

# CENTRO PAULA SOUZA

---

**FACULDADE DE TECNOLOGIA DE AMERICANA**  
**Curso Tecnologia em Análise e Desenvolvimento de Sistemas**

Lucas Matheus Souza Serra

**DESENVOLVIMENTO DE UM SISTEMA DE ORDEM DE SERVIÇO**  
**MULTIPLATAFORMA UTILIZANDO UM WEB SERVICE**

**Americana, SP**

**2015**

# CENTRO PAULA SOUZA

---

**FACULDADE DE TECNOLOGIA DE AMERICANA**  
**Curso Tecnologia em Análise e Desenvolvimento de Sistemas**

Lucas Matheus Souza Serra

## **DESENVOLVIMENTO DE UM SISTEMA DE ORDEM DE SERVIÇO MULTIPLATAFORMA UTILIZANDO UM WEB SERVICE**

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do (a) Prof.<sup>(a)</sup> Me. Diógenes de Oliveira

Área de concentração: Desenvolvimento de Sistemas

**Americana, S. P.**

**2015**

S496d	<p>Serra, Lucas Matheus Souza</p> <p>Desenvolvimento de um sistema de ordem de serviço multiplataforma utilizando um <i>web service</i>. / Lucas Matheus Souza Serra. – Americana: 2015. 51f.</p> <p>Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza. Orientador: Prof. Me. Diógenes de Oliveira</p> <p>1. Internet – rede de computadores 2. Dispositivos móveis – aplicativos I. Oliveira, Diógenes de II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.</p> <p>CDU: 681.519</p>
-------	--

Lucas Matheus Souza Serra

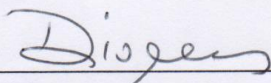
**DESENVOLVIMENTO DE UM SISTEMA DE ORDEM DE  
SERVIÇO MULTIPLATAFORMA UTILIZANDO UM WEB SERVICE**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

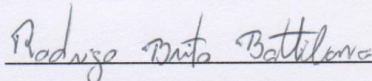
Área de concentração: Desenvolvimento de Sistemas.

Americana, 09 de Dezembro de 2015.

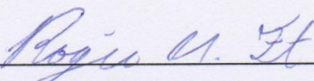
**Banca Examinadora:**



Diógenes de Oliveira (Membro)  
Mestre  
Fatec Americana



Rodrigo Brito Battilana (Membro)  
Bacharel  
Fatec Americana



Rogério Nunes de Freitas (Membro)  
Especialista  
Fatec Americana

## **AGRADECIMENTOS**

À minha família, que sempre me apoiou em minhas decisões e incentivou-me durante o curso.

Ao Professor/Orientador Diógenes de Oliveira, pela orientação desta monografia e pelos ensinamentos.

Aos professores do curso de Tecnologia em Análise e desenvolvimento de Sistemas, pelos ensinamentos e apoio.

## DEDICATÓRIA

Aos meus pais, pelo apoio moral, amor e dedicação que contribuíram para chegar até este momento importante de minha vida.

## RESUMO

Com o avanço da tecnologia, o uso dela tornou-se cada vez mais presente no cotidiano de pessoas e até mesmo de empresas. A tecnologia da informação deixou de ser uma ferramenta de auxílio e passou a ser uma ferramenta fundamental para as organizações atingirem seus objetivos. A presente monografia conceitua o estudo sobre TI dentro de empresas e organizações e a forma com que ela ajuda a atingir seus objetivos. Apresentar o estudo sobre conceitos relacionados a software, engenharia de software, dispositivos móveis e desenvolvimento de sistemas envolvidos no assunto tratado no decorrer da monografia, o qual é o desenvolvimento de um sistema de ordem de serviço para a seção de informática de uma instituição, com o propósito de substituir o sistema atual, o qual é feito de forma manual. Para isto, foram desenvolvidas duas aplicações, uma para a plataforma *web* e outra para plataforma Android, intermediadas por um *Web Service*.

**Palavras Chave:** *Aplicação Web; Aplicação Android; Web Service.*

## **ABSTRACT**

Owing to the technology advances, its use has been becoming part of people's and companies' routines. The information technology passed from an auxiliary tool from a fundamental tool to companies to reach their goals. This monograph conceptualizes the study about IT within companies and organizations and the way it helps to achieve their goals. It also introduces the study of concepts related to software, software engineering, mobile devices and systems development. The latter is about a system development that was made to provide a work order system for a computer section of an institution. It's proposes is to replace their current system which is done manually, to a new one. For this study, it was developed two applications, the first one for web platform and the second one for Android platform. Both are intermediated by a Web Service.

**Keywords:** *Web Application, Android Application, Web Service.*



## SUMÁRIO

<b>INTRODUÇÃO</b> .....	<b>14</b>
<b>1 REVISÃO BIBLIOGRÁFICA</b> .....	<b>16</b>
<b>1.1 Automação e informatização de empresas</b> .....	<b>16</b>
1.1.1 Objetivos da automação e informatização nas empresas .....	16
<b>1.2 Software</b> .....	<b>17</b>
<b>1.3 Dispositivos Móveis</b> .....	<b>18</b>
<b>1.4 Softwares e aplicativos para dispositivos móveis</b> .....	<b>18</b>
1.4.1 Aplicações móveis nativas .....	19
1.4.2 Aplicações móveis web .....	19
1.4.3 Aplicações Web .....	19
1.4.4 Web Services .....	20
<b>1.5 Engenharia de Software</b> .....	<b>21</b>
1.5.1 UML (Unified Modeling Language).....	22
1.5.2 Modelo de processo de software Cascata .....	24
<b>1.6 Ferramentas para desenvolvimento</b> .....	<b>25</b>
1.6.1 Microsoft Visual Studio .....	25
1.6.2 C# e ASP.NET.....	26
1.6.3 MySQL .....	26
1.6.4 Padrão MVC.....	27
1.6.5 Android Studio .....	28
<b>2 DESENVOLVIMENTO</b> .....	<b>29</b>
<b>2.1 Levantamento de Requisitos</b> .....	<b>29</b>
<b>2.2 Banco de Dados</b> .....	<b>30</b>
2.2.1 DER (Diagrama de Entidade-Relacionamento).....	30
2.2.2 Modelo Lógico .....	31
2.2.3 Dicionário de dados .....	33

<b>2.3</b>	<b>Aplicação Web .....</b>	<b>36</b>
2.3.1	Usuário Comum .....	37
2.3.2	Usuário Suporte .....	37
2.3.3	Usuário Administrador .....	38
<b>2.4</b>	<b>Web Service .....</b>	<b>39</b>
<b>2.5</b>	<b>Aplicação Android .....</b>	<b>40</b>
<b>3</b>	<b>RESULTADOS OBTIDOS.....</b>	<b>41</b>
<b>4</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>45</b>
<b>4.1</b>	<b>TRABALHOS FUTUROS .....</b>	<b>46</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>47</b>
	<b>APÊNDICE.....</b>	<b>49</b>

## LISTA DE FIGURAS

<b>Figura 1</b> – Exemplo de funcionamento de um <i>Web Service REST</i> .....	20
<b>Figura 2</b> – Exemplo de Diagrama de Classe .....	22
<b>Figura 3</b> – Exemplo de Diagrama de Caso de Uso .....	24
<b>Figura 4</b> – Exemplo das fases do modelo cascata .....	25
<b>Figura 5</b> – Exemplo de funcionamento da estrutura MVC .....	27
<b>Figura 6</b> – Diagrama de Entidade-Relacionamento do banco de dados .....	31
<b>Figura 7</b> – Modelo lógico do banco de dados .....	32
<b>Figura 8</b> – Diagrama de caso de uso do usuário comum .....	37
<b>Figura 9</b> – Diagrama de caso de uso do usuário suporte .....	38
<b>Figura 10</b> – Diagrama de caso de uso do usuário administrador .....	39
<b>Figura 11</b> – Imagem da página inicial do sistema.....	41
<b>Figura 12</b> – Imagem da página inicial do sistema.....	42
<b>Figura 13</b> – Imagem da página de detalhes de uma ordem de serviço .....	42
<b>Figura 14</b> – Imagem da tela de login do aplicativo Android .....	43
<b>Figura 15</b> – Imagem das notificações do aplicativo Android.....	44
<b>Figura 16</b> – Diagrama de classes do sistema web .....	49
<b>Figura 17</b> – Diagrama de classes do aplicativo Android.....	50
<b>Figura 18</b> – Exemplo de retorno de um método do Web Service .....	51

## LISTA DE TABELAS

<b>Tabela 1</b> – Dicionário de dados da entidade OrdemServico .....	33
<b>Tabela 2</b> - Dicionário de dados da entidade Usuario .....	34
<b>Tabela 3</b> - Dicionário de dados da entidade OSFinalizada .....	34
<b>Tabela 4</b> - Dicionário de dados da entidade Claim .....	35
<b>Tabela 5</b> - Dicionário de dados da entidade Operacao .....	36
<b>Tabela 6</b> - Tabela de métodos do Web Service .....	39

## LISTA DE ABREVIATURAS E SIGLAS

.NET	dotNET
ASP	<i>Active Server Pages</i>
C#	C Sharp
CORBA	Common Object Request Broker Architecture
CSS	Cascading Style Sheets
DCOM	Distributed Component Object Model
DER	Digrama Entidade-Relacionamento
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IIOp	Internet-Orb Protocol
ISO	International Organization Standardization
JSON	JavaScript Object Notation
MVC	Model-View-Controller
PDA	Personal Digital Assistant
REST	Representational State Transfer
RMI	Java Remote Method Invocation
SGBD	Sistema Gerenciador de Banco de Dados
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TI	Tecnologia da Informação

UML	Unified Modeling Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language

## INTRODUÇÃO

Com o avanço da tecnologia, a mesma está cada vez mais presente nos ambientes de trabalho de empresas e instituições. Mesmo com a ascensão dos meios tecnológicos, muitos serviços ainda são realizados de forma obsoleta, utilizando recursos que poderiam ser substituídos e que trariam vários benefícios, tanto na praticidade, como até mesmo ecologicamente. Um grande exemplo disso são como as empresas trabalham com as ordens de serviços. Grande parte delas ainda utilizam documentação em papel e armazenando-as em pastas.

A substituição de um sistema de ordem de serviços manual por um digital, não só facilitaria o trabalho de armazenamento dos dados e ajudaria o meio ambiente, pois elimina o uso de papel e também abre possibilidades de funcionalidades que o sistema antigo não proporcionaria. Como por exemplo, a possibilidade de uma área administrativa do sistema que seja possível atribuição de tarefas e gerar relatórios com gráficos. Além disso, possibilita o acesso fácil ao sistema independentemente do local ou do dispositivo utilizado, necessitando apenas de conexão com a Internet.

A automatização dos processos em uma empresa é fundamental, pois além de ser possível um maior controle sobre as atividades, há outros benefícios como maior fluidez nos processos, maior organização de dados, o modo em que os processos são executados são mais eficazes, e também, pode implicar em uma maior produção para a empresa.

