

---

Curso de Ciência da Computação  
Universidade Estadual de Mato Grosso do Sul

---

# Um Aplicativo Android para Trocas de Objetos

Adriana de Oliveira Palermo

Dr. Nilton César de Paula (Orientador)

Dourados – MS

2015



# Um Aplicativo Android para Trocas de Objetos

Adriana de Oliveira Palermo

Novembro de 2015

**Banca Examinadora:**

Prof. Dr. Nilton César de Paula (Orientador)  
Área de Computação – UEMS

Prof. Dr. Fabrício Sérgio de Paula  
Área de Computação – UEMS

Prof. Dr. Ricardo Luís Lachi  
Área de Computação – UEMS



# Um Aplicativo Android para Trocas de Objetos

Adriana de Oliveira Palermo

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso, devidamente corrigida e defendida por Adriana de Oliveira Palermo e aprovada pela Banca Examinadora, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Dourados, 23 de novembro de 2015.

Dr. Nilton César de Paula (Orientador)



# Agradecimentos

Agradeço a Deus por ter me guiado até aqui, dando força para superar todos os obstáculos e a Nossa Senhora por interceder por mim junto ao pai.

Aos meus pais Fernando Palermo e Luz Marina Palermo por tudo e principalmente por me proporcionar a oportunidade de estudar. E também às minhas irmãs Fernanda e Heliana.

Ao meu amado esposo Bruno que esteve partilhando comigo todos os momentos ajudando a ter forças para superar a cada dia.

A todos meus amigos que me ajudaram direta e indiretamente durante a graduação, Aline, Gleyciane, Natalia, Tiago Pereira, Tiago Antônio, Juliana, Geisi, a todos os meus colegas de trabalho e meus amados irmãos da Missão Enchei-vos que me acolheram e incentivaram.

Ao Professor Nilton César de Paula pela disponibilidade e toda orientação neste trabalho.

A todos os professores, colaboradores da Universidade Estadual de Mato Grosso do Sul, em especial os professores André Chastel Lima, Felipe Carbone, Jónison Almeida dos Santos e Rubens Barbosa Filho.



# Resumo

Atualmente com o crescimento nas compras, preços e prazos as pessoas passam a obter diversos itens e objetos, muito desses objetos passam a ser acumulados e ficam sem uso. Essas reservas de objetos e itens podem atrapalhar as pessoas, diminuindo o espaço físico e até estragar pela falta de uso. Para suprir essa necessidade e proporcionar uma solução, este trabalho tem a finalidade de construir uma aplicação de intermediação de trocas de objetos, semelhante à um Classificados, desenvolvida para dispositivo móvel com sistema operacional *Android*. O aplicativo de trocas, denominado Vai & Vem, tem como objetivo facilitar a troca de objetos incentivando a não se acumular itens e objetos inutilizados. Essa prática já era utilizada pelos nossos ancestrais e conhecida como escambo. Na época eram realizadas trocas comerciais entre os habitantes e a negociação era realizada, se as duas partes estivessem de acordo.

Palavra-chave: *Android*, Trocas, Classificados, Intermediação.



# Sumário

Agradecimentos.....	vii
Resumo.....	ix
Sumário.....	xi
Lista de Siglas.....	xiii
Lista de Figuras.....	xv
Lista de Tabelas.....	xvii
Capítulo 1.....	19
Introdução.....	19
1.1    Objetivos.....	19
1.1.1    Objetivos específicos.....	20
1.2    Justificativa e motivação.....	20
1.3    Metodologia.....	20
1.1    Organização do texto.....	21
Capítulo 2.....	23
Android.....	23
2.1    Arquitetura do <i>Android</i> .....	24
2.1.1    Núcleo Linux.....	24
2.1.2    Bibliotecas.....	25
2.1.3 <i>Android</i> Tempo de Execução (Runtime).....	25
2.1.4    Quadro de Aplicações.....	27
2.1.5    Aplicações.....	27
2.1    Versões do <i>Android</i> .....	27
Capítulo 3.....	29
Desenvolvimento.....	29
3.1    Funcionamento.....	29
3.2    Ambiente de desenvolvimento.....	30
3.3    Descrição da aplicação.....	31
3.4    Principais classes.....	37
3.4.1    Módulo aplicativo.....	37
3.4.2    Módulo aplicação PHP <i>WEB</i> .....	39
3.4.3    Módulo banco de dados.....	40
3.5    Diagrama de caso de uso.....	43
Capítulo 4.....	45
Interfaces do Aplicativo.....	45
Capítulo 5.....	59

Conclusão .....	59
5.1    Trabalhos futuros .....	59
Referências Bibliográficas.....	61
Apêndice A.....	63
Instalação e configuração do <i>Android Studio</i> .....	63
Apêndice B .....	69
Instalação e configuração do XAMPP.....	69
Apêndice C .....	75
Instalação e configuração do PostgreSQL .....	75

# Lista de Siglas

AOT - *Ahead-of-time*

API - *Applications Programming Interface*

ART - *Android Runtime*

HTTP - *Hypertext Transfer Protocol*

IDE - *Integrated Development Environment*

JSON - *JavaScript Object Notation*

MV - Máquina Virtual

OHA - *Open Handset Alliance*

PHP - *PHP: Hypertext Preprocessor*

SDK - *Software Development Kit*

SGBD - Sistema Gerenciador de Banco de Dados

SO - Sistema Operacional



# Lista de Figuras

Figura 1 - Arquitetura da plataforma Android .....	24
Figura 2 - Máquina JAVA tradicional e Máquina Virtual Dalvik.....	26
Figura 3 - Representação gráfica do login de acesso .....	32
Figura 4 - Representação gráfica do cadastro de usuário .....	32
Figura 5 - Representação gráfica do Cadastro de Objeto.....	33
Figura 6 - Representação gráfica da lista de categorias .....	34
Figura 7 - Representação gráfica da lista de objetos.....	34
Figura 8 - Representação gráfica dos detalhes do objeto .....	35
Figura 9 - Representação gráfica da Solicitação de Troca .....	36
Figura 10 - Representação gráfica do cancelamento de troca.....	36
Figura 11 - Representação gráfica da efetivação de troca .....	37
Figura 12 - Diagrama Entidade Relacionamento.....	41
Figura 13 - Diagrama de Caso de Uso .....	43
Figura 14 - Tela Splash do aplicativo Vai & Vem.....	45
Figura 15 - Tela de Login do aplicativo .....	46
Figura 16 - Tela de Cadastro de Usuário.....	47
Figura 17 - Tela que lista as Categorias que possui ao menos um objeto cadastrado.....	47
Figura 18 - Lista de objetos da Categoria Eletrônicos .....	48
Figura 19 - Detalhes do objeto e botão Solicitar Troca .....	49
Figura 20 - Tela de Efetivar Solicitação .....	49
Figura 21 - Tela com todas as trocas solicitadas pelo usuário .....	50
Figura 22 - Status do produto em negociação .....	51
Figura 23 - Objeto em negociação .....	51
Figura 24 - Lista de funcionalidades presentes no menu de opções .....	52
Figura 25 - Tela Meus Dados .....	53
Figura 26 - Tela de Cadastro de Produto .....	53
Figura 27 - Tela Meus Produtos.....	54
Figura 28 - Trocas Recebidas .....	54
Figura 29 - Aceitar ou cancelar uma solicitação de troca .....	55
Figura 30 - Status da troca CANCELADA.....	55
Figura 31 - Objeto disponível novamente .....	56
Figura 32 - Troca efetivada .....	56
Figura 33 - Objeto saiu da lista depois de ter sido trocado .....	57
Figura 34 - Instalação e configuração do Android Studio .....	63
Figura 35 - Instalando o Java .....	64
Figura 36 - Configurar o JAVA HOME.....	65

Figura 37 - Telas de boas-vindas do Android Studio .....	66
Figura 38 - SDK Manager .....	66
Figura 39 - Instalação e configuração do XAMPP .....	69
Figura 40 - Download do XAMPP .....	70
Figura 41 - Instalação do XAMPP .....	70
Figura 42 - Defina o diretório onde o XAMPP será instalado.....	71
Figura 43 - Selecione os arquivos à serem instalados.....	72
Figura 44 - Progresso da instalação do XAMPP.....	72
Figura 45 - Alerta de segurança do Windows .....	73
Figura 46 - Finalização da instalação do XAMPP .....	73
Figura 47 - Painel de Controle do XAMPP .....	74
Figura 48 - Instalação e configuração do PostgreSQL.....	75
Figura 49 - Página de Download do PostgreSQL .....	76
Figura 50 - Instalação do PostgreSQL .....	76
Figura 51 - Defina o local onde o PostgreSQL será instalado .....	77
Figura 52 - Cadastre uma senha para seu servidor PostgreSQL.....	77
Figura 53 - Definir porta .....	78
Figura 54 - Deixe a opção padrão selecionada.....	78
Figura 55 - Progresso da instalação.....	79
Figura 56 - Finalizar instalação .....	79

# Lista de Tabelas

Tabela 1 - Versões do Android.....	27
Tabela 2 - Principais classes do aplicativo Android .....	38



# Capítulo 1

## Introdução

Nos dias atuais, seja pela comodidade, fácil comunicação, lazer ou acesso à Internet, a grande maioria das pessoas tem como objeto inseparável um dispositivo móvel. Com a inovação tecnológica, a computação móvel vem se destacando como um novo modelo computacional muito utilizado (FIGUEIREDO; NAKAMURA, 2003). Percebendo isso, grandes empresas e desenvolvedores tem realizado grandes investimentos na criação e desenvolvimento de aplicativos que auxiliam o cotidiano das pessoas.

O *Android*, sistema operacional para dispositivos móvel do Google, se destaca neste cenário, pois está presente em mais de 81% dos *smartphones* em todo o mundo, ficando à frente do iOS, sistema operacional da Apple, a qual está presente em torno de 14% dos aparelhos e do Windows Phone, que soma 2,7% dos dispositivos em todo o mundo (AND-MER-MUN). Mas esse sucesso todo não é mérito somente do Google, por trás do desenvolvimento estão os grandes profissionais do mercado da mobilidade, fabricantes e operadoras. O grupo que ajuda no desenvolvimento dessa plataforma se chama OHA (*Open Handset Alliance*), que se divide em várias empresas de renome como Intel, Sprint, Nextel, ASUS, Acer, Dell, Garmin etc. Atualmente o *Android* está disponível para dispositivos móvel e tablets, além de relógios, televisões e carros (LECHETA, 2015).

### 1.1 Objetivos

Este trabalho tem como objetivo principal um estudo, o projeto e o desenvolvimento de um aplicativo móvel, no estilo Classificados, para a plataforma *Android*, cuja finalidade é a intermediação da troca de objetos entre alunos da universidade, que não possui mais serventia para seus donos e preferencialmente ainda sejam úteis no meio acadêmico.

### 1.1.1 Objetivos específicos

O resultado esperado com o desenvolvimento deste trabalho é a construção de um aplicativo que possibilite um melhor reaproveitamento de objetos, através da intermediação de trocas.

Os objetivos específicos deste trabalho são:

- Pesquisar sobre as características da plataforma *Android*;
- Pesquisar sobre programação *WEB*, utilizando a linguagem PHP;
- Pesquisar sobre Sistema de Bancos de Dados;
- Modelar o Diagrama de Entidade e Relacionamento;
- Projetar e implementar uma simples aplicação PHP que será a intermediária entre as informações do banco de dados e o aplicativo *Android*;
- Projetar e implementar o aplicativo *Vai & Vem* que possibilite a intermediação de troca de objetos entre alunos através de dispositivos móveis.

### 1.2 Justificativa e motivação

Diversos alunos, podem possuir inúmeros objetos e materiais “encostados” em suas casas, que estão sem utilidades para eles, os quais poderiam ser muito importantes para outros alunos em diversas situações. A motivação é disponibilizar um aplicativo capaz de viabilizar a intermediação de troca de produtos, onde aquele produto que não lhe serve mais, pode ser trocado por algo que lhe seja muito útil.

### 1.3 Metodologia

Para a pesquisa sobre a plataforma *Android* foi utilizado artigos na Internet, bem como teses e dissertações e livros, sobretudo o livro *Google Android*, de Ricardo Lecheta (LECHETA, 2015).

O servidor que hospedará a aplicação *WEB* e o banco de dados serão executados no Sistema Operacional (SO) Windows.

O Sistema Gerenciador de Banco de Dados (SGBD) escolhido foi o PostgreSQL (PSQL, 2015) que é um projeto de código aberto. Para gerenciar as informações do banco, foi utilizado o PGADMIN III (PSQL, 2015). Para a modelagem do banco foi utilizado um software chamado pgModeler (PGMODEL, 2015).

Para o desenvolvimento da aplicação *WEB*, que fará toda a comunicação entre o aplicativo *Android* e o Banco de Dados, foi escolhida a linguagem PHP (PHP, 2015).

Para a implementação do aplicativo foi escolhida a versão 5.0 do *Android* com API level a partir de 18, e foi utilizada a IDE de desenvolvimento *Android Studio* (AND-STUDIO, 2015).

Toda a comunicação do aplicativo com o servidor de hospedagem é feita através de *webservices* via protocolo HTTP.

## **1.1 Organização do texto**

O texto está organizado em um único volume. Além deste primeiro capítulo que apresentou o conteúdo introdutório deste trabalho, bem como seus objetivos gerais e específicos, a justificativa e motivação e a metodologia aplicada, o trabalho é organizado em outros 4 capítulos, e 3 apêndices cujos conteúdos são sumarizados a seguir:

### **Capítulo 2**

#### ***Android***

No segundo capítulo é apresentado um estudo geral sobre a plataforma *Android*, bem como seus conceitos, sua arquitetura e as versões existentes até o momento.

### **Capítulo 3**

#### **Desenvolvimento**

No terceiro capítulo é apresentado o projeto de desenvolvimento do aplicativo *Vai & Vem*, bem como o funcionamento da aplicação.

## **Capítulo 4**

### **Interfaces do Aplicativo**

No quarto capítulo é apresentado os testes realizados visando determinar se o projeto atingiu suas especificações e verificar se seus objetivos foram alcançados.

## **Capítulo 5**

### **Conclusão**

No quinto capítulo são apresentadas as considerações finais do trabalho e algumas melhorias futuras para a aplicação.

### **Referências Bibliográficas**

Nesta seção são disponibilizadas as referências bibliográficas.

### **Apêndice A**

No Apêndice A são apresentadas as instruções para instalação e configuração da IDE *Android Studio*.

### **Apêndice B**

No Apêndice B são apresentadas as instruções para instalação e configuração da XAMPP.

### **Apêndice C**

No Apêndice C são apresentadas as instruções para instalação e configuração da PostgreSQL.

# Capítulo 2

## Android

O *Android* é a primeira plataforma de aplicações móveis de código aberto no mercado, pois várias empresas e também desenvolvedores podem contribuir para a melhoria da plataforma. Baseado no Linux, o sistema operacional *Android* teve seu desenvolvimento iniciado em 2003 pela empresa *Android Inc*, que tinha como objetivo desenvolver sistemas operacionais com funcionalidades específicas para câmeras digitais. Como esse mercado não obteve o sucesso esperado, resolveram mudar o foco para o desenvolvimento mobile (A-AND, 2015). Em 2005, a empresa foi adquirida pelo Google, onde nasceu a divisão de pesquisa em tecnologia móvel, a Google Mobile Division, que hoje lidera o desenvolvimento do sistema (MONTEIRO, 2013).

Um fato marcante para o nascimento do *Android* teve início em 2007 quando gigantes do mercado de telefonia, *software* e *hardware*, como Intel, LG, Samsung, Motorola, Sony, Dell, Acer, dentre outras, formaram um grupo liderado pelo Google, cujo principal objetivo era o desenvolvimento de uma plataforma para dispositivos móveis que fosse completa, aberta e gratuita. Chamado de *Open Handset Alliance – OHA*, o grupo era formado por mais de 80 potências da tecnologia (OHA, 2015) e segundo elas, mais de 1 bilhão de pessoas tem acesso à internet e cerca de 3 bilhões tem um telefone celular, tornando o aparelho um dos produtos de consumo mais bem-sucedidos no mundo. Dessa maneira, construir novas tecnologias para um aparelho celular melhoraria a vida de inúmeras pessoas no mundo todo (LECHETA, 2015).

Além de definir uma plataforma única e aberta para celulares e deixar os consumidores mais satisfeitos com o produto final, outro objetivo principal da OHA, é criar uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas, e o resultado dessa união foi o nascimento do *Android* (LECHETA, 2015).

## 2.1 Arquitetura do *Android*

O *Android* é uma pilha de *software* composta por cinco camadas, que são divididas em níveis: Nucleo Linux, Bibliotecas, *Android* Tempo de Execução, Quadro de Aplicações e Aplicações (MONTEIRO, 2013), conforme ilustra a **Figura 1**.



