**MINISTÉRIO DA DEFESA**

**EXÉRCITO BRASILEIRO**

**DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA**

**INSTITUTO MILITAR DE ENGENHARIA**

**Seção de Engenharia de Computação / SE8**

**Felipe da Costa Rasinhas**

**Jonathan Correia Carvalhosa**

**PROJETO DE ANÁLISE E DESENVOLVIMENTO DE UMA SOLUÇÃO MÓVEL INTEGRADORA**

**Rio de Janeiro**

**2013**

**INSTITUTO MILITAR DE ENGENHARIA**

**Felipe da Costa Rasinhas**

**Jonathan Correia Carvalhosa**

**PROJETO DE ANÁLISE E DESENVOLVIMENTO DE UMA SOLUÇÃO MÓVEL INTEGRADORA**

Projeto de fim de curso apresentado ao Curso de Graduação de Engenharia de Computação como requisito parcial para a obtenção do título de Engenheiro.

Orientador: Prof. Ricardo Choren Noya

Rio de Janeiro

2013

c2013

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro - RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

XXXX CARVALHOSA, Jonathan Correia. RASINHAS, Felipe da Costa.

Projeto de análise e desenvolvimento de uma solução móvel integradora / Jonathan Correia Carvalhosa, Felipe da Costa Rasinhas. Rio de Janeiro: Instituto Militar de Engenharia, 2013.

29 f.

Projeto de fim de curso (graduação) – Instituto Militar de Engenharia, 2013.

1. ###PALAVRAS CHAVE###

**INSTITUTO MILITAR DE ENGENHARIA**

**JONATHAN CORREIA CARVALHOSA**

**FELIPE DA COSTA RASINHAS**

**PROJETO DE ANÁLISE E DESENVOLVIMENTO DE UMA SOLUÇÃO MÓVEL INTEGRADORA**

Projeto de fim de curso do Curso de Engenharia de Computação do Instituto Militar de Engenharia, como requisito para colação de grau no curso de Engenharia de Computação.

Orientador: Prof. Ricardo Choren Noya

Aprovada em \_\_ de \_\_\_\_\_\_\_\_de 2013 pela seguinte Banca Examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Ricardo Choren Noya, D.C. , do IME– Presidente

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Maj QEM Anderson Fernandes P. dos Santos, D.Sc.., do IME

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Profa. Raquel Coelho Gomes Pinto, D.Sc., do IME

Rio de Janeiro

2013

SUMÁRIO

Lista de Figuras 4

1 INTRODUÇÃO 7

1.1 OBJETIVO 7

1.2 MOTIVAÇÃO 7

1.3 METODOLOGIA 8

1.4 ORGANIZAÇÃO DA MONOGRAFIA 8

2 DESENVOLVIMENTO 9

2. 1 FERRAMENTAS UTILIZADAS 9

2.1.1 RIO DATAMINE [1] 9

2.1.1.1 AUTENTICAÇÃO 9

2.1.1.2 REQUISIÇÃO 10

2.1.1.3 RESPOSTA 10

2.1.2 IOS [2] 10

2.1.3 OBJECTIVE C 11

2.1.4 PYTHON [3] 11

2.1.5 DJANGO [4] 11

2.1.6 MYSQL [5] 11

2.1.7 SOUTH [6] 12

2.1.8 ASI-HTTP-Request 12

2.1.9 SBJSON 12

2.1.10 GIT 12

2.2 ARQUITETURA 12

3 DOCUMENTAÇÃO 14

3.1 USUÁRIOS 14

3.1.1 REGISTRAR USUÁRIO 14

3.1.2 LOGIN 15

3.1.3 LOGOUT 16

3.1.4 EDITAR DADOS 17

3.2 PESQUISAS 18

3.2.1 UTILIDADE PÚBLICA 18

3.2.2 RESTAURANTES 19

3.2.3 ESPORTES 19

3.2.4 CULTURA 20

3.3 GERENCIAMENTO DE PREFERÊNCIAS 20

5 CONCLUSÃO 21

6 REFERÊNCIAS BIBLIOGRÁFICAS 23

# Lista de Figuras

FIG. 2.2.1 Arquitetura do Sistema..............................................................................13

