



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

LUCAS CORTEZ

**SISTEMA DE CONTROLE DE PEDIDOS  
INTEGRAÇÃO UTILIZANDO WEB SERVICE**

Assis  
2014

LUCAS CORTEZ

**SISTEMA DE CONTROLE DE PEDIDOS**  
**INTEGRAÇÃO UTILIZANDO WEB SERVICE**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis – IMESA e Fundação Educacional do Município de Assis - FEMA, como requisito do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Esp. Célio Desiró

Área de Concentração: Análise e Desenvolvimento de Sistemas

Assis

2014

CORTEZ, Lucas

SISTEMA DE CONTROLE DE PEDIDOS - INTEGRAÇÃO UTILIZANDO  
WEB SERVICE / Lucas Cortez. Fundação Educacional do Município de Assis –  
FEMA – Assis, 2014.

74 Pag.

Orientador: Prof. Esp. Célio Desiró.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de  
Assis – IMESA.

1. Integração. 2. Web Service.

CDD: 001.61

Biblioteca FEMA

LUCAS CORTEZ

**SISTEMA DE CONTROLE DE PEDIDOS  
INTEGRAÇÃO UTILIZANDO WEB SERVICE**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis – IMESA e Fundação Educacional do Município de Assis - FEMA, como requisito do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, analisado pela seguinte comissão examinadora:

Orientador: Prof. Esp. Célio Desiró

Analisador: Prof. Me. Douglas Sanches da Cunha

Assis

2014

## **Dedicatória**

Dedico este trabalho primeiramente a DEUS, que com muito amor me guardou em toda essa trajetória da minha vida, nesta caminhada em preparação da minha vida, da minha carreira profissional a qual tanto desejo. Dedico também à minha namorada, que tanto me ajudou, me deu forças, me motivou a lutar por esse objetivo que vai muito nos auxiliar no nosso futuro. E também, à minha família, que me deram apoio e orientação durante essa caminhada rumo ao meu futuro.

## **Agradecimentos**

Agradeço primeiramente a DEUS por me ter me dado forças e sabedoria para concluir com êxito meus estudos, etapa importante da minha vida.

Ao professor e orientador Célio Desiró pela paciência, apoio e ajuda que muito me auxiliou no desenvolvimento deste trabalho.

A Fundação Educacional do Município de Assis pela sua qualidade de ensino e pela qualidade dos professores que ela possui que ajudaram tanto a mim quanto a outros alunos.

A todos os meus professores, que compartilharam seus conhecimentos profissionais para que eu pudesse ser um profissional capacitado.

A meu irmão Thiago Henrique Cortez, que me auxiliou muito durante o meu curso e também compartilhou comigo seus conhecimentos profissionais na área da programação.

A meus amigos de curso, que foram e serão meus irmãos, que muito me ajudaram em tudo.

E a todos aqueles que, direta ou indiretamente, torceram por mim na conclusão deste curso.

## RESUMO

O Sistema de Controle de Pedidos tem por objetivo controlar pedidos de compras de produtos realizados por diversos clientes. Esse sistema conta com três aplicativos, sendo um *web* em Java, um móvel para dispositivos Android e outro *web* em C Sharp.

O foco deste trabalho é implementar a integração de software nesse sistema, fazendo com que esses três aplicativos se comuniquem uns com os outros utilizando *Web Service*.

O *Web Service* faz com que os recursos da aplicação do software estejam disponíveis sobre a rede de uma forma padronizada. Existem muitas tecnologias que fazem a mesma coisa, como os browsers da Internet que utilizam a internet HTTP e HTML, porém, essas tecnologias não foram bem sucedidas na comunicação e integração de aplicações.

**Palavras-Chave: Integração, Web Service.**

## **ABSTRACT**

The Orders control system aims to control application for purchases of products made by several customers. This system has three applications, a Java web, a mobile for Android devices and other web in C sharp.

The focus of this work is to implement that system software integration, making these three applications to communicate with each other using Web Service.

