Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Guia do Aluno UnB: Aplicativo Georreferenciado de Apoio a Atividades Acadêmicas

Frederico Fernandes de Faria
Jean Victor Ribeiro Vieira

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientadora
Prof.a Dr.a Aletéia Patrícia Favacho de Araújo

Coorientadora
Prof.a Dr.a Maristela Terto de Holanda

Brasília
2017

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

# Guia do Aluno UnB: Aplicativo Georreferenciado de Apoio a Atividades Acadêmicas

Frederico Fernandes de Faria
Jean Victor Ribeiro Vieira

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof.a Dr.a Aletéia Patrícia Favacho de Araújo (Orientadora)
CIC/UnB

Prof.a Dr.a Fernanda Lima
CIC/UnB

Prof. Dr. Wilson Henrique Veneziano
CIC/UnB

Prof. MSc. Pedro A.D Rezende
Coordenador do Curso de Computação — Licenciatura

Brasília, 23 de agosto de 2017

# Dedicatória

Dedicamos este trabalho a nossos pais e familiares, assim como nossos estimados amigos, pelo apoio incondicional e empenho solícito e diligente em prol do nosso bem próprio. Também o dedicamos a Dra. Aletéia Patrícia, pelo desvelo e dedicação com que nos acolheu durante a orientação acadêmica. Excelsa, dedicamos a Hevelson e Doris, juntamente com Claúdio, Ivanete e Luis, pela ventura e zelo em nos apoiar. E em memória a Paulo de Deus, mentor da computação e da vida.

# Agradecimentos

Agradecemos ao corpo docente da UnB por nos proporcionar esmera formação acadêmica, assim como aos desenvolvedores parceiros que nos auxiliaram, especialmente à Dra. Aletéia Patrícia e à Dra. Maristela Terto que proporcionaram perfeitas condições para que o projeto fosse concluído com excelência.

# Resumo

Este trabalho apresenta o software Guia do Aluno UnB, um aplicativo móvel de mapeamento georreferenciado da Universidade de Brasília, incluindo seu processo de elaboração, implementação e desenvolvimento. O aplicativo é composto por um conjunto de menus com ferramentas destinadas a Universidade de Brasília. Dentre essas ferramentas, há um mapa interativo com os estabelecimentos, as salas de aulas, as secretarias e os departamentos da Universidade. No âmbito administrativo, o aplicativo propõe ferramentas de gestão da informação, tais como acesso aos documentos institucionais, gestão de faltas e tarefas, assim como acesso aos principais portais da UnB. Este aplicativo se destina a todos os usuários da Universidade, vinculados ou visitantes, como software de apoio a atividades acadêmicas.

**Palavras-chave:** Android, Georreferenciamento, Gestão Acadêmica, UnB

# Abstract

This work presents the software Guia do Aluno UnB, a mobile georeferencing mapping application from University of Brasília, including it's elaboration process, implementation and development. The application consists on a set of menus with tools made for the University of Brasília. Among these tools, there is an interactive map with establishments, classrooms, secretaries and university departments. At the administrative level, the application proposes information management tools, such as access to institutional documents, absence and tasks management, as well as access to UnB's main websites. This application is intended for all users of the University, linked or visitors, as a software to support academic activities.

**Keywords:** Android, Georeferencing, Academic Management, UnB

# Sumário

**1 Introdução** **1**
1.1 Objetivos . . . 2
1.2 Metodologia . . . 2
1.3 Estrutura do Trabalho . . . 3

**2 Georreferenciamento** **4**
2.1 Sistemas de Informação Geográfica (SIG) . . . 4
2.2 Ciência de Informação Geográfica (CIG) . . . 6
2.3 SIG Móvel . . . 8
2.4 Sistema de Posicionamento Global (GPS) . . . 10
2.5 OpenStreet Maps (OSM) . . . 11
2.5.1 Funcionamento do OpenStreetMap . . . 13
2.5.2 OpenStreetMap no Brasil . . . 16
2.5.3 Ferramentas OSM . . . 17
2.6 Considerações Finais . . . 18

**3 Tecnologia Envolvida** **19**
3.1 Java . . . 19
3.1.1 Características . . . 20
3.2 XML . . . 23
3.2.1 Modelagem . . . 24
3.3 Android . . . 25
3.3.1 Histórico de Versões . . . 26
3.3.2 Arquitetura . . . 27
3.3.3 Kit de Desenvolvimento . . . 31
3.4 Web Service . . . 32
3.5 Considerações Finais . . . 35

**4 Guia do Aluno UnB** **36**
4.1 Visão Geral . . . 36

4.2 A Universidade de Brasília . . . . . 38
4.3 Levantamento dos Requisitos . . . . . 39
4.4 Funcionalidades do Aplicativo Guia do Aluno UnB . . . . . 44
4.4.1 Disciplinas . . . . . 45
4.4.2 Mapa . . . . . 46
4.4.3 Faltas . . . . . 47
4.4.4 Documentos . . . . . 49
4.4.5 Tarefas . . . . . 50
4.4.6 Telefones . . . . . 50
4.4.7 MatriculaWeb . . . . . 51
4.4.8 Site UnB . . . . . 51
4.5 Considerações Finais . . . . . 53

**5 Resultado** **54**
5.1 Testes . . . . . 54
5.1.1 Testes para Validação da Posição . . . . . 54
5.1.2 Testes de Portabilidade . . . . . 55
5.1.3 Testes para Validação de Dados . . . . . 57
5.1.4 Testes de Acesso a Internet . . . . . 60
5.1.5 Testes de Uso do GPS . . . . . 62
5.2 Validação . . . . . 65
5.3 Considerações Finais . . . . . 67

**6 Conclusão** **68**

**Referências** **70**

# Lista de Figuras


2.1 Sistema de Informação Geográfica (SIG)[1]. . . . . 5
2.2 Mapeamento Automatizado em CIG[2]. . . . . 7
2.3 Sistema de Posicionamento Global (GPS)[3]. . . . . 10
2.4 Modelo do Mapa OSM [4]. . . . . 12
2.5 Estrutura Neo4j Spatial do Mapa [5]. . . . . 14
2.6 Gráfico de Uso de Pontos do OpenStreetMap no Brasil [6]. . . . . 17

3.1 Exemplo de Código Java [7]. . . . . 20
3.2 Funcionamento de um *Activity* em *Java* [8]. . . . . 22
3.3 Exemplo de Código XML com DTD [9]. . . . . 25
3.4 Logotipo Android criada por Irina Blok [10]. . . . . 27
3.5 Pilha de Software Android adaptado de [11]. . . . . 29
3.6 Exemplo de Arquivo JSON. . . . . 32
3.7 Exemplo de *Web Service*. . . . . 33

4.1 Tela Inicial do Aplicativo. . . . . 37
4.2 Respostas da Pergunta: "Você Possui Algum *Smartphone/Tablet*? Se Sim, Você Sabe Qual é o Sistema Operacional Dele?". . . . . 41
4.3 Respostas da Pergunta: "Você é aluno (a)/professor (a) ou já foi aluno (a)/professor (a) da Universidade de Brasília, campus Darcy Ribeiro?". . . 42
4.4 Respostas da Pergunta: "Ao tentar localizar uma sala de aula pela primeira vez, você já sentiu dificuldades em encontrá-la?". . . . . 42
4.5 Respostas da Pergunta: "Com que frequência você frequenta a Universidade de Brasília, campus Darcy Ribeiro?". . . . . 43
4.6 Respostas da Pergunta: "Quantas vezes você já se perdeu na Universidade de Brasília Campus Darcy Ribeiro?". . . . . 44
4.7 Respostas da Pergunta: "Você utilizaria um aplicativo que contém suas informações acadêmicas (horários, disciplinas, documentos) e um GPS que o guiasse a endereços na UnB?". . . . . 44
4.8 Telas do Menu de Disciplinas. . . . . 45

4.9 Telas da Função Mapa. . . . . 46
4.10 Tela Principal do Menu Faltas. . . . . 48
4.11 Itens do Menu Faltas. . . . . 48
4.12 Tela Inicial do Menu Documentos. . . . . 49
4.13 Tela do Menu Tarefas. . . . . 50
4.14 Imagem da Função Telefones. . . . . 51
4.15 Imagem da Função MatrículaWeb. . . . . 52
4.16 Imagem da Função Site UnB. . . . . 52

5.1 Imagens de *Smartphones* Usando o Aplicativo. . . . . 56
5.2 Telas Mostrando a Validação de Dados na Função Faltas. . . . . 58
5.3 Tela Exibindo a Validação de Dados na Função Mapa. . . . . 59
5.4 Tela Exibindo a Validação de Dados na Função Tarefas. . . . . 60
5.5 Tela Exibindo Mensagens de Erro na Função Disciplinas e Mapa. . . . . 61
5.6 Telas Mostrando Processos de Verificação do Uso do GPS. . . . . 62
5.7 Imagem Mostrando a Precisão do GPS. . . . . 63
5.8 Imagem Mostrando o Modo Bússola do GPS. . . . . 64
5.9 Imagem de *Status* do Aplicativo Guia do Aluno UnB [12]. . . . . 65
5.10 Exemplo de Comentários e Avaliações [12]. . . . . 66
5.11 Gráfico com Todas as Avaliações do Aplicativo Guia do Aluno UnB [12]. . 67

# Lista de Tabelas

3.1 Versões do Android [13]. . . . . 28

4.1 Tabela com as Posições das Instalações da Universidade de Brasília - Campus Darcy Ribeiro. . . . . 40

4.2 Tabela com as Posições das Instalações da Universidade de Brasília - Campus Darcy Ribeiro (continuação). . . . . 41

# Capítulo 1

# Introdução

Nas últimas décadas, o rápido crescimento dos centros urbanos e a quantidade de informação produzida em escala resultaram em dificuldades de gestão de espaço e informação, em problemas de locomoção e de acesso à informação. Todavia, o avanço tecnológico proporcionou o surgimento de soluções de gestão da informação e de uso do espaço geográfico distribuído de acordo com as necessidades sociais.

Assim, o advento dessas tecnologias consolidaram modificações nos hábitos populacionais, e deram origem a novos padrões de consumo da informação, sua produção e divulgação. Consequentemente, nesse contexto, surge a necessidade de uma participação no âmbito da tecnologia para o estabelecimento do bem social, e essa necessidade intrínseca prevê prioridades que potencializem estratégias, tais como meio de comunicação ágil, facilidade de acesso à informação no espaço geográfico distribuído com auxílio de estratégias eficientes para propagar a informação.

Nesse cenário surgiram os artefatos tecnológicos, tais como a computação móvel que é uma infraestrutura de comunicação sem fio e, com ela, após anos de pesquisa e desenvolvimento, os *smartphones* e aplicativos celulares. Esses dispositivos contam, hoje, com acesso à *Internet* e também localizações geográficas.

A tecnologia móvel aliada ao espaço geográfico é um fenômeno que está em constante crescimento, dado o grande leque de possibilidades que as aplicações podem proporcionar. Com essa evolução, surge a necessidade de monitorar, validar e gerenciar a informação de maneira transparente, para garantir a qualidade e a eficiência da produção de conhecimento epistêmico, pois esse fator é o que difere uma informação útil de uma outra imprófícua. Em suma, a necessidade de acesso aos dados armazenados de forma distribuída em dispositivos móveis, demandou que a gerência da informação construísse metodologias específicas para centralizar a informação, gerir, produzir e disseminar com estratégias e cuidados que preservam sua qualidade diante de uma sociedade que a produz em grande quantidade [14].

Diante do exposto, este trabalho propõe o aplicativo Guia do Aluno UnB, que tem como objetivo ser uma solução, no âmbito de aplicação móvel, para problemas acadêmicos que a Universidade de Brasília (UnB) apresenta. Assim o Guia do Aluno UnB é uma ferramenta gratuita que tem como função facilitar a vida da comunidade acadêmica, em especial, dos alunos.

## 1.1 Objetivos

O objetivo geral deste trabalho é desenvolver um aplicativo móvel georreferenciado para apoio a atividade acadêmica. Dentre suas funcionalidades, o software deve apresentar um conjunto de ferramentas facilitadoras e apoiadoras às atividades acadêmicas. Assim, para cumprir o objetivo geral deste trabalho, foram definidos os seguintes objetivos específicos:

- Mapear os prédios da Universidade e seus respectivos departamentos, secretarias, serviços e salas de aula;
- Desenvolver um mapa que propicie acesso fácil e direto às localidades da Universidade de maneira intuitiva;
- Criar um acervo de informação pública e de contatos dos principais órgãos da Universidade para aprimorar a comunicação interna;
- Implementar um web service que filtre as disciplinas ofertadas pela Universidade e guiar o aluno até sua respectiva sala de aula;
- Criar um sistema gestor de falta e de tarefas como apoio a atividades acadêmicas;
- Elaborar mecanismos de localização de serviços prestados pela Universidade ou por terceiro;
- Aprimorar processos e facilitar acesso à informação na Universidade de Brasília.

## 1.2 Metodologia

O trabalho foi dividido em etapas, as quais foram baseadas na metodologia *Scrum* - um processo de desenvolvimento iterativo e incremental de forma ágil. Assim, a primeira etapa consistiu na problematização, no levantamento de dados e na elaboração de planos de atuação. A segunda etapa concentrou-se em escolher e definir a arquitetura do *software* e as ferramentas utilizadas, o estabelecimento de datas para o ciclo de entregas e os testes do *software*, realizada em reuniões constantes e embates sobre melhores tecnologias do mercado.

A terceira etapa consistiu no desenvolvimento da aplicação, na modelagem e na implementação dos módulos de teste. Além disso, foi realizada nesta etapa o envio de versões para testes e avaliação do público alvo, nos eventuais ajustes e correção de falhas.

A quarta e última etapa consistiu na análise de desempenho com seus respectivos relatórios, e na escrita desta monografia. Nesta última etapa, fez-se também a publicação do *software* e a disponibilização para os usuários finais. A publicação do *software* foi feita em parceria com o Departamento de Ciência da Computação da Universidade de Brasília.

## 1.3 Estrutura do Trabalho

Este trabalho está dividido, além deste capítulo introdutório, nos seguintes capítulos:

- Capítulo 2: Este capítulo está associado ao georrefrenciamento, que foi usado amplamente no desenvolvimento deste trabalho;
- Capítulo 3: Apresenta a computação móvel e as definições da arquitetura usada no desenvolvimento do *software*, indicando o sistema operacional que foi escolhido, a modelagem do banco de dados da aplicação, a escolha da arquitetura do mapa, as bibliotecas e as aplicações das funcionalidades. Além disso, este capítulo apresenta a definição e a arquitetura de um *webservice*, e a maneira de obtenção dos dados;
- Capítulo 4: Este capítulo apresenta o aplicativo Guia do Aluno UnB, sua confecção e desenvolvimento, demonstrando suas funcionalidades, aplicações e a disposição das telas e dos recursos que a aplicação móvel proporciona;
- Capítulo 5: Neste capítulo são apresentados os resultados do desenvolvimento do aplicativo, metodologias usadas para o teste, validação do aplicativo Guia do Aluno UnB e considerações sobre a eficiência do *software*, além de um panorama geral de uso;
- Capítulo 6: Este capítulo apresenta as conclusões e alguns trabalhos futuros a serem desenvolvidos acerca do tema;

# Capítulo 2

# Georreferenciamento

O objetivo deste capítulo é apresentar, inicialmente, os conceitos de Georreferenciamento e salientar suas particularidades e características, que são pontos primordais para o entendimento deste trabalho. Em seguida, na Seção 2.1, serão detalhados os conceitos de Sistemas de Informação Geográfica (SIG), tais como o surgimento da ideia e os conceitos da arquitetura desses sistemas. Logo após, tem-se a Seção 2.2, na qual se abordará Ciência de Informação Geográfica (CIG), suas particularidades e características atreladas a Georreferenciamento e Sistemas de Informação Geográfica.

Na Seção 2.3 são apresentados alguns conceitos básicos sobre o SIG Móvel, uma derivação de um Sistema de Informação Geográfica convencional, mas com características singulares e imprescindíveis para o entendimento da disposição dos dados geográficos na atualidade. Em seguida, na Seção 2.4 será apresentado o conceito de Sistema de Posicionamento Global (GPS) como advento que revolucionou o entendimento do espaço Geográfico. Por fim, na Seção 2.5 há considerações finais sobre os assuntos abordados e uma síntese geral acerca de Georreferenciamento.

## 2.1 Sistemas de Informação Geográfica (SIG)

Sistema de Informação Geográfica (SIG) representa sistemas de captura, de armazenamento, de manipulação, de recuperação e análise de dados que estão referenciados geograficamente. Todo fenômeno geográfico contém localização associada a uma posição na superfície, como por exemplo a localização de um prédio que pode ser representada pela longitude e latitude [15]. Dentre as características de um SIG há a facilidade de manipular dados geográficos convencionais para análise e consulta integrada e, portanto, ter acesso às informações descritivas a partir da localização desse fenômeno geográfico.

Dessa forma, é possível classificá-los em dados convencionais e dados espaciais, sendo os respectivos aspectos que caracterizam dados georreferenciados [16]:

- Descrição: uma descrição geral do fenômeno geográfico;
- Posição: todo fenômeno geográfico acompanha uma localização;
- Relações: sua relação espacial com outros fenômenos;
- Tempo: intervalo de tempo de existência e validez do fenômeno.

Esses dados, além de apresentarem descrição, posicionamento, relação com outros dados espaciais e tempo de validez, também são classificados em categorias de dados, no qual o dado convencional é para armazenar dados descritivos e temporais com atributos. Os dados espaciais descrevem a localização geográfica e os relacionamentos espaciais. É possível que - fora os tipos padrão de dados convencionais e espaciais - o SIG possa apresentar dados pictórios, que são dados que armazenam imagens sobre regiões geográficas, ou seja, imagem propriamente ditas, normalmente, de satélites [17].

Por outro lado, os SGBDs (Sistema de Gerenciamento de Banco de Dados) são parte fundamental para o gerenciamento de bancos de dados geográficos dos SIGs, pois para encontrar soluções adequadas, no âmbito de gerenciamento de dados georreferenciados, adotou-se a arquitetura integrada, que prevê que o SIG armazene dados espaciais em um SGBD. Em síntese, é afirmar a necessidade de precaver que os dados sejam armazenados de maneira desorganizada e prover condições para seu catálogo e eficiência no acesso.

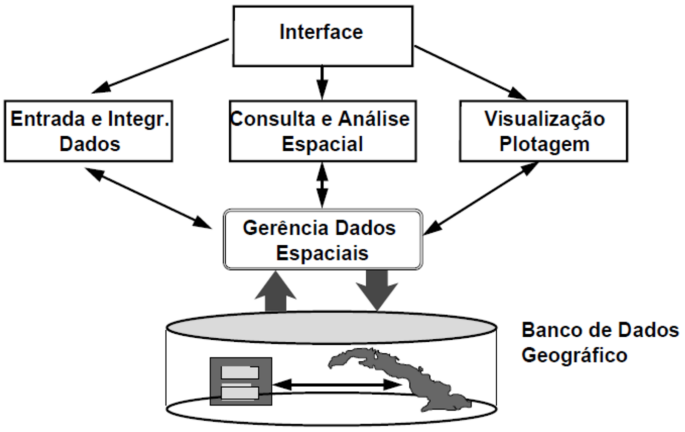


Figura 2.1: Sistema de Informação Geográfica (SIG)[1].

A vantagem da arquitetura integrada em armazenar dados espaciais em um SGBD, conforme apresentado na Figura 2.1, é a utilização dos recursos do sistema gerenciador que proporciona maior controle diante da manipulação de dados espaciais. Dentre esse mecanismo de controle, o controle de integridade e de concorrência, e a gerência de transações se apresentam como principais recursos. Para exemplificar a integração de um Sistema de Informação Geográfica, tais componentes são relevantes [18]: Sistemas de Gerenciamento

de Banco de Dados Geográficos - BDGeo, que é utilizado para obtenção e exibição de dados com representação gráfica; a interface que é a interação do usuário com o sistema computacional; e também a Entrada e Integração de Dados, que é um mecanismo de conversão de dados e consulta para análise espacial.

Para análise de dados espaciais é utilizado algoritmos com operações topológicas com uso de grafos matemáticos e álgebra de mapas com amparo da estatística espacial, ou seja, uma coleção de estudos matemáticos para formação de dados espaciais e sua análise, no qual, geralmente, a maioria dos dados de uma aplicação SIG é apresentada em forma de mapas. Mapa é a maneira mais convencional de se interpretar esses dados.

Dessa maneira, um SIG provê funções equivalentes e às vezes idênticas a cartogramas, pois há a possibilidade de colocar rótulos, legendas e escalas gráfica graças ao avanço da tecnologia de sistemas de informação que despertou possibilidades para visualizar dados geográficos de forma mais fácil e intuitiva. Esse advento de evolução da tecnologia fomentou uma área de pesquisa chamada Ciência da Informação Geográfica [19], a qual será descrita na próxima seção.

## 2.2 Ciência de Informação Geográfica (CIG)

O tema de pesquisa Ciência da Informação Geográfica (CIG), supracitado neste trabalho, surgiu após o advento do SIG - Sistema de Informação Geográfica. Com o advento dos SIG's, o campo de estudo da Geografia elevou a riqueza de detalhes e o quantitativo de pesquisas realizadas no âmbito do espaço geográfico. De acordo com Goodchild [20], problemas técnicos relacionados a construção de um SIG preocuparam os usuários e a necessidade de soluções para esses respectivos problemas tornou-se necessária.

A CIG está presente de maneira intrínseca na Computação, Geografia e Estatística, nas quais trabalham em conjunto. Por essa relevância foi construído um *software*, um *framework* conceitual desenvolvido e mantido pelo *National Center for Geographic Information and Analysis* (NCGIA) [21], uma instituição de pesquisa independente, que relaciona a computação, a sociedade e o indivíduo. Portanto, CIG é uma ciência multidisciplinar que aborda o espaço geográfico e o analisa.

Assim, dados geográficos digitais são o objeto de estudo da Ciência da Informação Geográfica. Eles são representados por caracteres e números e indicam, por exemplo, a localização e os atributos de um espaço geográfico. Segundo Goodchild [22], existem alguns tipos de propriedades desses dados, sendo elas:

- O espaço geográfico: sistema de coordenadas geográficas de latitudes e longitudes são usados para definir posições, embora imprecisões possam acontecer pelo formato

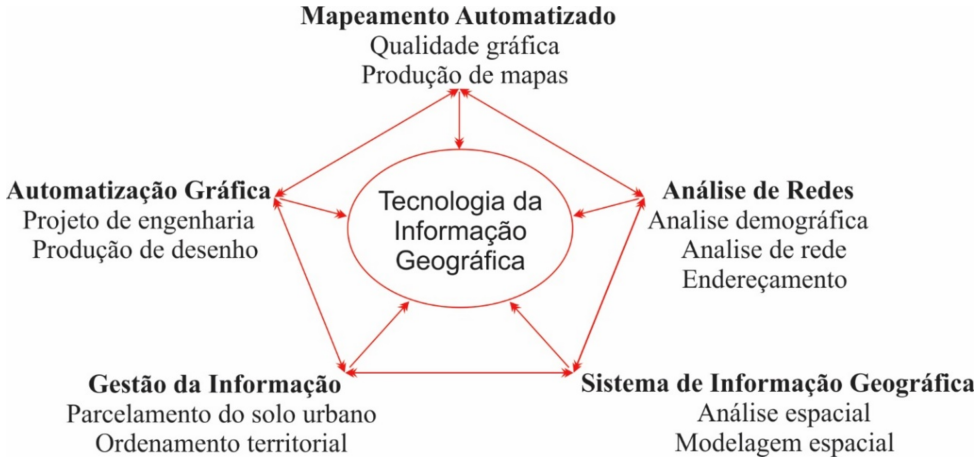


Figura 2.2: Mapeamento Automatizado em CIG[2].

esférico irregular da Terra. Em caso de elevação, visto que os terrenos são irregulares, é necessário que os pontos estejam com localizações dispostas ao formato correto;

- Espaço geográfico contínuo: um dado espacial contém uma chave espacial em duas dimensões contínuas, sem interrupções na superfície terrestre. Dessa maneira, pode-se aferir que todas as posições terrestres podem ser informadas por um sistema de coordenadas, ou seja, uma entidade geográfica pode assumir qualquer quantitativo de pontos, e essa informação é seletiva e aproximada;

- Dados geográficos: localização espacial e atributos não espaciais numa base de dados espaciais têm duas chaves, cujos registros são acessados por atributos ou localização. A chave espacial permite a definição de operações não incluídas no padrão SQL;

- Dados geográficos espacialmente dependentes: propriedade que um local tem de influenciar ao outro semelhante, chamada de dependência espacial. Essa definição está baseada na Primeira Lei da Geografia de Waldo Tobler [23], que afirma que tudo há relação com tudo, e que objetos próximos estão mais relacionados.