Figura 1 - Arquitetura da plataforma *Android*

A base do sistema *Android*, a camada nível zero, é uma versão modificada do *kernel* do Linux 2.6, cuja responsabilidade é disponibilizar serviços essenciais como segurança, rede e gerenciamento de processos e memória. O *kernel* é a camada de comunicação entre o *hardware* e as outras camadas do *software*. No *Android* vários aplicativos e processos podem ser executados ao mesmo tempo, ficando sob responsabilidade do *kernel* o controle e o gerenciamento dos mesmos, podendo encerrá-los e iniciá-los quando for necessário (LECHETA, 2015).

### 2.1.1 Núcleo Linux

A camada do núcleo, ou *kernel*, é baseada no sistema operacional Linux versão 2.6 e é responsável pela comunicação entre o *hardware* e os aplicativos, e oferece serviços principais

do *Android*, como o gerenciamento de processos e memória (PAULA, 2013). Várias funções do núcleo Linux são utilizadas diretamente pelo *Android*, mas muitas modificações foram realizadas com o passar do tempo para otimizar a memória e tempo de processamento das aplicações. Algumas dessas modificações são a inclusão de novos dispositivos de drivers, adições no sistema de gerenciamento de energia e um sistema que possibilita terminar processos de maneira eficiente e criteriosa quando há pouca memória disponível.

### 2.1.2 Bibliotecas

As bibliotecas são responsáveis por dar suporte e manipular os arquivos de áudio, vídeo, gráficos, banco de dados e browser. A grande maioria das bibliotecas são desenvolvidas em C/C++ e chamadas através de uma interface JAVA. As funcionalidades disponibilizadas pelas bibliotecas são acessadas através do *framework* de aplicações (AND-STUDIO, 2015).

### 2.1.3 *Android* Tempo de Execução (Runtime)

O *Android* Runtime permite que cada *thread* execute sua própria instância da máquina virtual (MV). É fato que a linguagem JAVA é utilizada para construir as aplicações para o *Android*, mas elas não são executadas em uma máquina virtual JAVA tradicional. Na verdade, o que temos é uma máquina virtual chamada Dalvik que é otimizada especialmente para execução em dispositivos móveis (LECHETA, 2015). A **Figura 2** mostra uma comparação entre a máquina virtual JAVA e a máquina virtual Dalvik.

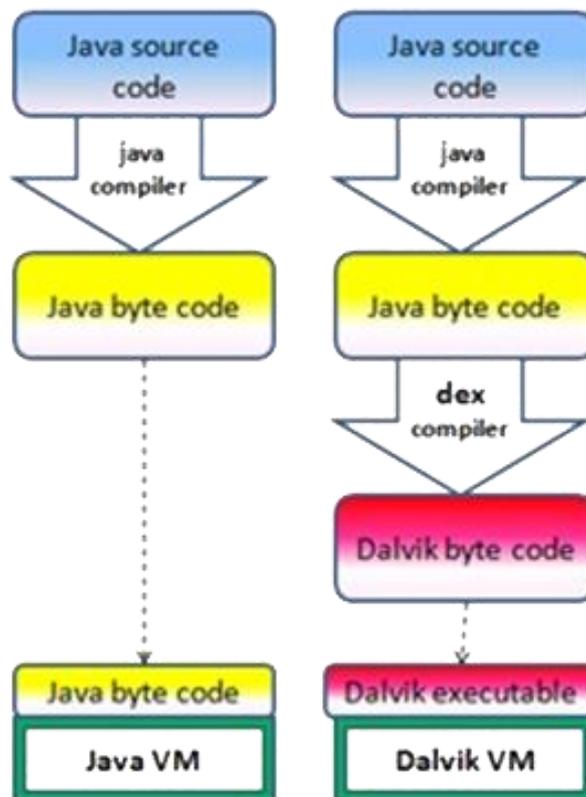


Figura 2 - Máquina JAVA tradicional e Máquina Virtual Dalvik

Ao desenvolver as aplicações para o *Android*, todos seus recursos são utilizados normalmente, porém depois que o *bytecode* é compilado, ele é convertido para o formato *.dex* (*Dalvik Executable*), que representa a aplicação do *Android* compilada. Posteriormente os arquivos *.dex* são compactados juntamente com as imagens em um único arquivo com a extensão *.apk* (*Android Package File*), que representa a aplicação final, pronta para ser distribuída e instalada (LECHETA, 2015).

A partir do *Android* 4.4 (KitKat), com o objetivo de substituir a Dalvik, foi criada uma máquina virtual opcional chamada ART – *Android Runtime*, que contém melhorias na compilação *Ahead-of-time* (AOT), cujo objetivo é a otimização do código ao máximo, melhorando assim também o desempenho. Já na versão 5.0 (*Lollipop*), o ART se tornou a máquina virtual padrão, substituindo de vez a Dalvik.

### 2.1.4 Quadro de Aplicações

O Quadro (*Framework*) de Aplicações disponibiliza todas as APIs (*Applications Programming Interface*) utilizadas para a criação de aplicações do sistema operacional *Android*. Com isso os desenvolvedores têm o mesmo acesso ao sistema que os aplicativos da camada de aplicativos possuem e podem usufruir de todas as funcionalidades que o sistema pode oferecer (GARGENTA, 2011). O objetivo principal deste *Framework* é abstrair a complexidade e simplificar a reutilização de procedimentos.

### 2.1.5 Aplicações

A camada de aplicações está no topo da arquitetura do *Android*. Ela é composta por todos as aplicações presentes no sistema, incluindo as nativas, como o cliente de *e-mail*, despertador, calendário, mapas e as desenvolvidas por terceiros e adquiridas através do Google Play.

## 2.1 Versões do *Android*

A primeira versão do *Android* chegou ao mercado no ano de 2008 e a cada versão são adicionadas novas funcionalidades ao sistema operacional, corrigindo os erros existentes e criando novas funções. A Tabela 1 mostra todas as versões e o ano de lançamento de cada uma. A partir da versão 1.5, cada versão do *Android* possui o nome de doce em ordem alfabética (LECHETA, 2015).

Tabela 1 - Versões do *Android*

NOME	VERSÃO	ANO
Alpha	1.0	2008
Beta	1.1	2009
Cupcake	1.5	2009
Donut	1.6	2009
Eclair	2.0 – 2.1	2009
Froyo	2.2 – 2.2.3	2010
Gingerbread	2.3 – 2.3.7	2010
Honeycomb	3.0 – 3.2.6	2011
Ice Cream Sandwich	4.0 – 4.0.4	2011

Jelly Bean	4.1 – 4.3.1	2012
KitKat	4.4 – 4.4.4	2013
Lollipop	5.0 – 5.0.2	2014
Marshmallow	6.0	2015

# Capítulo 3

## Desenvolvimento

Este capítulo é dedicado a detalhar o projeto de desenvolvimento do aplicativo Vai & Vem, destinado aos alunos que desejam trocar todo e qualquer tipo de objeto utilizado na universidade. Descreve como será o funcionamento da aplicação, bem como sua parte estrutural, detalhando sobre o ambiente de desenvolvimento utilizado. Em seguida serão destacadas algumas das principais classes do projeto, bem como o diagrama de caso de uso e o diagrama entidade relacionamento, finalizando com alguns testes realizados.

### 3.1 Funcionamento

A grande maioria dos alunos possui algum objeto que já não tem tanta utilidade como um dia já teve em sua vida, seja ele um livro, um caderno, uma mochila, um computador, uma calculadora, dentre tantas outras coisas. Com o passar do tempo na faculdade, acabamos “esquecendo” este objeto em algum lugar da casa por dias, meses, ou até anos. Ele fica lá, sem utilidade alguma, apenas ocupando espaço que poderia ser aproveitado para outra coisa e quando nos “lembramos” de sua existência, jogamos fora. O que não pensamos é que ele poderia ter uma grande utilidade na vida de outra pessoa.

Pensando nisso, foi proposto o desenvolvimento de alguma solução que possibilitasse que as pessoas trocassem seus objetos sem utilidade com outras pessoas que pudessem estar precisando. O nome do projeto vem do conceito de troca, ou seja, substituir uma coisa por outra, onde um objeto vai e outro vem em seu lugar.

O Aplicativo consiste na intermediação de trocas de objetos entre usuários cadastrados no sistema. Para poder cadastrar um objeto ou visualizar os já disponíveis para troca, é necessário realizar o *login* de acesso. Caso ainda não seja cadastrado, o usuário deve realizar um breve cadastro informando seu nome, seu *e-mail*, o número de seu celular, um *login* e uma senha.

Os objetos estarão classificados por categorias pré-cadastradas, o que facilita para o usuário na hora de realizar uma busca de um objeto de seu interesse. Ao cadastrar um objeto, o usuário deverá tirar uma foto, informar um título, uma breve descrição, selecionar a categoria que ele pertencerá e informar seu tipo de troca. Existem dois tipos de troca: a troca

Restrita e a troca Aberta. Um objeto que seja cadastrado como troca Restrita, só pode ser trocado por um outro objeto específico informado pelo usuário. Já o tipo troca Aberta, pode ser trocado por qualquer objeto.

Para fazer a busca de um objeto, o usuário deverá selecionar uma categoria dentre as listadas e então uma nova lista é apresentada, desta vez com todos os objetos da categoria selecionada. Com isso basta o usuário procurar aquele objeto que seja de seu interesse e selecioná-lo. Ao selecionar, algumas informações do produto são apresentadas, assim como um botão que possibilita realizar uma solicitação de troca e disponibiliza as informações de contato do dono do objeto.

Para que a aplicação de trocas de objetos funcione, existe, além do aplicativo *Android*, uma aplicação *WEB*, que nada mais é do que um comunicador intermediário entre o aplicativo *Android* e o banco de dados que armazena as informações. Esta aplicação *WEB* foi desenvolvida na linguagem PHP 5.2 (PHP, 2015) e o Sistema Gerenciador de Banco de Dados (SGBD) que faz o gerenciamento das informações é o PostgreSQL 9.4 (PSQL, 2015). Toda a comunicação entre o aplicativo Vai & Vem e a aplicação *WEB* é feita via internet, através do protocolo *HTTP*.

### **3.2 Ambiente de desenvolvimento**

Para desenvolver o aplicativo *Android* de trocas Vai & Vem, foi necessário preparar e configurar o ambiente de desenvolvimento instalando o *Software Development Kit* (SDK – Kit de Desenvolvimento de Software) (AND-STUDIO, 2015), que possui o emulador e todas as ferramentas necessárias. O *Android* SDK (AND-STUDIO, 2015) é o software utilizado para desenvolver aplicações no *Android*, que possui um emulador para poder simular um dispositivo móvel, ferramentas utilitárias e uma completa API para a linguagem JAVA. Já para o desenvolvimento da aplicação *WEB* foi utilizado o servidor XAMPP (XAMPP, 2015), que é independente de plataforma. Sua instalação acompanha o servidor *WEB* Apache (APACHE, 2015), o MySQL (MYSQL, 2015) e os interpretadores para linguagem de script: PHP (PHP, 2015) e Perl (PERL, 2015). O “X” da sigla significa que o pacote pode ser utilizado em qualquer sistema operacional (XAMPP). O banco de dados utilizado foi o PostgreSQL (PSQL, 2015), em sua versão 9.4 e o gerenciador para controlar as informações do banco foi o

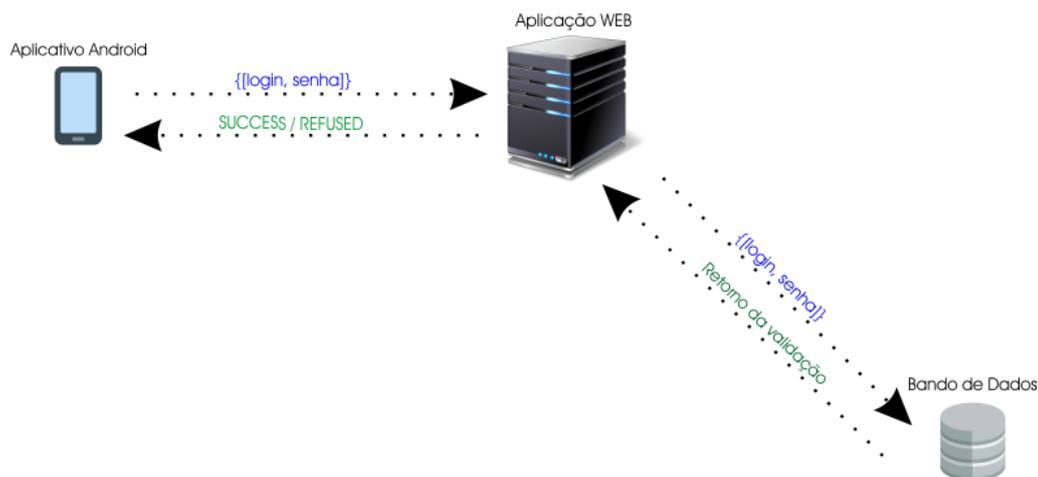
pgAdmin III (PSQL, 2015), que vem acompanhado no instalador do PostgreSQL (PSQL, 2015).

No projeto do aplicativo Vai & Vem, foi utilizado o *Android Studio 1.4* (AND-STUDIO, 2015), IDE oficial de desenvolvimento para *Android*. Até 2013 a IDE mais utilizada para o desenvolvimento de aplicativos *Android* era o Eclipse ADT (ECLIPSE, 2015), uma ferramenta bastante conhecida entre os desenvolvedores JAVA, porém, em uma conferência de programadores que é organizada anualmente pelo Google, chamada Google I/O, foi anunciado pela própria empresa que o *Android Studio* seria a IDE oficial do *Android*. Dentre alguns diferenciais importantes se comparado ao Eclipse (ECLIPSE, 2015), o *Android Studio* (AND-STUDIO, 2015) possui um editor visual com mais opções, um sistema de *build* mais moderno baseado em *Gradle*, diversas utilidades e facilidades ao desenvolver para *Android*, sendo totalmente integrado ao *Android SDK*, *templates* de projetos para *smartphones*, *tablets*, relógios e atualizações e melhorias frequentes (LECHETA, 2015).

### 3.3 Descrição da aplicação

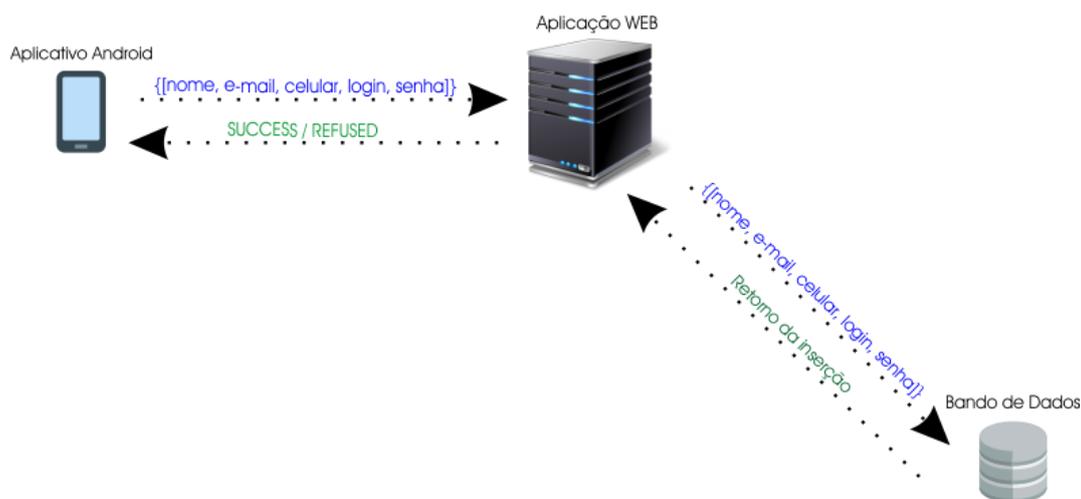
Ao iniciar o aplicativo é exibida uma tela de apresentação do sistema. Esta tela é conhecida como a tela *Splash* e fica visível por alguns segundos antes de surgir a tela principal do aplicativo. Usa-se esta tela *Splash* para apresentar informações básicas e iniciais do aplicativo. No caso do Vai & Vem é apresentado um logo da aplicação.

Após a tela *Splash* o usuário é redirecionado para a tela de *login*, onde deverá efetuar o acesso, caso já seja cadastrado, informando seu *login* e senha. Os dados de acesso são enviados pelo aplicativo *Android* à aplicação *WEB* no servidor, onde a mesma fará a validação junto ao banco de dados e retornará uma resposta ao aplicativo informando se os dados de acesso estão corretos ou não. A **Figura 3** ilustra como é feito o login de acesso ao aplicativo de trocas.



