



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

TÁREK HOLANDA SILVA

**DESENVOLVENDO UM APLICATIVO HÍBRIDO PARA CRIAÇÃO E
COMUNICAÇÃO DE EVENTOS EM UMA UNIVERSIDADE**

QUIXADÁ

2017

TÁREK HOLANDA SILVA

DESENVOLVENDO UM APLICATIVO HÍBRIDO PARA CRIAÇÃO E COMUNICAÇÃO DE
EVENTOS EM UMA UNIVERSIDADE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Orientador: Prof. Dr. David Sena de Oli-
veira

QUIXADÁ

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S584d Silva, Tárek Holanda.
Desenvolvendo um aplicativo híbrido para criação e comunicação de eventos em uma universidade /
Tárek Holanda Silva. – 2017.
36 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,
Curso de Sistemas de Informação, Quixadá, 2017.
Orientação: Prof. Dr. David Sena de Oliveira.

1. Eventos-Organização. 2. Desenvolvimento. 3. Aplicativos híbridos. I. Título.

CDD 005

TÁREK HOLANDA SILVA

DESENVOLVENDO UM APLICATIVO HÍBRIDO PARA CRIAÇÃO E COMUNICAÇÃO DE
EVENTOS EM UMA UNIVERSIDADE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Aprovada em: __/__/__

BANCA EXAMINADORA

Prof. Dr. David Sena de Oliveira (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Me. Regis Pires Magalhães
Universidade Federal do Ceará (UFC)

Prof. Me. Anibal Cavalcante de Oliveira
Universidade Federal do Ceará (UFC)

À minha mãe, Kátia. À minha avó, Marnice.
Aos meus amigos. Aos meus professores.

AGRADECIMENTOS

Ao Prof. Dr. David Sena de Oliveira por me orientar durante praticamente todo o meu período de estudo na universidade.

À todos os professores da Universidade Federal do Ceará (UFC) por me ensinarem durante todos os meus anos na universidade.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

Aos bibliotecários da Universidade Federal do Ceará: Eliene Maria Vieira de Moura, Francisco Edvander Pires Santos, Izabel Lima dos Santos, Juliana Soares Lima, Kalline Yasmin Soares Feitosa pela revisão e discussão da formatação utilizada neste *template*.

À minha mãe, Kátia, minha avó, Marnice, minha namorada, Beatriz, e meu amigo, Antoniel, por estarem sempre ao meu lado.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

E à Universidade Federal do Ceará (UFC) por proporcionar uma experiência incrível de conhecimento e educação.

“É bem mais difícil julgar a si mesmo que julgar os outros.”

(Antoine de Saint-Exupéry)

RESUMO

Um dos problemas encontrados na Universidade Federal do Ceará (UFC) está no processo de divulgação e acompanhamento de eventos da universidade. Este problema está diretamente relacionado a falta de eficiência e flexibilidade, e a dificuldade de *feedback* no processo que vai da criação à execução do evento. Como tentativa de disponibilizar uma melhor solução para este problema, este trabalho visa o desenvolvimento de um aplicativo híbrido que ajude a comunidade acadêmica na divulgação e acompanhamento de eventos da universidade. Dessa forma, será possível criar eventos no aplicativo para convidar pessoas, ter um controle da quantidade de presentes e notificar convidados em caso de mudanças, sem que isso seja difícil ou demorado.

Palavras-chave: Eventos. Aplicativo híbrido. Divulgação de eventos. Acompanhamento de eventos. Desenvolvimento.

ABSTRACT

One of the problems found in the University Federal of Ceará (UFC) is on the process of announcement and tracking event of an university. This problem is directly related to the lack of efficiency and flexibility, and the difficult feedback on the process that goes from creation to the execution of an even. As an attempt to provide a better solution to this problem, this project main goal is develop an hybrid application that helps the academic community to announce and to track the university events. Thus, it will be possible to create events in the application to invite people, to have the control of the amount of presents and to notify the guests in case of changes, without this being hard or time-consuming.

Palavras-chave: Events. Hybrid Application. Events Announcement. Events tracking. Development.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Convite para palestra na UFC via papel colado no mural | 12 |
| Figura 2 – Convite para célula na UFC via <i>Facebook</i> | 13 |
| Figura 3 – Ferramentas de desenvolvimento para aplicativos híbridos | 17 |
| Figura 4 – Código de uma página com <i>Ionic</i> | 18 |
| Figura 5 – Código de uma página com <i>AngularJS</i> | 19 |
| Figura 6 – Tela inicial - versão 1.0 x versão 1.1 | 22 |
| Figura 7 – Tela de evento - versão 1.0 x versão 1.1 | 23 |
| Figura 8 – Esquema do banco de dados do sistema | 24 |
| Figura 9 – Pergunta de confirmação de presença da entrevista de validação | 27 |
| Figura 10 – Pergunta sobre convites da entrevista de validação | 27 |
| Figura 11 – Pergunta sobre lembrete da entrevista de validação | 27 |

SUMÁRIO

| | | |
|-------|--|----|
| 1 | INTRODUÇÃO | 11 |
| 2 | TRABALHOS RELACIONADOS | 14 |
| 3 | OBJETIVOS | 16 |
| 3.1 | Objetivo geral | 16 |
| 3.2 | Objetivos específicos | 16 |
| 4 | FUNDAMENTAÇÃO TEÓRICA | 17 |
| 4.1 | <i>Front end, back end e banco de dados</i> | 18 |
| 4.1.1 | <i>Front end com Ionic e AngularJS</i> | 18 |
| 4.1.2 | <i>Back end e banco de dados com PHP</i> | 20 |
| 4.2 | Aplicativo híbrido | 20 |
| 4.3 | Eventos no Campus Quixadá | 21 |
| 5 | DESENVOLVIMENTO DO PROJETO | 22 |
| 5.1 | Definição dos Requisitos | 22 |
| 5.2 | Definição e validação dos requisitos funcionais | 23 |
| 5.3 | Coleta de dados de eventos | 23 |
| 5.4 | Implementação do banco de dados e do back end com <i>PHP</i> | 24 |
| 5.5 | Implementação do <i>front end</i> com <i>Ionic e AngularJS</i> | 25 |
| 5.6 | Integração do front end com o back end | 25 |
| 5.7 | Disponibilização e utilização | 25 |
| 5.8 | Verificação e avaliação de utilização | 25 |
| 6 | RESULTADOS | 26 |
| 7 | TRABALHOS FUTUROS | 28 |
| | REFERÊNCIAS | 29 |
| | APÊNDICE A – PRÓTOTIPO INICIAL | 30 |
| | APÊNDICE B – PRÓTOTIPO FUNCIONAL | 32 |
| | APÊNDICE C – DOCUMENTOS DE REQUISITOS | 35 |

1 INTRODUÇÃO

Dispositivos móveis estão se tornando cada vez mais presentes na sociedade atual (DEGROSSI et al., 2013). Os serviços on-line estão se tornando mais utilizados por pessoas e empresas. Estes serviços são disponibilizados através de ferramentas on-line com objetivo de aumentar a praticidade no dia-a-dia de seus usuários. Com o aumento do uso da tecnologia da informação (TI) em produtos e serviços, as empresas de TI passaram a ter alcance mundial, com atendimento customizado ao cliente.

As chamadas *self-service technologies* modificaram o processo de compra ou prestação de serviço, mantendo o foco no consumidor. Procedimentos como o online banking facilitam o dia a dia das pessoas, permitindo que clientes realizem operações financeiras no momento mais propício, sem precisar se deslocar até o endereço geográfico de seu banco (BITNER, 2001). Como atual exemplo, pode-se citar o *Nubank*, que é uma das mais conhecidas empresas de gerenciamento financeiro no Brasil e que trabalha cem por cento virtualmente com seus clientes. Nele o gerenciamento de contas é feito exclusivamente através do aplicativo.

O presente trabalho visa contribuir para solução de um problema de comunicação que envolve a divulgação e o acompanhamento dos diversos eventos na comunidade acadêmica. O estudo de caso é realizado na Universidade Federal do Ceará, no Campus Quixadá.

Existem problemas no processo de divulgação e acompanhamento dos eventos que acontecem na UFC. Após entrevistas com alunos e servidores, foram definidas três métricas para avaliação das soluções. As métricas são: eficiência, flexibilidade e dificuldade de feedback no processo que vai da criação à execução do evento.

