UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA JAVA

DANILO DAVID PEREIRA COSTA

APLICATIVO PARA DISPOSITÍVO MÓVEL ANDROID PARA SUPORTE AO PROCESSO DE INVENTARIAMENTO DE ESTOQUE

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2015

DANILO DAVID PEREIRA COSTA

APLICATIVO PARA DISPOSITÍVO MÓVEL ANDROID PARA SUPORTE AO PROCESSO DE INVENTARIAMENTO DE ESTOQUE

Trabalho de Conclusão de Curso, apresentado ao IV Curso de Especialização em Tecnologia Java, do Curso de Especialização em Tecnologia Java, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná, Campus Pato Branco, como requisito parcial para obtenção do título de Especialista.
Orientador: Prof. Robison Cris Brito

PATO BRANCO-PR

2015

APLICATIVO PARA DISPOSITIVO MÓVEL ANDROID PARA SUPORTE AO PROCESSO DE INVENTARIAMENTO DE ESTOQUE

Por

Danilo David Pereira Costa

Esta monografia foi apresentada às 16h30 do dia 28 de outubro de 2015 como requisito parcial para a obtenção do título de ESPECIALISTA, no III Curso de Especialização em Tecnologia Java, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. O acadêmico foi arguido pela Banca Examinadora composto pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Msc. Robison Cris Brito
Orientador
UTFPR – Campus Pato Branco

Prof. Msc. Rubia Eliza de Oliveira Schutz Ascari
Banca
UTFPR – Campus Pato Branco

Profa. Dr. Beatriz Borsoi
Banca
UTFPR – Campus Pato Branco

Prof. Msc. Robison Cris Brito
Coordenador do curso de Especialização
UTFPR – Campus Pato Branco

RESUMO

COSTA, Danilo David Pereira Costa.Aplicativo para dispositivo móvel Android para suporte ao processo de inventariamento de estoque. 2015. Monografia (Trabalho de Conclusão de Curso) – Curso de Especialização em Tecnologia Java, Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Pato Branco, 2015.

O propósito deste projeto, é desenvolver um aplicativo para dispositivos móveis, para auxiliar na coleta de dados para o processo de inventariamento de produtos dos clientes da empresa Dream Sistemas. Os clientes da empresa Dream Sistemas, são clientes que atuam no ramo farmacêutico e para conferência do seu estoque, precisam de uma ferramenta prática para auxiliar o processo. Com base na identificação do produto através do seu código de barras foi estudada a possibilidade da captura da imagem desse código e sua leitura através de uma aplicação Androide para dispositivos móveis. Para alcançar esse objetivo foi utilizada a biblioteca Zxing que permite utilizar a câmera do celular para capturar a imagem do código de barras do produto e retornar o código correspondente. Por meio da captura e leitura do código via celular, os clientes identificam os seus produtos e quantidades, que são armazenados e processados para auxiliar na conferência periódica do seu estoque.

Palavras-chave: Android, aplicativo, Java, móvel, inventarimento.

ABSTRACT

COSTA, Danilo David Pereira Costa. App to Android mobile device to give support at process of inventory. 2015. Monografia (Trabalho de Conclusão de Curso) – Curso de Especialização em Tecnologia Java, Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. Pato Branco, 2015.

The propose of this project, is to develop a app to mobile device to help in the data collect to the inventory of products process to the company Dream System's customers. The customers of the Dream Systems company act in the pharmaceutical industry and to check their inventory, they need of a practice tool to this process. In the base of the identification of products through code bar, it was studied the possibility do capture the image of this code, and read it through the android app to mobile device. To reach this objective, was utilized the Zxing lib that through the camera of the mobile phone, it capture the code bar image and return the code of the image. Through of the capture and the reading of the code through of the mobile phone, the customers will identify your products and quantities, it will be stored and process this data to help in the periodic check of their inventory.

Keywords: Android, app, Java, mobile, inventory.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE CÓDIGOS

LISTA DE SIGLAS

ADT	*Android Developer Tools*
ANVISA	*Agência Nacional de Vigilância Sanitária*
API	*Application Programming Interface*
EAN	*European Article Number*
GPS	*Global Positioning System*
IDE	*Integrated Development Environment*
SDK	*Source Development Kit*
SNGPC	*Sistema Nacional de Gerenciamento de Produtos Controlados*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
UML	*Unified Modeling Language*
USB	*Universal Serial Bus*

SUMÁRIO

1 INTRODUCAO

Este capítulo apresenta as considerações iniciais, objetivos e a justificativa pela qual foi proposto o desenvolvimento deste trabalho. As considerações iniciais abordam a perspectiva inicial e amplia a importância do projeto. Os objetivos estão classificados como objetivo geral, generalizando a finalidade do trabalho, e os específicos, os eventos para alcançar o objetivo geral. A justificativa, expressa a razão pela qual o trabalho foi sugerido. E no fechamento deste capítulo, é apresentada a estrutura a do trabalho.

1.1 CONSIDERAÇÕES INICIAIS

A acuracidade de armazenamento de produtos em uma empresa consiste em garantir a integridade de dados de armazenamento de produtos, presumindo possíveis quebras, perdas e extravios, resultando na disponibilidade dos produtos armazenados na empresa a serem comercializados. Este conceito pode ser atribuído à técnica do controle de estoque.

> Entende-se por estoque quaisquer quantidades de bens físicos que sejam conservados, de forma improdutiva, por algum intervalo de tempo; constituem estoques tanto os produtos acabados que aguardam venda ou despacho, como matérias-primas e componentes que aguardam utilização na produção. ” Controle de estoques: o lote econômico. (MOREIRA, 1993, p. 463).

O controle do fluxo de produtos em uma empresa exige minuciosa gestão e informatização para que a acuracidade dos seus itens em relação ao estoque físico e sistema seja garantida. A Figura 1 apresenta o modelo da dinâmica de estoque adaptado de Moreira (1993).

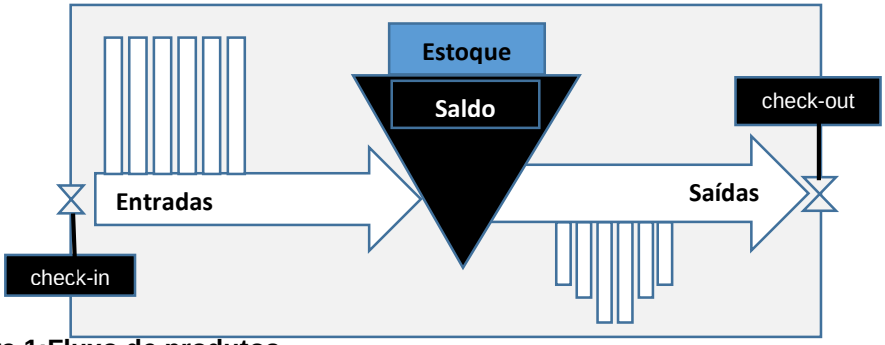

Figura 1:Fluxo de produtos

Fonte: Dinâmica de Estoque, adaptado de MOREIRA (1993, p. 23)

Segundo MOREIRA (1993), o fluxo de matérias é dividido em:

- Entradas: As entradas de materiais são transações que aumentam o saldo do item.
- Saídas: As saídas são transações que diminuem o saldo do item.

Saldo: O saldo é a quantidade disponível, sendo decorrência do saldo anterior, somadas as entradas e subtraídas as saídas no período.

Castiglione (2009), diz que a gestão de inventário é de extrema importância para a empresa para evitar possíveis desvios e garantir a disponibilidade dos estoques para o atendimento ao cliente final.

O processo de controle de inventário ou controle de estoque, aplicado no momento da venda, auxilia os gestores a analisar as divergências na acuracidade dos seus itens, para inspecionar o estoque contábil com o estoque físico para confrontar erros e acertos.

De acordo com DIAS (2008), as empresas vêm crescendo e se destacando cada vez mais, principalmente na cadeia de suprimentos. A grande quantidade de itens armazenados na empresa faz com que o processo de controle de estoque seja auxiliado por processos informatizados.

Há alguns anos, o processo de inspecionamento de estoque era feito manualmente devido à falta de recursos informatizados. O processo manual dependia de mão de obra para identificação do produto para a contagem do estoque.

Com a aplicação do código de barras, a definição desses padrões segundo CORONADO (2011), facilita na identificação e rastreabilidade de produtos de consumo, locais e serviços.

No mercado, dispositivos como leitores de código de barras são usados para scanear códigos de barras impressos a fim de identificar o produto por meio de fonte de luz infravermelho, para traduzir os impulsos ópticos em elétricos. O leitor contém um circuito decodificador para analisar os dados de imagem do código de barras fornecido pelo sensor e enviar o conteúdo do código de barras permitindo que este seja tratado como um número em um sistema computacional.

Em relação ao custo/benefício para compra de dispositivos ópticos específicos para empresas, torna-se um investimento alto conforme a quantidade adquirida que dependerá do tamanho da empresa. Pois empresas com grande fluxo de atendimentos e produtos, exige eficiência em seus processos.

Outras tecnologias podem ser utilizadas para capturar e processar o código de barras. A popularização dos dispositivos móveis como tablets e smatphones, permite aos usuários desses dispositivos o acesso a ferramentas como câmera, possibilitando ao usuário a captura e o armazenamento e processamento de imagens.

Com a tecnologia Android, é possível utilizar a biblioteca Zxing, que e a captura de imagem pela câmera do smartphone para implementar ao usuário uma aplicação, sem a utilização de dispositivos de leitores ópticos, a fim de o auxiliar no inventariamento de estoque.

1.2 OBJETIVOS

1.2.1 Objetivo geral

Desenvolvimento de um aplicativo para dispositivo móvel para plataforma Android que automatize o processo de inventariamento de estoque para os clientes da empresa de desenvolvimento de software Dream Sistemas.

1.2.2 Objetivos Específicos

- Definir layout do arquivo a ser criado pelo software da Pharma Manager para transferência dos itens para o sistema proposto;
- Utilizar biblioteca Zxing para capturar código de barras;
- Criar layout para o arquivo gerado pelo sistema proposto para importação no sistema da farmácia;
- Desenvolver aplicativo móvel;
- Integrar aplicativo móvel ao software Pharma Manager.

1.3 JUSTIFICATIVA

A proposta para o desenvolvimento do aplicativo de inventariamento tem como objetivo oferecer maior praticidade aos clientes da empresa Dream Sistema no processo de inventariamento. O aplicativo permite que funcionários utilizem smartphones para identificar os produtos nas prateleiras (usando a câmera destes dispositivos para scanear a imagem do código de barras), enviando estes dados para um servidor de aplicação, automatizando o processo de inventariamento.

A tecnologia Android foi escolhida devido a sua integração com aplicativos móveis como tablets e smartphone. Para garantir sua integração com os mais diferentes dispositivos (desde os mais antigos até os mais modernos), a plataforma é dividida em versões, como mostra a Figura 2.

Platform Versions.

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.1%
4.1.x	Jelly Bean	16	14.7%
4.2.x		17	17.5%
4.3		18	5.2%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	11.6%
5.1		22	0.8%

Data collected during a 7-day period ending on June 1, 2015.
Any versions with less than 0.1% distribution are not shown.