*Figura 3 - Representação gráfica do login de acesso*

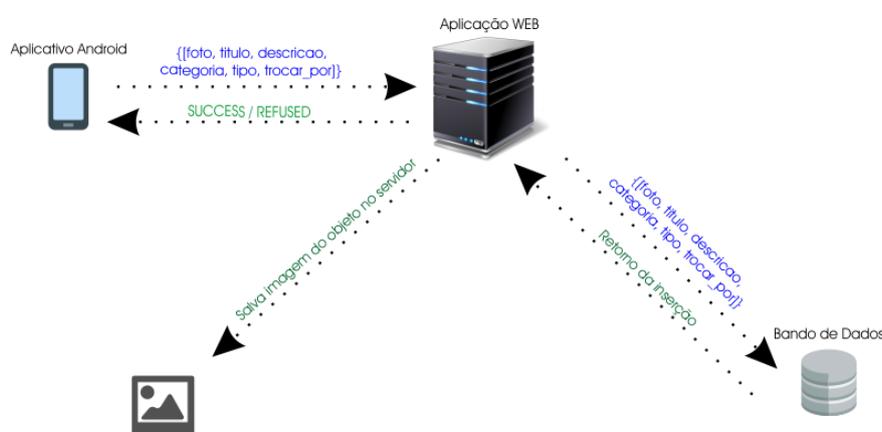
Caso ainda o usuário não seja cadastrado ele deverá realizar o cadastro para acessar o sistema. Este cadastro pode ser feito clicando no botão “Cadastre-se” da tela de *login*. As informações necessárias para o cadastro são: nome, *e-mail*, número do celular, *login* e senha. Para processar o cadastro do usuário, o aplicativo envia as informações para a aplicação *WEB* que realizará a inserção no banco de dados. Se as informações forem incluídas com sucesso ou não, o banco de dados informará à aplicação *WEB* que por sua vez informará o aplicativo *Android*. A **Figura 4** ilustra o procedimento de cadastro de usuário.



*Figura 4 - Representação gráfica do cadastro de usuário*

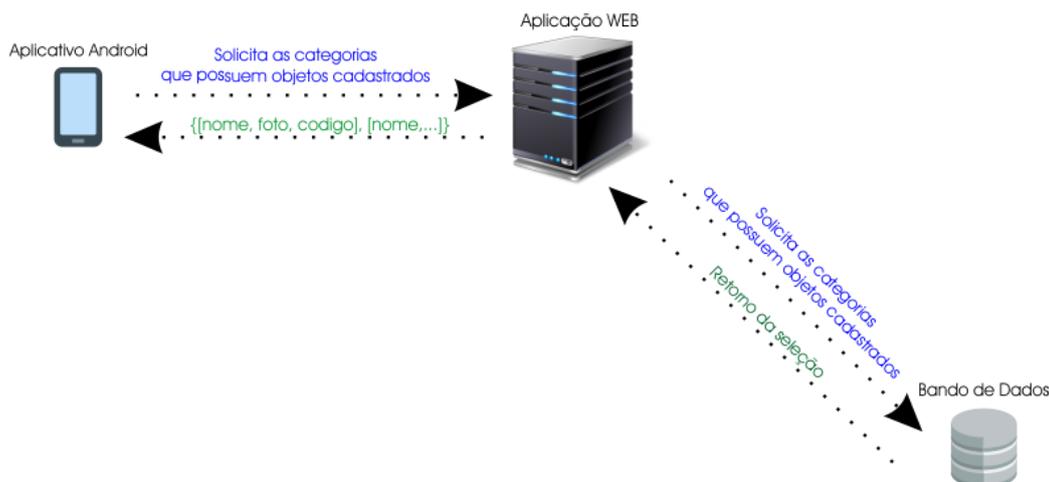
Para cadastrar um novo objeto o usuário deverá selecionar a opção Cadastrar Produto

e uma nova tela será aberta. Nesta opção o usuário deverá inserir as informações referentes ao objeto que está sendo cadastrado, sendo: uma foto, um título, uma breve descrição, a categoria que ele pertencerá e o tipo de troca. Sendo o tipo de troca Restrita, então ele deve fornecer também por qual objeto deseja trocar o seu. A **Figura 5** ilustra o Cadastro de Objeto. A única particularidade em relação às outras funcionalidades do sistema, é que o arquivo da foto do objeto deve ser salvo no servidor, através da aplicação *WEB*. Após salvar o arquivo, as informações são inseridas no banco de dados.



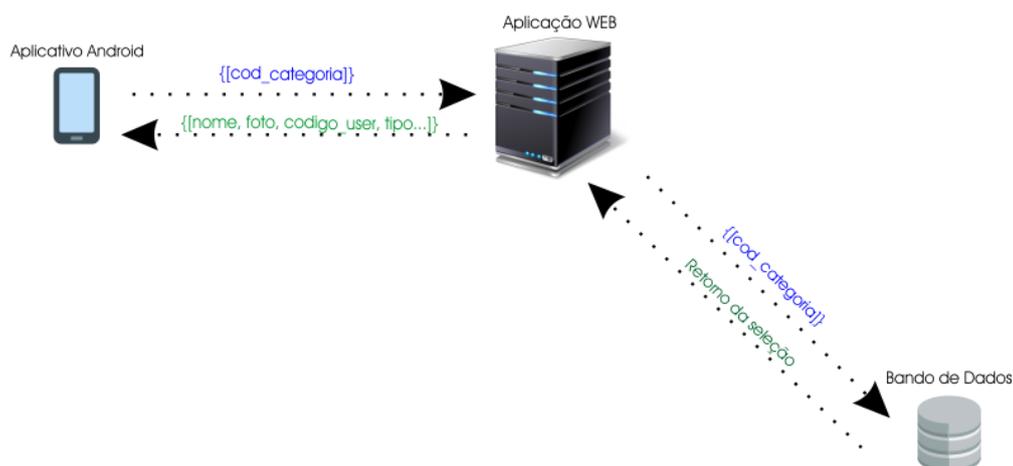
*Figura 5 - Representação gráfica do Cadastro de Objeto*

Após efetuar o *login* de acesso ou realizar o cadastro com sucesso, o aplicativo abre a tela que lista a categoria dos objetos. Nesta tela o usuário já pode visualizar todas as categorias cadastradas que possuem algum objeto disponível para troca. Todas as categorias listadas estão cadastradas no banco de dados, assim como seus respectivos objetos. Para o aplicativo listar as categorias na tela, o processo de recuperação das informações no banco de dados é semelhante ao do *login* e cadastro, conforme mostra a **Figura 6**. As informações retornadas do banco são armazenadas em uma estrutura no formato de texto conhecida como *JavaScript Object Notation* (JSON), onde a partir dessa estrutura de dados é formado a tela no aplicativo *Android*.



*Figura 6 - Representação gráfica da lista de categorias*

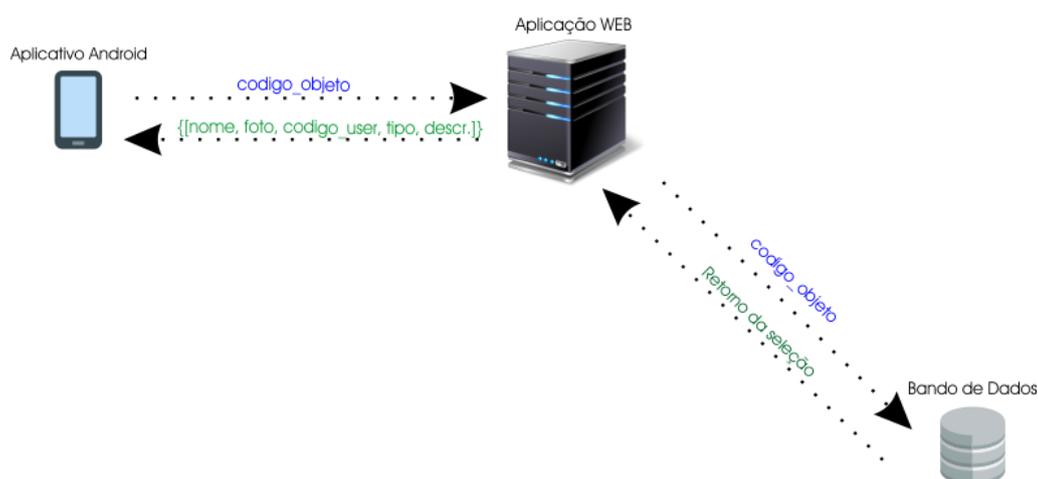
Quando o usuário seleciona uma das categorias uma nova tela é apresentada contendo uma lista com todos os seus objetos cadastrados. Ao clicar em uma categoria, o aplicativo *Android* envia para a aplicação *WEB* o código de identificação desta categoria, onde será realizada uma busca na base de dados e será retornado todas as informações dos objetos pertencentes à ela. A **Figura 7** ilustra este procedimento.



*Figura 7 - Representação gráfica da lista de objetos*

Quando o usuário seleciona um objeto de seu interesse uma nova tela é apresentada contendo todas as informações do objeto, como nome, foto, descrição, tipo de troca que ele pertence e por qual objeto ele pode ser trocado, caso seja do tipo de troca Restrita. Como em

todas as outras funcionalidades do sistema, a busca pelas informações do produto no banco de dados segue a mesma característica. É enviada a identificação do objeto selecionado para a aplicação *WEB*, que por sua vez busca os dados no banco, que retorna para a aplicação e que se comunica com o aplicativo *Android* novamente. A **Figura 8** ilustra esta funcionalidade.



*Figura 8 - Representação gráfica dos detalhes do objeto*

Quando um usuário deseja realizar uma solicitação de troca, ele deve clicar no botão Solicitar Troca. Ao clicar neste botão, o aplicativo *Android* envia à aplicação *WEB* as informações dos objetos que serão utilizados na troca e também as informações dos usuários envolvidos. A aplicação *WEB* então insere as informações na tabela de trocas e logo em seguida altera, na tabela produtos, os status dos objetos envolvidos para 'EM NEGOCIACAO'. Com isso a solicitação de troca foi concluída pelo usuário. Quando um objeto está com o status 'EM NEGOCIACAO' ele pode ser visualizado na lista de objetos, porém o aplicativo não permite que se faça solicitação de trocas para ele. A **Figura 9** ilustra a solicitação de troca.

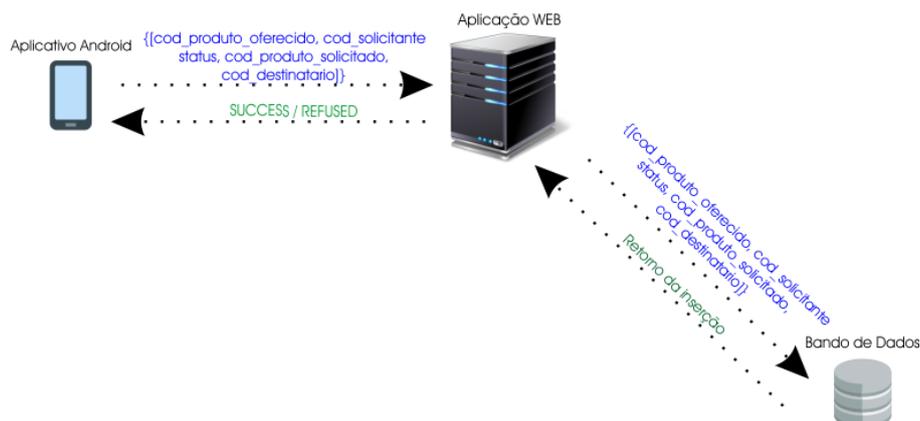


Figura 9 - Representação gráfica da Solicitação de Troca

Tanto o usuário que solicitou uma troca quanto o que recebeu podem cancelar uma solicitação. Quando uma solicitação de troca é cancelada o aplicativo *Android* envia para a aplicação *WEB* as informações referentes à troca. A aplicação *WEB* então atualiza na tabela de trocas o status desta troca para ‘CANCELADA’ e logo em seguida altera, na tabela produtos, os status dos objetos envolvidos para ‘DISPONIVEL’. Com isso estes objetos podem receber solicitações de trocas novamente. A **Figura 10** ilustra o cancelamento de uma solicitação de troca.

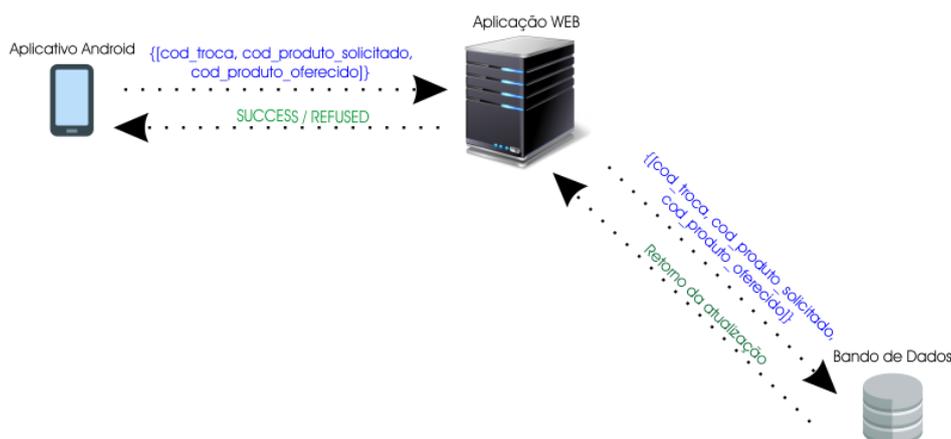
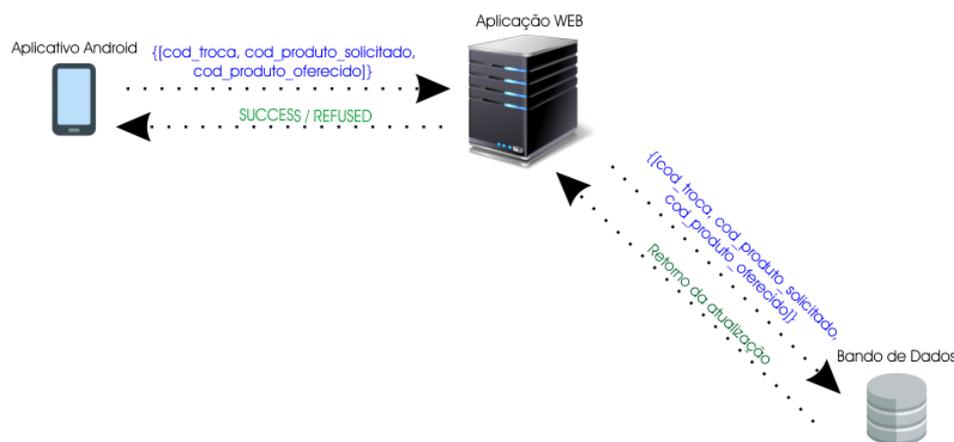


Figura 10 - Representação gráfica do cancelamento de troca

O usuário que recebeu uma solicitação de troca, além de poder recusá-la (cancelar), pode se interessar pelo produto oferecido e efetivar essa troca. Quando uma solicitação de troca é efetivada, significa que o usuário aceitou trocar seu objeto pelo que ofereceram. Com

isso o aplicativo *Android* envia para a aplicação *WEB* as informações referentes à troca, que então atualiza na tabela de trocas o status desta troca para 'FINALIZADA' e logo em seguida altera, na tabela produtos, os status dos objetos envolvidos para 'FINALIZADO'. Isso significa que estes objetos já foram trocados e não estarão mais disponíveis na lista de objetos. Uma troca que já foi finalizada não deve ser cancelada ou efetivada novamente. A **Figura 11** ilustra a efetivação de troca.



*Figura 11 - Representação gráfica da efetivação de troca*

Para todas as demais telas do aplicativo, o mesmo funcionamento descrito e ilustrado acima é aplicado, são elas: Meus Dados, Meus Produtos, Trocas Solicitadas e Trocas Recebidas. A tela Meus Dados permite o usuário atualizar suas informações pessoais: Nome, E-mail, Celular, *Login* e Senha. A tela Meus Produtos mostra uma lista com todos os objetos do usuário cadastrados. As telas de trocas solicitadas e recebidas mostra uma lista com todas as trocas envolvendo o usuário.

### 3.4 Principais classes

O aplicativo *Android* se conectará com a aplicação *WEB* por meio da Internet através do protocolo *HTTP*. A implementação de todo o sistema foi realizada na plataforma Windows.

#### 3.4.1 Módulo aplicativo

A seguir são listadas e apresentadas as principais classes e métodos que incorporam a implementação do aplicativo *Android*.