O Campus Quixadá, atualmente, possui algumas estratégias de divulgação. A primeira utiliza cartazes colados nos murais do Campus e postagens no Facebook para o processo de divulgação. A utilização de cartazes como meio de convites para eventos permite que todas as pessoas que passem pelo mural na universidade os vejam e saibam do evento (Figura 1). O desafio consiste em garantir que esses convites sejam percebidos e vistos em tempo hábil. Não há um meio de *feedback* dos convidados para o criador do evento. No caso de alguma mudança será necessário atualizar o cartaz. No processo de atualização não existe modo simples de sinalizar as mudanças ou garantia de alcance das novas informações.

A segunda estratégia de divulgação utiliza a ferramenta de eventos do *Facebook* que permite que qualquer pessoa crie um evento e convide outros usuários. É realizada uma publicação em um grupo ou marcação direta para que todos os convidados sejam notificados

Figura 1 – Convite para palestra na UFC via papel colado no mural



Fonte: mural do Campus Quixadá.

e recebam o *link* para a página do evento. Os convites dos eventos, normalmente criados por alunos, são enviados para o grupo do *Facebook* do Campus Quixadá. Além disso, o *Facebook* permite que os usuários comentem a postagem, que pode servir como *feedback* para a pessoa que realizou o convite.

Na Figura 2 é mostrado um convite para a célula de Italiano. A Figura 2 não utiliza a ferramenta de eventos do *Facebook*. É uma postagem com um endereço para uma *Fanpage*. A ferramenta de eventos quando utilizada, permite que os usuários confirmem sua participação e interesse em um evento futuro. Entretanto, ela não permite a criação de eventos periódicos, mas apenas de eventos únicos. Por isso, grupos de reunião periódica não costumam utilizá-la. Acontece, de forma recorrente, que os criadores dos eventos, semanalmente fazem uma postagem convidando ou lembrando as pessoas, o que gera uma sobrecarga do canal de comunicação.

Este trabalho tem como foco a problemática da comunicação de eventos na comunidade acadêmica. Através de entrevistas, o processo foi modelado e para validar a solução proposta foi desenvolvido um aplicativo móvel. O aplicativo fará uma gestão colaborativa de eventos. Ele deve permitir ao usuário criar e editar eventos, convidar outros usuários, confirmar

Figura 2 – Convite para célula na UFC via *Facebook*



Fonte: grupo do *Facebook* do Campus Quixadá.

e cancelar presença. Deve notificar o usuário que participa de um evento, caso haja mudanças e lembrá-lo quando estiver próximo ao horário do evento. Será desenvolvido com tecnologia híbrida, ou seja, um aplicativo implementado com linguagens de programação para web e que são executados como aplicativos móveis nativos. No decorrer do trabalho, serão apresentados os requisitos coletados, o conteúdo mais relevante colhidos nas entrevistas, a evolução do modelo de interação humano computador e requisitos que aconteceram durante o desenvolvimento da solução.

2 TRABALHOS RELACIONADOS

Este trabalho diferencia dos aplicativos com funcionalidades similares existentes por unir pontos positivos de todos eles sem possuir os mesmos problemas. Nas atuais soluções, um não possui eventos dinâmicos, outro não permite a frequência de eventos, outro limita a quantidade de eventos, e outro não possui esses problemas é pago.

A idéia deste trabalho é ajudar a comunidade acadêmica a gerir seus eventos colaborativamente. Dois importantes projetos foram fundamentais para a implementação e motivação deste trabalho, contribuindo assim na criação e desenvolvimento deste trabalho como um todo.

Cardoso e Pinto (2011) realizaram um estudo sobre como as redes sociais tornaram-se uma ferramenta de divulgação de produtos e serviços no meio digital. Em apenas dois anos, entre 2009 e 2011, o *Facebook* apresentou uma taxa de crescimento de usuários de duzentos e cinquenta e oito por cento (CARDOSO; PINTO, 2011).

Mesmo com o crescimento constante na utilização de redes sociais para divulgação de eventos, ainda existe uma carência na área. Cardoso e Pinto (2011) também propõem o desenvolvimento de um site que possibilite a divulgação de eventos no meio digital de uma maneira simples e eficiente. O foco é a facilidade de divulgação e compartilhamento.

A proposta de Cardoso e Pinto (2011) busca facilitar a divulgação e compartilhamento de eventos nas redes sociais, mas não resolve os desafios encontrados na utilização das soluções atuais da comunidade acadêmica da UFC de Quixadá, citados na introdução, que são a falta de eficiência, a falta de flexibilidade e a dificuldade de feedback.

Outro projeto que contribui para o desenvolvimento deste trabalho é a tese de bacharelado de Vandecandelaere (2015). Ele desenvolve um aplicativo híbrido que tem como objetivo ajudar os estudantes de sua universidade a gerenciar e divulgar eventos, e encontrar suas localizações com maior facilidade e organização.

O aplicativo possui quatro funções principais. Permitir que o usuário divulgue um evento pelo aplicativo, envie uma mensagem a outros usuários, anuncie a venda de produtos na seção de propaganda e veja uma lista de todos os prédios e instalações no campus.

Além do objetivo semelhante com o deste trabalho, o aplicativo de Vandecandelaere (2015) foi desenvolvido utilizando Ionic e AngularJS para implementação do front end, e PHP para implementação do back end, assim como ocorrerá com este trabalho.

Vandecandelaere (2015) demonstrou de maneira simplificada o passo a passo para desenvolver um aplicativo híbrido, contribuindo para o entendimento do conceito de aplicativo

híbrido para o desenvolvimento deste trabalho.

A principal diferença entre este trabalho e o de Vandecandelaere, além de não possuir as funções de anúncio e venda de produtos, é que este trabalho propõe a divulgação de eventos que acontecem regularmente, como aulas, monitorias e células. O aplicativo de Vandecandelaere, assim como a ferramenta de eventos do Facebook, permite apenas a criação e divulgação de eventos únicos.

3 OBJETIVOS

3.1 Objetivo geral

Apresentar o desenvolvimento de um sistema de gestão de eventos, desde sua concepção, requisitos, implementação e aceitação pelo cliente.

3.2 Objetivos específicos

Definir requisitos do sistema, implementar telas do sistema utilizando *Ionic*, implementar funcionalidades do *front end* utilizando *AngularJS*, implementar funcionalidades do *back end* utilizando *PHP*, implementar a persistência do aplicativo utilizando *MySQL*, integrar todas as partes do sistema e possibilitar a instalação e utilização do sistema.

4 FUNDAMENTAÇÃO TEÓRICA

Um aplicativo possui várias arquiteturas de software implementadas, geralmente em diferentes linguagens de programação, que trabalham integradas para que todo o sistema funcione: *front end*, *back end* e banco de dados.

Front end e *back end* são termos generalizados que se referem às etapas inicial e final de um processo. O *front end* é responsável por coletar a entrada do usuário em várias formas e processá-la para adequá-la a uma especificação em que o *back end* possa utilizar.

Back end é o código do lado do servidor responsável pelo gerenciamento de manipulação de dados, assim como sua disponibilidade para utilização no *front end*.

Banco de dados, é um sistema computadorizado de manutenção de registros. Pode ser equivalente a um armário eletrônico de arquivamento, ou seja, um repositório de arquivos de dados computadorizados (DATE, 2004).

Algumas das principais ferramentas de desenvolvimento de *front end* de aplicativos híbridos são *Onsen UI*, *Mobile Angular UI* e *Ionic Framework*. Todas elas trabalham integradas com *AngularJS*, o que indica quão poderosa essa ferramenta é para esse tipo de desenvolvimento.

Figura 3 – Ferramentas de desenvolvimento para aplicativos híbridos



Fonte: onsen.io, mobileangularui.com, ionicframework.com.

As principais ferramentas de desenvolvimento de *software* que serão utilizadas para a implementação deste trabalho serão o *Ionic Framework*, *AngularJS*, e *PHP*. As ferramentas responsáveis pelo *front end* do sistema serão *Ionic* e *AngularJS*. O *PHP* será o responsável pelo *back end*. A integração entre o *front end* e o *back end* serão feitas em *PHP* e *AngularJS*.