The Web Service makes the features of the software application are available over the network in a standardized way, there are many technologies that do the same thing, as the Internet browsers using the internet HTTP and HTML. But, however, these technologies have not been successful in the communication and integration of applications.

**Keywords: Integration, Web Service.**



## LISTA DE ILUSTRAÇÕES

Figura 1: Caso de Uso do Sistema Completo.....	26
Figura 2: Caso de Uso do sistema web em Java.....	27
Figura 3: Caso de Uso de manter pedido.....	28
Figura 4: Caso de Uso de manter produto.....	29
Figura 5: Caso de Uso do sistema móvel.....	31
Figura 6: Caso de Uso de manter usuário.....	32
Figura 7: Caso de Uso de manter pedido.....	34
Figura 8: Caso de Uso do Sistema Web em C#.....	35
Figura 9: Caso de Uso de receber pedido.....	36
Figura 10: Caso de Uso de manter estado do Pedido.....	37
Figura 11: Diagrama de Classe do Sistema de Controle de Pedido.....	38
Figura 12: Diagrama de Classe do Sistema Web em Java.....	39
Figura 13: Diagrama de Classe do Aplicativo Móvel.....	40
Figura 14: Diagrama de Classe do Sistema Web em C#.....	41
Figura 15: Diagrama de Entidade e Relacionamento do sistema completo.....	42
Figura 16: Diagrama de Entidade e Relacionamento da aplicação Web em Java.....	43
Figura 17: Diagrama de Entidade e Relacionamento na aplicação móvel.....	44
Figura 18: Diagrama de Entidade e Relacionamento na aplicação Web em C#.....	45
Figura 19: Diagrama de atividade do Web Site em Java.....	46
Figura 20: Diagrama de Atividade da Aplicação Móvel.....	47
Figura 21: Diagrama de Atividade do Web Site em C#.....	48
Figura 22: Diagrama de Sequência do Web Service em Java - Manter Pedido.....	49
Figura 23: Diagrama de Sequência do Web Site em Java - Manter Produto.....	50
Figura 24: Diagrama de sequência da Aplicação Móvel - Cadastro de Usuário Móvel.....	51
Figura 25: Diagrama de Sequência da Aplicação Móvel - Manter Pedido.....	52
Figura 26: Diagrama de Sequência do Web Site em C# - Manter Pedido.....	53
Figura 27: Tela inicial da aplicação Java.....	55
Figura 28: Tela de Login.....	56
Figura 29: Tela de Pedidos.....	56
Figura 30: Tela de Produtos.....	57
Figura 31: Tela de Cadastro de Produtos.....	58
Figura 32: Tela Inicial.....	59
Figura 33: Tela de cadastro de Usuário.....	60
Figura 34: Tela de Opções do Usuário.....	61
Figura 35: Tipos de Produto.....	62
Figura 36: Lista de Produto.....	63
Figura 37: Descrição do Produto.....	64
Figura 38: Tela de Login.....	65
Figura 39: Tela de Pedidos.....	65
Figura 40: Método Post REST para cadastro de Usuário Móvel.....	66
Figura 41: Método REST para consulta de Usuário Móvel.....	67
Figura 42: Método REST para consulta de todos os produtos.....	68

Figura 43: Método de Acesso ao REST para consultar Usuário Móvel .....	69
Figura 44: Método de Acesso ao REST para consulta de produtos.....	70
Figura 45: Método Genérico de acesso ao REST para consultas .....	71

## LISTA DE ABREVIATURAS E SIGLAS

ADT – Android Development Tools

DER – Diagrama de Entidade e Relacionamento.

EJB - Enterprise JavaBeans.

IBGE - Instituto Brasileiro de Geografia e Estatística.

JEE - Java Enterprise Edition .

JSF – Java Server Faces.

JSON – JavaScript Object Notation

JVM – Java Virtual Machine (Maquina Virtual Java).

O.O – Orientação a Objetos.

SGDB – Sistema Gerenciador de Banco de Dados.

S.O – Sistema Operacional.

UML - Linguagem Unificada de Modelagem ( Unified Modeling Language).

