UNIVERSIDADE TUIUTI DO PARANÁ

GIOVANNI LUIZ CORTIANO

SPORTAPP – APLICATIVO PARA APROXIMAÇÃO DE PRATICANTES DE ESPORTE

CURITIBA

2018

GIOVANNI LUIZ CORTIANO

SPORTAPP - APLICATIVO PARA APROXIMAÇÃO DE PRATICANTES DE ESPORTE

Trabalho de conclusão de curso apresentado ao curso de Tecnologia em análise e desenvolvimento de sistemas da Universidade Tuiuti do Paraná como pré-requisito para a obtenção do título de Tecnólogo em análise e desenvolvimento de sistemas.

Professor (a): Sérgio Marques.

CURITIBA

2018

TERMO DE APROVAÇÃO

GIOVANNI LUIZ CORTIANO

SPORTAPP – APLICATIVO PARA APROXIMAÇÃO DE PRATICANTES DE ESPORTE

Este trabalho de conclusão de curso foi julgado e aprovado para obtenção do título de Tecnólogo em analise e desenvolvimento de sistemas da Universidade Tuiuti do Paraná

Curitiba, 13 de dezembro de 2018

Tecnologia em análise e desenvolvimento de sistemas

Orientador: Prof. Sérgio Ramos

UTP – FACET

Prof. Patrícia R. de Bassi

UTP – FACET

Prof. Ângela Zatti

UTP – FACET

*Dedico este trabalho aos meus pais, Gerson Luiz Cortiano e Tânia Mara Cortiano que sempre me deram todo o apoio necessário para estudar. Dedico também a todos os amigos que de alguma forma contribuíram com ideias para o aplicativo, e a todos os professores que fizeram parte da minha formação acadêmica.*

RESUMO

O presente trabalho tem por intuito apresentar o projeto de construção de um aplicativo para dispositivos móveis que permita a aproximação de pessoas praticantes de esporte, utilizando todo o conhecimento adquirido durante o curso. Atualmente em consultas realizadas no Google Play, loja de aplicativos do sistema operacional Android, não foi encontrado aplicativos que tenham a finalidade de aproximar as pessoas de acordo com os esportes que elas gostam ou praticam. Uma grande parte dos esportes exige mais de uma pessoa para que possa ser praticado, exemplo: futebol, vôlei, tênis, basquete, entre outros. Esta pesquisa justifica-se porque propõe uma alternativa para aproximar pessoas de acordo com a sua localização e os esportes de seu gosto, incentivando a prática de esportes e ajudando as pessoas que querem praticar um esporte e não tem com quem fazer isso.

A pesquisa tem o objetivo de construir um aplicativo de celular que funcione no ambiente ANDROID e que consiga aproximar as pessoas de acordo com os esportes de seu interesse, permitindo que possam se conhecer, conversar, marcar jogos, etc. O sistema construído não pode gerar a necessidade de administração de um servidor de armazenamento de dados, deve ter disponibilidade em tempo integral para que o público alvo possa utiliza-lo a qualquer momento e não pode gerar custos em dinheiro, uma vez que o aplicativo é um experimento de inovação e no momento não tem fins lucrativos.

Para atingir o objetivo, o sistema foi construído utilizando a linguagem de programação Java e o ambiente de desenvolvimento integrado Android Studio, foi aplicado à metodologia de desenvolvimento de orientação a objetos e utilizada à linguagem UML para construção dos diagramas de casos de uso, diagrama de classes, diagrama de sequências e especificação dos casos de uso. Também foi utilizada a plataforma de desenvolvimento Firebase do Google para realizar o armazenamento de dados, autenticações, envio de e-mails, envio de notificações e mensagens em tempo real. Outras técnicas e tecnologias também foram aplicadas, como padrões de projeto GOF e utilização da localização do GPS do dispositivo para obter o posicionamento geográfico dos usuários.

Ao longo da pesquisa os objetivos foram alcançados, permitindo que o sistema fique disponível o tempo todo, sem gerar custos financeiros, proporcionando a possibilidade dos usuários encontrarem pessoas próximas de acordo com localização do GPS, com gostos dos mais variados esportes, permitindo conversas em tempo real para facilitar a aproximação das pessoas.

Palavras chave: Aproximação de pessoas por esporte. Aproximação de praticantes de esporte. Aplicativo de esporte. Aplicativo para incentivar a prática de esporte. Firebase.

**LISTA DE FIGURAS**

FIGURA 1 - PERCENTUAL CRESCENTE DE UTILIZAÇÃO DE DISPOSITIVOS MÓVEIS COM ACESSO A INTERNET NO BRASIL 11

FIGURA 2 - DIAGRAMA DE CASOS DE USO DO SISTEMA SPORTAPP 19

FIGURA 3 - TELA INICIAL DO SISTEMA 23

FIGURA 4 - TELA DE CADASTRO 24

FIGURA 5 - TELA DE CONFIGURAÇÕES, ATUALIZAÇÃO DE PERFIL E EXCLUSÃO DE CONTA. 25

FIGURA 6 - E-MAIL DE CONFIRMAÇÃO DE CONTA 26

FIGURA 7 - CONFIRMAÇÃO DE CONTA CONCLUÍDA 27

FIGURA 8 - TELA DE LOGIN 30

FIGURA 9 - TELA DE LOGIN AVISANDO SOBRE A CONFIRMAÇÃO DE CONTA 31

FIGURA 10 - TELA DE RECUPERAÇÃO DE SENHA 32

FIGURA 11 - E-MAIL DE RECUPERAÇÃO DE SENHA 33

FIGURA 12 - PÁGINA WEB PARA REDEFINIÇÃO DE SENHA 34

FIGURA 13 - APLICATIVO AUTENTICADO, EXIBINDO AS OPÇÕES DE CONFIGURAÇÕES E SAIR/LOGOFF 35

FIGURA 14 - APLICATIVO AUTENTICADO EXIBINDO UMA LISTA DE PESSOAS FICTÍCIAS 38

FIGURA 15 - APLICATIVO AUTENTICADO, EXIBINDO UM USUÁRIO FICTÍCIO EM UMA PESQUISA REALIZADA PELO PRATICANTE DE ESPORTE 39

FIGURA 16 - TELA VISUALIZAÇÃO DE PERFIL COM BOTÃO PARA INICIAR UMA CONVERSA 42

FIGURA 17 - TELA DE CHAT COM UMA CONVERSA INICIANDO 43

FIGURA 18 - TELA DE CHAT COM UMA CONVERSA ESTABELECIDA 44

FIGURA 19 - NOTIFICAÇÃO DO ANDROID AVISANDO QUE ALGUÉM ENVIOU UMA MENSAGEM PARA O PRATICANTE DE ESPORTE AUTENTICADO 45

FIGURA 20 - APLICATIVO AUTENTICADO EXIBINDO A ABA DE CONVERSAS 46

FIGURA 21 - DIAGRAMA DE CLASSES DO SPORTAPP 53

FIGURA 22 - DIAGRAMA DE SEQUENCIA DO CASO DE USO AUTENTICAR-SE 54

FIGURA 23 - DIAGRAMA DE SEQUENCIA DO CASO DE USO PROCURAR PESSOAS 55

FIGURA 24 - DIAGRAMA DE SEQUENCIA DO CASO DE USO MANTER CONVERSAS 55

FIGURA 25 - DIAGRAMA DE SEQUENCIA DO CASO DE USO NOTIFICAR PESSOAS 56

FIGURA 26 - TELA DE ADMINISTRAÇÃO DO BANCO DE DADOS DO SPORTAPP, BASEADO NA ESTRUTURA NOSQL 60

FIGURA 27 - TELA DE ADMINISTRAÇÃO DOS USUÁRIOS CADASTRADOS NO APLICATIVO 61

FIGURA 28 - TELA DE ADMINISTRAÇÃO DO ARMAZENAMENTO DE IMAGENS DO SPORTAPP 61

sumário

1 introdução 10

2 LEVANTAMENTO DE REQUISITOS 12

2.1 Requisitos funcionais 12

2.2 Requisitos não funcionais 13

3 escolha da metodologia 15

4 uml – modelagem de dados unificada 17

5 especificação de casos de uso 18

5.1 MANTER PRATICANTE DE ESPORTE 19

5.1.1 Pré-condições 19

5.1.2 Fluxo básico 20

5.1.3 Caminhos alternativos 20

5.2 AUTENTICAR-SE 27

5.2.1 Pré-condições 27

5.2.2 Caminhos alternativos 28

5.3 PROCURAR PESSOAS 36

5.3.1 Pré-requisitos 36

5.3.2 Fluxo básico 36

5.3.3 Fluxos alternativos 36

5.4 MANTER CONVERSAS 40

5.4.1 Pré-requisitos 40

5.4.2 Fluxo básico 40

5.4.3 Fluxos alternativos: 41

5.5 MANTER GRUPOS 46

5.5.1 Pré-requisitos 46

5.5.2 Fluxo básico 47

5.5.3 Caminhos alternativos 47

5.6 NOTIFICAR PESSOAS 49

5.6.1 Pré-requisitos 49

5.6.2 Fluxo básico 49

5.6.3 Fluxo alternativo 49

5.7 ATUALIZAR TOKENS 49

5.7.1 Pré-requisitos 50

5.7.2 Fluxo básico 50

5.7.3 Caminhos alternativos 50

6 diagrama de classes 51

7 diagrama de sequencia 54

8 FIREBASE 57

8.1 ArMAZENAMENTO DE DADOS 57

8.2 Armazentamento de imagens 58

8.3 Autenticação 58

8.4 noticações 59

8.5 utilização do firebase 59

9 ANDROID STUDIO 62

10 padrões de projeto 63

10.1 padrão de projeto observer 63

10.2 padrão de projeto dao 64

11 conclusão 65

12 referencias 67

13 glossario 68

# introdução

Atualmente em consultas realizadas no Google Play, loja de aplicativos do sistema operacional Android, não foram encontrados aplicativos que tenham a finalidade de aproximar as pessoas de acordo com os esportes que elas gostam ou praticam e de acordo com a localização geográfica de cada uma. Uma grande parte dos esportes exige mais de uma pessoa para que possa ser praticado, exemplo: futebol, vôlei, tênis, basquete, entre outros. Somado a isso, de acordo com o Consumer Barometer With Google, ferramenta para ajudar a entender como as pessoas utilizam a Internet pelo mundo (GOOGLE LLC, 2018), no Brasil a cada ano aumenta o uso de dispositivos móveis com acesso a Internet, como pode ser observada na FIGURA 1.

Esta pesquisa propõe uma alternativa para aproximar pessoas de acordo com a sua localização geográfica e os esportes de seu gosto, incentivando a prática de esportes e ajudando as pessoas que querem praticar um esporte e não tem com quem fazer isso. O objetivo da pesquisa é construir um aplicativo de celular, que funcione no sistema operacional Android, que proporcione a possibilidade dos usuários encontrarem e conhecerem pessoas praticantes de esporte, conversarem, trocarem contatos, redes sociais e consequentemente possam marcar jogos com os contatos que mais tiverem afinidade.

Para atingir o objetivo, o sistema foi projetado de uma maneira que não gere custos financeiros de administração do próprio aplicativo, custos com armazenamento de dados ou de utilização de ferramentas pagas, pois o projeto trata-se de um experimento de inovação e no momento não tem fins lucrativos. Além disso, outro pré-requisito do projeto é que o sistema tenha disponibilidade em tempo integral, para que o público alvo possa utilizar o sistema a qualquer momento, sem a necessidade de manter servidores caseiros de armazenamento de dados, que ficariam sujeitos a desligamento devido a quedas de energia, precisariam de algum tipo de administração, gastariam energia elétrica, entre outras desvantagens.

Nesta pesquisa será demonstrado o levantamento de requisitos do sistema, a utilização da metodologia orientada a objetos para construção, bem como os diagramas da metodologia, sendo eles, diagrama de casos de uso, diagrama de classes, diagrama de sequencia, será demonstrado à especificação de casos de uso, será feito um detalhamento de como foram utilizadas as ferramentas que auxiliaram na construção do aplicativo, como o Firebase, o Android Studio, entre outras, além disso, serão demonstrados os casos de testes, os resultados obtidos, o diagrama de navegação nas telas do sistema e por fim, a conclusão desta pesquisa.

FIGURA 1 - PERCENTUAL CRESCENTE DE UTILIZAÇÃO DE DISPOSITIVOS MÓVEIS COM ACESSO A INTERNET NO BRASIL



FONTE: GOOGLE LLC

# LEVANTAMENTO DE REQUISITOS

Neste capítulo será abordado o levantamento de requisitos do sistema, deixando de forma explicita quais serão as suas funcionalidades, ou seja, o que o sistema fará, o que os usuários poderão executar dentro de aplicativo e também serão levantados os requisitos não funcionais, explicando os recursos que o aplicativo precisará utilizar no dispositivo, em qual sistema operacional funcionará o sistema, padrões de tela que devem ser seguidos, requisitos de segurança, limites de conexões de usuários, entre outros. Todos os requisitos serão compostos por um código de identificação, RF trata-se de requisito funcional e RNF trata-se de requisito não funcional. Os requisitos funcionais também terão prioridade que serão divididas em prioridade “essencial” e prioridade “desejável”, uma prioridade essencial significa que é uma funcionalidade indispensável para atingir o objetivo do sistema, de aproximar pessoas pelo esporte, já um requisito funcional “desejável” significa que é uma funcionalidade que é interessante existir, porém não é essencial para atingir o objetivo.

## Requisitos funcionais

**RF01** - O sistema permitirá a manutenção de usuários, contendo nome, e-mail, senha, foto de perfil, localização atual, raio de quilômetro onde deseja encontrar outros usuários e os esportes de seu interesse. Prioridade: Essencial.

**RF02** - O sistema enviará e-mail para confirmação de cadastro, para garantir que o usuário que solicitou o cadastro tinha acesso à conta de e-mail cadastrado. Prioridade: Essencial.

**RF03** – O sistema enviará e-mail para redefinição de senhas esquecidas. Prioridade: Essencial.

**RF04** – O sistema realizará autenticação de usuários, que confirmaram sua conta, com e-mail e senha. Prioridade: Essencial.

**RF05** – O sistema buscará outros usuários que estão dentro do raio de quilômetro definido pelo usuário que está autenticado, mostrando o nome, a foto, os esportes e a distância em quilômetros destes usuários listados. Prioridade: Essencial.

**RF06** – O sistema realizará filtros por esporte, permitindo que o usuário autenticado possa encontrar outros usuários que tem interesse em um esporte específico. Ex: Futebol, Tênis, Vôlei, etc. Prioridade: Essencial.

**RF07** – O sistema permitirá que os usuários possam trocar mensagens via CHAT, para que possam se conhecer e marcar jogos. Prioridade: Essencial.

**RF08** – O sistema permitirá que um usuário autenticado possa visualizar as conversas ele iniciou com outros usuários, ou que outros usuários iniciaram com ele. Prioridade: Essencial.

**RF09** – O sistema permitirá que o usuário autenticado possa filtrar as conversas já existentes com outros usuários, procurando pelo nome de um determinado usuário. Prioridade: Essencial.

**RF10 –** O sistema permitirá que um usuário possa excluir sua conta. Prioridade: Essencial.

**RF11 –** O sistema enviará notificações de mensagens novas. Propriedade: Desejável.

**RF12 –** O sistema permitirá que os usuários possam montar grupos de um esporte específico, contendo até trinta participantes, para poderem trocar mensagens em grupo. Prioridade: Desejável.

## Requisitos não funcionais

**RNF01** – O sistema deverá rodar em plataforma Android 5.0 Lollipop ou superior.

**RNF02** – O sistema deverá utilizar o Firebaseauthentication do Google para realizar a autenticação de usuários.

**RNF03** – O sistema deverá armazenar os dados em um banco de dados Firebase Real Time Database do Google, para mantê-los disponíveis para consulta 24 horas por dia.

**RNF04** – O sistema deverá utilizar o Firebasestorage do Google para armazenar as fotos e mantê-las disponíveis para consulta 24 horas por dia.

**RNF05** – As telas do sistema devem seguir o padrão Material Design do Google, proporcionando uma usabilidade parecida com aplicativos conhecidos de mensagem instantânea.

**RNF06** – O sistema deve ter acesso à internet.

**RNF07** - O sistema deve ter acesso à localização do dispositivo.

**RNF08** - O sistema deve ter acesso à câmera do dispositivo.

**RNF09** –O sistema deve ter acesso ao armazenamento interno do dispositivo.

**RNF10** – O sistema seguirá as recomendações de segurança do Firebase, onde apenas usuários autenticados, com uma conta confirmada, podem ler e escrever no banco de dados.

**RNF11** – O sistema não gravará nenhum dado importante no armazenamento interno do dispositivo, para evitar que dados sejam roubados em caso de perda do celular.

