

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

ALEXANDRE DOTTI

APLICATIVO MOBILE PARA DIVULGAÇÃO DE EVENTOS

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2023

ALEXANDRE DOTTI

APLICATIVO MOBILE PARA DIVULGAÇÃO DE EVENTOS

Mobile Application for Promoting Events

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Vinicius Pegorini.

PATO BRANCO
2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ALEXANDRE DOTTI

APLICATIVO MOBILE PARA DIVULGAÇÃO DE EVENTOS

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas
da Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 20/junho/2023

Vinicius Pegorini
Mestrado
Universidade Tecnológica Federal do Paraná

Andreia Scariot Beulke
Mestrado
Universidade Tecnológica Federal do Paraná

Edilson Pontarolo
Doutorado
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2023

RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo para a divulgação e o gerenciamento de eventos. A proposta desse trabalho foi motivada pelo crescimento de aplicações para dispositivos móveis, principalmente para o sistema operacional Android. Com o passar do tempo, o *smartphone* está cada vez mais presente no dia a dia do usuário. É uma tecnologia capaz de simplificar e facilitar a vida de seus proprietários, o que possibilitou aos desenvolvedores se adaptarem e lançarem aplicativos no mercado com o objetivo de auxiliar os usuários em suas tarefas diárias. O Diretório Central de Estudantes da Universidade Tecnológica Federal do Paraná, apresenta certa carência de aplicativos para gerenciamento de eventos. Com base nisso, foi desenvolvido um software para administrar e divulgar eventos, contribuindo para o controle de tarefas e a interação da comunidade à rotina universitária. O projeto foi implementado utilizando a linguagem de programação Java, e a plataforma do Google Firebase para armazenar os dados do aplicativo, controlar a autenticação de usuários, apresentar métricas sobre a aplicação, mensurar o comportamento do usuário e enviar notificações. Como resultado, tem-se um aplicativo com as funcionalidades de cadastro de eventos, inscrição nos eventos, comentar e curtir eventos, consultar de eventos, cadastro de informativos e consulta de informativos. Para administradores o aplicativo apresenta as funcionalidades de consulta de usuários cadastrados por perfil de acesso, consulta de áreas que possuem mais interesse pelos usuários e definir o perfil de acesso de cada usuário. Foi desenvolvido também um gerenciador de notificações para alertar os usuários sobre os eventos marcados no dia, prevenindo o não comparecimento.

Palavras-chave: gerenciamento de eventos; android; diretório central de estudantes.

ABSTRACT

This work presents the development of an application for the promotion and management of events. The proposal of this work was motivated by the growth of applications for mobile devices, especially for the Android operating system. As time goes by, the smartphone is increasingly present in the user's day to day life. It is a technology capable of simplifying and facilitating the lives of its owners, which has enabled developers to adapt and launch applications on the market with the aim of helping users in their daily tasks. The Central Student Directory of the Federal Technological University of Paraná has a certain lack of applications for event management. Based on this, a software was developed to manage and promote events, contributing to the control of tasks and the interaction of the community with the university routine. The project was implemented using the Java programming language, and the Google Firebase platform to store the application's data, control user authentication, present metrics about the application, measure user behavior, and send notifications. As a result, we have an application with the functionalities of registering events, registering for events, commenting and liking events, consulting events, registering newsletters, and consulting newsletters. For administrators, the application has the functionalities to query registered users by access profile, query areas that have more interest for users, and define the access profile of each user. A notification manager was also developed to alert users about the events scheduled for the day, preventing non-attendance.

Keywords: management of events; android; the central student directory.

LISTA DE FIGURAS

Figura 1 - Acesso Redes Sociais	15
Figura 2 - Sympla	16
Figura 3 - Eventify	17
Figura 4 - Lets.events	18
Figura 5 - Caso de uso	26
Figura 6 - Diagrama de Entidade-Relacionamento	29
Figura 7 - Tela de Autenticação	30
Figura 9 - Tela Inicial para Administradores	31
Figura 11 - Cadastro de Eventos	32
Figura 13 - Detalhes do Evento	33
Figura 14 - Tela de Dashboard	34
Figura 15 - Permissão do Usuário	35
Figura 16 - Cadastro de Informativos	36
Figura 17 - Estrutura dos Pacotes do Aplicativo	37
Figura 18 - Estrutura da Camada Visual	41

LISTA DE QUADROS

Quadro 1 - Lista de ferramentas e tecnologias.....	19
Quadro 2 - Registro de Eventos.....	24
Quadro 3 - Gerenciar Evento.....	24
Quadro 4 - Consultar Evento.....	25
Quadro 5 - Levantamento de Dados.....	25
Quadro 6 - Enviar Notificação.....	25
Quadro 7 - Publicação de Materiais.....	26

LISTAGEM DE CÓDIGOS

Listagem 1 - Método Cadastrar/Alterar Evento.....	38
Listagem 2 - Consulta de Eventos	39
Listagem 3 - Consulta de Áreas	40
Listagem 4 - Cadastro de Informativos	41
Listagem 5 - Envio de Notificações	43

LISTA DE ABREVIATURAS E SIGLAS

DCE	Departamento Central dos Estudantes
DTO	<i>Data Transfer Object</i>
GPS	<i>Global Positioning System</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
RF	Requisitos Funcionais
RNF	Requisitos Não-Funcionais
SCIELO	<i>Scientific Electronic Library Online</i>
TIC	Tecnologia da Informação e Comunicação
UML	<i>Unified Modeling Language</i>
UTFPR	Universidade Tecnológica Federal do Paraná
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Considerações Iniciais	11
1.2	Objetivos	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	12
1.3	Justificativa	12
1.4	Estrutura do Trabalho	13
2	REFERENCIAL TEÓRICO	14
2.1	Sociedade e Comunicação	14
2.2	Redes Sociais	14
2.3	Aplicativos Similares	16
3	MATERIAIS E MÉTODO	19
3.1	Materiais	19
3.1.1	Android Studio	19
3.1.2	Linguagem de Programação Java	20
3.1.3	Firebase	20
3.1.4	Visual Paradigm	20
3.2	Método	21
4	RESULTADOS	23
4.1	Escopo do Sistema	23
4.2	Modelagem do Sistema	24
4.3	Apresentação do Sistema	30
4.4	Implementação do Sistema	36
5	CONCLUSÃO	44
	REFERÊNCIAS	45

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho. No final do capítulo é apresentada a organização do texto que exhibe a organização dos seus capítulos.

1.1 Considerações Iniciais

O uso de dispositivos móveis no Brasil para acessar aplicações bancárias, redes sociais, agenda e *e-mail*, cresce a cada ano. De acordo com a Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas, realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP), o número de *smartphones* em 2019 teve um aumento de 10 milhões em relação ao ano anterior, totalizando 230 milhões de aparelhos ativos no país (MEIRELLES, 2019).

Segundo Ribeiro (2019), no Brasil, 42% dos jovens entre 16 e 24 anos de idade afirmaram fazer uso exclusivo dos *smartphones* para suas atividades *on-line*, principalmente pela possibilidade de comunicação cada vez mais imediata e simplificada, estendendo para funções relacionadas ao entretenimento, à segurança e ao controle pessoal. Com base nesses dados, percebe-se que o aparelho celular substitui com eficiência diversos itens cada vez menos utilizados para controle pessoal, tais como, agendas, despertador, aparelhos de *Global Positioning System* (GPS), lanterna, gravador e relógios.

Atualmente, com os recursos presentes e acessíveis, o usuário está sempre buscando na área da tecnologia, aplicativos e serviços que gerenciam e simplificam seu trabalho. Assim abriu-se uma área para profissionais explorarem e aumentarem a oferta dos serviços para esse tipo de mercado.

Nesse contexto, neste trabalho foi desenvolvido um aplicativo que tem a função de auxiliar o Diretório Central dos Estudantes (DCE) da Universidade Tecnológica Federal do Paraná, Campus Pato Branco (UTFPR-PB) para divulgar eventos e alertar o usuário sobre eventos que é do seu interesse evitando que ele não compareça por falta de informação, esquecimento ou atenção.

1.2 Objetivos

A seguir são apresentados o objetivo geral e os específicos do trabalho. O objetivo geral está relacionado com o resultado principal da realização deste aplicativo e os objetivos específicos complementam o geral em termos de funcionalidades do sistema.

1.2.1 Objetivo Geral

Desenvolver um aplicativo para dispositivos móveis para divulgação, acompanhamento e notificação de eventos aos usuários.

1.2.2 Objetivos Específicos

- Possibilitar que os usuários possam manter seus próprios eventos.
- Permitir que os usuários possam acompanhar e receber notificações sobre os eventos cadastrados por outros usuários.
- Registrar e apresentar estatísticas de utilização do aplicativo para os usuários que cadastraram os eventos.
- Possibilitar a publicação de projetos acadêmicos, artigos e matérias informativas.

1.3 Justificativa

Anteriormente, quando os eventos da UTFPR Campus Pato Branco eram agendados, a divulgação na maioria dos casos era realizada por cartazes e rádios, e ainda é, porém agora para divulgação existe outras opções, como as redes sociais, sites e aplicativos que auxiliam e organizam na divulgação dos eventos. A rotina das pessoas conta com um número cada vez maior de atividades para serem administradas. Muitas vezes torna-se difícil administrar o tempo e controlar todas as atividades diárias, por exemplo, reuniões, palestras, cursos.