FIG. 3.2.1 Diagrama de Sequência............................................................................24

**RESUMO**

Nos próximos anos, o Rio de Janeiro sediará eventos de caráter global, e por este motivo, haverá um grande número de turistas na cidade durante este período. Nesse contexto, é importante o desenvolvimento de softwares de apoio, que possam ajudar essas pessoas, desde a escolha do transporte a ser utilizado até a chegada no local e a localização de pontos estratégicos, como polícia, bombeiros, hospitais e banheiros públicos.

Este trabalho consiste no desenvolvimento de uma aplicação *mobile* que fornece essas informações através de consultas a uma base de dados de apoio do governo, o Rio DataMine. Em paralelo a isso, será configurado um servidor web para centralizar as consultas e armazenar informações sobre os usuários.

Com esse trabalho, espera-se facilitar o deslocamento dos turistas pela cidade, disponibilizando informações que tornem sua estadia mais prazerosa.

**ABSTRACT**

In the next years, Rio de Janeiro will host global scale events, and for that reason, there will be an enormous number of tourists in town at this period. Within this context, the development of assist softwares is very important, so it can help them choosing the transportation, finding the event places and localizing strategic points, like police departments, fire departments, hospitals and public bathrooms.

This project consists in the development of a mobile application that provides this information through queries to an auxiliary database from the government, the Rio DataMine. In parallel to the mobile application, a web server will be configured, so it centralizes the queries and stores data from the users.

This work is expected to improve the mobility of tourists around town, providing information to make their stay more enjoyable.

# 1 INTRODUÇÃO

## ****1.1 OBJETIVO****

Este trabalho tem como objetivo o desenvolvimento de uma aplicação para dispositivos móveis que proporcione a integração dos dados da Prefeitura para melhorar sua acessibilidade e mobilidade. Desta forma, pretende ajudar os turistas a se deslocarem com mais facilidade pela cidade do Rio de Janeiro, tirando dúvidas quanto a localização dos meios de transporte, dos serviços de utilidade pública, e sugerindo locais a serem visitados de acordo com as suas preferências.

## 1.2 MOTIVAÇÃO

Prestes a protagonizar o papel de centro das atenções mundiais, o Rio de Janeiro precisa oferecer aos turistas informações claras, concisas e corretas. Não é o que se vê hoje. Entidades públicas e privadas raramente fornecem dados de forma sistêmica. Com isso, informações essenciais ficam soltas no ar. Por outro lado, é muito difícil ao turista incauto ter acesso às mesmas.

Com uma segurança pública que ainda deixa bastante a desejar e uma sinalização das vias públicas ineficiente, o turista fica perdido. A principal motivação para esse trabalho é disponibilizar ao turista as informações de que ele precisa para que sua estadia na cidade seja mais prazerosa.

Outra grande motivação é o desafio de aprender mais sobre o desenvolvimento de aplicativos para iPhone, que é um mercado que está em grande ascensão. Através da *app store*, a Apple permite que os desenvolvedores do aplicativo ganhem 70% de todo o lucro gerado pela venda. Além disso, a documentação presente no site destinado aos desenvolvedores é amigável e possui muitos exemplos que facilitam o aprendizado.

## 1.3 METODOLOGIA

A base de dados utilizada foi a do site Rio DataMine [2]. As principais vantagens que influenciaram nessa decisão foram a facilidade de uso e a alta disponibilidade, além de ser uma ferramenta criada pela própria prefeitura com o objetivo de estimular a criação de aplicativos. A primeira parte do trabalho foi o estudo do formato exigido nos métodos HTTP para as requisições e para as respostas do site e, para a próxima etapa será desenvolvido um serviço para automatizar o acesso a base de dados.