**RNF12** – As senhas de usuário não serão visíveis mesmo para alguém que tenha acesso direto ao banco de dados.

**RNF13** – O sistema usará o plano Spark do Firebase, que é gratuito, porém estará sujeito aos limites de armazenamento e conexões simultâneas deste plano: No máximo 100 conexões de usuários simultâneas, 1GB de armazenamento de texto, 5GB de armazenamento de imagens. (GOOGLE LLC. 2018)

# escolha da metodologia

A orientação a objetos existe desde o ano de 1960, aproximadamente, e tem sido cada vez mais utilizada por todo o setor de desenvolvimento de software, é presente em diversas linguagens de programação como Java, C++, C# entre outras. Quando se deu inicio a programação para computadores, os programadores utilizavam linguagens de programação binária que eram inseridas diretamente na memória do computador, porém os códigos em linguagens binárias eram difíceis para dar manutenção e ficavam propensos a erros e falta de estrutura, tornando a manutenção quase impossível. (SINTES, 2002, p. 4)

Com o passar do tempo foram aparecendo novas linguagens de programação classificadas como procedurais, como o Fortran por exemplo. As linguagens procedurais permitiram que os programadores separassem os códigos em procedimentos que executam o processamento de dados e esses procedimentos eram chamados em uma ordem sequencial, definindo a estrutura dos programas. O paradigma procedural trouxe melhorias para a manutenção dos códigos, uma vez que, funcionalidades pequenas de um sistema ficavam agrupadas dentro de um procedimento, facilitando o entendimento do código. Porém, no paradigma procedural ainda existiam dificuldades como, por exemplo, os dados do programa e os próprios procedimentos, ficavam separados, exigindo que cada procedimento saiba como manipular dados corretamente, fazendo com que procedimentos tenham acesso a dados que não deveriam ou não precisariam ter, e ainda, a dificuldade dos programadores de conseguirem separar os procedimentos para cada assunto que o software trata, causando códigos muito extensos e sem coesão. (SINTES, 2002, p. 5)

Após algumas evoluções nos paradigmas, surge à orientação a objetos, que divide os programas em vários objetos. Cada objeto representa um aspecto do problema que o software tenta resolver. Nos objetos os dados são representados pelos atributos e os procedimentos são representados pelas funções, cada atributo guarda uma característica do objeto e cada função executa um procedimento relacionado a aquele objeto, além disso, é possível encapsular os dados e procedimentos, fazendo com que os outros objetos do software só conheçam os dados e métodos que realmente serão necessários para cumprir seu papel na execução do programa. Os objetos interagem entre si, permitem que o programador possa separar melhor os códigos e que seja possível representar melhor o mundo real dentro do software. As principais características da orientação a objetos são:

* Herança: Capacidade de um objeto herdar comportamentos de outros objetos, permitindo a reutilização de código.
* Abstração: Capacidade de abstrair o mundo real dentro do programa, representando qualquer entidade do mundo real por meio das classes.
* Encapsulamento: Capacidade de isolamento dos métodos e atributos de acordo com a necessidade de acesso de cada objeto
* Polimorfismo: Capacidade de mudar o comportamento de um método dentro de diferentes objetos. (SINTES, 2002, p. 6)

Para a codificação deste projeto, foi escolhida a linguagem de programação Java, que permite a programação para o sistema operacional Android e não gera nenhum custo financeiro. A linguagem Java é orientada a objetos, por este motivo a metodologia escolhida para aplicar ao projeto é a orientação a objetos, além disso, é possível perceber que a orientação a objetos faz parte de uma evolução nos paradigmas de programação trazendo vantagens, citadas anteriormente, em relação a outros paradigmas existentes.

# uml – modelagem de dados unificada

A UML, sigla em inglês para linguagem de modelagem unificada, é uma linguagem visual, representada por diagramas, utilizada para modelar sistemas orientados a objetos. Não é uma linguagem de programação, trata-se de uma linguagem de modelagem para auxiliar na definição de requisitos, funcionalidades, comportamento e estrutura do software. (GUEDES, 2011, P. 19)

Nesta pesquisa, serão apresentados os seguintes diagramas da UML: Diagrama de casos de uso, que permitirá visualizar todas as ações que o usuário pode fazer com o aplicativo e como o sistema irá se comportar, o diagrama de classes que definirá toda a estrutura das classes existentes no projeto e diagrama de sequencia, que mostrará visualmente o fluxo de execução dos principais procedimentos do aplicativo. Dentro da linguagem de modelagem de linguagem unificada existem vários outros diagramas além dos citados neste projeto, porém nessa pesquisa serão aplicados estes três diagramas.

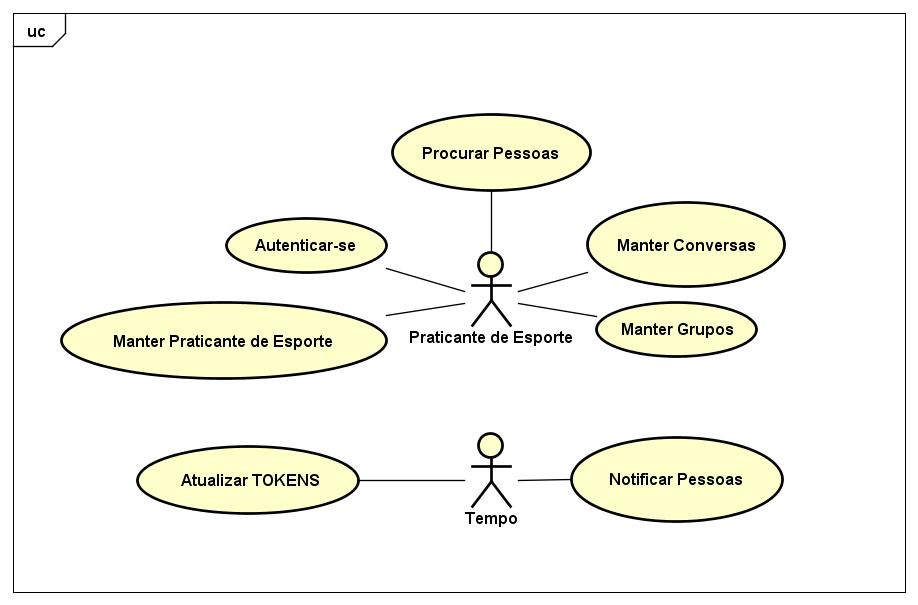
# especificação de casos de uso

A especificação dos casos de uso detalha de forma textual cada passo que o ator precisa fazer para executar uma ação dentro do sistema e como o sistema se comporta, inclusive caso aconteça qualquer tipo de exceção em relação ao fluxo padrão.

O diagrama de casos de uso demonstra uma linguagem simples, para apresentar como o sistema irá se comportar. Identifica os atores, representados por um boneco, que serão efetivamente os usuários do sistema e quais ações estes usuários podem ter durante a execução do programa, estas ações representadas por balões ovais, contendo o nome da ação. (GUEDES, 2011, P. 31)

A FIGURA 2 mostra o diagrama de casos de uso do sistema Sportapp, é possível visualizar um ator denominado “praticante de esporte”, que será o usuário praticante de esporte que pretende encontrar outras pessoas. É possível visualizar também um ator chamado “Tempo”, esse ator representa o próprio sistema que com o passar do tempo pode encadear algumas ações automáticas, como por exemplo, a ação de notificar um usuário “B”, quando um usuário “A” envia uma mensagem e este usuário “B” não está olhando para a tela do celular.

FIGURA 2 - DIAGRAMA DE CASOS DE USO DO SISTEMA SPORTAPP

****

FONTE: O PRÓPRIO AUTOR

## MANTER PRATICANTE DE ESPORTE

O caso de uso permite ao usuário praticante de esporte cadastrar-se no aplicativo, bem como, atualizar seus dados e até mesmo excluir seu próprio cadastro. As figuras 3, 4, 5, 6 e 7, exibidas após o detalhamento deste caso de uso, mostram as telas que o ator utiliza para fazer as ações presentes neste caso de uso.

### Pré-condições

Para cadastrar-se não existem pré-condições. Para atualizar ou excluir dados é necessário estar autenticado no aplicativo. Para todos os casos é necessário conexão com a internet.

### Fluxo básico

1. Solicitar a tela de cadastro de novos usuários: O caso de uso se inicia quando o praticante de esporte solicita a tela de cadastro de novos usuários. O sistema exibe a tela solicitada.
2. Preencher campo nome: O praticante de esporte digita seu nome e o sistema mantém os dados pendentes para inclusão.
3. Preencher campo e-mail: O praticante de esporte digita seu e-mail e o sistema mantém os dados pendentes para inclusão.
4. Preencher campo senha: O praticante de esporte digita sua senha e o sistema mantém os dados pendentes para inclusão.
5. Cadastrar usuário: O praticante de esporte clica no botão “Cadastrar”, o sistema grava os dados do usuário no banco de dados, envia um e-mail de confirmação de cadastro para o endereço de e-mail utilizado pelo praticante de esporte, mostra a mensagem “E-mail de confirmação enviado. Verifique sua caixa de entrada.”, redireciona para a tela inicial de LOGIN, mostrando a seguinte mensagem fixa na tela, com a cor vermelha, “Entre na sua caixa de e-mail e confirme sua conta. Caso não tenha recebido o e-mail de confirmação, toque aqui para reenviar” e o caso de uso termina.

### Caminhos alternativos

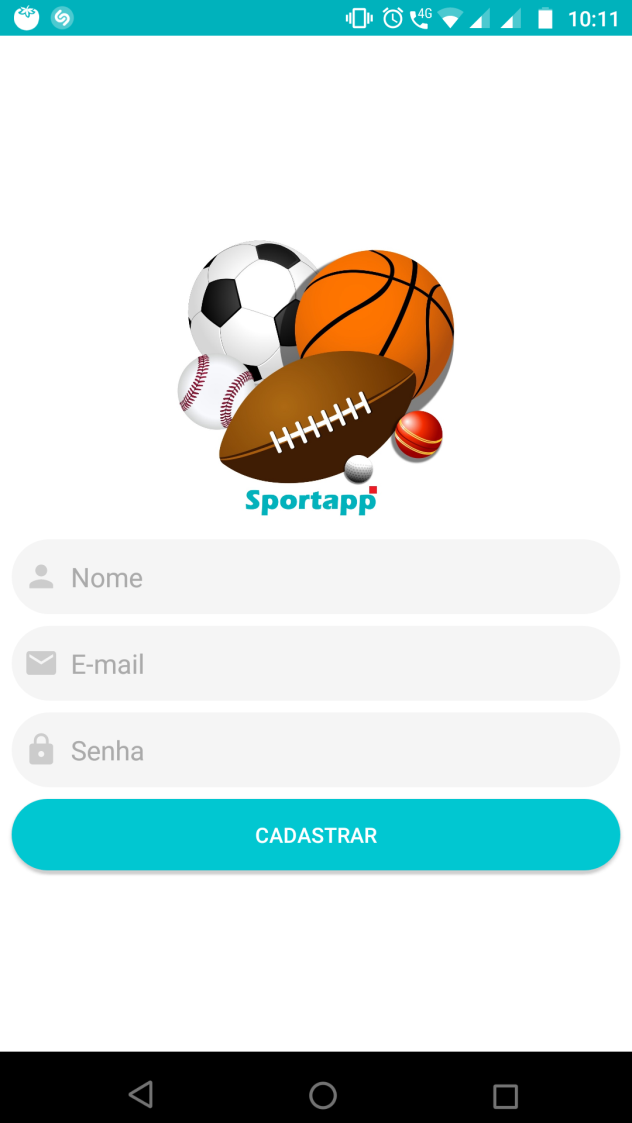
1. Não é possível estabelecer uma conexão com o banco de dados: Durante a execução do passo “Cadastrar usuário” caso não seja possível estabelecer uma conexão com o banco de dados o sistema mostra a mensagem “Erro desconhecido. Verifique sua conexão com a internet ou tente novamente mais tarde.” e o caso de uso continua sua execução a partir do passo “Cadastrar usuário”.
2. Campo nome não preenchido: Durante a execução do passo “Cadastrar usuário” caso o campo “Nome” não esteja preenchido, o sistema mostra a seguinte mensagem “Preencha o campo nome.”. O sistema mantém a execução a partir do passo “Preencher campo Nome”.
3. Campo e-mail não preenchido: Durante a execução do passo “Cadastrar usuário” caso o campo “E-mail” não esteja preenchido, o sistema mostra a seguinte mensagem “Preencha o campo e-mail”. O sistema mantém a execução a partir do passo “Preencher campo E-mail”.
4. Campo senha não preenchido: Durante a execução do passo “Cadastrar usuário” caso o campo “Senha” não esteja preenchido, o sistema mostra a seguinte mensagem “Preencha o campo senha”. O sistema mantém a execução a partir do passo “Preencher campo Senha”.
5. E-mail já cadastrado: Durante a execução do passo “Cadastrar usuário” caso o e-mail utilizado pelo praticante de esporte já esteja cadastrado no banco de dados, o sistema mostra a seguinte mensagem “E-mail já cadastrado”. O sistema mantém a execução a partir do passo “Preencher campo E-mail”.
6. E-mail inválido: Durante a execução do passo “Cadastrar usuário**”,** caso o e-mail digitado pelo praticante de esporte seja inválido ou não exista, o sistema mostra a seguinte mensagem “E-mail inválido.”. O sistema mantém a execução a partir do passo “Preencher campo E-mail”.
7. Senha fraca: Durante a execução do passo “Cadastrar usuário**”,** caso a senha digitada pelo usuário não seja aceita pelo FIREBASEAUTH, o sistema mostra a seguinte mensagem “Digite uma senha mais forte.”. O sistema mantém a execução a partir do passo “Preencher campo Senha”.
8. Atualizar dados: Estando autenticado, o usuário pode a qualquer momento solicitar a tela de configurações para atualizar os dados do seu perfil. O sistema abre a tela de configurações exibindo os dados do praticante de esporte. O praticante de esporte pode atualizar os dados e o sistema mantém estes dados pendentes para atualização. O praticante de esporte toca no botão “Atualizar”, o sistema grava os dados do usuário no banco de dados, mostra a mensagem “Sucesso ao gravar dados do usuário.” e o caso de uso termina.
9. Atualizar a foto do perfil tirando uma nova foto: Estando autenticado, o usuário pode a qualquer momento solicitar a tela de configurações para atualizar a foto de seu perfil. O sistema abre a tela de configurações exibindo alguma foto cadastrada anteriormente ou a foto de perfil padrão do sistema. O praticante de esporte toca no botão para tirar uma nova foto, o sistema abre a câmera e o usuário tira uma nova fotografia. Após tirar a nova foto o usuário confirma se deseja utilizar essa foto ou se deseja tirar outra fotografia. Caso o usuário confirme a utilização da nova foto, o sistema exibe esta imagem na tela de configurações, faz a gravação no banco de dados e mostra a seguinte mensagem “Imagem atualizada com sucesso.” e o caso de uso termina.
10. Atualizar a foto do perfil utilizando uma imagem do armazenamento do dispositivo: Estando autenticado, o usuário pode a qualquer momento solicitar a tela de configurações para atualizar a foto de seu perfil. O sistema abre a tela de configurações exibindo alguma foto cadastrada anteriormente ou a foto de perfil padrão do sistema. O praticante de esporte toca no botão para escolher uma imagem da galeria do dispositivo, o sistema abre a galeria de fotos do dispositivo. O praticante de esporte escolhe a imagem que deseja usar, o sistema exibe esta imagem na tela de configurações, faz a gravação no banco de dados e mostra a seguinte mensagem “Imagem atualizada com sucesso.” e o caso de uso termina.
11. Excluir conta: Estando autenticado, o usuário pode a qualquer momento solicitar a tela de configurações para excluir sua própria conta. O sistema abre a tela de configurações exibindo os dados já cadastrados anteriormente. O usuário toca na frase fixa de cor vermelha “APAGAR CONTA” que existe na tela, o sistema emite uma pergunta se o usuário deseja realmente apagar sua conta e caso a resposta seja positiva, o sistema excluí os dados do usuário do banco de dados, desfaz a autenticação, retorna para a tela inicial do aplicativo e o caso de uso termina.
12. Confirmar cadastro: Durante a execução do passo “Cadastrar usuário”, o praticante de esporte recebe um e-mail para confirmar sua conta. Ao acessar seu e-mail o praticante de esporte clica no LINK contido no conteúdo do e-mail, neste momento o sistema grava a confirmação do cadastro do usuário, o praticante de esporte é redirecionado para uma página WEB informando que seu cadastro foi confirmado e o caso de uso termina.

