

UNIVERSIDADE FEDERAL DO PARANÁ

DIEGO ANTONIO VONSOVICZ  
DOUGLAS EDUARDO ZEN  
GABRIEL MATUELLA  
NILSON SANGY JUNIOR  
ROGER CAMPOS DE OLIVEIRA



## GARÇOMNET:

UMA PLATAFORMA INTEGRADORA PARA AUTOMATIZAR PEDIDOS

M6  
004.167  
846 V  
2013

CURITIBA

2013

DIEGO ANTONIO VONSOVICZ  
DOUGLAS EDUARDO ZEN  
GABRIEL MATUELLA  
NILSON SANGY JUNIOR  
ROGER CAMPOS DE OLIVEIRA

## GARÇOMNET:

UMA PLATAFORMA INTEGRADORA PARA AUTOMATIZAR PEDIDOS

Trabalho de conclusão de curso apresentado como requisito à conclusão do Curso de Tecnologia em Análise e desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná.

Orientador: Prof. Dr. Alessandro Brawerman

CURITIBA

2013

## TERMO DE APROVAÇÃO

DIEGO ANTÔNIO VONSOVICZ  
DOUGLAS EDUARDO ZEN  
GABRIEL MATUELLA  
NILSON SANGY JUNIOR  
ROGER CAMPOS DE OLIVEIRA

## GARÇOMNET:

UMA PLATAFORMA INTEGRADORA PARA AUTOMATIZAR PEDIDOS

Monografia aprovada como requisito à conclusão do Curso de Tecnologia em Análise e desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, pela seguinte banca examinadora:

Banca: Prof. Dr. Alessandro Brawerman  
Prof. Mário de Paula Soares  
Prof. Pedro Torres

Orientador: Prof. Dr. Alessandro Brawerman  
Tecnologia em Análise e desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná

Membros: Diego Antônio Vonsovicz  
Douglas Eduardo Zen  
Gabriel Matuella  
Nilson Sangy Junior  
Roger Campos de Oliveira

Curitiba, 21 de março de 2013

## SUMÁRIO

<b>LISTA DE ILUSTRAÇÕES.....</b>	<b>6</b>
<b>LISTA DE TABELAS.....</b>	<b>7</b>
<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>8</b>
<b>LISTA DE SÍMBOLOS.....</b>	<b>9</b>
<b>RESUMO.....</b>	<b>10</b>
<b>ABSTRACT.....</b>	<b>11</b>
<b>1. INTRODUÇÃO.....</b>	<b>12</b>
1.1. OBJETIVO GERAL.....	12
1.2. OBJETIVOS ESPECÍFICOS.....	12
1.3. APRESENTAÇÃO DA MONOGRAFIA.....	13
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>14</b>
2.1. ANDROID.....	14
2.2. INTEGRANDO SISTEMAS WEB E MÓVEIS.....	15
2.3. TRABALHOS RELACIONADOS.....	16
<b>3. METODOLOGIA.....</b>	<b>18</b>
3.1. MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE.....	18
3.2. PLANO DE ATIVIDADE.....	19
3.2.1. WBS.....	19
3.2.2. GRÁFICO DE GANTT.....	19
3.3. PLANO DE RISCOS.....	20
3.4. RESPONSABILIDADES.....	21
3.5. MATERIAIS.....	22
3.5.1. DALVIK.....	22
3.5.2. REST.....	22
3.5.3. JSON.....	23
3.6. DESENVOLVIMENTO DO PROJETO.....	25
3.7. VALIDAÇÃO.....	28
3.7.1. TESTES DO ESTABELECIMENTO.....	28
3.7.2. TESTES DO APLICATIVO.....	30
<b>4. APRESENTAÇÃO DO SOFTWARE.....</b>	<b>31</b>
4.1. INSTALAÇÃO, CONFIGURAÇÃO E EXECUÇÃO.....	31

4.1.1. WEBSERVICE.....	31
4.1.2. APLICATIVO DO DISPOSITIVO MÓVEL.....	32
4.1.3. SOFTWARE DO ESTABELECIMENTO.....	32
4.2. PROTÓTIPO DE TELAS.....	33
4.2.1. TELAS DO APLICATIVO MÓVEL.....	33
4.2.2. TELAS DO ESTABELECIMENTO.....	38
<b>5. CONSIDERAÇÕES FINAIS.....</b>	<b>41</b>
<b>6. REFERÊNCIAS.....</b>	<b>42</b>
<b>7. APÊNDICES.....</b>	<b>45</b>
7.1. DIAGRAMA DE CASOS DE USO.....	45
7.2. ESPECIFICAÇÕES DE CASOS DE USO.....	46
7.2.1. UC001 – FAZER PEDIDO.....	46
7.2.2. UC002 – REALIZAR LOGIN.....	51
7.2.3. UC003 – ACOMPANHAR PEDIDO.....	53
7.2.4. UC004 – PEDIR DESTAQUE.....	55
7.2.5. UC005 – PEDIR FAVORITO.....	56
7.2.6. UC006 – REALIZAR LOGIN ESTABELECIMENTO.....	58
7.2.7. UC007 – GERIR PEDIDO.....	59
7.3. DIAGRAMA DE CLASSES.....	64
7.3.1. DIAGRAMA DE CLASSES CLIENTE.....	64
7.3.2. DIAGRAMA DE CLASSES SERVIDOR.....	65
7.4. DIAGRAMAS DE SEQUÊNCIA.....	66
7.4.1. DIAGRAMA DE SEQUÊNCIA VISÃO GERAL.....	66
7.4.2. DIAGRAMA DE SEQUÊNCIA APLICATIVO MÓVEL.....	67
7.4.3. DIAGRAMA DE SEQUÊNCIA ESTABELECIMENTO.....	68
7.5. DICIONÁRIO DE DADOS.....	69
7.6. DIAGRAMA ENTIDADE RELACIONAMENTO (DER).....	71
7.7. WBS.....	72
7.8. DIAGRAMA DE GANTT.....	73

## LISTA DE ILUSTRAÇÕES

Ilustração 1: Fluxograma do tráfego de informações.....	18
Ilustração 2: Diagrama sistema REST com JSON.....	25
Ilustração 3: Menu.....	33
Ilustração 4: Lista Cidades.....	34
Ilustração 5: Cardápio.....	34
Ilustração 6: Produto.....	35
Ilustração 7: Carrinho.....	35
Ilustração 8: Login.....	36
Ilustração 9: Cadastro Usuário 1.....	36
Ilustração 10: Cadastro Usuário 2.....	36
Ilustração 11: Acompanhamento.....	37
Ilustração 12: Produtos Pedidos.....	37
Ilustração 13: Favoritos.....	38
Ilustração 14: Login Estabelecimento.....	38
Ilustração 15: Gerenciamento Estabelecimento.....	39
Ilustração 16: Produtos Pedido.....	39
Ilustração 17: Pedidos Arquivados.....	40
Ilustração 18: Gráfico Produtos.....	40
Ilustração 19: Diagrama de Casos de Uso.....	45
Ilustração 20: Diagrama de Classes Cliente.....	64
Ilustração 21: Diagrama de Classes Servidor.....	65
Ilustração 22: Diagrama de Sequência Visão Geral.....	66
Ilustração 23: Diagrama de Sequência Aplicativo Móvel.....	67
Ilustração 24: Diagrama de Sequência Estabelecimento.....	68
Ilustração 25: Diagrama de Entidades e Relacionamentos.....	71
Ilustração 26: Work Breakdown Structure.....	72
Ilustração 27: Diagrama de Gantt GarçomNET.....	73

## LISTA DE TABELAS

Tabela 1: Distribuição das versões Android.....	15
Tabela 2: Distribuição de responsabilidades.....	21

## LISTA DE ABREVIATURAS E SIGLAS

- API – *Application Programming Interface*
- APP – *Aplicativo para dispositivo móvel*
- GUI – *Graphical User Interface*
- HTTP – *Hyper Text Transfer Protocol*
- IDE – *Integrated Development Environment*
- Inc. – *Incorporation ou Incorporated*
- JSON – *JavaScript Object Notation*
- JSR – *Java Specification Request*
- LAN – *Local Area Network*
- REST – *Representational State Transfer*
- RESTful – *Sistemas que utilizam o protocolo REST*
- SDK – *Software Development Kit*
- SO – *Sistema Operacional*
- SOAP – *Simple Object Access Protocol*
- UC – *Use Case*
- URL – *Universal Resource Locator*
- UTFPR – *Universidade Tecnológica Federal do Paraná*
- VM – *Virtual Machine*
- WBS – *Work Breakdown Structure*
- WS – *Webservice*
- XML – *Extensible Markup Language*

## LISTA DE SÍMBOLOS

- @ – Arroba
- % – Por cento

## RESUMO

Atualmente, a maioria das pessoas possui muito pouco tempo sobrando devido à rotina frenética do mundo capitalista. Com isto em mente e tentando tornar mais cômodas as refeições do dia a dia, o GarçomNET vem propor um sistema que visa eliminar as filas nos estabelecimentos fast-food, e com isso, facilitar o acesso dos clientes aos produtos, alimentos ou refeições, proporcionando comodidade ao cliente.

Este sistema permite a realização do pedido mesmo o cliente não estando no local desejado. Sendo que, o usuário tem acesso a todos os estabelecimentos cadastrados no sistema, juntamente com o cardápio de cada um. Com isso, há a possibilidade de realizar o pedido sem a necessidade de enfrentar uma fila, podendo até mesmo fazer o pedido a caminho do local desejado. Na outra ponta do enlace, o estabelecimento é interligado ao sistema, o qual pode visualizar os pedidos realizados e atualizá-los, sendo que essas atualizações são repassadas automaticamente pelo sistema ao cliente.

Palavras-Chaves: Restaurantes, automação, integração de sistemas, aplicativos móveis.

## ABSTRACT

Nowadays, most of the people have less free time due to the frenetic routine of the capitalist world. With this in mind and trying to render more conveniently the meals, the GarçomNET comes to propose a system that seeks to eliminate the queue on fast-food establishments, and with this, facilitate the access of customers to products, foods and meals, providing convenience to the customer.

This system permits to place an order without the client being on the desired place. The user has access to all the registered establishments in the system, each one with the respective menu. Therefore, there is a possibility to place the order without the necessity to face a queue. Besides, one can even make the request on the way to the desired location. On the other hand of the link, the establishment is connected to the system and it can visualize the orders made and update them.

Key Words: Restaurants, automation, systems integration, mobile applications.

## 1. INTRODUÇÃO

É notável o crescimento na utilização de dispositivos móveis no presente momento. Mais de 13 milhões de dispositivos somente no Brasil. O projeto GarçomNET então explorou esse novo mercado para automatizar o processo de realização de pedidos. Para isso, é proposta uma plataforma composta por um aplicativo móvel, um servidor central e por diversos sistemas gerenciadores de pedidos em restaurantes. O aplicativo utiliza a internet dos dispositivos para integrar o cliente ao estabelecimento no qual o pedido será realizado. Com essa integração proporcionada pela plataforma, o usuário disporá de maior comodidade e facilidade na realização de um pedido.

### 1.1. OBJETIVO GERAL

O objetivo principal da plataforma GarçomNET é interligar clientes e estabelecimentos de uma forma mais rápida e eficaz. Isso sem a necessidade de enfrentar filas, sem fazer com que o cliente perca seu tempo e visando dar mais comodidade ao mesmo na escolha do lanche ou refeição.

### 1.2. OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo proposto, uma série de pequenos passos foram planejados. Estes são apresentados abaixo como objetivos específicos do projeto, são eles:

- Desenvolvimento de um aplicativo móvel capaz de gerenciar pedidos de clientes;
- Ter a possibilidade de executar este aplicativo no maior número possível de dispositivos móveis;
- Desenvolvimento de um sistema central capaz de interligar clientes e estabelecimentos de uma forma transparente para ambos;
- Desenvolvimento de um sistema simples para gerenciar pedidos de clientes no estabelecimento;

- Desenvolvimento de serviços Web para integração dos diversos sistemas envolvidos no projeto;
- Conter uma extensa base de dados de estabelecimentos juntamente com os cardápios de cada um, para que haja variedade para o cliente.
- Realizar validação e testes da plataforma.

### 1.3. APRESENTAÇÃO DA MONOGRAFIA

O restante deste documento está dividido da seguinte forma, o Capítulo 2 apresenta alguns conceitos fundamentais para proporcionar ao leitor um melhor entendimento do projeto. O Capítulo 3 relata a metodologia utilizada no desenvolvimento da plataforma. No Capítulo 4, o software é ilustrado através das especificações de cada funcionalidade. E finalmente, o Capítulo 5 conclui a monografia.

## 2. FUNDAMENTAÇÃO TEÓRICA

O GarçomNET é uma plataforma integrada baseado na arquitetura cliente – servidor em que várias tecnologias são aplicadas. Dentre estas, a utilização do sistema operacional Android, servidor web RESTful, troca de dados entre sistemas no formato JSON e aplicações multi-threads Java Desktop. Essas tecnologias estão descritas e detalhadas nos tópicos seguintes.

### 2.1. ANDROID

O Android é um sistema operacional mantido pelo Google, baseado em Linux, com máquina virtual Java própria, de código aberto, desenvolvido para funcionar em tablets e smartphones (LECHETA, 2010). Após ser adquirida pela Google, a Android Inc. desenvolveu uma plataforma de dispositivos móveis com o objetivo de ser flexível e de fácil migração para os fabricantes.

Segundo (Ramos, 2013), mal começou o ano e já há indicações de quem será o vencedor da corrida de sistemas operacionais anual. De acordo com uma pesquisa realizada pelo Instituto ABI (ABI RESEARCH, 2013), o Android estará na frente da concorrência com uma folga considerável. Visto que dos 1,4 bilhão de smartphones que estarão circulando pelo mercado no final de 2013, 798 milhões rodarão o sistema da Google Inc.. Para se ter uma ideia do que isso significa, a Apple com o iOS ocupará a segunda posição com "apenas" 294 milhões de aparelhos. Windows Phone 8 e Blackberry 10 vêm na sequência, com 45 e 20 milhões, respectivamente. Portanto, pela popularidade do sistema operacional, foi escolhida a plataforma Android para desenvolvimento do GarçomNET.

Além da popularidade, o Android possui muitas outras vantagens, como código aberto e frequentes atualizações gratuitas de versão do SO (Sistema Operacional). Com isso, novas versões cada vez mais robustas e modernas são lançadas com o passar do tempo. Entretanto, muitos usuários ainda utilizam as versões mais antigas, pelo fato da maioria dos aparelhos não suportar atualizações de versão devido à limitação do hardware. Com isso, a versão escolhida como ideal para rodar o GarçomNET foi a GINGERBREAD, tomando-se como parâmetro a

Tabela 1, disponibilizada no site Developer Android (DEVELOPER, 2013), onde há a distribuição dos usuários por API utilizada. Note a porcentagem de usuários da versão Gingerbread.