Tabela 2 - Principais classes do aplicativo Android

CLASSES	DESCRIÇÃO
SplashActivity	Classe inicial do aplicativo: mostra informações básicas durante alguns segundos.
MainActivity	Classe principal do aplicativo: o usuário pode fazer o login de acesso ou se cadastrar no sistema
CategoriasActivity	Classe responsável por listar as categorias disponíveis
ProdutosActivity	Classe responsável por listar os objetos de uma categoria
ProdutosDetActivity	Classe responsável por detalhar as informações de um objeto
CadastrarProdutoActivity	Classe responsável por cadastrar um objeto
MeusDadosActivity	Classe responsável por atualizar as informações pessoais do usuário
MeusProdutosActivity	Classe responsável por listar todos os produtos do usuário
SolicitarTrocaActivity	Classe responsável por realizar uma solicitação de troca de objetos
TrocasRecebidasActivity	Classe responsável por listar todas as solicitações de trocas recebidas
TrocasSolicitadasActivity	Classe responsável por listar todas as solicitações de trocas realizadas
<i>AndroidManifest</i>	Arquivo XML principal de configuração de todo o aplicativo <i>Android</i>
Categorias	Classe que auxilia a montagem da tela que lista as categorias cadastradas
Produtos	Classe que auxilia a montagem da tela que lista os objetos cadastrados
Image e WrapData	Classes que auxiliam a utilização de arquivos Bitmap no <i>Android</i>
Base64	Classe responsável em converter o arquivo Bitmap em uma codificação base64 para envio à aplicação servidor
HttpConnection	Classe responsável pela conexão e comunicação com a aplicação <i>WEB</i>

Conforme podemos ver na **Tabela 2**, existe algumas classes que possuem a palavra *Activity* junto ao seu nome. Isso significa que estas classes estendem características da classe *Activity* do *Android*.

A classe *Activity* representa uma tela da aplicação, sendo responsável por controlar o estado e os eventos da tela e definir qual *View* será responsável por desenhar a interface gráfica do usuário, por isso para cada tela do aplicativo é estendida a classe *Activity* em sua classe. No entanto, a classe *Activity* por si só não sabe desenhar nada na tela, para isso precisa da ajuda da classe *View*, que por sua vez se encarrega de desenhar os componentes visuais, como campos de textos, botões, imagens. Cada *Activity* deve ser obrigatoriamente declarada

no arquivo *AndroidManifest.xml*. O arquivo *AndroidManifest.xml* é a base de uma aplicação *Android* e contém todas as configurações necessárias para executar a aplicação (LECHETA, 2015).

Portanto, para que fique claro o entendimento, pode-se associar cada tela que se vê no aplicativo como sendo uma *Activity*, então se for necessário criar uma nova tela, deverá ser criada uma nova *Activity* e esta deverá ser declarada, obrigatoriamente, no arquivo *AndroidManifest.xml*.

As classes *Categorias* e *Produtos* são responsáveis apenas pela organização das listas de elementos do aplicativo, como a lista de categorias e a lista de objetos. As classes *Image*, *WrapData* e *Base64* auxiliam na manipulação de imagens *Bitmap* dentro do *Android*. Já a classe *HttpConnection* é responsável pela conexão via *HTTP* do aplicativo *Android* com a aplicação *WEB*.

### 3.4.2 Módulo aplicação PHP WEB

A seguir são listadas e apresentadas os principais arquivos e funcionalidades da aplicação PHP (PHP, 2015) *WEB*.

Tabela 3 - Principais arquivos da aplicação WEB

ARQUIVOS	DESCRIÇÃO
Login.php	Este arquivo recebe os dados de acesso do usuário, faz a verificação junto ao banco de dados e retorna ao aplicativo se os dados estão corretos ou incorretos.
Cadastrar.php	Recebe os dados de cadastro do aplicativo, faz a inserção no banco de dados e retorna ao aplicativo se a inserção foi realizada com sucesso.
Categorias.php	Busca no banco de dados todas as categorias cadastradas que possuem ao menos um objeto cadastro. Retorna essa lista para o aplicativo.
Produtos.php	Recebe do aplicativo o código de uma categoria, faz a busca de todos seus objetos cadastrados e retorna os objetos localizados para o aplicativo.
Dados.php	Recebe do aplicativo o login e senha de um usuário, busca no banco todas suas informações e retorna seu nome, e-mail, celular, login e senha para o aplicativo.
MeusDados.php	Recebe as novas informações do usuário e atualiza o banco de dados. Retorna se a atualização foi realizada com sucesso ou não.
CadastrarProduto.php	Este arquivo realiza o cadastro de um objeto no banco de dados. Além disso ele salva em uma pasta do servidor a imagem do objeto. Seu retorno é se o cadastro ocorreu com sucesso ou não.

List_categorias_cadastrar.php	Este arquivo busca no banco de dados todas as categorias cadastradas no banco e retorna-as para o aplicativo. Essas informações serão mostradas na tela de cadastrar novo objeto, onde o usuário deverá selecionar a qual categoria pertence.
Meus_produtos.php	Recebe o código do usuário e busca no banco todos os seus objetos cadastrados. Retorna as informações dos objetos cadastrados de um usuário para o aplicativo.
SolicitarTroca.php	Recebe o código do objeto oferecido, o código do usuário que solicitou, o código do objeto solicitado, o código do usuário que recebeu a solicitação e o status da troca. Insere na tabela de trocas essas informações recebidas. Em seguida atualiza o status dos dois produtos para 'EM NEGOCIACAO'.
CancelarTroca.php	Recebe o código da troca, o código do objeto solicitado e o código do objeto oferecido. Atualiza na tabela de trocas o status desta troca para 'CANCELADA'. Depois atualiza o status dos dois produtos para 'DISPONIVEL'.
EfetivarTroca.php	Recebe o código da troca, o código do produto solicitado e o código do produto oferecido. Atualiza na tabela de trocas o status desta troca para 'EFETIVADA'. Depois atualiza o status dos dois produtos para 'FINALIZADO'.
TrocasSolicitadas.php	Recebe o código do usuário e retorna uma lista com todas as trocas que ele solicitou.
TrocasRecebidas.php	Recebe o código do usuário e retorna uma lista com todas as trocas que ele recebeu.
Conexao.php	Arquivo de conexão com o banco de dados.

### 3.4.3 Módulo banco de dados

Conforme citado anteriormente, o SGBD escolhido foi o PostgreSQL (PSQL, 2015) em sua versão 9.4. Para o funcionamento do aplicativo Vai & Vem, foram criadas 4 tabelas no banco de dados: USUARIOS, PRODUTOS, CATEGORIAS e TROCAS. A **Figura 12** mostra o Diagrama Entidade Relacionamento (DER) do banco de dados do sistema. Nota-se que a relação USUARIOS – PRODUTOS é uma relação 1 para N, onde um usuário pode cadastrar vários objetos. A relação CATEGORIAS – PRODUTOS é uma relação de 1 para N, onde uma categoria poder possuir vários objetos cadastrados. A relação USUARIOS – TROCAS é uma relação 1 para N, onde um usuário pode possuir várias trocas. A relação PRODUTOS – TROCAS é uma relação 1 para N, pois cada produto poder estar em várias trocas, no caso das trocas terem sido canceladas.

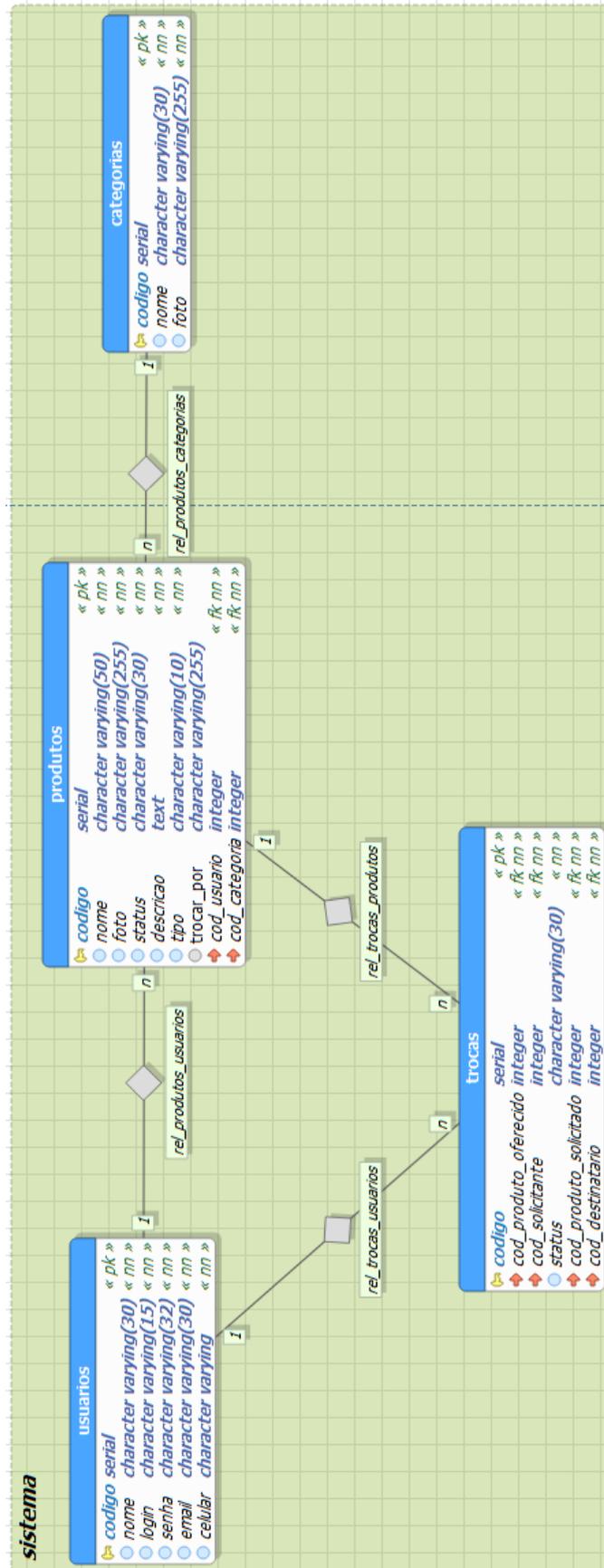


Figura 12 - Diagrama Entidade Relacionamento

A tabela USUARIOS armazenas as informações pessoais dos usuários como: NOME, EMAIL, CELULAR, LOGIN, SENHA, além do código que identifica cada registro desta tabela, CODIGO. O campo CODIGO é do tipo serial, ou seja, é um campo do tipo *integer* e *auto-increment* e é a chave primária desta tabela, onde ele será utilizado para referenciar os usuários nas outras tabelas.

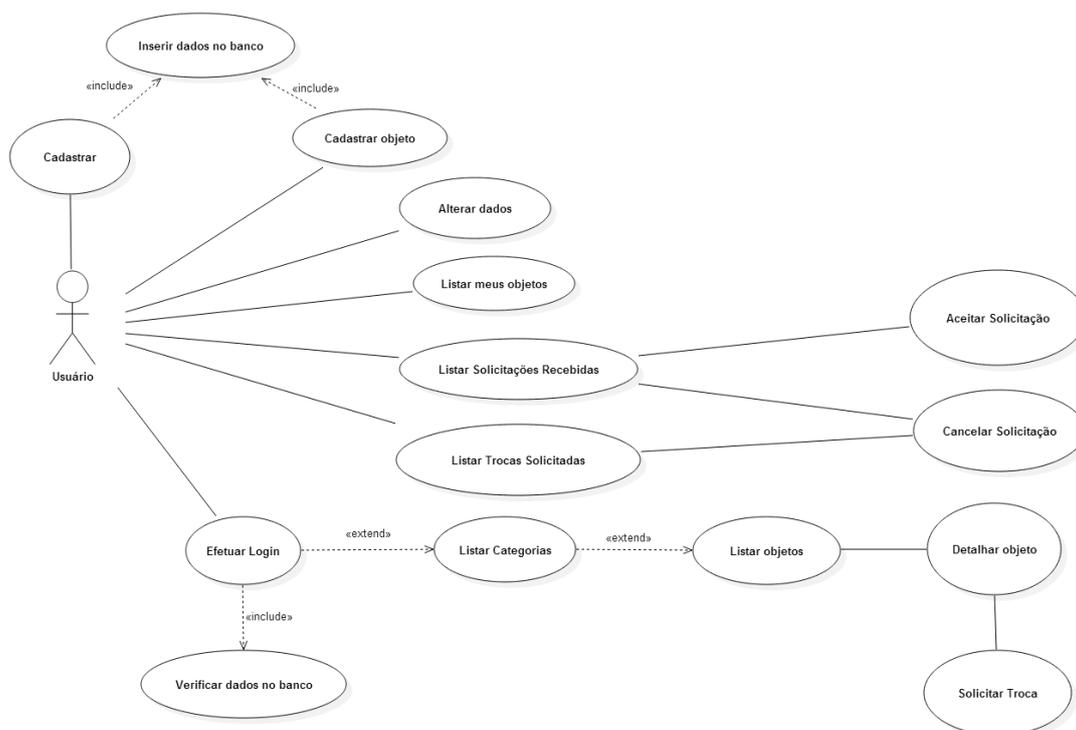
A tabela CATEGORIAS contém as informações das categorias cadastradas como: NOME, FOTO e o código de identificação de cada registro desta tabela, CODIGO. Assim como na tabela usuarios, o campo CODIGO é do tipo serial, ou seja, é um campo do tipo *integer* e *auto-increment* e é a chave primária desta tabela, onde ele será utilizado para referenciar as categorias na tabela de produtos.

A tabela PRODUTOS armazena as informações de cada objeto cadastrado como: NOME, FOTO, STATUS, DESCRICAO, TIPO, TROCAR\_POR, COD\_USUARIO, COD\_CATEGORIA e o código de identificação de cada registro na tabela, CODIGO. Assim como na tabela usuarios e na tabela produtos, o campo CODIGO é do tipo serial, ou seja, é um campo do tipo *integer* e *auto-increment* e é a chave primária desta tabela, onde ele será utilizado para referenciar os objetos na tabela de trocas. Os campos COD\_USUARIO e COD\_CATEGORIA são as chaves estrangeiras da tabela, que referenciam um usuário e uma categoria respectivamente, já que cada objeto pertence a uma categoria e a um usuário. O campo STATUS serve para indicar o status do objeto (“DISPONIVEL”, “EM NEGOCIACAO” ou “FINALIZADO”). O campo TIPO serve para indicar a que tipo de troca este objeto pertence (“ABERTA” ou “RESTRITA”). O campo TROCAR\_POR serve para indicar por qual objeto o usuário deseja trocar o seu objeto, no caso de a troca ser do tipo Restrita.

A tabela TROCAS armazena as informações referentes às solicitações de trocas realizadas no aplicativo: COD\_PRODUTO\_OFERECIDO, COD\_SOLICITANTE, STATUS, COD\_PRODUTO\_SOLICITADO, COD\_DESTINATARIO e o código de identificação de cada registro na tabela, CODIGO. Assim como nas outras tabelas, o campo CODIGO é do tipo serial, ou seja, é um campo do tipo *integer* e *auto-increment* e é a chave primária desta tabela. Esta tabela possui quatro chaves estrangeiras que serve para identificar os objetos e usuários envolvidos na troca. O campo STATUS serve para indicar qual o status da troca (“EM NEGOCIACAO”, “CANCELADA”, “FINALIZADA”).

### 3.5 Diagrama de caso de uso

A **Figura 13** representa o caso de uso do aplicativo. Para utilizar as funcionalidades, o usuário deverá estar cadastrado na base de dados. Estando cadastrado, basta realizar o login e o sistema já disponibilizará as telas do aplicativo.



*Figura 13 - Diagrama de Caso de Uso*

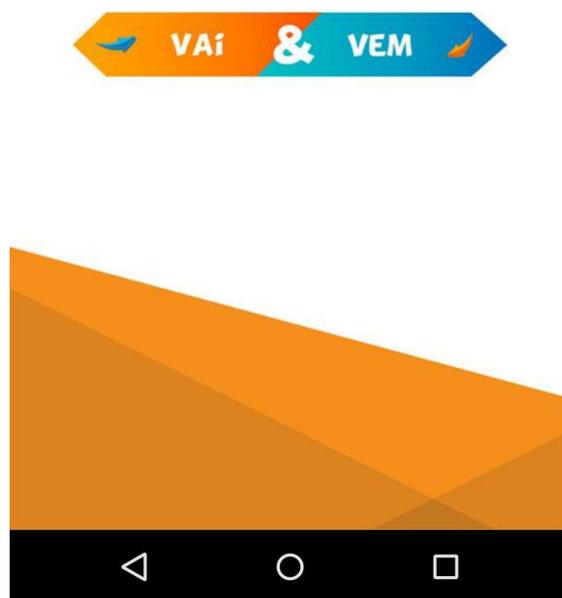


# Capítulo 4

## Interfaces do Aplicativo

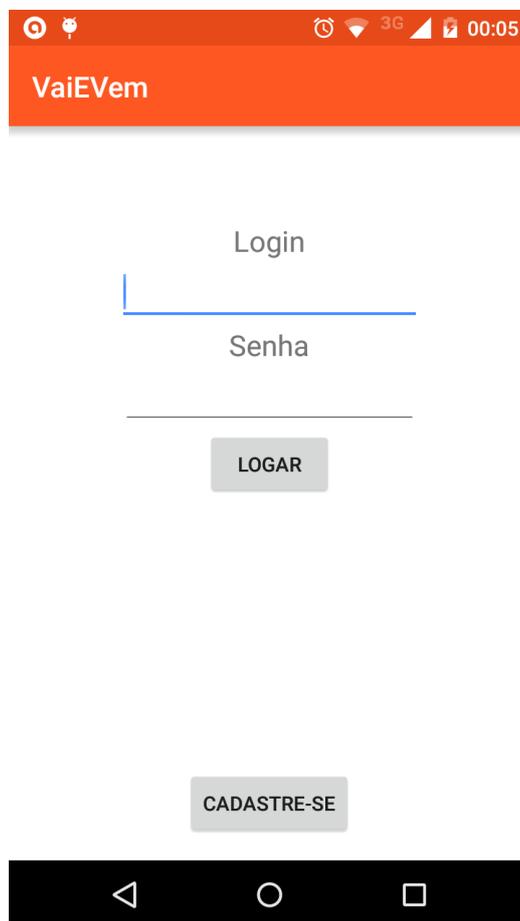
Este capítulo é dedicado a detalhar os testes do aplicativo realizados em ambiente de desenvolvimento. Eles visam determinar se o projeto atingiu suas especificações e se funcionou de maneira esperada. Nos testes foram utilizados tanto dispositivos simulados pelo emulador do *Android*, quanto dispositivos móveis reais. Para funcionamento da aplicação, inicialmente as categorias foram inseridas manualmente no banco de dados, para que ao cadastrar os produtos, elas fossem disponibilizadas na lista.