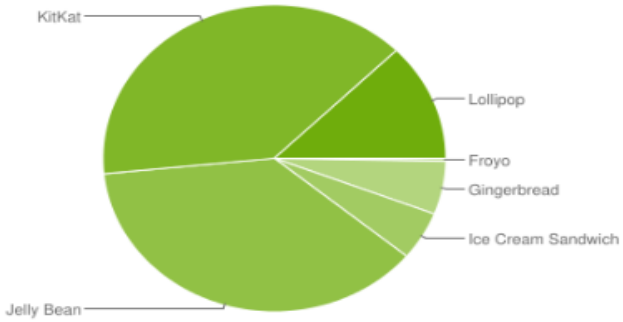

Figura 2:Dispositivos distribuídos por versão do Android
Fonte:http://developer.android.com, (2015)

A Figura 2 apresenta a constante atualização do Android. A coluna version apresenta a numeração de suas *releases* lançadas. O *codename*, como elas são chamadas. O conjunto de ferramentas que cada release abrange

chamou de API (*Application Programming Interface)*. Na coluna API segue a numeração respectiva de cada realize.

O processo de inventariamento realizado por meio de anotações além de tornar uma atividade exaustiva e demorada, pode acarretar a ocorrência de inconsistências no levantamento dos dados. O software sugerido fará com que o usuário tenha menos chances de correr erros na contagem de seu estoque.

Não será necessário o uso de leitores específicos como scanners para leitura dos códigos, pois a leitura será feita com as câmeras disponíveis em smartphones gerando uma grande economia, pois cada scanner pode custar cerca de R$ 800,00.

2 FUNDAMENTAÇÃO TEORICA

2.1. SOFTWARE PHARMA MANAGER

Com mais de quinze anos no mercado, a empresa de software Dream Sistemas, de Concórdia, Santa Catarina atua no desenvolvimento de sistemas para farmácias, sendo seu principal produto o Pharma Manager.

O Pharma Manager, é divido em dois módulos principais, o módulo de vendas e o retaguarda. No módulo de vendas, destaca-se todas os processos referentes a vendas e no retaguarda abrange cadastros, relatórios, processos administrativos e internos da farmácia.

Dentre as diversas funcionalidades do software, o sistema disponibiliza aos usuários o modulo de Inventariamento. O usuário neste módulo faz os ajustes das quantidades em estoque dos seus produtos estocados.

Figura 3: Tela de entrada do sistema Pharma Manager

Fonte: Dream Sistemas

Para realizar o inventariamento o usuário precisa logar-se no sistema instalado em um computador desktop, navegar pelo menu Entrada/ Saídas, opção Ajuste de Estoque.

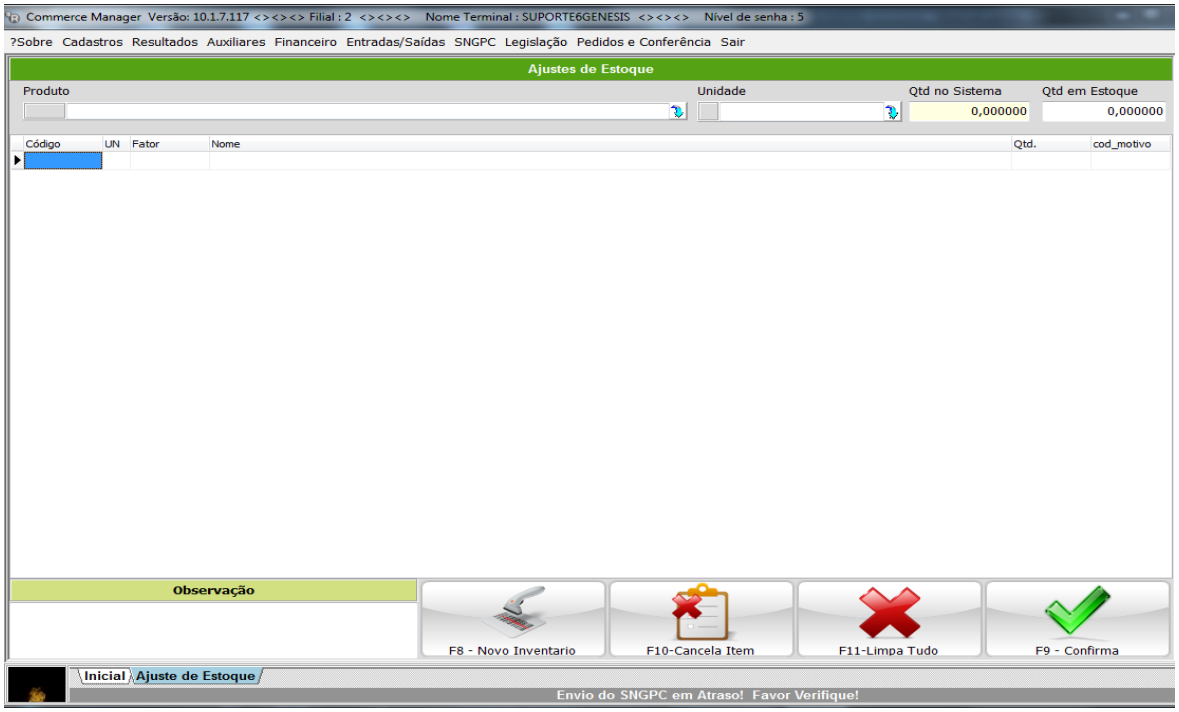

Figura 4: Tela de Ajuste de Estoque

Fonte: Dream Sistemas

Ao inicializar o inventário, o sistema apresenta uma grid em branco, indicando a abertura do novo inventário, como mostra a Figura 4. O usuário tem a opção de acrescentar os produtos, buscando o produto pelo código de cadastro, ou pela descrição ou pelo código de barras. O usuário pode informar a unidade de medida e a quantidade que contém eu seu estoque físico.

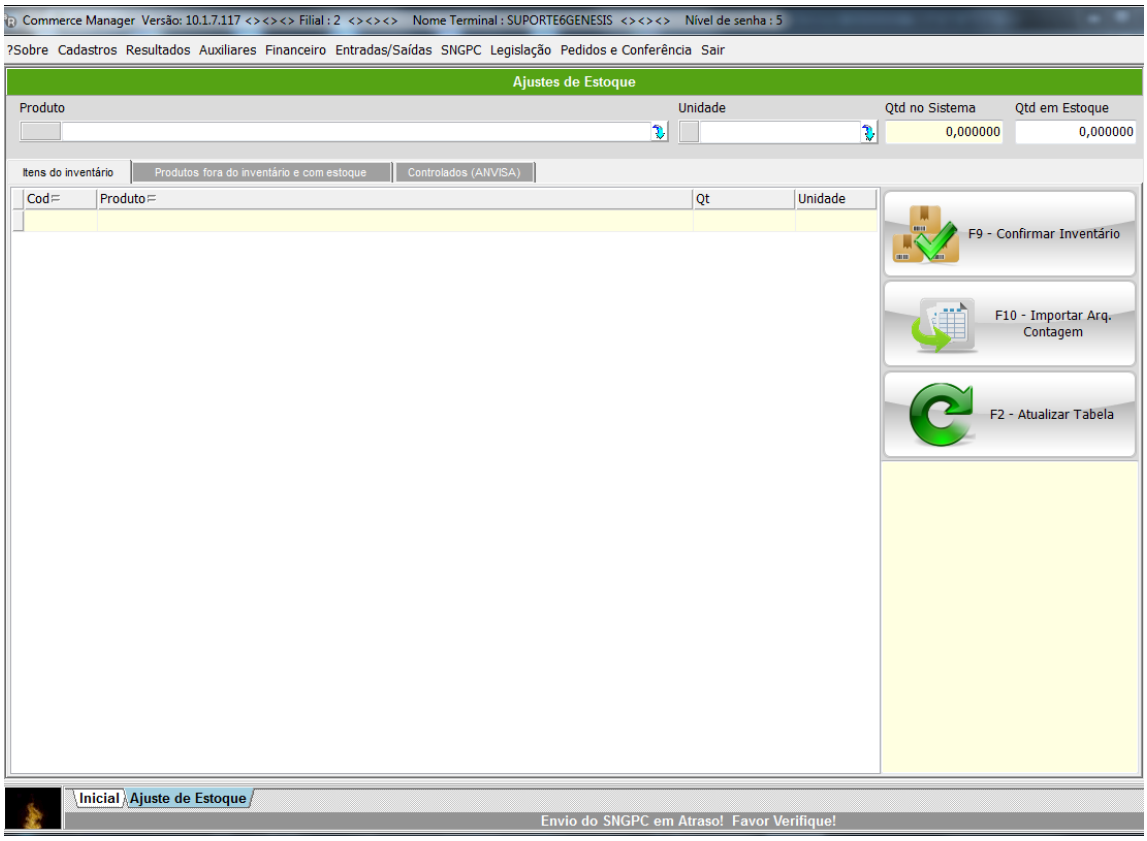

Figura 5: Tela de Inventariamento de estoque

Fonte: Dream Sistemas

Na aba Produtos fora do inventário com estoque, o sistema traz os itens com estoque positivo no sistema, como mostra a Figura 6. Em vez de informar o estoque, o usuário poderá marcar os produtos individualmente com um duplo clique e informar no inventário a mesma quantidade apresentada pelo sistema, ou poderá adicionar todos os itens com suas quantidades respectivas.

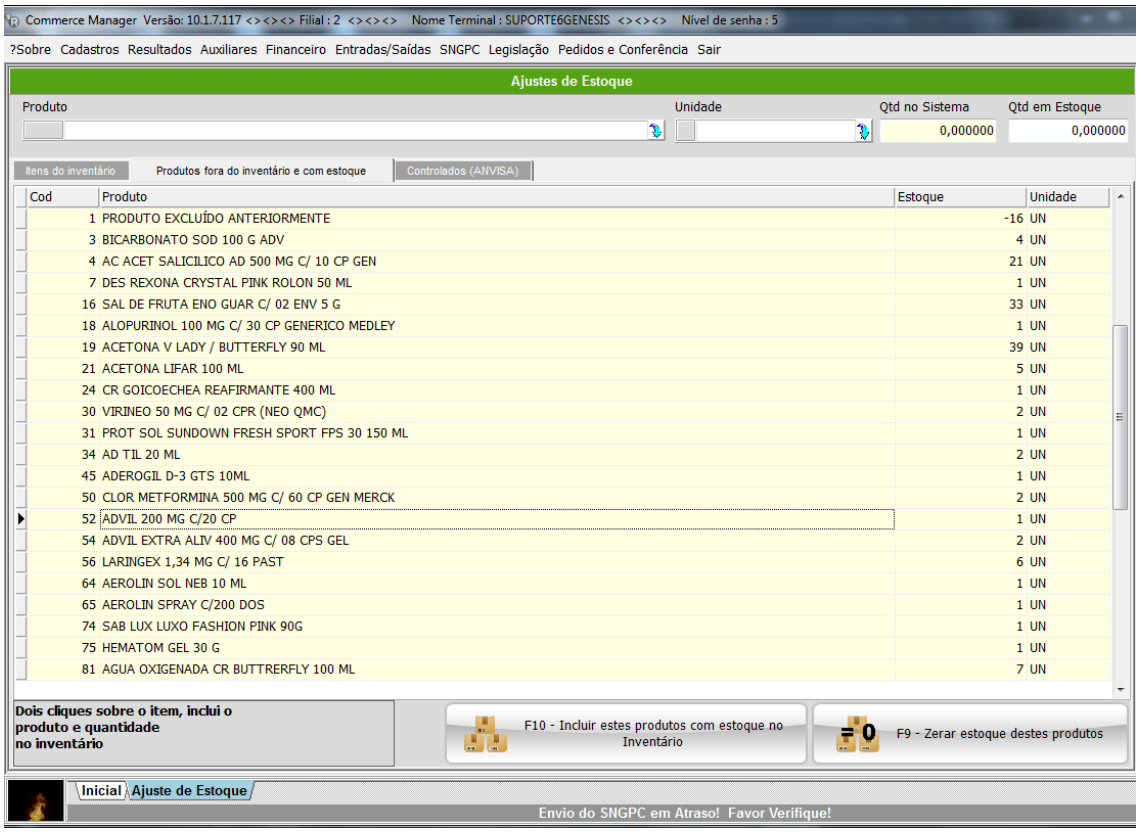

Figura 6: Aba Produtos fora do inventário e com estoque

Fonte: Dream Sistemas

Uma outra função no módulo de inventariamento, na aba Produtos fora do inventário e com estoque, permite ao usuário zerar o estoque de todos os itens, antes de adicioná-los no novo inventário.

