



**Fundação Educacional do Município de Assis**  
**Instituto Municipal de Ensino Superior de Assis - IMESA**

**Jean Rafael Gonçalves**

**SISTEMA PARA GERENCIAMENTO DE RESTAURANTES  
UTILIZANDO ANDROID E JAVA**

**Assis, SP**  
**2012**



**Fundação Educacional do Município de Assis**  
**Instituto Municipal de Ensino Superior de Assis - IMESA**

## **SISTEMA PARA GERENCIAMENTO DE RESTAURANTES UTILIZANDO ANDROID E JAVA**

Trabalho de conclusão de curso apresentado ao (IMESA) Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação.

Orientador: Fernando Cesar de Lima

Área de Concentração: Informática, Desenvolvimento de Aplicações

**Assis, SP**  
**2012**

## **Ficha Catalográfica**

**GONÇALVES, Jean Rafael**

Sistema para Gerenciamento de Restaurantes utilizando Android e Java / Jean Rafael Gonçalves. Fundação Educacional do Município de Assis – FEMA – Assis, 2012.

20p.

Orientador: Profº Fernando Cesar de Lima  
Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

**1. Android 2. Java 3. Mobilidade**

CDD: 001.6  
Biblioteca da FEMA



**Fundação Educacional do Município de Assis**  
Instituto Municipal de Ensino Superior de Assis - IMESA

## **SISTEMA PARA GERENCIAMENTO DE RESTAURANTES UTILIZANDO ANDROID E JAVA**

Jean Rafael Gonçalves

Trabalho de conclusão de curso apresentado ao (IMESA) Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação, analisado pela seguinte comissão examinadora:

Orientador: Fernando Cesar de Lima

Avaliador: Alex Sandro Romeo de Souza Poletto

**Assis, SP**  
**2012**

## RESUMO

A tecnologia, que já se encontra em praticamente todos os setores do mercado, continua a se espalhar com novas faces. Atualmente, os dispositivos móveis vêm ganhando muito espaço, e o crescimento da popularidade do Android é uma das causas para isso. Ter um software para o controle de um estabelecimento comercial traz diversas vantagens, que agregadas, resultam no grande objetivo do empresário: maior lucro. Assim acontece no contexto deste trabalho, que são ambientes como restaurantes, lanchonetes, pizzarias, etc.

Quem vai a este tipo de estabelecimento deseja receber o que pediu de forma rápida e, obviamente, receber realmente o que foi pedido. Com o tradicional “papelzinho” usado para anotar os pedidos, são muito comuns erros que resultam em uma longa espera do cliente, produtos entregues diferentes dos que foram pedidos, entre outros problemas.

O presente trabalho tem por objetivo desenvolver um sistema que auxilie no processo de pedido e entrega, onde no lugar de papel e caneta, os garçons usarão um dispositivo móvel (smartphone ou tablet) para que os pedidos sejam enviados a lugares diferentes sem que os garçons tenham que levá-los, ajudando a resolver assim, problemas como os que foram citados anteriormente.

**Palavras-chave:** Android, Java, mobilidade

## **ABSTRACT**

The technology, which is already in practically all the sectors of the market, continues to spread with new faces. Currently, the mobile devices have gained much space, and the growth of Android's popularity is one of the causes for it. Having a software to control a commercial establishment brings many advantages, which aggregate, result in the ultimate goal of the entrepreneur: higher profits. So it is in the context of this work, which are environments like restaurants, snack bars, pizzerias, etc.

Who goes to this type of establishment wants receive what asked quickly and, obviously, receive what was requested really. With the traditional "paper" used to note the orders, there are many common errors that result in a long waiting of the client, products delivered different from those were requested, among other problems.

The present work aims to develop a system that assists in the ordering and delivering process, where instead of paper and pen, the waiters will use a mobile device (smartphone or tablet) for requests to be sent to different places without the waiters have to take them, helping to solve this way, problems like those mentioned above.

**Keywords:** Android, Java, mobility

## Lista de Figuras

Figura 1 – Camadas da Plataforma Android.....	14
Figura 2 – Diagrama de Casos de Uso.....	17
Figura 3 – Diagrama de Classes.....	18
Figura 4 – Diagrama de Sequência .....	19
Figura 5 – Diagrama de Estados .....	20
Figura 6 – Telas iniciais da Aplicação Android .....	21
Figura 7 – Tela Novo Item - Aplicação Android.....	22
Figura 8 – Telas de composição de pizza - Aplicação Android.....	23
Figura 9 – Itens adicionados a comanda - Aplicação Android.....	24
Figura 10 – Tela Fechar Comanda - Aplicação Java.....	26
Figura 11 – Exibição de comanda para fechamento - Aplicação Java.....	27

# SUMÁRIO

<b>1. Introdução</b> .....	<b>08</b>
1.1 Trabalhos Relacionados.....	09
1.2 Justificativa.....	10
1.3 Objetivo.....	11
1.4 Metodologia de Desenvolvimento.....	11
1.5 Estrutura do trabalho.....	11
<b>2. Tecnologia Java</b> .....	<b>12</b>
2.1 JVM (Java Virtual Machine) e JDK (Java Development Kit).....	12
<b>3. Tecnologia Google Android</b> .....	<b>13</b>
3.1 Breve História.....	13
3.2 Arquitetura da Plataforma Android.....	14
3.2.1 Camada Applications.....	15
3.2.2 Camada Application Framework.....	15
3.2.3 Camada Libraries.....	15
3.2.4 Camada Android Runtime.....	15
3.2.5 Camada Linux Kernel.....	15
3.3 Máquina virtual Dalvik.....	16
3.4 Android SDK.....	16
<b>4. Modelagem do Sistema</b> .....	<b>17</b>
4.1 Diagrama de Casos de Uso.....	17
4.2 Diagrama de Classes.....	18
4.3 Diagrama de Sequência.....	18
4.4 Diagrama de Estados.....	19
<b>5. Desenvolvimento do Sistema</b> .....	<b>21</b>
5.1 Aplicação Android.....	21
5.2 Aplicação Java.....	26
<b>6. Conclusão</b> .....	<b>28</b>

## 1. Introdução

Os restaurantes, pizzarias e lanchonetes atraem um grande número de pessoas, principalmente nos finais de semana. Seja um encontro entre amigos, festa de aniversário ou passeio com a família, estes lugares tornaram-se uma ótima opção para unir momentos de lazer com uma boa refeição.

Para satisfação total dos clientes, os serviços prestados em estabelecimentos como restaurantes, lanchonetes e pizzarias, necessitam, principalmente, destes dois atributos trabalhando em harmonia: agilidade e precisão. Isto fica muito mais evidente nos dias de maior movimento, como feriados e finais de semana. Nestes lugares, geralmente os pedidos são efetuados manualmente, e como consequência, são muito comuns problemas como: papel do pedido ilegível, esquecimentos (por parte dos garçons ou cozinheiros), idas desnecessárias dos garçons à cozinha, entre outros.