Versão	Codename	API	Distribuição
4.2	Jelly Bean	17	1.4%
4.1		16	12.2%
4.0	Ice Cream Sandwich	15	29.0%
3.2	Honeycomb	13	1.0%
3.1		12	0.3%
2.3.3 – 2.3.7	Gingerbread	10	45.4%
2.3 – 2.3.2		9	0.2%
2.2	Froyo	8	8.1%
2.1	Eclair	7	2.2%
1.6	Donut	4	0.2%

Tabela 1: Distribuição das versões Android

## 2.2. INTEGRANDO SISTEMAS WEB E MÓVEIS

A integração de sistemas de diferentes portes é muito comum em soluções de computação nos dias de hoje. Nas soluções mais comuns estão a integração de sistemas Web com outros sistemas Web, por exemplo, validação e realização de pagamentos via cartão de crédito em sites de compra, verificação de CPF em formulários web, ou então busca de logradouro através de CEP e vice versa. Essa integração é contida também em sistemas Web e aplicativos para dispositivos móveis, por exemplo, aplicativos de internet *banking*, redes sociais, monitoramento do clima e *newspapers*. Portanto, a maneira mais comum e eficiente de se integrar plataformas é a partir do uso de web services.

Um WS (Web Service) é uma coleção de protocolos e padrões para a troca de informação entre sistemas (W3SCHOOLS, 2013). Isso significa que eles têm como principal objetivo realizar a interconexão de sistemas, de forma independente de linguagem de programação ou sistema operacional.

Atualmente muitos smartphones e tablets possuem pouco poder de processamento e memória. Por essa razão, é necessário reduzir e limitar ao máximo o processamento causado pelos aplicativos de um dispositivo. Com isso, a solução mais usual é o desenvolvimento de um webservice RESTful, o qual não é orientado

a conexão, ou seja, não grava o estado da conexão e não utiliza *cookies* (FIELDING, 2000). Dessa forma, o aplicativo se torna muito mais leve e com melhor desempenho.

Além da economia de processamento nos smartphones, foi pensado também no tráfego das informações utilizando as conexões atuais, 3G e GPRS. Como essas tecnologias ainda estão bem aquém das expectativas dos usuários, esse requisito teve de ser considerado. Com isso, foi utilizado o protocolo REST juntamente com JSON, o qual possui uma formatação leve, fácil de interpretar e gerar. Em suma, esta tecnologia é um arquivo em formato texto e completamente independente de linguagens, pois usa convenções que são familiares à maioria delas. Estas propriedades fazem com que o JSON seja um formato ideal para troca de dados.

### 2.3. TRABALHOS RELACIONADOS

A tecnologia abordada no GarçomNET está com grande popularidade em vários sistemas. Com uma rápida pesquisa em artigos, sites e revistas, percebe-se que a utilização de Android e web service está em grande ascensão.

Exemplo disso são as propostas para sistemas voltados aos visitantes que estarão no Brasil no período na copa do mundo FIFA 2014. Uma dessas propostas é apresentada em (FIT TECNOLOGIA, 2013), e propõe dois diferentes tópicos:

- aspectos importantes relacionados ao desenvolvimento de uma aplicação móvel e
- possíveis funcionalidades de um aplicativo móvel baseado em uma rede social.

Essa proposta tem como contexto o seguinte: "Como os turistas podem ter serviços e conteúdos mais acessíveis quando estiverem em visita ao Brasil? ". Contudo, a manipulação desses serviços vai ser feita pelo próprio visitante. A ideia desse projeto é de não se limitar apenas à Copa do Mundo 2014, e sim para o cenário turístico futuro também. Portanto, essa aplicação se assemelha ao GarçomNET na utilização da plataforma Android, e no uso de um Web service que realizará as operações do sistema, respondendo ao dispositivo móvel as requisições

do cliente. Vê-se a mesma arquitetura cliente dispositivo móvel juntamente com servidor web.

Um outro trabalho (DE LIMA, 2012) explorou a situação de um usuário do transporte coletivo. Esse artigo propôs um aplicativo que visa fornecer informações aos passageiros para que este possa tomar decisões sobre qual ônibus ou rota utilizar, além de haver a possibilidade dos clientes contribuírem para o crescimento do aplicativo através das redes sociais. Esse trabalho é similar ao GarçomNET na ideia de utilizar os dispositivos móveis para trazer comodidade ao cliente e facilitar o dia a dia das pessoas, além de utilizar a mesma arquitetura integrada aplicativo cliente e Web service.

Outra proposta, feita por alunos da Universidade de Passo Fundo (PRESTES, *et. al.*, 2011), foi elaborar um Web service para previsões de tempo e clima através de dados georreferenciados, pois uma grande parcela da população não possui o conhecimento suficiente ou acesso à uma consulta de informações desse gênero. Esse sistema utiliza PHP para o acesso via navegador pelo cliente e Java no WS, como o GarçomNET, que também utiliza servidor Java.

Há também um sistema elaborado por alunos da UTFPR (JUNIOR, *et. al.*, 2010), o qual tem o intuito de auxiliar vendedores disponibilizando um aplicativo móvel que, no momento de uma negociação, forma preços de venda baseados nos preços de vários concorrentes. Esse sistema utiliza um Web service no qual o aplicativo se conecta para realizar uma busca em sites de comércio eletrônico. Essa busca retorna o preço de venda atual de cada produto desejado em vários estabelecimentos. Com isso, os gestores têm a disposição informações importantes para uma tomada de decisão mais assertiva e fundamentada. E isso de forma cômoda e prática, o que se assemelha muito ao GarçomNET, que também visa a comodidade e praticidade, além de também utilizar WS e aplicativo para dispositivos móveis.

### 3. METODOLOGIA

O GarçomNET foi desenvolvido baseado numa arquitetura cliente – servidor, utilizando a metodologia de programação orientada a objetos, sendo que, no cliente o sistema operacional escolhido foi o Android. Também foi utilizado um Web service, ou seja, um servidor Web, onde é feita toda a gerência e armazenamento das transações de pedidos do sistema utilizando o protocolo REST. Por outro lado, foi decidido utilizar uma aplicação Java Desktop nos estabelecimentos, visto que aplicativos Java são amigáveis e de simples manuseio. Além da possibilidade que o Java fornece de instalação em qualquer sistema operacional.

Logo a seguir, a Ilustração 1 mostra a transação das informações entre o cliente e o servidor quando o usuário realiza um pedido. Do outro lado, o estabelecimento busca os pedidos solicitados e atualiza as informações do mesmo.

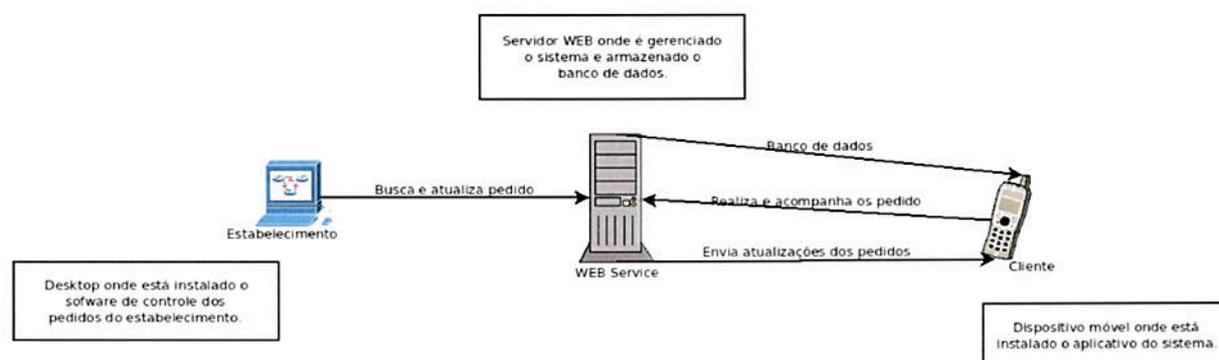


Ilustração 1: Fluxograma do tráfego de informações

#### 3.1. MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

A metodologia de engenharia de software utilizada no GarçomNET foi a de Orientação a Objetos, sempre se fundamentando nos conceitos essenciais de classe, objeto, instância, atributo, método, mensagem, herança, abstração e polimorfismo.

Para que houvesse uma visão completa das funções do sistema, foi elaborado o Diagrama de Casos de Uso (Ilustração 19). Na sequência, foram elaborados os Diagramas de Classes Cliente (Ilustração 20) e Servidor (Ilustração

21) para que o desenvolvimento pudesse ser norteado através das classes descritas por esses diagramas.

Foi ainda criado o Diagrama de Entidade Relacionamento (Ilustração 25), que discrimina as tabelas, atributos e relacionamentos necessários para o bom funcionamento do GarçomNET. Finalmente, para visualizar o fluxo de telas juntamente com o tráfego de informações na realização de um pedido, foram elaborados os Diagramas de Sequência Visão Geral (Ilustração 22), Aplicativo Móvel (Ilustração 23) e Estabelecimento (Ilustração 24). Toda essa documentação pode ser consultada nos apêndices deste documento.

## 3.2. PLANO DE ATIVIDADE

### 3.2.1. WBS

O WBS, apresentado na Ilustração 26 localizada nos apêndices, mostra a distribuição das atividades do GarçomNET de forma modular, desde o planejamento da plataforma até o encerramento das tarefas. Sendo que se observa, desde o início, três frentes de trabalho para que todos os objetivos do sistemas fossem atingidos, as quais foram a elaboração do aplicativo Android, a disponibilização do Web service, e o desenvolvimento do sistema do estabelecimento. Nessas frentes foram divididos os integrantes da equipe, de forma que cada um fosse responsável por determinada parte do desenvolvimento.

### 3.2.2. GRÁFICO DE GANTT

A Ilustração 27 mostra o gráfico de Gantt. Este mostra o calendário das atividades, desde o início do projeto em 01/10/2012 até a entrega aos avaliadores em 13/03/2013. Inicialmente, foram gastas cerca de três semanas para que o GarçomNET pudesse ser planejado e estruturado. Após isso, foi iniciado o desenvolvimento do aplicativo para dispositivo móvel, além do Web service e do software para os estabelecimentos.

Percebe-se ainda na Figura 3 que até dez/2012, o desenvolvimento seguia conforme o planejado. Entretanto, em meados do fim de dezembro e mês de janeiro, os planejamentos foram reestruturados para que o equívoco na configuração inicial do Web service pudesse ser sanado, e então, os cronogramas dos três sistemas sofreram alterações.

Nota-se também que, somente após a realização da tarefa "Refazer WS usando REST e JSON", é que as tarefas referentes à integração das plataformas puderam ser realizadas, visto que o Web service é objeto fundamental na arquitetura integrada do GarçomNET.

### 3.3. PLANO DE RISCOS

Na elaboração do GarçomNET alguns riscos foram identificados, e por conseguinte, certas ações foram planejadas e realizadas para que esses riscos pudessem ser diminuídos ou até mesmos extinguidos.

RISCO: Desconhecimento dos integrantes sobre funcionamento de Web services.

AÇÕES: Estudos sobre sistemas integrados utilizando WS, sobre Web services utilizando REST, e também sobre a utilização do JSON para tráfego de informações.

PROBABILIDADE: Alta

IMPACTO: Atraso no desenvolvimento do WS e erros na modelagem do sistema.

CLASSIFICAÇÃO: Grave

RISCO: Desconhecimento dos integrantes sobre programação para aplicativos móveis de plataforma Android.

AÇÕES: Matrícula na matéria de programação para dispositivos móveis na UFPR. Além de estudos extras em fontes como (LECHETA, 2010).

PROBABILIDADE: ALTA

IMPACTO: Atraso no desenvolvimento do Aplicativo Móvel e erros nas interfaces e lógica de programação.

CLASSIFICAÇÃO: Grave

RISCO: Falha na conexão entre Aplicativo Móvel, Web service e Estabelecimento.

AÇÕES: Criação de uma rede local (LAN) utilizando um *access point*, para que os sistemas pudessem estar conectados numa rede sem qualquer tipo de bloqueio ou filtro que impeça o tráfego dos pacotes entre as plataformas.

PROBABILIDADE: MÉDIA

IMPACTO: Não funcionamento da plataforma.

CLASSIFICAÇÃO: Gravíssimo

### 3.4. RESPONSABILIDADES

A Tabela 2 exibe as atividades executadas pelos integrantes do grupo no desenvolvimento do trabalho:

INTEGRANTES	ATIVIDADES EXECUTADAS
Diego Antônio Vonsovicz	<ul style="list-style-type: none"> <li>- Planejamento e desenvolvimento das telas do aplicativo.</li> <li>- Planejamento e desenvolvimento do Web service.</li> </ul>
Douglas Eduardo Zen	<ul style="list-style-type: none"> <li>- Planejamento e desenvolvimento do sistema do estabelecimento.</li> <li>- Planejamento do banco de dados.</li> <li>- Elaboração das especificações de telas da plataforma do estabelecimento.</li> </ul>
Gabriel Matuella	<ul style="list-style-type: none"> <li>- Desenvolvimento das telas Favoritos, Destaques e Estabelecimentos do aplicativo.</li> </ul>
Nilson Sangy Junior	<ul style="list-style-type: none"> <li>- Planejamento e desenvolvimento do Web service.</li> <li>- Planejamento e desenvolvimento do banco de dados.</li> <li>- Desenvolvimento das telas Acompanhamento e Produto do aplicativo.</li> <li>- Elaboração da documentação.</li> </ul>
Roger Campos de Oliveira	<ul style="list-style-type: none"> <li>- Planejamento e desenvolvimento do banco de dados.</li> <li>- Elaboração da documentação.</li> </ul>

Tabela 2: Distribuição de responsabilidades

### 3.5. MATERIAIS

Abaixo estão descritas as tecnologias utilizadas no GarçomNET mais detalhadamente, as quais são a VM da Google Inc., além dos protocolos REST e JSON.

#### 3.5.1. DALVIK

A Google Inc. criou a máquina virtual Dalvik, a qual é utilizada juntamente com o Android com o intuito de melhorar alguns pontos em que a JVM, máquina virtual mantida pela Oracle, falha quando se trata de dispositivos móveis (DEX2JAR, 2012).

Segundo (ANDROID, 2012), Dalvik é uma máquina virtual que roda aplicativos e códigos em Java com o intuito de, com a ajuda de um compilador padrão do Java, transformar o código em *bytecode*, compilando as classes para uma extensão *.dex*, as quais são lidas, interpretadas e executadas pela VM.

Sendo assim, os pontos positivos que a Dalvik possui são o desempenho, a segurança e a total liberdade para evoluir o Android sem a necessidade da especificação JSR (Java Specification Request) da Oracle, que é um processo bastante burocrático e necessário para que sejam inseridos novos recursos na plataforma. Com isso, o Google tem o controle completo e pode impulsionar a adoção de novas funcionalidades de uma forma mais rápida e simples (MOBILIDADE FM, 2012).

