campo.

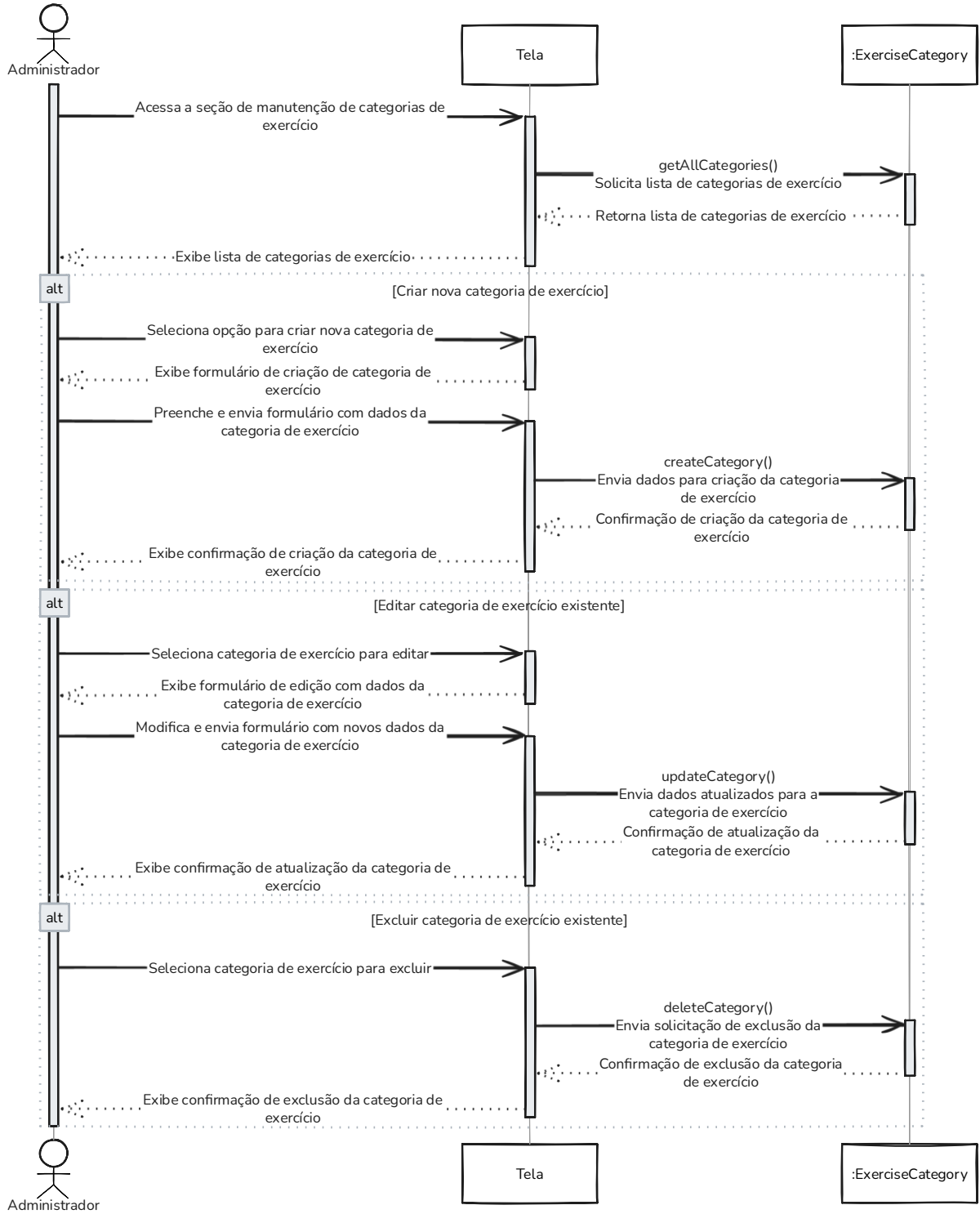
<b>Critério 2</b>
Deve cadastrar categorias de exercícios.
<b>Regra</b>
Quando todos os campos obrigatórios estiverem preenchidos corretamente, o sistema deve salvar no banco de dados a nova categoria.

<b>Critério 3</b>
Deve realizar a alteração de categoria de exercícios.
<b>Regra</b>
Quando a tela for carregada em modo de alteração, todos os campos devem estar preenchidos com as informações do banco de dados. Após a confirmação, salvar o registro com as alterações feitas, exibir a mensagem de sucesso e retornar à tela anterior.

<b>Critério 4</b>
Deve excluir a categoria de exercício.
<b>Regra</b>
Quando o botão de exclusão for clicado, um modal deve aparecer com os dados da categoria de exercício e perguntar se o usuário tem certeza da exclusão. Havendo confirmação, a categoria deve ser excluída do banco de dados.

## Diagrama de Sequência 025 – Manter categoria de exercício

Figura 52 - Diagrama de Sequência - Manter Categoria de Exercício



## HU 26. HISTÓRIA DE USUÁRIO 026 – MANTER CATEGORIA DE PUBLICAÇÃO

<b>Sendo</b>	Um administrador do sistema
<b>Quero</b>	Realizar operações de inclusão, alteração e exclusão de categoria de publicação
<b>Para</b>	Manter atualizada a base de dados de categorias disponíveis

Figura 53 - Tela de Manter Categoria de Publicação

### Categorias de publicação

Organize as publicações em categorias para facilitar a busca.

Nome

Descrição

Adicionar

#### Educativo

Editar

Remover

Publicações criadas com o objetivo de promover o acesso à informação. Inclui tutoriais, tira-dúvidas e dicas

#### Engajamento

Editar

Remover

Publicações focadas em gerar interação. Abrange enquetes, perguntas, desafios, convites à participação e chamadas para comentários.

#### Motivacional

Editar

Remover

Publicações criadas para inspirar ou estimular o público. Inclui mensagens de incentivo, histórias de superação e conteúdos que reforçam propósito ou ambição.

### Critérios de Aceitação

Critério 1
Não deve permitir o cadastro de categorias de publicações se algum campo obrigatório estiver vazio.
Regra
Quando for pressionado o botão “Adicionar” e houver um campo obrigatório vazio, deve ser exibido o erro “Este campo é obrigatório” logo abaixo do campo.

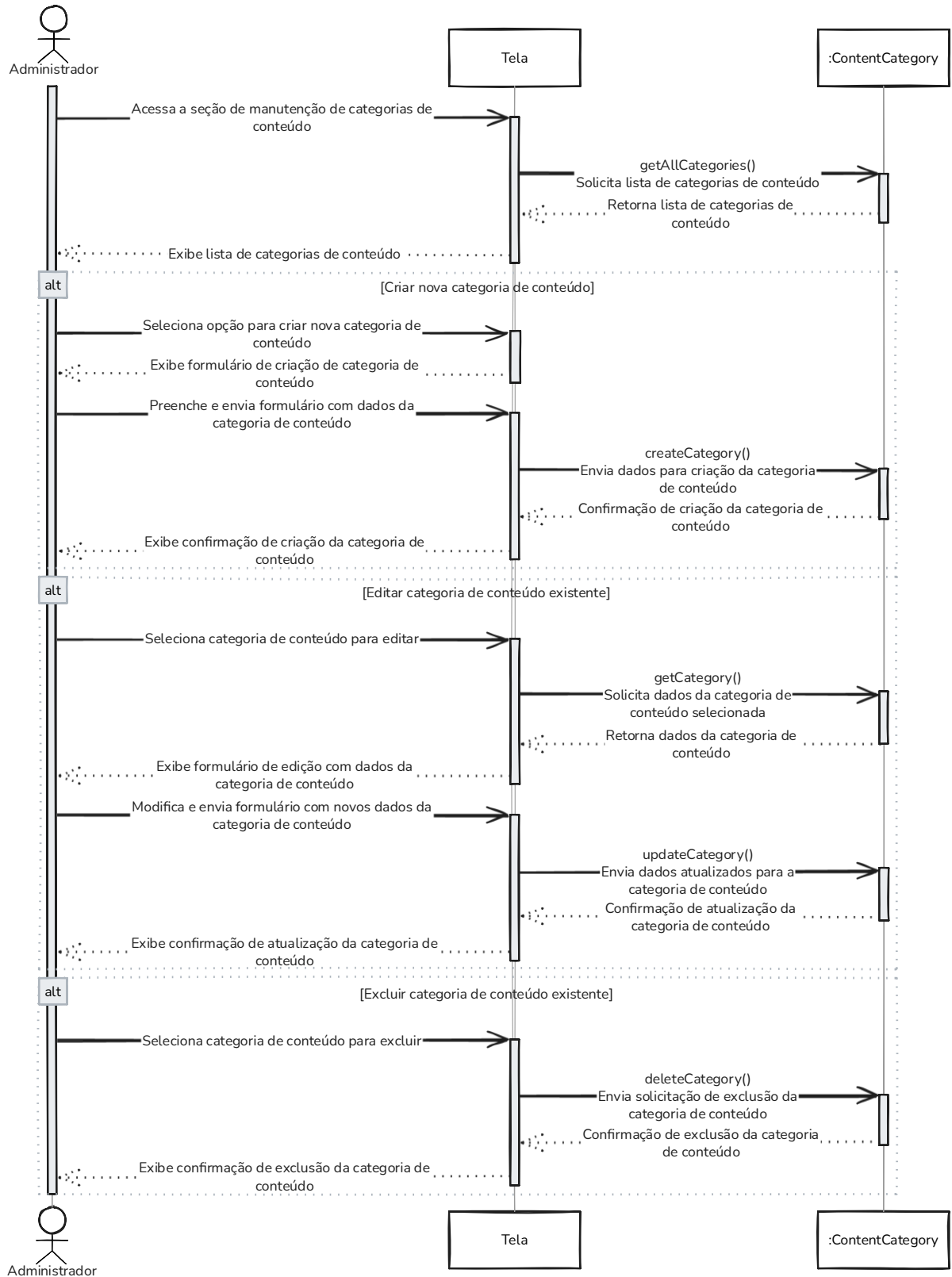
<b>Critério 2</b>
Deve cadastrar categorias de publicações.
<b>Regra</b>
Quando todos os campos obrigatórios estiverem preenchidos corretamente, o sistema deve salvar no banco de dados a nova categoria.

