

Desenvolvimento de um aplicativo para divulgação de eventos, estágios, residências e informações profissionais e acadêmicas

Mateus Padoin Brutti¹, Gustavo Stangherlin Cantarelli¹

¹Curso de Ciência da Computação – Universidade Franciscana (UFN)
CEP 97010-032 – Santa Maria – RS – Brasil

{mateuspbrutti, gus.cant}@gmail.com

Abstract. *This article presents a proposal for an application, for Android platform, with the purpose of disseminating academic and professional news. The methodology adopted for elaboration was Feature-Driven Development. The application will be developed in the Java programming language, making use of technologies such as Google Maps, Firebase Authentication, Cloud Storage and the data storage will be done through the database Firebase Database Realtime. In this way, the project the purpose of propagating such information, assisting students and professionals of diverse areas.*

Resumo. *Este artigo apresenta a proposta de um aplicativo, para plataforma Android, com a finalidade de disseminar e divulgar informações, notícias acadêmicas e profissionais. A metodologia adotada para a elaboração foi a Feature-Driven Development. A aplicação será desenvolvida na linguagem de programação Java, fazendo o uso de tecnologias como, Google Maps, Firebase Authentication, Cloud Storage e o armazenamento de dados será efetuado através do banco de dados Firebase Realtime Database. Desse modo, o projeto tem a finalidade de propagar tais informações, auxiliando estudantes e profissionais de diversas áreas.*

1. Introdução

São grandes as dificuldades encontradas por alunos, professores, e até mesmo profissionais, na procura de determinadas oportunidades, principalmente pela carência de um canal de comunicação específico e abrangente. No decorrer da pesquisa, o problema encontrado, relaciona-se ao fato de existir uma brecha entre a comunicação e os indivíduos.

Na maioria das vezes, a única forma que os indivíduos têm para o acesso de informações, tais como, eventos, vagas de emprego, residências e estágios, é por meio de indicações de pessoas que tem contato com o criador deles, deste modo a divulgação fica reduzida.

Com o passar dos anos os *smartphones* se tornam cada vez mais utilizados, pois há uma grande variedade de dispositivos que se encaixam a todos os tipos de público, assim, com um aplicativo, facilitando o acesso para tais informações. Segundo Munhoz, et al. (2018), a adoção de um aplicativo para os eventos pode ser uma forma de garantir que os participantes do mesmo estejam, efetivamente, recebendo aquilo que você comunica, sem ruídos ou imprecisões no alcance.

Junto ao curso de Física Médica, foi constatada a carência de aplicativos, ou tecnologias, que supram as necessidades por buscas de notícias e oportunidades em áreas específicas para a formação. Analisando os demais cursos que a Universidade

Franciscana dispõe, ficou evidente a deficiência que tais áreas sofrem pela falta de tecnologias voltadas para tal objetivo.

O presente trabalho tem como origem o Projeto PROBIT (Programa Institucional de Bolsas de Iniciação Científica Tecnológica) intitulado “Desenvolvimento de um Aplicativo para Profissionais de Física Médica”, juntamente com o Trabalho Final de Graduação (TFG) de autoria de Francine Kohls Shumacker, ambos do curso de Física Médica.

1.1. Objetivo Geral

O objetivo principal deste trabalho é desenvolver um aplicativo para a plataforma Android que seja capaz de proporcionar informações acadêmicas a um amplo número de pessoas, facilitando assim a comunicação e disseminação de propostas de estágios, eventos e informações profissionais.

1.2. Objetivos Específicos

Os objetivos específicos definem como o estudo e a pesquisa foram realizados para a elaboração deste projeto.

- Utilizar a metodologia FDD (*Feature Driven Development*) na aplicação do projeto.
- Utilizar a linguagem de programação Java para a implementação do aplicativo.
- Utilizar o banco de dados *Firebase Realtime Database* para armazenar os dados na nuvem.
- Utilizar o *Firebase Authentication* para validar os usuários.
- Utilizar o *Firebase Cloud Storage* para armazenar imagens representativas dos eventos.
- Implementar o projeto e com suas respectivas funcionalidades conforme os requisitos.
- Validar o aplicativo.

2. Referencial Teórico

Esta seção apresenta o referencial teórico do trabalho com a finalidade de fundamentar a proposta a partir de pesquisas relacionadas.

2.1. Informação e Disseminação

Disseminar informação supõe tornar público a produção de conhecimentos gerados ou organizados por uma instituição. A noção de disseminação é comumente interpretada como equivalente à de difusão, ou mesmo de divulgação [Lara 2003]. Segundo Lara (2003) a entrada e a saída de informação científica dos institutos de pesquisa dependem das relações estabelecidas com o ambiente em que atuam.

Para Leite (2016) os desenvolvimentos em tecnologias de informação e comunicação influenciam o próprio ciclo da informação, potencializando suas funções desde a geração até a utilização da informação. E esses avanços proporcionam novas possibilidades e oportunidades para processos informacionais.

2.2. Tecnologias

Esta seção apresentará as tecnologias computacionais necessárias para o desenvolvimento do aplicativo proposto.

2.2.1. Google Maps

O Google Maps é um produto que pertence a empresa Google, o mesmo, oferece mapas para o usuário se localizar. A API (*Application Programming Interface*) lida automaticamente com o acesso aos servidores do Google Maps, download de dados, exibição de mapas e resposta a gestos de mapas. Podendo usar chamadas de API para adicionar marcadores, polígonos e sobreposições a um mapa básico e alterar a visualização do usuário de uma área específica do mapa [Google Developers 2018].

Na aplicação proposta, o Google Maps tem como função auxiliar na localização, tanto no cadastro dos eventos como publicação, e os usuários podem visualizar o endereço em um fragmento de mapa.

2.2.2 Android

Segundo Pereira et al. (2009) “O Android é uma plataforma para tecnologia móvel completa, envolvendo um pacote com programas para smartphones, já com um sistema operacional, *middleware*, aplicativos e interface do usuário.

Seu objetivo principal é criar uma plataforma onde desenvolvedores possam inserir suas ideias e inovações resultando em um produto que realmente aprimora a experiência do usuário [Android Developers 2018].

Uma característica da plataforma Android é que não existe diferença entre os aplicativos integrados e os aplicativos criados com o SDK (*Software Development Kit*). Isso significa que se pode desenvolver aplicativos que acessam os recursos disponíveis no dispositivo [Ableson et al. 2012].

2.2.3 Firebase

O Firebase é um SGBD (Sistemas de Gestão de Base de Dados) que possui funções para a construção de aplicativos mobile e web através de ferramentas e infraestruturas que visam ajudar desenvolvedores a construir aplicativos visando a qualidade [Firebase Database 2019].

O Firebase contém alguns recursos que possibilitam ao desenvolvedor uma maior segurança e facilidade no desenvolvimento de um aplicativo, dentre eles são:

- *Firebase Realtime Database*: é um banco de dados hospedado na nuvem. Usa a sincronização de dados. Sempre que os dados são alterados, todos os dispositivos conectados recebem essa atualização em milissegundos [Firebase Database 2019].
- *Firebase Authentication*: fornece serviços de *back-end*, SDKs simples de usar e bibliotecas de IU prontas para autenticar usuários no seu aplicativo. Ele oferece suporte à autenticação por meio de senhas, números de telefone e provedores [Firebase Database 2019].
- *Firebase Cloud Storage*: Foi criado para armazenar dados gerados pelo usuário como fotos ou vídeos. Os *uploads* e *downloads* são feitos independentemente da qualidade da rede, o que significa que eles são reiniciados no ponto em que foram interrompidos, reduzindo tempo e largura de banda dos usuários [Cloud Storage, 2019].

2.3. Trabalhos correlatos

Nesta seção serão apresentados os trabalhos relacionados ao aplicativo proposto, os quais possuem algumas características semelhantes ao presente trabalho, com o objetivo de contribuir como base de conhecimento para a construção e organização desse projeto.

2.3.1. Aplicação Android direcionada para ONGs e projetos objetivando viabilizar a adoção de animais domésticos.

Nesse trabalho Souza et al. (2018) tem como objetivo o desenvolvimento de um software na plataforma Android visando facilitar o sistema de adoção de animais domésticos, mais especificadamente para cães e gatos, assim como, na divulgação de campanha para os órgãos responsáveis pelos bichos para arrecadação de recursos, sendo preferencialmente ONGs. Desse modo, auxiliando para a retirada dos animais que moram na rua, proporcionando também, um melhor bem-estar e minimizando o problema de saúde pública que se encontra hoje no Brasil.

Foi esperado como resultado a contribuição para a minimização da superlotação de cães e gatos que vivem nas ruas, através do crescimento de doções aguardado pelo aplicativo.

2.3.2. Aplicação Móvel para Divulgação de Campanhas, Eventos e Pedidos de Doações por Instituições do Terceiro Setor.

O trabalho realizado por Rossdeutscher et al. (2017) teve como objetivo elaborar uma aplicação móvel para proporcionar que instituições do terceiro setor possam realizar divulgações de campanhas, eventos e pedidos de doações, visando a arrecadação de doações e fundos.

Neste trabalho, o resultado esperado é de que, com a aplicação o número de doações cresça significativamente, e proporcionando a instituições do Terceiro Setor, um modelo próprio de captação de recursos.

2.3.3. Considerações sobre os trabalhos correlatos

Os dois trabalhos apresentados estão enfatizando a problemática do terceiro setor, fazendo o uso de tecnologias que contém no presente trabalho, no entanto, este se distingue dos demais por aplicar essas técnicas para auxiliar, primeiramente, o curso de Física Médica. O aplicativo proposto visa proporcionar a divulgação de eventos, residências e informações que contribuam na formação profissional.

3. Metodologia

Nesta seção é apresentada a metodologia utilizada para a elaboração do trabalho, com o intuito de mostrar os principais diagramas, fluxos, requisitos, e as principais funcionalidades do aplicativo.

3.1. *Feature-Driven Development (FDD)*

Agilidade tornou-se a palavra principal quando é descrito um moderno processo de desenvolvimento de software [Sommerville 2007].

FDD é uma metodologia ágil de desenvolvimento de software. Ela permite desenvolver sistemas de forma rápida, podendo facilmente ser adicionadas funcionalidades no decorrer (ou até mesmo depois de concluído) do desenvolvimento [Monteiro 2013]. Segundo Silva et al. (2009) são cinco os processos do FDD: “Desenvolver um Modelo Abrangente”, “Construir uma Lista de Funcionalidades”, “Planejar por Funcionalidade”, “Arquitetar por Funcionalidade” e “Construir por Funcionalidade”.

