

Matheus Bernardo Ferreira

**Aplicação Móvel em Prol do Patrimônio  
Histórico de Diamantina**

Diamantina

2017

Matheus Bernardo Ferreira

# **Aplicação Móvel em Prol do Patrimônio Histórico de Diamantina**

Trabalho apresentado ao curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM, como pré-requisito para obtenção do grau de bacharel, sob orientação do Prof. Me. Eduardo Pelli.

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI  
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
CURSO DE SISTEMAS DE INFORMAÇÃO

Orientador: Eduardo Pelli

Diamantina

2017

Monografia de projeto final de graduação sob o título “Aplicação Móvel em Prol do Patrimônio Histórico de Diamantina”, defendida por Matheus Bernardo Ferreira e aprovada em 22 de março de 2017, em Diamantina, Minas Gerais.

Banca Examinadora:



---

Prof. Msc. Eduardo Pelli  
Orientador



---

Profª. Msc. Cinthya Rocha Tameirão



---

Prof. Msc. Marcelo Ferreira Rego

*Aos meus avós Bené, Zezé, Zélia e Antônio*

# AGRADECIMENTOS

Agradeço a meus pais, pois me ensinaram que o conhecimento é a chave para se obter sucesso na vida, por terem me dado uma ótima educação e por terem me mostrado o caminho correto a seguir. Ao meu irmão que sempre me apoiou e esteve ao meu lado. Aos meus avós Bené, Zezé, Zélia e Antônio por todo apoio que me deram.

A Laya pela companhia, paciência, apoio, momentos vividos e pelos conselhos que me fazem enxergar as coisas de outro ângulo, e assim possibilita meu crescimento.

Ao meu orientador Eduardo Pelli pela oportunidade, paciência e conhecimentos compartilhados. Agradeço também aos professores e técnicos do Departamento de Computação da UFVJM, que foram extremamente importantes e me guiaram durante essa minha jornada.

Aos amigos que trilharam comigo o tortuoso caminho da graduação. A todos ex-integrantes e agregados da República Catapulta que tornaram as horas difíceis mais fáceis. Aos meus amigos de longa data que mesmo a distância foram de fundamental importância para eu me manter firme nessa caminhada.

*"Não devemos nos questionar porque algumas coisas nos acontecem  
e sim o que podemos fazer com o tempo que nos é dado."  
(Gandalf: O Senhor dos Anéis)*

# RESUMO

O objetivo geral deste trabalho é planejar e desenvolver um aplicativo para dispositivos móveis que auxilie o visitante na cidade de Diamantina – Minas Gerais. Diferentes setores da sociedade têm sido modificados a partir da inserção de aparatos tecnológicos que visam facilitar o acesso das pessoas a informação. Em função disso, propõe-se a desenvolver um aplicativo móvel que tem a finalidade de promover uma melhor interação entre o turista e os monumentos históricos da cidade de Diamantina. Para alcançar o objetivo proposto, foram utilizadas tecnologias como o *Quick-Responsive Code* e os Serviços Baseados em Localização(LBS), de modo a apoiar e elaborar a construção do sistema. A integração dessas tecnologias pode contribuir no sentido de prover o conteúdo da melhor maneira possível aos usuários. O trabalho aborda questões relacionadas ao planejamento, estudo sobre as tecnologias disponíveis e levantamento dos dados referentes aos patrimônios. Os resultados obtidos se relacionam com a construção de um aplicativo que procura servir como um guia virtual de Diamantina com praticidade e atratividade.

**Palavras-chave:** Diamantina. Dispositivos Móveis. Patrimônio Histórico. QR Code. LBS.

# ABSTRACT

The objective of this work is to design and develop a mobile application that helps the visitor in the city of Diamantina - Minas Gerais. In Information Era different sectors of society have been modified from the insertion of technological devices that aim to facilitate people's access to information. As a result, it is proposed to develop a mobile application that has the purpose of promoting a better interaction between the tourist and the historical monuments of the city of Diamantina. In order to achieve the proposed objective, technologies such as Quick-Responsive Code and Location-Based Services (LBS) were used to support and elaborate the construction of the system. The integration of these technologies can contribute to providing content in the best possible way to users. The paper addresses issues related to planning, study of available technologies and collection of data on assets. The results obtained are related to the construction of an application that seeks to serve as a virtual guide to Diamantina with practicality and attractiveness.

**Keywords:** Diamantina. Mobile Devices. Historical Patrimony. QR Code. LBS.



# LISTA DE ILUSTRAÇÕES

|  |    |
|--|----|
| Figura 1 – Números de Aparelho por Plataforma . . . . .                          | 17 |
| Figura 2 – iPhone exibindo três aplicações distintas. . . . .                    | 19 |
| Figura 3 – Evolução do sistema móvel da Microsoft . . . . .                      | 20 |
| Figura 4 – Comparação entre as Plataformas . . . . .                             | 22 |
| Figura 5 – HTC Dream, primeiro dispositivo comercializado com Android . . . . .  | 23 |
| Figura 6 – Evolução das Versões do Android . . . . .                             | 24 |
| Figura 7 – Arquitetura da plataforma Android . . . . .                           | 25 |
| Figura 8 – Arquitetura dos LBS . . . . .   | 28 |
| Figura 9 – Mapa de Salerno de duas fontes distintas . . . . .                    | 30 |
| Figura 10 – Exemplo de um QR Code . . . . .                                      | 31 |
| Figura 11 – Comparação de armazenamento: QR Code x Código de Barra . . . . .     | 31 |
| Figura 12 – Divulgação de duas aplicações com QR Code . . . . .                  | 33 |
| Figura 13 – Aplicação Memoriall . . . . .  | 33 |
| Figura 14 – QR Code do projeto QRio . . . . .                                    | 34 |
| Figura 15 – Comunicação via Web Services . . . . .                               | 36 |
| Figura 16 – Ciclo de vida de um processo por prototipagem . . . . .              | 39 |
| Figura 17 – Funcionamento JVM . . . . .  | 41 |
| Figura 18 – IDE - Android Studio . . . . .                                       | 42 |
| Figura 19 – Funcionamento SQLiteStudio . . . . .                                 | 44 |
| Figura 20 – Diagrama de Caso de Uso da aplicação . . . . .                       | 50 |
| Figura 21 – Diagrama de Classes da Aplicação . . . . .                           | 51 |
| Figura 22 – Diagrama de Sequência Acesso ao Mapa . . . . .                       | 52 |
| Figura 23 – Diagrama de Sequência Acesso a um Patrimônio . . . . .               | 52 |
| Figura 24 – Diagrama de Sequência Acesso ao Leitor QR Code . . . . .             | 53 |
| Figura 25 – Protótipo Tela Inicial . . . . .                                     | 54 |
| Figura 26 – Ilustração da Base de Dados gerado via <i>SQLiteStudio</i> . . . . . | 55 |
| Figura 27 – Acesso a base de dados. . . . .                                      | 56 |
| Figura 28 – Patrimônios mais visitados pelos turistas. . . . .                   | 57 |
| Figura 29 – Antes e Depois da personalização do mapa. . . . .                    | 58 |
| Figura 30 – Método para Navegação . . . . .                                      | 59 |
| Figura 31 – Classe do Leitor QRCode . . . . .                                    | 60 |
| Figura 32 – Leitor <i>QR Code</i> ZXing . . . . .                                | 60 |
| Figura 33 – Gerador <i>QR Code</i> . . . . .                                     | 61 |
| Figura 34 – Gerador <i>Classe Navigation View</i> . . . . .                      | 62 |
| Figura 35 – Modelo de <i>Navigation View</i> . . . . .                           | 62 |

|   |    |
|---|----|
| Figura 36 – Tela de Apresentação da Aplicação . . . . . | 64 |
| Figura 37 – Tela Principal da Aplicação . . . . .       | 65 |
| Figura 38 – Tela de Navegação . . . . .                 | 66 |
| Figura 39 – Tela do Leitor QR Code . . . . .            | 66 |
| Figura 40 – Tela de Listagem dos Patrimônios . . . . .  | 67 |
| Figura 41 – Tela de Detalhes dos Patrimônios . . . . .  | 68 |

# LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1 – Requisitos Funcionais da Aplicação . . . . .     | 47 |
| Tabela 2 – Requisitos não-funcionais da Aplicação . . . . . | 48 |

# LISTA DE ABREVIATURAS E SIGLAS

|       |  |
|-------|--|
| API   | Application Programming Interface                        |
| DLL   | Dynamic-Link Library                                     |
| HTC   | High-Tech Computer Corporation                           |
| HTML  | HyperText Markup Language                                |
| IDE   | Integrated Development Environment                       |
| IPHAN | Instituto do Patrimônio Histórico e Artístico Nacional   |
| JDK   | Java Development Kit                                     |
| JVM   | Java Virtual Machine                                     |
| OHA   | Open Handset Alliance                                    |
| OS    | Operating System   |
| QR    | Quick Responsive   |
| SDK   | Software Development Kit                                 |
| SQL   | Structured Query Language                                |
| UFVJM | Universidade Federal dos Vales do Jequitinhonha e Mucuri |
| UML   | Unified Modeling Language                                |
| URI   | Uniform Resource Identifier                              |
| URL   | Uniform Resource Locator                                 |
| WI-FI | Wireless Fidelity  |
| W3C   | World Wide Web Consortium                                |
| WWW   | World Wide Web   |
| XML   | Extensible Markup Language                               |

# SUMÁRIO

|            |   |           |
|------------|---|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                                   | <b>13</b> |
| <b>1.1</b> | <b>Motivação</b>                                    | <b>14</b> |
| <b>1.2</b> | <b>Objetivos</b>                                    | <b>14</b> |
| 1.2.1      | Objetivos Específicos                               | 15        |
| <b>1.3</b> | <b>Estrutura do Trabalho</b>                        | <b>15</b> |
| <b>2</b>   | <b>REFERENCIAIS TEÓRICOS</b>                        | <b>16</b> |
| <b>2.1</b> | <b>Mobilidade e Plataforma Móveis</b>               | <b>16</b> |
| 2.1.1      | Plataformas Móveis                                  | 17        |
| 2.1.1.1    | Android   | 18        |
| 2.1.1.2    | iOS   | 18        |
| 2.1.1.3    | Windows Phone                                       | 19        |
| 2.1.1.4    | Blackberry  | 20        |
| 2.1.2      | Comparação entre Plataformas                        | 21        |
| 2.1.3      | Plataforma Android                                  | 22        |
| 2.1.3.1    | Histórico e Evolução                                | 22        |
| 2.1.3.2    | Características Gerais                              | 24        |
| 2.1.3.3    | Arquitetura Android                                 | 25        |
| 2.1.3.4    | SQLite  | 26        |
| <b>2.2</b> | <b>Serviços Baseados em Localização</b>             | <b>27</b> |
| 2.2.1      | Arquitetura   | 28        |
| 2.2.2      | Mapas   | 29        |
| <b>2.3</b> | <b>Quick Response Code</b>                          | <b>30</b> |
| 2.3.1      | Exemplos de Aplicação                               | 32        |
| <b>2.4</b> | <b>Interface de Programação de Aplicativos(API)</b> | <b>34</b> |
| 2.4.1      | Web Services  | 35        |
| <b>2.5</b> | <b>Processos de Software</b>                        | <b>37</b> |
| 2.5.1      | Modelo por Prototipagem                             | 38        |
| <b>3</b>   | <b>FERRAMENTAS UTILIZADAS</b>                       | <b>40</b> |
| <b>3.1</b> | <b>Equipamentos Utilizados</b>                      | <b>40</b> |
| <b>3.2</b> | <b>Linguagens Utilizadas</b>                        | <b>40</b> |
| 3.2.1      | JAVA  | 40        |
| 3.2.2      | XML   | 41        |
| <b>3.3</b> | <b>Android Studio</b>                               | <b>42</b> |

|            |   |           |
|------------|---|-----------|
| <b>3.4</b> | <b>SQLiteStudio</b> . . . . .                 | <b>43</b> |
| <b>4</b>   | <b>DESENVOLVIMENTO DA APLICAÇÃO</b> . . . . . | <b>45</b> |
| <b>4.1</b> | <b>Desenvolvimento</b> . . . . .              | <b>45</b> |
| 4.1.1      | Identificação do Problema . . . . .           | 45        |
| 4.1.2      | Características do Sistema . . . . .          | 46        |
| 4.1.2.1    | Requisitos do Sistema . . . . .               | 46        |
| 4.1.2.2    | Requisitos Funcionais . . . . .               | 47        |
| 4.1.2.3    | Requisitos Não Funcionais . . . . .           | 47        |
| 4.1.2.4    | Limitações Identificadas . . . . .            | 48        |
| 4.1.2.5    | Soluções Propostas . . . . .                  | 48        |
| <b>4.2</b> | <b>Modelagem</b> . . . . .                    | <b>49</b> |
| 4.2.1      | Diagramas de Caso de Uso . . . . .            | 49        |
| 4.2.2      | Diagrama de Classes . . . . .                 | 50        |
| 4.2.3      | Diagrama de Sequência . . . . .               | 51        |
| 4.2.4      | Protótipos de Tela . . . . .                  | 53        |
| <b>4.3</b> | <b>Construção</b> . . . . .                   | <b>55</b> |
| 4.3.1      | Banco de Dados . . . . .                      | 55        |
| 4.3.2      | Levantamento dos Patrimônios . . . . .        | 56        |
| 4.3.3      | Mapa e Navegação . . . . .                    | 58        |
| 4.3.4      | Leitor QR Code . . . . .                      | 59        |
| 4.3.4.1    | Geração dos Códigos . . . . .                 | 61        |
| 4.3.5      | Menu Principal . . . . .                      | 61        |
| <b>5</b>   | <b>RESULTADOS OBTIDOS</b> . . . . .           | <b>64</b> |
| <b>5.1</b> | <b>Apresentação do Sistema</b> . . . . .      | <b>64</b> |
| 5.1.1      | Tela de Apresentação . . . . .                | 64        |
| 5.1.2      | Tela Principal . . . . .                      | 65        |
| 5.1.3      | Tela Navegação . . . . .                      | 65        |
| 5.1.4      | Tela Leitor QR Code . . . . .                 | 66        |
| 5.1.5      | Tela Listagem Patrimônios . . . . .           | 67        |
| 5.1.6      | Tela Patrimônio . . . . .                     | 67        |
| <b>6</b>   | <b>CONCLUSÃO</b> . . . . .                    | <b>69</b> |
| <b>6.1</b> | <b>Trabalhos Futuros</b> . . . . .            | <b>69</b> |
|            | <b>REFERÊNCIAS</b> . . . . .                  | <b>71</b> |

# 1 INTRODUÇÃO

Vivemos na era da informação em que a tecnologia está cada vez mais próxima das pessoas e por consequência, tornamo-nos cada vez mais dependentes dela. O avanço tecnológico ocasionou na invenção de vários instrumentos como, por exemplo, os microprocessadores, a rede de computadores, o computador pessoal, entre outros. Assim, podemos constatar que a Era Industrial foi deixada para trás, dando início a um processo de inovação sem limites, no qual não temos previsibilidade de seus impactos e avanços.

A utilização de computadores em prol da sociedade teve início na realização das tarefas e atividades que antes eram desenvolvidas de modo mecânico, por exemplo, arquivar documentos e realizar saldos financeiros. Essa utilização bem-sucedida impulsionou que os computadores fossem alvo de uma evolução desenfreada, sendo cada vez mais acessíveis, potentes e compactos. Hoje esses computadores estão em nossas mãos - são *smartphones*, *tablets*, *notebooks*, *netbooks* - e permitem desempenharmos tarefas de formas simples e prática. Esses dispositivos estão imersos na sociedade de maneira que alteram como as pessoas trabalham, divertem, interagem e socializam. (MADUREIRA, 2011).

Atualmente, os dispositivos móveis são as principais opções para acompanhar o desenvolvimento acelerado em um mundo globalizado, tendo em vista a velocidade que as informações são transmitidas e acessadas. A popularização, especialmente dos *smartphones*, vem modificando a forma de atuação de diversos setores como o turismo, lazer, financeiro, serviços, transporte e outros. Esses dispositivos portam aplicações cada vez mais robustas e com inúmeras funcionalidades, mais viáveis e atrativas aos usuários. Essa perspectiva é o que impulsiona o surgimento desenfreado de aplicativos que buscam trazer benefícios aos diferentes setores (MADUREIRA, 2011).

Propõe-se neste trabalho aliar o crescimento de aplicações móveis em prol do patrimônio histórico da cidade de Diamantina - Minas Gerais, seguindo a tendência de vários lugares pelo mundo onde os aplicativos móveis já são utilizados como forma de aproximação entre patrimônio, história e pessoas. Segundo Abrantes(2014):

O uso das tecnologias digitais nas práticas de preservação do patrimônio cultural, sobretudo neste momento histórico – que pode ser considerado um período de transição da sociedade, marcado por intensas transformações, resultado do intenso progresso tecnológico, em especial da comunicação–, mostra-se como fator básico para condução e ampliação do acesso à informação e aos bens culturais, e para a conservação destes.

A cidade dispõe de belos monumentos históricos e belezas naturais, mas não tira proveito dos benefícios da evolução tecnológica e falha no suporte aos turistas. Uma pesquisa realizada pelo curso de turismo da Universidade Federal dos Vales do Jequitinhonha e Mucuri

(UFVJM) mostrou que o acesso as informações turísticas são uma das maiores reivindicações dos visitantes e devem ser trabalhadas e melhoradas (UFVJM, 2012).

O desenvolvimento da aplicação apoiará-se em tecnologias como o *Quick-Responsive Code* e tecnologias de localização para apresentar o conteúdo da melhor maneira possível. O uso dessas tecnologias tem o intuito de fornecer uma interatividade e atratividade adequada aos visitantes. Além disso, espera-se que o aplicativo torne-se colaborativo para o enriquecimento do conteúdo disponível aos usuários.

Assim, o presente trabalho busca desenvolver uma aplicação que possa servir de ferramenta útil a qualquer um que queira conhecer mais sobre o patrimônio cultural de Diamantina. Julga-se necessário ressaltar que, a aplicação não pretende substituir os meios informacionais, mas sim, servir como uma ferramenta de auxílio na visitação dos patrimônios. O aplicativo surgiu a partir de um projeto de extensão que contou com o apoio de alunos e professores de diferentes departamentos da UFVJM<sup>1</sup>.

## 1.1 Motivação

Um dos maiores objetivos de uma universidade é possibilitar que o conhecimento adquirido pelos alunos seja aplicado de forma benéfica no âmbito social em que a instituição está inserida. Desse modo, buscou-se uma ideia que permitisse a utilização da tecnologia em prol de Diamantina. Através da leitura de algumas pesquisas desenvolvidas pela universidade, é possível notar que a cidade carece de tecnologias adequadas, e seu principal recurso, o patrimônio histórico, não dispõe do auxílio das ferramentas digitais.

Nesse contexto, foi identificado a possibilidade de criação de uma ferramenta móvel para colaborar com o setor turístico da cidade. O desenvolvimento de aplicações móveis é destaque no mercado de tecnologia de informação que vem provendo ferramentas úteis a diversos setores. Assim, a criação da aplicação permite dar um retorno ao município e também servirá como um aprendizado de grande valia em preparação ao mercado de trabalho.

## 1.2 Objetivos

O objetivo principal deste trabalho é planejar e desenvolver uma aplicação para dispositivos móveis que exercerá o papel de um guia turístico virtual para a cidade de Diamantina. A aplicação utilizará de tecnologias auxiliares, como o *Quick-Responsive Code* e serviços baseados em localização, para prover informações de forma prática e atrativa.

---

<sup>1</sup> O projeto faz parte do Programa Institucional de Bolsas de Extensão Universitária (PIBEX) da UFVJM e iniciou suas atividades no primeiro período de 2016



### 1.2.1 Objetivos Específicos

- a) Analisar as tecnologias necessárias para desenvolver o aplicativo;
- b) estudar a linguagem a ser utilizada para a implementação do projeto;
- c) realizar um estudo sobre as bibliotecas relevantes para o desenvolvimento;
- d) levantar os requisitos do aplicativo;
- e) modelar o aplicativo para dispositivo móvel;
- f) identificar os patrimônios mais relevantes para a aplicação;
- g) aplicar a utilização do *QR Code* como facilitador ao acesso à informação;
- h) criar um acervo informativo com dados escritos e audiovisuais que sirva como base de dados para o aplicativo;
- i) desenvolver uma aplicação que contribua com a sociedade em termos de visitação dos patrimônios.

## 1.3 Estrutura do Trabalho

No Capítulo 2 serão abordados definições e conceitos essenciais para o projeto. Analisa-se, também, as tecnologias que adotaremos neste trabalho.

No Capítulo 3 serão descritas as principais ferramentas utilizadas para elaboração do trabalho.

No Capítulo 4 serão detalhados o processo de desenvolvimento da aplicação, desde o planejamento até a implementação.

No Capítulo 5 serão mostrados os resultados obtidos como as telas disponíveis no aplicativo.