#### 3.5.2. REST

REST é um protocolo usado em Web services desenvolvido por Roy Fielding que foi baseado em SOAP, tornando este mais simples que qualquer outra arquitetura para WS.

Segundo Fielding em sua tese de PhD (FIELDING, 2000), REST em si é um estilo de alto nível que pode ser implementado utilizando muitas tecnologias diferentes, e utilizando diferentes valores para suas propriedades abstratas. Esse

protocolo inclui conceitos de recursos além de uma interface uniforme, ou seja, inclui a ideia de que todo recurso deveria responder aos mesmos métodos.

Sistemas que utilizam o protocolo REST são denominados sistemas RESTful, e os fundamentos desses Web services são as URLs e os *resources*. Sendo que, um *resource* é uma entidade, ou seja, um objeto com informação que pode ser representado por meio de um JSON. Em geral, a URL para acessar esse recurso é sempre a mesma, porém, para cada método HTTP (GET, POST, PUT, DELETE), o resultado da requisição ao sistema é diferente.

Mais detalhadamente, os métodos HTTP são os seguintes:

- GET: define um acesso de leitura do recurso sem efeitos colaterais. O recurso nunca é alterado através de uma requisição GET, por exemplo, o pedido não tem efeitos colaterais. Esse método é utilizado normalmente para se realizar pesquisas passando parâmetros pela URL.

- POST: cria um novo recurso, ou seja, insere novos dados ou objetos no servidor.

- PUT: atualiza um recurso existente. Normalmente o ID de algum objeto é passado na operação.

- DELETE: remove os recursos. Também é comum passar o ID de algum objeto nessa operação.

### 3.5.3. JSON

JSON (JavaScript Object Notation) é um padrão de dados mais simples, e atualmente é uma alternativa ao uso do XML na transmissão de dados entre cliente e servidor. O protocolo foi derivado da sintaxe de objetos em Javascript, mas apesar de estar relacionado ao mesmo, é um padrão para vários analisadores em diferentes linguagens, podendo servir para diferentes propósitos (JSON, 2013).

Em geral o formato JSON costuma ser menor que o equivalente em XML. Porém, a grande vantagem é que aquele protocolo possui uma formatação leve, fácil de interpretar e gerar. Isso pelo fato do JSON ser um arquivo em formato texto e completamente independente de linguagens, pois utiliza convenções que são

familiares às linguagens Java, Android, C#, PHP, entre outras (LINHADECODIGO, 2013).

A definição do JSON parece um dicionário, utilizando a estrutura chave e valor. A chave é sempre uma string, e precisa estar entre aspas duplas. Já o valor pode ser de vários tipos:

- um número, inteiro ou decimal,
- uma string, que precisa estar entre aspas duplas,
- um booleano (true ou false),
- um array, que deve estar entre colchetes,
- outro objeto, que deve estar entre chaves,
- ou null, quando não há valor.

Alguns exemplos para melhor entendimento:

```
{  
  "idConta": 251,  
  "nome": "José da Silva",  
  "ativo": true,  
  "endereco": {  
    "logradouro": "Rua do José, 1119",  
    "bairro": "Bairro do José",  
    "cidade": "São Paulo",  
    "estado": "SP" },  
  "telefones": [ "9999-9999", "8888-8888" ]  
}
```

Observe que no exemplo acima, o atributo “endereço” possui como tipo um objeto, por isso é aberto um novo par de chaves, como se estivesse sendo aninhando um objeto dentro do outro. O atributo “telefones” possui como tipo um array, e por isso foi aberto um par de colchetes.

A Ilustração 2 representa graficamente um sistema cliente – servidor onde é utilizado o protocolo REST com JSON, sendo que a aplicação cliente envia requisições GET, POST, PUT e DELETE com o arquivo JSON ao Web service; e após a requisição ser processada, o servidor responde com outro JSON à requisição.

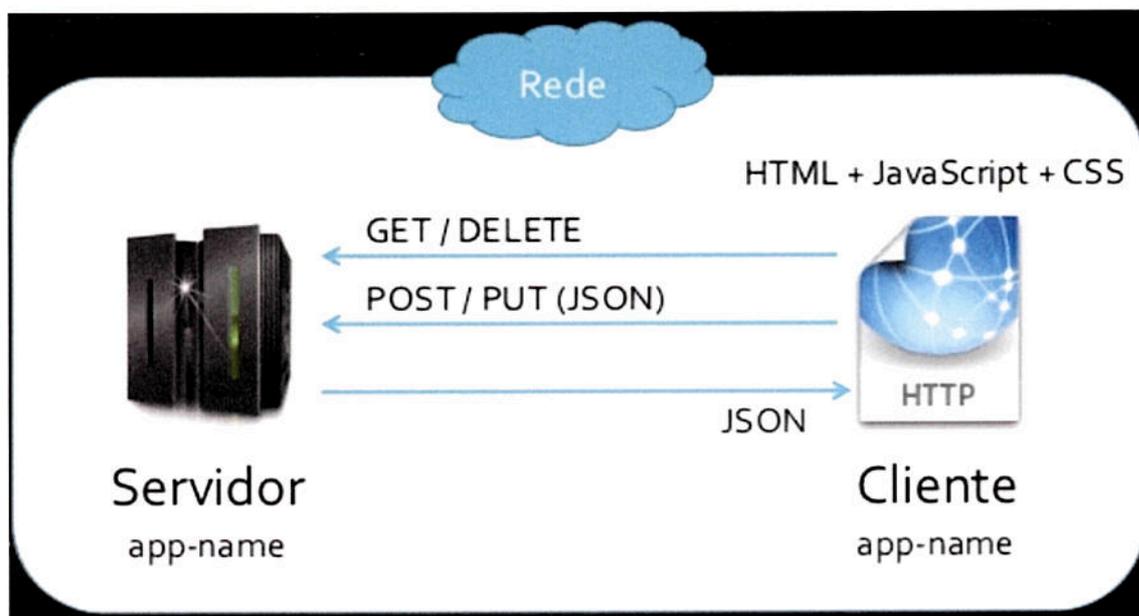


Ilustração 2: Diagrama sistema REST com JSON

### 3.6. DESENVOLVIMENTO DO PROJETO

O projeto GarçomNET foi iniciado aproximadamente em outubro de 2012, com a análise da situação feita pelos integrantes do grupo, onde os requisitos do sistema foram levantados, discutidos e avaliados em utilidade, importância, complexidade e custo homem/hora.

Com as diretrizes determinadas no decorrer do final de outubro, deu-se em novembro o início do projeto do sistema, e um modelo de fluxo das telas do aplicativo foi elaborado utilizando-se a ferramenta Prototype (PROTOTYPE, 2012),

para que o grupo tivesse uma noção geral de como o aplicativo poderia ser. Também foram iniciados os estudos para instalação e configuração de um Web service, o qual deveria utilizar o protocolo REST e trocar dados no formato JSON. Além do programa do estabelecimento, o qual desde o início teve a utilização de threads para gerar um número aleatório e a imprimir este valor num formulário na GUI. Este representava inicialmente o pedido de múltiplos clientes.

Por volta das três semanas seguintes, um protótipo de WS foi elaborado, o qual realizava a operação de inserção de um pedido numa base de dados. Juntamente com isso, a documentação sobre o Web service começou a ser elaborada. Porém, nesse período o projeto estava se encaminhando para uma diretriz errônea, visto que o servidor criado utilizava protocolo SOAP e trafegava arquivos XML, o que dificultaria o desenvolvimento futuro do GarçomNET.

Nos meses de novembro e dezembro de 2012 ocorreram várias discussões sobre funcionalidades das plataformas e fluxos das telas no aplicativo. Além disso, houve problemas consecutivos na instalação e configuração das ferramentas de desenvolvimento, as quais foram: Eclipse IDE, Android SDK, Apache Tomcat, Workbench, MySQL e Dropbox para o compartilhamento dos arquivos na nuvem.

Apesar do Web service SOAP ter sido criado, os integrantes sabiam que aquela solução não era a ideal, e continuaram a pesquisar sobre o assunto. Por volta do início do mês de janeiro de 2013, os membros do grupo conseguiram configurar o WS com REST e JSON, conforme foi proposto nas diretrizes iniciais. Esse fato proporcionou a aceleração no processo de desenvolvimento e integração das plataformas. Além de juntamente no mês de janeiro, o banco de dados ter sido modelado, configurado e alimentado para que as aplicações pudessem realizar as operações necessárias.

Em fevereiro o desenvolvimento do sistema se desenrolou na elaboração das telas do aplicativo, na criação e integração do software do estabelecimento com o Web service, além das melhorias realizadas nas operações do WS juntamente com o banco de dados. Em paralelo, a documentação do trabalho começou a ser formatada e alimentada com trabalhos relacionados e tecnologias utilizadas.

Na última semana de fevereiro e início de março, foram realizados os ajustes finais nas plataformas, como por exemplo a validação de usuário no aplicativo

Android e o aperfeiçoamento das telas e multiconexões ao servidor. Visto que, houve problema na conexão de vários usuários simultaneamente ao WS e para solucionar esse problema, o seguinte ambiente de teste foi montado:

- celular Android com o aplicativo instalado para que pedidos pudessem ser realizados;

- Web service rodando num notebook com Apache Tomcat e MySQL instalado;

- outro notebook com o *software* do estabelecimento no ar, para que este pudesse visualizar e editar o status dos pedidos.

Com a realização dos testes, o problema de multiconexões foi solucionado, pois foi verificada e alterada uma falha no algoritmo de conexão entre as plataformas. As linhas que solucionaram o problema, após serem inseridas, foram destacadas em negrito no código abaixo:

```
{
    .....
    Connection connection = new ConnectionFactory().getConnection();
    PreparedStatement stmt = null;
    ResultSet rs = null;
    String sql = " INSERT / UPDATE / DELETE ..... ";
    try {
        stmt=connection.prepareStatement(sql);
        stmt.execute();
        stmt.close();
        connection.close();
        System.out.println("Conexao fechada no servidor.");
    }
    .....
}
```

Com a inserção dessas linhas, as conexões que eram abertas no WS realmente passaram a ser fechadas após a realização da tarefa. O que, até então, não estava ocorrendo, e com isso o Web service lançava o erro de "too many connections".

Nas duas últimas semanas, juntamente com os ajustes nos sistemas, a documentação foi atualizada e completada com os requisitos exigidos pelo curso, sendo que os seguintes itens foram confeccionados: listas, resumo, metodologia, apresentação do software, considerações finais e apêndices. Quando então, o trabalho foi concluído e entregue ao professor orientador para que este pudesse avaliar e repassar aos outros avaliadores.

### 3.7. VALIDAÇÃO

#### 3.7.1. TESTES DO ESTABELECIMENTO

Caso de Teste	T001 – TESTAR LOGIN
Descrição do Caso de Teste	Teste de login do usuário.
Procedimento para Execução	1. Preencher campo nome com o valor : pizzahut@pizzahut.com.br 2. Preencher o campo senha com o valor: 6677
Resultado Esperado	1. O sistema deve retornar a mensagem "PizzaHut". 2. O sistema deve preencher o textField com "PizzaHut". 3. O sistema deve apresentar a tabela com os pedidos recebidos.
Resultado Obtido	<i>A ser preenchido quando o teste for executado.</i>

Caso de Teste	T002 – MUDAR STATUS NA FILA – PREPARANDO
Descrição do Caso de Teste	Teste de Mudança de Status NA FILA – PREPARANDO.
Procedimento para Execução	1. Clicar no campo status da tabela de pedidos na aba "Na Fila".
Resultado Esperado	1. O sistema deve retornar a mensagem "Preparando". 2. O programa deve carregar o pedido na ultima linha da tabela "Preparando".
Resultado Obtido	<i>A ser preenchido quando o teste for executado.</i>

Caso de Teste	T003 – MUDAR STATUS PREPARANDO – PRONTO
descrição do Caso de Teste	Teste de Mudança de Status PREPARANDO – PRONTO.
Procedimento para Execução	1. Clicar no campo status da tabela de pedidos na aba “Preparando”.
Resultado Esperado	1. O sistema deve retornar a mensagem “Pronto”. 2. O sistema deve carregar o pedido na última linha da tabela “Pronto”.
Resultado Obtido	<i>A ser preenchido quando o teste for executado.</i>

Caso de Teste	T004 – MUDAR STATUS PRONTO – ARQUIVADO
Descrição do Caso de Teste	Teste de Mudança de Status PRONTO – ARQUIVADO.
Procedimento para Execução	1. Clicar no campo status da tabela de pedidos na aba “Pronto”.
Resultado Esperado	1. O sistema deve retornar a mensagem “Arquivado”. 2. O sistema deve gravar no banco o pedido com status 4.
Resultado Obtido	<i>A ser preenchido quando o teste for executado.</i>

Caso de Teste	T005 – GERAR RELATÓRIO
Descrição do Caso de Teste	Teste de geração de relatórios.
Procedimento para Execução	1. Preencher o campo data Inicial com o valor: “03/03/2013”. 2. Preencher o campo data Final com o valor: “05/03/2013”. 3. Clicar na lupa de pesquisa. 4. Clicar no botão de geração de relatórios.
Resultado Esperado	1. O sistema deve preencher a tabela com os dados do período. 2. O sistema deve gerar o relatório e apresentar na tela.
Resultado Obtido	<i>A ser preenchido quando o teste for executado.</i>

## 3.7.2. TESTES DO APLICATIVO

Caso de Teste	T006 – REALIZAR PEDIDO
Descrição do Caso de Teste	Teste de Realização de um pedido.
Procedimento para Execução	<ol style="list-style-type: none"> <li>1. Clicar em “Fazer Pedido”.</li> <li>2. Selecionar “Curitiba”.</li> <li>3. Selecionar “Mc Donalds – Centro”.</li> <li>4. Selecionar “Big Mac”.</li> <li>5. Preencher o campo quantidade com o valor de “12”.</li> <li>6. Adicionar ao carrinho.</li> <li>7. Pressionar Finalizar.</li> </ol>
Resultado Esperado	<ol style="list-style-type: none"> <li>1. O sistema deve gravar o pedido no banco de dados via <i>webservice</i>.</li> <li>2. O sistema deve retornar à tela principal.</li> </ol>
Resultado Obtido	<i>A ser preenchido quando o teste for executado.</i>

Caso de Teste	T007 – LOGIN
Descrição do Caso de Teste	Teste de login do usuário.
Procedimento para Execução	<ol style="list-style-type: none"> <li>1. Preencher o campo nome com o valor “nilson@gmail.com”.</li> <li>2. Preencher o campo senha com o valor “2345”.</li> </ol>
Resultado Esperado	<ol style="list-style-type: none"> <li>1. O sistema deve logar no sistema.</li> <li>2. O sistema deve retornar à tela inicial.</li> </ol>
Resultado Obtido	<i>A ser preenchido quando o teste for executado.</i>

## 4. APRESENTAÇÃO DO SOFTWARE

O GarçomNET possui três plataformas integradas que utilizam a Internet para o tráfego de dados. Porém, como o Webservice ainda não foi disponibilizado na Internet, para que haja o pleno funcionamento do Aplicativo, Webservice e Sistema do Estabelecimento, essas plataformas devem estar conectadas juntas na mesma rede interna com faixa de IP 192.168.1.0/24. Sendo que, a máquina hospedeira do Webservice deve conter o IP 192.168.1.2, para que os clientes consigam visualizar e utilizar o servidor na rede.