De acordo com o *Open Geospatial Consortium* (OGC)[24], consórcio internacional de agências governamentais e universidades, para garantir interoperabilidade de soluções de geotecnologias, há um modelo de referência que agrupam modelos geográficos em discreto e contínuo. No caso dos discretos, eles são objetos facilmente reconhecíveis e bem definidos (rua, prédio, avenida), enquanto os fenômenos não tem dimensão definida e são relativos (calor, umidade). Portanto, os fenômenos discretos são dados vetoriais e os contínuos são dados matriciais e esse é o conceito de *raster data* [25].

Os dados vetoriais são um conjunto de primitivas geométricas (pontos, curvas, superfícies) com seus valores de atributos, enquanto dados matriciais são variáveis e instáveis,

tratados com um *array* regular de pontos ou células de elementos quadrados, chamados de *pixels* onde cada elemento representa um conjunto de valores. A forma de interpolação mais famosa de se obter dados matriciais é a partir de satélites de sensoriamento remoto, que capturam imagens que são avaliadas em uma estação [26].

## 2.3 SIG Móvel

A integração entre as técnologias de SIG e computação móvel conseguem criar respostas de grande aproveito, permitindo acesso às informações espaciais, bem como funcionalidades que o SIG oferece, em qualquer lugar. Essa integração é chamada de Sistemas de Informação Geográfica Móvel, ou SIG Móvel [27].

Segundo Longley *et al.* [28], existem as seguintes partes de um SIG:

- Rede Computacional: conjunto de computadores capazes de partilhar recursos e informações, interligados por um sistema de comunicação. É ele que possibilita a comunicação e a troca de informações digitais;
- Hardware: parte física, ou seja, é o conceito de hardware engloba todos os dispositivos físicos e equipamentos que processam informações. Nesse contexto, é um dispositivo cliente conectado por rede a um servidor;
- Software: parte lógica, ou seja, é o conjunto de instruções e de dados processados pelo hardware. Nesse contexto, pode ser projetado para executar em um Computador Pessoal (PC) ou empresarial;
- Banco de Dados: base de dados definida como representação digital de características geográficas.

Com o advento da Internet, por meio da interação entre cliente-servidor e hardware-software, surgiu o conceito de SIG Móvel pela idéia de sistema de informação distribuído. O modelo mais simples de SIG Móvel, portanto, apresenta um servidor e um cliente, e esse processamento de dados pode ser efetuado, por exemplo, por uma aplicação móvel. Cada serviço ou servidor tem uma URL específica acessível aos clientes, utilizando um protocolo HTTP [29] para enviar requisições ao servidor, e funcionando no esquema requisição-resposta. O servidor tem diversas maneiras de responder ao cliente, tais como os formatos HTML [30], XML [30], ou JSON [30]. Esses formatos são maneiras distintas de se representar respostas de dados.

Visto que um SIG móvel está completamente associado a *Web*, Sun e Fu [31] definiram um *SIGWeb* como um SIG que usa tecnologias *Web* para comunicação entre componentes.

Em sua definição e aspectos da topologia, o funcionamento de um ambiente móvel é o efeito da inclusão da mobilidade da computação móvel em sistemas de banco de dados.

Com baixos custos de aquisição e eficiente poder de processamento, a produção e a venda de *smartphones* alavancaram, e isso foi um fator positivo para a progressão de SIG móveis, que emergiram na década de 1990, ainda que a maioria desses dispositivos operassem sem acesso a *Internet*. O uso de informação geoespacial é essencial em casos como escolher uma melhor rota para determinado local, procurar locais por perto, em casos de serviços de emergência, e maximiza o uso de dados em papel (mapas, tutoriais, cartões de endereço).

Para usar esse tipo de serviço manipula-se dados geográficos. Assim as grandes vantagens do uso de SIG Móvel é a mobilidade de dispositivos não prejudicados por cabos, o grande volume de usuários e o reconhecimento de localização pela versatilidade de comunicação, e quantidade de recursos de *smartphones* modernos apoiados por informação em tempo real. Totavia, eles apresentam a desvantagem de não serem plataformas robustas para grandes bancos de dados pelo baixo poder de armazenamento, e no caso de alguns dispositivos não terem muito poder de processamento quanto a processamento de imagens de grande porte.

Segundo Longley [32], há diferentes tipos de arquitetura de sistemas computacionais que podem ser usadas na construção de um SIG. Dessa forma, as principais arquiteturas são:

- *Desktop*: projetada para atender apenas um usuário, as camadas da arquitetura SIG clássica são implementadas em um mesmo componente do hardware, normalmente, um computador pessoal;
- Cliente-Servidor: projetada para ambientes de trabalho, as camadas são instaladas em múltiplas máquinas por performance e flexibilidade, além de atender maior quantidade de usuários;
- *Desktop* Centralizado: usa-se uma arquitetura *thin client*, que significa uma rede altamente independente de um servidor, pois os dados se encontram nos computadores pessoais *desktop*.
- Servidor Centralizado: mais comum em meio corporativo, essa variação é implementada como um sistema servidor real, e executa em um servidor da camada de aplicação. Esses servidores podem estar conectados por uma rede local, mas aplicações de SIG Móvel e *Web* são bastante utilizadas nesse meio.

## 2.4 Sistema de Posicionamento Global (GPS)

O GPS (Sistema de Posicionamento Global) é uma ferramenta fundamental na computação móvel e está presente em grande parte dos celulares modernos, embutido em seu sistema operacional. O GPS é um sistema de navegação por satélite que consegue fornecer informação horária e posição, permitindo que uma pessoa com um receptor terrestre descubra uma determinada localização em qualquer lugar da terra utilizando três dimensões, que são: latitude, longitude e altura [33]. O surgimento do GPS remota a 1972, quando foi criado um sistema eletrônico de informação pelo exército norte-americano e desde 1995, pessoas ao redor do globo podem utilizar tal tecnologia.

Em princípio, o sinal GPS para uso civil era limitado e por conta disso, pouco utilizado. Entretanto, o GPS começou a ser utilizado pela sociedade e sua venda foi instigada com o advento da evolução da tecnologia móvel, mas seu uso não era satisfatório visto alguns empecilhos dessa mesma tecnologia, tais como posições inexatas, lentidão, custo e manuseio [34]. Segundo Richard J. Lisle [34] não se assume que o telefone celular funciona em quaisquer condições, mesmo subterrâneo ou em um local que o sinal GPS é fraco, e isso deve-se ao fato de que a cobertura do sinal GPS depende da localização, como por exemplo um usuário ter um sinal GPS forte o suficiente para usar a aplicação, e não ter a mesma intensidade de sinal em um local mais próximo.

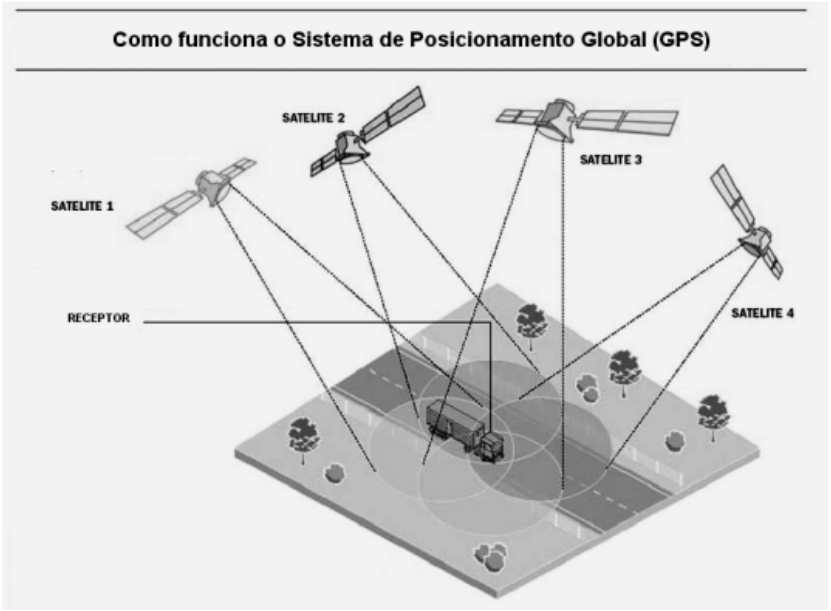


Figura 2.3: Sistema de Posicionamento Global (GPS)[3].

O GPS pode ser divido em diferentes partes, as quais são: a espacial, a controladora e a utilizadora. O sistema trabalha por meio dos satélites, que ficam orbitando ao redor da

Terra no espaço sideral, emitem sinais contendo diversas informações, como coordenadas geográficas. O controle destes satélites é feito por estações terrestres localizadas próximo à linha do Equador. Esta movimentação é a parte que denominamos controladora do processo. As estações terrestres da Zona Equatorial são encarregadas de manter os satélites orbitando corretamente, além de controlar seus relógios e os dados que eles emitem, conforme apresentado na Figura 2.3.

O funcionamento da parte utilizadora é realizada por um receptor GPS que capta os sinais emitidos pelos satélites. Assim sendo, o sistema de GPS funciona através de um aparelho receptor, no caso de aplicações móveis poderia ser representada pelo celular, através de sinais emitidos por satélites que ficam orbitando o planeta Terra. Para realizar esta tarefa, cada um destes satélites circula ao redor do planeta Terra duas vezes por dia.

Após o recebimento destas informações receptadas por sinais, o aparelho faz a leitura deste conteúdo e o cálculo das coordenadas para nos indicar o lugar em que estamos ou onde queremos ir, por exemplo. Pelo menos quatro satélites são utilizados simultaneamente na geração de cada uma destas informações, o que ocasiona em maior precisão e agilidade [35]. Para que o usuário entenda este conteúdo que é repassado pelo GPS, são necessários softwares e/ou banco de dados onde, nesse caso, o softwares possua uma API integrada que permite às aplicações guiarem os usuários através desses dados [36].

## 2.5 OpenStreet Maps (OSM)

Um dos problemas a serem enfrentados acerca de computação móvel e localizações geográficas é a exatidão posicional dos dados geográficos e sua real localização espacial. Para enfrentar a ausência de dados concretos ou a ambiguidade desses, aplicações modernas utilizam uma nova forma de informação geográfica, designada de Informação Geográfica Voluntária (IGV) [37].

A IGV representa um recurso crescente de dados potencialmente valiosos para muitas aplicações [38], no qual os usuários são responsáveis por produzir e publicar essa informação. A IGV, segundo Goodchild [39], é um fenômeno que rompe com todas as formas mais convencionais de informação geográfica, enfatizando também o esforço dos voluntários que a recolhem, a processam e a publicam.

O OpenStreetMap (OSM)[40] é um exemplo de IGV, e consiste num projeto colaborativo no qual os voluntários recolhem, processam e publicam dados geográficos de diverso tipo, como cartografia, imagens, atributos descritivos, entre outros. É uma fonte de dados geográficos de acesso livre, com a representação cartográfica de diversas entidades geográficas. Um dos pontos chave a serem tratados com IGV é a qualidade dos dados produzidos [38]. Alguns autores tem avaliado a exatidão da IGV comparando com outras

fontes de dados usados como referência, e também com mapas disponíveis na Internet, tal como o Google Maps [41] e o Nutiteq [41].

Contudo, nota-se que a cobertura dos dados OSM apresenta-se variável em função da região, sendo normalmente nas zonas urbanas, onde a população é maior que existe uma melhor completude. Em um estudo previamente realizado na Universidade de Brasília usando a aplicação móvel, foi analisada a exatidão posicional das rotas traçadas e as posições no mapa, tendo-se concluído que o uso das rotas do OSM possuem uma maior exatidão posicional obtidos a partir da integração de dados do OSM com dados extraídos com maior exatidão posicional que apenas os dados originais [42]. A Figura 2.4 representa como esse mapa se dispõe e como é a persistência dos dados.

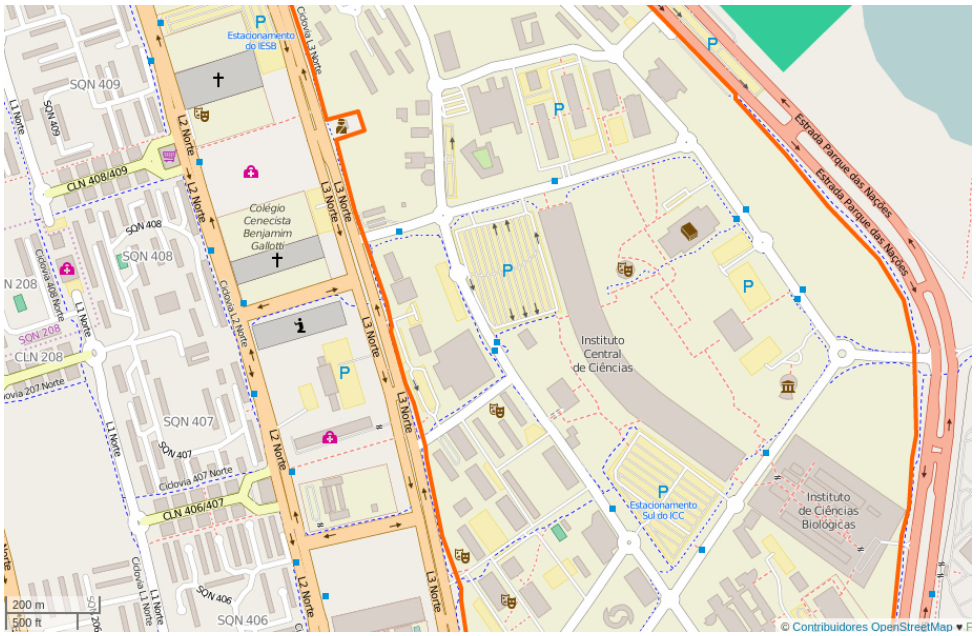


Figura 2.4: Modelo do Mapa OSM [4].

Nas últimas décadas, a produção da geoinformação sofreu mudanças devido à difusão das tecnologias computacionais. Essas tecnologias permitiram a disseminação das informações geoespaciais, bem como fizeram com que muitos indivíduos fossem conduzidos a uma nova perspectiva de uso e produção de mapas. Nesse sentido, o OpenStreetMaps foi desenvolvido não só na intenção de fornecer suporte de conhecimento para os pesquisadores engajados no entendimento das novas tendências na produção da geoinformação, bem como àqueles que propõem-se a compreender os problemas relacionados a qualidade de dados espaciais no contexto dos usuários produtores da geoinformação, com adendo a tecnologias atuais e verificando sua veracidade [40].

Para alcançar tal objetivo, pode-se dizer que, num primeiro momento, fez-se uma breve revisão historiográfica, na qual apresenta-se os conceitos que permitiram o surgimento

dos sistemas de informações geográficas voluntárias. A partir desta contextualização, relaciona-se esta temática ao problema da avaliação da qualidade da informação espacial, o qual tem travado, por exemplo, a utilização das informações geográficas voluntárias em órgãos oficiais de mapeamento, embora tenha se aplicado com perfeição diante de desafios sociais tais como o OpenStreetMaps.

Isso deve ocorrer porque os modelos de avaliação utilizados não contemplam as necessidades dos sistemas de informações geográficas voluntárias e, além disso, ressalta-se a importância do envolvimento da comunidade científica na solução de tais impasses, para que seja viabilizado o acesso à informações de qualidade real, ao invés de desmedidas colaborações da comunidade.

### 2.5.1 Funcionamento do OpenStreetMap

A informação é recolhida por muitos participantes, organizada em uma base de dados central, e distribuída em vários formatos digitais através da *World Wide Web* [43]. Além de ser *Open Source*, o Open Street Map tem várias APIs, auxiliando no acesso do mapa completo que está em seus servidores principais. Também pode ser feita a exportação de pequenos fragmentos do mapa. A API que disponibiliza tal serviço é uma aplicação web. Na página da API é possível selecionar uma área reduzida oferecendo quatro coordenadas geográficas que representarão a cidade que foi mapeada dentro dessa zona selecionada.

Para realizar o download do mapa é necessário apenas selecionar o botão de Exportar. Assim, será gerado um arquivo (*.osm*) reconhecido pelo *plugin* Neo4j-Spatial e de fácil importação para o banco de dados. Os mapas que foram importados pelo software tem uma estrutura complexa, formada por vários nós e relacionamentos que definem como um mapa real é representado computacionalmente. Essa estrutura pode ser representada graficamente pela Figura 2.5.

De acordo com o site oficial do OpenStreetMaps, essa estrutura é composta pelos relacionamentos [5]:

- OSM representa a relação entre o nó principal e o resto da estrutura;
- SPATIAL é a relação responsável por armazenar as informações espaciais inseridas no banco;
- LAYER é a camada na qual todos os NODES são inseridos;
- RTREE_METADATA armazena as informações do nó inicial e final de cada *layer*;
- RTREE_ROOT é a informação que representa a região inteira composta por RTREE _REFERENCE;

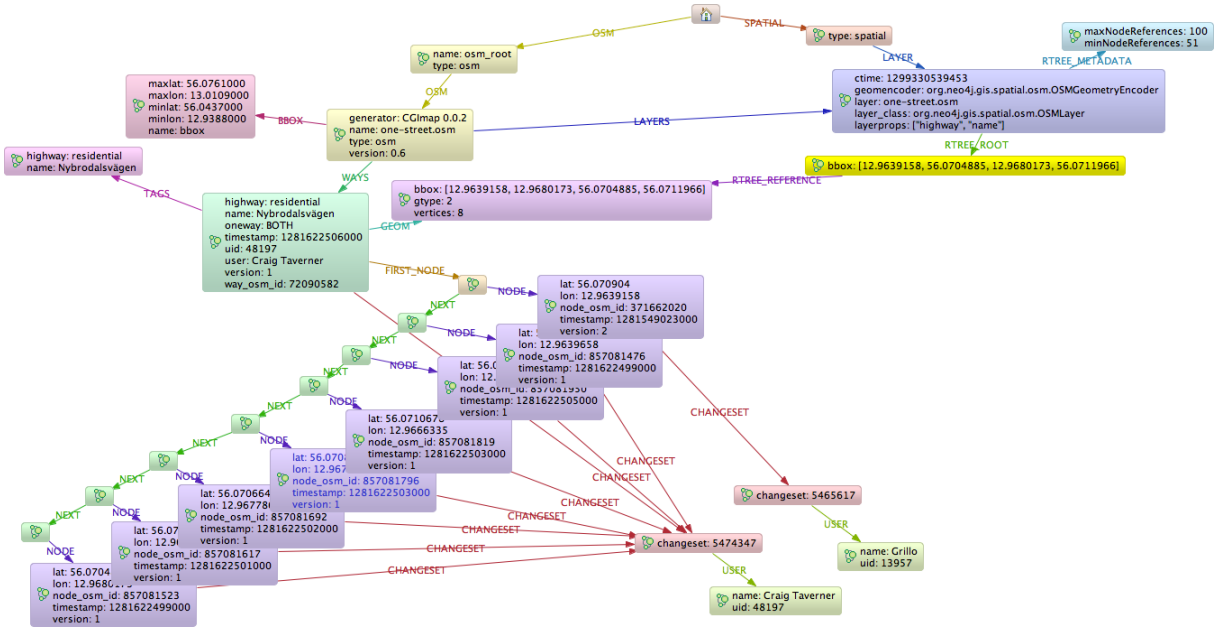


Figura 2.5: Estrutura Neo4j Spatial do Mapa [5].

- RTREE_REFERENCE é a referência para o inicio de uma arvore de nós que juntos representam uma rua, avenida etc;
- LAYERS é o inicio de uma árvore de nós que representam todos os *layers* do banco de dados.
- WAYS marcam o inicio de uma região;
- BBOX marca toda a área que o mapa *OSM* importado abrange;
- GEOM contém as informações geográficas do início e do fim de uma rua, avenida etc;
- TAGS contém as informações da região, se é residencial, comercial etc;
- FIRST_NODE marca o inicio de uma rua, avenida etc;
- NODE é um ponto geográfico propriamente dito no mapa, representando pontos de ruas e avenidas como cruzamentos, etc;
- NEXT liga um NODE a outro.

Diante do exposto, para compreender como o OpenStreetMaps funciona é necessário compreender que o termo geoprocessamento denota a disciplina do conhecimento que utiliza técnicas matemáticas e computacionais para o tratamento da informação geográfica,

influenciando de maneira crescente as áreas de Cartografia, Análise Transportes, Comunicações, Planejamento Urbano e Regional. As ferramentas computacionais para geoprocessamento, chamadas de Sistemas de Informação Geográfica (SIG), realizam análises complexas ao integrar dados de diversas fontes e ao criar bancos de dados georreferenciados, tornando ainda possível a produção automática de documentos cartográficos, como no caso do OpenStreetMaps.

Além de fornecer acesso aos dados visto, é necessário oferecer uma consulta a um metadado de um serviço de geoprocessamento para a exibição do mapa, ou seja, das entidades baseadas em coordenadas que o OpenStreetMaps utiliza. Para garantir acesso de forma distribuída há a necessidade de construir um *Web Services* de serviço independente, e trabalho colaborativo no qual componentes projetados para um dado serviço podem ser conectados para compor um serviço maior, de acordo com Doyle [44]. Já Lamas [45] propõe um SIG sensível a dispositivos móveis para auxiliar usuários a localizar eventos e setores de uma localidade. Para tal, há a necessidade de modelos baseados em iterações assistidas em ambos os casos visto a delicadeza dessas informações e a necessidade de atualizá-las com frequência [45].

O projeto OpenStreetMaps, datado das práticas cartográficas digitais, ou neogeografia, tornou-se, graças a disponibilidade de dados públicos de projetos colaborativos, o que foi replicado por pensadores diferentes como Goodchild [46]. Um campo de estudo inovador no âmbito da Geografia. Inclusive, de acordo com Gorman [47] existem tendências a neogeografia, que é a divulgação de dados geográficos por parte de entidades públicas a projetos colaborativos como OpenStreetMap, onde aplicações baseadas na geolocalização dependem intrinsicamente do volume de dados disponível.

Além disso, o OpenStreetMaps tomou como parâmetro o desenvolvimento de software livre, que são coordenados pela fundação *OSGeo (Open Source Geospatial Foundation)* [48], uma fundação criada para desenvolver sistemas de alta qualidade em código aberto para aplicações espaciais, para aproveitar de uma tecnologia conhecida e consolidada que não seja software proprietário o que garante liberdade em aditivos ao sistema.

A referência da API do OpenStreetMap (OSM) faz de uma aplicação móvel um projeto de mapeamento colaborativo que finda objetivo de criar um mapa editável e livre, baseados em mapas alcançados por processamento dos dados sob uma licença *Open Database License* [49].

Como consequência, todos que tenham acesso apenas a um navegador e a uma conexão de Internet, podem fazer o uso desta tecnologia conhecida por *WebGIS* [50]. No âmbito da locomoção, o estudo de variações das distribuições espaciais de incidência de eventos tem sido importante para investigar e sugerir hipóteses de causalidade em várias questões sociais como, por exemplo, meios de transporte [50].

Particularmente, os estudos de agravos em cálculo de rotas e fornecimento de algoritmos de exatidão posicional propicia avanços significativos quanto a locomoção para o OpenStreetMaps. Assim, o processo dinâmico de integração de dados e a construção de mapas, estabelece uma síntese de instrumentos ricos para transporte público, como por exemplo, a localização dos eventos no tempo e no espaço, o monitoramento e o controle de determinado evento, a identificação de áreas geográficas e grupos de população com maiores necessidades, e a estimativa de tempo percorrido no espaço.

Contudo, o uso de ferramentas computacionais em geoprocessamento não é uma tarefa trivial, pois demanda cautela especializada em garantir fidelidade aos dados, além da difículdade de encontrar ferramentas computacionais que levem em consideração o sinergismo existente entre os processos sociais e a maneira comportamental do transporte [51].

