UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

ANDRÉ DUARTE ANTUNES

APLICATIVO DE DIVULGAÇÃO DE DEFESAS DE TESES E APRESENTAÇÕES DE TCC

ALEGRE - ES
2018

ANDRÉ DUARTE ANTUNES

APLICATIVO DE DIVULGAÇÃO DE DEFESAS DE TESES E APRESENTAÇÕES DE TCC

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Prof. Dr. Antonio Almeida de Barros Junior.

ALEGRE - ES
2018

ANDRÉ DUARTE ANTUNES

APLICATIVO DE DIVULGAÇÃO DE DEFESAS DE TESES E APRESENTAÇÕES DE TCC

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 21 de novembro de 2018.

COMISSÃO EXAMINADORA

Prof. Dr. Antonio Almeida de Barros Junior
Universidade Federal do Espírito Santo
Orientador

Prof. M.Sc. Marcelo Otone Aguiar
Universidade Federal do Espírito Santo

Profª. M.Sc. Valeria Alves da Silva
Universidade Federal do Espírito Santo

DEDICATÓRIA

“Não tenha medo do fracasso. Tenha medo de não tentar.”

Roy T. Bennett

AGRADECIMENTOS

À minha família, por todo o esforço em me manter estudando todos esses anos, pelo apoio e incentivo.

À minha namorada Marcela Zonta que sempre esteve ao meu lado durante toda a minha jornada.

Ao meu orientador Antonio Almeida de Barros Junior pela oportunidade em desenvolver este projeto de monografia, pelas contribuições, disponibilidade em ajudar, paciência e direcionamentos.

À Universidade Federal do Espírito Santo e a todos os servidores e professores do Departamento de Computação, que contribuíram com meus conhecimentos.

À empresa Click Junior pela oportunidade de testar meus conhecimentos e me dar uma oportunidade de conhecer como funciona o mercado de trabalho.

Ao Núcleo de Tecnologia da UFES (NTI) pelo apoio e incentivos nos primeiros contatos direto à sistemas de computadores.

Aos meus amigos pelos momentos que levaremos por toda a vida.

Obrigado a todos.

RESUMO

Durante os cursos de graduação e pós-graduação é comum que os estudantes realizem apresentações de trabalhos desenvolvidos, sejam eles TCC, dissertações ou teses. Pelo tamanho dos campi e suas inúmeras apresentações ocorrendo ao mesmo tempo, um sério problema de falta comunicação ocorre entre alunos e universidade. Esse problema ocorre, principalmente, pela falta de meios de comunicação entre departamentos e até mesmo entre cursos. Muito embora informações úteis sejam pouco divulgadas, há uma tendência no uso de tecnologias onde pessoas compartilham a maior parte das informações do seu dia-dia, nos *smartphones*. Com o propósito de associar a falha na divulgação de informações ao meio de comunicação mais utilizado no mundo, o *smartphone*, este trabalho tem como objetivo facilitar o meio de comunicação de todos os campi da Universidade Federal do Espirito Santo, apresentando um aplicativo que possa ser capaz de utilização em diferentes plataformas de *smartphones*, com o principal foco de compartilhar onde serão realizadas as apresentações, informando sempre o que será apresentado e por quem será apresentado, o local, a data e horário que irá ocorrer.

Palavras chave: *Smartphone*; Aplicativo móvel; Apresentação TCC; Aplicativo de Gerenciamento.

ABSTRACT

During graduate and minors courses it is common for students to take the classes of developed works, like TCC, dissertations or theses. Because of the size of the campuses and their numerous presentations occurring at the same time, a serious problem of missing communication occurs between students and university. Caused mainly by the lack of communication means compared to the new technological trend that today share most of the new day-day information, the smartphones. With the purpose of associating the failure to divulge information to the most used communication medium in the word, the smartphone, this work aims to facilitate the communication medium of all Federal University of Espírito Santo campuses, presenting an application that can be capable of use in different smartphone platforms, with the main focus of sharing where the presentations will be made, always informing what will be presented and by whom will be presented, the location, date and time that will occurs.

Keywords: Smartphone; Mobile app; Presentation of TCC; Management application.

LISTA DE FIGURAS

Figura 1: Associação entre as camadas MVP. 16

Figura 2: Estrutura MVC. 17

Figura 3: Comunicação Aplicativo x Base de dados. 21

Figura 4: Diagrama de Estado. 22

Figura 5: Seleção dos campi. 23

Figura 6: Apresentação do curso e seus respectivos dados. 23

Figura 7: Diagrama de estados perfil Administrador. 24

Figura 8: Login Administrativo. 25

Figura 9: Área Administrativa. 25

Figura 10: Alterar, adicionar e/ou confirmar dados. 26

Figura 11: Campo da Model Apresentation. 29

Figura 12: Porcentagem dos questionados que têm conhecimento sobre as apresentações de TCC ou defesa de tese na faculdade. 31

Figura 13: Porcentagem dos questionados que possuem smartphone. 32

Figura 14: Porcentagem dos questionados que gostariam de utilizar o aplicativo. 32

Figura 15: Porcentagem de cada informação que os usuários gostariam que tivesse no aplicativo. Fonte: O autor, 2018. 33

Figura 16: Diagrama de Caso de Uso. 35

Figura 17: Diagrama de Atividade. 36

Figura 18: Diagrama de Classe. 37

Figura 19: Pseudocódigo CadastroLogin. 38

Figura 20: Pseudocódigo Getusuario. 38

Figura 21: Pseudocódigo CadastroApresentacao. 39

Figura 22: Pseudocódigo DeletarApresentacao. 40

LISTA DE TABELAS

Quadro 1: Linguagens nativas de sistemas operacionais. 18

Quadro 2: Aplicativos existentes no mercado. 30

Quadro 3: Requisitos funcionais. 34

Quadro 4: Dicionário de dados da classe Usuário..... 37

Quadro 5: Dicionário de dados da classe Apresentações..... 39

LISTA DE ABREVIATURAS E SIGLAS

App – Aplicativo

API – *Application Programing Interface*

HTTP – *HyperText Transfer Protocol*

MVC – *Model-View-Control*

MVP – *Model-View-Presenter*

REST – *Representation State Transfer*

TICs – Tecnologias de Informação e Comunicação

TIMs – Tecnologias de Informação Móveis

URI – *Unique Resource Identifiers*

SUMÁRIO

SUMÁRIO 2
1 INTRODUÇÃO **12**
1.1 O PROBLEMA E SUA IMPORTÂNCIA 13
1.2 OBJETIVOS 14
1.2.1 Objetivo geral 14
1.2.2 Objetivos específicos 14
1.3 ESTRUTURA DO TRABALHO 15
2 REVISÃO DE LITERATURA **16**
2.1 PADRÃO MVP E MVC 16
2.2 APLICAÇÕES MÓVEIS 17
2.2.1 *Framework* Phonegap 18
2.2.2 Interface Usuário 19
2.2.3 Adobe Phonegap Build 19
2.2.4 Uso de Banco de Dados 19
2.3 API 20
2.3.1 REST 21
2.3.1.1 Protocolo HTTP 21
3 METODOLOGIA **22**
3.1 PROTÓTIPOS 22
3.2 VALIDAÇÃO 26
3.3 ETAPA DO DESENVOLVIMENTO DO SISTEMA 27
3.3.1 Criação da Base de Dados 27
3.3.2 Aplicação Mobile 28
3.3.3 Geração da API 28
4 RESULTADOS **30**
4.1 PROJETO DA APLICAÇÃO 30
4.1.1 Requisitos 30
4.1.1.1 Requisitos Funcionais 34
4.1.2 Diagrama de Caso de Uso 34
4.1.3 Diagrama de Atividade 35
4.1.4 Diagrama de Classe 36
4.2 TESTES DE USABILIDADE DA APLICAÇÃO 40
5 CONCLUSÕES **42**

REFERÊNCIAS BIBLIOGRÁFICAS 43
ANEXOS 45
ANEXO A – Documento de *Feedbacks* 45
ANEXO B – Diagramas de Caso de Uso 46
ANEXO C – Fotos das telas tiradas do aplicativo 48

1 INTRODUÇÃO

Os computadores já fazem parte da nossa vida cotidiana. Quer tenha ou não, particularmente, algum conhecimento a respeito, o uso deste tipo de tecnologia é inevitável quando se trata de sacar dinheiro no caixa eletrônico, de fazer compras no supermercado, quando dirigimos um carro e ou, até mesmo quando usamos um *smartphone*.

Dados recentes da revista Exame (EXAME, 2017) mostram que, no Brasil, o acesso à internet por meio de celulares aumentou em 66% nos últimos 2 anos, principalmente na população jovem. Grande parte deste quantitativo pode-se relacionar aos estudantes universitários que utilizam este tipo de tecnologia para comunicação, acesso à internet e jogos. Além do mais, de acordo com o Estadão (ESTADÃO, 2017) estima-se que até o fim de 2017, no Brasil, haverá um *smartphone* por habitante.

Os chamados *smartphones*, computador de bolso, são capazes de substituir funções de uma série de equipamentos que hoje em dia são tidos como obsoletos. Mesmo que criados com os princípios básicos de realizar ligações, os celulares atingiram tamanha proporção que orientam nossas vidas e nos conectam com todo o mundo, podendo auxiliar uma melhor comunicação entre jovens estudantes e instituições de ensino (BAYER, 2016).

A comunicação no meio universitário encontra barreiras em decorrência da separação física dos departamentos e ou colegiados de curso. É comum que estudantes de um curso, frequentem apenas setores relacionados ao seu departamento, sem qualquer contato com outros setores do campus. Contudo, atividades ocorridas em um departamento podem ser de interesse de estudantes de diferentes departamentos.

Na Universidade Federal do Espírito Santo (UFES), por exemplo, não há um sistema eficiente e unificado de comunicação e divulgação das centenas de apresentações relacionadas aos trabalhos acadêmicos, tampouco das palestras interdisciplinares, motivacionais e de capacitação que ocorrem durante o semestre letivo. Essa divulgação, portanto, é feita de forma ineficiente e interna nos departamentos afins.
É importante ressaltar que os referidos trabalhos e palestras são de grande relevância para o desenvolvimento do senso crítico e analítico dos discentes. Neste contexto, realizar uma divulgação de forma efetiva e que alcance o maior número de estudantes

torna-se fundamental para o sucesso destes eventos.

Diante do que fora exposto, torna-se necessário a busca por mecanismos de comunicação que sejam capazes de difundir a informação no meio universitário de forma objetiva e moderna. Dentre os diversos meios de comunicação existentes, os aplicativos (Apps) utilizados para *smartphones* têm-se destacado como alternativa de difusão de informação.

O uso de aparatos de comunicação não se trata de uma novidade, bem como o fascínio que exercem e o status de “solução” atribuído a eles no contexto educacional. Com a ampliação e incrementos tecnológicos desses aparatos, reunidos às Tecnologias de Informação e Comunicação (TICs), expectativas e argumentos vêm sendo gerados em relação à apropriação dessas ferramentas para os mais diversos fins (GALVANI, 2015).