Quanto a abordagens para o desenvolvimento de aplicativos, Vandecandelaere (2015)

cita três principais: nativos, *web* e híbrido. Ele descreve uma visão panorâmica do desenvolvimento de aplicativos híbridos e os seus componentes, assim como as diferenças, similaridades, vantagens e desvantagens entre os aplicativos nativos, *web* e híbridos. A escolha dependerá dos recursos, tempo de desenvolvimento e objetivo do aplicativo (VANDECANDELAERE, 2015).

Nas seções a seguir, são apresentadas maiores informações sobre a escolha pela abordagem híbrida de desenvolvimento, bem como a arquitetura do *software* a ser desenvolvido, especialmente seu *front end*, *back end* e banco de dados.

4.1 *Front end*, *back end* e banco de dados

4.1.1 *Front end* com Ionic e AngularJS

A estrutura do *front end* é composta pelos *layouts* do aplicativo, abrangendo tudo aquilo que o usuário vê. Assim como as funcionalidades que interagem diretamente com o usuário, como o clique de um botão, a abertura de uma janela modal, ou seleção de uma lista, por exemplo.

Ionic é um pacote de desenvolvimento de *software HTML5* que ajuda a desenvolver aplicativos híbridos móveis muito parecidos com aplicativos nativos usando tecnologias web como *HTML*, *CSS* e *JavaScript*. Uma grande vantagem e uma das razões pelo qual foi escolhido para desenvolvimento deste trabalho é que, IONIC FRAMEWORK SITE OFICIAL (2017):

Ionic está voltado principalmente para a aparência e interação do seu aplicativo. Isso significa que não é um substituto para Phonegap ou o seu framework favorito JavaScript. Em vez disso, Ionic simplesmente se encaixa bem com estes projetos, a fim de simplificar a uma grande parte da sua aplicação: o *front end*.

Figura 4 – Código de uma página com *Ionic*

```

1 <ion-model-view>
2   <ion-header-bar class="bar bar-header bar-dark">
3     <h1 class="title" ng-if="event.class">Editar Evento/h1>
4     <h1 class="title" ng-if="event.class">Editar Aula/h1>
5     <button class="button button-clear button-light icon ion-close-circled" ng-click="modalDismissEvent()">
6     </button>
7   </ion-header-bar>
8   <ion-content scroll="true" class="text-center">
9     
11     <form name="myForm" ng-submit="saveEvent()" class="text-left">
12       <div class="list">
13         <label class="item item-input item-stacked-label">
14           <span class="input-label" ng-if="event.class">Evento/</span>
15           <span class="input-label" ng-if="event.class">Aula/</span>
16           <input type="text" name="name" placeholder="Evento..." ng-model="event.name"
17             ng-if="event.class" required/>
18           <input type="text" name="name" placeholder="Aula..." ng-model="event.name"
19             ng-if="event.class" required/>
20         </label>
21         <label class="item item-input item-stacked-label">
22           <span class="input-label">Descrição/</span>
23           <input type="text" placeholder="Descrição..." ng-model="event.description" required/>
24         </label>
25         <label class="item item-input item-stacked-label">
26           <span class="input-label">Local/</span>
27           <input type="text" placeholder="Local..." ng-model="event.local" required/>
28         </label>
29         <label class="item item-input">
30           <span class="input-label">Hora/</span>
31           <input type="time" ng-model="event.time" required/>
32         </label>
33         <label class="item item-input">
34           <span class="input-label">Data/</span>
35           <input type="date" ng-model="event.date" required/>
36         </label>
37         <label class="item item-input item-select">
38           <div class="input-label">Frequência/</div>
39           <select data-ng-options="option.name for option in options" data-ng-model="selected"></select>
40         </label>
41         <ion-list ng-if="event.often == 3" class="text-center">
42           <ion-checkbox ng-model="event.days.zero" class="daysCheckbox" ng-change="checkWeek()">
43           Sáb/Sexta

```

Fonte: autor (2017).

Ionic será a ferramenta base utilizada para o desenvolvimento do aplicativo. O *framework* será responsável pela interface, por grande parte do CSS, pela interação com o usuário e pela compilação do aplicativo, para possibilitar a execução do mesmo em três diferentes plataformas, *web*, *iOS* e *Android*, sendo assim, o que o define como um aplicativo híbrido.

Na Figura 4 é possível observar o código de uma página criada em *HTML* utilizando o *Ionic*. Algumas *tags* de *divs* preparadas e prontas para utilização como *ion-modal-view*, *ion-header-bar* e *ion-content*, e ainda classes como *bar-dark*, *button-clear* e *ion-close-circled*, facilitam muito o desenvolvimento.

Atualmente, o *Ionic* exige a integração e desenvolvimento utilizando *AngularJS*, a fim de trabalhar em seu pleno potencial. Segundo o próprio site, ANGULARJS SITE OFICIAL (2017):

AngularJS é uma ótima ferramenta para se trabalhar, extremamente poderosa e totalmente focada a velocidade e desempenho. HTML é ótimo por declarar documentos estáticos, mas falha quando tentamos usá-lo para declarar visualizações dinâmicas em aplicações web. AngularJS permite estender o vocabulário HTML para a sua aplicação. O ambiente resultante é extraordinariamente expressivo, legível e rápido para desenvolver.

AngularJS utiliza extensivamente *TypeScript* para desenvolvimento de aplicativos (RAVULAVARU, 2015). *AngularJS* será a ferramenta utilizada para implementação das funcionalidades do *front end* do aplicativo e fará, junto com *PHP*, a integração do *front end* com o *back end*.

Figura 5 – Código de uma página com *AngularJS*

```

    .state('tab.login', {
      url: '/login',
      views: {
        'tab.login': {
          templateUrl: 'templates/tab-login.html',
        }
      }
    });

$stateProvider.otherwise('/tabs/dashboard');

$httpProvider.interceptors.push('APIInterceptor');
})

.run(function ($rootScope, $state, LoginService, Backand) {
  $rootScope.admin = {
    user: "f91b33e0-9928-473d-9b1a-457e2e22fbcd",
    pass: "b35e7553-2106-11e7-8124-06bcf2b21c8c",
    url: "https://api.backand.com/"
  }
  function unauthorized() {
    console.log("user is unauthorized, sending to login");
    $state.go('tab.login');
  }
  function signout() {
    LoginService.signout();
  }
  $rootScope.$on('unauthorized', function () {
    unauthorized();
  });
  $rootScope.$on('$stateChangeSuccess', function (event, toState) {
    if (toState.name == 'tab.login') {
      signout();
    }
    else if (toState.name != 'tab.login' && Backand.getToken() === undefined) {
      unauthorized();
    }
  });
});

```

Fonte: autor (2017).

Na Figura 5 é possível ver um controlador do *AngularJS* que ajuda e facilita o desenvolvimento de aplicativos híbridos e a integração com o *front end* através do *Ionic*.

4.1.2 *Back end e banco de dados com PHP*

PHP é uma linguagem de programação flexível, que pode ser utilizada tanto para desenvolvimento no lado do servidor quanto no lado do cliente. (TATROE *et al.*, 2013)

PHP é uma simples e ainda poderosa linguagem designada a criar conteúdo HTML. Uma das características mais importantes do *PHP* é o seu amplo suporte para bancos de dados. *PHP* suporta todas as principais bases de dados (incluindo MySQL, PostgreSQL, Oracle, Sybase, MS-SQL, DB2 e bancos de dados compatível com ODBC). Mesmo os bancos de dados mais recentes como SQLite e MongoDB também são suportados. Com *PHP*, a criação de páginas web com conteúdo dinâmico a partir de um banco de dados é extremamente simples.

O *PHP* será utilizado para implementação das funcionalidades do back end do aplicativo e para integração do mesmo com o banco de dados, sendo responsável por receber as informações do banco de dados e disponibilizá-las dentro do aplicativo, possibilitando a utilização destes no *front end*.

4.2 Aplicativo híbrido

Um aplicativo híbrido é um aplicativo implementado com linguagens de programação para web e que são executados similarmente a aplicativos móveis nativos, assim como web. Para Vandecandelaere (2015),

aplicativos híbridos unem tecnologias web / práticas com aplicativos nativos. O aplicativo híbrido é desenvolvido utilizando tecnologias web comuns como HTML5, JavaScript e CSS3. Uma vez que o aplicativo está construído, um framework como Phonegap ou AppCelerator são utilizados para compilar o aplicativo a ser executado dentro de uma "WebView". Depois que o aplicativo é construído, ele é semelhante a um aplicativo nativo em termos de distribuição nas lojas de aplicativos, instalação e utilização.