Segundo Kohlrausch (em SOMENZI, 2005), a confiança na entrega ao cliente do produto, no prazo contratado, é o principal ponto a ser considerado na relação cliente - fornecedor.

Os dispositivos móveis, como *smartphones* (pequenos computadores pessoais) e celulares, apresentam a vantagem da conectividade e poder de uso, em qualquer lugar e hora, sendo extremamente útil não apenas para uso pessoal, mas também profissional (ABLESON; COLLINS; SEN, 2009).

Dispositivos que permitem comunicação e acesso a informação são tecnologias que sempre foram desejadas, principalmente pela grande diversidade de aplicações e benefícios que oferecem, auxiliando na eliminação do papel em processos comerciais e também ajudando no gerenciamento de compromissos e contatos (SIQUEIRA, 2005).

“A utilização de dispositivos móveis nas empresas pode trazer imensos benefícios, porém as principais vantagens estão associadas ao aumento da produtividade e eficiência nas transações, fazendo com que os funcionários e colaboradores possam trocar informações a qualquer tempo (*anytime*) e em qualquer lugar (*anywhere*)” (JUNIOR, 2008).

O Android consiste em uma plataforma de código aberto, destinada ao desenvolvimento de aplicativos para dispositivos móveis, baseada em um sistema operacional *Linux*, com diversas aplicações já instaladas e um ambiente de desenvolvimento bastante poderoso e flexível (LECHETA, 2009).

A plataforma do Google é a primeira para aplicações móveis completamente livre e de código aberto (*open-source*), fato este que influencia diretamente para sua

evolução, uma vez que programadores do mundo todo poderão contribuir para melhorar a plataforma (FARIA, 2008) e (LECHETA, 2009).

É exatamente com base nestes conceitos que surge a solução para o problema apresentado anteriormente. O sistema abordado neste trabalho será muito importante no sentido de obter mais agilidade no processo de pedido e entrega, bem como preservar a integridade das informações que compõem o cotidiano destes estabelecimentos.

Ele será desenvolvido para a empresa Pizzaria e Choperia Forno a Lenha, razão social Joice Moreira Pizzaria ME, CNPJ 08.010.115/0001-79, endereço Avenida Dom Antônio 470, Assis.

## 1.1 Trabalhos Relacionados

A fim de compreender melhor o contexto aqui abordado, são citados trabalhos que possuem alguma relação com este.

GARCIA, AFONSO e BARROS (2007) descrevem o desenvolvimento de desenvolvimento de uma ferramenta desenvolvida para PDA's, mais precisamente Pocket PC's, para auxiliar o acesso às informações relativas aos feirões de veículos. A ferramenta permite ao vendedor carregar consigo todas as informações pertinentes aos veículos do feirão e acessá-las sempre que necessário, pois estão imediatamente ao alcance de suas mãos. Ao ser questionado sobre informações a respeito de algum veículo existente no feirão, ou alguma de suas características, o vendedor, conectado à rede que utiliza um banco de dados central contendo todas as informações do feirão, pode fazer uma simples consulta e atender prontamente ao cliente, acelerando o processo de negociação.

O trabalho apresentado por MORELATO (2008) tem objetivos muito semelhantes aos aqui descritos, como por exemplo: diminuir falhas humanas referentes à escrita e facilitar uma correção quando necessária. O autor fala sobre o desenvolvimento de um sistema a ser executado em *Pocket* PCs, que acompanharão os agentes da Área Azul em suas tarefas. Neste sistema, os agentes preenchem as notificações de regularização, podendo corrigir facilmente erros, nas entradas de dados, que venham a ocorrer. Assim que estiver totalmente preenchida, a notificação é impressa a partir de uma mini-impressora (própria para *Pocket* PC) e fixada no automóvel. Assim que desejar, o agente pode enviar facilmente as notificações a um Banco de Dados, para que outros sistemas possam, eventualmente, automatizar outras partes do processo.

SENS (2009) relata o desenvolvimento de uma ferramenta destinada aos mesmos estabelecimentos citados neste trabalho, mas de uma forma diferente. Com o intuito de diminuir o prazo de entrega agilizando o processo e tornando a empresa (neste caso uma pizzaria) mais competitiva no mercado, tem-se a necessidade da utilização de um sistema de logística de entrega. Diante disso, foi desenvolvido um sistema de logística de entrega para encontrar as melhores rotas entre um ponto e outro da entrega, tendo em vista que um único entregador pode ter mais de uma rota por viagem. Sendo necessário encontrar um caminho que passe em cada um dos pontos de entrega e que leve o menor tempo possível.

## 1.2 Justificativa

Os restaurantes, pizzarias e lanchonetes atraem um grande número de pessoas, principalmente nos finais de semana e feriados.

Embora, na maioria das vezes, o momento seja de descontração, os clientes desejam receber o que pediram de forma rápida e, obviamente, receber realmente o que foi pedido. A maioria destes estabelecimentos usa o tradicional “papelzinho”, onde o garçom anota o pedido dos clientes de uma mesa e, em seguida, leva-o até a cozinha para ser preparado. Apesar de ser o mais conhecido e utilizado, este método pode ocasionar atrasos, erros e até algum tipo de constrangimento.

Aqui estão alguns exemplos:

- O papel do pedido chega à cozinha rasurado e ilegível.
- O garçom tem que, obrigatoriamente, levar os pedidos aos cozinheiros.
- O garçom ou o cozinheiro pode não perceber que o papel caiu em algum lugar, e conseqüentemente, o pedido não será atendido no tempo correto.
- Ao acumular vários pedidos em seu bolso, o garçom pode esquecer-se de entregar algum, ou então alterar a ordem dos mesmos.

Estas são apenas algumas das situações comuns que ocorrem em dias de movimento constante nestes locais. São estas e outras falhas que mostram a necessidade de um método mais eficaz no processo de pedido e entrega dos pedidos.

### **1.3 Objetivo**

O presente trabalho tem por objetivo desenvolver um sistema que auxilie no processo de pedido e entrega, onde no lugar de papel e caneta, os garçons usarão um dispositivo móvel (smartphone ou tablet) para que os pedidos sejam enviados a lugares diferentes sem que os garçons tenham que levá-los, ajudando a resolver assim, problemas como os que foram mencionados anteriormente.

### **1.4 Metodologia de Desenvolvimento**

A aplicação que será executada pelos dispositivos móveis será desenvolvida para a plataforma Google Android. O software que estará em máquinas desktop (computadores da cozinha e do caixa) será escrito na linguagem de programação Java.