Na tentativa de proporcionar uma melhor comunicação entre departamentos e discentes, bem como uma melhor divulgação dos eventos institucionais, é proposto neste trabalho um aplicativo (*App*) que facilite e conecte todos os estudantes com conteúdo sobre as apresentações que irão ocorrer, sendo atualizado e adicionado pela instituição, em especial que atenda a estrutura organizacional da Universidade Federal do Espírito Santo.

1.1 O PROBLEMA E SUA IMPORTÂNCIA

O alto poder de processamento computacional dos *smartphones* das últimas gerações, atrelado à liberdade de criação de novos aplicativos (oficiais ou não) vem gerando um novo ecossistema de compartilhamento de conteúdo dos mais variados meios, como o acadêmico, artístico, informativo e cultural facilitado pela usabilidade das interfaces (GEMINIS, 2011).

O uso das Tecnologias de Informação Móveis (TIMS), conforme Mendieta et. al. (2013) aponta, tem contribuído para as organizações inovarem suas práticas. Estas tecnologias acabam, de certa forma, sendo mediadoras para o empreendedorismo organizacional permitindo encontrar novas oportunidades de negócios. Neste contexto, o uso de *smartphone* contribui positivamente nas dimensões de

inovatividade, proatividade e autonomia da orientação empreendedora de uma organização.

Além disso, com o avanço da utilização de aparelhos celulares no ambiente acadêmico, identificou-se a oportunidade de unir esta ferramenta tecnológica com a divulgação de eventos relativos a este ambiente. Esta possibilidade de utilização de multimeios no sistema educacional pode ser realizada, por exemplo, pelo desenvolvimento de aplicativos informatizados capazes de facilitar a vida acadêmica dentro de uma universidade.

A criação de aplicativos de comunicação tornará possível a troca de informações nos dispositivos móveis presentes em um ambiente de uma rede local de computadores. Tornando-se, assim, um sistema prático para que os alunos possam buscar interação com os mais diversos assuntos de interesse em seus *smartphones* (GALVANI, 2015).

No que tange ao desenvolvimento de aplicativos informatizados, estes normalmente seguem padrões de programação denominados *Model-View-Presenter* (MVP) e *Model-View-Control* (MVC*)*. Quando utilizados especialmente no Phonegap[1], torna a aplicação mais fácil de projetar, revisar e realizar manutenção (MFSMOBILE, 2013).

1.2 OBJETIVOS

1.2.1 Objetivo geral

Este trabalho tem como objetivo principal facilitar a divulgação de eventos acadêmicos nos campi da UFES. Isto se dará por meio da criação de uma aplicação *mobile* permitindo a qualquer aluno o acesso por meio do próprio celular.

1.2.2 Objetivos específicos

Os objetivos específicos são apresentados a seguir.

a. Desenvolver o sistema em HTML5 responsivo, CSS3, JavaScript.

[1] Phonegap: estrutura de desenvolvimento de aplicativos móveis.

b. Criar uma API REST para fazer a ligação entre o aplicativo e base de dados.

c. Criar uma base de dados para o cadastro de eventos.

d. Realizar uma avaliação qualitativa da ferramenta desenvolvida.

e. Testar a aplicação gerada junto aos usuários-alvo.

1.3 ESTRUTURA DO TRABALHO

O restante deste trabalho está organizado da seguinte forma: no item 2 estão dispostas as abordagens encontradas na literatura sobre criação de aplicativo, no item 3 a metodologia utilizada para realização deste trabalho e nos itens 4 e 5, estão descritos os resultados obtidos e as conclusões, respectivamente.

2 REVISÃO DE LITERATURA

2.1 PADRÃO MVP E MVC

O padrão MVP, assim como o MVC, é um separador de códigos em camadas, com finalidade de expor a camada de apresentação de um lado, a de dados do outro e a regra de negócio em uma outra parte do código. Todas essas camadas possuem nome, são elas:

• *View* - Formulário que exibirá os dados, não contém regra alguma do negócio a não ser disparar eventos que notificam mudança de estado dos dados que ele exibe e processamento próprio dele, como por exemplo código para fechar o formulário. Um objeto *View* implementa uma interface que expõe campos e eventos que o *Presenter* necessita.

• *Model* - Objetos de manipulação de dados. Um objeto *Model* implementa uma interface que expõe os campos que o *Presenter* irá atualizar quando sofrerem alteração na *View*.

• *Presenter* - Faz a ligação e intermedia a camada *View* e *Model*. Também se define por atualizar o *View* quando o modelo é alterado e sincronizar o *Model* em relação ao *View*.

Pode-se visualizar melhor o funcionamento dessas camadas por meio da relação entre elas dispostas na Figura 1.

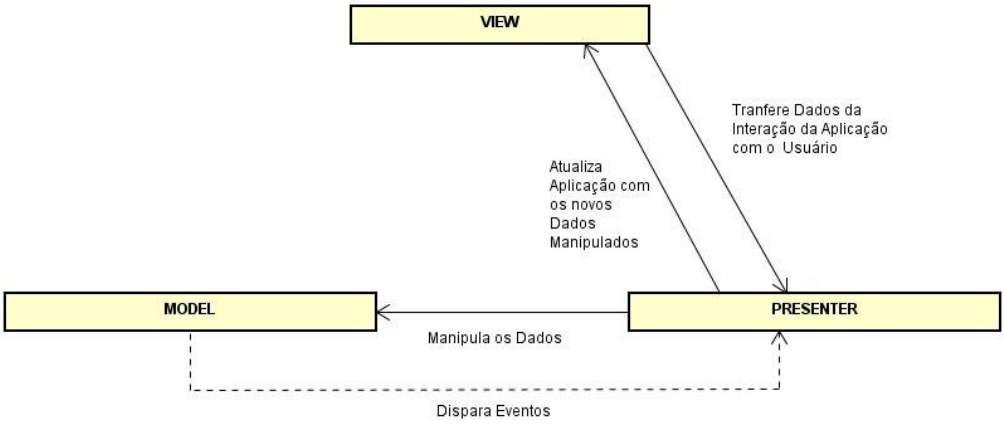


Figura 1: Associação entre as camadas MVP.

Fonte: O autor, 2017.

Como pode ser observado na Figura 1, no MVP, a camada *View* não interage com a *Model*. Assim, a *View*, neste padrão, é manipulada e atualizada essencialmente pela Presenter.

Entretanto, a ausência de interação entre a camada *View* e *Model* característica do padrão MVP, se mostrará diferente do MVC (Figura 2), em que a *Controller* pode interagir com a *Model* e esta pode atualizar a *View* (AUAD, 2017).

Estão dispostas na Figura 2, as relações de associação entre as respectivas camadas da MVC. Em que, diferente da MVP, a camada *Model* não dispara eventos diretos para a camada *Controller*.

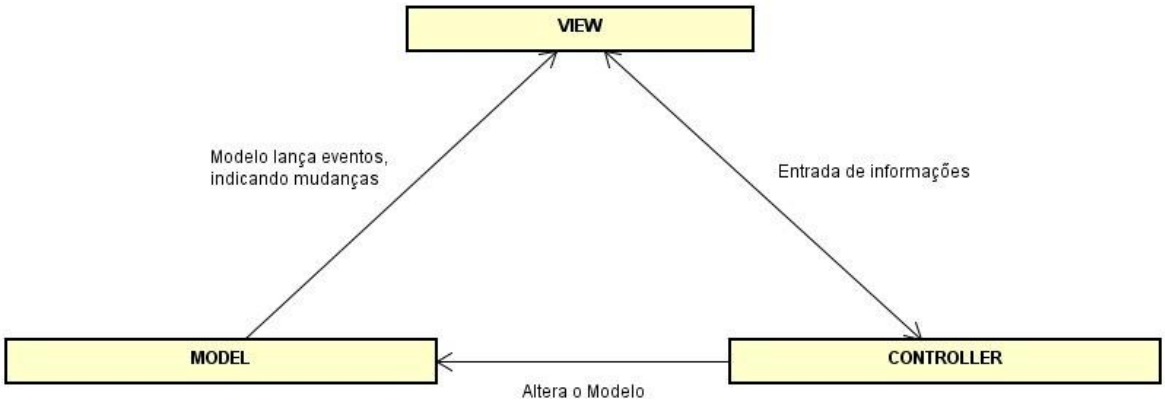


Figura 2: Estrutura MVC.

Fonte: O autor, 2017.

2.2 APLICAÇÕES MÓVEIS

Com o aumento das tecnologias nos aparelhos de celulares, houve um grande progresso no mercado de desenvolvimento de aplicativos para as mais diversas áreas com propósitos de trazer o lazer e/ou auxiliar como ferramenta em trabalhos de diferentes ramos.

Embora os aparelhos disponíveis no mercado sejam similares, suas plataformas são bem distintas, o que diferencia o processo de desenvolvimento destes aplicativos. Para cada plataforma, há um tipo específico de aplicação, variando entre *Web Apps*, Aplicações Nativas e Híbridas (PIRAN et al., 2014).

Os *Web Apps* são aplicações de tecnologias *web* que envolvem HTML5, CSS e JavaScript, manipulados de modo que se adequem aos requisitos da tela do

dispositivo. A vantagem deste tipo de tecnologia está na portabilidade.

Através do navegador é possível acessar a aplicação de diferentes *smartphones*, porém se ocorrer uma interrupção na internet há uma pausa nas possibilidades de acessos de informações do aplicativo (JUNIOR et al.,2016).

As Aplicações Nativas são um tipo de aplicação que exigem maior conhecimento por parte do programador, onde há uma necessidade de conhecer cada uma das linguagens de programação das plataformas que pretende desenvolver. Sobre cada aplicação há uma instalação direta no sistema operacional do telefone que executa sem a necessidade de conexão com Internet (PIRAN et al., 2014).

De acordo com Junior et al. (2016), por mais que exista comprometimento do programador sobre uma determinada linguagem em relação a um sistema operacional específico, as aplicações nativas são mais eficientes e rápidas do que as outras aplicações (JUNIOR et al., 2016). O Quadro 1 demonstra os diferentes tipos de linguagens e em quais plataformas são nativas.

Quadro 1: Linguagens nativas de sistemas operacionais.

Sistema Operacional	**Linguagem de Desenvolvimento**	**IDE**	**Sistema Operacional de Desenvolvimento**
Android	Java e XML	Equipe/Android Studio	Windows, Linux, OS, X
iOS	Objective-C	XCode	OS X
Windows Phone	C#	Visual Studio	Windows

Fonte: O autor, 2018.

Por fim, na aplicação híbrida existem características tanto da Nativa quando da *Web Apps*, uma vez que seu processo é executado dentro do componente nativo do aparelho e parte da utilização de HTML, CSS e JavaScript como um comunicador ao sistema para obter todas as funcionalidades do *smartphone*, sem a dependência direta a conectividade com a internet (MOREIRA, 2007).

2.2.1 *Framework* Phonegap

Phonegap, também conhecido como Cordova, provê de uma aplicação *open source* multiplataforma capaz de executar em diferentes tipos de plataformas móveis de forma transparente com o uso de linguagens voltadas para aplicações *Web*, ou seja,

HTML5, CSS3 e JavaScript (KASPERBAUER, 2012).