3.1.1. Desenvolver um Modelo Abrangente

Modelo Abrangente é responsável pelo estudo sobre o domínio do negócio e pela definição do escopo do projeto [Silva et. al 2009]. A visão geral do aplicativo pode ser observada através do Diagrama de Domínio, apresentado na Figura 1.

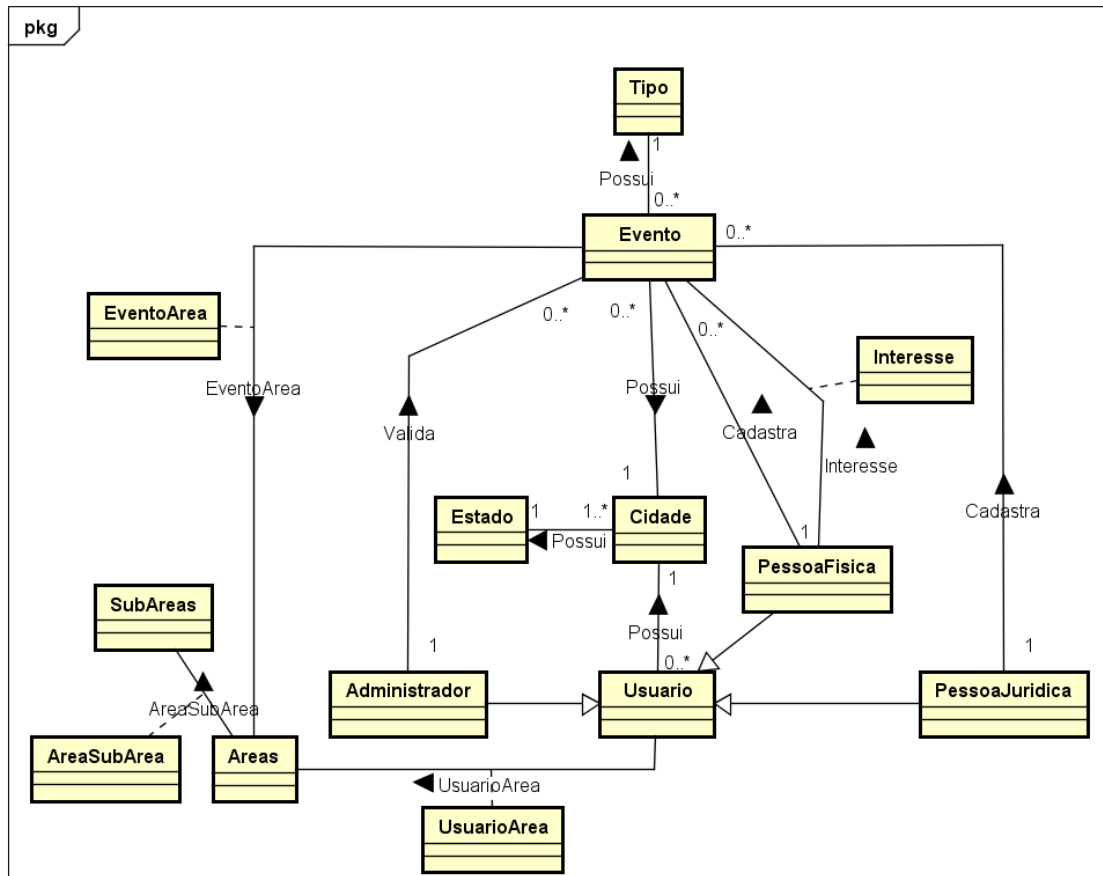


Figura 1. Diagrama de domínio

O diagrama apresentado na figura 1 é uma representação da estrutura de classes conceituais que servem de modelo para objetos. As classes *PessoaFisica*, *PessoaJuridica* e *Administrador*, são tipos de usuário, onde cada um contém sua cidade, estado, áreas e subáreas. A classe *Evento* representa todos os tipos de eventos que podem ser criados contendo seu tipo, cidade, estado, áreas e subáreas. O usuário do tipo *Pessoa Física* pode demonstrar o interesse em um ou mais eventos ficando na lista de interesse. A *Pessoa Física* e *Pessoa Jurídica* podem efetuar o cadastro de evento se desejarem, já o *Administrador* é quem fará a validação deste evento.

3.1.2. Construção da lista de funcionalidades

O objetivo dessa etapa é identificar as funcionalidades agrupadas em conjunto que satisfarão aos requisitos [Silva 2018]. Apresenta-se a Tabela 1 de Requisitos Funcionais (RF).

Tabela1. Requisitos funcionais do sistema

Requisitos Funcionais		
Funcionalidade	Descrição	Complexidade
RF01: Gerenciar usuário	Deve permitir as funções de cadastrar, buscar, alterar e excluir usuário.	Baixo
RF02: Cadastrar tipo evento	O aplicativo deve permitir o cadastro do evento, contemplando suas características.	Baixo
RF02.1: Escolher cidade do evento	O aplicativo deve possibilitar escolher a cidade na qual ocorrerá o evento fazendo o uso do Google Maps	Médio
RF02.2: Validar evento	O administrador deve validar ou invalidar um evento.	Baixo
RF02.3: Informar aprovação/reprovação	O aplicativo deve enviar um e-mail para o usuário informando a aprovação ou reprovação do cadastro do evento	Médio
RF03: Gerenciar evento	O aplicativo deve permitir as funções de cadastrar, consultar, alterar e excluir eventos de acordo com o tipo de usuário.	Baixo
RF04: Exibir eventos	O aplicativo deve exibir a lista de eventos cadastrados.	Baixo
RF04.1: Apresentar eventos conforme filtros	O aplicativo deve exibir os eventos cuja a cidade ou estado seja a mesma do filtro utilizado.	Médio
RF04.2: Exibir detalhes do evento	O aplicativo deve exibir as informações do evento e os contatos do responsável.	Baixo
RF04.3: Demonstrar interesse no evento	O aplicativo deve permitir o usuário demonstrar interesse no evento.	Médio
RF04.3.1: Notificar usuário do interesse	O aplicativo deverá informar o interesse ao usuário principal	Médio
RF05: Visualizar lista de interessados	Os usuários poderão visualizar a lista de interessados referente ao(s) evento(s) cadastrado(s) por ele	Médio

Apresenta-se na Tabela 2 os Requisitos Não Funcionais do sistema (RNF):

Tabela2. Requisitos não funcionais do sistema

Requisitos Não Funcionais	
Funcionalidade	Descrição
RNF01	O aplicativo deverá ser desenvolvido para plataforma Android.
RNF02	O aplicativo deverá ser desenvolvido na linguagem de programação Java.
RNF03	Os dados do software deverão ser armazenados na nuvem, através do banco de dados Firebase Realtime Database.
RNF04	Para o armazenamento das imagens dos eventos será feito o uso do <i>Firebase Cloud Storage</i> .

3.1.3. Planejar por Funcionalidade

Esta fase do FDD encontra-se detalhada no APÊNDICE E.

3.1.4. Arquitetar por Funcionalidade

Segundo Silva (2009), no processo Arquitetar por Funcionalidade, para cada funcionalidade da lista é definida uma atividade a ser realizada. Neste processo o protótipo da interface do usuário, juntamente com os diagramas de atividades foram gerados.

O fluxo geral do aplicativo e as principais funcionalidades na visão do usuário podem ser observadas através do Diagrama de Atividade e seus respectivos sub processos no Apêndice D.

O subprocesso Realizar Login, contempla o processo de cadastro do usuário, login e validação. Um ponto importante nesse subprocesso é a necessidade de validação por tipo de Pessoa.

O subprocesso Cadastrar Evento (APÊNDICE D), apresenta as atividades necessárias para o cadastro de um evento no aplicativo. O usuário deverá preencher as informações do evento e os dados de contato que ficarão visíveis ao público. Cada evento cadastrado será atribuído a uma cidade, para que, em seguida, a visualização dos eventos disponíveis seja definida por estado.

O subprocesso Visualizar Eventos (APÊNDICE D), apresenta as atividades necessárias para a visualização dos eventos disponíveis. De modo padrão, quando o usuário acessar o menu de visualização dos eventos, o sistema exibira a lista com todos eventos. Apesar disso, o aplicativo permitirá incluir parâmetros no filtro de visualização, para que seja possível visualizar eventos de outras localidades. Para visualizar mais detalhes sobre o evento, o usuário poderá clicar na imagem do evento e, assim, o sistema exibirá o cadastro completo do evento.

Nas Figura 2, é apresentado o diagrama de pacotes principal, contendo as classes que foram implementadas, tendo seus métodos e atributos. Os diagramas de classes contidas nos pacotes, estão exibidas no apêndice G.