No Capítulo 6 o trabalho será concluído com as devidas considerações e serão apresentadas propostas para trabalhos futuros.

## 2 REFERENCIAIS TEÓRICOS

Este capítulo é destinado a expor os conceitos fundamentais sobre as tecnologias que foram essenciais para a construção do trabalho.

Na seção sobre mobilidade e plataforma móveis serão apresentados definições, conceitos e um breve histórico sobre a evolução e a atual situação dessa tecnologia. Subsequente, constará uma breve análise das diferentes plataformas, e posteriormente uma comparação entre elas.

Ao final da seção, conceitos úteis sobre a plataforma que fora utilizada neste trabalho. Na seção Serviços Baseados e Localização (LBS) serão abordadas concepções gerais sobre esses serviços. Primeiramente, entenderemos os aspectos principais dessa tecnologia e posteriormente a exemplificação de aplicações que utilizam esses serviços.

Na seção QR Code serão apresentados os principais paradigmas dessa tecnologia e exemplos de como esta é utilizada e aproveitada nos dias de hoje. A última seção descreve o que são os *web services* e a importância desses para a construção deste trabalho.

### 2.1 Mobilidade e Plataforma Móveis

Entende-se que mobilidade tem por definição aquilo que é móvel ou capaz de se movimentar. Aplicando este conceito a tecnologia, pode-se compreender que a mobilidade tecnológica é capacidade de levar um dispositivo a qualquer lugar e que o mesmo possa ser utilizado quando em movimento (SACCOL; REINHARD, 2007). Os dispositivos que possuem essa tecnologia são hoje altamente utilizados pela sociedade; segundo a empresa TeleoGraphy's em um levantamento realizado em 2015, constatou que existem 7 bilhões de linhas móveis ativas, número que corresponde quase a população mundial.

Esse sucesso é o que justifica a difundida utilização de dispositivos móveis em quase todos os setores atualmente. Os aparelhos chegam até a sociedade na forma de *smartphones*, *tablets*, *consoles* que possuem a capacidade de carregar inúmeras funcionalidades como: câmera, *MP3 player* e, através da rede acessam e-mails, sites, jogos, *software* e redes sociais. É por meio dessas funcionalidades que a informação flui cada vez mais rápido e fácil.

Os dispositivos portáteis possuem um grande êxito global devido as suas potenciais características. Com hardware cada vez mais potente e sem acrescentar complexidade, conseguem manter a simplicidade de uso e permitem que usuários com capacidade educacional limitada desfrutem dos benefícios da mobilidade tecnológica. Além disso, a facilidade de instalação da estrutura de redes sem fio em relação as estruturas de redes convencionais, permitiu a fácil imersão dos dispositivos móveis em mercados emergentes, contribuindo ainda mais para a

popularização da tecnologia móvel (HARVARD, 2016).

O resultado dessa propagação é um número elevado de aplicações que modificam a forma de como a sociedade trabalha e comunica. Essa influencia afeta diretamente setores e mercados em todo mundo, modificando a forma com que esses são desenvolvidos. No Brasil, os aplicativos móveis já são o principal canal de transações para bancos como; Bradesco e Banco do Brasil, sendo as aplicações responsáveis por mais de 30% das transações (GOMES, 2016). Os benefícios por oferecer a mobilidade são altamente visados pelas organizações. Um levantamento realizado em 2011 pela Forrsights Networks And Telecommunications, mostrou que 64% das empresas Europeias e Norte Americanas possuem como principal prioridade dar um maior suporte a mobilidade. Esses números refletem que a busca das organizações por benefícios gerados pela mobilidade têm se tornado um fator estratégico crucial para o crescimento dessas instituições.

### 2.1.1 Plataformas Móveis

Plataforma móvel tem como principal função prover o acesso do usuário as funcionalidades dos dispositivos móveis. Para que os aparelhos possam usufruir de sistemas e serviços, esses necessitam de uma plataforma (FLING, 2009). Com a evolução dos dispositivos móveis diversas empresas entraram no desenvolvimento de plataformas, levando ao cenário que estamos inseridos atualmente, diferentes sistemas cada um com seus aspectos positivos e negativos.

A escolha de determinada plataforma para desenvolvimento de um sistema é uma parte fundamental do projeto, pois cada uma possui suas peculiaridades que oferece diferentes vantagens para construção de aplicações. Essas plataformas, também são nada mais que *software* e possuem a mesma divisão por licenças, próprias e código aberto. A figura 1 exibe dados divulgado em 2016 sobre os aparelhos ativados no mesmo ano:

Figura 1 – Números de Aparelho por Plataforma

| Operating System | 2Q16<br>Units    | 2Q16 Market<br>Share (%) | 2Q15<br>Units    | 2Q15 Market<br>Share (%) |
|------------------|------------------|--------------------------|------------------|--------------------------|
| Android          | 296,912.8        | 86.2                     | 271,647.0        | 82.2                     |
| iOS              | 44,395.0         | 12.9                     | 48,085.5         | 14.6                     |
| Windows          | 1,971.0          | 0.6                      | 8,198.2          | 2.5                      |
| Blackberry       | 400.4            | 0.1                      | 1,153.2          | 0.3                      |
| Others           | 680.6            | 0.2                      | 1,229.0          | 0.4                      |
| <b>Total</b>     | <b>344,359.7</b> | <b>100.0</b>             | <b>330,312.9</b> | <b>100.0</b>             |

Fonte: (GARTNER, 2015)

Os números mostram a significativa dominância da plataforma Android presente em mais de 80% dos aparelhos. Porém, em um mercado gigante como o de telefonia móvel, as plataformas com menor porcentagem de mercado são também fundamentais para a mobilidade e merecem a devida atenção. A concorrência entre essas plataformas contribui para a evolução acelerada dos sistemas e serviços. Veremos a seguir um pouco de cada uma das plataformas que aparecem como líderes de mercado.

#### 2.1.1.1 Android

Android é uma plataforma de desenvolvimento para aplicações móveis como *smarthphones* e *tablets* baseados no sistema operacional Linux. Em busca de acelerar e inovar as tecnologias móveis, a plataforma desenvolvida pela Open Handset Alliance (OHA) surgiu a partir de uma parceria liderada pela Google e contou com diversas empresas como LG, Motorola, Samsung, Toshiba e outras. A plataforma possui uma interface visual rica, GPS, suporta qualquer tipo de conexão sem fio (3G/4G/LTE, EDGE, Wi-Fi e bluetooth) e dispõe de um ambiente de desenvolvimento integrado (IDE) poderoso que utiliza a linguagem de programação Java (LECHETA, 2010).

O Android se destaca pelo crescente número de usuários superior a todos concorrentes e pelo grande arsenal de aplicativos. Por ser de código livre é um sistema altamente mütavel podendo ser modificado e adaptado sem perder seu baixo custo. Essa visibilidade aliada a política *open-source* faz com que essa plataforma esteja em constante desenvolvimento e recebendo melhorias.

Para a construção de aplicações Android não é requerido um sistema operacional específico, sendo indicado o uso da linguagem de programação *Java* e a *IDE Android Studio*. Para publicar um aplicativo na loja oficial da Google, a *Play Store*, é necessário um taxa única de US\$25. Antes de divulgados, esses aplicativos são revisados para assegurar que não comprometem as políticas estabelecidas pela empresa (ANDROID, 2016a). A implementação do modelo proposto por este trabalho foi desenvolvida na plataforma Android, logo, haverá uma seção que abordará detalhadamente todos conceitos desta plataforma.

#### 2.1.1.2 iOS

Desenvolvida pela Apple, a plataforma iOS foi criada para atender os dispositivos móveis da empresa, os conhecidos, iPod, iPhone e iPad. Diferente da plataforma Android, citada anteriormente, a Apple não licencia a plataforma para utilização em dispositivos não produzidos por ela. Quando lançada em 2007, o destaque para o sistema foi a inovações relacionadas às tecnologias de toque múltiplas adaptadas à multimídia (vídeos, imagens e músicas). A interação através de toques diretos em conjunto com o poder de mutação da aplicação no sistema permitiu que o conteúdo adequasse de maneira ideal ao usuário (COSTA; FILHO; DUARTE, 2012).

Figura 2 – iPhone exibindo três aplicações distintas.



Fonte: (THANH, 2014)

Na Figura 2 podemos visualizar o sistema tomando a forma adequada de cada aplicação, se nos dias de hoje é um fator comum termos esse comportamento em todas as plataformas, em 2007 era uma grande inovação que conquistou usuários em todas as partes do mundo. Uma limitação é que como o sistema é exclusivo da Apple suas aplicações estão disponíveis somente nas lojas da empresa. Porém, essa restrição trás um grande benefício aos níveis de segurança em geral, como a empresa controla todos aplicativos, a ocorrência de *malwares*<sup>1</sup> é bastante monitorada. A plataforma possui uma base satisfeita de usuários, favorecendo a aparição de mais desenvolvedores interessados em atender a essas demandas. É um sistema pioneiro com um importante nicho de mercado e com características poderosas que serão analisadas mais a frente (COSTA; FILHO; DUARTE, 2012).

### 2.1.1.3 Windows Phone

O Windows Phone, desenvolvida pela Microsoft, é uma versão licenciável e compactada do sistema Windows. A plataforma surgiu em 2010 como sucessora do Windows Mobile que não teve o sucesso esperado diferente da dominância da mercado que o sistema para computadores pessoais possui. A Nokia é o carro chefe na distribuição de aparelhos com a plataforma que ainda deixa a desejar em relação aos outros sistemas do mercado.

<sup>1</sup> Malware, é software nocivo ou software malicioso destinado a infiltrar-se em um sistema de computador alheio de forma ilícita, com o intuito de causar alguns danos, alterações ou roubo de informações (confidenciais ou não).

Figura 3 – Evolução do sistema móvel da Microsoft



Fonte: (THANH, 2014)

Atualmente, com a nova versão 8.1, o sistema trouxe melhorias na interação, aplicações e integrações com outros serviços da Microsoft. Futuramente, a empresa busca uma inserção maior com outros dispositivos, como o *Xbox*<sup>2</sup>, buscando uma unificação em um nível superior que a Google e a Apple já fazem com o iOS e Android (JESUS, 2014). Contudo, o sistema carece de aplicativos e adeptos ao desenvolvimento voltado ao sistema. A plataforma vem de uma crescente melhoria, porém não é possível afirmar se continuará sendo apenas mais um no mercado ou terá capacidade de se firmar ao lado do Android e iOS.

#### 2.1.1.4 Blackberry

O BlackBerry OS é um sistema desenvolvido pela BlackBerry e exclusiva para os aparelhos da empresa. A plataforma é fortemente utilizada pelo mercado corporativo, em razão de os dispositivos serem conhecidos por possuírem funções que permitem a produção de conteúdo profissional em qualquer lugar de forma segura. Após tentativas da empresa de aumentar seu espaço de mercado desenvolvendo um *smarthphone* com o sistema Android, a BlackBerry anunciou em 2016 que vai deixar de fabricar celulares e voltar toda sua atenção no desenvolvimento de *software* (COSTA; FILHO; DUARTE, 2012).

<sup>2</sup> Xbox é uma marca de consoles de videogame criada pela Microsoft

## 2.1.2 Comparação entre Plataformas

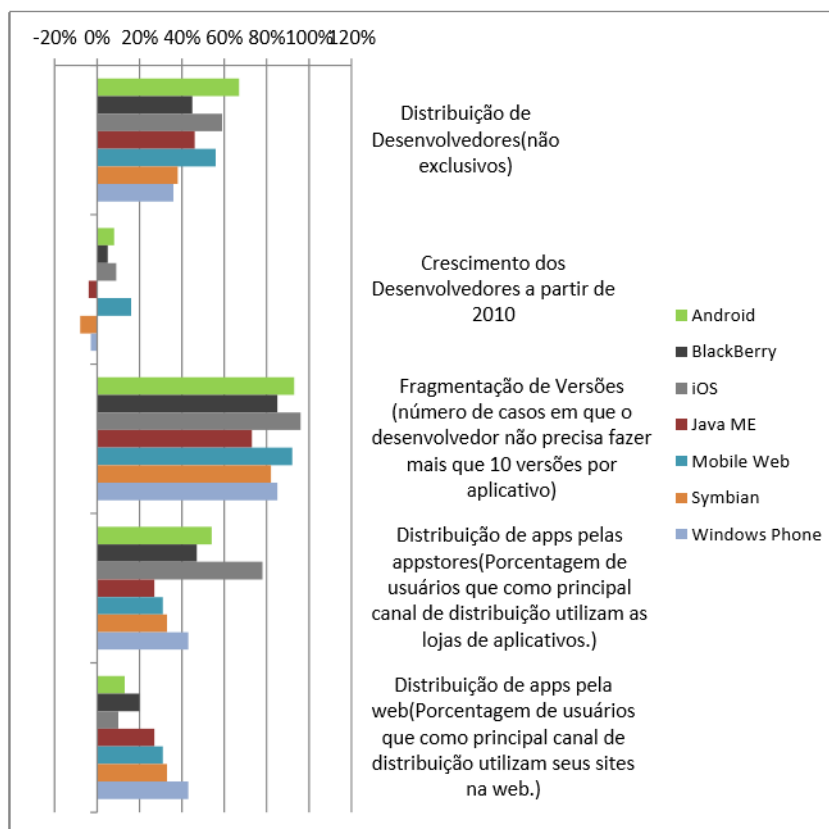
Para a comparação entre as plataformas, deve-se analisar as funcionalidades técnicas essenciais que precisam existir em um sistema operacional móvel, como também aquelas que deixam a plataforma mais robusta. Ademais, é necessário observar aspectos ligados a usabilidade desses sistemas, características que são evidenciadas diretamente pelos usuários como interface e facilidade de uso. Ao fim, considerar a viabilidade de desenvolvimento e a chance de mercado que cada plataforma oferece.

Conforme citado anteriormente, a pequena quantidade de usuários que utilizam os sistemas *Windows Phone* e *BlackBerry OS*, torna inviável que essas duas plataformas sejam escolhidas como o ambiente de desenvolvimento, tendo em vista que a aplicação proposta neste trabalho visa afetar o maior número de pessoas. Além disso, a quantidade de documentação disponível e material de suporte ao desenvolvimento é relativamente baixo se comparado ao Android o que dificulta ainda mais a adesão dessas plataformas.

O sistema iOS é projetado para uma integração de *hardware* e *software* exclusiva que proporciona uma grande vantagem em desempenho. Dispõe de uma arquitetura bem definida com diferentes núcleos que atendem todas funcionalidades técnicas necessárias com eficácia. O sistema foi pioneiro no conceito multi-toque para celulares e se destaca na usabilidade, sendo extremamente amigável ao usuário. Porém, por se tratar de uma plataforma exclusiva da Apple é necessário que utilize uma máquina da mesma empresa, enquanto as outras plataformas podem ser desenvolvidas em qualquer computador. Outro ponto é que o sistema detém uma fatia de mercado consideravelmente menor em relação ao Android e mesmo sobrando integração interna, o sistema peca em interagir com um ambiente externo (COSTA; FILHO; DUARTE, 2012).

A plataforma Android proporciona aos dispositivos uma fácil usabilidade, bom desempenho através de uma arquitetura bem definida e aplicações essenciais para um bom funcionamento. O sistema atualmente domina o mercado de dispositivo. A grande vantagem é por se tratar de uma plataforma de código aberto e livre, permitindo que cada fabricante possa realizar suas melhorias e criar o seu próprio sistema. As principais desvantagens encontradas são a carência e problemas de atualização do sistema, *malwares* na lojas de aplicativos e a incompatibilidade de aplicações entre aparelhos diferentes (COSTA; FILHO; DUARTE, 2012).

Figura 4 – Comparação entre as Plataformas



Fonte: (MOBILE, 2011)

Diante das características expostas e dos objetivos deste trabalho, o Android foi escolhido como a plataforma adequada. A adesão ao sistema torna possível um desenvolvimento robusto de baixo custo, auxiliado por uma vasta documentação e que possui potencial para atingir a maior quantidade de usuários.

### 2.1.3 Plataforma Android

Esta seção acrescenta uma série de informações sobre a plataforma escolhida neste trabalho, o Android. Primeiramente, veremos dados sobre a criação, evolução e como a plataforma está nos dias de hoje. Em seguida, entenderemos a arquitetura e as características gerais do sistema. Ao fim, uma descrição mais detalhada dos módulos utilizados pela a aplicação proposta neste trabalho.

#### 2.1.3.1 Histórico e Evolução

Em 2005, o programador Andy Rubin iniciou o desenvolvimento de um programa aberto com a finalidade de rodar em qualquer telefone. Após, tentativas falhas de encontrar investidores para o projeto, surgiu a oportunidade de mostrar o sistema para Larry Page, um



dos fundadores do Google. Com o sucesso da apresentação e de encontro com os interesses da empresa, em 2005 o Google adquiriu os direitos do sistema Android (MOON, 2008).

Dois anos depois, a Google se uniu com outros líderes da indústria de dispositivos móveis, tendo como objetivo inovar rapidamente e atender melhor as necessidades dos consumidores. Esse grupo conhecido como Open Handset Alliance existe até hoje e conta com empresas como; T-Mobile, Motorola, Samsung, Intel, Google e outras. Foi assim que em 2007 a plataforma foi liberada ao mundo sob a licença Apache 2.0<sup>3</sup>, permitindo que o sistema pudesse ser utilizado em quaisquer dispositivos sem nenhuma cobrança. Na época o mundo aguardava um dispositivo diferenciado, mas o CEO da Google, Erich Schmidt, deixou claro que a ambição da empresa era mais que um simples telefone, mas uma plataforma que atingiria vários dispositivos (GARGENTA, 2011).

Em 2008, a fabricante HTC lançou no mercado o primeiro dispositivo com Android que foi alvo de duras críticas e ficou longe de conquistar o mercado. Somente em 2009, com os lançamentos da linha Galaxy S da Samsung e o aparelho Droid da Motorola, o Android conseguiu ganhar o mercado e se igualar ao iPhone. Dados do instituto Gartner mostraram que em 2011 os aparelhos com sistema Android já representavam mais de 50% das vendas mundiais. A política de código aberto do sistema valorizou a terceirização da plataforma que fora implantada em dispositivos de diversas empresas. No ano de 2011, de acordo com a Google, o sistema estava presente em 90 dispositivos de 21 fabricantes (EXAME, 2011).

Figura 5 – HTC Dream, primeiro dispositivo comercializado com Android



Fonte: HTC

<sup>3</sup> Mais detalhes da licença pode ser encontrado em <https://www.apache.org/licenses/LICENSE-2.0>

Com o sucesso no mercado, a Google passou a liberar atualizações que cada vez mais incrementavam possibilidades e recursos ao sistema. Essas versões podem ser vistas na figura 6, a seguir. É importante para os desenvolvedores buscarem a Interface de Programação de Aplicativos(API) mais baixa que o sistema pode rodar e qual o nível de API mais presente nos dispositivos, assim, garante o funcionamento da aplicação na maior quantidade possível de aparelhos. Na figura 6 é possível identificar que menos de 2% de aparelhos utilizam a API 10 ou inferior e a maioria dos dispositivos empregam a API 15 ou superior.

Figura 6 – Evolução das Versões do Android

| Version          | Codename              | API | Distribution |
|------------------|-----------------------|-----|--------------|
| 2.2              | Froyo                 | 8   | 0.1%         |
| 2.3.3 -<br>2.3.7 | Gingerbread           | 10  | 1.7%         |
| 4.0.3 -<br>4.0.4 | Ice Cream<br>Sandwich | 15  | 1.6%         |
| 4.1.x            | Jelly Bean            | 16  | 6.0%         |
| 4.2.x            |                       | 17  | 8.3%         |
| 4.3              |                       | 18  | 2.4%         |
| 4.4              | KitKat                | 19  | 29.2%        |
| 5.0              | Lollipop              | 21  | 14.1%        |
| 5.1              |                       | 22  | 21.4%        |
| 6.0              | Marshmallow           | 23  | 15.2%        |

Fonte: (ANDROID, 2016b)

### 2.1.3.2 Características Gerais

A plataforma Interface de Programação de Aplicativos(API) dispõe de uma pilha de *software* composto por sistema operacional, *middleware* e aplicações principais. O *Android Software Development Kit(SDK)* oferece as ferramentas necessárias para o desenvolvimento de aplicações utilizando a linguagem de programação Java. As principais características são<sup>4</sup>:

- Webkit: Motor de renderização rápida utilizada por Safari, Chrome e outros;
- SQLite: Suporte completo para utilização de banco de dados SQL;
- OpenGL: Gráficos otimizados, suportando duas e três dimensões;

<sup>4</sup> Características retiradas de (ANDROID, 2016a)

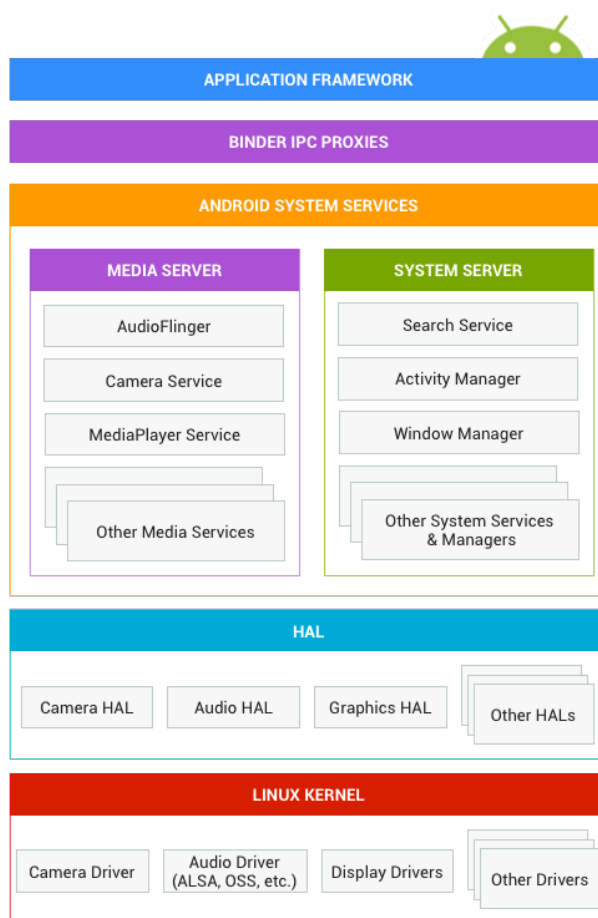
- Dalvik: Máquina virtual responsável por executar o código de aplicação nos dispositivos;
- Suporte a variados tipos de multimídia;
- OpenSSL: Camada de segurança do sistema;
- Framework de Aplicações: Fornece um rico ambiente com diversos serviços e bibliotecas que auxiliam no desenvolvimento.