O setor de eventos apresenta certa carência de sistemas gerenciadores específicos, por tratar-se de uma área em recente crescimento. Diante da necessidade de uma ferramenta que minimize as dificuldades para divulgar e alertar

sobre próximos eventos de diversas áreas de interesses do usuário, foi desenvolvido um aplicativo para dispositivos móveis com sistema operacional Android para simplificar e melhorar a divulgação das informações sobre eventos para usuários cadastrados. Essas informações serão expostas para o usuário se informar sobre o tema do evento e se cadastrar e acompanhar aqueles de seu interesse.

Na maioria das vezes o controle do gerenciamento por parte dos administradores de eventos é feito via rede social Facebook, ou manualmente em agendas, e acabam ficando sem estatísticas relevantes, tais como número de pessoas que irão comparecer no evento e temas mais procurados.

Para desenvolvimento do projeto foi escolhida a linguagem de programação Java e a plataforma do Google Firebase para armazenar os dados do aplicativo. Essa plataforma também serviu para gerar as métricas da aplicação, mensurar o comportamento do usuário, gerando relatórios do uso do aplicativo. O Firebase também disponibiliza um banco de dados NoSQL (Firebase Realtime Database) hospedado em nuvem, onde os dados são armazenados como JSON e sincronizados em tempo real com todos os clientes conectados.

Também será usado o Firebase para diariamente enviar notificações aos usuários sobre os eventos marcados no dia.

1.4 Estrutura do Trabalho

Este trabalho de conclusão de curso está organizado em capítulos, dos quais este é o primeiro e apresenta a justificativa do trabalho e os objetivos gerais e específicos. O Capítulo dois apresenta o referencial teórico sobre a Sociedade e Comunicação, redes sociais e alguns aplicativos similares a gestão de eventos. No Capítulo três são apresentados os materiais e o método utilizados para o desenvolvimento do trabalho. São apresentadas, no quarto capítulo, a modelagem, requisitos funcionais, diagramas e resultados obtidos após o desenvolvimento da aplicação proposta. Finalizando, as conclusões e referências bibliográficas que fundamentaram este trabalho de conclusão de curso.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta a fundamentação teórica deste trabalho, cujo conteúdo explana sobre o envolvimento da sociedade e comunicação nas redes sociais possibilitando a divulgação de eventos e notícias por meio das mesmas. Por fim, são apresentados alguns dos principais aplicativos utilizados na divulgação de eventos.

2.1 Sociedade e Comunicação

Ao longo do tempo, a sociedade tem passado por várias transformações sociais e estruturais, como as transições da sociedade agrícola para a industrial, da industrial para a sociedade da informação e do conhecimento. Atualmente, estamos vivenciando uma era de mudanças impulsionadas pela internet e pelas tecnologias digitais, o que está gerando um novo modelo social representado pela sociedade em rede.

As Tecnologias da Informação e Comunicação (TIC) podem ser caracterizadas como um conjunto de recursos tecnológicos, com um objetivo comum. As TICs são usufruídas de diferentes formatos, na indústria (no processo de automação), no comércio (no gerenciamento, nas diversas formas de publicidade), no setor de investimentos (informação simultânea, comunicação imediata) e na educação (no processo de ensino-aprendizagem). (VASCONCELOS *et al.*, 2010).

Neste sentido, a tecnologia é utilizada notavelmente no cotidiano das pessoas facilitando a comunicação por intermédio da mídia em geral e das mídias sociais em particular. Esse crescimento da comunicação obtido pelo crescente uso das redes sociais tem alertado empresas na área de eventos a procurar vantagens com o propósito alcançar um maior número de pessoas nas divulgações de seus eventos, obtendo resultados positivos ao organizador (PERINOTTO; BORGES; SOUSA, 2016).

2.2 Redes Sociais

No decorrer dos últimos anos, os meios comunicacionais estão em constante evolução. As redes sociais se destacam pela simplicidade de interação da sociedade, ofertando inúmeros recursos para a troca de informações, por exemplo, plataformas de *e-mails* como Gmail, Outlook, juntamente com o Facebook, e obtenção do

conhecimento por meio ferramentas de pesquisas de artigos e projetos como o Google Acadêmico e SCIELO. Também se destaca a aproximação de indivíduos e contatos entre amigos, familiares e profissionais.

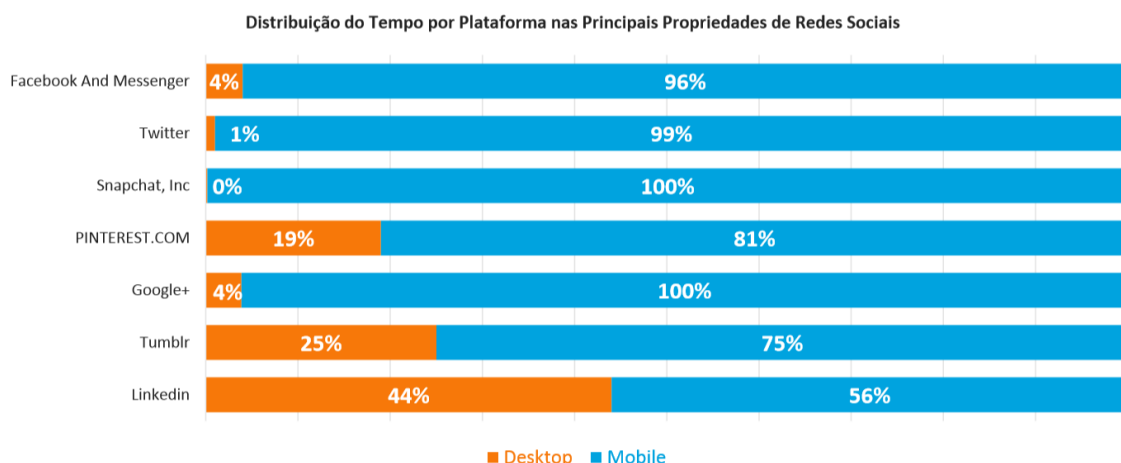
Na comunicação, as redes sociais é uma das formas mais usufruídas na atualidade. Neste contexto, uma rede social é definida como um conjunto de dois elementos: atores (pessoas, instituições ou grupos) e suas conexões (interação ou laços sociais), ou seja, estrutura social composta por pessoas ou organizações, conectadas por vários tipos de relações, construindo uma interação social (WASSERMAN e FAUST, 1994; DEGENNE e FORSÉ, 1999 apud RECUERO, 2009).

Pode-se, portanto, caracterizá-la com o objetivo de relacionar variados grupos sociais, concedendo a interação e relacionamento entre eles.

Segundo pesquisa da ComScore, realizada em junho de 2018 no Brasil, apontou que 26% dos usuários que acessam as redes sociais estão entre a faixa etária de 15 a 24 anos. A Figura 1, registra o sucesso dos dispositivos móveis divulgada pela comScore. A figura mostra que o usuário prefere acessar as redes sociais utilizando um dispositivo móvel.

Figura 1 - Acesso Redes Sociais

O mobile alavanca o sucesso das redes sociais



Fonte: Reproduzido de BURGER, OTAKA (2018)

Recuero (2009) afirma que o Facebook é um dos sistemas com maior base de usuários no mundo. Segundo Minhoto (2012) seus alunos demonstraram interagir

mais com o conhecimento utilizando a ferramenta Facebook devido a familiaridade com o aplicativo.

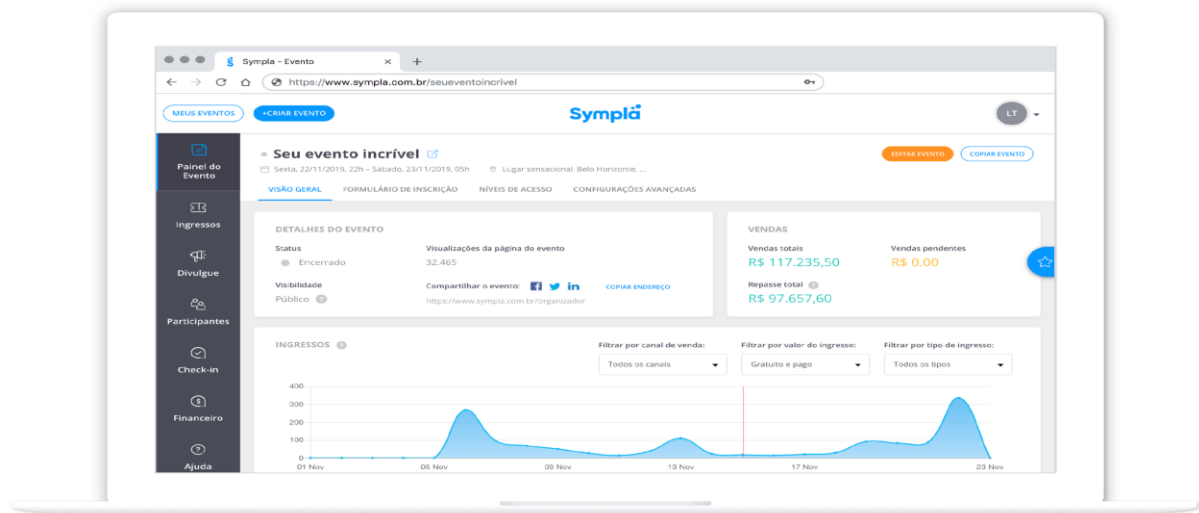
2.3 Aplicativos Similares

No mercado existem algumas empresas que desenvolvem sistemas para a divulgações de eventos, por exemplo Sympla, Eventify e Lets.events.