1	INTRODUÇÃO .....	14
1.1	OBJETIVO DO TRABALHO .....	14
1.2	PÚBLICO ALVO .....	15
1.3	JUSTIFICATIVA .....	16
2	MÉTODO DE DESENVOLVIMENTO .....	17
2.1	WEB SERVICE .....	17
2.2	ORIENTAÇÃO A OBJETO .....	17
3	LINGUAGEM JAVA .....	19
3.1.1	<b>Java Server Faces – JSF</b> .....	<b>19</b>
3.1.2	<b>Enterprise Javabeans – EJB</b> .....	<b>20</b>
3.1.3	<b>Hibernate</b> .....	<b>20</b>
3.2	ANDROID .....	21
3.2.1	<b>Android Development Tools - ADT-Plugin</b> .....	<b>21</b>
3.3	LINGUAGEM C# - C SHARP .....	21
3.4	UML – LINGUAGEM UNIFICADA DE MODELAGEM (UNIFIED MODELING LANGUAGE) 22	
3.5	JBOSS – SERVIDOR DE APLICAÇÃO .....	22
3.6	JSON – JAVASCRIPT OBJECT NOTATION .....	23
4	RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO DO PROJETO.....	24
5	ANÁLISE DE REQUISITOS .....	25
6	DIAGRAMA DE CASO DE USO .....	26
6.1	JAVA.....	27
6.2	ANDROID .....	31
6.3	C#.....	35
7	DIAGRAMA DE CLASSE .....	38
7.1	JAVA.....	39
7.2	ANDROID .....	40
7.3	C#.....	41
8	DIAGRAMA DE ENTIDADE E RELACIONAMENTO – DER .....	42
8.1	JAVA.....	43
8.2	ANDROID .....	44
8.3	C#.....	45
9	DIAGRAMAS DE ATIVIDADE.....	46
9.1	JAVA.....	46

9.2	ANDROID .....	47
9.3	C#.....	48
10	DIAGRAMA DE SEQUÊNCIA .....	49
10.1	JAVA.....	49
10.1.1	<b>Manter Pedido.....</b>	<b>49</b>
10.1.2	<b>Manter Produto .....</b>	<b>50</b>
10.2	ANDROID .....	51
10.2.1	<b>Cadastro Usuário Móvel.....</b>	<b>51</b>
10.2.2	<b>Manter Pedido.....</b>	<b>52</b>
10.3	C#.....	53
10.3.1	<b>Manter Pedido.....</b>	<b>53</b>
11	CRONOGRAMA.....	54
12	TELAS DO SISTEMA.....	55
12.1	APLICAÇÃO JAVA .....	55
12.2	APLICAÇÃO ANDROID.....	59
12.3	APLICAÇÃO C# .....	65
13	LINHAS DE CÓDIGO .....	66
13.1	JAVA.....	66
13.2	ANDROID .....	69
13.3	C#.....	71
14	CONCLUSÃO .....	72
	REFERÊNCIAS BIBLIOGRÁFICAS .....	73

# 1 INTRODUÇÃO

Existem diversas linguagens de programação utilizadas no desenvolvimento de software no mundo todo, e com cada uma dessas linguagens é possível desenvolver diversos tipos de softwares capazes de atender as necessidades do mercado. Parte desses softwares desenvolvidos com algumas dessas linguagens são para dispositivos móveis, que por sua vez estão sendo utilizados cada vez mais. Segundo o IBGE, o número de pessoas que usam celulares e internet em seis anos (entre 2005 e 2011) cresceram mais de 100% no Brasil.

Nos dias atuais os dispositivos móveis, como celulares, *smartphones*, *tablets* estão cada vez mais em constante uso pela sociedade, tornando-se parte da vida das pessoas. Com o seu uso constante, esses dispositivos se tornaram parte do dia a dia de seus usuários graças à diversidade de funcionalidades e recursos que eles oferecem.