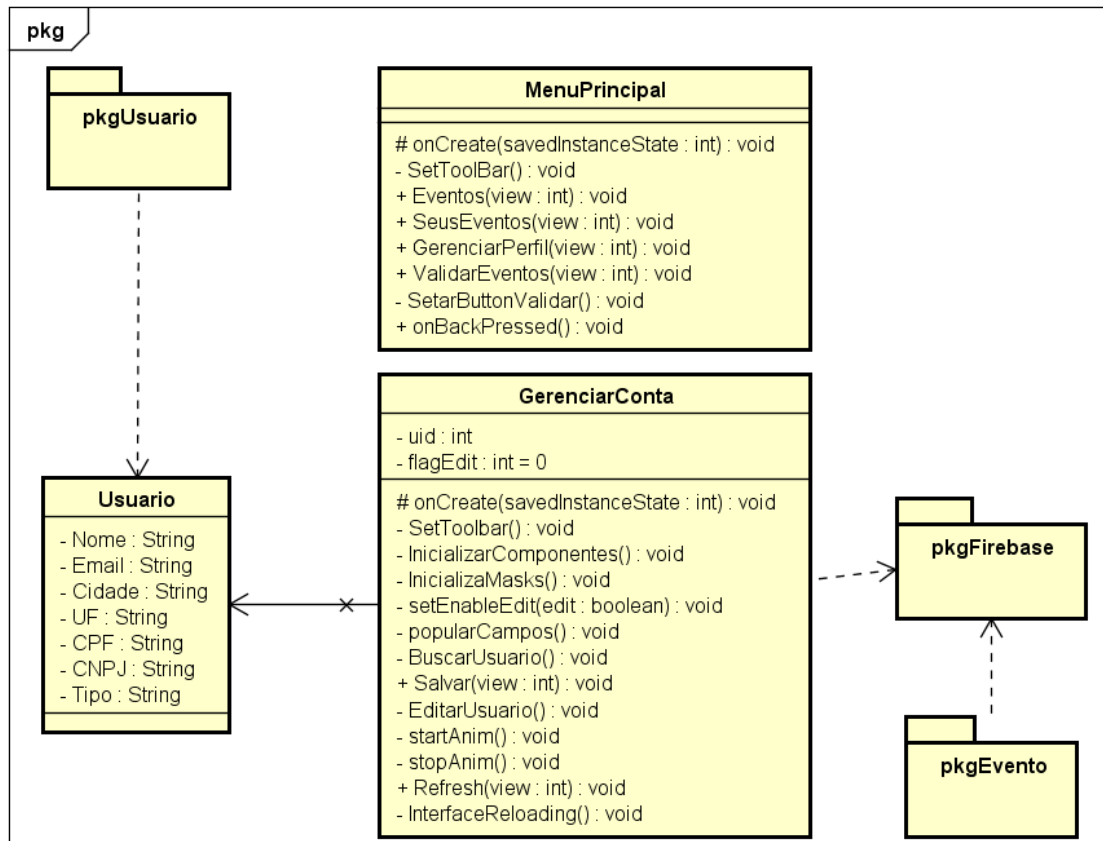


Figura 2. Diagrama de Pacotes

4. Construir por Funcionalidade

A construção por funcionalidade do software, é a quinta etapa da metodologia ágil FDD, onde foram elaboradas as codificações, fazendo o uso das tecnologias propostas no projeto.

Para a autenticação do usuário foi utilizado o *Firebase Authentication* por e-mail e senha, por meio do *SignInWithEmailAndPassword*, como mostra a Figura 3.

```

1 package com.example.mateu.informe_se;
2
3 import ...
4
28 public class Login extends AppCompatActivity {
29     auth.signInWithEmailAndPassword(email, password)
30     .addOnCompleteListener( activity: this, (task) -> {
31         if (task.isSuccessful()) {
32             Intent it = new Intent(getApplicationContext(), MenuPrincipal.class);
33             startActivity(it);
34             etSenha.setText(null);
35         } else {
36             String errorCode = ((FirebaseAuthException) task.getException()).getErrorCode();
37         }
38     });
39 }
40 
```

Figura 3. Trecho de código da autenticação do usuário

Um dos processos com maior relevância para a aplicação é o Cadastro de Eventos, onde os usuários poderão cadastrar seus eventos, fazendo o uso da API do Google Maps. E, dessa forma, obter uma maior facilidade em encontrar a localização exata, reduzindo tempo no preenchimento de local, assim como escolher uma imagem referente ao evento e, na sequência, enviar para análise de um administrador, representado pela Figura 4.


```

1 package com.example.mateu.informe_se.Eventos;
3 import ...
116 public class CadastrarEvento extends FragmentActivity implements OnMapReadyCallback, LocationListener {
561     if (validarDadosCadastro()) {
562         Evento evento = MontaEvento();
563         try{
564             String idEvento = ref.child("EVENTO").push().getKey();
565             evento.setIdEvento(idEvento);
566             ref.child("EVENTO").child(""+idEvento).setValue(evento);
567
568             Bitmap nomImage = BitmapFactory.decodeResource(getResources(),
569                 R.drawable.noimage);
570
571             if(img != null && img != nomImage){
572                 Bitmap bitmap = ((BitmapDrawable) ivImagemCard.getDrawable()).getBitmap();
573                 ByteArrayOutputStream baos = new ByteArrayOutputStream();
574                 bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, baos);
575                 byte[] data = baos.toByteArray();
576
577                 StorageReference storageRef = FirestoreStorage.getInstance().getReference();
578                 storageRef.child("FotoEvento/"+idEvento).putBytes(data);
579             }
580
581             _Evento = evento;
582             SetBarStatus();
583             setEnableButtons();

```

Figura 4. Trecho de código de Cadastro do Evento

Não menos importante, o procedimento de Listar Eventos faz o uso de um dos eventos disponibilizados pelo *Firestore Realtime Database*, o *ValueEventListener*, que, a partir da referência do banco de dados, é o responsável por listar os eventos, e cuidar quando houver algum tipo de modificação na tabela requerida pela referência, para atualizar os Eventos listados, tendo ou não filtros de busca, como determina a Figura 5.

```

1 package com.example.mateu.informe se.Eventos;
3 import ...
41 public class ListarEventos extends AppCompatActivity implements ChildEventListener {
151     eventTodos = new ValueEventListener() {
152         @Override
153         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
154             eventos = new ArrayList<>();
155             for (DataSnapshot objSnapshot: dataSnapshot.getChildren()){
156                 Evento event = objSnapshot.getValue(Evento.class);
157                 eventos.add(event);
158             }
159
160             Collections.sort (eventos, (Comparator) (evento1, evento2) -> {
161                 int tipoCompare = evento1.getTipo().compareToIgnoreCase(evento2.getTipo());
162                 if(tipoCompare != 0){
163                     return tipoCompare;
164                 }
165                 return evento1.getArea().compareToIgnoreCase(evento2.getArea());
166             });
167
168             eventoAdapter = new EventoAdapter( context: ListarEventos.this, eventos);
169             listView.setAdapter (eventoAdapter);
170             query.removeEventListener(eventTodos);
171
172             if(eventos.size() == 0){
173                 tvNEncontrado.setVisibility(View.VISIBLE);
174             }else {
175                 tvNEncontrado.setVisibility(View.GONE);
176             }
177
178             stopAnim();
179
180         }
181     }

```

Figura 5. Trecho de código Listar Eventos

O banco de dados usado é o *Firestore Realtime Database*, NoSQL, hospedado na nuvem e armazenado em um arquivo de formato JASON [Firestore Database 2019]. A estrutura a qual o software se aplica é representada na Figura 6.



Figura 6. Estrutura do banco de dados em Firestore Realtime Database

5. Resultados

O propósito do atual trabalho conteve a construção de um aplicativo para Smartphones com sistema operacional Android, no qual, oferece informações acadêmicas e profissionais, tendo em vista sua disseminação.

Na validação é feito uma série de diferentes testes, com o propósito de verificar a conformidade do software em relação aos requisitos. Os testes e a validação do software proposto, foram feitos junto ao curso de Física Médica em conjunto com o projeto PROBIT.

Primeiramente, foram feitos testes de usabilidade, inserindo alguns dados, para checar se o software, realmente faz o que deveria fazer. Após a certificação dos testes, o aplicativo foi disponibilizado para dois integrantes do projeto PROBIT para validação. Segundo a coordenadora do projeto, Ana Paula Schwarz.

O app informe-se ficou um aplicativo bonito e claro. Ficou fácil de se localizar nele, tanto para quem procura por eventos como para quem deseja cadastrar eventos. Gostei bastante na facilidade de cadastrar eventos, ficando os eventos cadastrados com os dados mais pertinentes na primeira tela e se caso me interessar pelo evento posso ter mais informações como a localização via *Google Maps* e acesso ao link do evento.

A segunda integrante do projeto PROBIT e, também, aluna do curso de Física Médica, Francine Kohls Schumacker, relatou:

O Informe-se é um aplicativo com manuseio fácil, rápido e intuitivo, com telas fáceis e práticas, tanto para inserir os eventos, quanto para visualizar os mesmos. Encontrei uma maneira bastante viável de se conectar aos eventos explorando essa ferramenta.

Devido à grande quantidade de interfaces elaboradas no aplicativo, serão exibidas apenas as consideradas mais importantes.

A primeira interface é destinada para a realização do *login* (Figura 7.1). Após a autenticação *login* o usuário é direcionado para o menu principal (Figura 7.2). Selecionado o botão EVENTOS, é mostrada a interface que lista todos os eventos (Figura 7.3). Por padrão, o aplicativo exhibe a lista de eventos ordenados por tipo e área. O usuário também pode adotar filtros de estado e ou cidade para refinar sua pesquisa.



Figura 7. Interfaces de login, menu principal e listar eventos

No momento em que é clicado em um evento listado (Figura 7.3), os detalhes do mesmo são apresentados, conforme mostra a Figura 8. O usuário pode interagir com o fragmento do *Google Maps*, onde é mostrado o local em que o evento ocorrerá, e demais estabelecimentos da proximidade, como hotéis e restaurantes. Assim como, o usuário pode demonstrar o interesse no evento.



Figura 8. Interface dos detalhes do evento

A Figura 9, exibe a interface de cadastro de um evento. Nela é possível escolher uma imagem, através da galeria do smartphone do usuário, ou por um link da internet. Para a localização do evento, o usuário pode pesquisar pela cidade, estado, cep e demais tipos. Ao clicar no mapa é adicionado um marcador sobre o local, e os campos local, cidade, estado e uf são preenchidos automaticamente através da tecnologia *GeoLocate*, disponibilizado pela *Google*. Se os dados preenchidos não estiverem de acordo, pode ser feita a alteração manualmente para a correção.

The image displays three sequential screenshots of a mobile application's event registration interface. The top bar is green with a white back arrow and the text 'CADASTRAR EVENTO'. The first screenshot shows a gallery icon for image selection and a link input field. The second screenshot shows a dropdown menu for 'ÁREA', a search bar for 'SubArea', and a Google map with a location marker. The third screenshot shows a form with fields for 'Local', 'Cidade', 'Estado', and 'UF', and a blue 'CADASTRAR' button at the bottom.

Figura 9. Interface do cadastro de evento

Este trabalho teve duas publicações como resumo, A primeira no XXII Simpósio de Ensino, Pesquisa e Extensão - SEPE no ano de 2018. A segunda publicação foi efetuada no IX Salão de Iniciação Científica – SIC no ano de 2019.

6. Problemas Encontrados

Ao longo do desenvolvimento foram encontrados problemas para utilizar algumas das tecnologias usadas. Foram encontradas dificuldades na implementação da codificação, para a comunicação com o banco de dados *Firestore Realtime Database*, devido sua documentação não estar devidamente atualizada. Um dos motivos é a grande mudança e atualização de versões assim como a alteração da referência que é utilizada para efetuar as requisições ao banco de dados.