<b>Critério 3</b>
Deve realizar a alteração de categoria de publicações.
<b>Regra</b>
Quando a tela for carregada em modo de alteração, todos os campos devem estar preenchidos com as informações do banco de dados. Após a confirmação, salvar o registro com as alterações feitas, exibir a mensagem de sucesso e retornar à tela anterior.

<b>Critério 4</b>
Deve excluir a categoria de publicação.
<b>Regra</b>
Quando o botão de exclusão for clicado, um modal deve aparecer com os dados da categoria e perguntar se o usuário tem certeza da exclusão. Havendo confirmação, a categoria deve ser excluída do banco de dados.

## Diagrama de Sequência 026 – Manter categoria de publicação

Figura 54 - Diagrama de Sequência - Manter Categoria de Publicação

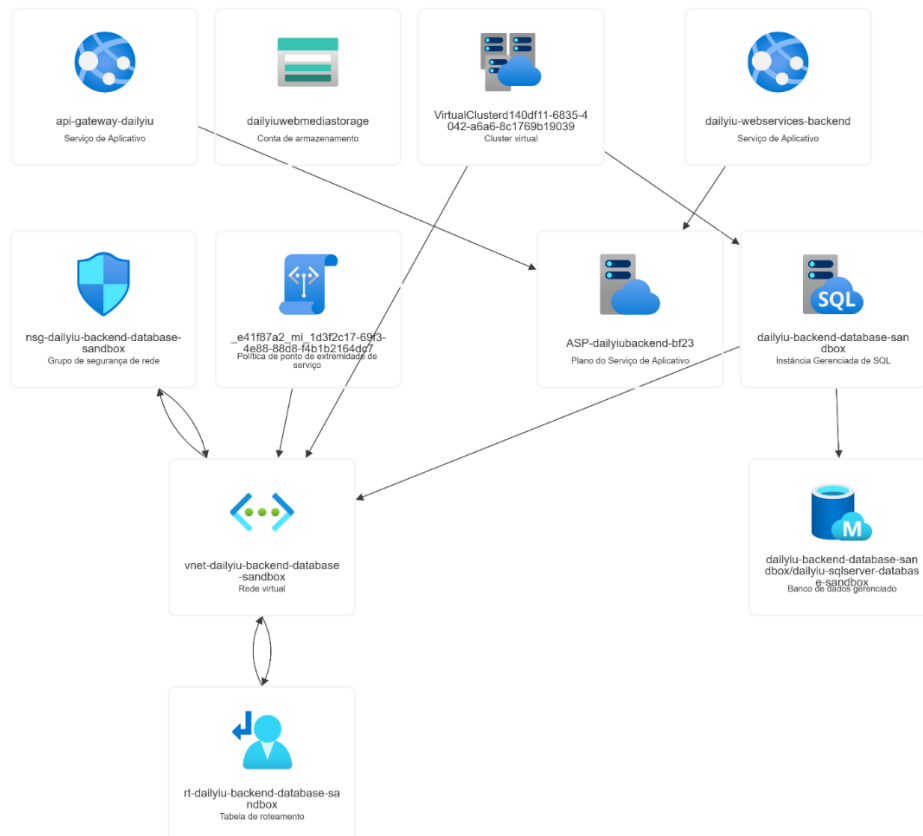




## APÊNDICE E - ARQUITETURA DA INFRAESTRUTURA

O diagrama a seguir representa a arquitetura dos recursos de infraestrutura hospedados na Azure. As setas representam a hierarquia e as relações de dependência entre os recursos, evidenciando sua organização de forma aninhada. O API Gateway (api-gateway-dailyiu) e o Backend (dailyiu-webservices-backend) são contêineres hospedados como Azure App Services, os quais estão contidos na instância do App Service Plan do Azure (ASP-dailyiubackend-bf23). De maneira análoga, o banco de dados (dailyiu-sqlserver-database-sandbox) encontra-se contido na instância gerenciada de banco de dados do Azure (dailyiu-backend-database-sandbox). A instância do banco de dados está inserida em um cluster virtual juntamente com a rede virtual privada, uma vez que ambos foram provisionados de forma integrada. Essa rede dispõe de um ponto de extremidade que permite o acesso pela internet, além de um Network Security Group (NSG) responsável pela definição e aplicação de regras de segurança interna, e de uma tabela de roteamento que gerencia o tráfego destinado aos contêineres de serviço.

Figura 56 - Diagrama da Arquitetura dos Recursos de Infraestrutura Hospedados na Azure



## **APÊNDICE F - ESPECIFICAÇÃO DE APPLICATION PROGRAMMING INTERFACES (APIS)**

O documento a seguir refere-se às especificações das APIs baseadas no Swagger e OpenAPI

**API Reference**

# API de Serviços Daily IU

API Version: 1.0.0

API de serviços para o projeto Daily IU, TCC da Universidade Federal do Paraná (UFPR), 2025.

# INDEX

<b>1. ADMIN</b>	<b>6</b>
1.1 POST /v1/admin/users/role	6
1.2 POST /v1/admin/reports/validate	8
1.3 POST /v1/admin/reports/strike	10
1.4 PATCH /v1/admin/users/status	11
1.5 GET /v1/admin/users	12
1.6 GET /v1/admin/reports	15
<b>2. ATRIBUTO DE EXERCÍCIO</b>	<b>18</b>
2.1 PUT /v1/exercise/attribute/{id}	18
2.2 DELETE /v1/exercise/attribute/{id}	19
2.3 GET /v1/exercise/attribute	21
2.4 POST /v1/exercise/attribute	22
<b>3. CATEGORIA DE CONTEÚDO</b>	<b>25</b>
3.1 PUT /v1/content/category/{id}	25
3.2 DELETE /v1/content/category/{id}	26
3.3 GET /v1/content/category	28
3.4 POST /v1/content/category	29
<b>4. CATEGORIA DE EXERCÍCIO</b>	<b>32</b>
4.1 PUT /v1/exercise/category/{id}	32
4.2 DELETE /v1/exercise/category/{id}	33
4.3 GET /v1/exercise/category	35
4.4 POST /v1/exercise/category	36
<b>5. COMENTÁRIOS</b>	<b>38</b>
5.1 GET /v1/content/comments/{id}	38
5.2 PUT /v1/content/comments/{id}	39
5.3 DELETE /v1/content/comments/{id}	41
5.4 PATCH /v1/content/comments/{id}	42
5.5 POST /v1/content/comments	44
5.6 GET /v1/content/comments/{id}/replies	46
<b>6. CONTATO</b>	<b>48</b>
6.1 POST /v1/contact/support	48
6.2 POST /v1/contact/professional-request	49
<b>7. CONTEÚDO</b>	<b>51</b>
7.1 GET /v1/content/{id}	51
7.2 PUT /v1/content/{id}	53
7.3 DELETE /v1/content/{id}	56
7.4 PATCH /v1/content/{id}	57
7.5 GET /v1/content	58