### 4.1. INSTALAÇÃO, CONFIGURAÇÃO E EXECUÇÃO

#### 4.1.1. WEBSERVICE

Para que o GarçomNET funcione corretamente, a máquina hospedeira do sistema deve conter os seguintes pré-requisitos:

1. Apache Tomcat instalado e executando com Port Number = 8085 para HTTP/1.1
2. MySQL instalado.
3. Eclipse IDE Indigo instalado.

Para instalação do servidor, basta seguir os seguintes passos:

1. Copie a pasta "/Servidor" do CD-ROM disponibilizado para o disco da máquina hospedeira.
2. Abra o Eclipse e importe o projeto da pasta copiada do CD-ROM. Para isso vá em File → Import → General → Existing Projects into Workspace, e selecione como "root directory" a pasta copiada do CD. Após importado, o projeto aparecerá na lista de projetos locais.
3. Copie o arquivo "/dump\_GarcomNET.sql" do CD-ROM disponibilizado para o disco local do servidor.
4. Execute o dump no banco de dados do Webservice com o seguinte comando pelo terminal: `mysql -u usuario -p garcomnet < dump_GarcomNET.sql`

5. Execute o projeto com o servidor Apache existente. Para isso clique com o botão direito no projeto Servidor → Run as → Run on Server, inclua o projeto na lista de projetos a serem executados pelo Apache e execute o serviço.

6. Pronto. O Webservice está pronto para ser utilizado. Para verificar se o serviço está funcionando corretamente, basta acessar de um navegador o endereço <http://192.168.1.2:8085/Servidor/pedido> para que o sistema retorne na tela uma string em formato JSON.

#### 4.1.2. APLICATIVO DO DISPOSITIVO MÓVEL

O dispositivo móvel para rodar o GarçomNET deve possuir sistema operacional Android igual ou superior à versão 2.3 – Gingerbread. Para a instalação do aplicativo, basta seguir os seguintes passos:

1. Copie o arquivo de instalação “/GarcomNET.apk” do CD-ROM de instalação disponibilizado para o dispositivo móvel;
2. Selecione o arquivo copiado para o dispositivo e abra as opções;
3. Selecione a opção “Instalar o aplicativo”;
4. Após o término da instalação, o aplicativo pode ser executado clicando-se no ícone do mesmo que aparece no menu do dispositivo;
5. Pronto. O aplicativo já pode ser utilizado.

#### 4.1.3. SOFTWARE DO ESTABELECIMENTO

Para que o sistema funcione corretamente no estabelecimento, o computador deve possuir Java JRE versão 7.0 ou superior instalado. Com isso, basta seguir os seguintes passos para instalação:

1. Copie o arquivo “Estabelecimento\_GarcomNET.jar” do CD-ROM de instalação disponibilizado para o disco local do computador.
2. Execute o arquivo com a máquina virtual Java. Para isso, basta clicar com o botão direito no arquivo → Abrir com → e selecionar o Runtime Java instalado.

3. Pronto. O sistema do estabelecimento já pode ser utilizado. Um login válido que pode ser utilizado no sistema é o seguinte:

usuário: McDonalds@McDonalds.com.br

senha: 5677

## 4.2. PROTÓTIPO DE TELAS

### 4.2.1. TELAS DO APLICATIVO MÓVEL



Ilustração 3: Menu

A Ilustração 3 apresenta a Tela Menu. Nesta o usuário pode acessar todas as principais funcionalidades do sistema. Nesta tela pode-se realizar ou acompanhar um pedido, pedir um destaque ou um favorito, realizar o login, ou visualizar a lista de estabelecimentos do sistema.



Ilustração 4: Lista  
Cidades

A Ilustração 4 apresenta a Tela Lista Cidades. Nesta o usuário seleciona a cidade do estabelecimento em que o pedido será realizado. O usuário também pode filtrar a lista digitando todo ou parte do nome da cidade.



Ilustração 5: Cardápio

A Ilustração 5 apresenta a Tela Cardápio. Nesta o usuário pode visualizar todos os produtos de determinado estabelecimento, e então, pode selecionar o produto que deseja pedir.



Ilustração 6: Produto

A Ilustração 6 apresenta a Tela Produto. Nesta o usuário pode visualizar os detalhes de um determinado produto. Pode dizer a quantidade que deseja do produto naquele pedido, e também pode inserir qualquer observação referente àquele produto que desejar naquele pedido.



Ilustração 7: Carrinho

A Ilustração 7 apresenta a Tela Carrinho. Nesta o usuário pode visualizar os produtos inseridos no carrinho, além de poder excluir algum produto, caso deseje. E depois de checar os produtos, o usuário pode finalizar o pedido.

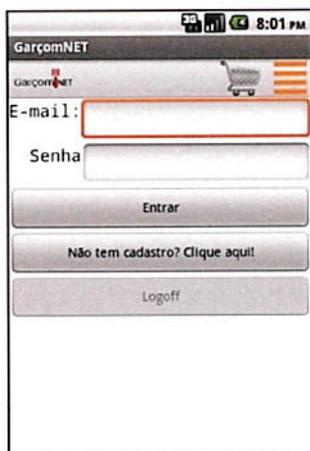


Ilustração 8: Login

A Ilustração 8 apresenta a Tela Login. Nesta o usuário pode realizar login ou logoff, ou então, ele pode realizar o cadastro caso ainda não possua uma conta.

Ilustração 9: Cadastro  
Usuário 1Ilustração 10:  
Cadastro Usuário 2

As Ilustrações 9 e 10 apresentam as Telas de Cadastro de Usuário. Nestas o usuário insere seus dados pessoais e os envia ao servidor para criar uma nova conta.

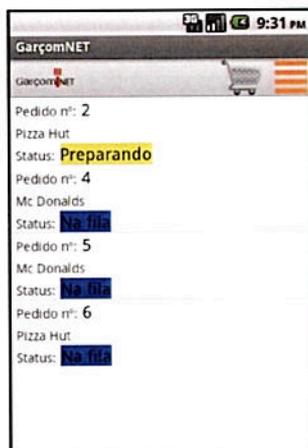


Ilustração 11:  
Acompanhamento

A Ilustração 11 apresenta a Tela Acompanhamento. Nesta o usuário pode acompanhar os pedidos já realizados verificando o status de cada um. Ele também tem a opção de clicar em algum pedido para visualizar os produtos solicitados no mesmo.

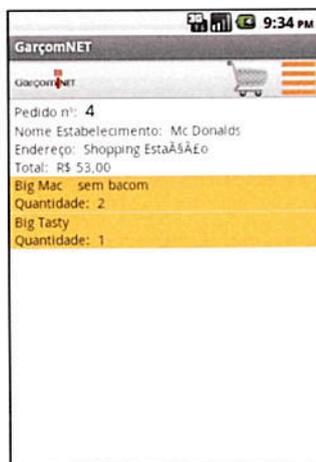


Ilustração 12: Produtos  
Pedidos

A Ilustração 12 apresenta a Tela Produtos Pedidos. Nesta o usuário pode visualizar os produtos solicitados em determinado pedido.



Ilustração 13:  
Favoritos

A Ilustração 13 apresenta a Tela Favoritos. Nesta o usuário pode visualizar os produtos que havia selecionado previamente como favoritos de sua conta. Também há a possibilidade de excluir determinado produto da lista de favoritos.

#### 4.2.2. TELAS DO ESTABELECIMENTO

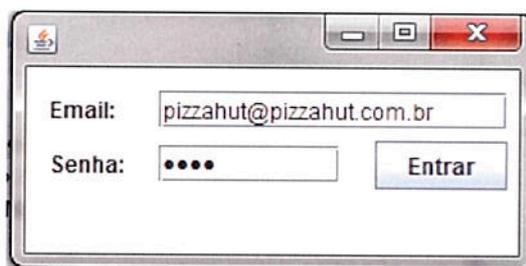


Ilustração 14: Login Estabelecimento

A Ilustração 14 apresenta a Tela Login Estabelecimento. Nesta o operador do sistema irá realizar o login no sistema.

N° Pedido	Status	Conta Usuário	Total	Produtos
1	Na fila	Roger Campos	R\$ 16.00	

Ilustração 15: Gerenciamento Estabelecimento

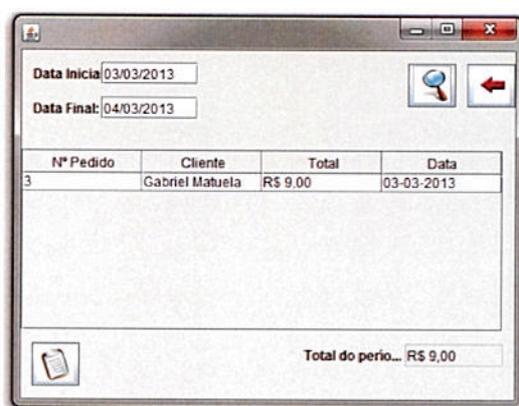
A Ilustração 15 apresenta a Tela Gerenciamento Estabelecimento. Nesta o operador pode visualizar os pedidos juntamente com seu respectivo status, podendo alterar a situação de cada um.

Produto	Quantidade
Big Mac	1
Batata Frita	1

Informações do Pedido  
sanduba sem cebola

Ilustração 16: Produtos Pedido

A Ilustração 16 apresenta a Tela Produtos Pedidos. Esta aparece quando o operador do estabelecimento seleciona um determinado pedido. Nesta tela, são listados os produtos do referido pedido, juntamente com as observações de cada um, caso haja.



Nº Pedido	Cliente	Total	Data
3	Gabriel Matuela	R\$ 9,00	03-03-2013

Total do período... R\$ 9,00

Ilustração 17: Pedidos Arquivados

A Ilustração 17 apresenta a Tela Pedidos Arquivados. Nesta o operador pode visualizar os pedidos arquivados do determinado período selecionado, além de poder gerar um relatório PDF das informações.

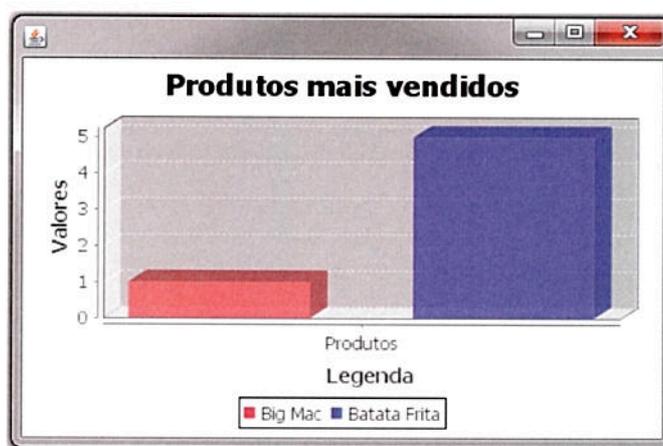


Ilustração 18: Gráfico Produtos

A Ilustração 18 apresenta a Tela Gráfico Produtos. Nesta pode ser visualizado um gráfico com os produtos mais vendidos juntamente com a respectiva quantidade vendida de cada um.

## 5. CONSIDERAÇÕES FINAIS

Devido a mobilidade explorada de um dispositivo móvel, foi possível possibilitar aos clientes uma forma divertida, cômoda e prática para realizar um pedido. Causa disso foi a instalação do GarçomNET nos *mobiles*, o qual possui uma interface amigável, ergonômica e intuitiva. Com isso, o objetivo da plataforma de interligar clientes e estabelecimentos foi atingido com êxito, evitando assim que os clientes enfrentassem filas para realizar um pedido.

No sistema do estabelecimento, a eficácia e simplicidade de operação foram o diferencial. Dessa forma, o operador do sistema não necessita de muito treinamento, o que evita gastos e possíveis rejeições.

A integração das plataformas foi outro objetivo atingido. Visto que, com o desenvolvimento do Web service, proporcionou-se uma unificação entre os sistemas dos clientes e estabelecimentos. Isso devido aos métodos disponibilizados no servidor para realização e acompanhamento de pedidos.

Por outro lado, além dos objetivos já atingidos, o projeto GarçomNET visa alguns trabalhos futuros, tendo em mente aumentar a credibilidade e funcionalidade da plataforma. Esses trabalhos podem ser resumidos na futura inclusão dos seguintes itens, expandir o sistema para outras plataformas como IOS, BlackBerry e Windows Phone; disponibilizar a opção de *Delivery*, para que os clientes possam fazer e receber os pedidos sem sair de casa; implantar a opção GPS, para que os clientes possam verificar os estabelecimentos conveniados mais próximos, e então, realizar pedidos nos mesmos; incluir no pedido a opção Agendar Horário, que permitirá ao usuário agendar um horário para retirada do pedido; integrar o GarçomNET aos sistemas nativos dos estabelecimentos, de forma que a plataforma seja personalizada e otimizada.

Com isto, o GarçomNET poderá se tornar uma plataforma integradora completa para automatizar pedidos. Com a popularização desse aplicativo, não haverá mais filas nos estabelecimentos, o que poderá modificar processos em restaurantes, além de proporcionar economia para os gerentes e clientes, tanto de tempo quanto de dinheiro.

