UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

BRUNO FERNANDES CAMPOS

Plataforma Móvel de Visualização de Locais de Entretenimento

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Leandro Krug Wives

Porto Alegre
2015

“If you think you can or think you can’t you are right.”
— HENRY FORD

RESUMO

O corrente trabalho pretende detalhar o desenvolvimento da aplicação para dipositivos móveis com o nome de Plataforma Mobile para a Visualização de Locais de Entretenimento. O objetivo principal do sistema é oferecer uma maneira fácil e simples de visualizar locais de entretenimento próximos ao usuário da aplicação. De maneira a oferecer este comportamento, a aplicação é integrada com redes sociais populares, como Instagram e Foursquare. Além disso, a aplicação também explora funcionalidades relacionadas à adição de conteúdo aos locais e visualização dos mesmos.

Palavras-chave: Mobilidade, iPhone, iOS, aplicação.

Mobile platform to discover places

ABSTRACT

The current work intends to detail the development of the mobile application named Mobile Platform to discover places. The main goal of the system is offer an easy way to visualize entertainment places around an user. In order to provide this behaviour, the application is integrated with popular social networks like Instagram and Foursquare. Furthermore, the application also explore features associated to the spot‘s content and its visualization.

Keywords: Mobility, iPhone, iOS, application.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
MVC	Model View Controller
IDC	International Data Corporation
UFRGS	Universidade Federal do Rio Grande do Sul
JSON	JavaScript Object Notation
IDE	Integrated Development Environment

SUMÁRIO

1 INTRODUÇÃO

Segundo o SEBRAE, com expansão anual em torno de 10%, o setor de alimentação fora de casa – ou de bares e restaurantes, como é chamado pelos comerciantes do ramo – gera cerca de 450 mil novas oportunidades de emprego por ano (SEBRAE, 2015).

Ainda, segundo o SEBRAE, até 2017 o mercado de entretenimento deve chegar aos 71 bilhões de dólares no Brasil. Essa previsão somada ao momento de visibilidade internacional alcançado pelo país por sediar eventos de grande porte, como a Copa do Mundo da FIFA de 2014, trazem boas perspectivas para os negócios.

Além do crescimento do mercado de entretenimento, é importante destacar o crescimento do mercado móvel, conforme dados do (IDC, 2014), de janeiro a dezembro de 2014, foram 54.5 milhões de aparelhos inteligentes comercializados, i.e., crescimento de 55% em comparação com 2013. Para 2015, apesar do cenário econômico desfavorável, a projeção é de alta de 16%.

É nesse contexto e tendo em vista a fornecer uma plataforma que possibilite a visualização remota de dados sobre locais de entretenimento que este trabalho se encaixa. O objetivo principal do trabalho é o desenvolvimento de um aplicativo com o intuito de facilitar o acesso a informações de locais de entretenimento em tempo real.

O documento está estruturado da seguinte forma. O capítulo seguinte apresenta a fundamentação teórica, isto é, os conceitos que estão por trás do sistema desenvolvido. Nele, também será abordado um breve histórico sobre as plataformas móveis e aplicativos semelhantes ao atual, os quais foram usados como fonte de inspiracão para o desenvolvimento do trabalho. No capítulo 3 serão abordadas as tecnologias utilizadas no desenvolvimento do sistema. No capítulo 4 tópicos relacionados ao projeto, arquitetura e modelagem dos sistema serão abordados. O capítulo 5 expõe aspectos relacionados ao desenvolvimento da aplicação, como metodologia, detalhes da conexão com as APIs e a estrutura da aplicação. Já no capítulo 6 será apresentado um guia da aplicação, cada tela do sistema será apresentada e detalhada nas seções deste capítulo. No capítulo 7 serão abordados tópicos relacionados à experiência do usuário no aplicativo e também detalhado um questionário de avaliação proposto à usuários do aplicativo. Finalmente, no capítulo final, serão descritas as conclusões, incluindo os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados alguns conceitos que foram usados como base para o desenvolvimento deste trabalho. Primeiramente será explicado o conceito de plataformas móveis, e logo após serão citados alguns aplicativos que possuem similaridades com o corrente trabalho.

2.1 Plataformas Móveis

De acordo com Fling (FLING, 2009), considerando o acesso a web via computadores desktop e via dispositivos móveis, é possível afirmar que o acesso à Web via dispositivos móveis tornou-se superior nos últimos anos, e que o número de acessos deve crescer ainda mais nos próximos anos. Além disso, como demonstrado na Figura 2.1, o número de smartphones já ultrapassa o número de computadores pessoais desde meados de 2012.

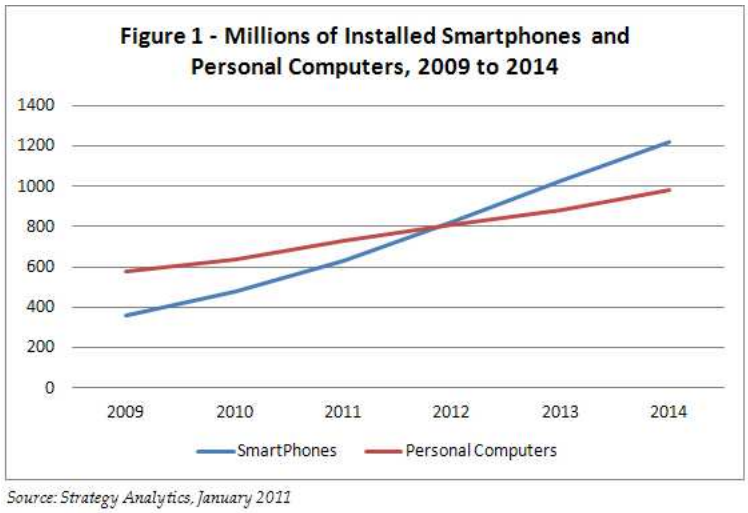

Figura 2.1: Gráfico do crescimento mobile
Fonte: (WEST M., 2011)

De acordo com (WEST M., 2011), os dispositivos móveis estão reformulando a sociedade, os meios de comunicação e a economia mundial. Com a ultrapassagem do número de smartphones ao número de computadores pessoais, a maneira com que as pessoas

acessam e compartilham informações tem mudado. Atualmente, os dispositivos móveis são utilizados para realizar pagamentos, acessar dados financeiros e realizar pesquisas a qualquer momento. Essa revolução no acesso da informação representa um ponto fundamental de mudança na história da humanidade. Hoje as pessoas não mais precisam ficar horas presas no trânsito, existe um aplicativo que pode ajudar qualquer um a traçar a melhor rota. Para contatar a polícia não é mais necessário realizar uma ligação, é possível contatá-la via mensagem de texto. Os exemplos acima são apenas algumas situações que mostram como a tecnologia móvel está revolucionando o mundo e a forma como os seres humanos se relacionam.

2.2 Trabalhos e aplicativos relacionados

Nesta seção serão apresentados aplicativos que tem um objetivo semelhante ao do corrente trabalho.

2.2.1 Instagram

O Instagram[1] é um aplicativo popular que possibilita aos usuários o compartilhamento de fotos e vídeos. O aplicativo é integrado com diversas redes sociais, como Foursquare, Facebook e Twitter. A interface do aplicativo é bastante simples e intuitiva. A Figura 2.2 mostra o painel de mídias relacionadas a um usuário, uma das funcionalidades do aplicativo.

O Instagram, como será introduzido no item 4.2.1, provê dados ao corrente sistema. Além disso, no início do aplicativo, para o usuário possuir acesso à aplicação, ele é obrigado a acessar a sua conta da plataforma do Instagram.

[1] https://itunes.apple.com/br/app/instagram/id389801252?mt=8

Figura 2.2: Tela de fotos de seus amigos no aplicativo instagram

2.2.2 AroundMe

AroundMe[2] é um aplicativo que disponibiliza informações sobre locais próximos a usuários. Apesar de disponibilizar funcionalidades semelhantes às do corrente aplicativo, este aplicativo não disponibiliza informações em tempo real sobre os locais. A Figura 2.3 mostra a tela do aplicativo relacionada ao filtro de locais disponibilizados pelo aplicativo.