Ainda de acordo com Kasperbauer (2012), a aplicação no *framework* Phonegap é capaz de ser convertida em aplicações com linguagem nativa ao tipo de sistema do aparelho de celular. Isso torna a programação mais rápida e fácil, sem necessidade de maiores conhecimentos específicos em diversas e inúmeras linguagens.

Além disso, a aplicação quando implementada sobre o sistema do aparelho de celular, tem-se acesso às *Application Programing Interface* (APIs), de modo *online* ou *offline*, e às ferramentas nativas do aparelho que tornam possível a utilização da câmera, da geolocalização e do acesso à contatos, por exemplo (PHONEGAP, s.d.).

2.2.2 Interface Usuário

Trice (2012) destaca como uma das características do *framework* Phonegap, o fato das camadas de usuário das aplicações serem interpretadas por um navegador que é capaz de preencher 100% da página, no que tange à largura e à altura do dispositivo. O Phonegap também utiliza *WebView* adaptativo ao tipo de celular implantado (TRICE, 2012). Isso faz com que seja possível haver aplicações por todo espaço da tela do dispositivo, sem que haja interferência de decorações de páginas *web* nativas, por exemplo.

2.2.3 Adobe Phonegap Build

O Adobe Phonegap Build é uma ferramenta facilitadora para compilação de projetos realizados no Phonegap, que faz com que seja possível compilar o mesmo projeto para seis plataformas diferentes ao mesmo tempo, por exemplo.

A referida ferramenta atua por meio do empacotamento da aplicação sobre os formatos dos aplicativos nativos de cada plataforma desejada. Isso gera um arquivo binário capaz de ser inserido nas mais diversas lojas de aplicativos em diferentes tipos de aparelhos de celulares (TRICE, 2012).

2.2.4 Uso de Banco de Dados

O Phonegap também atua como um cliente capaz de interagir com um servidor de aplicativos a fim de receber dados através de uma lógica de dados, independente da

comunicação com o repositório *back-end*. Isso ocorre, uma vez que trabalha com qualquer servidor por meio dos protocolos padrões da *web*, como o HTTP, em que o servidor executa as regras de negócio e recupera ou persiste os dados a partir de uma base de dados (TRICE, 2012).

Segundo Trice (2012), a comunicação do aplicativo geralmente não está diretamente relacionada a uma base de dados. Encaminha-se uma comunicação através do servidor de aplicativos, ou a uma API, podendo ser realizada por meio da HTTP para conteúdo HTML, serviços JSON, etc. Isso, semelhante às técnicas usadas em uma aplicação AJAX baseada em desktop-browser. As aplicações se utilizam, por meio convencional, de um modelo de tela única, em que a lógica da aplicação é composta de uma página HTML, ou seja, de modo que nunca seja descarregada da memória. Ademais, as atualizações realizadas só dizem respeito aos dados via DOM HTML, por meio da recuperação dos dados do servidor por meio de técnicas AJAX.

2.3 API

As API (*Application Programming Interface*) proveem de um conjunto de funcionalidades capazes de serem consumidas por serviços, que quando utilizadas, conectadas a uma aplicação, são chamadas de integração (COOKSEY, 2014).

Segundo Cooksey (2014), é uma ferramenta, aplicação do lado do servidor, que torna um serviço consumível para o computador. Sendo ele capaz de permitir visualização, edição e exclusão de dados de um determinado banco, conforme esquema demonstrado na Figura 3.

Uma das arquiteturas mais utilizadas atualmente é a REST, um estilo que torna possível o auxílio, melhor comunicação e troca de informações entre a base de dados e as aplicações por meio do protocolo HTTP, cujo objetivo é determinar a forma na qual as URIs (*Unique Resource Identifiers*) da API devem ser modeladas. (OLIVEIRA, 2018).

Figura 3: Comunicação Aplicativo x Base de dados.

Fonte: O autor, 2018.

2.3.1 REST

O REST (*Representation State Transfer*) é um estilo arquitetural de um sistema que se comunica sobre o HTTP, que oferece os métodos GET, POST, PUT e DELETE, interage com outros sistemas, e utiliza identificadores *web*, chamados URI (FERREIRA et al., 2017).

Essa arquitetura foi definida como um conjunto coordenado de restrições que quando aplicadas em conjunto, tornam possível a usabilidade, simplicidade, escalabilidade e extensibilidade do serviço (LI et al., 2011).

2.3.1.1 Protocolo HTTP

Segundo Oliveira (2018), o protocolo HTTP (*HyperText Transfer Protocol*) é a base sobre a qual funciona os serviços na internet, como o mecanismo de transporte de envio e recebimentos de mensagens e dados.

Existem vários métodos e maneiras para realização de requisições no navegador, com suas particularidades e aplicações de acordo com as necessidades do sistema. Os métodos mais utilizados são:

- GET: utilizado para recuperação de dados e/ou informação na base de dados;
- DELETE: exclusão de dados e/ou informação na base de dados;
- POST: realizar a criação de um novo dado e/ou informação na base de dados;
- PUT: atualizar um dado e/ou informação já existente na base.

3 METODOLOGIA

Para alcançar os objetivos deste trabalho, foi desenvolvida uma aplicação com HTML5, CSS3 e JavaScript, baseando-se em métodos responsivos. Todo o desenvolvimento foi pensado de forma que o aplicativo fosse compatível com diferentes plataformas, e adequado visualmente aos diferentes tipos de aparelhos e meios de comunicação.

A manipulação do banco de dados foi realizada via requisição a API utilizando JavaScript, o que possibilita a criação, armazenamento, busca e exclusão de qualquer informação. O Diagrama de Estados, representado na Figura 4, descreve as etapas de acesso e uso do aplicativo pelo aluno da universidade (usuário comum).

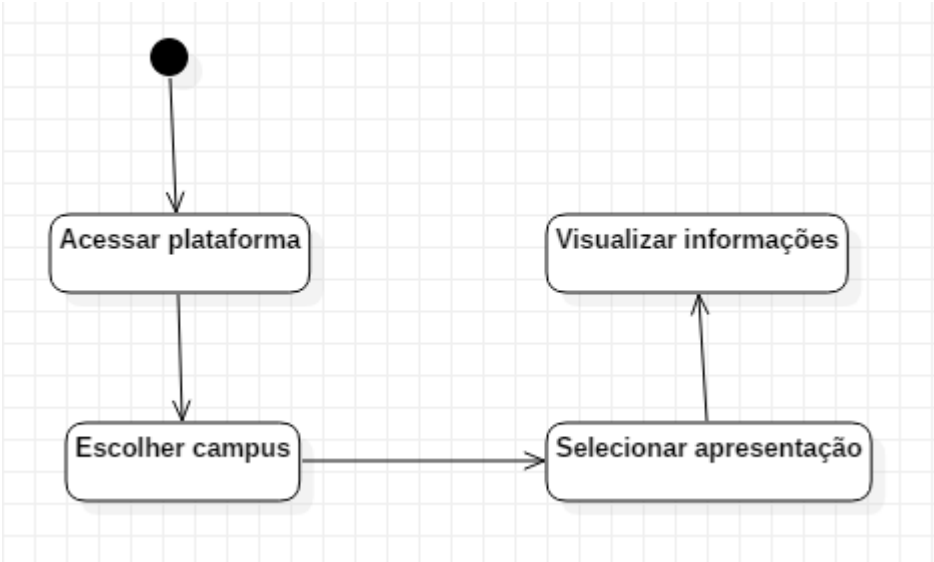


Figura 4: Diagrama de Estado.

Fonte: O autor, 2017.

3.1 PROTÓTIPOS

A primeira tela do aplicativo, como demonstrado na Figura 5, será dos campi da universidade.

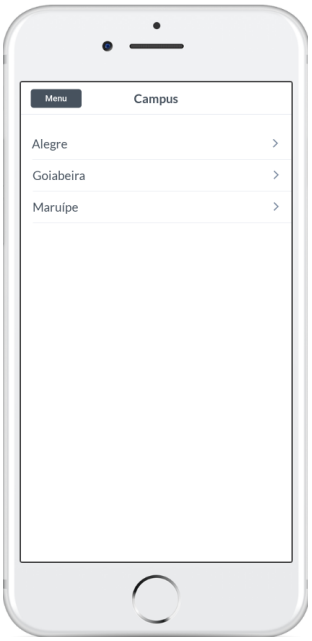


Figura 5: Seleção dos campi.

Fonte: O autor, 2018.

Na Figura 6a, são dispostas as apresentações correspondentes ao campus escolhido na tela anterior (Figura 5). Também são disponibilizadas informações básicas das apresentações, como o tema central, local, data e horário, conforme demonstrado na Figura 6b.

Figura 6: Apresentação do curso e seus respectivos dados.

Fonte: O autor, 2018.

É importante ressaltar que, todas as informações dispostas no aplicativo terão que passar pela aprovação dos coordenadores dos respectivos cursos, para que estes analisem e adicionem o conteúdo conforme a demanda necessária. Isso se dará por meio do painel administrativo do determinado curso, como demonstrado no diagrama de estados da Figura 7.

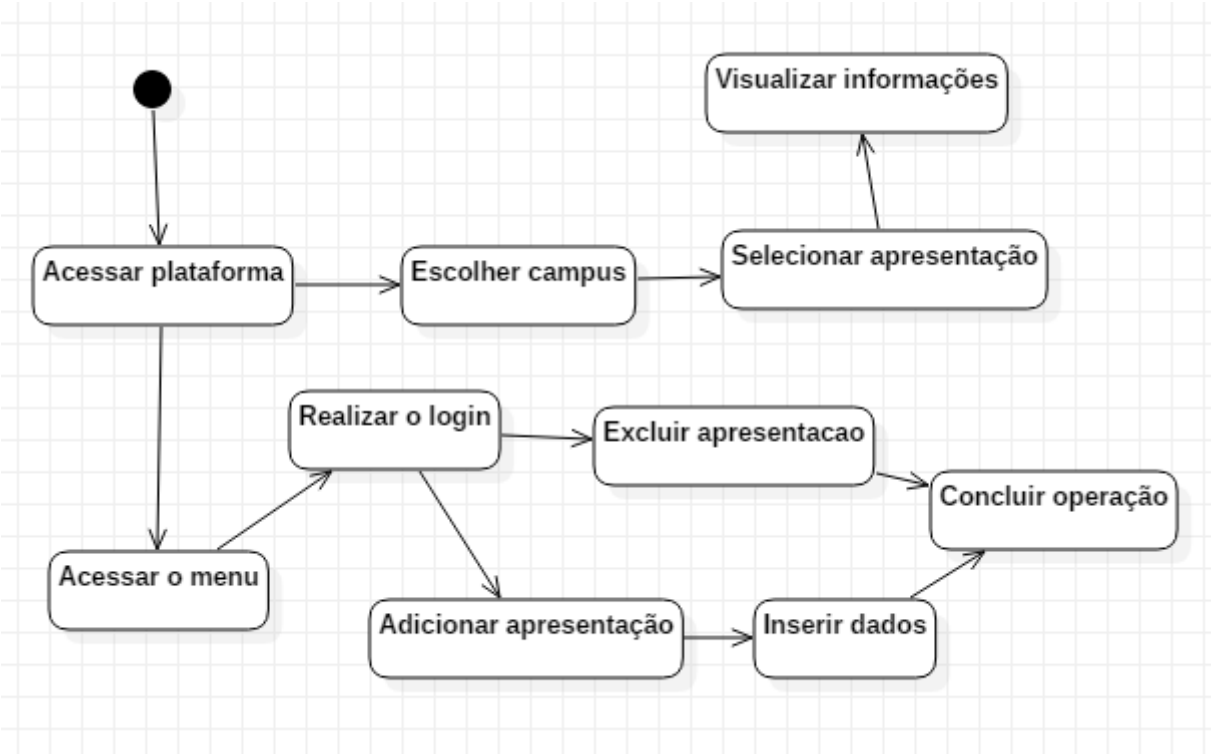


Figura 7: Diagrama de estados perfil Administrador.

