



UNIVERSIDADE FEDERAL DE MATO GROSSO  
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM  
SISTEMAS DE INFORMAÇÃO

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO  
APLICAÇÕES MOBILE COM LAYOUT ADAPTÁVEIS  
UTILIZANDO ANDROID FRAGMENTS**

**JUNIOR DA COSTA NASCIMENTO**

CUIABÁ – MT

2015

UNIVERSIDADE FEDERAL DE MATO GROSSO  
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM  
SISTEMAS DE INFORMAÇÃO

**RELÁTÓRIO DE ESTÁGIO SUPERVISIONADO  
APLICAÇÕES MOBILE COM LAYOUT ADAPTÁVEIS  
UTILIZANDO ANDROID FRAGMENTS**

**JUNIOR DA COSTA NASCIMENTO**

Relatório apresentado Instituto de  
Computação da Universidade Federal de  
Mato Grosso, para obtenção do título de  
Bacharel em Sistemas de Informação.

CUIABÁ – MT

2015

UNIVERSIDADE FEDERAL DE MATO GROSSO  
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM  
SISTEMAS DE INFORMAÇÃO

JUNIOR DA COSTA NASCIMENTO

Relatório de Estágio Supervisionado apresentado à Coordenação do Curso de Sistemas de Informação como uma das exigências para obtenção do título de Bacharel em Sistemas de Informação da Universidade Federal de Mato Grosso

Aprovado por:

---

Prof. MSc. Nilton Hideki Takagi  
Instituto de Computação  
(Coordenador de Estágios)

---

Prof. MSc. Thiago Meirelles Ventura  
Instituto de Computação  
(CONVIDADO)

---

Prof. Dr. Raphael de Souza Rosa Gomes.  
Instituto de Computação  
(ORIENTADOR)

---

Grad. João Luiz Oliveira Ribeiro  
(SUPERVISOR)

## **DEDICATÓRIA**

*Dedico este trabalho primeiramente a Deus por ter me dado sabedoria e me guiado  
durante esta caminhada.*

*A toda minha família pelo apoio e confiança que sempre depositaram em mim.*

*A todos os professores, pela importância que cada um tem durante o decorrer do  
curso.*

## **AGRADECIMENTOS**

A toda minha família, que sempre me apoiou e orientou para que pudesse tomar as melhores decisões.

A todos os professores, pelo conhecimento e experiências transmitidos durante o curso, ao meu orientador Prof<sup>o</sup> Raphael de Souza Rosa Gomes pela orientação, apoio e confiança.

A todos os colaboradores da WorkDev Tecnologia de contribuíram direta e indiretamente para realização deste trabalho, ao meu supervisor João Luiz Oliveira Ribeiro pela atenção e orientação em todas as etapas do desenvolvimento do trabalho.

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>7</b>
<b>LISTA DE TABELAS.....</b>	<b>8</b>
<b>LISTA DE SIGLAS E ABREVIATURAS .....</b>	<b>9</b>
<b>RESUMO .....</b>	<b>10</b>
<b>1. INTRODUÇÃO .....</b>	<b>11</b>
<b>2. REVISÃO DE LITERATURA .....</b>	<b>12</b>
2.1 INTRODUÇÃO AO ANDROID .....	12
2.1.1 <i>Linux Kernel</i> .....	14
2.1.2 <i>Libraries</i> .....	14
2.1.3 <i>AndroidRuntime</i> .....	14
2.1.4 <i>Application Framework</i> .....	14
2.1.5 <i>Componentes da Aplicação</i> .....	14
2.1.6 <i>Plataforma (Versão do Android)</i> .....	15
2.2 DESENVOLVIMENTO ÁGIL DE SOFTWARE.....	16
2.2.1 <i>Scrum</i> .....	16
<b>3. MATERIAS, TÉCNICAS E MÉTODOS .....</b>	<b>18</b>
3.1 FERRAMENTAS DE DESENVOLVIMENTO.....	18
3.1.1 <i>Eclipse</i> .....	18
3.1.2 <i>Android virtual Device Manager (AVD)</i> .....	19
3.1.3 <i>Jaybird</i> .....	19
3.2 MODELAGEM DA ARQUITETURA .....	21
3.3 ANDROID FRAGMENT .....	22
3.3.1 <i>Fragment</i> .....	22
3.3.2 <i>Ciclo de Vida Fragment</i> .....	23
<b>4. RESULTADOS .....</b>	<b>26</b>
<b>5. DIFICULDADES ENCONTRADAS .....</b>	<b>35</b>
<b>6. CONCLUSÕES.....</b>	<b>36</b>
<b>7. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>37</b>
<b>ANEXO I – LAYOUT EM TABLETS .....</b>	<b>39</b>

## LISTA DE FIGURAS

FIGURA 1- T-MOBILE G1. (ANDROID DEVELOPER, 2012).....	13
FIGURA 2- ARQUITETURA DO ANDROID (SOURCE ANDROID, 2015).....	13
FIGURA 3 - CICLO DO SCRUM (DESENVOLVIMENTO ÁGIL, 2013).....	17
FIGURA 4 - AMBIENTE DE DESENVOLVIMENTO ECLIPSE.....	18
FIGURA 5 - ANDROID VIRTUAL DEVICE AVD.....	19
FIGURA 6 - ARQUITETURA JAYBIRD (FIREBIRD, 2004).....	20
FIGURA 7 - ARQUITETURA DO SISTEMA.....	22
FIGURA 8 - DESIGN ANDROID FRAGMENT (ANDROID DEVELOPER, 2012).....	23
FIGURA 9 - CICLO DE VIDA FRAGMENT. (ANDROID DEVELOPER, 2012).....	24
FIGURA 10 - TRECHO DE CÓDIGO DA CLASSE ACTIVITY APLICATIVO.....	26
FIGURA 11 - CONFIGURAÇÃO.....	27
FIGURA 12 - LOGIN.....	28
FIGURA 13 – EMPRESA.....	28
FIGURA 14 – MENU.....	29
FIGURA 15 - ACERTO DE ESTOQUE.....	30
FIGURA 16 - INVENTARIO.....	30
FIGURA 17 - NÚMERO DE SÉRIE.....	31
FIGURA 18 – LOTE.....	31
FIGURA 19 - FRAGMENT – PRODUTO PADRAO.....	32
FIGURA 20 - FRAGMENT – PRODUTO INF.....	32
FIGURA 21 - TRECHO DE CÓDIGO CLASSE CONSULTA DE PRODUTO.....	33
FIGURA 22 - CONSULTA PREÇO.....	34
FIGURA 23 - CONSULTA DE PREÇO NO <i>TABLET</i> .....	34
FIGURA 24 – <i>TABLET</i> - LOGIN.....	39
FIGURA 25 – <i>TABLET</i> - MENU.....	39
FIGURA 26 – <i>TABLET</i> - ACERTO DE ESTOQUE.....	40
FIGURA 27 – <i>TABLET</i> - INVENTÁRIO DE ESTOQUE.....	40
FIGURA 28 – <i>TABLET</i> - CONSULTA DE PREÇO.....	41

## LISTA DE TABELAS

TABELA 1 - COMPONENTES DA APLICAÇÃO (ANDROID DEVELOPER, 2012).....	15
TABELA 2 - VERSÃO DO ANDROID E SUA DISTRIBUIÇÃO. (ANDROID DEVELOPER, 2015). ....	16
TABELA 3 - VERSÕES PARA DISTRIBUIÇÃO (FIREBIRD, 2004).....	21
TABELA 4 - PRINCIPAIS MÉTODOS DO CICLO DE VIDA DE UM FRAGMENT (ANDROID DEVELOPER, 2012). .....	25

## LISTA DE SIGLAS E ABREVIATURAS

OHA	Open Handset Alliance
HTC	High Tech Computer Corporation
SSL	Secure Socket Layer – Camada de Socket Seguro
GNU	General Public License
SDK	Software Development Kit.
APIs	Application Programming Interface
ANSI	American National Standards Institute
ADT	Android Development Tools
AVD	Android virtual Device Manager
JCA	Java EE Connector Architecture
JDBC	Java Database Connectivity
BSD	Berkeley Software Distribution
ERP	Enterprise Resource Planning
APIs	Application Programming Interface
IP	Internet Protocol
DNS	Domain Name System
XML	Extensible Markup Language

## **RESUMO**

Este trabalho descreve as atividades realizadas durante o período de estágio supervisionado realizado na empresa WorkDev Tecnologia do discente Junior da Costa Nascimento, orientado por Raphael de Souza Rosa Gomes e supervisionado por João Luiz Oliveira Ribeiro. O relatório tem intuito de mostrar o desenvolvimento de um aplicativo para dispositivos móveis na plataforma Android integrado com o software ERP. Para a realização do desenvolvimento do aplicativo foi utilizado a linguagem de programação Java, utilizando o SDK Android e API fragments. Durante a realização do desenvolvimento foi utilizado o framework ágil de desenvolvimento denominado SCRUM. Desenvolver uma aplicação com uma boa interface com o usuário não é simples e com diversos tamanho de telas como os tablets a API fragments trouxe resultados satisfatórios em múltiplos aparelhos, pois permite dividir a tela em vários pedaços e dar comportamento a cada pedaço (fragments). Tendo uma grande vantagem o uso desse recurso permite ao desenvolvedor criar aplicações que suportam múltiplos layouts.