[2]https://itunes.apple.com/br/app/aroundme/id290051590?mt=8

Figura 2.3: Tela de filtro de Locais no aplicativo AroundMe

2.2.3 Lucky

Lucky[3] é o aplicativo que mais se assemelha ao objetivo central do corrente trabalho, tem como objetivo principal fornecer uma plataforma para que usuários troquem informações sobre locais de entretenimento em tempo real. Possui as funcionalidades de avalição de local e visualização de locais em um mapa. Apesar de a ideia ser inovadora, o aplicativo tem uma interface pobre e o número de usuários é reduzido. A Figura 2.4 mostra a tela do aplicativo relacionada a um local de entretenimento e suas respectivas avaliações realizadas por usuários.

[3]https://itunes.apple.com/br/app/lucky/id919568081?mt=8

Figura 2.4: Tela de informações de um local do aplicativo Lucky

3 FERRAMENTAS E FRAMEWORKS

Este capítulo apresentará tópicos relacionados às ferramentas e aos frameworks utilizados durante o desenvolvimento do sistema. Serão introduzidos as ferramentas e os frameworks disponíveis, bem como suas vantagens e desvantagens. Dentre as opções de tecnologias existentes, a opção escolhida e a justificativa de sua escolha serão apresentadas ao fim de cada seção.

3.1 Ambiente de desenvolvimento

O objetivo desta seção é detalhar os ambientes de desenvolvimento disponíveis para a plataforma iOS, e posteriormente justificar a escolha do ambiente de desenvolvimento escolhido.

3.1.1 Xamarin

A IDE Xamarin permite o desenvolvimento de aplicativos para as plataformas iOS, Android e Windows Phone. A peculiaridade do Xamarin é a capacidade de permitir o desenvolvimento de um aplicativo capaz de suportar diversos sistemas operacionais móveis, isto é, o código utilizado para uma das plataformas suportadas pode ter seu uso compartilhado com outras plataformas possibilitando um grande reúso de código.

Com um código nativo baseado em C#, o Xamarin permite o desenvolvimento de aplicações sem necessitar o uso das linguagens nativas usadas para desenvolver um aplicativo para a plataforma iOS, Android e Windows Phone.

O Xamarin é muito utilizado hoje em dia como forma de acelerar o desenvolvimento de aplicativos nativos, isto é, seu uso possibilita o desenvolvimento de apenas uma aplicação para três plataformas distintas.

Apesar da IDE Xamarin possuir diversos aspectos positivos, ela peca em outros aspectos, quando se fala de estabilidade, robustez e confiabilidade, ela deixa a desejar. Cada plataforma móvel possui suas peculiaridades, os componentes da plataforma iOS e Android e a experiência de um usuário em ambas plataformas possuem diferenças. Existem diversos aplicativos no mercado hoje que possuem interfaces completamente distintas

para cada uma das plataformas móveis mais famosas, como iOS, Android e Windows Phone. É muito difícil satisfazer todas as peculiaridades das plataformas móveis em apenas uma aplicação, o ideal é que o Xamarin seja usado apenas para protótipos ou produtos que não estejam maduros ainda como forma de validação de um produto com usuários.

3.1.2 Xcode

O Xcode é o ambiente de desenvolvimento da Apple. Pode ser utilizado tanto para desenvolvimento de aplicativos para a plataforma Mac OS X quanto para desenvolvimento de aplicativos para a plataforma iOS. O Xcode suporta diversas linguagens de programação, dentre elas as linguagens Objective-C e a linguagem Swift. A linguagem Swift foi lançada recentemente em junho de 2014 e aberta aos desenvolvedores em setembro de 2014.

Por possuir extensa documentação e suporte na web, o corrente trabalho foi desenvolvido usando a linguagem Objective-C. A maioria dos aplicativos disponíveis na loja de aplicativos da Apple, a App store, utilizam o Xcode, já que ele permite o desenvolvimento nativo de aplicações para a plataforma móvel iPhone. É difícil encontrarmos um aplicativo já consolidado no mercado que não utilize o Xcode como ambiente de desenvolvimento e as linguagens nativas como Objective-C e Swift.[1]

3.1.2.1 Objective-C

Objective-C é uma linguagem orientada a objetos que utiliza o framework chamado *Cocoa*. O Cocoa permite que desenvolvedores criem aplicações para o Mac OS X, e uma versão dele permite a criação de aplicações para o iPhone. O Cocoa nada mais é que um conjunto de janelas e interfaces de usuário, além de ferramentas, que , junto com o Objective-C permitem o desenvolvimento de aplicações para as plataformas da Apple.

3.1.3 Resumo

Nesta seção foram apresentadas as IDEs disponíveis para o desenvolvimento de um aplicativo na plataforma iOS. Como já citado nas subseções anteriores, a utilização da IDE Xcode permite o desenvolvimento nativo de um aplicativo para a plataforma iOS, ela também permite que a experiência de usuário seja customizada e otimizada para esta plataforma. Além disso, sua utilização obriga que as linguagens nativas Objective-C ou Swift sejam usadas, o que acarreta a maior robustez do código e a consequente estabilidade do aplicativo. Por estas razões o Xcode foi a IDE escolhida para realizar o desenvolvimento do corrente trabalho.

[1] Esta subseção foi baseada em (GOLDSTEIN, 2009)

3.2 Gerenciamento de dependências

Ao desenvolver uma aplicação mobile, o desenvolvedor tem a possibilidade de usar diferentes bibliotecas e frameworks para integrar diferentes funcionalidades em sua aplicação. O objetivo desta seção é apresentar as opções disponíveis para gerenciamento de dependências na IDE Xcode e justificar a escolha da solução escolhida.

3.2.1 Xcode

A IDE Xcode oferece uma forma nativa de incluir bibliotecas e frameworks, isto é, suas dependências. A inclusão das dependências pode ser feita manualmente na tela de opções de projeto.

Usar a solução nativa oferecida pela IDE para integrar dependências é positivo pois dificilmente problemas poderão ocorrer no momento de incluir bibliotecas e frameworks, no entanto, esta solução não permite que o desenvolvedor possa controlar o versionamento das versões das dependências incluídas.

3.2.2 CocoaPods

CocoaPods[2] é um gerenciador de dependências de projetos para a plataforma iOS. Basicamente ele permite que a inclusão de bibliotecas e frameworks seja feita elegantemente pelo sistema. Além disso ele permite que o controle das versões das bibliotecas e frameworks utilizados seja feito com simplicidade. A figura 3.1 mostra o uso do CocoaPods, a partir de um arquivo, em que é especificado as bibliotecas usadas e as respectivas versões (se não é especificada a versão, a versão mais recente do framework é selecionada por padrão). As dependências são incluídas no projeto após a execução de alguns comandos no terminal.

O CocoaPods, por possuir um arquivo texto que especifica como as dependências e qual versão das mesmas deve ser incluída, permite que o desenvolvimento simultâneo de um projeto seja realizado com simplicidade, já que ao compartilhar o arquivo de texto usado para gerar as dependências, os desenvolvedores envolvidos estariam recebendo a "documentação"de como as dependências devem ser adicionadas ao projeto.

[2]https://cocoapods.org/

Figura 3.1: Arquivo podfile que especifica as bibliotecas e frameworks que devem ser adicionados ao projeto

3.2.3 Resumo

Nesta seção foram apresentadas as opções existentes para realizar o controle de dependências na IDE Xcode. O framework *CocoaPods* foi adotado no desenvolvimento do corrente trabalho, visto que, como já abordado na seção 3.2.2 , ele permite que o controle de versões e de dependências seja feito facilmente e que qualquer desenvolvedor com acesso ao arquivo podfile, exemplificado pela figura 3.1, possa incluir as dependências facilmente.

3.3 Prototipação da Interface

Durante a fase de projeto de um aplicativo, a fase de levantamento de requisitos é essencial. Para realizar o levantamento de requisitos, o corrente trabalho utilizou, junto com as *user stories*, uma ferramenta de prototipação de interfaces. O objetivo desta seção é introduzir algumas das diversas ferramentas de prototipação de interfaces existentes e justificar a escolha da solução adotada.

3.3.1 Mockinbird

O Mockinbird é uma ferramenta que permite a criacão de mockups online. Esta ferramenta possui diversos modelos de componentes para serem usados na montagem de uma interface, porém todos os componentes disponíveis por esta ferramenta tem seu layout baseado em aplicações web.