A Sympla foi fundada em 2012 na cidade de Belo Horizonte, com Rodrigo Cartacho sendo o CEO da empresa. Em 2017 a organização movimentou certa de R\$ 200 milhões de reais, contando com cerca de 200 colaboradores distribuídos em Belo Horizonte, São Paulo, Rio de Janeiro, Recife, Goiânia e Porto Alegre. A plataforma já realizou eventos em mais de 2.800 cidades totalizando mais de 600 mil eventos organizados, sendo uma plataforma para venda e gestão de ingressos e inscrições para eventos (SYMPLA, 2019).

Na Figura 2, é apresentada uma das telas do aplicativo, que exibe as principais funções que ele possui.

Figura 2 - Sympla

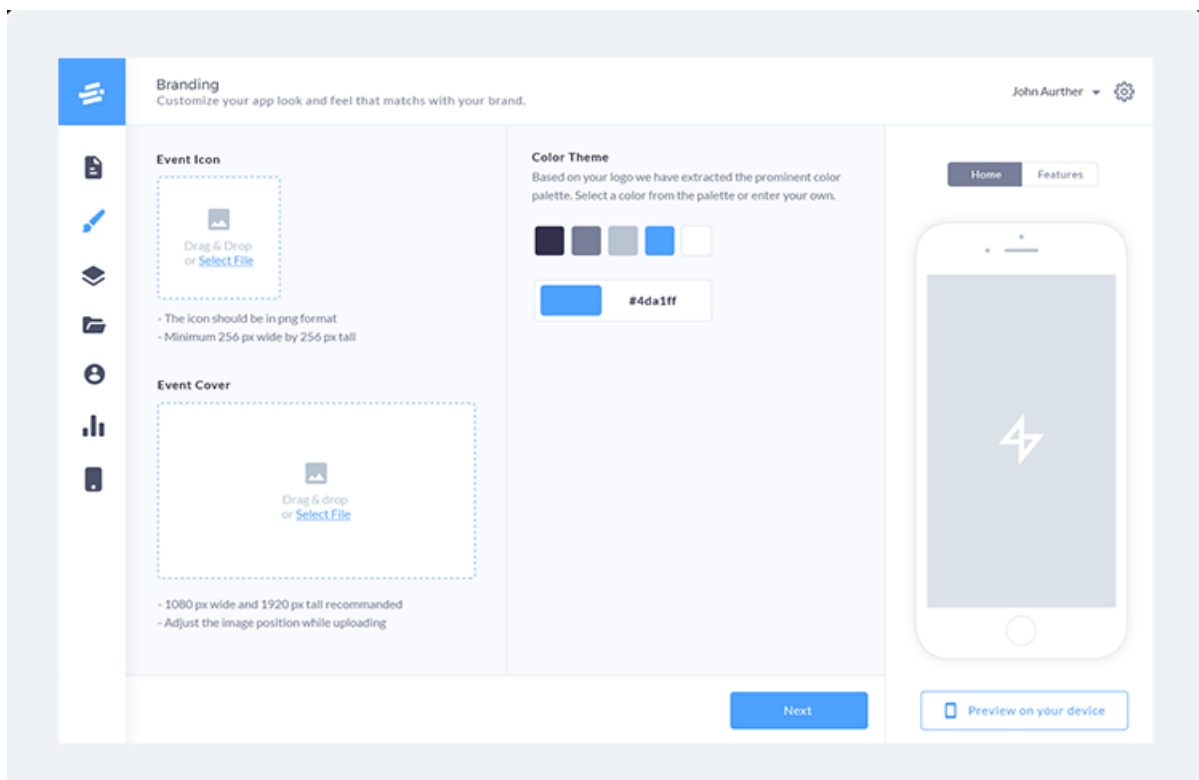


Fonte: Sympla (2019)

O Eventify é uma plataforma para a criação de aplicativos de eventos digitais. Oferece aplicativos móveis personalizados de plataforma para todos os tipos de eventos, com um recurso simples de arrastar e soltar. O aplicativo oferece alguns recursos como compromissos individuais, notificações *push* e mensagens privadas. Além disso, é possível criar o site do evento e hospedar os ingressos nesta plataforma de gerenciamento de eventos. (EVENTIFY, 2019).

Na Figura 3, é apresentado a tela de personalização do evento, facilitando ao usuário publicar seu evento da maneira desejada.

Figura 3 - Eventify

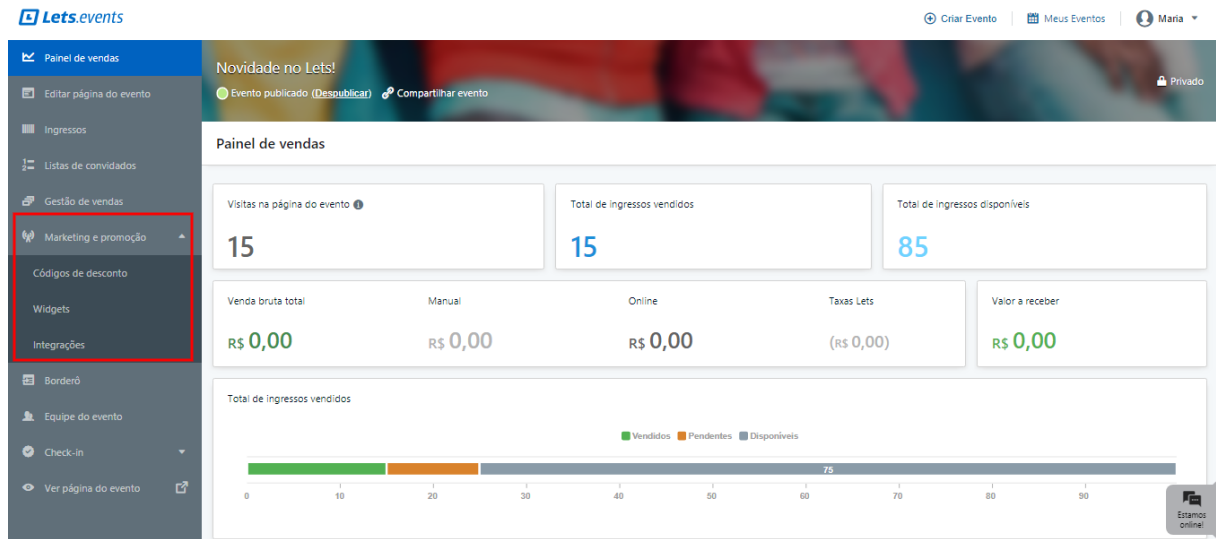


Fonte: Eventify (2019)

Lets.events é uma *startup* com o objetivo de oferecer soluções para o gerenciamento de listas de convidados e venda de ingressos *on-line*, visando economizar o tempo de organizadores de eventos. A plataforma tem uma parceria com a HackTown que é um evento de tecnologia e inovação que acontece em Minas Gerais (LETS.EVENTS, 2019).

A Figura 4 apresenta o menu de vendas do evento publicado pelo usuário.

Figura 4 - Lets.events



Fonte: Lets.events (2019)

3 MATERIAIS E MÉTODO

A ênfase deste capítulo está em reportar como o objetivo do trabalho foi atingido. Este capítulo é subdividido em duas seções, sendo uma para os materiais e outra para o método. A seção de materiais detalha os recursos utilizados durante a realização do trabalho e a seção do método descreve as etapas e abordagens adotadas para alcançar o objetivo proposto.

3.1 Materiais

No Quadro 1 são apresentadas as principais ferramentas utilizadas para as atividades de modelagem e implementação do sistema. Os materiais elencam as ferramentas e tecnologias utilizadas na modelagem, banco de dados, linguagem e ambientes para programação.

Quadro 1 - Lista de ferramentas e tecnologias

Ferramenta / Tecnologia	Versão	Finalidade
Android	9.0	Sistema Operacional
Android Studio	3.5	<i>Integrated Development Environment</i> (IDE) de desenvolvimento
Java	11	Linguagem de programação
FireBase	15	Banco de dados
Visual Paradigm	15	<i>Unified Modeling Language</i> (UML)

Fonte: Autoria própria (2023)

A seguir serão descritas as tecnologias utilizadas no desenvolvimento do trabalho.

3.1.1 Android Studio

O Android Studio foi lançado pelo Google para o desenvolvimento de aplicativos para o sistema operacional Android. Oferece um editor de código inteligente, que ajuda a trabalhar mais rápido e aumentar a produtividade oferecendo preenchimento automático de código avançado, refatoração e análise de código (GOOGLE, 2019).

A IDE possui um emulador para testes dos aplicativos, o qual simula um dispositivo real e facilita a realização de testes e a execução do aplicativo desenvolvido. Outra funcionalidade da IDE é a opção de *preview* de *layout*, em que é possível verificar como está ficando todo o *layout* da tela desenvolvida.