7.6	POST /v1/content	60
7.7	POST /v1/content/{id}/repost	63
7.8	POST /v1/content/{id}/report	65
7.9	PATCH /v1/content/{id}/save	67
7.10	GET /v1/content/saved	68
<b>8.</b>	<b>DIÁRIO</b>	<b>71</b>
8.1	GET /v1/calendar	71
8.2	PUT /v1/calendar	72
<b>9.</b>	<b>EXERCÍCIO</b>	<b>75</b>
9.1	PUT /v1/exercise/{id}	75
9.2	DELETE /v1/exercise/{id}	77
9.3	GET /v1/exercise	78
9.4	POST /v1/exercise	80
<b>10.</b>	<b>MÍDIA</b>	<b>83</b>
10.1	POST /v1/media/upload	83
<b>11.</b>	<b>PLANO DE TREINO</b>	<b>85</b>
11.1	PUT /v1/exercise/workout/plan/{id}	85
11.2	DELETE /v1/exercise/workout/plan/{id}	86
11.3	GET /v1/exercise/workout/plan	88
11.4	POST /v1/exercise/workout/plan	89
<b>12.</b>	<b>PREFERÊNCIAS</b>	<b>92</b>
12.1	GET /v1/preferences/notifications	92
12.2	PATCH /v1/preferences/notifications	93
12.3	GET /v1/preferences/accessibility	94
12.4	PATCH /v1/preferences/accessibility	96
<b>13.</b>	<b>QUESTIONÁRIO INICIAL</b>	<b>98</b>
13.1	GET /v1/questions/onboard	98
13.2	POST /v1/questions/onboard	99
<b>14.</b>	<b>RELATÓRIO</b>	<b>103</b>
14.1	GET /v1/report	103
<b>15.</b>	<b>TREINO</b>	<b>105</b>
15.1	PUT /v1/exercise/workout/{id}	105
15.2	DELETE /v1/exercise/workout/{id}	106
15.3	GET /v1/exercise/workout	108
15.4	POST /v1/exercise/workout	109
<b>16.</b>	<b>USUÁRIO</b>	<b>112</b>
16.1	GET /v1/users/{id}	112
16.2	PUT /v1/users/{id}	113

16.3	POST	/v1/users	115
16.4	POST	/v1/users/workout/feedback	118
16.5	POST	/v1/users/workout/completion	119
16.6	POST	/v1/users/password/reset	121
16.7	POST	/v1/users/password/forgot	122
16.8	POST	/v1/users/login	124
16.9	GET	/v1/users/workout/plan	126

# Security and Authentication

## SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
bearerAuth	http, bearer, JWT	

# API

## 1. ADMIN

Endpoints para tarefas administrativas

### 1.1 POST /v1/admin/users/role

#### Definir papel do usuário

Define ou altera o papel de um usuário

#### REQUEST

REQUEST BODY - application/json

```
{
  targetUserId*           integer
  description*            string
  permissionLevel*       integer
  reason*                 string
  hasDocument*           boolean
  documentType            string
  documentValue           string
  isDocumentInfoValid*   boolean
  isDocumentRequiredForPermissionLevel* boolean
}
```

#### RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - \*/\*

```
{
  id           integer
  name*        string
  email*       string
  role* {
    description*   string
    permissionLevel* integer
    reason*        string
    hasDocument*  boolean
    documentType  string
    documentValue  string
    conceivedBy {
      id           integer
      name*        string
      email*       string
      profilePicture {
        id           integer
        url*         string
        contentType* string
        contentSize* integer
        altText      string
        createdAt*  string
      }
    }
  }
}
```

```

    }
    profile {
        id            integer
        birthDate*   string
        gender*      string
        iciq3answer* integer
        iciq4answer* integer
        iciq5answer* integer
        iciqScore*   integer
        urinationLoss* string
        createdAt*   string
        updatedAt*   string
    }
    credential* {
        id            integer
        passwordHash* string
        salt*         string
    }
    preferences* {
        id                    integer
        highContrast*         boolean
        bigFont*              boolean
        darkMode*             boolean
        reminderCalendar*     boolean
        reminderCalendarSchedule string
        reminderWorkout*      boolean
        reminderWorkoutSchedule string
        encouragingMessages*  boolean
        notificationToken*    string
    }
    role*           {recursive}    Role
    strikes*        integer
    blocked*        boolean
}
conceivedAt*      string
}
status*           string
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

## STATUS CODE - 403: Forbidden

### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## STATUS CODE - 404: Not Found

### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## STATUS CODE - 500: Internal Server Error

### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## STATUS CODE - 501: Not Implemented

### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 1.2 POST /v1/admin/reports/validate

### Validar denúncia

Marca uma denúncia como válida ou inválida

### REQUEST

#### REQUEST BODY - application/json

```
{
  contentId* integer
  reportId*  integer
  valid*     boolean
}
```

### RESPONSE

**STATUS CODE - 204:** Denúncia validada com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
}
```

```
  details [string]
}
```

## 1.3 POST /v1/admin/reports/strike

### Aplicar advertência

Aplica uma advertência a um usuário com base em uma publicação oculta

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 204:** Advertência aplicada com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 1.4 PATCH /v1/admin/users/status

### Definir status do usuário (Ativo/Bloqueado)

Ativa ou bloqueia um usuário

#### REQUEST

##### REQUEST BODY - application/json

```
{
  targetUserId* integer
  blocked*      boolean
}
```

#### RESPONSE

##### STATUS CODE - 200: OK

##### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

##### STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

##### STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 1.5 GET /v1/admin/users

### Listar usuários

Recupera uma lista de todos os usuários

### REQUEST

#### HEADER PARAMETERS

NAME	TYPE	DESCRIPTION
page	int32	
size	int32	

### RESPONSE

STATUS CODE - 200: OK

## RESPONSE MODEL - \*/\*

[{

Array of object:

```
  id                integer
  name*             string
  email*            string
  role* {
    description*    string
    permissionLevel* integer
    reason*         string
    hasDocument*   boolean
    documentType    string
    documentValue   string
    conceivedBy {
      id            integer
      name*         string
      email*        string
      profilePicture {
        id          integer
        url*         string
        contentType* string
        contentType* integer
        altText     string
        createdAt*  string
      }
    }
    profile {
      id            integer
      birthDate*   string
      gender*       string
      iciq3answer* integer
      iciq4answer* integer
      iciq5answer* integer
      iciqScore*   integer
      urinationLoss* string
      createdAt*   string
      updatedAt*   string
    }
    credential* {
      id            integer
      passwordHash* string
      salt*         string
    }
    preferences* {
      id                integer
      highContrast*    boolean
      bigFont*         boolean
      darkMode*        boolean
      reminderCalendar* boolean
      reminderCalendarSchedule string
      reminderWorkout* boolean
      reminderWorkoutSchedule string
      encouragingMessages* boolean
      notificationToken* string
    }
  }
  role*           {recursive}   Role
  strikes*        integer
  blocked*        boolean
```

```
    }
    conceivedAt*      string
  }
  status*            string
}]
```

#### **STATUS CODE - 400: Bad Request**

##### **RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

#### **STATUS CODE - 401: Unauthorized**

##### **RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

#### **STATUS CODE - 403: Forbidden**

##### **RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

#### **STATUS CODE - 404: Not Found**

##### **RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

#### **STATUS CODE - 500: Internal Server Error**

##### **RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

#### **STATUS CODE - 501: Not Implemented**

##### **RESPONSE MODEL - application/json**

```

{
  code      integer
  origin    string
  message   string
  details   [string]
}

```

## 1.6 GET /v1/admin/reports

### Listar denúncias

Recupera uma lista de todas as denúncias, agrupadas por conteúdo

### REQUEST

#### HEADER PARAMETERS

NAME	TYPE	DESCRIPTION
page	int32	
size	int32	

### RESPONSE

STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```

[ {
  Array of object:
  id*           integer
  title*        string
  description*   string
  subtitle      string
  subcontent    string
  categories*   [string]
  author* {
    id*         integer
    name*       string
    profilePicture string
  }
  media* [ {
    Array of object:
    id           integer
    url*        string  min:1 chars
    contentType* string  min:1 chars
    contentType* integer
    altText     string
    createdAt   string
  } ]
  reports* [ {
    Array of object:
    id*         integer
    reason*     string
    createdAt*  string
  } ]
} ]

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 2. ATRIBUTO DE EXERCÍCIO

Endpoints para gerenciamento de atributos de exercício (Benefícios, Contraindicações, Dicas)

### 2.1 PUT /v1/exercise/attribute/{id}

#### Atualizar Atributo

Atualiza um atributo de exercício existente pelo ID

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

##### REQUEST BODY - application/json

```
{
  id          integer
  name*       string
  description* string  0 to 255 chars
  type*       integer  Tipo do atributo: 1 - Benefício, 2 - Contraindicação
}
```

#### RESPONSE

##### STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
{
  id          integer
  name*       string
  description* string  0 to 255 chars
  type*       integer  Tipo do atributo: 1 - Benefício, 2 - Contraindicação
}
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

##### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
}
```

```
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 2.2 DELETE /v1/exercise/attribute/{id}