Fonte: O autor, 2018.

Assim, o processo do perfil administrativo se dará por meio de um acesso restrito, disponível apenas os coordenadores de curso e chefes de departamento, como demonstrado na Figura 8.

Figura 8: Login Administrativo.

Fonte: O autor, 2018.

Com o acesso do perfil administrativo, será disponibilizado uma listagem das apresentações referentes ao curso cujo coordenador esteja associado, em que este poderá fazer alterações como excluir ou adicionar, por exemplo (Figura 9).

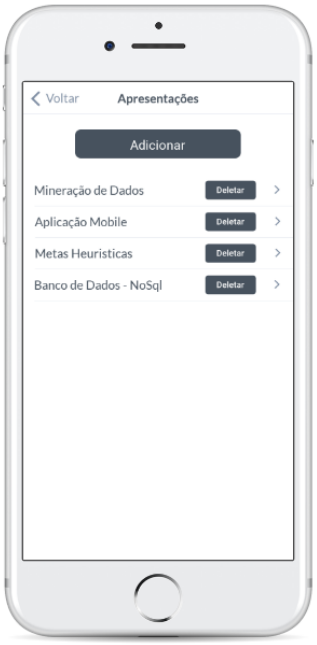


Figura 9: Área Administrativa.

Fonte: O autor, 2018.

No que se refere a adição e alteração de dados, exemplificados na Figura 10a, estes contêm todos os atributos necessários para inserção e/ou alteração antes do processo ser submetido à aprovação. Já a ação de excluir apenas será solicitada a confirmação (Figura 10b).

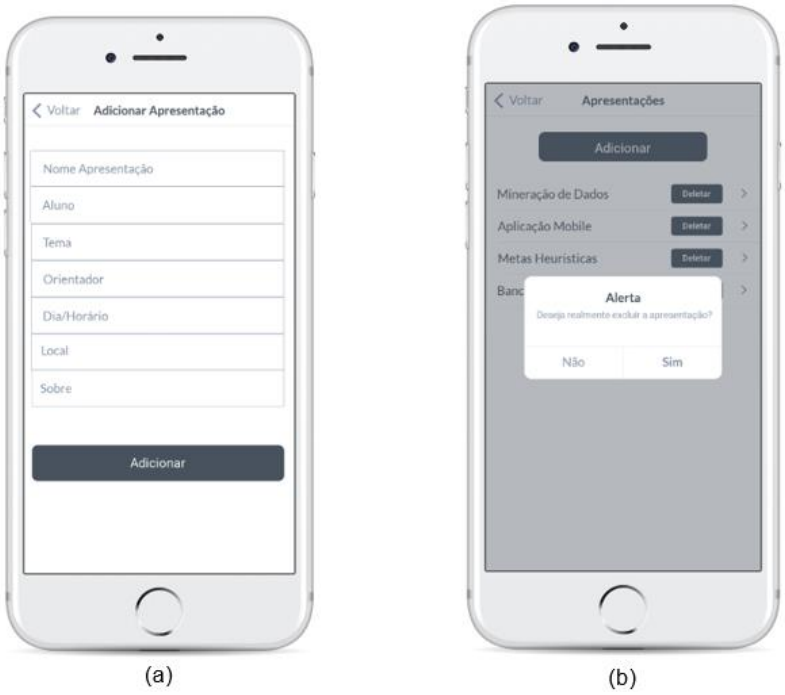


Figura 10: Alterar, adicionar e/ou confirmar dados.

Fonte: O autor, 2018.

Confirmada a ação, em ambos os casos dispostos na Figura 10, o programa realiza uma atualização no banco de dados e exibe a nova lista de apresentações em relação ao curso escolhido.

3.2 VALIDAÇÃO

O processo de validação do sistema foi realizado por meio da aplicação de um questionário qualitativo, no qual foram analisadas questões como facilidade de uso, importância, recursos e sugestões, em que estudantes foram convidados a participar.

Para facilitar a distribuição do aplicativo e permitir maior comodidade dos usuários e realização dos testes, inicialmente foi disponibilizado o arquivo de instalação do aplicativo para as plataformas de Android.

No que tange à usabilidade, foi elaborado um protótipo funcional evolutivo. Isto, pois, a aplicação tende a se adaptar da melhor forma possível com base no *feedback* dos usuários comuns, referente às mudanças desenvolvidas no aplicativo. Para facilitar o acesso às funcionalidades do sistema sem que haja necessidade de muitas informações de uso do programa.

Diante de questionários, e avaliações, ocorreu a necessidade de questionar os usuários do aplicativo acerca de suas maiores dificuldades diante as funcionalidades do sistema. Isto é, se o aplicativo, por si só, é autoexplicativo, sugestões do que poderia ser aprimorado e quais as maiores dificuldades, diante de uma visão pessoal, esse aplicativo sanaria.

Desta forma, através das análises das avaliações de usuários comuns diante de um protótipo, foi possível desenvolver um sistema final que atendesse melhor esse público alvo determinado, ou seja, que solucionasse grande parte dos problemas apontados durante as etapas de teste e sugestões.

3.3 ETAPA DO DESENVOLVIMENTO DO SISTEMA

A seguir será abordado o processo de desenvolvimento da aplicação, tomando como base todo o protótipo da metodologia, desenvolvido na plataforma Adobe XD, seguindo os diagramas e requisitos já apresentados.

3.3.1 Criação da Base de Dados

A base de dadas foi criada com base no MongoDB, utilizando um serviço oferecido pela mlab[2], na qual torna possível a hospedagem de um servidor gratuito com um limite de 0,5GB a ser utilizado com informações.

Todo processo de estruturação da relação entre as *collections* (*tables*) foi feito através da API criada. Assim que gerado o primeiro dado a base começa a criar sua estrutura de acordo com as *Models* implementadas. Essa parte será melhor abordada no tópico da geração da API.

[2] Mlab: serviço de banco de dados em nuvem totalmente gerenciado que hospeda bancos de dados do MongoDB.

3.3.2 Aplicação Mobile

Na aplicação *mobile* foi utilizado a plataforma Phonegap que permite o desenvolvimento de toda aplicação, utilizando as linguagens *web*, tornando possível a interpretação e geração do código para serem executados a partir de um *smartphone*.

Nessa parte foram geradas telas, seguindo as telas do protótipo presente na metodologia, utilizando as linguagens HTML para estruturar o código e os seus elementos presentes em cada página, o CSS no processo de tornar o *layout* mais agradável para visualização e utilização e o JavaScript que é utilizado no processo de requisição a API e também em algumas regras de funcionamento nas interações das páginas. Sem o JavaScript não seria possível realizar processos de busca, validação, criação ou exclusão dos dados que são exibidos na aplicação.

Em todas as requisições foi utilizada a Fetch API, onde passamos por parâmetro a URL que iremos consumir e também especificamos qual será seu método de requisição, passando ou não um corpo em formato JSON no envio.

Como retorno temos uma *promisse*, em que é retornado um objeto *response* com as informações da resposta do servidor, nesse caso, da API.

3.3.3 Geração da API

No processo da criação da API foi utilizado o NODE, um interpretador para JavaScript que é capaz de migrar todo o lado do cliente para o servidor, tornando assim possível a conexão entre a aplicação *mobile* ao banco de dados especificado pela *connectionstring* (*string* de conexão) armazenada. A API é definida por rotas e através do uso dessas rotas obtém-se os métodos que foram criados.

Para essa aplicação foram criadas rotas baseadas em Apresentacao, Campus e Usuário, onde são especificados quais os métodos que iremos tratar, são eles:

- PUT;
- GET;
- POST;

- DELETE.

Especificado os tipos de requisições sobre as devidas classes, são criados os métodos nas *Controllers* de cada rota para representar suas requisições e quais serão suas ações diante as representações das bases de dados em suas respectivas *Models*.

Através da criação das *Models* que definimos como será estruturada a base de dados no nosso servidor e a partir do primeiro documento inserido na *collection* a tabela é gerada. Na Figura 18, segue um exemplo de como especificar um campo a ser gerado na Model Apresentation:

```
orientador:{
                type: String,
                required: true,
                trim: true
                },
```

Figura 11: Campo da *Model Apresentation*.

Fonte: O autor, 2018.

Nesse caso temos o campo orientador na tabela de apresentação com o tipo de dado *string*, sendo obrigatório no momento da inserção e utilizando *trim* para desprezar campos vazios inutilizados.

4 RESULTADOS

Nos resultados teremos duas etapas, são elas: etapa do projeto da aplicação, demonstrado com base nos requisitos necessários do sistema, casos de uso, diagrama de atividades, diagrama de classes e etapa do desenvolvimento do sistema, em “ANEXO C” as fotos tiradas do aplicativo, e testes de usabilidade da aplicação.

4.1 PROJETO DA APLICAÇÃO

4.1.1 Requisitos

Através de análises nos aplicativos disponíveis para gerenciamento de atividades dentro das universidades. Tem-se uma grande deficiência nos processos de divulgação de atividades, além disso, há um foco apenas direcionado a aplicativos para plataforma Android, deixando um pouco a desejar às pessoas que possuem outro tipo de sistema operacional. No Quadro 2, temos uns dos poucos aplicativos voltados para divulgação de atividades com suas determinadas plataformas e algumas de suas principais atividades.

Quadro 2: Aplicativos existentes no mercado.

Plataforma	Nome	Atividades
Android	Union	Exibição de Notícias; Portais do aluno; Telefones uteis; Calendário Acadêmico; Cardápio RU.
	UFMGuia	Informações sobre bolsas; Normas acadêmicas; Horários de ônibus; Mapa dos campi.

Fonte: O autor, 2018.

Além da análise no mercado de aplicativos, foi realizado uma pesquisa com o intuito de levantar qual seria a importância de um aplicativo ao ponto de vista dos possíveis futuros usuários, a fim de avaliar a real necessidade do desenvolvimento do mesmo, do gerenciamento das apresentações, se há um interesse em obter essas

informações de forma rápida, simples e compartilhada entre as mais diferentes plataformas de *smartphones*.

Na pesquisa realizada, das questões disponíveis no questionário, as 6 primeiras são relacionadas aos perfis das pessoas, sejam elas usuários de aplicativos ou não e se eles possuem vínculo com alguma universidade. Perguntas de 7 a 11 são relacionadas ao nível de interesse das pessoas sobre as apresentações, e demais perguntas são relacionadas ao nível de importância de cada informação sobre a apresentação presente no aplicativo.