Podemos observar que a plataforma permite através de um ambiente de desenvolvimento adequado a construção de aplicações robustas que satisfazem as necessidades dos usuários.

### 2.1.3.3 Arquitetura Android

A arquitetura Android é baseada nas camadas apresentadas na Figura 7 e são explicadas com mais detalhes na sequência.

Figura 7 – Arquitetura da plataforma Android



Fonte: (ANDROID, 2016a)

- A camada superior *Application Framework* é comumente utilizada pelos desenvolvedores, pois é onde as aplicações são instaladas (ANDROID, 2016a);

- A camada *Binder IPC Proxies* liga a camada superior, *Application Framework*, a *Android System Services* permitindo que as APIs de alto nível interajam com os serviços do sistema da camada inferior. Essa comunicação é escondida do desenvolvedor (ANDROID, 2016a);
- Abaixo da *Binder IPC Proxies* está localizada a *System Services* que através de módulos permite acesso ao hardware (ANDROID, 2016a). Os serviços oferecidos por essa camada foram divididos em dois grupos:
  - *System Server*: São os serviços responsáveis por gerenciar buscas, notificações e telas,
  - *Media Server*: São os serviços responsáveis pelo gerenciamento de multimídia - áudio, câmera, vídeo e outros.
- A camada *Hardware Abstraction Layer* ou HAL fica abaixo da camada de *System Services* e permite que fornecedores de hardware implementem drivers e interfaces específicos. Assim, é possível a implementação de funcionalidades sem afetar as outras camadas do sistema (ANDROID, 2016a);
- A camada inferior é a *Linux Kernel* que utiliza uma versão do *Kernel* do Linux com adições de características importantes para plataforma móveis como, por exemplo, o sistema de gerenciamento de memória mais avançado. Esses incrementos são propriamente para funcionalidades do sistema e não afetam o desenvolvimento de *drivers* (ANDROID, 2016a).

#### 2.1.3.4 SQLite

O *Android* tem integração com o SQLite, um poderoso e leve banco de dados que possibilita a utilização de uma base de dados embutida na aplicação, ou seja, o sistema pode ter acesso a banco de dados SQL sem executar um processo de gerenciamento de base de dados separado. O SQLite é uma excelente opção para aplicações menos complexas já que ele pode ser entregue juntamente com o aplicativo, dispensando complicações maiores e evitando a instalação de um servidor (FELIX, 2016).

O SQLite é escrito em C e não possui licença, sendo totalmente disponibilizado pelos autores<sup>5</sup>, dessa maneira, qualquer pessoa pode utilizar o banco de dados. Assim, é a base de dados contidas em mais aplicações pelo mundo. O banco de dados, também é uma biblioteca compacta podendo ter menos de 500 kb com todas as funcionalidades necessárias, tornando-o a opção mais viável para sistemas com quantidade de memória restritas, como o caso de dispositivos móveis. (FELIX, 2016). Algumas informações<sup>6</sup> de destaque do SQLite são:

- a) Suporte total ao SQL92;

<sup>5</sup> Site oficial [www.hwaki.com](http://www.hwaki.com).

<sup>6</sup> Informações extraídas do site oficial do SQLite, <https://www.sqlite.org/about.html>.

- b) Bilhões e bilhões de instalações;
- c) Banco de dados de único arquivo;
- d) Código aberto;
- e) Todo código fonte em um único arquivo (sqlite3.c);
- f) Tamanho máximo do BD: 140 terabytes (247 bytes);
- g) Tamanho máximo da linha: 1 gigabyte;
- h) Dispensa Configuração;
- i) Transações segura, mesmo após falha de energia;
- j) Estável e Extensivo.

As funcionalidade e interesses do SQLite se encontram com os interesses do Android e fazem desse banco de dados o ideal em aplicações para a plataforma.

## 2.2 Serviços Baseados em Localização

Os Serviços Baseados em Localização(LBS) são serviços tecnológicos que fornecem informações criadas, selecionadas ou filtradas de acordo com as localizações atuais dos usuários, outras pessoas ou objetos móveis. Os LBS podem aparecer em conjunto com serviços convencionais como telefonia, por exemplo, para realizar o roteamento baseado em localização de chamadas. O atrativo é que os usuários não precisam introduzir informações de localização manualmente e são automaticamente rastreados e identificados (KüPPER, 2005).

Segundo Correia(2004), "Serviços baseados em localização podem ser definidos como serviços que integram a posição ou localização de um dispositivo móvel com outras informações para prover uma funcionalidade de maior valor agregado ". Podemos compreender uma aplicabilidade de maior valor observando situações comuns nos dias de hoje, por exemplo, quando um celular dispara eventos e notificações de acordo com a localização do dispositivo. Outra situação é quando utilizando o GPS e uma aplicação específica é possível visualizar informações em torno da localização, como lojas, eventos e outros.

É importante ressaltar que diferente dos LBS, um serviço de localização não processa ou trata a informação. Seu dever é somente de informar ao usuário a sua localização. Os responsáveis por executar funções de alto nível através dessas informações, são os LBS (KüPPER, 2005).

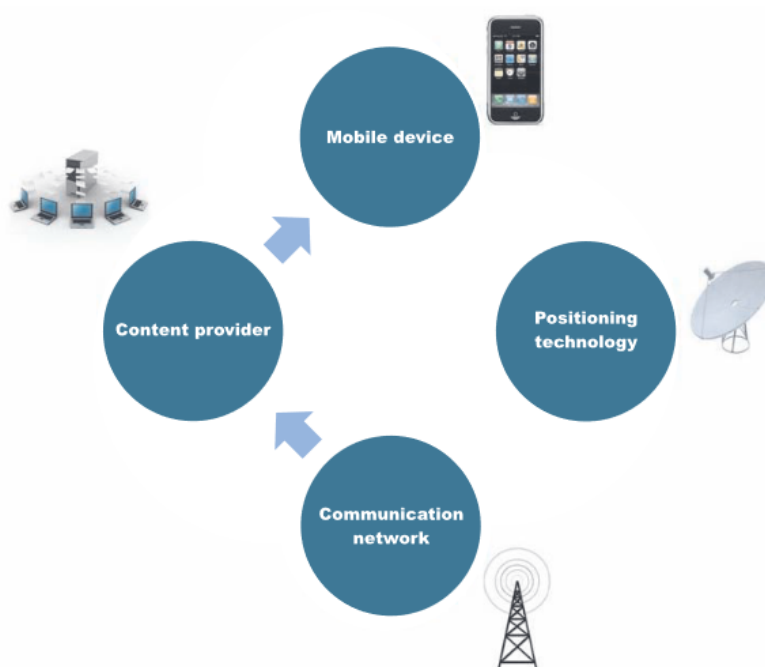
A aplicação dos LBS pode ser vista em diversas aplicações que são utilizadas diariamente pelos usuários de dispositivos móveis como, por exemplo, o Tinder; uma aplicação multi-plataforma que utiliza da localização das pessoas para encontros cruzando dados de outras aplicações e localizando as pessoas mais próximas (WILILAMS, 2015). Cada vez mais as aplicações portam informações sobre a localização dos usuários e com isso é interessante a

utilização dessas informações para um conteúdo de maior valor. Desse modo, o aplicativo a ser desenvolvido utilizará as informações de localização para facilitar a visualização do conteúdo ao usuário como já é aplicado por aplicações semelhantes do setor.

### 2.2.1 Arquitetura

Para entendermos melhor o funcionamento de aplicações LBS, serão detalhados os quatro componentes comuns a todas os serviços: dispositivo móvel, provedor de conteúdo, rede de dados e um componente de localização, conforme mostra à figura 8, a seguir.

Figura 8 – Arquitetura dos LBS



Fonte: (FERRARO; AKTIHANOGLU, 2011)

- a) Dispositivo Móvel (*Mobile Device*): Os *smartphones* tiveram um importante papel na rápida adoção dos LBS, devido as telas com tamanho adequado para visualização de mapas e a inclusão das tecnologias de localização, como o GPS. O que torna o uso desses serviços atrativos no celular é a facilidade em que uma vasta quantidade de informações podem ser acessadas com um simples toque na tela (FERRARO; AKTIHANOGLU, 2011);
- b) Provedor de Conteúdo (*Content Provider*): Um provedor de conteúdo é responsável pela informação que pode ser fornecida aos dispositivos móveis, de forma direta ou indireta. Esse componente tem sido cada vez mais importante devido ao crescimento exponencial das aplicações que necessitam desses para obter conteúdo. Um provedor não mantém todos os dados que são acessados pelo

usuários, por exemplo, um provedor de mapas disponibiliza informações pertinentes ao mapa e outros conteúdos são oferecidos por provedores de terceiros ((FERRARO; AKTIHANOGLU, 2011));

- c) Rede de Comunicação (*Communication Network*): O desenvolvedor do aplicativo LBS não possui controle sobre a rede de comunicação, mas pode administrar o tráfego dos dados utilizado pelo aplicativo, maximizando a velocidade de transferência ou minimizando a latência, bem como limitando as taxas de dados de acordo com diferentes planos que os usuários possuem ((FERRARO; AKTIHANOGLU, 2011));
- d) Componente de localização (*Positioning Technology*): Esse componente refere-se a tecnologia que permite identificar a localização do aparelho e repassar a informação para uma aplicação. Cada vez mais, esse componente tem sido escondido dos desenvolvedores, porém ainda há opção de escolha de um dos métodos de tecnologia de localização. Em dispositivos que portam mais de uma tecnologia de localização é interessante o uso híbrido dos métodos, assim, reduz as desvantagens de utilizar apenas uma tecnologia. Tem sido cada vez mais comum, determinar a localização através de componentes de *software* ou APIs que buscam estabelecer pelo menos uma localização aproximada ((FERRARO; AKTIHANOGLU, 2011)).

### 2.2.2 Mapas

Os mapas são de suma importância para os serviços baseados em localização. O provedor e a tecnologia de mapa podem ser o fator de sucesso ou incesso de uma aplicação. Por exemplo, um serviço lento, inadequado ou não-responsivo podem fazer com que o usuário migre para uma aplicação mais atrativa (FERRARO; AKTIHANOGLU, 2011). Dessa maneira, é fundamental a escolha de um serviço adequado e funcional que atenda às necessidades da aplicação e não prejudique a usabilidade da mesma.

A figura 12 mostra dois mapas de Salerno, Itália, gerados a partir de duas fontes diferentes. É nítido a diferença da qualidade, detalhe e informação que podem ser visualizadas. Uma aplicação de turismo é dependente da qualidade das informações que podem ser extraídas através do mapa, fazendo com que o provedor escolhido seja peça chave no sucesso da aplicação. Um mapa inadequado com limitação de detalhes pode levar a aplicação a não atingir o objetivo esperado.

Figura 9 – Mapa de Salerno de duas fontes distintas



Fonte: (FERRARO; AKTIHANOGLU, 2011)

Existem atualmente diversas APIs no mercado, sendo as mais conhecidas: Google Maps, MapQuest, Bing, Yahoo! e outras. Essas APIs acompanharam os avanços na última década e possuem os mais sofisticados recursos e provêm acesso completo para edição, interação e customização de acordo com cada aplicação. Dentre as diversas características, a representação gráfica e intuitiva na tela do dispositivo móvel é o fator mais importante de um serviço LBS (FERRARO; AKTIHANOGLU, 2011). Nesse trabalho, a API escolhida foi a *Google Maps* que possui um grande leque de funcionalidades, uma vasta documentação e atende bem a principal característica, citada anteriormente, sendo atualmente a mais utilizada<sup>7</sup> nas aplicações. Além disso, possui uma boa cobertura da cidade de Diamantina e permite ao usuário uma melhor visualização da cidade e dos monumentos que serão contemplados pela aplicação.

## 2.3 Quick Response Code

O *Quick Response Code* conhecido como *QR Code* é um código de barras em matriz bi-dimensional legível por leitores de código de barra QR e câmeras de dispositivos móveis. O código é representado por módulos pretos dispostos por padrões quadrados em um fundo branco, contendo informação codificada como: URL, textos ou outros tipos de dados (ROEBUCK, 2011). O termo QR, do inglês, *Quick Response* que significa resposta rápida, mostra a intenção da tecnologia de prover um objeto fácil e rápido de ser decodificado. A figura 10 é um exemplo de *QR Code*.

<sup>7</sup> Informações retiradas da página <http://compete.com>



Figura 10 – Exemplo de um QR Code



Fonte: (PANKIEWICZ, 2009)

Criado no Japão em 1994 pela empresa Denso Wave, surgiu da necessidade da empresa em criar um código que pudesse catalogar rapidamente um grande número de itens de seu inventário. Posteriormente, o *QR Code* foi padronizado como ISO/IEC18004 adaptado para realizar novas funções que aliadas a sua fácil utilização impulsionou o uso da tecnologia em diversos setores.

Diferente dos códigos de barras convencionais, de uma dimensão, o *QR Code* possui dados na direção horizontal e vertical, permitindo que ele armazene muito mais informação que um código de barras unidimensional. Um código de barras comum, contém em média não mais que 20 caracteres, enquanto um QR Code pode conter mais de 7000 caracteres. O código pode ser impresso com a utilização de qualquer combinação de cores, desde que essas sejam distintas. A figura 11 demonstra a diferença do código de barra convencional e o QR Code.

Figura 11 – Comparação de armazenamento: QR Code x Código de Barra



Fonte: Denso Wave

Neste pequeno espaço ocupado por um *Quick Responde Code*, podem ser armazenadas as seguintes quantidades de caracteres:

- Numéricos: 7.089.
- Alfa-numérico: 4.296.
- Binário (8 bits): 2.953.

- Kanji/Kana<sup>8</sup>: 1.817.

Além do poder de armazenamento superior ao código de barra comum, o QR Code possui outras vantagens no funcionamento. O código pode ser lido em qualquer direção, possuindo um padrão de detecção de posição nas suas extremidades. Outra característica fundamental é que o código dispõe de um sistema de segurança que permite a leitura precisa do código, caso contrário seria comum a interpretação errônea do conteúdo que criaria transtornos para os usuários (PANKIEWICZ, 2009).

Para a criação de um código legível existem diversos geradores disponibilizados gratuitamente na internet. A tecnologia também permite a criação de códigos mais atrativos que são amplamente utilizados por diversas empresas. Para ler um código, o usuário precisa ter um aplicativo de leitura, facilmente encontrado para todas plataformas móveis. Com a aplicação de leitura inicializada basta apontar a câmera do dispositivo para o código e as informações codificadas serão exibidas.

A praticidade e capacidade da tecnologia em conjunto com a disseminação dos dispositivos móveis, faz com que o *QR Code* seja cada vez mais utilizado em aplicações. No Brasil, já podemos ver o código em etiquetas, embalagens, propagandas, revistas e outros.

### 2.3.1 Exemplos de Aplicação

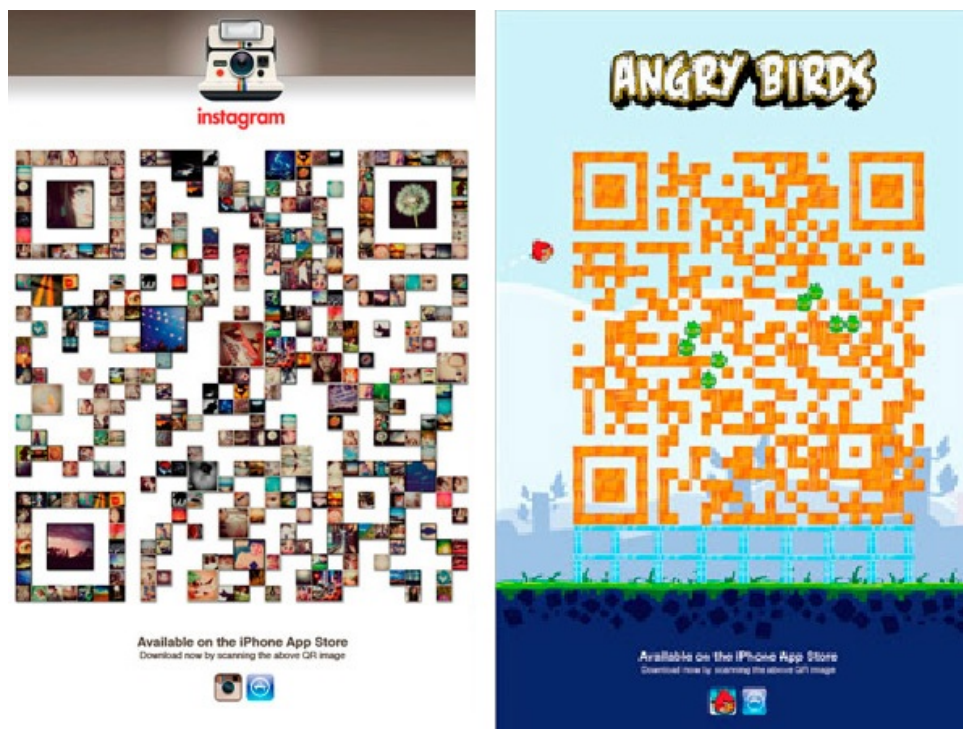
A facilidade na utilização faz com que o código seja ferramenta útil em diversos setores. Abaixo será descrito alguns exemplos de aplicações que usufruem da tecnologia em diferentes setores.

O *QR Code* começa a ter tradição no setor de marketing onde grandes empresas estão utilizando o código em etiquetas, embalagens, cartões, mensagens e outros. As características permitem que o código seja personalizado sem atrapalhar o seu funcionamento, possibilitando a criação de um código mais atrativo aos usuários. A figura 12 ilustra duas empresas utilizando o *QR Code* para divulgação das suas aplicações, ambos os códigos levam o usuário ao *download* dos aplicativos (JACOB, 2012).

---

<sup>8</sup> Caracteres utilizados no alfabeto japonês.

Figura 12 – Divulgação de duas aplicações com QR Code



Fonte: (JACOB, 2012)

Em alguns casos, os códigos são componentes chaves para o funcionamento da aplicação. É o caso do aplicativo Memoriall que permite aos usuários a partir da leitura de um *QR Code* colado na lápide do jazigo, acessar a biografia da pessoa falecida através de um dispositivo móvel. O código pode informar sobre a árvore genealógica da família, biografia, fotos, mensagens personalizadas, vídeos e outros (MEMORIALL, 2014). A figura 13 ilustra alguns códigos utilizados pela aplicação e a logo da mesma.

Figura 13 – Aplicação Memoriall



Fonte: (MEMORIALL, 2014)

Outra aplicação que utiliza da tecnologia é O QRio. Um projeto destinado a espalhar códigos *QR Code* nos principais pontos do Rio de Janeiro e foi uma das maiores motivações para o desenvolvimento deste trabalho. O projeto foi inspirado na ideia semelhante ocorrida em Portugal, quando Lisboa recebeu códigos no seus principais locais de visitação. No Rio de Janeiro, os pontos mais visitados receberam códigos que permitem a um visitante utilizar um leitor e obter informações relevantes dos locais. É uma forma interessante de divulgar informações úteis a baixo custo, sem denegrir o patrimônio e sem poluição visual. A flexibilidade na criação do QR Code, permitiu aos códigos serem expostos de forma criativa atraindo os visitantes a utiliza-los (FINCI, 2013). A figura 14 mostra um código colocado no Arpoador, Rio de Janeiro.