3.3.2 Justinmind

O Justinmind é uma ferramenta que permite a prototipação e criação de mockups para aplicações web e mobile. Como introduzido no início desta seção, esta ferramenta foi utilizada no presente trabalho em conjunto com outras formas, como as users stories, que serão introduzidas na sessão 4.4 , para realizar o levantamento de requisitos necessários da aplicação. O diferencial desta ferramenta é a possibilidade de uso dos componentes nativos das plataformas móveis. Por este motivo esta foi a ferramenta escolhida para o desenvolvimento dos mockups do presente aplicativo. Abaixo seguem os projetos de interface realizados com a ferramenta Justinmind.

Figura 3.2: Mockup da tela de login do aplicativo

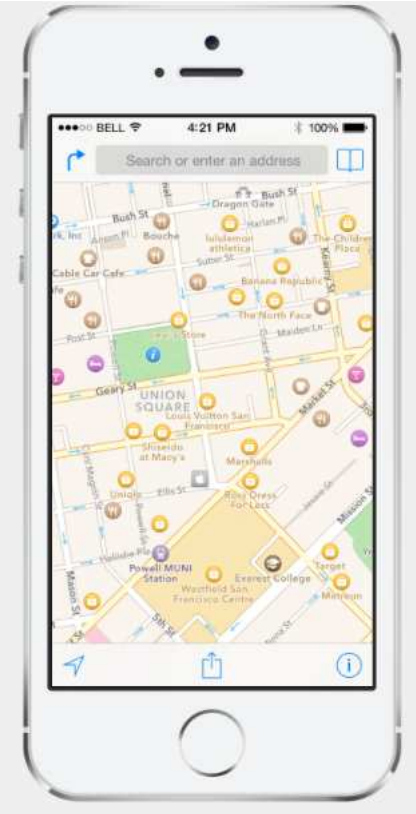

Figura 3.3: Mockup da tela de mapa do aplicativo

Figura 3.4: Mockup da tela de acesso à camera

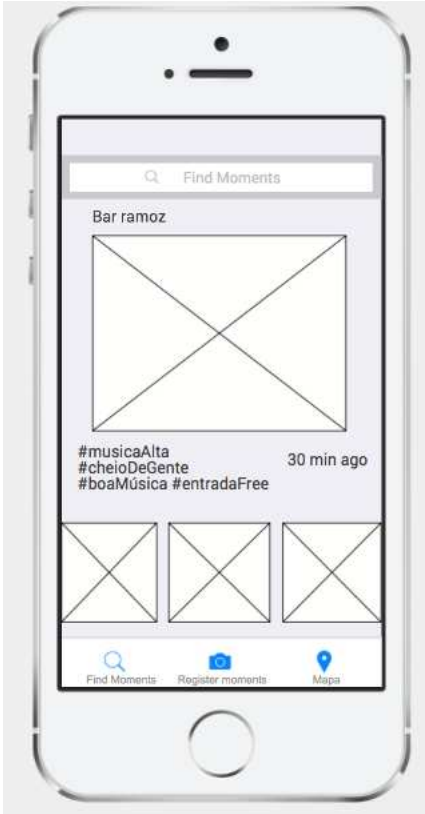

Figura 3.5: Mockup da tela da foto tirada

Figura 3.6: Mockup da tela de informações sobre um local

3.3.3 Resumo

Esta seção detalhou algumas das ferramentas disponíveis para realizar a prototipação de interfaces. A ferramenta escolhida para o corrente trabalho foi a ferramenta chamada JustInMind. A maior motivação para a escolha desta ferramenta foi a disponibilidade oferecida por ela em relação aos componentes relacionados à interface mobile.

4 MODELAGEM E PROJETO DA APLICAÇÃO

Neste capítulo tópicos relacionados à modelagem e ao projeto do sistema serão abordados. Na seção 4.1 será introduzida a arquitetura geral da aplicação, na seção 4.2 será introduzido o conceito de API, e serão introduzidas as APIs utilizadas, ainda, na seção 4.3 será introduzido o uso da plataforma Parse no corrente projeto. Por fim, na seção 4.4 a modelagem do sistema será detalhada.

4.1 Arquitetura da plataforma Móvel de visualização de locais de entretenimento

A arquitetura do sistema móvel de visualização de locais de entretenimento consiste da conexão do dispositivo móvel com as APIs e de sua conexão com a plataforma Parse. A arquitetura geral é definida abaixo pela figura 4.1. A maior parte dos dados da aplicação são alimentados pelas APIs do Foursquare e do Instagram. Como exibido pela figura 4.1, as APIs são utilizadas estritamente como fonte de dados, o Parse por sua vez, como introduzido pela seção 4.3 é utilizado como o banco de dados da aplicação, realizando o armazenamento das mídias relacionadas aos locais e também realizando o armazenamento de dados relacionados aos usuários.

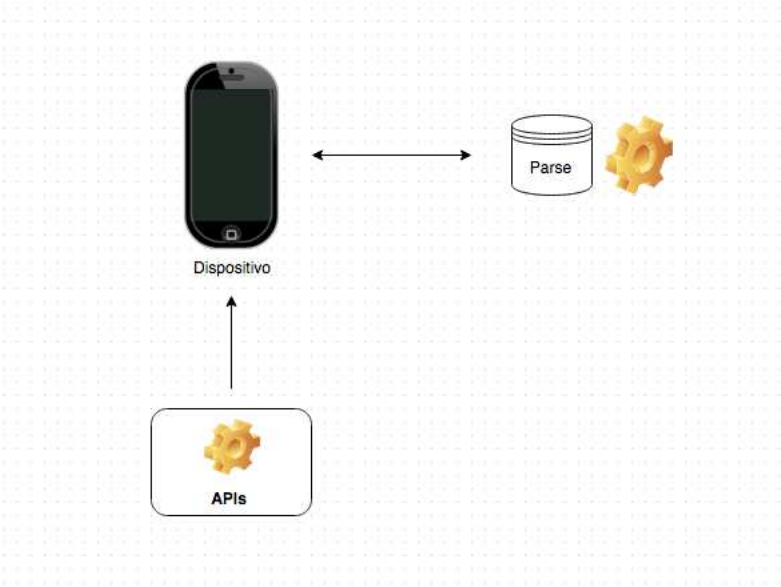

Figura 4.1: Visão geral do sistema

4.2 APIs

Uma das funcionalidades únicas do corrente trabalho em relação à aplicativos semelhantes é a sua conexão com diferentes APIs. Vale ressaltar também a importância do uso das APIs no corrente trabalho como fonte de dados para a maioria das funcionalidades. Por serem APIs de sistemas já maduros no mercado mobile, elas tornam-se uma fonte de dados robusta e completa. Para alimentar os dados do aplicativo e suprir os dados de funcionalidades básicas do aplicativo, duas APIs foram utilizadas, a API do Instagram, que será introduzida na sessão 4.2.1 e a API do Foursquare, que será introduzida na sessão 4.2.2.

4.2.1 Instagram

O Instagram é uma rede social de compartilhamento de fotos e vídeos online, a qual permite o compartilhamento destes por seus usuários. Possui integração com diversas outras redes sociais, como Foursquare, Tumblr, Flickr e Facebook. Sua API é usada no corrente trabalho como fonte de dados para as mídias relacionadas aos locais.

4.2.2 Foursquare

O Foursquare é uma rede geossocial e um pequeno blog, ele permite que o utilizador compartilhe a sua localização geográfica e que o utilizador possa encontrar lugares e pessoas próximas de sua localização. Além disso, ele também apresenta funcionalidades relacionadas à avaliação de locais, como comentários e opiniões. Sua API é utilizada no presente trabalho como fonte de dados para a base de dados de lugares.

4.3 Parse

O Parse é uma ferramenta que disponibiliza diversos recursos para aplicações, os quais incluem serviço de banco de dados, serviço de notificações via push, serviço de registro de dados da aplicação, serviço de hospedagem, entre outros. Para o corrente trabalho, apenas os serviços de análise de dados e de banco de dados foram usados.

4.3.1 Parse e a sua integração

A integração da ferramenta Parse nas plataformas móveis é feita via SDK (*Software Development Kit*). A partir desta integração, a aplicação é capaz de utilizar os recursos disponíveis pelo Parse, como execução de queries no banco de dados, e registro de dados sobre o sistema na ferramenta.