7. Conclusão

O estudo a partir de pesquisas relacionadas ao tema, apresentou o desenvolvimento de um aplicativo *mobile*, para a plataforma Android, destinado a disseminar e divulgar determinadas informações. O sistema contempla principalmente estudantes e profissionais. Com o uso do aplicativo, aguarda-se que ambos estejam cientes, de eventos gerais ou de áreas específicas de seu interesse, que ocorrem ao seu redor.

Para o desenvolvimento desse projeto, foi utilizada a metodologia ágil FDD, a qual, orientou o desenvolvimento desde o escopo inicial, até a implementação dos

requisitos tratados por ser compatível com a construção do presente trabalho. Por meio dela foram apresentados os principais diagramas da metodologia, com a finalidade de mostrar os requisitos, as principais funcionalidades do aplicativo e a arquitetura. Vale evidenciar, a grande importância do uso do banco de dados *Firebase Realtime Database*, que possibilitou uma transação de dados com mais facilidade e rapidez, colaborando positivamente, para a codificação.

A partir dos feedbacks de validação, disponibilizados por dois integrantes do projeto PROBIT, foi possível detectar, que o aplicativo pode suprir as necessidades do público alvo e contribuir para as dificuldades atribuídas no presente trabalho.

Para trabalhos futuros, sugere-se o desenvolvimento de funcionalidades adicionais, onde pode ser criado um sistema que propicia a criação de caravanas, oferecendo mobilidade há eventos distantes, proporcionando ainda, um auxílio a mais para o evento. Assim como desenvolver o aplicativo para a plataforma IOS da empresa Apple, devido grande aquisição de smartphones. Assim, abrangendo um maior público, e tornando o aplicativo multiplataforma.

8. Referências Bibliográficas

- Ableson, Frank; King, Chris; SEN, Robi. (2012) “Android em ação”. Elsevier Brasil.
- Android Developers. (2018) “Introdução ao Android”, <https://developer.android.com/guide/index.html>. Acesso em: outubro de 2018.
- Cloud Storage. (2019) “Cloud Storage”, <https://firebase.google.com/docs/storage>. Acesso em: junho de 2019.
- Firebase Database. (2019) “Firebase Realtime Database”, <https://firebase.google.com/docs/database/>. Acesso em: junho de 2019.
- Google Developers. SDK do Google Maps. (2018) <https://developers.google.com/maps/documentation/android-sdk/intro?hl=pt-br>. Acesso em: agosto de 2018.
- Lara, Marilda Lopes Ginez de; Conti, Vivaldo Luiz. (2003) “Disseminação da informação e usuários”. São Paulo em perspectiva, v. 17, n. 3-4, p. 26-34.
- Leite, Fernando César Lima; COSTA, Sely Maria de Souza. (2016) Modelo genérico de gestão da informação científica para instituições de pesquisa na perspectiva da comunicação científica e do acesso aberto. *Investigación bibliotecológica*, v. 30, n. 69, p. 43-74.
- Monteiro, Edivaldo Antonio. (2013) “Utilização de técnicas ágeis em projetos exclusivos de teste de software”. 58 p. Monografia (Especialização em Curso de Especialização em Informática) Universidade Federal do Paraná, Curitiba.
- Munhoz, Júlia Vidigal; Cerri, Natália; Rodrigues, Jonatan. (2018) “Tudo o que você precisa saber sobre Aplicativo de eventos”. <https://www.moblee.com.br/aplicativo-de-eventos/>. Acesso em: novembro de 2018.
- Pereira, Lucio Camilo Oliva; DA SILVA, Michel Lourenço. (2009) “Android para desenvolvedores”. Brasport.
- Ribeiro, L. (2016) “O que é UML e Diagramas de Caso de Uso: introdução prática à UML”, <http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>, Acesso em: outubro de 2018.
- Rossdeutscher, M. P. (2017) “Aplicação Móvel para Divulgação de Campanhas, Eventos, e Pedidos de Doações por Instituições do Terceiro Setor”, Trabalho de Conclusão de Curso (Graduação), Curso de Sistemas de Informação, Universidade Franciscana, Santa Maria, RS.
- Silva, Alessandra Galvão da. (2018) “A importância dos métodos ágeis na engenharia de software”, https://app.uff.br/riuff/bitstream/1/5488/1/TCC_ALESSANDRA_GALVAO_DA_SILVA%20%281%29.pdf. Acesso em: outubro de 2018.
- Silva, F. G.; Hoentsch, Sandra CP; Silva, Leila. (2009) “Uma análise das Metodologias Ágeis FDD e Scrum sob a Perspectiva do Modelo de Qualidade MPS”. *BR. Scientia Plena*, v. 5, n. 12.
- Sommerville, Ian et al. (2007) “Software engineering. Addison-wesley”.

Souza, M. A. (2017) “Aplicação Android direcionada para ONGs e projetos objetivando viabilizar a adoção de animais domésticos.”, Trabalho de Conclusão de Curso (Graduação), Curso de Sistemas de Informação, Universidade Franciscana, Santa Maria, RS.

Apêndices

Apêndice A. Regras de Negócio

Tabela 3. Regras de Negócio

Regras de Negócio	
RN	Descrição
RN01	São considerados como evento: Congressos, Convenções, Seminários, Workshops, Oficinas, Roadshows, Simpósios, Treinamentos, Cursos, Conferências, Exposição, Vagas de emprego e Residências.
RN02	Somente poderá ser cadastrado eventos dos tipos Vaga de Emprego, Curso, Residência e Treinamento se o usuário for do tipo Pessoa Jurídica.

Apêndice B. Descritivo de Caso de Uso (Gerenciar evento)

Tabela 4. Descritivo do Caso de Uso Gerenciar Evento

Identificação	UC01
Caso de Uso	Gerenciar evento
Descrição	Este descritivo de caso de uso descreve o processo por completo de como gerenciar o evento com Cadastro, Consulta, Edição e Exclusão.
Ator Principal	Usuário (Pessoa Física ou Pessoa Jurídica).
Ator(es) Secundário(s)	N/A.
Pré Condições	
PC1	O usuário deve ter realizado <i>Login</i> no sistema.
Pós Condições	
PO1	Operação realizada com sucesso.
Fluxo Principal	
FP1	O usuário pode escolher uma ação: Cadastrar Evento [FA1], Consultar Eventos [FA2], Alterar Evento [FA3] ou Excluir Evento [FA4].
FP2	Fim do caso de uso.
Fluxo Alternativo	
FA1	Se o usuário escolher a opção "Cadastrar Evento" do menu:
1.	O sistema apresenta a tela de cadastro de evento.
2.	O usuário realiza o preenchimento dos campos.
2.1.	O usuário escolhe a cidade na qual o evento ocorrerá.
2.2.	O usuário confirma/altera os dados para contato.
3.	O usuário finaliza o cadastro.
4.	O sistema valida o cadastro [FE1].
5.	O sistema registra o cadastro.
6.	O sistema retorna para [FP1].
FA2	Se o usuário escolher a opção "Consultar Eventos" do menu:
1.	O sistema apresenta os eventos cadastrados pelo usuário [FA3, FA4].
2.	Sistema retorna para [FP1].
FA3	Se o usuário escolher a opção "Alterar Evento" do menu:

(Continua)

(Conclusão)

1.	O usuário seleciona o evento que deseja alterar.
2.	O sistema apresenta os campos do evento preenchidos.
3.	O usuário altera os dados necessários.
4.	O sistema valida o cadastro [FE1].
5.	O sistema registra os dados alterados.
6.	O sistema retorna para [FA2].
FA4	Se o usuário escolher a opção “Excluir Evento” do menu:
1.	O usuário seleciona o evento que deseja excluir.
2.	O sistema solicita a confirmação da exclusão do evento.
3.	O usuário confirma a exclusão.
4.	O sistema executa a exclusão da doação [FE2].
5.	O sistema retorna [FA2].
Fluxo de Exceções	
FE1	Dados inválidos.
1.	O sistema exibe a mensagem de dados inválidos.
2.	O sistema retorna para o fluxo correspondente.
FE2	Impeditivos na exclusão.
1.	O sistema exibi a mensagem de exclusão malsucedida.
2.	O sistema retorna para o fluxo correspondentes.
Regras de Negócio	RN01, RN02
Casos de Uso Estendidos	N/A
Casos de Uso Incluídos	N/A

Apêndice C. Descritivo de Caso de Uso (Visualizar evento)

Tabela 5. Descritivo do Caso de Uso Visualizar Evento

Identificação	UC02
Caso de Uso	Visualizar evento
Descrição	Este descritivo de caso de uso refere-se à visualização dos eventos devidamente cadastrados.
Ator Principal	Usuário (Pessoa Física ou Pessoa Jurídica).
Ator(es) Secundário(s)	N/A.
Pré Condições	
PC1	O usuário deve ter realizado <i>Login</i> no sistema.
Pós Condições	
PO1	Operação realizada com sucesso.
Fluxo Principal	
FP1	O usuário acessa o menu de visualização de animais.
FP2	O sistema exibe a lista de eventos.
FP3	O usuário visualiza a lista de eventos.
FP4	O usuário seleciona um evento para visualização dos detalhes [FA1, FA2].
FP5	O sistema exibe o cadastro do evento.
FP6	O usuário visualiza o cadastro do evento.
FP7	O usuário clica no botão demonstrar interesse [FA3].
FP8	O sistema registra interesse do usuário no evento.
FP9	O sistema notifica o usuário responsável pelo evento.
FP10	Fim do caso de uso.
Fluxo Alternativo	
FA1	Se o usuário desejar alterar o filtro do estado e ou cidade:
1.	O usuário acessa o menu de filtros.
2.	O sistema apresenta as opções de filtro.
3.	O usuário altera o filtro para o estado e ou cidade desejada.
4.	O sistema filtra os eventos conforme o(s) novo(s) parâmetro(s).
5.	O sistema retorna para [FP2].
FA2	Se o usuário não desejar visualizar os detalhes do evento:
1.	O sistema retorna para [FP10].
FA3	Se o usuário não desejar demonstrar interesse no evento:
1.	O usuário clica no botão voltar [FA4].
2.	O sistema retorna para [FP2].
FA4	Se o usuário não desejar visualizar a lista de animais:
1.	O sistema retorna para [FP10].
Fluxo de Exceções	
FE1	O sistema impede o cadastro e/ou alteração se todos os campos obrigatórios não estiverem preenchidos.
FE2	O sistema impede a exclusão pois o evento possui vínculos.
Regras de Negócio	N/A
Casos de Uso Estendidos	N/A
Casos de Uso Incluídos	N/A