Diante do exposto, o OpenStreetMaps propicia um ambiente que oferece facilidade de acesso às informações espaciais, pois detém dados de múltiplos fornecedores. Para o acesso à essas informações, utiliza-se da Infraestrutura Nacional de Dados Espaciais (INDE)[52], que é definida como: "o conjunto integrado de tecnologias, políticas, mecanismos e procedimentos de coordenação e monitoramento; padrões e acordos, necessários para facilitar e ordenar a geração, o armazenamento, o acesso, o compartilhamento, a disseminação e o uso dos dados geoespaciais de origem federal, estadual, distrital e municipal", de acordo com o decreto No 6.666, aplicado em 2008 [53].

### 2.5.2 OpenStreetMap no Brasil

A INDE, Infraestrutura Nacional de Dados Espaciais, é um órgão governamental brasileiro que é mantido pela Comissão Nacional de Cartografia (CONCAR), e dispõe de diversas informações espaciais por meio de metadados MGB (Metadados Geográficos Brasileiro) de acordo com a norma ISO19115:2003 [54], que é administrada por um sistema *open source* chamado *Geonetwork* [55]. Nem sempre essas imagens são capturadas por sensoriamento remoto, necessitando de dados públicos para buscar essas informações. Para o uso desses dados, o OpenStreetMaps carece de mecanismos empíricos para a qualidade dessas informações e remoção dos erros, ainda que o colaborativismo seja solução indagada ao declínio no fornecimento de informações geográficas com recursos governamentais no mundo inteiro segundo Goodchild [56].

Para essa infraestrutura que estuda a movimentação do usuário, é necessário utilizar ontologia no âmbito de Ciência da Computação, com mapeamento de perfis de usuário, que de acordo com Gruber [57] "é uma especificação formal explícita de um conceito compartilhado", modelando os perfis dos usuários com suas informações geográficas. Para

Kokar [58], essa tendência significa um modelo de processos de informações de contexto com base em computador.

Dessa forma, é possível aferir com exatidão o número de pontos no Brasil, de acordo com as informações apresentadas na Figura 2.6.

Figura 2.6: Gráfico de Uso de Pontos do OpenStreetMap no Brasil [6].

Devido ao fato que o OSM possui muitos fornecedores de dados e ferramentas de interface, é possível traçar dados georreferenciados dentro de prédios. Um exemplo disso é a própria Universidade de Brasília, onde através do uso das ferramentas e pela contribuição de usuários, há rotas georreferenciadas sem o uso de automóveis dentro de prédios como Instituto Central de Ciências - ICC, Pavilhão João Calmon - PJC e Pavilhão Anísio Teixeira - PAT.

### 2.5.3 Ferramentas OSM

As ferramentas com infraestrutura de dados comuns são essenciais para manipular os dados do mapa. As novas cartografias digitais, denominadas como *living maps*, de acordo com Aguiton [59] é a ferramenta que permite a aplicação de interfaces dinâmicas em locais georreferenciados, permitindo que alteram a forma como usamos o espaço e nos relacionamentos com trajetórias e localizações, e consequentemente facilitando a busca de caminhos mais precisos.

Para a submissão de objetos, lugares, estabelecimentos e derivados, o OpenStreetMap não restringe as chaves e os valores atribuídos aos elementos (nós, caminhos e relações),

recomendando conjunto de elementos e as suas etiquetas correspondentes. Essas etiquetas são projetadas para usar critérios mensuráveis, minimizando as disputas com objetivo a um mapa livre e padronizado. Para isso, aplica-se uma técnica que traduz endereços em coordenadas, chamada *Geocoding*, cujo projeto do OSM é o *Nominatim* [60], que é uma ferramenta de busca de nomes e endereços pela resolução de nomes a pontos.

O *Nominatim* indexa elementos nomeados ou numerados a partir do conjunto de dados, além de um subconjunto de elementos não nomeados, ou pelo uso de palavras-chave que são traduzidas em etiquetas específicas do OpenStreetMap. Um programa de auxílio, CBERS, é uma parceria entre os governos brasileiro e chinês, para desenvolver tecnologias de sensoriamento remoto, que utiliza o satélite CBERS-2B Satélite, capaz de produzir imagens 2.7m para a América do Sul. *Nominatim* baseia-se no utilitário de importação osm2pgsql, utilizando a opção de saída alternativa *gazetteer*. A Indexação e pesquisa, por sua vez, combinam códigos escritos nas linguagens C e PHP.

Quando a Yahoo confirmou acesso de suas imagens aéreas para produção de mapas, acabou dando origem a um projeto independente *OpenAerialMap*, para guardar a base de dados de imagens aéreas em licença aberta, com amparo do sistema TIGER do governo dos Estados Unidos da América. Sua grande ascensão surgiu quando a Universidade de Oxford passou a usar o OpenStreetMap na sua página. Imagens de satélite *Landsat 7, Prototype Global Shorelines (PGS)* e Sistema TIGER dos EUA são alguns exemplos de fonte de mapas livres. Esses dados de uma licença compatível com a *Creative Commons Attribution-Share Alike* com direitos de autor expirados, domínio público ou outras licenças[61].

## 2.6 Considerações Finais

Neste capítulo foram descritos os conceitos de Georreferenciamento, sua arquitetura e os seus principais aspectos. Além disso, foi apresentado o conceito de Sistemas de Informação Geográfica (SIG) e seu arquétipo de funcionamento, assim como sua corroboração para a criação da Ciência de Informação Geográfica (CIG) e os desafios encontrados na atualidade para atender a mobilidade urbana e rural com o advento do objeto de estudo SIG Móvel.

Entretanto, para obter resultados com esse sistema é necessário um aparato tecnológico, e o seu entendimento quanto arquitetura, conceito e aplicação. Para isso, o Capítulo 3 apresenta conceitos de arquitetura e definições das linguagens de programação utilizadas, a indicação do sistema operacional escolhido, além de apresentar a definição e a arquitetura de um *webservice*.

# Capítulo 3

# Tecnologia Envolvida

O objetivo deste capítulo é apresentar as tecnologias envolvidas na construção desse projeto, suas particularidades e características, que são pontos primordiais para o entendimento deste trabalho. Assim, na Seção 3.1, será apresentada a linguagem de programação *Java*. Após, na Seção 3.2, será apresentado conceitos acerca de *XML*, apoiado pela Subseções 3.2.1 que explora os conceitos da linguagem e da Seção 3.2.2 com sua modelagem. Por fim, na Seção 3.5 há considerações finais sobre os assuntos abordados e uma síntese geral acerca das tecnologias envolvidas.

## 3.1 Java

Em síntese, *Java* é uma linguagem de programação multi-paradigma: baseado em classes, orientada a objetos e concorrente [62]. Foi criada por James Gosling, Mike Sheridan e Patrick Naughton durante o desenvolvimento de um projeto em junho de 1991, com o propósito de atender as necessidades da TV Interativa [63]. Esse mesmo projeto chamava-se *Oak*, por se tratar de uma árvore de carvalho que ficava ao lado do escritório de James Gosling. Em seguida, seu nome foi alterado para *Green* e respectivamente para *Java*, como referência a sementes de café produzidas na ilha da Indonésia.

Em 1995, a empresa Sun Microsystems lançou sua primeira implementação *Java* 1.0, com o propósito de proporcionar na programação um código que, escrito apenas uma vez, pudesse ser executado em qualquer plataforma, demonstrando segurança em sua execução nos navegadores *web* [64]. Por serem executados nas próprias páginas, a linguagem ganhou popularidade, algo que não era esperado pelos desenvolvedores.

Atualmente, *Java* é considerada a linguagem de mais uso no Mundo, dominando 20,846%, segundo a empresa TIOBE[65], que é uma empresa dedicada a qualidade de-software e responsável por criar um indicador de popularidade do uso de linguagens de programação usando como base de pesquisa vários engenheiros.

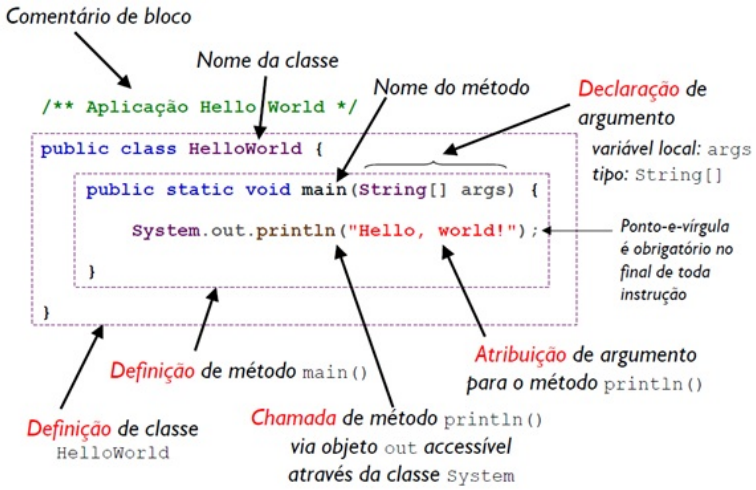


Figura 3.1: Exemplo de Código Java [7].

Para ilustrar sua estrutura, é possível perceber a definição de sua sintaxe na Figura 3.1. A linguagem *Java* é utilizada com bastante frequência para o desenvolvimento e distribuição de aplicações móveis e incorporadas, tais como jogos, conteúdo básico na *web* e *softwares* corporativos. Segundo o site oficial [66], *Java* é usado para criar programas por mais de 9 milhões de desenvolvedores em todo o Mundo e também mais de 3 bilhões de telefones celulares executam o *Java*. Segundo Robert Sebesta [67], *Java* é baseada em C++, porém, foi feita com o propósito de ser menor, mais simples e confiável.

### 3.1.1 Características

A linguagem de programação *Java* tem uma arquitetura robusta e muito sofisticada. Suas principal característica está na possibilidade de executar códigos compilados em *bytecode* em qualquer plataforma que possua a Máquina Virtual *Java*, independente da arquitetura do computador. Isso propicia que a linguagem *Java* seja executada em qualquer máquina. No caso da plataforma virtual, sua arquitetura está sustentada pela facilidade que os *Java Applets* propiciam.*Java Applets* são pequenas aplicações escritas em *Java* e enviadas para os usuários em forma de *bytecode*, executado em uma máquina virtual a parte dos navegadores. Em 2006, a empresa Sun lançou grande parte da máquina virtual *Java* como um *software open-source*, e em maio de 2007, essa mesma empresa finalizou e lançou todo o código da máquina virtual para acesso gratuito ao público, fazendo que idéias como o sistema operacional *Android* viessem a existir.

Linguagens de programação utilizam variáveis e estruturas de dados como maneira eficiente de manipular dados. Objetos em *Java* são parte fundamental de sua arquitetura e são criados por classes, uma abstração da linguagem para separar instâncias. Assim

como C++, *Java* tem mais classes do que primitivos, embora os vetores sejam instâncias de uma classe pré-definida, o que é diferente em C++. *Java* não utiliza ponteiros, mas as referências fornecem parte das propriedades dos ponteiros, no qual todos os objetos são alocados em um monte. *Java* também utiliza um tipo booleano chamado *boolean*, que é uma variável usada principalmente para as várias expressões de controle de suas respectivas sentenças de controle. Outra diferença que *Java* possui relaciona os subprogramas autocontidos e todos os subprogramas são métodos definidos em classes, e esses métodos podem ser chamados somente através de classes ou objetos, ocorrendo que *Java* oferece suporte apenas para programação orientada a objetos, enquanto C++ oferece suporte para programação procedural [68].

De acordo com Sebesta [67], mudanças de tipo de variáveis só ocorrem quando há necessidade de aumentar a tipologia de variáveis. Um exemplo seria mudar o tipo de uma variável inteira para ponto flutuante (*int* para *float*) através de um operador de atribuição. Essas possibilidades diferenciam as exceções da linguagem de programação. O tratamento de exceções do Java é baseado no mesmo da linguagem convencional C++, porém, é projetado para estar mais perto com o paradigma de linguagem orientada a objetos além de que o *Java* inclui uma coletânea de exceções pré-definidas que podem ser levantadas pela máquina virtual que *Java* utiliza. Além esse diferencial, Robert Sebesta [67] também avaliou a linguagem de programação *Java*, pois remover o excesso de recursos inseguros do C++, no qual essa linguagem foi baseada, fez bem para o desenvolvimento de programas feito em *Java*, como por exemplo, a exclusão de metade das coerções de atribuição que existiam em C++, o que gerou uma confiabilidade maior Assim como essas mudanças promoveram confiabilidade, a verificação de índices de acesso a vetores torna a linguagem mais segura e a adição de concorrência melhora o escopo de aplicações que podem ser escritas na linguagem *Java*. Um exemplo está na criação de uma *Activity*, que tem seu funcionamento descrito como na Figura 3.2.

Quanto a atribuição de características aos objetos, *Java* também não oferece herança múltipla, pois pode transformar o código em algo mais complicado e confuso. Dessa forma, a linguagem oferece apenas herança simples de classe, mas é possível usar algumas propriedades da herança múltipla através do uso de interfaces. *Java* também oferece um controle de concorrência por uma forma simples, através do seu modificador *synchronized*, que pode aparecer em métodos e blocos, fazendo com que um bloqueio seja anexado e, assim, garantindo acesso ou execução dos métodos ou blocos. *Java* utiliza o método coleta de lixo para a liberação de armazenamento dos objetos, fazendo que o programador fique livre da necessidade de remover os objetos quando esses não forem mais necessários. Para que não ocorresse a coleta de lixo, aconteceria um vazamento de memória, responsável que o armazenamento seja alocado na memória e não possa ser liberado, levando ao

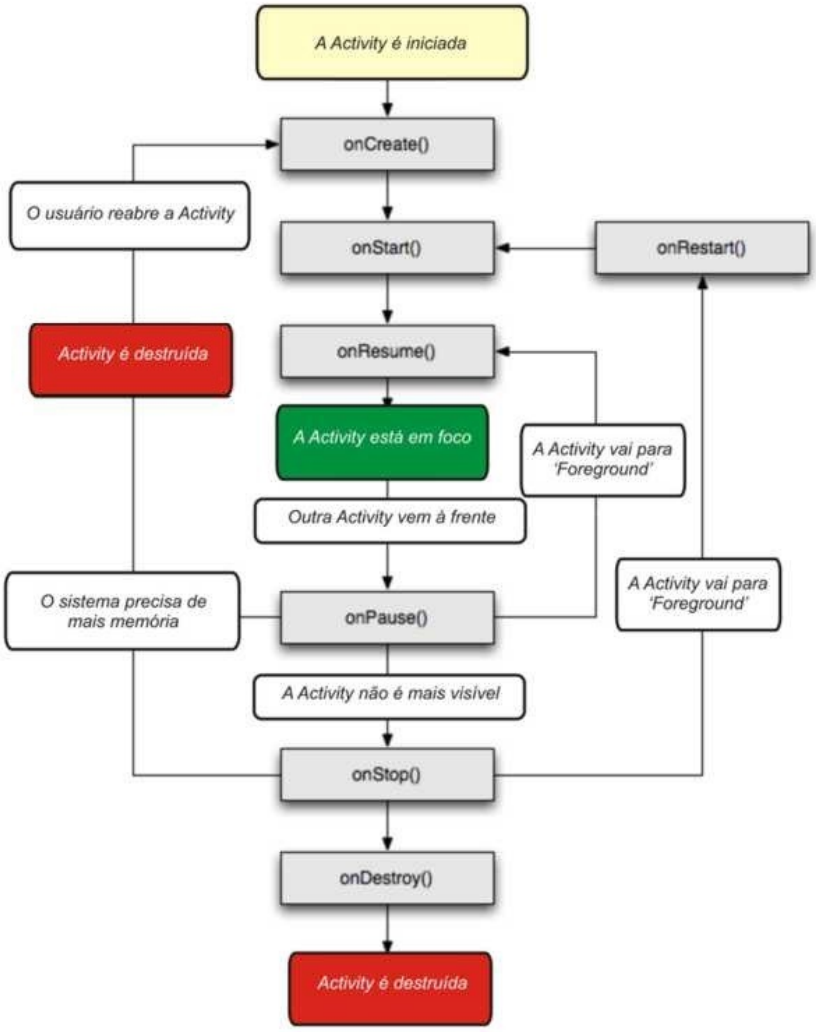


Figura 3.2: Funcionamento de um *Activity* em *Java* [8].

consumo de todo o armazenamento disponível. Essas características fizeram de *Java* uma linguagem portátil e de fácil distribuição.

Sendo assim, Sebesta [67] também relata que a portabilidade do *Java* é atribuída ao projeto da linguagem, que, segundo ele, qualquer linguagem pode ser traduzida para uma forma intermediaria que qualquer máquina virtual possa executar e o preço dessa portabilidade que o *Java* usa é o custo de interpretação. A primeira versão do interpretador que o *Java* utilizava foi conhecido como Máquina Virtual *Java* (*JVM*) e essa conseguia executar ao menos 10 vezes mais lento do que os programas equivalentes em C. Atualmente, muitos programas *Java* são traduzidos para código de máquina antes de serem executados, através dos compiladores *Just-in-Time* (JIT) [67], tornando a eficiência de programas *Java* equivalente com a dos programas escritos em linguagens compiladas como C++ e resolvendo esse quesito.

O suporte do *Java* para concorrência é simples, porém eficaz. Os mecanismos que o *Java* utiliza para fazer o tratamento de exceções são melhores do que a versão C++ utiliza, visto que as facilidades que *Java* possui para tratamento de exceção são a partir da cláusula *throws* e isso corrobora para a legibilidade de um programa. Sebesta [67] também justifica que *Java* se expandiu em relação a qualquer outra linguagem de programação pois esse fenômeno se dá ao valor da programação de documentos *web* dinâmicos, assim como o sistema de compilação e interpretação *Java* serem gratuitos e de fácil acesso na Internet. *Java* também passou a ser uma linguagem poderosa, assim como mais simples e segura, o que a fez escolhida para ser linguagem do sistema operacional *Android*.

## 3.2 XML

A linguagem de programação XML, abreviação de *eXtensible Markup Language* – Linguagem de marcação extensível, é uma linguagem de metamarcação usada para definir regras em um documento que seja legível para humanos e processados por computadores [62]. O objetivo que a linguagem XML atinge enfatiza simplicidade, padronização e usabilidade pela Internet [69] e foi definida na especificação feita pela W3C (*World Wide Web Consortium* – Consórcio da WWW) e por várias outras especificações.

A história da linguagem XML começa em 1970, quando a IBM criou a linguagem *GML (General Markup Language* – linguagem de marcação genérica) [70] com a necessidade de salvar uma grande demanda de informações da empresa. Por conta da atenção que a incorporou em ferramentas, a linguagem XML alcançou um nível muito alto, acima do que um simples publicador de documentos na Internet, chegando a ser escolhida como linguagem de metamarcação da plataforma Android, com o próposito de organizar a interface gráfica de uma aplicação.

Nesse contexto, segundo o artigo de Fernando Hadad [70], as principais características da XML são:

- Troca de dados: uma das principais utilizações da linguagem XML é a troca de dados por conta da grande diversidade de computadores, rodando diferentes sistemas, em diferentes plataformas, sistemas e banco de dados contêm dados em formatos que não são compatíveis. Como o conteúdo básico de documento que utiliza XML é apenas texto, a linguagem apresenta como possibilidade que computadores diferentes troquem dados pela Internet, bastando converter o formato do banco de dados no formato de texto, seguindo as regras e sintaxe estabelecidas nos documentos XML.
- Como a linguagem XML permite a definição dos marcadores e os dados armazenados têm um valor, logo ela permite que linguagens manipulem seu conteúdo, facilitando

a pesquisa, a inclusão, a alteração e a exclusão de dados. Ou seja, a linguagem tem uma grande capacidade de interoperabilidade de dados por possuir um padrão altamente flexível, independente do dispositivo.

- Numa determinada apresentação é possível ligar os dados contidos nos documentos HTML, com os dados contidos nos documentos XML. Assim, a mudança nos dados no arquivo XML vai ser refletido diretamente no HTML, eliminando a necessidade de sua atualização.
- A linguagem XML possui vários tratamentos para criar documentos, tal que diversas ferramentas podem analisar o conteúdo e apresentar um catálogo de informações. Um catálogo que está *online* e foi escrito em XML, ele pode ser considerado uma das partes mais importantes de um *cite*, por exemplo, pode conter informações que os clientes desejam tais como produtos, fotos e preços.
- Os sistemas desenvolvidos a partir da linguagem XML realizam a integração de empresas como por exemplo, a automação do processamento de compra, maior flexibilidade na escolha de fornecedores e mudança na cadeia de suprimentos. Isso se deve ao fato de que o XML define o conteúdo do documento, no qual o XML tem uma grande característica de separar a interface do usuário dos dados.

Em relação à troca de documentos eletrônicos e a mudança de dados em um formato neutro, em uma mesma empresa ou grandes organizações, o uso da XML se tornou a sintaxe preferida para troca de mensagens corporativas nessas organizações [70]. Essa é uma tecnologia que suporta sistemas interorganizacionais e está sendo constantemente usada nos sistemas de parceiros comerciais nos casos em que existe a necessidade de troca de documentos eletrônicos, com o propósito de automatizar os processos de negócio entre duas ou mais organizações.

### 3.2.1 Modelagem

A arquitetura da linguagem XML corresponde ao seu uso em termos de facilidade. A facilidade de uso da linguagem, por ser legível e compreensível, além de apresentar uma arquitetura equivalentemente simples foram pontos fundamentais para ajudar a linguagem a ser popular como é hoje. Além de uma arquitetura sucinta e a possibilidade de criação de analisadores de dados XML, essa possui uma estrutura sintática simples e uma forte carga semântica contida na estrutura hierárquica. A linguagem XML é uma mistura que possibilitou acertos com outras ferramentas. Ela é frequentemente utilizada nos componentes como Internet e processamento de documentos e banco de dados.

A linguagem possui uma gramática conhecida como DTD - *Document Type Definitions*, é nela que define a estrutura e os elementos e os atributos de um documento XML. Com essa gramática, é mais fácil a troca de dados entre dispositivos, pois uma aplicação pode verificar se o XML é válido ou não [9]. Essa gramática é definida no cabeçalho do documento XML quando o DTD está definido dentro desse mesmo arquivo XML. Caso o DTD seja definido por fora do arquivo XML, é necessário um documento de formato DTD para definir sua gramática, como é possível visualizar na Figura 3.3.