4.3.2 Parse e sua disponibilização de métricas

A plataforma Parse disponibiliza diversas informações sobre a aplicação, são disponibilizados dados como número de usuários, usuários ativos e inativos, número de downloads e também dados sobre o comportamento de usuários no sistema. Essas informações normalmente são utilizadas como métricas para medir o desempenho do aplicativo e de funcionalidades do mesmo.

4.3.3 Parse e seu armazenamento de dados

Como introduzido nesta seção, o Parse disponibiliza um serviço de banco de dados online, o qual é baseado na solução NoSQL. Conforme (VAISH, 2013), o NoSQL, conhecido como *not only sql*, surgiu no começo dos anos 2000, como tentativa de criar uma solução escalável para aplicações. O NoSQL pode ser definido como qualquer banco de dados que não segue o tradicional modelo relacional de banco de dados, sendo assim os dados usados não são relacionais e a linguagem de consulta ao banco não é o SQL.

4.4 User Stories

Normalmente, ao projetar um sistema, é necessário a definição dos requisitos e das funcionalidades. A representação dos requisitos e das funcionalidades do corrente trabalho foram definidas a partir de *User stories*. *User stories* consistem de pequenas narrativas a partir de um usuário que descrevem uma funcionalidade do sistema e a consequente interação entre usuário e sistema decorrente desta funcionalidade (COHN, 2004). As *user stories* são definidas a partir do seguinte formato:

Como um <papel de usuário> eu quero <objetivo/razão>.

Na tabela abaixo serão apresentados os requisitos do aplicativo a partir do uso das histórias de usuário.

Nº	Como um	Eu quero
1	Usuário padrão	Poder fazer login no sistema
2	Usuário padrão	Que se o login no sistema falhar, tal falha seja sinalizada
3	Usuário padrão	Que uma vez logado no sistema, essa informação seja persistida para que em futuros acessos o login não seja necessário
4	Usuário padrão	Que o mapa exiba os locais de lazer próximo a mim
5	Usuário padrão	Que ao clicar em um local exibido no mapa, informações sobre o local e sobre o número de pessoas no local sejam mostradas
6	Usuário padrão	Que seja possível tirar uma foto e compartilhá-la no aplictivo
7	Usuário padrão	Que seja possível, após tirar a foto, selecionar o local onde a foto foi tirada
8	Usuário padrão	Que sejam listados os locais que possuem mais pessoas no momento
9	Usuário padrão	Que sejam exibidas as últimas fotos tiradas em um local selecionado

Tabela 4.1: *User stories* do sistema

5 DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo serão apresentados tópicos relacionados ao desenvolvimento do sistema. Na seção 5.1 será abordada a metodologia utilizada no desenvolvimento, bem como a ferramenta utilizada para o gerenciamento do projeto. Na seção 5.2 será abordada em detalhe a conexão com as APIs do Foursquare e Instagram. Já na seção 5.3 serão abordados tópicos relacionados à arquitetura e implementação do sistema.

5.1 Metodologia

Por se tratar de um sistema mobile, por possuir funcionalidades dinâmicas e pelos requisitos e funcionalidades não terem sido todos definidos detalhadamente no início do projeto, a metodologia seguida no corrente trabalho foi a metodologia ágil. Já que o desenvolvimento do corrente trabalho não foi feito em equipe e não havia mais de uma pessoa envolvida no projeto, a metodologia ágil que o trabalho seguiu se baseou em uma versão adaptada do SCRUM. As tarefas e funcionalidades eram definidas em partes visando a entrega de um sistema funcional ao final de cada iteração, a qual se caracterizava pelo início de uma nova reunião com o professor orientador Leandro Wyves, em que as funcionalidades desenvolvidas eram validades e novas funcionalidades eram definidas.

A ferramenta usada para a gerência do trabalho é chadama de *Trello* e será introduzida na subseção 5.1.1[1].

5.1.1 Trello

O Trello é uma aplicação web utilizada para o gerenciamento de projetos que utiliza o paradigma Kanban. Conforme (KNIBERG; SKARIN, 2009), o Kanban consiste em dividir um objetivo em items e fixá-los em uma parede, na qual é possível posicionar itens em colunas, as quais podem possuir um limite de itens a ser inserido.

[1]Esta seção foi baseada em http://www.mountaingoatsoftware.com/agile/scrum/overview

5.1.1.1 Trello e sua organização

A organização de projetos na ferramenta Trello é caracterizada por um quadro, em que cada quadro contém colunas com pequenos cards. Para o corrente trabalho, foram usadas três colunas, a primeira representa os objetivos, a segunda coluna representa o que está em progresso e a terceira coluna representa o que foi finalizado.

5.1.1.2 Utilização do Trello no projeto

A figura 5.1 exibe como foi feita a organização do projeto. Cada grupo de cards exibidos na figura 5.1 foram definidos em reuniões periódicas com o professor orientador Leandro Wyves e inseridos na lista caracterizada por *To Do*, a qual é caracterizada por tarefas que ainda não começaram a ser desenvolvidas. A coluna *Doing* é caracterizada por tarefas que estão sendo desenvolvidas e a coluna *Done* é caracterizada por tarefas que foram desenvolvidas.

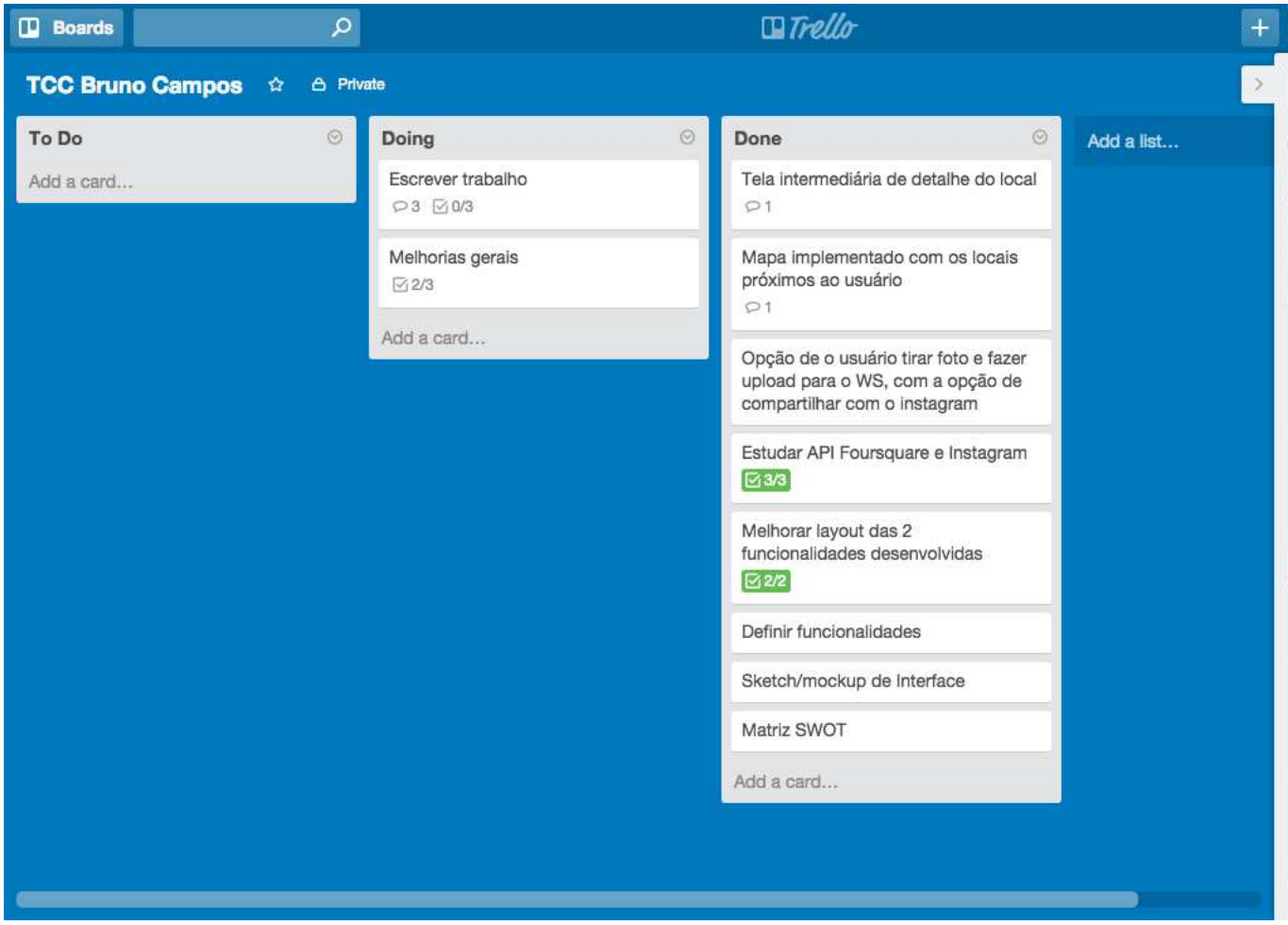

Figura 5.1: Trello e sua organização do corrente trabalho

5.2 Conexão com as APIs

Nesta seção será abordado como o sistema faz a conexão com as APIs e quais recursos o mesmo consome.