Apêndice D. Diagramas de Atividades e subprocessos

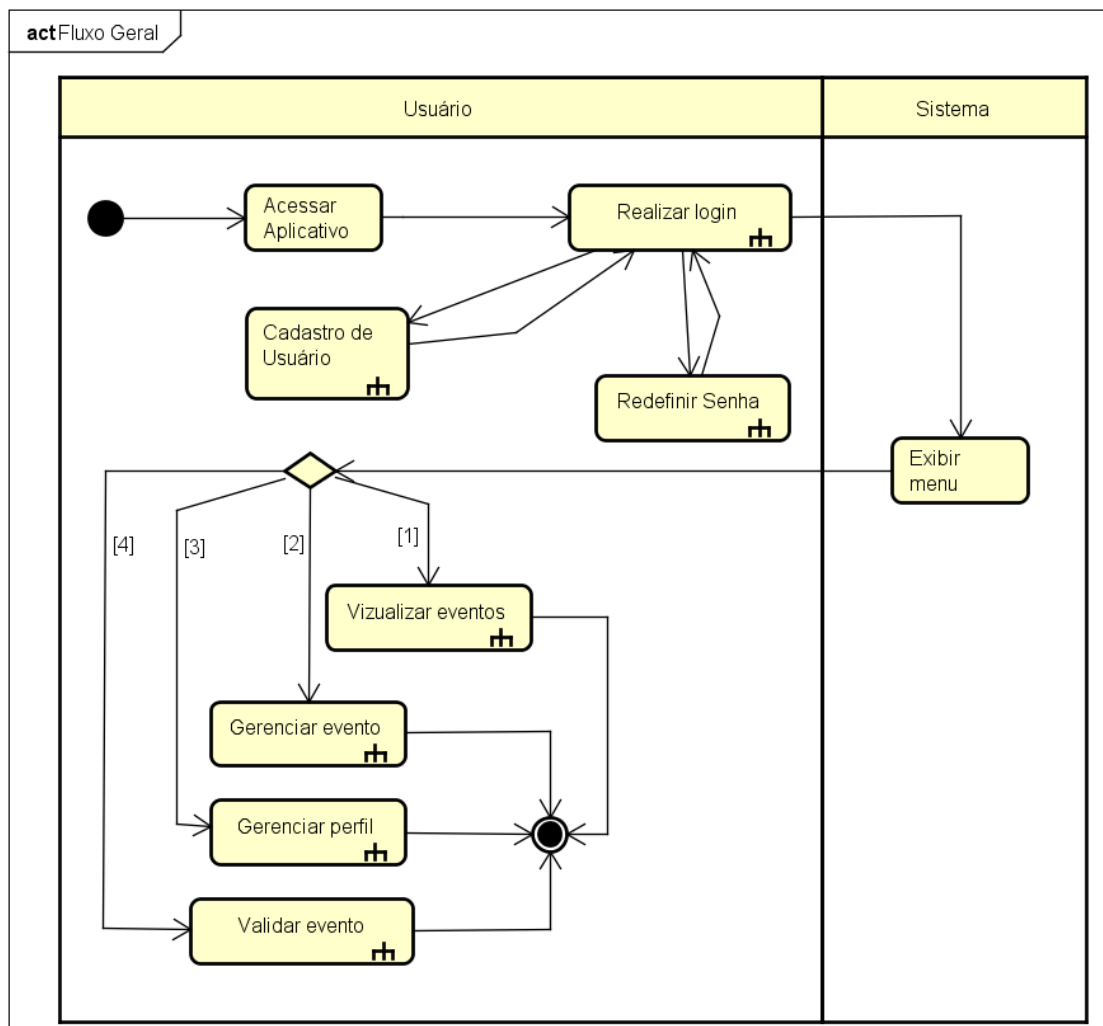


Figura 10. Diagrama de Atividade do fluxo geral do aplicativo

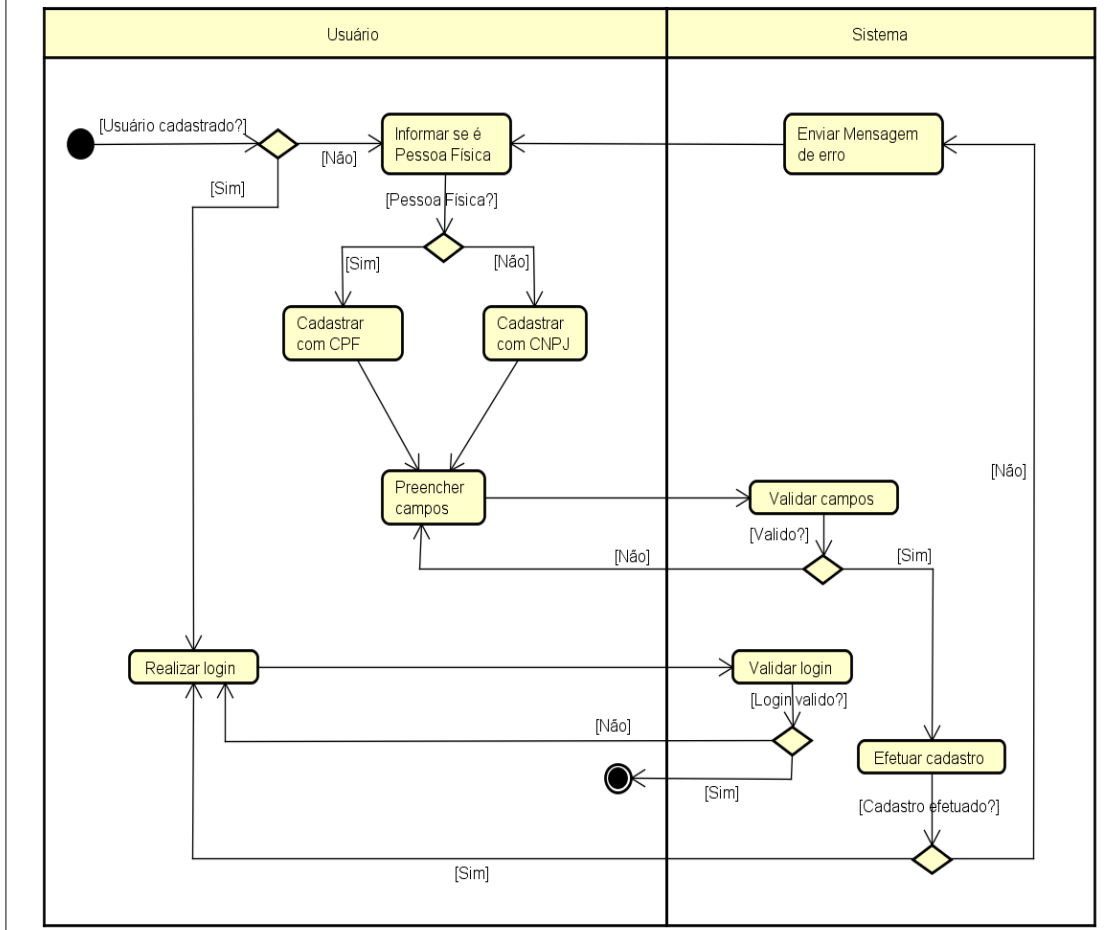


Figura 11. Subprocesso Realizar Login e Cadastro de Usuário

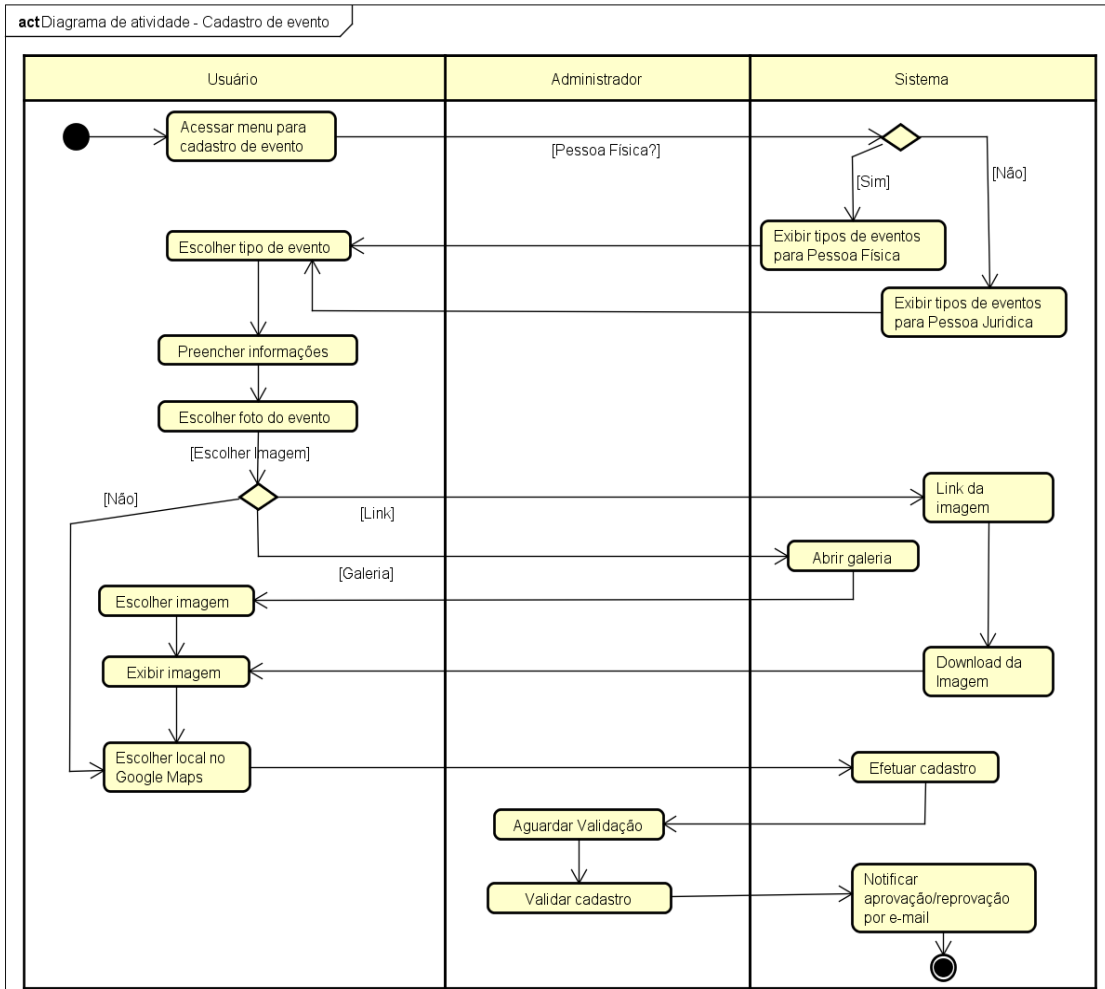


Figura 12. Subprocesso Cadastrar Evento

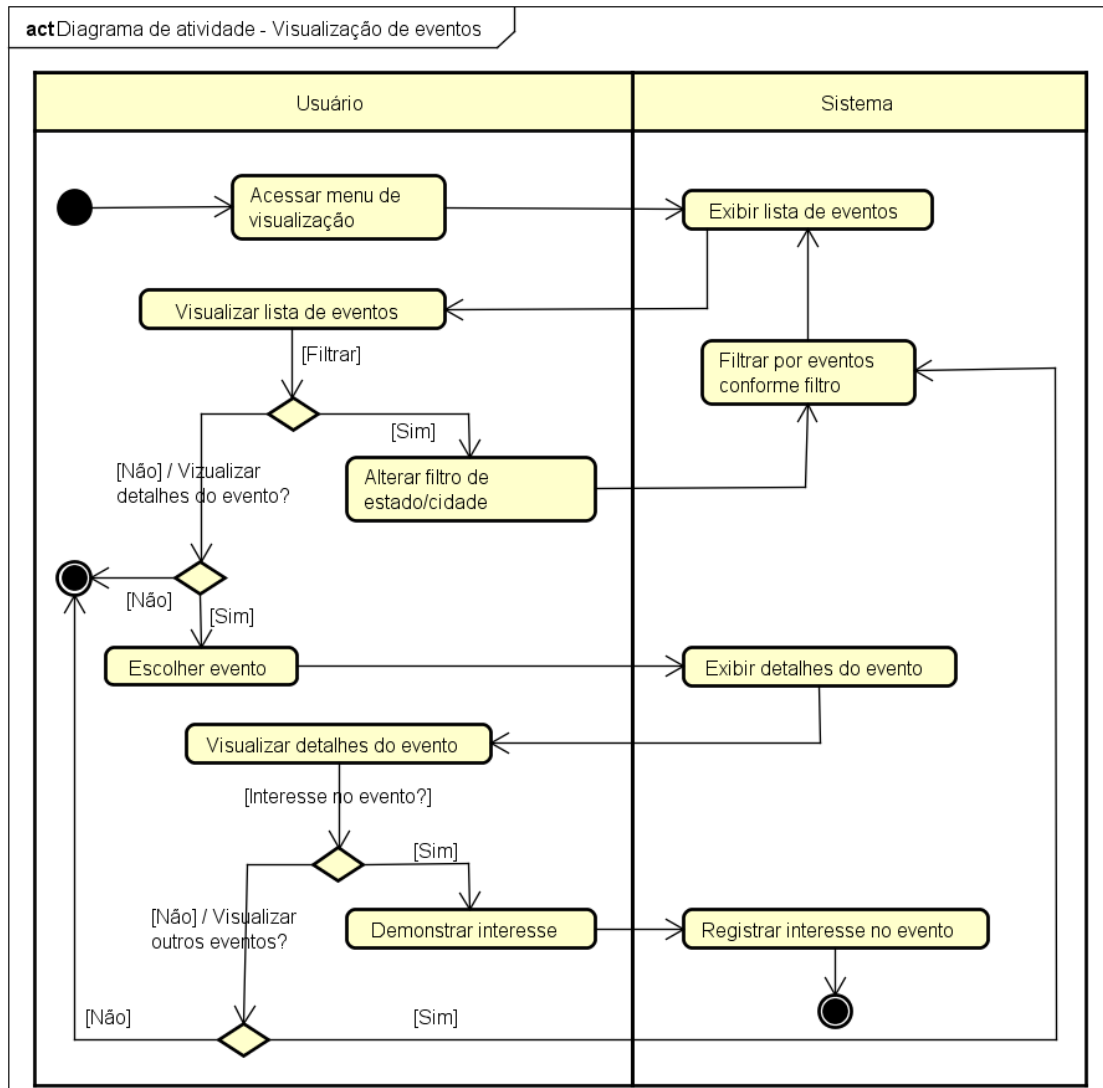


Figura 13. Subprocesso Visualizar Eventos

Apêndice E. Planejar por Funcionalidades.

O objetivo desta etapa é construir um plano de desenvolvimento proprietários de conjuntos de funcionalidades. Essas atividades são consideradas em conjunto após serem feitos diversos refinamentos [Silva 2018].

Mostra-se, através da Tabela 6, a ordem que será seguida para a implementação deste projeto que forma descritas na Tabela 1. Para o cálculo da estimativa de tempo, utilizou-se a seguinte métrica:

- 10 horas por semana para implementação do aplicativo;
- 5 meses desenvolvimento, ou seja, 21 semanas ao total;
- Com a multiplicação das horas semanais e quantidade de semanas, chega-se a 210 horas para ser distribuídas entre os requisitos;