O questionário foi aplicado através da plataforma de formulários do Google e foram obtidas 27 manifestações. Dentre as respostas, há um destaque onde 100% dos questionados não ficam cientes de todas apresentações que ocorrem nas suas universidades, como mostra a Figura 11. Também observou-se que 100% dos entrevistados possuem *smartphones*, como mostra a Figura 12. E 85,2% dos usuários questionados gostariam de obter um aplicativo com informações sobre as apresentações de toda a universidade, apresentado na Figura 13. As informações apresentadas como opção para estarem no aplicativo foram: aluno, orientador, um breve resumo, data e hora, local, tema, um breve sobre, artigo produzido, como mostra a Figura 14.

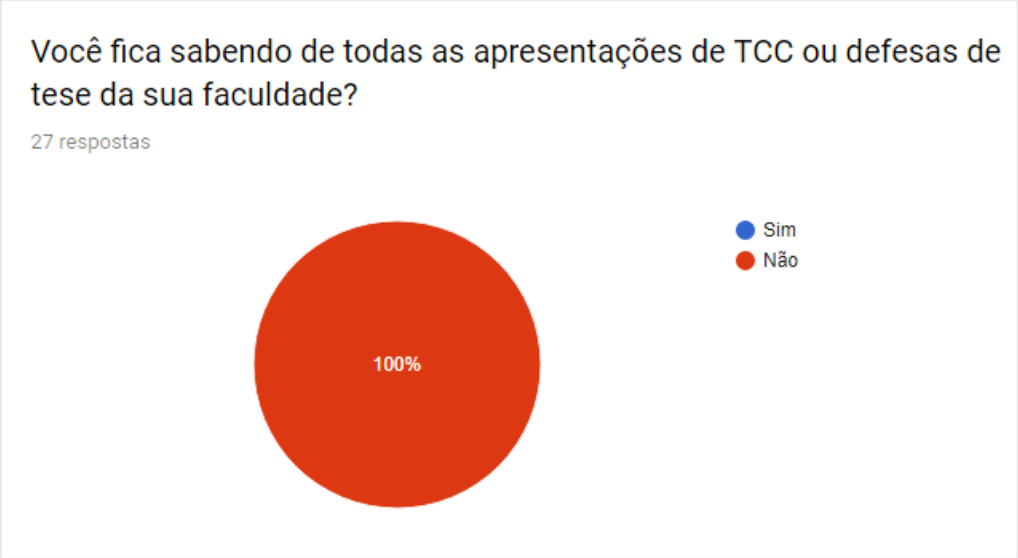


Figura 12: Porcentagem dos questionados que têm conhecimento sobre as apresentações de TCC ou defesa de tese na faculdade.

Fonte: O autor, 2018.

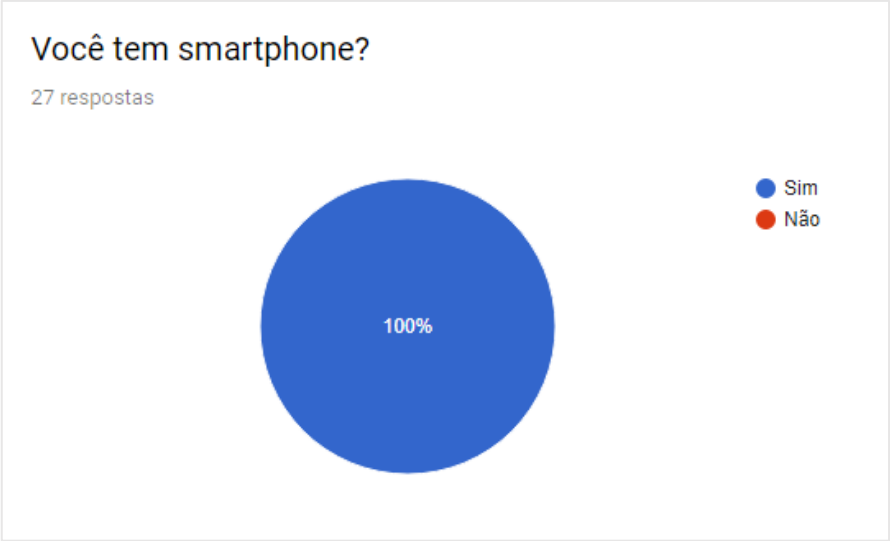


Figura 13: Porcentagem dos questionados que possuem *smartphone*.

Fonte: O autor, 2018.

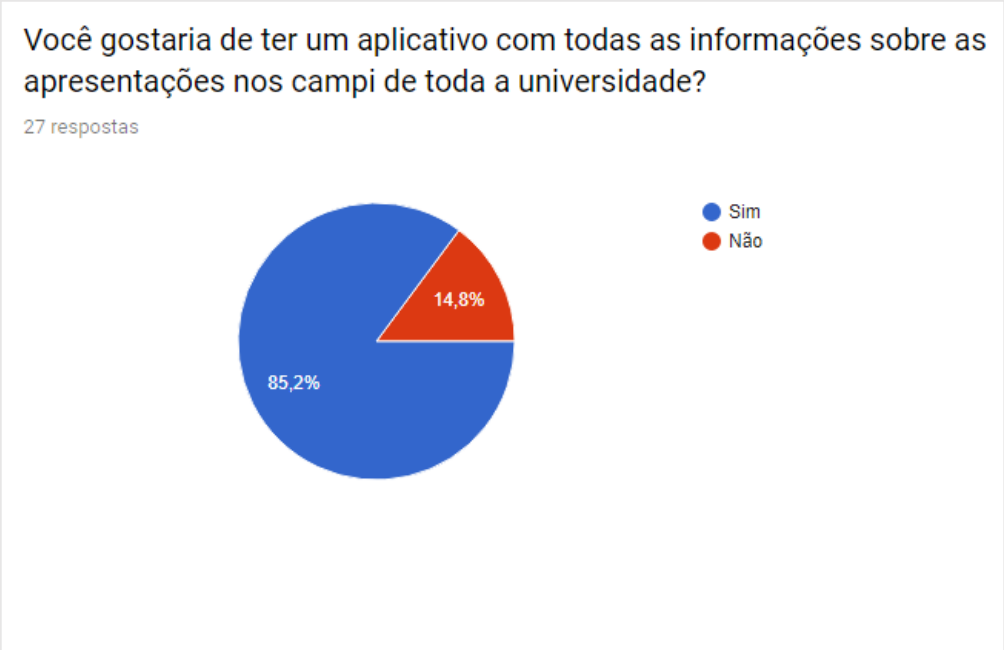


Figura 14: Porcentagem dos questionados que gostariam de utilizar o aplicativo.

Fonte: O autor, 2018.

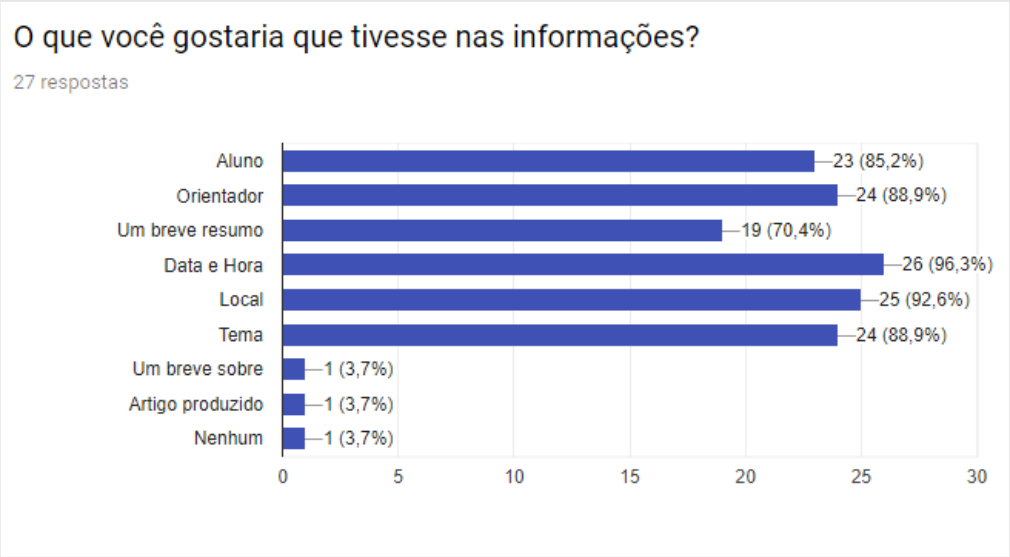


Figura 15: Porcentagem de cada informação que os usuários gostariam que tivesse no aplicativo.
Fonte: O autor, 2018.

A partir do levantamento das respostas obtidas através do questionário, foi possível identificar que o aplicativo proposto neste trabalho é de grande interesse e uso por parte das pessoas, visto que todos os entrevistados possuem um *smartphone* e a grande maioria informou que usaria um aplicativo para divulgação das apresentações, uma vez que não são informados sobre todas presentes nos campi.

Entre os interessados no uso do aplicativo, foi levantado o nível de interesse com base nas informações necessárias sobre as apresentações, contendo os maiores interesses nas seguintes informações:

- Aluno – responsável pela apresentação;
- Orientador – responsável pela orientação ao aluno durante a execução do trabalho;
- Data e Hora – dia e horário em que irá ocorrer a apresentação;
- Local – onde ocorrerá a apresentação;
- Tema – qual será o tema da apresentação;
- Um breve resumo / Um breve sobre – resumo sobre o que será apresentado/abordado na apresentação.

4.1.1.1 Requisitos Funcionais

Após a análise dos aplicativos e a pesquisa realizada com os possíveis usuários, foram levantados os seguintes requisitos, dispostos no Quadro 3:

Quadro 3: Requisitos funcionais.

ID	NOME	DESCRIÇÃO
RF001	Cadastrar Administrador	Aplicação deve permitir inclusão de novos administradores, utilizando a tela de cadastro, passando as informações de usuário e senha, e principalmente a chave.
RF002	Cadastrar Apresentação	Aplicação deve permitir a inclusão de novas apresentações, utilizando a tela de cadastro de apresentações, passando todos os parâmetros necessários e apresentados no formulário.
RF003	Deletar Apresentação	Aplicação deve permitir a exclusão das apresentações, na tela de atividades recentes, após o login administrativo.
RF004	Listar Apresentações	Aplicação deve listar e consultar todas as apresentações presentes para cada campi listado na tela inicial.
RF005	Listar Campi	Aplicação deve listar e consultar todos os campi já cadastrados na base dedados.

Fonte: O autor, 2018.

4.1.2 Diagrama de Caso de Uso

Com os requisitos funcionais já definidos, na Figura 15 temos o diagrama de caso de uso, com a finalidade de descrever e definir todas as funcionalidades da aplicação.

No diagrama temos apenas um ator, o administrador da aplicação e 8 casos de uso.