## 6. REFERÊNCIAS

ABI RESEARCH. Technology Market Intelligence, 45 Million Windows Phone and 20 Million BlackBerry 10 Smartphones in Active Use at Year-end; Enough to Keep Developers Interested. Disponível em:

<http://www.abiresearch.com/press/45-million-windows-phone-and-20-million-blackberry>

Acessado em: 09/02/2013

ANDROID: Open Source Project. Dalvik technical Information. Disponível em:

<http://source.android.com/tech/dalvik/index.html>

Acessado em: 12/12/2012

DE LIMA Vanessa Gomes, RÉGO Filipe M. M., TITO Adriano de Oliveira, SANTOS Rodolfo A. S., RISTAR Arley Ramalho R., SANTOS Luana Martins S., VIEIRA Vaninha, SALGADO Ana Carolina. Um Sistema de Identificação e Sugestão de Rotas de Ônibus Baseado em Informações de Redes Sociais. Disponível em:

<http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2012/0050.pdf>

Acessado em: 09/02/2013

DEX2JAR. Tools to work with android .dex and java .class files. Disponível em:

<http://code.google.com/p/dex2jar/wiki/DalvikVSJvm>

Acessado em: 12/12/2012

FIELDING, Roy. 2000. Architectural Styles and the Design of Network – Chapter 5 and 6 (REST). Disponível em:

<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

Acessado em: 27/02/2013

FIT TECNOLOGIA. Rossana M. C. Andrade, Márcio E. F. Maia e Paulo Henrique M. Maia – Desenvolvendo aplicações móveis integradas às redes sociais para o suporte de turistas no Brasil. Disponível em:

<http://www.fit-tecnologia.org.br/press-releases/rossana-m-c-andrade-desenvolvendo-aplicacoes-moveis-integradas-a-redes-sociais-para-o-suporte>

Acessado em: 27/02/2013

IDGNOW, 2012. Chega a 640 milhões o número de dispositivos iOS e Android em uso. Disponível em:

<http://idgnow.uol.com.br/mobilidade/2012/08/28/chega-a-640-milhoes-o-numero-de-dispositivos-ios-e-android-em-uso/>

Acessado em: 02/03/2013

JSON. Introdução ao JSON. Disponível em:

<http://www.json.org/json-pt.html>

Acessado em: 01/03/2013

JUNIOR Claudinei Rodrigues, MATOS Simone Nasser. Coordenação de Informática - UTFPR. Um Web Service para busca de Preços de Produtos. Disponível em:

<http://200.134.81.19:8080/arcabomk/Itens/Resultados/Artigos/ART.RODRIGUES.SEMINCO.2010.pdf>

Acessado em: 06/03/2013

LECHETA, Ricardo R.. Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 2ª Edição. São Paulo, 2010. 352p.

LINHADECODIGO. Linha de Código. Conhecendo JSON. Disponível em:

<http://www.linhadecodigo.com.br/artigo/3623/conhecendo-json.aspx>

Acessado em: 01/03/2013

MACEDO, Alexandre. K19 Treinamentos, Webservice Restful utilizando JSON. Disponível em:

<http://www.k19.com.br/artigos/webservice-restful-utilizando-json/>

Acessado em: 09/02/2013

MOBILIDADE FM. Máquina Virtual Dalvik. Disponível em:

<http://mobilidade.fm/iniciante-android/2011/05/maquina-virtual-dalvik/>

Acessado em: 13/12/2012

PRESTES João Paulo Moreira, HÖLBIG Carlos Amaral, PAVAN Willingthon, FERNANDES José Maurício Cunha. Web Service para previsão de tempo e clima através de dados georeferenciados. Disponível em:

<http://www.upf.br/seer/index.php/rbca/article/view/1371>

Acessado em: 06/03/2013

PROTOTYPE. A Foundation for ambitious web user interfaces. Disponível em:

<http://prototypejs.org/>

Acessado em: 15/10/2012

RAMOS, Junior Durval. Reportagem TecTudo, Android vai dominar o mercado de smartphones em 2013. Disponível em:

<http://www.tecmundo.com.br/android/36378-android-vai-dominar-o-mercado-de-smartphones-em-2013.htm>

Acessado em: 09/02/2013

W3SCHOOLS. Introduction to Web Services. Disponível em:

[http://www.w3schools.com/webservices/ws\\_intro.asp](http://www.w3schools.com/webservices/ws_intro.asp)

Acessado em: 28/02/2013

## 7. APÊNDICES

### 7.1. DIAGRAMA DE CASOS DE USO

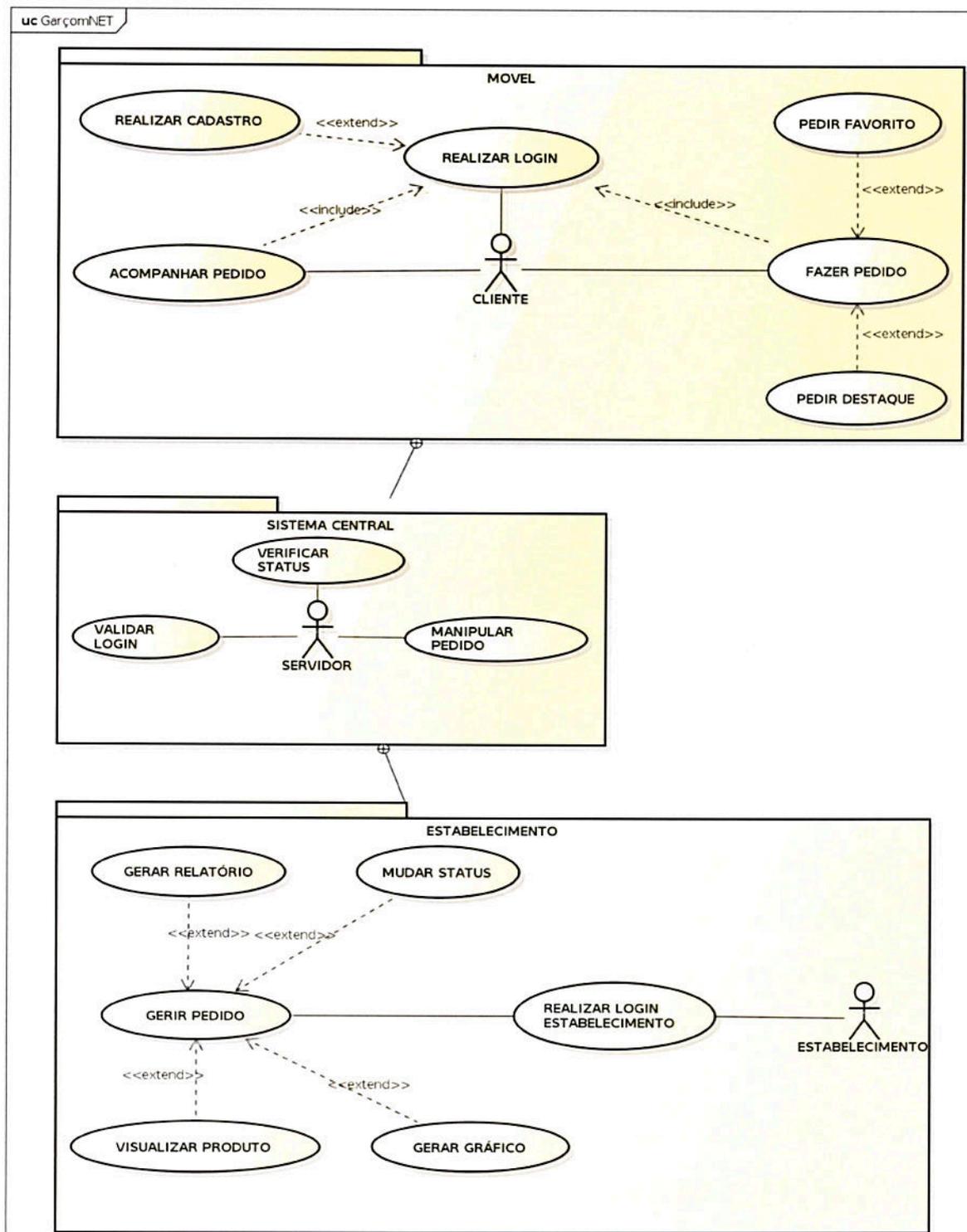


Ilustração 19: Diagrama de Casos de Uso

## 7.2. ESPECIFICAÇÕES DE CASOS DE USO

### 7.2.1. UC001 – FAZER PEDIDO

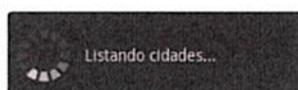
Descrição: Este caso de uso serve para fazer um pedido pelo aplicativo.

Data View:

#### DV1 – Tela Menu



#### DV2 – Alerta Listando Cidades

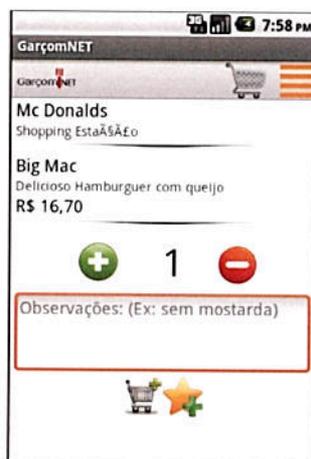


#### DV3 – Tela Lista Cidades





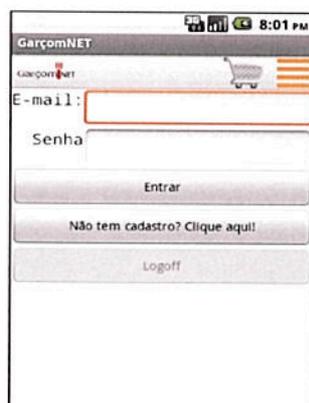
## DV7 – Tela Produto



## DV8 – Tela Carrinho



## DV9 – Tela Login



Pré-condições: não há.

Pós-condições: o sistema deve:

1. Ter enviado para o webservice o pedido.
2. Retornar para a tela (DV1).

Ator Primário: Usuário.

Fluxo de Eventos Principal:

1. O sistema apresenta a tela (DV1).
2. O usuário seleciona a opção Fazer Pedido.
5. O sistema apresenta a tela (DV3). (R1)
6. O usuário seleciona a cidade do estabelecimento no qual será realizado o pedido. (R2)
7. O sistema apresenta a tela (DV5). (R3)
8. O usuário seleciona o estabelecimento que deseja fazer o pedido. (R4)
9. O sistema apresenta a tela (DV6). (E1)
10. O usuário seleciona o produto que deseja pedir. (A1)
11. O sistema apresenta a tela (DV7). (E1)
12. O usuário coloca a quantidade do produto que deseja.
13. O usuário pressiona o botão Adicionar.
14. O sistema valida o campo quantidade. (E2)
15. O sistema insere o produto com a respectiva quantidade no carrinho do cliente.
16. O sistema fecha a tela (DV7).
17. O sistema apresenta a tela (DV6). (E1)
18. O usuário seleciona Meu Carrinho. (A3)
19. O sistema apresenta a tela (DV8). (E1)
20. O usuário pressiona Finalizar. (A2)
20. O sistema valida o login do usuário. (E3)
21. O sistema envia o pedido para o webservice. (E1)
21. O sistema retorna para a tela (DV1).
22. O caso de uso é finalizado.

#### Fluxos Alternativos:

##### A1: Selecionado Meu carrinho:

1. O sistema apresenta a tela (DV8).
2. O sistema preenche a lista dos produtos que estão no carrinho. (A2)

##### A2: Pressionado o botão X:

1. O sistema exclui o respectivo produto do carrinho.

##### A3: Selecionado novo produto:

1. O UC atual retorna para o passo 11.

#### Fluxos de Exceção:

##### E1: Caso ocorra falha na conexão ao webservice:

1. O sistema deve apresentar a mensagem “Falha na conexão ao servidor GarçomNET, verifique sua conexão à internet”.

##### E2: Caso o campo quantidade não seja preenchido corretamente:

1. O sistema retorna a mensagem: “ A quantidade desejado do produto deve ser inserida corretamente”.
2. O sistema retorna para a tela (DV7) com o campo quantidade limpo.

##### E3: Caso o usuário não esteja logado no sistema:

1. O UC002 – REALIZAR LOGIN é iniciado.

#### Regras de Negócio:

R1. Enquanto o sistema carrega do *webservice* a lista de cidades, é apresentada ao usuário a tela (DV2) sobre a tela (DV3). (E1)

R2. A medida que o usuário digita no campo texto da tela, o sistema filtra a lista de cidades para que apareçam somente as que o nome confira parcialmente ou totalmente ao digitado.

R3. Enquanto o sistema carrega do *webservice* a lista de estabelecimentos, é apresentada ao usuário a tela (DV4) sobre a tela (DV5). (E1)

R4. A medida que o usuário digita no campo texto da tela, o sistema filtra a lista de estabelecimentos para que apareçam somente os que o nome confira parcialmente ou totalmente ao digitado.

R5. O sistema valida o campo quantidade, verificando se realmente foi um número natural inserido. (E1)

## 7.2.2. UC002 – REALIZAR LOGIN

Descrição: Este caso de uso tem a finalidade de logar o usuário no aplicativo.

Data View:

DV10 – Tela Cadastrar Usuário 1



The screenshot shows the registration form for 'GarçomNET'. The form includes the following fields: 'Nome completo', 'E-mail', 'CPF', 'Senha', 'Confirmar senha', and 'Endereço'. A 'Próximo >>' button is located at the bottom of the form. The status bar at the top shows the time as 9:37 PM.

DV11 – Tela Cadastrar Usuário 2



The screenshot shows the registration form for 'GarçomNET' with the following information displayed: 'Nome: Fulano de Tal', 'Email: fulano@gmail.com', 'CPF: 02933322392', 'Endereço: Rua 24 de mai', 'Cidade:', 'Bairro:', 'UF:', and 'Bairro'. A 'Cadastrar' button is located at the bottom of the form. The status bar at the top shows the time as 9:40 PM.

Pré-Condições: Não há.

Pós-Condições: Após o usuário logar, o sistema deve ter retornar para a tela ([DV1](#)).

Ator Primário: Usuário

Fluxo Principal:

1. O sistema apresenta a tela ([DV9](#)).
2. O usuário preenche o campo e-mail.

3. O usuário preenche o campo senha.
4. O usuário pressiona o botão entrar. (A1)(A2)
5. O sistema valida os dados inseridos.
6. O sistema desabilita o botão entrar.
7. O sistema redireciona para a tela (DV1).
8. O caso de uso é finalizado.

#### Fluxo Alternativo

A1: Botão “Não tem cadastro? Clique aqui” selecionado:

1. O sistema apresenta a tela (DV10).
2. O usuário preenche o campo nome
3. O usuário preenche o campo email.
4. O usuário preenche o campo senha.
5. O usuário preenche o campo confirmar senha.
6. O usuário preenche o campo endereço.
7. O usuário pressiona o botão Próximo.
8. O sistema apresenta a tela (DV11).
9. O sistema preenche os campos digitados pelo usuário. (R1)
10. O usuário preenche o nome do bairro no campo texto. (R2)
11. O usuário seleciona o bairro correto.
12. O sistema preenche o campos restantes. (R3)
13. O usuário pressiona o botão Cadastrar.
14. O sistema grava o novo usuário no *webservice*. (E1)
15. O Sistema direciona para a tela (DV1).
16. O caso de uso é finalizado.

A2: Botão “Logoff” selecionado:

1. O sistema finaliza sessão do usuário.
2. O sistema direciona para a tela (DV9).
3. O sistema desabilita o botão “Logoff”.

Fluxos de Exceção:

E1: Caso ocorra falha na conexão ao *webservice*:

1. O sistema deve apresentar a mensagem “Falha na conexão ao servidor GarçomNET, verifique sua conexão à internet”.

Regras de Negócio:

R1: O sistema preenche os campos Nome, Email, CPF e Endereço com os dados preenchidos pelo usuário na tela anterior.