Figura 14 – QR Code do projeto QRio



Fonte: (FINCI, 2013)

## 2.4 Interface de Programação de Aplicativos(API)

API é o acrônimo de Application Programming Interface ou, em português, Interface de Programação de Aplicativos. É uma área compartilhada entre dois programas diferentes, permitindo que eles possam se comunicar um com o outro. Costa(2015) explica que "uma API é a forma que terceiros disponibilizam uma interface de modo que possamos consumir um determinado serviço deles sem nos preocuparmos com a implementação do mesmo". Uma boa exemplificação é o caso do Windows que possui APIs como a Win16 API, Win32 API ou Telephony API, em todas as suas versões. Desse modo, ao executar um programa que envolva algum processo do sistema operacional é provável que a aplicação faça uma conexão com alguma API do Windows.

Como esclarecido, o objetivo principal é disponibilizar recursos de uma aplicação para serem usados por outro sistema, dispensando os detalhes da implementação como também restringindo o acesso a esses recursos com regras específicas para tal (VENTURA, 2015). Dentre as diversas API's, pode-se destacar a Google Maps API que foi amplamente utilizada neste

trabalho e permite ao desenvolvedor a manipulação completa do mapa em uma aplicação. É possível observar a utilização dessa API em diversos sites, aplicativos e outras soluções.

Uma API tem três principais objetivos. Primeiro, possibilitar reuso de lógicas, já que expondo os métodos o código não precisa ser duplicado no contexto da solução. Segundo, centralizar a lógica, viabilizando num ponto único de uma solução o acesso a tal lógica. Por ultimo, viabilizar a integração com outras aplicações quando a conexão é mais "baixo nível", demandando acesso a recursos internos da aplicação (VENTURA, 2015).

Teoricamente, a API é única, mas quando falamos de desenvolvimento de *software*, uma API pode ser desenvolvida de diferentes maneiras. As mais utilizadas são:

- a) Dynamic-link library(DLL) são executáveis resultantes de classes ou conjuntos de classes que portam diferentes métodos e funções que podem ser utilizadas por terceiros;
- b) Plugins são atualmente um dos modos mais utilizados devido a maturidade da WWW. Eles provêm diferentes recursos e funcionalidades as mais variadas aplicações;
- c) Framework prove um conjunto de API's para consumo. Atualmente na web o uso dessas ferramentas está difundida entre os desenvolvedores; grandes exemplos são o JQuery e Bootstrap. Vale salientar que quando o framework é exclusivo de um sistema não trata-se de uma API, pois não há exposição de recursos nem integração com outros sistemas (VENTURA, 2015);
- d) WebAPI são API's utilizadas em soluções web. Nesse contexto falamos de Serviços Web que foram essenciais para este trabalho e serão detalhados posteriormente.

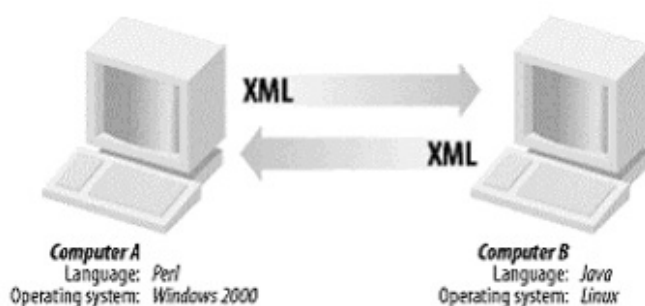
Existe uma confusão entre Web Services e API. Basicamente, a distinção é que os serviços web são uma API que obrigatoriamente utiliza a rede. Sendo assim, todos os Web Services são API, porém nem todas as API são Web Service. Por essa restrição, os serviços não podem executar todas tarefas que uma API normalmente executa (VENTURA, 2015). A próxima seção descreverá o funcionamento detalhado dos *Web Services*.

### 2.4.1 Web Services

Inicialmente, a computação era centralizada com as aplicações apenas em ambientes de grande porte. Com a evolução das redes de computadores, surgiram os Web Services como solução para uma melhor comunicação entre sistemas distribuídos, tornando-os uma importante técnica para criação e integração de sistemas. Segundo a W3C, *Web Services* é um programa que permite a interação máquina-máquina em redes. O serviço possibilita a comunicação entre duas aplicações através de uma interface bem definida, onde os dados são transitados no formato XML (ERL, 2015).

A independência de linguagem e plataforma dos *Web Services* tornou-os úteis na integração de sistemas, sendo o único requisito para um sistema se comunicar a um *Web Service* é interpretar uma mensagem XML. Na prática é o que torna viável uma aplicação desenvolvida em .Net e rodando em um servidor Microsoft, acessar um serviço feito em Java rodando em um servidor Linux. O objetivo dos *Web Services* é possibilitar a interoperabilidade entre sistemas distribuídos independente da linguagem de programação e/ou plataforma utilizada por eles, disponibilizando uma melhor interligação entre as soluções, conforme pode-se observar na figura 15 (RECKZIEGEL, 2006).

Figura 15 – Comunicação via Web Services



Fonte: (RECKZIEGEL, 2006)

Reckziegel descreve os benefícios na utilização dos *Web Services*:

Esta interligação tem como princípio facilitar os processos de negócios, proporcionando a *software* isolados passarem a funcionar de forma conjunta com os demais. Um projeto bem elaborado busca a diminuir custos, aumentar a produtividade e uma maior oportunidade de rendimento. Mas, diversos quesitos devem ser levados em conta, para evitar que esta comunicação de plataformas possa trazer prejuízos em vez de benefícios (RECKZIEGEL, 2006).

Essas vantagens são arquitetadas em uma estrutura que se baseia na comunicação entre três componentes: Provedor de Serviços, Consumidor de Serviços e Registro dos Serviços. A ligação destes componentes envolve as operações de publicação, pesquisa e ligação (KREGER, 2001). Abaixo está detalhado o papel que desempenha cada participante da comunicação.

- a) Provedor de serviços: É o responsável pela criação do *Web Services* e disponibiliza o serviço para utilização. Para que isso ocorra, o *Web Service* deve ser descrito em um formato padrão que seja compreensível para qualquer um que deseje utilizar o serviço. Também, é necessário divulgar as informações sobre o *Web Service* em um registro central que esteja disponível (KREGER, 2001);
- b) Consumidor de serviços: É qualquer um que utilize o serviço disponibilizado pelo provedor de serviço. A partir da descrição disponibilizada pelo provedor, o consumidor reconhece a funcionalidade e recupera os detalhes através da

pesquisa sobre o registro publicado. Este conhece a funcionalidade do *Web Service*, a partir da descrição divulgada pelo provedor de serviços, recuperando os seus detalhes através de uma pesquisa no registro publicado (KREGER, 2001);

- c) Registro dos serviços: Um registro de serviços é a localização central onde o provedor relaciona seus *Web Services*, e no qual um consumidor pode acessá-los (KREGER, 2001).

Desse modo, o provedor de serviços estabelece a descrição do serviço para o *Web Service* e a publica para o consumidor no registro de serviços. O consumidor utiliza a descrição do serviço publicada para se ligar ao provedor de serviços, para que possa invocar ou interagir com a implementação do *Web Service*.

Este trabalho utilizou desses serviços para acesso ao conteúdo e serviços de grande relevância ao funcionamento do aplicativo proposto.

## 2.5 Processos de Software

Para o desenvolvimento de *software* é necessário o planejamento e a adoção de técnicas adequadas que levem a um sistema de alta qualidade, contornando, assim, problemas eminentes como desempenho, prazo e custo. Essas tarefas bem estruturadas são definidas como processo de *software* e compõem um arcabouço de tarefas bem estabelecidas para construção de *software*. O processo define o contexto no qual os métodos técnicos são aplicados, os produtos de trabalho são desenvolvidos, os prazos são estabelecidos, as modificações bem gerenciadas e consequentemente, a qualidade é assegurada (PRESSMAN, 2006).

Infelizmente, o processo ainda mais utilizado é um modelo de alto risco, conhecido como "codifica-remenda". O modelo é atraente por não requerir nenhuma sofisticação técnica ou gerencial. Desenvolvedores começam a codificar a partir apenas da especificação da solução e corrigem os problemas a medida em que são descobertos (FILHO, 2009)

Pressman (2006), define um arcabouço genérico de processo que serve como base para a grande maioria dos projetos de *software*. As atividades genéricas são aplicadas no desenvolvimento de sistemas de qualquer porte, desde pequenas aplicações até grandes e complexas soluções. Os detalhes são bem diferentes em cada caso, mas as atividades são as mesmas (PRESSMAN, 2006). Estas etapas, são descritas abaixo:

- a) Comunicação: Envolve a comunicação entre os interessados para o levantamento de requisitos e outras atividade relacionadas;
- b) Planejamento: Estabelece um plano para o desenvolvimento do trabalho, descrevendo as tarefas, riscos, recursos, produtos e o prazo para a construção do projeto;
- c) Modelagem: Essa etapa inclui a criação de modelos que permitem aos interessa-

dos compreenderem os requisitos e como o projeto irá atendê-los;

- d) Construção: Essa etapa combina a criação de código e os testes necessários para correção de erros;
- e) Implantação: Fase de entrega do sistema que é avaliado pelo cliente que fornece um *feedback* a partir da avaliação.

É importante ressaltar que esse conjunto de tarefas devem ser escolhidos de maneira a se adequarem ao projeto e seus componentes, ou seja, o modelo deve ser adaptado as necessidades individuais de cada projeto. O processo escolhido para o desenvolvimento deste trabalho foi o modelo por prototipagem que será detalhado na próxima seção.

### 2.5.1 Modelo por Prototipagem

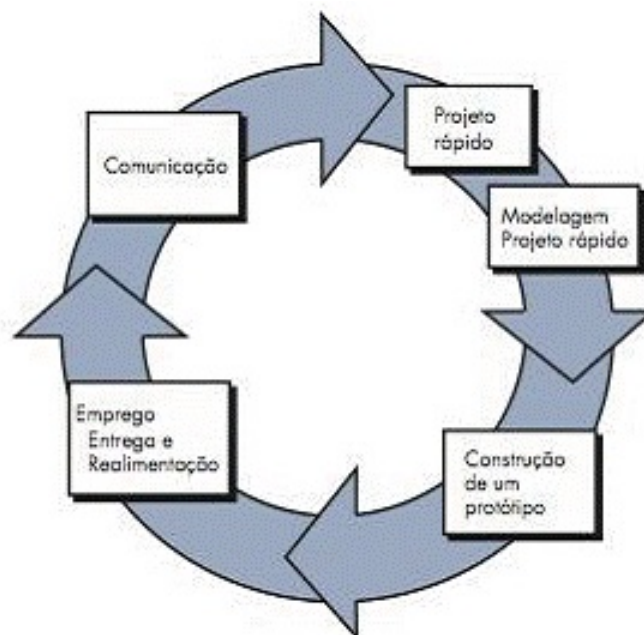
O modelo por prototipagem é conhecido como um modelo iterativo que permite aos desenvolvedores a criarem versões cada vez mais completas do sistema. É um modelo adequado para situações em que o objetivo é bem definido, porém, nem todos os detalhes de funcionamentos são conhecidos. Dessa forma, cria-se uma versão reduzida com um conjunto de funcionalidades básicas que são incrementadas com o tempo (PRESSMAN, 2006).

A prototipagem pode ser utilizada como modelo de processo independente tão como uma técnica complementar implementada dentro de outros processos. Em ambas aplicações, o paradigma de prototipagem auxilia os interessados a entenderem melhor o que deve ser construído. Casos em que o desenvolvedor está inseguro sobre a performance de algoritmo, ou de como será a usabilidade do sistema, a prototipagem proporciona uma melhor abordagem (PRESSMAN, 2006).

O processo de prototipagem inicia-se com a definição dos objetivos do *software*, identificando as necessidades principais. Em seguida, é realizado um planejamento e uma modelagem de forma rápida, levando a construção de um projeto funcional concentrado em aspectos que estarão visíveis aos usuários; o resultado é a implementação de um protótipo que será avaliado pelos utilizadores. O modelo servirá, então, como um mecanismo para identificação dos requisitos da aplicação e a partir do *feedback* fornecido é possível aos desenvolvedores analisarem as adequações necessárias para um produto mais congruente às necessidades dos usuários (PRESSMAN, 2006). A figura 16 ilustra o ciclo de vida do processo.



Figura 16 – Ciclo de vida de um processo por prototipagem



Fonte: (PRESSMAN, 2006)

O paradigma de prototipagem agrada todos envolvidos, já que o modelo oferece um produto funcional com sabor de sistema real. Contudo, o protótipo pode gerar problemas se as regras não forem bem definidas, por exemplo, o cliente vê o *software* funcionando e ignora que seja apenas uma aplicação com funcionamento precário. Logo, o sistema é mantido e o usuário desconsidera a necessidade de reconstrução do sistema com qualidade mais elevada, ou, o desenvolvedor querendo rapidamente construir o protótipo, utiliza de técnicas inadequadas que aceleram o processo de construção. Porém, quando é necessário um novo desenvolvimento as técnicas que já estão familiarizadas são mantidas, prejudicando a qualidade final do *software* (PRESSMAN, 2006).

Apesar de quase todos os requisitos estarem definidos antes do início de desenvolvimento, não se sabia como modelar a aplicação de forma a atender a todos eles, por essa razão foi escolhida a modelagem por prototipação. Com a adoção do modelo foi necessário também algumas adequações de suas etapas para melhor atender o trabalho. A falta de um cliente ou uma pessoa que identificasse os requisitos, tornou-se dispensável a fase de comunicação. Essas mudanças são permitidas e devem ser trabalhadas para que o modelo por prototipagem se adeque ao trabalho e consiga um melhor resultado.

## 3 FERRAMENTAS UTILIZADAS

No processo de desenvolvimento deste trabalho foram utilizadas diversas ferramentas de apoio, sejam elas para representações gráficas, manipulação de dados ou para codificação. Neste capítulo, serão detalhadas as principais ferramentas utilizadas e como elas contribuíram para o desenvolvimento do projeto.

### 3.1 Equipamentos Utilizados

Para a composição do trabalho utilizou-se um smartphone da marca Motorola e modelo MOTO G (Quarta Geração)<sup>9</sup> que possui um processador *Octa-Core* 1.4GHz, 2Gb de Memória RAM, tela de 5.5 polegadas com *touchscreen* capacitativa e sistema operacional Android 6.0.1. Para a codificação foi utilizado um *notebook* da marca Lenovo, modelo y50-70, que possui 8gb de Memória RAM, processador Intel Core i7-4700HQ CPU 2.40GHz, placa de video GFORCE GTX 870m e com sistema operacional Windows 10. Esses equipamentos foram fundamentais pra um desenvolvimento ágil, sem complicações e não apresentaram nenhum problema ao suportar as ferramentas e tecnologias propostas.

### 3.2 Linguagens Utilizadas

As duas subseções contidas nesta subseção descrevem as linguagens que foram utilizadas na codificação da aplicação.

#### 3.2.1 JAVA

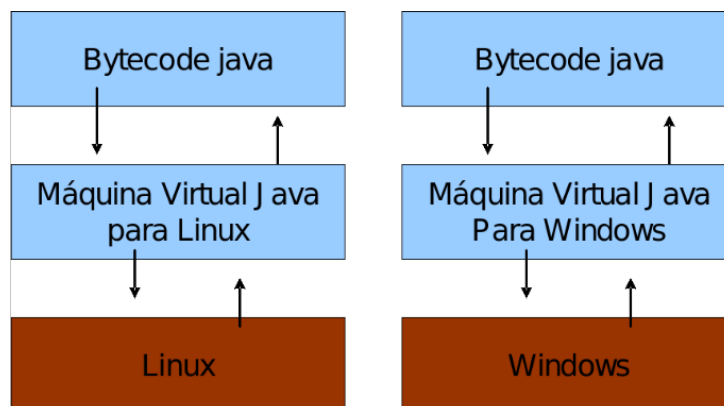
No início dos anos noventa a sociedade começava a acreditar que os dispositivos eletrônicos mudariam o modo de agir das pessoas. Foi então, que a Sun Microsystems financiou um projeto que levaria a James Gosling desenvolver uma linguagem baseada em C e C++. Inicialmente chamada de Oak, o nome Java foi sugerido após descobrirem que já existia uma linguagem de nome Oak. No começo o projeto enfrentou algumas dificuldade, mas com a explosão da World Wide Web(WWW), o Java surgiu como opção para desenvolvimento de páginas e ganhou seu espaço no mercado (DEITEL, 2010). Hoje, o Java domina o mercado de aplicações para dispositivos com mais de 2.5 bilhões de dispositivos compatíveis.

Diferente de algumas linguagens como o C na qual o código fonte é transformado para código de máquina específico para rodar em determinada plataforma, o Java utiliza o

<sup>9</sup> Todas características do aparelho pode ser visualizada em: <http://www.tudocelular.com/Motorola/fichas-tecnicas/n3205/Motorola-Moto-G4.html>

conceito de máquina virtual. Entre o sistema operacional e a aplicação, uma camada extra traduz o código de forma adequada para cada tipo de plataforma. Dessa forma, o desenvolvedor não precisa se preocupar com que sistema a aplicação está rodando. Assim, a aplicação roda sem envolvimento com sistema operacional, e sempre comunicando, dessa forma, com a camada conhecida como Java Virtual Machine(JVM) (CAELUM, 2016). A figura 17 ilustra o funcionamento da JVM.

Figura 17 – Funcionamento JVM



Fonte: (CAELUM, 2016)

A linguagem dispõe de uma quantidade vasta de bibliotecas gratuitas para quaisquer tipos de serviços, permitindo ao desenvolvedor criar aplicações sofisticadas sem nenhum custo. O uso do Java é interessante em aplicações de qualquer porte e devido a utilização da máquina virtual é ideal em aplicações multi-plataformas (CAELUM, 2016)

### 3.2.2 XML

O eXtended Markup Language(XML) foi criado pelo World Web Consortium para prover a Web uma forma simples de ultrapassar as limitações do HTML e permitir aplicações mais flexíveis. Esse sistema pode ser entendido como uma forma de organizar as informações, sendo um padrão de formatação de dados na forma de texto simples (MEDINA, 2014). A linguagem é recomendada pela World Wide Web Consortium(W3C) para geração de documentos com dados hierárquicos como textos, desenhos vetoriais ou banco de dados (MEDINA, 2014).

O XML possui uma sintaxe básica que pode utilizada para compartilhar informações entre diferentes computadores e aplicações. Em conjunto com outros padrões, é possível definir o conteúdo de um documento separadamente de seu formato, tornando simples a reutilização do código para outros propósitos. A linguagem tem o intuito de trazer flexibilidade e poder as aplicações, logo, trás benefícios para desenvolvedor e usuários, como: busca mais eficiente, aplicações flexíveis, integração de dados de fonte diferentes, fácil distribuição, fácil compreensão e outros (MEDINA, 2014).

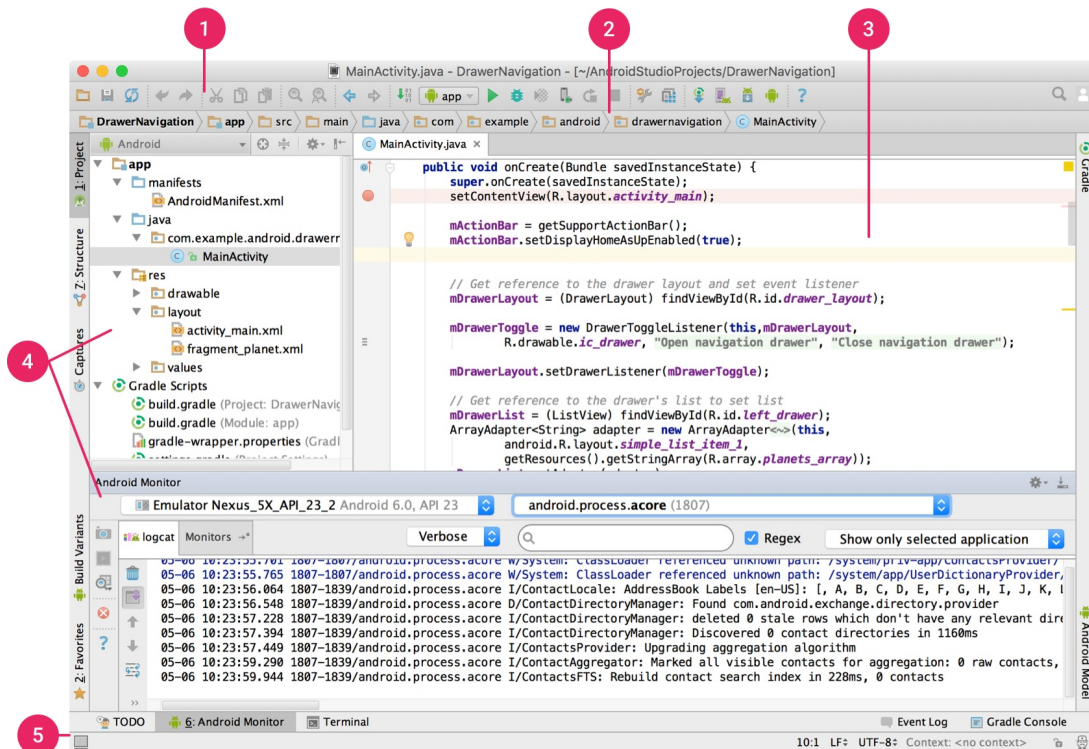
### 3.3 Android Studio

O ambiente de desenvolvimento integrado utilizado neste projeto foi a plataforma oficial para desenvolvimento de aplicativos Android, o Android Studio. O ambiente é baseado no IntelliJ IDEA, oferecendo ferramentas avançadas e muitos recursos que incrementam a produtividade do processo de desenvolvimento (ANDROID, 2016b). Alguns recursos de destaque, são:

- a) Emulador ágil com muitos recursos;
- b) Ambiente unificado que permite o desenvolvimento para todos dispositivos Android;
- c) Instant Run, que permite enviar atualizações a aplicações sem compilar uma nova aplicação;
- d) Ferramentas de verificação de código suspeito que detecta problemas rotineiros de desenvolvimento;
- e) Compatibilidade com C++, NDK e Google Cloud Platform.