FIGURA 3 - TELA INICIAL DO SISTEMA



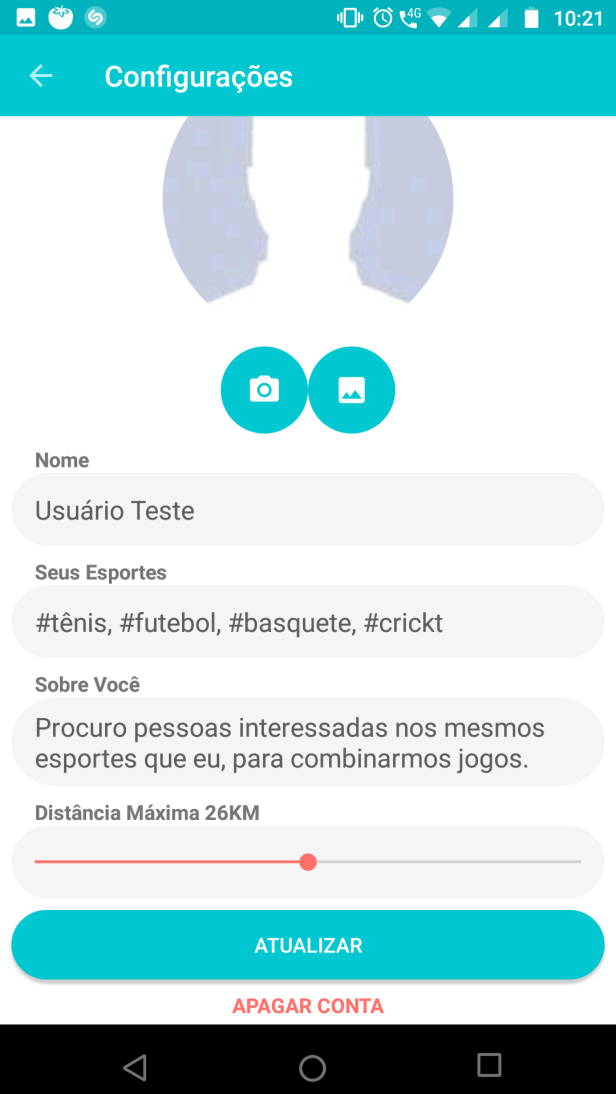
FONTE: O PRÓPRIO AUTOR

FIGURA 4 - TELA DE CADASTRO



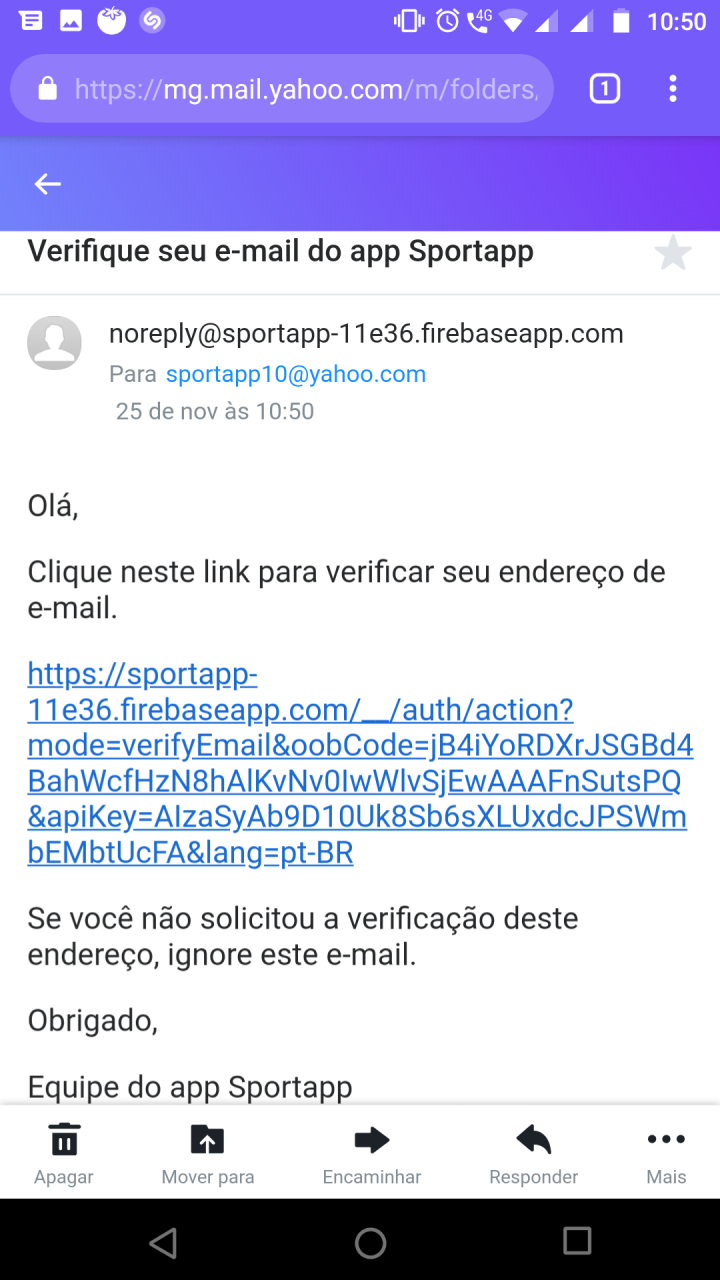
FONTE: O PRÓPRIO AUTOR

FIGURA 5 - TELA DE CONFIGURAÇÕES, ATUALIZAÇÃO DE PERFIL E EXCLUSÃO DE CONTA.



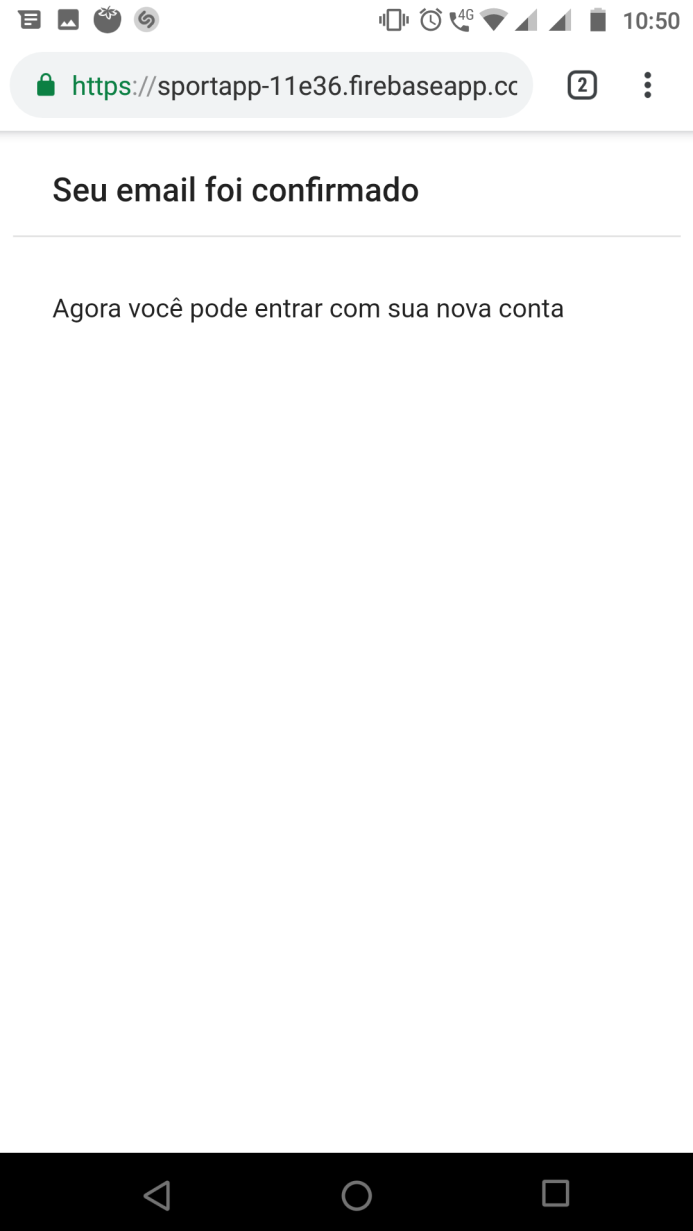
FONTE: O PRÓPRIO AUTOR

FIGURA 6 - E-MAIL DE CONFIRMAÇÃO DE CONTA



FONTE: O PRÓPRIO AUTOR

FIGURA 7 - CONFIRMAÇÃO DE CONTA CONCLUÍDA



FONTE: O PRÓPRIO AUTOR

## AUTENTICAR-SE

O caso de uso permite ao usuário praticante de esporte autenticar-se no aplicativo. As figuras 8, 9, 10, 11, 12 e 13, exibidas após o detalhamento deste caso de uso, mostram as telas que o ator utiliza para fazer as ações presentes neste caso de uso.

### Pré-condições

Necessário que o praticante de esporte já tenha se cadastrado no aplicativo e que sua conta de e-mail já tenha sido confirmada, necessário conexão com a internet.

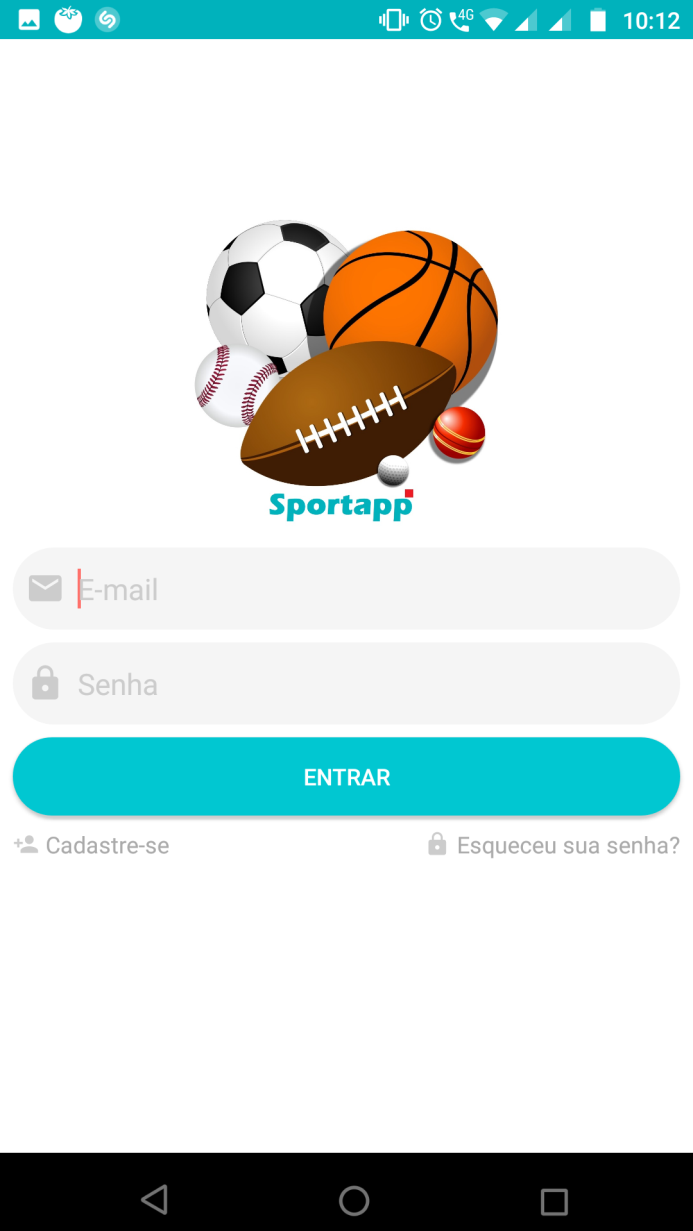
Fluxo básico

1. Preencher campo “E-mail”: O caso de uso se inicia quando o praticante de esporte abre o sistema e não está autenticado, o sistema abre a tela inicial de LOGIN. O praticante de esporte preenche o campo “E-mail”. O sistema mantém o dado pendente para utilização.
2. Preencher campo “Senha”: O praticante de esporte preenche o campo “Senha”. O sistema mantém o dado pendente para utilização.
3. Autenticar-se: O praticante de esporte clica no botão “Entrar”. O sistema confirma no banco de dados se o usuário e senha são válidos, caso sejam, o sistema redireciona o usuário para a tela de busca de pessoas e o caso de uso termina.

### Caminhos alternativos

1. Usuário não cadastrado: Durante a execução do passo “Autenticar-se” caso o e-mail digitado não esteja cadastrado no banco de dados, o sistema mostra a seguinte mensagem “E-mail não cadastrado.”. O sistema continua a execução a partir do passo “Preencher e-mail”.
2. Senha inválida: Durante a execução do passo “Autenticar-se”, caso o e-mail digitado esteja cadastrado no banco de dados, porém a senha digitada não esteja correta, o sistema mostra a seguinte mensagem “Senha incorreta”. O sistema continua a execução a partir do passo “Preencher senha”.
3. Não é possível estabelecer uma conexão com o banco de dados: Durante a execução do passo “Autenticar-se”, caso o não seja possível estabelecer uma conexão com o banco de dados, o sistema mostra a seguinte mensagem: “Erro desconhecido. Verifique sua conexão com a internet ou tente novamente mais tarde”. O sistema continua a execução a partir do passo “Autenticar-se”.
4. E-mail não confirmado: Durante a execução do passo “Autenticar-se”, caso o praticante de esporte ainda não tenha confirmado seu e-mail de cadastro, o sistema mostra uma mensagem fixa na tela, com a cor vermelha “Entre na sua caixa de e-mail e confirme sua conta. Caso não tenha recebido o e-mail de confirmação, toque aqui para reenviar”. A execução continua a partir do passo “Autenticar-se”.
5. Reenviar e-mail de confirmação: Durante a execução do fluxo alternativo “E-mail não confirmado”, caso o praticante de esporte toque na mensagem para reenviar o e-mail de confirmação, o sistema reenvia o e-mail confirmação de cadastro para o e-mail do usuário e mostra a seguinte mensagem “E-mail de confirmação enviado. Verifique sua caixa de entrada.”. A execução continua a partir do passo “Autenticar-se”.
6. Recuperar senha: Durante a execução do passo “Autenticar-se”, caso o praticante de esporte não se lembre da sua própria senha, o praticante de esporte toca no botão “Esqueceu sua senha?”, o sistema abre a tela de recuperação de senha. O praticante de esporte digita o seu e-mail de cadastro e o sistema envia um e-mail para o endereço de e-mail cadastrado contendo um LINK para uma página WEB onde o praticante de esporte pode digitar uma nova senha e confirmar. Após confirmar sua senha na página WEB, o sistema grava a nova senha no banco de dados e o caso de uso termina.
7. Realizar LOGOFF: Após autenticar-se, o praticante de esporte pode a qualquer momento tocar na opção “Sair”. O sistema faz o LOGOFF do praticante de esportes, redireciona para a tela inicial de LOGIN e o caso de uso termina.