### Excluir Atributo

Exclui um atributo de exercício pelo ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

## RESPONSE

**STATUS CODE - 204:** Atributo excluído com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
```

```
    origin string
    message string
    details [string]
}
```

## 2.3 GET /v1/exercise/attribute

### Buscar Todos Atributos

Recupera todos os atributos de exercício

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200: OK**

**RESPONSE MODEL - \*/\***

```
[{
  Array of object:
    id integer
    name* string
    description* string 0 to 255 chars
    type* integer Tipo do atributo: 1 - Benefício, 2 - Contraindicação
}]
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 2.4 POST /v1/exercise/attribute

### Criar Atributo

Cria um novo atributo de exercício

### REQUEST

**REQUEST BODY - application/json**

```
{
  id            integer
  name*         string
  description*  string  0 to 255 chars
  type*         integer  Tipo do atributo: 1 - Benefício, 2 - Contraindicação
}
```

### RESPONSE

**STATUS CODE - 201: Atributo criado com sucesso**

**RESPONSE MODEL - \*/\***

```
{
  id            integer
  name*         string
  description*  string  0 to 255 chars
  type*         integer  Tipo do atributo: 1 - Benefício, 2 - Contraindicação
}
```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

# 3. CATEGORIA DE CONTEÚDO

Endpoints para gerenciamento de categorias de conteúdo

## 3.1 PUT /v1/content/category/{id}

### Atualizar Categoria

Atualiza uma categoria de conteúdo existente pelo ID

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

##### REQUEST BODY - application/json

```
{
  id          integer
  name*       string
  description* string 0 to 255 chars
  auditable*  boolean
  createdAt   string
}
```

#### RESPONSE

##### STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
{
  id          integer
  name*       string
  description* string 0 to 255 chars
  auditable*  boolean
  createdAt   string
}
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

##### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code    integer
}
```

```
    origin  string
    message string
    details [string]
  }
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 3.2 DELETE /v1/content/category/{id}

### Excluir Categoria

Exclui uma categoria de conteúdo pelo ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

## RESPONSE

**STATUS CODE - 204:** Categoria excluída com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
```

```
    origin string
    message string
    details [string]
}
```

### 3.3 GET /v1/content/category

#### Buscar Todas Categorías

Recupera todas as categorías de conteúdo

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200: OK**

**RESPONSE MODEL - \*/\***

```
[{
  Array of object:
    id integer
    name* string
    description* string 0 to 255 chars
    auditable* boolean
    createdAt string
}]
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

```
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 3.4 POST /v1/content/category

### Criar Categoria

Cria uma nova categoria de conteúdo

### REQUEST

**REQUEST BODY - application/json**

```
{
  id          integer
  name*       string
  description* string 0 to 255 chars
  auditable*  boolean
  createdAt   string
}
```

### RESPONSE

**STATUS CODE - 201: Categoria criada com sucesso**

**RESPONSE MODEL - \*/\***

```
{
  id          integer
  name*       string
}
```

```
description* string 0 to 255 chars
auditable*   boolean
createdAt    string
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
```

```
code    integer
origin  string
message string
details [string]
}
```

---

## 4. CATEGORIA DE EXERCÍCIO

Endpoints para gerenciamento de categorias de exercício

### 4.1 PUT /v1/exercise/category/{id}

#### Atualizar Categoria

Atualiza uma categoria de exercício existente pelo ID

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

##### REQUEST BODY - application/json

```
{
  id          integer
  name*       string
  description string 0 to 255 chars
}
```

#### RESPONSE

##### STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
{
  id          integer
  name*       string
  description string 0 to 255 chars
}
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

##### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 4.2 DELETE /v1/exercise/category/{id}

### Excluir Categoria

Exclui uma categoria de exercício pelo ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

### RESPONSE

**STATUS CODE - 204: Categoria excluída com sucesso**

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 4.3 GET /v1/exercise/category

### Buscar Todas Categorías

Recupera todas as categorías de exercício

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200: OK**

**RESPONSE MODEL - \*/\***

```
[{  
  Array of object:  
  id          integer  
  name*       string  
  description string 0 to 255 chars  
}]
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
}
```

```
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 4.4 POST /v1/exercise/category

### Criar Categoria

Cria uma nova categoria de exercício

### REQUEST

**REQUEST BODY - application/json**

```
{
  id          integer
  name*       string
  description string 0 to 255 chars
}
```

### RESPONSE

**STATUS CODE - 201:** Categoria criada com sucesso

**RESPONSE MODEL - \*/\***

```
{
  id          integer
  name*       string
  description string 0 to 255 chars
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

```
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

# 5. COMENTÁRIOS

Endpoints para gerenciar comentários em conteúdos

## 5.1 GET /v1/content/comments/{id}

### Buscar comentários por ID do conteúdo

Recupera comentários de um item de conteúdo específico pelo seu ID, com paginação opcional

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
page	int32	

#### RESPONSE

##### STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
[{  
  Array of object:  
  id*           integer  
  author* {  
    id*         integer  
    name*       string  
    profilePicture string  
  }  
  text*        string  
  createdAt*   string  
  updatedAt*   string  
  likesCount*  integer  
  isLiked*     boolean  
  repliesCount* integer  
}]
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 5.2 PUT /v1/content/comments/{id}

### Atualizar Comentário

Atualiza um comentário existente pelo ID

### REQUEST

## PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

### REQUEST BODY - application/json

```
{
  contentId*      integer
  authorId*      integer
  text*          string
  replyToCommentId integer
}
```

## RESPONSE

### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
{
  id*            integer
  author* {
    id*          integer
    name*        string
    profilePicture string
  }
  text*          string
  createdAt*     string
  updatedAt*     string
  likesCount*   integer
  isLiked*      boolean
  repliesCount* integer
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

### STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

### STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{
  code    integer
```

```
    origin string
    message string
    details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

## 5.3 DELETE /v1/content/comments/{id}

### Deletar Comentário

Deleta um comentário pelo seu ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

### RESPONSE

**STATUS CODE - 204: Comentário deletado com sucesso**

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
```

```
code    integer
origin  string
message string
details [string]
}
```

#### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 5.4 PATCH /v1/content/comments/{id}

## Curtir/Descurtir Comentário

Alterna o status de curtida para um comentário específico pelo seu ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

#### REQUEST BODY - application/json

```
{
  userId*  integer
  control* boolean
}
```

### RESPONSE

**STATUS CODE - 204:** Status de curtida alterado com sucesso

**STATUS CODE - 400:** Bad Request

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 401:** Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 403:** Forbidden

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 404:** Not Found

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
}
```

```
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 5.5 POST /v1/content/comments

### Criar Comentário

Cria um novo comentário em um conteúdo

### REQUEST

**REQUEST BODY - application/json**

```
{
  contentId*      integer
  authorId*       integer
  text*           string
  replyToCommentId integer
}
```

### RESPONSE

**STATUS CODE - 201:** Comentário criado com sucesso

**RESPONSE MODEL - \*/\***

```
{
  id*           integer
  author* {
    id*         integer
    name*       string
    profilePicture string
  }
  text*         string
  createdAt*   string
  updatedAt*   string
  likesCount*  integer
  isLiked*     boolean
  repliesCount* integer
}
```

```
}
```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

#### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

#### STATUS CODE - 501: Not Implemented

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
}
```

```
    details [string]
  }
```

## 5.6 GET /v1/content/comments/{id}/replies

### Obter respostas de comentário

Recupera respostas para um comentário específico pelo seu ID, com paginação opcional

## REQUEST

### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
page	int32	

## RESPONSE

### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
[{
  Array of object:
  id*          integer
  author* {
    id*        integer
    name*      string
    profilePicture string
  }
  text*       string
  createdAt*  string
  updatedAt*  string
  likesCount* integer
  isLiked*    boolean
  repliesCount* integer
}]
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

### STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

---

# 6. CONTATO

Endpoints relacionados ao contato e suporte

## 6.1 POST /v1/contact/support

### Enviar e-mail de contato

Envia um e-mail usando o formulário de contato

#### REQUEST

REQUEST BODY - application/json

```
{
  userEmail* string min:1 chars
  subject*   string min:1 chars
  text*      string min:1 chars
}
```

#### RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

STATUS CODE - 401: Unauthorized

RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

STATUS CODE - 403: Forbidden

RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 6.2 POST /v1/contact/professional-request

### Enviar e-mail de solicitação profissional

Envia um e-mail de solicitação profissional usando o formulário de contato

#### REQUEST

##### REQUEST BODY - application/json

```
{
  name*   string
  email*  string
  reason* string min:1 chars
}
```

#### RESPONSE

STATUS CODE - 204: E-mail enviado com sucesso

STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 409: Usuário bloqueado não pode enviar solicitações profissionais**