## 1. INTRODUÇÃO

O estágio foi realizado na empresa WorkDev tecnologia, uma empresa de desenvolvimento de software voltado para automação comercial, atendendo os seguimentos de supermercados, materiais de construção e distribuidoras.

A partir da necessidade de minimizar erros na identificação de etiquetas de preços até a impressão das alterações necessárias, estava sendo realizada de forma manual. Tendo em vista melhorar esses processos, reduzir o uso de papeis e formulários e reduzir incidência de erros na execução dos processos seria necessário automatizar essa tarefa, que traria de vantagens: alteração de informações dos produtos em tempo real, redução do tempo de realização do inventário, precisão e redução de custos.

O relatório tem intuito de mostrar o desenvolvimento de um aplicativo modular para dispositivos móveis utilizando a plataforma Android para auxiliar os gestores da empresa na gestão dos processos de estoque. O sistema permite a realização de Auditoria de Preços, Inventário e Acerto de Estoque com leitura de código de barras através de câmera do aparelho e scanner portáteis. O aplicativo funciona integrado com o software ERP da empresa através da rede interna utilizando WIFI.

O aplicativo foi desenvolvido desde o início por uma equipe formada por um desenvolvedor, um analista e um gerente de projeto. O framework de desenvolvimento utilizado durante todo processo foi o SCRUM, em um intervalo de 7 dias era disponibilizada uma versão funcional do aplicativo para que os analistas pudessem realizar os testes pertinentes e dar um *feedback* da versão disponibilizada, assim repetindo esse ciclo até a o produto final.

O documento está organizado da seguinte forma: O capítulo 2 aborda a revisão literatura utilizado durante o período de estágio. No capítulo 3 descreve os materiais, técnicas e métodos e ferramentas utilizadas para realização do trabalho. O capítulo 4 demonstra os resultados através de imagens e descritivo. As dificuldades encontradas são descrita no capítulo 5 e por fim o capítulo 6 é feitas as considerações finais sobre os resultados obtidos no trabalho.

## 2. REVISÃO DE LITERATURA

No decorrer do curso foram adquiridos vários conceitos fundamentais para o desenvolvimento das atividades envolvidas no trabalho. Neste capítulo são abordados os fundamentos teóricos e práticos, sob os quais foram realizados no estágio supervisionado.

### 2.1 Introdução ao Android

O Android é uma plataforma aberta que foi desenvolvido pela OHA (Open Handset Alliance), liderada pela gigante Google e outras empresas como HTC, LG, Motorola, Samsung, Sony Ericsson, Toshiba, HTC, Huawei, Sprint Nextel, China Mobile, T-Mobile, ASUS, Intel, Acer, Dell, Garmin entre outras, assim o objetivo dessas empresas é definir uma plataforma para inovar e acelerar o desenvolvimento de aplicações móveis com recursos ricos e flexíveis a todos os usuários de sua plataforma, sendo baseado no sistema operacional Linux para dispositivos móveis, tais como tablets e smartphones.

“Android fornece uma estrutura de aplicativo rico que lhe permite construir aplicativos e jogos inovadores para dispositivos móveis em um ambiente de linguagem Java.”  
(ANDROID DEVELOPER, 2012).

A primeira versão beta do Android, foi lançado pela Google em 2007, sendo que a primeira versão comercial Android 1.0 foi lançado em setembro de 2008. Em outubro de 2008 começa a ser comercializado o primeiro aparelho com Android, o T-Mobile G1 da HTC com um design sólido e robusto, sendo considerado inovador, contendo um sistema de notificações e integração forte com serviços do Google como descreve a Figura 1. (ANDROID DEVELOPER, 2012).

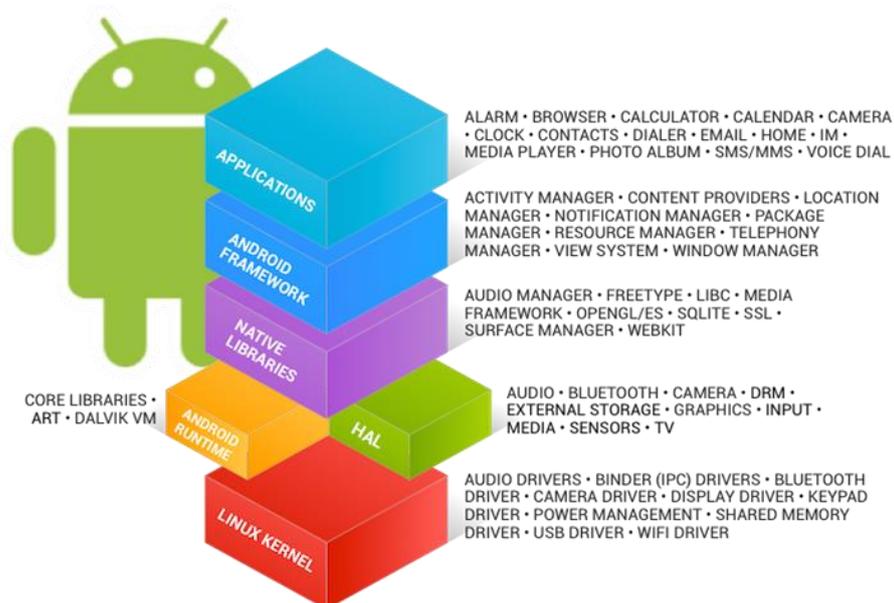


**Figura 1-** T-Mobile G1. (Android Developer, 2012).

O Android está disponível sobre a licença de software livre e *open source*, sendo que sua maior parte é publicada sobre a licença Apache 2.0 e a alteração do Kernel Linux sobre a GNU 2.0.

A estrutura geral da plataforma Android é composta pela SDK um kit de desenvolvimento que disponibiliza sua APIs e ferramentas que são necessárias para desenvolver aplicações, baseada na linguagem Java.

O sistema operacional Android está dividido em cinco seções e quatro camadas principais, como descreve a Figura 2.



**Figura 2-** Arquitetura do Android (Source Android, 2015).

### 2.1.1 Linux Kernel

A camada Linux Kernel é o núcleo do sistema operacional Android e, é baseado na versão do Kernel 2.6, esta camada é responsável pelas funcionalidades básica do sistema como, gerenciamento de processos, memória, dispositivos como WIFI, áudio, display, câmera, teclado etc. A segurança do Android é baseada na segurança do Linux, onde cada aplicação é executada em um único processo e cada processo possui thread responsável pelo processo (Android Developers, 2012).

### 2.1.2 Libraries

Essa camada contém um conjunto de bibliotecas C/C++ que são utilizadas pelo sistema, como bibliotecas multimídias, SSL responsável pela segurança de internet, navegação web, banco de dados SQLite repositório para armazenamento e compartilhamento de dados.

### 2.1.3 AndroidRuntime

Responsável pelo tempo de execução dos aplicativos através da máquina virtual Dalvik, criada para cada aplicação, possui também um conjunto de bibliotecas. A máquina virtual Dalvik executa classes compiladas pelo compilador Java, os arquivos *.class* gerados são transformados no formato *.dex* pela ferramenta *dx*, incluídas no SDK.

### 2.1.4 Application Framework

A camada de Application framework permite a interação de mais alto nível com o sistema Android a partir de sua APIs para desenvolvimento de aplicações.

### 2.1.5 Componentes da Aplicação

Os componentes são blocos de construção essenciais para desenvolvimento dos aplicativos Android. Conforme a Tabela 1 existe quatro tipos de componentes de aplicativos, cada um com sua finalidade e ciclo de vida distinta que define como o componente é criado e destruído (ANDROID DEVELOPER, 2012).

Componentes	Função
Activities	Atividades de interação do usuário com o smartphone.
Services	Serviços lidam com o processamento em segundo plano dos aplicativos.
Broadcast Receivers	Responsável pela comunicação entre o sistema operacional e aplicativos.
Content Providers	Gerenciamento dos dados e questões de gestão de banco de dados.

