



UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

Aplicativo móvel para encontrar grupos, parceiros e instrutores de atividades físicas.

Miguel Duarte Grieder

FLORIANÓPOLIS - SC

2024/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

APLICATIVO MÓVEL PARA ENCONTRAR GRUPOS, PARCEIROS E INSTRUTORES DE
ATIVIDADES FÍSICAS.

MIGUEL DUARTE GRIEDER

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de Informação

Orientador: Prof. Dr. Frank Augusto Siqueira.

FLORIANÓPOLIS - SC

2024/2

MIGUEL DUARTE GRIEDER

APLICATIVO MÓVEL PARA ENCONTRAR GRUPOS, PARCEIROS E
INSTRUTORES DE ATIVIDADES FÍSICAS.

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Frank Augusto Siqueira (Orientador Pedagógico)

Banca Examinadora

Jose Eduardo de Lucca

Robson Fernando Duda

SUMÁRIO

1 INTRODUÇÃO.....	17
1.1 CONTEXTUALIZAÇÃO.....	17
1.2 OBJETIVOS.....	18
1.2.1 Objetivos Gerais.....	18
1.2.2 Objetivos Específicos.....	19
1.3 MÉTODO DE PESQUISA.....	19
1.4 ORGANIZAÇÃO DO TEXTO.....	21
2 FUNDAMENTOS TECNOLÓGICOS.....	23
2.1 SISTEMAS MÓVEIS.....	23
2.1.1 iOS.....	23
2.1.2 Android.....	24
2.1.3 React Native.....	24
2.1.4 Expo.....	25
2.2 DESENVOLVIMENTO BACKEND.....	25
2.2.1 Clerk.....	25
2.2.2 Railway.....	26
2.2.3 Docker.....	26
2.2.4 MongoDB.....	27
2.2.5 Python.....	27
2.2.6 FastAPI.....	27
2.3 ENGENHARIA DE USABILIDADE.....	28

2.3.1 Protótipo de Interface.....	28
2.3.2 System Usability Scale.....	28
3 TRABALHOS RELACIONADOS.....	30
4 PROJETO DO APLICATIVO.....	33
4.1 REQUISITOS DO SISTEMA.....	33
4.1.1 Requisitos Funcionais.....	33
4.1.2 Requisitos Não Funcionais.....	35
4.2 HISTÓRIAS DE USUÁRIOS.....	36
4.3 CASOS DE USO.....	50
4.3.1 Usuário Casual.....	50
4.3.2 Treinador.....	52
4.3.3 Empresa.....	53
4.4 ARQUITETURA DO SISTEMA.....	54
4.5 PROTÓTIPOS DE INTERFACE.....	55
4.6 ESQUEMA DO BANCO DE DADOS.....	62
4.6.1 Dados de Atividade no MongoDB.....	63
4.6.2 Dados de usuário no Clerk.....	66
5 DESENVOLVIMENTO.....	69
5.1 APLICATIVO (FRONTEND).....	70
5.1.1 Tela Inicial.....	73
5.1.2 Criação e Atualização de Atividades.....	74
5.1.3 Atualização de Perfil do Usuário.....	76
5.1.4 Telas de Login.....	80

5.1.5 Listagem de Atividades.....	81
5.1.6. Filtro de Atividades.....	83
5.1.7 Listagem de Treinadores.....	84
5.1.8 Perfil do Usuário.....	85
5.1.9 Detalhes da Atividade.....	86
5.2 BACKEND.....	87
5.2.1 Endpoints de Atividades (/activities/)	89
5.2.2 Endpoints de Usuários (/users/)	92
5.2.3 Segurança e Autenticação.....	94
5.2.4 Integração com o frontend.....	97
5.2.5 Testes.....	98
5.3 INFRAESTRUTURA EM NUVEM.....	99
5.3.1 Hospedagem do Backend e Banco de Dados.....	101
5.3.2 Hospedagem do Frontend.....	101
5.3.3 Integração Contínua (CI) e a Entrega Contínua (CD).....	102
5.4 UTILIZAÇÃO DE RECURSOS DA NUVEM.....	105
5.5 MARKETING.....	107
6 AVALIAÇÃO.....	109
6.1 DEFINIÇÃO DA AVALIAÇÃO.....	109
6.3 ORGANIZAÇÃO DE PARTIDA DE FUTEBOL.....	114
6.4 USUÁRIO CASUAL BUSCANDO UMA ATIVIDADE DURANTE UMA VIAGEM...	116
6.5 RESULTADO DA AVALIAÇÃO.....	117
7 CONCLUSÃO.....	120

7.1.1 Novas Funcionalidades.....	121
7.1.2 Estudos de Impacto.....	122
REFERÊNCIAS BIBLIOGRÁFICAS.....	123

LISTA DE FIGURAS

Figura 1 - Casos de uso do Usuário Casual.....	50
Figura 2 - Casos de uso do Treinador.....	53
Figura 3 - Casos de uso da empresa.....	54
Figura 4 - Telas de Início(a) / Login(b) / Cadastro(c) / Cadastro - Informações e Cadastro(d) - Tipo de Conta(e).....	56
Figura 5 - Tela de Explorar(a) / Detalhes de atividade(b).....	57
Figura 6 - Tela de Entrar em Atividade(a) / Sair da Atividade(b).....	58
Figura 7 - Tela de Chats(a) / Perfil(b) / Configurações(c).....	59
Figura 8 - Tela de Mapa Interativo - Abrir Listagem(a) / Listagem(b) / Mapa(c) / Filtro(d)..	60
Figura 9 - Tela de Criar Atividade(a).....	61
Figura 10 - Tela de Listagem de Treinadores(a).....	62
Figura 11 - Documento de uma Atividade no MongoDB no Railway.....	65
Figura 12 - Metadados no Clerk de um usuário de teste.....	66
Figura 13 - Diagrama da infraestrutura completo para o aplicativo.....	69
Figura 14 - Tela inicial do Social Sweat.....	74
Figura 15 - Exemplos de avisos de erro ao usuário na criação de uma atividade.....	75
Figura 16 - Tela da página Criar Atividade.....	76
Figura 17 - Tela para definir nome, nome de usuário e foto de perfil.....	77
Figura 18 - Tela para inserir a data de nascimento.....	78
Figura 19 - Tela para definir tipo de usuário e esportes principais.....	79
Figura 20 - Tela de atualização de informações sociais.....	80

Figura 21 - Telas de Login(a) / Cadastro(b) / Verificação de código(c).....	81
Figura 22 - Telas de Listagem de atividades: Atividades(a) / Atividades criadas(b) / Minhas próximas Atividades (c).....	82
Figura 23 - Tela de Filtros de Atividades - estado inicial (a) / filtros aplicados (b).....	84
Figura 24 - Tela de Listagem de Treinadores.....	85
Figura 25 - Tela do perfil do usuário.....	86
Figura 26 - Tela de Detalhes da Atividade.....	87
Figura 27 - Swagger do microserviço.....	88
Figura 28 - Modelagem de classe de Atividade.....	89
Figura 29 - Modelagem de classe de modelos relacionados a atividade (a).....	90
Figura 30 - Modelagem de classe de modelos relacionados a atividade (b).....	90
Figura 31 - Modelagem de classe de UserModel.....	92
Figura 32 - Modelagem de classe de outros modelos relacionados ao usuário.....	93
Figura 33 - Middleware de autenticação.....	95
Figura 34 - Função para verificar se o user_id recebido é do mesmo usuário do JWT.....	95
Figura 35 - Modelo de atividade visto pelo Swagger do microserviço.....	96
Figura 36 - Modelo de usuário visto pelo Swagger do microserviço.....	97
Figura 37 - Exemplo de função no frontend de chamada ao microserviço: fetchUsers.....	98
Figura 38 - Retorno da execução dos testes presentes no tox.....	99
Figura 39 - Diagrama de atualização dos serviços em nuvem.....	100
Figura 40 - Visualização do projeto Social Sweat App Expo no Expo Go.....	102
Figura 41 - Visualização dos diferentes envios a produção do microserviço no Railway.....	

Figura 42 - Etapa CI no Gitlab executada com sucesso.....	103
Figura 43 - Conjunto de etapas CI no Gitlab e Railway executadas com sucesso.....	104
Figura 44 - Etapa CI no Gitlab com problema.....	104
Figura 45 - Visualização de um envio do microserviço que não foi a produção no Railway.	105
Figura 46 - Métricas de utilização de recursos do volume MongoDB no Railway.....	106
Figura 47 - Métricas de utilização de recursos do microserviço no Railway.....	107
Figura 48 - Perfil do Social Sweat no Instagram.....	108
Figura 49 - Documento de incorporação da startup no estado de Delaware.....	120

LISTA DE QUADROS

Quadro 1 - Requisitos Funcionais.....	33
Quadro 2 - Requisitos Não-Funcionais.....	35
Quadro 3 - História de Usuário 01.....	38
Quadro 4 - História de Usuário 02.....	39
Quadro 5 - História de Usuário 03.....	40
Quadro 6 - História de Usuário 04.....	41
Quadro 7 - História de Usuário 05.....	42
Quadro 8 - História de Usuário 06.....	43
Quadro 9 - História de Usuário 07.....	44
Quadro 10 - História de Usuário 08.....	45
Quadro 11 - História de Usuário 09.....	46
Quadro 12 - História de Usuário 10.....	47
Quadro 13 - História de Usuário 11.....	48
Quadro 14 - História de Usuário 12.....	49

LISTA DE TABELAS

Tabela 1 - Participante 1: Busca por Personal Trainer, Participante Cadastrado.....	112
Tabela 2 - Participante 2: Busca por Personal Trainer, Participante Novo.....	113
Tabela 3 - Participante 3: Organização de Partida de Futebol, Participante Cadastrado.	114
Tabela 4 - Participante 4: Organização de Partida de Futebol, Participante Novo.....	115
Tabela 5 - Participante 5: Usuário casual buscando uma atividade durante uma viagem, Participante Cadastrado.....	116
Tabela 6 - Participante 6: Usuário casual buscando uma atividade durante uma viagem, Participante Novo.....	117
Tabela 7 - Resultado final do SUS.....	118

LISTA DE ABREVIATURAS E SIGLAS

CLI	Command-Line Interface
API	Application Programming Interface
JSON	JavaScript Object Notation
JWT	JSON Web Token
SMS	Short Message Service
ID	Identification
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
SUS	System Usability Scale
ENUM	Enumeration
BSON	Binary JSON

RESUMO

Com o advento dos computadores e *smartphones*, as pessoas tornaram-se cada vez mais sedentárias (YANG, 2019). A internet e as redes sociais, que deveriam "conectar o mundo" e tornar a comunicação entre amigos e familiares instantânea, acabaram por ter um efeito quase contrário, criando uma realidade virtual própria e aumentando a sensação de solidão (HUNT, 2018).

Este Trabalho de Conclusão de Curso propõe o desenvolvimento do aplicativo móvel Social Sweat para sistemas iOS e Android, com o intuito de facilitar a conexão entre pessoas. O aplicativo oferece uma plataforma interativa e divertida para encontrar grupos, parceiros, instrutores e locais para praticar atividades físicas. A solução apresenta uma abordagem com um mapa interativo, onde usuários casuais podem visualizar e participar de atividades de interesse, como academias, artes marciais e partidas de futebol. Além disso, os usuários têm a capacidade de criar eventos, como jogos ou grupos para trilhas, promovendo encontros para a prática de atividades físicas.

O aplicativo também oferece suporte a instrutores, guias e *personal trainers*, permitindo que criem atividades do tipo sessão, que são como aulas que acontecem semanalmente. Eles também podem ser encontrados por outros usuários na aba de treinadores, onde suas informações estarão visíveis. Para empresas que possuem locais fixos, como academias, piscinas (natação), artes marciais e *CrossFit*, é possível adicionar o local permanentemente no aplicativo.

O método de desenvolvimento do aplicativo inclui a implementação de recursos como criação de perfis, integração de mapas interativos e sistema de atividades. Espera-se que o aplicativo promova a socialização, facilitando a prática de atividades

físicas em grupo, melhorando a experiência do usuário e incentivando um estilo de vida mais saudável.

Palavras-chave: aplicativo móvel, atividades físicas, socialização, treinadores.

ABSTRACT

With the advent of computers and smartphones, people have become increasingly sedentary (YANG, 2019). The internet and social media, which were supposed to "connect the world" and make communication between friends and family instant, ended up having an almost opposite effect, creating a virtual reality of their own and increasing feelings of loneliness (HUNT, 2018).

This Graduation Project proposes the development of the application Social Sweat for iOS and Android systems, with the goal of facilitating connections between people. The app provides an interactive and fun platform to find groups, partners, instructors, and places to engage in physical activities. It offers an innovative approach with an interactive map, where casual users can view and participate in activities of interest, such as gyms, martial arts, and soccer matches. Additionally, users have the ability to create events, such as games or hiking groups, promoting meet-ups for physical activities.

The app also supports instructors, guides, and personal trainers, allowing them to create session-type activities, similar to classes that occur weekly. They can also be found by other users in the trainer tab, where their information will be visible. For businesses with fixed locations, such as gyms, swimming pools, martial arts studios, and CrossFit, it is possible to add their location permanently to the app.

The app development method includes implementing features such as profile creation, interactive map integration, and an activity system. The app is expected to promote socialization, facilitate group physical activities, enhance user experience, and encourage a healthier lifestyle.

Keywords: mobile app, physical activities, socialization, trainer.

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O sedentarismo é um problema crescente na sociedade moderna, com graves consequências para a saúde física e mental, estando associado a um aumento do risco de várias doenças crônicas, como obesidade, diabetes e hipertensão (TEODOR, 2018). Além disso, o sedentarismo, juntamente com a falta da prática de atividades físicas pode levar à solidão, pois dificulta o contato social e a participação em atividades coletivas (BUENO-ANTEQUERA, 2023). Um dos fatores contribuintes para esse aumento são as novas tecnologias, como computadores, *smartphones*, redes sociais e *internet* (YANG, 2019), que consomem o tempo que normalmente seria destinado à prática de atividades físicas.

As plataformas e redes sociais específicas para o conteúdo *fitness*, como Strava e Nike Run Club, têm êxito na criação de perfis pessoais abrangentes, possibilitando recordar e acessar o histórico de exercícios, participar de desafios e integrar grandes comunidades e grupos. Contudo, carecem da facilidade de encontrar locais, parceiros e instrutores de maneira regional, simples e intuitiva. A navegação nessas plataformas, semelhante às redes sociais convencionais, dificulta a descoberta de novas pessoas e de eventos locais, pois o que o usuário visualiza depende de suas buscas e do algoritmo de recomendação de conteúdos. No caso do Strava, o usuário tem acesso a recomendações dos principais grupos e atividades da sua região, mas não há uma forma de organizar grupos diretamente nestas atividades.

Diante disso, este projeto visa preencher essa lacuna por meio de um aplicativo centrado em encontrar locais, atividades, pessoas, grupos e instrutores próximos, com o objetivo imediato da prática desejada. A principal forma de navegação será um mapa, no

qual haverá as atividades. Nele, os usuários poderão adicionar temporariamente eventos, que também serão identificados no mapa, como um torneio de futebol, informando a data, endereço e outras informações. Dessa forma, os participantes poderão se unir ao evento, interagir com outros interessados e combinar a prática da atividade desejada.

Com esse propósito, espera-se proporcionar aos usuários uma plataforma simplificada, centrada na realização de atividades físicas em grupo ou com profissionais qualificados. Por meio dessas práticas, busca-se diminuir os índices de sedentarismo e aprimorar a condição física e socialização, promovendo benefícios para a saúde tanto física quanto mental.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

Este projeto tem como objetivo principal o desenvolvimento de um aplicativo móvel destinado a iniciantes, entusiastas, profissionais de atividades físicas e empresas do ramo que possuam academias, centros de treinamento e locais onde são realizadas essas atividades. O aplicativo visa facilitar a conexão entre esses usuários, permitindo que encontrem grupos para a prática de suas atividades, promovendo uma abordagem saudável e coletiva, com ênfase na socialização. Além disso, oferecerá a oportunidade para instrutores e *personal trainers* disponibilizarem seus serviços. A navegação no aplicativo será inovadora, utilizando um mapa interativo que destaca os locais propícios para a realização das atividades físicas.

1.2.2 Objetivos Específicos

Para que o objetivo geral seja alcançado, faz-se necessário que os seguintes objetivos específicos sejam alcançados:

1. Desenvolver um protótipo da interface do aplicativo e realizar uma pesquisa sobre possíveis problemas de usabilidade e segurança, adaptando a interface conforme o necessário.
2. Escolher a estrutura de desenvolvimento, como tecnologias, tipos de banco de dados, infraestrutura *cloud*, bibliotecas e ferramentas de versionamento de código.
3. Desenvolver tanto o aplicativo *mobile* quanto o *backend*, mantendo os serviços online.
4. Disponibilizar o aplicativo nas plataformas iOS e Android através da plataforma Expo Go e avaliar seu funcionamento e uso de serviços online.
5. Disponibilizar os idiomas português e inglês, podendo ser trocado pelo usuário.

1.3 MÉTODO DE PESQUISA

Quanto à natureza do trabalho a ser conduzido, este estudo se enquadra como uma pesquisa aplicada (ou tecnológica), visando a criação de produtos inovadores com propósitos imediatos, fundamentados em conhecimentos pré-existentes. Em relação aos objetivos da pesquisa, caracteriza-se como um projeto de desenvolvimento, buscando a criação prática de um produto, em contraste com uma pesquisa puramente teórica. A metodologia deste projeto de pesquisa é estruturada em distintas etapas, que incluem:

Etapa 1: levantamento de requisitos, recursos iniciais, arquitetura e características básicas para o aplicativo.

- A1.1: definir as histórias de usuário, requisitos funcionais e não funcionais.
- A1.2: identificar os recursos técnicos necessários (hardware, software e serviços).
- A1.3: elaborar a arquitetura inicial do sistema, estabelecendo as características básicas do aplicativo.

Etapa 2: desenvolver um *template* da interface do aplicativo, avaliando sua usabilidade.

- A2.1: definir o guia de *design* do aplicativo móvel.
- A2.2: produzir um protótipo da interface do aplicativo no Figma.
- A2.3: realizar um teste na interface.
- A2.4: analisar os resultados e corrigir pontos negativos.

Etapa 3: Realizar uma pesquisa sobre problemas de segurança aos usuários, adaptando o protótipo da interface conforme o necessário.

- 3.1: buscar sobre problemas de segurança e precauções ao desenvolver um aplicativo *online*.
- A3.2: analisar os resultados da busca com o protótipo desenvolvido.
- A3.3: readaptar a interface para corrigir os riscos presentes.

Etapa 4: escolher a estrutura de desenvolvimento, como tecnologias, banco de dados, infraestrutura de *cloud*, bibliotecas e versionamento de código.

- A4.1: pesquisar e escolher as tecnologias necessárias para atender ao protótipo da interface para todo o aplicativo.

- A4.2: esquematizar os fluxos de dados.
- A4.3: preparar casos de testes automatizados de *software*.

Etapa 5: desenvolver o aplicativo *mobile*.

- A5.1: criar as telas básicas do aplicativo, com a navegação entre elas.
- A5.2: realizar integração com o *backend*.
- A5.3: finalizar o desenvolvimento passando nos testes de *software* planejados.

Etapa 6: desenvolver o *backend*, APIs e infraestrutura necessária.

- A6.1: configurar infraestrutura *cloud* e de versionamento de código.
- A6.2: desenvolver APIs conforme o necessário para atender o aplicativo, juntamente com testes automatizados.
- A6.3: finalizar o desenvolvimento ao suportar todas funções do aplicativo.

Etapa 7: realizar o lançamento do aplicativo através do Expo Go, que deve suportar Android e iOS, e implementar uma estratégia de *marketing*.

- A7.1: definir uma estratégia de *marketing* para o lançamento do aplicativo.
- A7.2: disponibilizar o aplicativo no Expo Go.
- A7.3: realizar a estratégia de *marketing*.
- A7.4: monitorar a utilização de recursos do aplicativo e os resultados do *marketing*.

1.4 ORGANIZAÇÃO DO TEXTO

Este trabalho está estruturado nos seguintes capítulos:

FUNDAMENTOS TECNOLÓGICOS: Apresenta as principais tecnologias utilizadas na realização do projeto, divididas em três subtópicos: Sistemas Móveis, Desenvolvimento Backend e Engenharia de Usabilidade.

TRABALHOS RELACIONADOS: Aborda artigos, projetos de TCC e outros aplicativos similares, discutindo suas diferenças em relação a este projeto.

PROJETO DO APLICATIVO: Detalha a preparação para o desenvolvimento do aplicativo, incluindo requisitos, histórias de usuário, casos de uso, arquitetura do sistema, protótipo de interface e esquema de banco de dados.

DESENVOLVIMENTO: Descreve o desenvolvimento do aplicativo, abrangendo a parte *frontend*, *backend*, serviços em nuvem, integração contínua, entrega contínua e estratégias de *marketing*.

AVALIAÇÃO: Relata os testes realizados por participantes no aplicativo, incluindo uma avaliação com o teste SUS, e comenta os resultados obtidos.

CONCLUSÃO: Apresenta as conclusões do projeto, destacando os resultados alcançados.

2 FUNDAMENTOS TECNOLÓGICOS

Este capítulo apresenta as principais tecnologias utilizadas no planejamento, desenvolvimento e conclusão do projeto, organizadas em três tópicos conforme o tipo de tecnologia. São explicados desde os sistemas operacionais, linguagens de programação e *frameworks* até os serviços em nuvem empregados.

2.1 SISTEMAS MÓVEIS

Mesmo antes do lançamento do iPhone, já haviam celulares que realizavam operações avançadas como enviar *e-mails*, acessar a internet móvel e rodar jogos, e embora num primeiro momento fossem menos eficientes, a mudança de paradigma trazida pelo iPhone acabou mudando completamente o mercado (HRISTOV, 2017). Com o passar do tempo, o sistema dos novos *smartphones* se tornaram parte fundamental do dispositivo, principalmente com a loja de aplicativos, onde se tornou comum a instalação de aplicativos de terceiros no aparelho (FAAS, 2010).

2.1.1 iOS

O iOS é o sistema operacional disponível nos dispositivos móveis da Apple Inc. Sendo reconhecido por sua interface de usuário intuitiva e elegante, além de ser altamente otimizado para o *hardware* proprietário também desenvolvido pela Apple. O iOS é construído sobre o núcleo do sistema operacional Darwin, derivado do Unix, e suas aplicações nativas são desenvolvidas nas linguagens de programação Swift e Objective-C (APPLE INC., 2020).

2.1.2 Android

O Android é um sistema operacional móvel mantido pela Open Handset Alliance (OHA), mas com a Google sendo a principal desenvolvedora. É amplamente utilizado em dispositivos de diversas fabricantes, sendo possível até rodar em *desktops*, por possuir seu sistema de código aberto e baseado no *kernel* do Linux, permitindo maior flexibilidade e customização se comparado ao iOS (GOOGLE, 2024).

As linguagens de programação nativas do Android são o Java e Kotlin, e ambas são compiladas para *bytecode* Java e executadas no Android Runtime (ART) nas versões mais novas do Android. O ART utiliza uma combinação de compilação Ahead-Of-Time (AOT) e Just-In-Time (JIT), o que permite uma execução eficiente dos aplicativos. Portanto, em termos de execução, ambas as linguagens funcionam de maneira semelhante no Android (BOSE, 2018).

2.1.3 React Native

O React Native foi desenvolvido e é mantido pela Meta (desde quando ainda chamava-se Facebook), sendo um *framework* usado principalmente para o desenvolvimento de aplicativos móveis para iOS e Android. Utiliza React, uma biblioteca para construção de interfaces de usuário (FOJTÍK, 2023). O uso deste *framework* facilita o desenvolvimento de aplicativos para *smartphones*, pois possibilita usar uma base de código compartilhada para executar em diversos sistemas operacionais, não sendo necessário criar uma versão em cada linguagem nativa das plataformas, o que aumenta a eficiência e reduz o tempo de desenvolvimento. Isso é possível devido à representação de árvore de elementos de interface do usuário em JavaScript, que é compilada em tempo

real de execução para código nativo específico de cada plataforma, sendo que este trabalho fica por conta do *framework* (PAUL, 2019).

2.1.4 Expo

Expo é uma ferramenta que simplifica o desenvolvimento e a implantação de projetos React Native. Ela permite que os desenvolvedores subam seus aplicativos na nuvem de forma gratuita, facilitando o compartilhamento de *apps* sem a necessidade de passar pelas lojas de aplicativos móveis. Com o Expo, é possível testar e distribuir aplicações diretamente aos usuários ou à equipe de desenvolvimento através de *links* ou códigos QR, o que agiliza o processo de *feedback* iterativo no desenvolvimento móvel.

2.2 DESENVOLVIMENTO *BACKEND*

Em um aplicativo, o desenvolvimento *backend* é construir a lógica de negócios, processamento de dados, interações com bancos de dados, interligar APIs, serviços *cloud* e mais. Esses serviços não são armazenados e processados no dispositivo do cliente, ficando em servidores e bases de dados específicos (QUVVATOV, 2024).

2.2.1 Clerk

Clerk é uma plataforma que oferece soluções completas de autenticação e gerenciamento de usuários para aplicações *web* e móveis. Com suporte para cadastro de diversas formas, componentes pré-construídos e APIs flexíveis, o Clerk simplifica a implementação de sistemas de *login* seguros e modernos e permite que os

desenvolvedores integrem funcionalidades de autenticação de forma rápida e eficiente, sem se preocupar com a infraestrutura subjacente.

2.2.2 Railway

Railway é uma plataforma de infraestrutura em nuvem que permite aos desenvolvedores provisionar recursos e implantar serviços próprios na nuvem. Sua facilidade de uso destaca-se com uma interface intuitiva estilo Canva; a Railway abstrai a complexidade do gerenciamento de servidores e bancos de dados, permitindo que os desenvolvedores consigam colocar serviços como bancos de dados MongoDB e subir microsserviços FastAPI em contêineres Docker em segundos.

2.2.3 Docker

Um contêiner é um pacote de *software* leve, independente e executável que inclui tudo o que é necessário para executar uma aplicação, como código, bibliotecas e ferramentas do sistema. Os contêineres fornecem um ambiente de execução consistente, garantindo que a aplicação possa ser executada de forma confiável, independentemente da configuração do sistema (YEPURI, 2023).

A containerização transformou o desenvolvimento de *software* e implantação com Docker sendo um dos mais amplamente adotados plataformas de código aberto para construir, enviar e executar aplicativos distribuídos. Esses contêineres são muito úteis para aplicações com arquiteturas baseadas em microsserviços, pois tornam muito simples o empacotamento (*packaging*) e a implantação (*deploy*) deles em produção (YEPURI, 2023).

2.2.4 MongoDB

O MongoDB é um banco de dados NoSQL orientado a documentos, que ganhou uma grande adoção mundial por causa de sua arquitetura escalável e flexível. Ele armazena dados em formato BSON, semelhante ao JSON (DUBEY, 2023).

2.2.5 Python

Python é uma linguagem de programação de alto nível, conhecida por sua simplicidade de desenvolvimento, legibilidade e grande número de bibliotecas de terceiros. Seus principais usos são análise de dados, desenvolvimento *web*, *machine learning* e automação (DHRUV, 2020).

2.2.6 FastAPI

O FastAPI é uma estrutura moderna e de alto desempenho para a construção de APIs na linguagem Python com agilidade e eficiência. Ele se destaca como uma das estruturas de API mais rápidas, tanto no aprendizado mais suave quanto na velocidade de execução, em comparação com *frameworks* como Flask e Django. Outros grandes pontos fortes são o assincronismo, uso extensivo de dicas de tipo do Python, documentação no formato OpenAPI automática e a integração de modelos de dados (MA, 2024).

Além de seu desempenho rápido, FastAPI reduz significativamente os erros humanos, em aproximadamente 40%, e acelera o desenvolvimento de funcionalidades, permitindo que os desenvolvedores concretizem suas ideias rapidamente (ABDULAZEEZ, 2022).

2.3 ENGENHARIA DE USABILIDADE

A engenharia de usabilidade é usada principalmente para melhorar a eficiência e a eficácia das interfaces. O processo de engenharia de usabilidade envolve iterar no *design* da UI e também realizar testes para garantir que a informação seja claramente comunicada e esteticamente agradável, ao mesmo tempo em que direciona a atenção do usuário para os componentes mais importantes (DESIDERIA, 2020).

2.3.1 Protótipo de Interface

O protótipo de interface é um modelo detalhado e funcional da interface do usuário que é desenvolvido para testar e refinar o *design* antes de desenvolver um aplicativo ou programa. Os *designers* criam esse protótipo de alta fidelidade que já possui um *layout* bem equilibrado da interface do usuário, que é então submetido a testes com usuários, onde participantes interagem com esse protótipo, realizando tarefas específicas. Esse processo de desenvolver um protótipo e testá-lo é crucial para coletar dados sobre a atenção dos usuários e validar a eficácia do *design* inicial, permitindo ajustes e otimizações subsequentes baseadas em *feedback* real (DESIDERIA, 2020).