Por se tratar da IDE oficial do Android com funcionalidades robustas e atualizações frequentes que agregam mais opções para o desenvolvimento a plataforma foi escolhida como o ambiente para a implementação do projeto. A seguir será feita uma apresentação breve da IDE.

Figura 18 – IDE - Android Studio



Fonte: (ANDROID, 2016b)

A enumeração corresponde as diversas áreas lógicas na janela principal da plataforma conjuntamente com suas descrições.

- 1) Barra de ferramentas: Permite a execução de várias ações como a execução de aplicativos e a inicialização de ferramentas;
- 2) Barra de navegação: Auxilia na navegação do projeto, abertura de arquivos e oferece uma visualização mais compacta da estrutura do projeto;
- 3) Janela do editor: Local de criação e edição, a depender do tipo de arquivo o editor muda adequadamente a cada tipo de arquivo;
- 4) Janelas de ferramenta: Permitem o acesso às tarefas como gerenciamento de projetos, pesquisa, controle de versões e outras;
- 5) Barra de Status: Exibe a situação do projeto e da IDE, e demais avisos ou mensagens.

O ambiente é distribuído gratuitamente e pode ser baixado no site oficial da plataforma<sup>10</sup>.

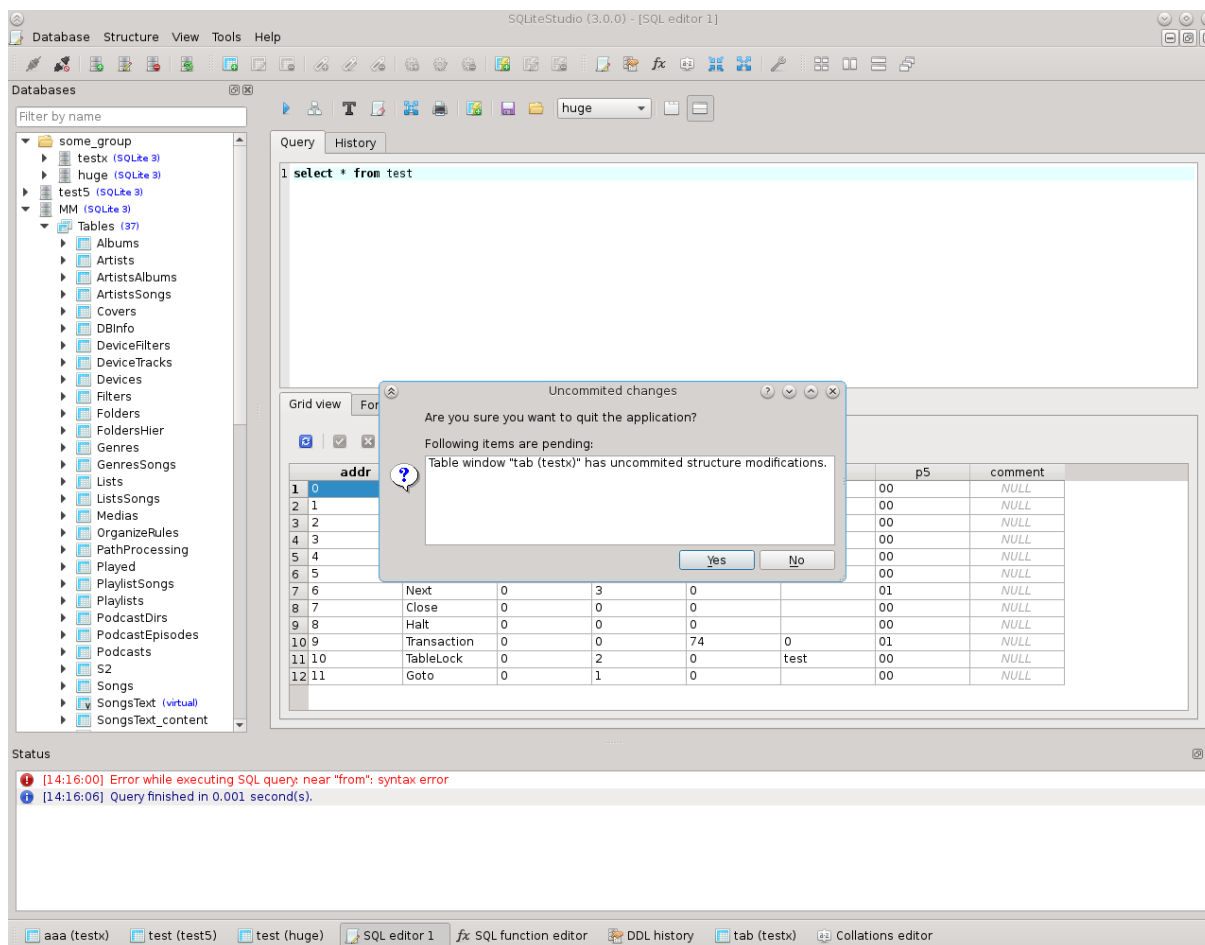
## 3.4 SQLiteStudio

O SQLiteStudio é um gerenciador de banco de dados, intuitivo e portátil, dispensando instalação, pois funciona com a execução de um único arquivo. É uma aplicação poderosa, leve e rápida que suporta as diferentes versões de SQLite e também é uma solução multiplataforma. O gerenciador permite a exportação e importação de dados em diversos formatos (CSV,SQL,XML,PDF e outros) e pode ser configurado ao gosto do usuário (SOUZA, 2013).

Diante dessas características e por ser um sistema de código livre e aberto, o gerenciador é uma excelente opção no auxílio ao gerenciamento de banco de dados SQLite. A figura 19 abaixo exibe o funcionamento da aplicação.

<sup>10</sup> Disponível em : <https://developer.android.com/studio/index.html>

Figura 19 – Funcionamento SQLiteStudio



Fonte: Imagem retirada do site oficial <https://sqlitestudio.pl>

# 4 DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo será descrito as etapas necessárias para o desenvolvimento da aplicação. O aplicativo foi construído a partir de atividades estabelecidas pelo processo de prototipagem que será detalhado na primeira seção. As demais partes desse capítulo estão descritas como foram realizadas as atividades propostas que foram guiadas pelo modelo de processo adotado.

## 4.1 Desenvolvimento

Nesta seção está descrita detalhadamente as atividades do arcabouço do processo de modelagem por prototipagem. Vale salientar que adequações foram necessárias para tornar a metodologia viável ao trabalho. Primeiro, será especificado o problema a ser solucionado - representando a fase de comunicação - e o planejamento da solução, contendo requisitos, limitações e soluções propostas. Segundo, será mostrado a modelagem que serviu de base a codificação da aplicação. Após a etapa de modelagem será descrito as principais implementações realizadas na fase de construção. Ao fim, uma análise do primeiro protótipo e as melhorias realizadas a partir dos pontos levantados.

### 4.1.1 Identificação do Problema

Segundo Maynard(2012), Diamantina possui uma demanda turística elevada que precisa ser atendida adequadamente através de iniciativas que atendam as tendências e exigências do visitante. A cidade oferece diversos patrimônios e roteiros, porém não consegue receber os visitantes de maneira eficaz. É o que mostra a pesquisa realizada em 2009, pela SECTUR<sup>11</sup> que identificou pouca informação, ausência de um guia ou monitor, as principais queixas dos turistas.

Dessa forma, através do estudo sobre LBS e dispositivos móveis foi possível observar a possibilidade da criação de uma aplicação que suprisse à falta de informação ao visitante. A capacidade de funcionalidades aliada ao crescimento na utilização dos aparelhos móveis faz com que uma aplicação tenha potencial para minimizar esses problemas. Além disso, Santos e Kuazaqui(2009) identificou que Diamantina precisa de uma estratégia de marketing e roteiros que ocupem o tempo do turista, fatores que a favorecem ainda mais o desenvolvimento da aplicação, já que esta pode desempenhar tais funções. Desse modo, foi definido o seguinte problema: disponibilizar ao usuário portador de um dispositivo móvel, informações sobre os

<sup>11</sup> Secretaria Municipal de Cultura, Patrimônio e Turismo de Diamantina

principais patrimônios históricos de Diamantinas com ferramentas que auxiliem na visitação destes.

## 4.1.2 Características do Sistema

No desenvolvimento de projetos a sempre limitações e empecilhos que podem impactar negativamente no produto final. Portanto, faz-se necessário ao desenvolvedor identificar e buscar soluções que minimizem o impacto. No desenvolvimento para dispositivos móveis algumas limitações são rotineiras e aparecem na grande maioria das aplicações, sendo o tratamento dessas de suma importância para a aplicação. A seguir estão descritas as limitações identificadas e as soluções buscadas para tentar reduzir os possíveis obstáculos que poderão surgir.

### 4.1.2.1 Requisitos do Sistema

A identificação e o gerenciamento dos requisitos é de primordial importância para o sucesso do projeto e a produção de uma aplicação de qualidade. Independente do tamanho, os sistemas podem se beneficiarem da atenção dada aos requisitos. Esses requisitos, definem os serviços que a aplicação deve oferecer (CARVALHO, TAVARES, 2002).

Na primeira etapa da modelagem é fundamental para o desenvolvimento de um sistema levantar as necessidades dos interessados, com o intuito de possibilitar a identificação dos requisitos que satisfaçam essas necessidades. As premissas dessa aplicação segue a estrutura de requisitos funcionais e não-funcionais descritos a seguir:

- **Requisitos Funcionais:** Descrevem os serviços que o sistema deve fornecer e como o sistema deve se comportar. Em alguns casos, também explicitam o que o sistema não deve fazer. Em suma, os requisitos funcionais de um sistema descrevem o que ele deve fazer (SOMMERVILLE, 2011);
- **Requisitos não funcionais:** Descrevem as restrições aos funcionalidades e serviços oferecidos pelo sistema, como restrições de timing, restrições no processo de desenvolvimento e outras impostas por normas. Ao contrário de características individuais, normalmente, os requisitos não funcionais aplicam-se ao sistema como um todo (SOMMERVILLE, 2011).

Além disso, os requisitos serão divididos quanto ao seu grau de importância e contribuição ao objetivo da aplicação. Para exemplificar as categorias, utilizaremos como exemplo um aplicativo fictício de pedido de pizza. As categorias são descritas a seguir:

- **Obrigatório:** Requisitos de maior prioridade que devem ser atendidos para que a aplicação possua o funcionamento básico desejado, por exemplo, a tela que o usuário preenche os dados para a entrega de um pedido;
- **Importante:** São os requisitos que se atendidos agregam valor a aplicação e ajudam a cumprir os objetivos desejados. Todavia, caso não atendido, não pre-



judicam as funcionalidades básicas da aplicação. Em um aplicativo de pedidos podemos considerar a tela de rastreamento de pedidos, que disponibiliza uma informação útil ao cliente, mas não afeta o funcionamento da aplicação;

- Desejável: São requisitos que acrescentaria funcionalidades pouco expressivas ou que não possuem relação direta com o objetivo da aplicação. Por exemplo, disponibilizar os valores nutricionais dos itens disponíveis.

#### 4.1.2.2 Requisitos Funcionais

Na tabela 1 serão apresentados os requisitos funcionais:

| Identificação | Descrição  | Categoria   |
|---------------|--|-------------|
| RF-01         | Disponibilizar uma lista com os principais patrimônios de Diamantina.                                    | Obrigatório |
| RF-02         | Permitir a visualização e interação no mapa dos patrimônios contidos na aplicação.                       | Obrigatório |
| RF-03         | Disponibilizar informações detalhadas de cada patrimônio, como histórico, imagens e dados de tombamento. | Obrigatório |
| RF-04         | Disponibilizar um leitor de QR Code adequado dentro da aplicação.  | Obrigatório |
| RF-05         | Permitir o usuário busca por texto de um patrimônio específico.  | Obrigatório |
| RF-06         | Disponibilizar a visualização e o uso da localização atual do usuário.                                   | Obrigatório |
| RF-07         | Permitir o usuário visualizar os patrimônios por divisão de categoria.                                   | Importante  |
| RF-08         | Disponibilizar informações climáticas da cidade de Diamantina.   | Desejável   |
| RF-09         | Disponibilizar as informações em outras línguas.   | Desejável   |
| RF-10         | Enviar notificações de patrimônios próximos ao usuário.  | Desejável   |

Tabela 1 – Requisitos Funcionais da Aplicação

#### 4.1.2.3 Requisitos Não Funcionais

Na tabela 2 serão apresentados os requisitos não-funcionais:

| Identificação | Descrição  | Categoria   |
|---------------|--|-------------|
| RNF-01        | A aplicação deve fornecer uma interface intuitiva e de fácil usabilidade ao usuário    | Obrigatório |
| RNF-02        | A aplicação deve usar os recursos adequadamente, sem exceder ou desperdiçar recursos   | Obrigatório |
| RNF-03        | O sistema deve ter um desempenho eficiente sem deixar o usuário demasiadamente ocioso  | Obrigatório |
| RNF-04        | A aplicação deve tratar os erros do sistema evitando comportamentos inesperados        | Importante  |
| RNF-05        | O aplicativo deve ter um visual atrativo condizente com as atuais tendências de layout | Desejável   |

Tabela 2 – Requisitos não-funcionais da Aplicação

#### 4.1.2.4 Limitações Identificadas

- a) Espaço de armazenamento: A distribuição da aplicação está exposta a inúmeros modelos de dispositivos, dessa forma o desenvolvedor não tem como desenvolver o aplicativo para um hardware específico;
- b) Conexão com a internet: Deve-se considerar que nem sempre o dispositivo terá acesso a uma conexão, ou, a garantia de qualidade e velocidade da rede quando estiver conectado;
- c) Telas e modo de interação: Assim como a limitação de armazenamento, a visualização da aplicação estará exposta a diferentes modelos com qualidade, formato e capacidade de exibição totalmente distintas. Se não tratada, o aplicativo pode ter um comportamento indesejado em alguns aparelhos.

#### 4.1.2.5 Soluções Propostas

Esta seção sugere as soluções para minimizar as limitações identificadas anteriormente, buscando diminuir os impactos negativos que a aplicação possa sofrer.

- a) Espaço de armazenamento: Como a aplicação deve ser compatível com qualquer modelo que a suporte, o aplicativo carrega somente dados essenciais para seu funcionamento. Alguns arquivos multimídia - como imagens dos patrimônios - aumentam consideravelmente o tamanho da aplicação. Dessa forma cada patrimônio porta suas informações essenciais e conteúdos extras são visualizados de forma remota. Além disso, a utilização de um *software* adequado que monitora e faz uma limpeza na aplicação, removendo dados desnecessários;
- b) Conexão com a internet: Para que a aplicação não seja totalmente dependente da conexão, o aplicativo faz consultas e mantém um banco de dados interno que permite ao usuário acessar às informações sobre os patrimônios. O uso da internet acarreta em mais conteúdo e a melhor utilização do mapa;

- c) Telas e modo de interação: É previsto que o *layout* e as interações propostas possam se adaptar aos diferentes tipos de dispositivos. Todo o *design* foi desenvolvido de forma responsiva para a aplicação se adequar a qualquer modelo com as especificações técnicas que suporte o aplicativo.

## 4.2 Modelagem

No contexto da Engenharia de Software, a modelagem do sistema é parte do processo de desenvolvimento de *software*. Para a modelagem da aplicação proposta neste trabalho, utilizou-se o padrão UML, linguagem que auxilia desde sistemas simples aos mais complexos. Booch, Rumbaugh e Jacobson (2005) definiram a UML como uma linguagem padrão para o planejamento da estrutura de projetos de *software* na qual poderá ser utilizada para a visualização, a especificação, a construção e a documentação de artefatos utilizados em sistemas.

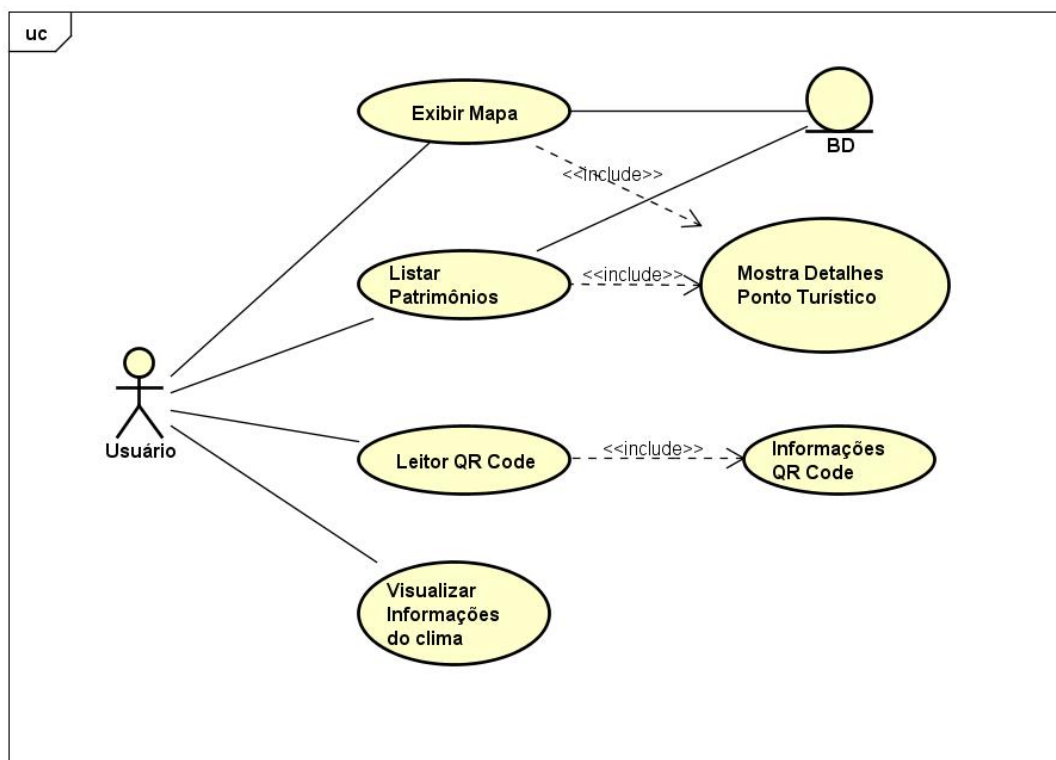
Booch *et al*(2005) explica de maneira simples que "Um modelo é uma simplificação da realidade" e "construímos modelos para compreender melhor o sistema que estamos desenvolvendo". Dessa forma, podemos compreender que a modelagem é um caminho essencial para atingir o objetivo da melhor maneira possível e permitindo contornar os problemas que surgem durante o desenvolvimento. (BOOCH; RUMBAUGH; JACOBSON, 2005).

A UML utiliza de diagramas para permitir a visualização do sistema em diferentes perspectivas, ou seja, o diagrama é a projeção de uma determinada aplicação (BOOCH; RUMBAUGH; JACOBSON, 2005). A linguagem inclui treze tipos de diagramas que são utilizados para a apresentação gráfica do comportamento de um sistema. A aplicação proposta neste trabalho não tem uma complexidade elevada que justifique a utilização de todos os tipos de diagramas, logo, ilustraremos três diagramas que serviram como base para a implementação do aplicativo.

### 4.2.1 Diagramas de Caso de Uso

Um diagrama de caso de uso ilustra a interação entre usuário e sistema, sendo importante para organização e modelagem dos comportamentos de um sistema. A figura 20 a seguir apresenta o caso de uso da aplicação, na qual o usuário acessa o sistema sem autenticação e possui três funções disponíveis.