Para o desenvolvimento dos dois programas citados acima serão utilizadas as seguintes tecnologias:

- Eclipse Indigo Service Release 2, com o plug-in ADT (Android Development Tools): ambiente de desenvolvimento de software.
- SQL Server 2008 R2: banco de dados.
- ArgoUML 0.28.1: preparação da análise.

### **1.5 Estrutura do Trabalho**

Este trabalho tem a seguinte estrutura de capítulos:

1. Introdução
2. Tecnologia Java
3. Tecnologia Google Android
4. Modelagem do Sistema
5. Desenvolvimento do Sistema
6. Conclusão

## 2. Tecnologia Java

Em 1991, a Sun Microsystems desenvolvia um projeto chamado The Green Project, conduzido inicialmente por Patrick Naughton, Mike Sheridan e James Gosling. Foi neste projeto que surgiu a tecnologia Java, criada como uma ferramenta de programação.

O projeto continuou, e em maio de 1995, o então diretor da Sun, John Cage, e um executivo da Netscape, Marc Andreessen, lançaram formalmente a plataforma Java, composta pela Java Virtual Machine (JVM) e pela API (Application Programming Interface) Java (MENDES, 2009).

A tecnologia Java ganhou grande popularidade, e hoje é amplamente utilizada em vários ambientes diferentes, como desktop, web, e também dentro da plataforma Android, explicada no próximo capítulo.

Atualmente, ocupando 16.25% do mercado, Java aparece como a segunda linguagem de programação mais utilizada no mundo, ficando atrás somente da linguagem C, que detém 17.72% (TIOBE, 2012).

A Figura 2 mostra o logotipo que representa a tecnologia Java.

### 2.1 JVM (Java Virtual Machine) e JDK (Java Development Kit)

A execução de um programa escrito em Java ocorre da seguinte forma:

1. O código-fonte (.java) é compilado pelo compilador Java
2. Esta compilação gera os bytecodes (.class)
3. A JVM transforma os bytecodes em linguagem de máquina
4. Hardware executa a aplicação traduzida

Os passos descritos acima evidenciam que a JVM é a grande responsável pela portabilidade da tecnologia Java, pois, basta que um computador possua uma JVM instalada para que o mesmo possa executar aplicações Java.

O JDK é o kit de desenvolvimento Java que pode ser baixado gratuitamente. Ele contém os utilitários necessários para a criação e execução de aplicativos Java, como por exemplo: APIs, o compilador Java e a JVM.

### 3. Tecnologia Google Android

O Android é uma plataforma de código aberto lançada pelo Google para o desenvolvimento de aplicações para dispositivos móveis como smartphones e tablets. Desde o seu lançamento, o Android vem proporcionando uma revolução no desenvolvimento de aplicações para tais dispositivos, pois é uma plataforma moderna que, aliada a linguagem de programação Java e um ambiente de alto nível como o Eclipse, permite um simples desenvolvimento e integração de aplicações.

Para isso, o Android consiste em um sistema operacional baseado no Linux, trazendo diversas aplicações, um navegador de Internet, integração com o Google Maps, GPS, banco de dados integrado e muitas outras funcionalidades (LECHETA, 2010).

#### 3.1 Breve História

A história do Android começou em 2003, na cidade de Palo Alto, Califórnia, onde Rich Miner e mais dois amigos fundaram uma empresa de desenvolvimento de sistemas operacionais para celulares, a Android Inc. Entretanto, todos seus projetos eram secretos.

Em 2005, a OHA (Open Handset Alliance), grupo de grandes empresas de TI e telefonia, liderada pela Google, comprou a Android Inc. Desta forma, a Google entrou no mercado de software para dispositivos móveis.

Como resultado, o celular HTC Dream G1 foi lançado em outubro de 2008, sendo o primeiro dispositivo a operar sobre a plataforma Android. Ele já possuía janela de notificações, profunda integração com o Gmail, central de download de aplicativos (Android Market), e muito mais. Tais funcionalidades eram bastante avançadas para a época.

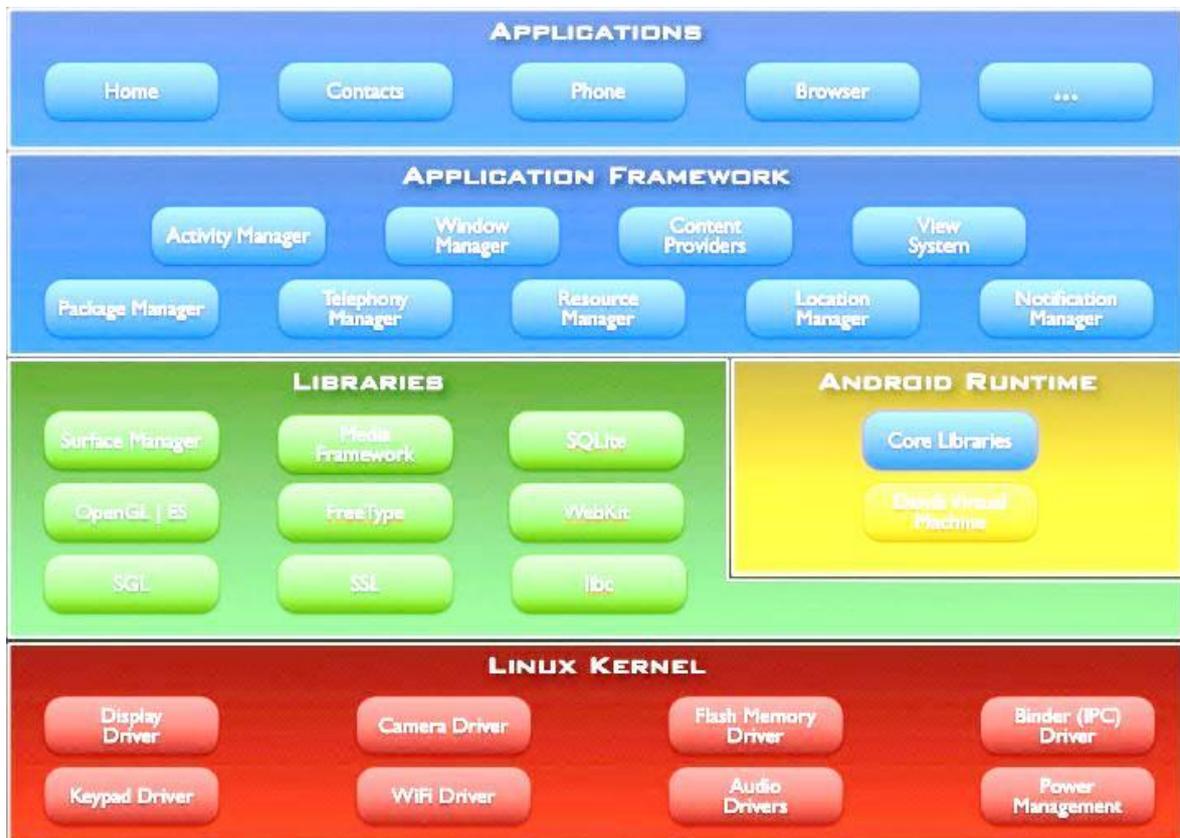
Dando os primeiros passos de uma grande evolução, em abril de 2009 foi lançada a versão 1.5 do Android. Nomeada Cupcake, ela trouxe novidades como teclado virtual, integração com o YouTube e comandos *Copiar* e *Colar*. Em setembro do mesmo ano, veio a versão 1.6 Donut.