2.3.2 System Usability Scale

O *System Usability Scale* (SUS) é uma ferramenta amplamente utilizada para medir a usabilidade de sistemas. Desenvolvido em 1986, o SUS é um questionário composto por 10 itens, cada um avaliado em uma escala Likert de 5 pontos que varia de "discordo fortemente" a "concordo fortemente". Os itens são equilibrados entre declarações

positivas e negativas, e a pontuação final varia de 0 a 100. O SUS é conhecido por ser uma abordagem "rápida e suja" para avaliar a usabilidade, oferecendo uma visão geral confiável sobre a facilidade de uso de um sistema (HYZY, 2022). Estudos mostram que uma pontuação média de 68 é considerada um ponto de referência útil.

3 TRABALHOS RELACIONADOS

O artigo "*HealthyTogether: exploring social incentives for mobile fitness applications*" de Yu Chen (2014) investiga a eficácia de diferentes incentivos sociais no aumento da atividade física dos usuários por meio de um aplicativo móvel. O estudo envolveu 18 duplas de usuários, que foram divididas em três grupos distintos: cooperação, híbrido e competição. Os resultados indicaram que o grupo de cooperação apresentou o maior aumento na atividade física, seguido pelo grupo híbrido, enquanto o grupo de competição foi eficaz apenas entre usuários com desempenho físico equivalente. A análise das mensagens trocadas entre os participantes revelou que o apoio mútuo e a comunicação positiva foram cruciais para a motivação. O estudo destaca a importância dos incentivos sociais e da comunicação para o *design* de aplicativos de exercícios que visam promover a atividade física de forma eficaz.

O trabalho "*Personal Activity: Um aplicativo para auxiliar profissionais de Educação Física nas atividades de Personal Trainer*" de Morgana Leite (2014) descreve o desenvolvimento de um aplicativo com o objetivo de facilitar as atividades dos profissionais de educação física. O aplicativo inclui funcionalidades como a inserção de alunos, gerenciamento de treinos e avaliações físicas, controle de pagamento e gerenciamento de horários. O desenvolvimento do aplicativo envolveu a coleta de dados junto com profissionais da área para identificar funcionalidades essenciais. Esse trabalho oferece uma ferramenta útil e acessível para *personal trainers*, possibilitando o gerenciamento de suas atividades diretamente de um *smartphone* Android, mesmo sem conexão à *internet*.

Com vários aplicativos de *fitness* populares, como Strava, Nike Training Club e Apple Health, o mercado é dominado por plataformas que se destacam em vários aspectos de monitoramento de condicionamento físico individual, planos de treino

personalizados, guias, agregação de dados de saúde e interação social. Existem algumas categorias de aplicativos que adotam abordagens diferentes para serem um bom aplicativo de *fitness*, eles não necessariamente competem diretamente entre si, mas ainda existem muitas semelhanças e os mesmos recursos entre eles.

O Strava, conhecido por seu estilo de mídia social e ampla variedade de recursos de rastreamento. O maior diferencial é a rede social pública mais ativa do mercado, com grupos comunitários locais espalhados por diversas cidades e países, ultrapassando 100 milhões de usuários em todo o mundo. Cada pessoa tem uma página pessoal, pode participar de grupos, comunidades, fazer desafios e encontrar lugares populares. Existem também aplicativos semelhantes, como o AllTrails, que se concentram mais em corrida, caminhada e ciclismo. Ambos os aplicativos têm uma página de mapa para encontrar locais de atividades próximas, mas que é limitada para ver detalhes do local, avaliações, caminho e previsão do tempo. Esses aplicativos, diferentemente dos citados anteriormente, possuem uma versão gratuita e outra paga. Com o nível gratuito você ainda pode acessar as funcionalidades básicas, mas o acesso às funções mais avançadas é permitido somente na versão paga.

Em contraste com estes aplicativos estabelecidos, a solução proposta segue um caminho divergente ao introduzir uma dimensão social e geográfica única à experiência das aplicações de *fitness*. Enquanto outras aplicações se concentram principalmente em realizações individuais, treinos estruturados e numa integração social como as principais aplicações de redes sociais, a inovação reside no aspecto comunitário, permitindo aos utilizadores ligarem-se com base na sua proximidade geográfica e interesse partilhado em atividades físicas. Essa camada social transforma o aplicativo em uma plataforma dinâmica onde os usuários podem descobrir e interagir perfeitamente com atividades e outros entusiastas do *fitness* em tempo real e de uma forma única. Mesmo em aplicativos

como Strava ou AllTrails, não é um convite para ingressar em outro grupo para realizar uma atividade – é semelhante a um grupo do Facebook.

4 PROJETO DO APLICATIVO

Neste capítulo será detalhado o planejamento para o desenvolvimento do aplicativo, com os requisitos e histórias que o aplicativo deve atender, e também há a arquitetura do sistema, banco de dados e o protótipo de interface necessária detalhada.

4.1 REQUISITOS DO SISTEMA

Os requisitos funcionais foram definidos através das funcionalidades necessárias para o aplicativo proposto neste projeto, tendo as histórias de usuário completas no subcapítulo 4.2. Para os requisitos não funcionais, foram considerados requisitos mínimos de uma aplicação *online* e o uso de tecnologias descritas no capítulo 2.

4.1.1 Requisitos Funcionais

Quadro 1 - Requisitos Funcionais.

#	Título do requisito	Descrição
RF01	Suporte a diferentes tipos de usuário	O sistema deve suportar diferentes tipos de usuário (Usuários Casuais, Treinadores, Empresas) com privilégios distintos.
RF02	Busca de atividades	O sistema deve permitir que os usuários busquem atividades utilizando palavras-chave e filtros diversos, como tipo de atividade (Evento, Sessão, Local), esporte e outros critérios.
RF03	Listagem de atividades correspondentes	O sistema deve exibir uma lista de atividades correspondentes aos critérios de busca, permitindo aos usuários visualizar detalhes e selecionar atividades de interesse.

RF04	Visualização de detalhes da atividade	O sistema deve permitir que os usuários visualizem detalhes de uma atividade, incluindo data, hora, local, descrição, número de participantes e informações sobre o organizador.
RF05	Criação de atividades	O sistema deve permitir a criação de atividades de acordo com o tipo de usuário e suas permissões: Usuário Casual: criar atividades do tipo Evento; Treinador: criar atividades dos tipos Evento, Sessão; Empresa: criar atividades dos tipos Evento, Sessão e Local.
RF06	Edição de atividades	O sistema deve permitir que os usuários editem atividades que tenham criado, possibilitando a atualização de informações como data, hora, local e descrição.
RF07	Participação em atividades	O sistema deve permitir que os usuários participem e saiam de atividades, atualizando automaticamente a lista de participantes.
RF08	Busca de treinadores	O sistema deve permitir que os usuários busquem treinadores, aplicando filtros como esporte principal.
RF09	Visualização de perfil de treinadores	O sistema deve permitir que os usuários visualizem o perfil de treinadores, incluindo esportes principais, biografia e informações de contato.
RF10	Criação e atualização de perfil	O sistema deve permitir que os usuários criem e atualizem seus perfis pessoais, definindo esportes principais, tipo de usuário, foto, nome de usuário, biografia e informações de contato.
RF11	Alteração de idioma da interface	O sistema deve permitir que os usuários alterem o idioma da interface entre português e inglês.

RF12	<i>Logout</i>	O sistema deve permitir que os usuários façam <i>logout</i> de suas contas de forma segura.
------	---------------	---

Fonte: elaborado pelo autor.

4.1.2 Requisitos Não Funcionais

Quadro 2 - Requisitos Não-Funcionais.

#	Título do requisito	Descrição
RNF01	Transmissão de dados segura	O sistema deve garantir transmissão de dados segura.
RNF02	Autenticação segura	O sistema deve utilizar mecanismos de autenticação seguros, como <i>e-mail</i> , senha, código verificador e <i>tokens</i> JWT, para proteger o acesso às contas dos usuários.
RNF03	Controle de acesso baseado em papéis	O sistema deve implementar controle de acesso baseado em papéis, garantindo que cada tipo de usuário acesse apenas as funcionalidades permitidas.
RNF04	Compatibilidade com iOS e Android	O sistema deve ser compatível com as versões mais recentes dos sistemas operacionais iOS e Android, proporcionando uma experiência consistente em diferentes dispositivos.
RNF05	Arquitetura de microsserviços	O sistema deve ser implementado usando microsserviços com FastAPI, operando em contêineres Docker para escalabilidade e manutenção eficiente.
RNF06	Suporte a múltiplos	O sistema deve suportar os idiomas

	idiomas	português e inglês, permitindo que os usuários escolham sua preferência de idioma.
--	---------	--

Fonte: elaborado pelo autor.

4.2 HISTÓRIAS DE USUÁRIOS

As histórias de usuário fornecem uma forma simples e direta de capturar o que o cliente realmente deseja ao descrever funcionalidades sob a perspectiva de quem as utiliza (Pressman, 2014). Elas consistem em três elementos principais (Cohn, 2009):

- **Descrição escrita:** utilizada para planejamento e como lembrete das funcionalidades a serem implementadas.
- **Conversas sobre a história:** servem para detalhar aspectos específicos e esclarecer dúvidas.
- **Testes:** ajudam a transmitir e documentar detalhes, além de determinar quando a história está completa.

Neste contexto, as histórias de usuário neste trabalho são definidas pelos seguintes componentes:

- **Identificador único (US00):** código único para cada história de usuário, atualizado conforme novas histórias são adicionadas.
- **Critérios de aceite:** conjunto de condições que a funcionalidade deve cumprir para ser realizado o caos.
- **Requisitos:** requisitos funcionais ou não funcionais que a funcionalidade descrita na história deve atender para satisfazer as necessidades necessárias.

- **Protótipo de tela e Resultado:** nome do protótipo de tela para esta funcionalidade, que está visível no subcapítulo 4.5 PROTÓTIPOS DE INTERFACE.

O resultado corresponde a esta tela no aplicativo desenvolvido.

Quadro 3 - História de Usuário 01.

US01 - Como novo usuário, eu preciso me registrar especificando o tipo de usuário (Usuário Casual, Treinador, Empresa) para que eu tenha acesso às funcionalidades apropriadas.
Critérios de aceite: US01.01 - O usuário deve poder selecionar seu tipo de usuário durante o registro. US01.02 - Cada tipo de usuário possui privilégios distintos que limitam ou concedem acesso a determinadas funcionalidades. US01.03 - O sistema deve armazenar o tipo de usuário associado ao perfil registrado.
Requisitos: RF01
Protótipos de tela: <u>Cadastro - Tipo de Conta</u> Resultado:

Fonte: elaborado pelo autor.

Quadro 4 - História de Usuário 02.

US02 - Como usuário, eu preciso buscar atividades utilizando palavras-chave e filtros para encontrar atividades de meu interesse.

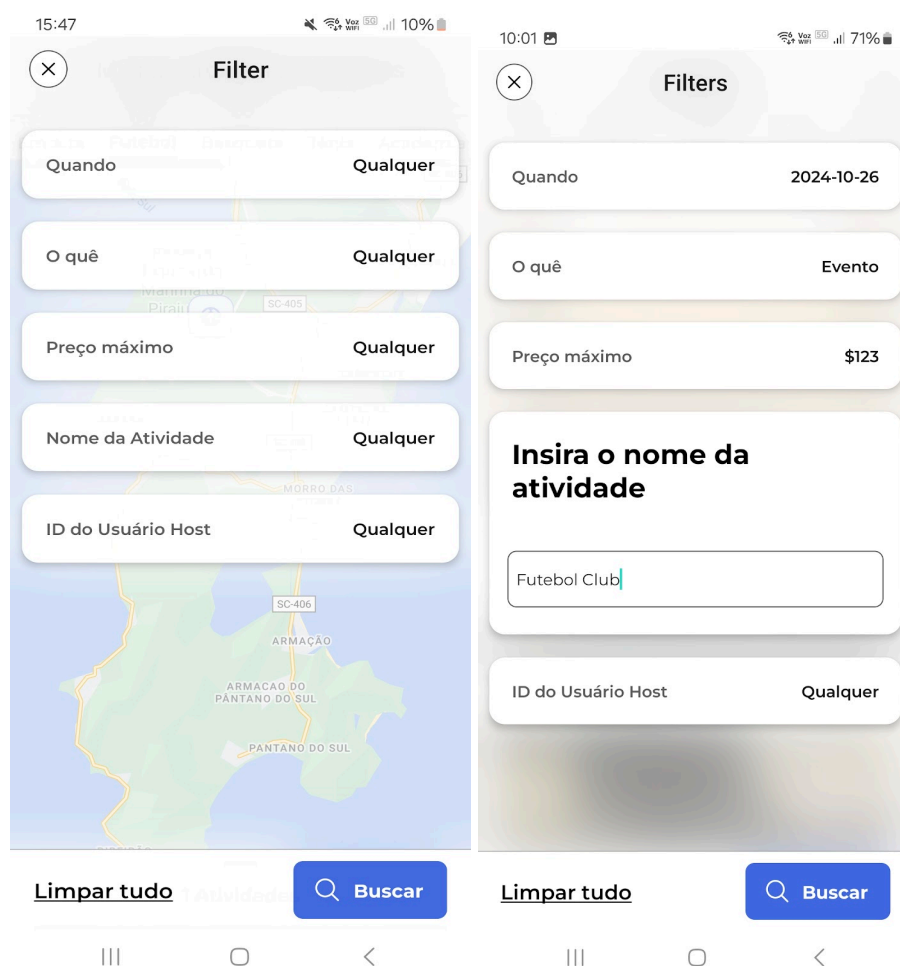
Critérios de aceite:

US02.01 - O usuário pode aplicar filtros como tipo de atividade, esporte, preço, palavras-chave e outros critérios.

US02.02 - O sistema retorna atividades que correspondem aos critérios de busca.

Protótipos de tela: Mapa Interativo - Filtro

Resultado:



Fonte: elaborado pelo autor.

Quadro 5 - História de Usuário 03.

US03 - Como usuário, eu quero visualizar uma lista de atividades correspondentes aos meus critérios de busca para que eu possa escolher as que me interessam.

Critérios de aceite:

US03.01 - O sistema exibe uma lista de atividades que correspondem aos critérios de busca realizada.

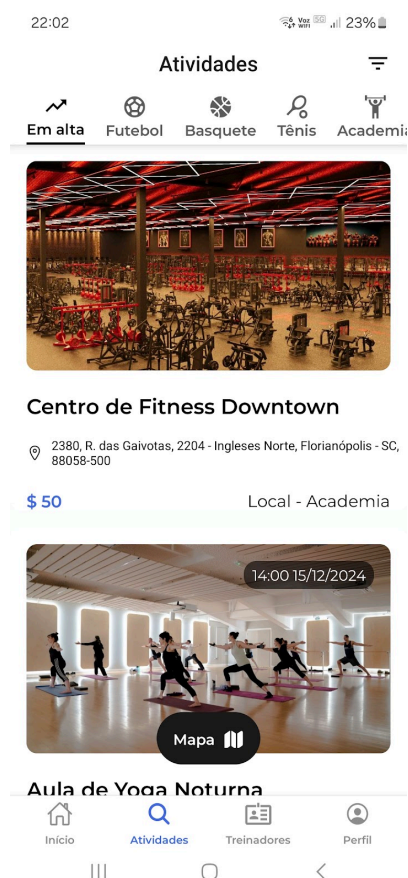
US03.02 - A lista exibe detalhes resumidos de cada atividade, como nome, data, endereço, preço e imagem.

US03.03 - O usuário pode selecionar atividades de interesse a partir da lista.

Requisitos: RF03

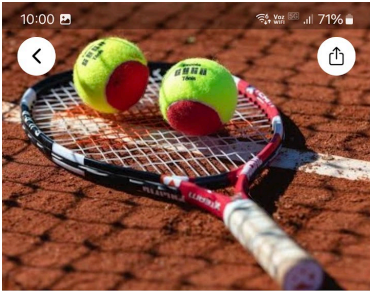
Protótipos de tela: Mapa Interativo - Listagem

Resultado:



Fonte: elaborado pelo autor.

Quadro 6 - História de Usuário 04.

US04 - Como usuário, eu preciso visualizar os detalhes completos de uma atividade para obter informações completas antes de participar.
Critérios de aceite: US04.01 - O sistema exibe todas as informações de uma atividade, como nome, imagem, descrição, data, hora, local, localização, descrição, número de participantes total e atual, e informações sobre o organizador. US04.02 - O usuário requisita a participação na atividade. US04.03 - O sistema retorna ao usuário um aviso com o resultado da requisição
Requisitos: RF04
Protótipos de tela: <u>Detalhes de atividade</u> Resultado:  All tennis cup Evento - Tênis Av. Ricardo Flores Magón 6BIS, Guerrero, Cuauhtémoc, 06300 Ciudad de México, CDMX, Mexico Início: 12:42 23/10/2024 Fim: 13:42 23/10/2024 Participantes: 1 / 50  Organizado por Grieder Media Organizador desde 2024-09-22 Torneio de tênis gratuito! GRÁTIS Cadastro Sair Agora III □ <

Fonte: elaborada pelo autor.

Quadro 7 - História de Usuário 05.

US05 - Como usuário com permissões de criação, eu quero criar atividades de acordo com meu tipo de usuário para oferecer eventos, sessões ou locais.

Critérios de aceite:

US05.01 - O usuário entra na página de criar uma atividade.

US05.02 - Usuário Casual pode criar atividades do tipo Evento.

US05.03 - Treinador pode criar atividades dos tipos Evento e Sessão.

US05.04 - Empresa pode criar atividades dos tipos Evento, Sessão e Local.

US05.05 - O usuário insere as informações necessárias para a criação da atividade.

US05.06 - O sistema verifica os dados informados, retornando avisos específicos avisando os erros de validação caso contenha..

US05.07 - O sistema permite a inserção de informações necessárias para a atividade caso não tenha erros de validação.

Requisitos: RF05

Protótipos de tela: Criar Atividade

The image displays two mobile application prototypes for the 'Criar Atividade' (Create Activity) screen. Both prototypes are shown in a mobile view with a status bar at the top indicating the time as 09:56 and 09:57, and battery level at 72%.

Left Prototype (09:56):

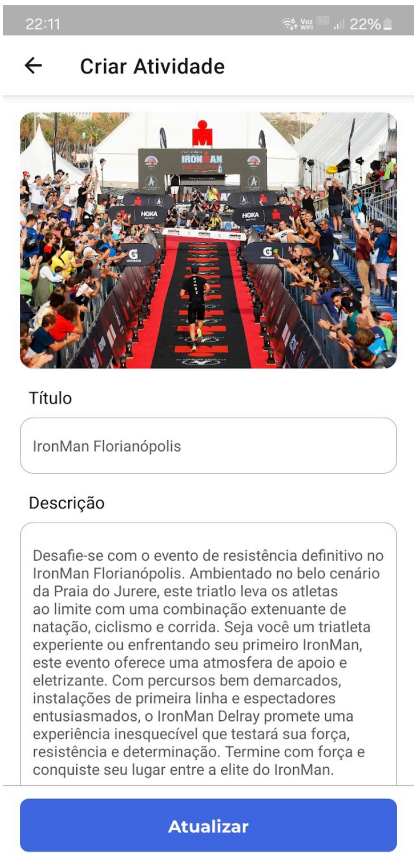
- Header:** 'Criar Atividade' with a back arrow.
- Image:** A photo of a man in a swimming pool.
- Título:** 'Estádio Swim - Natação'.
- Descrição:** 'Entre si na nossa comunidade do estádio suíno a maior piscina de natação do Estado'.
- Descrição privada:** 'Seja bem vindo!'.
- Button:** 'Criar Atividade'.

Right Prototype (09:57):

- Header:** 'Criar Atividade' with a back arrow.
- Tipo de Atividade:** 'Local'.
- Esporte:** 'Natação'.
- Preço:** '200'.
- Moeda:** 'R\$ - Real'.
- Selecione o Local:** A map showing a location near 'Praça Lauro Muller' and 'R. Heitor Luz'.
- Button:** 'Criar Atividade'.

Fonte: elaborado pelo autor.

Quadro 8 - História de Usuário 06.

US06 - Como usuário, eu quero editar atividades que criei para atualizar informações relevantes.
Critérios de aceite: US06.01 - O usuário entra na página de editar atividade. US06.02 - O usuário solicita a atualização de informações como nome, imagem, data, hora, local e descrição. US06.03 - As alterações são salvas e refletidas no sistema.
Requisitos: RF06
Protótipos de tela: <u>Criar Atividade</u> Resultado: 

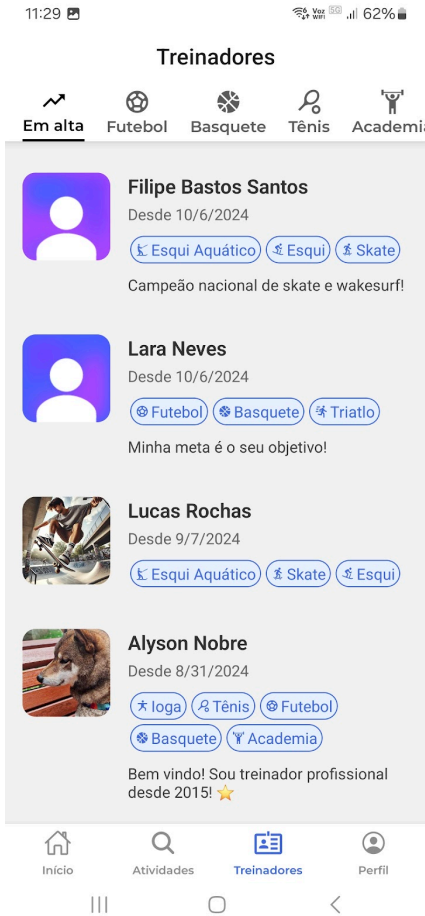
Fonte: elaborado pelo autor.

Quadro 9 - História de Usuário 07.

US07 - Como usuário, eu quero participar ou sair de atividades para gerenciar minha agenda.
Critérios de aceite: US07.01 - O usuário entra na página da atividade. US07.02 - O usuário pode se inscrever na atividade. US07.03 - O usuário pode cancelar sua participação em uma atividade. US07.04 - A lista de participantes e agenda do usuário é atualizada automaticamente.
Requisitos: RF07
Protótipos de tela: <u>Sair da Atividade</u> e <u>Entrar em Atividade</u> The image displays two side-by-side mobile app prototypes. Both screens show a soccer match background with a player in a white jersey (number 10) and a player in a blue jersey. A white success dialog box is centered on each screen. The left screen shows the message 'Sucesso' and 'Você saiu da atividade.' with an 'OK' button. The right screen shows the message 'Sucesso' and 'Você participou da atividade.' with an 'OK' button. Below the dialog boxes, both screens display the same information: 'Fim: 01:07 29/12/2024', 'Participantes: 0 / 3' (left) and 'Participantes: 1 / 3' (right), 'Organizado por Lucas Rochas', 'Organizador desde 2024-09-07', and a promotional text 'Participe da nossa Copa de Futebol! Este torneio está aberto para todas as idades e'. At the bottom, both screens show a price of '\$ 123' and a 'Cadastro' button. The left screen has a 'Junte-se Agora' button, and the right screen has a 'Sair' button. The bottom navigation bar of the app is visible at the very bottom of each screen.

Fonte: elaborado pelo autor.

Quadro 10 - História de Usuário 08.

US08 - Como usuário, eu quero buscar treinadores aplicando filtros para encontrar profissionais adequados.
Critérios de aceite: US08.01 - O usuário entra na página de treinadores. US08.02 - O usuário pode aplicar o filtro de esporte ensinado pelo treinador. US08.03 - O sistema retorna treinadores que correspondem ao critério do filtro.
Requisitos: RF08
Protótipos de tela: <u>Listagem de Treinadores</u> Resultado: 

Fonte: elaborado pelo autor.

Quadro 11 - História de Usuário 09.

US09 - Como usuário, eu quero visualizar o perfil de treinadores para conhecer suas qualificações.

Critérios de aceite:

US09.01 - O usuário acessa o perfil do treinador.

US09.02 - O sistema exibe todas as informações da página de perfil, que incluem informações de contato e redes sociais.

Requisitos: RF09

Protótipos de tela: Perfil

Resultado:



Fonte: elaborado pelo autor.

Quadro 12 - História de Usuário 10.

US10 - Como usuário, eu quero criar e atualizar meu perfil para que outras pessoas possam me conhecer melhor.
Critérios de aceite: US10.01 - Na criação de sua conta o usuário pode definir esportes principais, tipo de usuário, foto, nome de usuário, biografia e informações de contato e redes sociais externas. US10.02 - O usuário pode editar seu perfil a qualquer momento, editando essas informações. US10.03 - As atualizações são salvas no sistema.
Requisitos: RF10
Protótipos de tela: Telas de <u>Cadastro</u> Resultado: 

Fonte: elaborado pelo autor.

Quadro 13 - História de Usuário 11.

US11 - Como usuário, eu quero alterar o idioma da interface para português ou inglês conforme minha preferência.

Critérios de aceite:

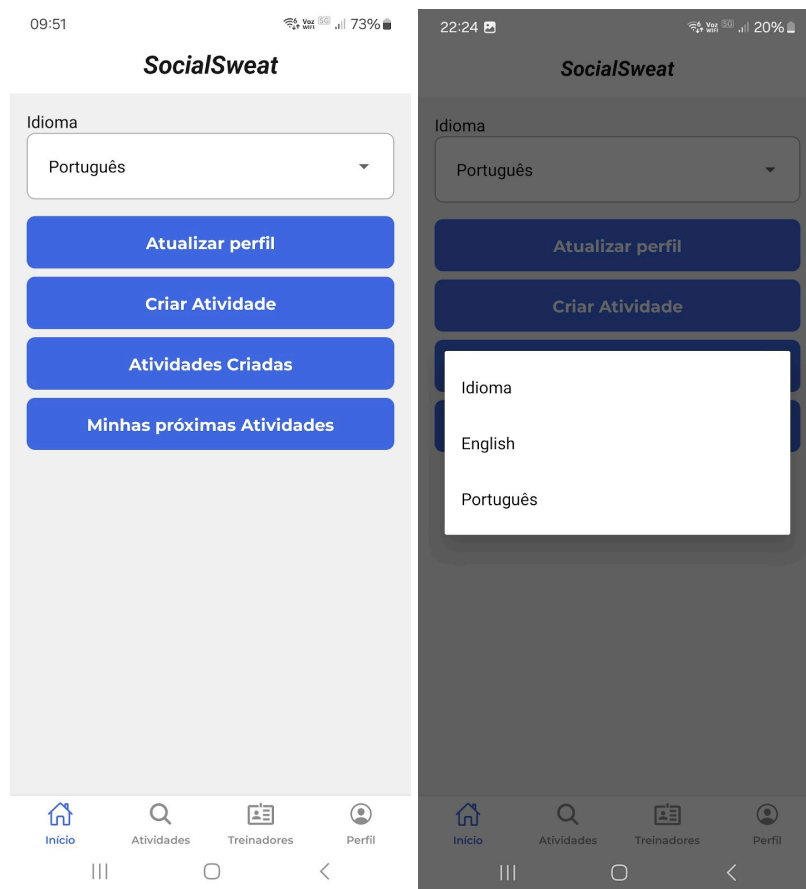
US11.01 - O usuário pode selecionar entre português e inglês.

US11.02 - A interface do sistema atualiza para o idioma escolhido, e grava a opção selecionada.

Requisitos: RF11, RNF06

Protótipos de tela: Configurações

Resultado:



Fonte: elaborado pelo autor.

Quadro 14 - História de Usuário 12.

US12 - Como usuário, eu quero fazer <i>logout</i> de minha conta.
Critérios de aceite: US12.01 - Na página de perfil ou configurações, deve haver um botão para o usuário sair de sua conta no dispositivo, caso ele já esteja em uma conta. US12.02 - Após o <i>logout</i>, o usuário é redirecionado para a tela inicial do aplicativo.
Requisitos: RF12, RNF02
Protótipos de tela: <u>Configurações</u> Resultado: 

Fonte: elaborado pelo autor.

4.3 CASOS DE USO

Os casos de uso foram separados para as 3 categorias de usuários possíveis, definidas no capítulo 1 deste projeto. Vários casos de uso são compartilhados entre as diferentes categorias de usuários, e estão referenciadas.

4.3.1 Usuário Casual

O usuário casual, também chamado apenas de usuário no aplicativo, como o nome sugere, é qualquer usuário que não seja um treinador e não seja uma empresa, que apenas deseje participar de atividades e criar eventos esporadicamente. A Figura 1 mostra os casos de uso para este tipo de usuário.

Figura 1 - Casos de uso do Usuário Casual.



Fonte: elaborada pelo autor.