A aba Controlados (ANVISA), são medicamentos que possuem controle especial, medicamentos controlados e antibióticos, monitorados pela vigilância sanitária. Esses itens somente podem sofrer alterações em seus estoques por meio de compra e venda. Para alterar o estoque desses itens somente por meio da abertura de inventario no SNGPC (*Sistema Nacional de Gerenciamento de Produto Controlado*).

No módulo de inventariamento no Pharma Manager, os medicamentos controlados e antibióticos, são apresentados na aba Controlados (ANVISA) somente para apresentar ao usuário a quantidade no estoque físico, não sendo possíveis alterações de estoque para esses itens.

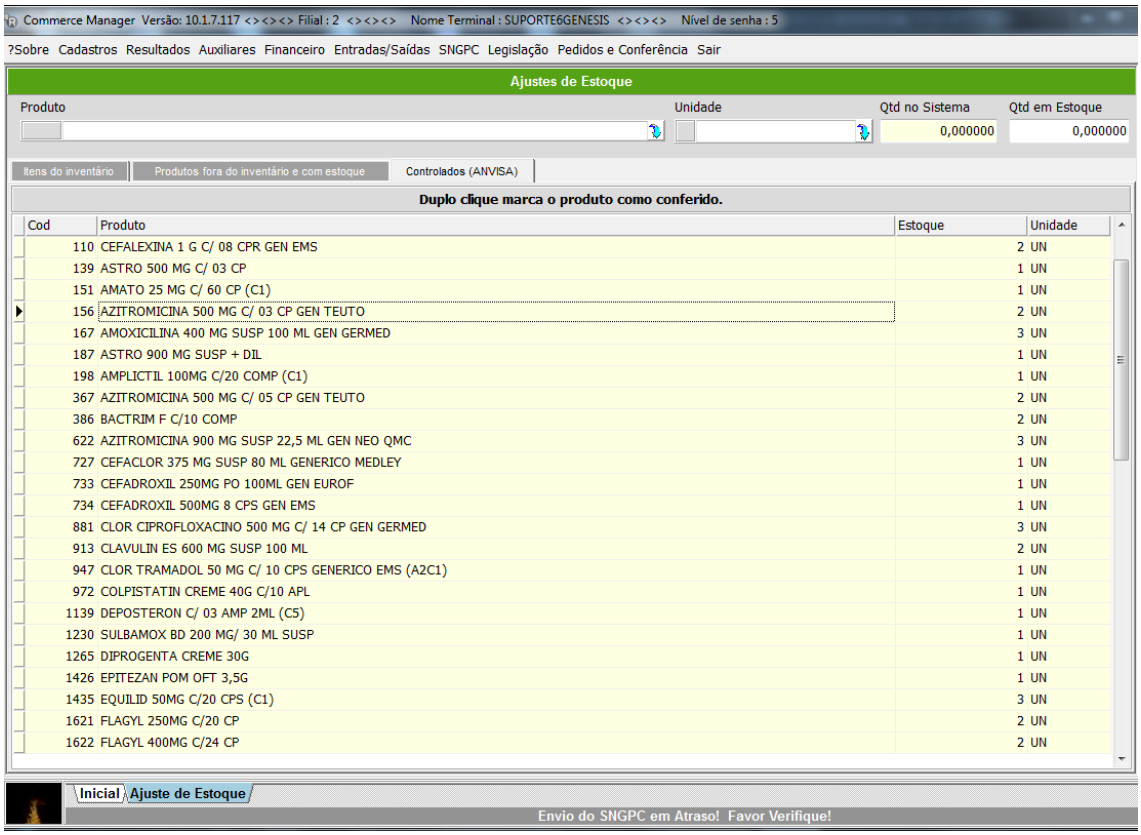

Figura 7: Aba Controlados (ANVISA)

Fonte: Dream Sistemas

A Figura 5, apresenta o botão F10 – Importar Arq. Contagem, ao qual busca o arquivo gerado pelo aplicativo proposto neste projeto para apresentar na tela os itens do inventário.

2.2 CÓDIGO DE BARRAS

O código de barras é uma representação gráfica de dados numéricos ou alfanuméricos a fim de promover a identificação e ou a rastreabilidade do objeto codificado.

O código de barras é um padrão utilizado para identificar produto de consumo exposto na gôndola para o consumidor final até as unidades logísticas, permitindo a rastreabilidade dos produtos (CORONADO, 2011).

Códigos de barra possibilitam que produtos sejam rastreados de maneira eficiente e precisa a taxas não possíveis usando sistemas manuais de entrada de dados (LIU, YANF, YANG, 2008).

Geralmente, a leitura de um código de barras é realizada pela decodificação de um scanner através da emissão de um raio infravermelho, absorvendo a luz das barras escuras e refletindo para o leitor.

O código de barras pode ser usado para aprimorar qualquer processo que envolva controle de mercadorias. Por suas próprias características, o sistema é ideal para operações com grande número de itens (ZYNGIER, 1991).

Zyngier (1991) cita que, de forma geral, o sistema de código de barras é a forma mais racional de gerenciamento do controle de fluxo e estoque de produtos acabados ou não.

Os códigos de barras são classificados pelas regras de simbologia. A simbologia define como os dados serão codificados. Para cada segmento de mercado, um código especifico é aplicado. Para segmento de varejo, é usado o padrão de código de barras EAN para identificação dos bens de consumo.

2.2.1 Código de barras padrão EAN

O padrão EAN/UPC é um padrão internacional usado no segmento varejista, para identificar itens vendidos, movimentados e armazenados.

Figura 8: Código de barras padrão EAN
Fonte:https://www.gs1br.org, (2015)

A Figura 8 apresenta um código de barras no padrão EAN. A estrutura do código apresenta as seguintes informações:

- Os três primeiros dígitos, como na Figura 1, 789, representam o país responsável pelo controle e licenciamento da numeração;
- Do quarto ao nono digito, identifica a empresa proprietária do prefixo. Como na Figura 1, os dígitos 835741;

- Do décimo ao décimo segundo dígitos, como na Figura 1, 001, representa a identificação do produto;
- O último dígito, é o dígito verificador.

O padrão EAN contém três estruturas de numeração padronizadas apropriadas para identificação de itens comerciais e unidades logísticas (CORONADO, 2011):

1. Código de barras EAN-13 - É a identificação básica e de uso geral para os produtos que recebem preço e podem ser comercializados na cadeia de suprimentos.

A estrutura EAN-13 é composta por 13 dígitos, que acompanham os produtos ou serviços, garantindo-lhes a exclusividade das numerações no mercado.

2. Código de barras EAN-8 é utilizado para identificação de produtos com tamanho reduzido. Exemplos: ampolas, cigarros etc.

3. Código de barras EAN-14 - É utilizado com freqüência em unidades logísticas - caixas, fardos, paletes, contêineres - com o objetivo de padronizar as quantidades de itens movimentadas em um processo logístico interno, facilitando o registro dos dados.

2.3 ANDROID

O Android é uma plataforma operacional em camadas baseado no Kernel Linux V2.6. Inicialmente, sua aplicação era voltada a celulares. Atualmente a plataforma Android tem se tornado robusto e estável para o desenvolvimento de uma gama de dispositivos móveis, incluindo interfaces ricas, tratamento de eventos e persistência de dados. A Figura 9, apresenta as camadas do software Android.

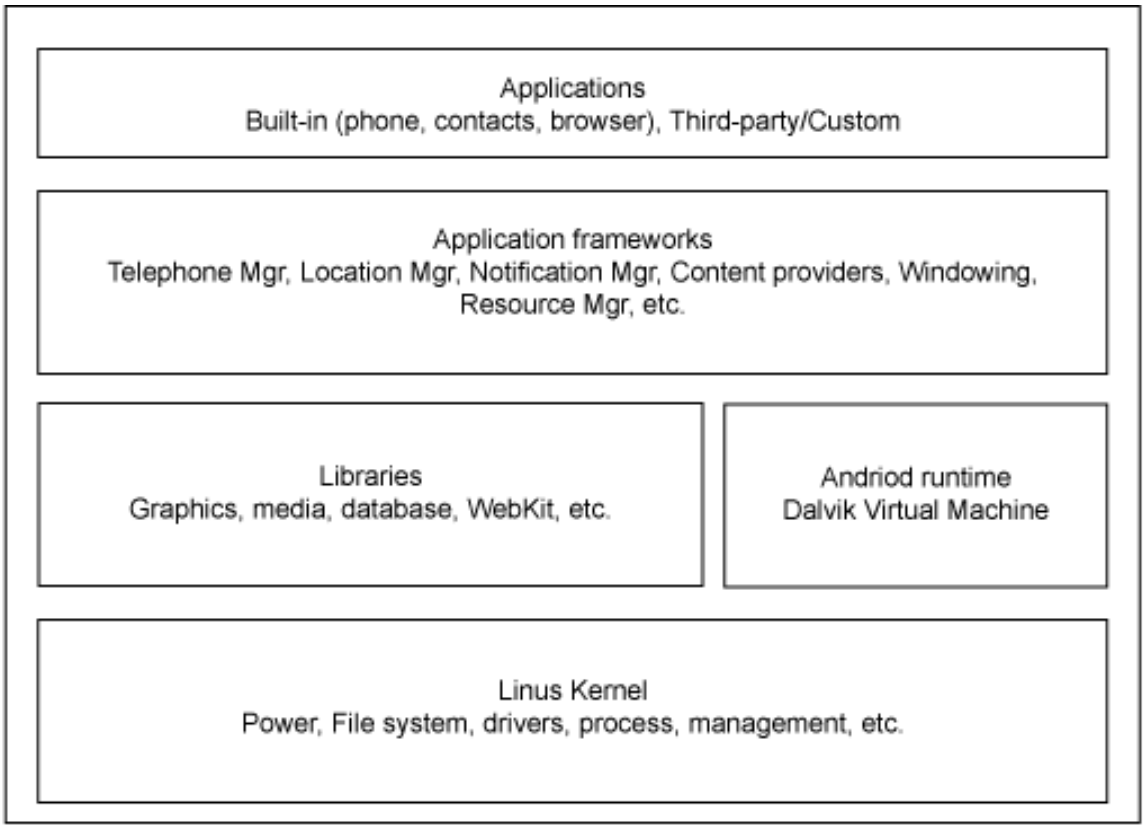

Figura 9: Camadas do software Android

Fonte: http://www.ibm.com, (2015)

A plataforma Android tem disponível várias opções de bibliotecas e frameworks que auxiliam no desenvolvimento de aplicativos e interatividade com os dispositivos de celular, como por exemplo a câmera; opções de conectividade, incluindo WiFi e Bluetooth; e estabelece interatividade com os serviços do Google, como por exemplo, Google Maps e Google Drive.