Já no mês seguinte, o sistema operacional permitia o uso de múltiplas contas de e-mail, suporte a bluetooth 2.1 e a HTML 5. A versão 2.0 Eclair também trouxe novas ferramentas de calendário e navegação pelo Google Maps.

A plataforma da Google continuou a melhorar e a ganhar mercado até chegar à versão atual, a 4.0 Ice Cream Sandwich, liberada em outubro de 2011, a qual é considerada uma das melhores plataformas móveis do mundo. Entre suas novidades estão o desbloqueio por reconhecimento e maior facilidade no compartilhamento de arquivos (SUPER ABRIL, 2012).

### 3.2 Arquitetura da Plataforma Android

A plataforma Android é composta de cinco camadas, como mostra a Figura 1. Elas serão detalhadas em seguida.



**Figura 1:** Camadas da plataforma Android

### **3.2.1 Camada Applications**

Como próprio nome indica, nesta camada encontram-se todos os aplicativos do Android, como por exemplo, cliente de e-mail, navegador web, agenda de contatos, jogos, e muitos outros. Todos estes aplicativos são escritos em Java (FARIA, 2008).

### **3.2.2 Camada Application Framework**

Esta camada fornece todas as APIs e recursos utilizados pelos aplicativos, como classes visuais, provedores de conteúdo (para troca de informações entre aplicações) e gerenciador de recursos (FARIA, 2008).

### **3.2.3 Camada Libraries**

Trazendo um conjunto de bibliotecas C/C++ utilizadas pelo sistema, esta camada inclui a biblioteca C padrão e também aquelas das áreas de multimídia, visualização de camadas 2D e 3D, funções para navegadores web, funções de acesso ao banco SQLite, entre outras. Ao desenvolver um aplicativo, o programador tem todas estas funcionalidades incorporadas ao framework (FARIA, 2008).

### **3.2.4 Camada Android Runtime**

Esta camada é uma instância da máquina virtual Dalvik. No Android, cada aplicação é executada sobre sua própria instância da máquina virtual Dalvik, tendo esta um ótimo desempenho, pois é integrada com a nova geração de hardware e otimizada para consumo mínimo de memória, bateria e CPU (FARIA, 2008).

### **3.2.5 Camada Linux Kernel**

Baseada no kernel 2.6 do Linux, é nesta camada que estão todas as funcionalidades do sistema operacional, como gerenciamento de memória, processos e threads, segurança dos arquivos e pastas, redes e drivers.

### 3.3 Máquina virtual Dalvik

Já foi mencionado anteriormente que todos os aplicativos Android são escritos em Java. Todavia, a plataforma Android não dispõe de uma Máquina virtual Java (JVM), e sim da Máquina virtual Dalvik.

O desenvolvedor poderá utilizar todos os recursos da linguagem Java normalmente, mas no Android, depois de compilado o bytecode (.class), ele é convertido para o formato .dex (Dalvik Executable), que representa a aplicação Android compilada.

Em seguida, é gerado um arquivo com a extensão .apk (Android Package File), que representa a aplicação final, pronta para distribuição. Este arquivo, que é o instalador da aplicação, é composto pelos arquivos .dex citados acima, além de imagens e outros recursos.

A geração deste arquivo .apk ocorre automaticamente quando se utiliza o ambiente de desenvolvimento Eclipse, com o plug-in ADT (Android Development Tools) (LECHETA, 2010).

### 3.4 Android SDK

O Android SDK (Software Development Kit) é o software usado para o desenvolvimento de aplicações Android, ele fornece ferramentas utilitárias e uma API completa para a linguagem Java, com todas as classes necessárias.

Com o plug-in ADT para o Eclipse, ocorre a integração do ambiente de desenvolvimento Java com o emulador disponível no SDK, pois é possível inicia-lo diretamente do Eclipse, instalar a aplicação e depurar o código-fonte como em qualquer outro software Java.

Para facilitar os testes em dispositivos reais e obter maior produtividade de desenvolvimento, é possível plugar um celular ou tablet na porta USB do computador e executar os aplicativos diretamente neste aparelho. Isto proporciona ao desenvolvedor uma análise do desempenho e usabilidade da aplicação ainda na fase de desenvolvimento (LECHETA, 2010).