FIGURA 8 - TELA DE LOGIN



FONTE: O PRÓPRIO AUTOR

FIGURA 9 - TELA DE LOGIN AVISANDO SOBRE A CONFIRMAÇÃO DE CONTA



FONTE: O PRÓPRIO AUTOR

FIGURA 10 - TELA DE RECUPERAÇÃO DE SENHA

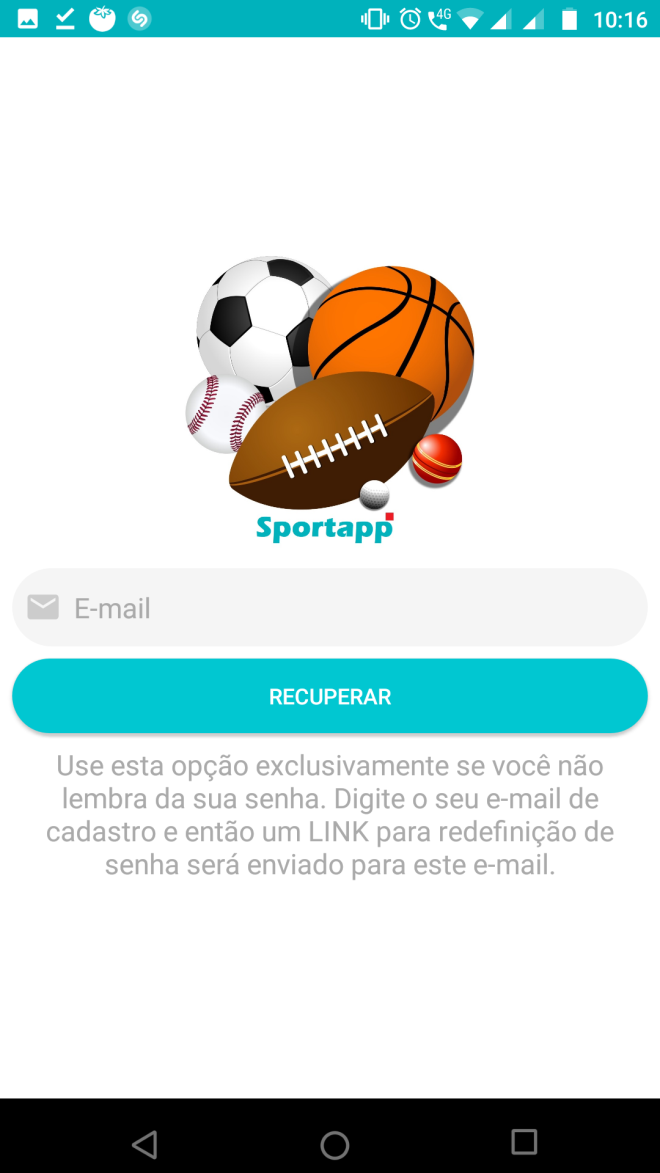
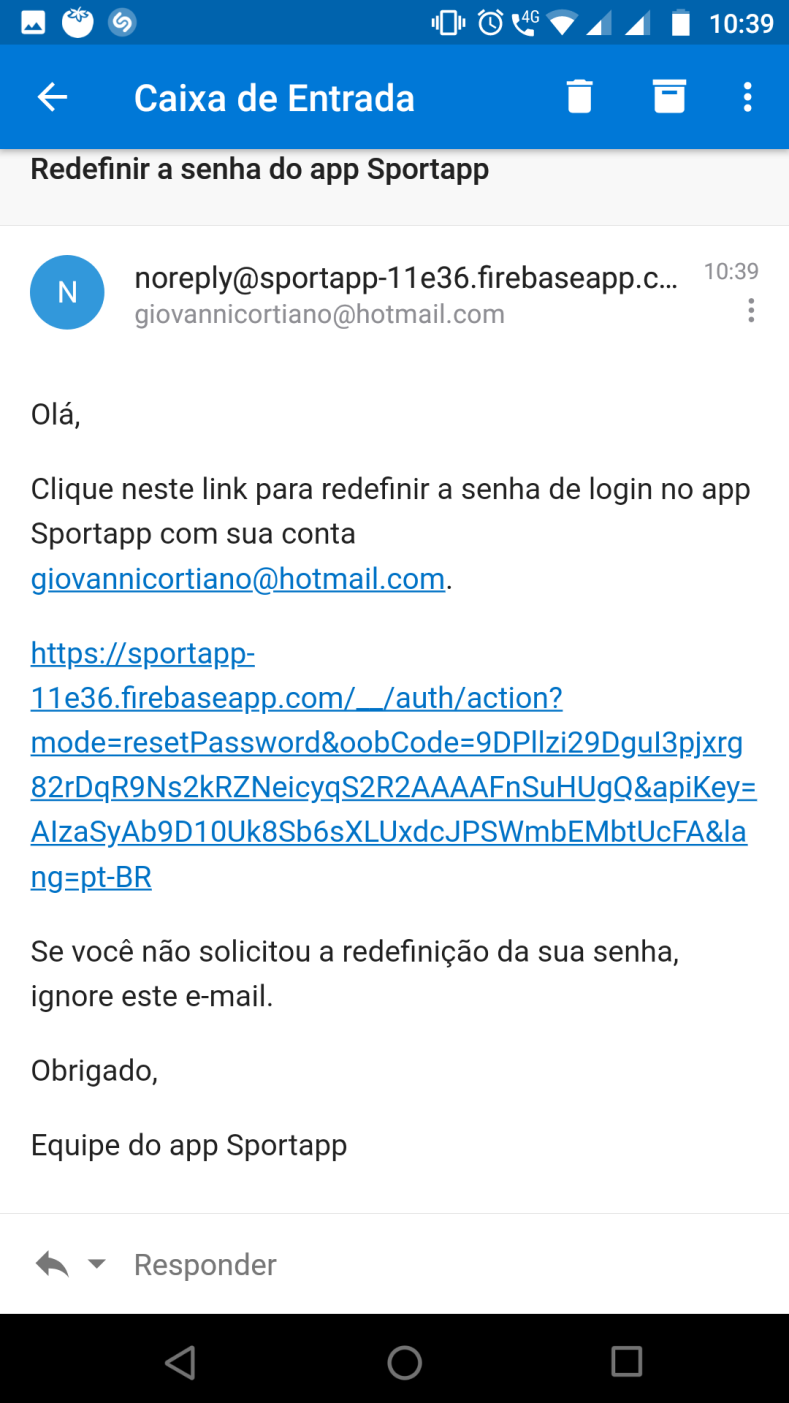
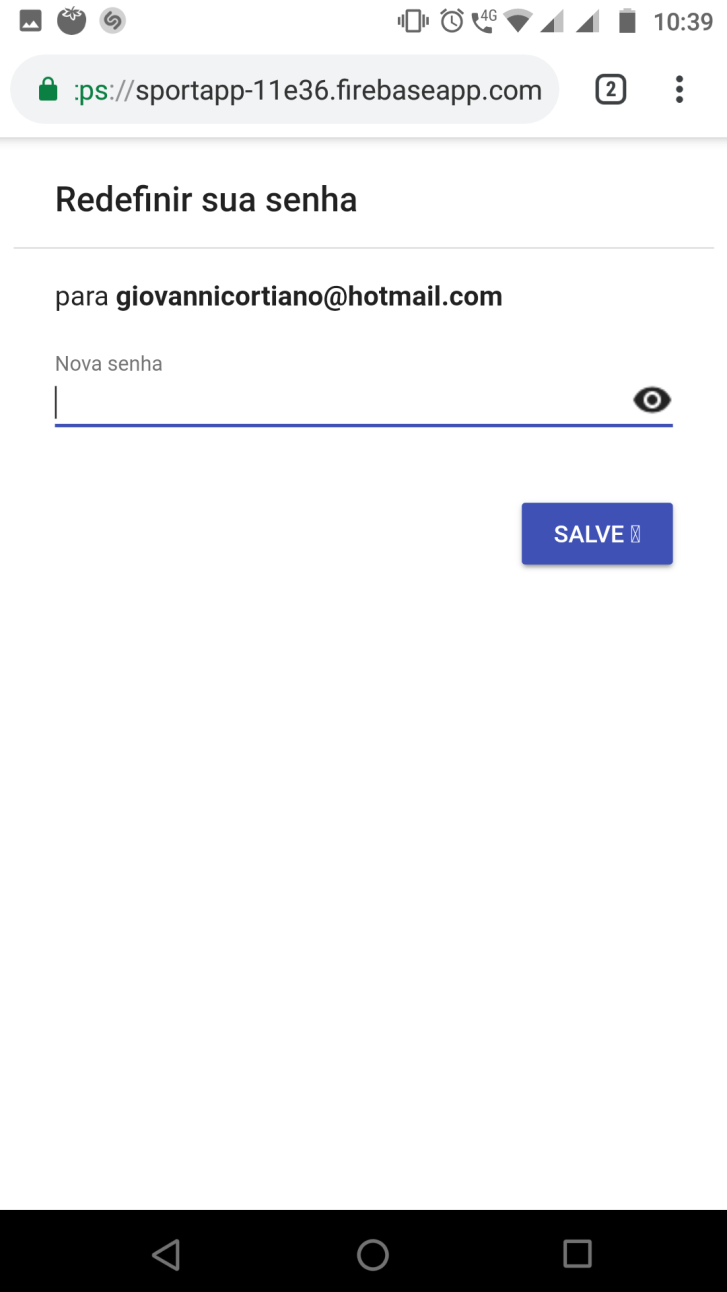


FIGURA 11 - E-MAIL DE RECUPERAÇÃO DE SENHA



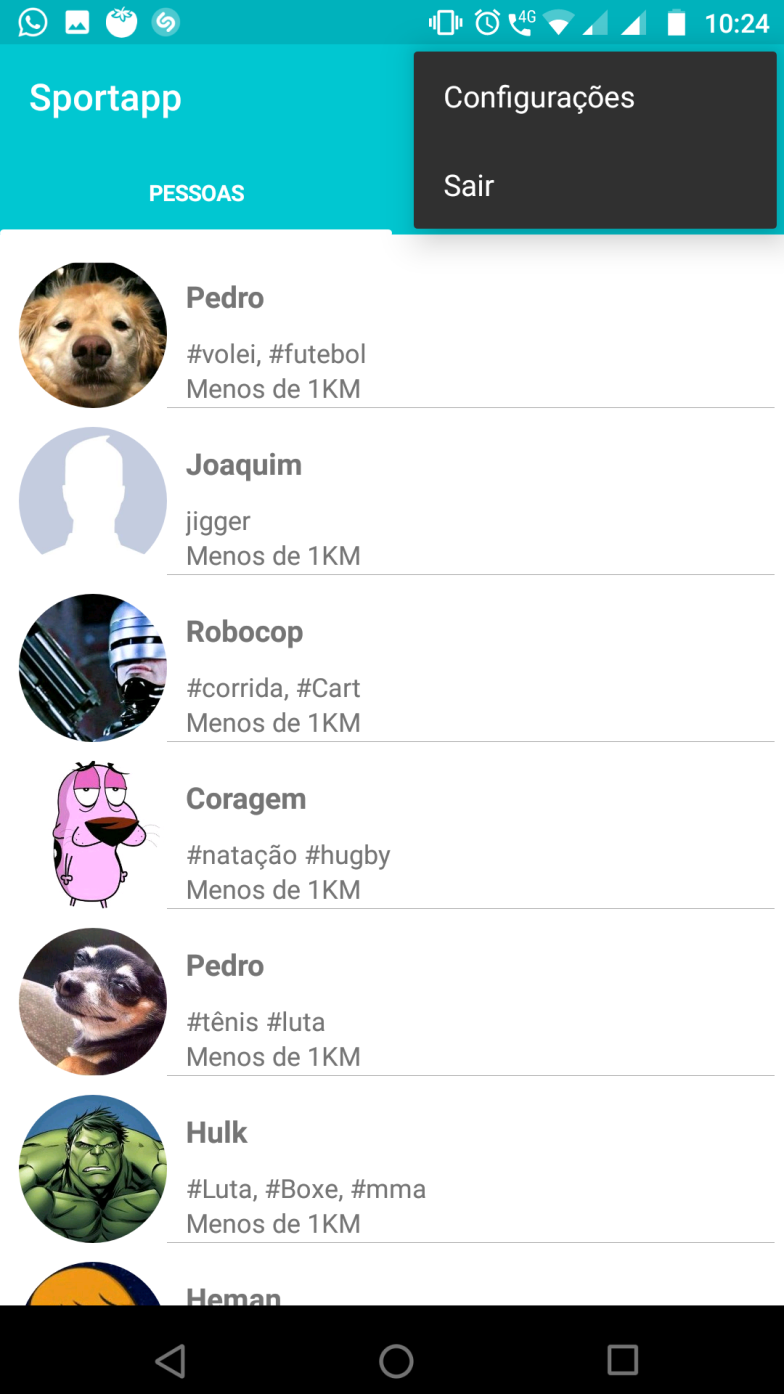
FONTE: O PRÓPRIO AUTOR

FIGURA 12 - PÁGINA WEB PARA REDEFINIÇÃO DE SENHA



FONTE: O PRÓPRIO AUTOR

FIGURA 13 - APLICATIVO AUTENTICADO, EXIBINDO AS OPÇÕES DE CONFIGURAÇÕES E SAIR/LOGOFF



FONTE: O PRÓPRIO AUTOR

## PROCURAR PESSOAS

O caso de uso permite que um praticante de esporte possa visualizar as pessoas que estão cadastradas no aplicativo e que tem uma distância próxima. Além disso, também permite que um praticante de esporte possa procurar por um esporte em específico, como vôlei, por exemplo, e então encontrar outras pessoas próximas que também tem interesse em vôlei. As figuras 14 e 15, exibidas após o detalhamento deste caso de uso, mostram as telas que o ator utiliza para fazer as ações presentes neste caso de uso.

### Pré-requisitos

Estar autenticado, conexão com a internet.

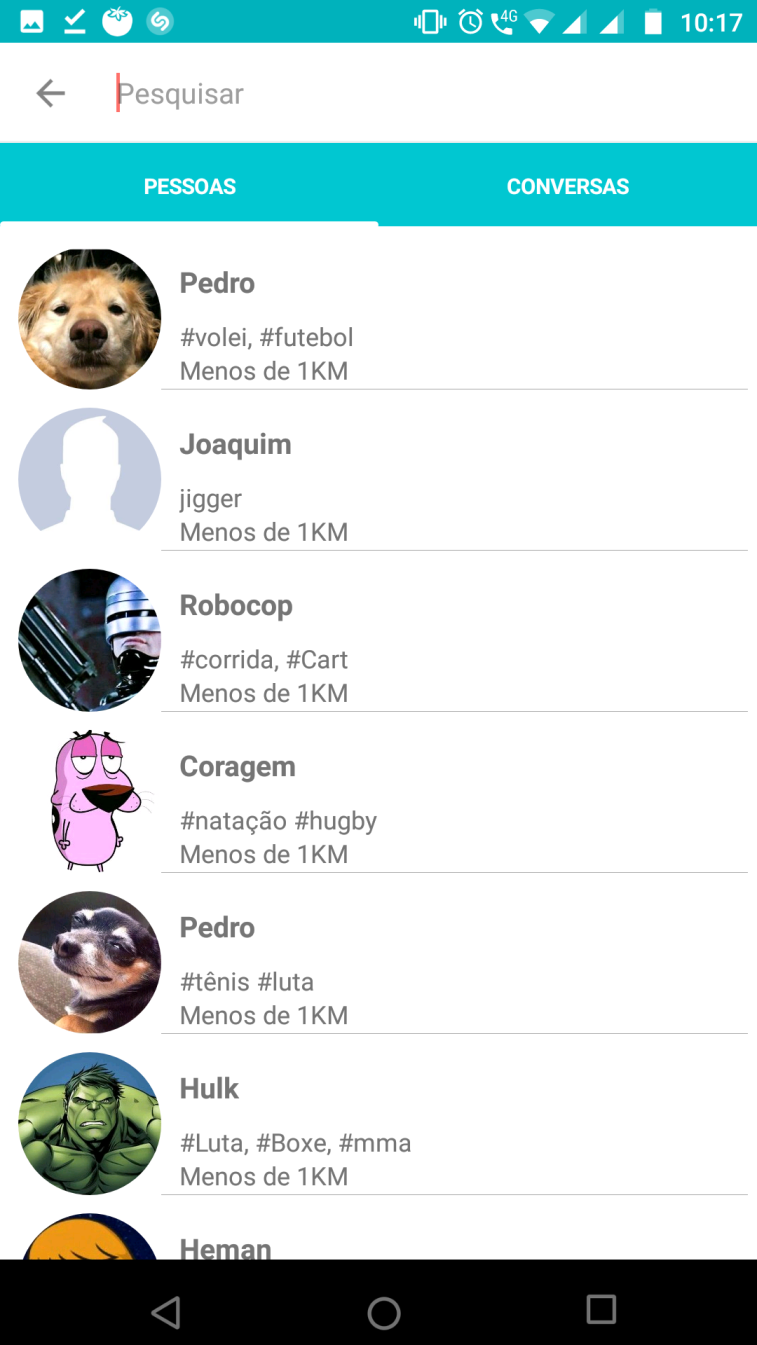
### Fluxo básico

1. Encontrar pessoas: O caso de uso se inicia quando o praticante de esportes faz a autenticação no sistema. O sistema abre a tela de pesquisa de pessoas, lista cinquenta pessoas que estejam dentro do raio de quilometro do usuário, independente de esporte, mostrando o nome, os esportes a foto e a distância aproximada em quilômetros dessas pessoas. O sistema fica disponível para que o praticante de esportes possa navegar na lista de pessoas ou digitar um esporte.
2. Pesquisar Esporte: O praticante de esporte digita na barra superior do sistema, o esporte que deseja procurar, o sistema faz um filtro no banco de dados, listando cinquenta pessoas que estejam dentro do raio de quilometro do praticante de esporte e que contenham em seus perfis o esporte digitado. O sistema mostra o nome, os esportes a foto e a distância aproximada em quilômetros dessas pessoas, fica disponível para que o praticante de esporte possa navegar na lista de pessoas ou procurar outro esporte e o caso de uso termina.

### Fluxos alternativos

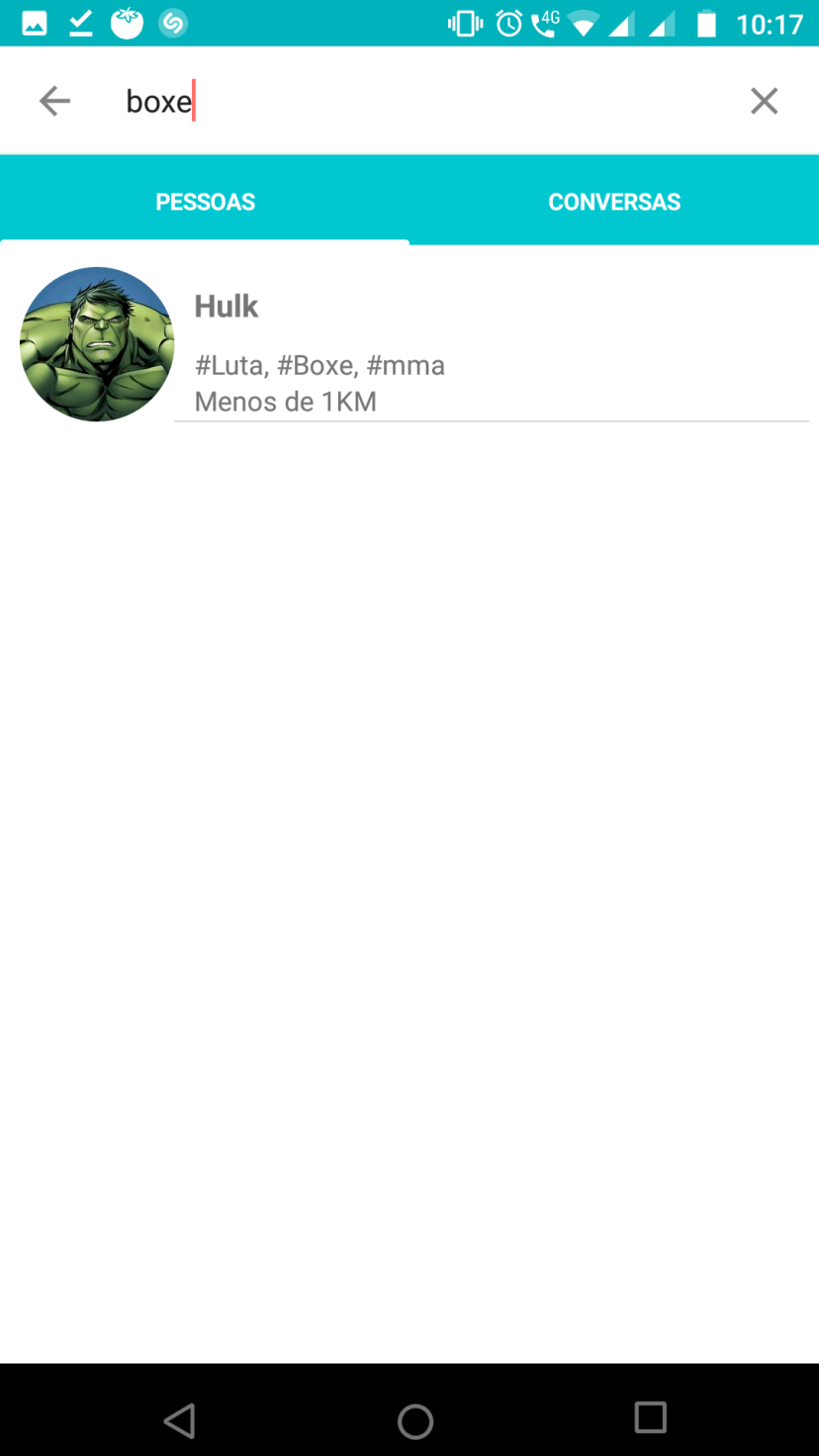
1. Listar mais pessoas: Durante a execução de qualquer passo, caso o praticante de esporte navegue até o final da lista de cinquenta pessoas, o sistema busca mais cinquenta pessoas caso existam e lista esses dados em tela, mostrando o nome, os esportes a foto e a distância aproximada em quilômetros. O sistema fica disponível para que o praticante de esportes navegue pela lista de pessoas e o caso de uso termina.
2. Nenhuma pessoa encontrada: Durante a execução dos passos “Encontrar pessoas” ou “Pesquisar Esporte”, caso não exista nenhum dado para ser mostrado, o sistema exibe a mensagem “Nenhuma pessoa encontrada” de forma fixa na tela.
3. Visualizar perfil: Durante a execução de qualquer passo, caso o praticante de esporte clique sobre o perfil de qualquer pessoa da lista, o sistema abre uma tela para visualização do perfil dessa pessoa selecionada, contendo a foto ampliada, os esportes da pessoa, a descrição do perfil e distância aproximada em quilômetros entre o praticante de esporte e a pessoa selecionada.
4. Reiniciar pesquisa: Durante a execução do passo “Pesquisar Esporte”, caso o praticante de esporte apague o conteúdo da pesquisa, o sistema desfaz o filtro existente e volta a mostrar cinquenta pessoas que estejam dentro do raio de quilometro, independe de esporte.
5. Não é possível estabelecer uma conexão com o banco de dados: Durante a execução de qualquer passo, caso o não seja possível estabelecer uma conexão com o banco de dados, o sistema mostra a seguinte mensagem: “Erro desconhecido. Verifique sua conexão com a internet ou tente novamente mais tarde”. O sistema continua a execução a partir do passo “Listar pessoas”.