3.1.2 Linguagem de Programação Java

Java é uma linguagem de programação interpretada orientada a objetos. Um ponto positivo do Java é a quantidade de bibliotecas gratuitas para realizar os mais diversos trabalhos, tais como relatórios, gráficos, sistemas de busca, geração de código de barra, manipulação de *eXtensible Markup Language* (XML), tocadores de vídeo, manipuladores de texto, persistência transparente, impressão, entre outros. A linguagem de programação Java é utilizada para desenvolvimentos de aplicação *desktop*, *web* e *móveis* (CAELUM, 2019).

3.1.3 Firebase

O Firebase “oferece suporte ao compartilhamento de recursos, como banco de dados, configurações e notificações entre aplicativos multiplataformas” (MORONEY, 2017). O Firebase foi desenvolvido com o intuito de facilitar o desenvolvimento de aplicativos (Android e iOS) e de aplicações web. Ele disponibiliza todos os recursos de *back-end* necessários para o desenvolvedor em um único lugar.

A ferramenta também oferece diversas funcionalidades, como, banco de dados NoSQL utilizando JSON para armazenamento de dados, suporte para autenticação de usuários via email, Facebook, GitHub, Google Sign-In e Twitter, relatórios de erros, e *insights* sobre o aplicativo.

3.1.4 Visual Paradigm

Visual Paradigm é uma ferramenta de software projetada para equipes de desenvolvimento de software para modelar sistema de informações empresariais e gerenciar processos de desenvolvimento. É uma ferramenta gratuita voltada para a modelagem de diagramas *Unified Modeling Language* (UML).

A ferramenta Visual Paradigm Community Edition, apresenta suporte para os 13 tipos de diagramas UML oferecidos, são eles: diagrama de classe, caso de uso,

sequência, comunicação, máquina de estado, atividade, componentes, implantação, pacote, objeto, estrutura composta, perfil, tempo e visão geral da interação.

3.2 Método

O método utilizado neste trabalho foi o processo sequencial linear de Pressman e Maxim (2016). Esse modelo foi adotado porque os requisitos estão bem definidos e são estáveis. Nesse modelo, as atividades e as tarefas ocorrem sequencialmente, com diretrizes de progresso definidas começando com a especificação dos requisitos do cliente e avançando pelas fases de modelagem, construção e disponibilização e terminando no suporte contínuo do software concluído (PRESSMAN, MAXIM, 2016, p. 42). Esse processo define as fases de comunicação, planejamento, modelagem, construção e entrega.

Na fase de comunicação ocorre o início do projeto com o levantamento de requisitos. O levantamento dos requisitos iniciou tendo como objetivo desenvolver um aplicativo móvel que possibilite a divulgação de eventos, acompanhamento e notificação dos mesmos aos usuários, e também visa proporcionar ao DCE o controle e gerenciamento dos eventos publicados.

Para definir alguns requisitos mais específicos foi realizada uma reunião entre o DCE juntamente com o professor orientador deste trabalho. Após esta etapa, foram definidos os requisitos funcionais e não funcionais do sistema. Os requisitos funcionais referem-se as funcionalidades do sistema, como o sistema irá se comportar em relação a à interação do usuário. A etapa do planejamento consiste na definição do cronograma para a realização de cada uma das etapas. Na etapa de modelagem foi realizada a análise do projeto com a construção dos diagramas de caso de uso. A partir desse documento foram geradas informações para a modelagem do banco de dados por meio do diagrama de entidade e relacionamentos.

A etapa inicial de implementação teve como foco a criação das coleções no Firestore Database, as classes *model* do projeto e a configuração para se comunicar com a plataforma do Firebase.

Após o projeto se comunicar com o Firebase, foi desenvolvido as classes de *Data Transfer Object* (DTO), usado para transferir dados de um local a outro na aplicação e também com o objetivo de realizar a comunicação da parte de acesso aos

dados à parte visual, interpretando as requisições trazidas pela parte visual para realizar as ações mapeadas para essas requisições.

A etapa seguinte foi implementar as telas de cadastro e autenticação do aplicativo com o auxílio do Firebase Authentication que faz o reconhecimento de identidade de usuários, permitindo que usuários possam fazer *login*. Seguindo o desenvolvimento, foi realizada a implementação da parte visual de todos os cadastros, fazendo com que sejam geradas requisições para o servidor (Firebase) incluir, editar ou excluir dados dos cadastros.

A etapa seguinte foi realizar as consultas que trazem informações sobre os eventos, informativos, níveis de acesso de usuários e outros. Todas as consultas poderão ser filtradas.

4 RESULTADOS

Este capítulo apresenta o escopo, a modelagem e o resultado obtido com o desenvolvimento do sistema para divulgação de eventos.

4.1 Escopo do Sistema

O aplicativo visa auxiliar o DCE da Universidade Tecnológica Federal do Paraná (UTFPR) a gerenciar eventos realizados na universidade, desde a divulgação e demonstração de interesses dos usuários até a geração de dados para o aprimoramento da realização de eventos, além de enviar notificações aos usuários sobre os eventos que estão marcados no dia. O acesso dos usuários e dos administradores deverá ser feito por meio de um *smartphone* com sistema operacional Android. O acesso é dividido em três perfis, sendo eles: usuários interessados nos eventos, usuários responsáveis pelos eventos, e administradores.

Os usuários interessados possuem acesso aos eventos cadastrados no aplicativo, podendo filtrá-los por data, idade mínima para entrada, público-alvo (no caso de eventos para acadêmicos ou para a comunidade em geral), tipo do evento (curso, palestra, oficina, *show*), cobrança para acesso (se gratuito ou pago) e área (música, teatro, área acadêmica, etc.). Podem demonstrar interesse em participar de eventos, curtir e comentar sobre o evento. Deve ser possível ao usuário também escolher as áreas de interesse para geração de métricas aos administradores em relação a quais áreas possuem mais interesses dos usuários.

Os usuários responsáveis pelos eventos deverão preencher um formulário de submissão que compreende as informações competentes a eventos, além de ter acesso a todas as funcionalidades de um usuário comum.

Os administradores são responsáveis por aceitar ou recusar um evento que foi submetido por um usuário que preencheu um formulário de submissão, selecionando o que deve ou não ser publicado no aplicativo, além de fazer as alterações cabíveis das informações anunciadas pelo usuário.

Os usuários administradores também possuem acesso às informações gerenciais do sistema, como o número total de cadastros de usuários com o perfil de interessados e com o perfil para submissão de eventos, lista de eventos passados e

eventos correntes, usuários que demonstraram interesse em um evento, áreas que os usuários apresentam mais interesse. Esses dados serão analisados posteriormente pelo usuário administrador com o intuito de aprimorar a ferramenta e aumentar o aproveitamento dos usuários.

O aplicativo também comporta a publicação de materiais relevantes ao público em geral como artigos, matérias informativas, exposição de projetos acadêmicos, entre outros.

4.2 Modelagem do Sistema

Os quadros numerados de 2 a 7 apresentam a listagem e descrição dos requisitos funcionais e os requisitos não-funcionais do sistema, representados pelas siglas RF e RNF, respectivamente.

Quadro 2 - Registro de Eventos

RF1 – Registro de Eventos	
Descrição: O sistema deve permitir que o usuário responsável cadastre o evento para o administrador validar.	
Requisitos Não-Funcionais	
Nome	Restrição
RNF 1.1 – Preencher campos obrigatórios	O usuário deve preencher todos os campos obrigatórios. Os campos nome do evento, tipo, público-alvo, área, data, idade mínima para entrada são obrigatórios.
RNF 1.2 – Validação	Todos os campos serão validados se preenchidos corretamente. Alertas visuais na tela serão exibidos enquanto houver erros de dados inseridos.

Fonte: Autoria própria (2023)

Quadro 3 - Gerenciar Evento

RF2 – Gerenciar Evento	
Descrição: O sistema deve permitir que o administrador avalie se o evento deve ou não ser publicado.	
Requisitos Não-Funcionais	
Nome	Restrição
RNF 2.1 – Editar Formulário	O administrador poderá alterar algum dado do formulário dependendo de sua avaliação.
RNF 2.2 – Atualizar evento	O administrador poderá atualizar o evento e publicar no aplicativo.

RNF 2.3 – Pesquisa de evento	O administrador poderá pesquisar eventos ainda não publicados, filtrando por tipo, público-alvo, área, data, idade mínima para entrada e nome do evento.
------------------------------	--

Fonte: A autoria própria (2023)

Quadro 4 - Consultar Evento

RF3 – Consultar Evento	
Descrição: Os usuários interessados terão acesso aos eventos publicados na plataforma.	
Requisitos Não-Funcionais	
Nome	Restrição
RNF 3.1 – Pesquisa de evento	O usuário poderá filtrar a pesquisa por data, idade mínima para entrada, público-alvo, tipo do evento e cobrança para acesso e área.

Fonte: A autoria própria (2023)

Quadro 5 - Levantamento de Dados

RF4 – Levantamento de dados	
Descrição: O sistema deverá apresentar informações estatísticas dos usuários, que serão analisadas, posteriormente, com o intuito de aprimorar a ferramenta e aumentar o aproveitamento dos usuários.	
Requisitos Não-Funcionais	
Nome	Restrição
RNF 4.1 – Permissões	Somente administradores poderão visualizar as estatísticas.