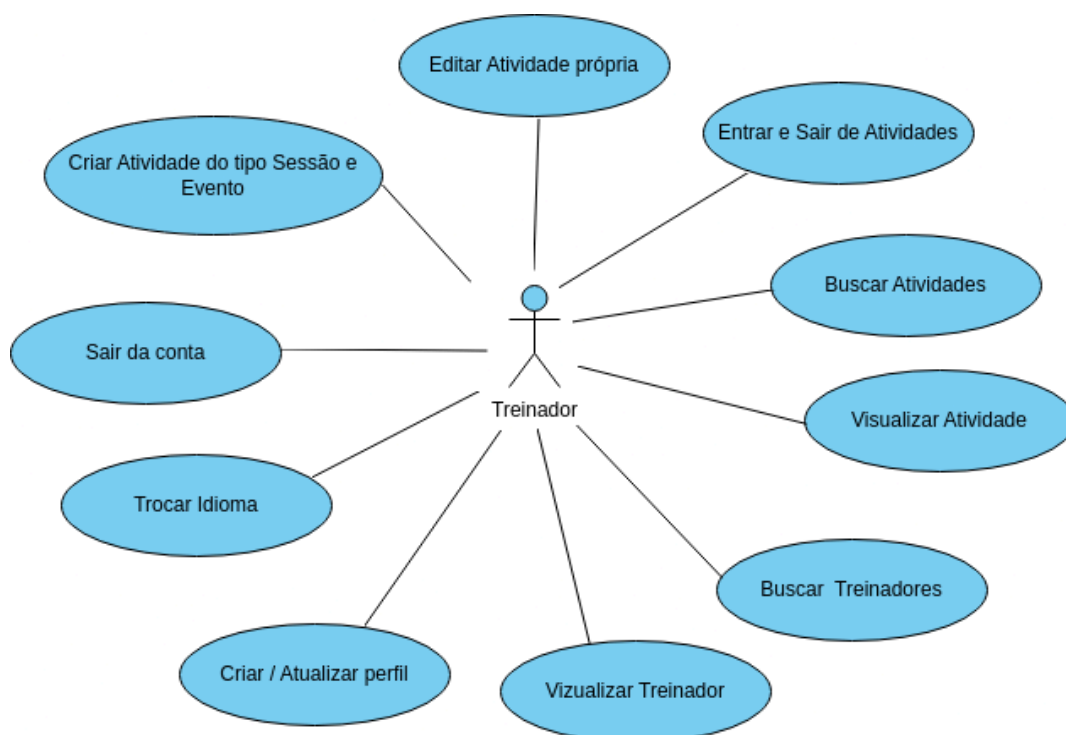
- **Criar Atividade do Tipo Evento:** o usuário casual tem a capacidade de criar atividades do tipo evento. Acessando a opção de criar atividade, ele preenche os detalhes do evento, como data, hora, local e descrição. O sistema então salva o evento e o torna visível para outros usuários.
- **Editar Atividade Própria:** o usuário casual pode editar qualquer atividade que tenha criado. Ao acessar a lista de suas atividades, ele seleciona a atividade que deseja editar, modificar as informações necessárias, como dados, desativá-la e ativá-la ou remover um participante e salva as alterações. O sistema atualiza os detalhes da atividade.
- **Entrar e Sair de Atividades:** o usuário casual pode participar ou sair de atividades. Ao visualizar uma atividade de interesse, ele clica em "Participar" para entrar ou em "Sair" para deixar a atividade. O sistema atualiza a lista de participantes.
- **Buscar Atividades:** o usuário casual pode utilizar a funcionalidade de busca para encontrar atividades específicas. Na página de atividades, ele insere palavras-chave ou aplica filtros na barra de busca, e o sistema retorna uma lista de atividades que correspondem aos critérios inseridos.
- **Visualizar Atividade:** o usuário casual pode selecionar uma atividade da lista de resultados para visualizar seus detalhes. O sistema exibe informações como data, hora, local, descrição, número de participantes e organizador.
- **Buscar Treinadores:** o usuário casual pode buscar treinadores, filtrando por esporte. O sistema retorna uma lista de treinadores que correspondem aos critérios.
- **Visualizar Perfil:** o usuário casual pode visualizar o perfil de outros usuários. Ao selecionar um treinador da lista de resultados, ou acessar o criador de uma

atividade, o sistema exibe o perfil do usuário, incluindo seus esportes principais, biografia e informações de contato.

- **Criar/Atualizar Perfil:** o usuário pode criar ou atualizar seu perfil. Acessando as configurações da conta, ele pode definir esportes principais, tipo de usuário, foto, nome de usuário, biografia e informações de contato. O sistema salva as alterações e atualiza o perfil público do usuário.
- **Buscar Treinadores:** o usuário pode buscar treinadores, filtrando por esporte. O sistema retorna uma lista de treinadores que correspondem aos critérios.
- **Trocar Idioma:** o usuário pode trocar o idioma do aplicativo entre português e inglês. O sistema atualiza a interface para o idioma selecionado.
- **Sair da Conta:** o usuário pode sair de sua conta. Nas configurações, ele seleciona a opção "Sair". O sistema encerra a sessão do usuário e retorna à tela inicial.

4.3.2 Treinador

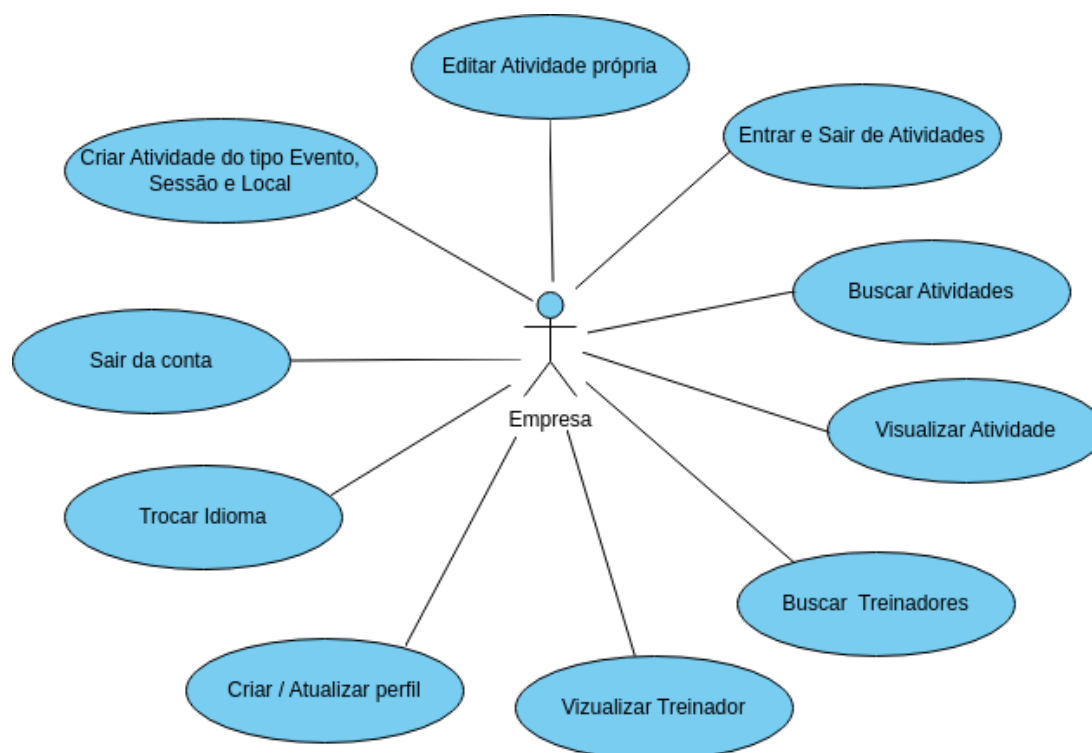
O usuário do tipo treinador também pode utilizar todas as funcionalidades disponíveis para o usuário casual: Editar Atividade Própria, Entrar e Sair de Atividades, Buscar Atividades, Visualizar Atividade, Buscar Treinadores, Visualizar Usuário, Criar/Atualizar Perfil, Trocar Idioma e Sair da Conta. A diferença é que o seu caso de uso Criar Atividade do Tipo Evento também tem atividades do tipo Sessão, ficando como Criar Atividade do Tipo Evento e Sessão. Seu caso de uso completo é visível na Figura 2.

Figura 2 - Casos de uso do Treinador.

Fonte: elaborada pelo autor.

4.3.3 Empresa

A empresa também pode utilizar todas as funcionalidades disponíveis para usuários casuais e treinadores: Editar Atividade Própria, Entrar e Sair de Atividades, Buscar Atividades, Visualizar Atividade, Buscar Treinadores, Visualizar Usuário, Criar/Atualizar Perfil, Trocar Idioma e Sair da Conta. A única mudança é que o seu caso de uso de Criar Atividade fica como Criar Atividade do Tipo Evento, Sessão e Local, como é visto na Figura 3.

Figura 3 - Casos de uso da empresa.

Fonte: elaborada pelo autor.

4.4 ARQUITETURA DO SISTEMA

O aplicativo foi desenvolvido em React Native, permitindo compartilhar código entre Android e iOS para melhorar a eficiência de desenvolvimento. O *backend* usa um microserviço FastAPI, encapsulado em um container Docker. Esta escolha facilita a manutenção, escalabilidade e distribuição consistente dos serviços em diferentes ambientes. As informações sensíveis, como privilégios de usuários e autenticação, serão armazenadas de forma segura na Clerk, enquanto os dados operacionais, como atividades, serão armazenadas no MongoDB que se conecta ao microserviço.

A escolha do Clerk como o serviço de gerenciamento de identidades e acesso para clientes (CIAM) foi motivada pela praticidade e simplicidade da integração, garantindo que

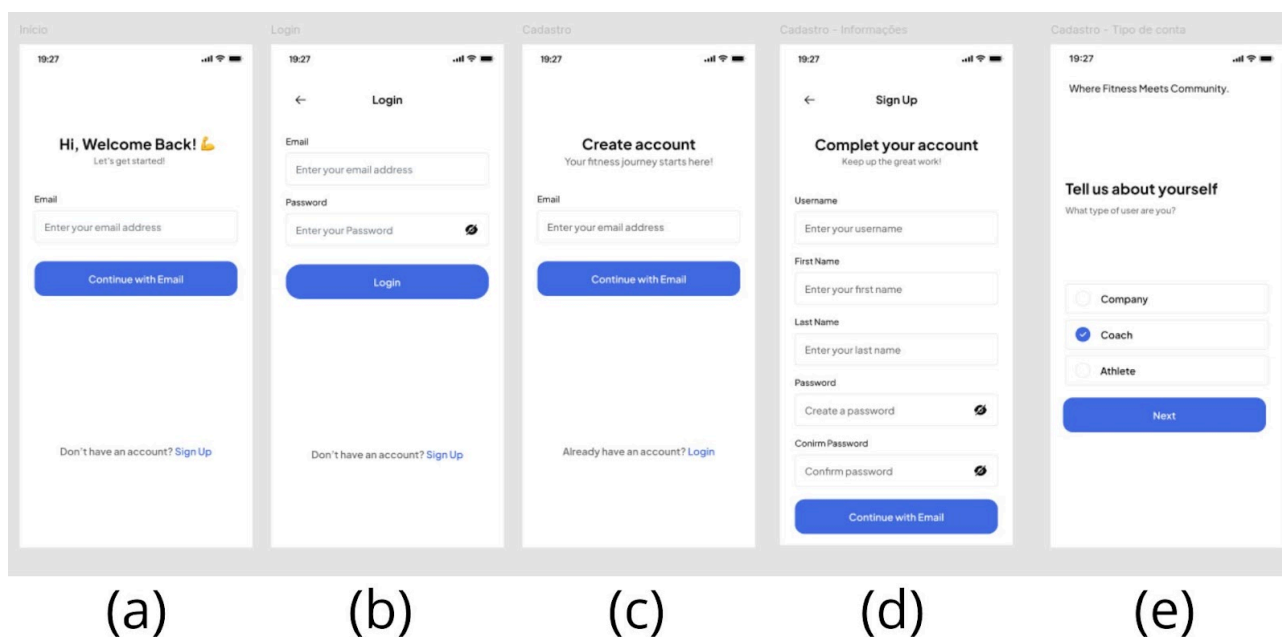
os dados privados e sensíveis dos usuários fiquem armazenados em locais próprios para isto.

4.5 PROTÓTIPOS DE INTERFACE

Os protótipos de interface desenvolvidos atendem aos requisitos e casos de uso inicialmente propostos, que sofreram algumas alterações para a versão final. A principal mudança foi a retirada de conversas por mensagens direto no aplicativo, pois já existem plataformas externas que já são consolidadas para isto, como o WhatsApp, Facebook, SMS e Instagram, e o perfil de um usuário já pode ter atalho para estas outras redes sociais. Os atalhos dinâmicos da tela inicial foram definidos apenas como botões para páginas específicas, como a página de suas próximas atividades em que você irá participar e a página de atividades que você criou. A aba específica de configurações foi retirada, com as configurações presentes estarem definidas na própria página inicial. Avaliações para treinadores não foram implementadas. Foi criada uma aba específica para visualizar os treinadores. Houve mudanças gerais na implementação de todas páginas em geral, mas o estilo e tema permanece o mesmo.

- **Início:** Figura 4(a) - Tela inicial ao entrar pela primeira vez no aplicativo.
- **Login:** Figura 4(b) - Tela para entrar em uma conta já existente.
- **Cadastro:** Figura 4(c) - Tela para inserir o *e-mail* para criar a conta.
- **Cadastro - Informações:** Figura 4(d) - Tela para adicionar as informações do usuário.
- **Cadastro - Tipo de Conta:** Figura 4(e) - Tela para escolher o tipo de conta do usuário.

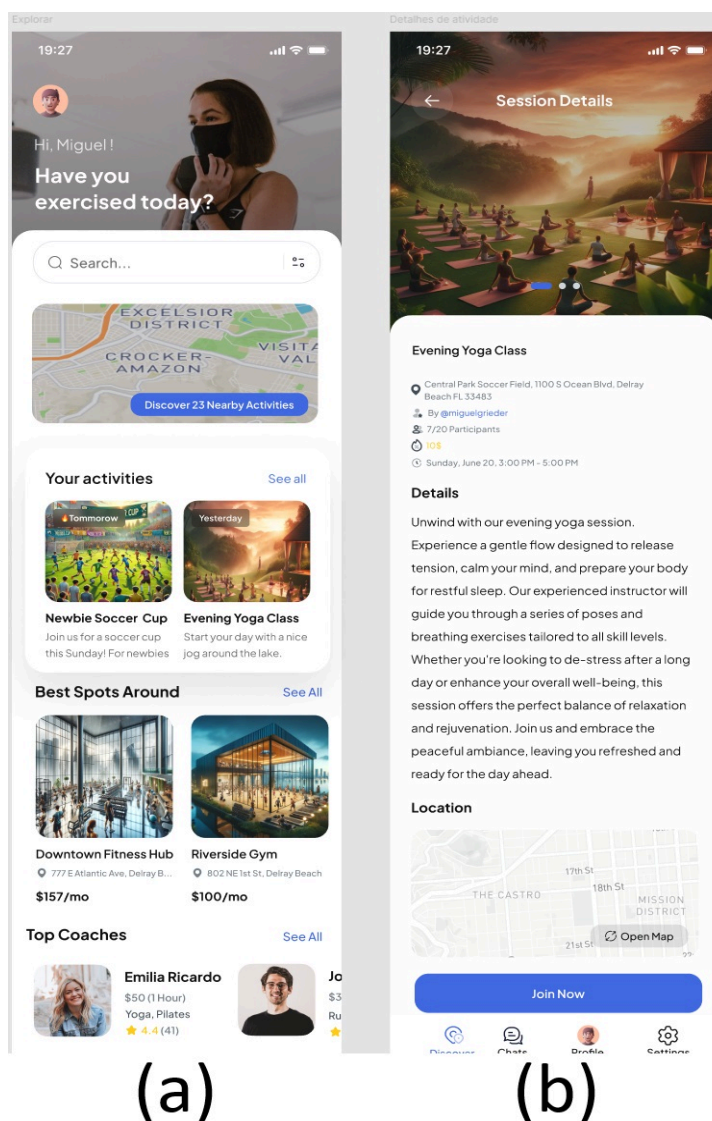
Figura 4 - Telas de Início(a) / Login(b) / Cadastro(c) / Cadastro - Informações e Cadastro(d) - Tipo de Conta(e).



Fonte: elaborada pelo autor.

- **Explorar:** Figura 5(a) - Tela principal do aplicativo, onde mostra todas categorias, buscas de atividades e treinadores disponíveis. Irá ter uma busca por palavras ou categorias no topo. Em baixo, terá um *widget* de mapa com as atividades próximas e listas de categorias relevantes para o usuário, podendo ser de treinadores, atividades grátis, atividades populares e outros.
- **Detalhes de atividade:** Figura 5(b) - tela que mostra todas as informações de uma atividade, e permite entrar na atividade.

Figura 5 - Tela de Explorar(a) / Detalhes de atividade(b).



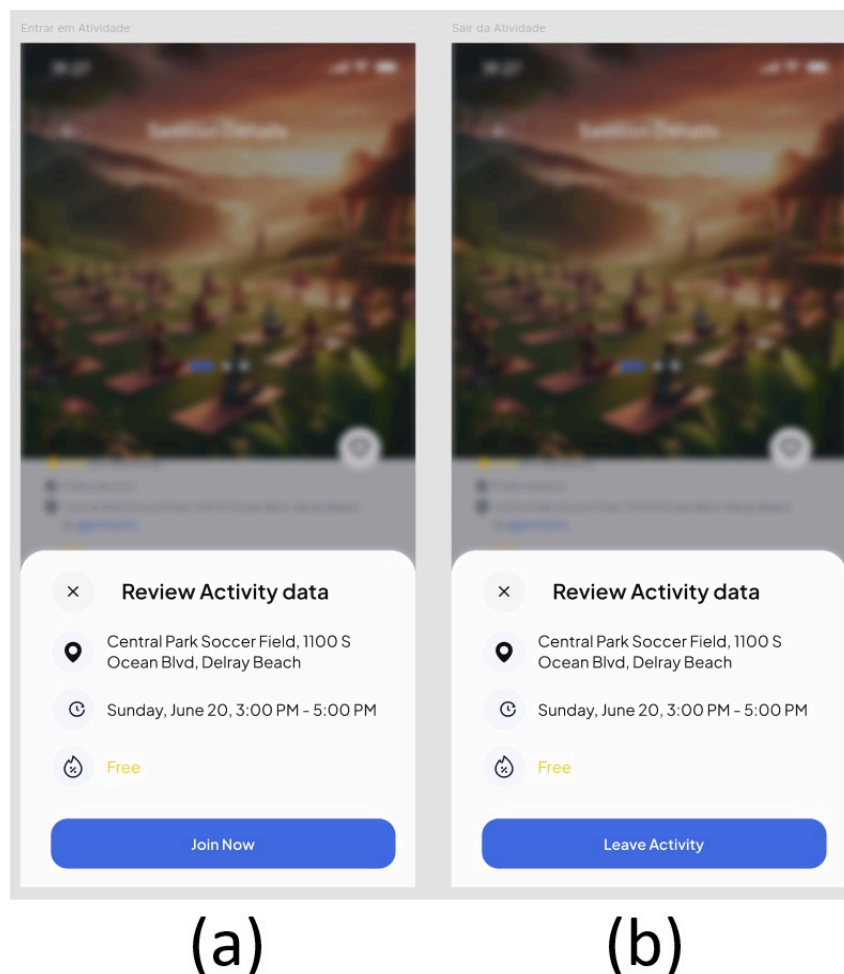
(a)

(b)

Fonte: elaborada pelo autor.

- **Entrar em Atividade:** Figura 6(a) - Tela de Detalhes de Atividade com o aviso de confirmação para entrar na atividade.
- **Sair da Atividade:** Figura 6(b) - Tela de Detalhes de Atividades com o aviso de confirmação para sair na atividade.

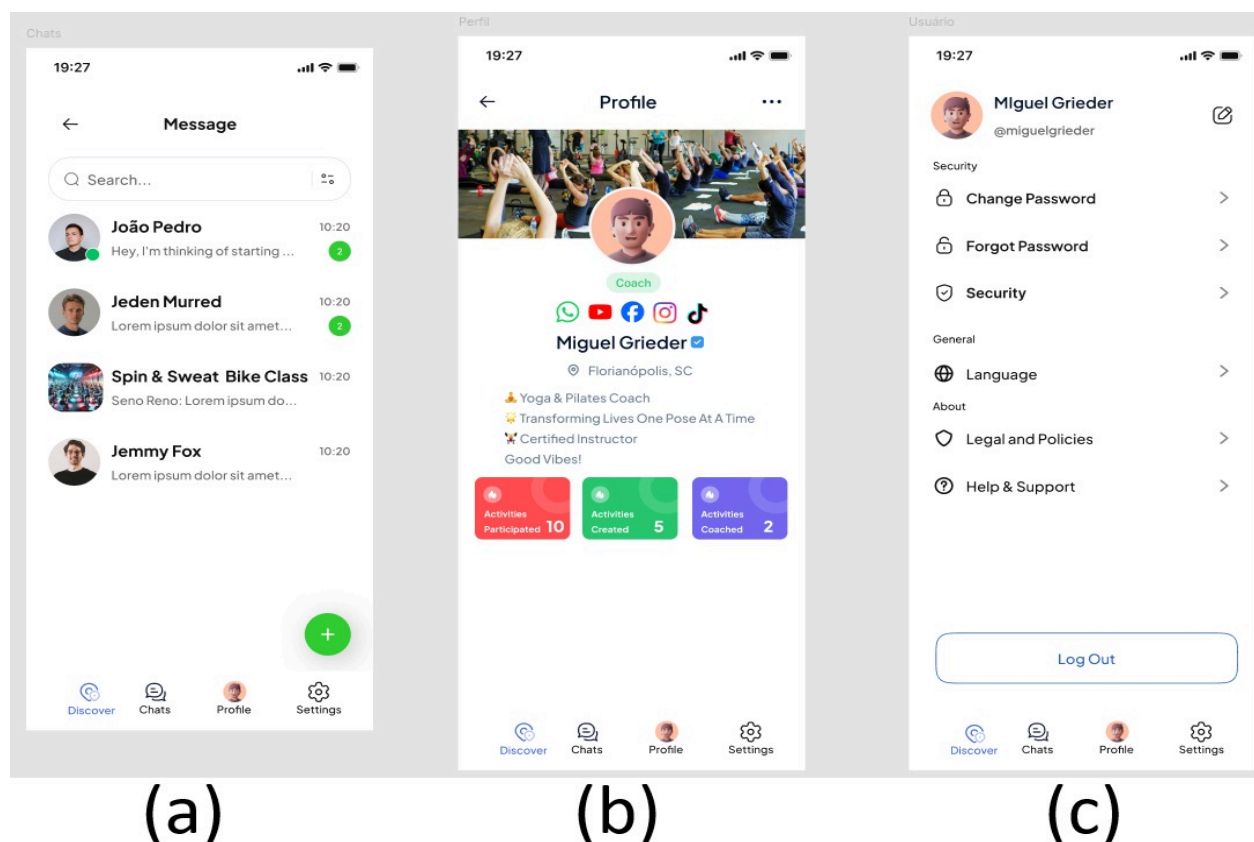
Figura 6 - Tela de Entrar em Atividade(a) / Sair da Atividade(b).



Fonte: elaborada pelo autor.

- **Chats:** Figura 7(a) - Tela das conversas do usuário, com outros usuários ou de atividades que ele participa. Irá permitir pesquisar pelo nome de conversa.
- **Perfil:** Figura 7(b) - Tela que mostra as informações de um perfil de usuário.
- **Configurações:** Figura 7(c) - Tela de configurações da conta do usuário, que inclui opções de segurança, como a alteração de senha e de idioma, além de informações legais e opções de ajuda e suporte.

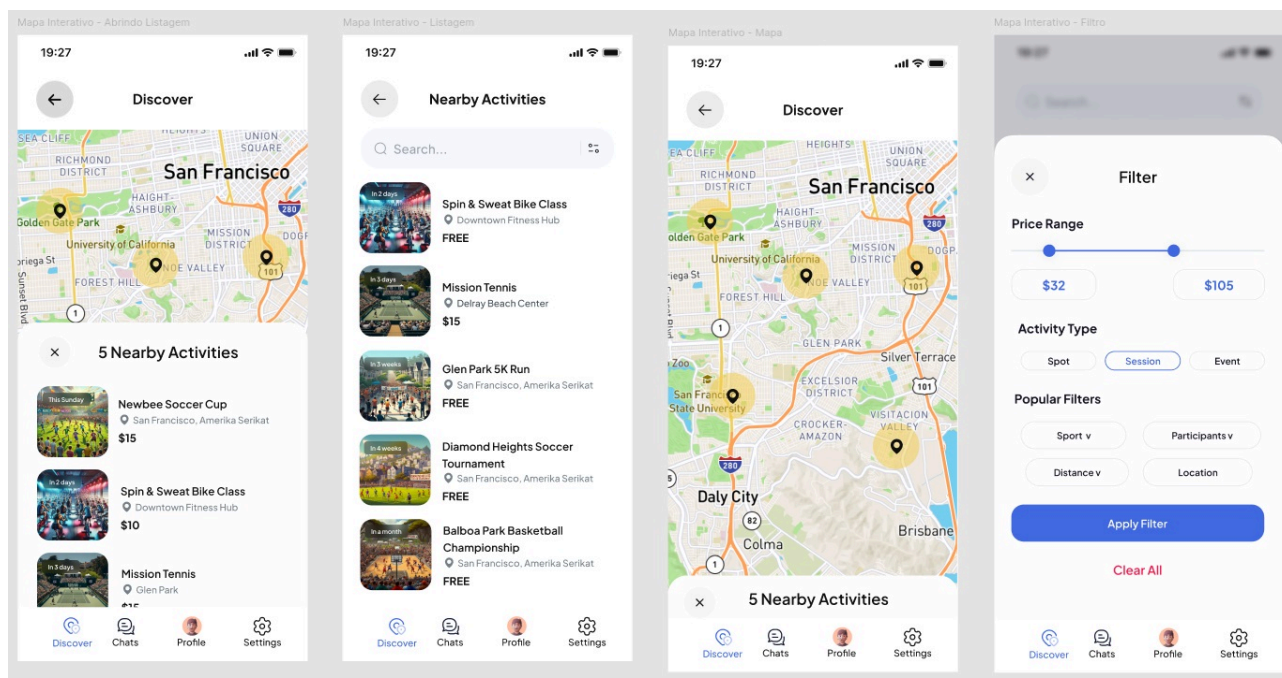
Figura 7 - Tela de *Chats(a)* / Perfil(b) / Configurações(c).



Fonte: elaborada pelo autor.

- **Mapa Interativo - Abrir Listagem:** Figura 8(a) - Tela de mapa quando está abrindo a listagem de atividades.
- **Mapa Interativo - Listagem:** Figura 8(b) - Tela de mapa quando a listagem está aberta.
- **Mapa Interativo - Mapa:** Figura 8(c) - Tela de mapa com a listagem fechada e mostrando apenas o mapa com os ícones de atividades em suas posições.
- **Mapa Interativo - Filtro:** Figura 8(d) - Tela de mapa quando abre o filtro da pesquisa.

Figura 8 - Tela de Mapa Interativo - Abrir Listagem(a) / Listagem(b) / Mapa(c) / Filtro(d).



(a)

(b)

(c)

(d)

Fonte: elaborada pelo autor.

- **Criar Atividade:** Figura 9 - Tela para criar uma atividade.

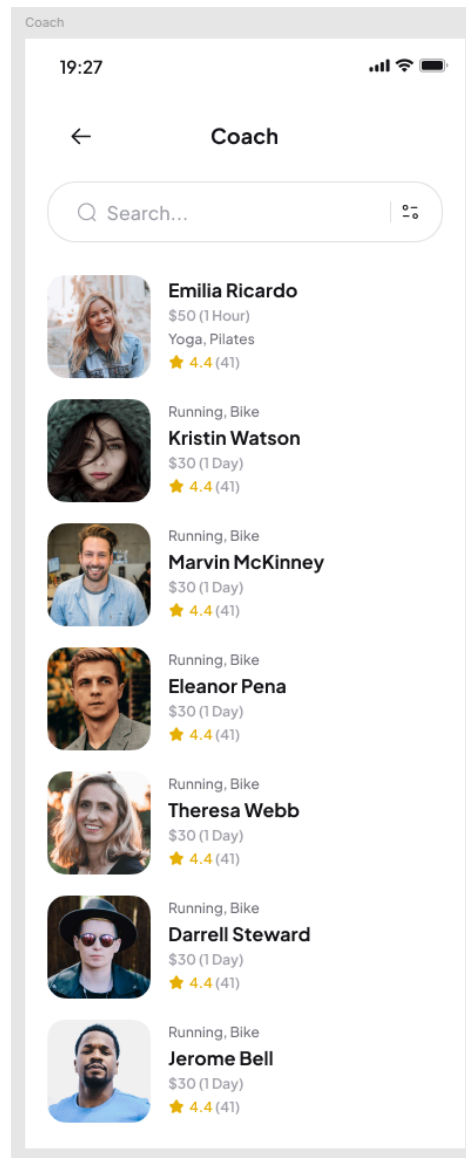
Figura 9 - Tela de Criar Atividade(a).

The screenshot displays a mobile application interface for creating a new activity. At the top, the title 'Criar Atividade' is visible. Below it, a status bar shows the time '19:27' and signal indicators. The main heading is 'Add Activity' with a back arrow. A dashed box contains a camera icon and the text 'Upload Activity Photo'. Below this are several form fields, each with an asterisk indicating it is required: 'Title*' (placeholder: Activity title), 'Description*' (placeholder: Activity description), 'Activity type*' (placeholder: Event v), 'Price*' (placeholder: 0.00 \$, with a currency icon), 'Date & Hour*' (placeholder: Sunday, June 20, 3:00 PM), 'Location*' (placeholder: Activity location), and 'Sport*' (placeholder: Activity sport). At the bottom is a blue button labeled 'Create Activity'.