FIGURA 14 - APLICATIVO AUTENTICADO EXIBINDO UMA LISTA DE PESSOAS FICTÍCIAS



FONTE: O PRÓPRIO AUTOR

FIGURA 15 - APLICATIVO AUTENTICADO, EXIBINDO UM USUÁRIO FICTÍCIO EM UMA PESQUISA REALIZADA PELO PRATICANTE DE ESPORTE



FONTE: O PRÓPRIO AUTOR

## MANTER CONVERSAS

O caso de uso permite que um praticante de esporte possa se comunicar com outro por meio de mensagens de texto, cumprindo o objetivo principal do aplicativo, que é realizar a aproximação de pessoas que tem um interesse em comum, o esporte, seja qual for. As figuras 16, 17, 18, 19 e 20, exibidas após o detalhamento deste caso de uso, mostram as telas que o ator utiliza para fazer as ações presentes neste caso de uso.

### Pré-requisitos

Estar autenticado no sistema, ter outro usuário praticante de esporte cadastrado no sistema além do próprio praticante de esportes autenticado, conexão com a internet.

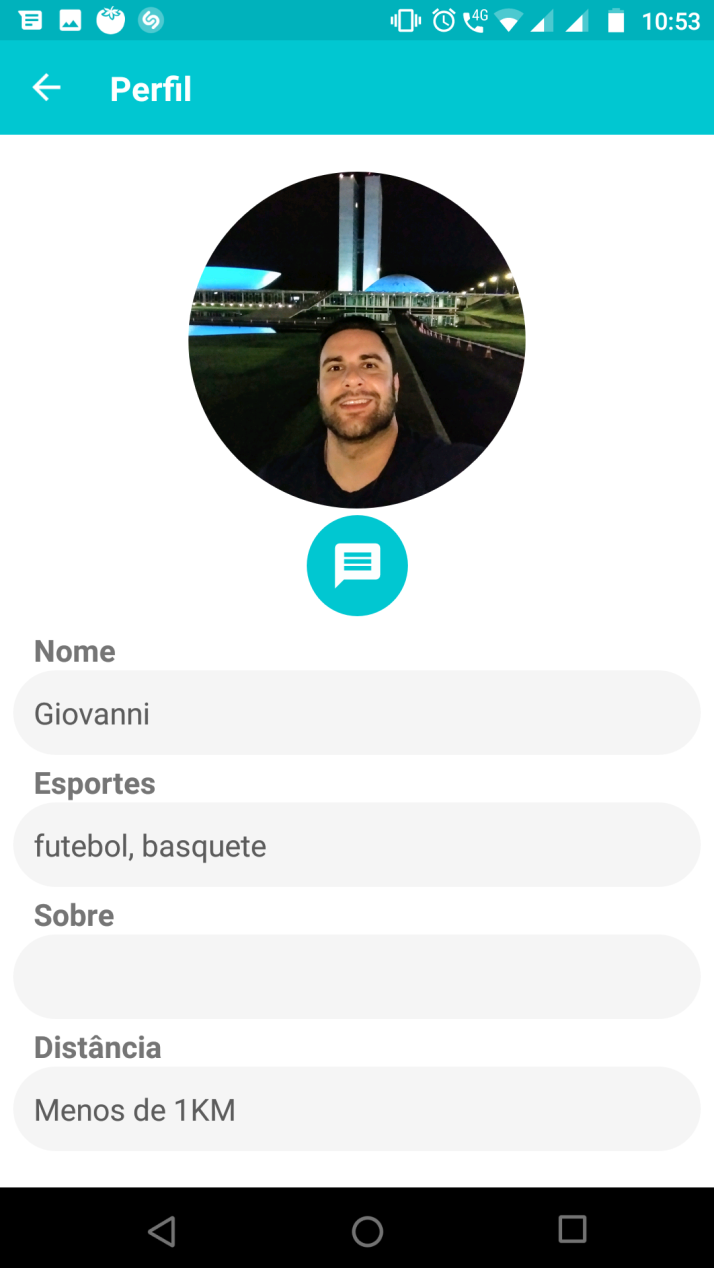
### Fluxo básico

1. Iniciar conversa: O caso de uso se inicia quando o praticante de esportes está visualizando a tela de perfil de outra pessoa, caso o praticante de esporte toque no botão de mensagem, o sistema abre a tela de CHAT para que uma conversa possa iniciar.
2. Enviar mensagem: O praticante de esporte digita a mensagem que deseja enviar e toca no botão de enviar mensagem. O sistema grava esta mensagem no banco de dados e mostra na tela de conversa, dentro de um balão azul, para que o praticante de esporte identifique que foi uma mensagem enviada por ele à outra pessoa. O sistema se mantém disponível para que o praticante de esportes possa enviar mais mensagens.
3. Receber mensagem: Ao abrir a tela de CHAT, o sistema adiciona um LISTNER no banco de dados para receber em tempo real qualquer mensagem que a pessoa do outro lado da conversa envie. Ao receber uma mensagem o sistema mostra o conteúdo desta dentro de um balão da cor cinza, para que o praticante de esportes possa identificar que se trata de uma mensagem vinda da outra pessoa participante da conversa e o caso de uso termina.

### Fluxos alternativos:

1. Visualizar perfil da pessoa: Durante a execução de qualquer passo, caso o praticante de esporte toque foto da outra pessoa participante da conversa, o sistema abre a tela de perfil mostrando os dados do outro participante da conversa e fica disponível para que o praticante de esporte possa voltar para a conversa ao tocar no botão voltar.
2. Visualizar conversa já existente: Durante e execução de qualquer passo, o praticante de esporte acessa a aba “Conversas”, o sistema lista todas as conversas que o praticante de esporte tem registrado com outras pessoas, mostrando dados do outro participante da conversa, como foto, nome e última mensagem da conversa. O sistema fica disponível para que o praticante de esporte possa navegar entre a lista de conversas.
3. Abrir conversa: Durante a execução do passo “Visualizar conversa já existente”, o praticante de esporte toca em uma conversa da lista e o sistema abre a tela de CHAT, listando as mensagens que foram enviadas anteriormente, sendo os balões de cor azul utilizados para identificar as mensagens que o praticante de esporte enviou e os balões de cor cinza utilizados para identificar as mensagens que a outra pessoa participante da conversa enviou. O sistema fica disponível para o envio de novas mensagens.
4. Não é possível estabelecer uma conexão com o banco de dados: Durante a execução de qualquer passo, caso o não seja possível estabelecer uma conexão com o banco de dados, o sistema mostra a seguinte mensagem: “Erro desconhecido. Verifique sua conexão com a internet ou tente novamente mais tarde” e o caso de uso termina.

FIGURA 16 - TELA VISUALIZAÇÃO DE PERFIL COM BOTÃO PARA INICIAR UMA CONVERSA



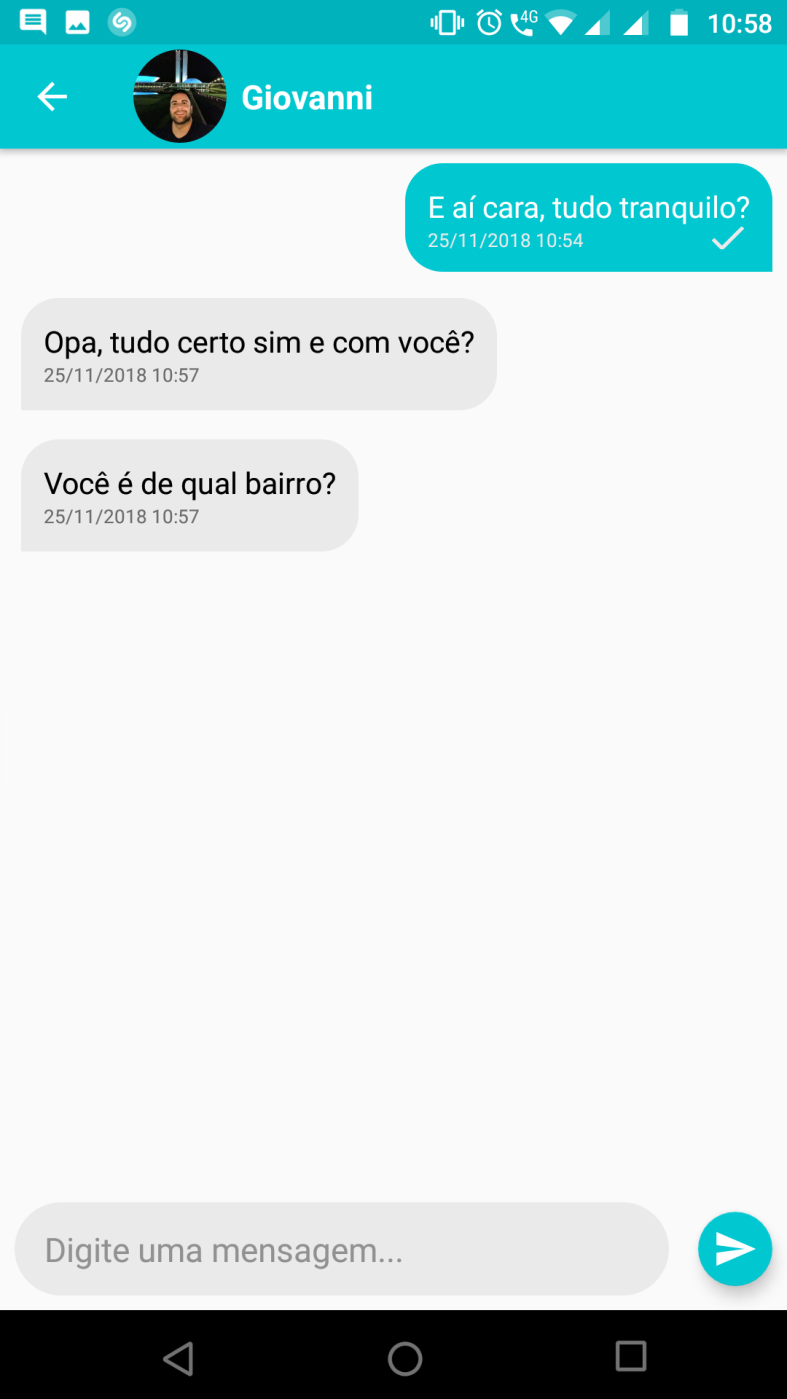
FONTE: O PRÓPRIO AUTOR

FIGURA 17 - TELA DE CHAT COM UMA CONVERSA INICIANDO



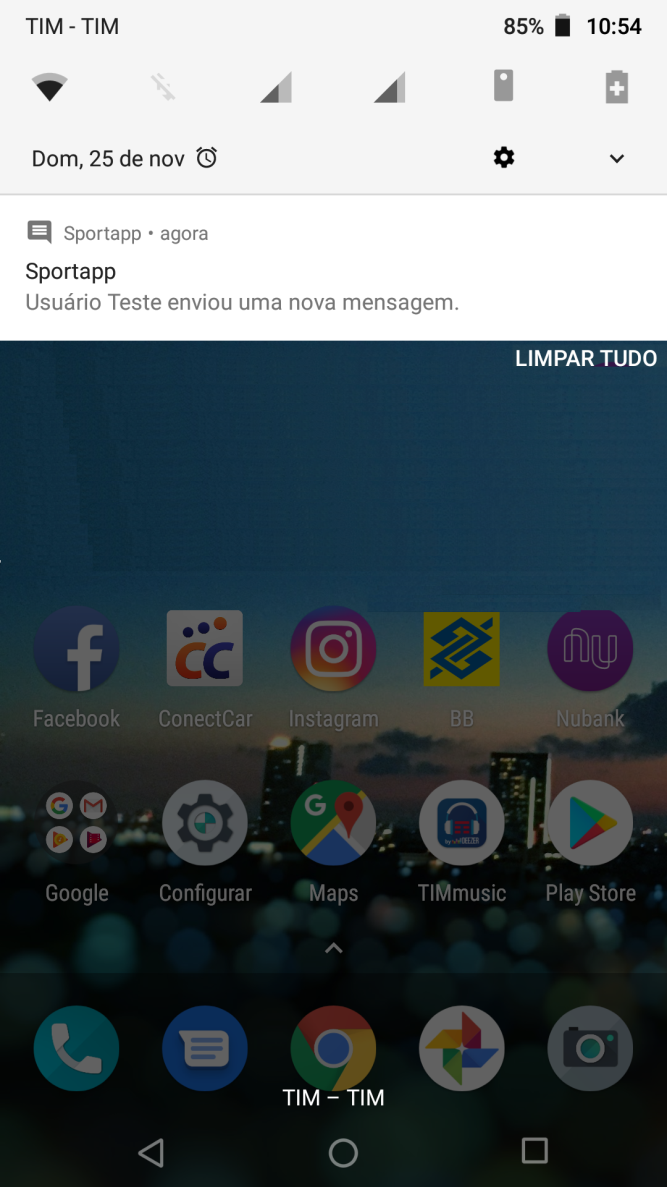
FONTE: O PRÓPRIO AUTOR

FIGURA 18 - TELA DE CHAT COM UMA CONVERSA ESTABELECIDA



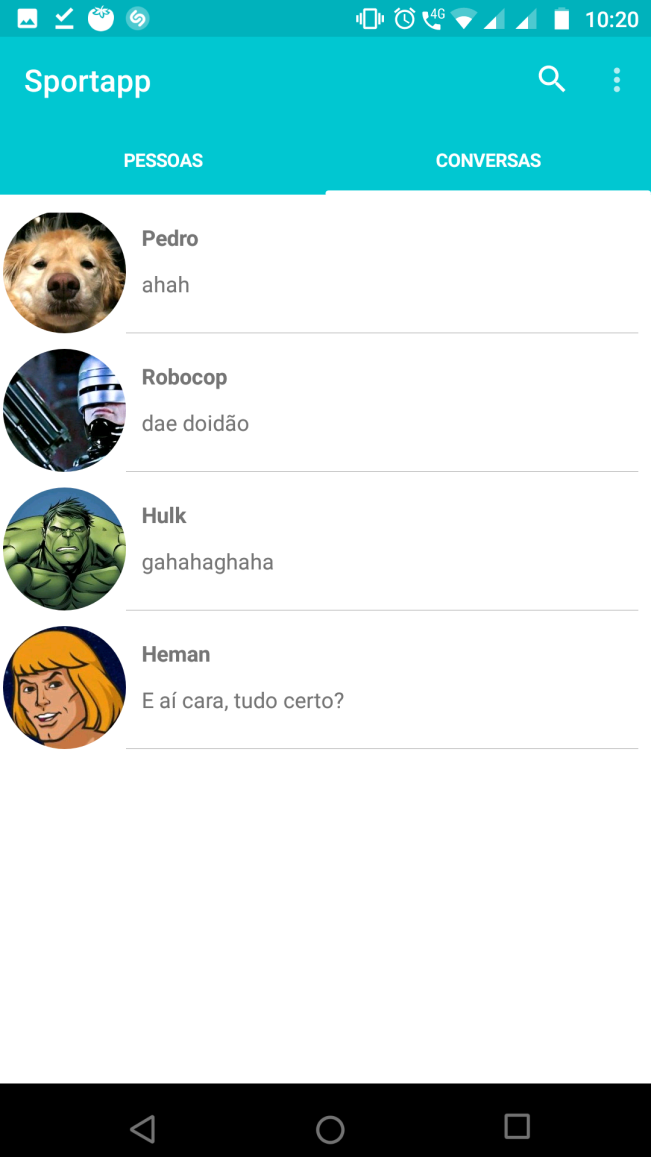
FONTE: O PRÓPRIO AUTOR

FIGURA 19 - NOTIFICAÇÃO DO ANDROID AVISANDO QUE ALGUÉM ENVIOU UMA MENSAGEM PARA O PRATICANTE DE ESPORTE AUTENTICADO



FONTE: O PRÓPRIO AUTOR

FIGURA 20 - APLICATIVO AUTENTICADO EXIBINDO A ABA DE CONVERSAS



FONTE: O PRÓPRIO AUTOR

## MANTER GRUPOS

O caso de uso permite que um praticante de esporte possa montar um grupo de pessoas para conversar. Um time ou dois times de futebol inteiros, por exemplo, podem estar dentro de um grupo e então permitir que estas pessoas conversem coletivamente para marcar jogos, horários e trocar ideias sobre seus objetivos.