O objetivo geral é desenvolver um sistema de ordem de serviços multiplataforma, onde suas principais funcionalidades serão concentradas em um *Web Service*. Haverá uma aplicação *web* onde os usuários poderão utilizar o sistema e também terá uma parte administrativa que será utilizada pela equipe de Tecnologia da Informação (TI). Por último, será desenvolvido um aplicativo para Android que possibilitará o usuário interagir com o sistema e receber notificações do mesmo. A aplicação *web* juntamente com o *Web Service* serão desenvolvidos em ASP.NET utilizando como linguagem de programação o *C Sharp* (C#). A aplicação Android será desenvolvida em Java.

Como objetivos específicos temos:

- Pesquisa sobre automação e informatização de empresas;
- Levantamento de requisitos;

- Modelagem dos sistemas com diagramas *Unified Modeling Language* (UML);
- Modelagem do banco de dados utilizando o Diagrama de Entidade Relacionamento (DER) e criação do banco de dados;
- Desenvolvimento do sistema *web* e *Web Service*.
- Desenvolvimento da aplicação Android.
- Implantação do sistema.

O sistema será implementado na seção de suporte a informática de instituição de ensino, com intuito de que os funcionários possam abrir um pedido de serviço no sistema sem sair do local onde trabalham, por isso é necessário um sistema *web* para que o usuário acesse de qualquer lugar e de qualquer dispositivo ou sistema operacional. A ideia do aplicativo Android, foi projetada pelo propósito de ocorrer um problema no computador, desta forma, o usuário poderá abrir uma ordem de serviço pelo celular e receber notificações sobre o andamento e situação da ordem de serviço.

Com base nos tópicos levantados, foi necessário um estudo sobre automação de empresas, que é a finalidade para qual o sistema foi desenvolvido. Este trabalho foi desenvolvido em algumas etapas: levantamento de requisitos, definição das tecnologias, desenvolvimento, testes e implantação.

No tocante ao modelo de desenvolvimento utilizado neste trabalho foi o modelo cascata, devido ao fato de ser um método de desenvolvimento que tem uma linearidade e uma sequência, podendo assim, dividir o projeto em fases para sua elaboração.



## **1 REVISÃO BIBLIOGRÁFICA**

De acordo com o que já foi exposto e conforme o objetivo deste trabalho, que é o desenvolvimento de um sistema de ordem de serviço multiplataforma utilizando um *Web Service* para integração das aplicações *web* e Android, foi buscado referências para fundamentar a elaboração do projeto. Neste capítulo, será abordado sobre temas de Automação e informatização de empresas, Aplicações para dispositivos móveis, Aplicações *Web*, *Web Service*, Modelo de processo de software Cascata, e por fim, Ferramentas para Desenvolvimento.

### **1.1 Automação e informatização de empresas**

A inclusão tecnológica dos recursos da informática no cotidiano das empresas é indispensável, pois os benefícios que ela proporciona são muito importantes, principalmente em empresas privadas, em que sua prioridade é gerar lucro.

Segundo SEBRAE (2003, p. 11)

Nos últimos vinte anos, com o avanço das tecnologias na área de informática e com o barateamento dos softwares e dos equipamentos, o processo de informatização da sociedade vem crescendo rapidamente, colaborando de forma substancial no aumento da competitividade das empresas.

Informatização é um termo que é aplicado no conceito de empresas digitais, o que significa a procura em maximizar a utilização da tecnologia da informação em uma empresa, com a finalidade de exercer suas operações de uma forma mais eficaz e produtiva, (VIDAL; ZWICKER; SOUZA, 2005).

De acordo com Marques (2007), a automatização dos processos de uma empresa requer uma análise sobre as atividades realizadas por ela, para se tomar o entendimento de como as atividades são realizadas, assim permitindo um estudo para melhorar e aperfeiçoar a forma com que os processos são executados. Portanto, esse estudo sobre como as atividades da empresa funcionam é essencial para o conhecimento da empresa e qual é sua necessidade.

#### **1.1.1 Objetivos da automação e informatização nas empresas**

Marques (2007) afirma que para se manter uma grande competitividade no mercado, cada vez mais é necessária a ajuda da tecnologia da informação nos

ambientes corporativos, pois o mercado exige uma grande qualidade e fluidez sobre os produtos e serviços de uma empresa, o que sem o uso de *softwares* e máquinas com o intuito de atingir as exigências do mercado.

O processo de informatização de uma empresa tem vários objetivos como, diminuir o encargo de tarefas executadas manualmente, o que além de muitas vezes simplificar as tarefas, reduz custos de mão de obra. Também pode ser eliminado totalmente ou parcialmente o uso de alguns recursos materiais como o papel, o que além de reduzir a verba destinada à compra do mesmo, traz um grande benefício para o meio ambiente. Outra melhoria gerada pelo processo de informatização é em relação a execução de tarefas em um período de tempo menor, o que gera mais produtividade e conseqüentemente mais lucro para a empresa. A segurança de uma atividade realizada por meios tecnológicos é uma grande vantagem, pois os dados são armazenados e executados por sistemas protegidos, o que dá mais credibilidade da informação, além de reduzir consideravelmente a probabilidade de erros, o que em uma tarefa manual é grande, (EAN, 2005, apud MENDES, 2005, p. 11-12).

## **1.2 Software**

Um software não é apenas um programa de computador, tudo relacionado a ele como documentação do planejamento e qualquer outra coisa relacionada a ele interfira no funcionamento do programa, ou seja, software não é apenas um programa de computador, a documentação do software também faz parte dele, na visão de (SOMMERVILLE, 2007).

De acordo com (SOMMERVILLE, 2007), existem basicamente dois tipos de produto de software, são eles Produtos genéricos, onde são desenvolvidos por uma organização e são vendidos a qualquer cliente interessado, e Produtos sob encomenda ou personalizados, que um determinado cliente entra em contato com o desenvolvedor solicitando o desenvolvimento de um software específico para ele, atendendo aos requisitos e funcionalidades que ele descrever.

### 1.3 Dispositivos Móveis

Nas últimas duas décadas houve uma revolução tecnológica na área da telefonia com a criação de infraestruturas que permitem a o acesso a informações a longas distancias e a qualquer momento, como serviços via satélite e redes *wireless*.

Para (MATEUS; LOUREIRO, 1998), o avanço tecnológico que possibilitou a criação dos meios de comunicação sem fio, originou um novo paradigma computacional chamado de Computação móvel, surgida como uma quarta revolução na computação. Este paradigma dá a possibilidade para que os usuários dessa tecnologia tenham acesso a serviços de qualquer lugar, e que permita uma grande mobilidade. Tudo isso, foi possível devido ao advento da comunicação sem fio que permite a comunicação sem a necessidade do aparelho estar conectado a uma rede fixa.

Pode-se chamar de dispositivo móvel um celular, smartphone, notebook, PDA (Personal Digital Assistant), ou qualquer outro aparelho eletrônico que tenha a capacidade de se comunicar com algum tipo de rede, e em alguns casos, com outros dispositivos móveis.

### 1.4 Softwares e aplicativos para dispositivos móveis

Aplicativos para dispositivos móveis estão se tornando cada vez mais presentes no nosso cotidiano e também no ambiente de trabalho. Muitas empresas estão investindo em desenvolvimento de aplicativos para dispositivos móveis na rotina da empresa.

Segundo Fling (2009), as aplicações para dispositivos móveis devem ser desenvolvidas a fim de operar sob as limitações do dispositivo. Os dispositivos móveis como o *smartphone*, por exemplo, muitas vezes não possuem um *hardware* robusto, portanto, qualquer aplicação destinada a estes dispositivos deve ser desenvolvida utilizando muita otimização de código, para a aplicação consumir menos recursos de *hardware* do aparelho.

Ao desenvolver aplicações móveis, deve-se ficar atento ao tipo de aplicação que irá ser utilizada, pois implica diretamente em como será o desenvolvimento e como os recursos serão utilizados.

### 1.4.1 Aplicações móveis nativas

As aplicações nativas são aquelas que foram desenvolvidas para determinada plataforma e que funcionam exclusivamente sob a plataforma em que foram criadas, (CHARLAND; LEROUX, 2011).

Com isso, para cada plataforma desejada, deve ser desenvolvida outra versão da aplicação. Com as aplicações nativas, os projetos de interface são mais fáceis de serem implementados, pois a execução do aplicativo será apenas naquela plataforma. Outra vantagem é o desempenho, uma aplicação nativa leva menor tempo para a interpretação do código, pois é dispensado o uso de um interpretador de outra linguagem para executar a mesma. O acesso aos recursos do aparelho também é uma vantagem, pois a aplicação irá rodar sob um sistema operacional de mesma linguagem, portanto é possível utilizar todos os recursos disponíveis.

### 1.4.2 Aplicações móveis web

De acordo com (FREIRE; RIBEIRO, 2013), também chamadas de *Web View*, são um objeto nativo que pode ser implementado na aplicação e através de um *browser* do sistema operacional do dispositivo, será exibida a aplicação, que é disponibilizada por um serviço de hospedagem em que a mesma está instalada.

Uma das grandes vantagens deste tipo de aplicação é que por ser baseada em tecnologias *web* como *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS), há um número maior de desenvolvedores no mercado. Outra vantagem é que praticamente não há incompatibilidade de plataformas e o processo de desenvolvimento é mais rápido. Porém, nas aplicações do tipo *web*, o acesso aos recursos do dispositivo é mais complicado, o que pode tornar inviável certos tipos de operações que poderiam ser facilmente realizadas por uma aplicação do tipo nativa.

### 1.4.3 Aplicações Web

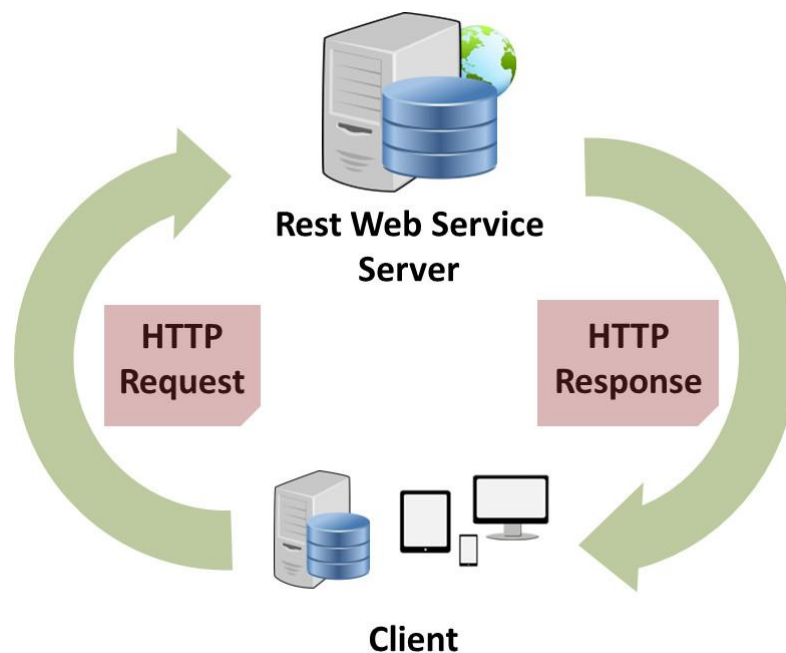
Segundo (OLIVEIRA; VARANDA; HENRIQUES, 2005), uma aplicação *web* é executada em um ambiente distribuído, ou seja, elas são executadas em máquinas distintas, o que é conhecido como cliente-servidor e que se comunicam através de protocolos de rede como o *Hypertext Transfer Protocol* (HTTP) e sua interface é apresentada ao usuário através de um navegador (*browser*).