**Tabela 1** - Componentes da Aplicação (ANDROID DEVELOPER, 2012).

#### 2.1.6 Plataforma (Versão do Android)

No Android uma versão do sistema operacional é conhecida como plataforma. Cada plataforma tem um código identificador, que é chamado de API Level. A Tabela 2 mostra as versões disponíveis hoje no mercado e sua distribuição.

“Esses dados mostram dispositivos que executam as mais recentes app da loja Google Play, que é compatível com Android 2.2 ou superior. Os Dados representam todos os dispositivos que visitaram o Google Play Store na prévia de 7 dias.” (ANDROID DEVELOPER, 2015).

Versão	Nome	API	Distribuição
2.2	Froyo	8	0,4%
2.3.3 - 2.3.7	Gingerbread	10	6,9%
4.0.3 -4.0.4	Ice Cream Sandwich	15	5,9%
4.1.x	JellyBean	16	17,3%
4.2.x		17	19,4%
4.3		18	5,9%
4.4	KitKat	19	40,9%
5	Lollipop	21	3,3

**Tabela 2** - Versão do Android e sua Distribuição. (Android Developer, 2015).

## 2.2 Desenvolvimento Ágil de Software

Segundo Pressman (2011), a engenharia de software ágil combina filosofia com um conjunto de princípios de desenvolvimento. Onde essa filosofia defende a satisfação do cliente e a entrega incremental prévio; equipes de projetos pequenas e altamente motivadas; métodos informais; artefatos de engenharia de software mínimos e, acima de tudo, simplicidade no desenvolvimento geral. Os princípios de desenvolvimento priorizam a entrega mais que a análise e projeto, também priorizam a comunicação ativa e contínua entre desenvolvedores e clientes.

### 2.2.1 Scrum

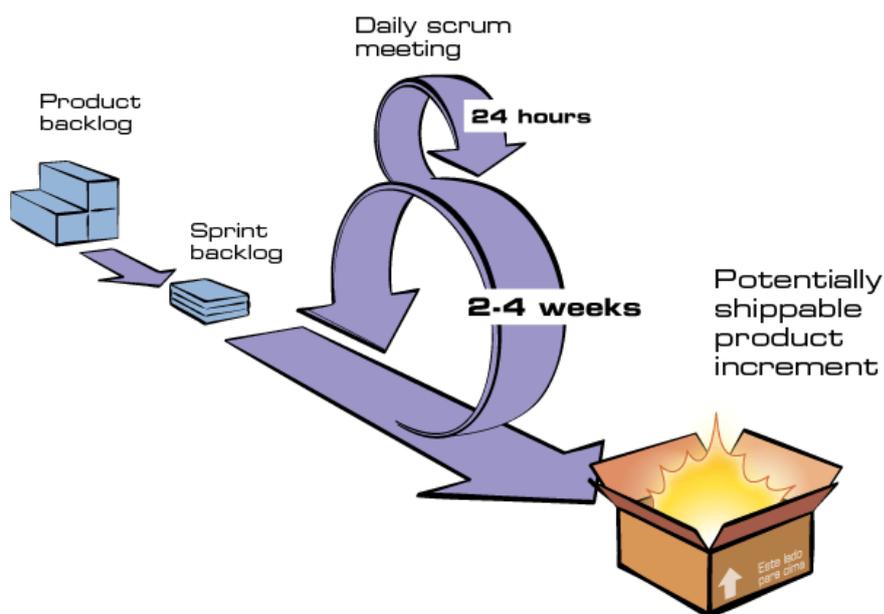
O *Scrum* é um *framework* para organizar e gerenciar trabalhos complexos, tais como desenvolvimento de software. O *Scrum* fornece um conjunto de valores, princípios e praticas para que as organizações se organizem de maneira que produza um produto de alta qualidade (ÁGIL, 2001).

O *Scrum* consiste em três papéis básicos: *Product Owner*, *ScrumMaster* e *Time Scrum*. O *Product Owner* determina o que precisa ser construído e qual deve

ser sua ordem. É responsável em comunicar a todos os outros participantes do *Scrum* uma visão clara do projeto.

O *ScrumMaster* é o responsável por ajudar a melhorar continuamente o processo, a equipe e o produto que está sendo criado, tendo um papel de facilitador. O *Time Scrum* é simplesmente a junção de todas as pessoas que são responsáveis pela concepção, construção e teste do produto. Um Time de desenvolvimento tem entre 5 e 9 pessoas, sendo que seus membros devem ter todas as habilidades necessárias para produzir com qualidade o produto, a Figura 3 demonstra o ciclo de vida de um scrum.

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista chamada *Product Backlog*, onde a cada *Sprint* realiza-se um *Sprint Planning Meeting*, que é uma reunião de planejamento onde é definido o *Product Owner*, e a equipe define as atividades do Sprint que se inicia. O *Daily Scrum* é uma breve reunião que tem o objetivo disseminar o conhecimento sobre o que foi realizado no dia anterior e ao final de cada Sprint a equipe apresenta as funcionalidades que foram implementadas em um *Sprint Review Meeting* e por fim, realiza-se um *Sprint Retrospective*, onde a equipe planeja o próximo Sprint.



**Figura 3** - Ciclo do Scrum (Desenvolvimento Ágil, 2013).

## 3. MATERIAS, TÉCNICAS E MÉTODOS

No desenvolvimento das atividades deste estágio foram utilizadas algumas técnicas e métodos para um melhor planejamento do projeto.

### 3.1 Ferramentas de Desenvolvimento

#### 3.1.1 Eclipse

O Eclipse é um ambiente de desenvolvimento que facilmente permite adicionar recursos para o desenvolvimento Android.

O Android oferece um *plugin* para o eclipse, chamado de ADT (Development Tools). Através do plug-in ADT é possível executar um emulador do Android diretamente do eclipse com mostra a Figura 4, e utilizar alguns recursos como debug, *garbage collector* entre outros. Para distribuir uma aplicação Android, é necessário compilar o projeto Java em uma extensão .apk (Android Package File), um arquivo binário compactado com todas as classes compiladas e seus recursos.

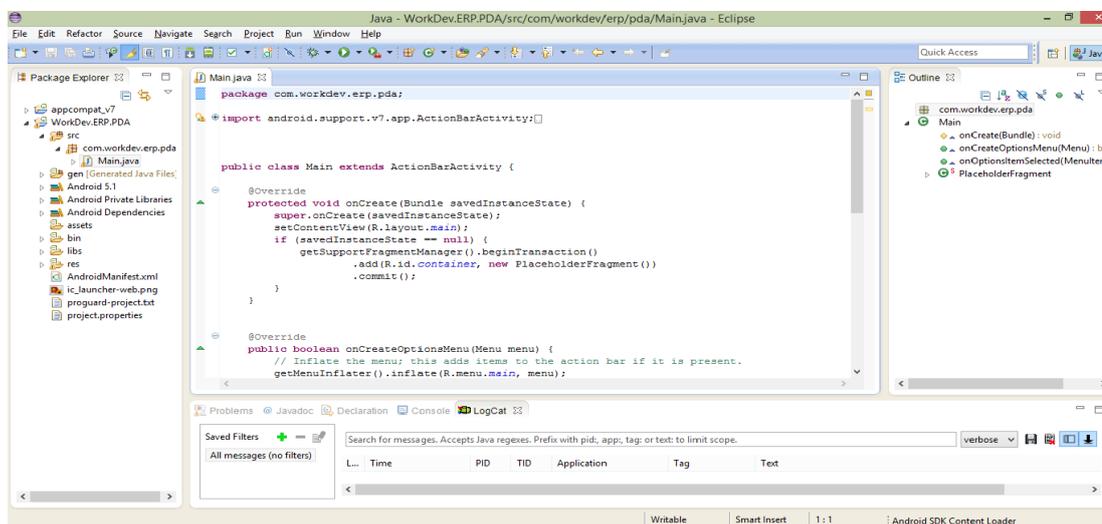
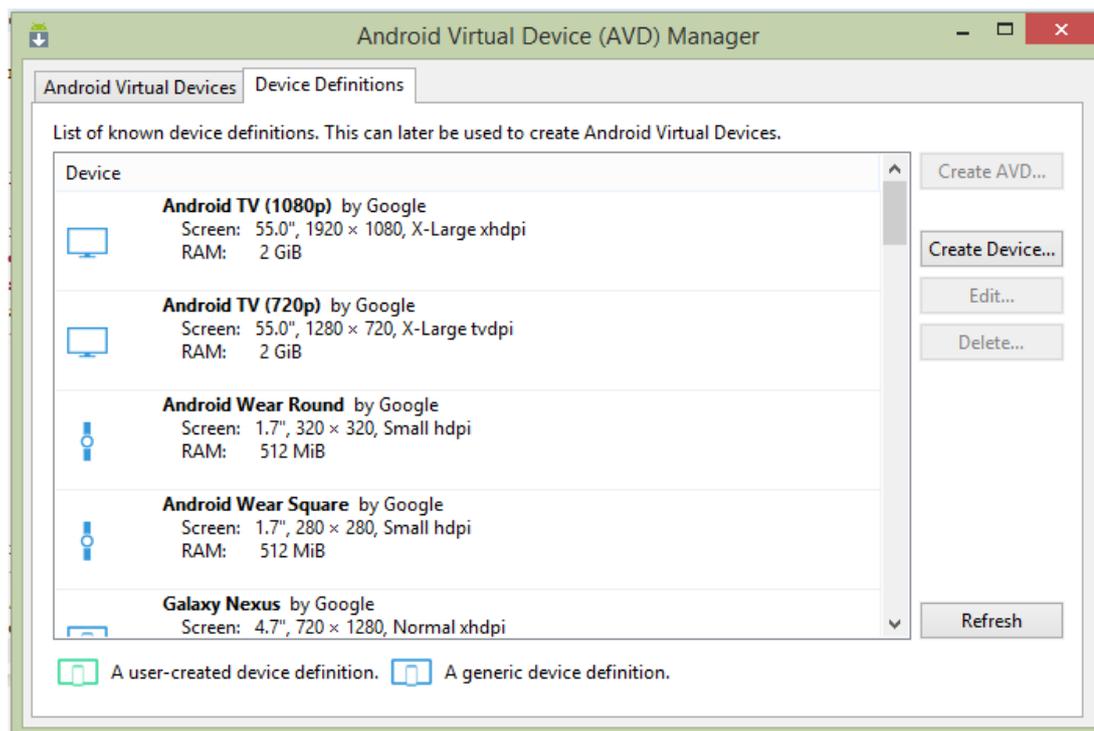