A grande aposta do Android são os novos aparelhos celulares, mais conhecidos como smartphones, celulares com grande poder de processamento que integram vários recursos, como alta conectividade com a Internet, GPS (*Global Positioning System*), sensores, telas sensíveis ao toque e etc. (OGLIARI, BRITO, 2014)

2.3.1 Zxing

Zxing abreviação de "zebra crossing" é uma biblioteca open-source, para leitura de código de barra multi-format 1D/2D implementada para linguagem java e outras linguagens.

A finalidade da biblioteca Zxing é a captura do código de barras como uma imagem por meio de uma câmera e a leitura e reconhecimento do padrão utilizado. A Figura 10, apresenta os tipos de códigos suportados pela biblioteca Zxing.

1D product	1D industrial	2D
UPC-A	Code 39	QR Code
UPC-E	Code 93	Data Matrix
EAN-8	Code 128	Aztec (beta)
EAN-13	Codabar	PDF 417 (beta)
	ITF	
	RSS-14	
	RSS-Expanded	

Figura 10: Tipo de código de barras suportadas pela biblioteca Zxing
Fonte:https://github.com, (2015)

3 MATERIAIS E METODOS

O presente capítulo apresenta os materiais e métodos definidos para o desenvolvimento do sistema proposto.

Os materiais definem as tecnologias, linguagem de programação, IDE (*Integrated Development Environment*) e ferramentas utilizadas. Enquanto os métodos definem os procedimentos para o projeto, implementação e desenvolvimento da aplicação.

3.1 MATERIAIS

A linguagem utilizada para o desenvolvimento do aplicativo foi o Java. A IDE para o desenvolvimento foi a IDE Eclipse Luna, utilizada como ambiente de programação. Foi utilizado o Android SDK (*Source Development Kit*), pacote necessário para o desenvolvimento de aplicativos para plataforma Android. Para captura e leitura do código de barras pela câmera do celular, foi utilizada a biblioteca Zxing. Para os testes do sistema foi utilizada a depuração via USB (*Universal Serial Bus*) no smartphone LG-D325f8. A Figura 11 apresenta o modelo do smartphone utilizado para os testes.

Figura 11: Smartphone LG-D325f8

As tecnologias e ferramentas utilizadas no desenvolvimento do projeto estão apresentadas no Quadro 1:

Nome	URL	Valor	Utilização
Eclipse IDE	www.eclipse.org/	Sem Custo	IDE para ambiente de programação
Android SDK	https://developer.android.com/sdk	Sem Custo	Pacote necessário para desenvolver aplicativos para plataforma Android
Plugin ADT	http://developer.android.com/tools/sdk/eclipse-adt.html	Sem Custo	Plugin utilizado para desenvolvimento Android no Eclipse IDE
Zxing	https://code.google.com/p/tesseract-ocr/	Sem Custo	API para captura de código de barras pela câmera do smatphone

Quadro 1: Ferramentas utilizadas para o projeto

A IDE Eclipse é um ambiente de programação utilizado para diversas linguagens, destaca-se no desenvolvimento para aplicações Java. A IDE é extensiva, composta por um conjunto de serviços para o desenvolvimento de aplicativos. Tornou-se umas das mais importantes ferramentas utilizadas para o desenvolvimento de aplicativos para plataforma Android.

O Android SDK inclui as API’s e as ferramentas necessárias para o desenvolvimento, debug e testes de código para linguagem Java. Para usar o SDK é necessário importá-la na IDE usada para o desenvolvimento.

ADT (*Android Developer Tools*) é um plugin para Eclipse que provê ferramentas que são integradas com o Eclipse IDE, oferecendo recursos para aplicações Android. ADT disponibiliza componentes visuais para construção de interfaces.

A biblioteca Zxing, permite captura e leitura de código de barras por meio da captura da imagem do código de barras pela câmera do smatphone sem a utilização de um leitor óptico.

3.2 MÉTODOS

Para conhecer os objetivos de um sistema é necessário analisar as suas características e funcionalidades, ou seja, ressaltar seus requisitos, que segundo

Sommerville (1995, p. 118), “os definem como uma descrição abstrata dos serviços que o sistema deve prover, independendo da situação a qual deve manter seu funcionamento.

“Requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar, para atingir os seus objetivos. ” (PFLEGGER, 2004, p. 111).

Um processo importante na análise dos requisitos é a documentação dos mesmos, pois a documentação facilita futuras manutenções, e até mesmo na busca de erros e soluções de problemas no sistema.

Para o levantamento dos requisitos do aplicativo, foram levantados os requisitos funcionais e não funcionais.

“Requisitos funcionais são necessidades de informação do usuário final que são vinculadas aos recursos de hardware, software, rede, dados e humanos que os usuários finais que presentemente utilizam ou poderão utilizar no novo sistema. ” (O’BRIEN, 2004, p. 332).

Os requisitos não funcionais segundo PFleeger (2004) descrevem uma restrição no sistema que limita a criação para solução do problema, ou seja, as mínimas configurações para o funcionamento do sistema.

Após o levantamento dos requisitos, foi desenvolvido um diagrama UML de implementação a fim de apresentar a organização da estrutura física do funcionamento do sistema, mostrando os objetos e a comunicação entre eles.

A comunicação entre o dispositivo móvel e o computador do usuário será realizada via socket TCP/IP (*Transmission Control Protocol/Internet Protocol*) para troca de informações via arquivo texto. Esse vai ser utilizado para transmitir os itens do sistema de vendas da farmácia e transmitir o inventário com os dados capturados pelo usuário via smartphone.

O desenvolvimento do aplicativo se deu a partir do desenho de suas telas e implementação de suas funcionalidades a partir dos requisitos levantados.

Os testes foram realizados simultaneamente com o desenvolvimento.

4 RESULTADOS

O presente capítulo apresenta a aplicação dos métodos informados no capitulo três para o desenvolvimento do sistema proposto.

4.1 APRESENTAÇÃO E ANÁLISE DE DADOS

Este capítulo apresenta uma breve análise das principais funcionalidades do aplicativo, dividida em requisitos funcionais, não funcionais e o diagrama UML de implementação.

4.1.1 Requisitos Funcionais

Os requisitos funcionais têm como finalidade levantar as funcionalidades do sistema a fim de alcançar os resultados esperados. O Quadro 1 apresenta os requisitos funcionais do sistema.

Requisitos	Descrição	Observações
Download e atualização da lista de itens do cliente	O usuário fará o download dos itens cadastros no sistema da farmácia e será armazenado no banco do sistema de inventariamento para que ao informar um código de barras o sistema faça a pesquisa desse item, não permitindo ao usuário informar itens que não estejam cadastrados no sistema da farmácia.	O sistema da farmácia irá gerar o arquivo conforme layout designado
Apresentar itens adicionados no inventário	O sistema apresentará os itens adicionados no inventário pelo usuário em forma de lista, agrupando por código de barras, apresentando a	Se o usuário informar o mesmo item mais que uma vez, o sistema apresentará na lista o registro com a soma de quantidades informadas pelo usuário

	quantidade de itens adicionados	
Exclusão de itens do inventário	O sistema permitirá excluir itens individuais da lista do inventário, apresentando mensagem de confirmação da operação	O sistema não permitirá edição da quantidade do registro. O usuário terá que excluir e informar novamente sua quantidade correta
Exclusão do inventário	O usuário poderá excluir todo o inventário.	Opção estará no menu oculto, por questão de segurança ao usuário.
Configuração de quantidade automática	O sistema permitirá ao usuário informar a quantidade do estoque do item, ou o sistema informará a quantidade de 1 unidade ao ler através da câmera o código de barras	Tanto pela digitação ou pela quantidade automática o sistema irá somar os itens informados
Ler código de barras do item por meio da câmera	O sistema irá apresentar o código de barras dos itens por meio da captura de imagem da câmera do smartphone e fará pesquisa do item através da tabela dos itens armazenados em banco	
Upload de inventário	O sistema irá transferir o inventário armazenado em banco para o ftp em arquivo texto conforme layout definido	

Quadro 2: Requisitos Funcionais

4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais são requisitos relacionados ao uso da aplicação em relação ao desempenho, da usabilidade, confiabilidade, segurança, disponibilidade, manutenção e recursos envolvidos.

Requisito	Descrição
Implementação 1	O sistema e o banco de dados deverão apresentar suporte a plataforma Android
Implementação 2	A implementação da leitura de código de barras se dará pela câmera do smatphone
Implementação 3	A troca de informação do sistema de inventariamento e do sistema de gestão do usuário se dará por arquivo texto via socket TCP/IP
Usabilidade 1	Apresentar telas com funções objetivas evitando aglomeração de componentes.
Usabilidade 2	Em telas que apresentar lista disponibilizar ao usuário a opção de pesquisa.
Confiabilidade 1	Apresentar mensagens de confirmação de ações aos usuários.
Segurança 1	Apresentar caixa de diálogo em ações que dependem de confirmação do usuário.

Quadro 3: Requisitos não funcionais

4.1.3 Diagrama de Implementação

O Diagrama de Implementação tem como objetivo representar a organização da estrutura física para o funcionamento do sistema. Nele são apresentadas máquinas, protocolo de comunicação, transmissão de dados, e etc. A Figura 12 representa o Diagrama de Implementação do sistema.