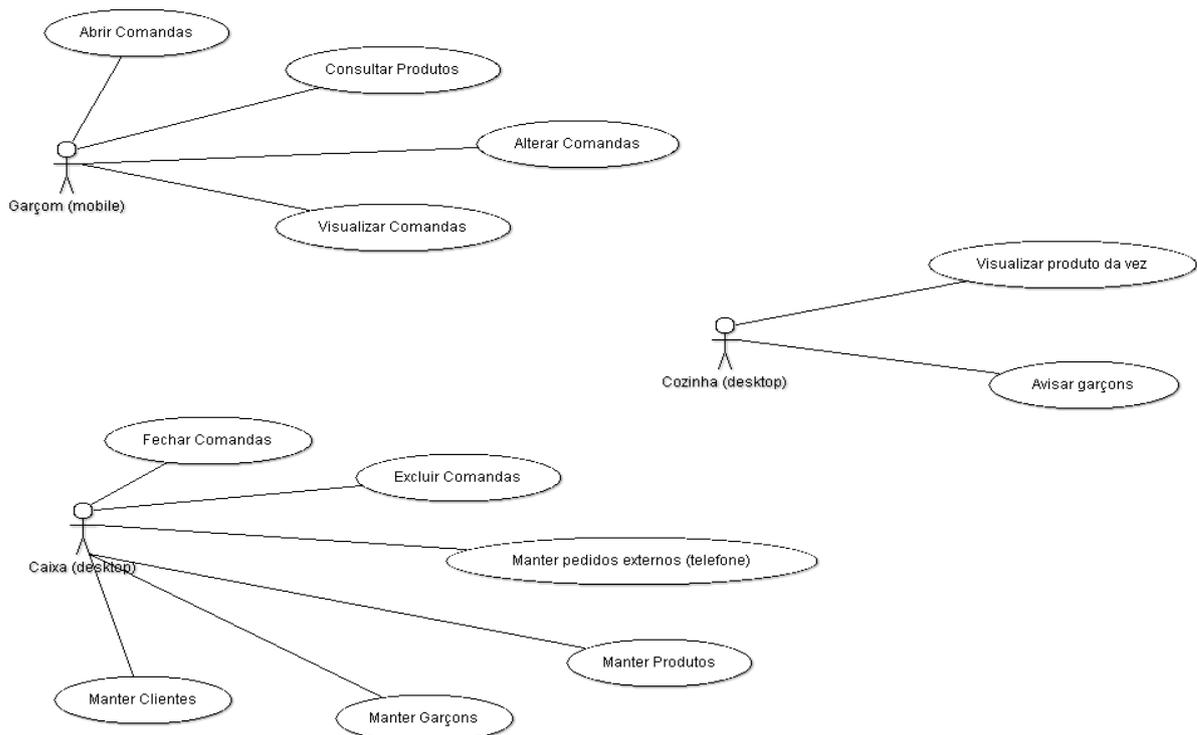
## 4. Modelagem do Sistema

Neste capítulo serão apresentados alguns diagramas da UML (Unified Modeling Language), com o objetivo de compreender melhor aspectos como a arquitetura, funcionalidades e comportamento do sistema desenvolvido neste trabalho.

### 4.1 Diagrama de Casos de Uso

O Diagrama de Casos de Uso tem como objetivo representar graficamente os requisitos do sistema. Nele se encontram os atores, os processos que serão automatizados e o relacionamento entre ambos (LIMA, 2012).

A Figura 2 mostra o Diagrama de Casos de Uso do sistema desenvolvido neste trabalho.



**Figura 2:** Diagrama de Casos de Uso

## 4.2 Diagrama de Classes

O Diagrama de Classes traz uma visualização clara dos objetos usados no sistema, onde estes são representados pelas classes presentes no diagrama. Nele também estão relações, dependências e comportamento das classes (LIMA, 2012).

A Figura 3 mostra o Diagrama de Classes do sistema desenvolvido neste trabalho.

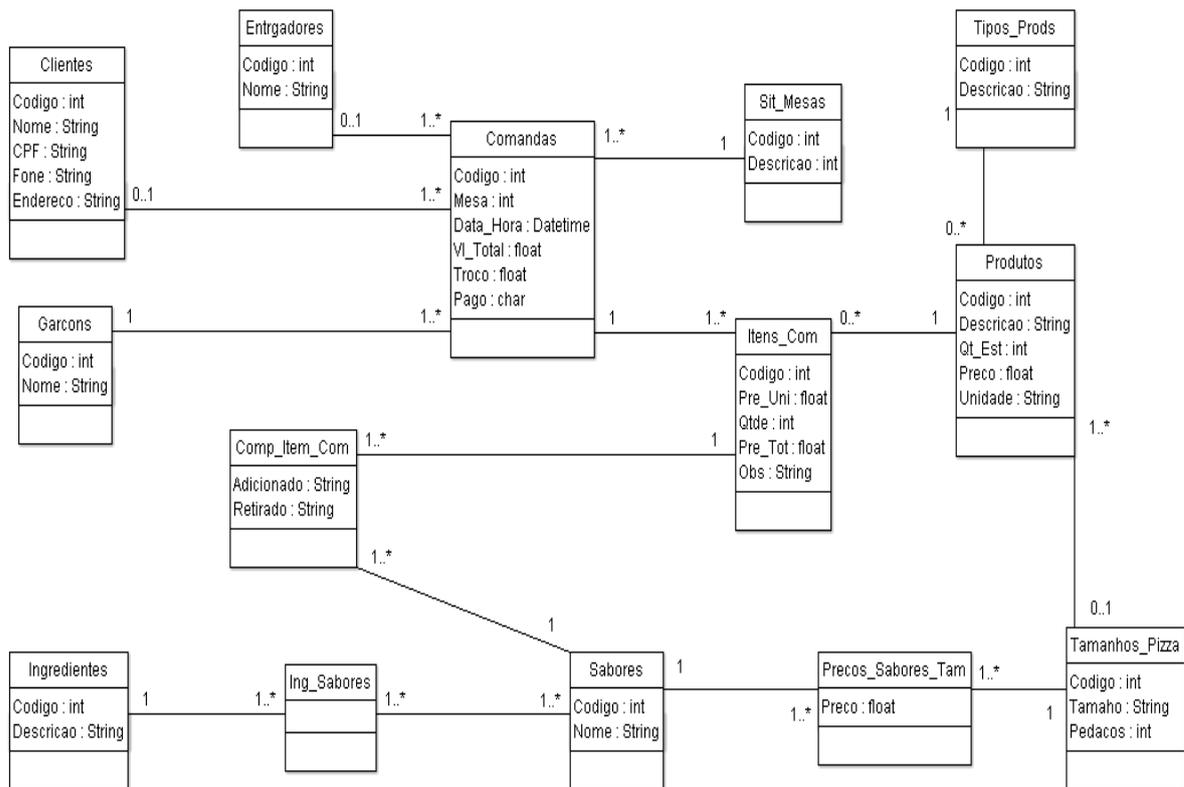
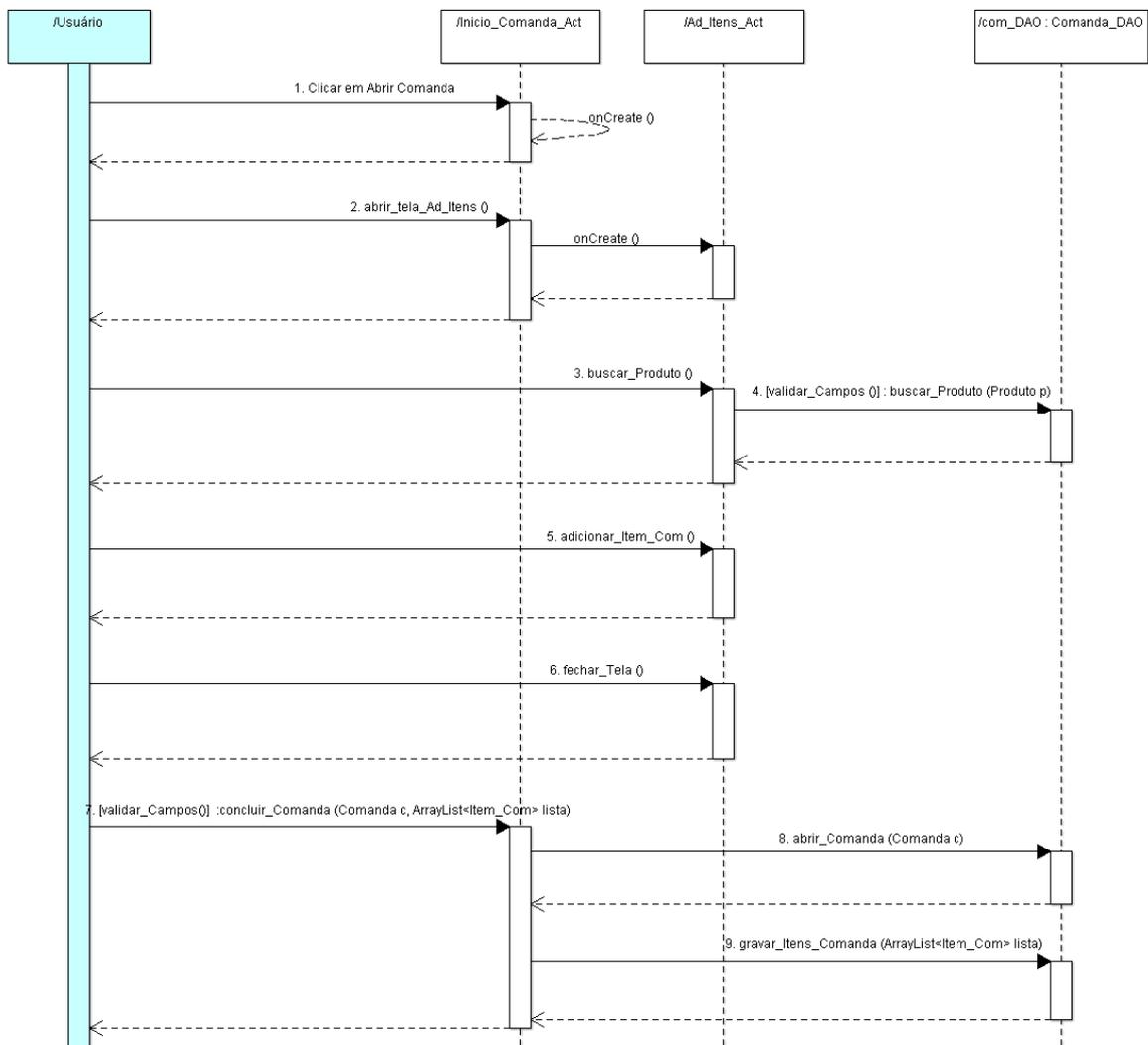