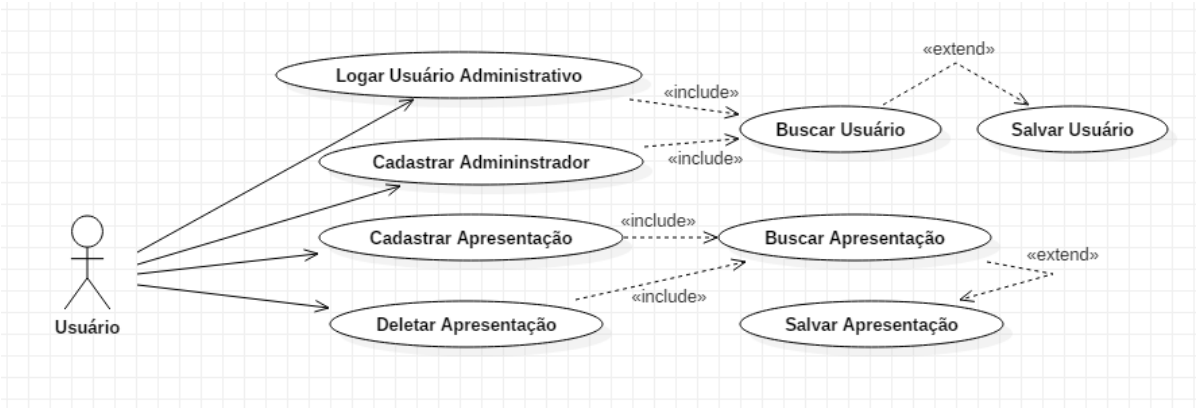


Figura 16: Diagrama de Caso de Uso.

Fonte: O autor, 2018.

No “ANEXO B” temos as descrições dos casos de uso, informando sobre cada um, a sua descrição, qual o será o principal atuador, suas pré e pós condições, o fluxo principal e, caso houver, um alternativo.

4.1.3 Diagrama de Atividade

Foi desenvolvido um diagrama de atividades (Figura 16) com o propósito de tornar mais claro todo o fluxo presente dentro dos casos de uso, mostrando as partes do uso através do público quanto aos processos do aplicativo.

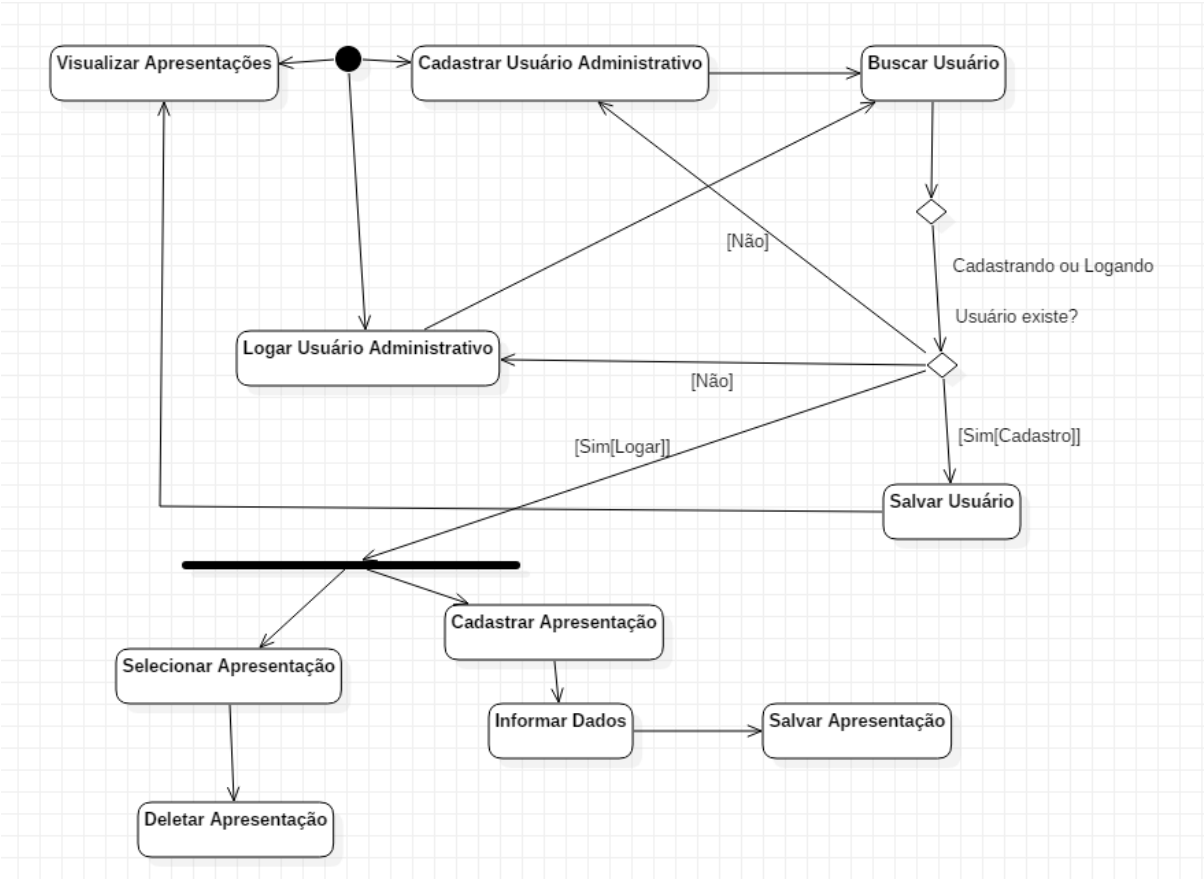


Figura 17: Diagrama de Atividade.

Fonte: O autor, 2018.

4.1.4 Diagrama de Classe

Um diagrama de classe tem como objetivo mostrar as classes que irão compor o aplicativo e seus relacionamentos.

Para aplicação foram criadas três classes, como apresentado na Figura 17: “Usuario”, “Campus” e “Apresentacao”. Nessas, existe uma relação de 1 para “n” entre as classes “Campus” e “Apresentacao”, sendo possível um campus obter várias apresentações, mas restringindo uma única apresentação especifica a um determinado campus. E quanto a classe de “Usuario” não existe uma relação, apenas é utilizada para acesso de área restrita para o método de cadastro de apresentações.

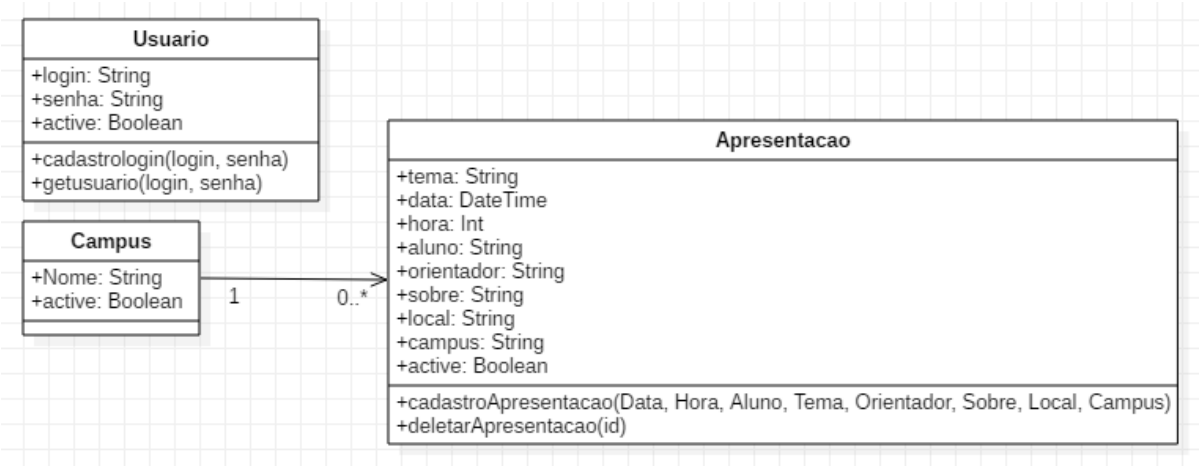


Figura 18: Diagrama de Classe.

Fonte: O autor, 2018.

A classe Campus tem os atributos: Nome e Active. O nome do campus é do tipo *string* e o atributo Active é do tipo booleano para manter a regra de apenas visualizar os campi que estão ativos, ou com valor *true*, na base de dados caso seja necessário tornar inativo algum dos já existentes.

Na classe Campus, inicialmente já é adicionada uma carga de dados com os atuais campi presente na UFES, tornando não necessário um método especifico para realizar os cadastros.

A classe Usuário tem os atributos: Login, Senha e Active. No quadro 8 temos um dicionário de dados referente aos atributos da classe.

Quadro 4: Dicionário de dados da classe Usuário

Atributo	**Tipo de Atributo**	**Especificação**
Login	String	Usuario de login
Senha	String	Senha referente ao usuário de login
Active	Bool	Atributo para descrever uma apresentação como ativa ou não

Fonte: O autor, 2018.

Na classe Usuário temos dois métodos:

- cadastrologin;
- getusuario.

No método de cadastrologin é realizado a validação se os campos de Usuário e Senha estão preenchidos, e também se a chave secreta está corretamente inserida e correspondente com a chave pré-definida diretamente no sistema. O retorno do método é *void* mas assim que realizado o cadastro, caso não haja problema no cadastro do dado na base, uma mensagem de cadastro realizado com sucesso é apresentada. Pseudocódigo do método na figura 18.

```
1.   FUNCAO(s)
2.   INÍCIO
3.       CRIAR (uma variável usuario, senha e chave);
4.       SE (chave = "2012204292")
5.                INSERIR (conta de usuário passando a variável usuário e senha)
6.                ESCREVA "Cadastro realizado com sucesso"
7.       SENAO
8.           ESCREVA "Chave incorreta"
9.       FIM-SE;
10.  FIM.
```

Figura 19: Pseudocódigo CadastroLogin.

Fonte: O autor, 2018.

Já no método de getusuario temos como parâmetro os valores inseridos em Usuário e Senha, os quais são validados através de uma consulta na base de dados, se existem ou não, com a finalidade de validar o *login* que está sendo realizado, caso os valores coincidam com os presentes na base de dados, um campo de entrar é mostrado na aplicação para que seja redirecionado para área administrativa. Pseudocódigo do método na figura 19.

```
1.   FUNCAO(s)
2.   INÍCIO
3.       CRIAR (uma variável usuário e senha);
             SE (Ao chamar o método buscarUsuario na interface passando as
4.           variáveis usuário e senha, se houver um usuário com o valor
             respectivo das variáveis, se retornar o valor = true)
5.               EXIBIR Botão entrar na tela
6.           SENAO
7.              ESCREVA "Usuario/Senha inválido"
8.           FIM-SE;
9.   FIM.
```

Figura 20: Pseudocódigo Getusuario.

Fonte: O autor, 2018.

A classe Apresentações têm os seguintes atributos: Tema, Data, Hora, Aluno, Orientador, Sobre, Local, Campus e Active. No quadro 9 temos um dicionário de dados referente aos atributos da classe.

Quadro 5: Dicionário de dados da classe Apresentações

Atributo	Tipo de Atributo	Especificação
Tema	String	Tema da apresentação
Data	Date	Data da apresentação
Hora	Int	Hora em que irá ocorrer apresentação, tipo inteiro de tamanho 4 que condiz com horas e minutos
Aluno	String	Aluno que fará apresentação
Orientador	String	Orientador responsável pelo Aluno
Sobre	String	Descrição sobre o trabalho que será realizado/apresentado
Local	String	Local que irá ocorrer a apresentação
Campus	String	Campus da apresentação
Active	Bool	Atributo para descrever uma apresentação como ativa ou não

Fonte: O autor, 2018.