- O total de horas de implementação conforme distribuição na Tabela 3 é de 184 horas;
- As 26 horas restantes serão alocadas para testes do aplicativo.

Tabela6. Estimativa do tempo de implementação das funcionalidades

Código	Funcionalidade	Tempo (Horas)
RF01	Gerenciar usuário	40
RF02	Cadastrar evento	6
RF02.1	Escolher cidade do evento	5
RF02.2	Validar evento	7
RF02.3	Informar aprovação/reprovação	7
RF03	Gerenciar evento	40
RF04	Exibir eventos	40
RF04.1	Apresentar eventos conforme filtros	10
RF04.2	Exibir detalhes do evento	10
RF04.3	Demonstrar interesse no evento	4
RF04.3.1	Notificar usuário do interesse	10
RF05	Visualizar lista de interessados	5

Na Figura 14 é apresentado o Diagrama de Casos de Uso, responsável por escrever as principais funcionalidades do sistema, mostrando a interação destas com os usuários e não se importando com detalhes técnicos que expliquem como o sistema realiza o processo [Ribeiro 2016].

Nos Apêndices B e C são apresentados dois Descritivos de Casos de Uso, considerados os mais relevantes para a implementação da aplicação em questão.

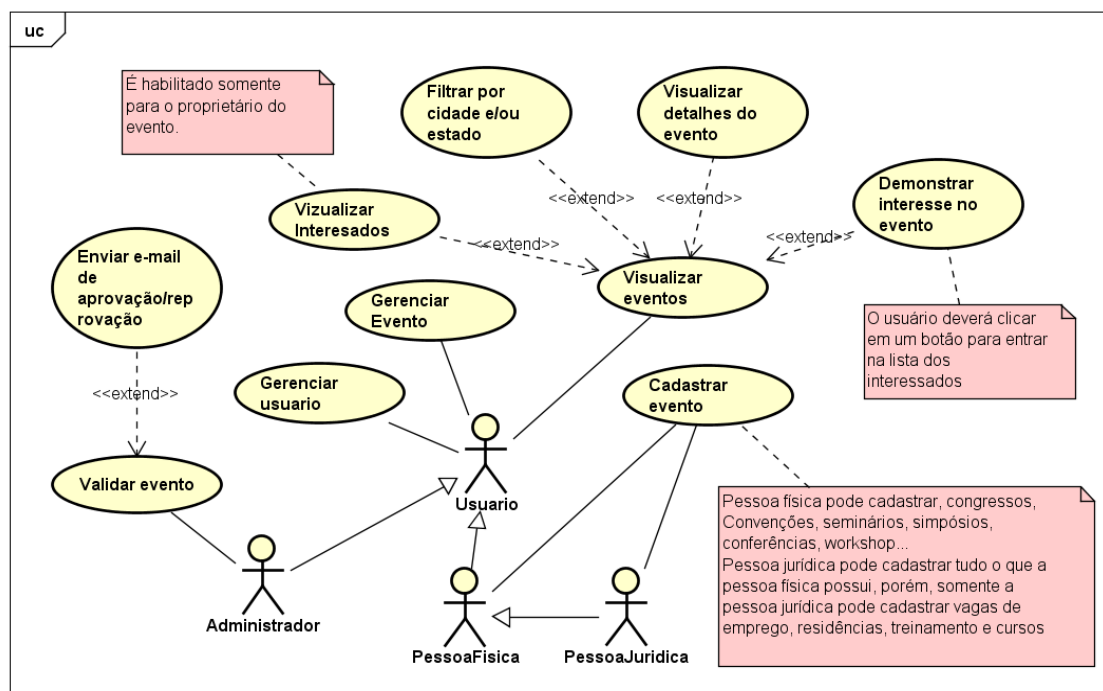


Figura 14. Diagrama de Caso de Uso

Apêndice F. Interfaces do software.