Fonte: A autoria própria (2023)

Quadro 6 - Enviar Notificação

RF5 – Enviar Notificação	
Descrição: O sistema deverá notificar os usuários sobre a data do evento.	
Requisitos Não-Funcionais	
Nome	Restrição
RNF 5.1 – Tempo	Será enviado diariamente uma notificação para cada evento marcado na data do envio.

Fonte: A autoria própria (2023)

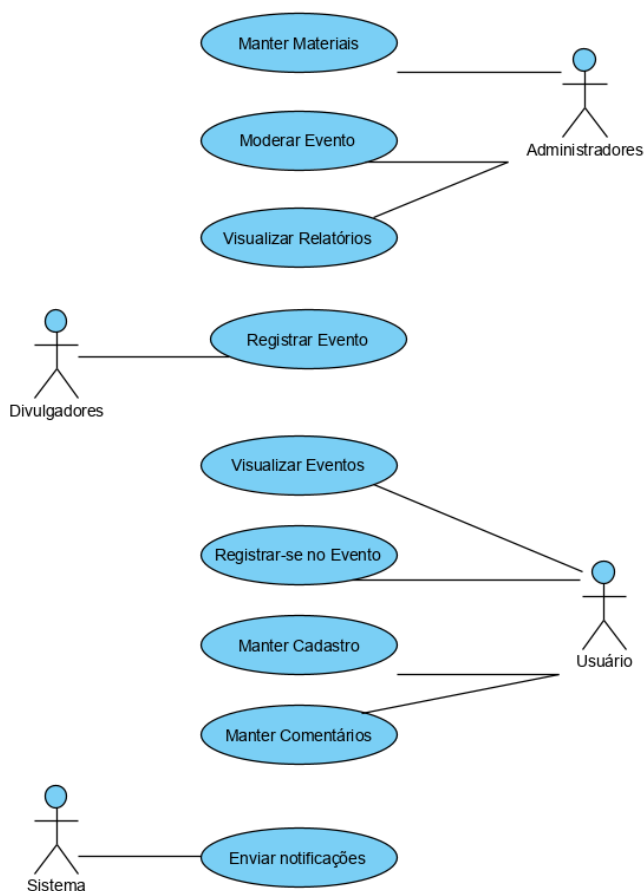
Quadro 7 - Publicação de Materiais

RF6 – Publicação de Materiais	
Descrição: O sistema deverá possibilitar a publicação de materiais relevantes ao público em geral como artigos, matérias informativas, exposição de projetos acadêmicos, que possam beneficiar o desenvolvimento pessoal dos usuários.	
Requisitos Não-Funcionais	
Nome	Restrição
RNF 6.1 – Preencher campos obrigatórios	O usuário deve preencher todos os campos obrigatórios. Os campos título do informativo e tipo são obrigatórios.
RNF 6.2 – Validação	Todos os campos serão validados se preenchidos corretamente. Alertas visuais na tela serão exibidos enquanto houver erros de dados inseridos.

Fonte: Autoria própria (2023)

A Figura 5 apresenta o diagrama de casos de uso definidos para o sistema. Esse diagrama demonstra as funcionalidades essenciais do sistema que são realizadas por seus atores representados pelo sistema, usuário, divulgadores ou administradores.

Figura 5 - Caso de uso



Fonte: Autoria própria (2019)

O administrador é responsável por aceitar ou recusar um evento que foi submetido por um divulgador que preencheu um formulário de submissão, selecionando o que deve ou não ser publicado no aplicativo. Além disso, poderá alterar informações anunciadas pelo divulgador do evento, também poderá publicar materiais relevantes ao público em geral como artigos, matérias informativas, exposição de projetos acadêmicos, e poderá visualizar relatórios que serão analisados com o intuito de aprimorar a ferramenta e aumentar o aproveitamento dos usuários.

O divulgador será responsável por registrar eventos por meio de um formulário que compreende todas as informações referentes a eventos.

O usuário terá acesso aos eventos cadastrados na plataforma, podendo filtrá-los por data, idade mínima para entrada, público-alvo, tipo do evento, cobrança para acesso e área. Podem demonstrar interesse em participar, comentar e curtir o evento. O usuário também poderá escolher as áreas que possui mais interesse para ser gerada métricas.

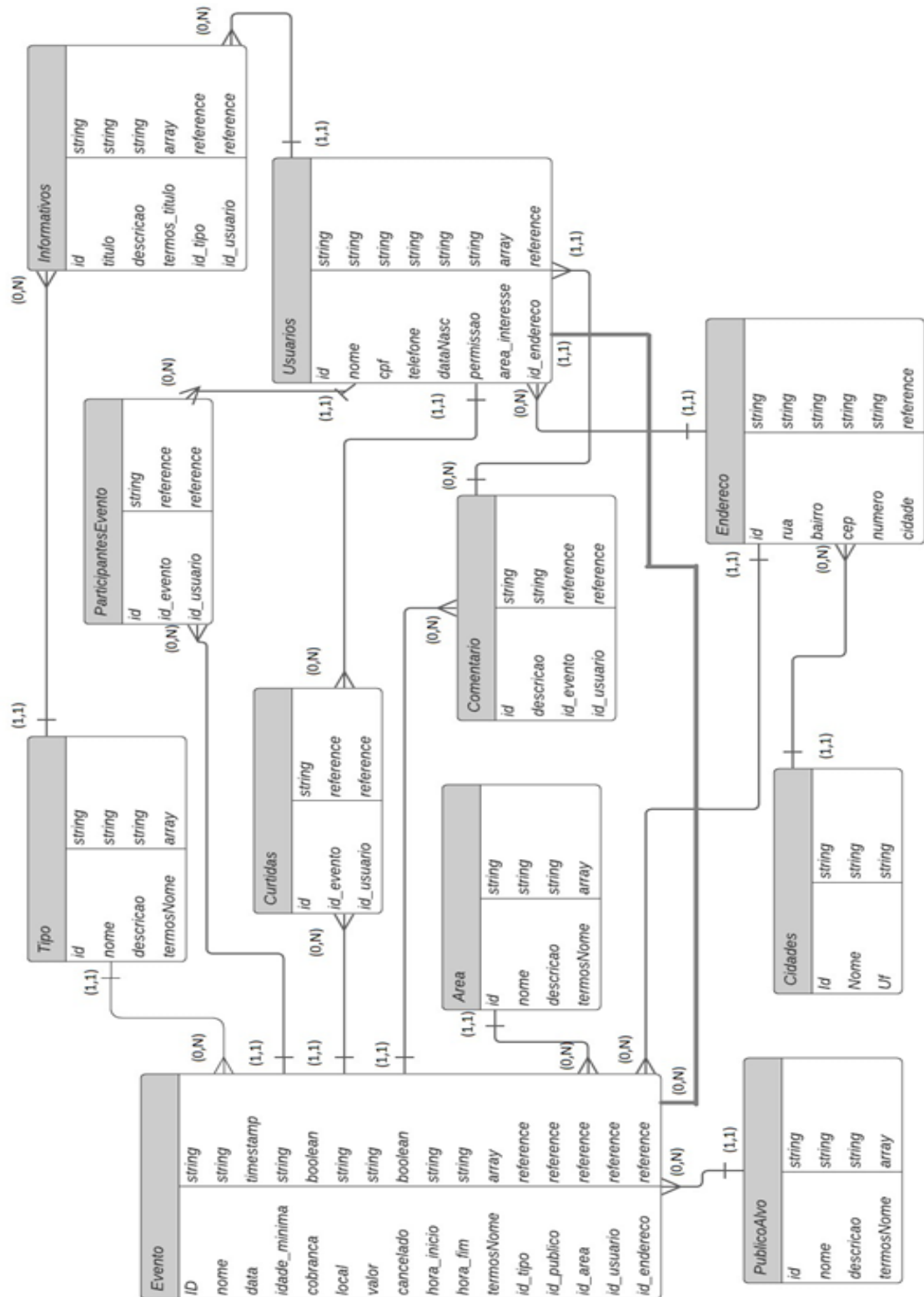
O sistema deverá enviar diariamente notificações para usuários sobre os eventos marcados no dia, com o objetivo de lembrar sobre a data do evento que o usuário estará cadastrado.

A Figura 6 apresenta o diagrama de entidade-relacionamento do sistema. As principais tabelas são Evento e Usuario. A tabela Evento possui os campos nome (nome do evento), data (realização do evento), idade_mínima (idade mínima para entrada), cobranca (se possui cobrança ou não), local (local do evento), cancelado (se o evento foi cancelado), valor (se o evento possuir cobrança, essa coluna armazenará o valor do ingresso), hora_inicio (a hora do início do evento), hora_fim (hora do final do evento), termosNome (array de palavras existentes no nome do evento) possui relações com outras tabelas como a de publico_alvo, que seria para qual público este evento pertence.