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

---

# 7. CONTEÚDO

Endpoints para gerenciar o conteúdo da rede social

## 7.1 GET /v1/content/{id}

### Obter Conteúdo por ID

Recupera uma publicação de conteúdo específica pelo seu ID

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

#### RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - \*/\*

```
{
  id*                integer
  title*             string
  description*       string
  subtitle           string
  subcontent         string
  categories*       [string]
  author* {
    id*              integer
    name*            string
    profilePicture  string
  }
  cover {
    id               integer
    url*            string  min:1 chars
    contentType*    string  min:1 chars
    contentSize*    integer
    altText         string
    createdAt       string
  }
  media* [{
    Array of object:
    id               integer
    url*            string  min:1 chars
    contentType*    string  min:1 chars
    contentSize*    integer
    altText         string
    createdAt       string
  }]
  isSaved*          boolean
  isLiked*          boolean
}
```

```

likesCount*           integer
comments* [{
  Array of object:
    id*                integer
    author* {
      id*              integer
      name*            string
      profilePicture  string
    }
    text*              string
    createdAt*        string
    updatedAt*        string
    likesCount*       integer
    isLiked*          boolean
    repliesCount*     integer
  }]
commentsCount*       integer
isReposted*          boolean
repostedFromContentId integer
repostedByUser {
  id*                integer
  name*              string
  profilePicture    string
}
createdAt*           string
updatedAt*           string
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 404: Not Found**

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.2 PUT /v1/content/{id}

### Atualizar Conteúdo

Atualiza uma publicação de conteúdo existente pelo ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

#### REQUEST BODY - application/json

```
{
  title*           string   min:1 chars
  description*     string   min:1 chars
  subtitle         string
  subcontent       string
  media* [{
    Array of object:
      id            integer
      url*          string   min:1 chars
      contentType* string   min:1 chars
      contentSize* integer
      altText       string
      createdAt     string
  }]
}
```

```
}
```

## RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - \*/\*

```
{
  id*           integer
  title*        string
  description*   string
  subtitle      string
  subcontent    string
  categories*   [string]
  author* {
    id*         integer
    name*       string
    profilePicture string
  }
  cover {
    id          integer
    url*        string  min:1 chars
    contentType* string  min:1 chars
    contentSize* integer
    altText     string
    createdAt   string
  }
  media* [{
    Array of object:
    id          integer
    url*        string  min:1 chars
    contentType* string  min:1 chars
    contentSize* integer
    altText     string
    createdAt   string
  }]
  isSaved*      boolean
  isLiked*      boolean
  likesCount*   integer
  comments* [{
    Array of object:
    id*         integer
    author* {
      id*       integer
      name*     string
      profilePicture string
    }
    text*       string
    createdAt*  string
    updatedAt*  string
    likesCount* integer
    isLiked*    boolean
    repliesCount* integer
  }]
  commentsCount* integer
  isReposted*   boolean
  repostedFromContentId integer
}
```

```
repostedByUser {
  id*          integer
  name*        string
  profilePicture string
}
createdAt*    string
updatedAt*    string
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.3 DELETE /v1/content/{id}

### Deletar Conteúdo

Deleta uma publicação de conteúdo pelo seu ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

### RESPONSE

**STATUS CODE - 204:** Conteúdo deletado com sucesso, sem conteúdo na resposta

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.4 PATCH /v1/content/{id}

### Curtir/Descurtir Conteúdo

Altera o status de curtida para uma publicação de conteúdo específica pelo seu ID

## REQUEST

**PATH PARAMETERS**

NAME	TYPE	DESCRIPTION
*id	int64	

**REQUEST BODY - application/json**

```
{
  userId*  integer
  control* boolean
}
```

## RESPONSE

**STATUS CODE - 204:** Operação realizada com sucesso, sem conteúdo na resposta

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.5 GET /v1/content

## Obter Conteúdos

Recupera publicações de conteúdo para um usuário. Se o cabeçalho 'x-profile' for verdadeiro, busca conteúdos específicos do perfil com base no ID do usuário, caso contrário, busca publicações recomendadas para o usuário.

## REQUEST

### HEADER PARAMETERS

NAME	TYPE	DESCRIPTION
*x-user-id	int64	
x-profile	boolean	
page	int32	
size	int32	

## RESPONSE

### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
[{
  Array of object:
  id                integer
  title*           string
  categories*      [string]
  author* {
    id*            integer
    name*          string
    profilePicture string
  }
  cover {
    id              integer
    url*           string  min:1 chars
    contentType*   string  min:1 chars
    contentSize*   integer
    altText        string
    createdAt      string
  }
  section          [string]
  isReposted*     boolean
  createdAt*      string
  updatedAt*      string
}]
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.6 POST /v1/content

### Criar Conteúdo

Criar uma nova publicação de conteúdo

### REQUEST

## REQUEST BODY - application/json

```
{
  title*           string      min:1 chars
  description*     string      min:1 chars
  subtitle         string
  subcontent       string
  categoryIds*    [integer]
  authorId*       integer
  media* [{
    Array of object:
      id           integer
      url*         string      min:1 chars
      contentType* string      min:1 chars
      contentType* integer
      altText      string
      createdAt    string
  }]
}
```

## RESPONSE

STATUS CODE - 201: Conteúdo criado com sucesso

### RESPONSE MODEL - \*/\*

```
{
  id*           integer
  title*        string
  description*   string
  subtitle      string
  subcontent     string
  categories*   [string]
  author* {
    id*         integer
    name*       string
    profilePicture string
  }
  cover {
    id           integer
    url*         string      min:1 chars
    contentType* string      min:1 chars
    contentType* integer
    altText      string
    createdAt    string
  }
  media* [{
    Array of object:
      id           integer
      url*         string      min:1 chars
      contentType* string      min:1 chars
      contentType* integer
      altText      string
      createdAt    string
  }]
  isSaved*      boolean
  isLiked*      boolean
  likesCount*   integer
  comments* [{
    Array of object:

```

```

    id*           integer
    author* {
      id*         integer
      name*       string
      profilePicture string
    }
    text*         string
    createdAt*   string
    updatedAt*   string
    likesCount*  integer
    isLiked*     boolean
    repliesCount* integer
  }]
  commentsCount* integer
  isReposted*    boolean
  repostedFromContentId integer
  repostedByUser {
    id*         integer
    name*       string
    profilePicture string
  }
  createdAt*    string
  updatedAt*    string
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{

```

```
code    integer
origin  string
message string
details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.7 POST /v1/content/{id}/repost

### Repostar Conteúdo

Repostar uma publicação de conteúdo existente pelo ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

#### REQUEST BODY - application/json

```
{
  repostedFromContentId* integer
  repostedByUserId*     integer
}
```

### RESPONSE

**STATUS CODE - 200: OK**

**RESPONSE MODEL - \*/\***

```
{
  id*           integer
  title*        string
  description*  string
  subtitle      string
  subcontent    string
}
```

```

categories*           [string]
author* {
  id*                 integer
  name*               string
  profilePicture     string
}
cover {
  id                 integer
  url*               string   min:1 chars
  contentType*      string   min:1 chars
  contentSize*      integer
  altText            string
  createdAt          string
}
media* [{
Array of object:
  id                 integer
  url*               string   min:1 chars
  contentType*      string   min:1 chars
  contentSize*      integer
  altText            string
  createdAt          string
}]
isSaved*              boolean
isLiked*              boolean
likesCount*           integer
comments* [{
Array of object:
  id*                 integer
  author* {
    id*               integer
    name*              string
    profilePicture    string
  }
  text*               string
  createdAt*          string
  updatedAt*          string
  likesCount*         integer
  isLiked*            boolean
  repliesCount*       integer
}]
commentsCount*        integer
isReposted*           boolean
repostedFromContentId integer
repostedByUser {
  id*                 integer
  name*               string
  profilePicture     string
}
createdAt*            string
updatedAt*            string
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  code    integer

```

```
    origin string
    message string
    details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

## 7.8 POST /v1/content/{id}/report

### Reportar Conteúdo

## Reportar uma publicação de conteúdo pelo seu ID

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

#### REQUEST BODY - application/json

```
{
  reporterId* integer
  reason*     string
}
```

### RESPONSE

**STATUS CODE - 204:** Report enviado com sucesso, sem conteúdo na resposta

**STATUS CODE - 400:** Bad Request

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 401:** Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 403:** Forbidden

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 404:** Not Found

#### RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.9 PATCH /v1/content/{id}/save

### Salvar/Remover dos Salvos

Altera o status de salvo para uma publicação de conteúdo específica pelo seu ID

### REQUEST

**PATH PARAMETERS**

NAME	TYPE	DESCRIPTION
*id	int64	

**REQUEST BODY - application/json**

```
{
  userId*  integer
  control* boolean
}
```

### RESPONSE

**STATUS CODE - 204:** Operação realizada com sucesso, sem conteúdo na resposta

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 7.10 GET /v1/content/saved

### Obter Conteúdos salvos

Recupera publicações de conteúdo salvos por um usuário.