Muitos dos aplicativos móveis desenvolvidos possuem a necessidade de se integrar com softwares desenvolvidos em plataformas diferentes. A tecnologia mais utilizada nesse processo de integração é a tecnologia de Web Services. Essa tecnologia permite que sistemas desenvolvidos em linguagens ou plataformas distintas se comuniquem uns com os outros para que possam realizar tarefas conjuntas sem que haja nenhuma restrição.

## 1.1 OBJETIVO DO TRABALHO

Esse trabalho propõe desenvolver algumas aplicações simples utilizando linguagens de programação diferentes, inclusive uma aplicação para Android, e integrá-las através de Web Services. Será desenvolvido um controle de pedidos on-line de uma pizzaria, através do qual um usuário móvel poderá efetuar um pedido e a pizzaria, recebendo esse pedido, poderá preparar o que o seu cliente solicitou e fazer a sua entrega.

O sistema de pedido proposto para exemplificar o conceito de integração terá três módulos, que serão aplicações isoladas desenvolvidas em linguagens diferentes, mas que se integram através de Web Services enviará arquivos em formatos JSON para as aplicações fazendo então o compartilhamento de informações e recursos.

O módulo Móvel será uma aplicação desenvolvida para o sistema operacional Android, utilizado para que os clientes se cadastrem e realizem pedidos on-line.

O módulo de Controle de Pedidos e Produtos será uma aplicação Web desenvolvida em Java, na qual o usuário mantém o cadastro de seus produtos e tem o controle dos pedidos recebidos.

Por último será desenvolvida uma aplicação Web em C#, a qual funcionará localmente na cozinha da pizzaria com objetivo de receber os itens de cada pedido. O banco de dados utilizado para desenvolver o sistema será o MySQL.

## 1.2 PÚBLICO ALVO

Todos os quais necessitem de conhecimentos sobre o assunto abordado no desenvolvimento deste trabalho.

Os mecanismos de buscas na internet sempre terão seus papéis importantes para os desenvolvedores de softwares, tanto para aprender uma nova tecnologia como para buscar exemplos de como implementar determinados recursos. Este trabalho servirá para complementar esses mecanismos de busca, fornecendo ao desenvolvedor um conteúdo simples que facilite o entendimento sobre integração de sistemas com Web Services e o envio e consumo de arquivos em formato JSON.

### 1.3 JUSTIFICATIVA

O sistema colocará em prática o desenvolvimento de aplicativos web e aplicativo móvel para Android, bem como irá mostrar a integração desses três aplicativos desenvolvidos em plataformas distintas para que ambos realizem tarefas conjuntamente. A ideia de desenvolver este sistema surgiu através de um interesse de aprender a teoria e a prática de integração de sistemas e desenvolvimento de aplicativos móveis para o sistema operacional Android e sua integração com outras plataformas.



## 2 MÉTODO DE DESENVOLVIMENTO

Para o desenvolvimento do controle de pedidos serão utilizados os conceitos de orientação a objetos, que tem como principais objetivos diminuir a complexidade no desenvolvimento do software e aumentar a produtividade, e também a tecnologia de Web Services, que permite criar diferentes aplicativos em diferentes plataformas e fazer com que esses aplicativos se interajam um com o outro mesmo sendo de plataformas distintas.

A seguir são apresentadas as tecnologias utilizadas e algumas de suas características principais.

### 2.1 WEB SERVICE

Web Service é uma tecnologia que permite a integração de sistemas e a comunicação entre aplicações diferentes. Com ela é possível fazer com que uma nova aplicação se integre e comunique com uma aplicação já existente desenvolvida em plataforma diferente.

O Web Service permite que as aplicações enviem e recebam os dados necessários para a comunicação. Os dados podem ser em formato XML ou JSON. A mais conhecida e utilizada atualmente é o formato JSON.

### 2.2 ORIENTAÇÃO A OBJETO

A orientação a objetos tem por objetivo representar situações do mundo real através de uma coleção de objetos. Ela tenta aproximar o mundo real com o mundo virtual para simular o mundo real dentro de um computador.