O autor mostrou muitas vantagens no desenvolvimento de um aplicativo híbrido em relação a um aplicativo nativo, o que fez com este trabalho também escolhesse a implementação de um aplicativo híbrido, mesmo considerando que um aplicativo nativo tem o desempenho um pouco mais eficiente do que um híbrido. O principal benefício no desenvolvimento de um aplicativo híbrido é a redução de custo e tempo (VANDECANDELAERE, 2015). O autor também contribuiu na escolha do *framework Ionic* juntamente com *AngularJS*, por afirmar que, (VANDECANDELAERE, 2015)

possui várias ótimas propriedades que, quando combinadas, formam uma rede de transporte ideal para uma aplicação híbrida. Também é importante notar que

AngularJS abrange completamente o código JavaScript, assim não há métodos adicionais de compilação necessários para lidar com AngularJS.

Os resultados da análise de Degrossi (et al., 2013) sobre o tempo de implementação *Android* e *HTML5*, também justificam a escolha do *framework Ionic* para o desenvolvimento deste trabalho. Esses resultados sugerem que o desenvolvimento de aplicações *HTML5* possui, em média, menor tempo de implementação quando comparado com o desenvolvimento de aplicativos *Android* na linguagem Java (DEGROSSI et al., 2013). Sabendo que *Ionic* é totalmente baseado e utilizado juntamente com *HTML5*, e tendo em vista a comprovação de Degrossi (et al., 2013) que mostrou que a tecnologia *HTML5* possui uma média de tempo para implementação aproximadamente trinta e cinco por cento menor que a tecnologia *Android*.

A escolha das ferramentas mais adequadas para implementação de um projeto são fundamentais para um bom desempenho no desenvolvimento do projeto. Logo, escolher a melhor abordagem para o desenvolvimento de uma aplicação depende das necessidades específicas da organização e de diferentes parâmetros (DEGROSSI et al., 2013). Portanto, para implementação deste trabalho, não será feito o desenvolvimento de um aplicativo nativo, e sim um aplicativo híbrido, utilizando o *framework Ionic* com *AngularJS*.

4.3 Eventos no Campus Quixadá

Os eventos que ocorrem na UFC Capus Quixadá, possuem diversas categorias e objetivos. As categorias de eventos podem ser divididas em palestras, cursos, oficinas, apresentação de artigos, monitoria de disciplinas, células de aprendizagem cooperativa, encontros artísticos, treinos esportivos, campeonatos e competições internas e externas.

Os objetivos de cada evento variam entre difundir atividades de ensino, pesquisa, extensão, arte e cultura, ajudar no desempenho de disciplinas, desenvolver conhecimento em novas áreas, e promover a interação, diversão e prática de esportes entre a comunidade acadêmica.

O trabalho apresentado não lidará com submissões de trabalhos para os Encontros Universitários, solicitação de bolsas de estudo, criação de células de aprendizagem cooperativa ou formação de grupos de estudos. O trabalho apresentado tem como intuito a divulgação de eventos ocorrentes na UFC Campus Quixadá.

5 DESENVOLVIMENTO DO PROJETO

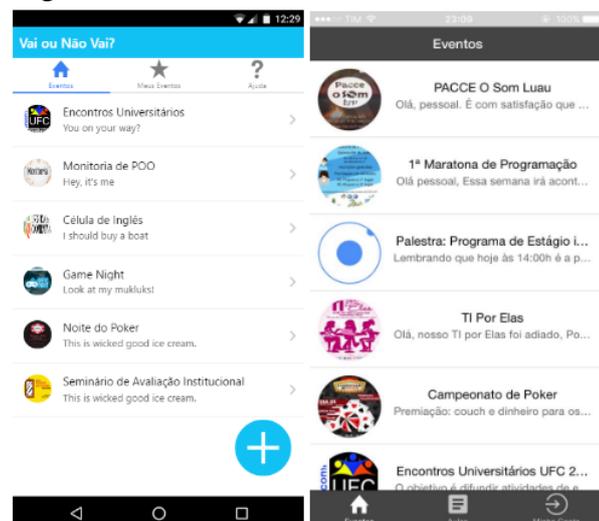
5.1 Definição dos Requisitos

Os requisitos do aplicativo foram definidos em várias reuniões com um profissional com experiência em desenvolvimento de aplicativos híbridos, atualmente trabalhando na empresa norte americana *Heavy Connect*. Raimundo Leonel Junior é Bacharel em Sistemas de Informação, formado na UFC Quixadá. Como *stakeholder*, Leonel acompanhou todo o desenvolvimento do aplicativo, desde a definição dos requisitos até a utilização pela comunidade acadêmica da UFC Quixadá.

Após a definição dos requisitos, uma equipe de profissionais com experiência em desenvolvimento de aplicativos híbridos formados em diferentes cursos da UFC Quixadá, fizeram a verificação dos requisitos. Permitindo os mesmos avaliarem os requisitos definidos, buscando filtrar os requisitos que mais agregam valor ao software.

Requisitos como a confirmação de presença horas antes e a notificação constante de mudanças no evento, perderam força durante o processo de validação e testes de usuários. Já alguns requisitos que não eram considerados antes, como a habilidade de compartilhar um evento fora do aplicativo e o convite externo, via Whatsapp ou Facebook, por exemplo, foram adicionados a partir de pedidos dos próprios usuários.

Figura 6 – Tela inicial - versão 1.0 x versão 1.1

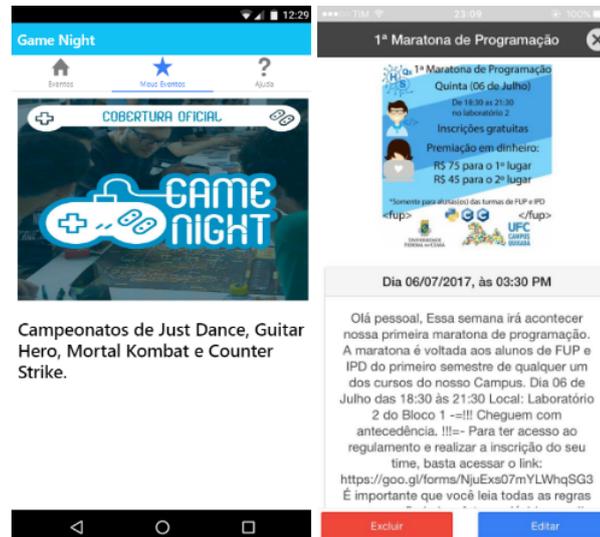


Fonte: autor (2017).

Além dos requisitos, as telas do aplicativo também sofreram mudanças a partir da sequência de testes dos usuários interessados, como é possível ver a seguir na Figura 6 e na

Figura 7. O tema do aplicativo ganhou um tom mais escuro, o menu de abas foi movido para baixo, entre outros detalhes de layout que é possível perceber observando as telas da primeira versão do aplicativo e as telas da versão atual.

Figura 7 – Tela de evento - versão 1.0 x versão 1.1



Fonte: autor (2017).

5.2 Definição e validação dos requisitos funcionais

Os requisitos funcionais do sistema também foram definidos a partir de várias reuniões com o stakeholder do projeto. O processo levou em consideração todas as funcionalidades requeridas ao sistema discutidas e documentadas nas reuniões. A partir dessas funcionalidades os requisitos foram definidos de acordo com as funcionalidades que agregam mais valor ao sistema.

Após definição, os requisitos do sistema foram validados através de várias entrevistas feitas com prováveis usuários do aplicativo, alunos que atualmente divulgam eventos no Campus utilizando outras soluções, citadas na introdução.

5.3 Coleta de dados de eventos

Para a coleta das informações dos atuais eventos na UFC Quixadá, foram feitas algumas entrevistas com um servidor da secretaria acadêmica para que este pudesse repassar todas as informações necessárias para criação da linha do tempo inicial do aplicativo.