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Teste1</to>
<from>Teste 2</from>
<heading>Lembrete</heading>
<body>Exemplo de DTD Interno</body>
</note>
```

Figura 3.3: Exemplo de Código XML com DTD [9].

No XML é possível que os desenvolvedores criem e definam seus próprios elementos, podendo existir um elemento com o mesmo nome, porém definidos em vocabulários diferentes. O uso do "*namespace*"permite que elementos com os mesmos nomes, porém com pápeis diferentes, em um mesmo documento XML, pois possuem um vocabulário diferente do outro. No caso das interfaces XML executadas pela máquina virtual *Dalvik* ou *Android Runtime* (ART), abordada na próxima seção, as telas em XML são construídas acerca do padrão de arquitetura de software MVC (*Model-View-Controller*, Modelo-Visão-Controlador) padronizada pelo Android. Segundo Gamma [71], o *Model* é o objeto de aplicação, a *View* é a apresentação na tela e o *Controller* é o que define a maneira como a interface do usuário reage às entradas do mesmo. O XML é o responsável por essas telas.

## 3.3 Android

Visto que telefones celulares permitem fácil acesso a informação e uma vasta gama de recursos que o usuário precisa e dado o fato dos celulares serem portáteis e mais baratos

que computadores, além de possuirem em funções que computadores oferecem, como o acesso *web*, de maneira mais prática, a computação móvel ganhou espaço no cenário tecnológico e tem sido difundida de maneira gradual.

O sistema operacional móvel Android foi desenvolvido pela empresa Google com o foco em dispositivos móveis. Android é um projeto desoftware de código aberto, regido por um grupo formado pela própria empresa Google chamado *Open Handset Alliance* com o princípio de melhorar o sistema operacional. O Android fornece um sistema de interação com o usuário simples, sendo utilizado em televisores digitais, câmeras e até mesmo refrigeradores, além dos costumeiros *smartphones*. Esse sistema operacional é o mais usado atualmente no Mundo, de acordo com a pesquisa feita pela *International Data Corporation* (IDC), no primeiro trimestre de 2015, a popularidade no uso do sistema *Android* atingiu 78% do mercado de telefones celulares, demonstrando que o sistema operacional visa atender uma grande parte dos usuários de computação móvel no Mundo.

O sistema operacional *Android* foi baseado no *kernel* 2.6 do *Linux*, e esse é o responsável por gerenciar a memória e os processos, permitindo que várias aplicações sejam executadas em segundo plano. No *Android*, cada aplicação é executada em um processo e cada processo possui uma *thread* dedicada, permitindo que vários aplicativos possam ser executados ao mesmo tempo, no qual o *kernel* do sistema é o responsável por esse controle de memória, podendo esse terminar processos para liberar memória caso necessário [72].

O Android utiliza uma maquina virtual feita especialmente para dispositivos móveis. Na versão mais antiga, a máquina virtual era conhecida como *Dalvik* e era otimizada para executar em pouca memória usando os métodos de paginação e mapeamento de memória [73]. Na versão 4.4, conhecida como "*KitKat*", foi implementada uma segunda maquina virtual conhecida como "*Android Runtime*" (ART), que foi feita para substituir a *Dalvik* com o intuito de melhorar o uso com um coletor de lixo mais otimizado, um depurador melhorado e com um recurso de compilação antecipada chamada "*Ahead-of-time*" (AOT), que melhora a performance de um aplicativo consideravelmente [74].

### 3.3.1 Histórico de Versões

O sistema operacional *Android*, propriedade da *Google*, não foi criado ao longo dos anos por essa mesma empresa. O desenvolvimento inicial do *Android* começou em 2003 com o propósito de revolucionar a indústria móvel, e em 2005 foi vendido para a empresa *Google* [75]. No ano de 2007, o desenvolvimento do sistema operacional *Android* foi iniciado com o lançamento da versão Alfa, com codinomes de robôs fictícios.

Além disso, foi criado também o logotipo do sistema operacional, desenhado por Irina Blok [10], que é utilizado no sistema operacional até hoje. A versão Beta foi criada em 5 de Novembro de 2007, e é considerada uma data importante na comunidade *Android* desde

então. Após 7 dias, foi criado o kit de desenvolvimento desoftware para o sistema [76], incentivando os desenvolvedores a criarem aplicativos e disponibilizando dados a respeito do sistema, e prêmios de até 10 milhões de dólares como incentivo a fomentar o mercado de aplicativos, conhecido hoje como *Play Store* [77].

Figura 3.4: Logotipo Android criada por Irina Blok [10].

No decorrer de quase 10 anos foram criadas 17 versões com codinomes diferentes, contando as versões Alfa e Beta. A partir da versão 1.5, foram dados nomes para cada versionamento do sistema operacional seguindo a letra alfabética, começando na letra C. Todas as versões possuem um nome que são referências a nomes de doces. A Tabela 3.1 apresenta todo o Histórico das versões do sistema Android, sendo a última versão lançada foi chamada de "*Nougat*".

Em uma pesquisa feita no dia 8 de Agosto de 2017 pela própria empresa *Android* [13], foi constatado que a versão 4.4 atende mais de 90% dos usuários que possuem a plataforma em seus respectivos *smartphones* (aproximadamente 89,6% dos usuários). Além disso, os aplicativos compilados nas versões de API 4.4 conseguem ser executados nas versões mais recentes do sistema operacional. Esses aplicativos costumam ser desenvolvidos com a ferramenta *Android Studio*, que é um Ambiente de Desenvolvimento Integrado (IDE) oficial da empresa *Android*, baseado na ferramenta *IntelliJ*. Esse ambiente apresenta melhorias e atualizações mais constantes do que outros ambientes que são utilizados para criar aplicativos *Android*.

### 3.3.2 Arquitetura

Conforme citado nas seções anteriores, o Android utiliza as linguagens de programação Java, que é utilizada para agregar dados computacionais como cálculos e, também, a linguagem XML, utilizada para criar a interface de interação humano-computador. Porém,

Tabela 3.1: Versões do Android [13].

| **Codinome** | **Versão** | **Data de lançamento** | **Nível API** |
|---|---|---|---|
| (Sem Codinome) | 1.0 | 23 de Setembro de 2008 | 1 |
| (Conhecida como *Petit Four*) | 1.1 | 9 de Fevereiro de 2009 | 2 |
| *Cupcake* | 1.5 | 27 de Abril de 2009 | 3 |
| *Donut* | 1.6 | 15 de Setembro de 2009 | 4 |
| *Eclair* | 2.0 - 2.1 | 26 de Outubro de 2009 | 5-7 |
| *Froyo* | 2.2 - 2.2.3 | 20 de Maio de 2010 | 8 |
| *Gingerbread* | 2.3 - 2.3.7 | 6 de Dezembto de 2010 | 9-10 |
| *Honeycomb* | 3.0 - 3.2.6 | 22 de Fevereiro de 2011 | 11-13 |
| *Ice Cream Sandwich* | 4.0 - 4.0.4 | 18 de Outubro de 2011 | 14-15 |
| *Jelly Bean* | 4.1 - 4.3.1 | 9 de Julho de 2012 | 16-18 |
| *KitKat* | 4.4 - 4.4.4 | 31 de Outubro de 2013 | 19-20 |
| *Lollipop* | 5.0 - 5.1.1 | 12 de Novembro de 2014 | 21-22 |
| *Marshmallow* | 6.0 - 6.0.1 | 5 de Outubro de 2015 | 23 |
| *Nougat* | 7.0 - 7.1.2 | 22 de Agosto de 2016 | 24-25 |
| *O* | 8.0 | Para Anunciar | 26 |

há componentes feitos em C e C++ utilizados pelo sistema Android, permitindo criar programas nativos que utilize funções por meio dessas linguagens de programação [78]. Quando compilado, o código é encapsulado em um arquivo com o formato .apk (*Android Package*) e este possui todo o conteúdo feito em um aplicativo, como por exemplo imagens e arquivos.

Para o usuário efetuar a instalação de um aplicativo, é necessário usar esse arquivo com essa extensão. A Google oferece um repositório de aplicativos de seu sistema operacional, conhecido como *Google Play*, no qual usuários de vários dispositivos diferentes conseguem gerenciar as instalações e as atualizações dos aplicativos.

O Android é um sistema operacional de código aberto com sua base feita em Linux e a arquitetura é dividida em camadas, no qual cada parte do sistema é encarregado por administrar os vários processos atribuídos a sua camada respectiva [72]. A Figura 3.5 representa a arquitetura da plataforma Android que é dividida em 6 partes.

A parte fundamental do sistema operacional Android é o *kernel* do Linux [11]. Isso faz que a máquina virtual trabalhe em conjunto com a kernel do Linux para realizar funções como o encadeamento e o gerenciamento de memória de baixo nível. Ao usar a kernel do Linux, o Android consegue aproveitar os recursos de segurança que esse sistema operacional oferece, assim como empresas que desenvolve dispositivos conseguem efetivamente criar *drivers* de hardware.

A camada de abstração de hardware (*Hardware Abstraction Layer*, ou HAL) trabalha juntamente com o dispositivo para localizar e utilizar a infraestrutura que o dispositivo oferece, tais como sensores, áudio e quaisquer outro presente no aparelho. Essa camada

Figura 3.5: Pilha de Software Android adaptado de [11].

permite que o hardware do dispositivo comunique com toda a estrutura da Java API de maior nível [11]. A camada está dividida em módulos, na qual cada uma implementa uma interface para cada tipo de hardware, por exemplo, uma API deseja fazer uma chamada para acessar o hardware da câmera do dispositivo, o sistema Android deverá carrega esse módulo de hardware para efetivamente utilizar a câmera.

Os dispositivos Android que possuem uma versão mais nova, mais precisamente acima da versão 5.0, ou nível API 21, executam cada aplicativo num processo com uma instância da máquina virtual (*Android Runtime*), que foi projetado para executar várias máquinas

virtuais em dispositivos com baixa memória através de arquivos DEX, um formato de *bytecode* feito especificamente para Android [11]. No caso de dispositivos com a versão do Android anterior a versão 5.0, o dispositivo deverá usar a máquina virtual Dalvik para executar aplicativos. Porém, se o aplicativo conseguir ser efetivamente executado por meio da máquina virtual (*Android Runtime*), ele poderá ter o mesmo desempenho na máquina Dalvik, mas o inverso pode não acontecer.

Para economizar a energia da bateria do dispositivo móvel, o sistema Android gerencia o acesso de processos e aplicativos. Quando uma aplicação está inativa, o sistema suspende sua operação para economizar o uso do processador e do consumo da bateria, restringindo acesso a conexão de dados temporariamente [79]. Para extender o uso da bateria do dispositivo, a plataforma entra em um período de hibernação após um tempo predeterminado de inatividade, desligando a tela do dispositivo até o próximo uso. Os sistemas Android mais recentes utilizam recursos otimizados conhecidos como *Doze*, que faz que o sistema diminui o acesso ao processador e o uso da rede de dados após um longo período de inatividade. [80]

Componentes como a camada de abstração de hardware e a máquina virtual (*Android Runtime*) foram implementados utilizando bibliotecas C e C++ nativas. O sistema Android fornece através das API's de *Java Framework* para facilitar o acesso dessas bibliotecas nativas aos aplicativos. Assim, é possível, por exemplo, utilizar a biblioteca *OpenGL ES* para permitir que aplicativos de mapeamento utilize gráficos 3D para representar prédios e instalações [11]. É possível, também, criar aplicativos utilizando parte do código nas linguagens C e C++ através do conjunto de ferramentas *Android* NDK, para acessar diretamente parte dessas bibliotecas de plataforma nativa no código.

Os recursos do sistema operacional Android são acessados através das API's programadas em *Java*, tais API's são essenciais para a criação dos aplicativos *Android*, simplificando a reutilização de componentes e serviços [11], tais como:

- Os provedores de conteúdo permite que os aplicativos acessem dados de outros aplicativos, como o aplicativo Telefone;
- O sistema de visualização é utilizado para a criação de interfaces de usuário em um aplicativo, como o uso de caixas de texto, grades, botões, etc;
- O gerenciador de recursos fornece acesso a recursos como *strings*, arquivos de *layout*, gráficos, etc;
- O gerenciador de notificação permite que aplicativos criem alertas personalizados na barra de *status*.

- O gerenciador de atividade gerencia o funcionamento dos aplicativos através do ciclo de vida, além de fornecer uma pilha de navegação inversa.

Por fim, o sistema *Android* conta com um grupo de aplicativos principais para diversas operações como *e-mail*, navegador de Internet, entre outros. Os aplicativos do sistema são inclusos no sistema operacional, mas não se torna uso exclusivo do usuário, pois a plataforma permite que o usuário instale outros aplicativos para usar como aplicativo padrão para um tipo específico de operação. Os aplicativos de sistema fornecem o uso de atividades principais de um aparelho em um aplicativo, por exemplo, se um aplicativo deseja permitir que o usuário utilize um discador para fazer uma ligação, é possível fazer a chamada do aplicativo Telefone já instalado para fazer tal ligação [11].

### 3.3.3 Kit de Desenvolvimento

Um *kit* é um conjunto de ferramentas para o desenvolvimento de uma aplicação móvel. Em 23 de setembro de 2008 foi lançado o kit de desenvolvimento versão 1.0 para o público [76]. Já a versão 6.0.1 foi lançada em 7 de dezembro de 2015, codinome "*Marshmallow*" [81]. A *Google* oferece hoje como sua principal ferramenta de desenvolvimento o *Android Studio*, com o principal foco de migrar todos os desenvolvedores da plataforma *Android* para essa ferramenta, ao invés de usar o *framework Eclipse* adaptado.

Essa interface de desenvolvimento é gratuita e disponível para vários sistemas operacionais (*Windows, Linux, Mac OS*) apesar da existência de *plugins* de terceiros serem pagos e também a necessidade de pagar uma taxa em mercados de vendas de aplicativos *Android* como *Google Play* ou *Amazon*, levando em conta o mercado oficial da *Google* gratuito ao usuário, é necessário pagar vinte e cinco dólares como taxa única antes de enviar o seu aplicativo de tamanho máximo de cem *megabytes* [82].

O Ambiente de Desenvolvimento Integrado *Android Studio* foi criado pela própria *Google* baseado no ambiente *IntelliJ*, com o intuito de facilitar a codificação do programa, assim como a integração de *plugins*, API's e também dispositivos conectados no computador com a finalidade de testar em um dispositivo real. A escolha dessa IDE para criar uma aplicação móvel se dá ao fato de que ela está sendo constantemente atualizada, assim como a empresa *Google* pretende dar suporte para esse ambiente por tempo ilimitado. O aprendizado em desenvolvimento *mobile* tem grande curva de aprendizagem no início, porém o ambiente do *Android Studio* permite uma configuração mais fácil do que o programador está habituado, assim como a facilidade de consertar erros relacionados a aplicação ou ao ambiente [83].

Por fim, um outro fator importante se deve ao fato desse ambiente conseguir construir um aplicativo no emulador com rapidez e eficácia. Com um emulador já implementado

dentro da IDE, o *Android Studio* também permite o uso de outros emuladores e versões do sistema operacional, assim como qualquer *plugin* disponível que o desenvolvedor necessite para criar seu aplicativo, que seja facilmente atualizados com o próprio *Android Studio*.

## 3.4 Web Service

O *web service* é um serviço de transmissão de dados oferecido por um dispositivo eletrônico para outro dispositivo eletrônico, ou seja, um dispositivo comunica com outro através de uma conexão de *Internet*.

Tecnologias *web* como HTTP (*Hypertext Transfer Protocol*) que foram desenvolvidas para uma comunicação entre uma máquina e o usuário são utilizadas para realizar a comunicação entre as duas máquinas, mais especificamente para transferir arquivos de formatos legíveis como XML ou JSON. A organização W3C (*World Wide Web Consortium*) responsável pelas padronizações internacionais para o *World Wide Web* (WWW) descreve que o *web service* é um "sistema desoftware desenvolvido para suportar interações entre máquinas dentro de uma rede. Ele possui uma interface descrita como um formato processado somente por máquinas, como é possível ver na Figura 3.7.

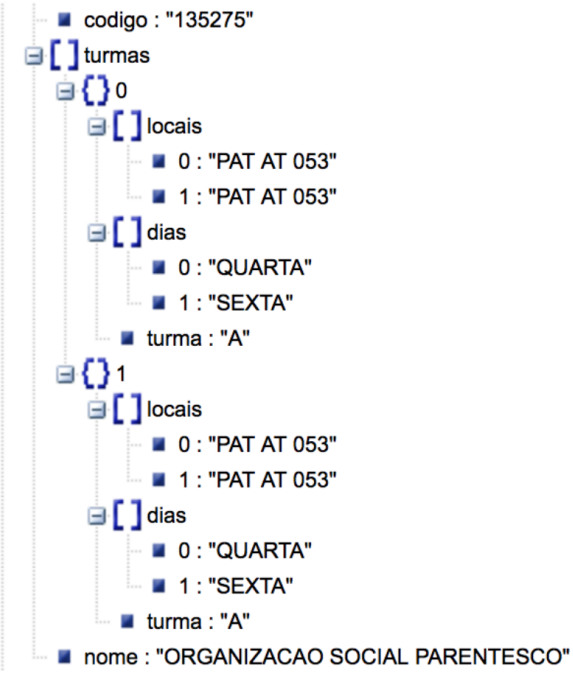


Figura 3.6: Exemplo de Arquivo JSON.

Outros sistemas interagem com o *web service* em uma maneira prescrita por sua descrição usando mensagens SOAP (*Simple Object Access Protocol*), que é um protocolo para

troca de informação estruturada na implementação de *web services* nas redes de computadores. Essas informações são tipicamente transportadas usando o protocolo HTTP com XML serializado em conjunção com outros padrões Web [84]. O termo *web service* descreve um caminho padronizado para integrar aplicações baseadas em *web* usando padrões XML (*eXtensible Markup Language*), SOAP (*Simple Object Access Protocol*), WSDL (*Web Service Definition Language*) e UDDI (*Universal Description Discovery and Integration*) em um protocolo de Internet nas redes principais, no qual o XML é utilizado para identificar o dado e o SOAP é utilizado para transferir o dado. O WDSL é usado para descrever os serviços disponíveis e UDDI lista quais serviços estão disponíveis para uso conforme mostra a figura a seguir.

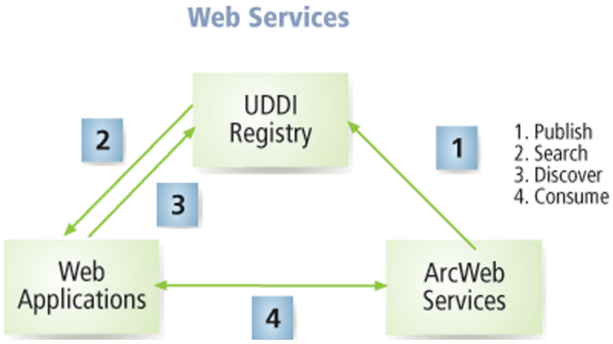


Figura 3.7: Exemplo de *Web Service*.

Como citado anteriormente, *web service* é um método de comunicação entre dois dispositivos eletrônicos em uma rede de conexão de dados, ou seja, é uma função de umsoftware fornecido em um endereço de rede na Internet e esse serviço deverá sempre funcionar em um aspecto de utilidade computacional. Muitas organizações usam sistemas desoftware múltiplos para gerenciamento de dados, ou seja, sistemas de *softwares* diferentes geralmente necessitam a troca de dados entre eles, e o *web service* é um método de comunicação que permite dois sistemas de *softwares* trocarem informações em toda a *Internet*.

O sistema desoftware que solicita dado é chamado de *service requester* (solicitante de serviços) assim como o sistema desoftware que deve processar e oferecer os dados é chamado de *service provider* (fornecedor de serviços). Vários *softwares* diferentes podem utilizar linguagens de programação diferentes, o que pode acarretar em um sistema não reconhecer o dado transferido ou conflito dos dados utilizados. Para isso, existe uma necessidade de um método de transferência de dado que não dependa unicamente de uma linguagem de programação particular. Entretanto, a maioria dos tipos de *softwares* podem interpretar abas XML ou JSON, e assim, *web services* podem utilizar arquivos XML para transferência de dados. Então existe a necessidade de criar regras de comunicação entre

diferentes sistemas, tais como um sistema pode pedir dados de outro sistema. Para isso, é possível perceber a necessidade das seguintes regras:

- Quais parâmetros especificamente são necessários no pedido de dados;