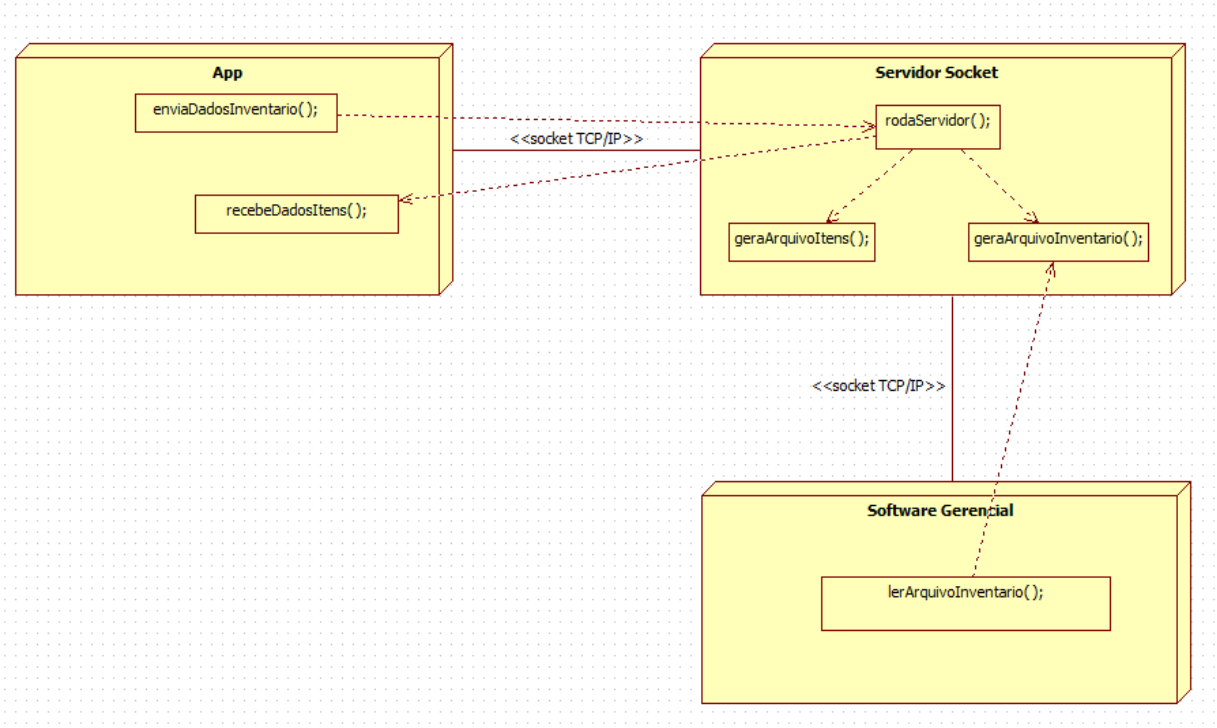

Figura 12: Diagrama de Implementação

O objeto App indica a aplicação móvel. Onde nele se encontram pacotes, classes e bibliotecas do sistema. Neste objeto, é apresentado somente os métodos que interagem com os componentes externos do sistema.

O objeto Software Gerencial, apresenta o Pharma Manager, o aplicativo de gerenciamento da farmácia. No aplicativo Pharma Manager, quando o cliente acessa a tela de inventariamento, e clica no botão novo inventário, será executado o aplicativo Servidor Socket que abrirá uma conexão socket no computador do cliente na porta especificada para a comunicação com o smatphone.

O aplicativo Servidor Socket terá o método rodaServidor () responsável por iniciar o servidor socket e receber e enviar os dados para a aplicação Android.

Dois métodos principais estão apresentados na Figura 1, o método geraArquivoItens() e geraArquivoInventario().

O método geraArquivoItens(), é o método responsável por conectar a base de dados do cliente e extrair os dados de código e descrição dos itens do

sistema da farmácia gerando o arquivo itens.txt em C:\Genesis\Commerce\Manutencao.

O método geraArquivoInventario(), lê a variável com os dados recebidos da aplicação Android pelo método rodaServidor(), e gera o arquivo inv.txt na máquina do cliente no local C:\Genesis\Commerce\Manutenção. O arquivo inv.txt possui os dados armazenados pelo cliente por meio da aplicação Android.

4.2 DESENVOLVIMENTO DO APLICATIVO

O levantamento dos requisitos do sistema formou um conjunto de finalidades básicas e generalizadas a fim de dirigir o desenvolvimento do sistema proposto em tempo para alcançar o objetivo deste projeto.

O desenvolvimento do sistema para a plataforma Android foi realizado utilizando a IDE Eclipse Luna, release 4.4.0. A Figura 13, apresenta a estrutura do projeto do sistema proposto.

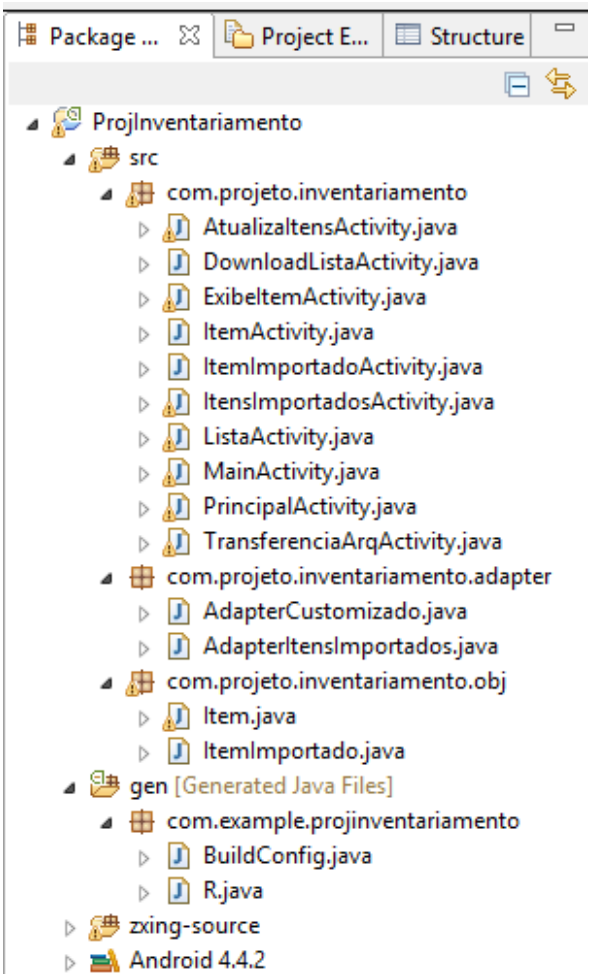

Figura 13: Estrutura do projeto

O projeto está divido em três pacotes. O pacote inventariamento, onde estão as activities, e onde está programada toda a ação das view's do sistema. O pacote adapter, onde está o código java para implantação da tela de listagem dos objetos e no pacote obj estão os objetos básicos do sistema.

A classe MainActivity, é a classe responsável pela integração com a API Zwing. Para a captura do código de barras por meio da imagem captura pela câmera do smartphone. A Figura 14 apresenta a estrutura dos objetos importados da API Zwing.

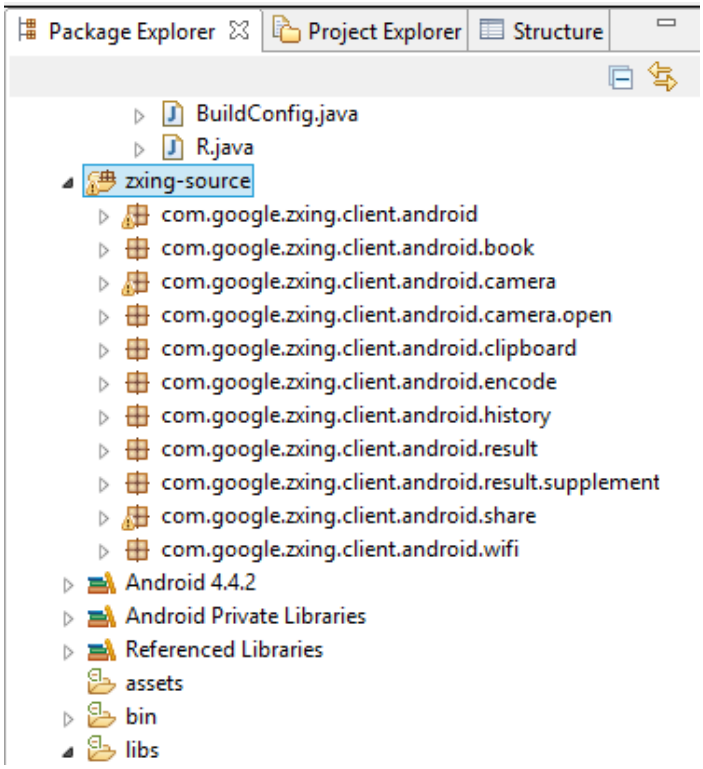

Figura 14: Pacote zwing-source

A Figura 14 apresenta a estrutura dos pacotes da API Zxing. Nesses pacotes estão as classes responsáveis pelo funcionamento da API.

A classe CaptureActivity, é a classe responsável pelo código de captura e leitura da imagem do código de barra dos itens localizada no pacote Android do pacote principal swing-source.

O desenvolvimento das telas do aplicativo para a interatividade com o usuário foram criadas pelo plugin ADT disponível na IDE, que disponibiliza componentes gráficos para criação. A Figura 15, mostra o funcionamento do ambiente com o plugin ADT.

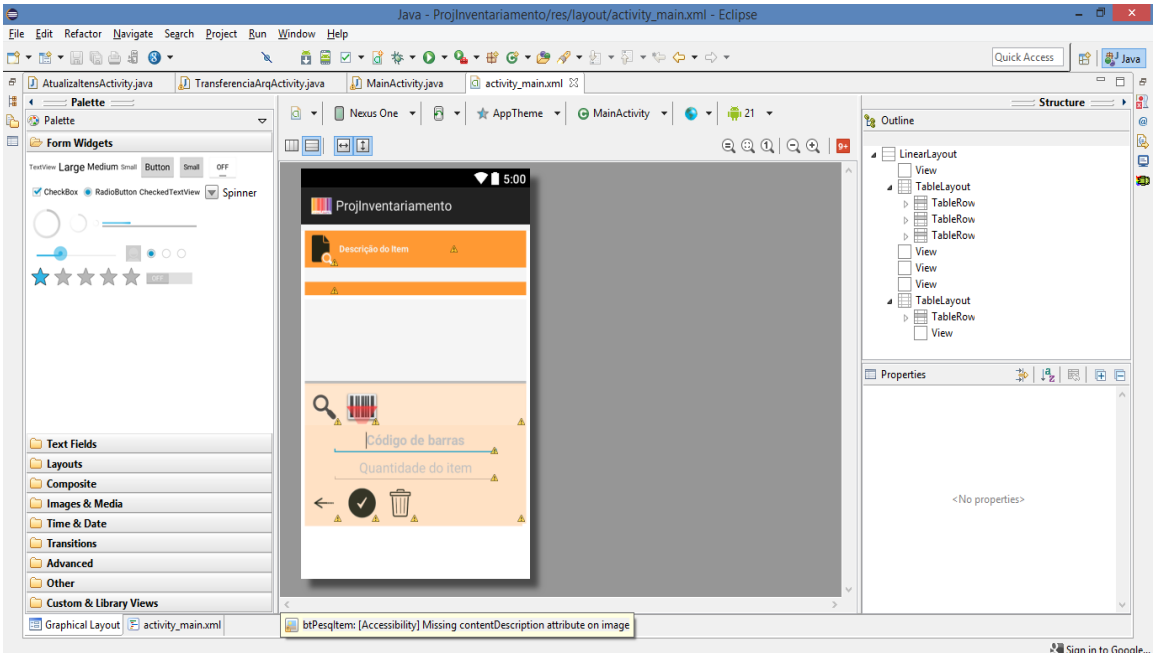

Figura 15: Ambiente de desenvolvimento

4.2.1 Telas do aplicativo

Ao executar o aplicativo, o sistema carrega uma tela de apresentação e na seqüência são apresentados os itens do inventário adicionados pelo usuário. A Figura 16 apresenta a tela do inventário. A tela do inventario, mostra o código de barras do item, sua descrição e quantidade do item em estoque. Dados esses informados pelo usuário.

Figura 16: Tela Itens do inventário.