Figura 3: Diagrama de Classes

## 4.3 Diagrama de Sequência

O Diagrama de Sequência mostra a ordem temporal das trocas de mensagens (métodos) entre os objetos de um único caso de uso (LIMA, 2012).

A Figura 4 mostra o Diagrama de Sequência referente ao caso de uso Abrir Comandas.

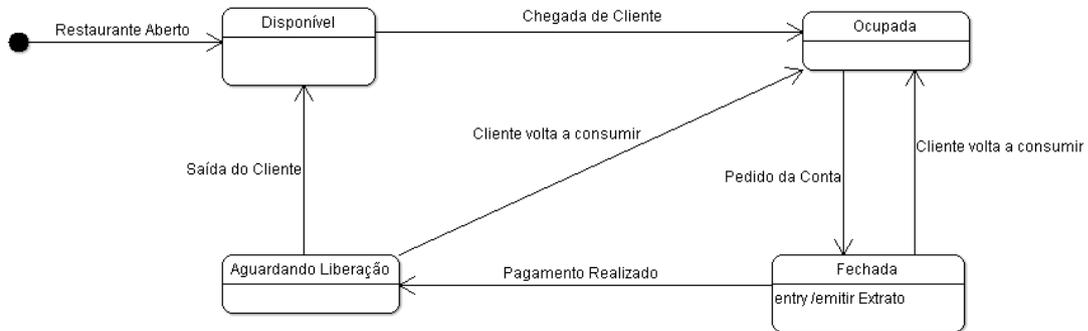


**Figura 4:** Diagrama de Sequência

#### 4.4 Diagrama de Estados

O Diagrama de Estados retrata as várias alterações que um objeto sofre durante seu ciclo de vida. Assim, é possível observar todos os seus estados.

A Figura 5 mostra o Diagrama de Estados do objeto Mesa.



**Figura 5:** Diagrama de Estados

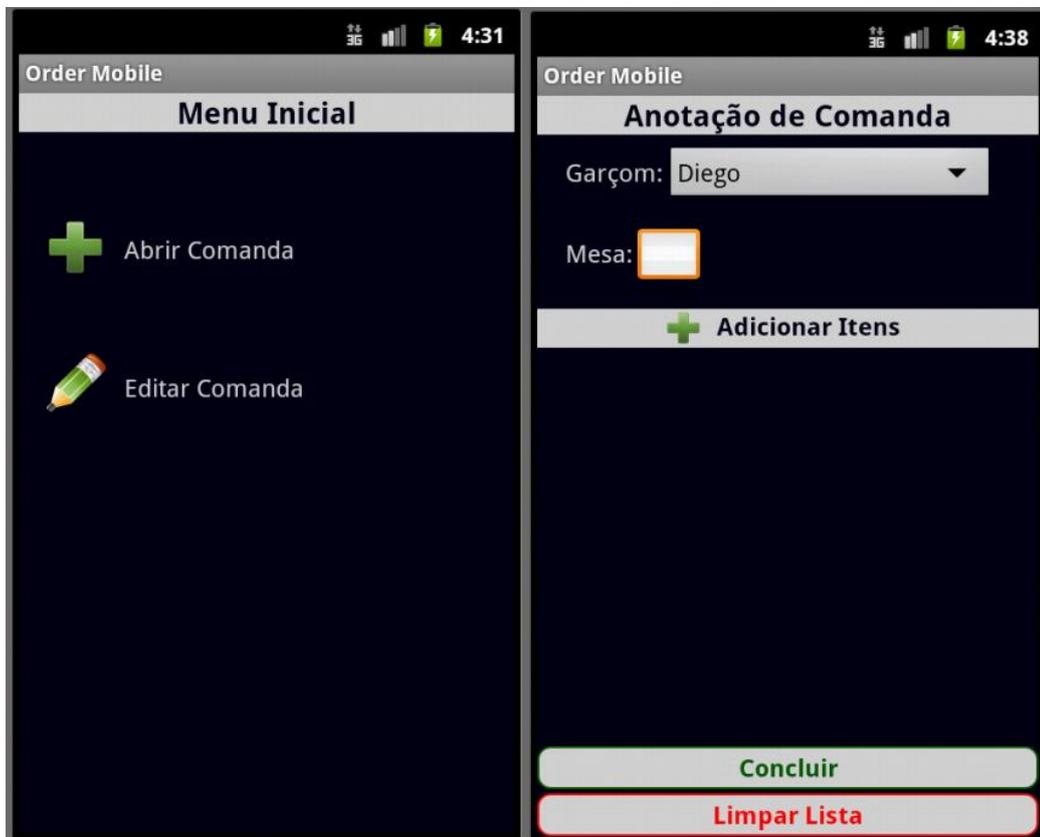
## 5. Desenvolvimento do sistema

Neste capítulo será apresentada a principal funcionalidade do sistema desenvolvido neste trabalho, que se trata de abrir um pedido (chamado de comanda) através de um dispositivo móvel (utilizado pelos garçons), e visualizá-lo e/ou fechá-lo através de um ambiente desktop (utilizado na cozinha e no caixa do estabelecimento). Desta forma, não se faz mais necessário que os garçons levem comanda ao seu destino, pois o sistema se encarrega de fazer este “transporte”.