#### 1.4.4 Web Services

Com o decorrer dos anos, a tecnologia foi se desenvolvendo cada vez mais e com isso o a quantidade de aplicações cresceu proporcionalmente. Além disso, ao surgir várias novas linguagens de programação, encontrou-se a necessidade de integrar sistemas. A fim de sanar estes problemas de integração entre sistemas distintos e desenvolvidos em linguagens diferentes, a tecnologia dos *Web Services* foi criada. O seu propósito é ser uma aplicação intermediária entre uma ou mais aplicações distintas, independentemente de sua plataforma ou linguagem em que foi desenvolvida, a fim de realizar operações e consumir informações, (MENÉNDEZ, 2002).

A Figura 1 apresentada abaixo exemplifica o funcionamento de um Web Service utilizando a arquitetura REST.

**Figura 1** – Exemplo de funcionamento de um *Web Service REST*



**Fonte:** ChemAxom, 2015.

A tecnologia dos *Web Services* tem uma característica importante que possibilita o desenvolvedor definir diferentes meios de transmissão de dados através da rede, podendo trabalhar com protocolos de rede como HTTP, *Simple Mail Transfer Protocol* (SMTP), *File Transfer Protocol* (FTP), *Java Remote Method Invocation – Internet-Orb Protocol* (RMI-IIOP) ou protocolos proprietários, (W3C, 2004, apud MORO; DORNELES; REBONATTO, 2011, p. 38).

O *Extensible Markup Language*(XML) foi a linguagem escolhida pelo *World Wide Web Consortium*(W3C) para troca de informações padrão entre as aplicações que se comunicam através do protocolo HTTP, mas também pode-se optar pela utilização do *JavaScript Object Notation* (JSON). Já as aplicações que trocam informações através do *Simple Object Access Protocol* (SOAP), que é uma tecnologia não proprietária, oferece alternativas aos protocolos tradicionais, como *Common Object Request Broker Architecture* (CORBA) e *Distributed Component Object Model* (DCOM), (GOMES, 2010, apud LIMA, 2012, p. 8).

Para que haja interação da aplicação Android com o sistema *web*, será utilizado um Web Service utilizando a arquitetura REST (*Representational State Transfer*), criado por Roy Fielding, um dos criadores do protocolo HTTP (FIELDING, 2000).

## 1.5 Engenharia de Software

A computação está presente em diversos países e seu crescimento está crescendo gradativamente com o passar dos anos e com isso, cada vez mais as áreas de conhecimento inserem a computação em sua rotina devido aos variados benefícios que a mesma traz. Tudo é controlado e monitorado por servidores através de softwares, por isso os softwares devem ser bem planejados para garantir alta disponibilidade de serviços e diminuir as chances de falhas, diz (SOMMERVILLE, 2007).

A engenharia de software é a área da computação criada afim de auxiliar o processo de desenvolvimento de software. Ou seja, a engenharia de software designa recursos e padrões a serem seguidos para o planejamento do desenvolvimento do software, visando não ultrapassar a margem de custo do projeto e obter um software de qualidade como produto final, de acordo com (SOMMERVILLE, 2007).

No tocante a Sommerville (2007, p. 3), “O conceito *de engenharia de software* foi inicialmente proposto em 1969, em uma conferência organizada para discutir o que foi então chamado de ‘crise de software’.” A partir dessa crise de software, um novo *hardware* computacional foi introduzido e isso possibilitou o desenvolvimento de softwares mais robustos.

### 1.5.1 UML (Unified Modeling Language)

A engenharia de software é a área que auxilia o desenvolvimento de um software utilizando padrões e recomendação para isto. A UML é uma dessas padronizações que a engenharia de software traz, usada para documentar projetos de software.

Segundo (PRESSMAN, 2011), a UML serve para ser usada da mesma forma que os arquitetos criam plantas e projetos a serem usados por uma empresa de construção, ou seja, os arquitetos de software criam os diagramas para auxílio ao desenvolvimento do software.

Criada na década de 1990 por Grady Booch, Jim Rumbaugh e Ivar Jacobson, a UML combinou uma variedade de notações de modelagens existentes na época. O padrão atual é a UML 2.0 que é um padrão ISO (International Organization Standardization), e oferece 13 modelos diferentes de diagramas para modelagem de software, de acordo com (PRESSMAN, 2011).

Os principais diagramas da UML utilizados serão apresentados a seguir, explicando os conceitos básicos de cada um deles.

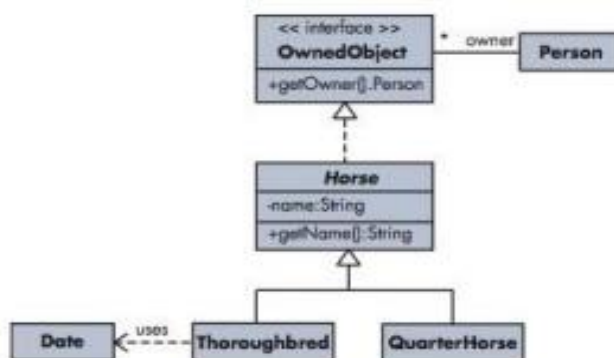
#### **Diagrama de Classe**

Utilizado para modelagem de classes, isto incluindo seus detalhes como atributos, métodos, relações e associações com outras classes. Através do diagrama de classe, é possível ter uma visão estrutural de um sistema, porém não mostra a forma em que o sistema se comunicará com os outros elementos, diz (PRESSMAN, 2011).

Os elementos principais do diagrama de classe são caixas utilizadas para representar classes, e são divididas em duas partes onde a parte superior é destinada a representação dos atributos da classe, e a parte inferior aos seus correspondentes métodos, afirma (PRESSMAN, 2011).

Abaixo na Figura 2, segue um diagrama de classe apresentado por Pressman que exhibe a representação gráfica das classes de um sistema, seus respectivos atributos e métodos e seus relacionamentos com outras classes do sistema.

**Figura 2** – Exemplo de Diagrama de Classe



Fonte: Pressman, 2011, p. 728.

Ao lado de cada atributo e métodos, é possível identificar que há um símbolo, que representa um nível de visibilidade do atributo ou método no sistema, podendo ser representado como -, #, ~ ou +, indicando respectivamente, visibilidade *private*, *protected*, *package* ou *public*, afirma (PRESSMAN, 2011).

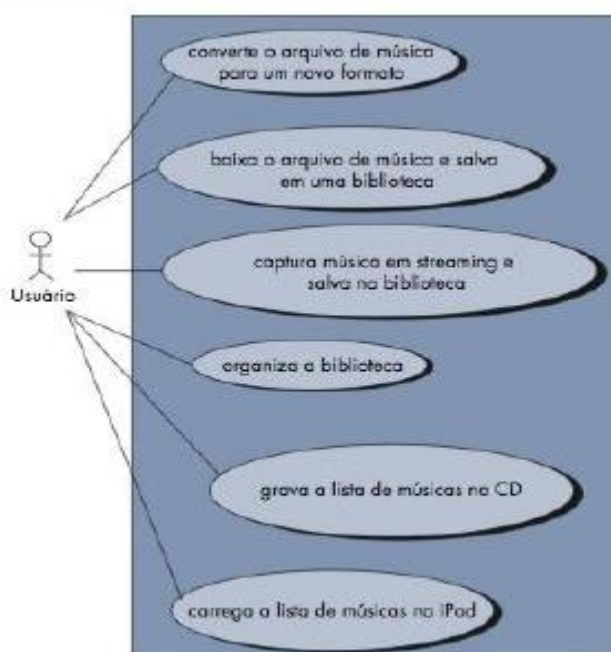
### Diagrama de Caso de Uso

O diagrama de caso de uso ajuda no entendimento das funcionalidades e características do software. Ele é o diagrama que representa como o usuário irá interagir com o sistema, especificando as atividades que ele irá fazer e em alguns casos especifica algumas sequências de interações, diz (PRESSMAN, 2011).

Abaixo na Figura 3, segue um diagrama de caso de uso referente a um aplicativo de música digital, apresentado por Pressman em seu livro.



**Figura 3** – Exemplo de Diagrama de Caso de Uso



**Fonte:** Pressman, 2011, p. 732.

Visualizando o diagrama, nota-se a figura de um usuário que é chamado de ator, podendo haver mais de um ator em um diagrama de acordo com a complexidade do sistema. O ator é conectado por meio de linhas às ações que ele executa no sistema, de acordo com (PRESSMAN, 2011).

### 1.5.2 Modelo de processo de software Cascata

Processos de *software* são sequências de práticas utilizadas para o desenvolvimento de um *software*. Essas práticas são técnicas específicas para cada fase do desenvolvimento, que são aplicadas do início do projeto até a implantação e manutenção. “Processo de *software* é definido como uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um *software* de alta qualidade” (PRESSMAN, 2011, p. 52). “Um processo de *software* define a abordagem adotada conforme um *software* é elaborado pela engenharia” (PRESSMAN, 2011, p. 52).

Sommerville (2007, p. 43) afirma que

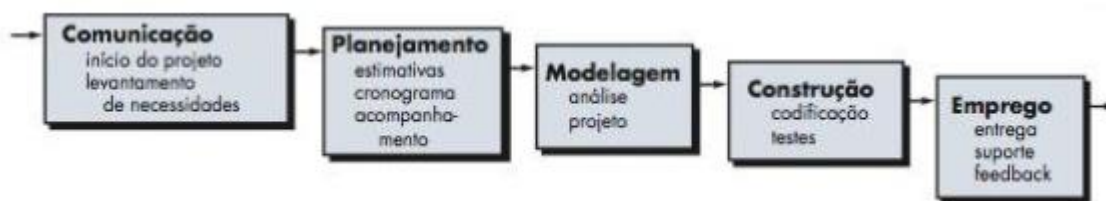
“Um modelo de processo de *software* é uma representação abstrata de um processo de *software*. Cada modelo de processo representa

um processo sob determinada perspectiva e, dessa forma, fornece somente informações parciais sobre esse processo.”

O modelo de processo de software utilizado nesse trabalho é o cascata, também chamado de ciclo de vida clássico, que engloba as principais atividades do processo, como especificação de requisitos, desenvolvimento e implantação e evolução. (SOMMERVILLE, 2007). Abaixo, segue um exemplo da sequência das atividades do modelo cascata, “levantamento de necessidades por parte do cliente, avançando pelas fases de planejamento, modelagem, construção, emprego e culminado no suporte contínuo do software concluído” (PRESSMAN, 2011, p. 52).

Na Figura 4, é possível identificar as fases e suas atividades existentes no modelo de processo de software Cascata.