Antes de adicionar um item ao inventário, o usuário terá que importar os itens cadastrados no sistema Pharma Manager, o sistema de gestão da

farmácia, no aplicativo. Para fazer o download, o usuário irá acessar a opção do menu Download Lista Itens. A Figura 17, apresenta o menu da tela Itens do inventário.

Figura 17: Menu da tela de itens do inventário

Na tela de download dos itens, apresentada pela Figura 18, o usuário irá informar o IP (*Internet Protocol*) de um dos computadores onde o sistema da farmácia está instalado. Via socket TCP/IP, o aplicativo irá fazer o download do arquivo itens.txt, localizado em C:\Genesis\Commerce\Aplicativos, armazenará no banco sqLite do próprio dispositivo Android.

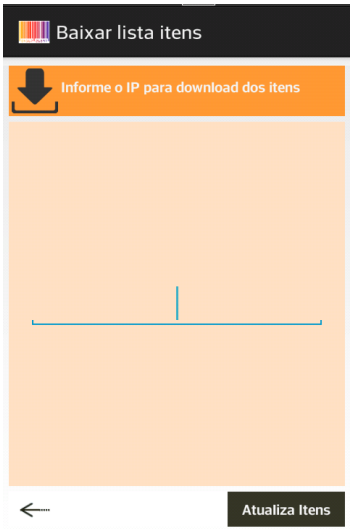

Figura 18: Tela de Download dos itens do Pharma Manager.

Depois de realizar o download, o usuário poderá visualizar os itens importados. A opção Ver Itens Impostado do menu da tela de itens do inventario irá apresentar ao usuário os itens que foram importados. O usuário poderá pesquisar pelos itens por meio da digitação do código de barras. A Figura 19 mostra a tela dos itens importados.

Figura 19: Tela itens importados

O usuário poderá pesquisar um item já informado no seu inventario, pesquisando por meio da digitação do código de barras e pressionando o botão apresentado pela lupa.

Para alterar a quantidade, o usuário terá que excluir o item, pressionando na tela a linha do item a ser excluído. A aplicação irá apresentar a tela de confirmação de exclusão de registro a fim de garantir a segurança ao usuário. A Figura 20mostra a mensagem de exclusão do item.

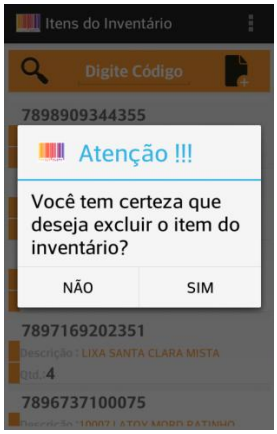

Figura 20: Exclusão de itens do inventário.

O botão direito da tela de itens do inventário, permite ao usuário adicionar um item ao seu inventário. A Figura 21 apresenta a tela de inserção de itens no inventário.

Figura 21: Tela de inserção de itens do inventario

Para adicionar um item, o usuário poderá informar o código de barras pela digitação do código ou captura do código pela câmera. O usuário irá digitar o código, caso ocorra algum problema com leitura da imagem, código de barras manchado, embalagem do item pode dificultar a leitura. O usuário irá pressionar o botão apresentado por uma lupa para a pesquisa do item.

Para busca pelo código de barras, o usuário irá pressionar o botão do código de barras, onde a aplicação vai abrir o visor da câmera para leitura e captura do código de barras. A Figura 22 apresenta a tela do visor da câmera.

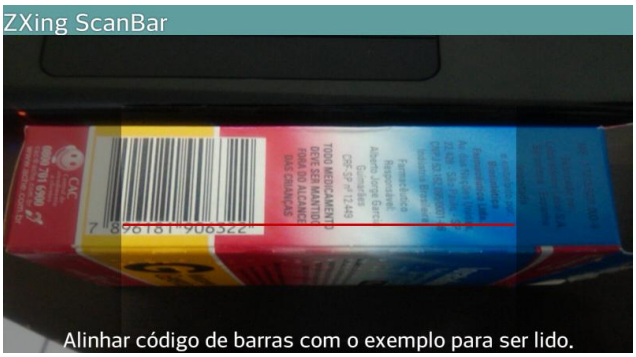

Figura 22: Tela de captura do código de barras

Ao informar o código válido, o sistema informará automaticamente a quantidade 1 no campo quantidade do item, possibilitando também que o usuário digite a quantidade de itens do código respectivo. Caso o usuário informe o mesmo item, o sistema irá agrupar o item apresentando na tela de itens do inventário a soma das quantidades informas.

A aplicação irá capturar o código de barras através da API Zwing, e irá apresentar a descrição do item. A cada código informado, o aplicativo soma o total dos códigos correspondentes.

Após concluído o inventário, o cliente irá acessar a opção Transferir Inventário do menu da tela dos itens do inventário e irá informar o IP do computador onde pretende salvar o arquivo. O aplicativo irá fazer o upload do arquivo inv.txt em C:\Genesis\Commerce\Aplicativos. A Figura 23 apresenta a tela de transferência do inventário.

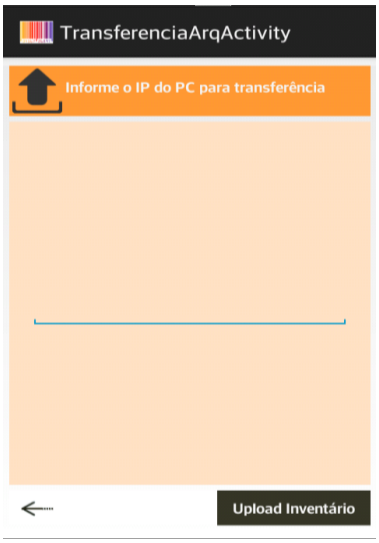

Figura 23: Tela de Upload do arquivo do inventario

4.2.2 Implementação Zxing

Para implementação da biblioteca Zxing, foi importado no projeto o diretório zxing-source, contendo todos os pacotes e respectivas classes responsáveis para implementar suas funcionalidades. A Figura 24 apresenta o diretório zxing-source importado no projeto e seus pacotes.

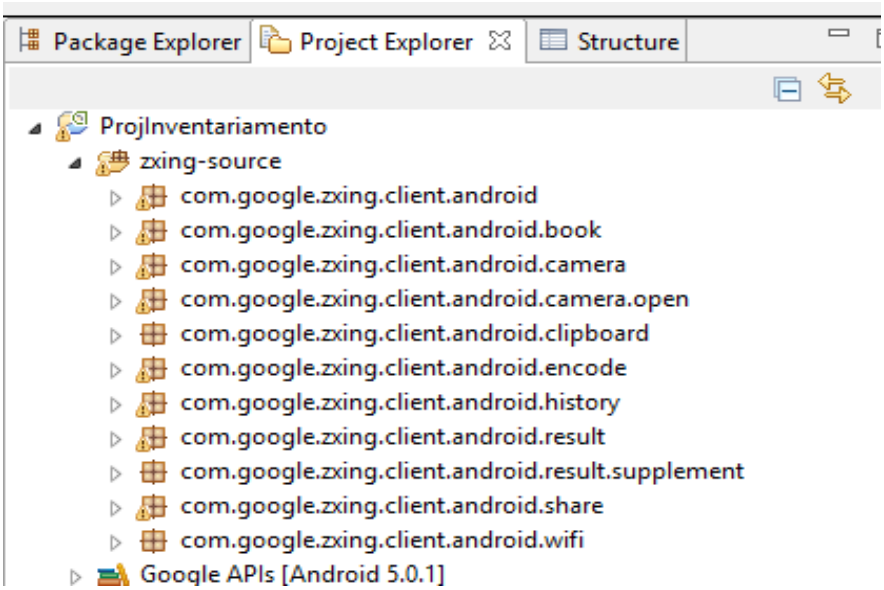

Figura 24: Pacote zxing-source

Para implementar a chamada da tela responsável pela captura do código de barras, foi necessário fazer alteração na classe com.google.zxing.client.androi.CaptureActivity, no método handleDecode(). A Listagem 1 apresenta o métodohandleDecode.

```
public void handleDecode(Resultraw Result, Bitmap barcode, float scaleFactor) {

Intentit = getIntent();
it.putExtra("SCAN_RESULT", rawResult.getText());
it.putExtra("SCAN_FORMAT", rawResult.getBarcodeFormat().toString());
setResult(Activity.RESULT_OK, it);
finish();

inactivityTimer.onActivity();
lastResult = rawResult;
ResultHandlerresultHandler = ResultHandlerFactory.makeResultHandler(this, rawResult);

booleanfromLiveScan = barcode != null;
if (fromLiveScan) {
        historyManager.addHistoryItem(rawResult, resultHandler);
        beepManager.playBeepSoundAndVibrate();
        drawResultPoints(barcode, scaleFactor, rawResult);
  }

switch (source) {
case NATIVE_APP_INTENT:
        case PRODUCT_SEARCH_LINK:
        handleDecodeExternally(rawResult, resultHandler, barcode);
        break;
case ZXING_LINK:
if (scanFromWebPageManager == null || !scanFromWebPageManager.isScanFromWebPage()) {
handleDecodeInternally(rawResult, resultHandler, barcode);
} else {
handleDecodeExternally(rawResult, resultHandler, barcode);
}
```

```
break;
case NONE:
SharedPreferencesprefs = PreferenceManager.getDefaultSharedPreferences(this);
if (fromLiveScan&&prefs.getBoolean(PreferencesActivity.KEY_BULK_MODE, false)) {
Toast.makeText(getApplicationContext(),
getResources().getString(R.string.msg_bulk_mode_scanned) + " (" + rawResult.getText() + ')',
Toast.LENGTH_SHORT).show();
restartPreviewAfterDelay(BULK_MODE_SCAN_DELAY_MS);
      } else {
handleDecodeInternally(rawResult, resultHandler, barcode);
      }
break;
   }
  }
```

Listagem 1: MétodohandleDecode da classe CaptureActivity.java

Por sua vez, a classe MainActivity.java, aciona a tela de captura do código de barras, através da ação onclick do botão btPesqItem, pelo método callZXing. O método onActivityResult captura o código de barras retornado como uma String pelo método handleCode da classe CaptureActiviy. A Listagem 2, apresenta os métodos onActivityResult e callZXing, da classe MainActivity.java.

```
private void onActivityResult(int requestCode, int resultCode, Intent data){
            if (REQUEST_CODE == requestCode&&RESULT_OK == resultCode) {
            edCodigoBarras.setText(data.getStringExtra("SCAN_RESULT"));
            edtQtdItem.requestFocus();tvDescItem.setText(retornaDescIten(edCodigoBarras.getText().toString()));
            }
}
publicvoidcallZXing(View v) {
            Intent i = newIntent(MainActivity.this,
            com.google.zxing.client.android.CaptureActivity.class);
            startActivityForResult(i, REQUEST_CODE);
}
```

Listagem 2: Método onActivityResult e callZXing da classe MainActivity

4.2.3 Layout arquivo de importação dos itens

Para evitar que o usuário adicione itens no inventário que não estejam no cadastro do sistema da farmácia, o aplicativo armazenará o código de barras e descrição do item através de socket TCP/IP. Será gerado o arquivo itens.txt em C:\Genesis\Commerce\Aplicativos. Os dados dos itens nesse arquivo serão enviados pelo aplicativo via socket. A Figura 25, apresenta o formato do arquivo itens.txt.