R2: A medida que o usuário insere caracteres, o sistema busca no *webservice* bairros que confirmam com o digitado e os apresentam numa lista abaixo do campo texto.

R3: O sistema preenche os campos Cidade, Bairro e UF com os dados do bairro selecionado pelo usuário.

### 7.2.3. UC003 – ACOMPANHAR PEDIDO

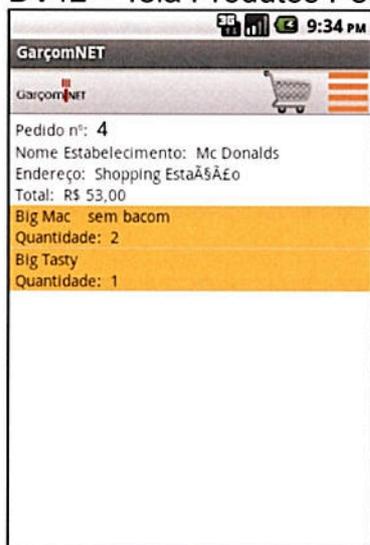
Descrição: Este caso de uso tem a finalidade de mostrar a situação do(s) pedido(s) feito(s) pelo usuário.

Data View:

DV11 – Tela Acompanhamento



## DV12 – Tela Produtos Pedidos



Pré-Condições: O usuário deve estar logado.

Pós-Condições: Não há.

Ator Principal: Usuário

Fluxo Principal

1. O sistema verifica o login do usuário. (E1)
2. O sistema apresenta a tela (DV11).
3. O sistema carrega todos os pedidos do usuário logado. (E1)
4. O usuário visualiza os pedidos na tela. (A1)(A2)(A3)

Fluxo Alternativo

A1: Pedido Selecionado:

1. O sistema busca os produtos do pedido selecionado. (E1)
2. O sistema apresenta os produtos na tela (DV12).
3. O caso de uso é finalizado.

A2: Botão "Carrinho" selecionado:

1. O Sistema apresenta a tela (DV8).
2. O usuário finaliza o pedido apresentado. (A4)(E1)
- 3 O caso de uso é finalizado.

A3: Botão "Menu" selecionado:

1. O sistema apresenta a tela (DV1).
2. O caso de uso é finalizado.

A4: Botão "X" pressionado:

1. O sistema exclui o produto da lista.

Fluxos de Exceção:

E1: Caso ocorra falha na conexão ao webservice:

1. O sistema deve apresentar a mensagem "Falha na conexão ao servidor GarçomNET, verifique sua conexão à internet".

#### 7.2.4. UC004 – PEDIR DESTAQUE

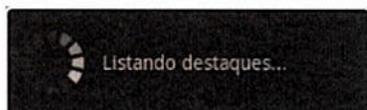
Descrição: Este caso de uso tem a finalidade de realizar o pedido de um produto em destaque no aplicativo.

Data View:

DV13 – Tela Destaques



## DV14 – Alerta Destaques



Pré-Condições: Não há.

Pós-Condições: Não há.

Ator Principal: Usuário

Fluxo Principal

1. O sistema apresenta a tela (DV13). (R1)
2. O usuário seleciona o produto desejado.
3. O UC001 – FAZER PEDIDO é iniciado. (R2)(E1)
4. O caso de uso atual é finalizado.

Fluxos de Exceção:

E1: Caso ocorra falha na conexão ao *webservice*:

1. O sistema deve apresentar a mensagem “Falha na conexão ao servidor GarçomNET, verifique sua conexão à internet”.

Regras de Negócio:

R1: Enquanto o sistema carrega do *webservice* a lista de destaques, é apresentada ao usuário a tela (DV14) sobre a tela (DV13). (E1)

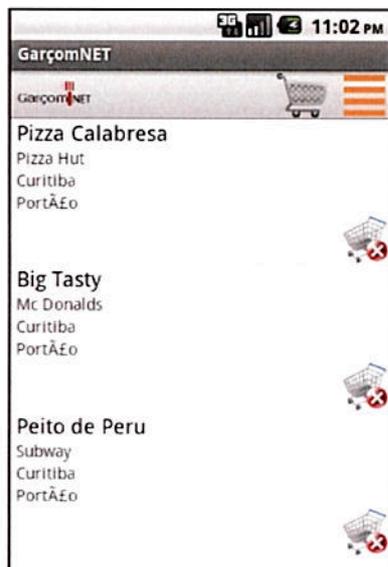
R2: O sistema passa como parâmetro o id do produto selecionado pelo usuário para o UC001 se iniciar no Passo 11 do Fluxo de Evento Principal.

### 7.2.5. UC005 – PEDIR FAVORITO

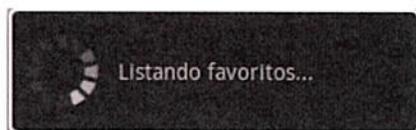
Descrição: Este caso de uso tem a finalidade de realizar o pedido de um favorito do usuário.

Data View:

## DV15 – Tela Favoritos



## DV16 – Alerta Favoritos



Pré-Condições: O usuário deve estar logado.

Pós-Condições: Não há.

Ator Principal: Usuário

Fluxo Principal

1. O sistema verifica o login do usuário. (E1)(E2)
1. O sistema apresenta a tela (DV15). (R1)
2. O usuário seleciona o produto desejado. (A1)
3. O UC001 – FAZER PEDIDO é iniciado. (R2)(E1)
4. O caso de uso atual é finalizado.

Fluxo Alternativo

A1: Botão "X" pressionado:

1. O sistema exclui o produto da lista de favoritos do usuário. (E1)
2. A lista é atualizada na tela. (E1)

Fluxos de Exceção:

E1: Caso ocorra falha na conexão ao *webservice*:

1. O sistema deve apresentar a mensagem “Falha na conexão ao servidor GarçomNET, verifique sua conexão à internet”.

E2: Caso o usuário não esteja logado:

1. O UC002 – REALIZAR LOGIN é iniciado.
2. O usuário realiza o login.
3. O sistema retorna ao caso de uso atual.

Regras de Negócio:

R1. Enquanto o sistema carrega do *webservice* a lista de favoritos, é apresentada ao usuário a tela (DV16) sobre a tela (DV15). (E1)

R2: O sistema passa como parâmetro o id do produto selecionado pelo usuário para o UC001 se iniciar no Passo 11 do Fluxo de Evento Principal.

#### 7.2.6. UC006 – REALIZAR LOGIN ESTABELECIMENTO

Descrição: Este caso de uso é utilizado para o acesso ao sistema.

Data View:

DV1 – Tela de Login:



Pré-Condições: Não há.

Pós-Condições: Após o fim normal deste caso de uso o sistema deve:

1. Ter chamado o UC007 – GERIR PEDIDO

Ator Primário: Usuário.

Fluxo de Evento Principal:

1. O sistema apresenta a tela (DV1).
2. O usuário preenche o campo "Email".
3. O usuário preenche o campo "Senha".
4. O usuário pressiona o botão "Entrar".
5. O sistema valida o usuário. (R1)(E1)
6. O sistema carrega o UC007 – GERIR PEDIDO.
7. O caso de uso é encerrado.

Fluxos Alternativos:

A1: Caso o login não seja válido:

1. O sistema apresenta a mensagem "Login inválido. Tente novamente."
2. O sistema retorna à tela (DV1) com os campos em branco.

Fluxos de Exceção:

E1: Caso ocorra falha na conexão ao *webservice*:

1. O sistema deve apresentar a mensagem "Falha na conexão ao servidor GarçomNET, verifique sua conexão à internet".

Regras de Negócio:

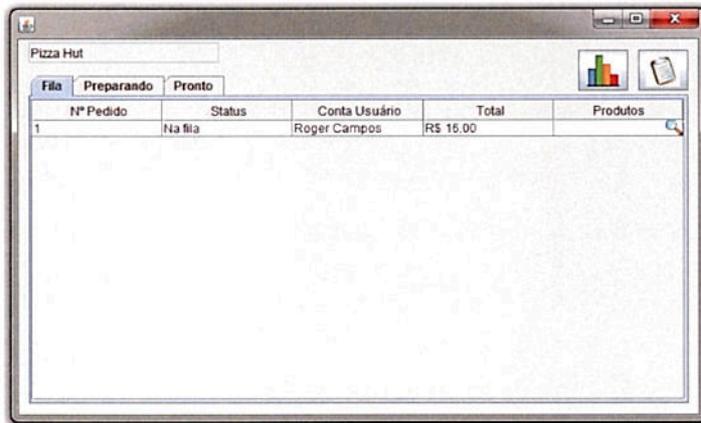
R1. O sistema envia o login ao *webservice* e recebe a validação do mesmo. (A1)

### 7.2.7. UC007 – GERIR PEDIDO

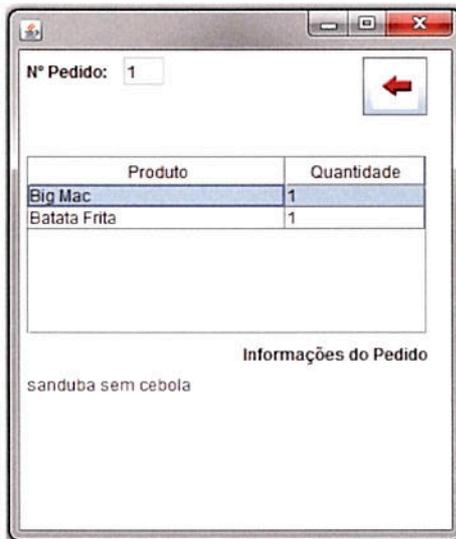
Descrição: Este caso de uso é utilizado para o gerenciamento e controle dos pedidos.

Data View:

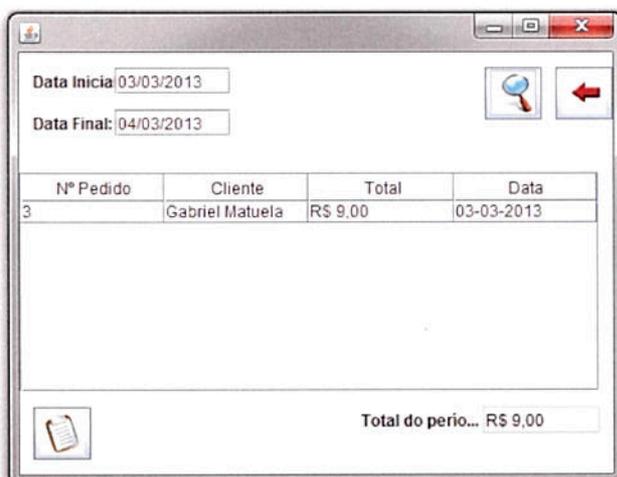
## DV1 – Tela de Gerenciamento



## DV2 – Tela de Produtos



## DV3 – Tela de arquivo



DV4 – Gráfico:



Pré-condições: Este caso de uso pode iniciar somente se:

1. O sistema tiver executado o UC006.

Pós-condições: Após o fim normal deste caso de uso o sistema deve:

1. Ter alterado os status dos pedidos selecionados.

Ator Primário: Usuário.

Fluxo de Eventos Principal:

1. O sistema preenche o campo com o nome do estabelecimento. (E5)
2. O sistema carrega os pedidos recebidos. (E5)
3. O sistema apresenta a tela (DV1).
4. O usuário altera o status dos pedidos. (A1)(A2)(A3)
5. O sistema muda o status na tela.
6. O sistema envia a alteração para o *webservice*. (E5)
7. O caso de uso é reiniciado. (E5)

Fluxo de Eventos Alternativo:

A1: O ícone  é selecionado:

1. O sistema apresenta a tela (DV2).
2. O sistema preenche o campo N° do pedido.
3. O sistema carrega os itens do pedido.
4. O usuário seleciona um produto
5. O sistema carrega a observação do referido produto no campo Informações do Pedido.

6. O usuário pressiona o botão  .

7. O caso de uso é reiniciado.

A2: O botão  pressionado:

1. O sistema apresenta a tela (DV3).

2. O usuário preenche o campo Data Inicial.

3. O usuário preenche o campo Data Final.

4. O usuário pressiona o botão  . (A4)

5. O sistema valida os campos da tela. (E1)(E2)(E3)(E4)

5. O sistema carrega os dados na tela.

6. O sistema preenche o campo "Total do período".

7. O usuário pressiona o botão  .

8. O sistema apresenta a tela (DV1)

A3: O botão  pressionado:

1. O sistema apresenta a tela (DV4).

2. O usuário pressiona o botão "X".

3. O caso de uso é reiniciado.

A4: O botão  pressionado:

1. O sistema gera o relatório em PDF.

2. O sistema apresenta o relatório na tela.

3. O caso de uso é reiniciado.

Fluxo de Exceções:

E1. Campo Data Inicial não preenchido:

1. O sistema apresenta a mensagem "Campo Data Inicial não preenchida".

2. O Caso de uso é reiniciado.

E2. Campo Data Final não preenchido:

1. O sistema apresenta a mensagem "Campo Data final não preenchida".

2. Caso de uso é reiniciado.

E3. Campo Data Inicial inválida:

1. O sistema apresenta a mensagem "Data Inicial Inválida".
2. Caso de uso é reiniciado.

E4. Campo Data Final inválida:

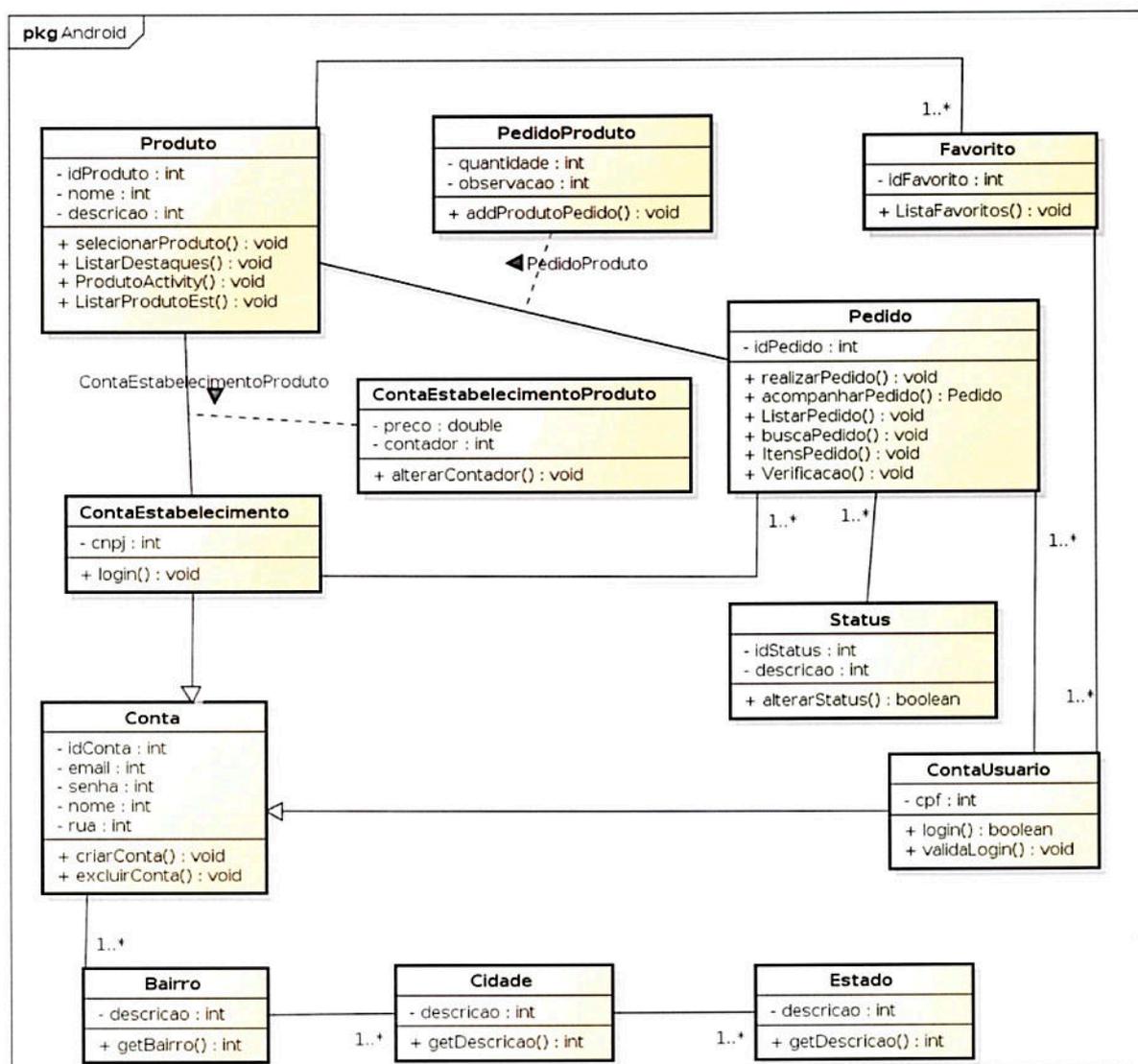
1. O sistema apresenta a mensagem "Data Final Inválida".
2. Caso de uso é reiniciado.

E5: Caso ocorra falha na conexão ao *webservice*:

1. O sistema deve apresentar a mensagem "Falha na conexão ao servidor GarçomNET, verifique sua conexão à internet".

## 7.3. DIAGRAMA DE CLASSES

## 7.3.1. DIAGRAMA DE CLASSES CLIENTE



powered by Astah

Ilustração 20: Diagrama de Classes Cliente

## 7.3.2. DIAGRAMA DE CLASSES SERVIDOR

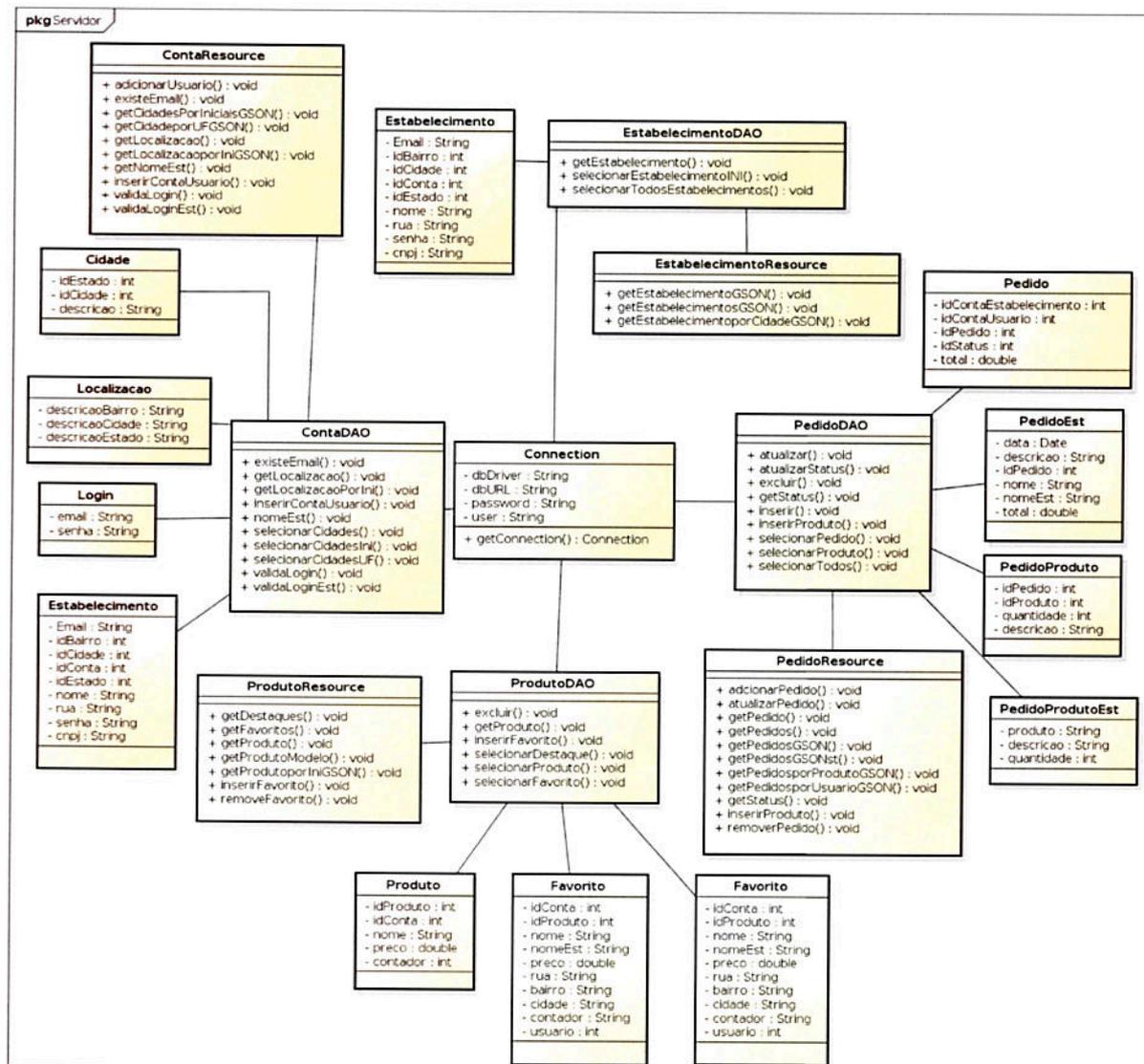
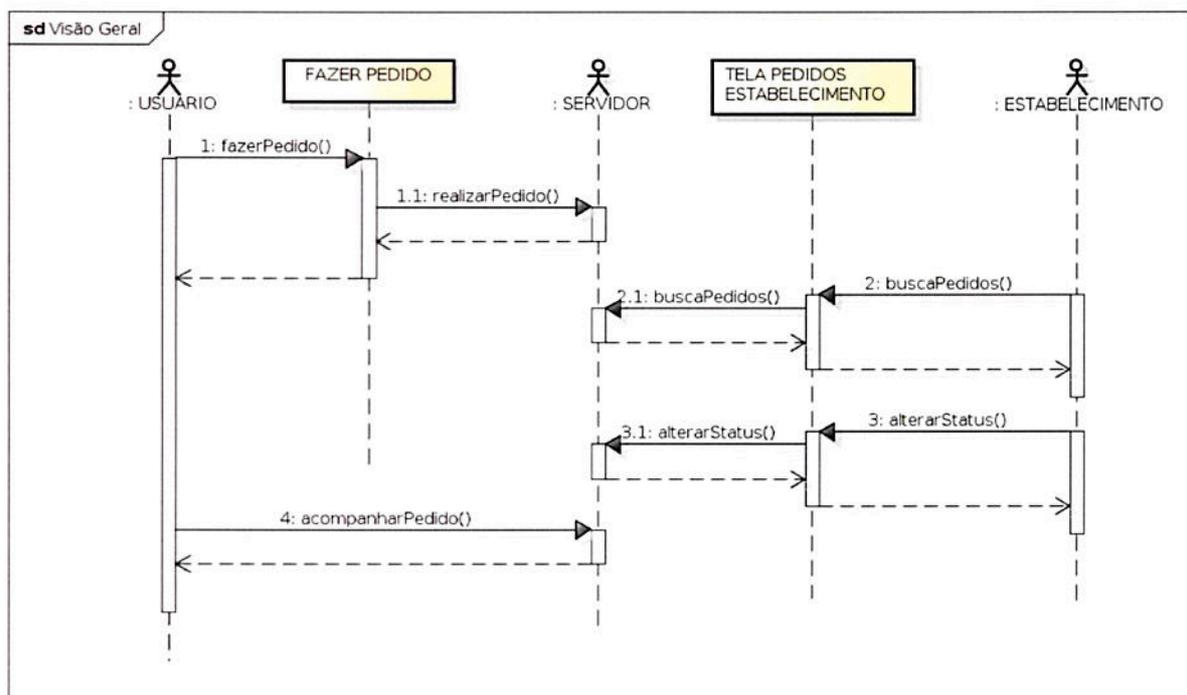


Ilustração 21: Diagrama de Classes Servidor

## 7.4. DIAGRAMAS DE SEQUÊNCIA

### 7.4.1. DIAGRAMA DE SEQUÊNCIA VISÃO GERAL



powered by Astah

Ilustração 22: Diagrama de Sequência Visão Geral

## 7.4.2. DIAGRAMA DE SEQUÊNCIA APLICATIVO MÓVEL

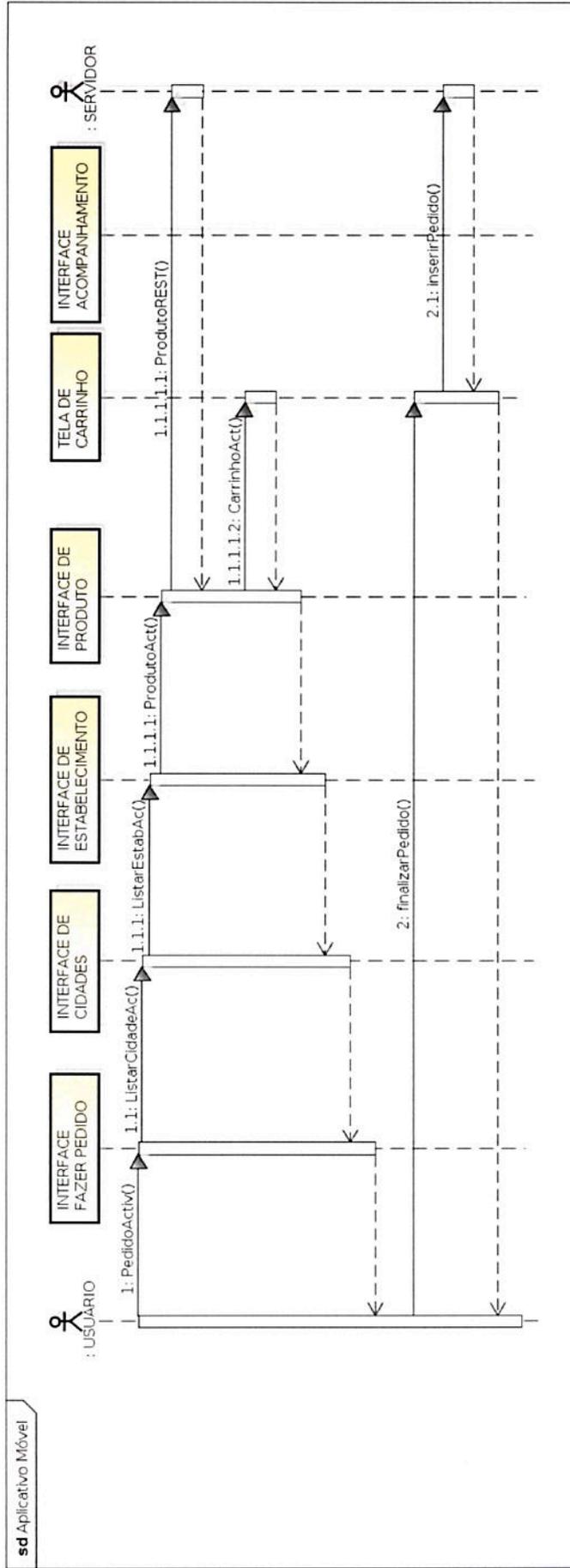
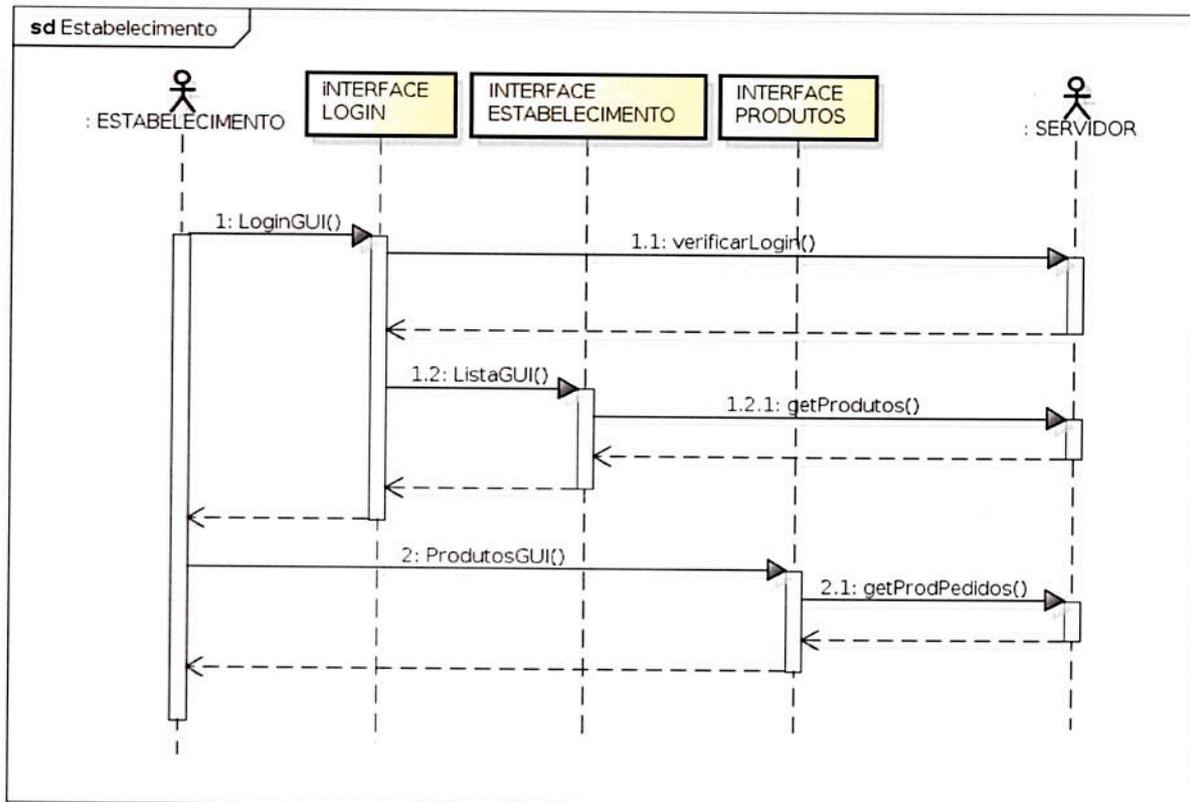