- Qual seria a estrutura produzida e normalmente esses dados são transferidos em arquivos XML;
- Quais mensagens de erro deverão ser mostradas quando certa regra de comunicação não for observada, a fim de melhorar o quadro de solução de problemas.

Essas regras de comunicação citadas acima são definidas em um arquivo chamado WSDL (*Web services Description Language*) no qual ele possui uma extensão *..wdsl* para leitura. Existe um diretório chamado UDDI (*Universal Description, Discovery and Integration*) cuja função define qualsoftware deverá ser chamado para um tipo especifico de dado. Por exemplo, quando um sistema desoftware precisar de um tipo especifico de dado, ele deverá ir para o diretório UDDI e descobrir qual outro sistema para receber esse dado, é necessário verificar se é possível o contato com tal sistema. Assim que o sistema desoftware descobre qual outro sistema consegue receber tal dado, ele então entraria em contato usando um protocolo especial SOAP. Esse sistema de provedor de serviço deverá validar primeiramente o pedido de dados referenciando o arquivo WSDL, e assim processar o pedido para finalmente mandar a data usando o protocolo SOAP.

Para fazer o *web service* em vários dispositivos, uma ferramenta que pode ajudar o desenvolvedor é *Web API* que é uma interface de programação de aplicações para um servidor ou navegador web, ela é utilizada para recuperar os dados que o usuário necessita em um banco de dados de um site. Essa ferramenta tem como foco de simplificar as comunicações baseadas em REST (*Representational State Transfer*), no qual esta é um estilo de arquitetura composta por um conjunto de componentes, conectores e elementos de dados coordenados dentro de um sistema de hipermídia distribuído. O objetivo dessa arquitetura é conduzir desempenho, simplicidade, portabilidade e confiabilidade na comunicação de dados e o foco dessa ferramenta está na função dos componentes e em um conjunto de interações entre os elementos de dados [85].

Desenvolver *web services* sem a arquitetura REST pode ser complexo de implementar e normalmente são criados por terceiros ou empresas, ao invés de implementados em código aberto. O trabalho relacionado com a visualização e captura de dado muda em cada tipo de *web service* e portanto a portabilidade desse serviço pode ter o custo bem alto. Ainda assim, é possível utilizar artefatos intermediários nas mudanças de computação e visualização de dados.

De forma geral, a API (*Application Programming Interface*) é um conjunto de rotinas, protocolos e ferramentas utilizadas para construção de softwares e aplicações para utilizar

suas funcionalidades por programas que não necessita de atenção em detalhes da implementação, e sim apenas usar seus serviços. Assim, uma API bem desenvolvida facilita o desenvolvimento de um programa ao providenciar os serviços requisitados pelo programador. A *Web* API providencia a mesma interface de programação, porém, conforme citado anteriormente, criado para um servidor ou navegador *web*, usualmente limitado em uma aplicação *web* no lado do cliente e pode não incluir detalhes de implementação de servidor ou navegador *web*.

Para o *Web API* em contato ao servidor, essa é uma interface programática que consiste em um ou mais *endpoints* públicos para definir um sistema de mensagem de resposta e pedido, expressada normalmente por JSON ou XML. Assim, tem-se por definição que *endpoint* é o nome para uma entidade em um terminal de uma conexão da camada de transporte, ou seja, são importantes aspectos de interação com o *Web API* do servidor, pois eles conseguem especificar quais recursos podem ser acessados por outros *softwares*. É importante que os *endpoints* sejam estáticos, ou seja, é importante manter seus valores sem modificações, pois caso o contrário, não vai ser garantido o funcionamento correto dosoftware que interage com o *web service*. Caso a localização de um recurso mude, assim como o *endpoint*, então osoftware que for utilizar esse serviço vai retornar um erro, dado que ele não consegue encontrar o recurso pedido. Portanto é necessário que o servidor consiga manter a consistência dos seus recursos.

## 3.5 Considerações Finais

Neste capítulo foi apresentado o conjunto de tecnologias envolvidas no desenvolvimento deste trabalho. Para tal, foram descritos os conceitos da linguagem de programação Java, sua definição e arquitetura. Após, foi apresentado o conceito da linguagem de programação XML e seu arquétipo de funcionamento, assim como sua corroboração para a tecnologia, seu conceito e sua modelagem. Em seguida, apresentou-se o sistema operacional Android, sua definição, o histórico de versões desenvolvido até hoje e sua arquitetura. Assim, o kit de desenvolvimento necessário para programar aplicações móveis. Por fim, foi apresentado o conceito de *web services* e sua síntese.

# Capítulo 4

# Guia do Aluno UnB

Este capítulo apresenta o aplicativo Guia do Aluno UnB. Para isso, a Seção 1 apresenta uma visão geral do aplicativo. Em seguida, a Seção 2 descreve sobre a Universidade de Brasília. A Seção 3 apresenta o levantamento de requisitos para entender as necessidades dos usuários. A Seção 4 apresenta as funcionalidades do aplicativo Guia do Aluno UnB.

## 4.1 Visão Geral

O aplicativo Guia do Aluno UnB tem como objetivo guiar alunos e usuários na Universidade de Brasília, informando a localização das instalações de todos os campi da Universidade de Brasília - UnB, os quais são Darcy Ribeiro, Ceilândia, Gama e Planaltina. O aplicativo também oferece um sistema de busca de matérias oferecidas no Matrícula Web, sistema de matrícula da universidade, mostrando o devido local no qual a matéria está sendo ministrada. Além disso, o aplicativo oferece um sistema de gerenciamento de faltas, gerenciamento de tarefas, um acervo de documentos oferecidos aos alunos da UnB, uma lista de telefones úteis e acesso fácil aos *sites* do Matrícula Web e o portal da UnB.

Assim, as principais funcionalidades disponibilizadas pelo aplicativo Guia do Aluno UnB são apresentadas na Figura 4.1, e listadas a seguir:

- Busca das instalações de todos os campi da Universidade de Brasília;
- Mostra a localização do usuário no mapa, e indica o melhor caminho para um determinado ponto de origem;
- Busca matérias que estão no Matricula Web e indica qual o local da sala em que a disciplina está sendo ministrada.
- Gerencia as faltas e as tarefas das matérias;
- Disponibiliza o acervo de documentos e telefones úteis da UnB;

- Acessa os sites Matrícula Web e o portal da UnB.

Figura 4.1: Tela Inicial do Aplicativo.

O aplicativo utiliza o sistema operacional Android, o qual é compatível com todas as versões 4.4 (Codinome: "*KitKat*") ou superiores. Além disso, é necessário que o aparelho possua uma conexão de dados com a Internet ativa, seja através do Wi-Fi ou da operadora de celular, bem como uma conexão GPS, para que todos os módulos do aplicativo funcionem efetivamente.

Além disso, o seviço de localização disponibilizado pelo aplicativo permite a seleção de como o usuário deseja fazer o percurso, o qual pode ser por meio de carro, bicicleta ou a pé. Outra vantagem do aplicativo é a possibilidade do usuário navegar pelo mapa do campus livremente, ou utilizar a barra de busca da lista de disciplinas que permite que o usuário navegue no mapa do campus através de um serviço automatizado, que informa o nome da disciplina e a turma correspondente, respectivamente. Neste segundo caso, o aplicativo indicará o melhor caminho para a chegada do usuário na sala de aula onde a disciplina é ministrada.

Assim sendo, aplicativo Guia do Aluno UnB se divide em duas propostas: sistema georreferenciado, e serviços administrativos e estudantis. No sistema georreferenciado tem-se funcionalidades que utilizam de análise do espaço geográfico, como o Mapa e as Disciplinas, ambas sessões do aplicativo que guiam o estudante à localizações precisas do serviço que o mesmo procura. Esse serviço está dividido em categorias, representadas por ícones, cada categoria representa um tipo diferente de recurso da universidade, por

exemplo: as salas de aula, as secretarias, os departamentos, as lanchonetes, os restaurantes, as bancas de jornal, os postos de gasolina, a barbearia, as bibliotecas, a reitoria, os estacionamentos, dentre outros, conforme apresentado na Figura 4.8 e Figura 4.9. A segunda é auxiliar o aluno em suas tarefas acadêmicas, tais como administrar as faltas de matérias, gerenciar tarefas, pesquisar telefones úteis, acessar um acervo de documentos da UnB e o rápido acesso aos sites da UnB, conforme apresentado na Figura 4.10, Figura 4.12, Figura 4.13, Figura 4.14, Figura 4.15 e Figura 4.16. Logo a proposta do aplicativo é auxiliar a comunidade universitária em suas tarefas do dia-a-dia.

## 4.2 A Universidade de Brasília

A Universidade de Brasília é a maior universidade situada no Distrito Federal. Possui quatro campi, sendo que o primeiro campus fundado, chamado Darcy Ribeiro, fica situado na região central da Asa Norte, na extensão entre a Quadra 16 e a Quadra 5, totalizando uma área de 3.950.569,07 $m^2$. Essa área é formada por:

- Área gramada: 1.650.000 $m^2$;
- Área de laboratório: 32.138 $m^2$;
- Área total construída: 513.767,16 $m^2$;

O campus Darcy Ribeiro é composto por 58 prédios, dentre esses salas de aulas, espaços administrativos, reitoria, etc. Os quais apresentam, geralmente, distâncias médias e longas que os intercalam. Diante dessa realidade, a disposição de prédios de uso exclusivo da universidade (em exceção bancos, postos de gasolina e prédios comerciais) é constituído pelas coordenadas e nome do estabelecimento, conforme apresentado na Tabela 4.1 e Tabela 4.2.

O campus Ceilândia, no qual possui o foco em ampliar e criar cursos na área de saúde, possui três edifícios, dentre esses salas de aulas, laboratórios multidisciplinares, laboratórios de informática e auditórios. Além desses espaços, o campus de Ceilândia conta também, temporariamente, com um prédio localizado ao lado do Centro de Ensino Médio 4, o qual possui também de salas de aula e laboratórios.

Além desses, o campus Gama, que é focado em cursos de Engenharia, possui uma área total de 335.534 $m^2$ e uma área construída de 4.795 $m^2$. O campus conta com três prédios, que são compostos por salas de aulas, laboratórios de Engenharia, laboratórios de Informática e auditórios. Um fator que é destaque neste campus é que nele há vários laboratórios e centro acadêmicos dentro de *containers*, o que acaba economizando espaço e dinheiro.

Por fim, o campus Planaltina é composto por cursos da área de ciências agrárias. Possui uma área total de 301.847,06 $m^2$, e uma área construída de 2.860,26 $m^2$. O campus é composto por quatro prédios, que são constituídos por salas de aulas, laboratórios e auditórios, no qual um desses prédios, é um dormitório para alunos. O campus Planaltina é o único, além do Darcy Ribeiro, que possui tal tipo de instalação.

A Universidade de Brasília possuia no primeiro semestre de 2016 (2016.1), 27.053 alunos. Desse total, 22.602 são alunos do curso diurno, e 4.451 alunos do curso noturno [86]. Isso ressalta a necessidade de um aplicativo que auxilie toda a comunidade universitária a localizar as instalações de suas salas, pois como dito, os prédios são fisicamente distantes. Além disso, há muitos espaços, e de um modo geral, a comunidade universitária não conhece todos os espaços da universidade.

Dessa forma, constata-se que qualquer aluno que não tenha o conhecimento prévio do seu local de destino vai passar por dificuldades para localizar as instalações, assim como as salas dentro de tais instalações.

Diante do exposto, com um aplicativo de georreferenciamento, o usuário tem o mapa dos campi, com as devidas localizações para a pesquisa tanto de instalações quanto de salas que essas instalações possuem, diretamente em seu dispositivo móvel. Assim, é necessário apenas que o usuário acesse o aplicativo, que esse automaticamente encontrará o seu local por coordenadas e, informa o seu local de destino, retornando o resultado com o melhor caminho (cálculo eficiente de rota) para chegar ao destino solicitado.

## 4.3 Levantamento dos Requisitos

Para esclarecer as necessidades dos usuários é preciso levantar requisitos funcionais e não-funcionais. Além do estudo de caso acerca da geografia da universidade, foi elaborada uma pesquisa com a comunidade acadêmica que frequenta ou já frequentou a Universidade de Brasília. Assim, no dia 28 de Junho de 2016, foram entrevistadas 45 pessoas, conforme demonstra-se na pesquisa.

Nessa pesquisa foram feitas 8 perguntas. A primeira pergunta consistiu em saber o quantitativo de aparelhos telefônicos modernos na Universidade de Brasília. No resultado da pergunta "Você possui algum *Smartphone/Tablet*? Se sim, você sabe qual é o sistema operacional dele?", com os resultados apresentados, ficou claro que se o aplicativo fosse feito no sistema operacional Android, ele atenderia mais usuários do que qualquer outro sistema operacional para *smartphone*. Outro fator importante que ficou evidente no resultado dessa pergunta foi que nenhum dos entrevistados respondeu usando a alternativa "Não possuo *Smartphone* / Não sei o sistema operacional", ou seja, todos os entrevistados tinham pelo menos um *smartphone* compatível com as alternativas dadas. Portanto, se o

Tabela 4.1: Tabela com as Posições das Instalações da Universidade de Brasília - Campus Darcy Ribeiro.

| **Posição** | **Instalação** |
|---|---|
| -15.76062, -47.87053 | ICC Norte - Insituto Central de Ciências |
| -15.77454, -47.86681 | Centro de Apoio ao Desenvolvimento Tecnológico - CDT |
| -15.77115, -47.86689 | Laboratório de Estudos Geodinâmicos e Ambientais |
| -15.77275, -47.86578 | CPD - Centro de Informática |
| -15.76491, -47.87148 | SG4 - Departamento de Música |
| -15.76443, -47.87142 | SG2 - Música |
| -15.77286, -47.86675 | INFRALAB |
| -15.77255, -47.86746 | Diretoria de Manutenção de Equipamentos - DIMEQ |
| -15.77171, -47.86810 | Centro de Referência em Conservação da Natureza e Recuperação de Áreas Degradadas - CRAD |
| -15.77193, -47.86576 | CEBRASPE (Cespe) |
| -15.77092, -47.87141 | Fundação Osvaldo Cruz - FIOCRUZ |
| -15.77322, -47.87096 | Fundação de Empreendimentos Científicos e Tecnológicos - FINATEC |
| -15.77592, -47.87092 | Fundação Universitária de Brasília - FUBRA |
| -15.77208, -47.87044 | Autotrac |
| -15.77159, -47.87400 | Hospital Universitário - HUB |
| -15.77306, -47.86706 | Centro Interdisciplinar de Estudos em Transportes - CEFTRU |
| -15.76968, -47.86769 | Núcleo de Medicina Tropical |
| -15.76963, -47.87028 | Centro de Excelência em Turismo - CET |
| -15.76872, -47.86946 | Centro de Desenvolvimento Sustentável - CDS |
| -15.76885, -47.86623 | Faculdade de Ciências da Saúde - FS |
| -15.76850, -47.86481 | Instituto de Química - IQ |
| -15.76680, -47.86672 | Bloco de Salas de Aula Sul - BSAS |
| -15.76592, -47.86594 | Instituto de Ciências Biologicas - IB |
| -15.76503, -47.87197 | SG10 - CEPLAN / Núcleo de Vivência |
| -15.75804, -47.86832 | Centro Comunitário Athos Bulcão |
| -15.75698, -47.87037 | Bloco de Salas de Aula Eudoro de Sousa - BAES |
| -15.74920, -47.87704 | Hospital Veterinário - HVET |
| -15.73642, -47.88352 | Estação Experimental de Biologia |
| -15.75703, -47.87130 | Bloco de Salas de Aula Norte - BSAN |
| -15.76706, -47.86965 | Pavilhão Multiuso 1 CEAM - PMU 1 |
| -15.76563, -47.87046 | Instituto de Artes da Universidade de Brasília - IdA |
| -15.75750, -47.87281 | Instituto de Ciências Sociais - ICS |
| -15.76347, -47.87235 | Faculdade de Tecnologia - FT |
| -15.76522, -47.87248 | SG12 - Laboratório de Engenharia Civil |
| -15.76567, -47.86687 | ICC Sul - Insituto Central de Ciências |
| -15.76397, -47.86591 | Memorial Darcy Ribeiro - Beijodromo |
| -15.76326, -47.86055 | Faculdade de Educação Física - FEF |
| -15.76820, -47.87100 | Faculdade de Educação 1 - FE1 |
| -15.76759, -47.87140 | Faculdade de Educação 3 - FE3 |

Tabela 4.2: Tabela com as Posições das Instalações da Universidade de Brasília - Campus Darcy Ribeiro (continuação).

| Posição | Instalação |
|---|---|
| -15.76809, -47.87182 | Faculdade de Educação 5 - FE5 |
| -15.76797, -47.86885 | Pavilhão Multiuso 2 CIFMC/IFD - PMU 2 |
| -15.76448, -47.87180 | SG8 - Auditório de Música |
| -15.76345, -47.85767 | Centro Olímpico - CO |
| -15.76534, -47.85785 | Casa do Estudante Universitário - Bloco A |
| -15.76701, -47.87228 | Casa do Professor - ADUnB |
| -15.76652, -47.87147 | SG9 - Laboratório de Engenharia Mecânica |
| -15.76591, -47.87213 | SG11 - Laboratório de Engenharia Elétrica |
| -15.75955, -47.87212 | Faculdade de Direito - FD |
| -15.75857, -47.87185 | Faculdade de Administração, Contabilidade, Economia e Gestão de Políticas Públicas - FACE |
| -15.76639, -47.87249 | Alumni UnB - Associação dos Ex-Alunos da UnB |
| -15.75911, -47.87076 | PAT - Pavilhão Anísio Teixeira |
| -15.75842, -47.87023 | PJC - Pavilhão João Calmon |
| -15.75824, -47.86962 | IREL/IPOL - Instituto de Relações Internacionais e Ciência Política |
| -15.75862, -47.86904 | Prédio de Ciência da Computação e Estatística - CIC/EST |
| -15.76570, -47.87274 | Observatório Sismológico |
| -15.77028, -47.86847 | Laboratório de Termobiologia |

aplicativo Guia do Aluno UnB fosse implementado para celulares com Android, ele será de fácil acesso para a maioria dos indivíduos que usualmente frequentam a Universidade de Brasília. Os resultados dessa pergunta são apresentados na Figura 4.2:

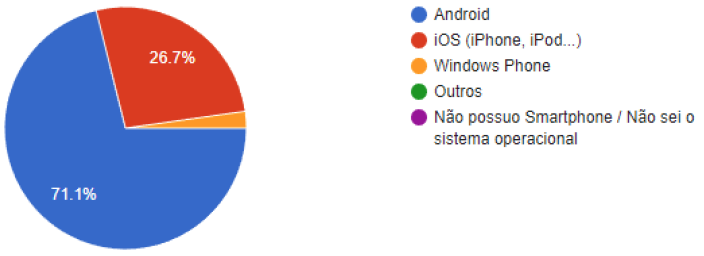


Figura 4.2: Respostas da Pergunta: "Você Possui Algum *Smartphone/Tablet*? Se Sim, Você Sabe Qual é o Sistema Operacional Dele?".

A segunda pergunta feita para os entrevistados fez referência ao vínculo dos mesmos com a Universidade de Brasília. O resultado da pergunta "Você é aluno (a)/professor (a) ou já foi aluno (a)/professor (a) da Universidade de Brasília, campus Darcy Ribeiro?" mostrou que mais de 97% do usuários tem ou teve algum vínculo com a Universidade de

Brasília, ou seja, possuem algum conhecimento sobre a estrutura e as instalações de pelo menos um campus da Universidade de Brasília, conforme apresentado na Figura 4.3.

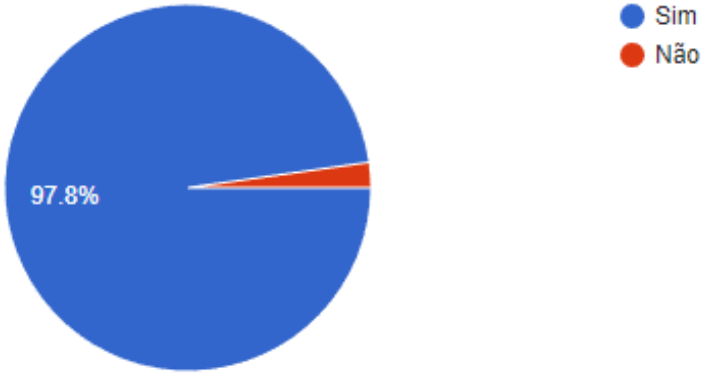


Figura 4.3: Respostas da Pergunta: "Você é aluno (a)/professor (a) ou já foi aluno (a)/professor (a) da Universidade de Brasília, campus Darcy Ribeiro?".

A terceira pergunta "Ao tentar localizar uma sala de aula pela primeira vez, você já sentiu dificuldades em encontrá-la?" mostrou que um indivíduo novato não consegue chegar a uma determinada sala de aula com sucesso, em 95,6% dos entrevistados. Isso demonstrou que nos campi não há instruções suficientes para uma pessoa localizar um espaço na UnB. Os resultados dessa pergunta são apresentados na Figura 4.4:

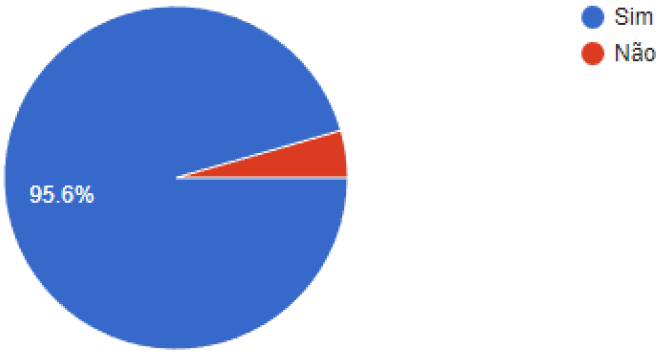


Figura 4.4: Respostas da Pergunta: "Ao tentar localizar uma sala de aula pela primeira vez, você já sentiu dificuldades em encontrá-la?".

Na quarta pergunta, "Com que frequência você frequenta a Universidade de Brasília, campus Darcy Ribeiro?", os resultados mostraram que boa parte dos entrevistados frequenta a Universidade de Brasília, campus Darcy Ribeiro, com uma alta frequência,

pois 62,2% dos entrevistados responderam que vão, pelo menos, cinco vezes ou mais por semana para a universidade, e que 17,8% dos entrevistados frequentam quatro vezes por semana a Universidade de Brasília, campus Darcy Ribeiro, ou seja, somando os dois resultados, tem-se que 80% dos entrevistados frequentam a Universidade de Brasília pelo menos quatro vezes por semana. Os resultados para essa pergunta são mostrados na Figura 4.5:

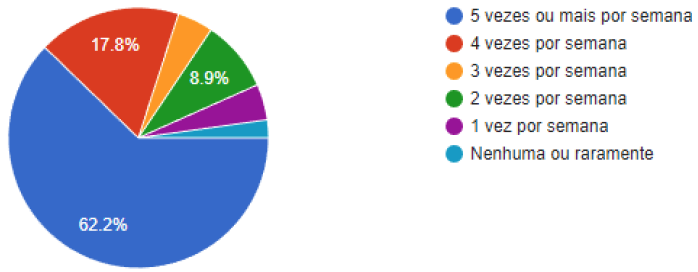


Figura 4.5: Respostas da Pergunta: "Com que frequência você frequenta a Universidade de Brasília, campus Darcy Ribeiro?".

Na quinta pergunta, "Quando calouro, você já teve dificuldades em se localizar na Universidade de Brasília?", a alternativa 'Sim' foi respondida por 91,1% dos entrevistados, e esse resultado prova mais uma vez que não há instruções o suficiente para guiar os novatos na Universidade de Brasília, campus Darcy Ribeiro, ou pelo menos não há informações para guiar os usuários nas instalações da universidade.

Na sexta pergunta, "Quantas vezes você já se perdeu na Universidade de Brasília Campus Darcy Ribeiro?", o resultado foi surpreendente em relação ao que era esperado, mais de 42% dos entrevistados responderam que ficaram perdidos ao menos cinco vezes na Universidade de Brasília – campus Darcy Ribeiro. Pelo menos 95% dos entrevistados ficaram perdidos ao menos uma vez na universidade, conforme os resultados apresentados na Figura 4.6:

Na sétima pergunta, "Você já teve dificuldades de locomoção e/ou localização de endereços na Universidade de Brasília?" a resposta "Sim" foi respondida por aproximadamente 93% dos entrevistados, onde de fato eles relatavam que enfrentaram dificuldades para encontrar a sala de aula da sua matéria ao menos uma vez.

Por fim, a última pergunta faz referência ao interesse dos entrevistados em utilizar um aplicativo para dispositivos móveis. Assim, a pergunta foi "Você utilizaria um aplicativo que contém suas informações acadêmicas (horários, disciplinas, documentos) e um GPS que o guiasse a endereços na UnB?". A resposta "Sim" foi respondida por aproximadamente 89% dos entrevistados, conforme mostrado na Figura 4.7:

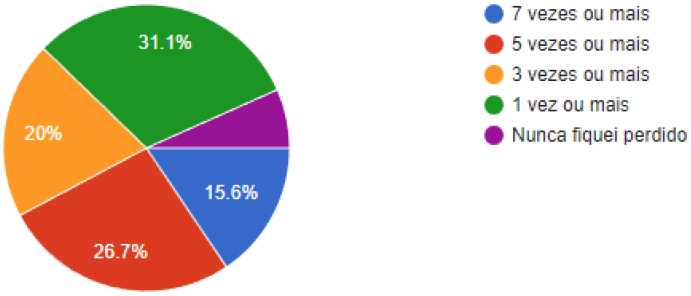


Figura 4.6: Respostas da Pergunta: “Quantas vezes você já se perdeu na Universidade de Brasília Campus Darcy Ribeiro?”.

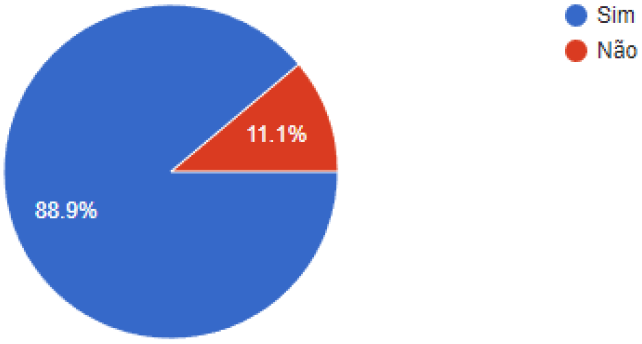


Figura 4.7: Respostas da Pergunta: “Você utilizaria um aplicativo que contém suas informações acadêmicas (horários, disciplinas, documentos) e um GPS que o guiasse a endereços na UnB?”.

A pesquisa foi realizada com o propósito de levantar os requisitos funcionais, e são funcionais para o desenvolvimento do aplicativo, a fim de reconhecer os problemas encontrados pelos alunos no cotidiano da vida acadêmica.

Portanto, a partir desses resultados foi possível levantar as necessidades que uma aplicação móvel georreferenciada deveria ter para atender a comunidade acadêmica visando atingir majoritariamente seus usuários.

## 4.4 Funcionalidades do Aplicativo Guia do Aluno UnB

Como dito anteriormente, o aplicativo Guia do Aluno UnB tem duas grandes áreas de atuação, as quais são a geolocalização dos espaços nos campi da UnB e no auxílio nas tarefas acadêmicas. Assim, a tela inicial do aplicativo, conforme mostrado na Figura 4.1, apresenta nove ícones, os quais são detalhados nas próximas seções.

### 4.4.1 Disciplinas

O *menu* de Disciplinas do aplicativo é alimentado por toda as disciplinas ofertadas durante o semestre letivo na Universidade de Brasília. Esses dados são extraídos a partir do site da UnB por meio de um *web service*, desenvolvido para analisar dados do sistema de matrícula em disciplinas da UnB, chamado MatriculaWeb. Após executar esse *web service*, os dados em um arquivo do tipo .JSON como dado público, e o aplicativo lê esses dados de forma sistemática, criando uma lista de oferta de disciplinas, conforme apresentado na Figura 4.8.

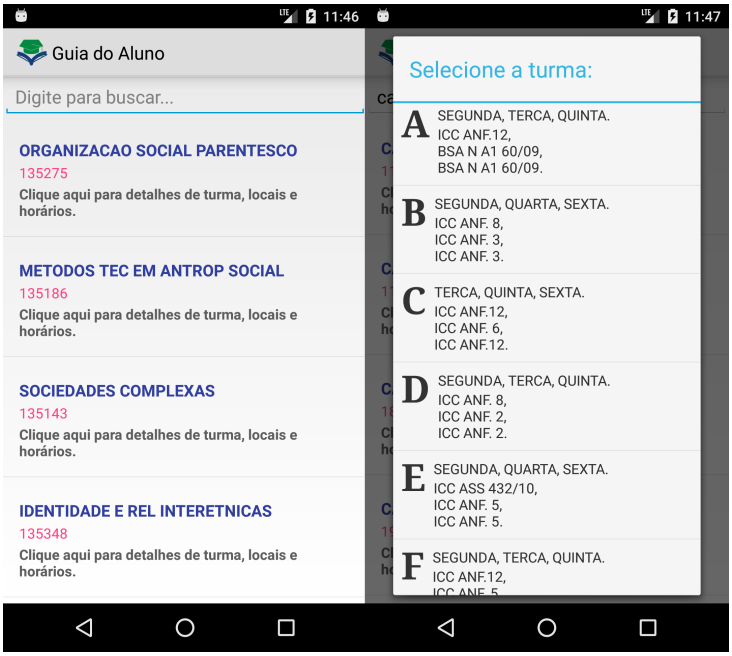


Figura 4.8: Telas do Menu de Disciplinas.

Dessa forma, na lista apresentada na Figura 4.8 é possível pesquisar o nome da disciplina desejada e, ao encontrá-la haverá seu código de matrícula referente. Ao selecionar uma disciplina, o aplicativo abre uma janela *pop-up* referente a disciplina escolhida com seus dias de aula, turmas disponíveis e dias da semana em que haverão as aulas. Ao selecionar a turma referente a sua matrícula, o aplicativo analisa sintaticamente o local de sala, e compara a uma lista de prédios existentes na UnB. Caso o prédio exista e seja válido, abre-se o mapa com o endereço, do destino encontrado, traçando a rota mais próxima para que o aluno possa ir para o endereço solicitado.

### 4.4.2 Mapa

O aplicativo Guia do Aluno UnB tem o foco principal de auxiliar, qualquer usuário do aplicativo, na localização das instalações da Universidade de Brasília, guiando-o por meio de um mapa digital que traça caminhos de uma posição inicial do usuário até seu destino final. Para isso, é necessário que o usuário informe somente o ponto de destino, pois o ponto inicial é obtido automaticamente, buscando a posição do usuário pelo GPS. Assim, o aplicativo deverá retornar o caminho mais curto estabelecido nas rotas predefinidas da API.

Figura 4.9: Telas da Função Mapa.

Dentre as possibilidades do mapa, há diversas *views*, que é a maneira que o mapa é apresentado na tela, seja ela como desenho ou como imagens de satélite reais, ou seja, isso é a formação gráfica do mapa na tela do *smartphone*, conforme apresentado na Figura 4.9. Além disso, por meio do uso da biblioteca OpenGL ES, o aplicativo também consegue efetivamente representar os prédios por meio de figuras 3D, porém essa função não existe nas áreas rurais.

Por padrão, é mostrada a localização do Tela do Menu de Disciplinas, principal prédio do campus Darcy Ribeiro da Universidade de Brasília e, normalmente, o mais acessado por professores, alunos e funcionários. Após o usuário indicar o local de destino, o aplicativo traça a rota mais ágil de sua localização atual até a de destino, e oferece as opções de transporte por veículo motorizado, por bicicleta ou semelhantes, e também via caminhada. Para pesquisar um local de destino, basta escrever seu respectivo nome na barra de busca.

Todavia, é possível mudar a posição da tela entre os campi da Universidade de Brasília rapidamente através de um botão de acesso.

Após selecionar um determinado marcador no mapa, o aplicativo mostrará uma janela conhecida como "*InfoWindow*". Essa janela mostrará dados como uma imagem que representa o ponteiro, informações como horário de funcionamento (somente dia útil) e telefone para contato (conforme apresentado na Figura 4.9), sendo que no último basta selecionar o número do telefone na janela que será discado automaticamente. Para definir destinos no mapa, basta o usuário pressionar na tela o local desejado, com isso o aplicativo deverá pedir a confirmação daquele local como destino, e assim traçar a melhor rota. É possível requisitar rotas usando um marcador como destino, basta utilizar a janela *InfoWindow* para gerar o caminho desejado, veja a Figura 4.9.

### 4.4.3 Faltas

O *menu* Faltas é um sistema de gestão de faltas para estudantes da Universidade de Brasília. Quando matriculado em uma disciplina, o aluno assume um compromisso de ter presença mínima de 25% nas aulas lecionadas dessa disciplina. Para tal, são necessárias assiduidade e responsabilidade para não exceder o número máximo de faltas. Cada disciplina tem um número de créditos relacionado a quantidade de horas aula que a mesma provê, de acordo com um calendário acadêmico. O calendário do ano letivo da Universidade é estabelecido pelo Conselho de Ensino, Pesquisa e Extensão (CEPE), sendo que as atividades acadêmicas desenvolvidas no ano letivo ocupam, no mínimo, 200 (duzentos) dias de trabalho escolar efetivo. Há, no ano, dois períodos letivos regulares de atividades, podendo haver um período especial. O semestre tem duração de 17 semanas. Portanto, uma disciplina de 60 horas, ou de quatro créditos, no período diurno, será ministrada em 34 aulas de 110 minutos.

Assim, essa função dispõe de um serviço de alerta automatizado, no qual o aluno receberá um alerta através da vibração do dispositivo quando estiver próximo do limite de faltas, de acordo com o número de créditos da disciplina. O aluno poderá ver o progresso de suas faltas através de uma barra que está relacionada a matéria, onde essa muda de cor de acordo com o número de faltas e o número de créditos da matéria, conforme a Figura 4.10. Assim a cor Verde significa que o número de faltas está abaixo do limite de reprovação; a cor amarelo significa atenção para o número de faltas; a cor laranja demanda cuidado; e a cor vermelha alerta o aluno para uma situação crítica, na qual ele poderia ser reprovado por falta.

Portanto, o aplicativo estabelece que o aluno entre com o nome da disciplina, seu respectivo quantitativo de créditos e sempre que houver uma falta, registre na aplicação, veja a Figura 4.11. Assim, quando chegar próximo ao nível máximo de faltas, o aluno

Figura 4.10: Tela Principal do Menu Faltas.

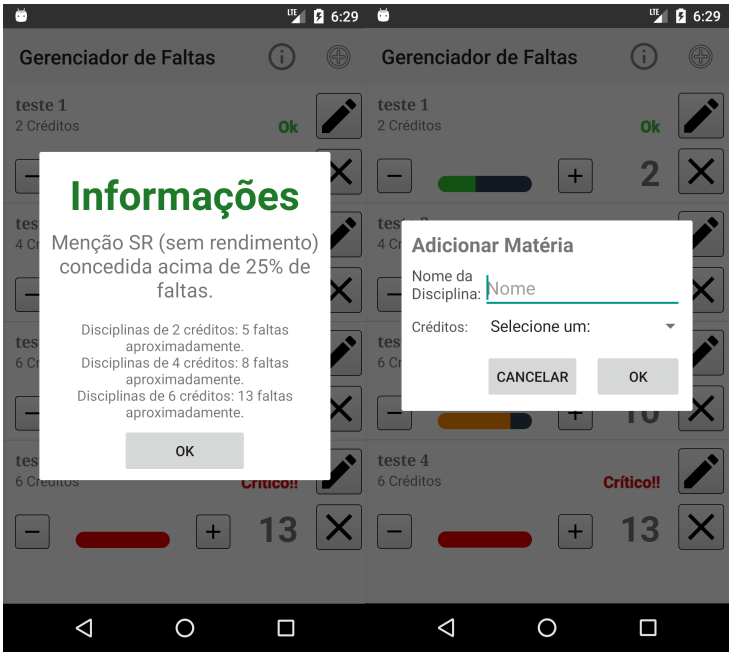


Figura 4.11: Itens do Menu Faltas.

pode se precaver de uma reprovação por falta, a qual implica em uma menção SR (Sem Rendimento) que equivale a nota 0. Essa menção não é benéfica para o currículo estudantil e pode ser um fator negativo no histórico de formação do aluno.

### 4.4.4 Documentos

O *menu* Documentos apresenta diversos documentos organizacionais, de regimento interno, burocracias processual, regras, guia e inferências sobre a história, o funcionamento e os serviços da Universidade de Brasília. Esses documentos foram selecionados pelos autores do aplicativo como sendo os documentos necessários para o usuário da Universidade, nos quais é possível encontrar informações úteis sobre os procedimentos acadêmicos, órgãos e serviços disponibilizados à comunidade acadêmica. Esses documentos complementam as demais iniciativas de boas-vindas, e serve como primeira fonte de consulta para inserção na vida universitária na UnB, conforme apresentado na Figura 4.12.

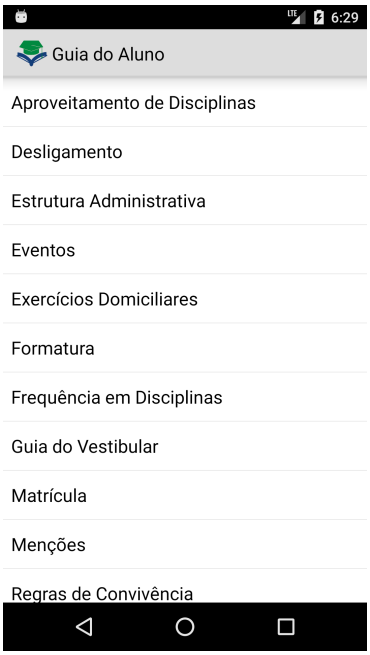


Figura 4.12: Tela Inicial do Menu Documentos.

Esse *menu* compartilha informações que são úteis no cotidiano acadêmico da UnB, começando com uma breve apresentação da história, da estrutura de gestão acadêmica e dos serviços de apoio disponibilizados aos estudantes. Em seguida, são apresentados os principais procedimentos previstos no calendário acadêmico. Uma lista é fornecida no final deste guia com os respectivos endereços e mapas de localização das unidades acadêmicas, administrativas, de apoio e de serviços, para ajudá-lo no reconhecimento dos espaços e na sua movimentação nos campi.

### 4.4.5 Tarefas

O *menu* Tarefas é um sistema gestor de tarefas baseado em atividades simples a serem cumpridas no âmbito acadêmico e pessoal. Esse *menu* contém um botão para adicionar novas tarefas, as quais, após serem cumpridas, podem ser excluídas, veja a Figura 4.13.

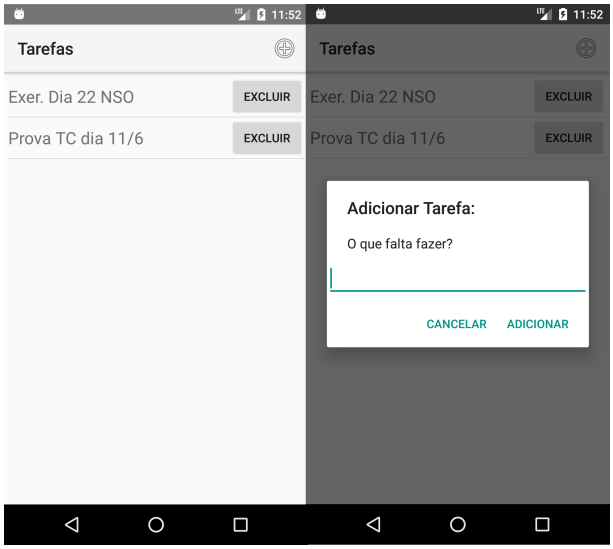


Figura 4.13: Tela do Menu Tarefas.

Após adicioná-las, o aplicativo gera uma lista com todas as atividades a serem exercidas, facilitando a gestão do aluno quanto suas atividades. Essa lista é gravada diretamente em um banco de dados do próprio celular, e quando deletada, é excluída do banco de dados, evitando o de acúmulo de dados desnecessários.

### 4.4.6 Telefones

O *menu* Telefones é uma compilação feita com recolhimento de arquivos e a criação de um acervo virtual com os telefones das diversas unidades da Universidade. O objetivo desse catálogo virtual é para facilitar a comunicação e a agilidade para entrar em contato com qualquer departamento, setor administrativo, secretaria e outros setores de serviço que compõem a Universidade, conforme apresentado na Figura 4.14.

Assim, ao pesquisar, utilizando a barra de busca por um telefone, e selecionar um número de telefone referente ao resultado da pesquisa, o aplicativo abre o discador do celular com o número selecionado, para que seja efetuada uma ligação. Essa ligação é de responsabilidade da operadora do usuário, assim o aplicativo não se responsabiliza por gastos excedentes, visto que o Guia do Aluno UnB é um aplicativo gratuito.

Figura 4.14: Imagem da Função Telefones.

### 4.4.7 MatriculaWeb

O *menu* MatriculaWeb é um acesso direto ao *site* de matrículas da Universidade de Brasília. O calendário de matrícula é divulgado no portal da UnB, veja a Figura 4.15. A matrícula em disciplinas de graduação na UnB tem o suporte informatizado, dado pelo Sistema de Graduação (SIGRA) e pelo sistema de matrícula via Internet (Matrícula Web).

Para o estudante ter acesso ao sistema Matrícula Web, é preciso ter *login* e senha, de modo que o *login* é o número de matrícula, e a senha é fornecida no momento do registro do aluno na universidade. Todos os pedidos de matrícula realizados pelo Matrícula Web são processados pelo SIGRA, durante o processamento coletivo de matrícula, considerando os seguintes critérios de prioridade: posição da disciplina pleiteada com relação à posição do estudante no fluxo, modalidade da disciplina pleiteada, rendimento do estudante na disciplina pleiteada, posição absoluta do estudante no fluxo, desempenho do estudante no último período, e prioridade do pedido.

### 4.4.8 Site UnB

O *menu* Site UnB é um acesso ao *site* principal da Universidade de Brasília. No portal da UnB (www.unb.br) encontram-se todas as informações úteis ao estudante em busca permanente pela transparência da Universidade de Brasília, que dispõe de uma série de publicações, núcleos e serviços para prestar contas e dialogar com a sociedade, além de

Figura 4.15: Imagem da Função MatrículaWeb.

prestar atendimento ao aluno e informações ao público, conforme apresentado na Figura 4.16.

Figura 4.16: Imagem da Função Site UnB.

## 4.5 Considerações Finais

Neste capítulo foi apresentado o aplicativo Guia do Aluno UnB. Para isso, foram apresentados o levantamento de dados geográficos da Universidade de Brasília e os resultados de uma pesquisa de campo elaborada com objetivo de analisar o comportamento dos usuários da Universidade, e a estrutura tecnológica que a mesma propicia. Em seguida, apresentaram-se as funcionalidades da aplicação que são divididas entre um sistema georreferenciadas e um sistema de tarefas. No primeiro, apresentou-se parte da aplicação que sintetiza a geografia da Universidade em um sistema digital, e utiliza da mesma para gerar um mapa catalogado da Universidade. No segundo, apresentou-se o sistema de tarefas cotidianas dos usuários da Universidade.

No próximo capítulo, será abordado o resultado de testes dessas funções apresentadas, mostrando uma avaliação geral do desempenho da aplicação e também dados relacionados a usabilidade do aplicativo.

# Capítulo 5

# Resultado

Este capítulo visa apresentar a avaliação geral acerca do desempenho da aplicação, metodologias de testes, resultados e também usabilidade do aplicativo em termos práticos, e a sua eficiência quanto a utilidade pública. Assim sendo, neste capítulo serão descritas os ambientes nos quais foram feitos os testes do aplicativo, as metodologias de testes, a validação e a usabilidade do aplicativo.

## 5.1 Testes

Os testes de dados, usabilidade e validação foram categorizados e catalogados com critérios de engenharia de software, incluindo correção de *bugs* em função de mensagens de erro inesperadas. Além disso, também foram levados em consideração o tempo de resposta do aplicativo, bem como a precisão ao usar os dados.

### 5.1.1 Testes para Validação da Posição

Durante o processo de levantamento de requisitos, foram criados aplicativos teste nas plataformas Android Studio para desenvolvimento nativo e Ionic [87] para o desenvolvimento híbrido entre as plataformas iOS e Android. Os testes foram realizados no *smartphone* Samsung Galaxy Note 3, no qual ambos os aplicativos foram instalados e testados nas instalações da UnB (campus Darcy Ribeiro), mais precisamente no térreo e subsolo do Insituto Central de Ciências - ICC.

Os testes realizados foram de acordo com a funcionalidade do aplicativo, onde após inicializado a função Mapa, o dispositivo deverá indicar a posição do usuário naquele momento, por meio do mapa oferecido pela Google Maps. Primeiramente, foram realizados testes no térreo da instalação pois não havia obstáculos que poderiam prejudicar o sinal GPS. Em seguida, foram realizados testes no subsolo do mesmo prédio, a fim de testar

se seria possível utilizar o aplicativo mesmo debaixo de obstáculos que poderiam efetivamente atrapalhar o desempenho da localização do usuário [88]. Ao todo foram feitos 3 vezes cada teste.

De início, foi utilizado o aplicativo feito pelo Android Studio, utilizando somente um módulo do Google Maps e um botão para inicializar a busca pela posição. Ao ativar a função de busca, no térreo do prédio, o aplicativo imediatamente mostrou no módulo de Mapa, o prédio ICC, porém com um raio de precisão grande nos primeiros segundos, tomando uma parte do ICC Norte. Porém, após aproximadamente 5 segundos, o sinal GPS mostrou a posição com uma precisão melhor, no qual o raio de precisão esteve significativamente menor e mais preciso.

Ao fazer os testes no subsolo, após reiniciar o dispositivo para refazer a busca, debaixo de obstáculos que poderiam comprometer a localização, o resultado foi relativamente diferente, pois houve uma demora maior ao buscar a posição, e quando buscou, o *smartphone* mostrava um raio de precisão pouco pior em relação ao teste anterior, demorando segundos a mais para buscar uma posição melhor em relação à busca no térreo, mostrando uma posição um pouco diferente do que o teste anterior.

Em seguida, foram feitos testes usando o aplicativo feito em Ionic, usando a biblioteca Cordova [89] para fazer o uso do GPS, primeiramente no térreo do prédio ICC. Em relação ao teste utilizando o aplicativo nativo, o aplicativo feito em Ionic demonstrou uma demora maior ao encontrar a posição do usuário, em torno de 5 segundos a mais, porém nada que implicaria um impedimento no uso do GPS. Todavia, ao fazer o mesmo teste no subsolo do prédio, o resultado foi ainda pior, só foi possível encontrar o local em todos os testes corretamente aproximadamente, 25 segundos, sendo que em um dos testes, inicialmente, o GPS indicou um caminho completamente errado, com uma margem de precisão pouco pior em relação aos testes feitos nativamente.