Fonte: elaborada pelo autor.

- **Listagem de Treinadores:** Figura 10 - Tela que lista os treinadores disponíveis.

Figura 10 - Tela de Listagem de Treinadores(a).



Fonte: elaborada pelo autor.

4.6 ESQUEMA DO BANCO DE DADOS

Neste subcapítulo está comentado como será a estrutura dos dados armazenados pela aplicação. Nenhum dado ficará salvo no dispositivo do usuário, todos os dados são salvos no *backend*. Os dados sobre atividades estão salvos no MongoDB, enquanto os dados referente aos usuários estão salvos no próprio Clerk.

4.6.1 Dados de Atividade no MongoDB

A principal funcionalidade do aplicativo são as atividades. Elas ficarão registradas no MongoDB como documentos de uma coleção e sua estrutura é:

1. id: Identificador único da atividade (*String*).
2. enabled: Indica se a atividade está ativa (*Boolean*).
3. name: Nome da atividade (*String*).
4. description: descrição detalhada da atividade (*String*).
5. description_private: descrição privada da atividade, visível apenas para usuários participantes (*String* opcional).
6. activity_type: tipo de atividade (ENUM ActivityType - *String*).
7. sport_type: tipo de esporte, por exemplo (ENUM SportType - *String*).
8. price:
 - a. value: valor do preço da atividade (*Float*).
 - b. unit: unidade monetária. (ENUM PriceUnit - *String*)
9. location:
 - a. country: país onde a atividade ocorre. (ENUM Country - *String*)
 - b. area: área ou estado da atividade. (*String*)
 - c. city: cidade onde a atividade ocorre. (*String*)
 - d. smart_location: endereço completo ou descrição da localização. (*String*)
 - e. geometry:
 - i. type: Tipo de geometria, sempre sendo "Point" (*String*).
 - ii. coordinates:
 1. latitude: Latitude do local (*Float*).
 2. longitude: Longitude do local (*Float*).
10. participants:

- a. participants_user_id: lista de IDs dos usuários que estão participando da atividade (*Array de Strings*).
 - b. max: Número máximo de participantes permitidos (Int opcional).
11. pictures: Lista de URLs ou base64 das fotos da atividade (*Array de Strings*).
12. host:
- a. host_user_id: ID do usuário que criou a atividade. (*String*)
13. datetimes:
- a. datetime_created: Data e hora de criação da atividade (ISODate - *String*).
 - b. datetime_deleted: Data e hora de exclusão da atividade (ISODate - *String* ou *null*).
 - c. datetime_start: Data e hora de início da atividade (ISODate - *String* opcional).
 - d. datetime_finish: Data e hora de término da atividade (ISODate - *String* opcional).

Detalhes sobre os campos ENUM utilizados no banco de dados são descritos a seguir:

- activity_type: ActivityType - enumeração que indica o tipo de atividade:
 - "spot" (Local)
 - "session" (Sessão)
 - "event" (Evento)
- sport_type: SportType - Enumeração que especifica o tipo de esporte:
 - "gym", "basketball", "soccer", "tennis", "yoga", "triathlon", "run",
"martial_arts", "motorsports", "volleyball", "handball", "hockey", "ski",
"ski_water", "baseball", "skateboard", "esports", "swim", "other".

- price.unit: PriceUnit - Enumeração para a unidade monetária:
 - "\$", "R\$", "€"
- location.country: Country - Enumeração para o país:
 - "brazil", "united_states", "germany", "argentina", "paraguay", "chile", "canada", "mexico", etc.

Essa estrutura permite armazenar todas as informações necessárias sobre uma atividade, facilitando a gestão e a busca por parte dos usuários no aplicativo. Na Figura 11 é exibido um exemplo real de atividade, que está hospedada em nuvem no provedor Railway.

Figura 11 - Documento de uma Atividade no MongoDB no Railway.

```
_id: "6718664b01d40c345eeeb541"
id: "1729652298814"
enabled: true
name: "FutPé - escola de futebol"
description: "Entre na nossa escola de futebol aula toda terça 15 horas"
description_private: "Seja bem-vindo! Contato em 48 98888-3334"
activity_type: "session"
sport_type: "soccer"
▼ price: Object
  value: 150
  unit: "R$"
▼ location: Object
  country: "brazil"
  area: ""
  city: "Porto alegre"
  smart_location: "R. Ricardo Seibel de Freitas Lima, 401 - Farrapos, Porto Alegre - RS, ..."
  ▼ geometry: Object
    type: "Point"
    ▼ coordinates: Object
      latitude: -29.974774157944655
      longitude: -51.19457881897688
▼ participants: Object
  ▼ participants_user_id: Array (empty)
  max: 999
▼ pictures: Array (1)
  0: "data:image/png;base64,/9j/4AAQSkZJRgABAQAAQABAAQ/4gHYSUNDX1BST0ZJTEU..."
▼ host: Object
  host_user_id: "user_21s2lsB19U008tyyheXtoNwVakK"
▼ datetimes: Object
  datetime_created: "2024-10-23T02:58:18.814Z"
  datetime_deleted: "2024-10-23T02:59:07.994Z"
  datetime_start: "2024-12-01T18:00:00.000Z"
  datetime_finish: "2024-12-03T20:00:00.000Z"
```

Fonte: elaborada pelo autor.

4.6.2 Dados de usuário no Clerk

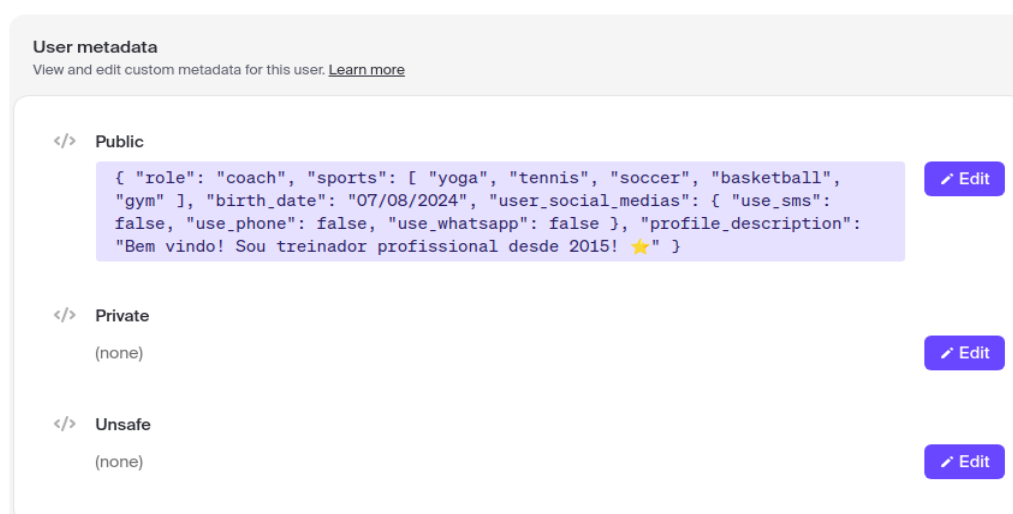
As informações pessoais dos usuários do sistema serão armazenadas na nuvem utilizando o Clerk, organizadas em três categorias distintas de metadados: *private*, *public* e *unsafe*.

- **Private Metadata:** contém informações restritas, acessíveis exclusivamente pela API do *Backend*.
- **Public Metadata:** inclui dados que podem ser acessados para leitura tanto pelo *frontend* quanto pela API do *backend*, mas podendo ser escrita só pela última.
- **Unsafe Metadata:** pode ser lido e atualizado diretamente pelo aplicativo móvel utilizando o pacote do React Native, bem como pela API do *backend*.

Para ser utilizado a API do *backend*, requer credenciais de acesso que não são compartilhadas com o aplicativo móvel, garantindo a segurança e a privacidade dos dados do usuário.

Todos os dados salvos no Clerk pela aplicação estão salvos no campo de *Public Metadata* como visto na Figura 12, e representam dados específicos do usuário em questão, sem o seu relacionamento com as atividades com o qual participa ou cria. Também será detalhado cada campo salvo.

Figura 12 - Metadados no Clerk de um usuário de teste.



Fonte: elaborada pelo autor.

1. role: papel do usuário dentro do sistema, que pode ser um dos seguintes: *coach*, *user*, ou *company* (Enum).
2. sports: lista de atividades preferidas ou oferecidas pelo usuário. (Array de *String*: *ActivityType*)
3. birth_date: data de nascimento do usuário no formato dd/mm/yyyy. Validada para atender ao padrão de data (*String*).
4. user_social_medias: informações e preferências de contato do usuário em redes sociais. Contém os seguintes atributos:
 - a. user_youtube: *username* do perfil do usuário no YouTube (*String*, máx. 100 caracteres).
 - b. user_instagram: *username* do perfil do usuário no Instagram (*String*, máx. 100 caracteres).
 - c. user_facebook: *username* do perfil do usuário no Facebook (*String*, máx. 100 caracteres).
 - d. user_tiktok: *username* do perfil do usuário no TikTok (*String*, máx. 100 caracteres).
 - e. user_strava: *username* do perfil do usuário no Strava (*String*, máx. 100 caracteres).
 - f. user_phone: número de telefone no formato internacional, conforme especificado por *PHONE_NUMBER_REGEX* (*String*, máx. 16 caracteres).

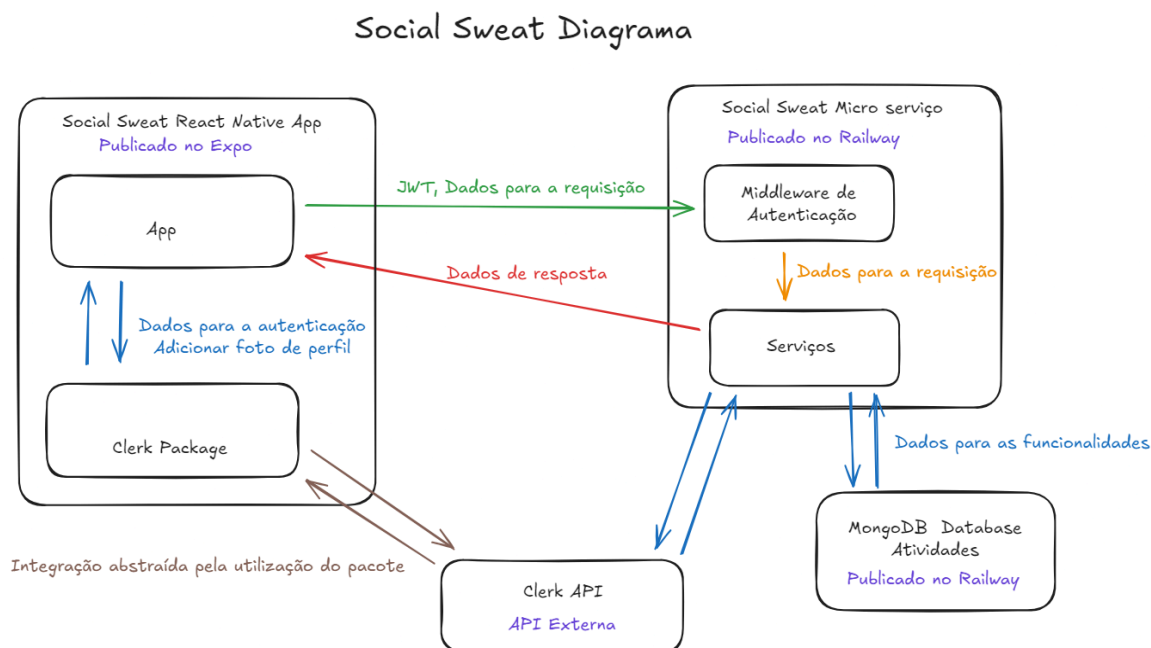
- g. use_phone: indica se deve ter o atalho no perfil para adicionar na discagem o user_phone do usuário (*Boolean*).
 - h. use_sms: indica se deve ter o atalho no perfil para receber SMS no user_phone (*Boolean*).
 - i. use_whatsapp: indica se deve ter o atalho no perfil para o WhatsApp do número user_phone do cliente (*Boolean*).
5. profile_description: breve descrição do perfil do usuário, com um limite de até 500 caracteres (*String*).

5 DESENVOLVIMENTO

Com a compreensão dos conceitos teóricos fundamentais, este capítulo detalha as etapas envolvidas no desenvolvimento do aplicativo Social Sweat, abrangendo desde a concepção das telas de interface até a implementação do *backend* e a configuração da infraestrutura em nuvem. Serão descritas de maneira abrangente todas as fases de desenvolvimento e uso da aplicação.

A Figura 13 ilustra o diagrama da infraestrutura e fluxo de dados do aplicativo, destacando as diferentes partes que compõem a arquitetura, os fluxos de dados entre elas, o tipo de dado transmitido e onde cada componente está hospedado na nuvem. Esse diagrama oferece uma visão geral do processo de integração e comunicação entre os elementos do sistema.

Figura 13 - Diagrama da infraestrutura completo para o aplicativo.



* APIs externas que não tratam de dados persistentes para aplicação não foram incluídas, como consultas em mapas

Fonte: elaborada pelo autor.

A seguir, serão abordados os aspectos principais da implementação, divididos em quatro seções: **Aplicativo (Frontend)**, **Backend** e **Infraestrutura em Nuvem** e **Conclusão da Implementação**.

5.1 APLICATIVO (FRONTEND)

O desenvolvimento do *frontend* do Social Sweat foi realizado com React Native, um *framework* amplamente reconhecido para a criação de aplicativos móveis multiplataforma. É usado Expo e seu *template* padrão, criado com o comando “`npx create-expo-app`”, que inicializa o projeto com várias pré-configurações e pacotes adicionais, facilitando um começo rápido e eficiente. A interface do usuário foi projetada para ser intuitiva, responsiva e visualmente atraente, proporcionando uma experiência fluida e agradável para todos os usuários.

O projeto utiliza variáveis de ambiente para configurar parâmetros sensíveis e específicos do ambiente de desenvolvimento. O uso de arquivos `.env` garante que informações como chaves de API e URLs de *backend* sejam gerenciadas de forma segura e eficiente, com exemplos fornecidos em `.env.example` para facilitar a configuração inicial.

Diversas bibliotecas e pacotes foram integrados para aprimorar as funcionalidades e a usabilidade do aplicativo. A seguir, são detalhadas as principais bibliotecas utilizadas e suas respectivas funções:

- **expo**: permite que o projeto seja executado diretamente no aplicativo Expo Go, facilitando o desenvolvimento, teste e iteração contínua do aplicativo. Expo fornece um ambiente de desenvolvimento integrado que acelera o processo de criação,

depuração e compartilhamento para outros usuários do aplicativo. Disponível em: <https://github.com/expo/expo>.

- **expo-router**: facilita a navegação dentro do aplicativo, permitindo a criação de rotas de forma simples e eficiente. O Expo Router simplifica a gestão das telas e a transição entre elas, contribuindo para uma estrutura de navegação organizada e intuitiva, de forma similar a como as requisições HTTP funcionam. Disponível em: <https://github.com/expo/expo>.
- **react-native-maps**: implementa mapas interativos no aplicativo, essenciais para a visualização das atividades geolocalizadas. Esta biblioteca permite a integração de mapas detalhados, possibilitando aos usuários explorar locais de atividades físicas com facilidade. Disponível em: <https://github.com/react-native-maps/react-native-maps>.
- **clerk/clerk-expo**: gerencia a autenticação e o gerenciamento de contas dos usuários. O pacote Clerk Expo oferece funcionalidades robustas para *login*, registro e gerenciamento de perfis, garantindo segurança e simplicidade no acesso dos usuários ao aplicativo. Disponível em: <https://github.com/clerk/javascript>.
- **apisauce**: baseada no Axios, outra biblioteca npm, o apisauce facilita a realização de chamadas HTTP para a comunicação com o *backend*. Apisauce simplifica a configuração de requisições, tratamento de respostas e gerenciamento de erros, otimizando a interação entre o *frontend* e o *backend*. Disponível em: <https://github.com/infinitered/apisauce>.
- **i18n, intl-pluralrules, i18next e i18n-js**: conjunto de bibliotecas que fornecem suporte completo para internacionalização e localização. Essas ferramentas permitem que o aplicativo ofereça múltiplos idiomas, adaptando-se às preferências

linguísticas dos usuários de forma eficiente. Disponível em:
<https://github.com/i18next>;

- **gorhom/bottom-sheet**: implementa listas de atividades que podem ser arrastadas para cima sobre o mapa interativo. Gorhom Bottom Sheet melhora a experiência do usuário ao permitir a visualização dinâmica das atividades sem sobrecarregar a interface principal. Disponível em:
<https://github.com/gorhom/react-native-bottom-sheet>.
- **react-native-geocoding**: facilita a obtenção de dados de localização a partir de coordenadas geográficas escolhidas no mapa. Esta biblioteca automatiza a adição de atividades com base na localização selecionada, simplificando o processo de criação de eventos geolocalizados. Disponível em:
<https://github.com/marlove/react-native-geocoding>.
- **react-native-map-clustering**: utilizado para agrupar indicadores de atividades no mapa, facilitando a visualização e a navegação. Esta biblioteca agrupa nós próximos em um único ponto de agrupamento, exibindo a quantidade de atividades existentes sobre esse nó, o que melhora a clareza e a eficiência na interação com o mapa.
- **react-native-modal-datetime-picker**: utilizado como seletor de datas tanto na tela de inserção da data de nascimento quanto no filtro avançado de atividades, permitindo que os usuários selecionem facilmente as datas requeridas.
- **react-native-picker-select**: cria *combo-boxes* de opções, melhorando a interface de seleção de dados dentro do aplicativo. React Native Picker Select oferece componentes personalizáveis para *dropdowns*, permitindo uma seleção intuitiva de opções pelos usuários. Disponível em:
<https://github.com/lawnstarter/react-native-picker-select>.

- **react-native-reanimated:** Oferece utilitários para implementar animações em diversos componentes, utilizado para o componente de filtro completo de atividades e para o efeito da imagem de uma atividade ao deslizar sobre ela. Disponível em <https://github.com/software-mansion/react-native-reanimated>.

A combinação dessas bibliotecas e pacotes permite que o Social Sweat ofereça uma experiência de usuário robusta e funcional, atendendo às necessidades de conectividade, interatividade e personalização dos usuários. A utilização dessas ferramentas garante que o aplicativo seja não apenas funcional, mas também escalável e fácil de manter. Não ficam dados do usuário salvos em seu próprio dispositivo, todos seus dados ficam salvos em serviços do *backend*, de modo que ele pode trocar de dispositivo ou ter mais de um dispositivo conectado em sua conta sem que ocorram problemas de inconsistência ou perda de dados.

No próximo tópico, são descritas as principais telas do aplicativo e suas funcionalidades.

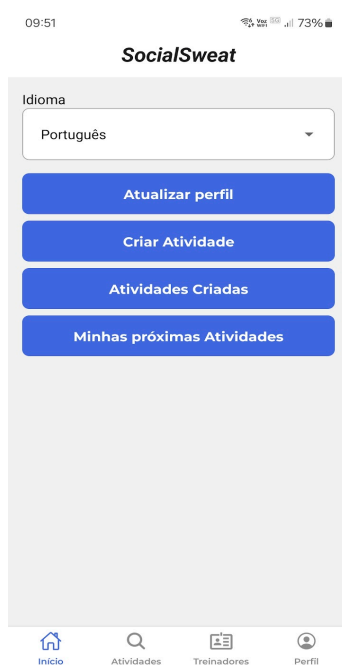
5.1.1 Tela Inicial

A tela inicial (`index.tsx`), vista na Figura 14 serve como o ponto de entrada para os usuários, oferecendo opções essenciais para navegação e configuração inicial:

- **Seletor de Idioma:** permite aos usuários escolherem entre os idiomas disponíveis (português e inglês), personalizando a experiência de uso conforme sua preferência.
- **Atualização de Perfil:** um botão dedicado para que os usuários atualizem ou configurem seus perfis, garantindo que suas informações estejam sempre atualizadas.

- **Botões de Acesso Rápido a Atividades:** incluem botões para acessar páginas de atividades específicas, como Criar Atividade, Atividades Criadas e Minhas Próximas Atividades.

Figura 14 - Tela inicial do Social Sweat.



Fonte: elaborada pelo autor.

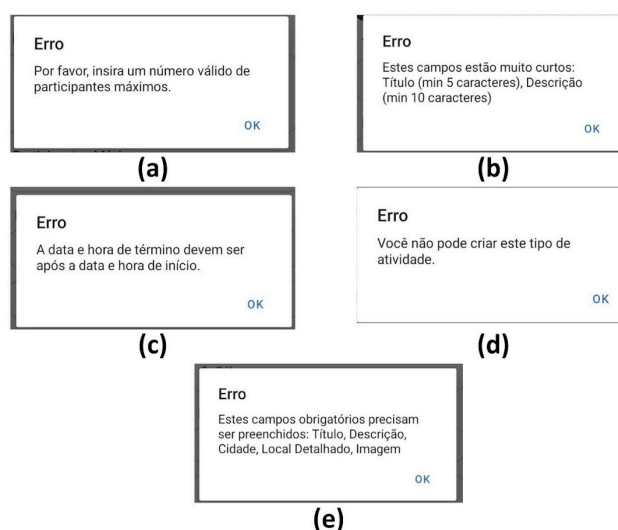
5.1.2 Criação e Atualização de Atividades

As telas de Criação de Atividades (Figura 16) permitem aos usuários criar novas atividades e atualizar informações de atividades existentes:

- **Formulário de Criação (activity-creation.tsx) e de Atualização de Atividade (activity-update.tsx):** inclui campos para inserir detalhes como nome da atividade, data, hora, local, descrição e outras informações relevantes. Algumas informações, como tipo de atividade (Evento, Sessão, Local) não são possíveis de serem trocadas se o usuário estiver editando uma atividade já criada.

- **Validação de Dados:** implementação de validações para garantir que todos os campos obrigatórios sejam preenchidos corretamente antes do envio (também são verificadas no *backend*).
- **Feedback ao Usuário:** na Figura 15 são mostradas mensagens de confirmação ou erro são exibidas para informar o usuário sobre o sucesso ou falha na operação, e informando o que precisaria ser ajustado.

Figura 15 - Exemplos de avisos de erro ao usuário na criação de uma atividade.



Fonte: elaborada pelo autor.

Figura 16 - Tela da página Criar Atividade.

The image displays two side-by-side screenshots of a mobile application interface for creating an activity. Both screens have a status bar at the top showing the time (09:56 and 09:57) and battery level (72%).

Left Screenshot (09:56):

- Header:** A back arrow and the text "Criar Atividade".
- Image:** A photo of a person swimming in a pool.
- Título:** A text input field containing "Estádio Swim - Natação".
- Descrição:** A text input field containing "Entre si na nossa comunidade do estádio suíno a maior piscina de natação do Estado".
- Descrição privada:** A text input field containing "Seja bem vindo!".
- Button:** A blue button labeled "Criar Atividade".

Right Screenshot (09:57):

- Header:** A back arrow and the text "Criar Atividade".
- Tipo de Atividade:** A text input field containing "Local".
- Esporte:** A text input field containing "Natação".
- Preço:** A text input field containing "200".
- Moeda:** A text input field containing "R\$ - Real".
- Seleção de Local:** A map view showing a location in Florianópolis, Brazil, with labels for "Praça Lauro Muller", "R. Hektor Luz", "Av. Mauro Ribeiro", and "Demétrio Ribeiro".
- Button:** A blue button labeled "Criar Atividade".

Fonte: elaborada pelo autor.

5.1.3 Atualização de Perfil do Usuário

Estas telas permitem a criação e atualização do perfil dos usuários:

- **Foto de Perfil e Nome de Usuário (photo-username-selection.tsx):** Figura 17 - permite aos usuários fazer *upload* de uma foto de perfil e definir um nome de usuário único e seu primeiro e último nome.

Figura 17 - Tela para definir nome, nome de usuário e foto de perfil.

23:54

<

Escolha uma foto de perfil e um nome de usuário

Nome

Miguel

Sobrenome

Grieder

Escolha seu nome de usuário

miguelgrieder

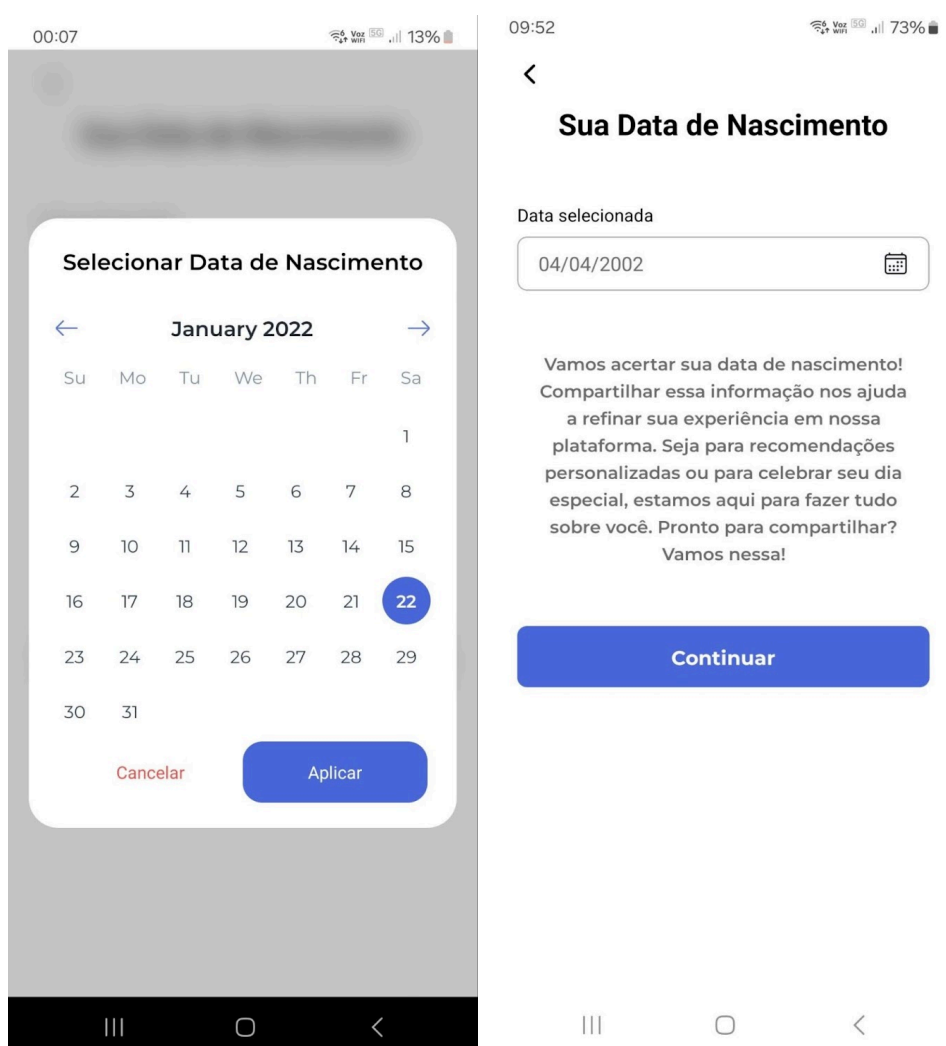
Aqui, você pode personalizar seu perfil com uma foto que te represente. Simplesmente selecione uma imagem do seu dispositivo, e nós cuidamos do resto. Certifique-se de que sua foto reflète quem você é, seja um selfie, uma foto

III ○ <

Fonte: elaborada pelo autor.

- **Data de Nascimento (birthdate-selection.tsx):** Figura 18 - campo para inserir a data de nascimento do usuário.

Figura 18 - Tela para inserir a data de nascimento.



Fonte: elaborada pelo autor.

- **Seleção de Função (role-selection.tsx):** Figura 19 - usuários escolhem seu tipo (Usuário Casual, Treinador, Empresa), determinando suas permissões e funcionalidades disponíveis. Também escolhem seus esportes principais. No caso de Usuário Casual, é seus esportes preferidos, no caso de Treinador, é os esportes principais em que atua e no caso de empresa é os esportes principais que oferece.

Figura 19 - Tela para definir tipo de usuário e esportes principais.



Fonte: elaborada pelo autor.

- **Informações Sociais (social-selection.tsx):** Figura 20 - permite aos usuários adicionar informações para suas redes sociais, outras informações de contato e definir uma biografia para o perfil.

Figura 20 - Tela de atualização de informações sociais.

The image displays two screenshots of a mobile application interface for updating social information. Both screens have a status bar at the top showing the time, signal strength, and battery level.

Left Screenshot (09:53, 73% battery):

- Header:** A back arrow and the title "Conecte-se com outros".
- Text:** "Compartilhe para os outros poderem te conhecer".
- Perfil Bio:** A text input field containing "Um atleta casual!".
- Número de Telefone:** A text input field containing "554899999999".
- Permitir Chamadas Telefônicas:** A toggle switch that is currently off.
- Permitir Mensagens SMS:** A toggle switch that is currently off.
- Permitir WhatsApp:** A toggle switch that is currently on.
- Instagram:** A text input field containing "miguelgrieder".
- Facebook:** A text input field that is empty.