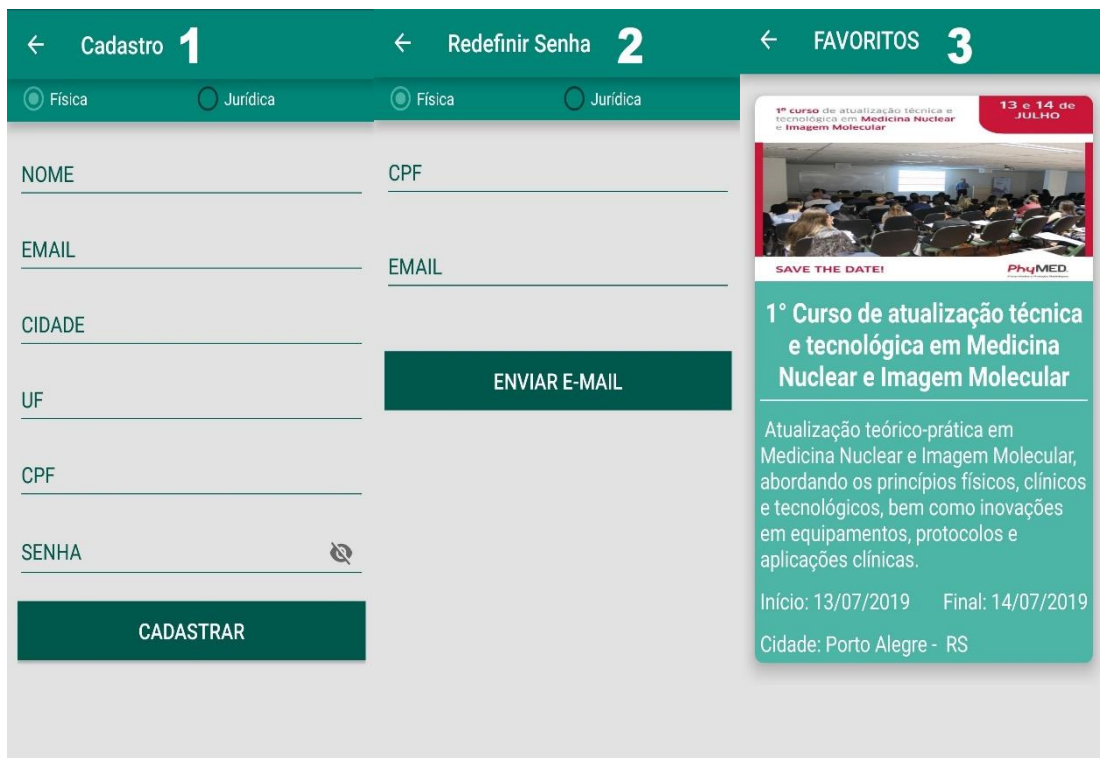


Figura 15. Interfaces de cadastro de usuário, redefinição de senha e eventos favoritos

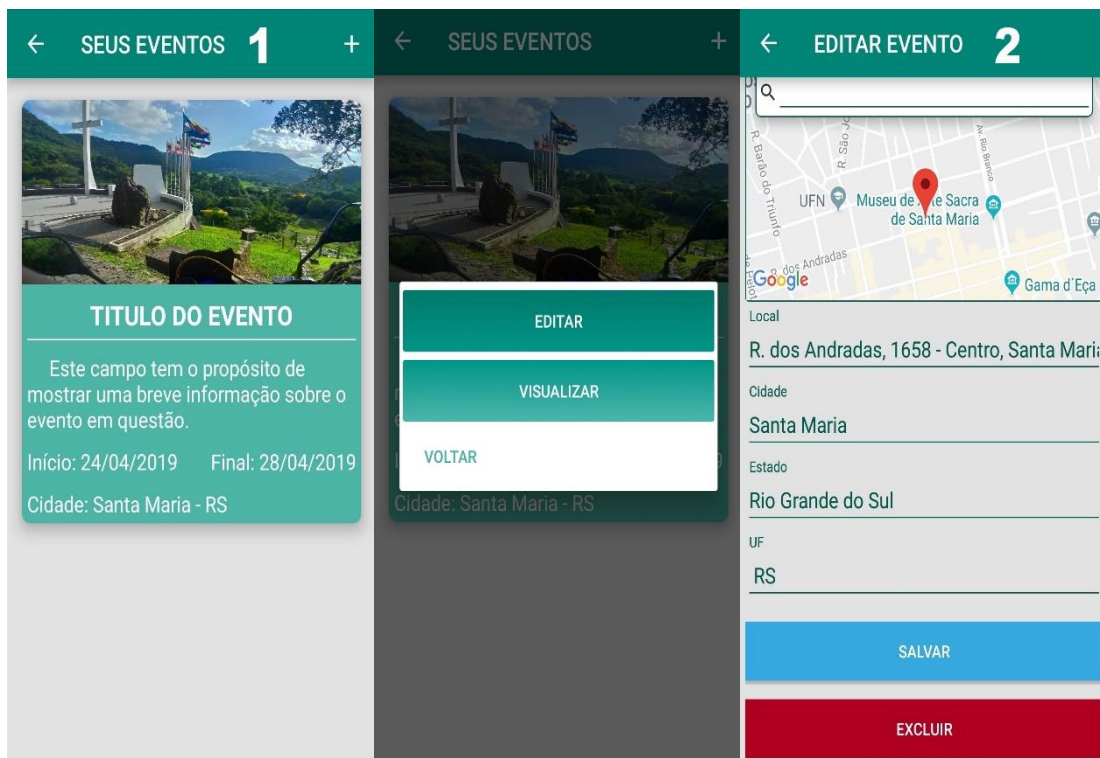


Figura 16. Interfaces dos eventos cadastrados pelo usuário (Seus Eventos), escolha para editar ou visualizar seu evento e Interface de validação, por parte do administrador

← EDITAR EVENTO
← EDITAR EVENTO
← EDITAR EVENTO

Em Análise...

Residência ▼

Título *

TITULO DO EVENTO

Descrição Parcial * 16 / 100

Este campo tem o propósito de mostrar uma breve informação sobre o evento em questão.

Descrição Completa 90 / 200

Data de Início * Data final *

24/04/2019 28/04/2019

Data de Início * Data final *

24/04/2019 28/04/2019

Carga Horária (Horas) Número de Vagas

10 20

Link do Site do Evento

https://www.google.com.br

Residência ▼

Ciências Tecnológicas ▼

SubArea

Ciência da Computação

Local

R. dos Andradas, 1658 - Centro, Santa Mari

Cidade

Santa Maria

Estado

Rio Grande do Sul

UF

RS

SALVAR

EXCLUIR

Local

R. dos Andradas, 1658 - Centro, Santa Mari

Cidade

Santa Maria

Estado

Rio Grande do Sul

UF

RS

SALVAR

EXCLUIR

Figura 17. Interface de edição do evento

Apêndice G. Diagramas de classes contidas no diagrama de pacotes principal.