#### REQUEST

No request parameters

#### RESPONSE

## STATUS CODE - 200: OK

### RESPONSE MODEL - \*/\*

```
[{
  Array of object:
  id          integer
  title*     string
  categories* [string]
  author* {
    id*       integer
    name*     string
    profilePicture string
  }
  cover {
    id          integer
    url*       string  min:1 chars
    contentType* string  min:1 chars
    contentSize* integer
    altText    string
    createdAt  string
  }
  section    [string]
  isReposted* boolean
  createdAt* string
  updatedAt* string
}]
```

## STATUS CODE - 400: Bad Request

### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## STATUS CODE - 401: Unauthorized

### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## STATUS CODE - 403: Forbidden

### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## STATUS CODE - 404: Not Found

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

---

# 8. DIÁRIO

Endpoints relacionados ao diário miccional do usuário

## 8.1 GET /v1/calendar

### Listar eventos do calendário

Retorna os eventos do calendário do usuário em um intervalo de datas. Se as datas não forem fornecidas, retorna eventos do mês atual.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
from	date	
to	date	

### RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - \*/\*

```
{
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

STATUS CODE - 401: Unauthorized

RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

STATUS CODE - 403: Forbidden

RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
}
```

```
message string
details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 8.2 PUT /v1/calendar

### Adicionar ou atualizar evento no calendário

Adiciona ou atualiza um evento no calendário do usuário para uma data específica.

#### REQUEST

**REQUEST BODY - application/json**

```
{
  date*           string
  leakageLevel*  string
  notesPreview   string
  urinationData [{
    Array of object:
    id            integer
    time*        string
    amount*      string
    leakage*     boolean
    reason       string
    urgency     boolean
  }]
}
```

## RESPONSE

### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
{
  date*           string
  leakageLevel*   string
  eventsCount*    integer
  completedExercises* integer
  notesPreview    string
  urinationData [{
    Array of object:
      id          integer
      time*       string
      amount*     string
      leakage*    boolean
      reason      string
      urgency     boolean
    }]
  dayTitle*       string
  dayNumber*      integer
  isToday*        boolean
  today           boolean WRITE-ONLY
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

### STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

### STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

---

# 9. EXERCÍCIO

Endpoints para gerenciamento de exercícios

## 9.1 PUT /v1/exercise/{id}

### Atualizar Exercício

Atualizar um exercício existente

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

#### REQUEST BODY - application/json

```
{
  title*           string      min:1 chars
  instructions*    string      min:1 chars
  categoryId*      integer
  media* [{
    Array of object:
    id              integer
    url*            string      min:1 chars
    contentType*    string      min:1 chars
    contentSize*    integer
    altText         string
    createdAt       string
  }]
  attributes*     [integer]
  repetitions*    integer
  sets*           integer
  restTime*       integer
  duration*       integer
}
```

### RESPONSE

#### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
{
  id              integer
  title*          string
  instructions*   string
  category*       string
  media* [{
    Array of object:
    id              integer
    url*            string      min:1 chars
    contentType*    string      min:1 chars
  }]
}
```

```

    contentSize* integer
    altText      string
    createdAt    string
  }]
  benefits* [{
    Array of object:
    id           integer
    name*        string
    description* string 0 to 255 chars
    type*        integer Tipo do atributo: 1 - Benefício, 2 - Contraindicação
  }]
  repetitions*  integer
  sets*         integer
  restTime*     integer
  duration*     integer
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 500: Internal Server Error**

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 9.2 DELETE /v1/exercise/{id}

### Deletar Exercício

Deletar um exercício existente

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

### RESPONSE

STATUS CODE - 204: Exercício deletado com sucesso

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 9.3 GET /v1/exercise

### Listar Exercícios

Listar todos os exercícios disponíveis

### REQUEST

No request parameters

### RESPONSE

#### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
[{
  Array of object:
    id          integer
    title*     string
}
```

```

instructions*           string
category*              string
media* [{
  Array of object:
    id                 integer
    url*               string  min:1 chars
    contentType*      string  min:1 chars
    contentSize*      integer
    altText            string
    createdAt          string
  }]
benefits* [{
  Array of object:
    id                 integer
    name*              string
    description*       string  0 to 255 chars
    type*              integer  Tipo do atributo: 1 - Benefício, 2 - Contraindicação
  }]
repetitions*           integer
sets*                  integer
restTime*              integer
duration*              integer
}]

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  code    integer
}

```

```
    origin  string
    message string
    details [string]
  }
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 9.4 POST /v1/exercise

### Criar Exercício

Criar um novo exercício

### REQUEST

**REQUEST BODY - application/json**

```
{
  title*      string      min:1 chars
  instructions* string      min:1 chars
  categoryId* integer
  media* [{
    Array of object:
    id          integer
    url*        string      min:1 chars
    contentType* string      min:1 chars
    contentSize* integer
    altText     string
    createdAt   string
  }]
  attributes* [integer]
  repetitions* integer
  sets*        integer
  restTime*    integer
  duration*    integer
}
```

### RESPONSE

**STATUS CODE - 201: Exercício criado com sucesso**

## RESPONSE MODEL - \*/\*

```
{
  id                integer
  title*           string
  instructions*    string
  category*        string
  media* [{
    Array of object:
      id            integer
      url*          string  min:1 chars
      contentType* string  min:1 chars
      contentType* integer
      altText       string
      createdAt     string
    }]
  benefits* [{
    Array of object:
      id            integer
      name*         string
      description*  string  0 to 255 chars
      type*         integer  Tipo do atributo: 1 - Benefício, 2 - Contraindicação
    }]
  repetitions*     integer
  sets*            integer
  restTime*        integer
  duration*        integer
}
```

## STATUS CODE - 400: Bad Request

### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## STATUS CODE - 401: Unauthorized

### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## STATUS CODE - 403: Forbidden

### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

---

# 10. MÍDIA

Endpoints para gerenciar uploads de mídia

## 10.1 POST /v1/media/upload

### Enviar Mídia

Envie um ou mais arquivos de mídia

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*files	array of string	

#### RESPONSE

##### STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
[{  
  Array of object:  
  url*      string  
  contentType string  
  contentSize* integer  
}]
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

##### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

##### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{  
  code    integer
```

```
    origin  string
    message string
    details [string]
}
```

#### **STATUS CODE - 404: Not Found**

##### **RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### **STATUS CODE - 500: Internal Server Error**

##### **RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### **STATUS CODE - 501: Not Implemented**

##### **RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

---

# 11. PLANO DE TREINO

Endpoints para gerenciamento de planos de treino

## 11.1 PUT /v1/exercise/workout/plan/{id}

### Atualizar Plano de Treino

Atualizar um plano de treino existente

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

##### REQUEST BODY - application/json

```
{
  name*           string
  description*    string
  workoutIds* {
  }
  daysPerWeek*   integer between 1 and 7
  totalWeeks*    integer
  iciqScoreMin*  integer >=1
  iciqScoreMax*  integer <=12
  ageMin*        integer >=12
  ageMax*        integer <=120
}
```

#### RESPONSE

##### STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
{
  id              integer
  name*           string
  description*    string
  workouts* {
  }
  daysPerWeek*   integer
  totalWeeks*    integer
  iciqScoreMin*  integer
  iciqScoreMax*  integer
  ageMin*        integer
  ageMax*        integer
}
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 11.2 DELETE /v1/exercise/workout/plan/{id}

## Deletar Plano de Treino

Deletar um plano de treino existente

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

### RESPONSE

**STATUS CODE - 204:** Plano de treino deletado com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 11.3 GET /v1/exercise/workout/plan

### Listar Planos de Treino

Listar todos os planos de treino disponíveis

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200: OK**

**RESPONSE MODEL - \*/\***

```
[{
  Array of object:
  id            integer
  name*         string
  description*  string
  workouts* {
  }
  daysPerWeek* integer
  totalWeeks*  integer
  iciqScoreMin* integer
  iciqScoreMax* integer
  ageMin*      integer
  ageMax*      integer
}]
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 11.4 POST /v1/exercise/workout/plan

### Criar Plano de Treino

Criar um novo plano de treino

### REQUEST

#### REQUEST BODY - application/json

```
{
  name*           string
  description*    string
  workoutIds*    {
  }
  daysPerWeek*   integer between 1 and 7
  totalWeeks*    integer
  iciqScoreMin*  integer >=1
  iciqScoreMax*  integer <=12
  ageMin*        integer >=12
  ageMax*        integer <=120
}
```

## RESPONSE

#### STATUS CODE - 201: Plano de treino criado com sucesso

##### RESPONSE MODEL - \*/\*