O núcleo da aplicação foi desenvolvido a partir do modelo de casos de uso e será posteriormente revisado para que seja alcançada a versão final.

O dispositivo escolhido para hospedar o aplicativo foi o iPhone. Entre as principais vantagens que influenciaram esta decisão estão a grande quantidade de usuários do iPhone em todo o mundo e a facilidade de acesso a documentação da Apple. Para a próxima etapa do trabalho, será desenvolvida uma primeira interface para o aplicativo no iPhone.

## 1.4 ORGANIZAÇÃO DA MONOGRAFIA

O trabalho está distribuído em uma seção abordando as tecnologias utilizadas para o desenvolvimento do aplicativo, descrevendo todas as ferramentas a serem utilizadas ao longo do projeto e a arquitetura do sistema; seguida de uma seção onde é apresentada a documentação do projeto. A subseção 3.1 descreve a especificação dos casos de uso do sistema e a seção 3.2 expõe um diagrama de sequência relativo à principal funcionalidade do sistema. O relatório é finalizado na seção 4 com a apresentação de um cronograma com as atividades a serem cumpridas até o final do projeto.

# 2 DESENVOLVIMENTO

Nesta seção serão abordados tópicos relativos a escolhas feitas para o pleno desenvolvimento do software, bem como as ferramentas utilizadas para a implementação e a arquitetura do sistema.

## 2. 1 FERRAMENTAS UTILIZADAS

Para a execução do projeto, foi necessária a escolha de uma plataforma *mobile*, uma base de dados contendo as informações sobre a cidade e um framework de desenvolvimento web. Esta seção tem como objetivo esclarecer a escolha de cada um destes componentes do sistema além de citar algumas das características dos mesmos.

### 2.1.1 RIO DATAMINE [1]

O Rio DataMine é uma iniciativa do governo para melhorar a acessibilidade, a transparência e a mobilidade dos dados  públicos  pra os moradores da cidade. Para isso, o site fornece acesso a um servidor composto por bases de dados legíveis por computadores e dispositivos similares.

O acesso aos dados pode ser feito por qualquer pessoa através de requisições HTTP.  O formato esperado da requisição e da resposta será brevemente descrito, de acordo com os padrões especificados pela API do Rio DatamMine.

#### 2.1.1.1 AUTENTICAÇÃO

Para um aplicativo acessar os serviços da API do Rio DataMine, é necessária a posse de um *token* conhecido com *accesstoken*, cuja validade é de seis horas. Para obter o *accesstoken*, o aplicativo deve realizar um HTTP GET passando os parâmetros app\_id e app\_secret, que são identificadores de um aplicativos e são obtidos quando o aplicativo é cadastrado no site.

#### 2.1.1.2 REQUISIÇÃO

O formato esperado é: http://api.riodatamine.com.br/rest/DATASET?QUERY, onde *dataset*  e *query* são, respectivamente, o serviço e o conjunto de filtros que a serem aplicados na consulta.

#### 2.1.1.3 RESPOSTA

O formato da resposta é, por padrão, o formato JSON. Esse formato consiste em uma lista de pares (chave, valor) e é muito comum em aplicações web devido a facilidade na manipulação dos dados de um objeto json.

### 2.1.2 IOS [2]

O iOS é o sistema operacional móvel desenvolvido pela Apple. Ele foi desenvolvido originalmente para iPhone, mas acabou sendo usado também em iPod, iPad e Apple TV. Ele é dividido em quatro camadas básicas:

* Core OS Layer: É a camada mais básica, que se comunica diretamente ao hardware e serve diretamente às camadas superiores.
* Core Services Layer: Contém os serviços mais fundamentais, que todas as outras aplicações podem usar. Ela cuida, por exemplo, da conexão com a internet, do contador de referência das variáveis, da criptografia e do serviços de compartilhamento de arquivos.
* Media Layer: Contém as ferramentas gráficas e as tecnologias de vídeo, áudio disponíveis no aparelho.
* Cocoa Touch Layer: É a camada mais externa, e contém as tecnologias necessárias para o contato direto entre o aparelho e o usuário, como captura dos toques na tela e envio de notificações por push.