### Pré-requisitos

Estar autenticado no sistema, ter outra pessoa praticante de esporte cadastrada no sistema além do próprio praticante de esporte autenticado, conexão com a internet.

### Fluxo básico

1. Criar grupo: O caso de uso se inicia quando o praticante de esporte toca no botão “Novo Grupo”. O sistema abre a tela para criação de grupo.
2. Preencher campo “Nome do grupo”. O praticante de esporte preenche o nome do grupo e o sistema mantém o dado pendente para inclusão.
3. Adicionar participantes: O praticante de esporte seleciona outros usuários para participarem do grupo, o sistema mantém os dados pendentes para inclusão.
4. Selecionar imagem do grupo: O praticante de esporte toca no botão para selecionar a imagem do grupo, o sistema abre a galeria de imagens do dispositivo para que uma foto seja selecionada. O praticante de esporte seleciona a imagem e o sistema mantém o dado pendente para inclusão.
5. Gravar grupo: O praticante de esportes clica no botão “Criar”, o sistema grava os dados no banco de dados, mostra a mensagem “Novo grupo de usuários criado.”, fecha a tela de criação de grupos e o caso de uso termina.

### Caminhos alternativos

1. Enviar mensagem em grupo: O praticante de esportes toca sobre um grupo na lista de conversas, o sistema abre a tela de CHAT do grupo. O praticante de esportes digita uma mensagem e toca no botão enviar. O sistema grava a mensagem no banco de dados e mostra o conteúdo da mensagem dentro de um balão azul na tela de CHAT. O sistema se mantém disponível para que o praticante de esportes possa enviar mais mensagens.
2. Receber mensagens: Ao abrir a tela de CHAT de um grupo, o sistema adiciona um LISTNER no banco de dados para receber em tempo real qualquer mensagem que outras pessoas do grupo enviem. Ao receber uma mensagem o sistema mostra o conteúdo desta dentro de um balão da cor cinza, contendo também o nome do usuário que enviou a mensagem, para que o praticante de esportes possa identificar que se trata de uma mensagem vinda da outra pessoa participante do grupo. O sistema se mantém disponível para envio de mais mensagens.
3. Adicionar mais pessoas no grupo: Durante a execução do passo “Enviar mensagem em grupo”, caso o praticante de esporte criador do grupo toque no botão “Adicionar pessoas” o sistema abre uma tela para a escolha de outras pessoas. O praticante de esporte seleciona as novas pessoas que irão participar do grupo e toca no botão confirmar. O sistema grava os dados no banco de dados, mostra a seguinte mensagem “Novos participantes adicionados ao grupo” e volta para a tela de conversa em grupo, ficando disponível para envio de mais mensagens.
4. Campo “Nome do grupo” não preenchido: Durante e execução do passo “Gravar grupo”, caso o campo “Nome do grupo” não esteja preenchido, o sistema mostra a seguinte mensagem “Nome do grupo não preenchido”, o caso de uso continua a execução a partir do passo “Preencher campo nome do grupo”.
5. Nenhum participante selecionado: Durante e execução do passo “Gravar grupo”, caso nenhum participante tenha sido selecionado, o sistema mostra a seguinte mensagem “Nenhum participante selecionado” e o caso de uso continua a partir do passo “Adicionar participantes”.
6. Não é possível estabelecer uma conexão com o banco de dados: Durante a execução de qualquer passo, caso o não seja possível estabelecer uma conexão com o banco de dados, o sistema mostra a seguinte mensagem: “Erro desconhecido. Verifique sua conexão com a internet ou tente novamente mais tarde” e o caso de uso termina.
7. Sair do grupo: Qualquer participante do grupo pode clicar no botão “sair do grupo” na tela de conversa em grupo. Ao clicar no botão “sair do grupo”, o sistema emite a seguinte pergunta “Confirma realmente sair do grupo?”, se o praticante esporte responder a pergunta de forma positiva, o sistema remove o praticante de esporte do grupo e o grupo não é mais exibido na lista de conversas. Caso o praticante de esporte for o criador do grupo, o sistema faz a exclusão do grupo no banco de dados, o grupo deixa de existir para todos os outros participantes e o caso de uso termina.

## NOTIFICAR PESSOAS

O caso de uso permite que as pessoas destinatárias das mensagens, recebam notificações que informam que outros usuários do sistema estão tentando se comunicar. Por exemplo, um praticante de esporte pode estar com o celular no bolso, neste momento o celular irá vibrar ou emitir algum som, dependendo da configuração de cada dispositivo, e este usuário perceberá que alguém está tentando conversar com ele por meio do Sportapp.

### Pré-requisitos

Praticante de esporte precisa estar autenticado para receber notificações de mensagem, conexão com a internet.

### Fluxo básico

1. Notificar pessoas: Durante a execução do caso de uso “Manter conversa”, quando um praticante de esporte envia uma mensagem, o sistema envia uma notificação para a pessoa destinatária da mensagem, desde que, essa pessoa destinatária não esteja com a própria tela de conversa aberta. O conteúdo da notificação tem o nome do usuário que enviou a mensagem e uma informação que uma nova mensagem chegou. O caso de uso termina.

### Fluxo alternativo

1. Durante a execução do passo “Notificar pessoas”, caso a notificação não possa ser entregue, por falta de conexão com a Internet da pessoa destinatária, por exemplo, o sistema mantém a notificação pendente no Firebase Cloud Message para que possa ser entregue quando o destinatário estiver disponível e o caso de uso acaba.

## ATUALIZAR TOKENS

Para que os usuários do sistema possam receber notificações, cada um deles precisa ter uma chave única identificada como TOKEN, que é um código único atribuído pelo Firebase ao dispositivo do usuário. Este código é atribuído ao dispositivo na primeira vez que o praticante de esporte abre o aplicativo, toda vez que este usuário apaga os dados do aplicativo no dispositivo ou toda vez que o Sportapp é desinstalado e instalado novamente. Toda vez que este código é atualizado, o sistema internamente recebe uma notificação por parte do Firebase que existe um novo código foi atribuído ao dispositivo e então o próprio aplicativo Sportapp faz o vínculo do novo TOKEN ao usuário, gravando no banco de dados. Este caso de uso mostra como este fluxo funciona.

### Pré-requisitos

Conexão com a internet.

### Fluxo básico

1. Receber novo TOKEN: Ao abrir o sistema e estando conectado com a internet, o Sportapp recebe internamente uma notificação que um TOKEN foi atribuído ao dispositivo. O sistema grava este código SHARED PREFERENCES do dispositivo, para que possa ser vinculado ao praticante de esporte quando este realizar sua autenticação, pois neste momento o sistema ainda não sabe qual praticante de esporte que vai se autenticar.
2. Gravar TOKEN no banco de dados: Durante a execução do caso de uso “Autenticar-se”, ao concluir a autenticação de um praticante de esporte, o sistema grava no banco de dados o TOKEN que estava pendente no SHARED PREFERENCES, vinculado ao praticante de esporte que foi autenticado. O praticante de esporte não recebe nenhum aviso que esta operação aconteceu, pois não se trata de uma mensagem de valor para o usuário final do sistema. O caso de uso termina.

### Caminhos alternativos

1. Praticante de esporte apagou os dados do Sportapp no dispositivo: Ao abrir o sistema novamente, o caso de uso continua a partir do passo “Receber novo TOKEN”.
2. Praticante de esporte desinstalou o sistema e reinstalou: Ao abrir o sistema novamente, o caso de uso continua a partir do passo “Receber novo TOKEN”.

# diagrama de classes

O diagrama de classes pode ser considerado um dos mais utilizados de toda a UML e um dos mais importantes. Este diagrama define e mostra de forma visual a estrutura das classes de todo o sistema, determinando os atributos, métodos, encapsulamento, heranças e como os objetos interagem entre si, por meio de relacionamentos. (GUEDES, 2011, P. 101)

Uma vez que o projeto utiliza o paradigma orientado a objetos, é neste diagrama que estarão representadas todas as classes do sistema, é a representação visual da estrutura do código fonte do sistema.

No diagrama, uma classe é representada um retângulo com três divisões, sendo delas:

* A primeira divisão contém o nome da classe
* A segunda divisão contém o nome dos atributos e o tipo de dado que este atributo armazena
* A terceira divisão contém o nome das funções existentes na classe.

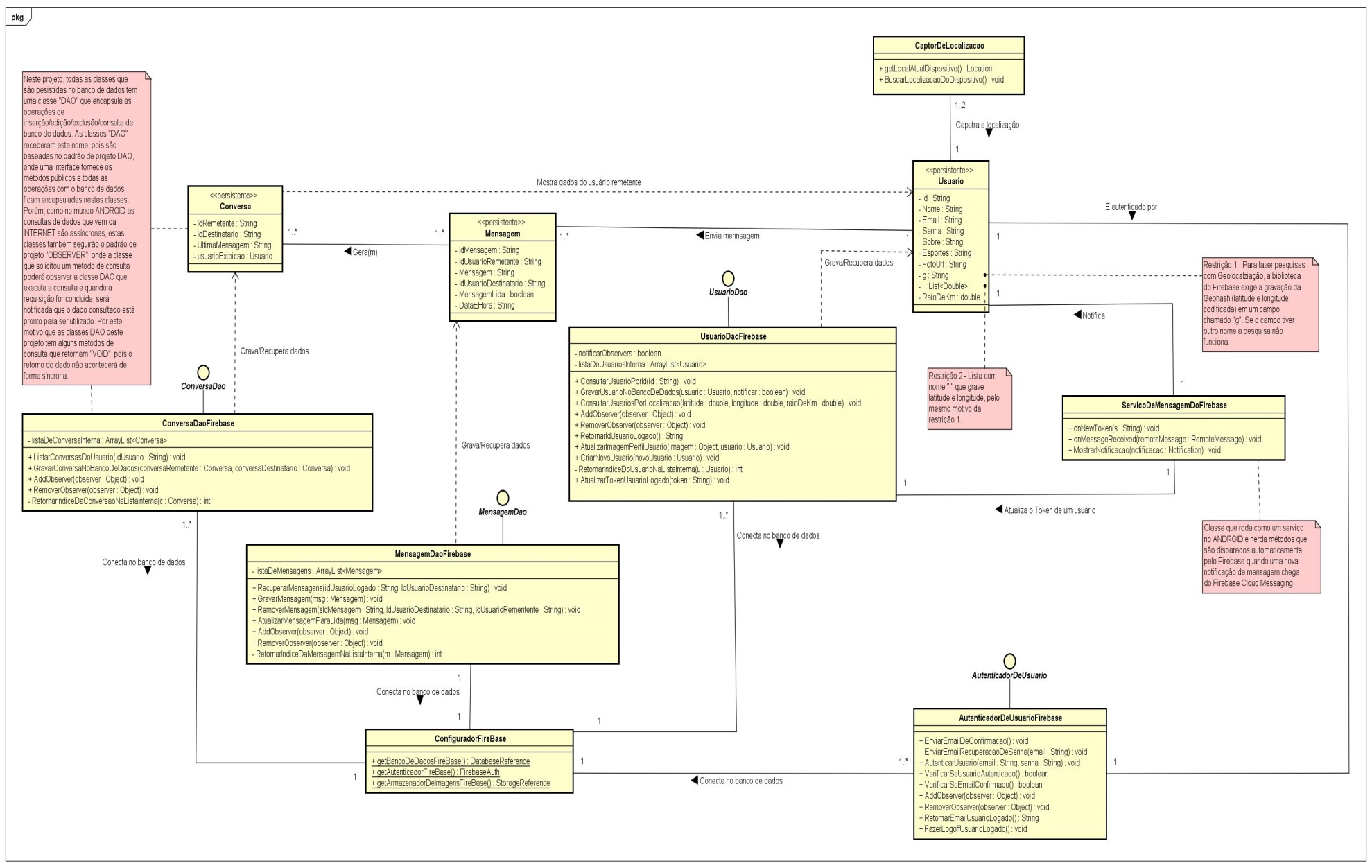
O símbolo “+” representa que o atributo ou método pode ser acessado por qualquer classe, o símbolo “-“, representa que o atributo ou método só pode ser acessado pela própria classe e o símbolo <<persistente>>, trata-se de um estereótipo que informa que a classe que contém este estereótipo é uma classe que seus atributos serão gravados no banco de dados. (GUEDES, 2011, p. 103, p 126)

As relações entre as classes são representadas por uma linha que liga uma classe a outra. Uma linha tracejada com uma seta apontando para uma classe, significa que estas classes tem uma relação de dependência, a seta aponta para a classe dependente. A linha contínua, sem tracejados ligando uma classe a outra, demonstra que uma classe se associa com outra, uma associação demonstra como os objetos se relacionam entre si durante a execução do programa. (GUEDES, 2011, p. 106). As associações também demonstram com quantas instâncias de um objeto as classes se relacionam, por meio da multiplicidade (GUEDES, 2011, p. 108), sendo as utilizadas neste projeto:

* 1..\* - Uma ou várias instâncias de uma classe se relacionam com “x” instâncias de outra classe.
* 1 – Apenas uma instância de uma classe se relaciona com “x” instâncias de outra classe.
* 1..2 – Uma ou duas instâncias de uma classe se relacionam com “x” instâncias de outra classe.

Além disso, é possível que cada associação tenha um nome, que representa de forma escrita como as classes se relacionam. Também se pode visualizar um símbolo representado por um triângulo escuro, que demonstra o sentido em que a relação deve ser lida. Outros símbolos utilizados no diagrama de classes do Sportapp foram o símbolo de interface, neste diagrama representado por um círculo ligado diretamente à classe, isso significa que esta classe precisa implementar algum serviço, no caso uma ou várias funções fornecidas pela interface, para que a classe tenha um conjunto de características fornecido pela interface. (GUEDES, 2011, p. 118), o último símbolo presente no diagrama de classes deste projeto, é o de notas explicativas, representado por um retângulo com uma ponta dobrada, como se fosse uma folha de papel com uma pequena dobra na parte superior esquerda. As notas explicativas foram utilizadas para demonstrar o porquê alguns atributos ou classes receberam determinados nomes. A figura 21 mostra o diagrama de classes do Sportapp, o diagrama foi inserido na orientação de paisagem devido ao seu tamanho, não cabendo na orientação de retrato.