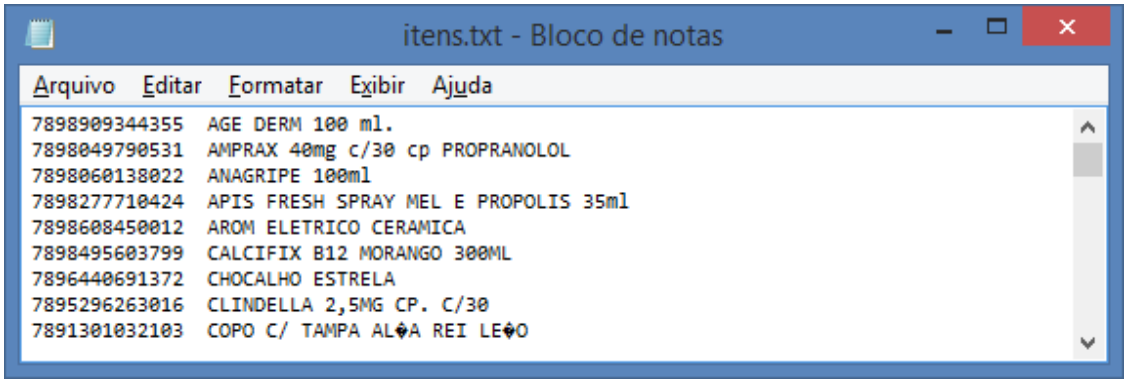

Figura 25: Arquivo itens.txt

O layout do arquivo compreende as 13 primeiras posições para o código de barras e a descrição do item. Os dados são separados por um espaço.

O método responsável por receber e armazenar os itens recebidos do arquivo itens.txt, está implementado na classe AtualizaItensActivity.java, o método recebeDadosItens (). A Listagem 3, apresenta o código do método.

```
public void recebeDadosItens()  {
                        itensImportados.clear();
                        dialog = ProgressDialog.show(AtualizaItensActivity.this,"MyInv.app","Recebendo Dados . .
.", false, true);
            dialog.setCancelable(false);
              Thread t = new Thread() {
        @Override
publicvoidrun() {
try {
            Socket cliente = new  Socket(edtIPDownload.getText().toString(),7999);
            ObjectInputStream entrada = new     ObjectInputStream(cliente.getInputStream());
                        StringdadosRecebidos = (String) entrada.readObject();
                        Typetype = new TypeToken<List<ItemImportado>>(){}.getType();
                        itensImportados = new Gson().fromJson(dadosRecebidos, type);

                        armazenaItensBD();
                        geraArquivoItensTxt();

                        dialog.dismiss();
                        entrada.close();
                        cliente.close();
            } catch (Exception e) {
            msgErroThread = "AtualizaItensActivity.recebeDadosItens!"+e.getMessage();
            }

        }
              };

            t.start();
   }
```

Listagem 3: Método recebeDadosItens () da classe AtualizaItensActivity

4.2.4 Layout arquivo de importação do inventario

Os dados do inventário, ou seja, o código de barras e a quantidade dos itens, que são dados informados pelo usuário e armazenados no aplicativo, são enviados via socket TCP/IP ao computador do usuário e para gerar o arquivo inv.txt. O arquivo é alocado em C:\Genesis\Commerce\Aplicativos. A Figura 26, apresenta o layout do arquivo inv.txt.

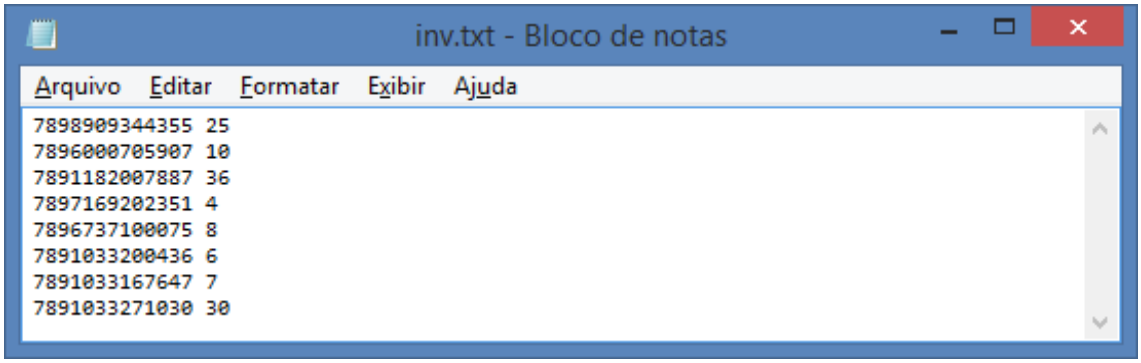

Figura 26: Arquivo inv.txt.

O layout do arquivo inv.txt, é formado pelo código de barras e a quantidade do item. Os dados são separados por espaço.

O método responsável pelo envio dos dados do inventário, está implementado na classe TransferenciaArqActivity.java, o método enviaDadosInventario (). A Listagem 4, apresenta o método codificado.

```
public void enviaDadosInventario()  {
                        dialog = ProgressDialog.show(TransferenciaArqActivity.this,"MyInv.app","Enviando Dados .
. .", false, true);
            dialog.setCancelable(false);
            new Thread() {
          @Override
publicvoidrun() {
try {
            Socket cliente = new Socket(edtIPUpload.getText().toString(),7999);
            ObjectOutputStreamsaida = new ObjectOutputStream(cliente.getOutputStream());
            StringjsonItens = new Gson().toJson(itens);
            saida.flush();
            saida.writeObject(jsonItens);
                        dialog.dismiss();
                        saida.close();
            } catch (Exception e) {
            System.out.println("Erro >>>>>"+e.getMessage());  }
          }
}.start();
if (!msgErroThread.isEmpty()) {
            Toast.makeText(this, msgErroThread, Toast.LENGTH_LONG).show();
      }
```

```
}
```

Listagem 4: MétodoenviaDadosInventario () da classe TransferenciaArqActivity

4.2.5 Servidor socket

O software servidor, ao ser executado implementa um servidor socket, na porta 7999, aguardando as requisições vindas do dispositivo Android através da rede interna.

O objetivo deste servidor é basicamente executar as requisições, processando-as, sendo que este processamento pode ser o envio da lista do software desktop do usuário, ou o resultado dos produtos lançados via dispositivo Android, o inventário. A Figura 27 representa a interface gráfica do servidor.

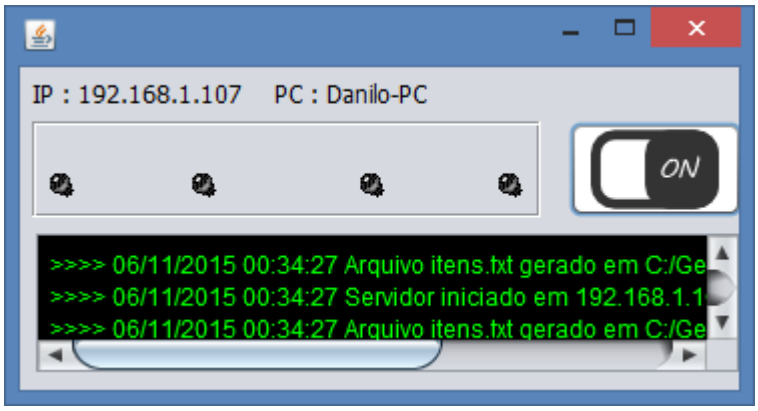

Figura 27: Interface gráfica do servidor socket.

4.2.6 Integração

Após capturar as informações do inventário por meio do aplicativo móvel, o usuário irá transferir os dados via socket TCP/IP para o computador desktop indicado pelo usuário. Será gerado um arquivo texto, inv.txt, que é alocado em C:\Genesis\Commerce\Aplicativos.

Para carregar o arquivo, o usuário irá logar-se ao Pharma Manager, acessar a tela do inventário pelo menu Entrada/Saídas opção Ajuste de Estoque, como mostrar a Figura 28.

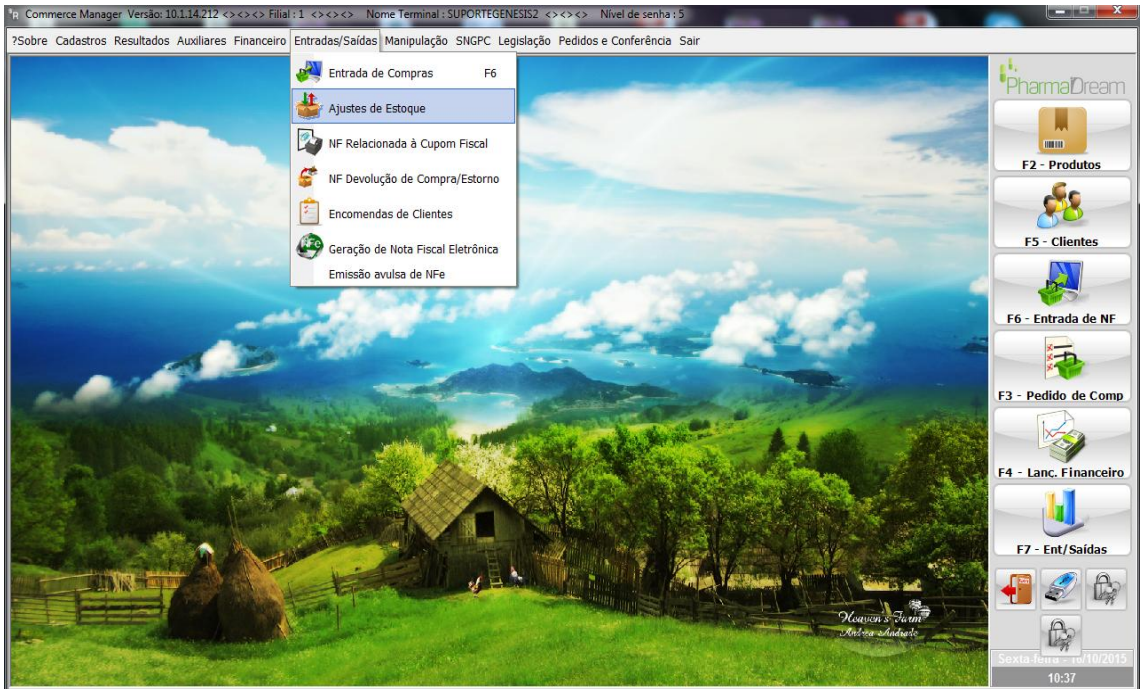

Figura 28: Acesso ao formulário de Inventariamento

A Figura 29, apresenta o formulário de Ajuste de Estoque. Por meio dessa tela é possível o usuário abrir um novo inventário e contabilizar os itens estocados.