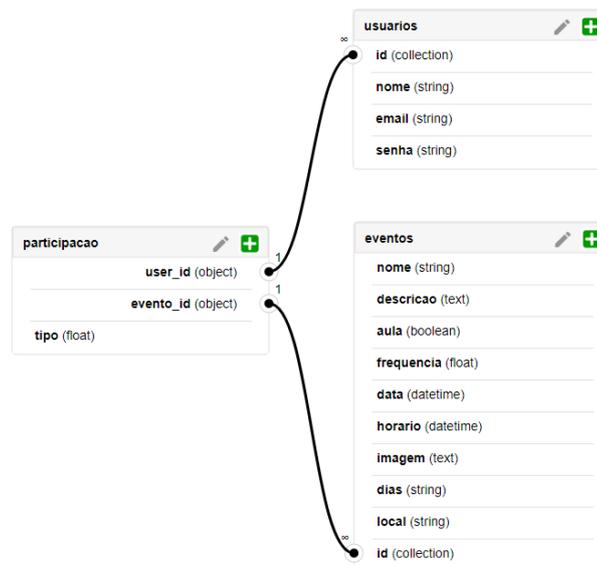
Foram coletadas também informações de atividades pelo grupo da UFC Quixadá no Facebook, e através do formulário feito para coleta inicial de eventos, que foi preenchido por

parte da comunidade acadêmica da UFC Quixadá, buscando também informações dos atuais eventos na universidade.

5.4 Implementação do banco de dados e do back end com *PHP*

No banco de dados, cada evento possui um número identificador único, um nome, uma descrição, uma imagem, um período semanal ou data e um local. Assim, utilizando *PHP*, o aplicativo pode receber todas as atividades do banco, ordená-las de diferentes formas e mostrá-las no aplicativo. Além disso, cada usuário possui um número identificador único, um nome, um email e uma senha. As participações ligam os eventos aos usuários e podem ser como participante ou como coordenador.

Figura 8 – Esquema do banco de dados do sistema



Fonte: autor (2017).

Na Figura 8 é possível ver o modelo do esquema do banco de dados do aplicativo. Os dados coletados sobre os eventos ocorrentes na UFC Quixadá foram adicionados no banco de dados para que possam ser utilizados no aplicativo.

Os usuários também podem adicionar eventos utilizando o aplicativo, o que fará com que novos dados de eventos sejam salvos no banco de dados através de funções implementadas em *PHP*.

5.5 Implementação do *front end* com *Ionic* e *AngularJS*

A implementação do *front end* foi feita através do *framework Ionic*, que é o responsável por toda a interface do aplicativo juntamente com um *CSS* personalizado, e através do *AngularJS*, que é responsável por todas as funcionalidades do *front end* e pela integração do mesmo com o *back end*.

5.6 Integração do *front end* com o *back end*

A integração do *front end* com o *back end* consiste simplesmente em adicionar o código *PHP* nos arquivos *HTML*, de modo a tornar possível a comunicação entre o mesmo e os arquivos *JavaScript* do *AngularJS*, para então disponibilizar e mostrar os dados requeridos na tela do aplicativo através do *Ionic*.

5.7 Disponibilização e utilização

Pelo fato da conta de desenvolvedor da *Apple Store* ser muito cara, a primeira versão do aplicativo estará disponível para download apenas para *Android*, na *GooglePlay*. Após o término do desenvolvimento da primeira versão o aplicativo será publicado pelo desenvolvedor e é esperado que o aplicativo passe a ser utilizado pela comunidade acadêmica da UFC, Campus Quixadá.

5.8 Verificação e avaliação de utilização

Após a disponibilidade da primeira versão do aplicativo na *Google Play* é previsto que seja realizada uma verificação da quantidade de usuários que após baixar o aplicativo, continuam com o aplicativo em seu dispositivo, se eles continuam utilizando o aplicativo e frequência de utilização.

Após a verificação citada acima, será feita uma entrevista, com dez perguntas em um formulário, a respeito da satisfação dos usuários em relação a utilização do aplicativo e uma pesquisa buscando melhorias e possíveis novas funcionalidades para o sistema.

6 RESULTADOS

Como citada nas seções 5.1 e 5.2, a definição dos requisitos deste trabalho foi feita através de várias reuniões com o stakeholder do projeto Leonel Junior. Ele ajudou a definir novos requisitos e a pensar em novas funcionalidades para o aplicativo.

As principais funcionalidades definidas para o aplicativo foram que o sistema permitisse o usuário ver, cadastrar, editar, mostrar e participar de um evento, assim como notificar os usuários participantes ou criadores de um evento em caso de mudanças e emitir lembretes com confirmação de presença a eles. Além disso, o sistema também deve permitir o convite de eventos e compartilhamento em outras redes sociais.

Para validar os requisitos e estimar os benefícios de um aplicativo de gerenciamento de eventos à comunidade acadêmica, foram realizadas várias reuniões com desenvolvedores experientes de aplicativos híbridos e algumas pesquisas com alunos e usuários interessados da UFC Quixadá.

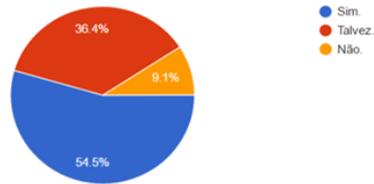
Os resultados da pesquisa mostraram que mais de noventa por cento das pessoas entrevistadas já precisou divulgar algum evento e encontrou dificuldade ou desejou que houvesse alguma maneira melhor ou mais fácil de divulgação e também mostrar interesse em eventos e que outras pessoas pudessem mostrar interesses em seus eventos. Além disso, todos os entrevistados gostariam de saber sobre todos os eventos da universidade em um único lugar.

Mais de setenta por cento dos entrevistados gostaria de poder cadastrar, editar e mostrar seus eventos, e todos eles gostariam de notificar e serem notificados sobre eventos que mostraram interesse e de saber quantas pessoas irão ao evento que criou ou que se interessou. É possível ver na representação dos gráficos abaixo, outros resultados da pesquisa.

Na Figura 9, na Figura 10 e na Figura 11, é possível ver o gráfico das respostas para as perguntas de confirmação de presença, convites para eventos e lembrete de eventos, respectivamente.

Figura 9 – Pergunta de confirmação de presença da entrevista de validação

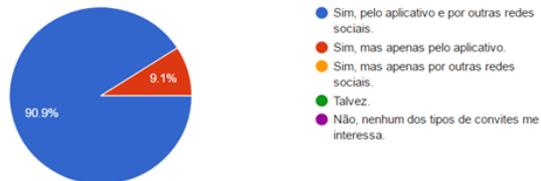
Você gostaria de confirmar presença em um evento 24 horas antes ou que pessoas confirmassem presença no seu evento?



Fonte: autor (2017).

Figura 10 – Pergunta sobre convites da entrevista de validação

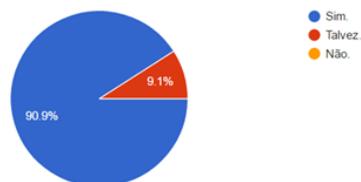
Você gostaria de convidar pessoas para um evento através do aplicativo ou de outras redes sociais?



Fonte: autor (2017).

Figura 11 – Pergunta sobre lembrete da entrevista de validação

Você gostaria de ser lembrado de um evento que você está interessado ou lembrar as pessoas que se interessaram em seu evento algum tempo antes?



Fonte: autor (2017).

7 TRABALHOS FUTUROS

O principal objetivo futuro deste trabalho é a disponibilização e utilização do aplicativo pela comunidade acadêmica da UFC Quixadá. Além disso, é esperado que seja implementada a funcionalidade de participação de eventos e de notificação, onde usuários poderão participar de eventos como coordenadores, aqueles que podem editar e excluir, ou apenas participantes. Os usuários que participarem e que coordenarem um evento serão notificados sempre que houver qualquer tipo de mudança no evento que estão ligados.

Considerando a utilização do aplicativo com o tempo, este trabalho pretende se adaptar com as necessidades da comunidade acadêmica e aumentar a praticidade na criação e divulgação de eventos da comunidade acadêmica da UFC Quixadá. No Apêndice 1 está o documento de Requisitos Inicial construído a partir das interações iniciais. No Apêndice 2 está a primeira versão de algumas telas do Protótipo Inicial de validação construído. No Apêndice 3 estão as telas atuais do Protótipo Funcional do aplicativo.

REFERÊNCIAS

ANGULARJS SITE OFICIAL. **AngularJS**. 2017. Disponível em: <<http://www.https://angularjs.org/>>. Acesso em: 10 jun. 2017.