Figura 4 - Ambiente de desenvolvimento Eclipse.

### 3.1.2 Android virtual Device Manager (AVD)

O AVD manager é uma ferramenta que permite criar e gerenciar dispositivos virtuais Android, que permite definir as configurações para o emulador Android como mostra a Figura 5.



**Figura 5** - Android Virtual Device AVD.

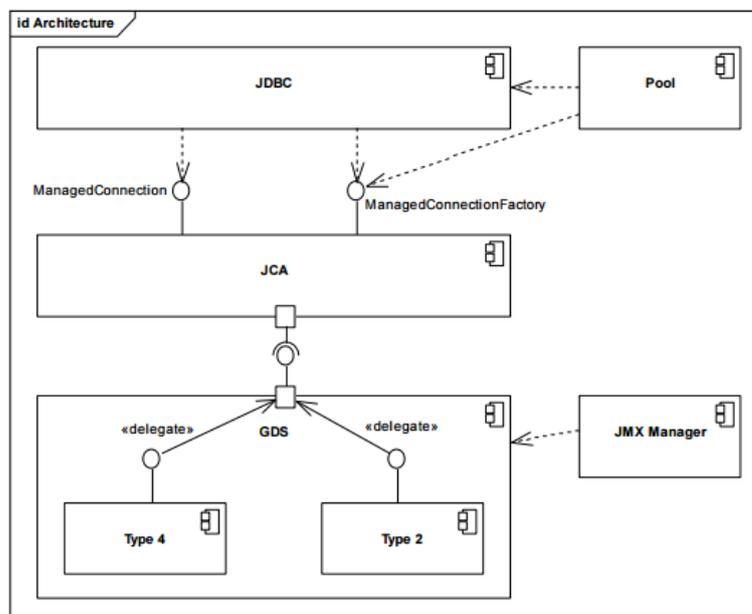
### 3.1.3 Jaybird

O Jaybird é um driver JCA/JDBC para realizar conexão com o servidor de banco de dados Firebird utilizando aplicação Java. O driver é baseado no novo padrão JCA para servidores de aplicativos, o padrão JCA especifica uma arquitetura na qual uma aplicação servidor possa utilizar o driver, onde o servidor é responsável por gerenciar transações, segurança e pool de recursos. O Jarybird requer ao menos a versão do JDK 1.3.1 para que possa compilar e executar (Firebird, 2004).

O Jaybird JCA/JDBC é distribuído gratuitamente sobre a licença GNU.

Segundo a SUN, API JDBC é um padrão para a conectividade independente que permite o acesso ao banco de dados baseado em SQL e outras fontes de dados tabulares, como planilhas ou arquivos simples (SUN, JDBC).

Conforme a Figura 6 o JCA (Java EE Connector Architecture) define uma arquitetura para sistema Java EE que permite conectar servidores de aplicativos e sistemas de informação (Java Community Process).



**Figura 6** - Arquitetura Jaybird (Firebird, 2004).

O jaybird 2.2.x foi testado com as versões do Firebird 2.1.5 e 2.5.2, porem é compatível com o Firebird 1.0 e versões acima. Este driver não suporta servidores interbase devido alterações no Firebird, onde as especificações de protocolos e fixação do banco de dados que são enviados ao servidor.

Firebird é um banco de dados relacional que oferece muitas características padrão ANSI SQL que funciona em Linux, Windows, e em uma variedade de plataformas Unix. Firebird oferece concorrência excelente, de alto desempenho e suporte a poderosa linguagem para stored procedures e triggers. (Firebird, 2000).

O jaybird 2.2.x tem suporte para Java 5 (JDBC 3.0), Java 6 (JDBC 4.0) e Java 7 (JDBC 4.1), para a versão Java 8 será adicionado na versão jaybird 2.2.4. A próxima versão não terá suporte ao Java 5 (Firebird, 2004). Na tabela 3 descreve a versão disponível.

<b>Versão</b>	<b>Descrição</b>
jaybird-2.1.6.jar	O arquivo contém o driver o JCA/JDBC.
Jaybird-full-2.1.6.jar	Além dos driver acima contém as classes pool de conexão.
jaybird21.dll	Versão pré-compilada da biblioteca JNI para o Tipo 2 e driver de servidor embutido para 32-bit plataforma Windows.
libjaybird21.so	Versão pré-compilada da biblioteca JNI para o Tipo 2 e drivers de servidor embutido para plataformas Linux de 32 bits.

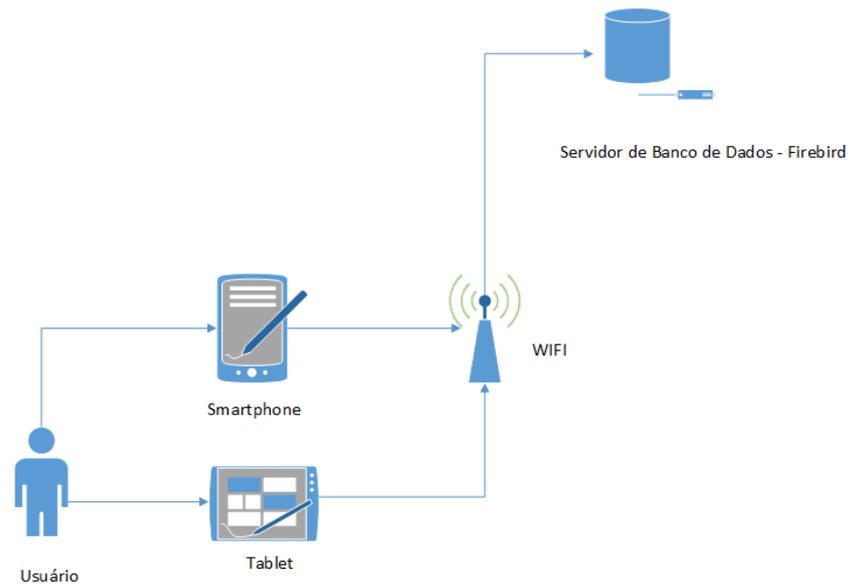
**Tabela 3** - Versões para distribuição (Firebird, 2004).

### **3.2 Modelagem da Arquitetura**

O projeto do aplicativo é modular para controle e gerência dos processos realizados no estoque, utilizando dispositivos móveis para leitura de código de barras através da câmera do aparelho e scanner portáteis. O sistema permite a realização de auditoria de preços e inventário de estoque. O aplicativo funciona integrado com o software ERP da empresa através da rede interna utilizando WIFI.