5.2.1 Conexão com a API do Foursquare

Como introduzido na subseção 4.2.2, uma das APIs utilizada como fonte de dados é a API do Foursquare. O Foursquare disponibiliza diversos recursos, os recursos utilizados por este aplicativo se limitam a recursos relacionados à busca de locais. Abaixo serão listados os recursos utilizados

5.2.1.1 ***Procura por locais***

Recurso: `https://api.foursquare.com/v2/venues/search`[2]

Descrição: Este recurso é utilizado como fonte de dados na tela de mapa de locais e também na tela de procura por local.

Parâmetros: Este recurso aceita diversos parâmetros relacionados a um local, mas os parâmetros passados pelo sistema são latitude e longitude do usuário, dados de autenticação da aplicação, cidade do usuário, versão da API do foursquare e categoria dos locais a serem retornados. Este último parâmetro informa à API que o resultado da consulta deve ser filtrado de acordo com as categorias passadas, no caso do corrente trabalho, as categorias consultadas são bares, clubes e restaurantes.

Este recurso é utilizado em 2 funcionalidades da aplicação, ao listar os locais no mapa e ao procurar por um local por nome.

Retorno: Este recurso retorna um objeto no formato JSON contendo diversos dados sobre o local. Os dados consumidos pelo aplicativo são o identificador único do local, o nome do local, e o número de check-ins realizados no local.

5.2.2 Conexão com a API do Instagram

Como introduzido na subseção 4.2.1 a outra API utilizada como fonte de dados é a API do Instagram. O Instagram disponibiliza múltiplos recursos, contudo este aplicativo se limitou a consumir os dados relacionados aos locais e às mídias referentes a estes locais. Além disso, o corrente sistema utiliza a API do instagram para realizar o login do usuário.

Abaixo serão listados os recursos utilizados.

5.2.2.1 ***Login de um usuário***

Recurso: `https://api.instagram.com/oauth/authorize`[3]

[2] https://developer.foursquare.com/docs/venues/search
[3] https://instagram.com/developer/authentication/

Descrição: Este recurso é utilizado para realizar o login do usuário. O recurso é utilizado posteriormente para autenticar as chamadas do usuário à API. O Instagram estabelece um limite de chamadas à API por token de usuário, obrigá-lo a se logar é uma maneira de não ultrapassar esse limite e de não deixar com que os serviços providos pelo sistema fiquem indisponíveis.

Parâmetros: Para a utilização deste recurso é necessário passar parâmetros relacionados à autenticação da aplicação, os quais são utilizados como identificadores de uma aplicação pela API do Instagram.

Retorno: Este recurso retorna um valor condicional que indica se o usuário obteve sucesso ou falha na autenticação de seu login e informações do token de acesso do usuário à API do Instagram.

*5.2.2.2 **Busca por identificação única do local***

Recurso: `https://api.instagram.com/v1/locations/search`[4]

Descrição: Este recurso é utilizado para fazer a conexão entre as APIs do Instagram e do Foursquare .

Parâmetros: Para a utilização deste recurso é necessário passar o token de acesso do usuário. O outro parâmetro passado é o identificador de um local na plataforma do Foursquare.

Retorno: Este recurso retorna o identificador correspondente utilizado pelo Instagram, que será usado para a consulta de mídias do local no próximo item.

*5.2.2.3 **Busca por Mídias de um local***

Recurso: `https://api.instagram.com/v1/locations/id/media/recent`[5]

Descrição: Este recurso é utilizado para consumir as mídias associadas a um local .

Parâmetros: Para a utilização deste recurso é necessário passar o token de acesso do usuário e o identificador único do local.

Retorno: Este recurso retorna um objeto no formato JSON que lista as mídias relacionadas ao local e suas respectivas informações.

5.3 Estrutura do Aplicativo

O corrente trabalho baseou-se no padrão arquitetural MVC (Model-View-Controller), segundo Sommerville(SOMMERVILLE, 2004), o MVC é um padrão arquitetural que separa a apresentação de dados e a interação dos dados do sistema. O MVC é composto de três tipos de objetos, o model é o objeto da aplicação, a view é a apresentação da tela, e o controller define a maneira como a interface reage à entrada do usuário.

[4]https://instagram.com/developer/endpoints/locations/
[5]https://instagram.com/developer/endpoints/locations/

Na imagem 5.2 a interação entre os componentes é representada em um diagrama, fica claro que os objetos view e model não interagem entre si diretamente, quem realiza tal interação é o objeto controller.

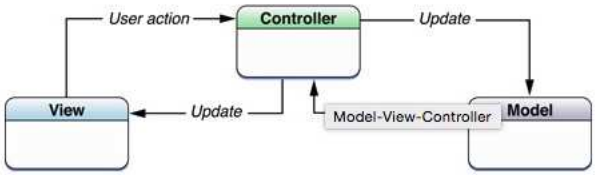

Figura 5.2: Interação entre os componentes do modelo arquitetural MVC

5.3.1 MVC aplicado ao Sistema

A figura 5.3 exibe a estrutura do sistema na ferramena Xcode . De acordo com a figura, a pasta *Models* representa o objeto model da arquitetura MVC, a pasta *ViewControllers* representa o objeto controller da arquitetura MVC, e o arquivo Main.storyboard representa o objeto view da arquitetura MVC. Abaixo serão comentadas as estruturas mais importantes utilizadas no projeto.

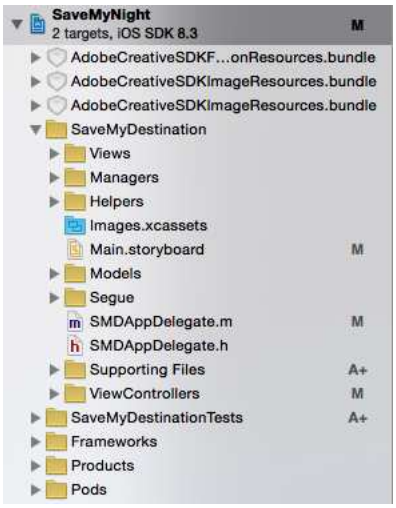

Figura 5.3: Estrutura do projeto

5.3.1.1 Helpers

A pasta Helpers, demonstrada em detalhe na figura 5.4, contém classes que foram criadas para ajudar no desenvolvimento da aplicação. Ela contém a classe SMDImageHelper, que possui funções relacionadas ao tratamento de imagens.

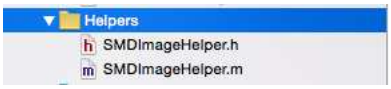

Figura 5.4: Detalhe da pasta Helper

5.3.1.2 Main.storyboard

Esta classe, a qual foi introduzida no início desta seção representa o objeto *View* na arquitetura MVC. Como representado pela figura 5.5 esta classe contém todas as telas utilizadas pelo projeto. É possível, a partir da figura 5.5 acompanhar a relação hierárquica das telas, caracterizada pelas setas.