### 2.1.3 OBJECTIVE C

O Objective C é uma linguagem de alto nível orientada a objeto, que adiciona eventos e troca de mensagens à tradicional linguagem C. Essa é principal linguagem utilizada pela Apple e é nela que devem ser desenvolvidos os aplicativos para iPhone.

### 2.1.4 PYTHON [3]

Python é uma linguagem de programação interpretada, de alto nível e orientada a objetos. Esta linguagem foi desenvolvida com o objetivo de priorizar a legibilidade do código e sua sintaxe possibilita expressar conceitos em menos linhas de código do que em linguagens como C e Java.

Os principais motivos da escolha do python como linguagem base para o desenvolvimento da plataforma web foram a facilidade e velocidade de se desenvolver um código na linguagem, além do conhecimento de vários frameworks de apoio para o desenvolvimento de uma aplicação web como o Django e o South.

### 2.1.5 DJANGO [4]

O Django é um framework *open source* para o desenvolvimento de aplicações web utilizando Python. Este framework é baseado no padrão MVC (Model-View-Controller) e fornece facilidades como uma modelagem de dados ORM (Object-Relational Mapping), uma linguagem de templates extensa e amigável, um sistema de serialização de dados que possibilita a representação de XML e JSON como instancias de objetos e um sistema de internacionalização.

### 2.1.6 MYSQL [5]

O MySQL é um sistema de gerenciamento banco de dados que utiliza a linguagem SQL. É uma uma aplicação open source e é utilizado um dos sistemas de bancos de dados mais populares do mundo. A escolha desta ferramenta deu-se principalmente por fatores como a baixa necessidade de recursos, a facilidade de uso, o bom desempenho e a compatibilidade com a linguagem de escolha através do MySQLdb.

### 2.1.7 SOUTH [6]

É uma biblioteca de migração de bases de dados para o framework Django. Essa ferramenta verifica se algo foi modificado em algum dos modelos do Django e gera migrações automáticas para o banco de dados para deixa-lo equivalente ao especificado no arquivo models.py.

### 2.1.8 ASI-HTTP-Request

Trata-se de uma biblioteca que visa facilitar a comunicação de aplicações para iPhone com servidores web através de requisições HTTP.

### 2.1.9 SBJSON

O SBJSON é uma API voltada para a manipulação de objetos do tipo json (javascript object notation) em objective-C. Esta ferramenta é utilizada basicamente para “parsear” e gerar objetos json para a comunicação dos dispositivos com o servidor web.

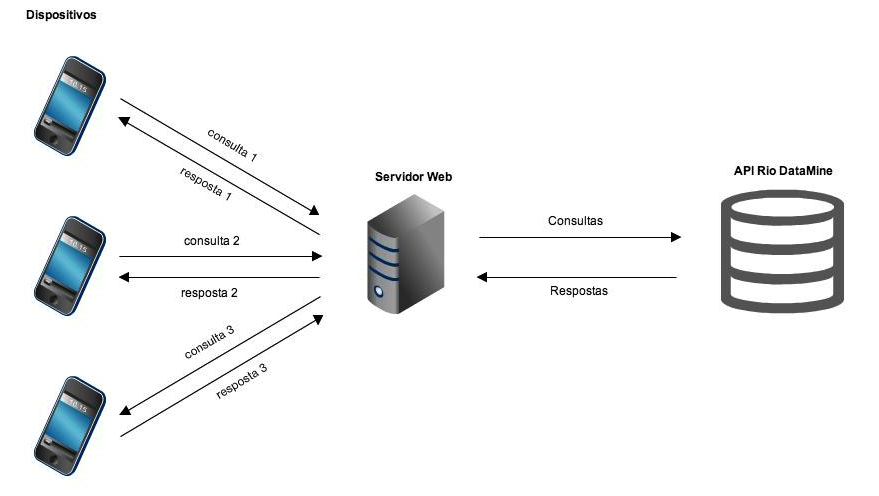
### 2.1.10 GIT

O Git é uma ferramenta de controle de versão desenvolvido originalmente por Linus Torvalds para versionar o Kernel Linux.