**Figura 4 – Exemplo das fases do modelo cascata**



**Fonte:** Pressman, 2011, p. 60.

## 1.6 Ferramentas para desenvolvimento

De acordo com o tema abordado, a seguir serão apresentadas ferramentas, linguagens e padrão de desenvolvimento utilizados para a elaboração do sistema. Foram considerados requisitos para a definição das ferramentas como portabilidade, evolução e manutenção das aplicações, segurança e praticidade de desenvolvimento.

### 1.6.1 Microsoft Visual Studio

No ambiente de desenvolvimento de sistemas são muito utilizadas ferramentas para o auxílio do mesmo para tornar o processo mais rápido e fácil de ser realizado. É muito comum o uso de ferramentas chamadas de IDE (*Integrated Development Environment*), que no português significa Ambiente de

Desenvolvimento Integrado, nesse caso, a ferramenta principal para o desenvolvimento do sistema, foi a IDE da Microsoft chamada Visual Studio.

A Microsoft (2015) diz que “Visual Studio é um conjunto completo de ferramentas para construir aplicativos desktop e aplicativos Web empresariais desenvolvidos por equipes”. Ou seja, o Visual Studio oferece as ferramentas principais para o desenvolvimento do projeto que foi feito em ASP.NET e C# em sua maior parte.

### **1.6.2 C# e ASP.NET**

Todo tipo de software seja ele para plataformas desktop, móbil ou web são desenvolvidos utilizando no mínimo um tipo de linguagem de programação. Criada pela Microsoft, a linguagem C# foi a escolhida para ser utilizada em grande parte do sistema.

Além da linguagem de programação, foi utilizada uma plataforma web também desenvolvida pela Microsoft, chamada ASP.NET (*Active Server Pages*). ASP.NET é uma plataforma Web que fornece todos os serviços necessários para construir aplicações web baseadas em servidor. ASP.NET é baseado em *.NET Framework*, então pode-se utilizar todos os recursos do *.NET Framework* no ASP.NET. (MICROSOFT, 2015).

O ASP.NET possui uma série de recursos que podem ser utilizados para auxílio no desenvolvimento de sistemas, neste trabalho foi utilizado um deles, chamado de ASP.NET Identity. De acordo com (RASTOGI et al., 2013), o ASP.NET Identity foi baseado no ASP.NET Membership, utilizado para implementar soluções relacionadas a contas de usuário e autenticações de sistemas, e o ASP.NET Identity possui as mesmas características, porém foi aperfeiçoado e incrementado com novas funcionalidades. Uma delas foi utilizada no sistema, o recurso de trabalhar com Claims, recurso similar ao de atribuir grupos ou papéis ao usuário, entretanto há mais flexibilidade, diz (RASTOGI et al., 2013).

### **1.6.3 MySQL**

O sistema desenvolvido trabalha constantemente com informações que necessitam ser armazenadas em algum local. Para isso, é necessário o uso de um banco de dados e um servidor gerenciador de banco de dados (SGBD), que aqui foi-se utilizado o *MySQL* para ser encarregado dessas tarefas.

De acordo com Neves e Ruas (2005, p. 22)

O MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional, de licença dupla (sendo uma delas de software livre), projetado inicialmente para trabalhar com aplicações de pequeno e médio portes, mas hoje atendendo a aplicações de grande porte e com mais vantagens do que seus concorrentes.

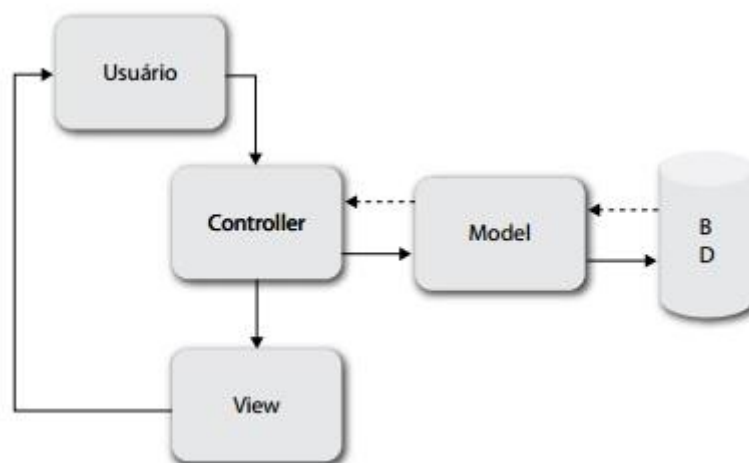
Isso significa que, além de um banco de dados o MySQL possui também um SGBD para o gerenciamento do mesmo, porém é possível utilizar outro software como SGBD, que no caso, foi utilizado o HeidiSQL.

#### 1.6.4 Padrão MVC

O padrão MVC (Model-View-Controller) é um padrão de arquitetura de software que é separado em três camadas. Essas camadas são chamadas de: *model*, *view* e *controller*. A camada *model* contém o código da camada de dados, a *view* implementa o design da aplicação e exibe dados, e por fim, o *controller* recebe as requisições do usuário (LOTAR, 2011, p. 15).

Segue abaixo na Figura 5 um exemplo do funcionamento da estrutura MVC:

**Figura 5** – Exemplo de funcionamento da estrutura MVC



**Fonte:** Lotar, 2011, p. 15.

Lotar (2011, p. 15) afirma que “O *controller* aceita a requisição do usuário, acessa o *view* e/ou o *model*. O *model* acessa o banco de dados. Ao final, o conteúdo do *view* é retornado ao usuário”.

### **1.6.5 Android Studio**

Assim como para o desenvolvimento do sistema *web* e *WebService*, foi necessária também a utilização de uma IDE para o desenvolvimento do aplicativo Android. A ferramenta escolhida foi o *Android Studio*, que é uma IDE designada para desenvolvimento Android e foi criada pelo Google, sendo baseada em IntelliJ IDEA.

## 2 DESENVOLVIMENTO

Neste capítulo, o desenvolvimento do sistema foi apresentado apontando todas suas etapas e seus principais detalhes.

Com base no sistema atual de ordem de serviço utilizado na seção de informática de uma instituição, o qual será substituído pelo sistema em desenvolvimento visando uma melhor praticidade e segurança que um sistema informatizado traz, já exemplificadas no capítulo anterior, foram definidos requisitos para o sistema junto a equipe da seção de informática da instituição, que serão apresentados no próximo tópico.

### 2.1 Levantamento de Requisitos

A partir da reunião citada na introdução deste capítulo, foi concluído que será necessário um sistema *web* que tenha todos os recursos utilizados no sistema atual. Além disso, foi sugerido o desenvolvimento de um aplicativo mobile para acompanhamento das solicitações.

Em relação ao sistema *web*, o mesmo deverá permitir que os usuários realizem um cadastro e façam autenticação no sistema para acessar o painel do usuário. O usuário terá como operações principais, abrir uma ordem de serviço, visualizar andamento da ordem de serviço, visualizar ordens de serviço, solicitar cancelamento de ordem de serviço. Essas mesmas funcionalidades serão disponibilizadas no aplicativo Android. Na área administrativa, deverão conter funcionalidades para visualizar as ordens de serviço em aberto, visualizar as ordens de serviço em andamento, finalizar uma ordem de serviço e visualizar as ordens de serviço finalizadas. A ordem de serviço deverá conter dados do solicitante como nome, departamento, data, localização, contato, descrição do problema, nome da pessoa para quem foi atribuída, data de conclusão e descrição da resolução do problema.

A respeito do aplicativo mobile, foi acordado que o mesmo seria desenvolvido para Android, visto que atende a maioria dos usuários. O aplicativo necessitará de um método de autenticação para acesso a tela de notificações para se manter atualizado do estado da ordem de serviço. Porém, para ter acesso aos dados do sistema *web*, a aplicação necessita de um intermediador, que no caso é um *Web Service*, já mencionado no capítulo anterior.

O *Web Service* tem a responsabilidade de receber requisições do aplicativo Android, consultar dados no banco e retornar dados ao aplicativo, utilizando estilo arquitetural REST.

## **2.2 Banco de Dados**

No ambiente computacional, diversas aplicações trabalham ou necessitam de armazenamento de informações para serem resgatadas em algum momento, por isso, muitas aplicações utilizam os bancos de dados para este fim, e que no presente trabalho também será necessário.

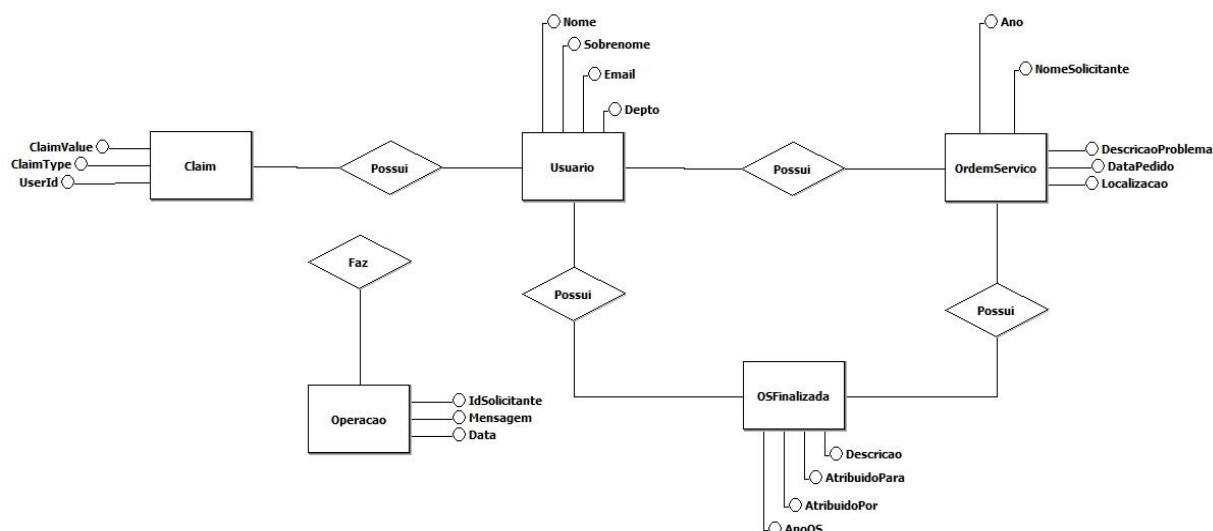
Como já mencionado, o SGBD utilizado foi o MySQL, e após a análise dos requisitos foram criados o DER e o Modelo Conceitual do banco de dados que foi desenvolvido.

Nos tópicos seguintes são apresentados os detalhes do banco de dados e suas respectivas representações gráficas.

### **2.2.1 DER (Diagrama de Entidade-Relacionamento)**

Para a representação conceitual do banco de dados criado, foi gerado o Diagrama de Entidade-Relacionamento com a finalidade de representar de maneira simples e objetiva as principais entidades, principais atributos e seus relacionamentos para que os *stakeholders* (partes interessadas) compreendam a estrutura básica do banco de dados, mesmo que não tenham conhecimento técnico sobre o assunto.

**Figura 6** – Diagrama de Entidade-Relacionamento do banco de dados



**Fonte:** Próprio autor.

Analisando o diagrama presente na Figura 7, pode-se concluir que foi identificada a necessidade da elaboração de 5 entidades ou tabelas, chamadas Claim, Usuario, OrdemServico, OSFinalizada e Operacao, e seus relacionamentos são descritos nos losangos interligados nas mesmas. Também é possível notar que em cada entidade contém alguns itens, esses chamados de atributos, que são os valores que serão propriamente armazenados por cada entidade no banco de dados.