Figura 20 – Diagrama de Caso de Uso da aplicação



Fonte: Elaborada pelo autor

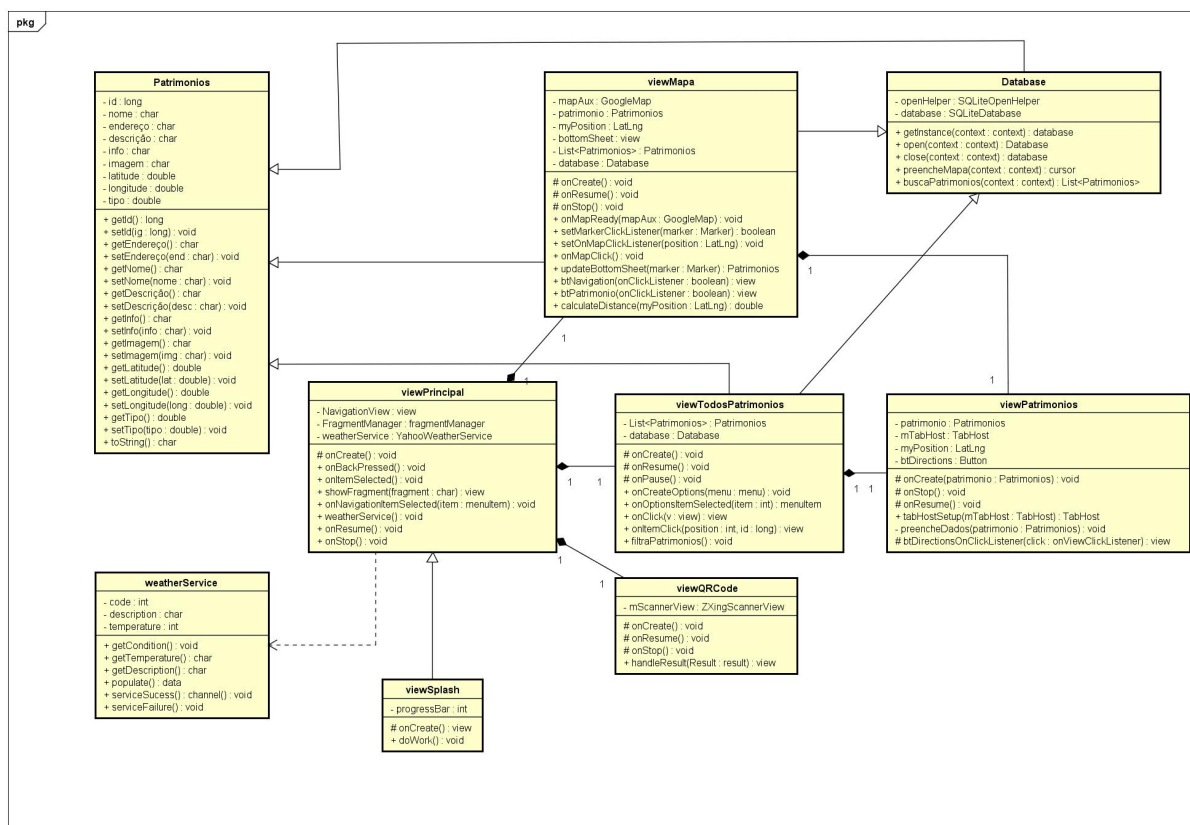
O diagrama representa bem o fluxo de eventos da aplicação que serviram como base inicial para compreender o funcionamento da aplicação. Há quatro situações possíveis quando o usuário abrir o aplicativo:

- O usuário entra no modo navegação, exibindo o mapa e a aplicação acessa o banco de dados e retorna o mapa preenchido com os patrimônios. Clicando em um patrimônio desejado, ele acessa às informações detalhadas;
- O usuário acessa o menu de listagem dos patrimônios, o aplicativo consulta o banco de dados e retorna os patrimônios disponíveis;
- O usuário abre o leitor de QR Code e pode ler o código de barra desejado;
- A aplicação exibe as informações sobre o clima da cidade de Diamantina;

## 4.2.2 Diagrama de Classes

No diagrama de classes são apresentadas as classes utilizadas no desenvolvimento do aplicativo. É possível observar a estrutura de classes com seus atributos, métodos e a relação entre as classes, conforme a figura 21.

Figura 21 – Diagrama de Classes da Aplicação

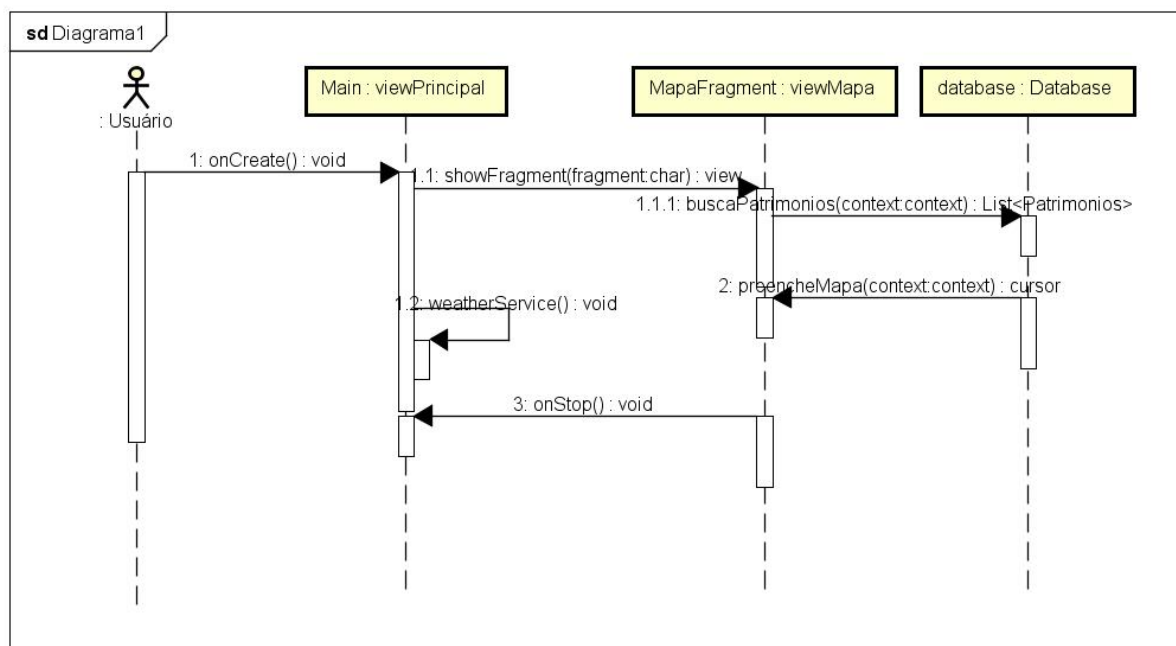


Fonte: Elaborado pelo o autor

### 4.2.3 Diagrama de Sequência

Nesta seção são exibidos os principais fluxos do processo de comunicação realizada entre as classes das camadas envolvidas em cada processo. A figura 22 exhibe o diagrama que demonstra o fluxo de eventos quando o usuário deseja acessar o modo navegação para visualizar o mapa com os patrimônios.

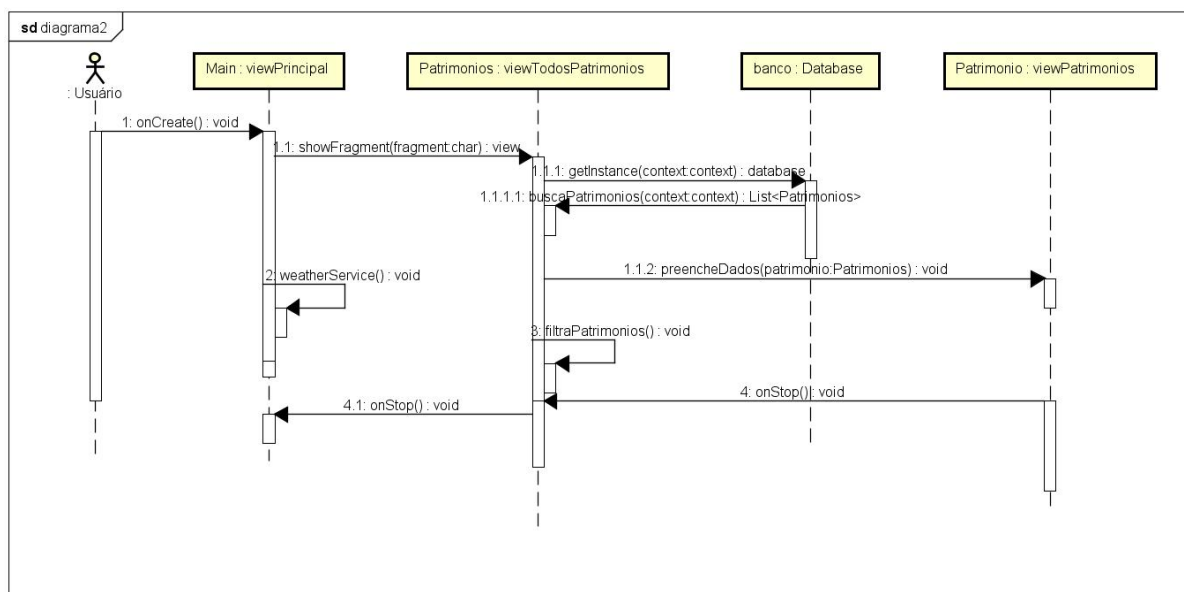
Figura 22 – Diagrama de Sequência Acesso ao Mapa



Fonte: Elaborada pelo autor

A figura 23 ilustra o fluxo da operação no aplicativo quando o usuário busca a listagem de todos patrimônios contidos na aplicação e acessa um específico para visualizar maiores detalhes.

Figura 23 – Diagrama de Sequência Acesso a um Patrimônio

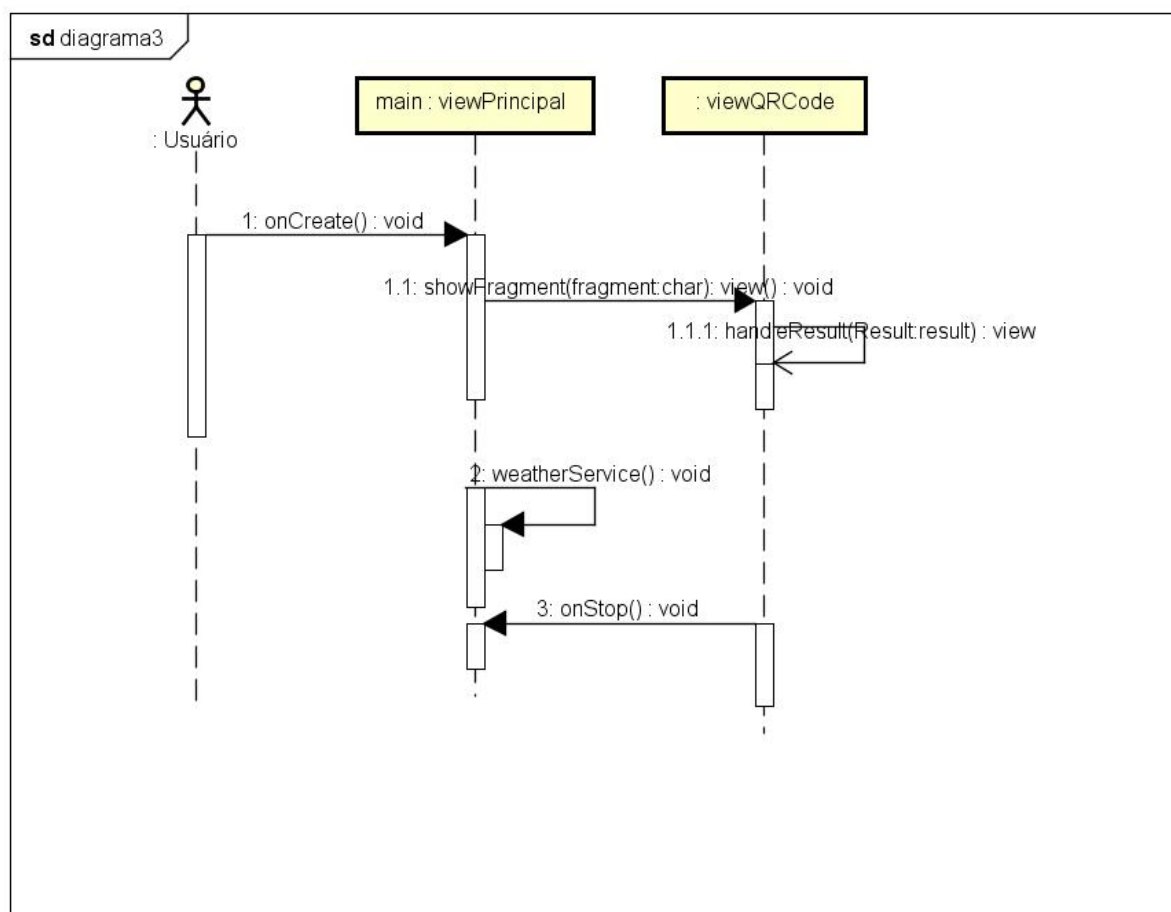


Fonte: Elaborada pelo autor

O diagrama mostrado na figura 24 representa o fluxo de eventos de quando o usuário

deseja utilizar o leitor *QR* da aplicação para ler um código.

Figura 24 – Diagrama de Sequência Acesso ao Leitor QR Code



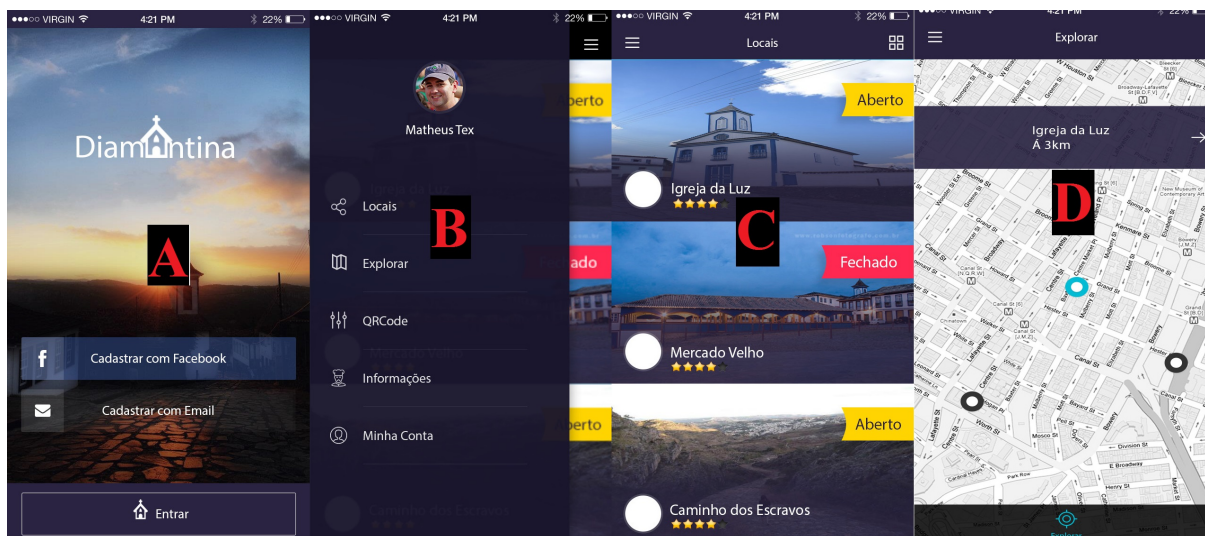
Fonte: Elaborada pelo autor

#### 4.2.4 Protótipos de Tela

A prototipagem de telas permite ao projeto uma melhor visualização e avaliação de algumas importantes funcionalidades. Os modelos criados para a aplicação foram desenvolvidos utilizando a ferramenta *Adobe Photoshop*. A criação de um protótipo auxilia a familiarizar usuários e desenvolvedores com o produto que está em fase de desenvolvimento, permitindo identificar se as principais necessidades da aplicação estão sendo atendidas (FLAILTON, 2017).

Os requisitos, limitações e diagramas orientaram a criação dos protótipos de telas que buscam se aproximar ao objetivo final do sistema e colaboram com a implementação da aplicação. Os protótipos de telas são apresentados na figura 25.

Figura 25 – Protótipo Tela Inicial



Fonte: Elaborada pelo autor

A tela indicada pela letra A exibe o esboço referente a tela inicial da aplicação foi pensada em ser simples e atrativa. A tela possui apenas funcionalidades de acesso e uma imagem que remete a cidade de Diamantina.

O protótipo da tela de menu, indicado pela letra B, foi baseado em oferecer as funcionalidades de forma ágil. Dispõe de um menu que pode ser acessado na barra fixa da aplicação - podendo ser utilizado a qualquer momento - e exibe todas as funcionalidades do aplicativo.

A tela referenciada pela letra C apresenta o protótipo da tela que é destinada a listar todos patrimônios cobertos pela aplicação. A listagem exibe imagem e nome de cada patrimônio contido no sistema e terá uma opção de busca indicado pelo ícone localizado na barra superior do aplicativo.

É possível ver na tela marcada com a letra C, a proposta da tela de navegação que basicamente têm patrimônios identificados com diferentes ícones no mapa. Quando os monumentos são acessados, uma nova exibe a opção para maiores detalhes.

Nota-se a ausência de um protótipo importante que é a tela de detalhes de patrimônios. Durante a elaboração inicial do aplicativo, não se tinha uma visão de como os dados seriam apresentados para o usuário. Todavia, a metodologia escolhida - que está descrita na seção 4.1.1 - permitiu adequações e criações quando as necessidades forem claras. Assim, a tela que exibe as informações dos patrimônios foi desenvolvida logo após análise do primeiro protótipo e será mostrada posteriormente.



## 4.3 Construção

Nesta seção está descrito os procedimentos que implementaram as etapas essenciais a construção da aplicação. O aplicativo foi desenvolvido na linguagem JAVA através da IDE *Android Studio* para funcionar em dispositivos móveis com plataforma mínima Android 4.0. Durante a construção do sistema, foi necessário o levantamento de dados e o aprofundamento no estudo das tecnologias utilizadas. Orientado pelo método de prototipagem, alguns pontos e requisitos foram alterados, buscando otimizar o desempenho e qualidade da aplicação.

### 4.3.1 Banco de Dados

Para acessar às informações coletadas referentes aos patrimônios, sem que haja a necessidade de uma conexão com a internet, os dados foram armazenadas em uma banco *SQLite*, que é a base suportada nativamente pela plataforma Android. Para facilitar a manipulação dos dados, utilizou-se o ambiente *SQLiteStudio* que serve como uma importante ferramenta de apoio para a preparação de uma base de dados ideal a aplicação. Os dados foram organizados em uma tabela, conforme podemos observar na figura 26.

Figura 26 – Ilustração da Base de Dados gerado via *SQLiteStudio*.

| Table: PATRIMONIOS |           |   |
|--------------------|-----------|---|
| Column             | Data type | Constraints                             |
| _id                | INTEGER   | •NOT NULL<br>•PRIMARY KEY AUTOINCREMENT |
| NOME               | VARCHAR   |   |
| INFORMACOES        | VARCHAR   |   |
| ENDERECO           | VARCHAR   |   |
| TOMBAMENTO         | VARCHAR   |   |
| IMAGEM             | VARCHAR   |   |
| LATITUDE           | DOUBLE    |   |
| LONGITUDE          | DOUBLE    |   |
| TIPO               | DOUBLE    |   |

Fonte: Elaborada pelo autor

Como a aplicação não necessita de acesso de usuário e depende apenas de uma entidade, o sistema foi implementado através de uma única tabela que contém todas as informações referentes a cada patrimônio. A tabela nominada como `PATRIMONIOS` possui colunas para os dados descritivo e referentes a localização, e também uma coluna identificada como `TIPO` que carrega a informação se a entidade é um museu, monumento religioso ou histórico.

Os dados foram tratados e inseridos exclusivamente no *SQLiteStudio* bastando a aplicação apenas acessar o conteúdo desejado. Para isso, utilizou-se da biblioteca nativa de

SQLite disponível no Android e o banco pode ser acessado de forma simples, como demonstrado na figura 27.