No desenvolvimento do software essa coleção de objetos compõe um sistema computacional fazendo com que cada objeto interaja entre si realizando tarefas de acordo com suas funcionalidades.

O conceito de orientação a objetos não é muito recente. A primeira linguagem a utilizar o conceito de O.O (Orientação a Objetos) foi o Simula 67 criado por Ole Johan Dahl e Kristen Nygaard em 1967. Mesmo este paradigma da programação sendo antigo, sua utilização ganhou força a pouco tempo e se espalhou por diversas linguagens modernas.

Algumas das linguagens modernas a usar O.O são: Java, C++, Ruby, C# e Python.

No desenvolvimento das aplicações propostas neste trabalho serão usadas as linguagens Java e C#, ambas abordando o conceito de O.O.

### 3 LINGUAGEM JAVA

Java é uma potente linguagem de programação que começou a ser criada na 1ª metade da década de 90 pela empresa Sun Microsystems. É uma linguagem que pode ser executada em qualquer sistema operacional desde que ele tenha uma JVM (Java Virtual Machine – “Maquina Virtual Java”) instalada, por esse motivo ela é considerada uma linguagem multiplataforma por não possuir uma compatibilidade limitada, pois ela é executada na JVM.

A Sun Microsystems, no início tinha como objetivo utilizar a linguagem recém criada em softwares de aparelhos eletrônicos, mas não obteve sucesso. Com o surgimento da internet, a empresa percebeu uma grande chance de utilizar a linguagem, e a partir daí ela alcançou o sucesso que era esperado.

Hoje, a linguagem Java está entre as mais usadas por profissionais no ramo da programação por ser uma linguagem potente e muito eficiente em ambientes complexos como redes de internet.

Atualmente, existem diversos frameworks desenvolvidos em Java que ajudam no desenvolvimento de aplicações para web. Um deles é o JSF (Java Server Faces) que será usado no trabalho e descrito na próxima sessão.

#### 3.1.1 Java Server Faces – JSF

Um framework é um projeto que serve como base para o desenvolvimento de outras aplicações.

O Java Server Faces (JSF) é um framework utilizado para desenvolvimento de interfaces para usuário em aplicativos web. Essa tecnologia é muito utilizada por desenvolvedores, pois facilita na criação de componentes para desenvolver a interface de usuário de aplicações web e esses componentes podem colocados em formulários e ligados em objetos facilmente, esse recurso permite com que o desenvolvedor se preocupe mais no desenvolvimento da regra de negocio, lógica, navegação e conexão.

Um dos componentes JSF bastante usado é o PrimeFaces que possui diversos componentes pré prontos que facilitam o desenvolvimento de componentes gráficos de interface web.

### **3.1.2 Enterprise Javabeans – EJB**

O EJB é uma arquitetura de componentes da plataforma JEE que é utilizada no ramo de desenvolvimentos de softwares que permite realizar o encapsulamento da lógica das regras de negócio, permitindo que o desenvolvimento das aplicações seja mais rápido e simplificado.

O EJB possui quatro tipos de componentes: Stateless Session Bean, Statefull Session Bean, Entity Bean e o Message Driven Bean.

- Stateless Session Bean: componente que é responsável por guardar as lógicas de negócio da aplicação.
- Statefull Session Bean: componente que atua como extensão no servidor utilizado por um cliente.
- Entity Bean: objeto remoto que gerencia dados persistentes, executa a lógica de negócio, utiliza objetos java dependentes e podem ser identificados por chave primária.
- Message Driven Bean: aplicação Java EE que processa mensagens de forma assíncrona, ou seja, através de transmissões de dados.

### **3.1.3 Hibernate**

Hibernate é um framework que possui ferramentas que ajudam no mapeamento de objetos e persistência de dados. Permite também trabalhar com SGBD's (Sistema Gerenciador de Banco de Dados) de maneira orientada a objetos fazendo com que um objeto tenha uma relação forte com uma tabela

dentro do banco de dados, ou seja, ele praticamente transforma as classes criadas dentro do sistema em tabelas dentro do banco de dados.