FIGURA 21 - DIAGRAMA DE CLASSES DO SPORTAPP



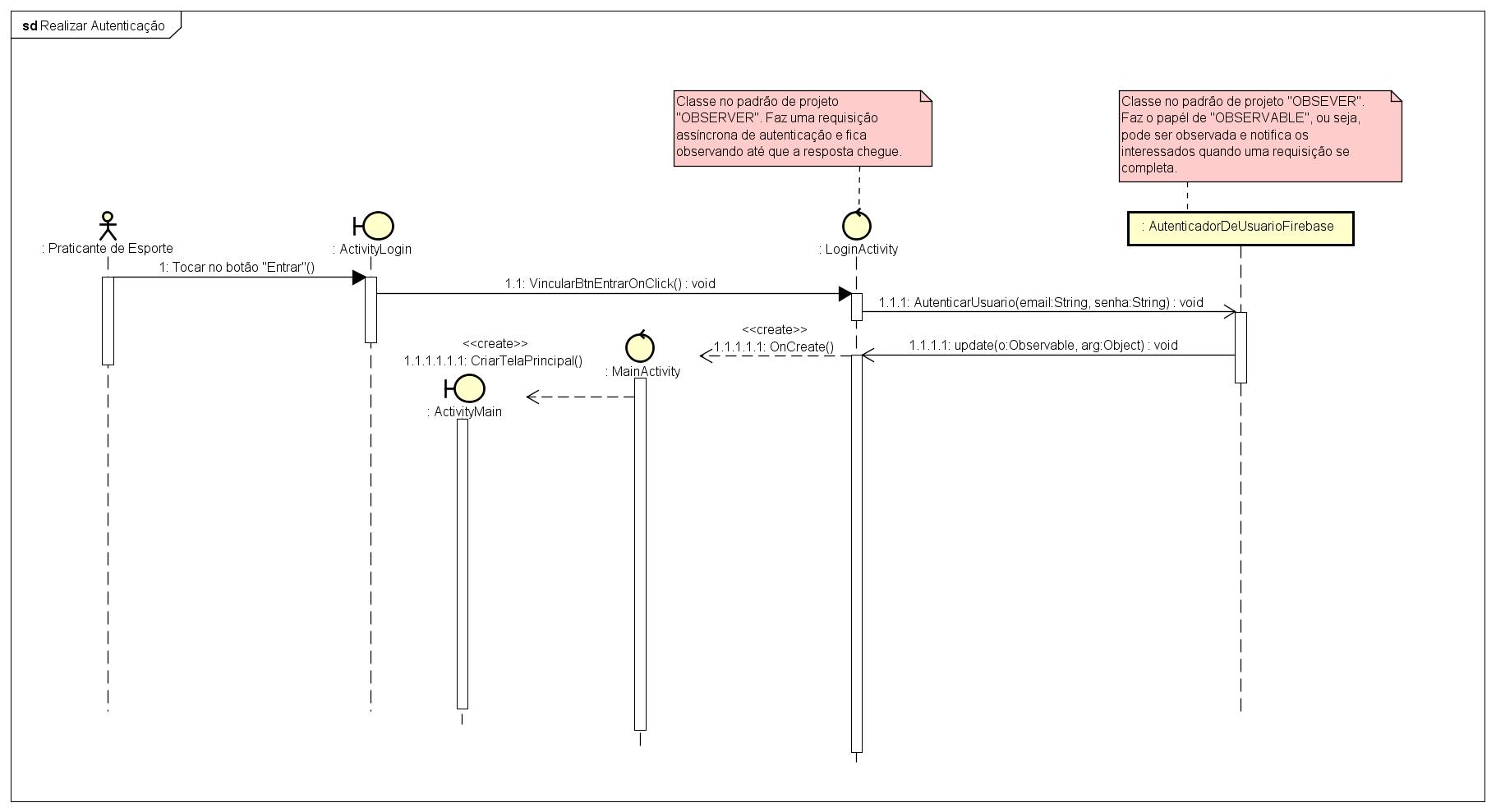
FONTE: O PRÓPRIO AUTOR

# diagrama de sequencia

O diagrama de sequencia trata-se de um diagrama comportamental que demonstra a sequencia de eventos que acontecem durante a execução de uma ação. Este diagrama é baseado no diagrama de casos de uso, onde pode existir um diagrama de sequencia para cada caso de uso existente no projeto. O diagrama de sequencia também depende do diagrama de classes, uma vez que ele mostra a ordem de chamada dos métodos contidos nas classes, para executar um caso de uso acionado por um ator, por exemplo. (GUEDES, 2011, p. 192),

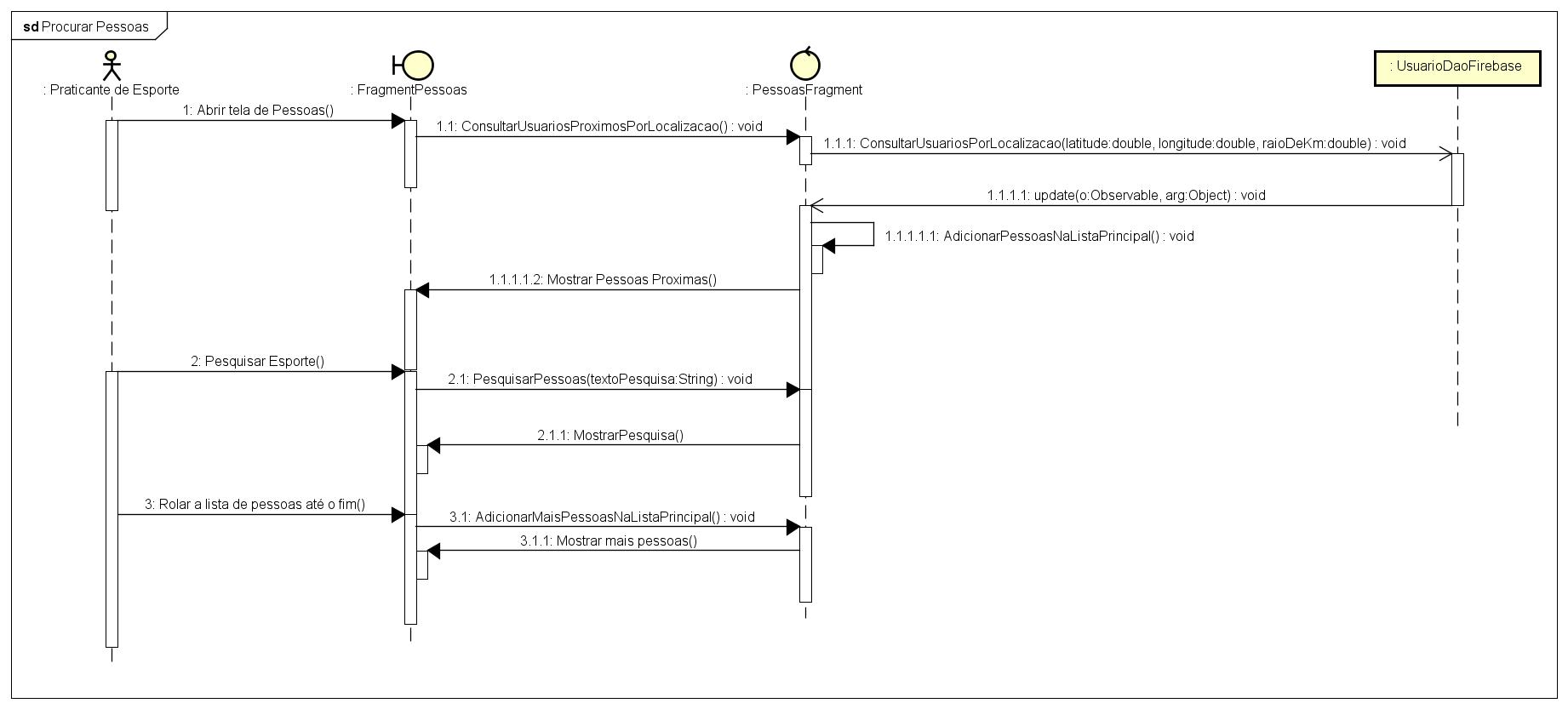
Neste projeto, foram selecionados os casos de uso considerados mais complexos do sistema para ser representado no diagrama de sequencia, sendo eles o caso de uso Autenticar-se, o caso de uso Manter conversas, o caso de uso Procurar Pessoas e o caso de uso Notificar pessoas, como pode ser visto nas figuras 22, 23, 24 e 25 exibidas a seguir.

FIGURA 22 - DIAGRAMA DE SEQUENCIA DO CASO DE USO AUTENTICAR-SE



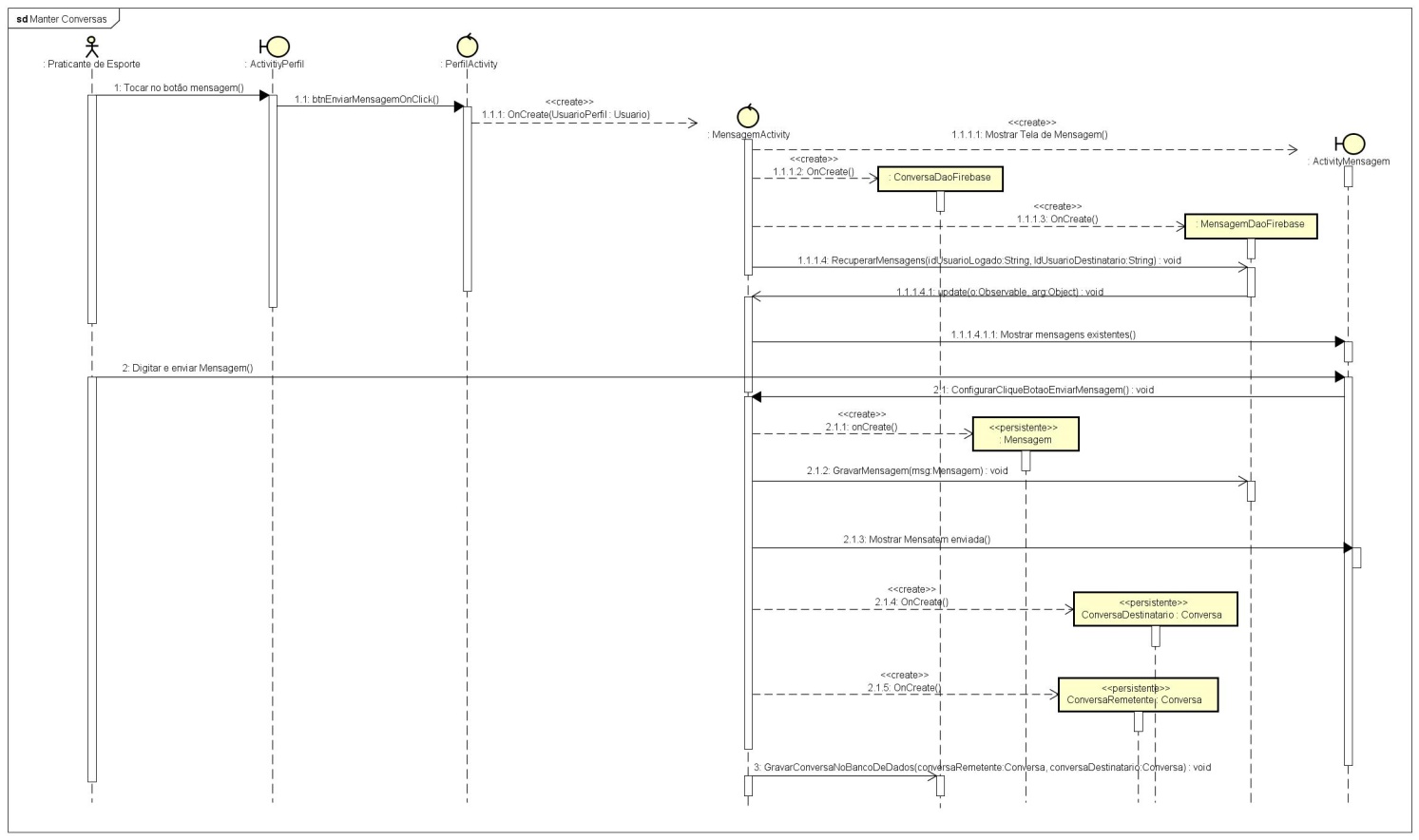
FONTE: O PRÓPRIO AUTOR

FIGURA 23 - DIAGRAMA DE SEQUENCIA DO CASO DE USO PROCURAR PESSOAS



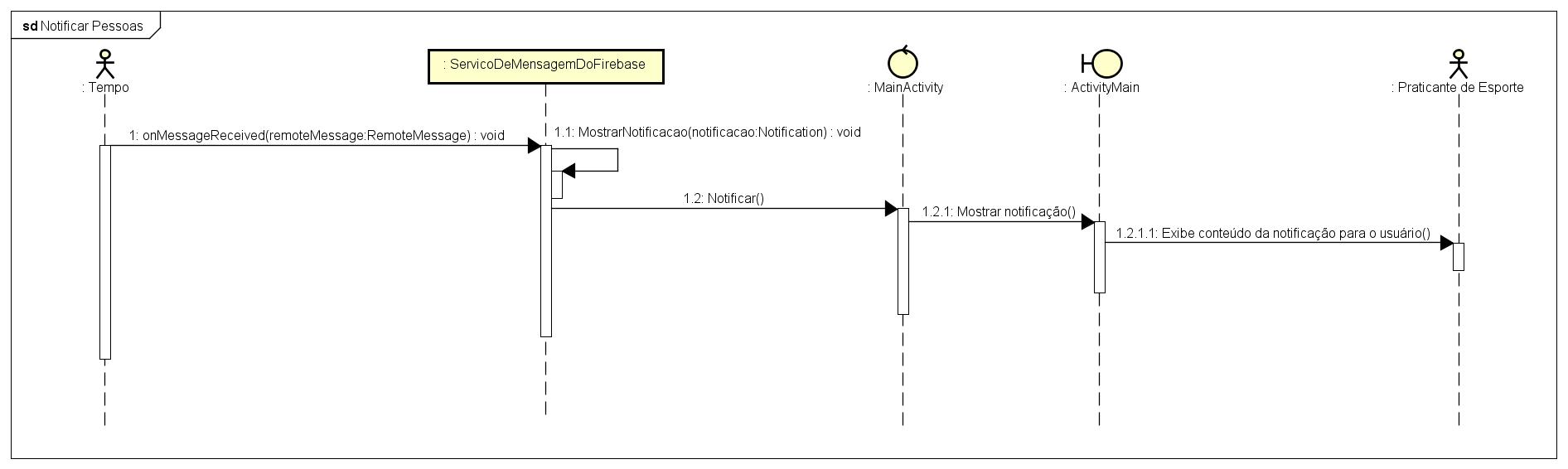
FONTE: O PRÓPRIO AUTOR

FIGURA 24 - DIAGRAMA DE SEQUENCIA DO CASO DE USO MANTER CONVERSAS



FONTE: O PRÓPRIO AUTOR

FIGURA 25 - DIAGRAMA DE SEQUENCIA DO CASO DE USO NOTIFICAR PESSOAS



FONTE: O PRÓPRIO AUTOR

# FIREBASE

Firebase é uma plataforma de desenvolvimento de software, desenvolvida pela Google, que permite a criação de aplicativos sem a necessidade de gerenciar a infraestrutura. É compatível com o Android, utiliza a infraestrutura do Google e é automaticamente dimensionado para atender a demanda de usuários. (FIREBASE INC, 2018)

A plataforma contém diversos planos de adesão, cada plano pode conter limitações técnicas e vantagens, dependendo do preço. Nessa pesquisa será utilizado o plano denominado Spark, que não tem custo financeiro, cumprindo um dos pré-requisitos do projeto. Este plano está sujeito a alguns limites técnicos, que serão detalhados a seguir, porém isso não impede o alcance do objetivo do aplicativo proposto na pesquisa.

## ArMAZENAMENTO DE DADOS

Um dos produtos do Firebase é o Realtime Database, que é um banco de dados NoSQL hospedado na nuvem (FIREBASE INC, 2018), com este produto é possível armazenar dados e deixa-los disponíveis para consulta o tempo todo, cumprindo mais um dos pré-requisitos do projeto, que é a disponibilidade em tempo integral.

No plano Spark, este armazenamento de dados é limitado a 1GB de texto (FIREBASE INC, 2018), porém como as imagens são armazenadas separadas em outro produto do Firebase, 1GB de texto é um número interessante para iniciar um aplicativo experimental, além disso, caso um aplicativo necessite de mais recursos com o passar do tempo, é possível fazer uma migração para um plano que contenha mais recursos mediante valores definidos pelo Firebase e que podem ser consultados na página oficial do produto. Outra limitação existente no plano utilizado no Sportapp é o limite de no máximo cem conexões simultâneas no banco de dados (FIREBASE INC, 2018), mais um limite que não é impeditivo para o objetivo da pesquisa e que pode ser resolvido fazendo uma migração de plano mediante necessidade.

O Firebase é um produto utilizado por empresas conhecidas mundialmente, como The New York Times, The Economist, Shazan e Alibaba.com (FIREBASE INC, 2018). O Realtime Database foi o banco de dados utilizado para o armazenamento de todos os dados do Sportapp, dados de usuários, conversas, entre outros.

## Armazentamento de imagens

O Cloud Storage, produto do Firebase para armazenamento em nuvem de imagens, áudios, vídeos e demais arquivos, foi utilizado na pesquisa para armazenar as fotos dos usuários do aplicativo, cada usuário poderá inserir sua própria foto em seu perfil do Sportapp para que ao encontrar e conversar com outras pessoas do aplicativo, possa ter a lembrança de que se trata de uma pessoa real. No plano Spark o limite de armazenamento de imagens será de 5GB, que também não é impeditivo para que o objetivo do projeto seja concluído.

## Autenticação

Para realizar a autenticação de todos os usuários do Sportapp foi utilizado o produto Firebase Authentication, que permite realizar a autenticação de usuários e é compatível com contas de E-mail, Facebook, GitHub, Google, Twiter, além de permitir autenticação com TOKENS enviados via SMS para o celular do usuário. Neste projeto o método de autenticação utilizado foi o de e-mail e senha, porém no futuro é possível expandir o aplicativo fazendo integração com outros métodos de autenticação. A própria biblioteca do Firebase fornece funcionalidades prontas para envio de e-mail de confirmação de cadastro e envio de e-mail para recuperação de senha, facilitando a vida do programador, fazendo com que não haja a necessidade de preocupação com este tipo de funcionalidade, que normalmente exige criação de interface separada para recuperar uma senha por exemplo. Outro ponto interessante deste produto, é que as senhas cadastradas pelos usuários nunca ficam visíveis, nem mesmo para o administrador ou criador do aplicativo, proporcionando uma maior segurança aos dados, pois mesmo uma pessoa que tenha acesso ao banco de dados, não saberia a senha dos usuários cadastrados.