Ilustração 23: Diagrama de Sequência Aplicativo Móvel

## 7.4.3. DIAGRAMA DE SEQUÊNCIA ESTABELECIMENTO



powered by Astah

Ilustração 24: Diagrama de Sequência Estabelecimento

## 7.5. DICIONÁRIO DE DADOS

TABELA BAIRRO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idBairro	Código do bairro	INT	11
	descricao	Nome do Bairro	VARCHAR	45
FK	idCidade	Código da Cidade	INT	11

TABELA: CIDADE				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idCidade	Código da Cidade	INT	11
	descricao	Nome da cidade	VARCHAR	45
FK	idEstado	Código do Estado	INT	11

TABELA CONTA				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idConta	Código da Conta	INT	11
	email	Email do usuário ou estabelecimento	VARCHAR	100
	senha	Senha do usuário ou estabelecimento	VARCHAR	20
	rua	Endereço do cliente ou estabelecimento	VARCHAR	100
	nome	Nome do cliente ou estabelecimento	VARCHAR	100
FK	idCidade	Código da Cidade	INT	11
FK	idBairro	Código do bairro	INT	11
FK	IdEstado	Código do Estado	INT	11

TABELA CONTAESTABELECIMENTO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
	cnpj	CNPJ do estabelecimento	VARCHAR	20
PK	idConta	Código da Conta	INT	11

TABELA CONTAESTABELECIMENTOPRODUTO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idConta	Código da Conta	INT	11
PK	idProduto	Código do Produto	INT	11
	preco	Preço do produto	DOUBLE	---
	contador	Contador sobre produto no estoque	INT	11

TABELA CONTAUSUARIO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
	cpf	CPF do Cliente	INT	11
PK	idConta	Código da Conta	INT	11

TABELA ESTADO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idEstado	Código do Estado	INT	11
	descricao	Nome do Estado	VARCHAR	45

TABELA FAVORITO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idEstado	Código do Estado	INT	11
FK	idProduto	Código do Produto	INT	11
FK	idContaUsuario	Código da ContaUsuario	INT	11

TABELA PEDIDO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idPedido	Código do Pedido	INT	11
FK	idStatus	Código do Status	INT	11
FK	idContaUsuario	Código da ContaUsuario	INT	11
FK	idContaEstabelecimento	Código do ContaEstabelecimento	INT	11
	total	Total da compra realizada	DOUBLE	---
	data	Data e hora da compra realizada	TIMESTAMP	---

TABELA PEDIDOPRODUTO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idPedido	Código do Pedido	INT	11
PK	idProduto	Código do Produto	INT	11
	quantidade	Quantidade de produto comprado no pedido	INT	11
	descricao	Descricao do pedido, se caso o cliente deseja algo diferente no pedido	VARCHAR	45

TABELA PRODUTO				
	CAMPO	DESCRIÇÃO	TIPO	TAM
PK	idProduto	Código do Produto	INT	11
	nome	Nome do Produto	VARCHAR	100
	descricao	Descrição do produto em detalhes	VARCHAR	200

## 7.6. DIAGRAMA ENTIDADE RELACIONAMENTO (DER)

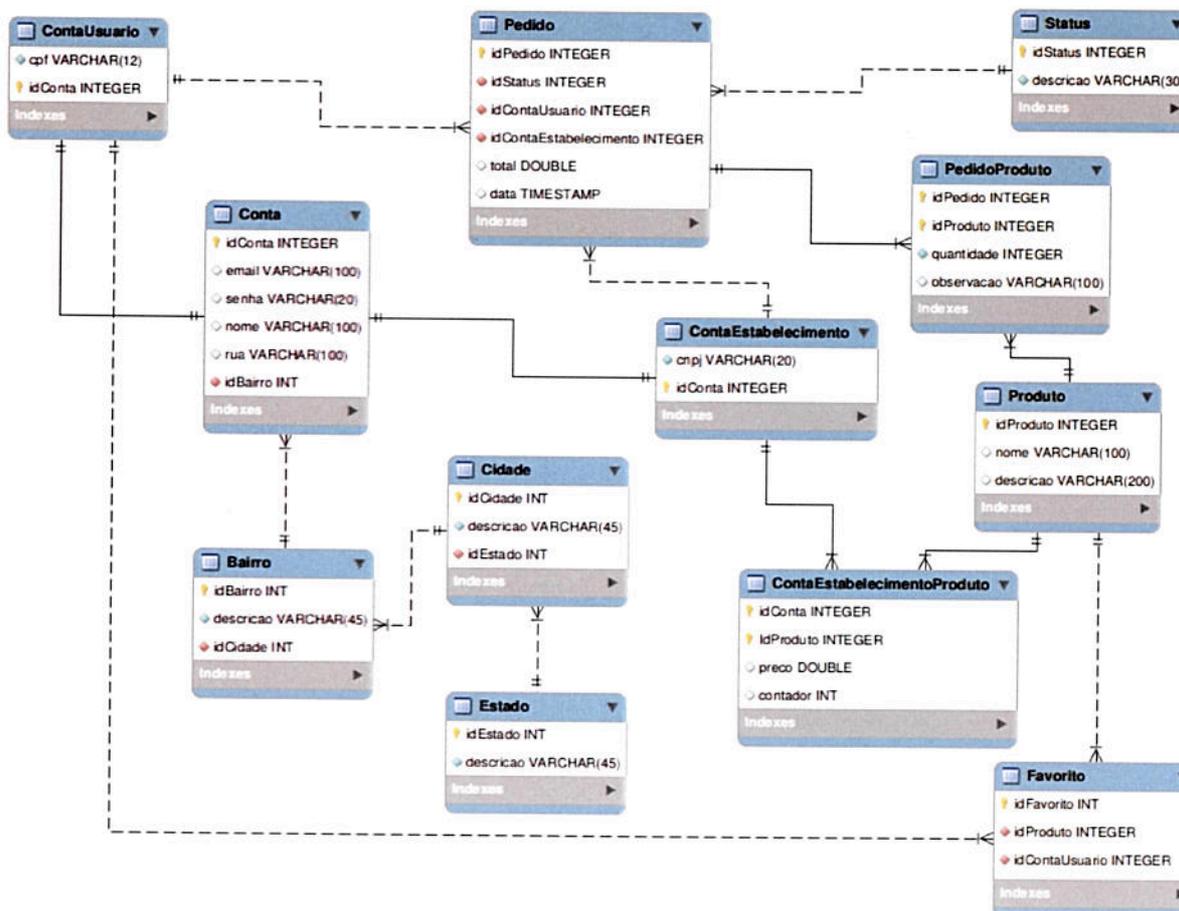


Ilustração 25: Diagrama de Entidades e Relacionamentos

## 7.7. WBS

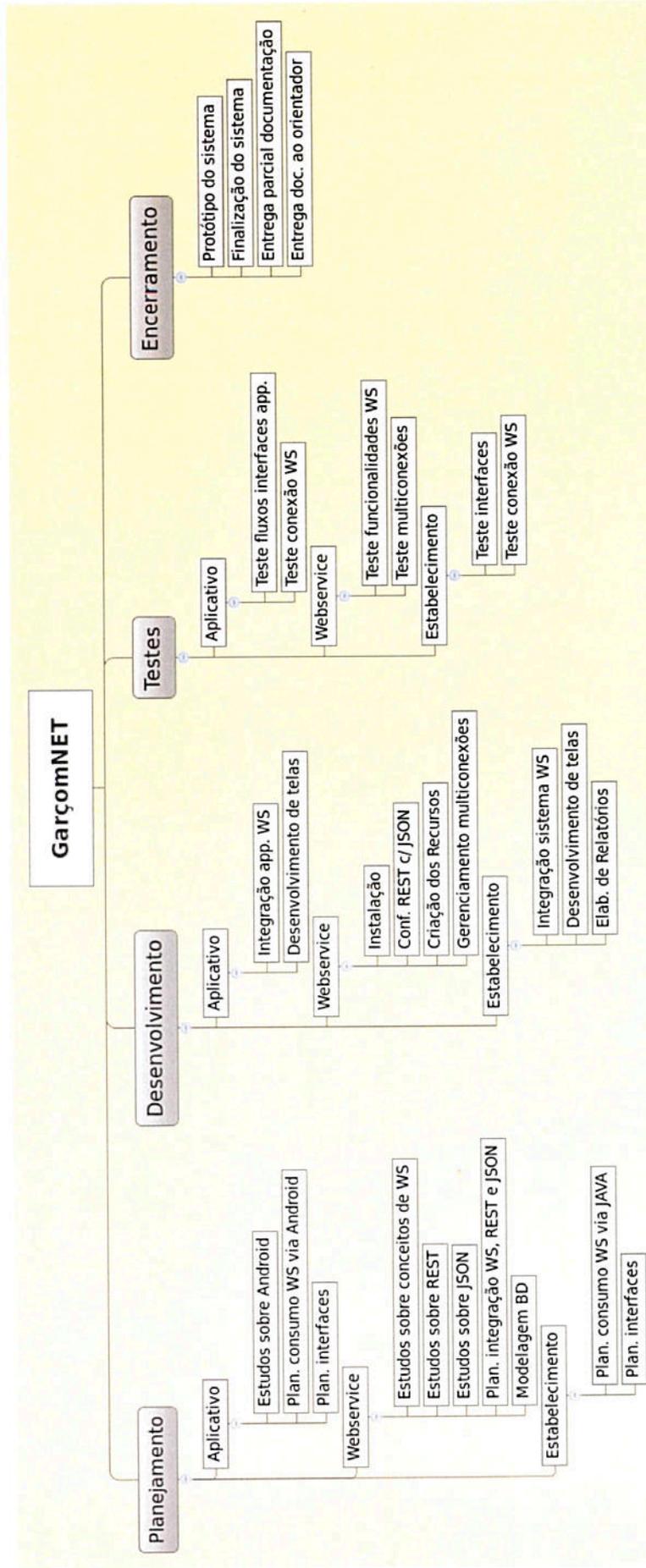


Ilustração 26: Work Breakdown Structure

## 7.8. DIAGRAMA DE GANTT

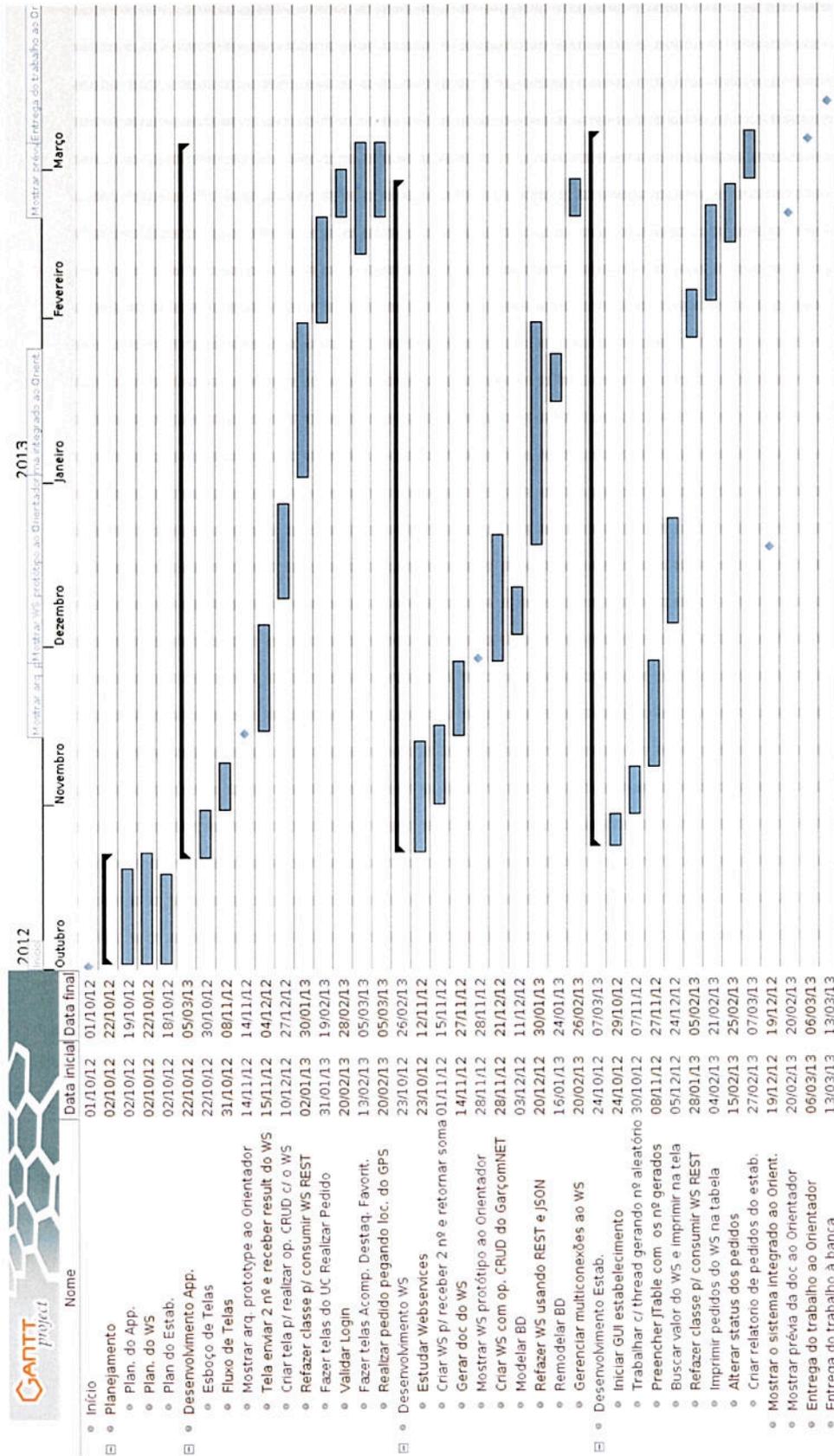


Ilustração 27: Diagrama de Gantt GarçomNET