## 2.2 ARQUITETURA

Para o desenvolvimento deste sistema foi considerada a utilização apenas da plataforma móvel porém esta escolha mostrou-se ineficiente devido a necessidade do token de autenticação do Rio DataMine para executar qualquer consulta. Se o software fosse desenvolvido utilizando esta arquitetura, seria necessário ter um identificador da API para cada dispositivo móvel que utilizasse o aplicativo.

Por este motivo verificou-se que seria mais eficiente centralizar estas consultas em um servidor web, que tem o trabalho de receber as requisições feitas por todos os dispositivos, e repassá-las ao servidor do Rio DataMine. Isso faz com que um identificador único, utilizado pelo servidor, seja suficiente para satisfazer a todos os dispositivos.



**FIG. 2.2.1 –** Arquitetura do Sistema

A escolha da utilização de um servidor web mostrou-se extremamente oportuna pois também centraliza todas as informações sobre as pesquisas feitas pelos usuários, fornecendo dados extremamente essenciais para possíveis melhorias a serem feitas no aplicativo posteriormente.

# 3 DOCUMENTAÇÃO

Neste capítulo serão apresentadas detalhadamente todas as funcionalidades do sistema.

O projeto é composto de duas partes fundamentais: o código embarcado no *iphone* de cada cliente, que é visto como um aplicativo de celular, e o servidor web usado para centralizar as requisições e armazenar o histórico. Dessa forma, a documentação abrangerá tanto o aplicativo de *iphone* como o servidor web, descrevendo as funcionalidades envolvidas e a utilização de cada uma delas.

## 3.1 USUÁRIOS

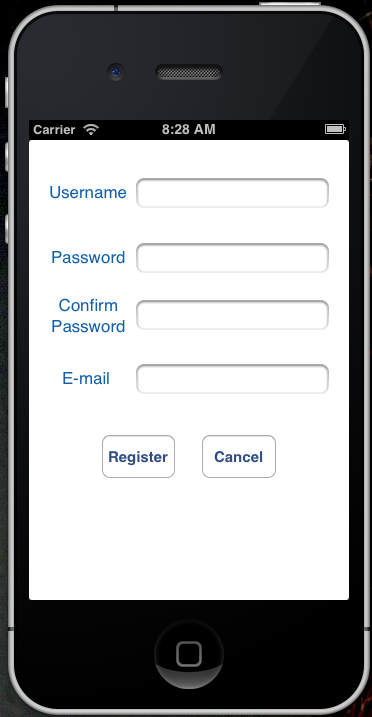
O projeto inclui a gerência de usuários, que é feita da seguinte maneira:

1. O cliente cria uma conta usando seu celular através da opção Registrar Usuário
2. O cliente faz *login* utilizando seu *username* e *password*.
3. O cliente pode modificar seus dados através da opção Editar Dados.

A seguir serão abordadas detalhadamente cada uma dessas funcionalidades.

### 3.1.1 REGISTRAR USUÁRIO

Quando o cliente obtém o aplicativo, essa é a única funcionalidade disponível. As informações necessárias são *username*, password e email, com um campo para a confirmação do password, conforme mostrado na imagem a seguir:

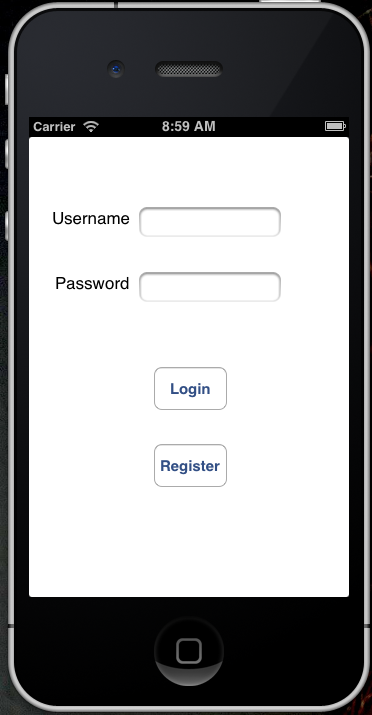


Assim que o botão *Register* é pressionado, o iphone verifica apenas se todos os campos foram preenchidos e a confirmação do *password* está correta e envia uma requisição com os dados para o servidor web.

A requisição é validada se o campo username for diferente do de todos os outros usuários anteriormente cadastrados. Nesse caso, um novo usuário será criado e armazenado no banco de dados. Com a resposta da requisição do servidor, o iphone notifica ao cliente de que o usuário foi criado, ou mostra os erros de validação que foram encontrados.

### 3.1.2 LOGIN

O *login* é a porta de entrada para o aplicativo e a sua execução pelo cliente permite que todas as outras funcionalidades sejam liberadas. A seguir, uma imagem com a tela de *login* no *iphone*:



Assim que o botão Login é pressionado, o celular verifica se os dados foram preenchidos e envia a requisição ao servidor, que faz uma busca no banco de dados para recuperar as informações do usuário e responde à requisição com o identificador do usuário ou uma mensagem de erro.

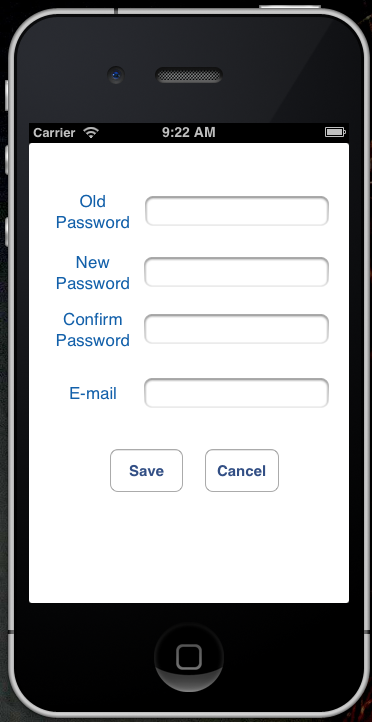
Assim que o aparelho receber o identificador do usuário, ele salva esse número em uma variável acessível em todo o ambiente do sistema, de forma que essa informação possa ser futuramente recuperada, e redireciona o cliente para uma tela com o menu principal.

### 3.1.3 LOGOUT

Quando o cliente quiser desconectar seu usuário do sistema ele pode utilizar a funcionalidade de Logout. Diretamente no menu inicial, ele aperta o botão Logout, que funciona sem o auxílio do servidor. Assim que o botão é pressionado, o iphone libera o identificador de usuário que estava armazenado e retorna para a tela de Login.

### 3.1.4 EDITAR DADOS

Caso tenha o interesse de mudar alguma informação relacionada ao seu usuário, o cliente pode usar a funcionalidade Edit Profile, cuja tela está representada na imagem a seguir:



O cliente preenche os campos, o iphone verifica se todos eles estão preenchidos e se a confirmação da senha nova está correta e envia a requisição ao servidor, acrescentando a informação do identificador do usuário, que passa a ficar disponível em todas as telas após a realização do Login.

O servidor, por sua vez, verifica se a senha antiga está correta e então atualiza as informações do usuário, retornando uma mensagem de sucesso para o celular.

\*\*\*\*FIXME\*\*\* se o usuário so quiser trocar o email basta deixar os outros campos em branco e vice versa.