O ERP é uma solução em automação comercial, com ênfase em Supermercados, Materiais de Construção e Distribuidora, com conceitos administrativos, financeiro e contábil, que consegue dar respaldo ao administrador em tomada de decisões.

Na Figura 7 demonstra a arquitetura de comunicação com o aplicativo, sendo que o usuário a partir de um dispositivo móvel realiza a comunicação com o banco de dados através da rede WIFI.



**Figura 7** - Arquitetura do sistema.

### 3.3 Android Fragment

Com o surgimento dos Tablets e Smartphones com telas maiores e com sua popularização, surgiu a necessidade de customizar o Android para que seus usuários pudessem usufruir ao máximo dos tamanhos de telas disponíveis nesses aparelhos.

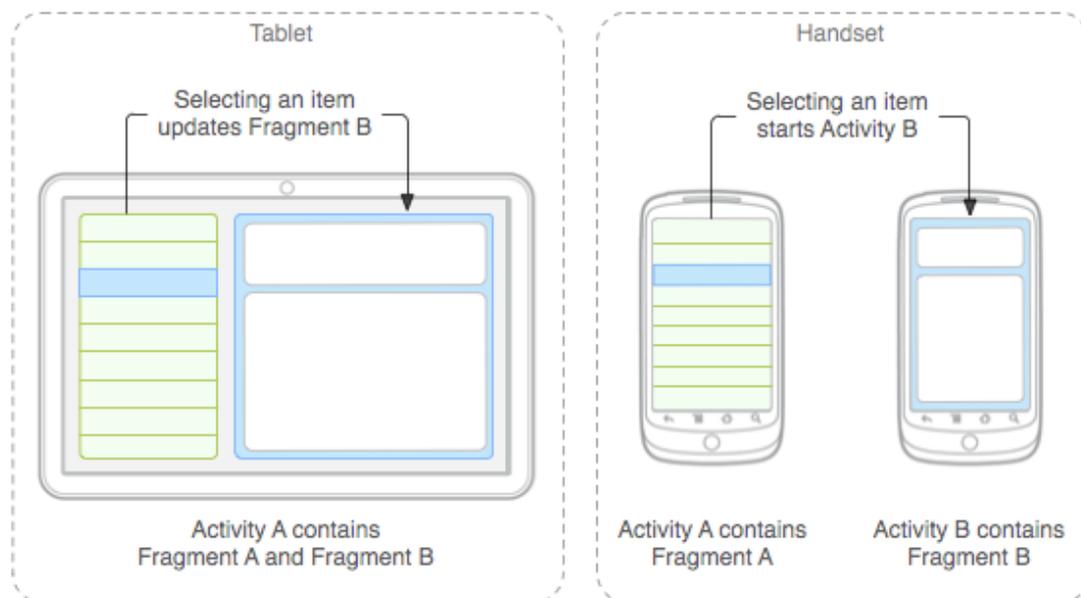
A partir da versão Android 3.0 Honeycomb API 11, a API Fragment foi adicionada para auxiliar os desenvolvedores criar aplicações que pudesse rodar em Tablets e Smartphones, visando uma arquitetura simples, de fácil manutenção e reaproveitamento de código (ANDROID DEVELOPER, 2012).

#### 3.3.1 Fragment

Fragment é um componente Android independente que pode ser usado por uma activity (Atividade). Um fragment encapsula a funcionalidade de maneira que possa ser reutilizado pelos layouts.

Cada fragment tem seu próprio ciclo de vida e sua própria interface de usuário. É possível definir um fragment sem interface. Um fragment é executado em um contexto de uma activity (ANDROID DEVELOPER, 2012).

A Figura 8 demonstra um exemplo típico com dois módulos de interface de usuário definidos por fragmentos que podem ser combinados em uma atividade, em um tablet e smartphone.



**Figura 8** - Design Android Fragment (Android Developer, 2012).

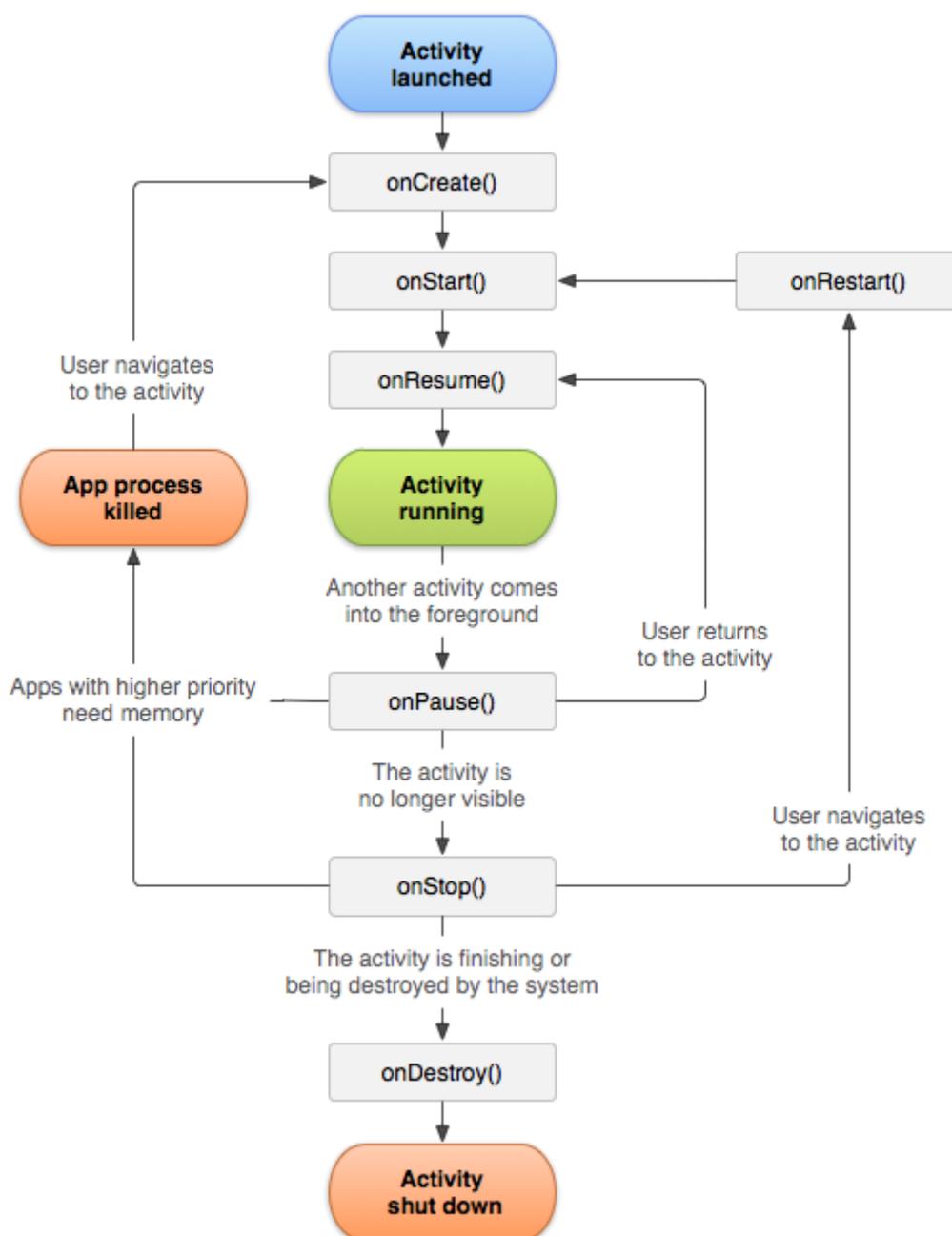
Uma activity pode ter vários fragment, primeiro de tudo deve decidir quantos fragmentos que você deseja usar em uma atividade. Baseada no número de fragmentos criado, criar classes que estendem a classe fragment. A classe fragment tem funções de retorno de chamada que pode substituída por qualquer uma das funções com base em suas necessidades.

### 3.3.2 Ciclo de Vida Fragment

Para criar um Fragment, onde é necessário criar uma subclasse de Fragment. A classe Fragment contém código parecido com a classe de uma Activity. Ele contém métodos de retorno de chamada semelhantes aos de uma atividade, como onCreate(), onStart(), onPause() e onStop(). O Android Fragment possui seu próprio ciclo de vida e esta ligada ao ciclo de vida de sua activity como demonstra a Figura 9.

O ciclo de vida completo de uma activity ocorre pela chamada do método onCreate(Bundle) até a chamada de onDestroy(). Sendo assim, uma activity irá executar tudo o que for “global” no onCreate() e liberar todos os recursos no onDestroy().