### 2.2.2 Modelo Lógico

O modelo lógico de um banco de dados é um diagrama que contém as mesmas informações do anterior, porém neste são exibidos mais detalhes para a partir dele gerar propriamente o banco de dados.

No diagrama abaixo, nota-se que há outros atributos que não foram exibidos no diagrama anterior, isto porque geralmente são dados secundários e não prioritários para o entendimento dos *stakeholders*.

É possível notar também que conceitos e termos da linguagem SQL utilizada para criar o banco de dados físico, são representados neste diagrama. A chave amarela exibida ao lado dos atributos denominados como Id identifica aquele atributo como o identificador único do registro que será inserido na tabela. Já a

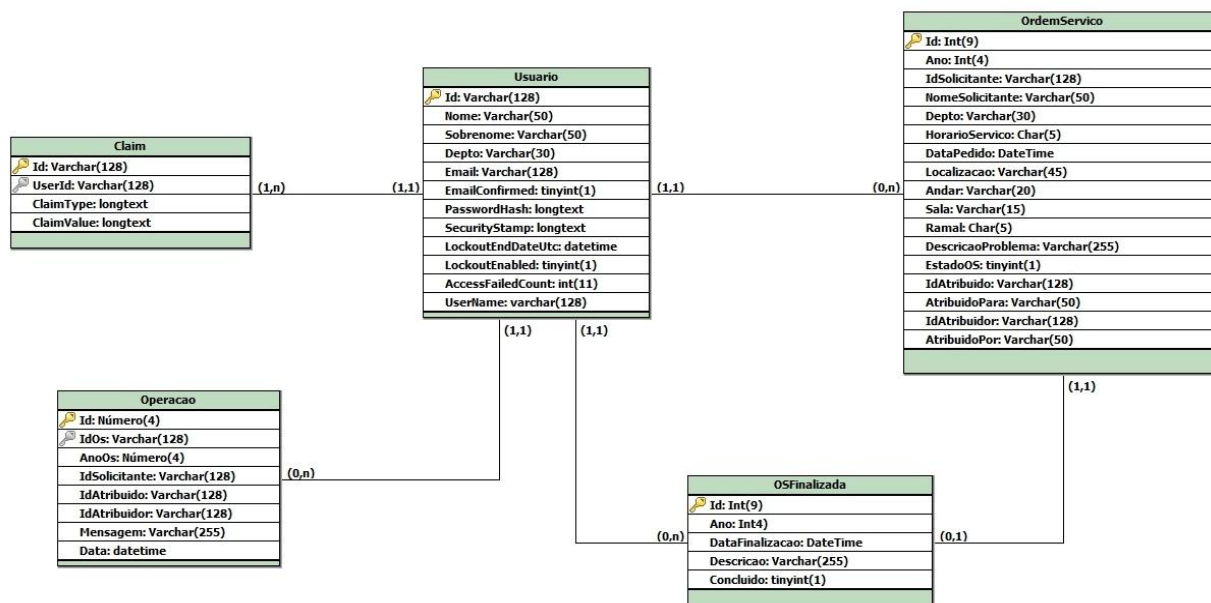


chave cinza é uma chave estrangeira, que indica que o valor daquele campo será originado da chave primária da outra tabela.

Analisando ambos os diagramas apresentados levando em consideração os relacionamentos e cardinalidades, pode-se concluir que o armazenamento de dados do sistema segue a seguinte forma: um usuário possui nenhuma ou vários claims e cada claim pertence apenas a um usuário. Um usuário tem nenhuma ou várias operações e cada operação pertence a um usuário. Um usuário possui nenhuma ou várias ordens de serviço e cada ordem de serviço pertence a um usuário. Um usuário possui nenhuma ou várias ordens de serviço finalizadas e cada ordem de serviço finalizada pertence a um usuário. E por fim, uma ordem de serviço pode não ter sido finalizada ou ter sido finalizada e cada ordem de serviço finalizada pertence a uma ordem de serviço apenas.

Na figura 8 apresentada a seguir foi demonstrado o modelo lógico do banco de dados tendo como base o diagrama da Figura 7 apresentado anteriormente, e utilizado para criar o modelo físico do banco de dados.

**Figura 7 – Modelo lógico do banco de dados**



**Fonte:** Próprio autor.

### 2.2.3 Dicionário de dados

As tabelas a seguir são denominadas de dicionário de dados e mostram detalhadamente cada atributo de uma tabela e suas respectivas informações.

**Tabela 1 – Dicionário de dados da entidade OrdemServico**

<b>Entidade: OrdemServico</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Determinante	Numérico	4	
Ano	Simple	Numérico	4	
NomeSolicitante	Simple	Texto	50	
Depto	Simple	Texto	30	
HorarioServico	Simple	Data		
DataPedido	Simple	Data		
Localizacao	Simple	Texto	45	
Andar	Simple	Texto	20	
Sala	Simple	Texto	15	
Ramal	Simple	Texto	5	
DescricaoProblema	Simple	Texto	255	
IdAtribuido	Simple	Texto	128	Id para quem foi atribuída
AtribuidoPara	Simple	Texto	50	Nome para quem foi atribuída
IdAtribuidor	Simple	Texto	128	Id de quem atribuiu
AtribuidoPor	Simple	Texto	50	Nome de quem atribuiu

**Fonte:** Próprio autor.

Observando as informações da Tabela 1 referente a entidade OrdemServico é possível notar mais claramente as características de cada atributo. Nesta entidade, destacam-se os atributos IdAtribuido, AtribuidoPara, IdAtribuidor e AtribuidoPor, que como descritos em suas respectivas descrições são atributos para armazenar as informações de para quem a ordem de serviço foi atribuída e por quem ela foi atribuída.

**Tabela 2** - Dicionário de dados da entidade Usuario

<b>Entidade: Usuario</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Determinante	Texto	128	
Nome	Simple	Texto	50	
Sobrenome	Simple	Texto	50	
Email	Simple	Texto	128	
EmailConfirmed	Simple	Numérico	1	Email confirmado
PasswordHash	Simple	Texto		Senha criptografada
SecurityStamp	Simple	Texto		Código de verificação
LockoutEndDateUtc	Simple	Data		Data desbloqueio da conta
LockoutEnabled	Simple	Numérico	1	Bloquear conta
AccessFailedCount	Simple	Numérico	11	Contador tentativas de login
UserName	Simple	Texto	128	Username utilizado para login

**Fonte** – Próprio autor.

Na Tabela 2, referente a entidade Usuario, os atributos que merecem destaque. O campo EmailConfirmed que indica se o email do usuário foi ou não confirmado. O campo PasswordHash é destinado ao armazenamento da senha criptografada do usuário. SecurityStamp é um campo que armazena com código de segurança que representa os dados do usuário, se a senha do usuário for alterada, seu SecurityStamp também será alterado. O atributo LockoutEndDateUtc é o que armazena a data de desbloqueio da conta caso o usuário exceda 5 tentativas de login sem sucesso. O atributo LockoutEnabled indica se a conta do usuário está ou não bloqueada. O campo AccessFailedCount é um contador de tentativas de login sem sucesso. E por fim, o campo UserName é o que armazena o username utilizado para login, que no caso é o email do usuário. Todos estes atributos são criados por padrão pelo *ASP.NET Identity* para auxílio no gerenciamento da conta de usuário.

**Tabela 3** – Dicionário de dados da entidade OSFinalizada

<b>Entidade: OSFinalizada</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Determinante	Numérico	9	Id da ordem de serviço

Ano	Simple	Numérico	4	Ano da ordem de serviço
DataFinalizacao	Simple	Data		
Descricao	Simple	Texto	255	Descrição da resolução
Concluido	Simple	Numérico	1	Estado se foi concluído ou não

**Fonte** – Próprio autor.

Observando a Tabela 3 referente a entidade OSFinalizada, é possível notar que os dois primeiros atributos da tabela são referentes a ordem de serviço e seus valores serão iguais aos da ordem de serviço correspondente. Logo abaixo da DataFinalização, há o campo Descricao onde um administrador ou membro da equipe de suporte descreve como a ordem de serviço foi solucionada ou o porque ela não foi. Por último, há o campo Concluido que indica se a ordem de serviço foi concluída ou não.

**Tabela 4** - Dicionário de dados da entidade Claim

<b>Entidade: Claim</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Determinante	Texto	128	
UserId	Simple	Texto	128	Id do usuário
ClaimType	Simple	Texto		Tipo do claim
ClaimValue	Simple	Texto		Valor do claim

**Fonte** – Próprio autor.

A entidade Claim exibida na Tabela 4 é responsável por armazenar dados do usuário relacionados a permissões de acesso. Também é utilizada para armazenar o código gerado para confirmação de email da conta do usuário. O destaque desta entidade são os atributos ClaimType e ClaimValue, que representam o tipo do claim, como por exemplo TipoUsuario e o valor do claim, como por exemplo Adm, respectivamente.

Tabela 5 - Dicionário de dados da entidade Operacao

Entidade: Operacao				
Atributo	Classe	Domínio	Tamanho	Descrição
Id	Determinante	Numérico	9	Id da operação
IdOS	Simple	Numérico	4	Id da ordem de serviço
AnoOS	Simple	Numérico	4	Ano da ordem de serviço
IdSolicitante	Simple	Texto	128	IdSolicitante da ordem de serviço
IdAtribuido	Simple	Texto	128	IdAtribuido da ordem de serviço
IdAtribuidor	Simple	Texto	128	IdAtribuidor da ordem de serviço
Mensagem	Simple	Texto	255	Mensagem da operação
Data	Simple	Data		Data da operação

Fonte – Próprio autor.

O dicionário de dados da Tabela 5 que representa a entidade Operacao mostra que os seis primeiros atributos são referentes a tabela OrdemServico e seus valores são obtidos através dela. O atributo Mensagem é a descrição da operação realizada, e posteriormente utilizada para compor as notificações na aplicação Android.

### 2.3 Aplicação Web

Com a facilidade de acesso à internet e seu crescimento, a implementação de sistemas *web* como solução de problemas se torna cada vez mais frequente, devido a sua maior disponibilidade e fácil acesso, necessitando apenas de conexão com a internet.

O sistema *web* proposto foi desenvolvido na plataforma ASP.NET utilizando a linguagem de programação C# e seguindo os padrões de desenvolvimento da arquitetura MVC. Para manipulação de dados com o MySQL, foi utilizado o *Entity Framework*, ferramenta para persistência de dados da Microsoft. Para customização do design, foi utilizado o *Bootstrap* criado pelo Twitter, *framework* para desenvolvimento *web* voltado para criação de páginas responsivas.