```
{
  id              integer
  name*           string
  description*    string
  workouts*      {
  }
  daysPerWeek*   integer
  totalWeeks*    integer
  iciqScoreMin*  integer
  iciqScoreMax*  integer
  ageMin*        integer
  ageMax*        integer
}
```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{
  code    integer
```

```
    origin  string
    message string
    details [string]
}
```

#### **STATUS CODE - 404: Not Found**

##### **RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### **STATUS CODE - 500: Internal Server Error**

##### **RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### **STATUS CODE - 501: Not Implemented**

##### **RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

---

# 12. PREFERÊNCIAS

Endpoints para gerenciar as preferências do usuário

## 12.1 GET /v1/preferences/notifications

### Obter Preferências de Notificação

Recupera as preferências de notificação de um usuário

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200: OK**

**RESPONSE MODEL - \*/\***

```
{
  reminderCalendar*      boolean
  reminderCalendarSchedule string
  reminderWorkout*       boolean
  reminderWorkoutSchedule string
  encouragingMessages*   boolean
  notificationToken*     string
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
}
```

```
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 12.2 PATCH /v1/preferences/notifications

### Definir Preferências de Notificação

Atualiza as preferências de notificação de um usuário

#### REQUEST

**REQUEST BODY - application/json**

```
{
  reminderCalendar*      boolean
  reminderCalendarSchedule string
  reminderWorkout*      boolean
  reminderWorkoutSchedule string
  encouragingMessages*  boolean
  notificationToken*    string
}
```

#### RESPONSE

**STATUS CODE - 200: OK**

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 12.3 GET /v1/preferences/accessibility

## Obter Preferências de Acessibilidade

Recupera as preferências de acessibilidade de um usuário

### REQUEST

No request parameters

### RESPONSE

**STATUS CODE - 200: OK**

**RESPONSE MODEL - \*/\***

```
{
  isBigFont*      boolean
  isHighContrast* boolean
  isDarkMode*     boolean
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 12.4 PATCH /v1/preferences/accessibility

### Definir Preferências de Acessibilidade

Atualiza as preferências de acessibilidade de um usuário

#### REQUEST

**REQUEST BODY - application/json**

```
{
  isBigFont*    boolean
  isHighContrast* boolean
  isDarkMode*   boolean
}
```

#### RESPONSE

**STATUS CODE - 200:** OK

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

```
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

---

# 13. QUESTIONÁRIO INICIAL

Endpoints para gerenciar questionário de onboarding

## 13.1 GET /v1/questions/onboard

### Obter perguntas iniciais

Recupera as perguntas iniciais do onboarding

#### REQUEST

No request parameters

#### RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - \*/\*

```
[{  
  Array of object:  
  id*      string  
  text*    string  
  type*    string  
  options [{  
    Array of object:  
    label* string  
    value* {  
    }  
  }]  
  min      integer  
  max      integer  
  step     integer  
  required boolean  
  placeholder string  
  hidden*  boolean  
  answer {  
  }  
}]
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

STATUS CODE - 401: Unauthorized

RESPONSE MODEL - application/json

```
{  
  code    integer
```

```
    origin string
    message string
    details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

## 13.2 POST /v1/questions/onboard

### Enviar respostas do onboarding

Envia as respostas para as perguntas do questionário de onboarding

#### REQUEST

**REQUEST BODY - application/json**

```
{
  DTO para envio das respostas do questionário de onboarding
  userId integer
}
```

```
    answers* {  
    }  
}
```

## RESPONSE

**STATUS CODE - 200:** Respostas enviadas com sucesso

**RESPONSE MODEL - \*/\***

```
{  
  profile* {  
    birthDate*    string  
    gender*       string  
    iciq3answer*  integer  
    iciq4answer*  integer  
    iciq5answer*  integer  
    iciqScore*    integer  
    urinationLoss* string  
  }  
  workoutPlan {  
    id            integer  
    plan* {  
      id          integer  
      name*       string  
      description* string  
      workouts* {  
      }  
      daysPerWeek* integer  
      totalWeeks*  integer  
      iciqScoreMin* integer  
      iciqScoreMax* integer  
      ageMin*      integer  
      ageMax*      integer  
    }  
    startDate*    string  
    endDate       string  
    totalProgress* integer  
    weekProgress* integer  
    currentWeek*  integer  
    nextWorkout   integer  
    lastWorkoutDate string  
    completed*    boolean  
  }  
}
```

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 409: Usuário já possui um plano de treino ativo**

**RESPONSE MODEL - \*/\***

```
{
  profile* {
    birthDate*  string
    gender*     string
    iciq3answer* integer
    iciq4answer* integer
    iciq5answer* integer
    iciqScore*  integer
    urinationLoss* string
  }
  workoutPlan {
    id          integer
    plan* {
      id          integer
      name*       string
      description* string
      workouts* {
      }
      daysPerWeek* integer
      totalWeeks* integer
      iciqScoreMin* integer
      iciqScoreMax* integer
      ageMin*     integer
      ageMax*     integer
    }
    startDate*  string
    endDate     string
    totalProgress* integer
  }
}
```

```
    weekProgress*   integer
    currentWeek*   integer
    nextWorkout    integer
    lastWorkoutDate string
    completed*     boolean
  }
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

---

# 14. RELATÓRIO

Endpoints para geração de relatórios de usuários

## 14.1 GET /v1/report

### Gerar relatório de usuário

Gera um relatório com base nas atividades do usuário dentro de um intervalo de datas especificado.

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*params	object	

#### RESPONSE

STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
{
  user* {
    name*   string
    age*    integer
    gender* string
  }
  history* [{
    Array of object:
    date*           string
    leakageLevel*  string
    eventsCount*   integer
    completedExercises* integer
    notesPreview   string
    urinationData [{
      Array of object:
      id           integer
      time*        string
      amount*      string
      leakage*     boolean
      reason       string
      urgency     boolean
    }]
    dayTitle*      string
    dayNumber*     integer
    isToday*       boolean
    today          boolean WRITE-ONLY
  }]
  generatedAt*    string
}
```

STATUS CODE - 400: Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

# 15. TREINO

Endpoints para gerenciamento de treinos

## 15.1 PUT /v1/exercise/workout/{id}

### Atualizar Treino

Atualizar um treino existente

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

##### REQUEST BODY - application/json

```
{
  name*           string   min:1 chars
  description*    string
  totalDuration* integer
  difficultyLevel* string   min:1 chars
  exerciseIds*   {
  }
}
```

#### RESPONSE

##### STATUS CODE - 200: OK

##### RESPONSE MODEL - \*/\*

```
{
  id             integer
  name*          string
  description*   string
  totalDuration* integer
  difficultyLevel* string
  exercises*    {
  }
  createdAt*    string
}
```

##### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

##### STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 15.2 DELETE /v1/exercise/workout/{id}

### Deletar Treino

Deletar um treino existente

### REQUEST

#### PATH PARAMETERS

---

NAME	TYPE	DESCRIPTION
------	------	-------------

---

*id	int64	
-----	-------	--

---

## RESPONSE

**STATUS CODE - 204:** Treino deletado com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

## RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 15.3 GET /v1/exercise/workout

### Listar Treinos

Listar todos os treinos disponíveis

### REQUEST

No request parameters

### RESPONSE

STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
[{
  Array of object:
  id            integer
  name*         string
  description*  string
  totalDuration* integer
  difficultyLevel* string
  exercises* {
  }
  createdAt*   string
}]
```

STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 15.4 POST /v1/exercise/workout

### Criar Treino

Criar um novo treino

### REQUEST

#### REQUEST BODY - application/json

```
{
  name*           string  min:1 chars
  description*    string
  totalDuration* integer
  difficultyLevel* string  min:1 chars
  exerciseIds*   {
  }
}
```

### RESPONSE

**STATUS CODE - 201: Treino criado com sucesso**

**RESPONSE MODEL - \*/\***

```
{
  id            integer
  name*        string
  description*  string
  totalDuration* integer
  difficultyLevel* string
  exercises* {
  }
  createdAt*   string
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
```

```
origin string
message string
details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

---

# 16. USUÁRIO

Endpoints para gerenciar autenticação de usuário e recuperação de senha

## 16.1 GET /v1/users/{id}

### Buscar Usuário por ID

Recupera os detalhes do usuário pelo ID do usuário

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

#### RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - \*/\*

```
{
  id          integer
  name*       string
  email*      string
  role*       string
  profilePictureUrl string
  curtidas*   integer
  salvos*     integer
  postagens*  integer
}
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 401: Unauthorized

RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 501: Not Implemented

#### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 16.2 PUT /v1/users/{id}

### Atualizar Usuário

Atualizar uma conta de usuário existente

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	int64	

##### REQUEST BODY - application/json

```
{
  name           string
  email          string
  profilePicture {
```

```

    id            integer
    url*          string   min:1 chars
    contentType*  string   min:1 chars
    contentType*  integer
    altText       string
    createdAt     string
  }
}