O onStart() até o onStop() é o ciclo da activity que pode ser visível pelo usuário, mesmo que não possa interagir, os dois métodos pode manter os recursos necessário para manter a activity para o usuário.



**Figura 9** - Ciclo de Vida Fragment. (Android Developer, 2012).

A Tabela 4 descreve os principais métodos do ciclo de vida de um fragment.

<b>Método</b>	<b>Descrição</b>
onAttach(activity)	Método é chamado logo depois do fragment ser associado à activity.
onCreateView(inflater, viewgroup,bundle)	Este método é o mais importante, e uma vez que o fragment cria sua interface com a activity.
onActivityCreated()	É chamado logo após onCreateView quando a activity foi criado, podendo criar instancia de objetos.
onDestroyView()	É chamado quando a view criada pelo fragment esta sendo destruída.
onDetach()	É chamado quando o fragment esta sendo desassociado da sua activity.

**Tabela 4** - Principais métodos do ciclo de vida de um fragment (Android Developer, 2012).

## 4. RESULTADOS

Neste capítulo será demonstrado os resultados do estágio através das principais telas e com um descritivo de cada uma. O primeiro elemento importante da API Android é classe Activity (android.app.Activity). Ela define uma tela da aplicação, controla seu estado, eventos, passagens e recebimento de parâmetros.

Uma tela é composta por vários elementos visuais, como botões, textos, entre outros elementos. Um componente visual no Android, é representada pela classe View (android.view.View), ou por arquivo XML como elementos.

Por tanto para criar uma tela da aplicação é necessário criar uma classe, estender de Activity e vincular esta classe a uma view.

Uma view definida por um XML deverá ser vinculado à uma activity pela instrução setContentView(view) passando como parâmetro a view. Este procedimento é invocado dentro do método onCreate() da activity, invocado no momento da criação da tela como descreve a Figura 10.

```
package br.com.workdev.erp.pda;

import android.app.Activity;

public class Aplicativo extends Activity
{
    private TextView SB_txtFantasia,
                    SB_txtEndereco,
                    SB_txtBairro,
                    SB_txtCidade,
                    SB_txtComplemento,
                    SB_txtFone,
                    SB_txtCep,
                    SB_txtSite,
                    SB_txtVersao;

    public static final int SAIR = 0;

    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.frm_aplicativo_sobre);

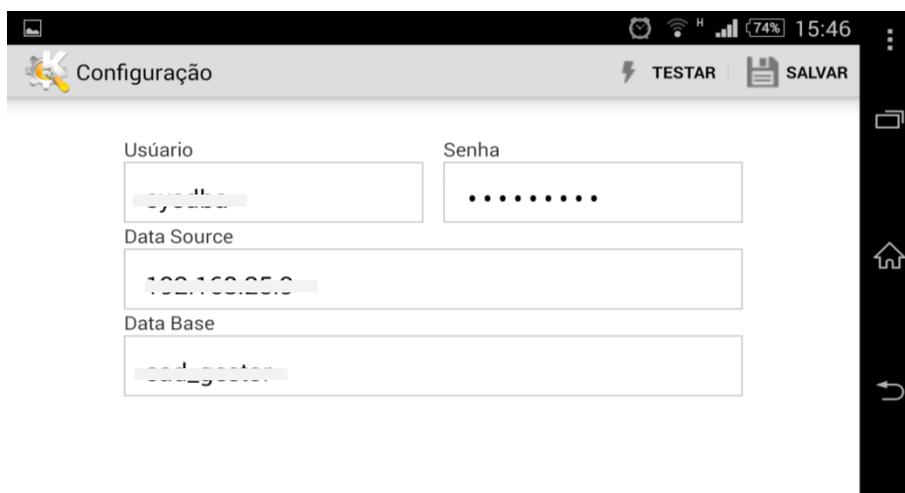
        SB_txtFantasia = (TextView) findViewById(R.id.SB_txtFantasia);
        SB_txtEndereco = (TextView) findViewById(R.id.SB_txtEndereco);
        SB_txtBairro = (TextView) findViewById(R.id.SB_txtBairro);
    }
}
```

Figura 10 - Trecho de código da classe Activity Aplicativo.

A Figura 11 mostra tela de configuração do aplicativo, que permite a comunicação com a base de dados do ERP.

O usuário configura a conexão informando o usuário e senha do banco de dados, o Data Source define o IP onde está localizado o banco de dados. É recomendado utilizar um número de IP ao invés do DNS da máquina, pois se houver problema na rede o aplicativo não pode localizar o servidor automaticamente.

Data Base define o caminho da pasta onde está localizado o Banco de Dados.



**Figura 11 - Configuração.**

A Figura 12 mostra a tela de login do aplicativo e engloba todo controle de acesso, ao informar o login e senha, o aplicativo irá checar as permissões de acesso ao aplicativo conforme cadastrado no ERP.



**Figura 12 - Login.**

A Figura 13 mostra o próximo passo após o login. Caso haja mais de uma empresa o aplicativo listará as empresa, para que o usuário possa selecionar e realizar as operações.



**Figura 13 – Empresa.**

Após selecionar a empresa a próxima tela a Figura 14 representa o menu com as funcionalidades disponíveis no aplicativo de acordo com as permissões de acesso de cada usuário.

O Acerto de Estoque é um controle interno que permite o ajuste de estoque de um determinado produto.

Inventário é utilizado para realizar a manutenção das informações dos produtos em estoque, torna-se possível uma gestão mais eficiente dos recursos, com acompanhamentos das contagem real do estoque e análise de divergências.

Consulta de Preço, opção de consulta rápida de um determinado produto, este módulo permite o gerenciamento das mercadorias na área de venda e controle das divergências nos preços em relação ao ERP, mantendo os preços sempre atualizados nas gôndolas.



**Figura 14 – Menu.**

Para uma empresa é crucial um controle adequado de seus estoques, pois estas informações auxiliam o gestor nas tomadas de decisões da empresa, tais como análise de desempenho e planejamento de reposição através de sugestões de compra por estoque mínimo e máximo, caso um produto tenha saldo divergente podem ser geradas sugestões de compra desnecessárias. Desta forma é aconselhável que não haja divergências de estoque.

A Figura 15 mostra a tela de acerto de estoque, que permite ao usuário ajustar a quantidade de um determinado produto em estoque, seja Físico ou Fiscal.

Estoque - Acerto

DataBase  
25/04/2015

Estoque  
Fisico

Local de Estoque  
Area de Venda  Acumular Saldo

Operação

Informe o Código/Barras do Produto

Descrição do Produto

Quantidade Atual Nova Quantidade  
0.000 1.000

+ - ✓

**Figura 15 - Acerto de Estoque.**

A Figura 16 mostra a tela de Inventário, é utilizado para realizar a manutenção de informações do inventario de acordo com a data e abertura no ERP, através deste são inseridas as informações do produto, tais como Quantidade e Número de Serie e Lote.

Estoque - Inventário

CONTAGEM

Data Base Abertura  
25/04/2015 0

Local de Estoque Estoque  
Area de Venda Fisico

Produto

Informe o Código/Barras Produto

Produto

Quantidade Atual Quantidade Apurada  
0.000 0.000

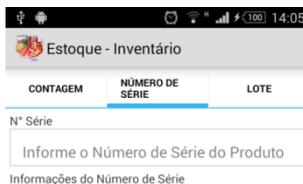
Nova Quantidade Quantidade Adicionar/Remover  
0.000 1.000

+ - ✓

**Figura 16 - Inventário.**

O Número de Série é uma identificação exclusiva de um produto junto ao seu fabricante, existe caso onde produtos são controlados por números de série. Caso o produto tenha número de serie será habilitado a Tab Número de Série conforme

mostra a Figura 17. Esta opção permite realizar controle por número de série dos produtos.



**Figura 17 - Número de Série.**

O Lote é uma informação sobre a produção de determinados itens, onde os produtos são fabricados sob as mesmas condições, possuem as mesmas características, físicas, químicas e mesmas dimensões.

Caso o produto tenha Lote será habilitado a Tab Lote conforme mostra a Figura 18. Esta opção permite o controle por lote do produto, de acordo com sua fabricação e vencimento e quantidade em estoque, que podem ser determinantes para uma gestão eficiente do estoque.



**Figura 18 – Lote.**

A Figura 19 e 20 mostra a implementação da tela de Consulta de Preço, onde foi definido dois fragment e suas respectivas view, o primeiro para as informações do produto e o segundo com as informações de última compra.