Na classe Apresentação temos dois métodos:

- cadastroApresentacao;
- deletarApresentacao.

No método de cadastroApresentacao são passados todos os campos de Data, Hora, Aluno, Tema, Orientador, Sobre, Local e Campus como parâmetro com o tipo de retorno *void*, em que seu propósito é armazenar na base de dados a nova apresentação com os valores de parâmetros passados. Com todos esses valores na base será criado um id que corresponde a apresentação por completo. O Pseudocódigo do método foi apresentado na figura 20.

```
1.   FUNCAO(s)
2.   INÍCIO
3.       CRIAR (uma variável Data,Hora, Aluno, Tema, Orientador, Sobre, Local e
         Campus);
                    (Ao chamar o método de InserirApresentacao, passando as
4.       SE         variáveis Aluno, Tema, Orientador, Sobre, Local, Hora, Data e
                    Campus, no retorno o valor = true)
5.                  ESCREVA "Apresentação cadastrada com sucesso"
6.       SENAO
7.                  ESCREVA "Ocorreu um erro ao cadastrar apresentação"
8.       FIM-SE;
9.   FIM.
```

Figura 21: Pseudocódigo CadastroApresentacao.

Fonte: O autor, 2018.

Já no método de deletarApresentação recebe apenas o id da apresentação gerada, com o objetivo de deletar, ou seja, torná-la inativa. Quando inativa a apresentação não

irá ser apresentada em nenhuma parte do sistema, apenas servirá de registro no histórico das apresentações, conforme o pseudocódigo do método na figura 21.

```
1.   FUNCAO(s)
2.   INÍCIO
3.       CRIAR (uma variável idApresentacao);
4.       SE (Ao chamar o método de deletarApresentacao da API passando o
         idApresentacao, se ela me retornar valor = true)
5.               ESCREVA "Apresentação deletada com sucesso"
6.       SENAO
                 ESCREVA "Ocorreu um erro ao deletar apresentação"
7.       FIM-SE;
8.   FIM.
```

Figura 22: Pseudocódigo DeletarApresentacao.

Fonte: O autor, 2018.

4.2 TESTES DE USABILIDADE DA APLICAÇÃO

O teste de usabilidade foi realizado com um pequeno grupo de 10 pessoas, nele foi solicitado a execução de algumas atividades, são elas:

- Realizar cadastro de login administrativo;
- Realizar login na área administrativa;
- Cadastrar uma nova apresentação;
- Deletar apresentação;
- Visualizar/Detalhar apresentação gerada ou existente em algum campus.

O processo de teste foi observado pelo desenvolvedor da aplicação na realização de todas as atividades e solicitado ao usuário que ao realizar alguma operação, descrevesse o que estava sendo feito e comentasse caso houvesse algum problema, no que diz respeito à falta de clareza em algumas funcionalidades da aplicação.

Para todos os testes aplicados, ao término, foi solicitado aos usuários que pontuassem quais foram suas maiores dificuldades, o que poderia ser melhorado ou adicionado na aplicação e se a aplicação atenderia o propósito de gerenciar e compartilhar sobre as apresentações dos campi da UFES.

Os resultados levantados na conclusão dos testes foram:

• Na versão inicial da aplicação, apenas para criação do *layout* foram levantadas questões de problemas na navegação das abas, onde ao apertar o botão de voltar a aplicação, tudo travava e a aplicação não respondia mais a qualquer tipo de comando até que fosse reiniciada;

• De todos os usuários, houveram questionamentos sobre as apresentações adicionadas, nas quais não estavam aparecendo no aplicativo instalado por eles mesmos em outro aparelho de celular, apenas nos próprios celulares;

• Mensagens não muito amigáveis quando ocorrido algum problema durante utilização das atividades.

Visto todos os *feedbacks*, conforme é possível verificar no “ANEXO A”, dos usuários que realizaram o teste, foi pensado e recriado todo o processo de alertas na execução das atividades, a fim de torná-las mais amigáveis e claras aos olhos dos usuários. Também foi refeito todo o processo de rotas em que as páginas interagem, tornando o acesso mais simples e sem falhas, sem a necessidade de reiniciar a aplicação.

Além disso, quanto ao compartilhamento das apresentações que não eram apresentadas na aplicação em outro celular, trata-se da base de dados inicial criada para aplicação, sendo ela WEB SQL, ou seja, não tornava a base compartilhada em um servidor, e sim armazenamento em memória cache dos aparelhos de celular.

Para resolver isso, foi gerada a API em meio às requisições, criando assim um servidor com dados a serem compartilhados e exibidos para todos os *smartphones* ao mesmo tempo.

5 CONCLUSÕES

Com base nos problemas apresentados de falta de comunicação da UFES e o aumento do uso de *smartphones* e seus aplicativos por meio da população universitária, nesse trabalho, foi desenvolvido um aplicativo capaz de gerenciar sobre todas as apresentações nos campi da universidade.

Tendo em vista os inúmeros tipos de *smartphones* e seus diferentes tipos de sistemas operacional, tem-se um grande problema de desenvolvimento para aplicativos apenas direcionados a determinados sistemas operacionais, mas com o avanço da tecnologia e a criação de novos métodos de implementação, surgiram meio de programação que possibilitam abranger diferentes meios com uma única aplicação, tornando o processo mais rápido e simples.

Sendo assim foi desenvolvido uma aplicação *mobile* com Phonegap no intuito de compartilhar um sistema simples e prático de informações entre diferentes plataformas e abranger todo o tipo de pessoa que possui *smartphone*, tornando acessível todas as informações sobre todas as apresentações nos campi da Universidade Federal do Espirito Santo.

O resultado obtido foi bastante satisfatório, demonstrando uma resposta bem rápida para todo o processo, em todos os tipos de sistemas operacionais, tanto na adição quanto na exclusão das atividades que ocorrem durante o dia-dia da universidade.

Não havendo divergência de processo entre celulares de outras marcas, tem-se que uma aplicação híbrida pode se tornar de grande foco diante os olhos dos desenvolvedores *mobile*.

Para trabalhos futuros, recomenda-se estudar novas bibliotecas e pacotes que são gerados sempre nos fóruns de programação, buscando inovar e integrar as aplicações de todos os sistemas operacionais em uma única aplicação. Além disso, é interessante que com o passar do tempo essa aplicação seja atualizada com novos recursos e funcionalidades, para sempre satisfazer a demanda dos estudantes.

REFERÊNCIAS BIBLIOGRÁFICAS

AUAD, T. **O Padrão MVP em contraste com MVC**. Disponível em: <https://tassioauad.com/2014/01/17/>. Acesso em: 10 de nov. 2017.

BAYER, J. **O poder do smartphone**. Disponível em:<https://www.bayerjovens.com.br/pt/materia/?materia=o-poder-do-smartphone>. Acesso em: 16 de nov. 2017.

COOKSEY, B. **An Introduction do APIs**. Zapier, 2014.

ESTADÃO. **Até o fim de 2017, Brasil terá um smartphone por habitante**. Disponível em:<http://link.estadao.com.br/noticias/gadget,ate-o-fim-de-2017-brasil- tera-um-smartphone-por-habitante-diz-pesquisa-da-fgv,70001744407>. Acesso em: 15 de nov. 2017.

EXAME. **Smartphone se consolida como meio preferido de acesso à internet**. Disponível em: <https://exame.abril.com.br/tecnologia/smartphone-se-consolida-como-meio-preferido-de-acesso-a-internet/7>. Acesso em: 15 de nov. 2017.

FERREIRA F.; SANTOS T.; MARTINS F.; LOPES A.; VASCONCELOS V. **Especificação de Interfaces Aplicacionais REST**. Universidade de Lisboa, Faculdade de Ciências, LaSIGE, 2017.

GALVANI, A. **Desenvolvimento de uma aplicação para suporte ao E-Learning**. 2015, 39 f. Dissertação (Mestrado em Televisão Digital) – Universidade Estadual Paulista, Bauru, 2015.

GEMINIS. São Paulo, ano 2, n.1, p. 1-311, jan./jun. 2011.

NASCIMENTO JUNIOR, L. C.; BONETTI, T. **Desenvolvendo aplicações multiplataformas para dispositivos móveis utilizando ionic e Phonegap**. Disponível em: <http://web.unipar.br/~seinpar/2016/publicacao/Luiz_Carlos_Junior.pdf >. Acesso em 18 de dez. 2017.

KASPERBAUER, M. **Desenvolvendo aplicativos multiplataforma com tecnologias web: Criando aplicativos para tablets e smartphones com o**

Phonegap. Disponível em: <http://www.univale.com.br/unisite/mundo-j/artigos/56_multiplataforma.pdf>. Acesso em: 21 de jun. 2017.

LI, L.; WU, C. **Design and Describe REST API without Violating REST: A Petri Net Based Approach**. 2011 IEEE International Conference on Web Services, p. 508–515, 2011.

MENDIETA, Aline Carrion Martens Cristina Dai Pra ; Bento, Fabio de Oliveira; Lacerda, Fabricio Martins**. Revista Eletronica de Estratégia e Negócios**. 2013, Vol.6(3), p.212(26).

MFSMOBILE. **MVC Architechture & OOPS in JavaScript using Phonegap**. Disponível em: https://mindfiremobile.wordpress.com/2013/07/31/mvc- architechture-oops-in-javascript-using-phonegap/>. Acesso em: 18 de nov. 2017.

MOREIRA, R. P. **Maw – uma abordagem híbrida de modelagem para aplicações webapps**. 2007. 74 p. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Lavras, Lavras, 2007.

OLIVEIRA, A. G. G. **Construção de Aplicações Distribuídas Utilizando-se de APIs REST**. Monografia, Universidade do Estado do Rio Grande do Norte. Mossoró, RN, 2018.

PIRAN, F.; LAZZARETTI, A. **Estudo da tecnologia Phonegap/Cordova e a aplicação em um estudo de caso**. Disponível em: <http://painel.passofundo.ifsul.edu.br/uploads/arq/20160331164543877842707.pdf> Acesso em: 18 de dez. 2017.

PHONEGAP. **About The Project**. Disponível em:<http://phonegap.com/about/> Acesso em: 13 de nov. 2017.

TRICE, A. **Phonegap Explained Visually**, 2012. Disponível em: <https://phonegap.com/blog/2012/05/02/phonegap-explained-visually/>. Acesso em: 20 de nov. 2017.

ANEXOS

ANEXO A – Documento de *Feedbacks*

FEEDBACKS

“Gostei muito do aplicativo, mas me parece que as únicas informações que aparecem no aplicativo são as que foram inseridas por mim.”

“Ótima iniciativa, gostei muito da ideia, vai ajudar bastante as universidades.”

“Achei interessante a ideia do app. Mas durante o uso notei algumas coisas que poderiam ser melhoradas, como por exemplo as mensagens “de erro” que poderiam ser mais amigáveis.”