Dessa forma, conclui-se que o aplicativo nativo tem uma melhor resposta do que o aplicativo feito em Ionic, com um nível de precisão melhor, e com um tempo de resposta significativamente menor. Por conta dos testes mencionados acima, optou-se por desenvolver o aplicativo Guia do Aluno UnB no Android Studio.

### 5.1.2 Testes de Portabilidade

Para fazer os testes iniciais do aplicativo Guia do Aluno UnB, durante o seu desenvolvimento, foi utilizado um dispositivo:

- Samsung Galaxy Note 3
  - Sistema operacional: *Android* 5.0 (*Lollipop*);
  - Processador: Quad-core 2.3 GHz;

- Tela: 5,7 Polegadas;
- Memória RAM: 3 GB;
- Memória Interna: 32 GB.

Após obter resultados satisfatórios durante o desenvolvimento do aplicativo, foram utilizados outros quatro dispositivos, conforme apresentado na Figura 5.1, os quais foram:

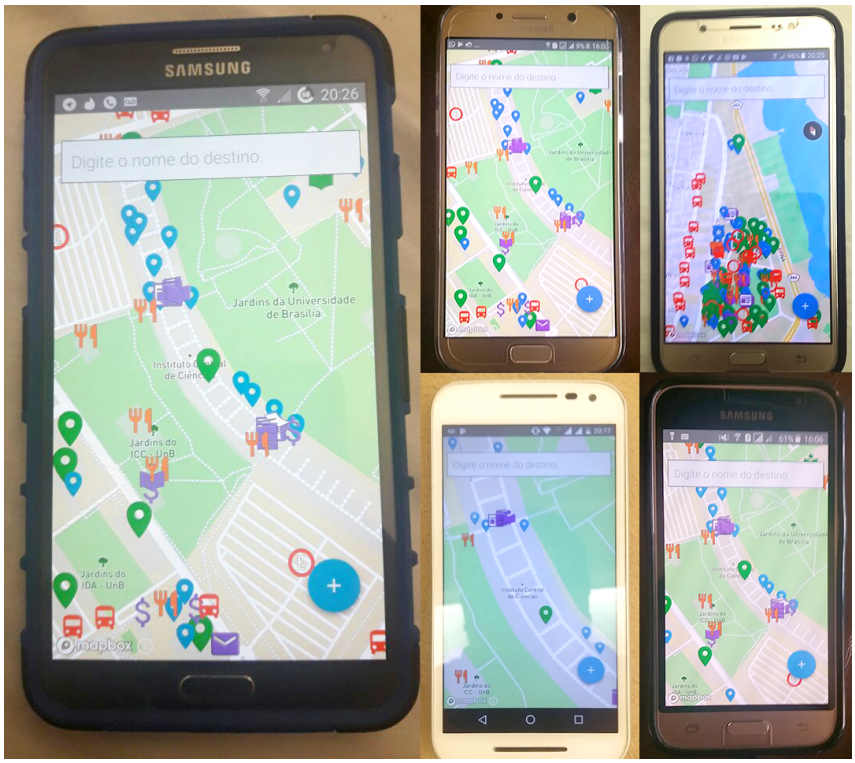


Figura 5.1: Imagens de *Smartphones* Usando o Aplicativo.

- Motorola Moto G (Terceira Geração)
  - Sistema operacional: *Android* 6.0 (*Marshmallow*);
  - Processador: Quad-core 1.4 GHz;
  - Tela: 4,5 Polegadas;
  - Memória RAM: 2 GB;
  - Memória Interna: 16 GB.
- Samsung Galaxy J7 Metal (Versão 2016)
  - Sistema operacional: *Android* 6.0.1 (*Marshmallow*);
  - Processador: Octa-Core 1.6 GHz;

  - Tela: 5,5 Polegadas;
  - Memória RAM: 2 GB;
  - Memória Interna: 16 GB.

- Samsung Galaxy J1
  - Sistema operacional: *Android* 5.1.1 (*Lollipop*);
  - Processador: Quad-core 1.3 GHz;
  - Tela: 4,3 Polegadas;
  - Memória RAM: 1 GB;
  - Memória Interna: 8 GB.

- Samsung Galaxy A5 (Versão 2017)
  - Sistema operacional: *Android* 6.0.1 (*Lollipop*);
  - Processador: Octa-core 1.9 GHz;
  - Tela: 5,2 Polegadas;
  - Memória RAM: 3 GB;
  - Memória Interna: 32 GB.

Assim, nota-se que o aplicativo apresentou sucesso no teste de portabilidade, pois todos os dispostivos utilizados para teste executam-no sem apresentar problemas que impossibilitam o uso das funções do aplicativo.

### 5.1.3 Testes para Validação de Dados

Quando o usuário necessita enviar dados de entrada para o aplicativo, há uma probabilidade de ocasionar erros, dependendo dos dados enviados. A fim de evitar problemas, o aplicativo Guia do Aluno UnB conta com funções que possam prevenir possíveis bloqueios quando um usuário tentar executar uma determinada função. Assim, existem três funções que foram implementadas as validações de dados: Faltas, Mapa e Tarefas.

A função Faltas necessita que o usuário obrigatoriamente informe dois atributos ao adicionar uma matéria e salvar no banco de dados. Esses atributos possuem as seguintes regras:

- Nome da Disciplina.
  - O usuário não consegue salvar a matéria se o nome da matéria estiver em branco;

  - O usuário deverá informar um nome no qual não seja idêntico ao nome de uma outra matéria previamente salvo no banco de dados.

- Número de créditos.
  - O usuário, obrigatoriamente, deverá selecionar uma das opções do controle giratório.

Caso nenhuma das regras citadas acima seja cumprida, o aplicativo emite um alerta na tela conhecido como *Toast*, que informa o erro cometido pelo usuário através de uma breve mensagem na parte inferior da tela. A Figura 5.2 mostra uma simulação de tais erros, e suas devidas mensagens de alerta. Na tela da esquerda na Figura 5.2, quando o usuário tenta salvar uma matéria cujo nome não há dados, o aplicativo não salva os dados, emite um alerta informando o erro, e assim substitui a cor do campo "Nome da matéria" para vermelho, deixando claro o campo que está errado.

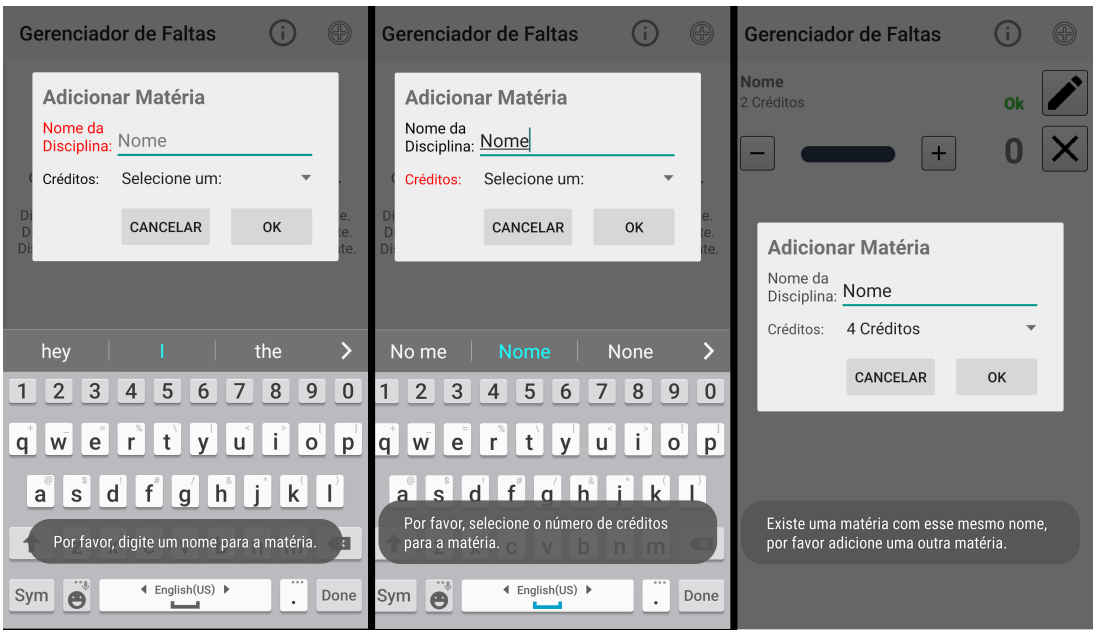


Figura 5.2: Telas Mostrando a Validação de Dados na Função Faltas.

O mesmo acontece quando o usuário tentar salvar o campo "Número de créditos" com a opção "selecione um", será emitido uma mensagem de alerta, e assim substituirá o nome do campo em vermelho conforme apresentado na tela do meio na Figura 5.2. A tela da direita da Figura 5.2 é mostrada quando um usuário deseja salvar uma determinada matéria com um nome, cujo mesmo já está gravado em uma outra matéria. Isso é considerado um erro porque os nomes são chaves-primária no banco de dados, o aplicativo acabaria

considerando os dois dados salvos como somente um, ocasionando uma sobreposição dos dados.

Na função Mapa, quando o usuário digitar o nome de um prédio na barra de pesquisa, o aplicativo deverá sugerir nomes de marcadores previamente salvos, e para o usuário navegar no mapa usando a barra de pesquisa, ele deverá obrigatoriamente usar uma das sugestões dadas pelo aplicativo. Quando o usuário não escolher uma das sugestões dadas pelo aplicativo, e mesmo assim utiliza a barra de busca, o aplicativo retornará uma mensagem de erro, conforme mostrado na Figura 5.3.

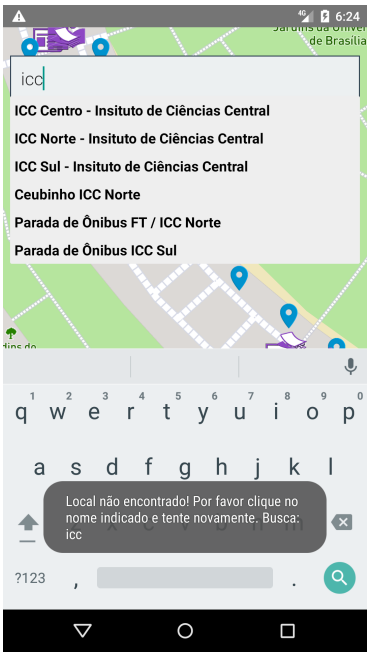


Figura 5.3: Tela Exibindo a Validação de Dados na Função Mapa.

Esta mensagem de erro mostra que o usuário não deve digitar somente o nome de uma instalação, e o aplicativo deverá fazer automaticamente a busca do prédio com base somente no texto escrito. De acordo com a mensagem de erro, o usuário deverá ficar ciente que é necessário utilizar as sugestões dadas pelo aplicativo, e assim navegar através da barra de pesquisa somente por meio dessa. Isso evite que o usuário procure por uma instalação inexistente na UnB.

A função Tarefas possui o mesmo esquema de funcionalidade da função Faltas, porém é menos complexa. A função Tarefas necessita salvar somente um dado, o qual é o texto da tarefa a ser adicionada. Assim, como a Tarefa necessita somente desse dado e ele é usado como um dado primário no banco de dados, foi implementada a mesma verificação de erro da função Faltas, conforme apresentado na Figura 5.4. Para evitar erros de conflito,

o aplicativo deverá permitir o armazenamento de uma tarefa no banco de dados se, e somente se, o nome da tarefa for único, ou seja, essa tarefa não existir no banco de dados.

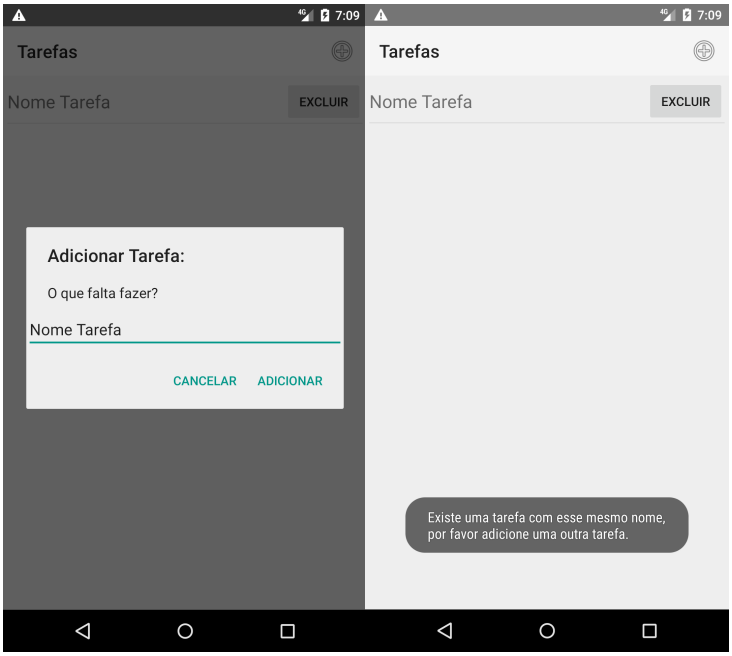


Figura 5.4: Tela Exibindo a Validação de Dados na Função Tarefas.

Os métodos apresentados nesta seção foram feitos para prevenir eventuais erros ao validar dados digitados pelo usuário, onde são eficazes e importantes para o funcionamento do programa. Ne seção seguinte, serão abordados os testes de acesso a Internet, os quais mostram o que o programa deverá fazer quando não há acesso a Internet.

### 5.1.4 Testes de Acesso a Internet

O acesso a Internet é um dos recursos primordiais que o aplicativo Guia do Aluno UnB necessita utilizar para o funcionamento de algumas de suas funções. Tais recursos são obtidos seja através do acesso ao Wi-Fi, ou por meio do uso de uma conexão de dados móvel. Os testes de acesso a Internet têm como principal objetivo verificar como o aplicativo comporta-se durante suas funcionalidades que necessitam do uso da Internet, nos momentos em que não há acesso à rede.

Os recursos utilizados do *web service* permite que o usuário acesse as informações das matérias por meio de um arquivo JSON. A fim de economizar o uso de espaço de memória ao fazer a transferência do aplicativo no dispositivo do usuário, o arquivo que a função Disciplinas usa está hospedado em um serviço de armazenamento em nuvem. Ou seja, toda vez que o usuário for executar a função Disciplinas, é necessário que o dispositivo

tenha acesso a conexão de dados para que ele consiga efetivamente fazer a transferência do arquivo JSON e fazer o uso do mesmo.

Com base nesses dados, foram feitos testes ao acessar a função Disciplinas sem acesso de dados qualquer, seja Wi-Fi ou rede móvel. Conforme é apresentado na Figura 5.5, o usuário recebe uma mensagem *toast* na qual é dito que não há conexão de dados, inferindo assim que o dispositivo não está com uma conexão de dados ativa. Esse erro também pode ocorrer caso o servidor de armazenamento em nuvem estiver *offline*, porém em nenhum momento durante os testes ou o desenvolvimento do aplicativo, os servidores que possuem o arquivo JSON falharam.

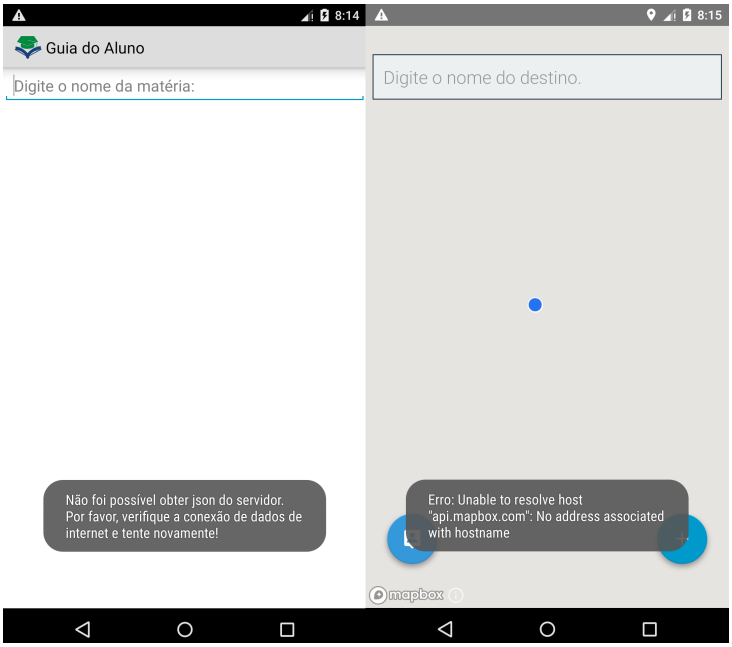


Figura 5.5: Tela Exibindo Mensagens de Erro na Função Disciplinas e Mapa.

A função Mapa necessita utilizar ambos os recursos de GPS e conexão de dados, sendo que neste último é necessário obter as imagens das camadas de navegação do mapa, bem como também calcular a rota entre dois pontos no mapa. Assim, quando a função Disciplinas é carregada, a primeira tela que é exibida é o campus Darcy Ribeiro por padrão, sendo que se ele já tiver sido carregado anteriormente, mostrará as imagens do mapa como se estivesse com acesso de dados. Porém, ao navegar em qualquer outra área que não tenha sido previamente carregada, a camada do mapa fica cinza, esperando que as imagens dessas camadas sejam carregadas. Na Figura 5.5 é possível ver que não há imagens do mapa em volta, porém é possível verificar que foi encontrada a posição do dispositivo. Ao tentar fazer o uso do cálculo da rota ainda sem conexão de dados, o aplicativo retorna uma mensagem *toast* genérica, na qual relata que não há conexão de dados com o servidor.

Dessa forma, concluí-se que o uso do acesso de dados é importante para o desempenho do aplicativo, de fato as funções que necessitam do uso de dados da Internet não conseguem efetivamente executar suas funções. Mas, mesmo com uma conexão de dados ativa relativamente lenta, o aplicativo consegue fazer suas devidas operações com éxito.

### 5.1.5 Testes de Uso do GPS

Conforme fortemente citado anteriomente, o aplicativo Guia do Aluno UnB oferece um guia por meio de mapas, o qual usa o módulo de GPS do dispositivo para mostrar a posição do usuário. Assim sendo, foram realizados vários testes com o uso do GPS durante o desenvolvimento, e também após a versão final. Os testes consistem em verificar o funcionamento do módulo GPS do aparelho em geral, verificar a posição do usuário e sua precisão, e o uso do modo bússola.

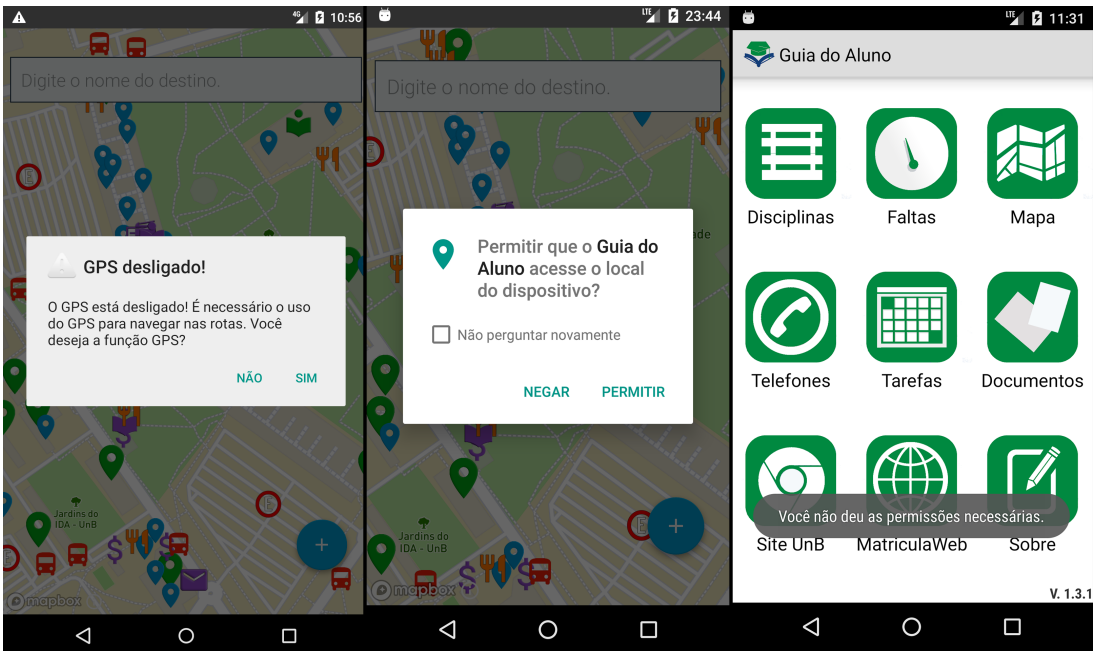


Figura 5.6: Telas Mostrando Processos de Verificação do Uso do GPS.

Após conseguir entrar na função Mapa, o usuário pode ativar a função de posicionamento para que seja exibido a posição do usuário na tela do dispositivo. Porém, o aplicativo passa por dois processos de verificação, conforme mostrado na Figura 5.6.

O primeiro processo de verificação antes de ativar o posicionamento é o estado do GPS. Se o módulo de GPS estiver desligado, o aplicativo deverá informar o usuário que o mesmo está desativado conforme mostrado na tela da esquerda na Figura 5.6. Se o usuário selecionar a opção 'Sim', o dispositivo direciona-o para uma configuração do GPS, na qual é possível ativar a função GPS para o uso.

Porém, ao retornar na função Mapa, após tentar ativar o módulo de posicionamento novamente, se o dispositivo usar um sistema Android cuja a versão seja 6.0 (nível API 23) ou acima, o aplicativo deverá pedir por permissão do uso do GPS como mostrado na tela do meio da Figura 5.6. O pedido de permissão só acontece uma vez, se o usuário negar o pedido, o aplicativo fecha a função Mapa e retorna ao *menu* principal com uma mensagem de erro, conforme apresentado na tela da direita da Figura 5.6. Quando o usuário acessar novamente a função Mapa, o mesmo processo de concessão de pedido acontece, após concedido, o aplicativo consegue exibir a posição do usuário no mapa.

Figura 5.7: Imagem Mostrando a Precisão do GPS.

A precisão da posição do GPS tem um papel muito importante para a exibir as posições dos usuários, porém essa precisão não indica exatamente a posição. Por conta disso, quando o GPS mostra a posição do usuário, ele deverá mostrar um segundo círculo em volta da posição do usuário, onde este é considerado a precisão do GPS, conforme apresentado na Figura 5.7.

Durante o início do desenvolvimento do aplicativo, a biblioteca Mapbox [90], que gera rotas e determina a posição do usuário utilizando o OpenStreet Maps, não tinha uma precisão boa o suficiente. No inicio do ano de 2016, a biblioteca estava na versão 3.0.0, mas a precisão do GPS usando essa biblioteca, por exemplo, cobria todo o prédio do Departamento de Ciência da Computação e Estatística ao requisitar a localização dentro

do mesmo prédio, o que era longe de ser satisfatório. Porém, no dia 21 de Julho de 2017, a atualização da biblioteca para a versão 5.1.1 trouxe uma grande melhora na precisão, conforme mostrado na Figura 5.7. No decorrer das atualizações da biblioteca, a precisão foi apresentando melhoras significantes, o que tornou o uso da localização na ferramenta mais preciso.

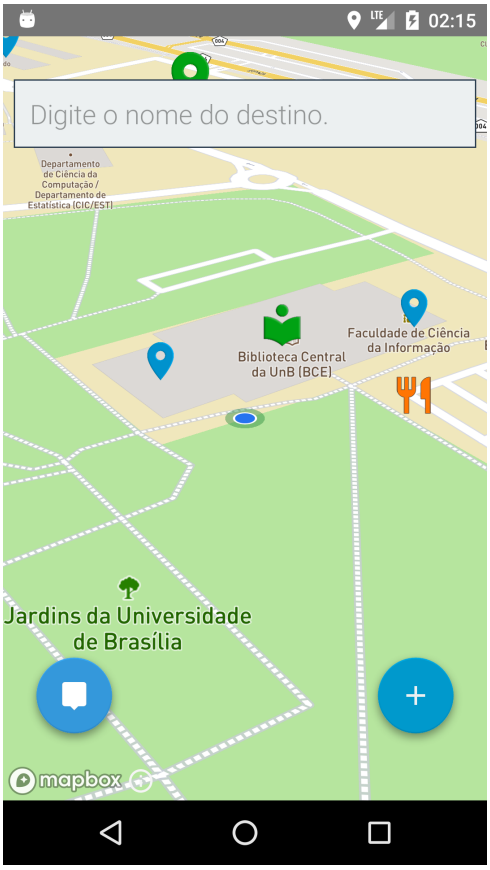


Figura 5.8: Imagem Mostrando o Modo Bússola do GPS.

O aplicativo também conta com um modo bússola, no qual é possível mudar de posição o sentido da tela automaticamente de acordo com a orientação do dispositivo nos pontos cardeais. Para usar esse modo, o celular deverá ficar na posição vertical, jamais deitado sobre uma superfície. Após uma série de testes, o modo apresentou ser preciso, de acordo com a direção que ele estiver apontando, como mostrado na Figura 5.8. Porém, com o decorrer do uso, o dispositivo perde um pouco a precisão, apontando direções com mínimas diferenças em relação a anteriomente, mas não atrapalha à experiência do usuário.

Essa seção conclui-se que os testes obtiveram resultados satisfatórios, sendo que o objetivo do software em atender aos requisitos levantados e descritos, focando nos critérios de utilidade e na usabilidade, foram cumpridos de maneira eficiente.

Na seção seguinte serão apresentados os dados referentes as availações dos usuários por meio da plataforma *Google Play*.

## 5.2 Validação

Após o término do desenvolvimento do aplicativo, e os testes apresentados na seção anterior, foi gerado uma versão de testes aberta para quaisquer usuários da plataforma *Google Play* [91] utilizar. O objetivo dessa versão foi verificar se o aplicativo atingia os objetivos especificados para auxiliar a vida acadêmica dos usuários.

No dia 1 de Julho, foi enviada a versão 1.0.0 para *download* no *Google Play*, na qual qualquer usuário com um dispositivo Android, com uma versão igual ou superior a 4.4, poderia fazer a transferência, e assim utilizar o aplicativo. Na versão 1.1.1 foram corrigidos erros relatados por alguns usuários e foi aprimorado a usabilidade de funções, mas foi a partir dessa versão que o aplicativo foi divulgado para a availação dos usuários.

A divulgação foi feita por meio de várias páginas relacionadas a UnB, no Facebook e nas instalações da Universidade de Brasília, como é mostrado na Figura 5.9 através da plataforma *Google Play Console*. Assim, no dia 10 de Agosto de 2017, o aplicativo atingiu a excelente margem de 1.153 *downloads*, e um total de 101 avaliações feitas espontaneamente pelos usuários. A avaliação média obtida foi de 4,84, no total de 5 estrelas, ou seja, o aplicativo obteve 96,8% de satisfação dos usuários.

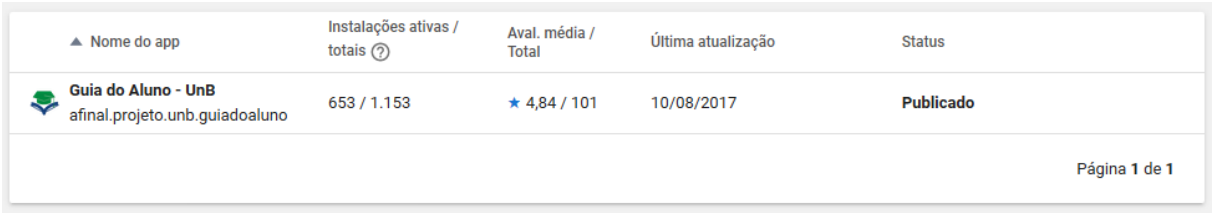


Figura 5.9: Imagem de *Status* do Aplicativo Guia do Aluno UnB [12].

O público que avaliou o aplicativo era, em sua maioria, composto por estudantes da Universidade de Brasília. As recomendações foram positivas e os comentários benevolentes quanto a qualidade da aplicação e suas funcionalides. As avaliações também ajudaram ao aperfeiçoamento do aplicativo, pois relataram alguns erros em versões específicas do Android, as quais foram corrigidas em versões posteriores. Algumas availações deram sugestões de implementações, o que poderiam ser utilizadas como base para sugestões de melhorias em versões futuras. Algumas dessas avaliações estão ilustradas na Figura 5.10.

A avaliação na *Google Play* era feita em dois passos simples, sendo que o usuário deverá obrigatoriamente fazer o *download* do aplicativo no seu dispositivo, e após o término da

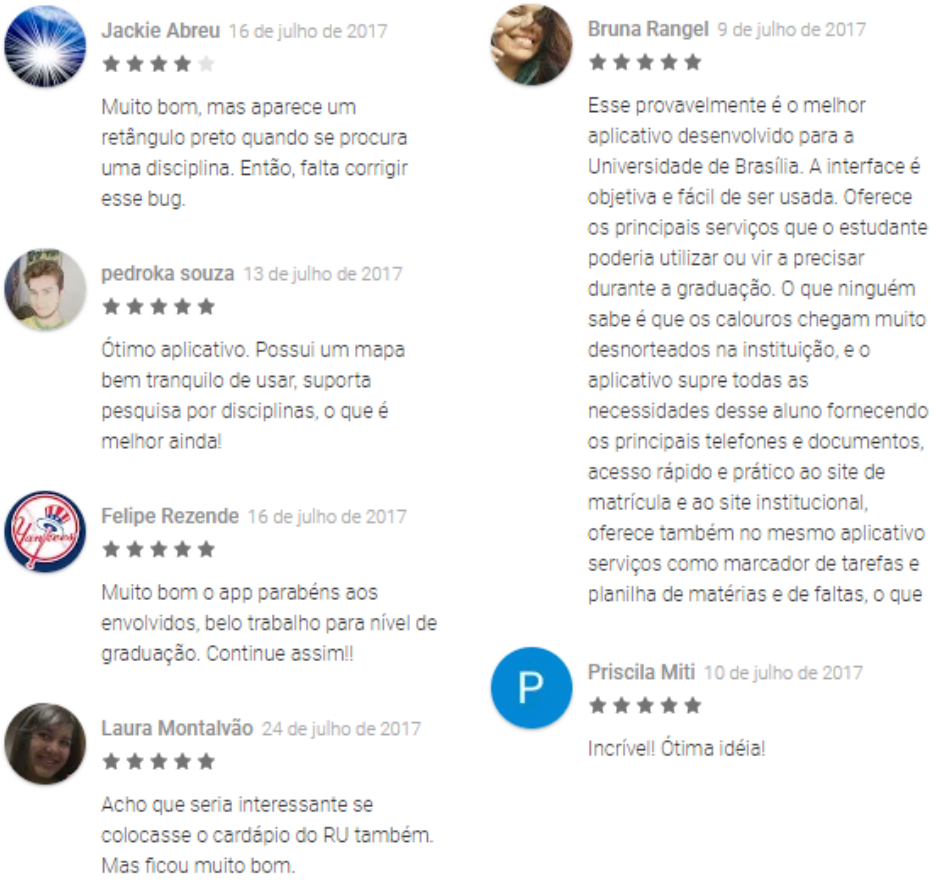


Figura 5.10: Exemplo de Comentários e Avaliações [12].

transferência, a plataforma *Google Play* permite que o usuário faça a avaliação entre uma a cinco estrelas. Assim, após o usuário avaliar, ele poderá acrescentar comentários em sua availação, porém esse último passo não é obrigatório. A avaliação teve um total de 101 avaliações como mostra a Figura 5.11, sendo:

- 5 estrelas = 91 avaliações;
- 4 estrelas = 7 avaliações;
- 3 estrelas = 1 avaliação;
- 2 estrelas = 1 avaliação;
- 1 estrelas = 1 avaliação;
- **Total**: 101 avaliações.

Por fim, as avaliações mostram que o aplicativo obteve êxito em atender as demandas dos usuários, atingindo 96,8% de satisfação dos usuários. Além disso, o aplicativo teve

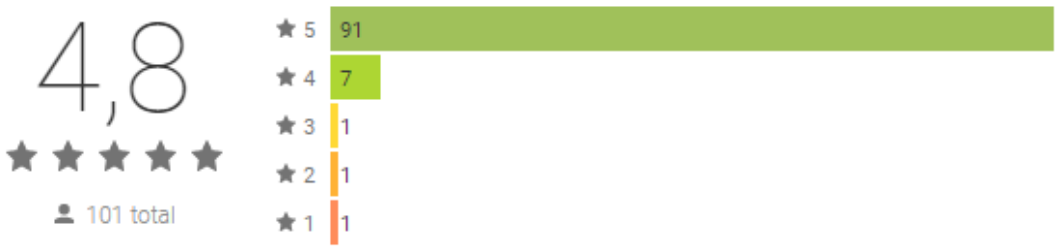


Figura 5.11: Gráfico com Todas as Avaliações do Aplicativo Guia do Aluno UnB [12].

um grande número de *downloads*, o que tem demonstrado que os alunos estão fazendo uso efetivo dele. Com base nesses dados, o programa Guia do Aluno UnB tem um grande potencial em ser uma ferramenta de amparo para os alunos da Universidade de Brasília.

## 5.3 Considerações Finais

Este capítulo expôs os testes feitos no aplicativo Guia do Aluno UnB, mostrando a capacidade de evitar possíveis erros ao manipular dados que se encontram no dispositivo do usuário. Para tal, foram descritos os dispositivos nos quais foram feitos os testes e as suas especificações. Logo em seguida, foram apresentados os testes com aplicativos nativos e híbrido, a fim de entender qual seria mais adequado para o propósito do aplicativo. Em seguida, foram apresentados os testes feitos ao manipular dados do usuário, para prevenir erros e conflitos. Respectivamente, foi apresentada a sequência de testes feitos quando não há Internet, no qual verificou-se a eficiência ao orientar o usuário onde há falta de sinal de dados. Logo em seguida, foram apresentados os testes feitos com o uso do GPS, bem como mostrar os processos de verificação de uso, a sua precisão e o uso do modo bússola. Por fim, foram apresentadas as avaliações dos usuários.

Para finalizar este trabalho, o Capítulo 6 apresenta a conclusão do aplicativo Guia do Aluno UnB.

# Capítulo 6

# Conclusão

O software Guia do Aluno UnB é um aplicativo móvel de mapeamento georreferenciado da Universidade de Brasília e é composto por um conjunto de menus com ferramentas destinadas a Universidade de Brasília. Dentre essas ferramentas, há um mapa interativo com os estabelecimentos, as salas de aulas, as secretarias e os departamentos da universidade. No âmbito administrativo, o aplicativo propõe ferramentas de gestão da informação, tais como acesso aos documentos institucionais da universidade e gestão de faltas e tarefas, assim como acesso aos principais portais da UnB. O software é detido pela Fundação Universidade de Brasília e registrado pelo orgão INPI.