O fragment permite adaptar o layout para tablet e smartphones, esta funcionalidade o torna uma estrutura mais poderosa, mantendo uma boa experiência de interface em todos os dispositivos Android.

Portanto é preciso criar uma subclasse de Fragment (android.app.Fragment), e pelo menos implementar o método onCreateView(...) para retornar a view que será responsável por exibir o conteúdo no espaço reservado ao fragment. O método onCreateView(...) é chamado durante o início do ciclo de vida do fragment e deve retornar uma view que será utilizada para preencher o espaço da tela. Para isso o método recebe como parâmetro um objeto LayoutInflater para auxiliar a criação da view. Podemos visualizar o código-fonte do fragment.

```
package br.com.workdev.erp.pda;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Estoque_Consulta_Produto_Padrao extends android.support.v4.app.Fragment
{
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
    {
        View view = inflater.inflate(R.layout.frm_es_cs_produto_padrao, container, false);

        return view;
    }
}
```

Figura 19 - Fragment – Produto Padrao.

```
package br.com.workdev.erp.pda;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Estoque_Consulta_Produto_Inf extends android.support.v4.app.Fragment
{
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
    {
        View view = inflater.inflate(R.layout.frm_es_cs_produto_inf, container, false);

        return view;
    }
}
```

Figura 20 - Fragment – Produto Inf.

A Figura 21 implementa a classe responsável pela criação da view fragment, a classe estende a classe `FragmentActivity` que contém o método `onCreate(bundle)` e utiliza uma view para criar a interface com método `setContentView(view)`.

```
package br.com.workdev.erp.pda;

import java.sql.Connection;

public class Estoque_Consulta_Produto extends Activity
{
    private TextView PR_txtProduto;
    private TextView PR_txtDescricao;
    private TextView PR_txtEstoque_Atual;
    private TextView PR_txtUnidade;
    private TextView PR_txtCusto;
    private TextView PR_txtPreco;
    private Button PR_btnConsulta;
    private Button PR_btnScanner;

    public static final int SAIR = 0;
    private Intent intent;

    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.frmesproduto_preco);

        PR_txtProduto = (TextView) findViewById(R.id.PR_txtProduto);
        PR_txtDescricao = (TextView) findViewById(R.id.PR_txtDescricao);
        PR_txtEstoque_Atual = (TextView) findViewById(R.id.PR_txtEstoque_Atual);
        PR_txtUnidade = (TextView) findViewById(R.id.PR_txtUnidade);
        PR_txtCusto = (TextView) findViewById(R.id.PR_txtCusto);
        PR_txtPreco = (TextView) findViewById(R.id.PR_txtPreco);
        PR_btnConsulta = (Button) findViewById(R.id.PR_btnConsulta);
        PR_btnScanner = (Button) findViewById(R.id.PR_btnScanner);
    }
}
```

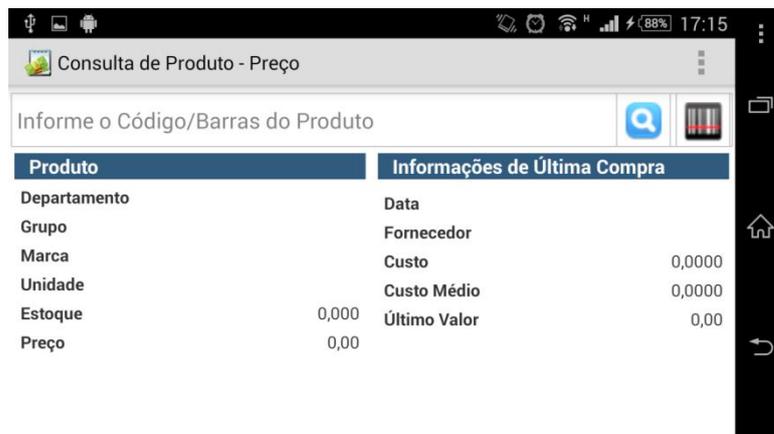
**Figura 21** - Trecho de código classe Consulta de Produto.

A Figura 22 mostra a tela de Consulta de Preço, que tem por objetivo mostrar uma visualização rápida de informações referentes ao preço de um determinado produto, com informações sobre a última aquisição, como: Fornecedor, Data, Custo, Custo Médio e Último Valor de compra.



**Figura 22** - Consulta Preço.

Na Figura 23 mostra a consulta visualizada via tablet.



**Figura 23** - Consulta de Preço no *tablet*.

No Anexo I, foram adicionadas outras imagens que mostram o comportamento do fragment em tablet para outras telas do software.

## **5. DIFICULDADES ENCONTRADAS**

Uma das dificuldades encontradas foi na criação do layout das telas que adequasse ao ambiente de trabalho que seria utilizado o aplicativo, manter uma experiência de interface gráfica independente do tamanho da tela que facilitasse a interação do usuário com o aplicativo de forma prática.

Além da visualização, a forma de processar e armazenar as aplicações varia de aparelhos para aparelhos. E tudo isso precisa ser ajustado a uma grande quantidade de aparelhos disponíveis no mercado.

## 6. CONCLUSÕES

O estágio abordou um estudo sobre as tecnologias e ferramentas envolvidas no desenvolvimento do aplicativo em dispositivos móveis utilizando a plataforma Android e linguagem de programação Java. O aplicativo desenvolvido cumpriu os requisitos analisados para a utilização e modernização dos processos realizados através da Auditoria de Preço, Inventário e Acerto de Estoque, o que anula o uso de papel e traz ganho em relação à diminuição do tempo em cada ação e a confiabilidade das informações, possibilitando uma rápida tomada de decisões.

As ferramentas utilizadas no ambiente de desenvolvimento mostraram ser suficientes para elaboração do aplicativo, não tendo custo para projeto, pois são ferramentas gratuitas ou de licença livre.

Com a modernização dos dispositivos móveis novas tecnologias estão sendo implementada e disponibilizadas no mercado a todo o momento, fazendo com que os processos sejam revistos e modernizados.

A realização do trabalho permitiu ao discente aplicar os conhecimentos adquiridos durante o curso, além de adquirir experiência na área de desenvolvimento mobile.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- AGIL, Scrum Metodologia Ágil; Disponível por www em.  
<http://www.desenvolvimentoagil.com.br/scrum/>. (Acessado em 11 de abril de 2015).
- ANDROID SOURCE, Introdução ao Android; Disponível por www em  
<http://source.android.com/source/index.html>. (Acessado: 20 de março de 2015).
- ANDROID DEVELOPERS, Introdução ao Android; Disponível por www em  
<http://developer.android.com/guide/index.html>.(Acessado: 21 de março de 2015).
- ANDROID, DEVELOPERS; Disponível por www.  
<http://www.tutorialspoint.com/android/index.htm>. (Acessado: 20 de março de 2015).
- ANDROID DEVELOPERS, Android Fragment; Disponível por www em  
<http://developer.android.com/guide/components/fragments.html>. (Acessado em 28 de março de 2015).
- FIREBIRD, Jaybird; Disponível por www em  
<http://jaybirdwiki.firebirdsql.org/jaybird/doku.php?id=info:start>. Acessado em 28 de março de 2015.
- FIREBIRD, Jaybird Arquitetura; Disponível por www em  
[http://www.firebirdsql.org/file/documentation/drivers\\_documentation/Jaybird\\_2\\_1\\_JDBC\\_driver\\_manual.pdf](http://www.firebirdsql.org/file/documentation/drivers_documentation/Jaybird_2_1_JDBC_driver_manual.pdf). Acessado em 28 de março de 2015.
- FIREBIRD; Introdução ao Firebird, Disponível por www em  
<http://www.firebirdsql.org/en/about-firebird/>. (Acessado em 28 de março de 2015).
- FRAGMENT, Android Fragment; Disponível por www em  
<http://www.vogella.com/tutorials/AndroidFragments/article.html>. (Acessado em 28 de março de 2015).
- JCA, (Java EE Connector Architecture); Disponível por www em  
<http://java.sun.com/products/jdbc/>.(Acessado em 29 de março de 2015).
- JAVA EE ; Disponível por www em  
<http://www.oracle.com/technetwork/java/javaee/overview/index.html>.(Acessado em 29 de março de 2015).
- OPEN HANDSET ALLIANCE; disponível por www em  
<http://www.openhandsetalliance.com/>. (Acessado em 20 de março de 2015).
- PRESSMAN, R. Engenharia de Software: Uma abordagem Profissional. 2011. 7ª edição. Editora Bookman

SUN, JDBC; Disponível por www em <http://docs.oracle.com/javase/6/docs/api/java/sql/package-summary.html>. (Acessado em 28 de março de 2015).