Right Screenshot (00:09, 13% battery):

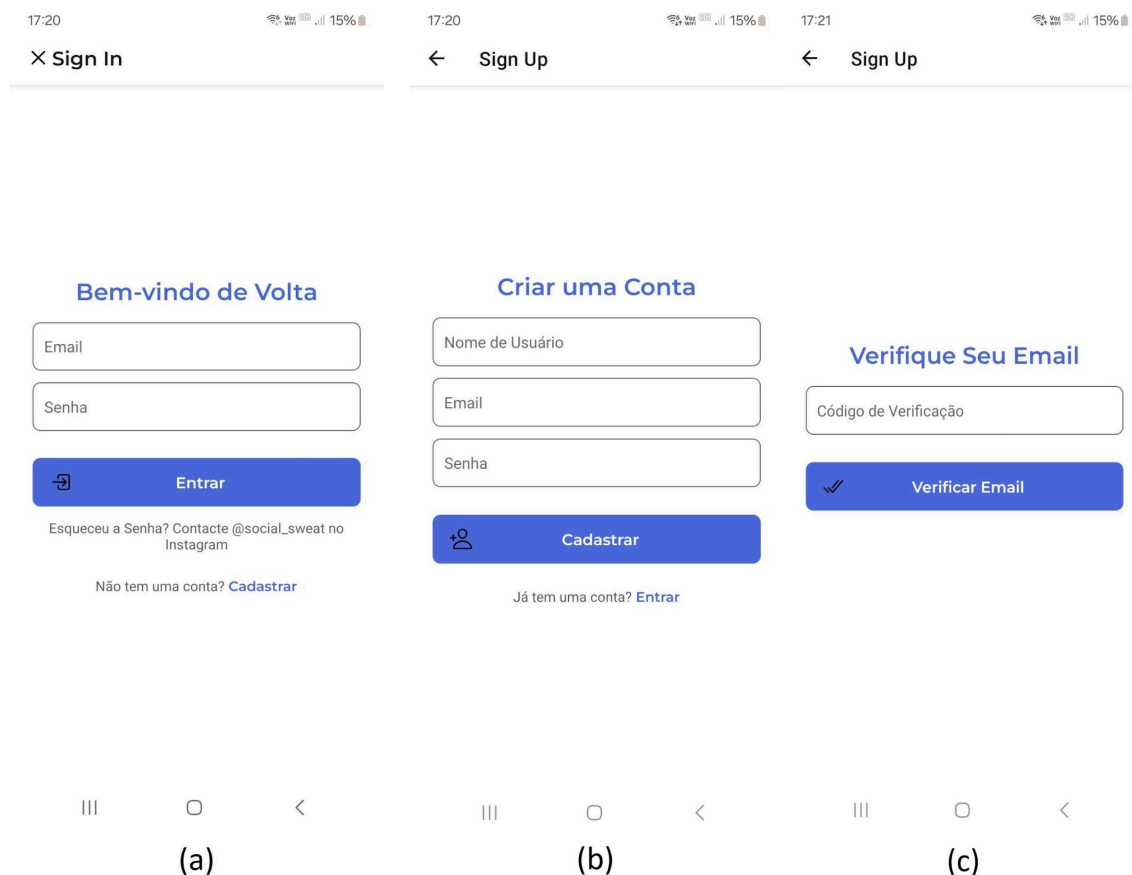
- Header:** A back arrow.
- Número de Telefone:** A text input field with the placeholder "Insira seu número de telefone".
- Instagram:** A text input field with the placeholder "Insira seu nome de usuário do Instagram".
- Facebook:** A text input field with the placeholder "Insira seu nome de usuário do Facebook".
- YouTube:** A text input field with the placeholder "Insira seu canal do YouTube".
- TikTok:** A text input field with the placeholder "Insira seu nome de usuário do TikTok".
- Strava:** A text input field containing "miguelgrieder".
- Footer:** A blue button labeled "Continuar".

Fonte: elaborada pelo autor.

5.1.4 Telas de *Login*

As telas de *login* e cadastro, apresentadas na Figura 21, permitem que o usuário acesse sua conta ou realize um novo cadastro. A criação de conta, ilustrada na Figura 21(b), exige o nome de usuário, *e-mail* e senha. Após o preenchimento desses campos, é solicitado um código de verificação enviado para o *e-mail* informado, conforme exibido na Figura 21(c). A tela de *login*, mostrada na Figura 21(a), requer o *e-mail* e a senha do usuário. O fluxo de recuperação de senha não foi implementado, sendo o processo realizado manualmente, conforme indicado na tela da Figura 21(a).

Figura 21 - Telas de Login(a) / Cadastro(b) / Verificação de código(c).



Fonte: elaborada pelo autor.

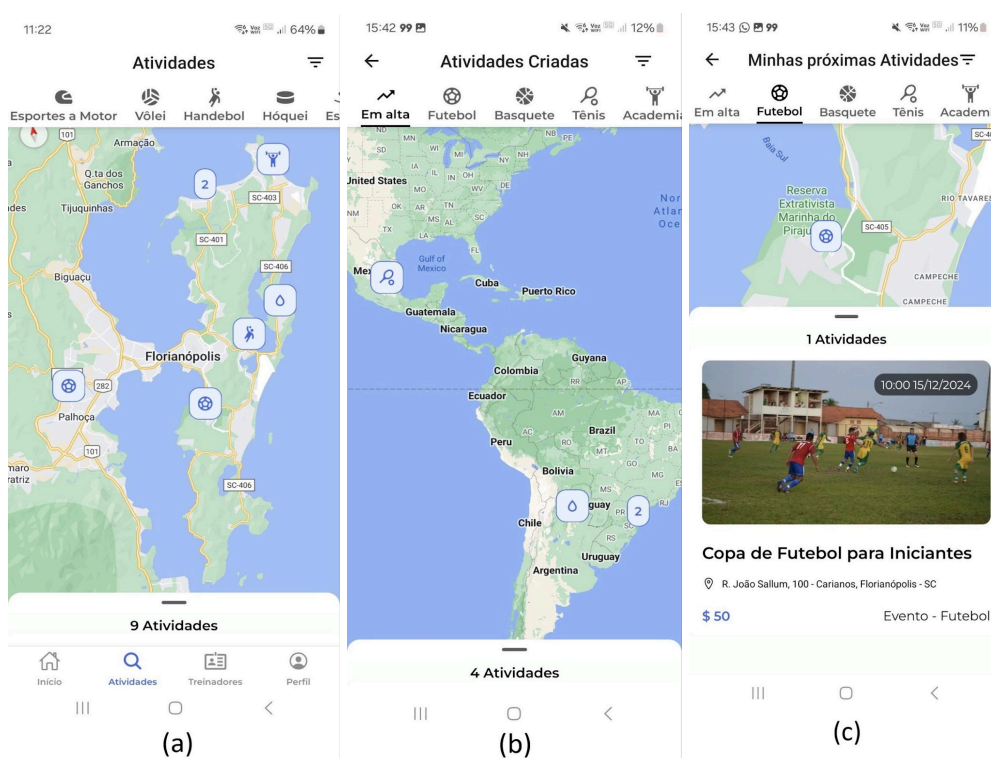
5.1.5 Listagem de Atividades

Estas telas, vistas na Figura 22, utilizam um componente de listagem compartilhado, adaptando-se aos diferentes contextos, que apenas mudam o filtro inicial e permanente aplicado. Essas mudanças no filtro inicial podem ser:

- **Atividades Gerais (activities.tsx):** Figura 22(a) - exibe todas as atividades disponíveis, essa tela permite aos usuários explorar e participar de atividades de interesse. É acessada a partir do menu de navegação inferior, na aba Atividades.

- **Atividades Criadas (hosted-activities.tsx):** Figura 22(b) - mostra apenas as atividades que o usuário criou, facilitando o gerenciamento e a edição das mesmas. Disponível a partir da página inicial do aplicativo.
- **Minhas Próximas Atividades (my-next-activities.tsx):** Figura 22(c) - lista as atividades futuras nas quais o usuário está inscrito, fornecendo seu calendário de eventos. Pode ser acessada a partir da página inicial do aplicativo.

Figura 22 - Telas de Listagem de atividades: Atividades(a) / Atividades criadas(b) / Minhas próximas atividades (c).



Fonte: elaborada pelo autor.

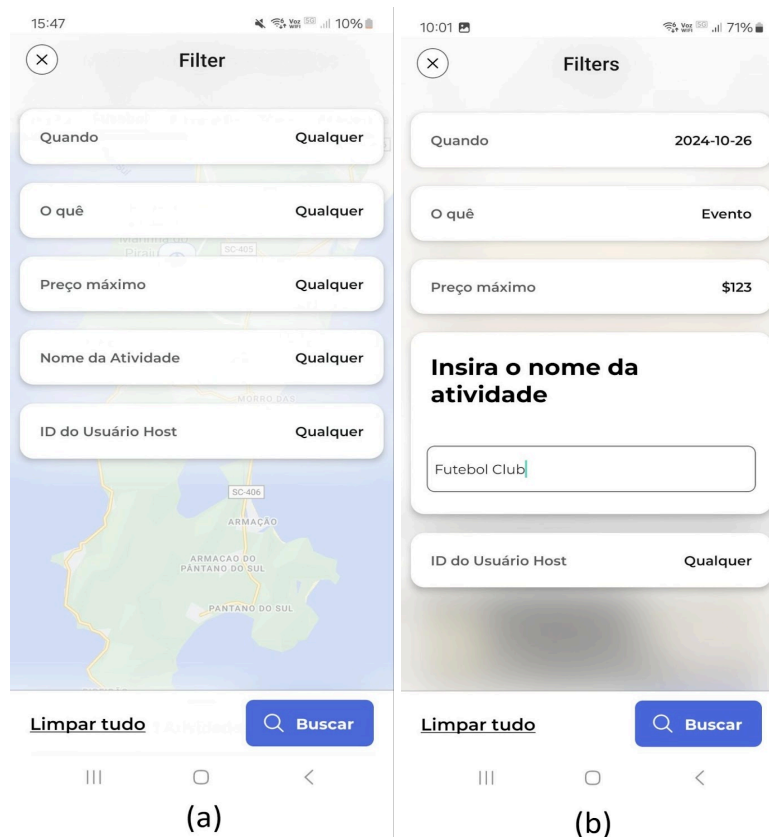
5.1.6. Filtro de Atividades

A página de filtro (`activities-filter.tsx`), visto na Figura 23, aprimora a funcionalidade de busca, permitindo aos usuários refinar os resultados de atividades. Estes filtros funcionam para todas as telas que possuem o componente de listagem de atividades, como discutido tópico 5.1.5 Listagem de Atividades.

- **Filtros Avançados:** opções para filtrar por tipo de atividade, data, preço, nome e identificador do criador.
- **Interface Intuitiva:** uso de *sliders*, caixas de seleção e menus suspensos para facilitar a aplicação dos filtros desejados.
- **Resultados Dinâmicos:** atualização em tempo real dos resultados conforme os filtros são aplicados, proporcionando uma navegação eficiente.

A tela (a) da Figura 23 apresenta a página em seu estado inicial, sem nenhum filtro. Já a tela (b) exibe a página com alguns filtros ativados e a aba de filtragem por nome de atividade aberta.

Figura 23 - Tela de Filtros de Atividades - estado inicial (a) / filtros aplicados (b).

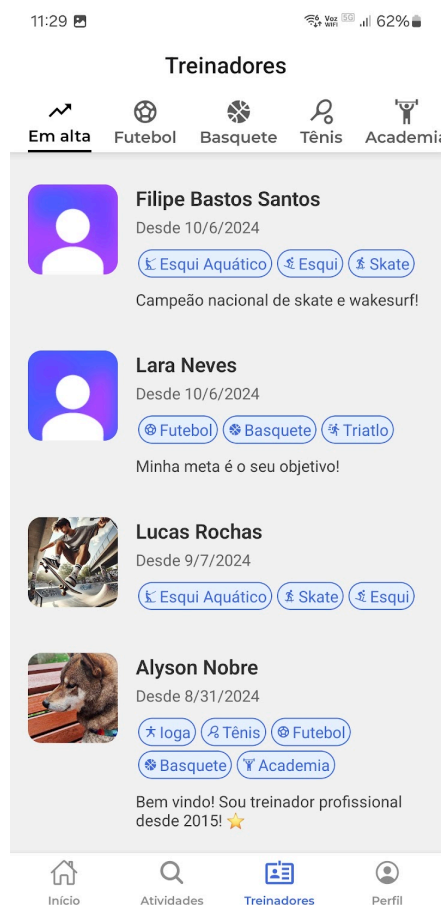


Fonte: elaborada pelo autor.

5.1.7 Listagem de Treinadores

Esta tela (coaches.tsx), apresentada na Figura 24, exibe uma lista com os treinadores disponíveis. Disponibiliza um seletor que permite filtrar os treinadores com base nos esportes que eles atuam. Além disso, ao clicar em um treinador na lista, o perfil completo do usuário é exibido, fornecendo detalhes abrangentes sobre o treinador selecionado. O perfil de um usuário é explicado no capítulo 5.1.8 Perfil do Usuário.

Figura 24 - Tela de Listagem de Treinadores.



Fonte: elaborada pelo autor.

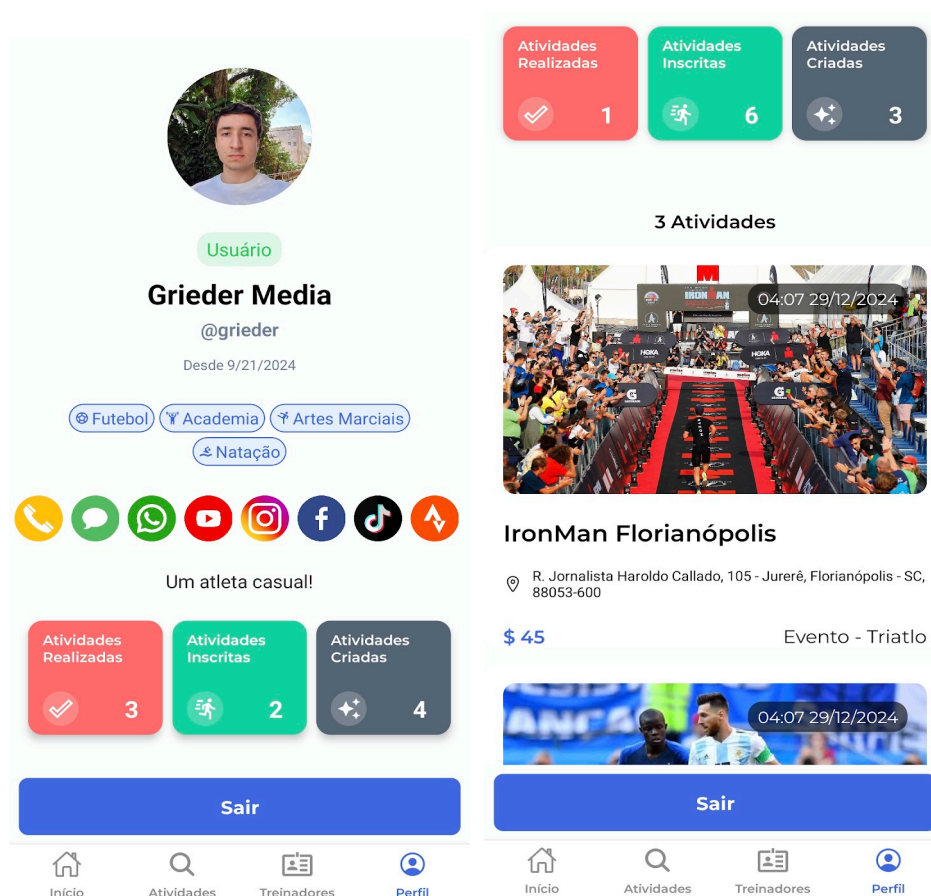
5.1.8 Perfil do Usuário

A tela de perfil de um usuário (profile.tsx), vista na Figura 25, exibe todas as informações relevantes do usuário. São seus componentes:

- **Informações Pessoais:** inclui foto de perfil, nome, usuário, data de criação da conta, tipo de usuário, biografia, esportes principais e informações de contato.
- **Métricas de usuário:** informa quantas atividades o usuário já participou, atividades inscritas (presentes e futuras) e criadas.

- **Atividades Criadas:** listagem dedicada para mostrar as atividades que o usuário criou.
- **Opções de Sair:** botão para sair (*logout*) de sua conta, caso esteja no perfil próprio.

Figura 25 - Tela do perfil do usuário.



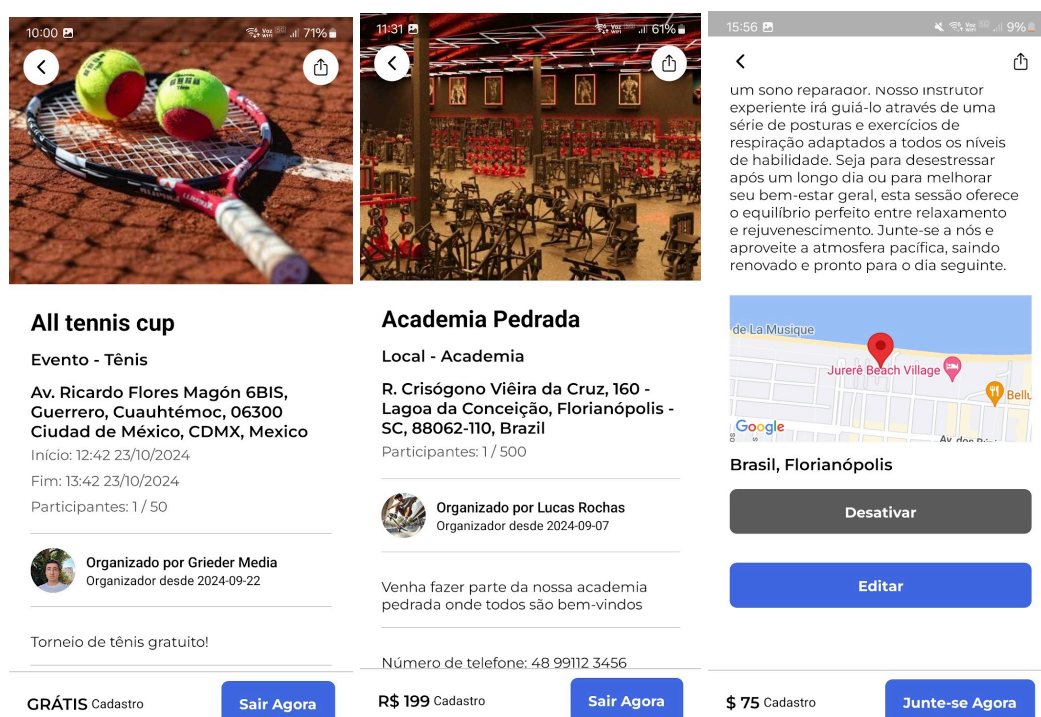
Fonte: elaborada pelo autor.

5.1.9 Detalhes da Atividade

Esta tela (`activity/[id].tsx`), vista na Figura 26, fornece uma os detalhes de uma atividade específica:

- **Informações Completas:** exibição de imagem, nome, tipo de evento, esporte, data, hora, local, descrição, número de participantes, preço, organizador e descrições.
- **Mapa Integrado:** visualização da localização da atividade no mapa, facilitando visualizar ela em um aplicativo de mapa próprio.
- **Interação do Usuário:** botões para participar ou sair da atividade.
- **Editar Atividade:** caso seja o criador da atividade, são exibidos os botões para ativar/desativar a atividade e editar as informações da mesma.

Figura 26 - Tela de Detalhes da Atividade.



Fonte: elaborada pelo autor.

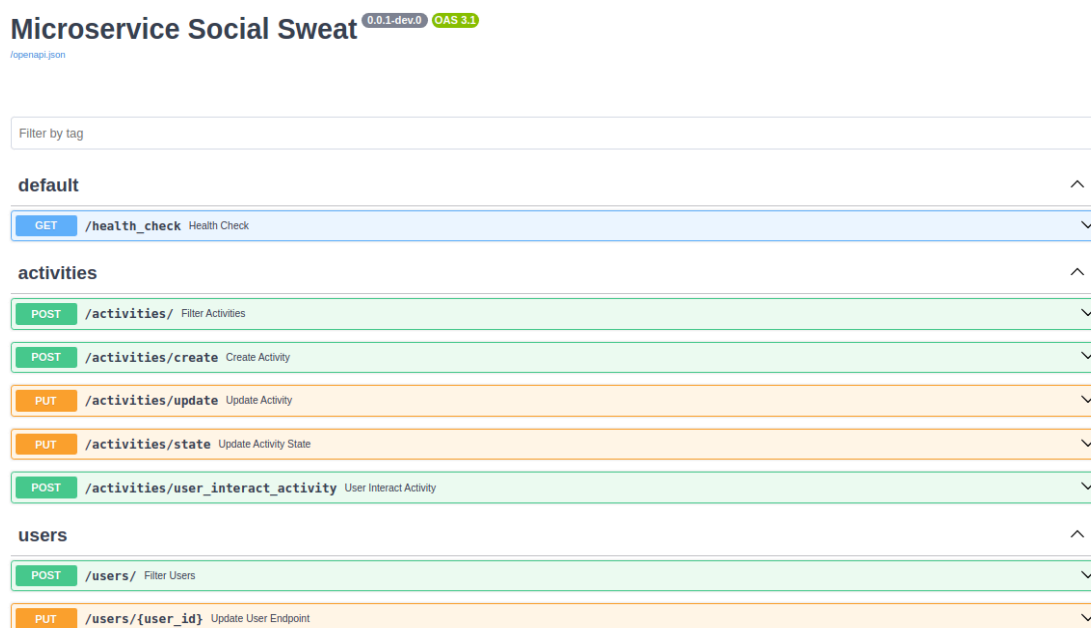
5.2 BACKEND

O *backend* do Social Sweat foi desenvolvido em Python utilizando FastAPI, um *framework* moderno e de alto desempenho para a construção de APIs Rest. A arquitetura do *backend* segue uma abordagem de microsserviços, fornecendo escalabilidade,

manutenção simplificada e desempenho eficiente. Nas próximas seções serão detalhadas as principais funcionalidades implementadas pelo *backend*.

A Figura 27 apresenta o Swagger do microserviço, uma página acessada por qualquer navegador de *internet* gerada automaticamente pelo *framework* que exibe todas as rotas disponíveis, junto com os dados esperados de entrada e saída. Essa página serve para documentação, facilita o compartilhamento com outros desenvolvedores e permite o uso direto da API, possibilitando a realização de requisições aos serviços disponíveis diretamente pelo Swagger.

Figura 27 - Swagger do microserviço.



Fonte: elaborada pelo autor.

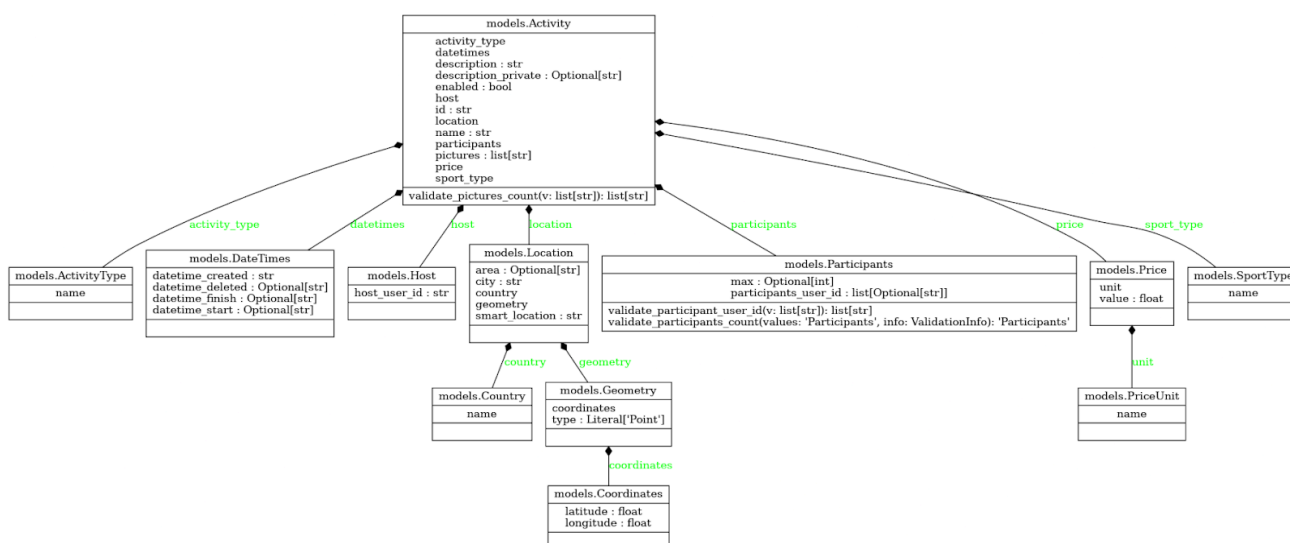
A seguir estão listados todas as rotas do microserviço desenvolvido, contendo seu tipo de requisição HTTP, *path*, nome, descrição e nomes das classes de entrada e resposta da rota.

As próximas seções apresentam figuras dos diagramas das classes utilizadas na implementação. Algumas classes possuem atributos que são de tipos de classes que estão em outra figura de diagrama, e para abreviação é mostrado apenas o título do diagrama referenciado. Classes que possuem apenas o atributo “name” são do tipo ENUM, e estão detalhadas na seção 4.6.

5.2.1 Endpoints de Atividades (/activities/)

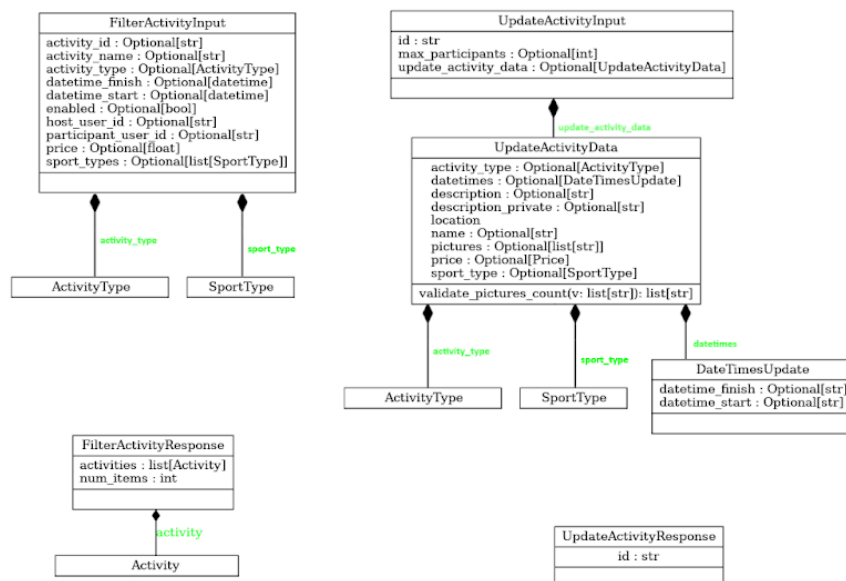
As Figuras 28, 29 e 30 apresentam a modelagem das classes de modelos relacionados às atividades. Na primeira figura é mostrado o diagrama da classe Atividade, e todas as classes que a compõem. Nas figuras seguintes estão os diagramas de entrada e saída das rotas.

Figura 28 - Modelagem de classe de Atividade



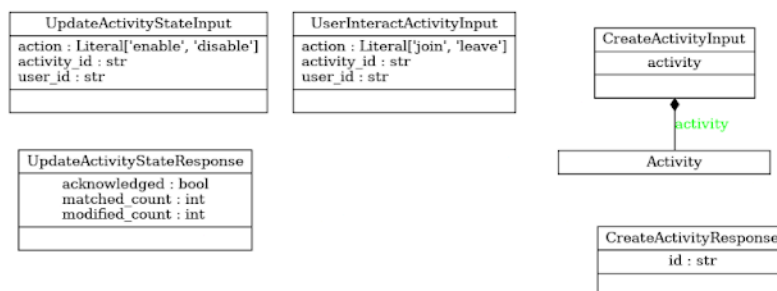
Fonte: elaborada pelo autor.

Figura 29 - Modelagem de classe de modelos relacionados a atividade (a)



Fonte: elaborada pelo autor.

Figura 30 - Modelagem de classe de modelos relacionados a atividade (b)



Fonte: elaborada pelo autor.

A seguir são listados os *endpoints* para acesso às atividades físicas disponibilizadas no aplicativo:

- **POST /activities/ - Filtrar Atividades:** permite aos usuários filtrar atividades com base em critérios específicos, como tipo, esporte, local, data, etc. Recebe parâmetros de filtro no corpo da requisição e retorna uma lista de atividades que correspondem aos critérios.

Entrada: FilterActivityInput

Saída: FilterActivityResponse

- **POST /activities/create - Criar Atividade:** responsável por criar uma nova atividade no sistema. Recebe os detalhes da atividade (nome, descrição, tipo, data, hora, local, etc.) e armazena no banco de dados, associando-a ao usuário que a criou.