Levando em consideração que o sistema seria acessado 3 tipos diferentes de usuários como usuário comum, usuário suporte e usuário administrador, o mesmo foi dividido em módulos e cada módulo, cada módulo é representado por um controller e possui funcionalidades distintas referentes a cada tipo de usuário. O tipo de um

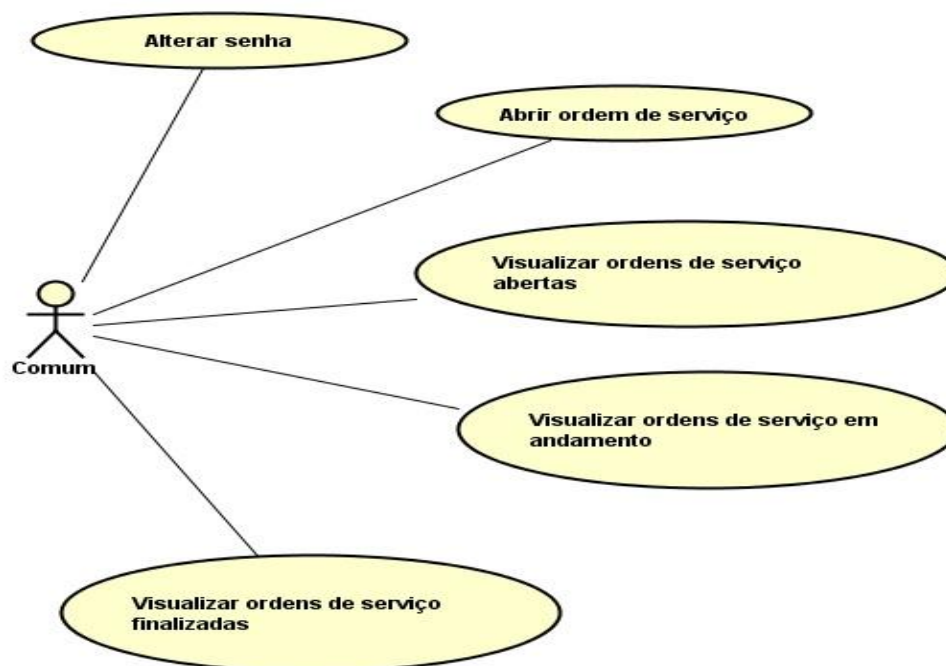
usuário é definido através de seus respectivos claims armazenados no banco de dados.

### 2.3.1 Usuário Comum

Este módulo é acessado por usuários que são clientes internos da instituição, ou seja, destinado aos usuários que necessitam de assistência técnica e por este módulo entram em contato com a equipe de suporte de informática.

O usuário comum por este módulo tem o acesso as funcionalidades de alteração de senha, abrir ordem de serviço, visualizar suas ordens de serviço abertas, visualizar suas ordens de serviço em andamento e visualizar suas ordens de serviço finalizadas, representadas pelo diagrama da Figura 8.

**Figura 8** – Diagrama de caso de uso do usuário comum



**Fonte:** Próprio autor.

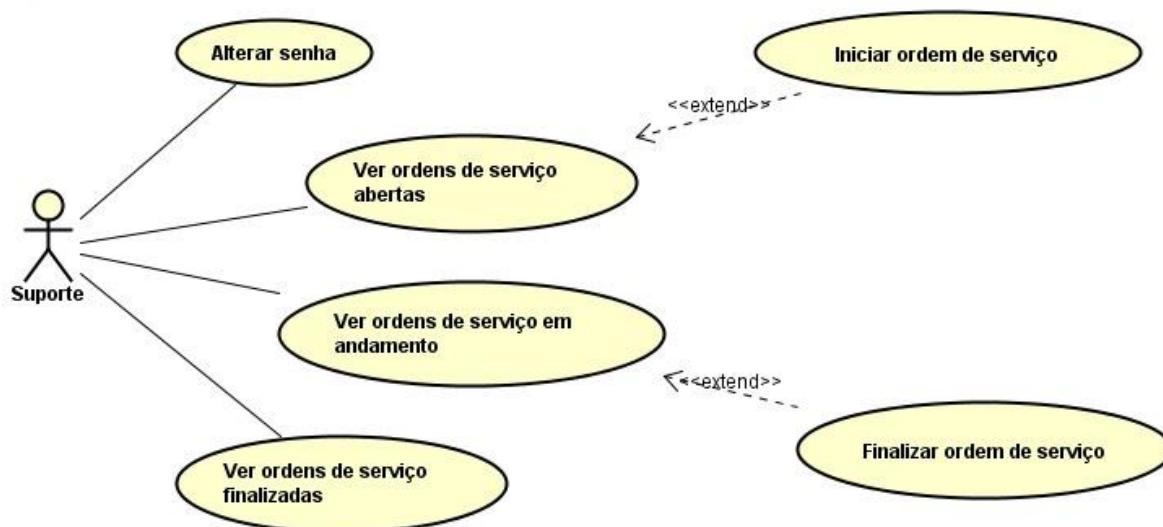
### 2.3.2 Usuário Suporte

Este módulo é acessado por usuários que fazem parte da equipe de suporte de informática da instituição.

É por este módulo que o usuário de suporte faz as seguintes interações, alterar senha, visualizar ordens de serviço abertas, visualizar ordens de serviço em andamento e visualizar ordens de serviço finalizadas. O usuário suporte pode

realizar outras duas operações, iniciar ordem de serviço e finalizar ordem de serviço, onde na primeira só é possível executá-la caso a ordem de serviço esteja aberta e ainda não em andamento, já a segunda operação só é possível caso a ordem de serviço esteja em andamento e atribuída a ele mesmo, ou seja, ele não tem permissão para finalizar uma ordem de serviço que não foi atribuída a ele. Essas funcionalidades foram representadas no diagrama da Figura 9.

**Figura 9** – Diagrama de caso de uso do usuário suporte



Fonte: Próprio autor.

### 2.3.3 Usuário Administrador

Este módulo é acessado por usuários que fazem parte da equipe de suporte da informática da instituição, porém apenas os analistas.

Com as mesmas funcionalidades apresentadas no módulo anterior, o módulo de usuário administrador permite que o usuário finalize uma ordem de serviço que não foi atribuída a ele, e permite também atribuir ordens de serviço a outros usuários. Este módulo também proporciona a funcionalidade de buscar qualquer tipo de usuário no cadastrado no sistema e editar seus respectivos claims, isto para a alteração do tipo do usuário. É possível conferi-las abaixo no diagrama da Figura 10.

**Figura 10** – Diagrama de caso de uso do usuário administrador



Fonte: Próprio autor.

## 2.4 Web Service

Como o Web Service utilizado segue a arquitetura REST, o mesmo utiliza o cabeçalho HTTP como um envelope de metadados, ou seja, os métodos e parâmetros serão enviados pela aplicação Android através de URL (Uniform Resource Location) pelo protocolo HTTP. Os métodos são do tipo GET e todos eles recebem parâmetros de entrada. As informações processadas pelo *WebService* são retornadas para a aplicação com o formato em JSON. Segue a Tabela 6 em que apresenta os métodos disponíveis.

**Tabela 6** - Tabela de métodos do Web Service

Método	Descrição
Autenticar	Recebe os dados de login e verifica se pode ser autenticado no sistema
DadosUsuario	Coleta os dados do usuário autenticado
Operacao	Verifica se há novas operações ou notificações

Fonte – Próprio autor.



## 2.5 Aplicação Android

O uso do smartphone no cotidiano das pessoas vem se tornando cada vez mais como e em todos os ambientes, como no ambiente de trabalho. Isso devida a facilidade de comunicação e acesso de informação em qualquer local que a pessoa esteja.

O desenvolvimento da aplicação Android foi sugerido a fim de o usuário possa ter fácil acesso às informações de uma ordem de serviço através de notificações de qualquer lugar que ele esteja, necessitando apenas de um meio conexão de internet com o smartphone, o que é totalmente acessível no ambiente da faculdade, já que há conexão *Wi-Fi* em todo o campus, além disso, é uma solução caso o usuário tenha problemas com seu computador.

Através da aplicação Android, o usuário se depara com as seguintes funcionalidades:

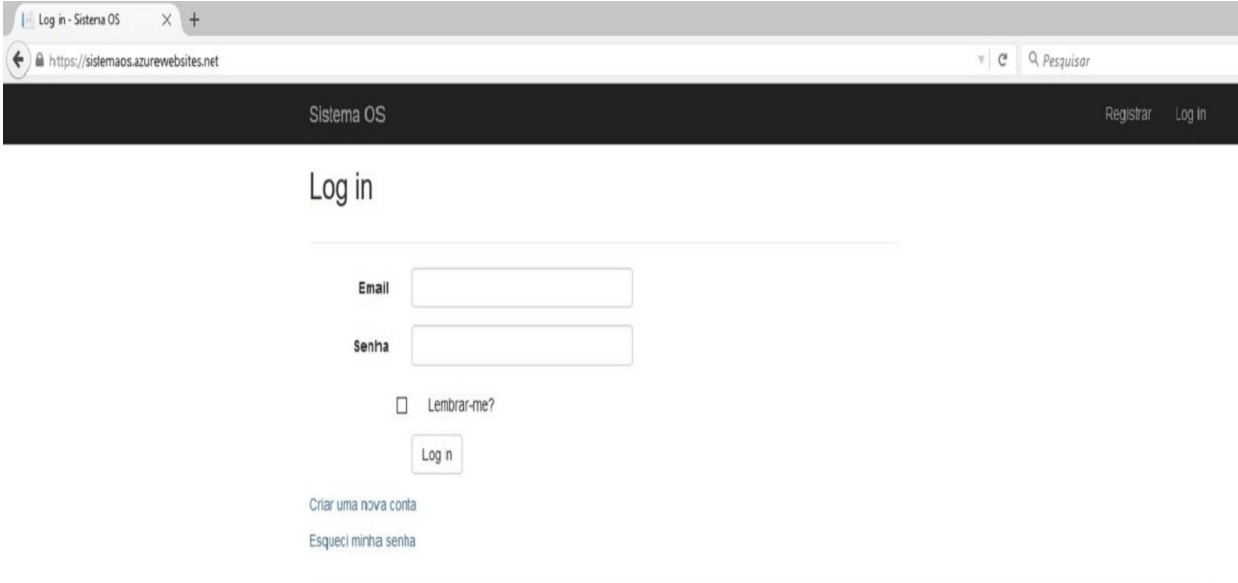
- Login: o usuário insere seus dados cadastrados no sistema *web* para realizar a autenticação na aplicação.
- Ver notificações: o usuário pode visualizar suas novas notificações.

### 3 RESULTADOS OBTIDOS

Como resultados desse trabalho foram construídos uma aplicação web, uma aplicação Android e um Web Service para a interação entre as duas aplicações. Exemplos da utilização dos sistemas foram apresentados a seguir com o auxílio de imagens.

Como principal parte de trabalho, o sistema web foi o representado em primeiro lugar a seguir exibindo suas principais telas.

**Figura 11** – Imagem da página inicial do sistema



A imagem mostra a interface de login de um sistema web. No topo, há uma barra de navegação com o título "Sistema OS" e links para "Registrar" e "Log in". Abaixo, o título "Log in" é exibido em uma fonte maior. O formulário de login contém dois campos de entrada: "Email" e "Senha". Abaixo dos campos, há uma caixa de seleção para "Lembrar-me?". Um botão "Log in" está posicionado abaixo da caixa de seleção. Na base do formulário, há dois links: "Criar uma nova conta" e "Esqueci minha senha".