```

## RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - \*/\*

```

{
  id            integer
  name*         string
  email*        string
  profilePictureUrl  string
  role*         string
  profile {
    birthDate*  string
    gender*     string
    iciq3answer* integer
    iciq4answer* integer
    iciq5answer* integer
    iciqScore*  integer
    urinationLoss* string
  }
  preferences* {
    highContrast*  boolean
    bigFont*       boolean
    darkMode*      boolean
    reminderCalendar*  boolean
    reminderCalendarSchedule  string
    reminderWorkout*  boolean
    reminderWorkoutSchedule  string
    encouragingMessages*  boolean
    notificationToken*  string
  }
}

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

STATUS CODE - 401: Unauthorized

RESPONSE MODEL - application/json

```

{
  code    integer
  origin  string
}

```

```
message string
details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 16.3 POST /v1/users

### Criar Usuário

Criar uma nova conta de usuário

### REQUEST

**REQUEST BODY - application/json**

```
{
  name*           string  min:1 chars
  email*          string  min:1 chars
  password*       string  min:1 chars
}
```

```

profile {
  birthDate*      string
  gender*         string
  iciq3answer*   integer
  iciq4answer*   integer
  iciq5answer*   integer
  iciqScore*     integer
  urinationLoss* string
}
role {
  description*    string
  permissionLevel* integer
  reason*        string
  documentType   string
  documentValue  string
}
profilePicture {
  id              integer
  url*           string  min:1 chars
  contentType*   string  min:1 chars
  contentSize*   integer
  altText        string
  createdAt      string
}
preferences {
  highContrast*   boolean
  bigFont*        boolean
  darkMode*       boolean
  reminderCalendar* boolean
  reminderCalendarSchedule string
  reminderWorkout* boolean
  reminderWorkoutSchedule string
  encouragingMessages* boolean
  notificationToken* string
}
workoutPlan {
  planId*        integer
  startDate*     string
  endDate        string
  totalProgress* integer
  weekProgress*  integer
  currentWeek*   integer
  completed*     boolean
}
isPatientProfileValid* boolean
}

```

## RESPONSE

STATUS CODE - 201: Usuário criado com sucesso

RESPONSE MODEL - \*/\*

```

{
  id              integer
  name*          string
  email*         string
  profilePictureUrl string
}

```

```

role*                string
profile {
  birthDate*         string
  gender*            string
  iciq3answer*       integer
  iciq4answer*       integer
  iciq5answer*       integer
  iciqScore*         integer
  urinationLoss*     string
}
preferences* {
  highContrast*      boolean
  bigFont*           boolean
  darkMode*          boolean
  reminderCalendar*  boolean
  reminderCalendarSchedule string
  reminderWorkout*   boolean
  reminderWorkoutSchedule string
  encouragingMessages* boolean
  notificationToken* string
}
}

```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
  message string
  details [string]
}

```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```

{
  code    integer
  origin  string
}

```

```
    message string
    details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 16.4 POST /v1/users/workout/feedback

### Registra Feedback de Treino

Cria uma nova entrada de feedback de treino

#### REQUEST

**REQUEST BODY - application/json**

```
[{
  Array of object:
  exerciseId* integer
  workoutId*  integer
  rating*     integer
  evaluation* string
  comments    string
  completedAt string
}]
```

#### RESPONSE

**STATUS CODE - 204:** Feedback de treino registrado com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 16.5 POST /v1/users/workout/completion

### Registrar Conclusão de Treino

Registrar a conclusão de um treino

#### REQUEST

## REQUEST BODY - application/json

```
[{  
  Array of object:  
  workoutId*   integer  
  completedAt* string  
}]
```

## RESPONSE

### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
{  
  id           integer  
  plan*        string  
  description* string  
  startDate*   string  
  endDate      string  
  totalProgress* integer  
  weekProgress* integer  
  currentWeek* integer  
  completed*   boolean  
}
```

### STATUS CODE - 400: Bad Request

#### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

### STATUS CODE - 401: Unauthorized

#### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

### STATUS CODE - 403: Forbidden

#### RESPONSE MODEL - application/json

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

### STATUS CODE - 404: Not Found

#### RESPONSE MODEL - application/json

```
{  
  code    integer
```

```
    origin  string
    message string
    details [string]
  }
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 16.6 POST /v1/users/password/reset

### Alterar Senha

Altera a senha do usuário utilizando o OTP enviado para o e-mail

### REQUEST

**REQUEST BODY - application/json**

```
{
  otp*          string
  email*        string
  newPassword*  string
}
```

### RESPONSE

**STATUS CODE - 204:** Senha alterada com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

## 16.7 POST /v1/users/password/forgot

### Iniciar redefinição de senha

Envia um OTP para o e-mail do usuário para redefinição de senha

## REQUEST

### HEADER PARAMETERS

NAME	TYPE	DESCRIPTION
------	------	-------------

---

NAME	TYPE	DESCRIPTION
*x-user-email	string	

---

## RESPONSE

**STATUS CODE - 204:** OTP enviado com sucesso

**STATUS CODE - 400:** Bad Request

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 401:** Unauthorized

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 403:** Forbidden

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 404:** Not Found

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 500:** Internal Server Error

**RESPONSE MODEL - application/json**

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

**STATUS CODE - 501:** Not Implemented

## RESPONSE MODEL - application/json

```
{
  code      integer
  origin    string
  message   string
  details   [string]
}
```

## 16.8 POST /v1/users/login

### Login do Usuário

Autentica um usuário e retorna um token JWT com os detalhes do usuário

### REQUEST

#### REQUEST BODY - application/json

```
{
  email*    string
  password* string
}
```

### RESPONSE

#### STATUS CODE - 200: OK

#### RESPONSE MODEL - \*/\*

```
{
  token*      string
  user* {
    id          integer
    name*       string
    email*      string
    profilePictureUrl string
    role*       string
    profile {
      birthDate* string
      gender*    string
      iciq3answer* integer
      iciq4answer* integer
      iciq5answer* integer
      iciqScore* integer
      urinationLoss* string
    }
    preferences* {
      highContrast* boolean
      bigFont*      boolean
      darkMode*     boolean
      reminderCalendar* boolean
      reminderCalendarSchedule string
      reminderWorkout* boolean
      reminderWorkoutSchedule string
      encouragingMessages* boolean
      notificationToken* string
    }
  }
}
```

```
}
```

**STATUS CODE - 400: Bad Request**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 401: Unauthorized**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 403: Forbidden**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 404: Not Found**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 500: Internal Server Error**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string  
  details [string]  
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{  
  code    integer  
  origin  string  
  message string
```

```
    details [string]
  }
```

## 16.9 GET /v1/users/workout/plan

### Obter Plano de Treino

Recuperar o plano de treino de um usuário

#### REQUEST

No request parameters

#### RESPONSE

**STATUS CODE - 200:** Plano de treino recuperado com sucesso

**RESPONSE MODEL - \*/\***

```
{
  id                integer
  plan* {
    id              integer
    name*           string
    description*    string
    workouts* {
    }
    daysPerWeek*   integer
    totalWeeks*    integer
    iciqScoreMin*  integer
    iciqScoreMax*  integer
    ageMin*        integer
    ageMax*        integer
  }
  startDate*       string
  endDate          string
  totalProgress*   integer
  weekProgress*    integer
  currentWeek*     integer
  nextWorkout      integer
  lastWorkoutDate  string
  completed*       boolean
}
```

**STATUS CODE - 204:** Nenhum plano de treino ativo encontrado para o usuário

**RESPONSE MODEL - \*/\***

```
{
  id                integer
  plan* {
    id              integer
    name*           string
    description*    string
    workouts* {
    }
    daysPerWeek*   integer
    totalWeeks*    integer
    iciqScoreMin*  integer
  }
}
```

```
    icipScoreMax* integer
    ageMin*       integer
    ageMax*       integer
  }
  startDate*     string
  endDate        string
  totalProgress* integer
  weekProgress*  integer
  currentWeek*   integer
  nextWorkout    integer
  lastWorkoutDate string
  completed*     boolean
}
```

#### STATUS CODE - 400: Bad Request

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 401: Unauthorized

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 403: Forbidden

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 404: Not Found

##### RESPONSE MODEL - application/json

```
{
  code    integer
  origin  string
  message string
  details [string]
}
```

#### STATUS CODE - 500: Internal Server Error

##### RESPONSE MODEL - application/json

```
{
  code    integer
```

```
origin string
message string
details [string]
}
```

**STATUS CODE - 501: Not Implemented**

**RESPONSE MODEL - application/json**

```
{
  code integer
  origin string
  message string
  details [string]
}
```

---