Figura 27 – Acesso a base de dados.

```
public Cursor preencheMapa (Context context)
{
    Cursor cursor = database.rawQuery("SELECT * FROM PATRIMONIOS", null);

    return cursor;
}
```

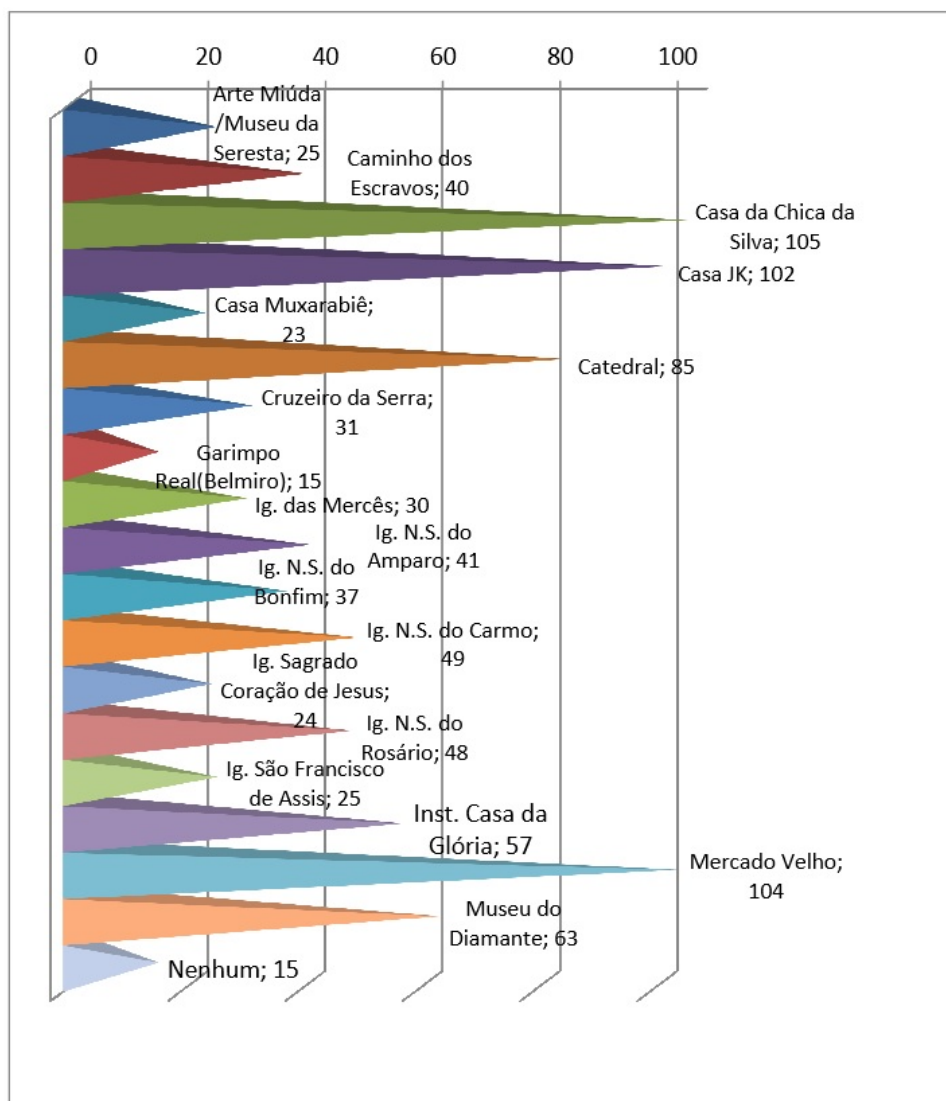
Fonte: O autor

A estrutura `cursor` executa através da classe `database` realiza uma busca completa por todos patrimônios contidos no banco. Assim, as informações são armazenadas na estrutura, que é retornada ao final da função.

### 4.3.2 Levantamento dos Patrimônios

O intuito da aplicação é prover uma base de informações completas sobre o patrimônios da cidade de Diamantina. Para atingir este objetivo, propõe-se que o sistema tenha caráter colaborativo e que com o tempo possa atingir um amplo e rico acervo de dados. Contudo, é necessário uma base inicial com uma quantidade de monumentos que tornem a aplicação útil e atrativa ao usuário, mas que condiz com o prazo e metas do trabalho. Para isso, o levantamento dos patrimônios foi orientado por uma pesquisa realizada em 2012 pela UFVJM, denominada Perfil da Demanda Turística Real de Diamantina e Região que identificou os patrimônios mais visitados pelos turistas, como pode ser visualizado na figura 28.

Figura 28 – Patrimônios mais visitados pelos turistas.



Fonte: (UFVJM, 2012)

O levantamento permitiu identificar os 18 locais mais frequentados por turistas e que foram selecionados para integrar o acervo inicial da aplicação. A diversidade desses patrimônios é interessante, pois coloca o aplicativo a prova de exercer todas suas funcionalidades em lugares com grande chance de receberem visitação. Uma quantidade excessiva de monumentos poderia acarretar a perda da qualidade de informações ou dados desnecessários.

Dessa forma, para a construção do acervo inicial foram necessários a colaboração e a utilização de ferramentas que pudessem aumentar a qualidade dos dados passados aos usuários. Primeiramente, as coordenadas geográficas foram conseguidas utilizando a plataforma *Google Maps*. Dados de tombamento e detalhes de cada patrimônio foram adquiridos com o apoio do IPHAN e colaboradores do Departamento de História da UFVJM, destaca-se a participação do aluno Rodrigo Rigon Giglio que elaborou os textos de forma adequada a aplicação. A base

de dados foi aberta aos colaboradores que contribuíram e adicionaram os textos para cada monumento. As imagens contou com a colaboração do fotógrafo Sérgio Miranda e foram retiradas especificamente para incrementar o acervo do aplicativo. É importante ressaltar que, desde o início o levantamento se apoiou no caráter colaborativo e esteve aberto a qualquer conteúdo que pudesse enriquecer o acervo da aplicação.

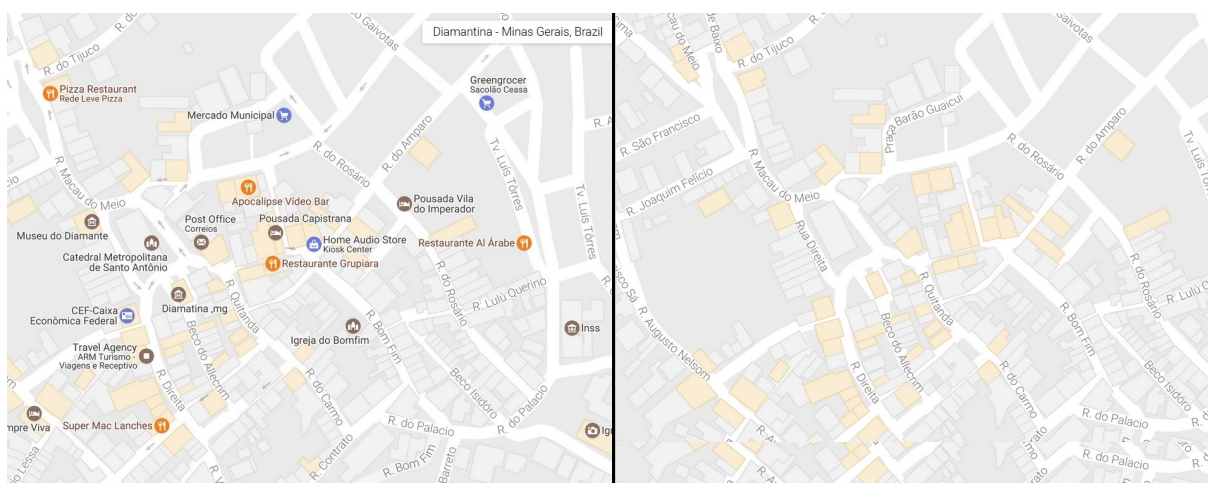
### 4.3.3 Mapa e Navegação

Para a implementação do módulo de navegação do sistema, foi utilizada a *Maps API*. Disponibilizada pela Google a biblioteca fornece suporte nativo para completa utilização de mapas em aplicações na plataforma Android através de uma fácil implementação. Para desfrutar das funcionalidades da API, o desenvolvedor deve portar uma chave Android que pode ser adquirida registrando o projeto do aplicativo no *Google Developers Console*. Após a aquisição da chave, essa deve ser inserida no código do aplicativo e a API estará disponível para uso.

Foi necessário que o mapa oferecido por padrão pela API sofresse alterações para se adequar as funcionalidades da aplicação, tornando-o mais intuitivo e atrativo aos usuários, por exemplo, a exibição de pontos de comércio foi removida, pois polui a visualização do mapa e não é essencial. Essas alterações são facilitadas pela simplicidade na implementação e por ferramentas disponibilizadas pela Google para personalização do mapa.

A primeira medida foi aplicada através da ferramenta *Map Style*<sup>12</sup> que retirou marcações e informações indesejadas, deixando o mapa visualmente mais atrativo e apto a receber as personalizações adequadas. A figura 29 exhibe a visualização do mapa antes, à esquerda, e após receber o novo estilo, à direita.

Figura 29 – Antes e Depois da personalização do mapa.



Fonte: Elaborado pelo autor

<sup>12</sup> Ferramenta para personalização de mapas que pode ser acessada em <https://mapstyle.withgoogle.com/>

Com o mapa pronto a receber os marcadores que identificam a localização de cada patrimônio, foram implementadas diferentes formas visuais que ilustram as marcações para cada tipo de monumento buscando facilitar a interação com o usuário.

Uma funcionalidade a ser implementada encontrou barreiras nos termos de privacidade da Google. Segundo a empresa, a cláusula 10.4.c descreve que não é permitido a navegação, ou seja, o serviço ou conteúdo da API não será utilizado para navegação em tempo real ou guia de direção (GOOGLE, 2017). Diante disso, quando um usuário quiser navegar até determinado patrimônio, este será redirecionado ao aplicativo *Google Maps* que possui mecanismos eficiente para orientar o utilizador até o local desejado. A figura 30 exibe o método implementado para ativar o modo navegação através da aplicação da Google.

Figura 30 – Método para Navegação

```
@Override
public void onClick(View v)
{
    String uri = String.format(Locale.ENGLISH, "http://maps.google.com/maps?saddr=%f,%f(%s)&daddr=%f,%f (%s)",
        myPosition.latitude, myPosition.longitude, "Minha Localização",
        marker.getPosition().latitude, marker.getPosition().longitude, marker.getTitle());
    Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(uri));
    intent.setClassName("com.google.android.apps.maps", "com.google.android.maps.MapsActivity");
    startActivity(intent);
}
```

Fonte: Elaborado pelo autor

A função opera da seguinte forma, uma `String` armazena os dados da localização atual e onde o usuário deseja chegar. Uma nova `view` é criada com os parâmetros especificados e com referência a classe da aplicação *Google Maps*. Ao fim, o método `startActivity` invocará a tela do aplicativo construída a partir dos parâmetros armazenados.

#### 4.3.4 Leitor QR Code

Uma das propostas do trabalho é a utilização de códigos QR que portem informações relevantes aos monumentos. Para tal, foi imprescindível a implantação de um leitor dentro da aplicação que permitisse a leitura desses códigos. A codificação da função foi realizado através do auxílio da biblioteca de código-livre ZXing<sup>13</sup> que provê todas as funcionalidades necessárias é escrita em JAVA e possui uma vasta documentação.

Para implementação da biblioteca, essa deve ser inserida no arquivo de compilação e também requisita uma permissão adicional para utilização da câmera do dispositivo. Após isso, a classe pode ser executada e as alterações concebidas de acordo com a necessidade de desenvolvimento. Os métodos são demonstrados na figura à seguir.

<sup>13</sup> Guia de utilização pode ser encontrado em <https://github.com/zxing/zxing/wiki/Getting-Started-Developing>

Figura 31 – Classe do Leitor QRCode

```
public class SimpleScannerActivity extends Activity implements ZXingScannerView.ResultHandler {
    private ZXingScannerView mScannerView;

    @Override
    public void onCreate(Bundle state) {
        super.onCreate(state);
        mScannerView = new ZXingScannerView(this); // Programmatically initialize the scanner view
        setContentView(mScannerView); // Set the scanner view as the content view
    }

    @Override
    public void onResume() {
        super.onResume();
        mScannerView.setResultHandler(this); // Register ourselves as a handler for scan results.
        mScannerView.startCamera(); // Start camera on resume
    }

    @Override
    public void onPause() {
        super.onPause();
        mScannerView.stopCamera(); // Stop camera on pause
    }

    @Override
    public void handleResult(Result rawResult) {
        // Do something with the result here
        Log.v(TAG, rawResult.getText()); // Prints scan results
        Log.v(TAG, rawResult.getBarcodeFormat().toString()); // Prints the scan format (qrcode, pdf417 etc.)

        // If you would like to resume scanning, call this method below:
        mScannerView.resumeCameraPreview(this);
    }
}
```

Fonte: Elaborado pelo autor

No método `onCreate` a classe `ZXing` é instanciada. Os métodos `onResume` e `onPause` são responsáveis por manipular o funcionamento da câmera quando a aplicação é interrompida ou resumida, como também o estado da `view`. A função `handleResult` é responsável por tratar a forma com que o código é lido e como será exibido. O funcionamento básico da biblioteca pode ser visualizado na figura 32.

Figura 32 – Leitor QR Code ZXing



Fonte: Elaborado pelo autor

#### 4.3.4.1 Geração dos Códigos

O funcionamento do leitor QR em conjunto com as informações pertinentes aos patrimônios já levantadas, tornou viável a criação dos códigos que auxiliam a aplicação. Perante a grande quantidade de geradores disponíveis na rede, a escolha se deu pelo QR Code Generator<sup>14</sup> que oferece uma interface simples, prática e gratuita. A figura 33 exibe a tela principal da aplicação.

Figura 33 – Gerador *QR Code*



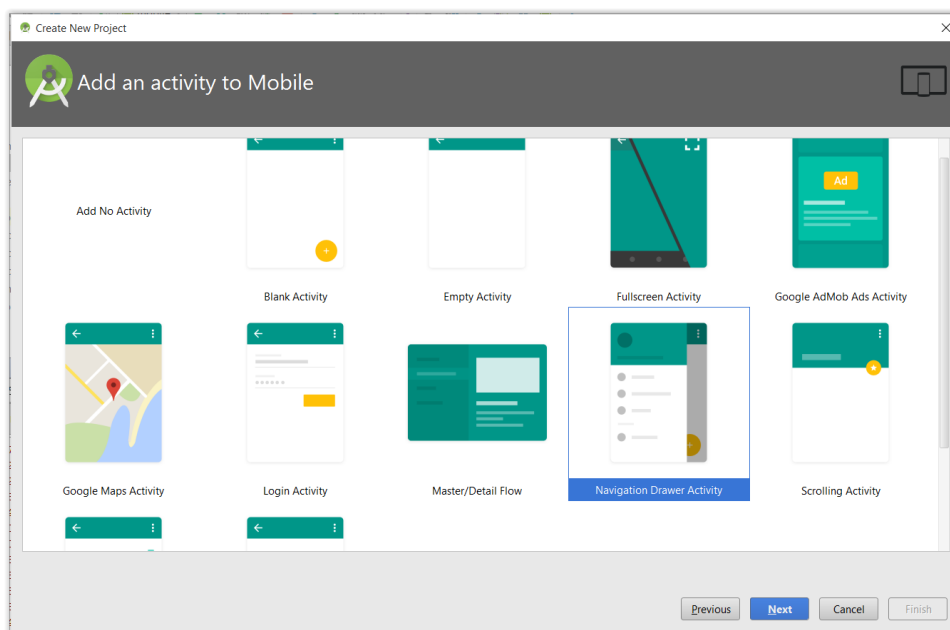
Fonte: Elaborado pelo autor

Para geração de um código, basta escolher o tipo de dados, inserir as informações e após selecionar a opção de criação, o código estará pronto para uso.

#### 4.3.5 Menu Principal

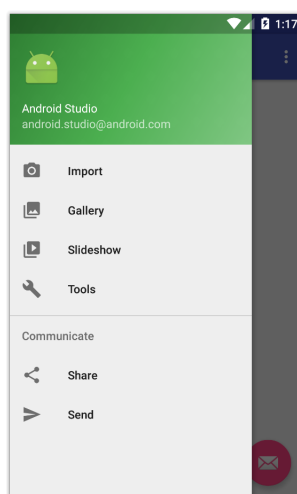
Para o bom funcionamento da aplicação, buscou-se uma forma em que as telas estivessem distribuídas claramente e que fossem de fácil acesso ao usuário. A solução foi a utilização da classe *NavigationView* que é de simples geração no ambiente de desenvolvimento e possui todos os recursos necessários para controlar as telas da aplicação. A classe pode ser implementada através do passo a passo disponibilizada pela IDE, conforme é apresentado na figura 34.

<sup>14</sup> Disponível em <http://br.qr-code-generator.com/>

Figura 34 – Gerador Classe *Navigation View*

Fonte: Elaborado pelo Autor

A *NavigationView* é dividida em duas partes; a parte superior - também chamada de *header* - é utilizada para dispor as informações que os desenvolvedores julgam ser necessárias. A segunda parte, é o *menu* que compõe as opções necessárias para acesso às funcionalidades do sistema. A classe é comumente utilizada em várias aplicações como, por exemplo, o *GoogleMaps*. A figura 35 exibe o modelo padrão criado pelo *Android Studio*.

Figura 35 – Modelo de *Navigation View*

Fonte: Elaborado pelo Autor

A *view* foi personalizada segundo as especificações e orientadas pela prototipagem descrita na seção 4.3.4. O menu recebeu as opções de acesso a todas as telas da aplicação e teve



seu *layout* alterado para uma melhor visualização por parte do usuário. A parte superior permitiu implantar o requisito RF-08 e cedeu espaço a API de serviços climáticos.

# 5 RESULTADOS OBTIDOS

O resultado deste trabalho foi uma aplicação nativa para a plataforma Android, que auxiliada por algumas tecnologias, como LBS e *QR Code*, pode desempenhar um importante papel de suporte aos visitantes que passam por Diamantina. Nas seguintes seções será realizada a apresentação do sistema e o detalhamento das funcionalidades de cada tela do aplicativo.

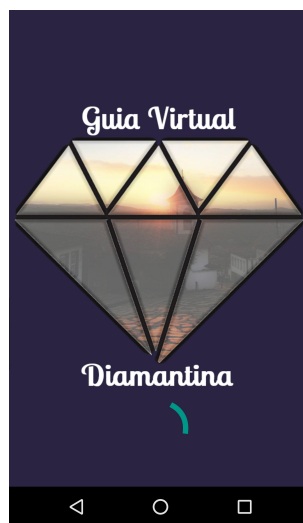
## 5.1 Apresentação do Sistema

O sistema denominado por "Guia Virtual Diamantina" passou por algumas versões e recebeu melhorias para atingir os requisitos levantados. O protótipo a ser apresentado é considerado a terceira versão e foi desenvolvida para suportar dispositivos que portem a plataforma Android em versão 4.0 (*Ice Cream Sandwich*) ou superior. A instalação é através de um arquivo com aproximadamente *10 megabytes* e ocupa cinco vezes mais espaço quando instalado.

### 5.1.1 Tela de Apresentação

Ao acessar o aplicativo, a primeira tela visualizada exibe apenas o logo e nome do aplicativo, porém sua principal função é mascarar os processos de carregamento que o aplicativo precisa executar. Por exemplo, enquanto a tela é exibida, o mapa principal do sistema faz o preenchimento dos pontos presentes na base de dados.

Figura 36 – Tela de Apresentação da Aplicação

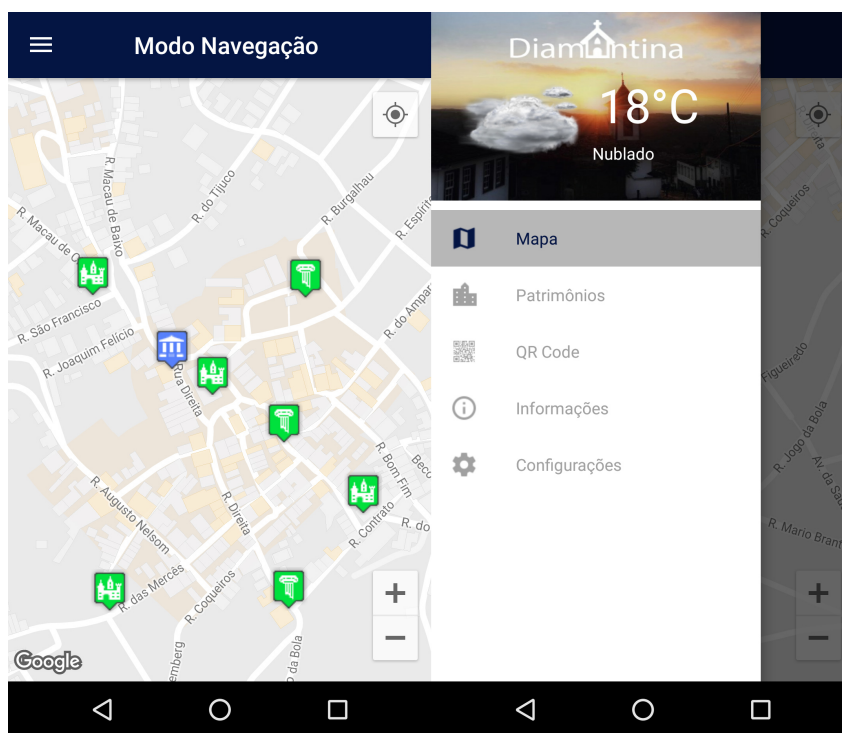


Fonte: Elaborado pelo Autor

### 5.1.2 Tela Principal

A tela principal contém as opções e ferramentas essenciais para o controle do sistema, sendo a responsável por fazer o fluxo pelas demais funcionalidades da aplicação. Podemos verificar a tela na figura 37.