SCRUM , A Metodologia Ágil Explicada de uma Forma Definitiva; Disponível por www em <http://www.mindmaster.com.br/scrum/>. (Acessado em 11 de abril de 2015).

## ANEXO I – LAYOUT EM TABLETS

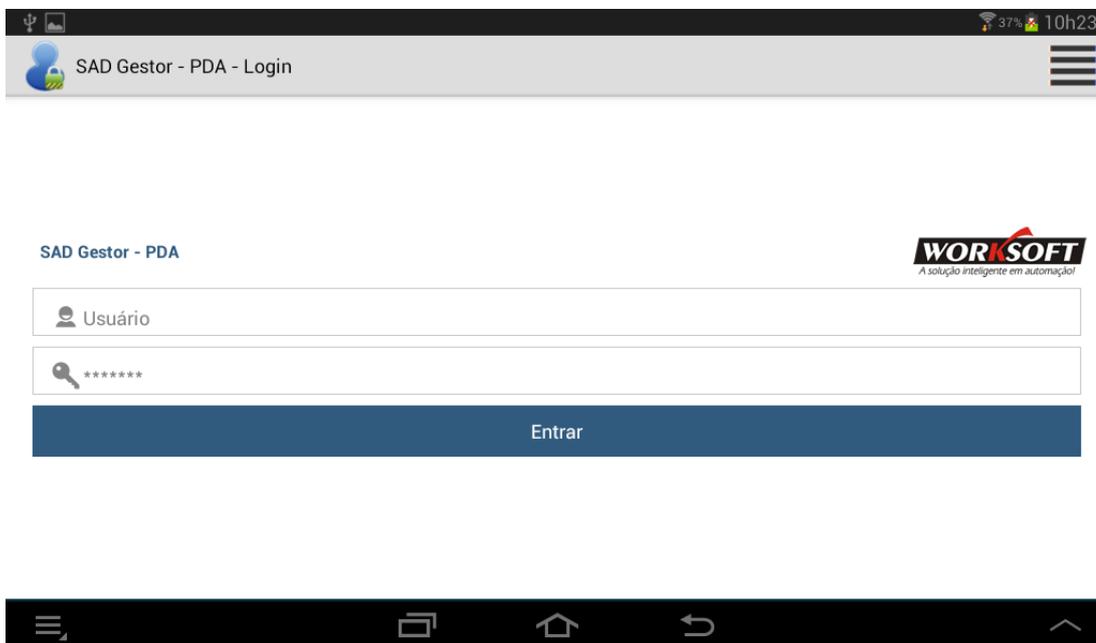


Figura 24 – Tablet - Login.

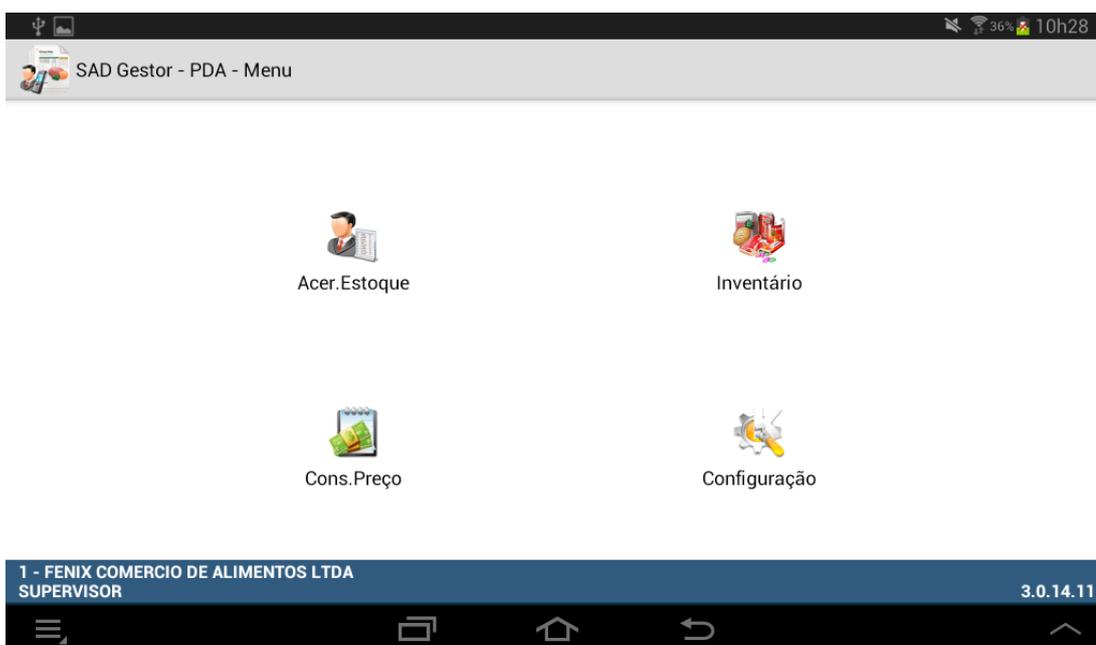


Figura 25 – Tablet - Menu.

Estoque - Acerto

DataBase  
12/11/2015

Estoque  
Fisico

Local de Estoque  
Area de Venda  Acumular Saldo

Operação

Informe o Código/Barras do Produto

Descrição do Produto

Quantidade Atual Nova Quantidade

+ - ✓

Figura 26 – Tablet - Acerto de Estoque.

Estoque - Inventário SALVAR

CONTAGEM

Data Base  
12/11/2015

Local de Estoque  
Area de Venda

Produto  
Informe o Código/Barras Produto

Abertura  
0

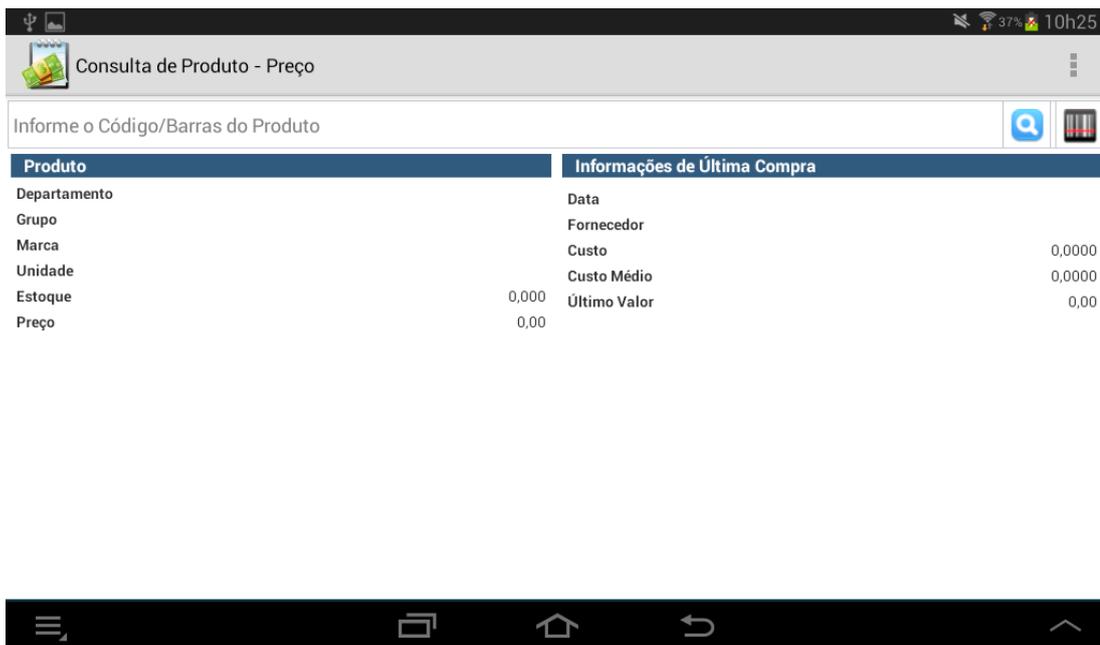
Estoque  
Fisico

Quantidade Atual Quantidade Apurada  
0.000 0.000

Nova Quantidade Quantidade Adicionar/Remover  
0.000 1.000

+ - ✓

Figura 27 – Tablet - Inventário de Estoque.



Consulta de Produto - Preço

Informe o Código/Barras do Produto

Produto		Informações de Última Compra	
Departamento		Data	
Grupo		Fornecedor	
Marca		Custo	0,0000
Unidade		Custo Médio	0,0000
Estoque	0,000	Último Valor	0,00
Preço	0,00		

Figura 28 – Tablet - Consulta de Preço.