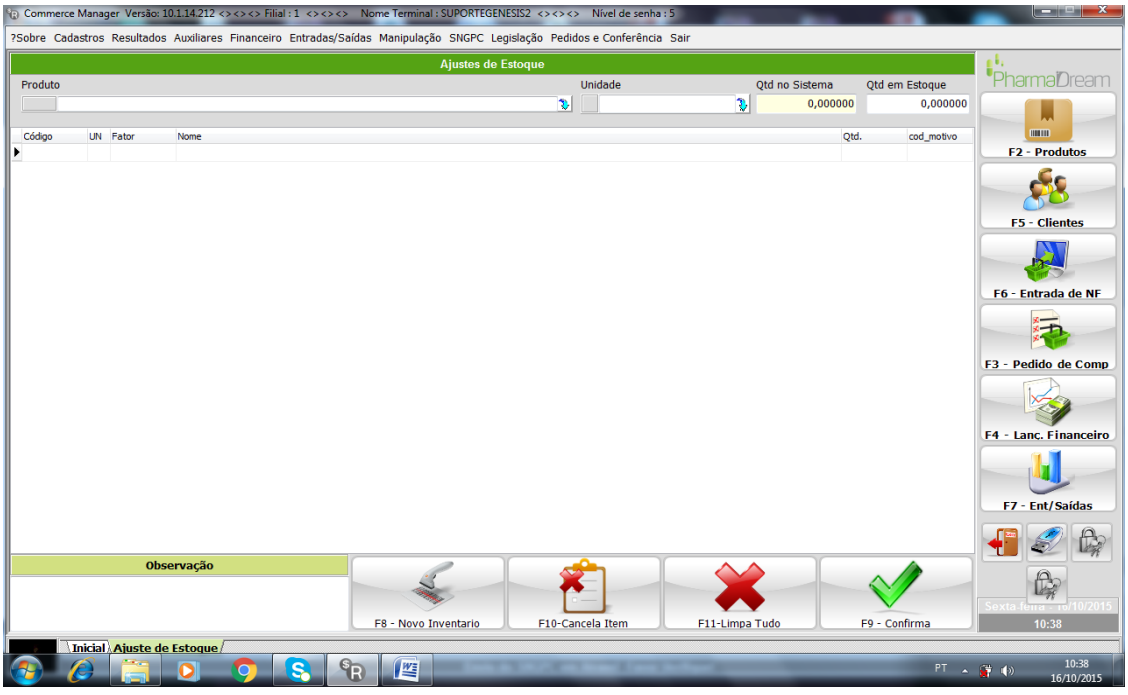

Figura 29: Formulário Ajuste de Estoque

Para realizar um novo inventário, o usuário terá que pressionar o botão F8 – Novo Inventário, e o sistema irá habilitar ao usuário o botão F10 – Importar Arq. Contagem, conforme apresenta a Figura 30. Uma janela do Windows irá abrir para que o usuário localize o arquivo inv.txt.

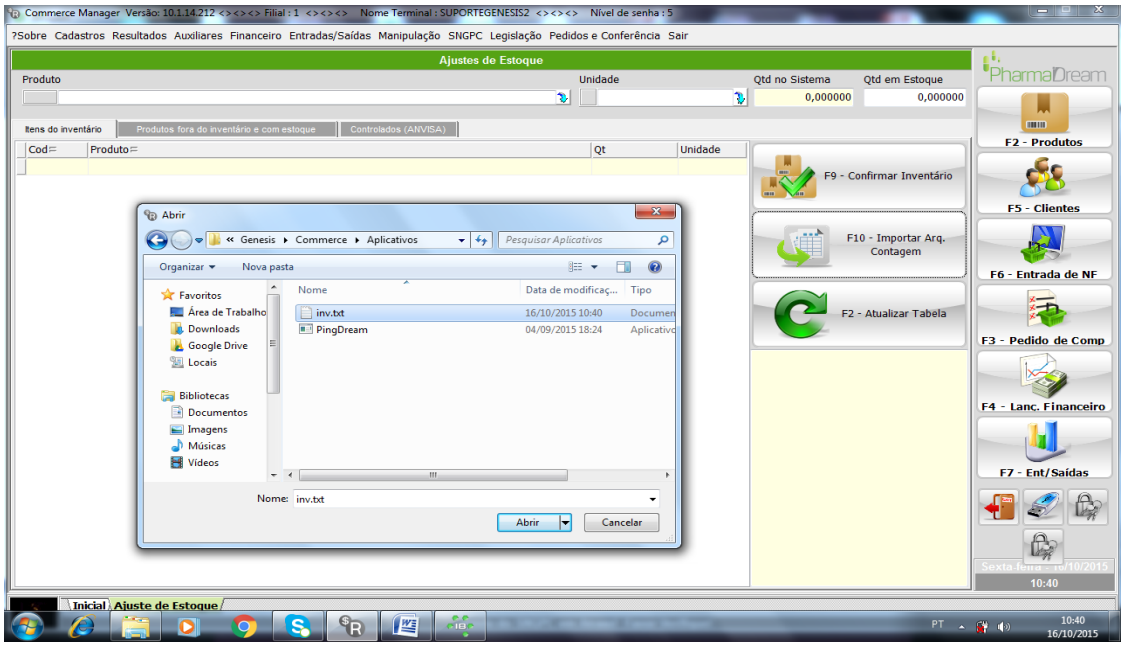

Figura 30: Abrir arquivo inv.txt

Ao selecionar o arquivo, o sistema irá carregar os itens apresentando ao usuário o código de registro do item, a descrição da sua unidade de medida e a quantidade informada no arquivo inv.txt. A Figura 31, mostra a tela com os itens carregados.

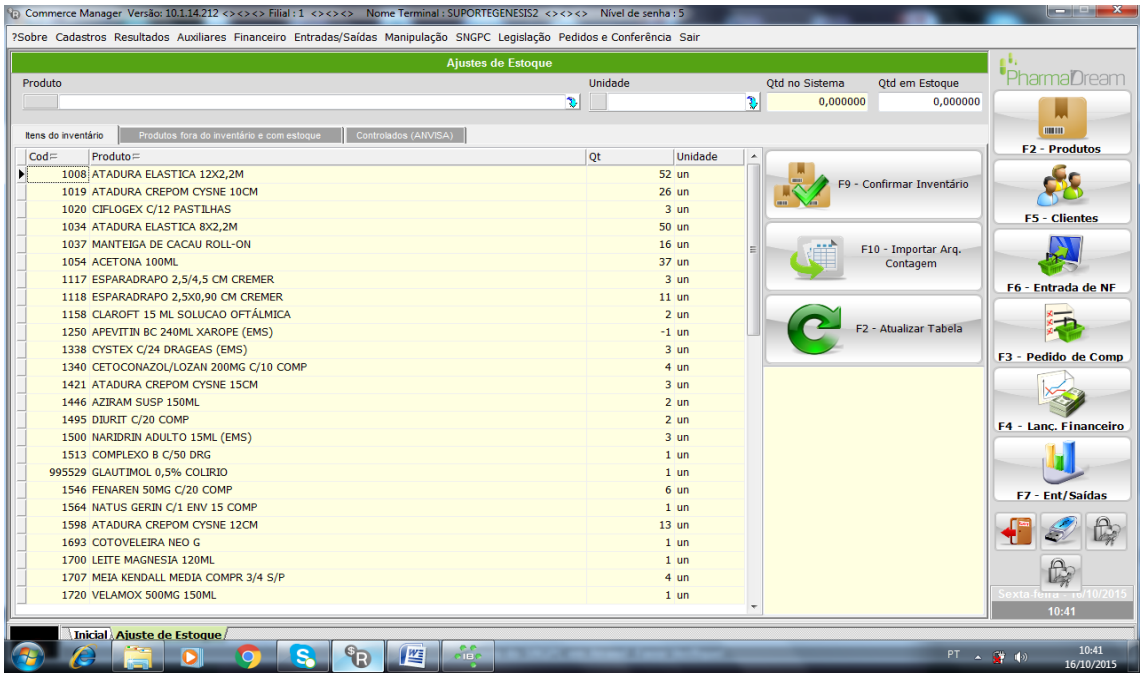

Figura 31: Carregamento dos itens do arquivo inv.txt

Ao confirmar o inventário, o usuário terá os itens do seu sistema com o estoque atualizado segundo o que há em seu estoque físico.

5 CONCLUSÃO

O objetivo deste trabalho foi o desenvolvimento de uma ferramenta para auxiliar a coleta de dados referente à quantidade em estoque de itens dos clientes da empresa de desenvolvimento Genesis Sistemas. A proposta foi automatizar o processo da coleta dos dados tornando o processo prático e eficiente utilizando tecnologias disponíveis para aplicativos móveis.

Para identificação dos itens, foi trabalhado com o código de barras, por ser um dado único de identificação. Foi definido um arquivo texto de itens extraído do sistema desktop do usuário a fim de importar no aplicativo móvel contendo o código de barras e descrição do item.

Foi utilizada a câmera do smartphone para captura da imagem do código de barras e por meio da biblioteca Zxing foi possível a leitura do código de barras para identificação dos itens. Isso fez com que o processo de captura de dados se tornasse prático ao usuário. Em vez de digitar o código, simplesmente passar a câmera no código da embalagem do item para informar o item desejado.

Após a captura dos dados do estoque, o código de barras e quantidades respectivas dos itens informados pelo usuário, o aplicativo móvel disponibiliza ao usuário um arquivo texto contendo esses dados para que possa importar em seu sistema e atualizar seu estoque. A automatização do processo de inventariamento aumenta a segurança no processo, evitando a duplicidade e inconsistência de dados.

A implementação da biblioteca Zxing, apresentou-se prática. Os componentes disponibilizados na biblioteca mostraram eficiência na aplicação, pois a captura e retorno do código de barras acontece de forma ágil a ponto de suprir as necessidades para a qual o aplicativo foi desenvolvido.

A interação com o Pharma Manager, o sistema desktop do usuário, a importação do arquivo do inventario, já existia. Foi somente adequado o layout do arquivo ao layout exigido.

Como trabalho futuro, pode-se citar a implementação da inclusão de dados como lotes e validades dos itens, para controle do estoque do usuário,e a comunicação com o site do governo para inventariar os medicamentos controlados.

REFERÊNCIAS

CASTIGLIONE, José Antônio de Mattos. **Logística operacional: Guia Prático**. 2. ed.São Paulo: Érica, 2009.

CORONADO, Osmar. **Logística integrada: modelo de gestão**. 1.ed. São Paulo: Atlas, 2011.

DIAS, Marco Aurélio P. **Administração de Materiais: princípios, conceitos e gestão**. 5.ed. São Paulo: Atlas, 2008.

LIU, Y., YANG, B., YANG, J. **Bar Code Recognition in Complex Scenes by Camera Phones. Fourth International Conference on Natural Computation**, IEE Computer Society. 2008.

MOREIRA, Daniel Augusto. **Administração da produção e operações**. 1 ed. de 1993. São Paulo: Pioneira Thomson Learning, 2002.

O'BRIEN, James A. **Sistemas de Informação e as decisões gerenciais na era da Internet**. 2. ed. São Paulo: Saraiva, 2004.

OGLIARI, Ricardo da Silva; BRITO, Robison Cris. **Android – Do Básico ao Avançado**. Editora Ciência Moderna, 2014

PFLEEGER, Shari Lawrence. **Engenharia de Software: teoria e prática**. 2. ed. São Paulo: Pretice Hall, 2004.

SOMMERVILLE, Ian. **Software Engineering**. 5. ed. England: Addison-Wesley, 1995.

Zyngier, Mauro Luiz e Grossmann, Fabio. **Código de barras: da teoria a prática**. São Paulo: Nobel, 1991.