Entrada: CreateActivityInput

Saída: CreateActivityResponse

- **PUT /activities/update - Atualizar Atividade:** permite aos usuários atualizarem informações de uma atividade existente. Recebe o ID da atividade e os novos dados a serem atualizados, garantindo que apenas o criador ou usuários com permissões adequadas possam realizar a atualização.

Entrada: UpdateActivityInput

Saída: UpdateActivityResponse

- **PUT /activities/state - Mudar Estado da Atividade:** facilita a alteração do estado de uma atividade, como ativar e desativar. Utilizado para gerenciar a disponibilidade das atividades conforme necessário.

Entrada: UpdateActivityStateInput

Saída: UpdateActivityStateResponse

- **POST /activities/user_interact_activity - Interagir com Atividade:** gerencia a interação dos usuários com as atividades, permitindo que participem ou saiam delas. Atualiza a lista de participantes da atividade se necessário.

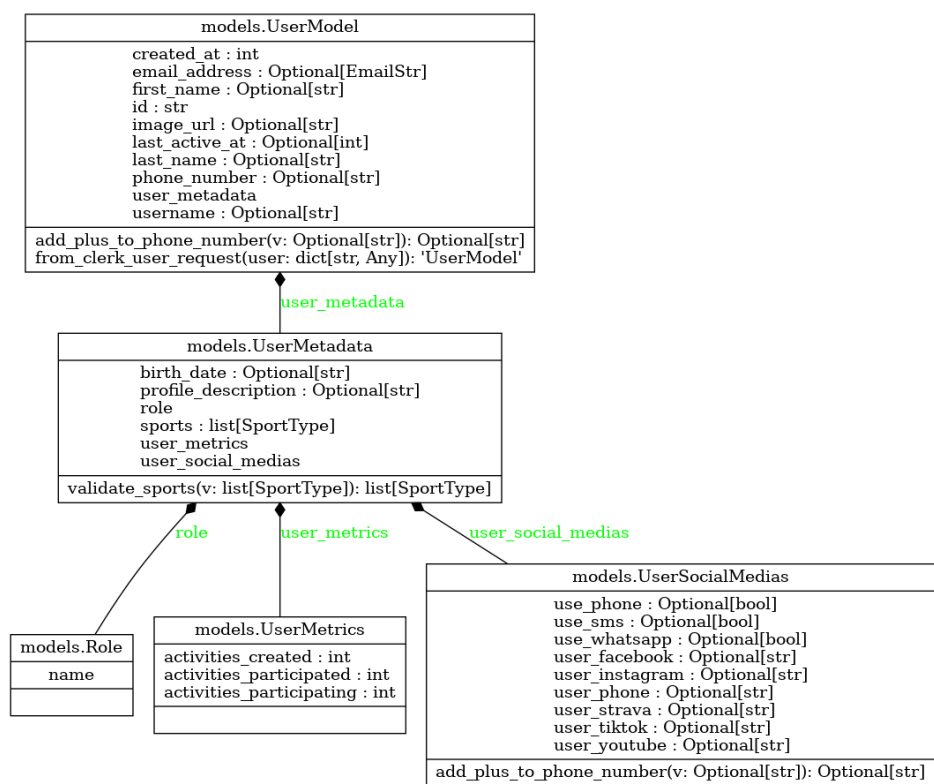
Entrada: UserInteractActivityInput

Saída: JSON (sem classe definida)

5.2.2 Endpoints de Usuários (/users/)

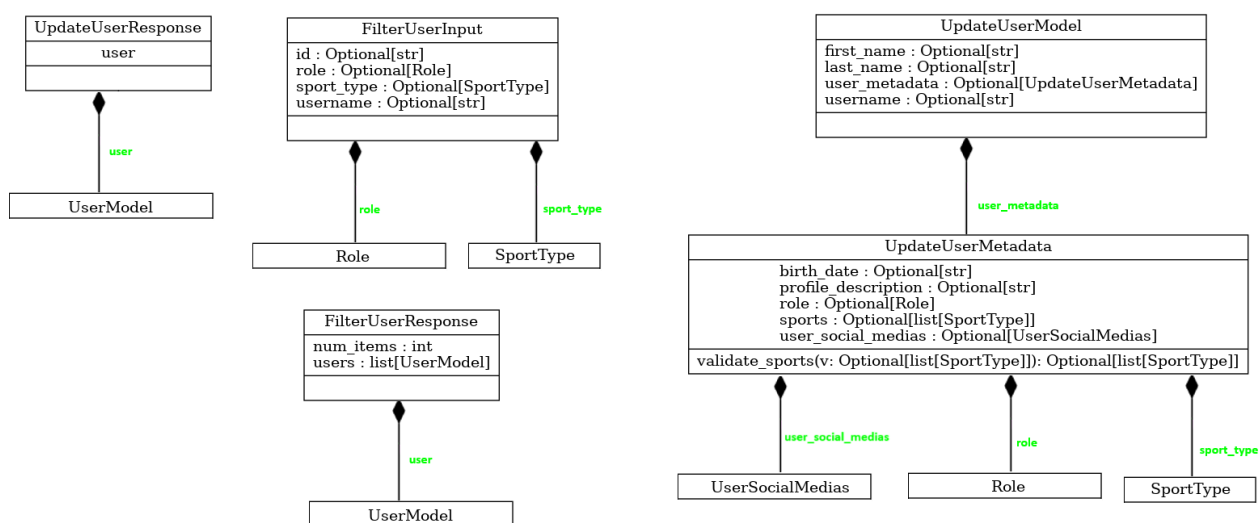
As Figuras 31 apresenta o modelo da classe UserModel e suas classes dependentes, que é o modelo base de usuário, e a Figura 32 apresentam a modelagem das classes de modelos relacionados ao usuário utilizada pelas rotas

Figura 31 - Modelagem de classe de UserModel



Fonte: elaborada pelo autor.

Figura 32 - Modelagem de classe de outros modelos relacionados ao usuário



Fonte: elaborada pelo autor.

A seguir são listadas as rotas sobre usuários disponibilizadas pelo *backend*:

- **POST /users/filter - Filtrar Usuários:** permite a busca e filtragem de usuários com base em critérios como tipo de usuário, id e esportes principais. Retorna uma lista de usuários que correspondem aos filtros aplicados.

Entrada: FilterUserInput

Saída: FilterUserResponse

- **PUT /users/{user_id} - Atualizar Dados do Usuário:** responsável por atualizar as informações do perfil do usuário. Recebe o ID do usuário na URL como parâmetro e os novos dados a serem atualizados, como nome, biografia, esportes principais e informações de contato.

Entrada: user_id e UpdateUserModel

Saída: UpdateUserResponse

5.2.3 Segurança e Autenticação

A autenticação é realizada utilizando tokens JWT. O microserviço possui um *middleware* que recebe o *token* JWT do aplicativo para autenticação segura, garantindo que apenas usuários autenticados possam acessar e interagir com os *endpoints* protegidos. São esses *endpoints* protegidos funcionalidades ligadas a um usuário como criar atividade, entrar e sair de atividades, atualizar dados do perfil, ativar ou desativar atividades é verificado se o *token* informado realmente é deste usuário a ação que ele deseja fazer. Para os *endpoints* básicos de listar e filtrar atividades e usuários, não é verificado o JWT para permitir usuários não logados de terem uma interação básica com o aplicativo, visualizando as atividades e treinadores.

O *backend* emprega um mecanismo de controle de acesso baseado em papéis, assegurando que cada tipo de usuário (Casual, Treinador, Empresa) tenha acesso apenas às funcionalidades permitidas. A verificação é realizada em conjunto com o JWT, de modo a assegurar que o usuário que enviou o *token* é o mesmo da ação, e que o tipo de usuário tem permissão para executar aquela ação. Caso contrário, a permissão para execução é negada.

Como visto na Figura 33, após validar o JWT recebido, o *middleware* salva no *state* da requisição o ID de usuário deste *token*, para ser verificado como visto na Figura 34 onde for executado alguma ação ligada ao usuário, por exemplo verificar se o *host_user_id* (id de usuário do criador da atividade) é o mesmo que o do JWT recebido.

Figura 33 - *Middleware* de autenticação.

```

16 class ClerkMiddleware(BaseHTTPMiddleware): 5 usages 1 Miguel Duarte Grieder
17     def __init__(self, app: ASGIApp) -> None: 1 Miguel Duarte Grieder
18         super().__init__(app)
19         self.public_key = settings.clerk_jwt_public_key
20
21 @
22     async def dispatch(self, request: Request, call_next: Callable[[Request], Any]) -> Any: 1 Miguel Duarte Grieder
23         if request.url.path in [
24             "/docs",
25             "/openapi.json",
26             "/activities/",
27             "/activities/",
28             "/users/",
29         ]:
30             return await call_next(request)
31         if request.method == "OPTIONS":
32             # Allow preflight requests to pass through without authentication
33             return await call_next(request)
34
35         auth_header: Optional[str] = request.headers.get("Authorization")
36         if not auth_header or not auth_header.startswith("Bearer "):
37             raise HTTPException(status_code=401, detail="Unauthorized")
38         token: str = auth_header.split("Bearer ")[1]
39
40         try:
41             payload = jwt.decode(token, key=self.public_key, algorithms=["RS256"])
42             request.state.user_id = payload.get("sub")
43         except PyJWTError as e:
44             raise HTTPException(status_code=401, detail="Invalid token") from e
45
46         # Proceed to the next middleware or endpoint
47         response = await call_next(request)
48         return response

```

Fonte: elaborada pelo autor.

Figura 34 - Função para verificar se o `user_id` recebido é do mesmo usuário do JWT.

```

4 def verify_user_id_is_the_same_from_jwt(request: Request, user_id: str) -> None: 8 usages 1 Miguel Duarte Grieder
5     if not user_id == request.state.user_id or not request.state.user_id or not user_id:
6         raise HTTPException(
7             status_code=401,
8             detail="Received JWT Token is not from the user id received to perform this action!",
9         )

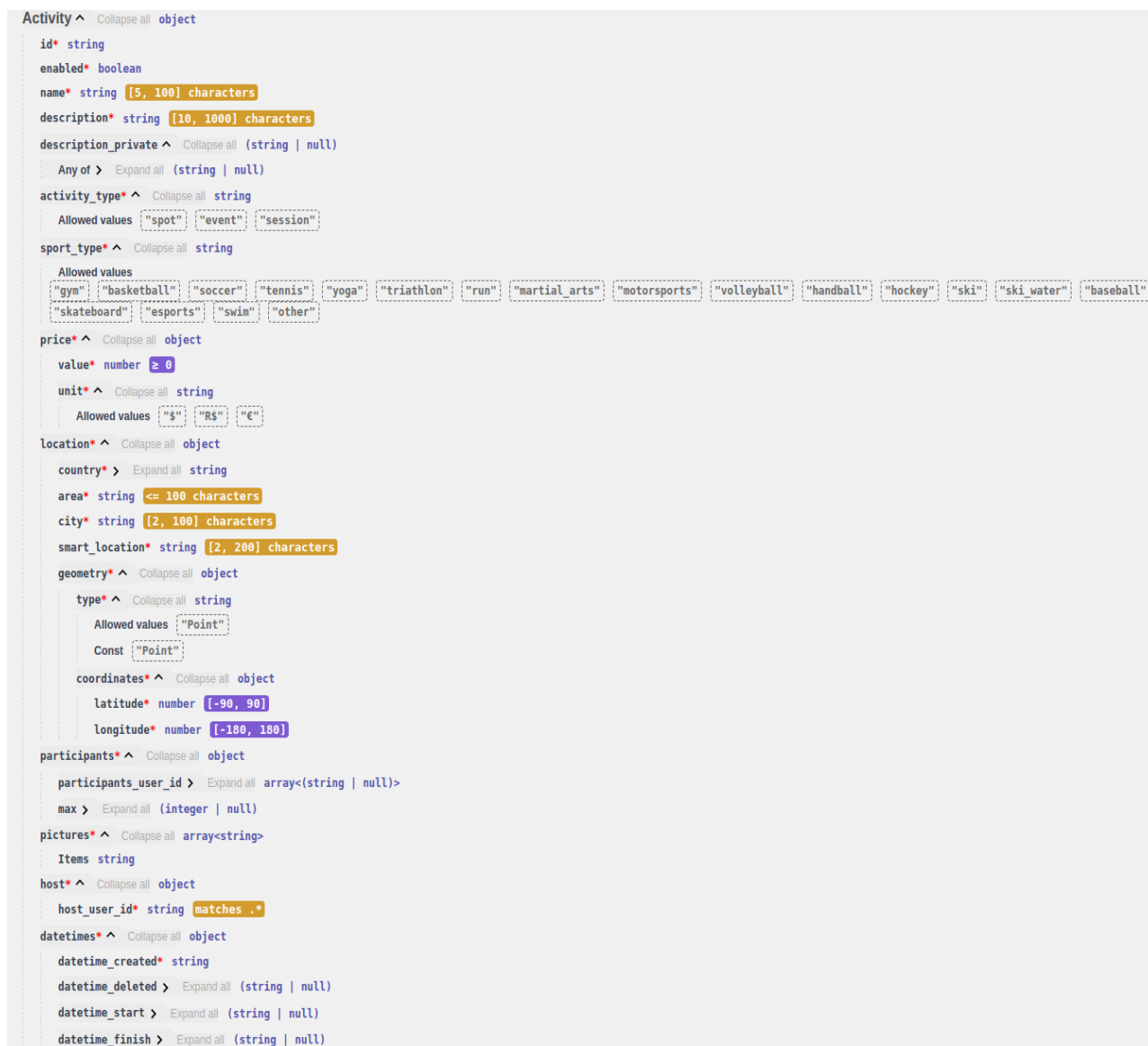
```

Fonte: elaborada pelo autor.

A validação dos dados recebidos pelo *backend* é realizada utilizando modelos de dados definidos com Pydantic para garantir a integridade e consistência das informações. Ao ler e salvar no banco de dados ou no metadata do usuário, é sempre feita a validação

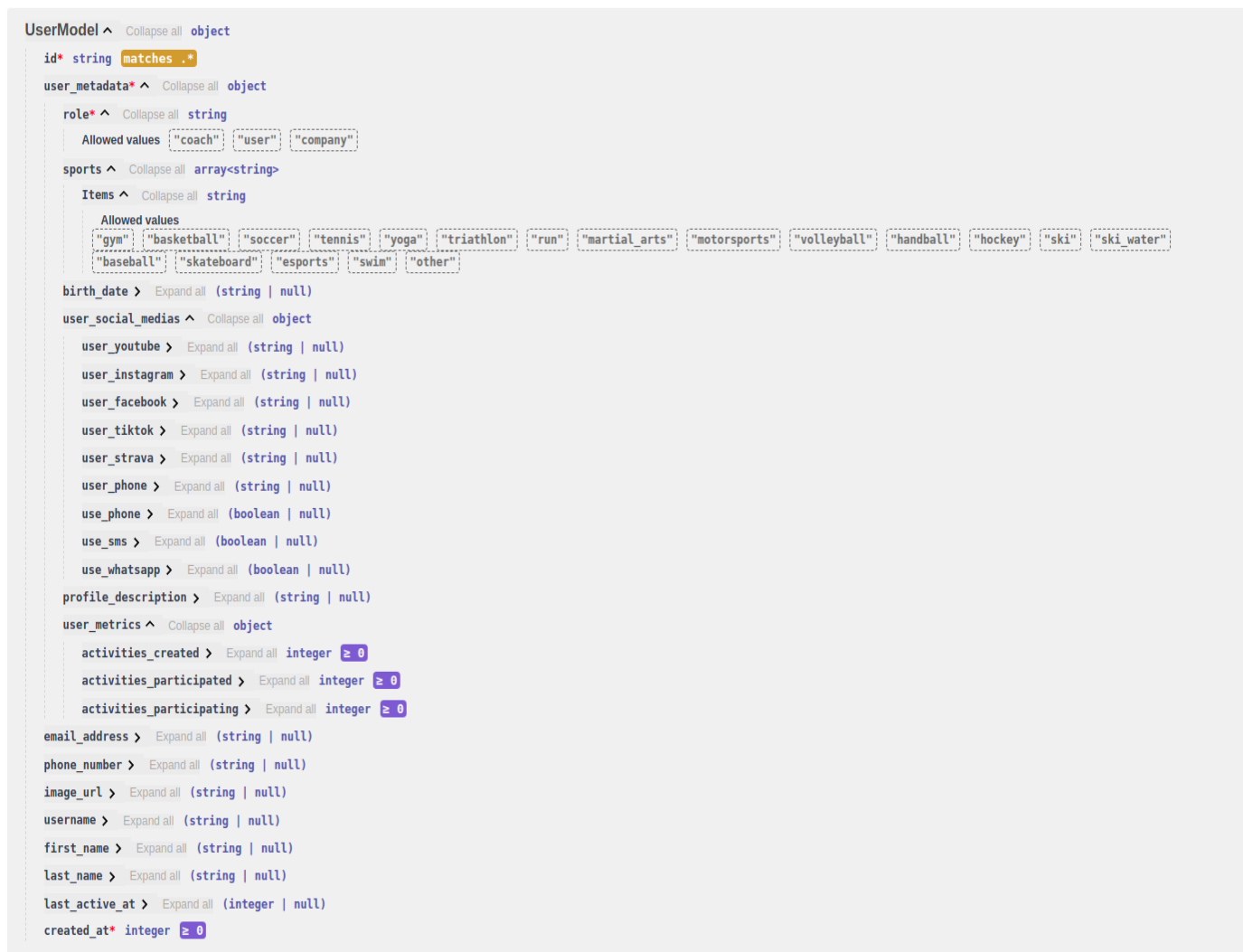
dos dados. Nas figuras 35 e 36 são mostrados os modelos de atividade (Activity) e usuário (User) no Swagger, com as validações de dados feitas pelo Pydantic.

Figura 35 - Modelo de atividade visto pelo Swagger do microserviço.



Fonte: elaborada pelo autor.

Figura 36 - Modelo de usuário visto pelo Swagger do microserviço.



Fonte: elaborada pelo autor.

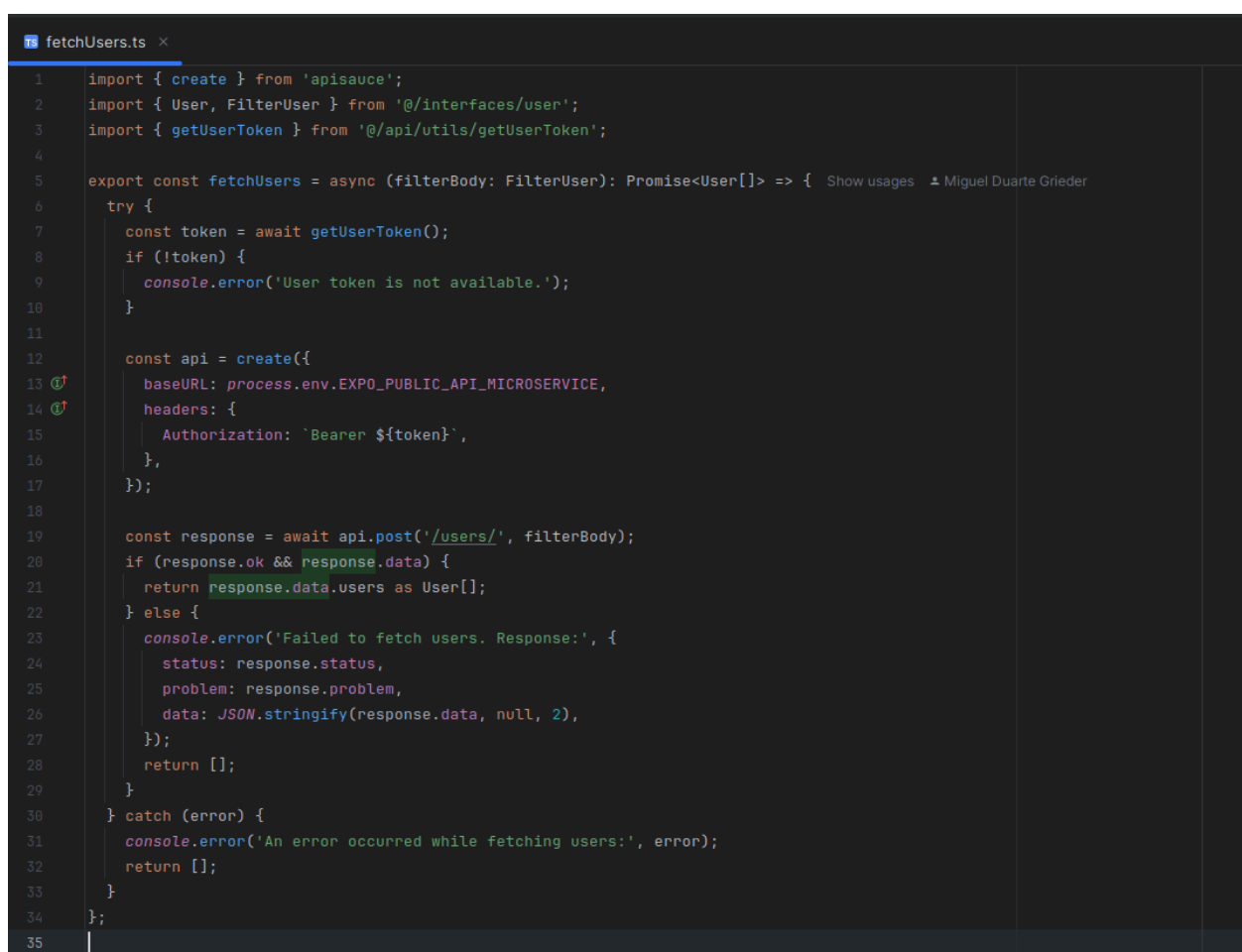
5.2.4 Integração com o *frontend*

O *backend* está totalmente integrado com o *frontend* desenvolvido em React Native, utilizando chamadas via HTTP para comunicação com os *endpoints* da API. A utilização de bibliotecas no *frontend* como *apisauce* facilita a gestão das requisições e o tratamento de respostas e erros, garantindo uma comunicação eficiente e robusta entre as camadas do aplicativo.

No diretório `src/api` do *frontend* há arquivos que contêm uma função específica para cada rota do microserviço, facilitando o uso dela em diferentes pontos do aplicativo.

Ela também garante receber os modelos já esperados pelo microserviço. Na Figura 37, há o exemplo da função `fetchUsers` no aplicativo, que é a função que realiza chamadas a rota de consulta de usuários.

Figura 37 - Exemplo de função no *frontend* de chamada ao microserviço: `fetchUsers`.



```
1 import { create } from 'apisauce';
2 import { User, FilterUser } from '@interfaces/user';
3 import { getUserToken } from '@api/utils/getUserToken';
4
5 export const fetchUsers = async (filterBody: FilterUser): Promise<User[]> => {
6   try {
7     const token = await getUserToken();
8     if (!token) {
9       console.error('User token is not available.');
```

Fonte: elaborada pelo autor.

5.2.5 Testes

O *backend* possui testes abrangentes que incluem estilização (black), qualidade de código (ruff), tipagem estática (mypy) e testes de código (py312 - pytest). Todos esses

testes são orquestrados pelo tox, permitindo que, com um único comando, todos sejam executados em um ambiente de teste separado do ambiente de desenvolvimento. Na Figura 38 é mostrado o relatório final da execução do comando tox.

Figura 38 - Retorno da execução dos testes presentes no tox.

```

===== 4 passed, 1 warning in 1.73s =====
py312: OK ✓ in 7.11 seconds
ruff: install_package: python -I -m pip install --force-reinstall --no-deps /home/migueldg/Documents/github/microservice-social-sweat/.tox/.tmp/package/409/microservice_social_sweat-0.0.1.dev0.tar.gz
ruff: commands[0]: ruff check .
All checks passed!
ruff: OK ✓ in 1.82 seconds
mypy: install_package: python -I -m pip install --force-reinstall --no-deps /home/migueldg/Documents/github/microservice-social-sweat/.tox/.tmp/package/410/microservice_social_sweat-0.0.1.dev0.tar.gz
mypy: commands[0]: mypy --config-file /home/migueldg/Documents/github/microservice-social-sweat/pyproject.toml .
Success: no issues found in 28 source files
mypy: OK ✓ in 2.19 seconds
black: commands[0]: black --config /home/migueldg/Documents/github/microservice-social-sweat/pyproject.toml --check .
All done! ✨ 🍰 ✨
28 files would be left unchanged.
py312: OK (7.11=setup[2.65]+cmd[4.46] seconds)
ruff: OK (1.82=setup[1.81]+cmd[0.01] seconds)
mypy: OK (2.19=setup[1.90]+cmd[0.30] seconds)
black: OK (0.12=setup[0.01]+cmd[0.12] seconds)
congratulations :) (11.30 seconds)
(.venv) migueldg@HP-XB:~/Documents/github/microservice-social-sweat$

```

Fonte: elaborada pelo autor.

5.3 INFRAESTRUTURA EM NUVEM

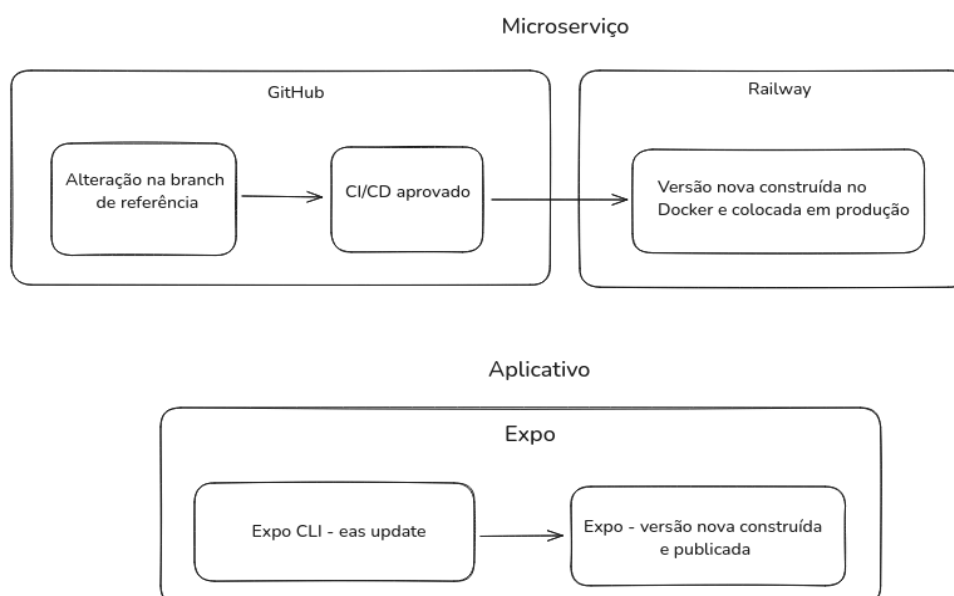
Para garantir a disponibilidade, escalabilidade e desempenho do Social Sweat, a infraestrutura do aplicativo foi planejada e implementada utilizando serviços de nuvem modernos. Os dados do aplicativo são salvos apenas em nuvem e atrelados à conta do usuário de forma segura, permitindo que o usuário use sua conta em diversos dispositivos diferentes e que apenas ele consiga alterar as informações de sua conta. A seguir, são descritos os principais componentes da infraestrutura.

O acesso ao aplicativo na nuvem é realizado por meio do aplicativo móvel e plataforma Expo Go, que serve como um *hub* para acessar projetos de aplicativos móveis. O endereço de acesso ao Social Sweat está disponível na biografia da página oficial do aplicativo no Instagram: https://www.instagram.com/social_sweat/.

A Figura 39 descreve o fluxo de atualização dos serviços em nuvem para o microsserviço (*backend*) e o aplicativo (*frontend*). No caso do microsserviço, o processo começa com uma alteração no código-fonte no GitHub que é feita na *branch* de referência apontada pelo Railway. Após a alteração, e a aprovação do fluxo de integração e entrega contínua (CI/CD) no GitHub, a nova versão é automaticamente construída em um container Docker e colocada em produção no Railway.

Para o aplicativo, o fluxo ocorre por meio da plataforma Expo usando seu sistema CLI no próprio terminal do projeto em desenvolvimento. Quando há uma atualização no aplicativo em que se deseja atualizar o aplicativo no Expo, o comando "eas update" é executado manualmente, o que inicia o processo de atualização. Uma vez que a atualização é concluída, uma nova versão do aplicativo é construída e publicada na Expo, tornando-a acessível aos usuários finais.

Figura 39 - Diagrama de atualização dos serviços em nuvem.



Fonte: elaborada pelo autor.

5.3.1 Hospedagem do *Backend* e Banco de Dados

O *backend* desenvolvido com FastAPI e o banco de dados MongoDB foram hospedados na plataforma Railway, que oferece uma infraestrutura simples, robusta e escalável para aplicações modernas.