O aplicativo destina-se aos professores, alunos, profissionais diversos e visitantes da Universidade de Brasília com o objetivo de ser uma ferramenta facilitadora a gestão da informação, minimizando dificuldades burocráticas e maximizando acesso a informação e guiando esses entes em prol de uma locomoção facilitada e eficiente, assim como promissão em atividades cotidianas e acadêmicas.

A interface simples, arquitetada especificamente para os aspectos da usabilidade, permite que o usuário acesse várias funções do aplicativo sem qualquer esforço. O mapa que possui as instalações dos prédios de todos os *campi* da Universidade de Brasília, facilita o empenho ao localizar as salas, buscar locais que possuem serviços como xerox, lanchonetes, postos de gasolina etc, bem como os horários de funcionamento dos prédios e os respectivos telefones. A função mapa também conta com a busca de rotas para os prédios e quaisquer lugares que o mapa aponta, e por meio de ferramentas como diferentes camadas de mapas para navegação, o uso de prédios 3D e o modo bússola facilita a busca de qualquer destino.

As funções Faltas e Tarefas foram feitas, em específico, para os alunos da Universidade de Brasília, com o objetivo de facilitar o gerenciamento de tarefas e as faltas das matérias que estão sendo cursadas pelo aluno, nos quais foram lhes ofertadas, por meio do uso de banco de dados. Além disso, a função Documentos possibita o usuário acessar uma

lista de documentos úteis, com informações pertinentes a UnB, tais como aproveitamento de disciplinas, desligamento, estrutura administrativa, eventos, exercícios domiciliares, formatura, frequência em disciplinas, guia do vestibular, matrícula, menções, regras de convivência, reintegração, revisão de menção, trancamento e vida acadêmica.

Como o aplicativo foi disponibilizado na plataforma *Google Play*, foi possível que os usuários avaliassem o aplicativo Guia do Aluno UnB, conseguindo, de fato, um exímio resultado por meio de mais de 1.150 usuários, onde o programa obteve 96,8% de satisfação, por meio dos votos de 101 avaliadores. Assim, conclui-se que o aplicativo atendeu as expectativas e as necessidades dos usuários.

Dentre as possíveis melhorias e trabalhos futuros, há a possibilidade de implementar as seguintes funcionalidades e advindos ao aplicativo:

- Criar categorias de marcadores na função Mapa para apagar ou mostrar diferentes tipos de marcadores;
- Salvar dados do usuário utilizando inteligência artificial;
- Integrar um gerenciador de notas junto a função Faltas;
- Integração com outros dispositivos e versão web;
- Tour e tutorial no início do aplicativo;
- Melhorias nos filtros de busca;
- Incremento de funcionalidades e manutenção de próximas versões;
- Acesso ao cardápio do RU;
- Integração com o sistema do Matrícula Web da UnB.

# Referências


[1] Ribeiro, J. A.: *Sistema de informação geográfica (sig)*, 2011. Disponível em: http://www.geomatica.eng.uerj.br/docentes/araujo/_media/te:aula24.jpeg?w=450. Acesso em Julho de 2017. ix, 5

[2] Melo, D.H; Silva, H.M; Brito P.L; Menezes L. A: *Mapeamento automatizado em cig*, 2015. Disponível em: http://mundogeo.com/blog/2015/09/10/sistema-de-informacao-geografica-uma-analise-sobre-o-desenvolvimento-tecnico-tecnologico-academico-e-conceitual/. Acesso em Julho de 2017. ix, 7

[3] Européia, Agência Espacial: *Sistema de posicionamento global (gps)*, 2011. Disponível em: http://3.bp.blogspot.com/-cyzmoSz87do/Tt3ty33Q7iI/AAAAAAAAAYA/P7XF7HqSMzc/s640/image005.png. Acesso em Julho de 2017. ix, 10

[4] *Open Street Map*: *Modelo do mapa osm*, 2016. Disponível em: https://www.openstreetmap.org/way/332888307#map=16/-15.7626/-47.8711. Acesso em Maio de 2016. ix, 12

[5] Taverner, C.: *Estrutura neo4j spatial do mapa*, 2017. Disponível em: http://neo4j-contrib.github.io/spatial/0.24-neo4j-3.1/index.html. Acesso em Agosto de 2017. ix, 13, 14

[6] *Open Street Map*: *Gráfico de uso de pontos do openstreetmap no brasil*, 2014. Disponível em: http://wiki.openstreetmap.org/wiki/Stats_-_Brasil. Acesso em Maio de 2016. ix, 17

[7] Oliveira, Edson Mendes De: *Exemplo de código* Java, 2017. Disponível em: http://videos.web-03.net/artigos/Edson_Mendes_De_Oliveira/IntroducaoJava/IntroducaoJava2.jpg. Acesso em Agosto de 2017. ix, 20

[8] Junior, João Torres: *Funcionamento da* Activity *em* Java, 2017. Disponível em: https://joaotorresjunior.files.wordpress.com/2015/08/2.jpg. Acesso em Agosto de 2017. ix, 22

[9] W3Schools: *Dtd tutorial*, 2017. Disponível em: https://www.w3schools.com/xml/xml_dtd_intro.asp. Acesso em Agosto de 2017. ix, 25

[10] Breeze, Mez: *Android, Apple, Starbucks and Nasa: what inspired these four world-famous logos?*, 2013. Disponível em: http://thenextweb.com/dd/2012/09/08/the-stories-inspiration-behind-world-famous-logos. Acesso em Maio de 2016. ix, 26, 27

[11] Android, Desenvolvedores: *Arquitetura da plataforma*, 2017. Disponível em: https://developer.android.com/guide/platform/index.html. Acesso em Agosto de 2017. ix, 28, 29, 30, 31

[12] Google: Google Play Console, 2017. Disponível em: https://play.google.com/store/apps/details ?id=afinal.projeto.unb.guiadoaluno. Acesso em Agosto de 2017. x, 65, 66, 67

[13] *Android Developers*: *Dashboards*, 2017. Disponível em: https://developer.android.com/about/ dashboards/index.html. Acesso em Agosto de 2017. xi, 27, 28

[14] Batista, Thiago: *A importância de se ter um ambiente de gerenciamento centralizado*, 2017. Disponível em: http://gestaonati.com.br/a-importancia-de-se-ter-um-ambiente-de-gerenciamento-centralizado/. Acesso em Agosto de 2017. 1

[15] Quoos, J. H.: *O que é latitude e longitude?*, 2007. Disponível em: http://coral.ufsm.br/cartografia/index.php?option=com_content&view=article&id=43&Itemid=39. Acesso em Julho de 2017. 4

[16] Filho, J. L; Iochpe, C.: *Introdução a sistemas de informações geográficas com Ênfase em banco de dados*. Universidade Federal de Viçosa, páginas 8–9, 1996. 4

[17] DPI/INPE, Spring: *Introdução ao geoprocessamento*, 2006. Disponível em: http://www.dpi.inpe.br/spring/portugues/tutorial/introducao_geo.html. Acesso em Agosto de 2017. 5

[18] São Paulo, Universidade de: *Sistemas de informação geográfica (sig)*, 2017. Disponível em: http://www.lem.ep.usp.br/Pef411/C̃aio%20Eduardo%20Aoki%20Kac/PEF411.htm. Acesso em Agosto de 2017. 5

[19] Prates, Izabela: *A ciência por trás da informação geográfica*, 2017. Disponível em: http://mundogeo.com/blog/2015/04/15/a-ciencia-por-tras-da-informacao-geografica/. Acesso em Agosto de 2017. 6

[20] Matias, L.F.: *Sistema de informações geográficas (sig): Teoria e método para representação do espaço geográfico*. Universidade de São Paulo, página 111, 2001. 6

[21] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, página 4, 2016. `Disponívelem:http://bdm.unb.br/handle/10483/13169`. 6

[22] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, páginas 4–5, 2016. `Disponívelem:http://bdm.unb.br/handle/10483/13169`. 6

[23] Tobler, W. R.: *A Computer Movie Simulating Urban Growth in the Detroit Region*. Economic Geography, 46(sup1):234–240, 1970. Disponívelem:http://www.tandfonline.com/doi/abs/10.2307/143141. 7

[24] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, página 6, 2016. Disponívelem:http://bdm.unb.br/handle/10483/13169. 7

[25] QGis: *Dados raster*, 2017. Disponível em: http://docs.qgis.org/2.0/pt_BR/docs/gentle_gis _introduction/raster_data.html. Acesso em Agosto de 2017. 7

[26] Faria, Caroline: *Sensoriamento remoto*, 2017. Disponível em: http://www.infoescola.com/cartogra fia/ sensoriamento-remoto/. Acesso em Agosto de 2017. 8

[27] Martins, W.S.; Monteiro, B.R.; Filho J.L.; Rocha M.N.: *Problemas e soluções no desenvolvimento de um sig móvel para aplicações urbanas*. Informática Pública, página 70, 2008. 8

[28] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, páginas 7–8, 2016. Disponívelem:http://bdm.unb.br/handle/10483/13169. 8

[29] Martins, W.S.; Monteiro, B.R.; Filho J.L.; Rocha M.N.: *Problemas e soluções no desenvolvimento de um sig móvel para aplicações urbanas*. Informática Pública, página 72, 2008. 8

[30] Fu, P.; Sun, J.: *Web GIS: Principles and Applications*. Esri Press, página 8, 2010. 8

[31] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, páginas 12–13, 2016. Disponívelem:http://bdm.unb.br/handle/10483/13169. 8

[32] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, páginas 9–10, 2016. Disponívelem:http://bdm.unb.br/handle/10483/13169. 9

[33] Branco, Anselmo Lazaro: *Coordenadas geográficas: Latitude, longitude e gps*, 2005. Disponível em: https://educacao.uol.com.br/disciplinas/geografia/coordenadas-geograficas-latitude-longitude-e-gps.htm. Acesso em Agosto de 2017. 10

[34] Lisle, R.J.; Brabham, P.; Barnes J.W.: *Basic Geological Mapping*, volume 5. John Wiley & Sons, 2011. 10

[35] Quora: *Why Are Four GPS Satellites Required To Locate Someone's Position?*, 2017. Disponível em: https://www.forbes.com/sites/quora/2017/01/18/why-are-four-gps-satellites-required-to-locate-someones-position/#7aa65e0f4fc3. Acesso em Agosto de 2017. 11

[36] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, página 8, 2016. Disponívelem:http://bdm.unb.br/handle/10483/13169. 11

[37] Branco, W. G.: *Comune : aplicativo android para questionários e coleta de relatos de usuários de serviços públicos*. Universidade de Brasília, página 15, 2016. Disponívelem:http://bdm.unb.br/handle/10483/13169. 11

[38] Bastin, L.; See, L.; Foody G.; Fonte L.F.: *Usability of VGI for Validation of Land Cover Maps*. International Journal of Geographic Information Science, páginas 1260–1291, 2015. 11

[39] Goodchild, M.F.: *Citizens as Sensors: The World of Volunteered Geography*. GeoJournal 69, páginas 211–215, 2007. 11

[40] Mata, D. Q.; Ferreira, F. G. N.: *Siu : um aplicativo móvel para coleta e visualização de dados geográficos aplicado ao transporte público do df*. Universidade de Brasília, página 26, 2013. Disponívelem:http://bdm.unb.br/handle/10483/6519. 11, 12

[41] Mata, D. Q.; Ferreira, F. G. N.: *Siu : um aplicativo móvel para coleta e visualização de dados geográficos aplicado ao transporte público do df*. Universidade de Brasília, páginas 37–38, 2013. Disponívelem:http://bdm.unb.br/handle/10483/6519. 12

[42] OpenStreetMap: *Accuracy*, 2017. Disponível em: http://wiki.openstreetmap.org/wiki/Accuracy. Acesso em Agosto de 2017. 12

[43] Haklay, M. M.; Weber, P.: *OpenStreetMap: User-generated street maps*, volume 4. IEEE Pervasive Computing, 2008. 13

[44] Doyle, A.; Carl, R.: *Introduction to OGC Web Services*. An Open Geospatial Consortium White Paper, 2001. Disponívelem:http://portal.opengeospatial.org/files/?artifact_id=14973. 15

[45] Lamas, A.R.; Filho, J.L.; Oliveira A.P.; Júnior R.M.A.B: *A mobile geographic information system managing context-aware information based on ontologies*. The First International Workshop on Managing Data with Mobile Devices, páginas 718–727, 2009. Disponívelem:http://www.ubicc.org/files/pdf/1_361.pdf. 15

[46] Harley, M.: *How good is Volunteered Geographical Information? A comparative study of OpenStreetMap and Ordnance Survey datasets*. Environment and Planning B: Planning and Design, 2008. 15

[47] Brunet, K.; Freire, J.: *Cultura digital e geolocalização: a arte ante o contexto técnico-político*. Encontro de Estudos Multidisciplinares em Cultura, 2010. 15

[48] Foundation, The Open Source Geospatial: *Osgeo*, 2017. Disponível em: http://www.osgeo.org. Acesso em Agosto de 2017. 15

[49] OpenStreetMap: *Open Database License*, 2016. Disponível em: http://wiki.openstreetmap.org/wiki/ Open_Database_License. Acesso em Agosto de 2017. 15

[50] Mata, D. Q.; Ferreira, F. G. N.: *Siu : um aplicativo móvel para coleta e visualização de dados geográficos aplicado ao transporte público do df.* Universidade de Brasília, páginas 7–8, 2013. Disponívelem:http://bdm.unb.br/handle/10483/6519. 15

[51] Mata, D. Q.; Ferreira, F. G. N.: *Siu : um aplicativo móvel para coleta e visualização de dados geográficos aplicado ao transporte público do df.* Universidade de Brasília, página 17, 2013. Disponívelem:http://bdm.unb.br/handle/10483/6519. 16

[52] INDE: *WebSite infraestrutura nacional de dados espaciais*, 2017. Disponível em: http://www.inde.gov.br/. Acesso em Agosto de 2017. 16

[53] Silva, Paulo Bernardo: *Decreto número 6.666 de 27 de novembro de 2008*, 2017. Disponível em: http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2008/Decreto/D6666.htm. Acesso em Agosto de 2017. 16

[54] Standardization, International Organization for: *Iso 19115:2003 geographic information metadata*, 2003. Disponível em: https://www.iso.org/standard/26020.html. Acesso em Agosto de 2017. 16

[55] Estatística, Instituto Brasileiro de Geografia e: *Catálogo de metadados*, 2017. Disponível em: http://www.metadados.geo.ibge.gov.br/geonetwork_ibge/srv/por/main.home. Acesso em Agosto de 2017. 16

[56] Goodchild, M. F: *Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0.* International Journal of Spatial Data Infrastructures Research, páginas 24–32, 2007. Disponívelem:http://ijsdir.jrc.ec.europa.eu/index.php/ijsdir/article/view/28/22. 16

[57] Thomas, G. R.: *Toward principles for the design of ontologies used for knowledge sharing.* Int. J. Human-Computer Studies, páginas 907–928, 1995. 16

[58] Kokar, M.M.; Matheus, C.J.; Baclawski K.: *Ontology-based situation awareness.* Information Fusion, páginas 83–98, 2009. 17

[59] Aguiton, C.; Cardon, D.; Smoreda Z.: *Living Maps: New data, new uses, new problems.* Orange Labs, 2009. 17

[60] OpenStreetMap: *Nominatim*, 2017. Disponível em: https://nominatim.openstreetmap.org/. Acesso em Agosto de 2017. 18

[61] OpenStreetMap: *Copyright and License*, 2017. Disponível em: http://www.openstreetmap.org/ copyright. Acesso em Agosto de 2017. 18

[62] Steele, G.; Bracha, G.; Buckley A.; Gosling J.; JOY B.: *The Java Language Specification Java SE 8 Edition*, volume 8. Oracle America, 2015. 19, 23

[63] Byous, Jon: *Java technology: the early years*, 1998. Disponível em: http://java.sun.com/features/1998/05/birthday.html. Acesso em Maio de 2016. 19

[64] Oracle: *The History of Java Technology*, 1995. Disponível em: http://www.oracle.com/technetwork/java/ javase/overview/javahistory-index-198355.html. Acesso em Maio de 2016. 19

[65] Tiobe: *Tiobe index for april 2016*, 2016. Disponível em: http://www.tiobe.com/tiobe_index ?page=index. Acesso em Maio de 2016. 19

[66] Java: *Porque os desenvolvedores de software escolhem o Java*, 2016. Disponível em: https://www.java.com/pt_BR/about/. Acesso em Maio de 2016. 20

[67] Sebesta, R.W.: *Concepts of Programming Languages*, volume 10. Pearson, 2015. 20, 21, 22, 23

[68] Java: *Porque os desenvolvedores de software escolhem o Java*, 2016. Disponível em: https://www.java.com/pt_BR/about/. Acesso em Maio de 2016. 21

[69] Mcqueen, C. M.; Bray, T.; Paoli J.; Maler E.: *Extensible Markup Language (XML) 1.0*, volume 5. W3C, 2008. 23

[70] Zaidan, Fernando Hadad: *O mundo xml*, 2007. Disponível em: http://www.fernandozaidan.com.br/pitagoras/Artigos_Diversos/O%20mundo %20XML.pdf. Acesso em Agosto de 2017. 23, 24

[71] Gamma, E.: *Padrões de Projeto - Soluções Reutilizaveis de Software Orientado a Objetos*, volume 1. Bookman, 2000. 25

[72] Lecheta, R.: *Google Android. Aprenda a criar aplicações para dispositivos móveis*, volume 2. Novatec, 2010. 26, 28

[73] Android, Desenvolvedores: *Overview of Android Memory Management*, 2017. Disponível em: https://developer.android.com/topic/performance/memory-overview.html. Acesso em Agosto de 2017. 26

[74] Android, Desenvolvedores: *ART and Dalvik*, 2017. Disponível em: https://source.android.com/devices/tech/dalvik/. Acesso em Agosto de 2017. 26

[75] Elgin, Ben: *Google Buys Android for Its Mobile Arsenal*, 2005. Disponível em: http://www.webcitation.org/5wk7sIvVb. Acesso em Maio de 2016. 26

[76] Perez, Juan Carlos: *Google releases Android SDK*, 2007. Disponível em: http://www.macworld.com/article/1061005/androidsdk.html. Acesso em Maio de 2016. 27, 31

[77] Twohill, Lorraine: *$10 million for Project 10î00 winners*, 2017. Disponível em: https://googleblog.blogspot.com.br/2010/09/10-million-for-project-10100-winners.html. Acesso em Agosto de 2017. 27

[78] Android, Desenvolvedores: *Add C and C++ Code to Your Project*, 2017. Disponível em: https://developer.android.com/studio/projects/add-native-code.html. Acesso em Agosto de 2017. 28

[79] Matos, Victor: *Lesson 3: Android Application's Life Cycle*, 2013. Disponível em: https://web.archive.org/web/20140222153131/Disponível em: http://grail.cba.csuohio.edu/matos/ notes/cis-493/lecture-notes/Android-Chapter03-Life-Cycle.pdf. Acesso em Agosto de 2017. 30

[80] Android, Desenvolvedores: *Power Management*, 2017. Disponível em: https://source.android.com/devices/tech/power/mgmt. Acesso em Agosto de 2017. 30

[81] Sharma, Ankur: *Google releases Android 6.0.1 factory images for Nexus 6p/6, Nexus 5x/5, Nexus player, Nexus 9 and Nexus 7*, 2015. Disponível em: http://www.gadgetzarena.com/news/google-releases-android-6-0-1-factory images-nexus-6p6-nexus-5x5-nexus-player-nexus-9-nexus-7-2013/. Acesso em Maio de 2016. 31

[82] Android, Desenvolvedores: *Launch Checklist*, 2017. Disponível em: http://developer.android.com/distribute/tools/launch-checklist.html. Acesso em Maio de 2017. 31

[83] Android, Desenvolvedores: *Android Develop Training*, 2017. Disponível em: https://developer.android.com/training/index.html. Acesso em Agosto de 2017. 31

[84] Hass, Hugo: Web services glossary, 2004. Disponível em: https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice. Acesso em Maio de 2016. 33

[85] Fielding, Roy Thomas: Architectural Styles and the Design of Network-based Software Architectures, 2000. Disponível em: http://www.ics.uci.edu/fielding/pubs/dissertation/top.htm. Acesso em Julho de 2016. 34

[86] Brasília, Decanato de Ensino de Graduação da Universidade de: *Número de alunos*, 2016. Disponível em: http://www.unb.br/administracao/decanatos/deg/downloads/cil/ numero_de_alunos.pdf. Acesso em Maio de 2016. 39

[87] Ionic: Ionic, the open source framework for building mobile apps, 2017. Disponível em: https://ionicframework.com/. Acesso em Agosto de 2017. 54

[88] Kumar, Arun: What Interferes with a GPS Signal and Methods to Work with Weak Signal GPS, 2017. Disponível em: http://www.brighthub.com/electronics/gps/articles/43060.aspx. Acesso em Agosto de 2017. 55

[89] Cordova: Cordova Geolocation, 2017. Disponível em: http://ngcordova.com/docs/plugins/geoloc ation/. Acesso em Agosto de 2017. 55

[90] Mapbox: Mapbox Android SDK, 2017. Disponível em: https://www.mapbox.com/. Acesso em Agosto de 2017. 63

[91] Play, Google: Guia do Aluno - UnB, 2017. Disponível em: https://play.google.com/store/apps/details?id=afinal.projeto.unb.guiadoaluno. Acesso em Agosto de 2017. 65