### 5.1 Aplicação Android

A seguir serão mostradas algumas telas da aplicação que poderá ser utilizada pelos garçons de uma pizzaria, por exemplo. Por ela é possível abrir e editar comandas. Abaixo será descrito o processo de abertura de uma comanda.

A Figura 6 mostra a tela inicial da aplicação Android e também a primeira tela apresentada quando o usuário escolhe, no menu inicial, a opção Abrir Comanda.



**Figura 6:** Telas iniciais da Aplicação Android

Naturalmente, em uma comanda devem estar os itens pedidos pelo cliente. Para adicionar estes itens, o usuário deve tocar no texto “Adicionar Itens” ou no ícone ao lado deste. Será então apresentada a tela Novo Item, mostrada na Figura 7.



**Figura 7:** Telas Novo Item - Aplicação Android

Nesta tela o usuário encontra várias opções de itens. Ao tocar em algum deles, é aberta uma tela que traz todos os produtos disponíveis de acordo com o tipo escolhido, para que o usuário possa escolher algum e adicioná-lo a comanda.

O único tipo de item que difere deste procedimento é a Pizza. Ela pode ser composta por até quatro sabores, e a cada sabor o cliente pode adicionar ou retirar ingredientes. Tocando no ícone ou no texto “Pizza” é aberta a tela para a composição da pizza e de seus sabores. Ao tocar no ícone que fica ao lado direito de algum sabor, é apresentada uma tela para que o garçom possa adicionar ou retirar ingredientes ao sabor escolhido.

A Figura 8 mostra as duas telas usadas na composição de uma pizza.



**Figura 8:** Telas de composição de pizza - Aplicação Android

Com a pizza montada, basta que o garçom informe a quantidade e toque no botão Adicionar Item.

Após adicionar os itens solicitados pelo cliente, o usuário deve voltar à tela mostrada na Figura 6, onde ele deve se identificar e também informar a mesa ocupada pelo cliente. Nesta tela são apresentados os itens adicionados até o momento.

A Figura 9 mostra esta tela com uma comanda pronta para ser aberta.



**Figura 9:** Itens adicionados a comanda - Aplicação Android

Ao tocar no botão Concluir, a comanda é gravada no Banco de Dados, ficando imediatamente disponível para ser acessada de outros lugares no restaurante.

Isso é possível, pois a aplicação Android, através da rede sem fio, grava as comandas diretamente no Banco de Dados situado em um computador que trabalhe como servidor, e aplicação Java utilizada no caixa e cozinha faz acesso a este mesmo Banco.

Como os aplicativos Android são escritos em Java, é possível acessar o Banco de Dados que está em outra máquina da mesma forma abordada em uma aplicação Java Desktop. Basta informar, na String de conexão, o IP da máquina onde se encontra o Banco de Dados.

```

public void conecta() throws Exception
{
    Class.forName(Conexao.driver);
    conexao = DriverManager.getConnection(Conexao.url);
    // Conexao.url = "jdbc:jtds:sqlserver://192.168.1.11:1433/TCC;user=sa;password=*****";

    st = conexao.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);

    Log.i("Conexão com o Banco", "Conectado com o Banco");
}

public void lista_garcons (ArrayList<Garcom> lista_garcons) throws Exception
{
    conecta();

    rs = st.executeQuery("select * from GARCONS order by nome");

    while (rs.next())
    {
        lista_garcons.add(new Garcom(rs.getInt("codigo"), rs.getString("nome")));
    }

    conexao.close();
}

```

Quando o garçom abre uma comanda, uma trigger faz, automaticamente, a atualização do estoque dos produtos envolvidos. Uma trigger é um procedimento que fica armazenado no Banco de Dados e é disparado (executado) automaticamente, de acordo com a ocorrência de um INSERT, UPDATE ou DELETE em uma determinada tabela.

Assim que acontece um INSERT na tabela ITENS\_COM (itens de comanda), a trigger ATUALIZA\_ESTOQUE identifica o produto e atualiza sua quantidade na tabela PRODUTOS.

```

ALTER TRIGGER [dbo].[ATUALIZA_ESTOQUE] ON [dbo].[ITENS_COM] AFTER INSERT AS
    DECLARE @cprod int,
            @qt int

    BEGIN
        SELECT @cprod = PRODUTO FROM INSERTED
        SELECT @qt = QTDE FROM INSERTED

        UPDATE PRODUTOS SET QT_EST = QT_EST - @qt
            WHERE CODIGO = @cprod AND QT_EST IS NOT NULL

    END

```

## 5.2 Aplicação Java

A aplicação Java será responsável por todos os cadastros necessários em um estabelecimento como uma pizzaria, por exemplo. Da mesma forma que esta aplicação tem acesso às comandas geradas pela aplicação Android, os dados cadastrados pelo software Java estarão imediatamente disponíveis para acesso dos garçons através do aplicativo Android, sem nenhuma necessidade de exportação. Abaixo será descrito o processo de fechamento de uma comanda.

Quando o cliente acaba de consumir o que pediu e decide deixar o estabelecimento, o garçom não precisa levar a comanda ao caixa, pois ela pode ser acessada pela aplicação Java.

A Figura 10 mostra a tela Fechar Comanda da aplicação Java.

Descrição	Preço Unitário	Qtde	Preço Total	Observação
-----------	----------------	------	-------------	------------

**Figura 10:** Tela Fechar Comanda - Aplicação Java

Ao clicar na combobox Mesa, é apresentada ao operador (a) de caixa uma lista de mesas que possuem comandas abertas, e basta escolher uma para que os dados desta comanda sejam mostrados na tela.

A Figura 11 mostra a tela de fechamento de comandas após a escolha de uma mesa.