## noticações

Um dos requisitos funcionais do projeto é o envio de autenticação aos usuários destinatários das mensagens enviadas via CHAT. Para atender este requisito foram selecionados dois produtos do Firebase que são eles:

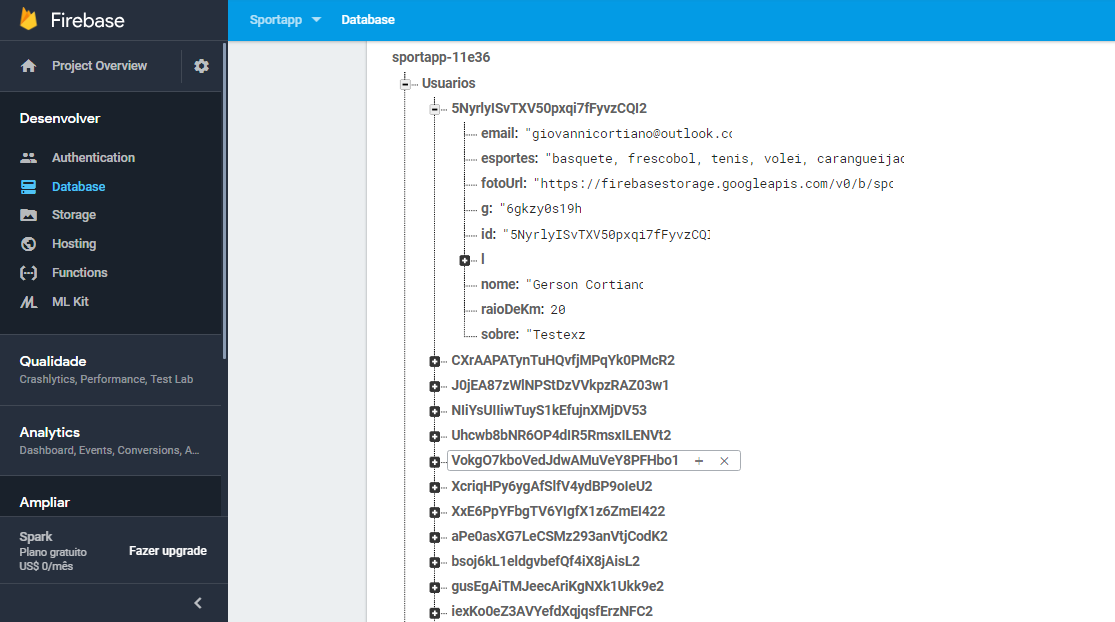
* Cloud Functions - Este produto permite programar funções que são disparadas e executadas dentro do servidor do Firebase, este produto pode ser utilizado para várias finalidades, como por exemplo, disparar a execução de uma função que verifica se em uma mensagem que chegou ao banco de dados existe palavras de baixo calão, pode ser utilizado para gravar a data e a hora do servidor em um dado que chegou ao banco de dados, pode também ser utilizado para apagar todos os dados de um usuário que cancelou sua conta, fazendo com que este processamento não fique no dispositivo do usuário final, existem vários exemplos de utilização na página oficial do produto.
* Cloud Messaging – Este produto permite que sejam enviadas notificações aos dispositivos dos usuários, é possível definir o conteúdo da notificação, o título da notificação, o destinatário da notificação que pode ser um único dispositivo, ou todos os dispositivos que estão cadastrados no projeto do Firebase, no caso o projeto Sportapp.

Nesta pesquisa, o produto Cloud Functions foi utilizado para disparar uma função programada toda vez que uma mensagem chega ao banco de dados do Sportapp. A função disparada captura quem é o remetente da mensagem e quem é o destinatário, a partir disso, é possível chamar o produto Cloud Messaging para enviar uma notificação para o destinatário informando que o remente da mensagem deseja se comunicar. Isso tudo acontece em tempo real, sem a necessidade de administração de nenhum tipo de servidor.

## utilização do firebase

Para utilizar o Firebase no desenvolvimento de um aplicativo, basta acessar a página oficial do produto, cadastrar-se, ou realizar a autenticação com uma conta do Google e então já estará disponível um painel para gerenciamento de todos os produtos. Todo produto tem um manual contendo todas as funcionalidades e recursos disponíveis, além de exemplos de utilização e exemplos e instruções de como fazer a integração do código fonte do sistema com o Firebase. As imagens a seguir mostram o painel de administração do projeto Sportapp, com dados fictícios cadastrados.

FIGURA 26 - TELA DE ADMINISTRAÇÃO DO BANCO DE DADOS DO SPORTAPP, BASEADO NA ESTRUTURA NOSQL



FONTE: O PRÓPRIO AUTOR

FIGURA 27 - TELA DE ADMINISTRAÇÃO DOS USUÁRIOS CADASTRADOS NO APLICATIVO

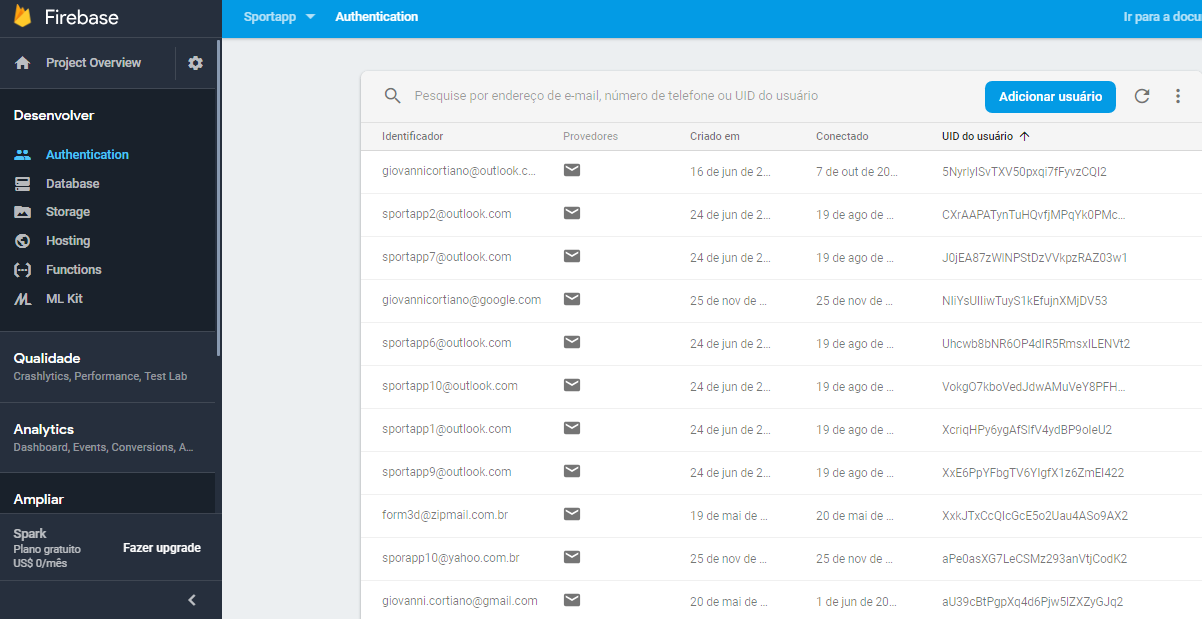
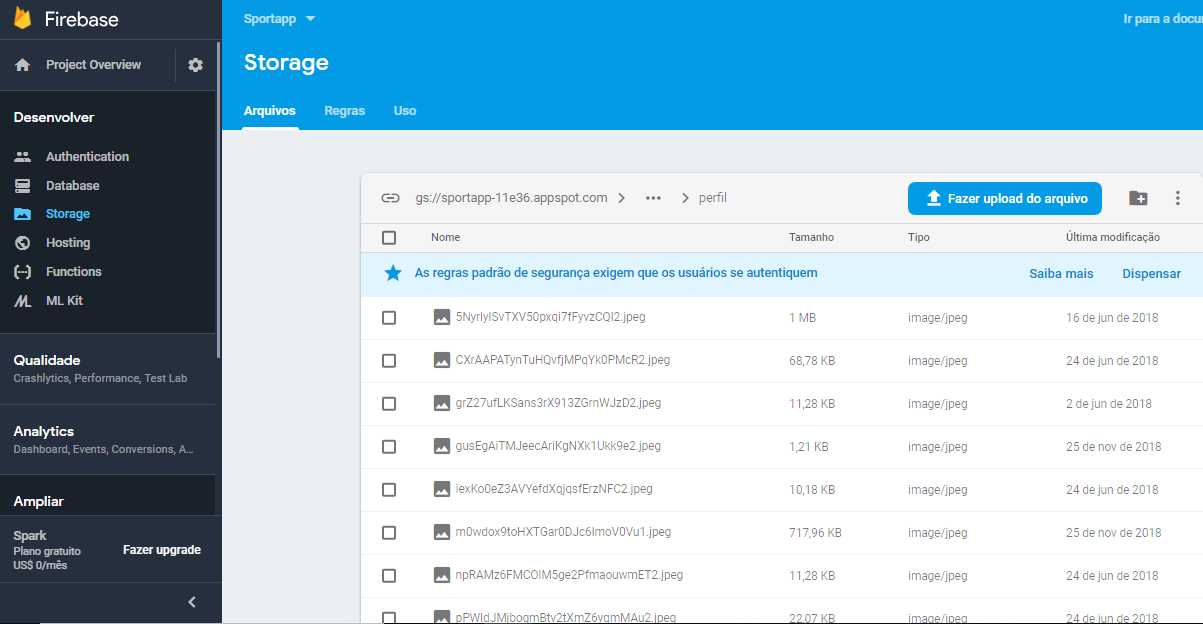


FIGURA 28 - TELA DE ADMINISTRAÇÃO DO ARMAZENAMENTO DE IMAGENS DO SPORTAPP



# ANDROID STUDIO

Android Studio é uma ferramenta do Google, disponibilizada gratuitamente, para a criação de aplicativos em todos os tipos de dispositivos com o sistema operacional Android. (GOOGLE LLC, 2018) Nesta ferramenta, uma das possibilidades é programar utilizando a linguagem de programação Java, por ser gratuito e permitir a criação de aplicativos para o sistema operacional Android, foi à ferramenta selecionada para realizar a codificação e a construção do aplicativo Sportapp.

# padrões de projeto

Os padrões de projeto identificam uma estrutura comum em um projeto, com a finalidade de solucionar um problema, ou definir um comportamento específico de algumas classes. Cada padrão de projeto tem um objetivo e segue um determinado padrão de estrutura, identificando as classes e as instancias participantes, definindo a responsabilidade e como cada classe interage com outra. (GAMMA, 2008 p. 20)

Neste projeto foram utilizados alguns padrões de projeto, dentre os mais variados existentes, para facilitar a codificação do sistema, o capítulo atual mostrará quais padrões foram utilizados e as suas respectivas definições.

## padrão de projeto observer

O objetivo deste padrão de projeto é definir uma dependência entre objetos, de uma forma que um objeto possa ser observado por outros objetos que precisam de alguma informação, por exemplo, e que quando este objeto observado tenha alguma mudança de estado, possa avisar ao objeto observador que houve esta mudança. (GAMMA, 2008 p. 274)

Neste projeto, todas as classes que fazem alguma consulta no banco de dados, seguem o padrão de projeto Observer, portanto essas classes permitem serem observadas por outras classes do projeto. Houve a necessidade de utilização deste padrão, pois no ambiente Android, toda consulta feita em dados armazenados na nuvem acontecem de maneira independente da execução do fluxo do sistema, portanto não é comum chamar uma funcionalidade que consulte os dados de um usuário, por exemplo, que estão armazenados na nuvem e que esta função retorne os dados imediatamente, vão existir fatores como a velocidade da Internet, por exemplo, que podem influenciar no tempo para essa consulta ser concluída, o aplicativo por sua vez continua sendo executado normalmente e processa de forma separada esta requisição de consulta de dados. Utilizando o padrão Observer, foi possível construir classes que tem a responsabilidade de consultar dados que estão armazenados na nuvem, porém como a execução do sistema não para e não se sabe quando a consulta dos dados terminará, a classe de consulta de dados avisa para as classes interessadas quando este dado estiver pronto para ser utilizado, por exemplo, uma classe solicita os dados do de um usuário para mostrar em tela, uma função de consulta é disparada dentro da classe que pode ser observada, a classe que solicitou a consulta passa a observar a classe que vai realizar a consulta e quando de fato por concluída essa requisição de dados, a classe que realizou a consulta avisa e passa os dados para a classe que solicitou, que por sua vez pode então mostrar os dados em tela para o usuário final.

## padrão de projeto dao

O padrão de projeto DAO – Data Acess Object, serve para separar classes que fazem as operações realizadas em bancos de dados, como inclusão, consulta inserção, exclusão, etc., das classes de regra de negócio, que são as classes que tem a inteligência do sistema, que fazem as tomadas de decisões conforme as ações vão acontecendo dentro do sistema. (LUCKOW, 2010, p. 177)

No projeto do Sportapp, este padrão de projeto foi aplicado com o intuito de separar as operações com o banco de dados, das classes de negócio, dessa forma, em todo o projeto, as únicas classes que conhecem e fazem operações no banco de dados, são as classes DAO. Neste projeto, cada classe DAO tem uma interface que fornece métodos que as classes DAO são obrigadas a implementarem, se no futuro fosse necessário trocar o banco de dados que o projeto utiliza, por exemplo, bastaria que novas classes DAO fossem criadas, implementando as mesmas interfaces, encapsulando o novo banco de dados escolhido e então, bastaria alterar o código fonte para instanciar as novas classes DAO e todo o restante do sistema continuaria funcionando normalmente, sem necessidade de alteração em outras classes, por este motivo, o padrão de projeto DAO foi aplicado neste projeto, ajudar na separação das responsabilidades das classes e consequentemente trazer vantagens para reutilização de código.

# conclusão

Durante a execução de todas as etapas desta pesquisa, foi possível aplicar diversos conhecimentos adquiridos ao longo do curso, além de ter que realizar várias pesquisas em livros ou em páginas oficiais das ferramentas utilizadas, para poder apresentar conceitos para a pesquisa e para sanar dúvidas que apareceram durante o transcorrer do projeto.

O objetivo principal do projeto foi atingido, pois foi possível construir um aplicativo, que tenha disponibilidade integral, que não gere custos financeiros e que de a possiblidade dos os usuários procurarem, encontrarem e conversarem com pessoas que tenham um gosto em comum, o esporte. O projeto ainda pode ser melhorado com várias funcionalidades que sejam interessantes para os usuários, como por exemplo, a criação de divulgação de locais para praticar esportes, a criação de autenticação por meio de redes sociais, a criação de competições entre times, porém como o tempo para conclusão da pesquisa não é grande, aliado as concorrências de atividades do cotidiano, são funcionalidades que podem ser incrementadas no futuro. Além disso, os testes realizados até o momento são com dados fictícios, não com usuários reais, caso houvesse tempo para testar com dados reais, publicando o aplicativo na loja do Google, isso daria uma visão se realmente o aplicativo pode ou não cair no gosto popular e ser útil para as pessoas.

Durante a execução da codificação, foram encontradas algumas limitações nas ferramentas do Firebase, como por exemplo, a dificuldade de fazer consultas paginadas, isso é, trazer um pouco dos dados para tela e caso o usuário precise ver mais dados, trazer outra carga de dados, fazendo com que o consumo de memória e de dados fique sempre baixo e seja utilizado somente o necessário. Para esta dificuldade foram criadas algumas lógicas dentro do próprio aplicativo, para consultar os dados aos poucos. Outra dificuldade encontrada na ferramenta foi que a consulta por geolocalização não permite mais filtros além da própria localização, essa pesquisa traz as pessoas que estão próximas do usuário de acordo com a localização do dispositivo, porém foi necessário trazer todas as pessoas próximas para a memória do dispositivo e então fazer os filtros por esporte diretamente na lista instanciada na memória do celular, isso faz com que dados sejam consumidos desnecessariamente, além de consumir mais memória por carregar dados que talvez não sejam utilizados pelo usuário, se existisse uma consulta por localização combinada com um filtro por outros dados da estrutura armazenada, seria o ideal para o caso dessa pesquisa.

É perceptível que apesar das limitações encontradas, o Firebase foi essencial para atingir o objetivo da disponibilidade integral do aplicativo e auxiliar no desenvolvimento de funcionalidades essenciais, como notificações, autenticação, entre outras. A escolha das ferramentas foi muito importante para cumprir o pré-requisito de um aplicativo funcional que não gere custo financeiro.

# referencias

LUCKOW, Décio Heinzelmann; MELO, Alexadre Altair. Programação Java Para a WEB. 1. ed. Novatec, 2010.

SINTES, Anthony. Aprenda Programação Orientada a Objetos em 21 dias. 1. ed. Makron Books, 2002.

GUEDES, Gilleanes T. A. UML 2 Uma abordagem prática 2. ed. Novatec, 2011.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. Padrões de Projeto Soluções reutilizáveis de software orientado a objetos. 1. ed. Artmed Editora, 2000.

FIREBASE - Produtos. Disponível em: < https://firebase.google.com/>

Acesso em: 25/11/2018

FIREBASE - Preços. Disponível em: < https://firebase.google.com/pricing/>

Acesso em: 25/11/2018

FIREBASE – Casos de uso. Disponível em: < https://firebase.google.com/ use-cases/>

Acesso em: 25/11/2018

CONSUMER BAROMETER WITH GOOGLE. Disponível em:

< https://www.consumerbarometer.com/en/>

Acesso em: 15/11/2018

ANDROID STUDIO. Disponível em: < https://developer.android.com/studio/?hl=pt-br>

Acesso em: 15/11/2018

# glossario

Firebase – Plataforma com diversas funcionalidades, para facilitar o desenvolvimento para aplicatvos móveis.

Android Studio – Ferramenta do Google para construção de aplicativos moveis para o sistema operacional Android.

Android – Sistema operacional encontrado em diversos dispositivos móveis.