BITNER, M. J. **Service and technology: opportunities and paradoxes**. MCB UP Ltd, v. 11, n. 6, p. 375–379, 2001.

CARDOSO, D. P.; PINTO, M. M. M. **As Redes Sociais como Ferramentas de Divulgação de Eventos**. FATEC, São Caetano do Sul, p. 30–40, 2011.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. 8. ed. São Paulo: Elsevier, 2004.

DEGROSSI, L. C.; JUNIOR, V. F.; SILVA, J. D.; BARBOSA, E. F. Uma comparação do tempo de implementação: Android vs. html5. **Workshop Latino Americano de Engenharia de Software Experimental**, São Paulo, p. 63, 2013.

IONIC FRAMEWORK SITE OFICIAL. **Ionic Framework**. 2017. Disponível em: <<http://www.https://ionicframework.com/>>. Acesso em: 10 jun. 2017.

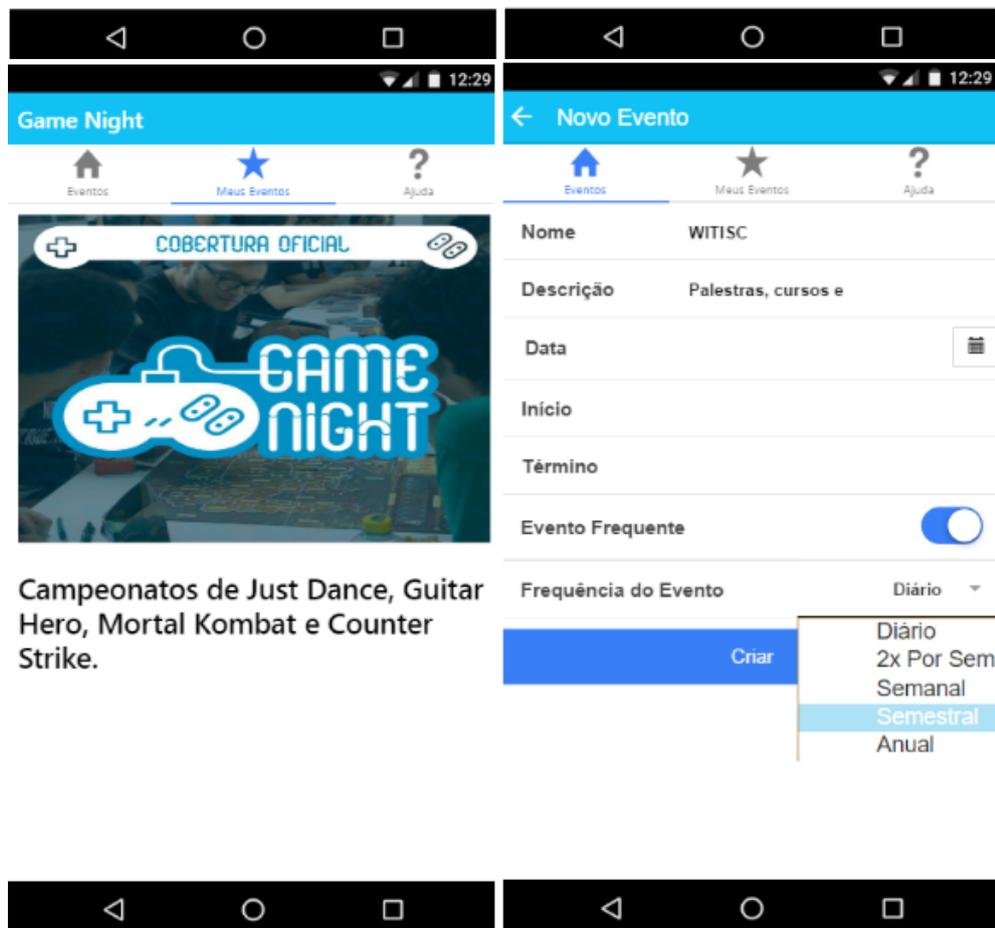
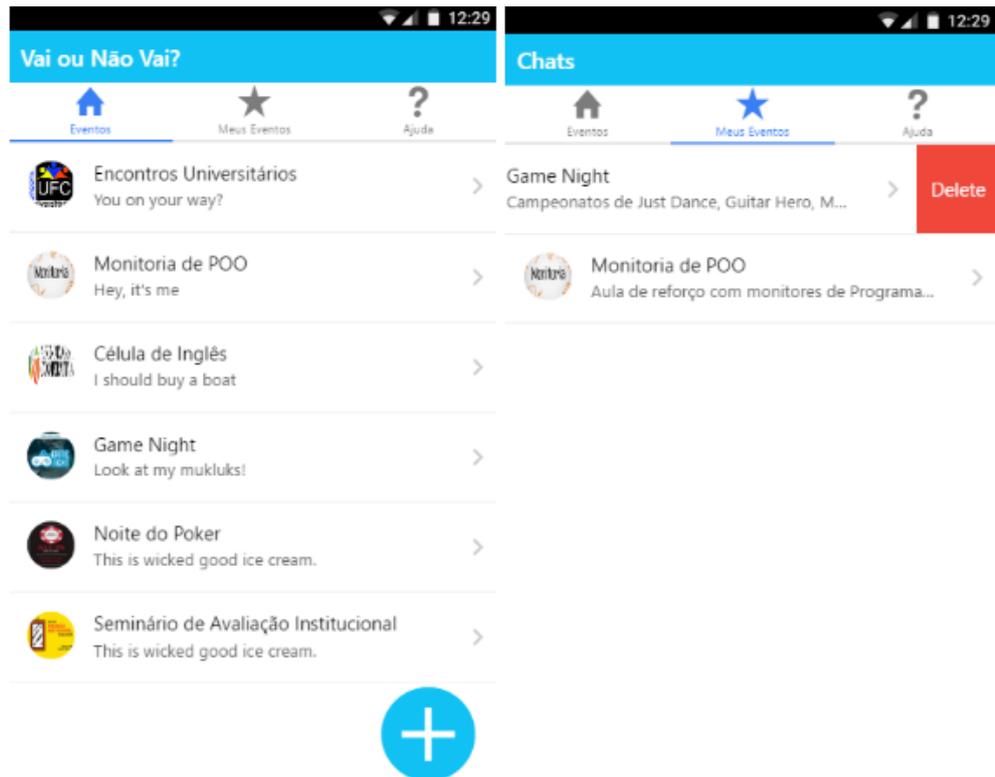
RAVULAVARU, A. **Learning Ionic - Build real-time and hybrid mobile applications with Ionic**. 2. ed. Birmingham: Packt Publishing, 2015.

TATROE, K.; MACINTYRE, P.; LERDORF, R. **Programming PHP**. 3. ed. [S.l.]: O'Reilly, 2013.

VANDECANDELAERE, B. **Developing the UDUBS-IT platform as a hybrid app with the ionic framework**. Western Cape: [s.n.], 2015. Bacharelado em Novas Mídias e Tecnologia de Comunicação.