## 3.2 PESQUISAS

Esta é a principal funcionalidade do aplicativo e pode ser acessada através do botão Search localizado no menu inicial do sistema. Através desta ferramenta o usuário consegue informações sobre serviços e locais na cidade além de um mapa plotando o local de interesse e a posição do usuário. Os pontos de interesse foram divididos em quatro grupos:

1. Utilidade Pública: Neste grupo o usuário consegue acessar informações sobre hospitais, delegacias e corpos de bombeiros.
2. Restaurantes: Este grupo fornece ao usuário informações sobre restaurantes na cidade.
3. Esportes: O usuário tem acesso a informações sobre complexos esportivos localizados na cidade.
4. Cultura: São fornecidas informações sobre eventos culturais realizados na cidade.

A seguir serão abordados cada um destas funcionalidades.

## 3.2.1 UTILIDADE PÚBLICA

Caso o usuário necessite da ajuda de algum serviço de utilidade pública, o mesmo deve selecionar a opção *Utilities* no menu de pesquisas. A seguir será apresentado um formulário onde deve ser preenchido o nome do bairro em que se deseja realizar a busca no campo *neighbourhood* e pressionado o botão relativo ao serviço que se deseja obter as informações.

O sistema fará uma requisição ao servidor web que armazenará a pesquisa feita pelo usuário no banco de dados e tentará consultar a base de dados do Rio DataMine. Caso a base de dados retorne uma resposta, o servidor enviará uma resposta em json ao aplicativo, que apresentara uma lista com os resultados na tela. Caso o Rio DataMine não retorne uma resposta, uma pesquisa será feita no banco de dados de apoio existente no servidor web e as respostas serão apresentadas.

Na lista de resultados, serão apresentadas várias informações sobre os pontos encontrados, como telefone, nome e endereço. Além disso, o usuário pode também clicar em um dos resultados para ter acesso a um mapa que indicará a posição do local selecionado e a localização atual do usuário caso lhe seja permitido.

## 3.2.2 RESTAURANTES

Se o usuário estiver procurando um local para se alimentar, basta selecionar a opção *Restaurants*  no menu de pesquisas. O sistema apresentará um formulário onde o usuário deverá selecionar o bairro em que deseja procurar o serviço, os tipos de restaurantes que deseja encontrar (fast food, restaurantes, BACON \*\*FIXME\*\* e bares), e também a faixa de preço que está disposto a pagar.

O sistema fará uma requisição ao servidor web e assim como na funcionalidade anterior apresentará uma lista com os locais que atendem aos filtros especificados pelo usuário.

A lista de resultados contara com dados como: nome, faixa de preço, endereço e uma breve descrição do estabelecimento. O usuário pode selecionar o item da lista que mais se interessou e visualizar o mapa.

## 3.2.3 ESPORTES

Se o usuário estiver interessado em acessar um complexo desportivo como um estádio ou um ginásio, poderá selecionar a opção *Sports* no menu de pesquisas. \*\*FIXME \*\* que filtros??

O sistema fará uma requisição ao servidor web e assim como na funcionalidade de busca por serviços de utilidades públicas apresentará uma lista com os locais que atendem aos filtros especificados pelo usuário.

A lista de resultados contara com dados como: nome e endereço. O usuário pode selecionar o item da lista que mais se interessou e visualizar o mapa.

## 3.2.4 CULTURA

Caso o usuário esteja a procura de um evento cultural para comparecer, ele pode selecionar a opção *Culture* no menu de pesquisas. \*\*FIXME \*\* que filtros??

O sistema fará uma requisição ao servidor web e assim como na funcionalidade de busca por serviços de utilidades públicas apresentará uma lista com os locais que atendem aos filtros especificados pelo usuário.

A lista de resultados contara com dados como: nome, endereço e data. O usuário pode selecionar o item da lista que mais se interessou e visualizar o mapa.

## 3.3 GERENCIAMENTO DE PREFERÊNCIAS

Esta funcionalidade tem como objetivo facilitar a utilização das opções de busca do usuário. No menu principal, o usuário poderá selecionar a opção *Manage Preferences*  para poder mudar as suas preferências de busca.