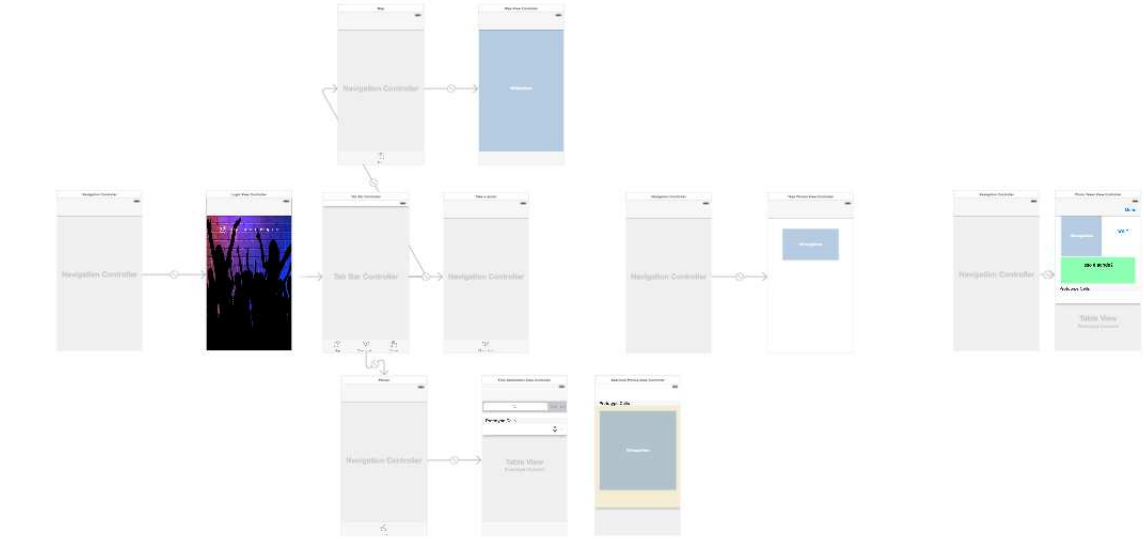

Figura 5.5: Detalhe da classe Main.storyboard

5.3.1.3 Models

A pasta Models contém classes que representam a estrutura *Model* definida pela arquitetura MVC. De acordo com a figura 5.6 ,serão citadas abaixo algumas das subpastas contidas nesta pasta.

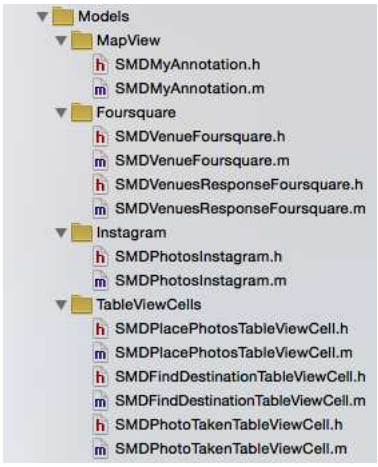

Figura 5.6: Detalhe da pasta Models

- Foursquare: Esta pasta contém classes relacionadas aos recursos que são consumidos da API do Foursquare. As classes são usadas para representar os objetos retornados pelos recursos da API.

- Instagram: Esta pasta contém classes relacionadas aos recursos que são consumidos da API do Instagram. Estas classes são usadas para represntar os objetos retornados pelos recursos da API.

- MapView: Esta pasta contém as classes que representam os pinos usados na tela de mapa.

- TableViewCells: Esta pasta contém classes que representam as células de um componente do framework *Cocoa* chamado TableView. A TableView é usada para representar dados em uma lista.

5.3.1.4 ViewControllers

A pasta ViewControllers representa o objeto Controller na arquitetura MVC, no corrente projeto as classes fazem o gerenciamento da interface e da interação dos modelos com a mesma.

5.3.2 Banco de Dados

Apesar da maioria dos dados da aplicação proverem das APIs do Foursquare e Instagram, o presente trabalho possui também o seu próprio banco de dados. Como introduzido no item 4.3 a aplicação utiliza o serviço de banco de dados disponibilizado pela plataforma Parse.

Neste item serão descritas as entidades utilizadas para modelar o banco de dados da aplicação. A aplicação não exigiu uma modelagem complexa de seu banco de dados, já que, como introduzido no início desta seção, a maioria dos dados provêm das plataformas

sociais. A figura 5.3.2 representa o diagrama de entidade relacionamento do banco de dados da aplicação.

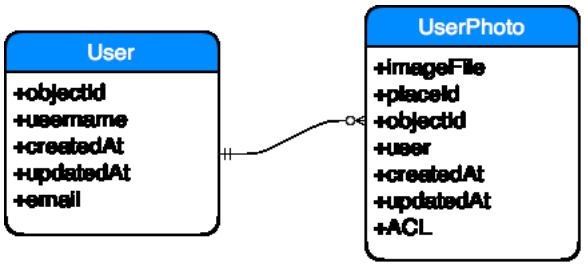

Figura 5.7: Diagrama entidade relacionamento do banco de dados do sistema

5.3.2.1 Entidade User

A entidade *User* representa um usuário do sistema. Dados do usuário como *objectId*, *createdAt*, *updatedAt* são dados automaticamente preenchidos pela plataforma Parse. O atributo *objectId* será usado posteriormente como chave primária e estrangeira para se relacionar com a entidade *UserPhoto*. Este mesmo atributo será representado pelo atributo *user* na entidade *UserPhoto*. Os atributos *email* e *username* são campos preenchidos manualmente pela aplicação para identificar o usuário que está conectado com o sistema.

5.3.2.2 Entidade UserPhoto

Para guardar dados relacionados às mídias fornecidas por usuários e dados sobre os locais relacionados às mesmas, o corrente trabalho faz uso da entidade **UserPhoto**. Esta entidade, que é representada pela tabela no banco de dados **UserPhoto**, é exibida na figura 5.3.2.2. Atributos como *objectId*, *user*, *createdAt*, *updatedAt* e *ACL* são campos criados e preenchidos automaticamente pela plataforma Parse. Já os campos *placeId* e *imageFile* são campos criados manualmente pela aplicação. A coluna *imageFile* corresponde à mídia guardada, já a coluna *placeId* corresponde ao identificador do local usado pela API do Foursquare.

objectId String	imageFile File	placeId String	user Pointer<_User>	createdAt Date	updatedAt Date	ACL ACL
objectId						
[illegible]seFpj	Image.jpg	4e286054814df9e4be5459ba	mNYQ76j3KR	May 03, 2015, 21:43	May 03, 2015, 21:43	mNYQ76j3KR
xk9Tuuy5k2	Image.jpg	4e286054814df9e4be5459ba	mNYQ76j3KR	May 03, 2015, 21:38	May 03, 2015, 21:38	mNYQ76j3KR
Ir1SpHqm12	Image.jpg	4e286054814df9e4be5459ba	mNYQ76j3KR	May 03, 2015, 21:36	May 03, 2015, 21:36	mNYQ76j3KR
OvWy3FfdEm	Image.jpg	4e286054814df9e4be5459ba	mNYQ76j3KR	May 03, 2015, 20:55	May 03, 2015, 20:55	mNYQ76j3KR
R2OCegR4R8	Image.jpg	4e286054814df9e4be5459ba	mNYQ76j3KR	May 03, 2015, 20:47	May 03, 2015, 20:47	mNYQ76j3KR
4M9vehlIiQ	Image.jpg	4e286054814df9e4be5459ba	mNYQ76j3KR	May 03, 2015, 20:31	May 03, 2015, 20:31	mNYQ76j3KR

Figura 5.8: Detalhe da tabela UserPhoto usada no projeto

6 FUNCIONAMENTO DO APLICATIVO

O objetivo deste capítulo é apresentar todas as telas do aplicativo. Serão detalhadas todas as telas existentes no aplicativo e também descritas as suas respectivas funcionalidades.

6.1 Tela de boas-vindas

A tela de boas-vindas é a primeira tela que é exibida ao usuário. Obrigatoriamente o usuário deve se logar com uma conta da plataforma Instagram, como consta na figura 6.1.

Figura 6.1: Tela de boas vindas ao usuário com a opção de logar

6.2 Tela de login

A tela de login é a tela originada da ação no botão de login detalhada na seção 6.1. O usuário deve obrigatoriamente se logar com uma conta da plataforma do Instagram, como consta na figura 6.2. Após o login, é retornado um token para o aplicativo, que será usado para a autenticação ao acessar os recursos da API do Instagram.

Figura 6.2: Detalhe da tela de login

6.3 Tela de mapa

A tela de mapa lista bares e restaurantes próximos ao usuário na forma de pinos. Ao clicar em um pino, informações como quantidade de pessoas e nome do local são exibidas na tela.

Figura 6.3: Detalhe da tela de mapa do aplicativo

6.4 Tela de tirar foto

A tela de tirar foto acessa a funcionalidade nativa da câmera do iOS. Nesta tela o usuário apenas tem a opção de tirar uma foto ou gravar um vídeo e voltar para a tela anterior.