Figura 37 – Tela Principal da Aplicação



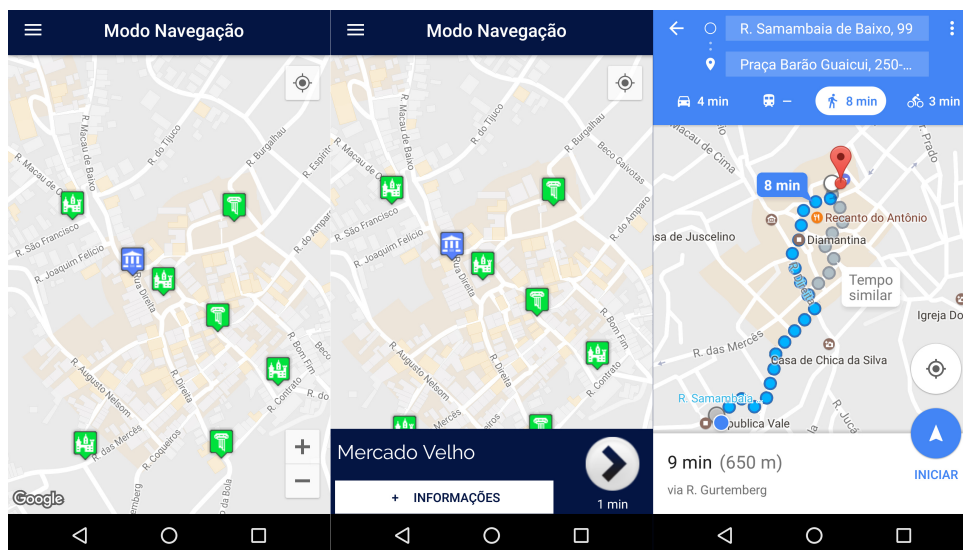
Fonte: Elaborada pelo autor

O menu pode ser acessado através do botão localizado na parte inferior esquerda e porta as opções para acessar as demais telas da aplicação. É possível observar na parte superior do menu, informações climáticas de Diamantina que atendem ao requisito RF-08. Destaca-se que o acesso ao menu foi implementado em todas as telas através da utilização de uma barra estática, dessa forma o botão de acesso estará sempre disponível ao usuário.

### 5.1.3 Tela Navegação

A tela de navegação possibilita a visualização do mapa preenchido com os patrimônios disponíveis no banco de dados. O usuário pode identificar cada monumento através de ícones que são colocados de acordo com a localização fornecida pela API *GoogleMaps*. Clicando no ícone, uma janela fornece a possibilidade de maiores informações sobre determinado local, ou acessar o modo navegação que traça o trajeto da localização atual do dispositivo até o patrimônio. A tela é ilustrada na figura 38.

Figura 38 – Tela de Navegação



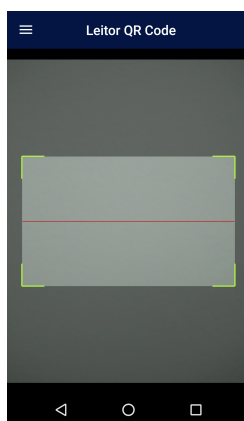
Fonte: Elaborada pelo autor

Para facilitar a visualização dos dados, cada local recebe um ícone personalizado para cada tipo de monumento, por exemplo, é possível reconhecer os patrimônios religiosos identificados com um ícone de igreja. Entre outras funcionalidades, próximo ao botão de navegação é possível verificar o tempo estimado de deslocamento até a localização desejada. O mapa, também foi personalizado para tornar a visualização mais limpa e clara possível.

### 5.1.4 Tela Leitor QR Code

A tela tem uma única função que é realizar a leitura de um código QR. Dessa forma, quando inicializada, ela ativa a câmera do celular e permite ao usuário a leitura do código. A tela é representada pela figura 39, a seguir.

Figura 39 – Tela do Leitor QR Code

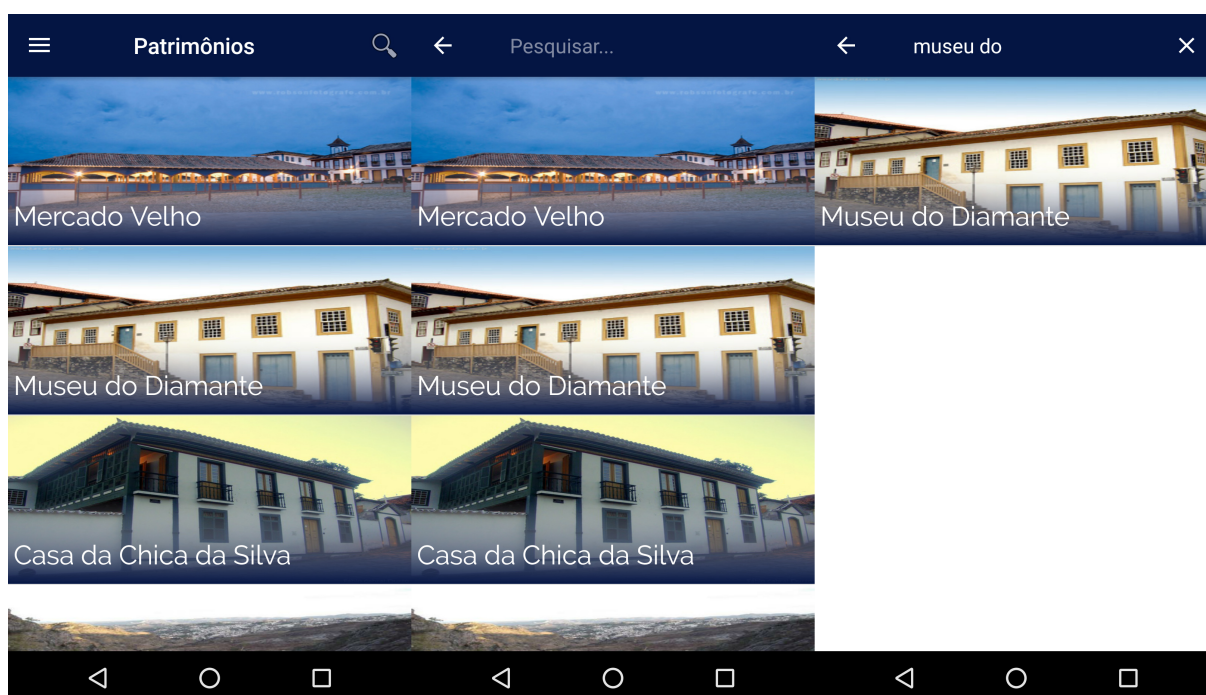


Fonte: Elaborada pelo autor

### 5.1.5 Tela Listagem Patrimônios

Acessada através do menu, a tela exibe todos patrimônios inseridos na base de dados. Sem muitas funcionalidades, a intenção é fornecer uma prévia visualização dos monumentos disponíveis, além de oferecer uma opção de busca para agilizar a procura do usuário. As imagens referentes a cada monumento são localizadas no banco de dados da própria aplicação, dispensando o uso de conexão com a internet e proporcionando um melhor desempenho. A tela e suas funcionalidades podem ser vistas na figura 40.

Figura 40 – Tela de Listagem dos Patrimônios



Fonte: O Autor

Quando um patrimônio for selecionado, o usuário será direcionado a tela com os detalhes referentes ao monumento. Para facilitar o processo de busca, a parte superior possui uma ferramenta de busca por caracteres. Ressalta-se que essas funcionalidades possibilitaram o cumprimento dos requisitos RF-01 e RF-05.

### 5.1.6 Tela Patrimônio

Responsável por exibir as informações de cada patrimônio ela pode é acessada quando o usuário deseja acessar dados detalhados de um monumento. O usuário pode visualizar informações, imagens, dados de tombamento e ativar o recurso de navegação até o local do monumento. A tela é exibida na figura 41 e utilizou o monumento conhecido como Mercado Velho para exemplificação.

Figura 41 – Tela de Detalhes dos Patrimônios



Fonte: O Autor

A opção de navegação pode ser ativada através do ícone na parte superior direita ou clicando no botão onde é indicado o endereço. Os dados são exibidos de forma alternada de acordo com o desejo do usuário.

De acordo com o desenvolvimento e planejamento do aplicativo, percebemos a sua materialização a partir do que foi discutido e apresentado neste capítulo. A expectativa, nesse momento, é fazer a divulgação do Guia Virtual de modo a contribuir com o acesso às informações sobre os patrimônios históricos, facilitando o tour dos turistas que visitam a cidade de Diamantina. Espera-se que com maior visibilidade a aplicação receba novos conteúdos para incrementar o acervo disponibilizado pelo aplicativo.

## 6 CONCLUSÃO

A cidade de Diamantina destaca-se por possuir um vasto acervo de monumentos históricos, religiosos e naturais. Porém, esses patrimônios carecem de suporte aos visitantes interessados em conhecê-los. O protótipo do aplicativo desenvolvido neste trabalho propõe-se a suprir essa carência, servindo como um guia turístico virtual a qualquer pessoa portadora de um dispositivo móvel. O desenvolvimento do sistema foi motivado pelas diversas aplicações, como o Portal do Patrimônio e o QRio que surgiram em âmbito nacional para auxiliar o setor turístico.

Ao longo da realização do trabalho, a partir do levantamento dos dados, ferramentas e tecnologias, definiu-se um arcabouço que viabilizasse a construção da aplicação. Esse escopo permitiu que o sistema fosse construído de forma adequada, buscando resolver o problema definido como objetivo do trabalho. As escolhas referentes aos materiais e métodos foram cruciais no desenvolvimento do trabalho, com destaque da utilização da plataforma Android, da utilização da API *Google Maps* e à adesão do ambiente de desenvolvimento *Android Studio*.

O resultado da implementação foi a aplicação Guia Virtual Diamantina. O sistema mostra que é possível a utilização de um aplicativo para auxiliar os turistas na escolha de lugares interessantes na cidade de Diamantina. Os visitantes podem desfrutar do sistema para visualizar informações e imagens, localizar patrimônios e navegar até os monumentos desejados. Alguns requisitos não foram implementados, pois não houve tempo hábil de serem executados da mesma maneira e qualidade que outros requisitos foram. Entretanto, esses são considerados o ponto de partida para futuras melhorias na solução.

O objetivo principal de implementar uma ferramenta de suporte ao visitante foi atendida e mostrou-se funcional. A construção do projeto contribuiu para o refinamento de conceitos aprendidos durante a graduação e conhecimento de novas ferramentas, teorias e *software* condizente com a realidade do mercado e meio acadêmico.

### 6.1 Trabalhos Futuros

Como possibilidades futuras para a continuação deste trabalho ficam todos os requisitos não alcançados e funcionalidades pensadas, porém não implementadas, com destaques para algumas que serão descritas a seguir:

- a) Adição de roteiros turísticos;
- b) Incluir outros idiomas;
- c) Permitir ao usuário dar nota aos pontos turísticos;
- d) Inclusão de novas camadas no mapa como bares, restaurantes, hotéis;

- e) Expansão do acervo à Estrada Real;
- f) Integração com redes sociais tais como *Facebook* e *Twitter*;
- g) Permitir ao usuário enviar conteúdo como imagens e dados para a aplicação.

Acredito que o aplicativo pode ser melhorado futuramente com a inserção e implementação das funcionalidades expostas no parágrafo anterior. Dessa forma, esperamos colaborar com o turismo local, aliando uma das maiores potencialidades econômicas da cidade com a aplicação de conhecimentos estudados no curso de graduação em Sistemas de Informação.



# REFERÊNCIAS

- ABRANTES, A. R. *Tecnologias digitais como instrumentos de preservação do patrimônio urbano edificado*. [S.l.: s.n.], 2014. Nenhuma citação no texto.
- ANDROID. *Android Interfaces and Architecture | Android Open Source Project*. 2016a. Disponível em: <<http://source.android.com/devices/index.html>>. Acesso em: 20 jan 2017. Citado 4 vezes nas páginas 18, 24, 25 e 26.
- ANDROID. *Meet Android Studio | Android Studio*. 2016b. Disponível em: <<https://developer.android.com/studio/index.html>>. Acesso em: 18 jan 2017. Citado 2 vezes nas páginas 24 e 42.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. 9. ed. Rio de Janeiro: Elsevier, 2005. Citado na página 49.
- CAELUM. *Apostila Java e Orientação a Objetos*. 2016. Disponível em: <<https://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/>>. Acesso em: 04 fev 2017. Citado na página 41.
- CERETO, C.; GAUDERETO, K. *Iphan-RJ lança Portal do Patrimônio, aplicativo para celular e tablet*. 2014. Disponível em: <<http://portal.iphan.gov.br/noticias/detalhes/146/>>. Acesso em: 5 fev 2017. Nenhuma citação no texto.
- COSTA, O. N.; FILHO, F. N.; DUARTE, F. A. *Avaliação Comparativa de Sistemas Operacionais para Dispositivos Móveis: Foco em suas funcionalidades*. 2012. Citado 4 vezes nas páginas 18, 19, 20 e 21.
- DEITEL, M. H. *Java: Como programar*. [S.l.: s.n.], 2010. Citado na página 40.
- ERL, T. *Introdução às tecnologias Web Services: SOA, SOAP, WSDL e UDDI - Parte 1*. 2015. Disponível em: <<http://imasters.com.br/artigo/4245/web-services/entendendo-os-webservices?trace=1519021197&source=single>>. Acesso em: 5 fev 2017. Citado na página 35.
- EXAME, R. I. *Por que o Android engoliu a Apple*. 2011. 58 p. Citado na página 23.
- FELIX, R. *SQLite - O Pequeno Notável*. 2016. Disponível em: <<http://www.devmedia.com.br/sqlite-o-pequeno-notavel/7249>>. Acesso em: 15 jan 2017. Citado na página 26.
- FERRARO, R.; AKTIHANOGLU, M. *Location Aware Applications*. New York, NY, USA: Manning, 2011. ISBN 978-1-935182-33-7. Citado 3 vezes nas páginas 28, 29 e 30.
- FILHO, W. de P. P. *Engenharia de Software: fundamentos, métodos e padrões*. 3. ed. Rio de Janeiro: [s.n.], 2009. Citado na página 37.
- FINCI, R. *Rio coloca QR codes em pontos turísticos da cidade*. 2013. Disponível em: <<http://www.guiadasemana.com.br/turismo/noticia/rio-coloca-qr-codes-em-pontos-turisticos-da-cidade>>. Acesso em: 5 fev 2017. Citado na página 34.

- FLAILTON. *Prototipagem: o que é e para que serve?* 2017. Disponível em: <<https://nuvemandroid.wordpress.com/2013/10/15/ferramentas-de-prototipagem-de-tela-para-dispositivos-moveis/>>. Acesso em: 15 jan 2017. Citado na página 53.
- FLING, B. *Mobile Design and Development. Sebastopol*. Sebastopol, CA: O'Reilly, 2009. Citado na página 17.
- GARGENTA, M. *Learning Android*. Sebastopol, CA: O'Reilly, 2011. Citado na página 23.
- GARTNER. *Gartner Says Worldwide Smartphone Sales Recorded Slowest Growth Rate Since 2013*. 2015. Disponível em: <<http://www.gartner.com/newsroom/id/3115517>>. Acesso em: 22 jan 2017. Citado na página 17.
- GOMES, H. S. *Apps para smartphone se tornam canal nº 1 de bancos brasileiros*. 2016. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2016/03/apps-para-smartphone-se-tornam-canal-n-1-de-bancos-brasileiros.html>>. Acesso em: 17 jan 2017. Citado na página 17.
- GOOGLE. *Google Maps APIs Terms of Service*. 2017. Disponível em: <<https://developers.google.com/maps/terms>>. Acesso em: 05 fev 2017. Citado na página 59.
- HARVARD. *Como a Mobilidade está Mudando o Mundo*. 2016. Disponível em: <[https://www.sapvirtualagency.com/FileExplorer/Partners/Sybase/MobileistheGameChanger/ptBR/How\\_Mobility\\_Is\\_Changing\\_the\\_World\\_-\\_Brazilian\\_Portuguese.pdf](https://www.sapvirtualagency.com/FileExplorer/Partners/Sybase/MobileistheGameChanger/ptBR/How_Mobility_Is_Changing_the_World_-_Brazilian_Portuguese.pdf)>. Acesso em: 03 fev 2017. Citado na página 17.
- JACOB, S. *5 Genius Examples of QR Codes in Marketing*. 2012. Disponível em: <<https://blog.kissmetrics.com/genius-qr-codes/>>. Acesso em: 5 fev 2017. Citado 2 vezes nas páginas 32 e 33.
- JESUS, A. *Windows Phone é a plataforma sucessora do Windows Mobile*. 2014. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/windows-phone.html>>. Acesso em: 10 jan 2017. Citado na página 20.
- KÜPPER, A. *Location-Based Services: Fundamentals and Operation*. [S.l.: s.n.], 2005. Citado na página 27.
- KREGER, H. *Web Services Conceptual Architecture (WSCA 1.0)*. [S.l.: s.n.], 2001. Citado 2 vezes nas páginas 36 e 37.
- LASTRES, H. M. M.; ALBAGLI, S. *Informação e globalização na era do conhecimento*. [S.l.: s.n.], 1999. Nenhuma citação no texto.
- LECHETA, R. R. *Google Android*. São Paulo, SP: Novatec, 2010. Citado na página 18.
- MADUREIRA, D. *Dispositivos Móveis e o seu impacto na sociedade*. [S.l.: s.n.], 2011. Citado na página 13.
- MEDINA, R. D. *Linguagem XML*. 2014. Disponível em: <<http://www-usr.inf.ufsm.br/~rose/curso3/cafe/XML-Cap1-Linguagem.pdf>>. Acesso em: 20 nov 2016. Citado na página 41.
- MEMORIALL. *Sobre nós*. 2014. Disponível em: <<http://www.memoriall.com.br/#about-1>>. Acesso em: 5 fev 2017. Citado na página 33.

- MOBILE, V. *Apps para smartphone se tornam canal nº 1 de bancos brasileiros*. 2011. Disponível em: <[\[Infographic\]TheMobilePlatformRace-Howdomobileplatformsstackup?>](#) Acesso em: 28 jan 2017. Citado na página 22.
- MOON, P. *A arma do google é o Android*. 2008. Disponível em: <<http://revistaepoca.globo.com/Revista/Epoca/0,,EMI6849-15273,00.html>>. Acesso em: 03 fev 2017. Citado na página 23.
- PANKIEWICZ, I. *O que são os QR Codes?* 2009. Disponível em: <<http://source.android.com/devices/index.html>>. Acesso em: 20 nov 2016. Citado 2 vezes nas páginas 31 e 32.
- PRESSMAN, R. S. *Engenharia de Software*. 6. ed. São Paulo: [s.n.], 2006. Citado 3 vezes nas páginas 37, 38 e 39.
- RECKZIEGEL, M. *Entendendo os WebServices*. 2006. Disponível em: <<http://imasters.com.br/artigo/4245/web-services/entendendo-os-webservices?trace=1519021197&source=single>>. Acesso em: 5 fev 2017. Citado na página 36.
- ROEBUCK, K. *QR Code*. Lexington, USA: Teboo, 2011. Citado na página 30.
- SACCOL, A. Z.; REINHARD, N. *Tecnologias de informação móveis, sem fio e ubíquas: definições, estado-da-arte e oportunidades de pesquisa*. 2007. 175-198 p. Citado na página 16.
- SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Education do Brasil, 2011. Citado na página 46.
- SOUZA, D. L. *SQLiteStudio - Gerenciador de banco de dados SQLite*. 2013. Disponível em: <<https://www.vivaolinux.com.br/dica/SQLiteStudio-Gerenciador-de-banco-de-dados-SQLite>>. Acesso em: 2 fev 2017. Citado na página 43.
- THANH, T. N. *Windows Phone Introduction*. 2014. Disponível em: <<https://www.slideshare.net/nikitakova/windows-phone-introduction>>. Acesso em: 15 jan 2017. Citado 2 vezes nas páginas 19 e 20.
- UFVJM. *Perfil da Demanda Turística Real de Diamantina e Região*. 2012. Disponível em: <[http://www.ufvjm.edu.br/cursos/bhu/eixo-de-disciplinas/doc\\_view/46-.html](http://www.ufvjm.edu.br/cursos/bhu/eixo-de-disciplinas/doc_view/46-.html)>. Citado 2 vezes nas páginas 14 e 57.
- VENTURA, P. *O que é API*. 2015. Disponível em: <<http://www.ateomomento.com.br/o-que-e-api/>>. Acesso em: 11 jun 2015. Citado 2 vezes nas páginas 34 e 35.
- WAZE. *Sobre nós*. 2017. Disponível em: <<https://www.waze.com/pt-BR/about>>. Acesso em: 5 fev 2017. Nenhuma citação no texto.
- WILILAMS, K. *Location-Based Services (LBS): Are there any other apps just like Tinder?* 2015. Disponível em: <<https://www.quora.com/Location-Based-Services-LBS-Are-there-any-other-apps-just-like-Tinder>>. Acesso em: 5 fev 2017. Citado na página 27.