1. Package Usuário.

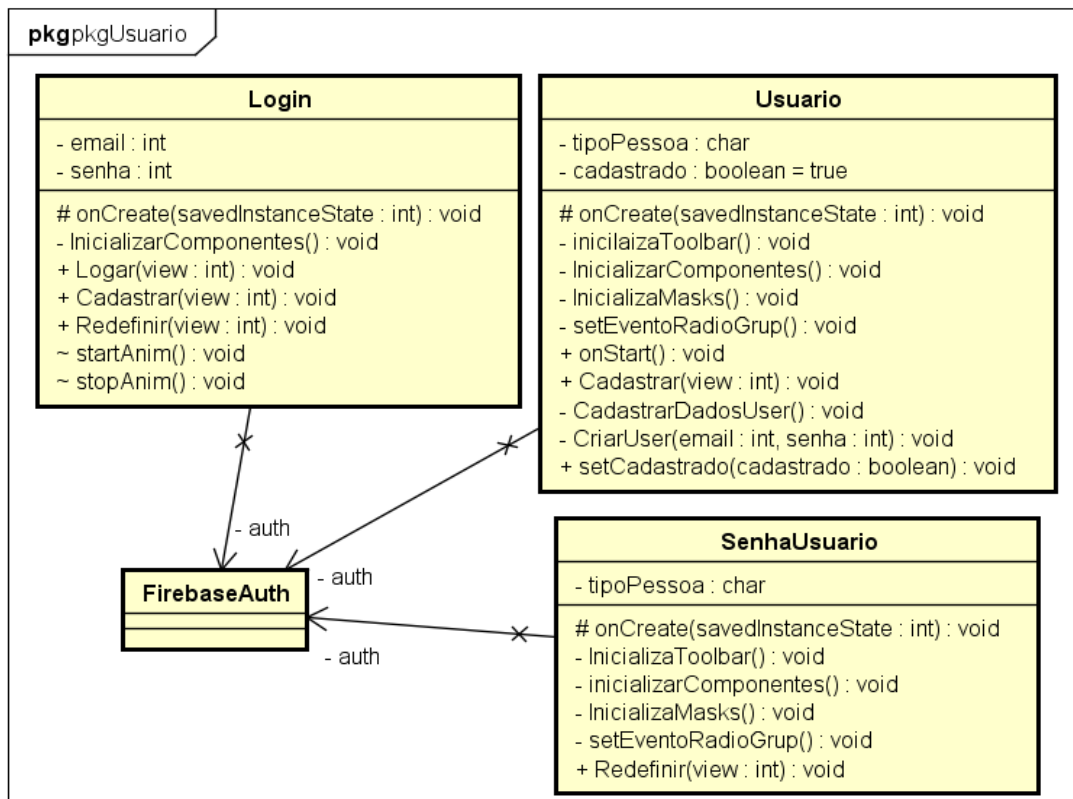


Figura 18. Diagrama de classes Usuário

2. Package Evento.

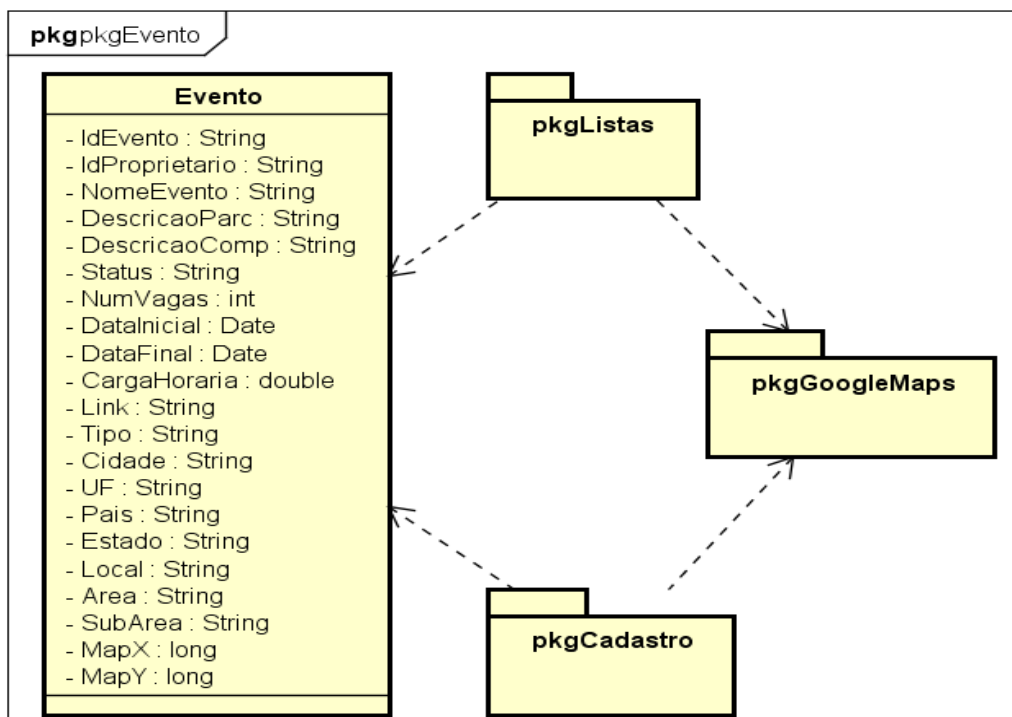


Figura 19. Diagrama de classes Evento

2.1. Package Listas.

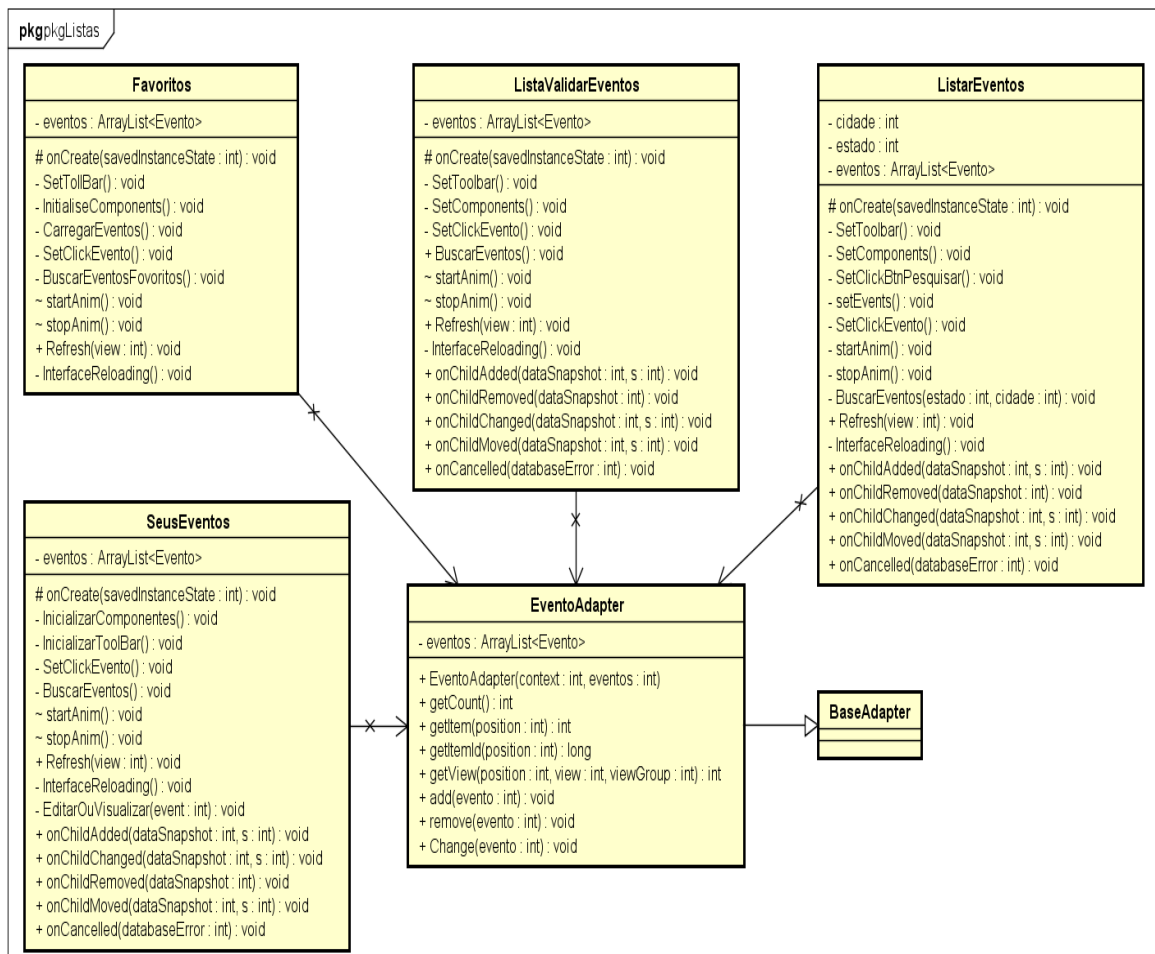


Figura 20. Diagrama de classes Listas

2.2. Package GoogleMaps.

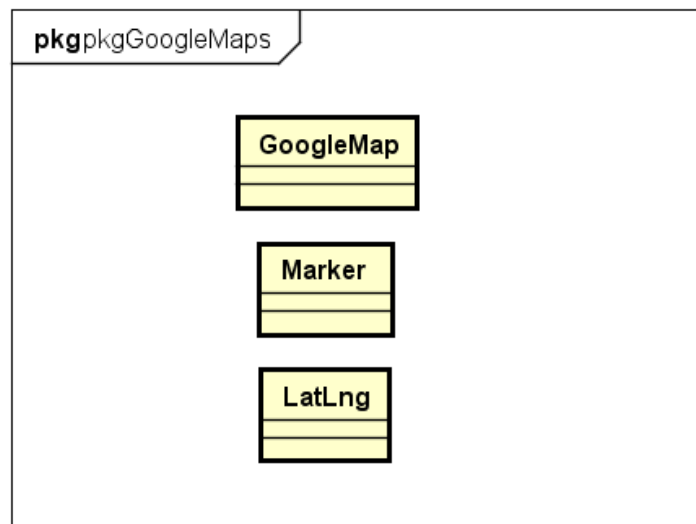


Figura 21. Diagrama de classes GoogleMaps

2.3. Package Cadastro.

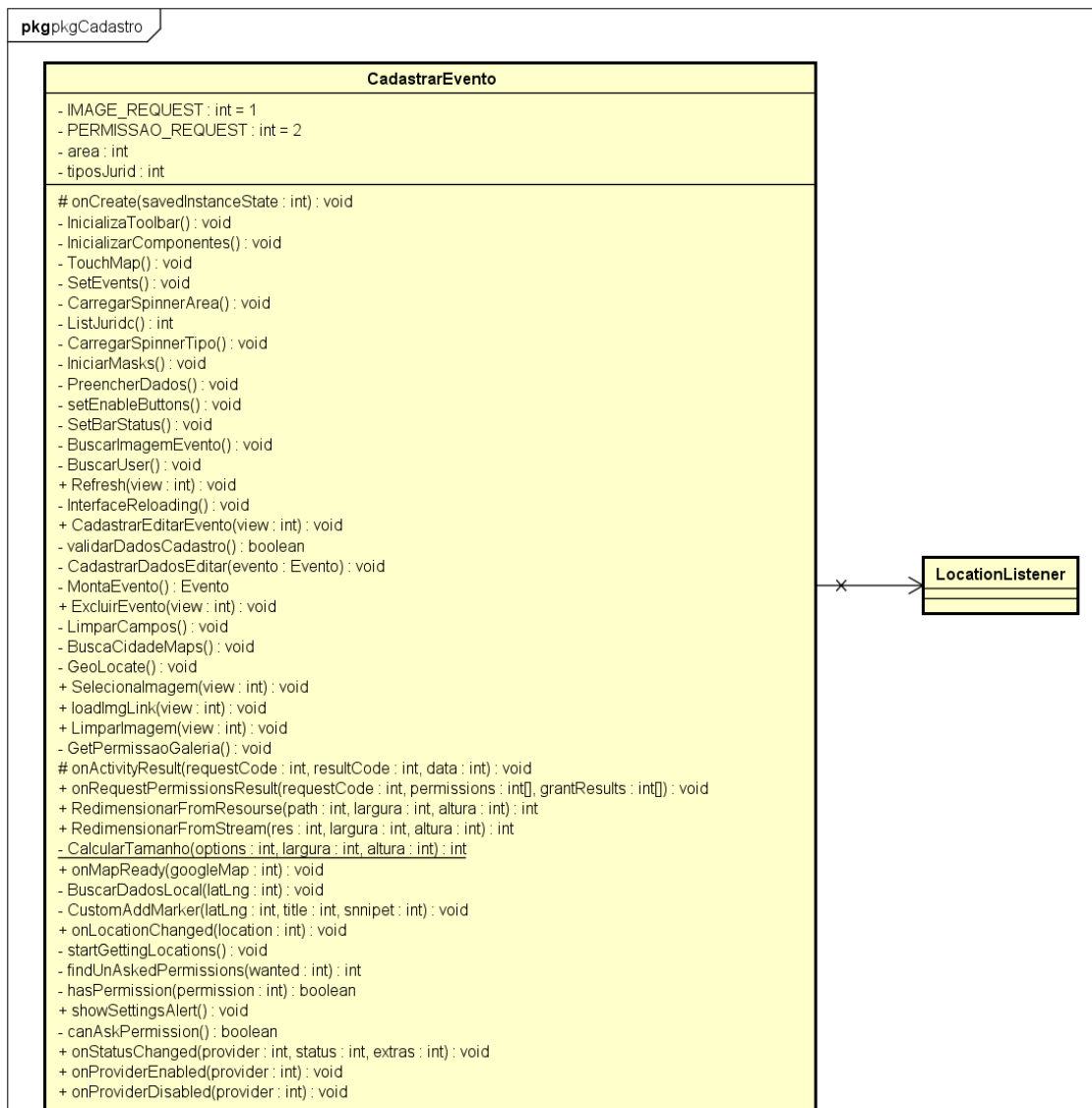


Figura 22. Diagrama de classes Cadastro

3. Package Firebase.

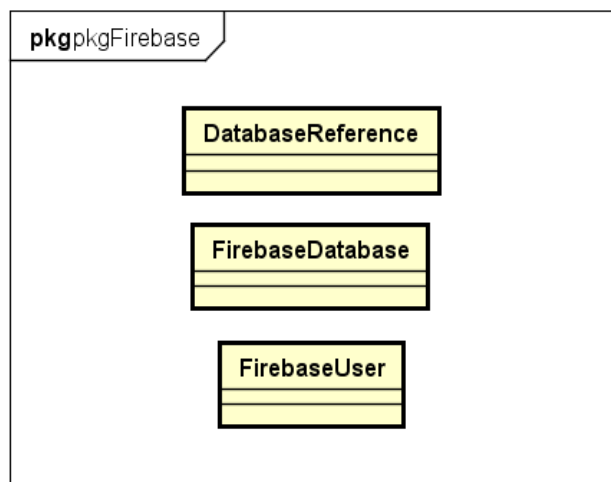


Figura 23. Diagrama de classes Firebase