## 3.2 ANDROID

O Android é um sistema operacional (S.O) criado pela Google com base no kernel do Linux. Esse S.O é utilizado em aparelhos móveis, como celulares e tablets, possui uma interface amigável para o usuário facilitando o seu uso.

O uso deste S.O tem crescido muito, e através deste crescimento também aumentou o desenvolvimento de diversos tipos de aplicações para os usuários.

Uma das ferramentas utilizadas pelos desenvolvedores de aplicativos Android é o Android Developer Tools (ADT).

### 3.2.1 Android Development Tools - ADT-Plugin

No desenvolvimento do aplicativo Android, a linguagem a ser usada será Java, através do IDE Eclipse. O ADT é um *plugin* que contém diversas ferramentas que ao serem instaladas no Eclipse, permitem o desenvolvimento de aplicações móveis para Android.

Ele contém também um emulador para simular o S.O Android e testar a aplicação desenvolvida antes de instalar no dispositivo móvel.

## 3.3 LINGUAGEM C# - C SHARP

É uma linguagem de programação de alto nível, orientada a objetos, baseada no C++ e desenvolvida pela Microsoft para a plataforma .NET.

O C# é uma linguagem simples e robusta que permite que o programador desenvolva excelentes aplicações. Sua sintaxe simplifica muitas complicações

da linguagem C++ e assim oferece melhores condições de desenvolvimento para o programador.

### 3.4 UML – LINGUAGEM UNIFICADA DE MODELAGEM (UNIFIED MODELING LANGUAGE)

A linguagem UML é usada para fazer modelagem de sistema, ou seja, fazer representações do software através de modelos orientados a objetos. Ela permite, através de desenhos, descrever e identificar a comunicação entre os processos do sistema permitindo melhor compreensão sobre suas funcionalidades, isso a torna uma grande ferramenta para os engenheiros de softwares.

A UML possui dois tipos de diagramas, os Estruturais e os Comportamentais:

- **Diagramas Estruturais:** De Classe, Objeto, Componentes, implantação, Pacotes, Estrutura.
- **Diagramas Comportamentais:** De Caso de Uso (Use Case), De Máquina de Estados, Atividades, Interação.

### 3.5 JBOSS – SERVIDOR DE APLICAÇÃO

O JBOSS é um servidor de aplicação desenvolvido em java baseado na plataforma J2EE e também possui seu código aberto. Ele pode ser executado em qualquer sistema operacional.

### 3.6 JSON – JAVASCRIPT OBJECT NOTATION

É uma estrutura de dados em JavaScript. Seu uso ganhou força recentemente, mas o JSON não é uma tecnologia nova. Ele pode substituir o xml e é muito mais eficaz, pois possui uma estrutura mais simples e sua leitura é muito mais rápido do que a do xml.

## **4 RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO DO PROJETO**

### **Humano:**

- 1 Analista / Programador

### **Equipamento:**

- 1 Notebook

### **Software:**

- Eclipse Kepler
- Microsoft Visual Studio 2012
- Servidor de aplicação JBOSS 7.1
- Servidor de aplicação Microsoft IIS
- Banco de Dados MySQL
- ADT – Plugin