Figura 6.4: Detalhe da tela de tirar foto

6.5 Tela de edição da foto

Após a captura da foto, o usuário pode editar a foto de acordo com o seu gosto. É disponibilizado ao usuário diversas opções de filtros, efeitos, orientação e iluminação da foto. Como exibido na figura 6.5, após a edição da foto, o usuário tem a opção de prosseguir para a tela seguinte do aplicativo clicando no canto superior direito da tela, no botão caracterizado na figura com a palavra *Pronto*. Também é disponibilizado ao usuário a opção de voltar e reproduzir a captura da foto, ao clicar no botão *Cancelar*.

Figura 6.5: Detalhe da tela de editar a foto

6.6 Tela de compartilhamento da foto

Na tela de compartilhamento da foto é exibida ao usuário a foto que ele tirou no canto superior esquerdo, como consta na figura 6.6. O usuário é obrigado a selecionar o local em que a foto foi tirada, deste modo esta foto passará a integrar o *feed* de fotos do local, detalhado na seção 6.8.

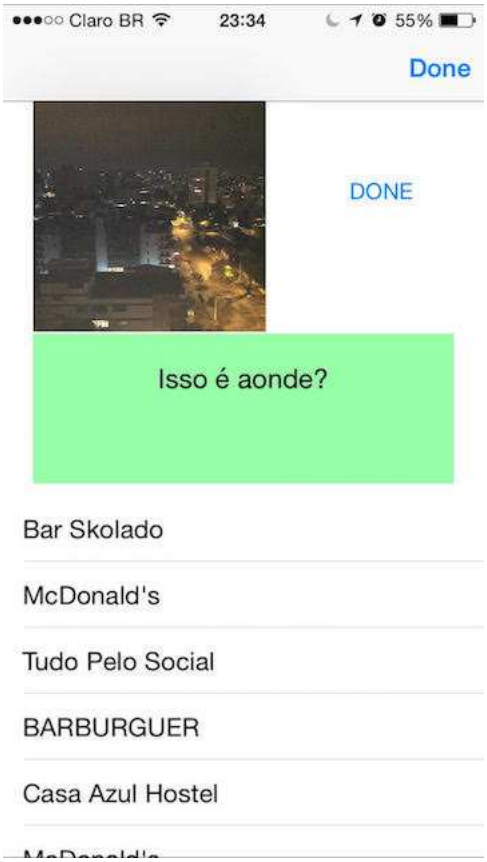

Figura 6.6: Detalhe da tela de compartilhamento da foto

6.7 Tela de lista de locais

Nesta tela são exibidos os locais próximos ao usuário e a respectiva informação do número de pessoas que fizeram *check-in* pela plataforma do Foursquare na última hora. O usuário também possui a opção de procurar pelo nome de um local, no caso do mesmo não estar incluído na lista.

A lista de locais exibidos é originada de recursos oriundos da API do Foursquare, que disponibiliza o nome dos locais próximos ao usuário e informações como o número de pessoas que usaram a funcionalidade de *check-in*. Como fica visível na figura 6.7, cada célula exibe o nome do local no lado esquerdo e o número de check-ins recentes feitos neste mesmo local no lado direito.

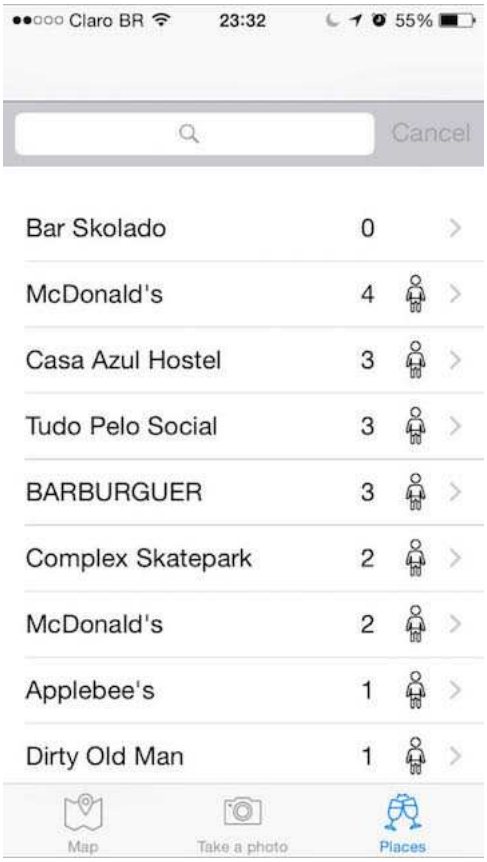

Figura 6.7: Detalhe da tela de lista de locais

6.8 Tela de fotos recentes de um local

Nesta tela são exibidas as mídias recentes relacionadas ao local em questão. As mídias estão ordenadas temporalmente, e são originadas de consultas à API do Instagram e ao banco de dados do aplicativo.

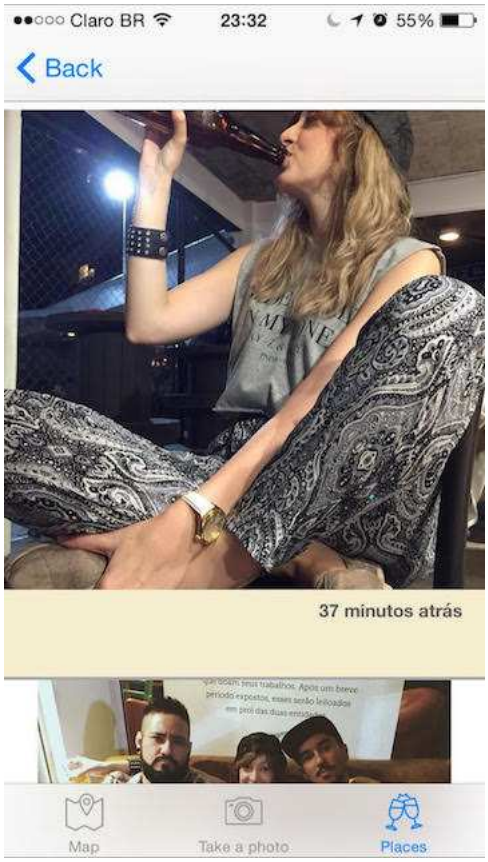

Figura 6.8: Detalhe da tela mídias recentes de um local

7 AVALIAÇÃO

O presente aplicativo seguiu os guias de design propostos pela Apple. A Apple aconselha aos desenvolvedores o uso de algumas tecnologias e componentes padrões. A maioria dos aplicativos desenvolvidos para a plataforma iOS seguem um padrão usual de layout, o mesmo seguido pelo presente trabalho. O objetivo deste capítulo é abordar os aspectos relacionados à experiência de usuário seguidos pelo presente trabalho, e também a avaliação do aplicativo, a qual foi proposta a alguns usuários[1].

7.1 Componentes padrões utilizados

O corrente trabalho utilizou diversos componentes padrões da plataforma iOS que facilitam o uso de um usuário nativo desta plataforma. Abaixo estes componentes serão detalhados.

7.1.1 Tab Bar

O uso do componente nativo da plataforma iOS chamado *Tab Bar* permite que o usuário de uma aplicação iOS possa navegar pelas funcionalidades facilmente. A figura 7.1.1 exibe o uso deste componente na plataforma iOS.

Figura 7.1: Uso do componente Tab Bar no aplicativo

7.1.2 MK Map View

O mapa utilizado pela aplicação, caracterizado pelo componente MK Map View, é um componente nativo de uma aplicação iOS. Além disso, a maneira com que os dados relacionados aos locais são exibidos no mapa, por utilizar o componente MK Map View,

[1]Este capítulo foi baseado em `https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html`

é semelhante à maneira com que outras aplicações com objetivos similares exibem os dados, o que torna a dificuldade de interação com a aplicação menor.

7.1.3 Table View

Para realizar a listagem de dados o aplicativo faz uso do componente nativo da plataforma iOS chamado de *Table View*. As telas de listagem de locais, introduzida na seção 6.7 e de listagem de mídias relacionadas a um local, introduzida na seção 6.8, são exemplos de uso do componente *Table View* da plataforma iOS.