Ainda na Figura 6, a tabela Area representa a área temática, ou seja, se o evento é musical, teatral ou outro. Possui relação com a tabela Comentario que armazena os comentários feitos pelos usuários sobre o evento. A tabela Curtidas tem o objetivo de armazenar as curtidas que o usuário deu em algum evento. A tabela Tipo permite o cadastro de variados tipos de eventos sendo cursos, palestras, entre outros. A tabela Participante Evento armazena os usuários que estão cadastrados em determinados eventos. Possui relação com a tabela de Endereco (para armazenar o

endereço do evento). A tabela Usuario possui os campos nome (nome do usuário), CPF, telefone, data de nascimento, permissao (o perfil de acesso do usuário), faz relação com a tabela Area (área de interesse do usuário para gerar métricas) e com a tabela Endereco (que possui os campos relacionados ao endereço do usuário). Na modelagem, foi criado uma tabela Informativos, que armazenará dados de materiais informativos, artigos, exposição de projetos acadêmicos, entre outros, deverá ser cadastrado o título do material, descrição, tipo e armazenará o usuário que publicou.

Figura 6 - Diagrama de Entidade-Relacionamento



Fonte: Autoria própria (2023)

4.3 Apresentação do Sistema

Nesta seção serão apresentadas as principais funcionalidades disponíveis no sistema desenvolvido.

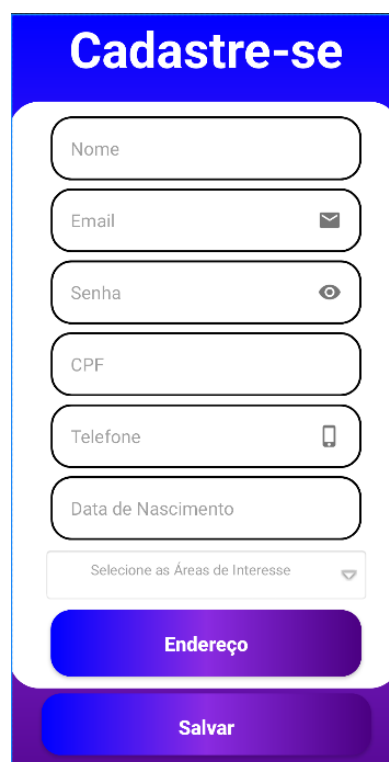
Na Figura 7 é apresentada a página de autenticação do sistema. Para autenticar-se é necessário informar o email e senha cadastrados.

Figura 7 - Tela de Autenticação

A tela de autenticação possui um fundo com gradiente de azul para roxo. No topo, há um ícone de mapa com pinos. Abaixo, há dois campos de entrada: 'Email' com um ícone de envelope e 'Senha' com um ícone de olho. Abaixo dos campos, há o link 'Esqueceu a Senha?' e um botão azul 'Entrar'. No rodapé, há um link 'Registrar-se'.

Fonte: Autoria própria (2023)

Figura 8 - Cadastro de Usuários

A tela de cadastro possui um fundo com gradiente de azul para roxo. No topo, há o título 'Cadastre-se'. Abaixo, há uma série de campos de entrada: 'Nome', 'Email' com ícone de envelope, 'Senha' com ícone de olho, 'CPF', 'Telefone' com ícone de telefone, 'Data de Nascimento' e 'Selecione as Áreas de Interesse' com uma seta para baixo. No rodapé, há dois botões: 'Endereço' e 'Salvar'.

Fonte: Autoria própria (2023)

Caso o usuário não tenha cadastro, é possível realizá-lo clicando no *link* Registrar-se para preencher os campos do formulário apresentados na Figura 8.

Ao realizar a autenticação, o usuário é direcionado para a página inicial do aplicativo, na qual é apresentado atalhos para as funcionalidades do sistema. Os usuários que possuem permissão de administrador, terão acesso às opções apresentadas na Figura 9.

Usuários com permissão de divulgadores de eventos ou usuários comuns, possuem acesso limitado para cadastrar e consultar eventos, além de visualizar o perfil, conforme apresentado na Figura 10.

Figura 9 - Tela Inicial para Administradores



Fonte: Autoria própria (2023)

Figura 10 - Tela Inicial para os Usuários



Fonte: Autoria própria (2023)

A Figura 11 demonstra como realizar um cadastro de evento. Nessa tela são apresentados os componentes que representam os dados do registro, sendo eles do tipo texto, caixa de seleção, e para informações como público-alvo, tipo do evento e área, é utilizado um componente do tipo *spinner*. Os botões ao lado de cada *spinner* servem para cadastro de um novo item conforme necessidade do usuário. O botão de endereço abre uma tela com as informações necessárias de preenchimento sobre a localização do evento. Ao clicar em salvar, as informações serão validadas e, caso o usuário possua acesso de publicador, o evento ficará aguardando os administradores realizarem a publicação. Caso o usuário que cadastrar o evento possua permissão de administrador, o evento será publicado automaticamente.

Figura 11 - Cadastro de Eventos

Fonte: Autoria própria (2023)

Figura 12 - Consulta de Eventos

Nome	Data	Idade Mínima	Cobrança	Valor	Local	Cancelado?	Hora do Início	Hora do Fim
Curso de Android 1	16/05/2023	1	Com Cobrança	3	UTFPR	Não	18:40	23:00
Curso de Java	02/05/2023	18	Com Cobrança	1	pato branco	Não	12:00	17:00
Palestra de Segurança	16/05/2023	18	Gratuito		UTFPR	Não	20:30	22:00

Fonte: Autoria própria (2023)

A Figura 12 apresenta a lista de eventos cadastrados no aplicativo. Nela são apresentadas as informações principais sobre o evento. É disponibilizado também alguns filtros relacionados ao evento e um campo para pesquisa. A lista possui condições específicas por nível de acesso, por exemplo, administradores podem ver todos os eventos cadastrados. Já usuários que possuem a permissão de publicar eventos podem ver seus eventos que ainda não foram publicados e os demais eventos já publicados. Por fim, usuários comuns podem ver eventos que já foram publicados e que ainda não aconteceram. Para visualizar mais informações sobre o evento é necessário clicar sobre o evento.

Ao clicar em algum evento da lista, é aberta uma tela com todas as informações sobre o evento, a quantidade de pessoas que estão participando e opções que

interagem entre usuários e o evento, conforme mostrado na Figura 13. A opção de editar evento estará sempre visível para administradores, já para o usuário que criou ela estará disponível desde que o evento ainda não esteja publicado. Administradores possuem o privilégio de remover a publicação de um evento ou realizar a publicação. As opções de comentar, visualizar comentários, curtir ou remover a curtida, participar ou sair do evento são visíveis para todos os tipos de usuários.

Figura 13 - Detalhes do Evento



Fonte: Autoria própria (2023)

Usuários administradores possuem acesso a tela de *Dashboard* que apresenta informações relacionadas a quantidade de usuários por nível de acesso, e uma lista com a quantidade de usuários que demonstraram interesse em cada área de eventos, como pode ser observado na Figura 14.

Figura 14 - Tela de Dashboard

Fonte: Autoria própria (2023)

A Figura 15 representa o menu Permissão Usuário localizado na tela inicial do sistema, esta funcionalidade contém o perfil de acesso que o usuário selecionado possui, permitindo que os administradores alterem o perfil de acesso de qualquer usuário cadastrado no aplicativo.

Figura 15 - Permissão do Usuário

A captura de tela mostra uma interface de usuário com o título 'Permissão do Usuário' em um cabeçalho azul. Abaixo do título, há um campo de texto arredondado contendo o nome 'Alexandre Dotti'. Logo abaixo, há uma barra de seleção com o texto 'USER' em azul. Na base da interface, há um botão retangular com o texto 'SALVAR' em branco sobre um fundo escuro. O fundo da tela apresenta um gradiente de cor, passando de azul no topo para roxo no fundo.

Fonte: Autoria própria (2023)

O cadastro de informativos é disponibilizado para qualquer tipo de usuário, são apresentados os componentes que representam os dados do registro, sendo eles do tipo texto, e para o tipo do informativo, é utilizado o componente *spinner*. O botão ao lado do *spinner* serve para cadastro de um novo item conforme necessidade do usuário, como pode ser observado na Figura 16.

Figura 16 - Cadastro de Informativos

Fonte: Autoria própria (2023)

4.4 Implementação do Sistema

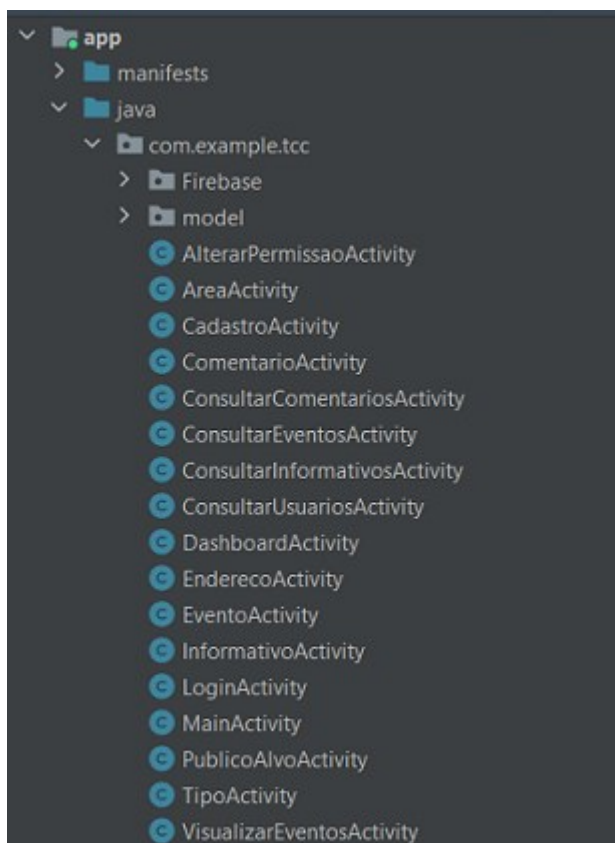
Para iniciar o desenvolvimento do aplicativo, foi definida uma estrutura de pacotes para manter o projeto organizado e conciso. Na Figura 17 é possível visualizar a estrutura utilizada durante o desenvolvimento do projeto, em que foi utilizado a linguagem de programação Java.