Mesa: 5 Garçom: Diego

Descrição	Preço Unitário	Qtde	Preço Total	Observação
PIZZA GRANDE	26,00	1	26,00	
PIZZA GRANDE	29,00	1	29,00	
COCA-COLA LATA	2,50	2	5,00	
SUCO MARACUJÁ	4,00	1	4,00	
SUCO ACEROLA	3,50	1	3,50	
CONTI BIER LATA	2,00	3	6,00	
PUDIM LEITE COND.	4,00	1	4,00	
TORTA DE FRAMBOESA	7,00	1	7,00	

Total da Comanda: **84,50** Dinheiro: **100,00** Troco: **15,50**

Fechar Comanda

**Figura 11:** Exibição de comanda para fechamento - Aplicação Java

Após a conferência realizada pelo cliente e operador, este último digita a quantia em dinheiro paga pelo cliente, e a aplicação informa o troco. Feito o pagamento, o usuário clica no botão Fechar Comanda, confirma sua ação, e a comanda é encerrada. Após este procedimento, a mesa fica disponível para uma nova abertura de comanda.

## 6. Conclusão

Os restaurantes, pizzarias e lanchonetes atraem um grande número de pessoas, principalmente nos finais de semana. Seja um encontro entre amigos, festa de aniversário ou passeio com a família, estes lugares tornaram-se uma ótima opção para unir momentos de lazer com uma boa refeição.

Para satisfação total dos clientes, os serviços prestados em estabelecimentos como restaurantes, lanchonetes e pizzarias, necessitam, principalmente, destes dois atributos trabalhando em harmonia: agilidade e precisão. Isto fica muito mais evidente nos dias de maior movimento, como feriados e finais de semana. Nestes lugares, geralmente os pedidos são efetuados manualmente, e como consequência, são muito comuns problemas como: papel do pedido ilegível, esquecimentos (por parte dos garçons ou cozinheiros), idas desnecessárias dos garçons à cozinha, entre outros.

O presente trabalho tem por objetivo desenvolver um sistema que auxilie no processo de pedido e entrega, onde no lugar de papel e caneta, os garçons usarão um dispositivo móvel (smartphone ou tablet) para que os pedidos sejam enviados a lugares diferentes sem que os garçons tenham que levá-los, ajudando a resolver assim, problemas como os que foram mencionados acima.

Para uma boa compreensão de aspectos como a arquitetura, funcionalidades e comportamento do sistema proposto, foi levantada uma análise com diagramas da UML, levando em consideração as necessidades presentes no cotidiano de uma pizzaria.

Como resultado, pode-se concluir que o objetivo traçado neste trabalho foi alcançado, pois o sistema desenvolvido promove aos seus usuários mais mobilidade, independência, precisão e agilidade em suas atividades. Com isso, os problemas citados como causa de transtorno e lentidão podem ser minimizados ao máximo. Tudo isso implica no bom andamento de uma pizzaria, restaurante ou lanchonete, pois a satisfação dos clientes é o principal fator para o lucro e crescimento deste tipo de empresa.

## Referências

ABLESON, W. Frank; COLLINS, Charlie; SEN, Robi. Unlocking Android. A Developer's Guide. 1. ed. Greenwich: Manning, 2009.

DEITEL, Paul; DEITEL, Harvey. JAVA - COMO PROGRAMAR. São Paulo: Pearson, 2010.

FARIA, Alessandro de Oliveira. Programe seu Andróide. Linux Magazine, Volume 1, Número 43, p. 73-77, 2008.

GARCIA, A. G. D.; AFONSO, F.; Barros, G.; Relatório Feirão de Automóveis. Universidade Federal de Viçosa, 2007.

JUNIOR, J. B. B; Dispositivos e Tecnologias Móveis nas Empresas: Novas Possibilidades para os Negócios na Era da Informação, 2008. Disponível em: <<http://publica.fesppr.br/index.php/rnti/article/view/55/71>>. Acesso em: 10 de Nov. 2011.

LECHETA, Ricardo R.. Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. São Paulo: Novatec, 2009.

LECHETA, Ricardo. Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 2. ed. São Paulo: Novatec, 2010.

LIMA, Fernando Cesar de. Casos de Uso - Introdução. Disponível em: <[http://200.230.71.24/moodle/file.php/120/Engenharia\\_01\\_CasosUsoIntroducao.pdf](http://200.230.71.24/moodle/file.php/120/Engenharia_01_CasosUsoIntroducao.pdf)>. Acesso em: 24 Jun. 2012.

LIMA, Fernando Cesar de. Diagrama de Classes. Disponível em: <[http://200.230.71.24/moodle/file.php/120/Engenharia\\_03\\_DiagramaClasses.pdf](http://200.230.71.24/moodle/file.php/120/Engenharia_03_DiagramaClasses.pdf)>. Acesso em: 24 Jun. 2012.

LIMA, Fernando Cesar de. Diagrama de Sequência. Disponível em: <[http://200.230.71.24/moodle/file.php/120/Engenharia\\_05\\_DiagramaSequencia.pdf](http://200.230.71.24/moodle/file.php/120/Engenharia_05_DiagramaSequencia.pdf)>. Acesso em: 28 Out. 2012.

LIMA, Fernando Cesar de. Diagrama de Estados. Disponível em: <[http://200.230.71.24/moodle/file.php/120/07\\_-\\_DiagramaEstados.pdf](http://200.230.71.24/moodle/file.php/120/07_-_DiagramaEstados.pdf)>. Acesso em: 28 Out. 2012.

MENDES, Douglas Rocha. Programação Java com Ênfase em Orientação a Objetos. São Paulo: Novatec, 2009.

MORELATO, D. R.; Sistema para Emissão de Notificações de Regularização em Área Azul Utilizando Pocket PC e Webservices sobre a Plataforma .NET. Universidade do Vale do Itajaí, 2008.

SENS, T. A.; Sistema de Pedidos de Pizza para Otimização de Rotas no Google Maps. Universidade Regional de Blumenau, 2009.

SIERRA, Kathy; Bates, Bert. Use a Cabeça! Java. Rio de Janeiro: Alta Books, 2007.

SIQUEIRA, J. R.; Programação do Pocket PC. 1. ed. São Paulo: Novatec, 2005.

SOMENZI, Sílvia. Qual o principal ponto a ser considerado na relação Cliente & Fornecedor? Porto Alegre: UOL, 2005. Disponível em: <<http://www.baguete.com.br/colunasDetalhes.php?id=1654>>. Acesso em: 28 Mar. 2012.

SUPER, Conheça a história do Android, o sistema operacional mobile da Google. Disponível em: <<http://super.abril.com.br/galerias-fotos/conheca-historia-android-sistema-operacional-mobile-google-688822.shtml>>. Acesso em: 17 de Jun. 2012.

TIOBE, TIOBE Programming Community Index for June 2012. Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. Acesso em: 23 de Jun. 2012.