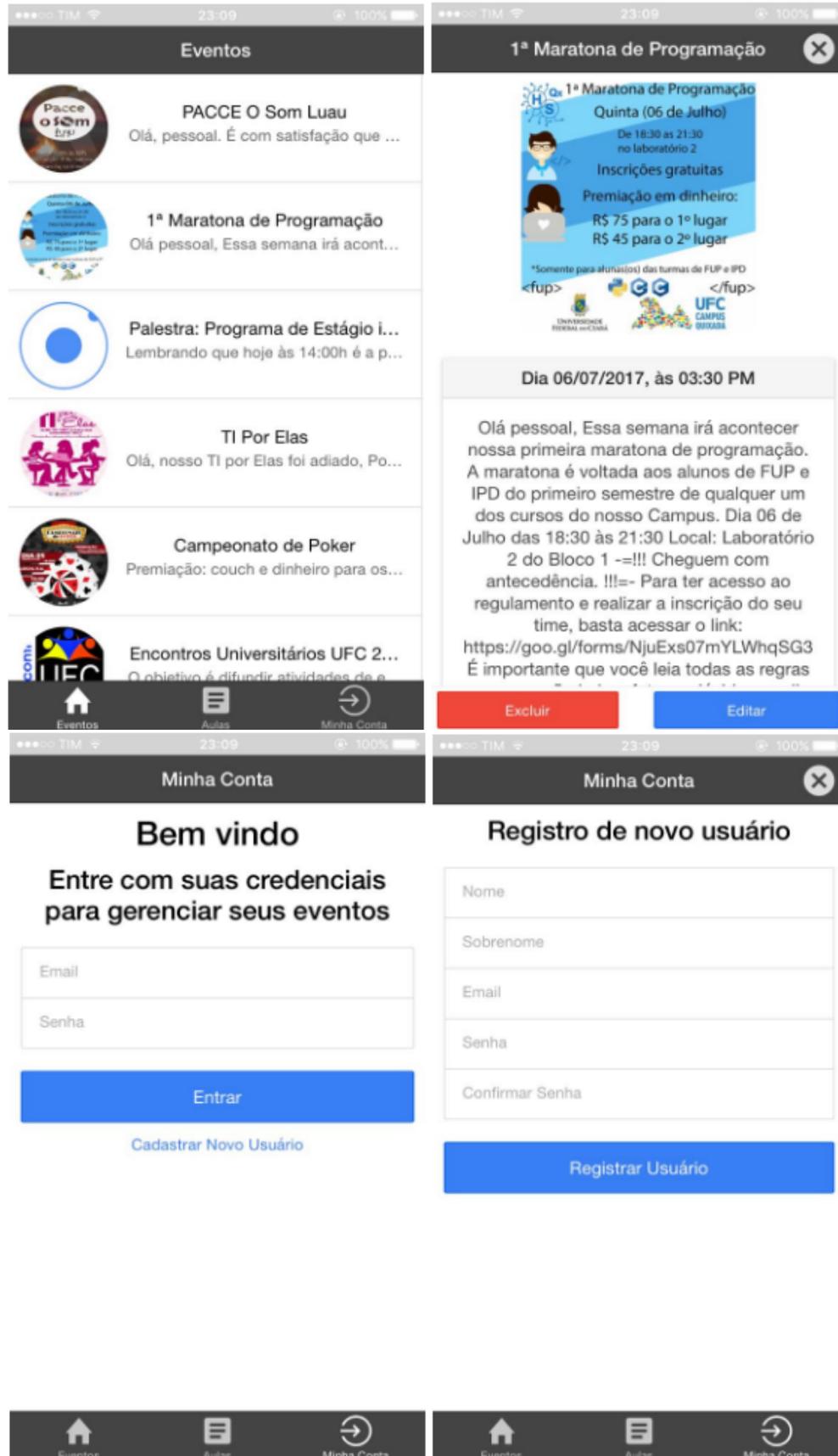
APÊNDICE A – PRÓTOTIPO INICIAL

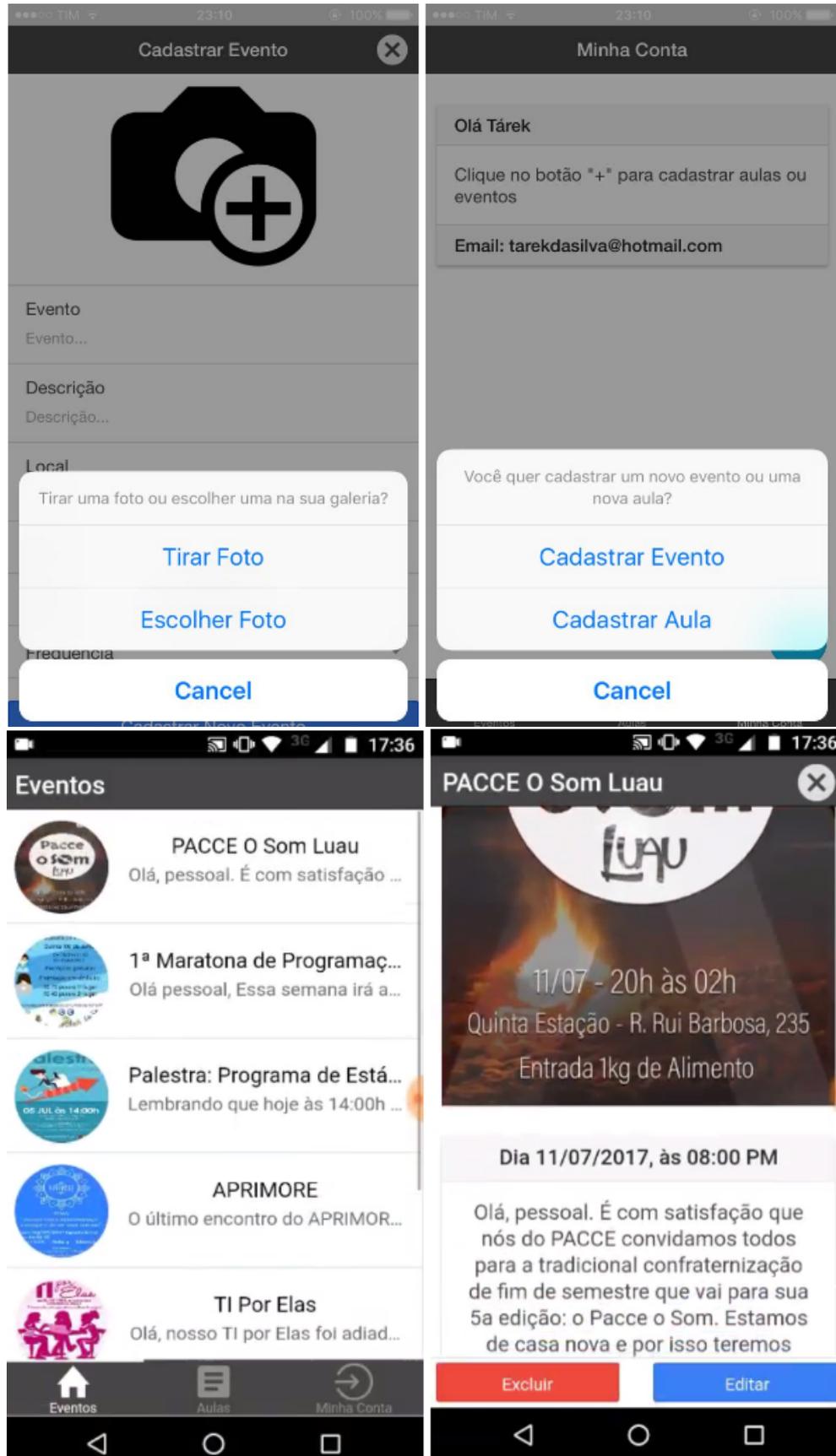
Neste apêndice, seguem algumas imagens da versão inicial do aplicativo com funcionalidades incompletas e que foi utilizado nas entrevistas na busca por melhorias e validação de requisitos.



APÊNDICE B – PRÓTOTIPO FUNCIONAL

Neste apêndice, seguem algumas imagens da versão funcional do aplicativo com funcionalidades completas e pronto para ser utilizado.





APÊNDICE C – DOCUMENTOS DE REQUISITOS

Neste apêndice, segue o documento de requisitos do projeto.

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE SISTEMAS DE INFORMAÇÃO

**Desenvolvendo um Aplicativo Híbrido para Criação e Comunicação de
Eventos em uma Universidade**

Documento de Requisitos

Tárek Holanda Silva

Orientador:

Prof. David Sena de Oliveira

QUIXADÁ

2017

Conteúdo

| | |
|---|----------|
| 1. INTRODUÇÃO | 3 |
| 1.1 VISÃO GERAL DO DOCUMENTO | 3 |
| 1.2 CONVENÇÕES, TERMOS E ABREVIACÕES | 3 |
| 1.2.1 Identificação dos requisitos | 3 |
| 1.2.2 Prioridades dos requisitos | 3 |
| 2. DESCRIÇÃO GERAL DO SISTEMA | 4 |
| 2.1 ABRANGÊNCIA E SISTEMAS RELACIONADOS | 4 |
| 3. REQUISITOS FUNCIONAIS (CASOS DE USO)..... | 4 |
| 3.1 CADASTRO | 4 |
| [RF001] Criar evento | 4 |
| [RF002] Excluir evento | 4 |
| [RF003] Alterar evento | 5 |
| 3.2 INTERFACE | 5 |
| [RF001] Visualizar evento | 5 |
| [RF002] Participar de evento | 5 |
| [RF003] Compartilhar evento | 5 |
| [RF004] Convidar para evento | 6 |
| [RF005] Visualizar presenças | 6 |
| 3.3 NOTIFICAÇÃO | 6 |
| [RF001] Notificar participante | 6 |
| [RF002] Emitir lembrete | 6 |
| [RF003] Confirmar presença | 7 |
| 4. REQUISITOS NÃO-FUNCIONAIS | 7 |
| [NF001] Usabilidade | 7 |
| [NF002] Desempenho | 7 |
| [NF003] Disponibilidade | 8 |
| 5. REFERÊNCIAS | 8 |

1. Introdução

Este documento especifica os requisitos do sistema *Vai ou Não Vai?*, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

1. **Seção 2 – Descrição geral do sistema:** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
2. **Seção 3 – Requisitos funcionais (casos de uso):** especifica todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
3. **Seção 4 – Requisitos não-funcionais:** especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
4. **Seção 5 – Referências:** apresenta referências para outros documentos utilizados para a confecção deste documento.