Ao iniciar o aplicativo é apresentada sua tela *Splash*, conforme mostra a **Figura 14**.



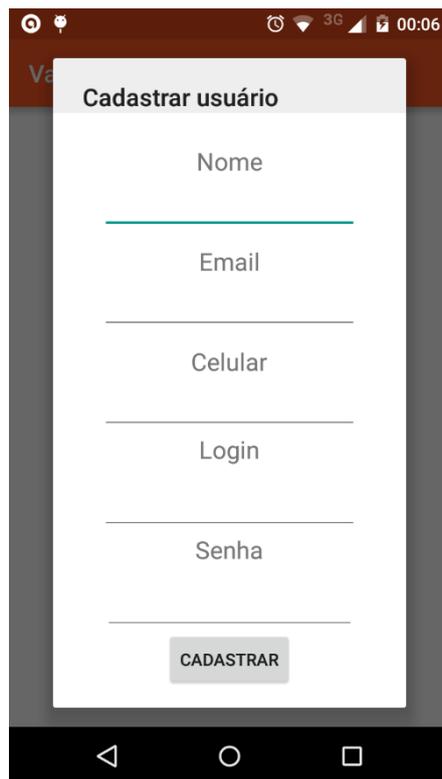
*Figura 14 - Tela Splash do aplicativo Vai & Vem*

Após mostrar a tela *Splash* o aplicativo abre a tela de *Login*, conforme mostra a **Figura 15**. Nesta tela usuário tem a opção de realizar o acesso informando seus dados de acesso como *login* e senha, ou caso ainda não seja cadastrado ele tem a possibilidade de cadastrar-se.



*Figura 15 - Tela de Login do aplicativo*

Para realizar um novo cadastro, basta o usuário clicar no botão CADASTRE-SE da tela de *Login*, conforme mostrado na Figura anterior. Ao clicar neste botão, uma nova janela aparecerá na tela, solicitando que o usuário insira as informações pessoais. A **Figura 16** mostra essa tela. Nota-se pela Figura, que o usuário para efetivar seu cadastro deve informar os seguintes campos: nome, *e-mail*, celular, login e senha.



*Figura 16 - Tela de Cadastro de Usuário*

Ao efetuar o acesso ao sistema, tanto pelo login como pelo cadastro, o aplicativo abre a tela que lista as categorias cadastradas, conforme mostra a **Figura 17**.



*Figura 17 - Tela que lista as Categorias que possui ao menos um objeto cadastrado*

Nota-se pela Figura anterior que o aplicativo já traz o nome do usuário que fez acesso ao sistema, uma forma de identificar quem está utilizando o aplicativo.

Ao selecionar uma categoria, abre-se uma nova tela contendo uma lista com todos os objetos desta categoria, conforme mostra a **Figura 18**.



*Figura 18 - Lista de objetos da Categoria Eletrônicos*

Ao selecionar um objeto, uma nova tela é mostrada contendo todas as informações deste objeto selecionado e um botão onde o usuário pode solicitar a troca por este objeto. A **Figura 19** mostra esta tela.



Figura 19 - Detalhes do objeto e botão Solicitar Troca

Para solicitar uma troca basta clicar no botão SOLICITAR TROCA. Uma nova tela será apresentada contendo uma lista com todos os objetos do usuário que poderão ser oferecidos e o nome do usuário dono do objeto. A **Figura 20** mostra esta tela.

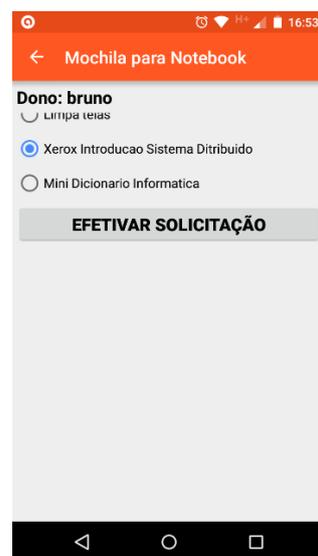
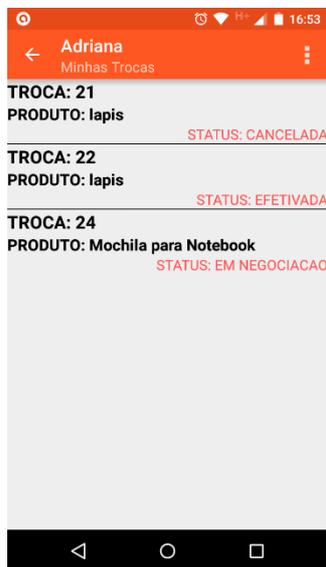


Figura 20 - Tela de Efetivar Solicitação

Para efetivar uma solicitação basta selecionar um objeto que irá oferecer e clicar no botão EFETIVAR SOLICITAÇÃO. Com isso o sistema irá salvar a troca no banco e alterar os status dos produtos para “EM NEGOCIACAO”. A **Figura 21** mostra a tela depois de efetivar a solicitação.



*Figura 21 - Tela com todas as trocas solicitadas pelo usuário*

A Figura acima mostra que o usuário já possui 3 solicitações de trocas cadastradas, onde uma foi cancelada, outra efetivada e esta última está em negociação. Como o objeto Mochila para Notebook está em uma negociação, seu status foi alterado, conforme mostra a **Figura 22**.



*Figura 22 - Status do produto em negociação*

Caso algum usuário tente solicitar a Mochila para Notebook para uma troca, o aplicativo informa que ele já está em negociação e não permite solicitar enquanto estiver em negociação. O botão que servia para Solicitar Troca, agora mostra que o objeto está Em Negociação. A **Figura 23** mostra esta tela.



*Figura 23 - Objeto em negociação*

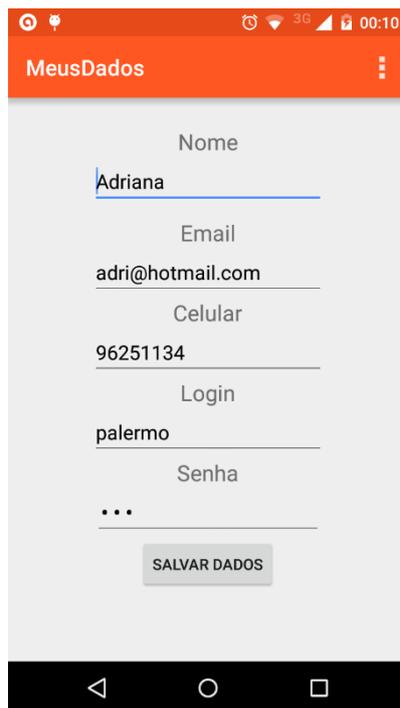
Em todas as páginas, na barra superior, no lado direito existe um menu de opções

identificado por uma imagem com 3 pontinhos. Este menu contém algumas funcionalidades que o usuário pode selecionar como: Meus Dados, Cadastrar Produto, Meus Produtos, Trocas Recebidas e Trocas Solicitadas. A **Figura 24** mostra este menu de opções.



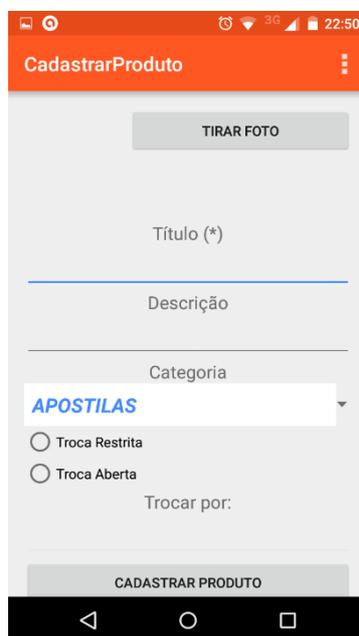
*Figura 24 - Lista de funcionalidades presentes no menu de opções*

Ao selecionar o item Meus Dados uma nova tela é aberta contendo todas as informações pessoais do usuário. Ali o usuário pode atualizar suas informações e salvar na base de dados. A **Figura 25** mostra esta tela.



*Figura 25 - Tela Meus Dados*

Ao selecionar a opção Cadastrar Produto, é apresentada uma tela onde o usuário poderá cadastrar um novo objeto. A **Figura 26** mostra esta tela.



*Figura 26 - Tela de Cadastro de Produto*

Ao selecionar a opção Meus Produtos, uma nova tela é apresentada contendo a lista de todos os objetos cadastrados pelo usuário. A **Figura 27** apresenta esta tela.



*Figura 27 - Tela Meus Produtos*

Ao selecionar a opção Trocas Recebidas é apresentada uma lista com todas as solicitações de trocas que o usuário recebeu. A **Figura 28** apresenta esta tela.



*Figura 28 - Trocas Recebidas*

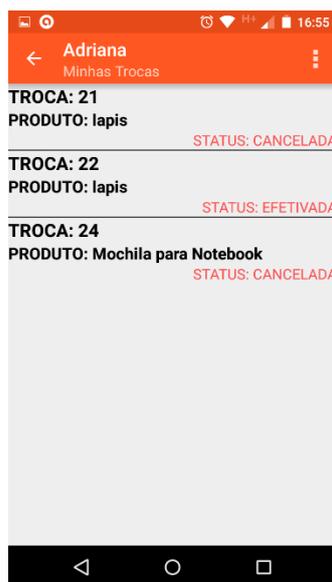
O usuário que recebeu uma solicitação tem a opção de recusá-la (cancelá-la) ou aceitá-la. Para isso basta clicar em cima de uma troca e escolher uma opção. A **Figura 29** mostra a

tela.



*Figura 29 - Aceitar ou cancelar uma solicitação de troca*

Se o usuário decidir recusar esta solicitação ele deverá clicar no botão CANCELAR TROCA. Com isso o status da troca será alterado para CANCELADA e os status dos produtos envolvidos voltará para DISPONIVEL. A **Figura 30** mostra a lista com a troca e seu status alterado.



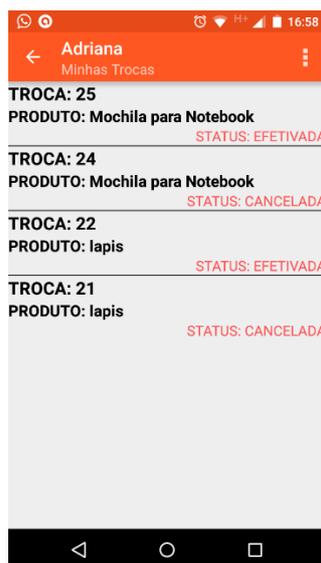
*Figura 30 - Status da troca CANCELADA*

Quando uma troca é CANCELADA, seus produtos envolvidos voltam a ficar disponíveis para novas trocas. A **Figura 31** mostra o produto disponível novamente.



*Figura 31 - Objeto disponível novamente*

Quando uma solicitação de troca é aceita, o status da troca é alterado para FINALIZADA e os status dos produtos são alterados para FINALIZADO. Com isso eles já nem aparecem mais na lista. A **Figura 32** mostra o status de uma troca efetivada para o objeto Mochila para Notebook.



*Figura 32 - Troca efetivada*

Como a troca foi finalizada, o objeto Mochila para *Notebook* não deve mais aparecer na lista de objetos disponíveis. A **Figura 33** mostra esta tela.



*Figura 33 - Objeto saiu da lista depois de ter sido trocado*



# Capítulo 5

## Conclusão

Cada vez mais as pessoas tendem a se desapegar de seus bens materiais, devido à rápida substituição que o passar do tempo provoca. No mundo universitário essa substituição fica ainda mais evidente, visto que um aluno, com o término de um período letivo, provavelmente não necessitará mais daquele livro que ele tanto usou, daqueles xérox que os professores passavam semanalmente ou daquele caderno contendo toda a matéria estudada. A probabilidade de o aluno “esquecer” estes objetos ou até mesmo jogá-los fora é muito grande, sendo que eles poderiam ser muito bem reaproveitados por outros alunos nos próximos períodos letivos. Foi justamente pensando nisso, de reaproveitar inúmeros objetos úteis e que possam ser reutilizados por outras pessoas na universidade, que este projeto foi proposto e desenvolvido.

Pôde-se verificar, através de todo o estudo e pesquisa realizados neste trabalho, o quanto a área de desenvolvimento de aplicativos móveis e da tecnologia móvel tem se expandido de uma maneira muito grande. O aplicativo *Android Vai & Vem* desenvolvido e apresentado neste trabalho trouxe uma possibilidade de as pessoas, principalmente os alunos de nossa universidade, de destinar aqueles objetos que não são mais úteis, às pessoas que possam usufruí-los de uma maneira melhor. Com isso, acabamos reaproveitando materiais que poderiam ser jogados fora e contribuímos com a natureza, já que menos fabricação ocorrerá e menos poluição atingirá o meio ambiente.

O Aplicativo proposto e desenvolvido para o ambiente acadêmico pode ser utilizado também em outros ambientes, tais como: empresarial, redes sociais, familiares e grupo de amigos.

### 5.1 Trabalhos futuros

Apresento aqui algumas possíveis funcionalidades e características que poderiam ser implementadas e acrescentadas no aplicativo, visando sua melhoria e facilidade.

- Acesso via redes sociais como *Facebook, Twitter, LinkedIn*;

- Sistema de mensagens dentro do aplicativo para facilitar a comunicação entre os usuários;
- Possibilitar o cadastro de mais de uma foto para cada objeto;
- Criar prazos para que um objeto fique disponível para trocas;
- Permitir o gerenciamento de categorias através do próprio aplicativo, através de uma senha de administrador.

# Referências Bibliográficas

A-AND. A história do *Android*. Disponível em: <<https://www.oficinadanet.com.br/post/13939-a-historia-do-Android>>, 2015. Acessado em: Abril de 2015.

AND-MER-MUN. *Android* e iOS detêm 96,3% do mercado mundial de smartphones. Disponível em: <<http://www.tudocelular.com/Android/noticias/n50237/Android-ios-96-3-mercado-mundial-smartphones.html>>, 2015. Acessado em: Junho 2015.

AND-STUDIO. Ambiente de Desenvolvimento *Android* Studio. Disponível em: <<http://developer.Android.com/intl/pt-br/sdk/index.html>>, 2015. Acessado em: Abril 2015.

APACHE. Disponível em: <<https://httpd.apache.org/>>, 2015. Acessado em: Abril 2015.

DES-AI. DescolaAi. Disponível em: <<http://www.descolaii.com/page/sobre-nos>>, 2015. Acessado em: Abril de 2015.

DES-AND. Desenvolvendo para *Android*: Arquitetura *Android*. Disponível em: <<http://www.tiselvagem.com.br/geral/developendo-para-Android-arquitetura-Android/>>, 2011. Acessado em: Junho de 2015.

ECLIPSE. Disponível em: <<https://eclipse.org/>>, 2015. Acessado em: Abril 2015.

FIGUEIREDO, M. S.; NAKAMURA, E. Computação Móvel: Novas Oportunidades e Novos Desafios. T&C Amazônia, 2003.

GARGENTA, M.: *Learning Android*. O'Reilly Media, 2011.

INFO-ES. InfoEscola Escambo. Disponível em: <<http://www.infoescola.com/economia/escambo/>>, 2011. Acessado em: Maio de 2015.

JSON. Introdução ao JSON. Disponível em: <<http://json.org/json-pt.html>>, 2015. Acessado em: Agosto 2015.

LECHETA, R. R. *GOOGLE ANDROID: Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. 4ª edição, 2015.

MONTEIRO, J. Google *Android* – Crie aplicações para celulares e tablets. 1º Edição, 2013.

MOTTIN, L.R. Troca-Troca de Roupas, UFPR- PR, 2014.