**Fonte** – Próprio autor.

Na Figura 11, a página inicial do sistema é a página de login, que pode ser acessada pela URL <https://sistemaos.azurewebsites.net>.

**Figura 12** – Imagem da página inicial do sistema

Sistema OS Olá Lucas! Sair

## Abrir OS

Horário Serviço

Andar

Sala

Ramal

Descrição do Problema

[Voltar](#)

**Fonte** – Próprio autor.

Na Figura 12, foi demonstrada a página designada ao usuário para a abertura de uma ordem de serviço, e seus respectivos dados devem ser preenchidos nesta página de acordo com a necessidade do usuário, o único campo obrigatório é a descrição do problema, onde o usuário insere as informações do problema.

**Figura 13** – Imagem da página de detalhes de uma ordem de serviço

Sistema OS Olá Admin! Sair

## Detalhes

**Número** 13  
**Ano** 2015  
**Nome Solicitante** Lucas Prog  
**Departamento** Departamento 1  
**Horário Serviço** 08:30  
**Data Pedido** 11/8/2015  
**Andar** 11  
**Sala** Sala 10  
**Ramal** 12234  
**Descrição do Problema** Problemas com o switch do departamento.

---

**Atribuído Por**  
**Atribuído Para** Rodrigo  
**Data Finalização** 11/8/2015 9:43:15 AM  
**Descrição da Finalização** Efetuada a troca do switch.

[Voltar](#)

**Fonte** – Próprio autor.

A imagem exibida anteriormente na Figura 13 mostra a página que exibe os detalhes de uma ordem de serviço. A ordem de serviço que está sendo mostrada nesse exemplo foi finalizada, por esse motivo os últimos cinco campos do relatório são exibidos com as informações da finalização da ordem de serviço.

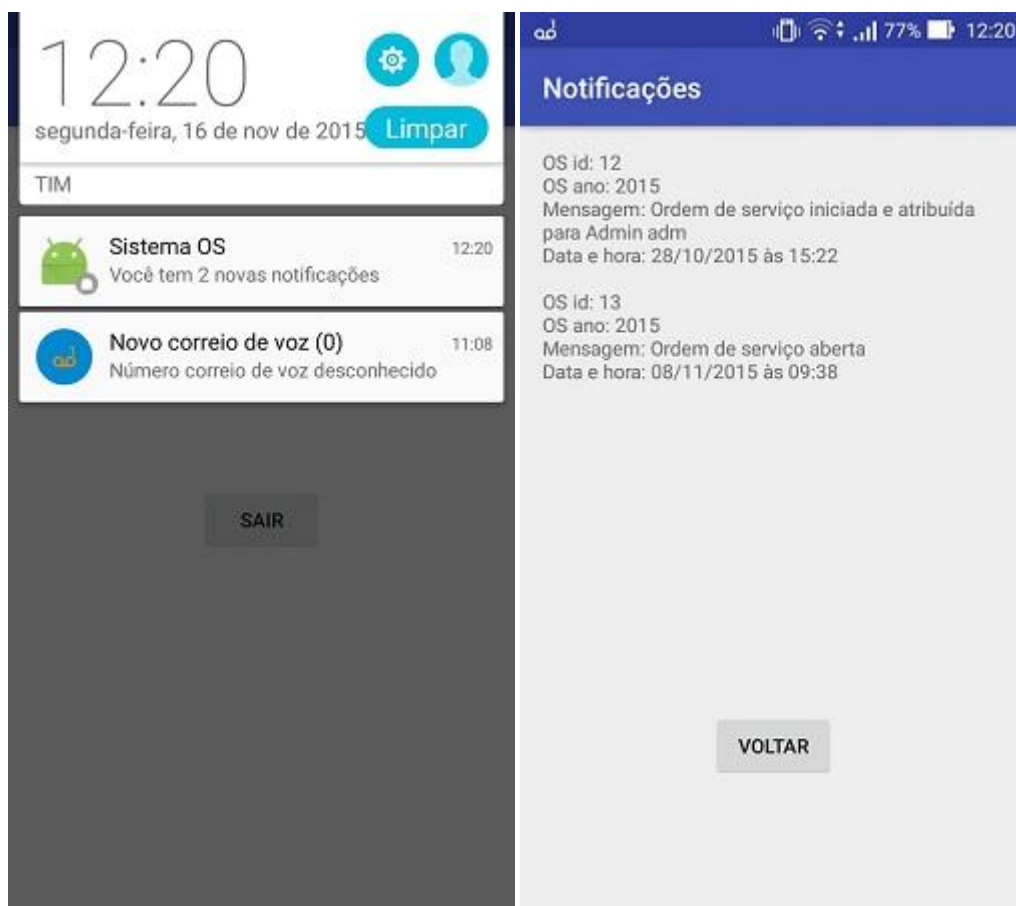
**Figura 14** – Imagem da tela de login do aplicativo Android



**Fonte** – Próprio autor.

Acima na Figura 14 foi apresentada a imagem que exibe a tela inicial do aplicativo *Andorid*, que é a tela de login do aplicativo.

**Figura 15** – Imagem das notificações do aplicativo Android



Fonte – Próprio autor.

A imagem exibida acima na Figura 15 mostra o alerta que o aplicativo gera para avisar ao usuário que ele tem novas notificações. Ao clicar neste alerta, o usuário é redirecionado a tela de notificações do aplicativo, onde é possível visualizar os status das ordens de serviço.

#### 4 CONSIDERAÇÕES FINAIS

Com este trabalho, foi possível identificar as vantagens em substituir sistemas utilizados de forma manual, por sistemas informatizados afim de atingir seus objetivos com maior precisão, agilidade e praticidade.

A proposta da monografia foi concluída com êxito, no qual o objetivo era o desenvolvimento de um sistema de ordem de serviço para facilitar o processo de abertura de solicitações de serviço e faço acesso as suas informações através de uma aplicação para *web* e uma para Android.

Como resultado do trabalho, foi criado um *web site* para que os usuários acessem o sistema e utilizem suas funcionalidades, um aplicativo para Android para que os usuários recebam notificações sobre as ordens de serviço e para a comunicação entre ambos, foi criado um *Web Service* baseado na arquitetura REST.

Em relação ao *Web Service*, foi possível notar o porquê é uma tecnologia muito utilizada ultimamente. Com ela é possível criar operações e deixa-las disponíveis para outros sistemas fazerem uso delas, independentemente de sua plataforma ou linguagem de programação utilizada. Outra grande utilidade, é a redução de código, por exemplo, se há um *Web Service* em uma empresa com uma operação de login e surgir a necessidade dessa empresa desenvolver uma nova aplicação que necessite de uma operação de login, pode ser usada a operação já existente do *Web Service*, economizando linhas de código e tempo.

#### 4.1 TRABALHOS FUTUROS

As aplicações que foram desenvolvidas permitem o acesso as funcionalidades necessárias para as interações com o sistema de ordem de serviço, mas ser aprimorado com implementações de novas funcionalidades e a fim de torna-lo mais eficiente.

Possíveis funcionalidades futuras identificadas:

- Recurso para o usuário comum avaliar o serviço realizado pela equipe de suporte ao finalizar uma ordem de serviço.
- Possibilidade de gerar relatórios com o intuito de coleta de dados, como identificar reincidentes, verificar satisfação do usuário, verificar tempo gasto para solucionar uma ordem de serviço.
- Implementação de um banco de dados local para armazenamento dos dados da aplicação Android.
- Implementação de *Web Views* no aplicativo Android, para o acesso fácil e rápido das funcionalidades completas do sistema através do aplicativo de notificações.

## REFERÊNCIAS BIBLIOGRÁFICAS

CHARLAND, Andre; LEROUX, Brian. **Mobile Application Development: Web vs. Native**. New York: Acm Queue, 2011. Disponível em: <http://dl.acm.org/citation.cfm?id=1968203> >. Acesso em: 11 jun. 2015.

CHEMAXOM. **JChem Web Services**. Disponível em: <https://www.chemaxon.com/products/jchem-web-services/>>. Acesso em: 25 jun. 2015.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado) - Curso de Information And Computer Science, University Of California, Irvine, 2000. Disponível em: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>. Acesso em: 05 nov. 2015.

FLING, Brian. **Mobile Design and Development**. 1st ed. California: O'Reilly, 2009.

FREIRE, Pedro J.; RIBEIRO, Rui. **Frameworks de Desenvolvimento Móvel Multiplataforma**. 2013. 11 f. - Universidade Lusófona, Lisboa, 2013. Disponível em: <http://copelabs.ulusofona.pt/scicommons/index.php/attachments/single/509>>. Acesso em: 11 jun. 2015.

LIMA, Jean Carlos Rosário. **WEB SERVICES (SOAP X REST)**. 2012. 41 f. TCC (Graduação) - Curso de Tecnologia em Processamento de Dados, Faculdade de Tecnologia de São Paulo, São Paulo, 2012. Disponível em: <http://www.fatecsp.br/dti/tcc/tcc00056.pdf>>. Acesso em: 25 jun. 2015.

LOTAR, Alfredo. **Programando com ASP.NET MVC: Aprenda a desenvolver aplicações web utilizando a arquitetura MVC**. São Paulo: Novatec, 2011.

MARQUES, José Alves. **Automatização de Processos**. jun. 2007. Disponível em: <http://www.link.pt/upl/{d6dfd44a-3c8a-43ec-9276-9a1bb4baa4f9}.pdf>>. Acesso em: 13 maio 2015.

MATEUS, Geraldo Robson; LOUREIRO, Antonio Alfredo Ferreira. **Introdução à Computação Móvel**. Rio de Janeiro: Dcc/im, Coppe/ufRJ, 1998. 189 p. Disponível em: [http://homepages.dcc.ufmg.br/~loureiro/cm/docs/cm\\_livro\\_1e.pdf](http://homepages.dcc.ufmg.br/~loureiro/cm/docs/cm_livro_1e.pdf)>. Acesso em: 13 nov. 2015.

MENDES, Roberto Damiani. **Desenvolvimento de um sistema de automação comercial para ambiente web**. 2005. 51 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal de Lavras, Lavras, 2005. Disponível em: [http://www.bcc.ufla.br/wp-content/uploads/2013/2005/Desenvolvimento\\_de\\_um\\_sistema\\_de\\_automacao\\_comercial\\_para\\_ambiente\\_web.pdf](http://www.bcc.ufla.br/wp-content/uploads/2013/2005/Desenvolvimento_de_um_sistema_de_automacao_comercial_para_ambiente_web.pdf)>. Acesso em: 08 jun. 2015.

MENÉNDEZ, Andrés Ignacio Martínez. **Uma ferramenta de apoio ao desenvolvimento de Web Services**. 2002. 97 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Campina Grande, Campina Grande, 2002. Disponível em:



<<https://ri.ufs.br/bitstream/123456789/1089/1/UmaFerramentaDeApoio.pdf>>. Acesso em: 21 maio 2015.