Nesta tela será apresentado um formulário onde o usuário deverá preencher os campos relativos aos grupos de busca como: *Neighbourhood* (para todos os grupos), tipos e faixa de preço para restaurantes, e tipos para eventos. Para salvar os dados, o usuário deverá pressionar o botão save. Os dados serão enviados ao servidor para serem salvos no banco de dados além de ser salvos no dispositivo.

Sendo assim, em qualquer momento que o usuário logar em sua conta (em qualquer dispositivo) essas preferencias serão salvas no novo dispositivo até o logout do usuário.

Tendo preenchido e salvo estas informações, para todas as buscas que o usuário fizer a partir deste instante, os formulários estarão pré preenchidos com as informações das preferencias do cliente.

# 5 CONCLUSÃO

Este trabalho é muito importante dada a situação atual da cidade do Rio de Janeiro. O Rio foi recentemente escolhido para ser sede de eventos de caráter mundial, que trazem à cidade um grande número de turistas. Dessa forma, aplicativos para ajudar na locomoção e na localização de pontos na cidade são extremamente úteis.

Foi utilizada uma grande variedade de ferramentas *open source* que foram essenciais para o desenvolvimento do trabalho. O código foi versionado através de um repositório criado no GitHub, o servidor de dados foi feito com o Django, que é um *framework* extremamente completo para Python e os *frameworks* para *Objective-c* TDBadgedCell, ASIHttpRequest e Json nos ajudaram a construir o código para o aplicativo no Iphone. O trabalho agora está disponível no GitHub para qualquer pessoa que se interesse, podendo ainda ser utilizado como fonte de consulta para outros trabalhos nessa área.

As dificuldades encontradas dizem respeito principalmente à base de dados da prefeitura, o RioDatamine. O serviço prestado por eles fica fora do ar com muita frequência, o que nos levou a fazer um *backup* das informações em nosso servidor para não deixar o erro chegar aos usuários.

Além disso, algumas informações não tinham o nível de detalhe que nós esperávamos. Por exemplo: as informações sobre eventos esportivos nos diziam apenas os lugares que sediavam os eventos, mas não o momento que esses eventos iam acontecer. Por isso acabamos por desistir da ideia de notificar os usuários quando a data de algum evento esportivo que fosse do interesse dele estivesse próxima.

Outro problema com o RioDatamine foi a codificação usada por eles. Alguns caracteres comuns na língua portuguesa, como ã, á, â, ç apresentam problemas de codificação, o que prejudica a leitura das informações e faz com que os usuários questionem a qualidade do aplicativo.

Para trabalhos futuros de caráter geral, seria muito interessante um trabalho para melhorar o visual do aplicativo. As telas poderiam ter mais imagens e botões personalizados que facilitassem a utilização e tornassem o aplicativo mais bonito. Além disso, as buscar poderiam ser mais inteligentes, para tratar possíveis erros de digitação, o que é comum para uma pessoa que não está ambientada com os nomes dos bairros ou expressões locais. O mapa que mostra a localização dos resultados também poderia ser melhorado, permitindo que os usuários buscassem maneiras de sair do local em que eles estão para ir para o local desejado.

# 6 REFERÊNCIAS BIBLIOGRÁFICAS

[1] RIO DATAMINE. Disponível em <riodatamine.com.br >

Acesso em 12 de Outubro de 2012.

[2] APPLE DEVELOPER CENTER. Disponível em <developer.apple.com >

Acesso em 12 de Outubro de 2012.

[3] PYTHON. Disponível em <www.python.org/>

Acesso em 14 de Março de 2013

[4] DOCS DJANGOPROJECT.

Disponível em < docs.djangoproject.com/en/1.5/>

Acesso em 13 de Março de 2013.

[5] MYSQL. Disponível em < www.mysql.com/>

Acesso em 14 de Março de 2013.

[6] SOUTH. Disponível em <south.readthedocs.org/en/0.7.6/>

Acesso em 14 de Março de 2013.