MYSQL. Disponível em: <<https://www.mysql.com/>>, 2015. Acessado em: Abril 2015.

OLIVEIRA, F. Protótipo de Aplicativo de força de vendas para dispositivos móveis baseados na plataforma *Android*, Blumenau-SC, 2014.

OHA. OPEN HANDSET ALLIANCE. Disponível em: <[http://www.openhandsetalliance.com/oha\\_members.html](http://www.openhandsetalliance.com/oha_members.html)>, 2015. Acessado em: Julho de 2015.

PAULA, L. J. D. Desenvolvimento de aplicativo para dispositivos móveis para coleta de dados georreferenciados através de reconhecimento de voz. Piracicaba – SP, 2013.

PER-DIG. Permuta Digital. Disponível em: <<http://www.permutadigital.com/pmt/conheca-permuta-digital>>, 2013. Acessado em: Maio de 2015.

PERL. Disponível em: <<https://www.perl.org/>>, 2015. Acessado em: Abril 2015.

PGMODEL. Disponível em: <<http://pgmodeler.com.br/>>, 2015. Acessado em: Abril 2015.

PHP. Disponível em: <<https://secure.php.net/>>, 2015. Acessado em: Abril 2015.

PSQL. Disponível em: <<http://www.postgresql.org/download/windows/>>, 2015. Acessado em: Abril 2015.

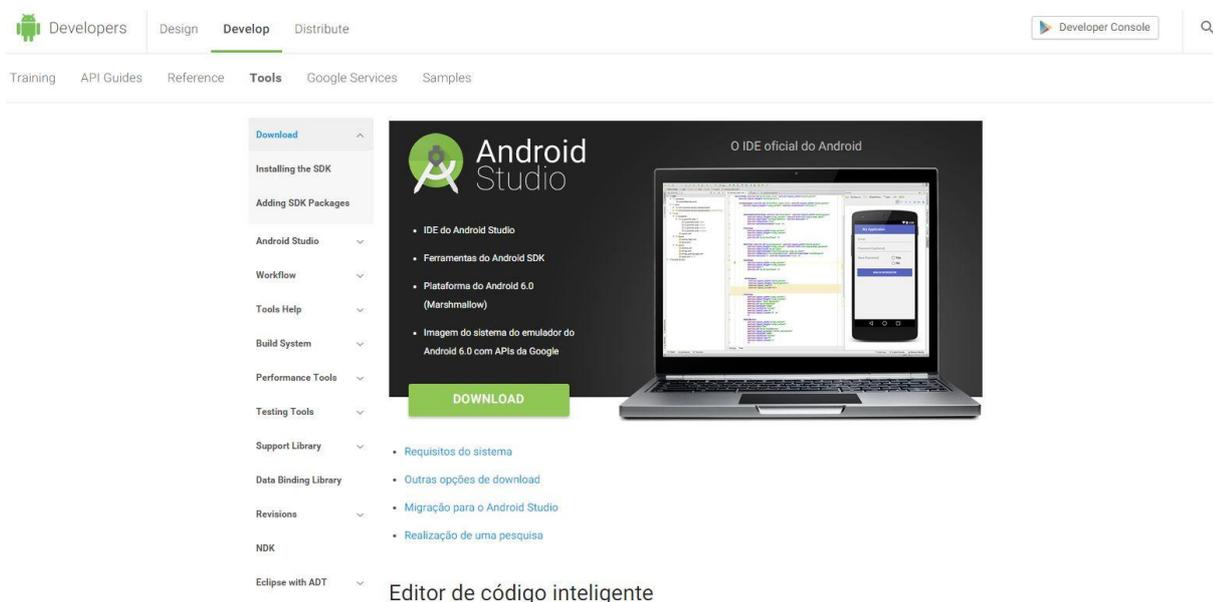
SIMÃO, A. M. D. L. Proposta de Método para Análise Pericial em Smartphone com Sistema Operacional *Android*. Disponível em: <[http://repositorio.unb.br/bitstream/10482/9938/1/2011\\_AndreMorumLimaSimao.pdf](http://repositorio.unb.br/bitstream/10482/9938/1/2011_AndreMorumLimaSimao.pdf)>, 2011. Acessado em: Maio de 2015.

XAMPP. Disponível em: <[https://www.apachefriends.org/pt\\_br/download.html](https://www.apachefriends.org/pt_br/download.html)>, 2015. Acessado em: Abril 2015.

# Apêndice A

## Instalação e configuração do *Android Studio*

O *Android Studio* deve ser baixado no site oficial, no navegador de internet vá para: <http://developer.Android.com/intl/pt-br/sdk/index.html> e clique no link para download. A **Figura 34** mostra a página principal do site da IDE.



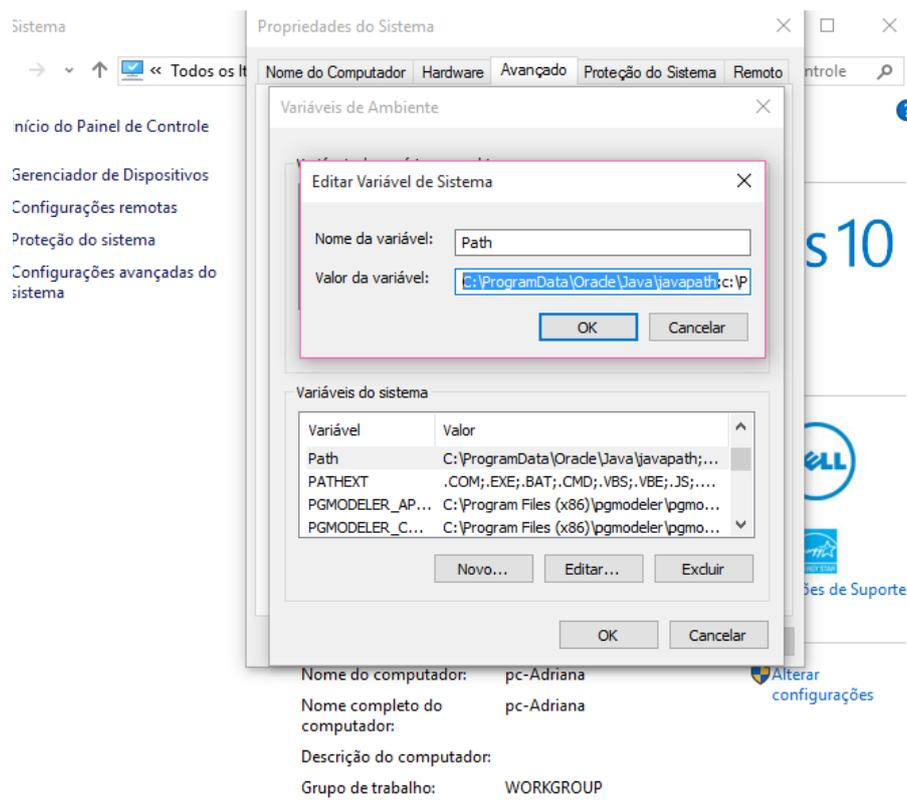
*Figura 34 - Instalação e configuração do Android Studio*

Passo 1: Caso não tenha já pré-instalado em sua máquina o JAVA JDK, faça o download apropriado (versão 32 ou 64 bits) para seu sistema no link a seguir: <http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk7-downloads-1880260.html>. A instalação segue padrão de instalação ( next, next, finish). Preste atenção ao caminho da instalação para configurar o JAVA\_HOME posteriormente. A **Figura 35** mostra o progresso da instalação.



*Figura 35 - Instalando o Java*

Passo 2: Configure o *JAVA HOME*. Abra o Painel de Controle > Sistemas > Configurações do Sistema > Avançado > Variáveis de Ambiente. Na seção “Variáveis do sistema” selecione o item “Path” e clique em “Editar”. Na janela que abriu “Editar variável de Sistema” clique no campo “Valor da variável”, vá até o final do texto e informe o caminho da instalação do seu JDK feito no passo 1 e clique em OK. A **Figura 36** mostra a janela que configura o *JAVA HOME*.



*Figura 36 - Configurar o JAVA HOME*

Passo 3: Finalizada a instalação do JDK e configuração do Java\_Home faça a instalação do *Android Studio* baixado. A instalação do *Android* segue o padrão (*Next, Next, Finish*). Após a instalação e execução do programa é apresentada a tela de boas-vindas. A **Figura 37** mostra esta tela.

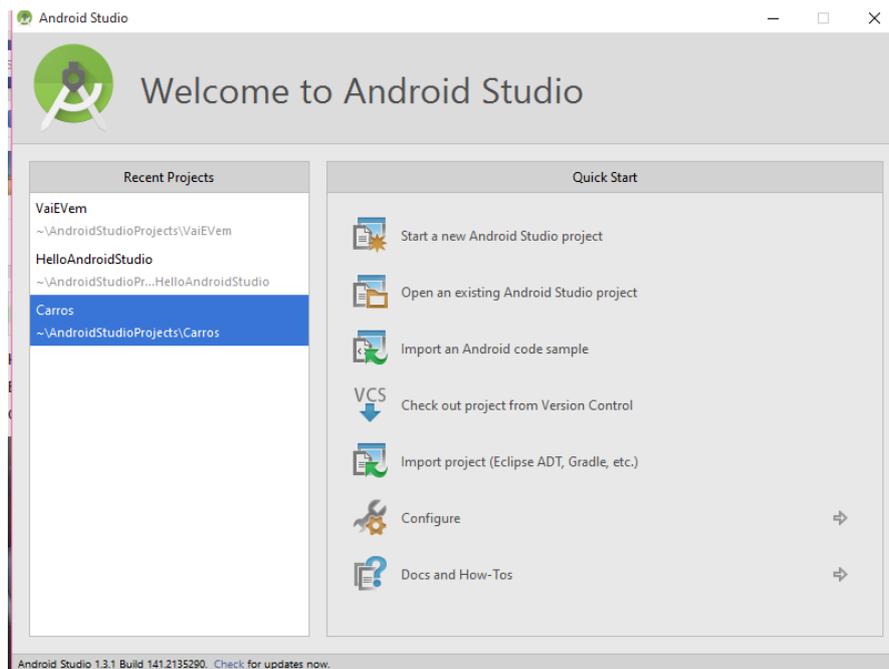


Figura 37 - Telas de boas-vindas do Android Studio

Passo 4: O *Android SDK Manager* é o responsável pelo gerenciamento dos pacotes e bibliotecas nativas da plataforma *Android*. Vamos utilizá-lo para instalar as versões e suas atualizações. Clique em *Configure > SDK Manager*.

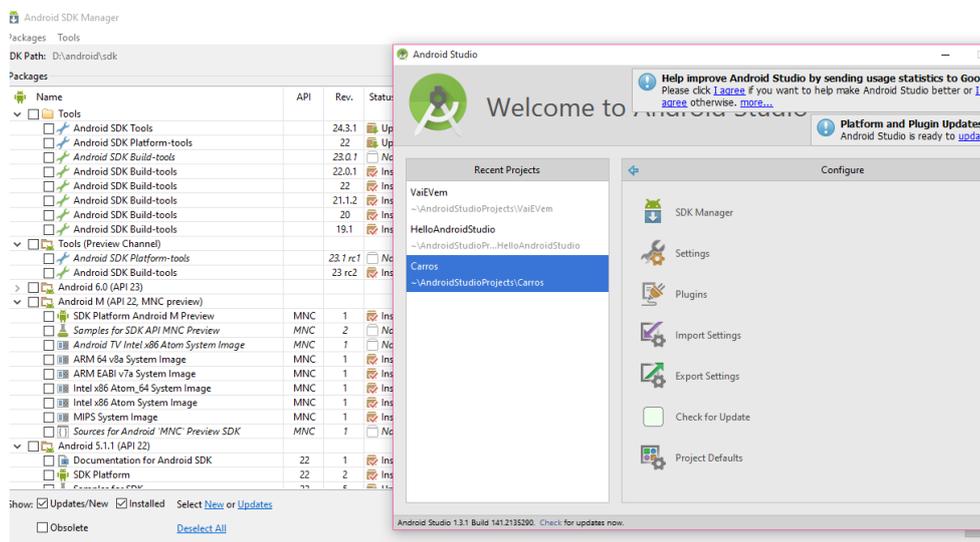


Figura 38 - SDK Manager

Passo 5: Com *Android Studio* aberto, faça todas as atualizações que a ferramenta pedir, clicando no botão *Install packages*. É necessário sempre aceitar os termos de uso no momento da instalação. Faça esse processo novamente até que todas as atualizações

indicadas sejam instaladas. Após seguir esses passos, você já pode usar o Ambiente de desenvolvimento *Android Studio*.



# Apêndice B

## Instalação e configuração do XAMPP

O XAMPP deve ser baixado no site oficial, no navegador de internet vá para: <https://www.apachefriends.org/download.html> e clique no link para download. A **Figura 39** mostra a página de *Download* do XAMPP.



*Figura 39 - Instalação e configuração do XAMPP*

Passo 1: Para iniciar o Download do instalador, basta clicar na opção de plataforma que você utiliza e o navegador pergunta se realmente deseja baixar este arquivo. A **Figura 40** mostra a janela de confirmação do *Download*.

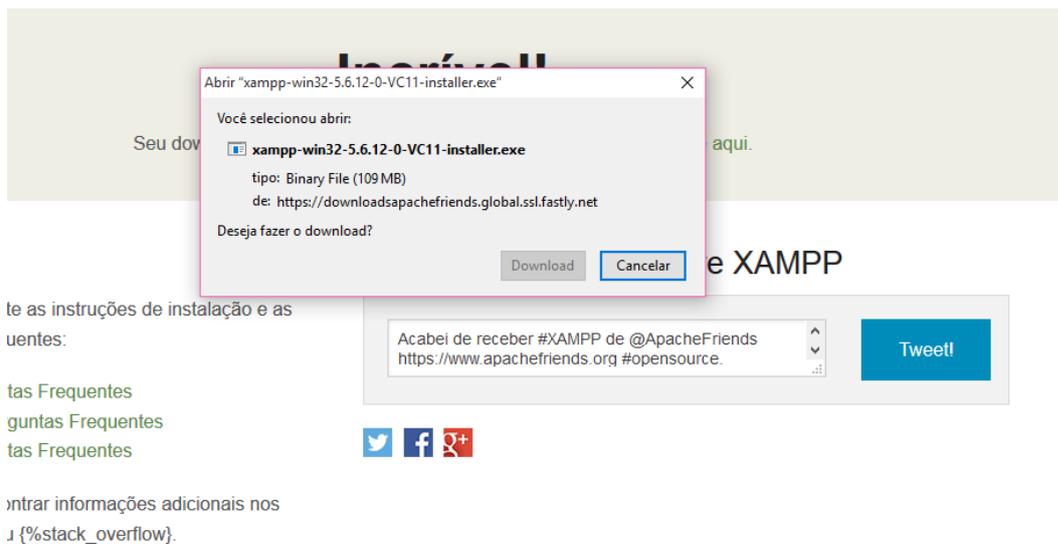


Figura 40 - Download do XAMPP

Passo 2: Após finalização do *Download* do instalador, execute e escolha o idioma de instalação. Siga clicando em *Next*. A **Figura 41** mostra este passo.