Os principais pacotes apresentados são:

- a) Model: contém todas as classes de persistência do sistema.
- b) Firebase: responsável pelo gerenciamento das notificações enviadas pelo Firebase.

São apresentadas também classes que possuem o papel de fazer o gerenciamento das telas do aplicativo.

Figura 17 - Estrutura dos Pacotes do Aplicativo



Fonte: Autoria própria (2023)

Na sequência do texto serão apresentados os principais métodos desenvolvidos para atender os requisitos do aplicativo. Na Listagem 1 é apresentado o método responsável por cadastrar ou alterar um evento. Esse método possui uma *View* como parâmetro, a *view* é um objeto da interface gráfica que será apresentado visualmente para o usuário, responsável por desenhar o componente na tela assim como responder aos seus eventos. O objeto “b” é do tipo *Bundle*, uma classe utilizada para transferir dados entre as telas do aplicativo.

No exemplo da Listagem 1 o objeto “b” armazena os dados do evento em caso de edição, com base neste objeto é possível definir se é para ser criado um evento ou apenas editar um evento já existente. Quando o usuário alterar um evento, é criado um objeto do tipo *DocumentReference* responsável por buscar a referência cadastrada na coleção Evento no Firebase, após possuir o código do evento a ser alterado é criado uma lista de chave-valor para armazenar todos os campos da coleção Evento, por fim é utilizando o método *update* do banco de dados Firebase passando a lista como parâmetro para ser atualizado. No mesmo método, caso o

usuário esteja cadastrando um evento é utilizado o método *set* do Firebase para criar um registro, passando o objeto evento que é instanciado a partir da classe com todos os atributos relacionados à um evento, é possível saber se o cadastro foi realizado com o método subscrito *onSuccess*, que possui o resultado do banco de dados. Finalizando com uma mensagem para o usuário ter a certeza de que o cadastro foi realizado, e voltando para a tela inicial do sistema.

Listagem 1 - Método Cadastrar/Alterar Evento

```
private void cadastrarEvento(View view) {
    if (b != null){
        DocumentReference dr = db.collection("Evento").document(eventoID);
        Map<String, Object> ev = new HashMap<>();
        ev.put("nome", evento.getNome());
        ev.put("data", evento.getData());
        ev.put("idade_minima", evento.getIdade_minima());
        ev.put("cobranca", evento.getCobranca());
        ev.put("local", evento.getLocal());
        ev.put("cancelado", evento.getCancelado());
        ev.put("valor", evento.getValor());
        ev.put("hora_inicio", evento.getHora_inicio());
        ev.put("hora_fim", evento.getHora_fim());
        ev.put("publicado", evento.getPublicado());
        ev.put("id_tipo", evento.getId_tipo());
        ev.put("id_publico", evento.getId_publico());
        ev.put("id_area", evento.getId_area());
        ev.put("termosNome", evento.getTermosNome());
        dr.update(ev);
        Intent intent = new Intent();
        setResult(RESULT_OK, intent);
    } else {
        DocumentReference dr = db.collection("Evento").document();
        dr.set(evento).addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Snackbar snackbar = Snackbar.make(view, mensagens[1],
                Snackbar.LENGTH_SHORT);
                snackbar.setBackgroundTint(Color.WHITE);
                snackbar.setTextColor(Color.BLACK);
                snackbar.show();
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Log.d("db_error", "Erro ao salvar os dados" + e.toString());
            }
        });
    }
    finish();
}
```

Fonte: Autoria própria (2023)

Na Listagem 2 é apresentado o código que é utilizado para realizar a consulta de eventos para administradores sem a utilização de filtros. O resultado obtido é atribuído a uma lista de *arrays* e essa lista é vinculada ao componente *ListView* que é um componente da camada visual da aplicação.

Listagem 2 - Consulta de Eventos

```

db.collection("Evento")
    .orderBy("nome")
    .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
@Override
public void onComplete(@NonNull Task<QuerySnapshot> task) {
    if (task.isSuccessful()) {
        for (QueryDocumentSnapshot document : task.getResult()) {
            arrayList.add(new Evento(document.getId(), document.getString("nome")
                , document.getDate("data"), document.getString("idade_minima")
                , document.getBoolean("cobranca"), document.getString("local")
                , document.getBoolean("cancelado"), document.getString("valor")
                , document.getString("hora_inicio"), document.getString("hora_fim")
                , document.getBoolean("publicado"),
document.getDocumentReference("id_tipo")
                , document.getDocumentReference("id_publico"),
document.getDocumentReference("id_area")
                , document.getDocumentReference("id_usuario"),
document.getDocumentReference("id_endereco")));
        }
        ArrayAdapter adapter_evento = new ArrayAdapter(ConsultarEventosActivity.this,
android.R.layout.simple_list_item_1, arrayList);
        list.setAdapter(adapter_evento);
    } else {
        Log.d("", "Error getting documents: ", task.getException());
    }
}
});

```

Fonte: Autoria própria (2023)

Uma das métricas apresentadas para os administradores do aplicativo é a consulta de áreas cadastradas e a quantidade de usuários que demonstraram interesse em cada área, conforme demonstra a Listagem 3. Para realizar a consulta ao banco de dados é utilizado o método *collection* que busca a coleção passada por parâmetro, é também definido uma ordenação pelo nome das áreas obtidas. O resultado é atribuído a uma lista de *arrays* e essa lista é vinculada ao componente *ListView* que está no menu Dashboard localizado na tela inicial do aplicativo.

Listagem 3 - Consulta de Áreas

```

private void getAreas() {
    arrayList.clear();

    db.collection("Area")
        .orderBy("nome")
        .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
            @Override
            public void onComplete(@NonNull Task<QuerySnapshot> task) {
                if (task.isSuccessful()) {
                    for (QueryDocumentSnapshot document : task.getResult()) {

                        db.collection("Usuarios")
                            .whereArrayContains("area_interesse", document.getId())
                            .get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
                                @Override
                                public void onComplete(@NonNull Task<QuerySnapshot> task) {
                                    if (task.isSuccessful()) {
                                        arrayList.add(new AreaDashboardDTO(document.getId(),
document.getString("nome"), task.getResult().size()));
                                        ArrayAdapter adapter_area = new ArrayAdapter(DashboardActivity.this,
android.R.layout.simple_list_item_1, arrayList);
                                        list.setAdapter(adapter_area);
                                    }
                                }
                            });
                    }
                } else {
                    Log.d("", "Error getting documents: ", task.getException());
                }
            }
        });
}
}

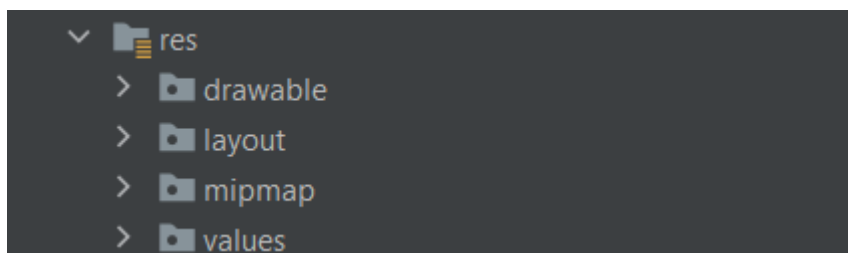
```

Fonte: Autoria própria (2023)

Na Figura 18 é possível observar a estrutura da camada visual do aplicativo. São arquivos no formato XML que contém a definição dos componentes utilizados nas telas. As principais pastas apresentadas são:

- a) Layout: contém o desenvolvimento dos layouts de todas as telas apresentadas no sistema.
- b) Drawable: possui ícones, definições de fundo para componentes e telas, e imagens utilizadas no aplicativo.
- c) Values: são arquivos com definições de estilos de botões, campos de edição de texto, criação de variáveis que são utilizadas no título das telas, e lista de filtros para eventos e informativos.

Figura 18 - Estrutura da Camada Visual



Fonte: Autoria própria (2023)

Na Listagem 4 está um trecho do código mostrando a utilização dos componentes mais importantes da tela de Cadastro de Informativos apresentada na Figura 16, o arquivo XML apresenta os componentes *TextView* utilizado para definir o título da tela, *EditText* é o componente responsável por receber os dados que serão utilizados para cadastrar um informativo. Para selecionar o tipo do informativo é implementado o componente *Spinner*, e o componente responsável por executar o método que cadastra o informativo é o *Button*.