1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.2.1 Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[nome da subseção. identificador do requisito]

Por exemplo, o requisito funcional [Recuperação de dados.RF016] deve estar descrito em uma subseção chamada “Recuperação de dados”, em um bloco identificado pelo número [RF016]. Já o requisito não-funcional [Confiabilidade.NF008] deve estar descrito na seção de requisitos não-funcionais de Confiabilidade, em um bloco identificado por [NF008].

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [NF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

1.2.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, nas seções 4 e 5, foram adotadas as denominações “essencial”, “importante” e “desejável”.

5. **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
6. **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
7. **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis

podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. Descrição geral do sistema

2.1 Abrangência e sistemas relacionados

O sistema *Vai ou Não Vai?* é uma ferramenta para o gerenciamento de eventos ocorrentes em uma universidade. Fornece uma maneira intuitiva e eficiente para a criação e visualização de eventos para a comunidade acadêmica.

Utilizando a ferramenta, o usuário - em geral, a comunidade acadêmica - poderá cadastrar novos eventos. Além disso, poderá alterar, remover e consultar eventos já criados.

A ferramenta permitirá que os usuários visualizem os eventos criados por outros usuários e que participem desses eventos e ainda receberem lembretes e notificações sobre mudanças em um evento.

Diante das várias ferramentas de gerenciamento de eventos existentes no mercado, o diferencial do *Vai ou Não Vai?* é uma linha do tempo de eventos por onde os usuários poderão visualizar os eventos mais próximos e por categorias.

3. Requisitos funcionais (casos de uso)

3.1 Cadastro

[RF001] Criar evento

Descrição do caso de uso: Este caso de uso permite que o usuário crie e armazene um novo evento no sistema.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: nome, data, local e descrição

Saídas e pós-condição: um evento é cadastrado no sistema

[RF002] Excluir evento

Descrição do caso de uso: Este caso de uso permite que o usuário exclua um evento do cadastro de eventos do sistema. Um evento pode ser excluído apenas pelo usuário que o criou.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: usuário tem que ter criado o evento

Saídas e pós-condição: o usuário consegue excluir o evento que deseja

[RF003] Alterar evento

Descrição do caso de uso: Este caso de uso permite que o usuário altere as informações de um evento.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: recebe como entrada o evento que se deseja alterar

Saídas e pós-condição: um evento é alterado no sistema

3.2 Interface

[RF001] Visualizar evento

Descrição do caso de uso: Este caso de uso permite que o usuário visualize os dados de um determinado evento.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: deve receber como entrada o evento que se deseja visualizar

Saídas e pós-condição: o usuário visualiza o evento desejado

[RF002] Participar de evento

Descrição do caso de uso: Este caso de uso permite que o usuário participe de um evento criado por outro usuário.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: recebe como entrada o evento que se deseja participar

Saídas e pós-condição: o usuário passa a ser um participante do evento

[RF003] Compartilhar evento

Descrição do caso de uso: Este caso de uso permite que o usuário compartilhe o evento em outras redes sociais.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: recebe como entrada o evento que se deseja compartilhar.

Saídas e pós-condição: o usuário compartilha o evento em outra rede social

[RF004] Convidar para evento

Descrição do caso de uso: Este caso de uso permite que o usuário convide outros usuários para um evento.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: recebe como entrada o evento que se deseja enviar o convite

Saídas e pós-condição: outro usuário recebe o convite enviado

[RF005] Visualizar presenças

Descrição do caso de uso: Este caso de uso permite que o usuário visualize quantas pessoas confirmaram presença em um evento.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: recebe como entrada o evento que se deseja visualizar as presenças

Saídas e pós-condição: o usuário consegue visualizar as presenças do evento desejado

3.3 Notificação

[RF001] Notificar participante

Descrição do caso de uso: Este caso de uso permite que o usuário receba notificação de um evento em caso de mudanças, inclusive a exclusão do mesmo.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: mudança em um evento que o usuário participa

Saídas e pós-condição: usuário é notificado sobre a mudança no evento

[RF002] Emitir lembrete

Descrição do caso de uso: Este caso de uso permite que o usuário receba um lembrete de um evento que participa algum tempo antes de o evento começar.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: deve receber como entrada a hora e data do evento que participa

Saídas e pós-condição: é emitido um lembrete para o usuário

[RF003] Confirmar presença

Descrição do caso de uso: Este caso de uso permite ao usuário confirmar presença em um evento.

Prioridade: Essencial Importante Desejável

Entradas e pré-condições: recebe como entrada a resposta do usuário confirmando que vai ao evento

Saídas e pós-condição: o evento recebe a contagem de presentes confirmados

4. Requisitos não-funcionais

[NF001] Usabilidade

A interface com o usuário é de vital importância para o sucesso do sistema. Principalmente por ser um sistema que será utilizado pela comunidade acadêmica e que geralmente não possui tempo disponível para aprender como utilizar o sistema.

O sistema terá uma interface amigável ao usuário primário sem se tornar cansativa aos usuários mais experientes. Em especial, a linha do tempo de eventos com opção de pesquisa por nome e filtro por categorias e datas.

Prioridade: Essencial Importante Desejável

[NF002] Desempenho

Embora não seja um requisito essencial ao sistema, deve ser considerada por corresponder a um fator de qualidade de software.

Prioridade: Essencial Importante Desejável

[NF003] Disponibilidade

Visando criar um produto com maior flexibilidade, o sistema deve ser desenvolvido como um aplicativo híbrido. Podendo ser executado em dispositivos móveis com sistema operacional Android e iOS.

O uso do framework Ionic permite que o aplicativo seja compilado para os dois sistemas operacionais.

Prioridade: Essencial Importante Desejável

5. Referências

ANGULARJS SITE OFICIAL. **AngularJS**. Disponível em: <<http://www.angularjs.org/>>. Acesso em: 10 jun. 2017.

BITNER, M. J. Service and technology: opportunities and paradoxes. MCB UP Ltd, v. 11, n. 6, p. 375-379, 2001.

CARDOSO, D. P; PINTO, M. M. As Redes Sociais Como Ferramentas de Divulgação de Eventos. FATEC, São Caetano do Sul, p. 30–40, 2011.

DATE C. J. **Introdução a Sistemas de Banco de Dados**. 8. ed. Americana. São Paulo: Elsevier, 2004.

DEGROSSI, L. C.; JUNIOR, V. F.; SILVA, J. D.; BARBOSA, E. F. Uma Comparação do Tempo de implementação: Android vs. HTML5. Workshop Latino Americano de Engenharia de Software Experimental, São Paulo, p. 63, 2013.

IONIC FRAMEWORK SITE OFICIAL. **Ionic Framework**. 2017. Disponível em: <<http://www.ionicframework.com/>>. Acesso em: 10 jun. 2017.

RAVULAVARU, A. **Learning Ionic - Build real-time and hybrid mobile applications with Ionic**. 2. ed. Birmingham: Packt Publishin, 2015.

TATROE, K.; MACINTYRE, P.; LERDORF, R. **Programming PHP**. 3. ed. [S.l.]: O'Reilly, 2013.

VANDECANDELAERE, B. **Developing the UDUBS-IT platform as a hybrid app with the ionic framework**. Western Cape: [s.n], 2015. Bacharelado em Novas Mídias e Tecnologia de Comunicação.