MICROSOFT. **ASP.NET e Visual Studio para Web**. Disponível em: <<https://msdn.microsoft.com/pt-br/library/dd566231.aspx>>. Acesso em: 19 set. 2015.

MICROSOFT. **Visual Studio**. Disponível em: <[https://msdn.microsoft.com/pt-br/library/52f3sw5c\(v=vs.90\).aspx](https://msdn.microsoft.com/pt-br/library/52f3sw5c(v=vs.90).aspx)>. Acesso em: 19 set. 2015.

MORO, Tharcis dal; DORNELES, Carina F.; REBONATTO, Marcelo Trindade. Web services WS-\* versus Web Services REST. **Reic - Revista de Iniciação Científica**, Passo Fundo, v. 11, n. 1, p.36-51, 2011. Disponível em: <<http://www.seer.ufrgs.br/reic/article/viewFile/22140/12928>>. Acesso em: 09 jun. 2015.

NEVES, Pedro M. C.; RUAS, Rui P. F.. **O Guia Prático do MySQL**. Centro Atlântico, 2005. Disponível em: <<https://novatec.com.br/livros/mysqlcompleto/capitulo8575221035.pdf>>. Acesso em: 19 set. 2015.

OLIVEIRA, Eva; VARANDA, Maria João; HENRIQUES, Pedro Rangel. **Compreensão de Aplicações Web: O Processo e as Ferramentas**. Bragança: Associação Portuguesa de Sistemas de Informação, 2005. Disponível em: <<http://hdl.handle.net/10198/577>>. Acesso em: 25 jun. 2015.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. São Paulo: Mcgraw Hill, 2011.

RASTOGI, Pranav et al. **Introduction to ASP.NET Identity**. 2013. Disponível em: <<http://www.asp.net/identity/overview/getting-started/introduction-to-aspnet-identity>>. Acesso em: 10 nov. 2015.

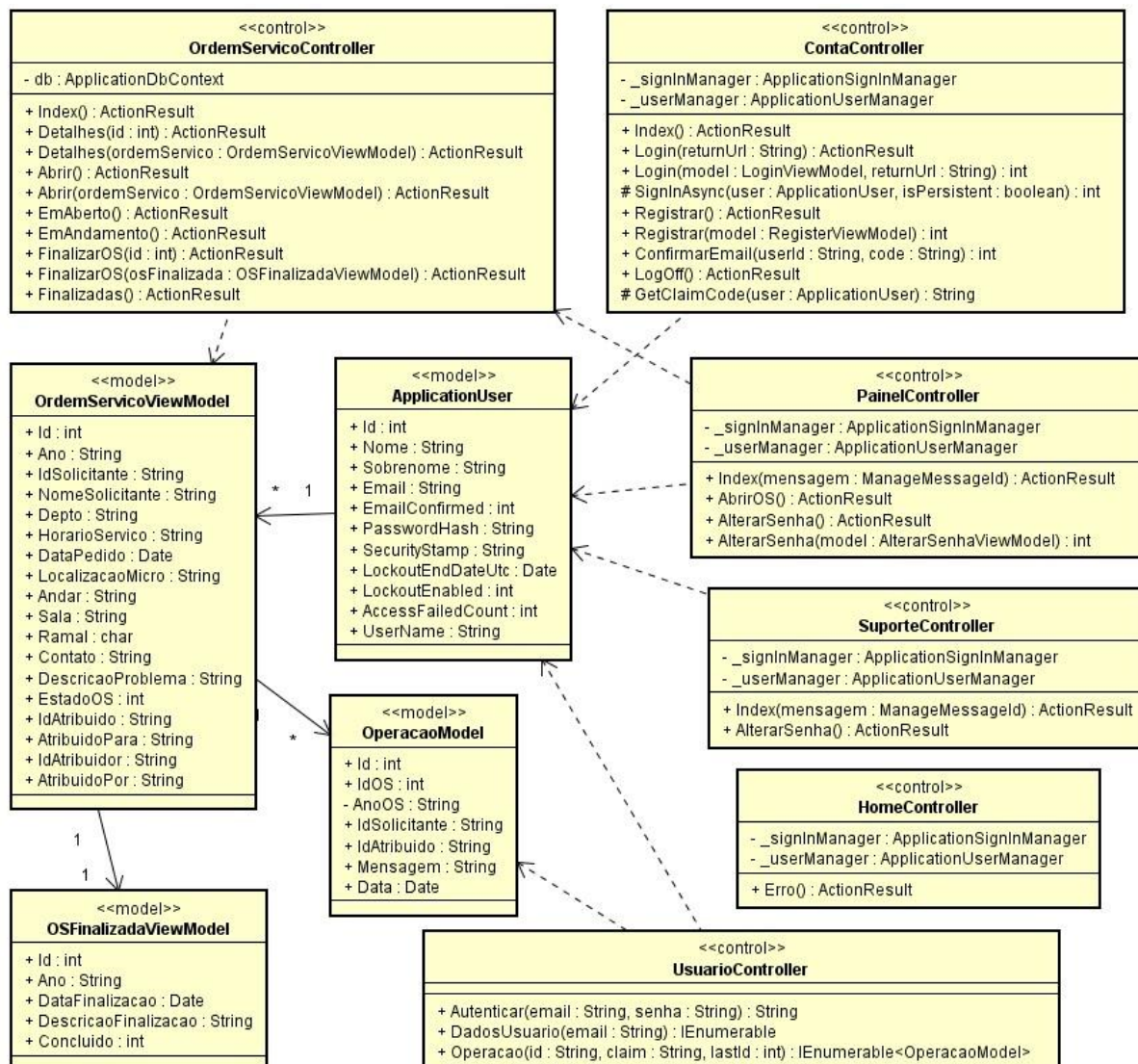
SEBRAE. **A informatização nas MPES paulistas**. 2003. Disponível em: <[http://www.sebraesp.com.br/arquivos\\_site/biblioteca/EstudosPesquisas/estudos\\_tematicos/informatizacao\\_mpes\\_2003.pdf](http://www.sebraesp.com.br/arquivos_site/biblioteca/EstudosPesquisas/estudos_tematicos/informatizacao_mpes_2003.pdf)>. Acesso em: 24 jun. 2015.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. Pearson Education, 2007.

VIDAL, Antonio Geraldo da Rocha; ZWICKER, Ronaldo; SOUZA, César Alexandre de. Um estudo da informatização em empresas industriais paulistas. **Revista de Administração Contemporânea**, Curitiba, jun. 2005. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1415-65552005000200009](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-65552005000200009)>. Acesso em: 24 jun. 2015.

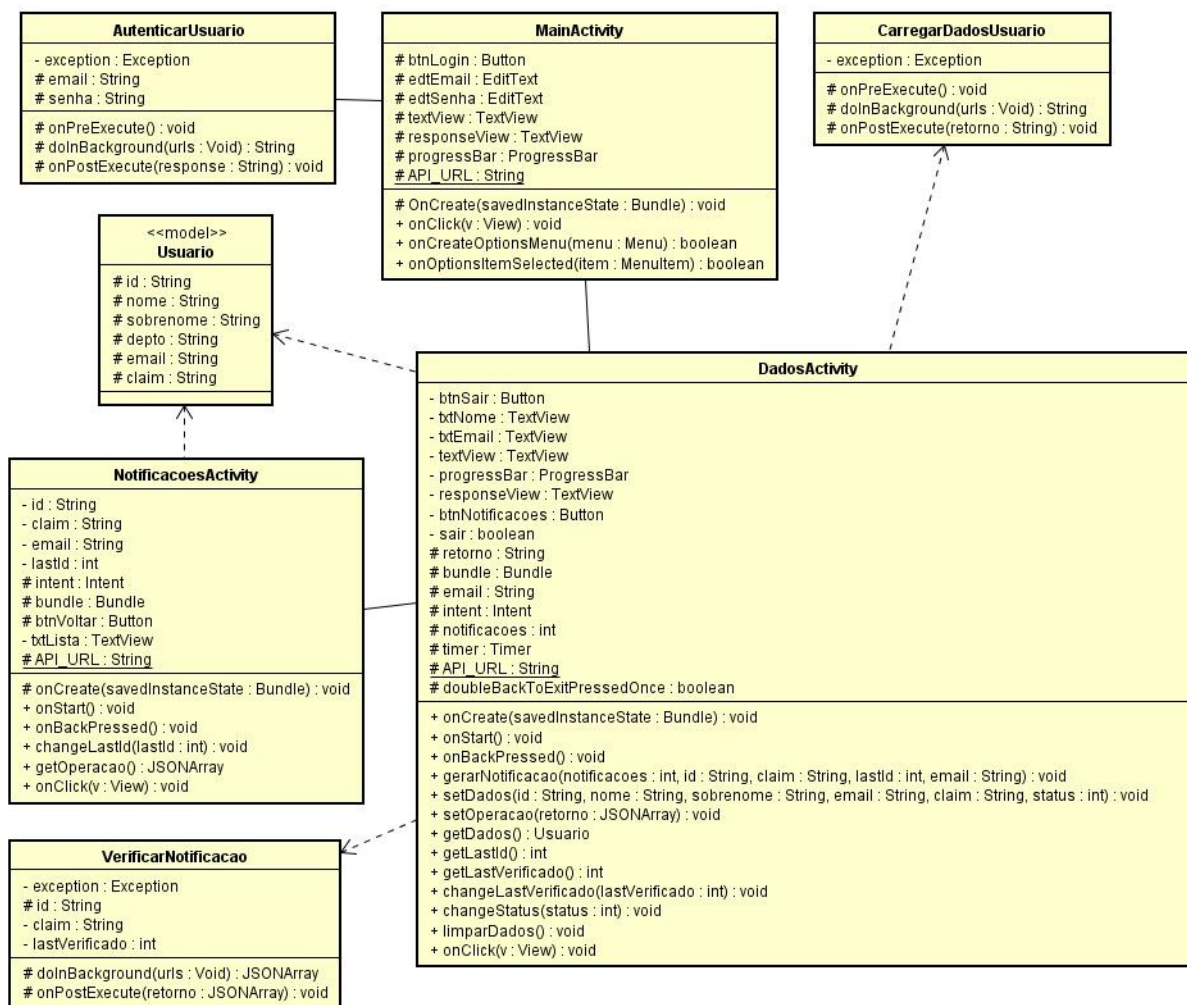
## APÊNDICE

Figura 16 – Diagrama de classes do sistema web



Fonte – Próprio autor.

Figura 17 – Diagrama de classes do aplicativo Android



Fonte – Próprio autor.

**Figura 18** – Exemplo de retorno de um método do Web Service

```
application/json, text/json

Sample:

[
  {
    "id": 1,
    "idos": 2,
    "anoOS": "sample string 3",
    "idSolicitante": "sample string 4",
    "idAtribuido": "sample string 5",
    "mensagem": "sample string 6",
    "data": "2015-11-18T15:19:34.6094231+00:00"
  },
  {
    "id": 1,
    "idos": 2,
    "anoOS": "sample string 3",
    "idSolicitante": "sample string 4",
    "idAtribuido": "sample string 5",
    "mensagem": "sample string 6",
    "data": "2015-11-18T15:19:34.6094231+00:00"
  }
]
```

Fonte – Próprio autor.