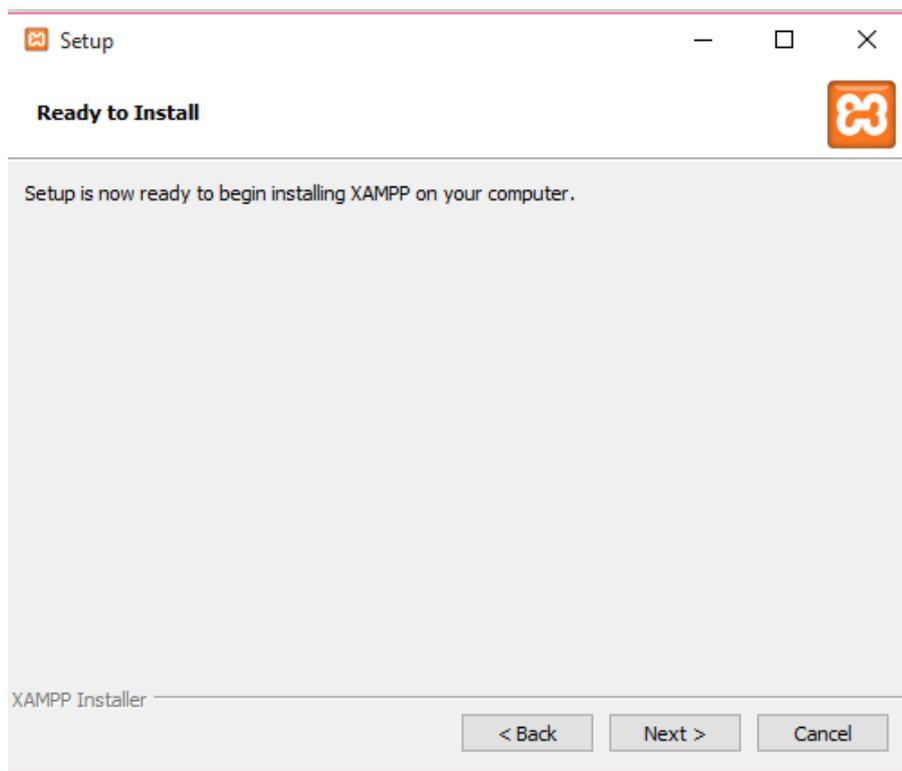
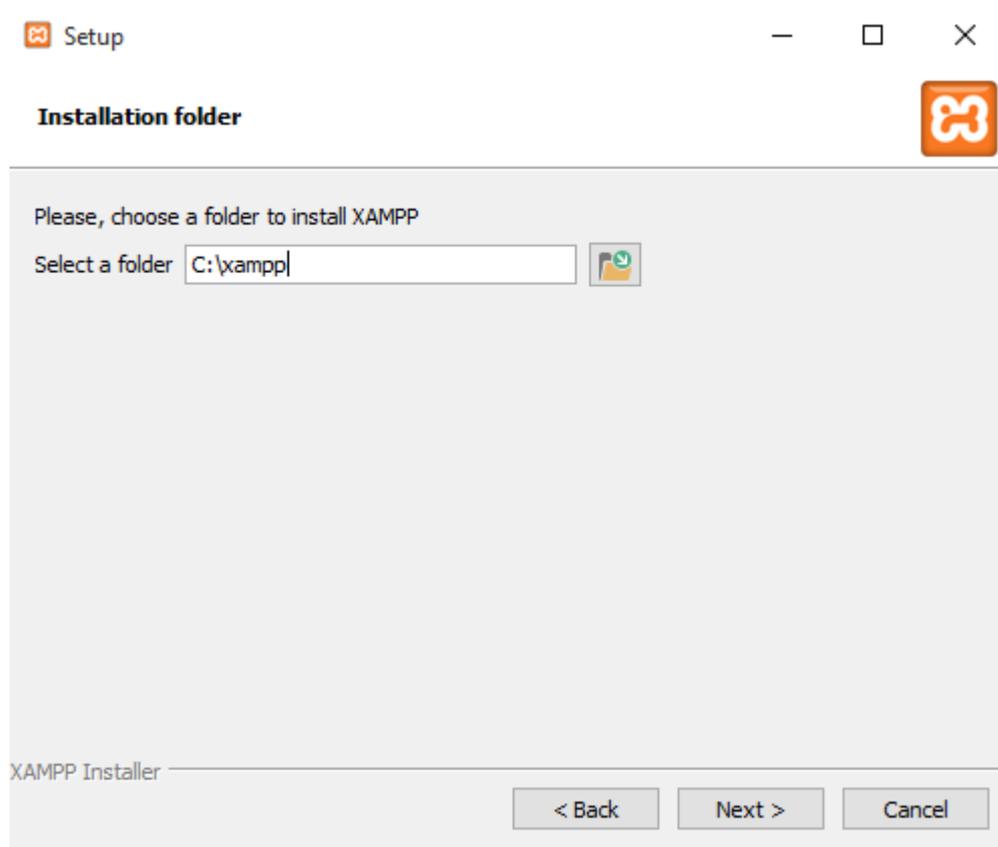
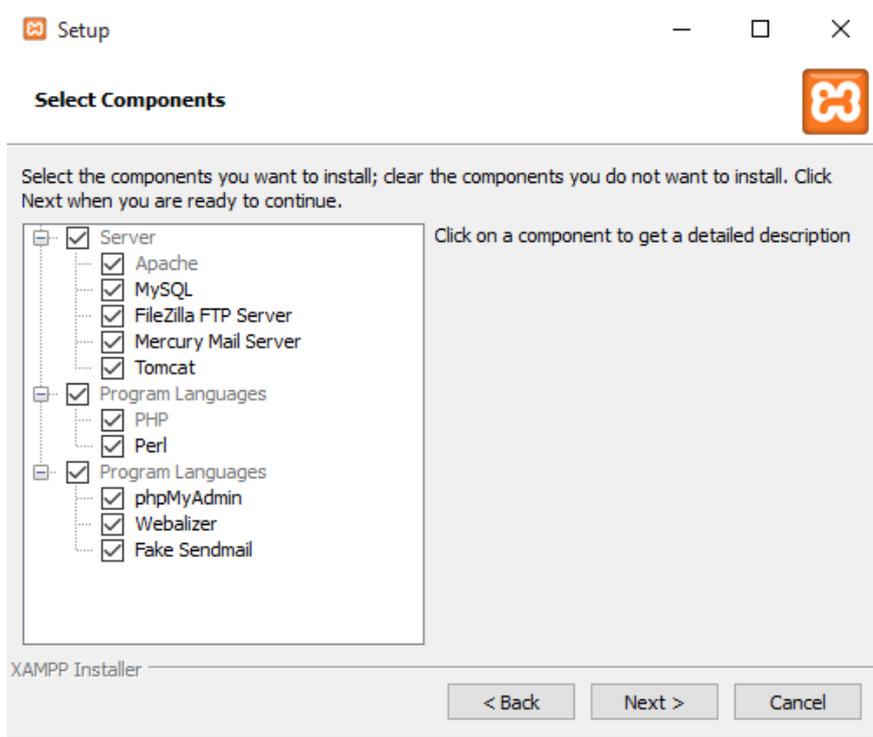


Figura 41 - Instalação do XAMPP

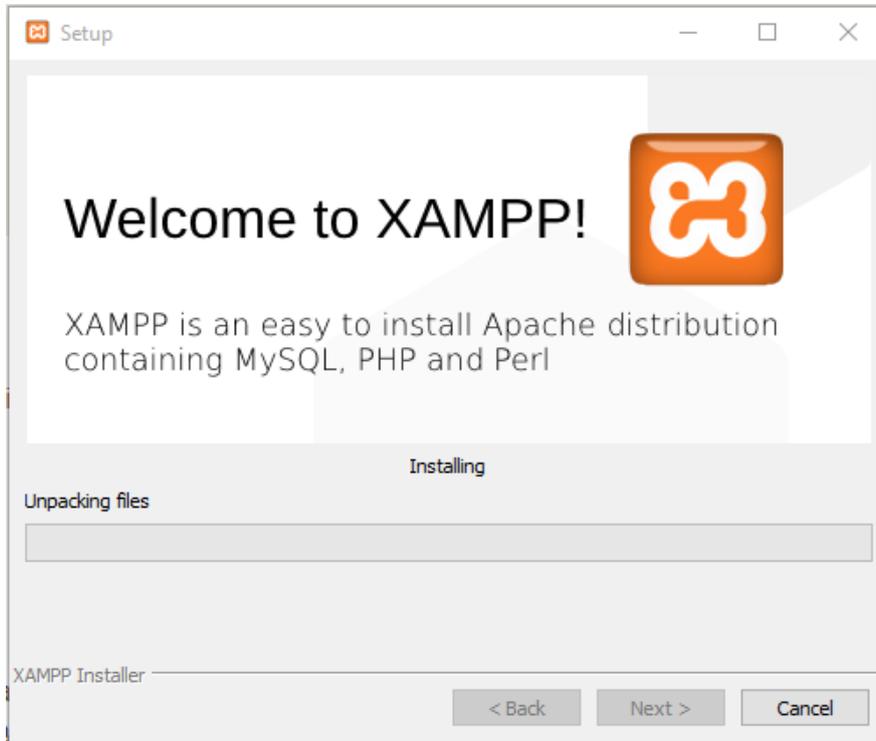
Passo 3: Escolha o diretório onde o XAMPP será instalado e clique em *Next*. Logo após, selecione os componentes que deseja instalar e clique em *Next* novamente. As **Figuras 42 e 43** mostram estes passos.



*Figura 42 - Defina o diretório onde o XAMPP será instalado*

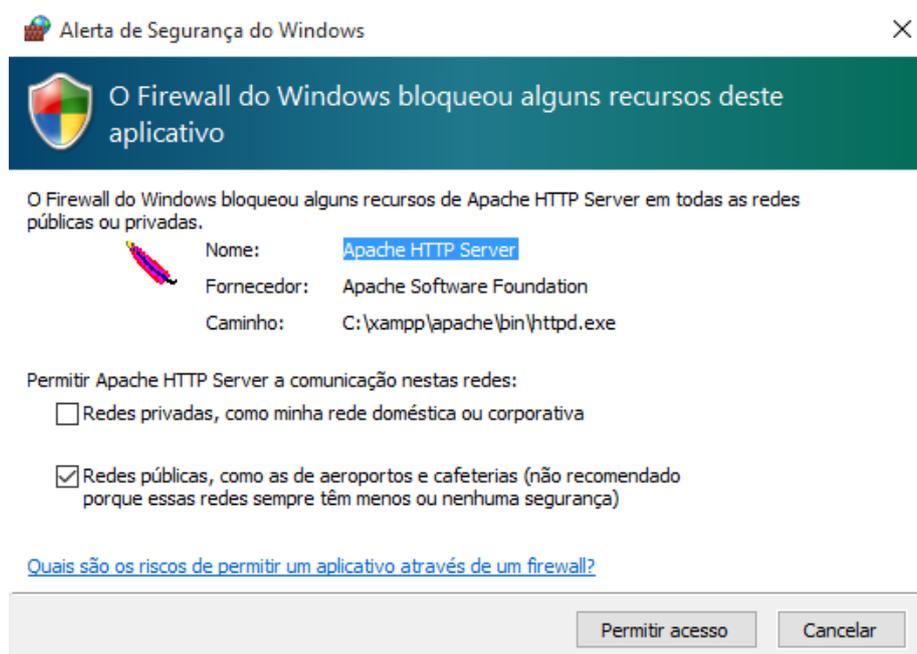


*Figura 43 - Seleção dos arquivos a serem instalados*

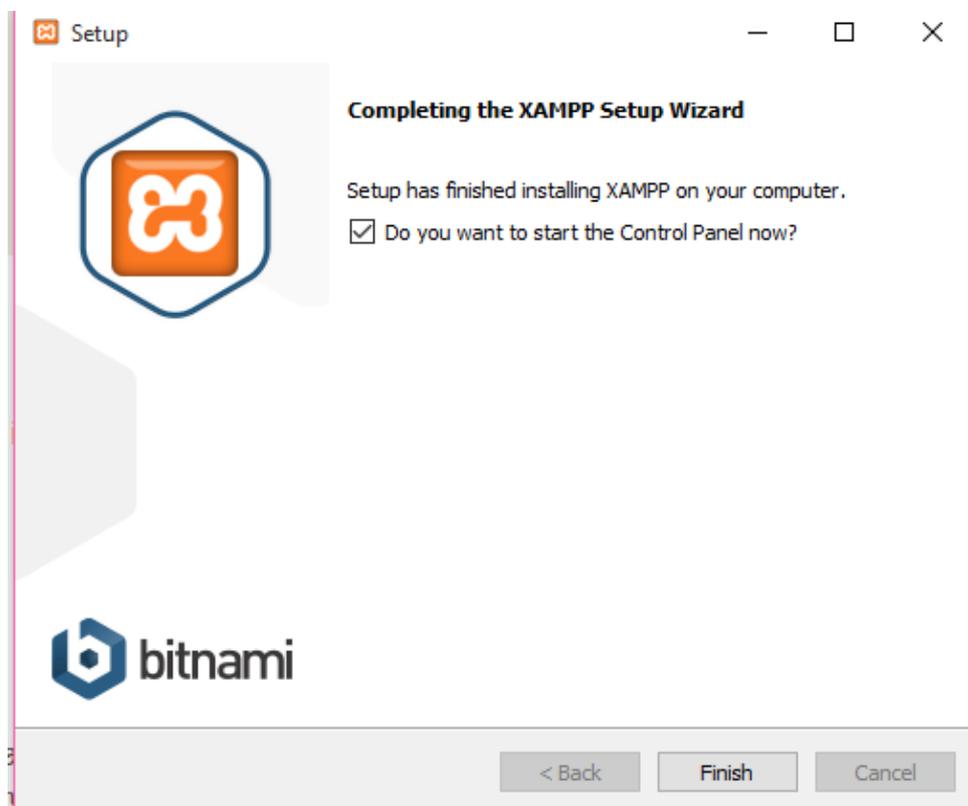


*Figura 44 - Progresso da instalação do XAMPP*

Passo 4: Aceite o alerta de permissão do *Windows* e finalize a instalação clicando em *Finish*. A **Figura 45** mostra este alerta de segurança.

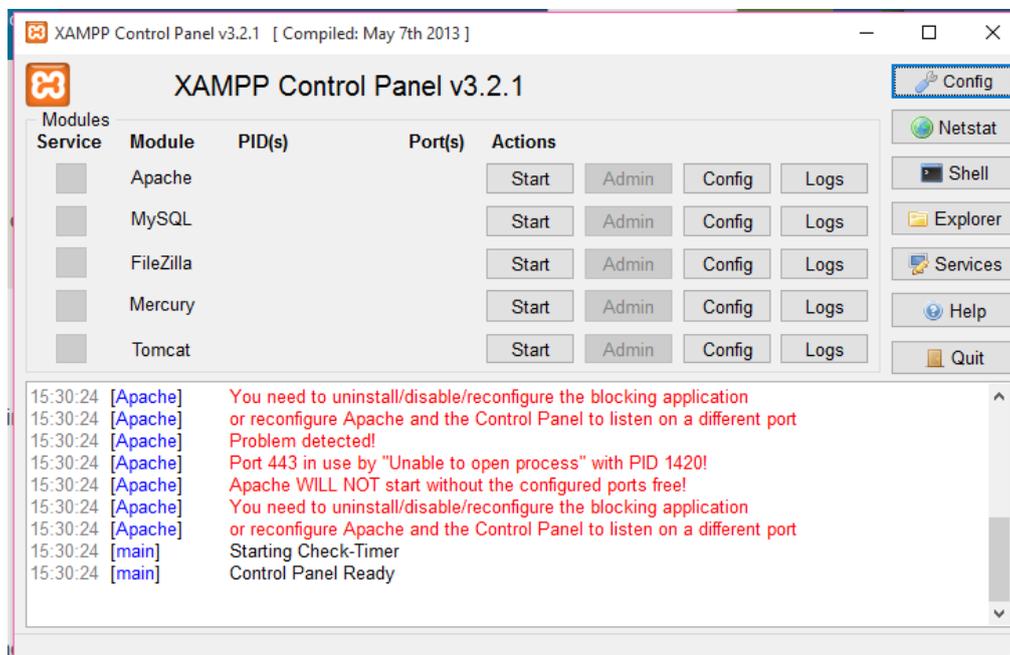


*Figura 45 - Alerta de segurança do Windows*



*Figura 46 - Finalização da instalação do XAMPP*

Passo 5: Feita a instalação, inicie o painel de controle do XAMPP e clique em “Start” para iniciar os serviços Apache. Para verificar a instalação do Apache clique no “Admin” do Apache no Painel de Controle. A **Figura 47** mostra o painel de controle do XAMPP.



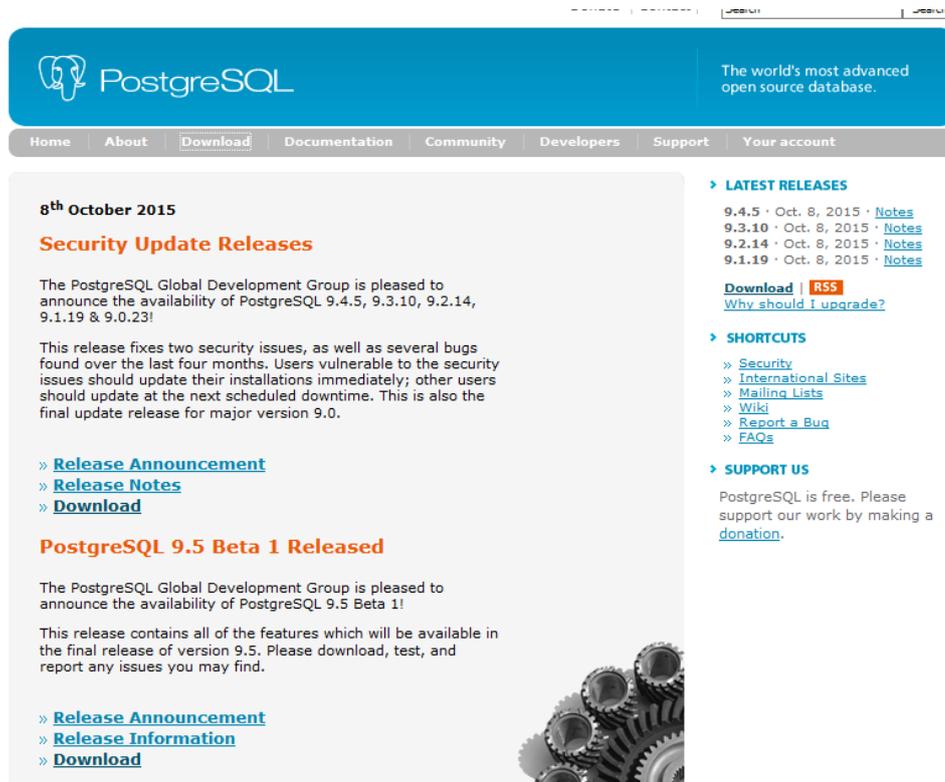
*Figura 47 - Painel de Controle do XAMPP*

Passo 6: Se as etapas forem bem-sucedidas, o XAMPP será instalado com sucesso em seu computador. Abra o navegador e digite “http://localhost” na sua barra de endereços. Você será direcionado para uma página dizendo que instalou corretamente o XAMPP em seu sistema.