Listagem 4 - Cadastro de Informativos

```

<TextView
    android:id="@+id/title_informativo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="468dp"
    android:text="@string/title_informativo"
    android:textColor="@color/white"
    android:textSize="50sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/etTitulo"
    style="@style/Edit_Text"
    android:layout_width="354dp"
    android:layout_height="64dp"
    android:layout_marginStart="29dp"
    android:layout_marginTop="106dp"
    android:layout_marginEnd="28dp"
    android:layout_marginBottom="365dp"
    android:hint="Título"
    android:inputType="text"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
  
```

```

<Spinner
    android:id="@+id/spTipoInformativo"
    android:layout_width="303dp"
    android:layout_height="50dp"
    android:layout_marginStart="29dp"
    android:layout_marginTop="399dp"
    android:layout_marginEnd="79dp"
    android:layout_marginBottom="86dp"
    android:background="@drawable/background_spinner"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints" />

<Button
    android:id="@+id/btSalvar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="160dp"
    android:layout_marginTop="457dp"
    android:layout_marginEnd="80dp"
    android:text="Salvar"
    android:textColor="#FFFFFF"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.833" />

```

Fonte: Autoria própria (2023)

Para o envio das notificações conforme apresentado na Listagem 5, foi implementado um servidor utilizando *Spring Framework* e a anotação *Scheduled* para definir que o método *enviaNotificacoes* irá ser executado a cada 24 horas. A variável *listEvento* recebe um *Array* de Eventos que acontecerão no dia que está sendo feito a consulta no banco de dados, já a variável *listUsuarios* recebe um *Array* de usuários cadastrados no aplicativo. A variável *pushNotificationRequest* representa uma classe com os dados necessários para enviar uma notificação, sendo eles, título, mensagem e o *token* do usuário cadastrado no aplicativo. Após as declarações das variáveis é implementado laços de repetições para as listas de usuários e eventos, atribuindo os dados para a variável *pushNotificationRequest*, e por fim é implementado o método *createPostWithObject* responsável por enviar a requisição da notificação para o servidor do Firebase que irá disparar a notificação para o usuário que é representado no *token*.

Listagem 5 - Envio de Notificações

```

@Scheduled(fixedRate = 86400000)
public void enviaNotificacoes() {

    listEvento = consultaEvento();
    listUsuarios = consultaUsuario();

    if (listEvento.size() > 0 && listUsuarios.size() > 0) {
        PushNotificationRequest pushNotificationRequest = new PushNotificationRequest();

        for (int i = 0; i < listUsuarios.size(); i++) {
            for (int j = 0; j < listEvento.size(); j++) {
                pushNotificationRequest.setTitle("Evento : " + listEvento.get(j).getNome());
                pushNotificationRequest.setMessage("Início : "+listEvento.get(j).getHora_inicio()+"
hora(s)," +
                " até as : "+listEvento.get(j).getHora_fim()+" hora(s).\n"+
                listEvento.get(j).getLocal());
                pushNotificationRequest.setToken(listUsuarios.get(i).getToken());
                createPostWithObject(pushNotificationRequest);
            }
        }
    }
}

```

Fonte: Autoria própria (2023)

5 CONCLUSÃO

Neste trabalho foi desenvolvido um aplicativo para a gestão de eventos. O projeto foi implementado utilizando a linguagem de programação Java para Android, e a plataforma do Google Firebase para armazenar os dados do aplicativo, controlar a autenticação de usuários, apresentar métricas sobre a aplicação, mensurar o comportamento do usuário e enviar notificações.

Todos os objetivos propostos ao trabalho foram alcançados, sendo o principal proporcionar o Diretório Central dos Estudantes da Universidade Tecnológica Federal do Paraná, Campus Pato Branco a divulgar, acompanhar e controlar eventos. Algumas melhorias foram feitas no banco de dados apresentado no trabalho de conclusão de curso 1, visando melhorar o desempenho da aplicação. Tais como novos campos nas tabelas de Informativos, Tipo, Área, Público-Alvo, Evento, para auxiliar a consulta de informações para o usuário, e na tabela de Usuario um novo campo do tipo *array* para gravar as áreas de interesse de mais afinidade.

Por não trabalhar diretamente com as tecnologias usadas, o autor foi desafiado para conseguir a melhor implementação do projeto. Resultando em dúvidas durante o desenvolvimento das rotinas do sistema relacionadas ao uso de alguns componentes, e a plataforma do Google Firebase, entretanto essas dúvidas foram sanadas por meio de pesquisas na documentação das bibliotecas, fóruns de discussão e com a ajuda do orientador.

Com trabalhos futuros esse aplicativo poderá ser melhorado para ser utilizado além da comunidade acadêmica. Podendo ser criado diferentes controles sobre eventos, mais modelos de métricas, aumentar a capacidade do dashboard, implementar melhorias visuais e melhorar o gerenciador de notificações podendo ser enviado notificações de outras funcionalidades.

O Repositório com o código fonte da aplicação está disponível em:

<https://github.com/alexandredotti/tcc>

REFERÊNCIAS

BURGER, Luciana; OTAKA, Felipe. **O Cenário das Redes Sociais no Brasil**. Brasil, 2018. Disponível em:

https://www.comscore.com/content/download/42824/2441584/file/WBNR_Cen__rio_Redres_Sociais_Brasil_FINAL_JUN2018-1.pdf. Acesso em 21 nov. 2019.

CAELUM. **Java e Orientação a Objetos**. Disponível em:

<http://www.inf.ufrgs.br/~tsrodrigues/utilidades/java.pdf>. Acesso em: 06 nov. 2019.

CARDOSO, Danilo; PINTO, Maria. **AS REDES SOCIAIS COMO FERRAMENTAS DE DIVULGAÇÃO DE EVENTOS**. Periódico Eletrônico da FATEC-São Caetano do Sul, São Caetano do Sul, v. 1, n. 5, Out/Dez 2011, p. 30 a 40. Disponível em: <http://fatecsaocaetano.edu.br/fascitech/index.php/fascitech/article/download/45/44>. Acesso em: 21 nov. 2019.

Cury, Cecília. **Por dentro da Sympla**. Disponível em: <https://blog.sympla.com.br/por-dentro-da-sympla/>. Acesso em: 20 nov. 2019.

DEGENNE, Alain e FORSÉ, Michel. **Introducing Social Networks**. London: Sage, 1999.

EVENTIFY. **Eventify**. Disponível em: <https://eventify.io/>. Acesso em 20 nov. 2019.

FERNANDES, Luís. **Redes Sociais Online e Educação: Contributo do Facebook no Contexto das Comunidades Virtuais de Aprendentes**. 2011. Disponível em: http://www.trmef.lfernandes.info/ensaio_TRMEF.pdf. Acesso em: 21 nov. 2019.

GOOGLE. **Android Studio**. Disponível em:

<https://developer.android.com/studio/intro>. Acesso em: 20 out. 2019.

LETS.EVENTS. **Lets.events**. Disponível em: <https://lets.events/pt/#>. Acesso em: 21 nov. 2019.

MEIRELLES, Fernando S. **30ª Pesquisa Anual do FGVcia da FGV/EAESP**. São Paulo: FGV, 2019. Disponível em:

https://eaesp.fgv.br/sites/eaesp.fgv.br/files/noticias2019fgvcia_2019.pdf. Acesso em: 05 set. 2019.

MINHOTO, Paula. **A utilização do Facebook como suporte à aprendizagem da biologia: estudo de caso numa turma do 12º ano**. Bragança: Escola Superior de Educação. Dissertação de Mestrado em Ensino das Ciências, 2012. Disponível em: https://bibliotecadigital.ipb.pt/bitstream/10198/6864/1/PaulaMinhoto_22696.pdf.

Acesso em 21 nov. 2019.

MORONEY L. **Firestore Cloud Messaging. O Guia Definitivo do Firestore**. Apress, Berkeley, Califórnia.

PERINOTTO, André; BORGES, Dilene e SOUSA, Julyana. **As redes sociais na divulgação de eventos e promovendo o turismo de eventos em Parnaíba-PI**. ECCOM, v. 7, n. 13, jan./jun. 2016. Disponível em: <http://unifatea.com.br/seer3/index.php/ECCOM/article/view/498/448>. Acesso em: 19 nov. 2019.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de software**. 8 ed. McGraw Hill Brasil, 2016.

RECUERO, Raquel. **Redes sociais na internet**. Porto Alegre. Sulina, 191 p. (Coleção Cibercultura), 2009. Disponível em: <http://www.ichca.ufal.br/graduacao/biblioteconomia/v1/wpcontent/uploads/redessocia isnainternetrecuero.pdf>. Acesso em: 21 nov. 2019.

RIBEIRO, José Carlos *et al.* **Conexão constante, privacidade e ambivalências nas dinâmicas interacionais mediadas por smartphones**. Brasil: Revista Crítica de Ciências Sociais 2019. Disponível em: <http://journals.openedition.org/rccs/9090>. Acesso em: 05 set. 2019.

SYMPLA. **Sympla**. Disponível em: <https://www.sympla.com.br/>. Acesso em: 21 nov. 2019.

VASCONCELOS, Maria *et al.* **AS TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO NA EDUCAÇÃO**. Brasil, 2010. Disponível em: <http://www.ice.edu.br/TNX/storage/webdisco/2011/02/11/outros/91e426c5bf41e7974ff11a1faa72f647.pdf>. Acesso em: 21 nov. 2019.

WASSERMAN, Stanley e FAUST, Katherine. **Social Network Analysis. Methods and Applications**. Cambridge, UK: Cambridge University Press, 1994.