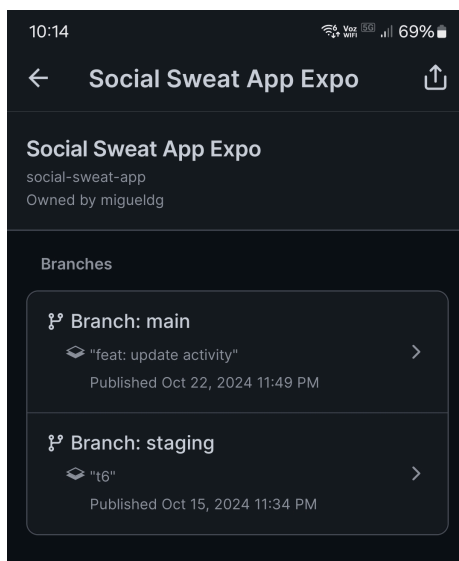
Os serviços do *backend* foram containerizados utilizando Docker, permitindo uma implantação consistente e simplificada em diferentes ambientes. Os contêineres Docker garantem que todas as dependências e configurações necessárias estejam corretamente configuradas, facilitando o processo de desenvolvimento e implantação.

O MongoDB, utilizado como banco de dados, armazena todas as informações das atividades e interações com os usuários de forma eficiente e flexível. A integração com Railway garante alta disponibilidade e segurança dos dados.

5.3.2 Hospedagem do *Frontend*

O *frontend* desenvolvido em React Native foi hospedado utilizando o Expo, uma plataforma que facilita o desenvolvimento e a distribuição de aplicativos móveis. A utilização do Expo Go permite que qualquer usuário com o aplicativo instalado em seu *smartphone* possa acessar e testar o Social Sweat de forma rápida e eficiente, e seu acesso é exibido na Figura 40. O acesso ao aplicativo é feito conforme descrito no início da seção 5.3.

Figura 40 - Visualização do projeto Social Sweat App Expo no Expo Go.

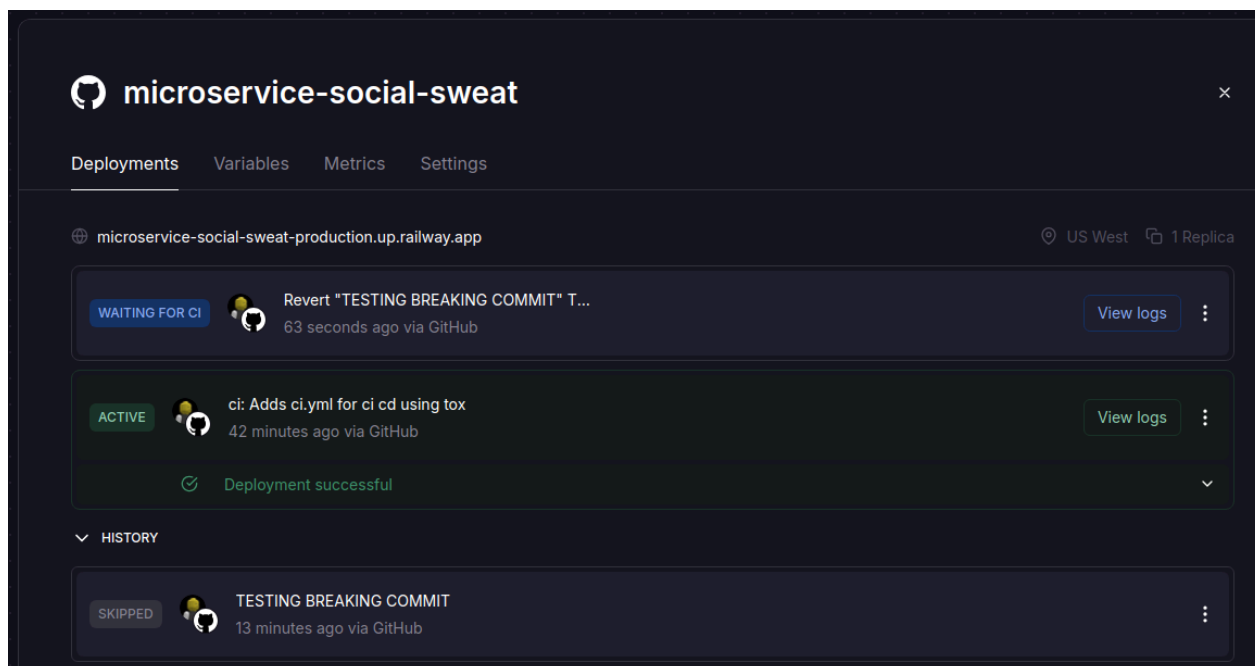


Fonte: elaborada pelo autor.

5.3.3 Integração Contínua (CI) e a Entrega Contínua (CD)

Para o *backend* (microserviço FastAPI), que possui os testes explicados no tópico 5.2.5, sempre que um *commit* é adicionado à *branch* de produção do repositório no GitHub, um *workflow* do GitHub Actions é automaticamente executado, realizando todos os testes com o *tox*. Isso garante a confiabilidade do código mesmo em um ambiente de execução isolado. O servidor na Railway aguarda a conclusão dos testes do GitHub Actions e coloca as novas alterações em produção automaticamente somente se os testes forem bem-sucedidos. Na Figura 41 é possível ver no Railway a espera pelo resultado do CI para colocar as novas alterações em produção.

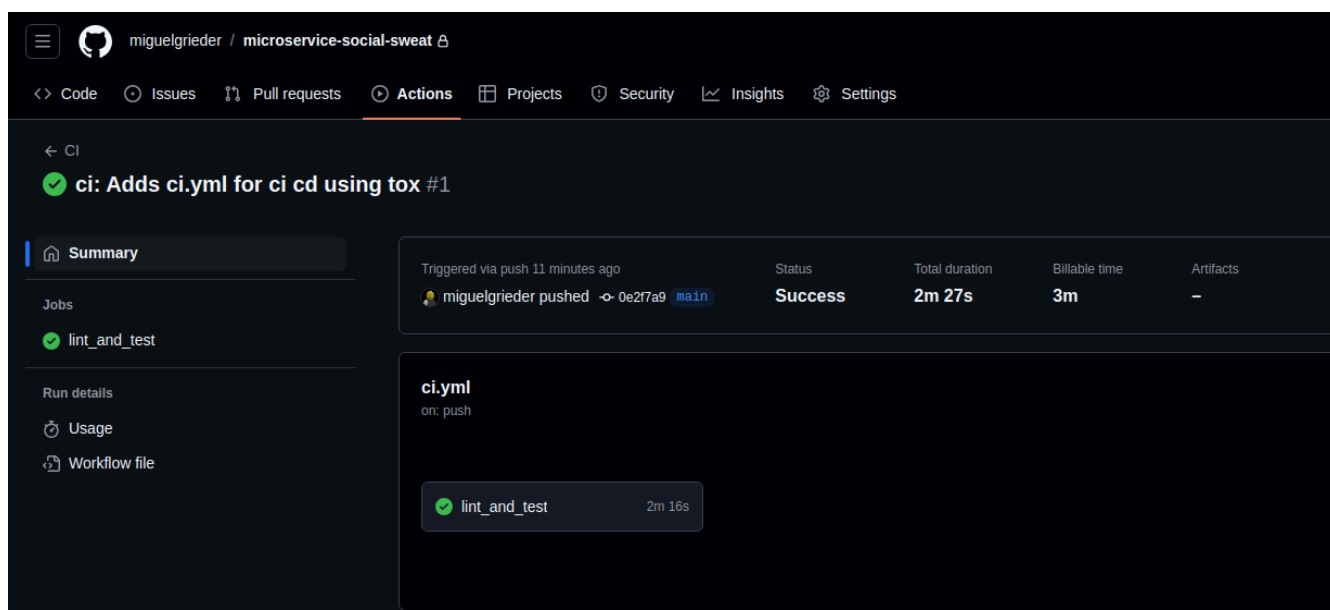
Figura 41 - Visualização dos diferentes envios a produção do microserviço no Railway.



Fonte: elaborada pelo autor.

No exemplo da Figura 42, foi adicionado o *commit* "ci: Adds ci.yml for CI/CD using tox", e na imagem é possível ver o relatório após a finalização bem-sucedida do fluxo.

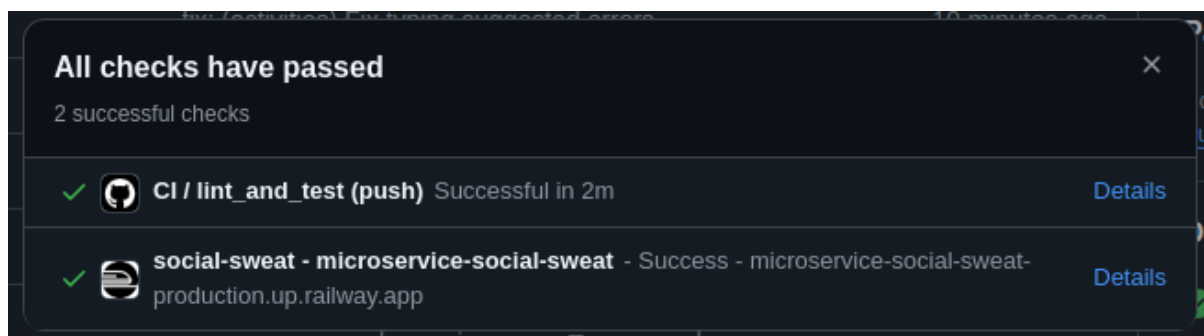
Figura 42 - Etapa CI no Gitlab executada com sucesso.



Fonte: elaborada pelo autor.

A Figura 43 ilustra o processo do GitHub Actions que foi executado com sucesso, seguido pelo *deploy* automático na Railway.

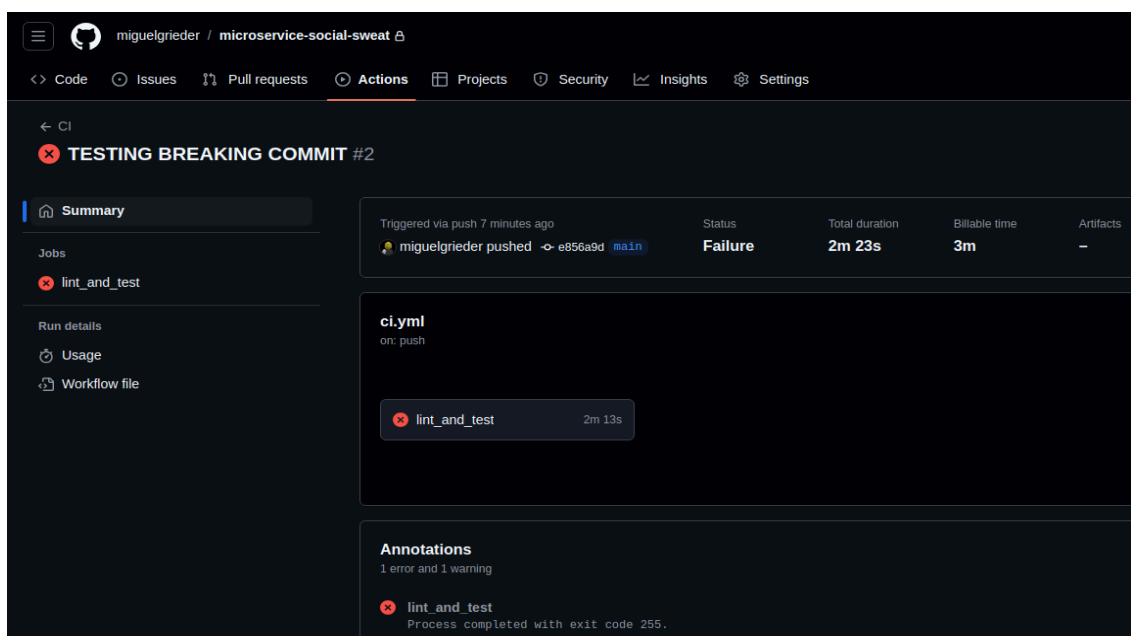
Figura 43 - Conjunto de etapas CI no Gitlab e Railway executadas com sucesso.



Fonte: elaborada pelo autor.

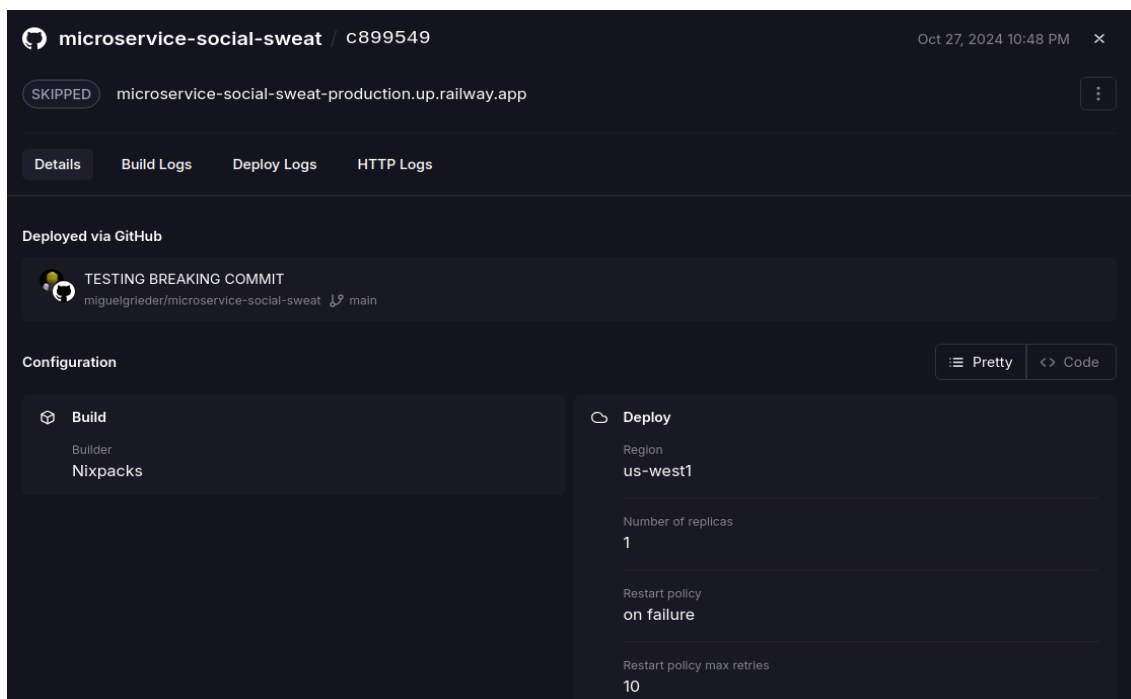
Ao publicar um commit que falharia em algum dos testes do tox, a falha torna-se evidente no GitHub Actions, mostrado na Figura 44, e é possível ver na Railway, como mostra a Figura 45, que essa atualização foi "*Skipped*", mantendo a versão anterior que havia passado nos testes.

Figura 44 - Etapa CI no Gitlab com problema.



Fonte: elaborada pelo autor.

Figura 45 - Visualização de um envio do microserviço que não foi a produção no Railway.



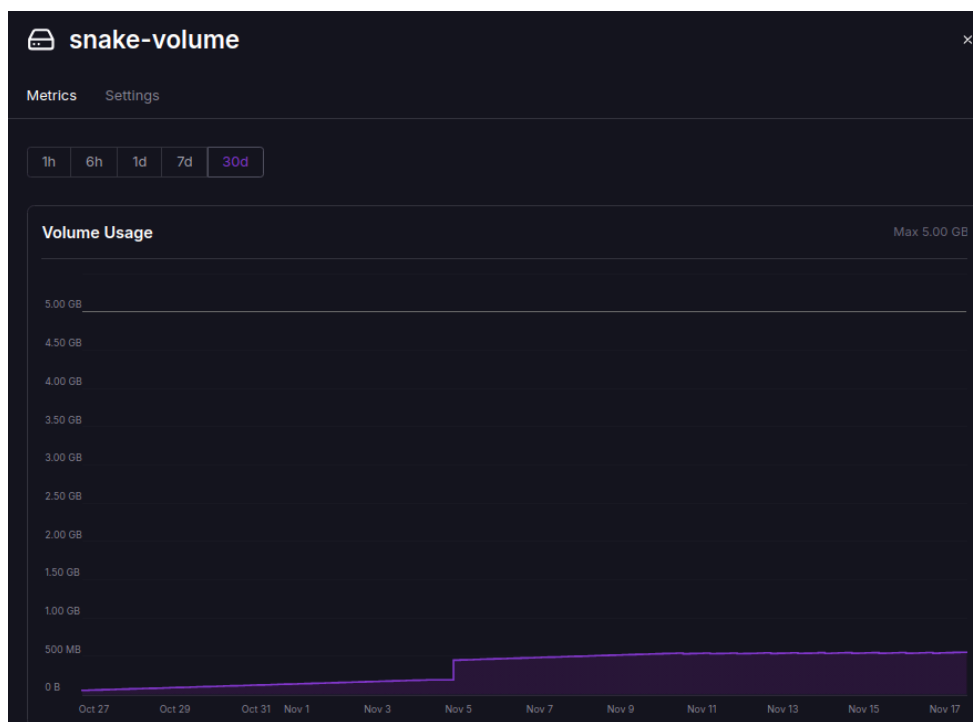
Fonte: elaborada pelo autor.

5.4 UTILIZAÇÃO DE RECURSOS DA NUVEM

A utilização dos serviços em nuvem pode ser avaliada para os componentes que operam no Railway, especificamente o volume de dados do MongoDB e o microserviço. O Expo Application Services (EAS), responsável por hospedar o *frontend*, não oferece uma ferramenta para monitorar o uso do serviço. Portanto, a análise será focada exclusivamente nos serviços fornecidos pelo Railway.

Em relação ao volume de dados do MongoDB, conforme ilustrado na Figura 46, observa-se um aumento gradual na ocupação, ultrapassando os 500 *megabytes*. Esse valor excede o limite da conta gratuita do Railway, o que tornou necessário a aquisição do plano pago da plataforma “Hobby”, que permite volumes de armazenamento de até 5 *gigabytes*.

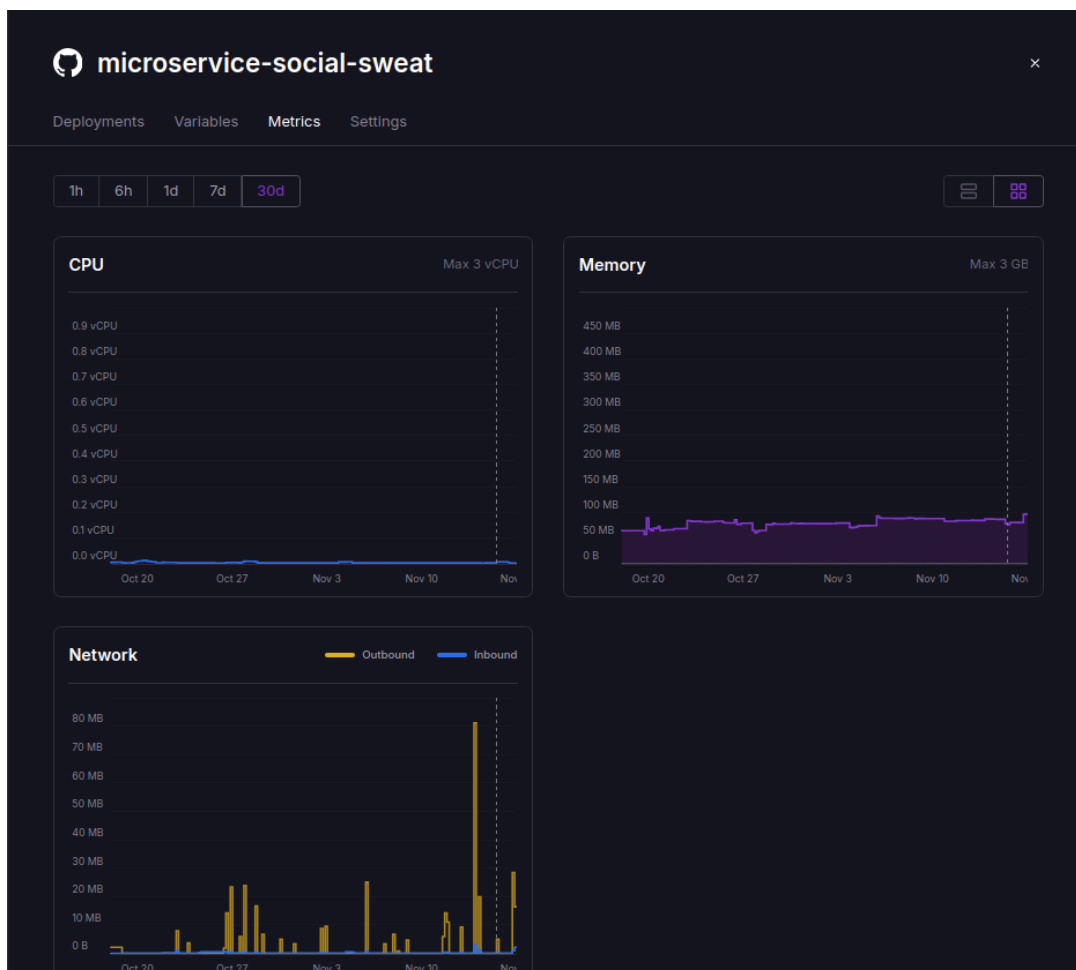
Figura 46 - Métricas de utilização de recursos do volume MongoDB no Railway.



Fonte: elaborada pelo autor.

Quanto ao microserviço, a maioria dos indicadores de desempenho se manteve estável, como visto na Figura 47. A única variação significativa foi observada no tráfego de rede (*Network*), que representa o fluxo de dados de entrada e saída no aplicativo. Embora tenham ocorrido momentos de maior uso, o consumo de recursos deste serviço não ultrapassou os limites estabelecidos pela conta gratuita do Railway.

Figura 47 - Métricas de utilização de recursos do microserviço no Railway.



Fonte: elaborada pelo autor.

5.5 MARKETING

Para promover e distribuir o aplicativo nesta fase de projeto de conclusão de curso, foi criada uma página no Instagram dedicada, mostrada na Figura 48, onde foram realizadas postagens destacando as funcionalidades do aplicativo e fornecendo orientações sobre seu uso. Essas postagens abordaram desde o processo de acesso ao aplicativo através do Expo Go até o detalhamento de suas principais funcionalidades, facilitando a compreensão e engajamento dos usuários. Também tem o *link* de acesso para o aplicativo no Expo.

Figura 48 - Perfil do Social Sweat no Instagram.



Fonte: elaborada pelo autor.

6 AVALIAÇÃO

A avaliação do aplicativo foi estruturada em três cenários base de uso distintos, com o objetivo de testar sua funcionalidade, usabilidade e eficácia nos principais contextos de utilização. Todas as avaliações utilizaram o System Usability Scale (SUS) para obter um resultado mensurável da usabilidade do aplicativo, além de coletar comentários adicionais dos usuários.

6.1 DEFINIÇÃO DA AVALIAÇÃO

Foram definidos três tópicos de avaliação:

- **Busca por *Personal Trainer*:** simulou-se a experiência de um usuário procurando um treinador de esqui aquático através da plataforma. Já havia um *personal trainer* de esqui aquático previamente cadastrado no aplicativo. O participante deve enviar uma mensagem pelo Instagram ao treinador.
- **Organização de Partida de Futebol:** testou-se a funcionalidade em que um usuário cria uma atividade do tipo evento, que é uma partida de futebol, buscando outros jogadores para completar os times.
- **Usuário casual buscando uma atividade durante uma viagem:** simulou-se um usuário buscando por uma atividade gratuita do tipo evento de futebol em uma cidade que está visitando. Esta atividade já estava criada previamente pelo autor, e o usuário deveria encontrá-la.

Após completarem a atividade, foi aplicado o teste SUS, que contém 10 perguntas:

1. Eu acho que gostaria de usar essa aplicação com frequência.
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.

4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muitas inconsistências.
7. Eu imagino que as pessoas aprenderão a usar esse sistema rapidamente.
8. Eu achei o sistema complicado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de usar o sistema.

Os participantes atribuíram notas de 1 a 5 para cada pergunta, seguindo a escala:

1 - Discordo Totalmente

2 - Discordo

3 - Neutro

4 - Concordo

5 - Concordo Totalmente

Além do teste SUS, foi disponibilizado um campo para que os usuários pudessem registrar comentários adicionais sobre a experiência.

Os resultados de cada avaliação estão organizados por participante, incluindo as métricas do teste SUS em tabelas. A tabela SUS inclui os itens de 1 a 10 (as questões do SUS), a escala escolhida pelo usuário e a contribuição de cada item para o resultado final do teste. As métricas se dizem a respeito de todo o uso do participante com o aplicativo, e não se restringe apenas aos tópicos da avaliação feita. Ao término deste capítulo é apresentado o resultado completo da avaliação.

Os participantes foram escolhidos por conveniência, sendo pessoas conhecidas do autor.

- **Participantes Cadastrados:** pessoas que antes de realizar o teste proposto já tinham conhecimento do projeto e experiência com o aplicativo, tendo familiaridade com seu propósito. Já haviam criado uma conta, criado uma atividade, participado de uma atividade e visto a listagem de treinadores na aba Treinadores.
- **Participantes Novos:** usuários sem nenhum conhecimento prévio do aplicativo, representando novos usuários em potencial.

Os participantes foram instruídos a realizar a tarefa proposta de maneira idêntica à descrita no início deste capítulo, sem orientações adicionais sobre como executar as atividades propostas. Cada participante só realizou uma tarefa, tendo assim 6 participantes no total.

A formação acadêmica ou profissional dos participantes não foi considerada um fator relevante na seleção, uma vez que o aplicativo é destinado ao uso geral pela população. Dessa forma, a avaliação buscou abranger diferentes perfis de usuários para obter um panorama abrangente sobre a usabilidade e eficácia do aplicativo nos cenários propostos.

6.2 BUSCA POR *PERSONAL TRAINER***Tabela 1** - Participante 1: Busca por *Personal Trainer*, Participante Cadastrado.

Questão	Escala					Contribuição
	1	2	3	4	5	
1			x			$3 - 1 = 2$
2	x					$5 - 1 = 4$
3					x	$5 - 1 = 4$
4		x				$5 - 2 = 3$
5				x		$4 - 1 = 3$
6		x				$5 - 2 = 3$
7					x	$5 - 1 = 4$
8	x					$5 - 1 = 4$
9					x	$5 - 1 = 4$
10		x				$5 - 2 = 3$
SOMA x 2,5						85

Fonte: elaborada pelo autor.

Tabela 2 - Participante 2: Busca por *Personal Trainer*, Participante Novo.

Questão	Escala					Contribuição
	1	2	3	4	5	
1			x			$3 - 1 = 2$
2		x				$5 - 2 = 3$
3				x		$4 - 1 = 3$
4			x			$5 - 3 = 2$
5					x	$5 - 1 = 4$
6	x					$5 - 1 = 4$
7					x	$5 - 1 = 4$
8	x					$5 - 1 = 4$
9				x		$4 - 1 = 3$
10			x			$5 - 3 = 2$
SOMA x 2,5						77.5

Fonte: elaborada pelo autor.

6.3 ORGANIZAÇÃO DE PARTIDA DE FUTEBOL

Tabela 3 - Participante 3: Organização de Partida de Futebol, Participante Cadastrado.

Questão	Escala					Contribuição
	1	2	3	4	5	
1					x	$5 - 1 = 4$
2		x				$5 - 2 = 3$
3					x	$5 - 1 = 4$
4	x					$5 - 1 = 4$
5					x	$5 - 1 = 4$
6		x				$5 - 2 = 3$
7				x		$4 - 1 = 3$
8	x					$5 - 1 = 4$
9					x	$5 - 1 = 4$
10	x					$5 - 1 = 4$
SOMA x 2,5						92.5

Fonte: elaborada pelo autor.

Tabela 4 - Participante 4: Organização de Partida de Futebol, Participante Novo.

Questão	Escala					Contribuição
	1	2	3	4	5	
1		x				$2 - 1 = 1$
2	x					$5 - 1 = 4$
3					x	$5 - 1 = 4$
4	x					$5 - 1 = 4$
5					x	$5 - 1 = 4$
6		x				$5 - 2 = 3$
7				x		$4 - 1 = 3$
8	x					$5 - 1 = 4$
9				x		$4 - 1 = 3$
10	x					$5 - 1 = 4$
SOMA x 2,5						86,5

Fonte: elaborada pelo autor.

6.4 USUÁRIO CASUAL BUSCANDO UMA ATIVIDADE DURANTE UMA VIAGEM

Tabela 5 - Participante 5: Usuário casual buscando uma atividade durante uma viagem, Participante Cadastrado.

Questão	Escala					Contribuição
	1	2	3	4	5	
1			x			$3 - 1 = 2$
2	x					$5 - 1 = 4$
3					x	$5 - 1 = 4$
4	x					$5 - 1 = 4$
5				x		$4 - 1 = 3$
6		x				$5 - 2 = 3$
7					x	$5 - 1 = 4$
8	x					$5 - 1 = 4$
9				x		$4 - 1 = 3$
10	x					$5 - 1 = 4$
SOMA x 2,5						87.5

Fonte: elaborada pelo autor.

Tabela 6 - Participante 6: Usuário casual buscando uma atividade durante uma viagem, Participante Novo.

Questão	Escala					Contribuição
	1	2	3	4	5	
1				x		$4 - 1 = 3$
2		x				$5 - 2 = 3$
3				x		$4 - 1 = 3$
4		x				$5 - 2 = 3$
5				x		$4 - 1 = 3$
6			x			$5 - 3 = 2$
7					x	$5 - 1 = 4$
8		x				$5 - 2 = 3$
9				x		$4 - 1 = 3$
10	x					$5 - 1 = 4$
SOMA x 2,5						77.5

Fonte: elaborada pelo autor.