“Gostei do aplicativo, mas algumas horas ele para de funcionar, nenhum botão funciona.**”**

“Me parece que o aplicativo está perdendo a rota das telas. Tirando isso está bacana.”

“Bem fácil de utilizar, tentei cadastrar e o aplicativo me trouxe uma mensagem estranha por não ter inserido a chave correta.**”**

“Excelente ideia, gostei bastante, mas não consigo ver apresentações inserida por outros.**”**

“Top de mais, muito irado. Só poderia melhorar um pouco as mensagens de erro.**”**

“Muito legal, aplicativo bem bacana para gerenciar apresentações, parabéns.”

“Fácil de utilizar, bem intuitivo. Botões claros e objetivos.”

ANEXO B – Diagramas de Caso de Uso

UC001 – Cadastrar Administrador	
Descrição	O usuário realiza o cadastro da conta administrativa
Ator	Usuário.
Pré-condições	Ter acesso a palavra chave secreta.
Pós-condições	Cadastro realizado com sucesso
Fluxo principal	[FP01] – Usuário insere login, senha e a chave secreta. [FP02] – Usuário clica no botão de “Cadastrar”. [FP03] – O aplicativo realiza o cadastro do usuário.
Fluxo alternativo	[FA01] – Usuário informa um login e senha incorreta, voltando para [FP01], e a mensagem “usuário e senha já existente”. [FA02] – Usuário informa a palavra chave incorreta, voltando para [FP01], e a mensagem “chave incorreta”.

UC002 – Logar Usuário Administrativo	
Descrição	O usuário realiza o login no aplicativo na área administrativa.
Ator	Usuário.
Pré-condições	Ter uma conta já cadastrada.
Pós-condições	Login realizado com sucesso.
Fluxo principal	[FP01] – Usuário insere login e senha. [FP02] – Usuário clica no botão de “Validar”. [FP03] – O aplicativo busca pelo usuário. [FP04] – O usuário clica no botão de “Entrar”. [FP05] – O aplicativo realiza o login do usuário.
Fluxo alternativo	[FA01] – Usuário informa um login e senha errado ou inexistente, voltando para [FP01], e a mensagem “usuário e/ou senha incorretas”.

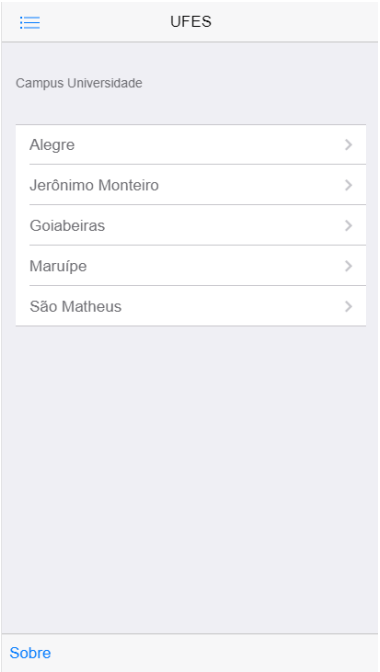
UC003 – Cadastrar Apresentação	
Descrição	O usuário realiza o cadastro da apresentação.
Ator	Usuário.
Pré-condições	Ter uma conta para acesso da área administrativa.
Pós-condições	Apresentação cadastrada com sucesso.
Fluxo principal	[FP01] – Usuário insere as informações de Aluno, Tema, Data, Hora, Local, Sobre e Orientador. [FP02] – Usuário clica no botão de “Adicionar”. [FP03] – O aplicativo realiza o cadastro da apresentação.
Fluxo alternativo	N/A

UC004 – Deletar Apresentação	
Descrição	O usuário deleta uma apresentação existente.
Ator	Usuário.
Pré-condições	Ter uma conta para acesso da área administrativa.
Pós-condições	Apresentação deletada com sucesso.
Fluxo principal	[FP01] – Usuário arrasta a apresentação para esquerda. [FP02] – Usuário clica no botão de “Deletar”. [FP03] – Usuário confirma a operação. [FP04] – O aplicativo deleta a apresentação.
Fluxo alternativo	N/A

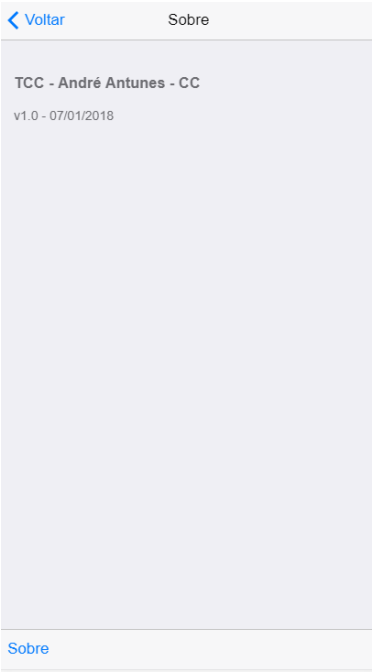
UC005 – Listar Apresentações Recentes	
Descrição	O usuário atualiza a lista de atividades recentes.
Ator	Usuário.
Pré-condições	Ter uma conta para acesso da área administrativa.
Pós-condições	Lista de apresentações recentes atualizada.
Fluxo principal	[FP01] – Usuário clica no botão de “Atualizar atividades recentes”. [FP02] – O aplicativo lista todas as atividades que estão presentes na base de dados.
Fluxo alternativo	N/A

ANEXO C – Fotos das telas tiradas do aplicativo

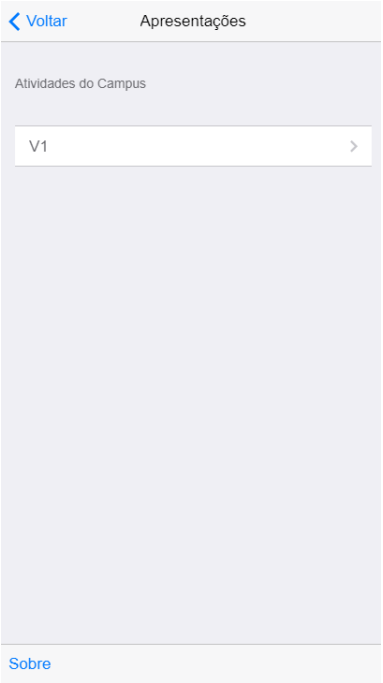
Tela principal com os campi



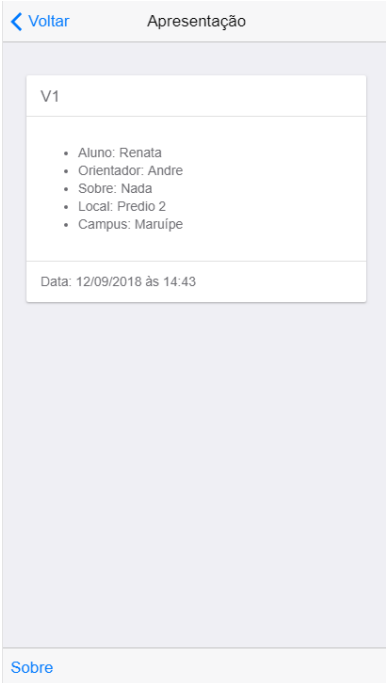
Tela informativa sobre



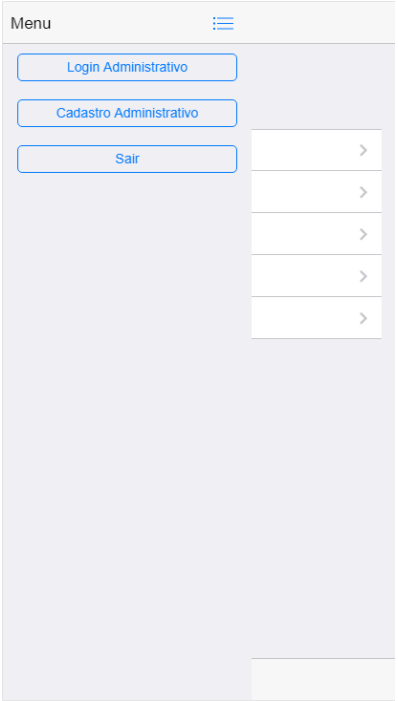
Tela apos selecionar campus



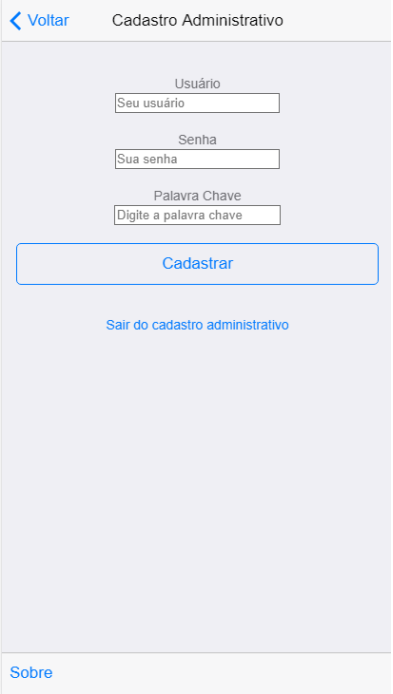
Tela após selecionar apresentação



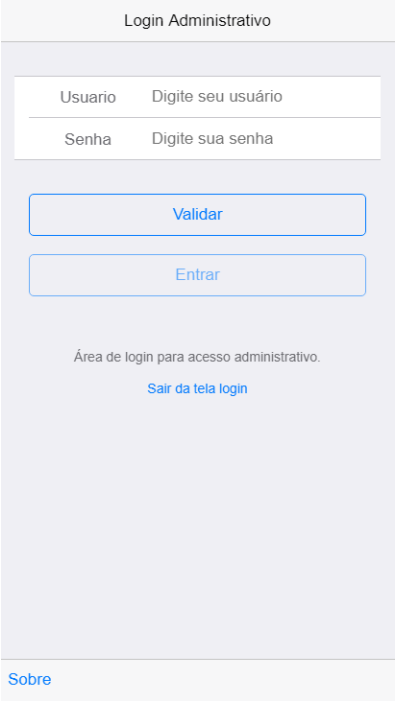
Tela menu aplicativo



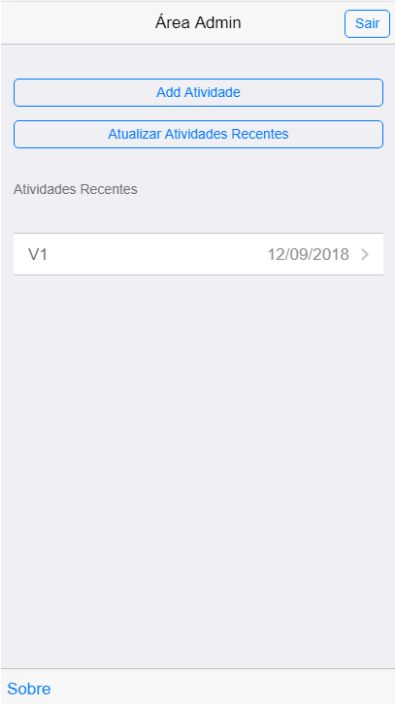
Tela cadastro administrativo



Tela login administrativo



Tela principal após login



Tela para adicionar apresentacao