# Apêndice C

## Instalação e configuração do PostgreSQL

O PostgreSQL deve ser baixado no site oficial, no navegador de internet vá para: <http://www.postgresql.org/download/windows/> e clique no link para download. A **Figura 48** mostra a página inicial do site do PostgreSQL.



*Figura 48 - Instalação e configuração do PostgreSQL*

Passo 1: Escolha a opção *Download* e você será redirecionado para uma tela onde deverá escolher o seu Sistema Operacional, a versão de bits e também a versão do PostgreSQL que deseja utilizar. A **Figura 49** mostra esta tela.

# Download PostgreSQL

Please Note: Cookies should be enabled for the download process to function correctly

Installer version Version 9.5.0 Beta 1



Installer version Version 9.4.5 [Readme file for customers interested in using PL/Perl, PL/Python or PL/Tcl]



Installer version Version 9.3.10



Installer version Version 9.2.14



Figura 49 - Página de Download do PostgreSQL

Passo 2: Feito isso, o download irá começar. Dê um duplo clique no arquivo baixado e aguarde começar a instalação. Após, você será direcionado a tela de boas-vindas. Clique em *Next* e siga. A **Figura 50** mostra este passo.

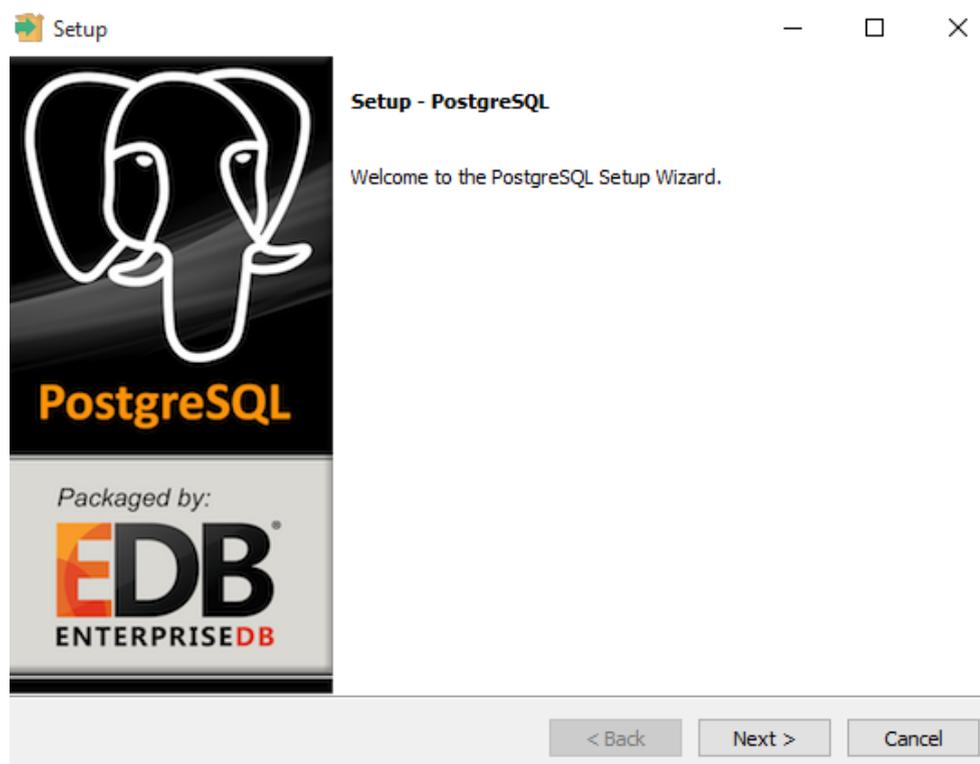


Figura 50 - Instalação do PostgreSQL

Passo 3: Escolha o diretório da instalação e siga clicando em Next.

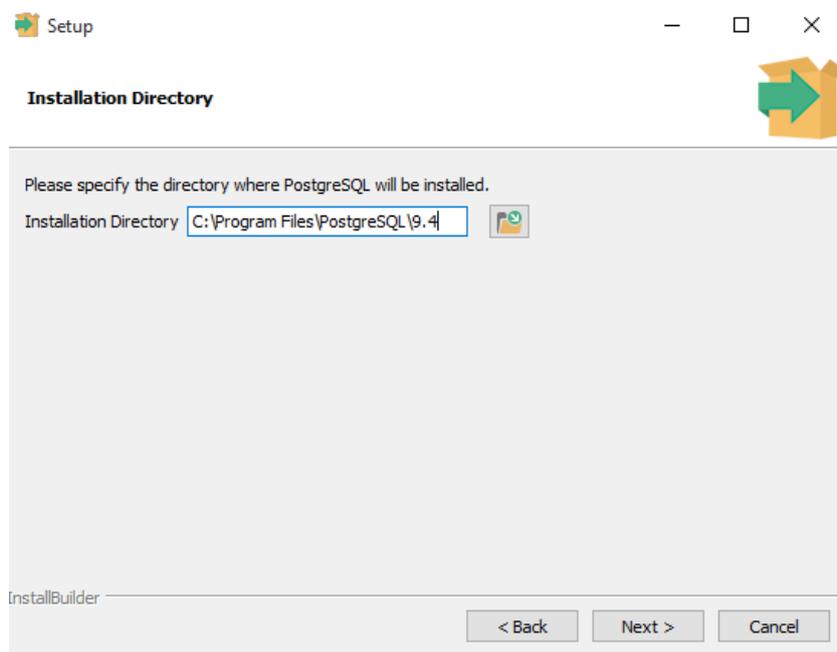


Figura 51 - Defina o local onde o PostgreSQL será instalado

Passo 4: Insira um *password* para seu Banco e clique em *Next*. Insira o número da porta que seu banco vai ficar, ou deixe a que está como *default*, só mude se a porta já estiver em uso, clique em Next. As **Figuras 52 e 53** mostram estes passos.

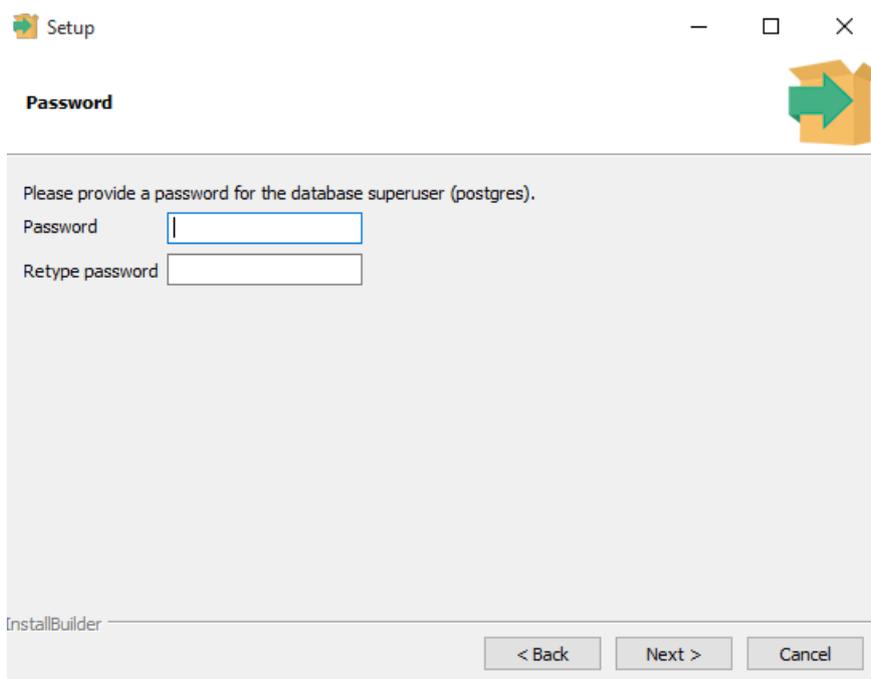
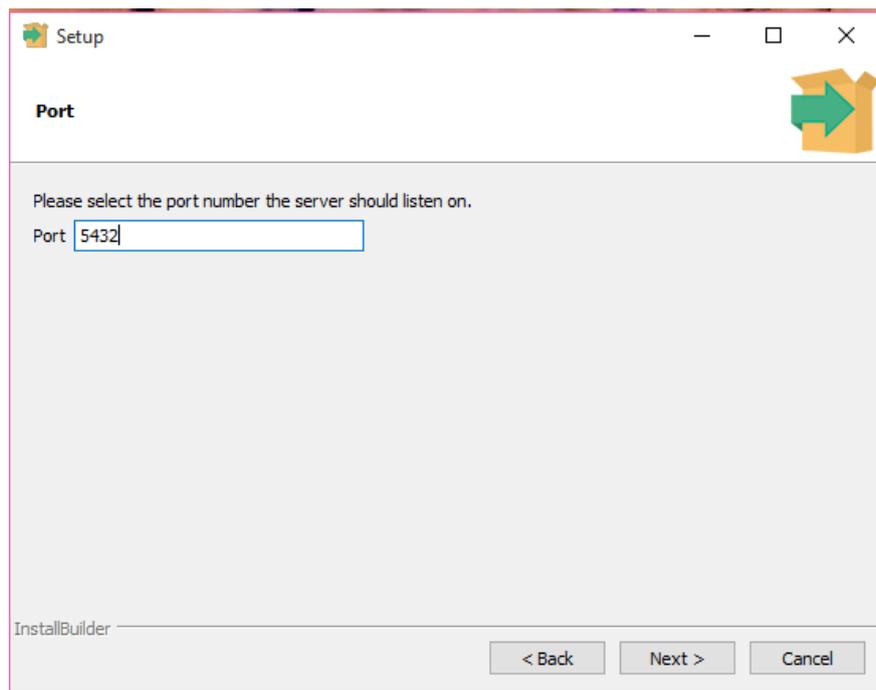
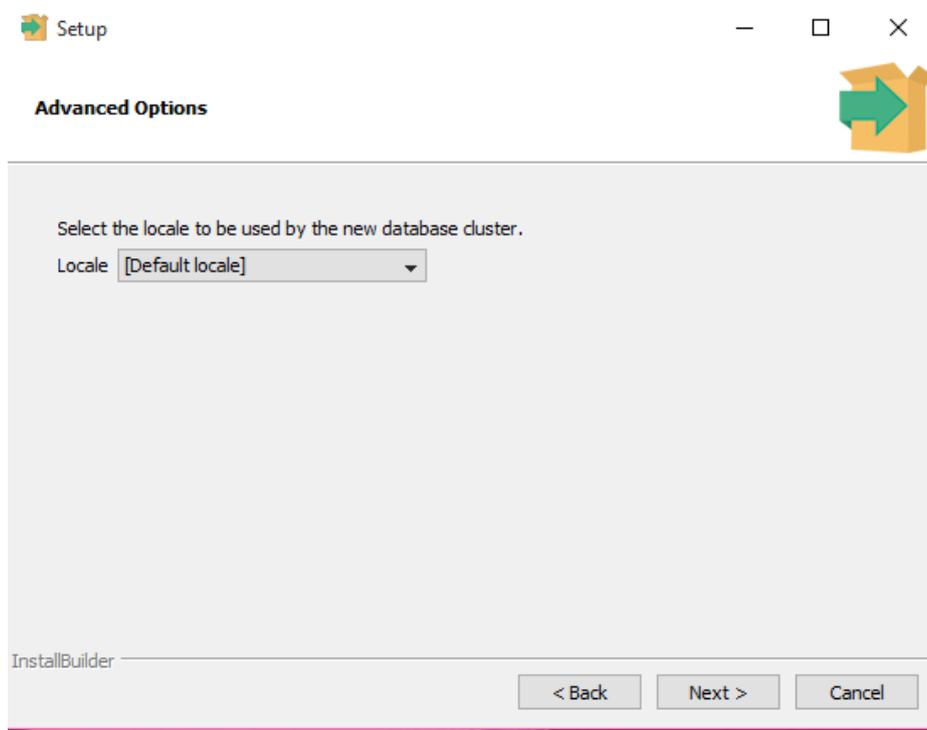


Figura 52 - Cadastre uma senha para seu servidor PostgreSQL



*Figura 53 - Definir porta*

Passo 5: Defina o *locale* como *portuguese Brazil*, clique em Next e em seguida em Next novamente. A **Figura 54** mostra esta opção.



*Figura 54 - Deixe a opção padrão selecionada*

Passo 6: Aguarde a instalação finalizar. A **Figura 55** mostra o progresso da instalação.

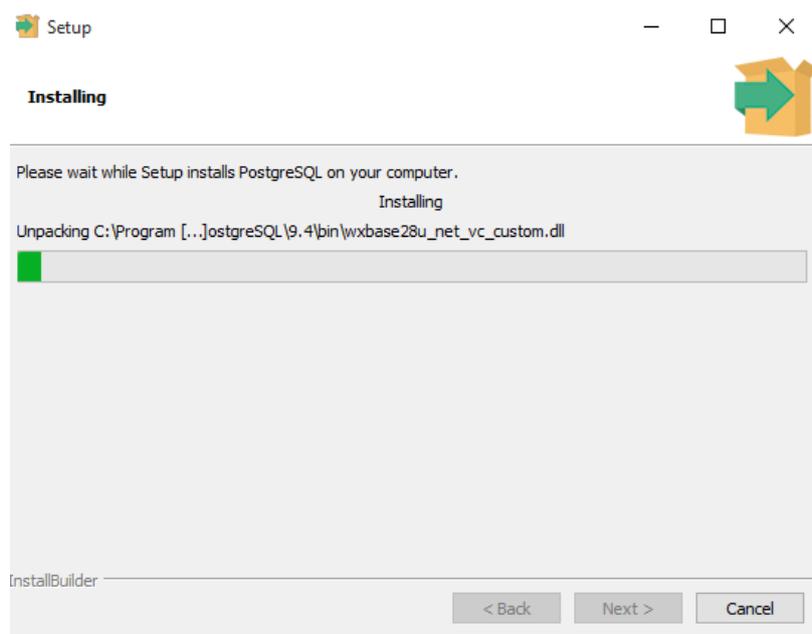


Figura 55 - Progresso da instalação

A Figura 56 mostra a tela de finalização da instalação do PostgreSQL.

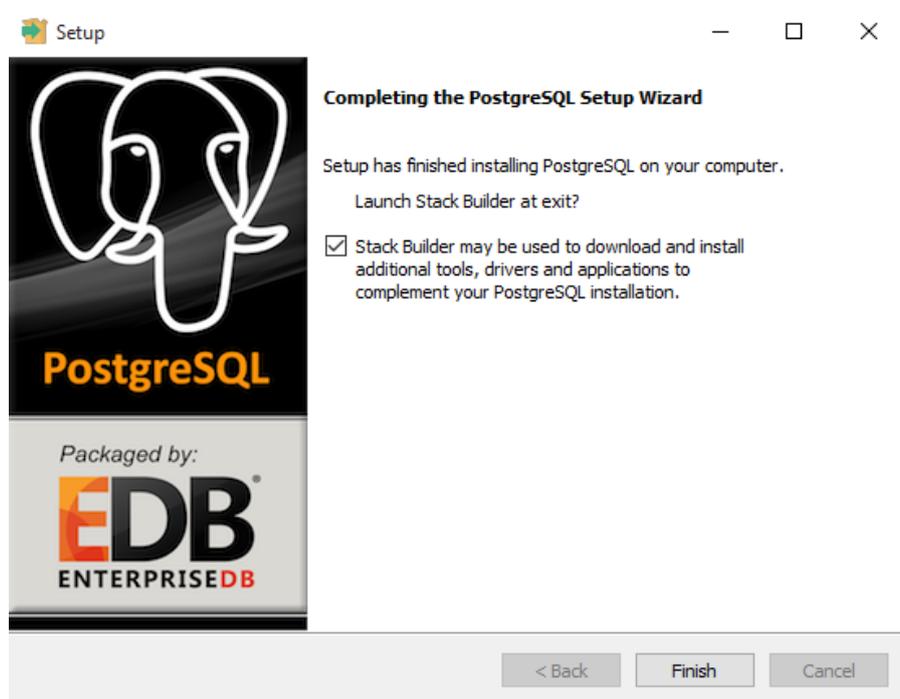


Figura 56 - Finalizar instalação