6.5 RESULTADO DA AVALIAÇÃO

A Tabela 7 apresenta os resultados dos testes SUS resumidos por participante, incluindo as somas, pontuações individuais do SUS e a média final dessas métricas. Os participantes com números ímpares (1, 3, 5) eram usuários cadastrados que já possuíam familiaridade com o aplicativo, enquanto os participantes pares (2, 4, 6) estavam utilizando o aplicativo pela primeira vez. A avaliação se diz sobre todo o uso que os

usuários tiveram com o sistema, então para os usuários cadastrados foi contado mais do que a avaliação pedida.

Tabela 7 - Resultado final do SUS.

Participante	Soma	Pontuação SUS
1	34	85
2	31	77,5
3	37	92,5
4	34	86,5
5	35	87,5
6	31	77,5
Média final	43,7	84,4

Fonte: elaborada pelo autor.

Observa-se que os participantes cadastrados obtiveram uma média de pontuação SUS de 88,3, enquanto os novos usuários alcançaram uma média de 80,5, resultando em uma diferença de quase 8 pontos. Essa diferença foi consistente em todos os casos, provavelmente devido à familiaridade prévia com o sistema.

O resultado médio do SUS foi de 84,4, o que pode ser interpretado como uma indicação de que o sistema é percebido como altamente utilizável, conforme as normas e classificações descritas na literatura. Pontuações acima de 80 no SUS frequentemente são classificadas como uma experiência acima da média (LEWIS, 2018), correspondendo a um "A+" na escala curvada de classificação (CGS). Dessa forma, a média de 84,4 está próxima do topo dessa categoria, refletindo um nível de usabilidade considerado excelente pela maioria dos padrões.

Os comentários foram positivos, com alguns pontos citados para melhorar. Os principais comentários para melhorar foram que alguns componentes não estavam sendo traduzidos para o português, como o escolhedor de datas e algumas mensagens de erro na criação de conta. Também, tiveram dificuldades em utilizar este escolhedor de datas, pois digitar o ano desejado no componente não funcionava como o esperado. Também, a tela de início estava servindo um propósito muito genérico, com e atalhos de atividades e o seletor de idioma juntos.

7 CONCLUSÃO

O desenvolvimento deste projeto resultou em um aplicativo funcional e eficiente, atendendo aos requisitos estabelecidos. As avaliações realizadas confirmaram tanto a usabilidade quanto a eficácia da plataforma, evidenciando seu potencial para alcançar os objetivos propostos.

A arquitetura do aplicativo foi projetada para suportar uma aplicação profissional, com facilidade de expansão para novas áreas e países. Isso inclui suporte completo a múltiplas línguas e moedas. Essa abordagem garante a escalabilidade do aplicativo, permitindo sua adoção em diversas localidades com adaptações mínimas.

Foi criada uma empresa de responsabilidade limitada (LLC) pelo autor e seu sócio americano, Collins, denominada X Platforms, como visto na Figura 49. Este empreendimento está em processo de captação de *venture capital* nos Estados Unidos, com o objetivo de viabilizar a transformação do projeto em uma *startup*.

Figura 49 - Documento de incorporação da *startup* no estado de Delaware.

STATE of DELAWARE
STATEMENT and RESIGNATION of the INCORPORATOR
A STOCK/PROFIT CORPORATION

The undersigned, the Incorporator of **X PLATFORMS INCORPORATED**, who signed and filed its Articles of Incorporation (or similar organizing document) with the Delaware Secretary of State (or other appropriate state office), appoints the following individuals to serve as directors of the corporation, who shall serve as directors until the first meeting of shareholders for the election of directors and until their successors are elected and agree to serve on the board:

Name and address of each initial director:

COLLINS [REDACTED]
 [REDACTED]
 [REDACTED]

MIGUEL GRIEDER
 [REDACTED]
 [REDACTED]

Additionally, the undersigned does hereby tender his/her resignation as Incorporator for the Corporation, and from any and all involvement with, control of, or authority over the Corporation, real or perceived, effective immediately.

Dated: June 17th, 2024


 Lovette Dobson, Incorporator

Fonte: elaborada pelo autor.

7.1 TRABALHOS FUTUROS

Para garantir a continuidade do desenvolvimento e aprimoramento do aplicativo, foram identificadas diversas áreas para serem aprimoradas e desenvolvidas como trabalho futuro.

7.1.1 Novas Funcionalidades

- **Sessões recorrentes:** implementar uma funcionalidade que permita a criação de sessões com atividades que se repetem automaticamente em datas e horários futuros. Isso facilitará a organização e eliminará a necessidade de criação manual de sessões repetitivas.
- **Gamificação:** introduzir elementos de gamificação, como conquistas e recompensas, além apenas das métricas do usuário no perfil, para aumentar o engajamento e a motivação dos usuários em suas atividades físicas.
- **Avaliações:** desenvolver a opção para que usuários possam registrar avaliações dos serviços oferecidos por treinadores e empresas, além das atividades de sessão e local realizadas. As avaliações incluirão uma classificação de uma a cinco estrelas, uma breve descrição textual e a possibilidade de adicionar imagens.
- **Cadastro e Login Diversificados:** implementar métodos de cadastro via OAuth, aumentando a praticidade e flexibilidade e segurança no acesso ao aplicativo.
- **Segmentação de atividades por Área:** utilizar o país da atividade e área (como códigos de área (*Area Code*) nos Estados Unidos e DDD (Discagem Direta à Distância) no Brasil) para segmentar atividades por região de forma eficiente. Isso assegura a escalabilidade do aplicativo, permitindo sua adoção em diversas localidades com adaptações mínimas e minimizando a sobrecarga nos sistemas. Assim, o aplicativo iria carregar apenas as atividades de uma área por vez.

- **Moderação de atividades:** implementar um mecanismo para a moderação das atividades criadas, para prevenir o uso indevido desta funcionalidade, com denúncia e análise preliminar das atividades criadas.
- **Garantia de treinadores e empresas:** requisitar documentos, certificados, licenças ou registros que assegurem a profissionalidade dos treinadores e a legitimidade das empresas cadastrados no aplicativo, verificando a legitimidade destes documentos.

7.1.2 Estudos de Impacto

É prevista a realização de estudos para avaliar o impacto do aplicativo na saúde e bem-estar dos usuários. Esses estudos fornecerão dados que orientarão melhorias futuras e verificarão se o aplicativo está atingindo seu propósito.

REFERÊNCIAS BIBLIOGRÁFICAS

YANG, L., CAO, C., & KANTOR, E. D. (2018). Trends in sedentary behavior among the US population, 2001-2016. **Journal of the American Medical Association**, 320(19), 2020-2028. doi:10.1001/jama.2018.14854

HUNT, M. G., MARX, R., LIPSON, C., & YOUNG, J. (2017). No more FOMO: Limiting social media decreases loneliness and depression. **Journal of Social and Clinical Psychology**, 36(10), 751-768. doi:10.1521/jscp.2017.36.10.751

BUENO-ANTEQUERA, J.; MUNGUÍA-IZQUIERDO, D. (2023). The association between social media use and loneliness: A meta-analysis. In *Integrated Science of Global Epidemics* (pp. 429-442). Springer, Cham. doi:10.1007/978-3-031-17778-1_19

TEODOR, D. SEDENTARISM AND BENEFITS OF REGULAR PHYSICAL ACTIVITY IN HEALTHY ADULTS. (2018) **Ovidius University Annals - Science, Movement and Health**, vol. XVIII, n. 1

LEITE, Morgana. Desenvolvimento de um aplicativo Android voltado para profissionais de Educação Física. 2022. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – **Universidade do Estado de Santa Catarina**, Florianópolis, 2022.

CHEN, Yu; PU, Pearl. HealthyTogether: exploring social incentives for mobile fitness applications. **University of California**, p. 10, 2014.

D'ADDARIO, Marco; BARETTA, Dario; ZANATTA, Francesco; GRECO, Andrea; STECA, Patrizia. Engagement Features in Physical Activity Smartphone Apps: Focus Group Study With Sedentary People. **JMIR Mhealth Uhealth**, v. 8, n. 11, p. e20460, 2020.

HRISTOV, Victor. "These were the best phones... before the iPhone." PhoneArena, 29 jun. 2017, 9:25 AM. Disponível em: <https://www.phonearena.com/news/These-were-the-best-phones...-before-the-iPhone_id95512>. Acesso em: 8 jun. 2024.

FAAS, R. J. Understanding iPhone Applications and the App Store. In: iPhone for Work: Increasing Productivity for Busy Professionals. **Berkeley**, CA: Apress, 2010. p. 273-285. Disponível em: <https://doi.org/10.1007/978-1-4302-2446-4_15>. Acesso em: 8 jun. 2024.

Google. Visão geral do AOSP. Documentação do Android. Disponível em: <<https://source.android.com/docs/setup/about?hl=pt-br>>. Acesso em: 7 jun. 2024.

FOJTÍK, Rostislav; **PAWLAS**, Jan. CROSS-PLATFORM AND NATIVE MOBILE APP DEVELOPMENT. In: Proceedings of the 4th International Conference on Decision Making for Small and Medium-Sized Enterprises (DEMSME). **Universidade de Twente**, 2023. p. 70. Disponível em: <https://ris.utwente.nl/ws/portalfiles/portal/325391243/DEMSME_2023_Proceedings.pdf#page=70>. Acesso em: 8 jun. 2024.

PAUL, A.; **NALWAYA**, A. React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications. 1st ed. **Springer**, 2019. Disponível em: <<https://link.springer.com/book/10.1007/978-1-4842-4454-8>>. Acesso em: 8 jun. 2024.

BOSE, S. et al. "A comparative study: java vs kotlin programming in android application development." International Journal of Advanced Research in Computer Science, v. 9, n. 3, p. 41-45, 2018. Disponível em:

<<https://pdfs.semanticscholar.org/c0ee/43434064520cdde7222318bf6c4d2db69177.pdf>>.

Acesso em: 8 jun. 2024.

QUVVATOV, B. WEB FRONT-END AND BACK-END TECHNOLOGIES IN PROGRAMMING. Theoretical aspects in the formation of pedagogical sciences, v. 3, n. 1, p. 208-215, 15 jan. 2024. Disponível em: <<http://econferences.ru/index.php/tafps/article/view/11716>>. Acesso em: 8 jun. 2024.

DUBEY, P.; **TIWARI**, A. K.; **RAJA**, R. Amazon Web Services: the Definitive Guide for Beginners and Advanced Users. **Bentham Science Publishers**, 2023. Disponível em: <<https://books.google.com.br/books?hl=en&lr=&id=rQbgEAAQBAJ&oi>>. Acesso em: 7 jun. 2024.

YEPURI, Vamsi Krishna; **POLAMARASETTY**, Venkata Kalyan; **DONTHI**, Shivani; **GONDI**, Ajay Kumar Reddy. **Containerization of a polyglot microservice application using Docker and Kubernetes**. arXiv preprint arXiv:2305.00600, 2023. Disponível em: <<https://arxiv.org/abs/2305.00600>>. Acesso em: 7 jun. 2024.

DUBEY, Atal et al. "MONGODB: A COMPREHENSIVE REVIEW." **IRJMETS**, [S.l.], v. 5, n. 11, p. 12-15, nov. 2023. Disponível em: <https://www.irjmets.com/uploadedfiles/paper/issue_11_november_2023/46052/final/fin_irjmets1699333129.pdf>. Acesso em: 7 jun. 2024.

DHRUV, Akshit J. et al. "Python: the most advanced programming language for computer science applications." **Science and Technology Publications, Lda**, 2021, p. 292-299. Disponível em: <<https://www.scitepress.org/Papers/2020/103079>>. Acesso em: 7 jun. 2024.

MA, Xiaoming. "Development and Automation of a Web Applications Using FastAPI, Jenkins, and Robot Framework." (2024). Disponível em: <<https://www.theseus.fi/handle/10024/859549>>. Acesso em: 7 jun. 2024.

DESIDERIA, Gladina; **BANDUNG**, Yoanes. "User efficiency model in usability engineering for user interface design refinement of mobile application." **J. ICT Res. Appl.**, 2020, v. 14, n. 1, p. 16-33. Disponível em: <<https://pdfs.semanticscholar.org/4aa6/0c5e617620e78b0b91b14d944dcd600f0a26.pdf>>. Acesso em: 12 jun. 2024.

HYZY, Maciej; **BOND**, Raymond; **MULVENNA**, Maurice; **BAI**, Li; **DIX**, Alan; **LEIGH**, Susan; **HUNT**, Steven. "System usability scale benchmarking for digital health apps: meta-analysis." **JMIR mHealth and uHealth**, 2022, v. 10, n. 8, p. e37290. Disponível em: <<https://mhealth.jmir.org/2022/8/e37290>>. Acesso em: 12 jun. 2024.

CHEN, Yu; **PU**, Pearl. HealthyTogether: exploring social incentives for mobile fitness applications. **University of California**, 2014. Disponível em: <<https://dl.acm.org/doi/10.1145/2592235.2592240>>. Acesso em: 17 jun. 2024.

LEWIS, James. "The System Usability Scale: Past, Present, and Future." *International Journal of Human-Computer Interaction*, 2018, v. 34, n. 7, p. 577-590. Disponível em: https://www.researchgate.net/profile/James-Lewis-8/publication/324116412_The_System_Usability_Scale_Past_Present_and_Future/links/652225343ab6cb4ec6c71aec/The-System-Usability-Scale-Past-Present-and-Future.pdf. Acesso em: 16 nov. 2024.

COHN, Mike. *User Stories Applied for Agile Software Development*. Addison-Wesley, 2009.

PRESSMAN, Roger S. Software Engineering: A Practitioner's Approach. 7. ed. Boston: McGraw-Hill Higher Education, 2010.

APÊNDICE A – Aplicativo móvel para encontrar grupos, parceiros e instrutores de atividades físicas.

Miguel D. Grieder

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC –
Brazil

miguel.grieder@grad.ufsc.br

Abstract. *With the rise of technology, people have become more sedentary and isolated due to social media (YANG, 2019). The Social Sweat app for iOS and Android aims to reverse this trend by connecting users for group physical activities. It provides a platform where users can create, view, and participate in any physical activity event. The app also supports trainers in offering their services to users and enables companies to register their locations.*

Resumo. *Com o aumento da tecnologia, as pessoas tornaram-se mais sedentárias e isoladas pelas redes sociais (YANG, 2019). O aplicativo Social Sweat para iOS e Android visa reverter isso, conectando usuários para a realização de atividades físicas em grupo. Ele oferece uma plataforma onde os usuários podem criar, visualizar e participar de qualquer evento de atividade física. O aplicativo também oferece suporte para os treinadores disponibilizarem seus serviços aos usuários e empresas terem seus locais registrados.*

1. Introdução

O sedentarismo, agravado pelo uso excessivo de tecnologias, representa um desafio significativo à saúde pública. Segundo Yang (2019), dispositivos digitais, como smartphones e redes sociais, consomem grande parte do tempo que poderia ser destinado à prática de atividades físicas.

Este trabalho apresenta o Social Sweat, um aplicativo para dispositivos móveis Android e iOS projetado para facilitar a busca por locais, atividades, pessoas, grupos e instrutores próximos para a prática de atividades físicas. A navegação será feita pelo mapa ou listagem onde os usuários poderão adicionar eventos temporários, como torneios de futebol, com informações de data, endereço e outros detalhes. O objetivo é promover a interação entre os interessados e facilitar a organização de atividades em grupo ou com profissionais, visando reduzir o sedentarismo, melhorar a saúde física e mental e incentivar a socialização.

2. Proposta

O Social Sweat tem o objetivo de desenvolver uma plataforma inclusiva e intuitiva, destinada a conectar indivíduos interessados em atividades físicas. As funcionalidades principais da plataforma incluem:

Descoberta de Atividades: Através da aba "Atividades", é possível descobrir novas opções de prática física. As atividades são apresentadas tanto em formato de mapa quanto em listagem, permitindo uma visualização abrangente. Além disso, a plataforma oferece filtros rápidos por tipo de esporte e filtros avançados para refinar a busca conforme as preferências específicas.

Participação de Atividades: É permitindo que os usuários se inscrevam de forma simples e rápida. Caso uma atividade esteja lotada, a plataforma impedirá novas inscrições, garantindo a capacidade máxima estabelecida. Ao se inscrever, uma descrição privada da atividade é disponibilizada, fornecendo detalhes adicionais para os participantes.

Criação de Atividades: A plataforma permite a criação de diferentes tipos de atividades, adaptando-se às diversas necessidades dos usuários:

- Locais: Espaços permanentes destinados à prática contínua, como academias, praças de skate.
- Sessões: Aulas recorrentes com horários fixos, como uma aula de arte marcial, aula de Yoga.
- Eventos: Atividades pontuais, como torneios, jogos de futebol esporádicos.

Cada tipo de usuário possui permissões específicas para criar atividades conforme seu perfil. Os organizadores têm a capacidade de editar, desativar e reativar suas atividades conforme necessário.

Cadastro e Login: O sistema de cadastro e login é gerenciado pela ferramenta Clerk, garantindo segurança e eficiência na gestão das contas dos usuários. O processo de cadastro e autenticação é realizado por meio de email, senha e um código de verificação enviado por email. Além disso, o sistema fornece avisos de erro sugestivos para facilitar a resolução de eventuais problemas durante o processo.

Criação de Perfil: Os usuários podem personalizar seus perfis adicionando uma imagem, biografia, esportes principais e links para outras redes sociais. A plataforma exibe métricas de participação, permitindo o acompanhamento do engajamento individual. Além disso, todas as atividades criadas pelo usuário são exibidas diretamente em seu perfil, facilitando o gerenciamento pessoal.

Treinadores Profissionais: Há uma aba específica dedicada a treinadores profissionais, onde esses profissionais são exibidos para todos os usuários da plataforma. Ao selecionar um treinador na lista, abre-se o perfil completo do profissional, proporcionando informações detalhadas sobre suas qualificações e serviços oferecidos.

Trocar de Idioma: A plataforma permite ao usuário escolher entre os idiomas inglês e português.

Atalhos de Atividade: São disponibilizados atalhos rápidos para acessar diretamente as seções "Atividades Criadas" e "Minhas Próximas Atividades". Essa funcionalidade facilita a navegação e o gerenciamento das participações e criações dos usuários dentro do aplicativo, otimizando a experiência de uso.

3. Desenvolvimento

Este capítulo descreve as tecnologias utilizadas, o fluxo de trabalho adotado e os principais componentes que integram o sistema, detalhando as decisões tomadas para garantir funcionalidade, segurança e escalabilidade.

3.1 Tecnologias Utilizadas

O desenvolvimento do aplicativo foi organizado em três principais componentes, cada um com responsabilidades e tecnologias específicas:

Frontend: Para o aplicativo em si, utilizou-se React Native com o Expo, permitindo o desenvolvimento compartilhado entre as plataformas Android e iOS, reduzindo tempo e custos. O Expo Publish Service foi escolhido para a publicação do aplicativo.

Backend: O *backend* foi desenvolvido utilizando um microsserviço com o FastAPI, encapsulado em um contêiner Docker, facilitando a escalabilidade, manutenção e consistência na distribuição entre diferentes ambientes.

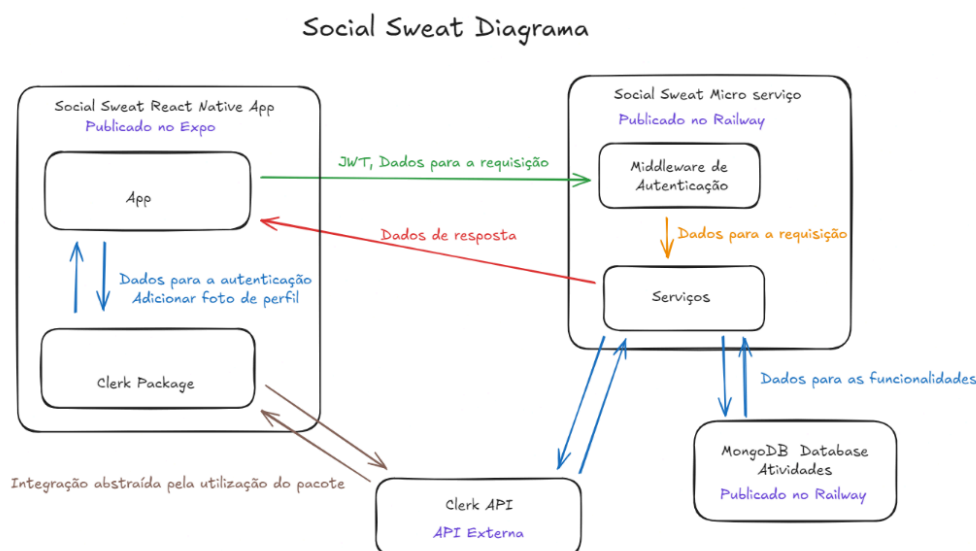
Banco de Dados: O banco MongoDB foi utilizado para armazenar as atividades criadas, fornecendo flexibilidade na modelagem de dados não-relacionais e escalabilidade para crescimento futuro.

Gerenciamento de Identidades: O serviço Clerk foi escolhido para gerenciar autenticação e privilégios de usuários, garantindo segurança e simplicidade na integração de gerenciamento de contas.

Infraestrutura na Nuvem: A hospedagem do microsserviço e do banco de dados (MongoDB de atividades) foi realizada na plataforma Railway, enquanto o Expo Publish Service foi utilizado para a distribuição do aplicativo.

3.2 Diagrama da Arquitetura e Fluxo de Dados

A Figura 1 apresenta o diagrama da arquitetura do aplicativo, destacando as conexões entre o *frontend*, e *backend*, o banco de dados e a API do Clerk. Esse diagrama detalha o fluxo de dados, especificando o tipo de informação transmitida e os pontos de integração entre os componentes.



* APIs externas que não tratam de dados persistentes para aplicação não foram incluídas, como consultas em mapas

Figura 1. Uma figura típica

Conforme ilustrado, o aplicativo envia o token de autenticação e os dados da requisição ao microsserviço para as operações. No microsserviço, esse token é validado por um middleware de autenticação antes de o serviço em si ser acionado, conectando-se com os sistemas necessários, como o banco de dados MongoDB, que armazena as atividades, e à API do Clerk. O aplicativo se

comunica diretamente com o Clerk apenas para gerenciar informações de autenticação e atualizar a foto de perfil do usuário.

3.3 Envio de atualizações a produção

O fluxo de Fluxo de Integração e Entrega Contínuas(CI/CD) implementado no Social Sweat automatiza a integração e entrega de novas versões do microserviço, garantindo confiabilidade e eficiência. No backend, cada nova atualização enviada à branch de produção no repositório GitHub aciona workflows configurados no GitHub Actions. Esses fluxos executam testes automatizados utilizando o tox, verificando a integridade do código em um ambiente isolado. Apenas após a aprovação nos testes, as alterações são implantadas automaticamente no servidor da Railway, onde o backend está hospedado.

Esse processo garante que atualizações sejam introduzidas apenas quando atenderem aos critérios de qualidade e testes estabelecidos. Caso a atualização apresente falhas nos testes, ele é descartado pelo sistema, preservando a versão estável em produção. Essa abordagem otimiza o ciclo de desenvolvimento, proporcionando uma entrega contínua de melhorias com alta confiabilidade. Na Figura 2 é exibido um caso de sucesso nos testes no GitHub seguido da atualização em produção do microserviço no Railway.

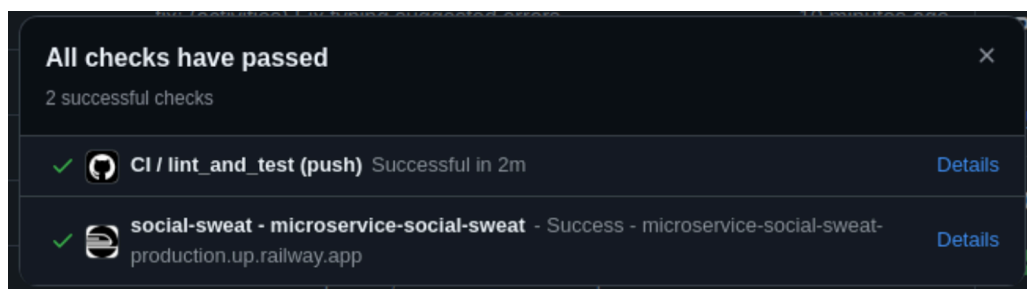


Figura 2. Conjunto de etapas CI no Gitlab e Railway executadas com sucesso.

Para o aplicativo, os envios são feitos manualmente executando o comando ‘`cas update`’ no terminal do projeto local, onde é construído e publicado a nova versão no servidor Expo automaticamente.

O fluxo para realizar o envio de atualizações a produção para o microserviço e aplicativo podem ser vistos na Figura 3.

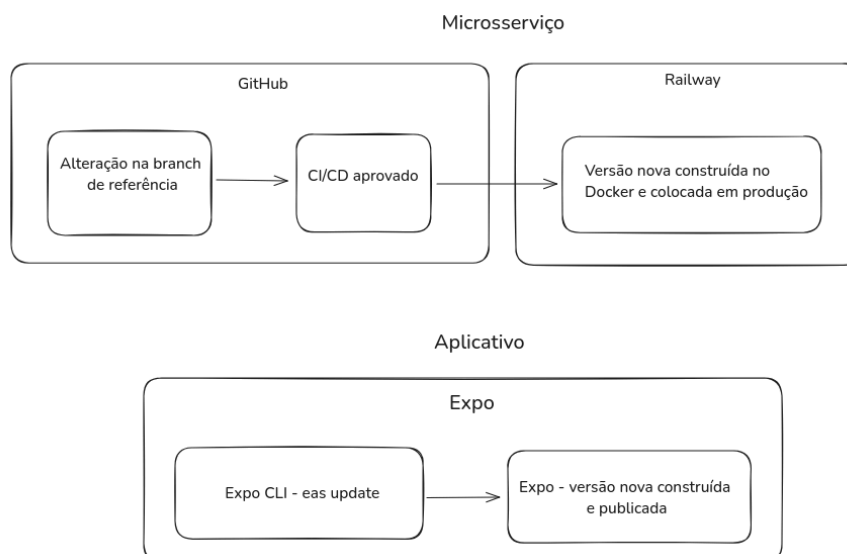


Figura 3. Diagrama de atualização dos serviços em nuvem.

4. Resultados

Foram realizados testes no aplicativo Social Sweat, estruturados para avaliar sua funcionalidade, usabilidade e eficácia em diferentes cenários de uso. Três situações distintas foram simuladas: a busca por um personal trainer, a organização de uma partida de futebol e a busca por uma atividade durante uma viagem. Com base nesses cenários, foi aplicado o System Usability Scale (SUS) para mensurar a usabilidade do sistema, complementado por comentários fornecidos pelos participantes.

Os resultados indicaram uma pontuação média de 84,4 no teste SUS, considerada excelente e acima da média, conforme os padrões de avaliação da literatura. Para a análise, os participantes foram divididos em dois grupos: aqueles que já possuíam familiaridade com o aplicativo, tendo-o utilizado anteriormente, e os novos usuários, que o experimentaram pela primeira vez durante o teste. Observou-se que os participantes familiarizados alcançaram uma média de 88,3 pontos, enquanto os novatos obtiveram 80,5. Essa diferença sugere a existência de uma curva de aprendizado, ainda que relativamente curta, no uso do aplicativo. Comentários adicionais destacaram a necessidade de melhorias pontuais, como a tradução completa de alguns componentes para o português e ajustes no seletor de datas. Apesar dessas observações, o aplicativo demonstrou alta usabilidade e funcionalidade, consolidando seu potencial para atingir os objetivos propostos e atender a um público amplo de forma escalável e eficiente.

5. Conclusão

Como resultado final, o aplicativo Social Sweat mostrou-se como uma plataforma que facilita a descoberta, organização e participação em atividades físicas, integrando usuários, treinadores profissionais e espaços dedicados ao esporte de forma simples e eficaz. O desenvolvimento e a infraestrutura adotados garantiram estabilidade, segurança e escalabilidade, tornando possível a expansão para novos contextos, países e línguas. Os testes de usabilidade demonstraram uma experiência altamente positiva, refletida em pontuações elevadas no SUS, o que indica a facilidade de uso do sistema. Ao mesmo tempo, os

comentários coletados durante as avaliações forneceram subsídios valiosos para futuras melhorias, contribuindo para um aperfeiçoamento contínuo. Dessa forma, o Social Sweat mostra-se pronto para cumprir sua proposta original, incentivando a prática de atividades físicas e promovendo a saúde e a interação social entre os usuários.

Referências

Yang, L., Cao, C., and Kantor, E. D. (2018) “Trends in sedentary behavior among the US population, 2001-2016”, *Journal of the American Medical Association*, Vol. 320, No. 19, pp. 2020–2028, doi:10.1001/jama.2018.14854.

APÊNDICE B – Código fonte

O código-fonte deste projeto está disponível na plataforma GitHub, organizado em duas partes: o aplicativo (*frontend*) e o microsserviço (*backend*). O repositório do aplicativo pode ser acessado em: <https://github.com/miguelgrieder/social-sweat-app>, enquanto o repositório do microsserviço está disponível em: <https://github.com/miguelgrieder/microservice-social-sweat>.