7.2 Avaliação do aplicativo

Como forma de avaliar o aplicativo e a experiência de um usuário ao interagir com o mesmo, foi proposta a realização de um questionário diante da execução de alguns cenários por usuários. Visto que o público-alvo da aplicação está entre a faixa etária de 15 até 35 anos, o questionário limitou-se a usuários desta faixa etária. Os cenários escolhidos foram baseados nas histórias de usuário introduzidas na seção 4.4. Os resultados da avaliação com cada cenário escolhido serão detalhados abaixo nesta seção.

Como forma de medir qualitativamente a experiência de usuário ao realizar os cenários, foi proposto que após a realização de um cenário, o usuário classificasse a realização da interação de 1 a 5 em função da dificuldade, isto é, se um usuário achasse difícil a realização de um cenário, ele classificaria a interação como 5, já se um usuário achasse fácil a realização do mesmo, ele classificaria a interação como 1. Por este motivo o questionário foi aplicado apenas com pessoas familiarizadas com o uso de smartphones.

Cada proposição de cenário aos usuários foi feita de modo oral, após a realização dos cenários, as informações relacionadas às dificuldadades e às facilidades encontradas foram coletadas.

O questionário foi aplicado com 5 usuários e a figura 7.2 exibe o seu resultado. Abaixo serão comentados os resultados obtidos levando em conta cada um dos cenários avaliados.

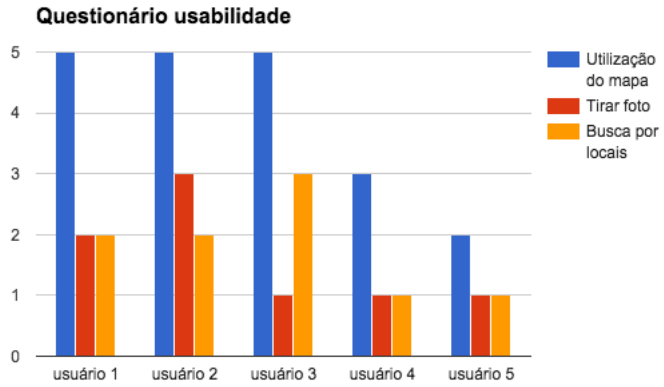

Figura 7.2: Gráfico com o resultado do questionário de usabilidade proposto aos usuários

7.2.1 Interação do usuário com o Mapa

Foi proposto aos usuários que interagissem com o mapa da aplicação, e que avaliassem a dificuldade na interação. Basicamente foi pedido aos usuários que procurassem por bares e restaurantes ao seu redor no mapa da aplicação e que tentassem visualizar informações sobre os mesmos.

Este foi o cenário pior avaliado pelos usuários. Muitos usuários atribuiram a dificuldade da interação à clareza da informação listada no mapa, alguns usuários tiveram dificuldade em clicar nos pinos vermelhos que caracterizam os locais, isto pois no mapa existem outros locais caracterizados por outros formatos diferentes dos pinos vermelhos, os quais não são clicáveis.

7.2.2 Interação do usuário com a funcionalidade de tirar a foto

A interação dos usuários com a funcionalidade de registrar fotos foi avaliada positivamente. Foi pedido aos usuários que tentassem capturar uma foto e que aplicassem efeitos na mesma. Como demonstrado pela figura 7.2 pela cor vermelha, maioria dos usuários consultados avaliou esta interação como positiva, alegando que a experiência de interação com a câmera é semelhante à outros aplicativos.

7.2.3 Interação de um usuário com a funcionalidade de busca local

A interação de usuários com a funcionalidade de busca por locais foi avaliada positivamente, o que é caracterizado pela figura 7.2 pela coluna de cor amarela. Foi pedido aos usuários que procurassem por um local na lista de locais e que tentassem visualizar informações sobre este. Diversos usuários reportaram que a funcionalidade é clara e transparente, já que ao acessar a tela de locais, uma lista de locais ordenada descrescentemente em ordem de número de check-ins realizados nos locais na última hora sempre é

exibida, o que facilitou o entendimento da funcionalidade da tela pelos usuários.

8 CONCLUSÃO

No decorrer do presente trabalho foram abordados pontos relacionados ao crescimento das plataformas móveis e ao mercado de entretenimento e lazer. A partir destes foi proposta a idéia de desenvolvimento de um aplicativo que facilite a visualização de locais de entretenimento.

A partir do desenvolvimento do sistema, foram abordados pontos relacionados ao desenvolvimento de um aplicativo para a plataforma móvel iPhone, bem como as ferramentas utilizadas para tal, frameworks adotados e metodologia seguida durante o desenvolvimento.

Ao final do trabalho foi detalhado o protótipo obtido como resultado da implementação da solução proposta. Foram detalhadas todas as telas que fazem interação com o usuário e suas respectivas funcionalidades. Finalmente, os resultados de um questionário aplicado aos usuários do aplicativo foram apresentados e comentados.

A idéia é que a partir do protótipo desenvolvido neste trabalho seja possível futuramente desenvolver alguns pontos do aplicativo e disponibilizá-lo na loja de aplicativos da Apple, a App Store.

Pelo sistema atual ser apenas um protótipo, há ainda espaços para melhorias e novas funcionalidades, as quais serão citadas na seção 8.1

8.1 Trabalhos Futuros

É natural que durante o desenvolvimento de um aplicativo surjam novas idéias e pontos em que melhorias possam ser adotadas. A idéia dessa seção é abordar as possíveis melhorias que possam ser adotadas no aplicativo e também novas plataformas em que o aplicativo possa ser desenvolvido.

8.1.1 Novas Plataformas

Além da plataforma para qual o presente trabalho foi desenvolvido, existem outras plataformas móveis como Android, Symbian OS e Windows Phone que suportariam o desenvolvimento de um aplicativo semelhante ao desenvolvido no presente sistema. O

desenvolvimento do mesmo aplicativo para qualquer uma das plataformas citadas acima seria possível. Como sugestão, a plataforma Android seria aconselhada, já que ferramentas empregadas no desenvolvimento deste software poderiam ser reaproveitadas. A ferramenta Parse, por exemplo, possui suporte para a plataforma Android, ou seja, o banco de dados utilizado para a aplicação poderia ser facilmente acessado por um aplicativo desenvolvido na plataforma Android.

8.1.2 Integração com outras redes sociais

Por se tratar de um aplicativo de uso social, diversas outras redes sociais, como o Facebook e Twitter, poderiam ser integradas na aplicação. A aplicação poderia se beneficiar da integração com o Facebook para obter acesso aos eventos de usuários e assim prover locais de entretenimento de uma forma mais esperta ao utilizador.

A ferramenta Twitter por sua vez poderia ser integrada como forma de fornecer atualizações em tempo real via tweets sobre os locais de entretenimento.

8.1.3 Independência da plataforma Instagram

O corrente sistema, para fazer uso de dados providos pela API do Instagram obriga que o usuário efetue o login utilizando uma conta da plataforma do Instagram. Esta imposição feita pelo sistema é decorrente do fato de a API do Instagram possuir um limite no número de requisições que possam ser feitas por usuário.

Uma forma alternativa de login poderia ser implementada no aplicativo de forma que o usuário não fosse impossibilitado de usá-lo no caso de não possuir uma conta da plataforma do Instagram.

REFERÊNCIAS

COHN, M. **User stories applied**: for agile software development. [S.l.]: Addison-Wesley Professional, 2004.

FLING, B. **Mobile design and development**: practical concepts and techniques for creating mobile sites and web apps. [S.l.]: O'Reilly Media, Inc., 2009.

GOLDSTEIN, N. **Objective-C for Dummies**. [S.l.]: Wiley Publishing, Inc, 2009.

IDC. Disponível em : <http://br.idclatin.com/releases/news.aspx?id=1801/>.Acesso em: Abril 2015.

KNIBERG, H.; SKARIN, M. **Kanban e Scrum obtendo o melhor de ambos**. [S.l.]: C4Media Inc, 2009.

SEBRAE. Disponível em : <http://www.sebrae.com.br/sites/PortalSebrae/artigos/Bares-e-restaurantes:-um-setor-em-expans%C3%A3o/>.Acesso em: Março 2015.

SOMMERVILLE, I. **Software Engineering.** [S.l.]: Addison Wesley, 2004.

VAISH, G. **Getting started with NoSQL**. [S.l.]: PACKT, 2013.

WEST M., D. **Ten Facts about Mobile Broadband**. Disponível em : <http://www.brookings.edu/research/papers/2011/12/08-mobile-broadband-west>.Acesso em: Junho 2015.