## 5 ANÁLISE DE REQUISITOS

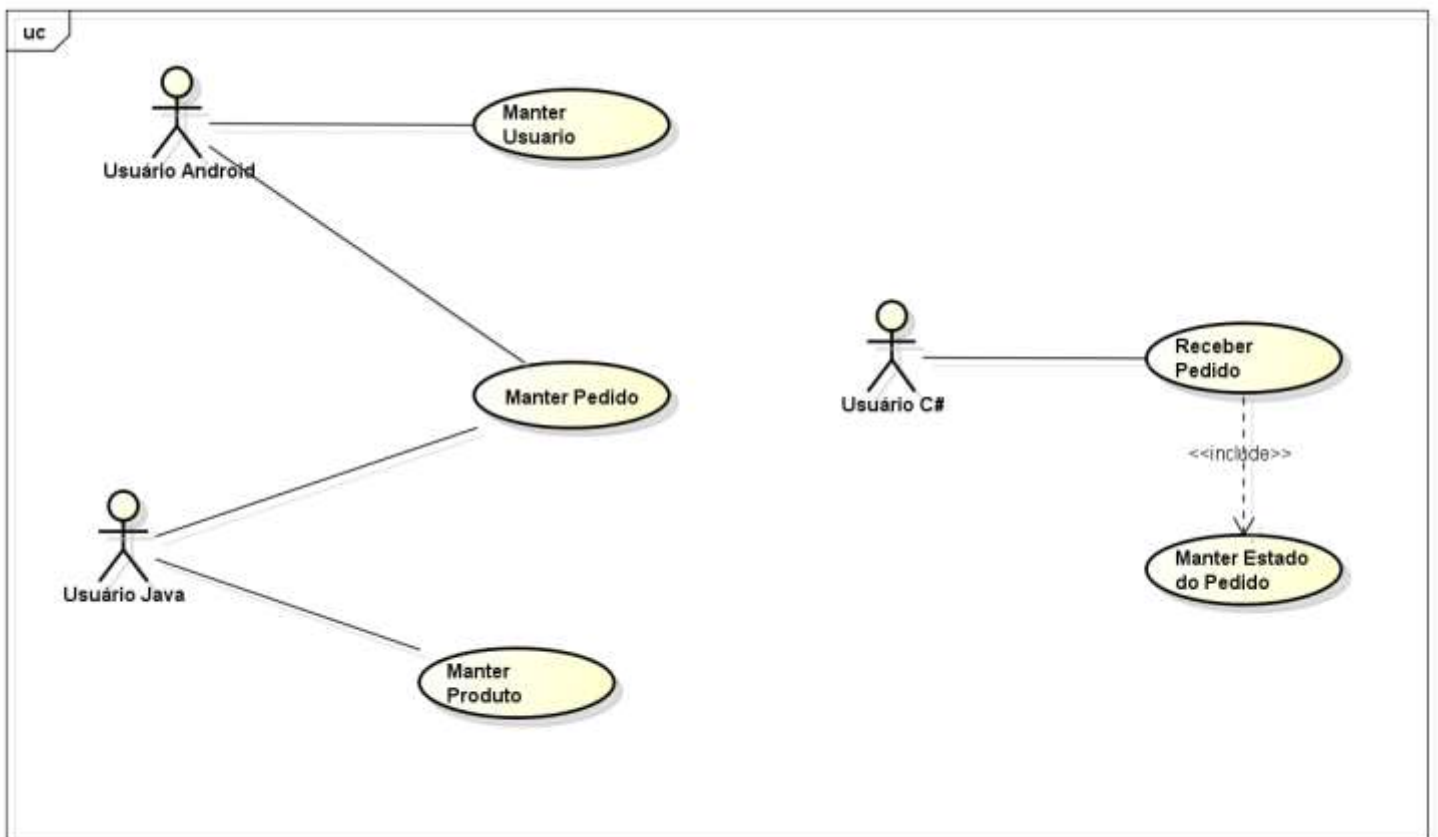
Para realizar o desenvolvimento do trabalho foi necessário buscar conhecimento sobre as ferramentas e tecnologias que serão utilizadas na programação do sistema. A principal, que também é o foco do trabalho é a integração de software utilizando Web Services. Também não foi necessário ideias de um cliente específico, pois o sistema poderá ser usado em vários ramos de serviços que necessite das funcionalidades que o software oferece.

Para a execução do sistema completo será necessário um microcomputador com configurações básicas com acesso à internet, um aparelho celular com o sistema operacional Android, também com acesso à internet.

## 6 DIAGRAMA DE CASO DE USO

O Diagrama de Caso de Uso é uma ferramenta que possibilita representar, através de desenhos gráficos, as funcionalidades proposta pelo sistema que está sendo desenvolvido, promovendo uma melhor compreensão do software.

O Caso de uso abaixo representa o sistema completo composto pelas três aplicações.

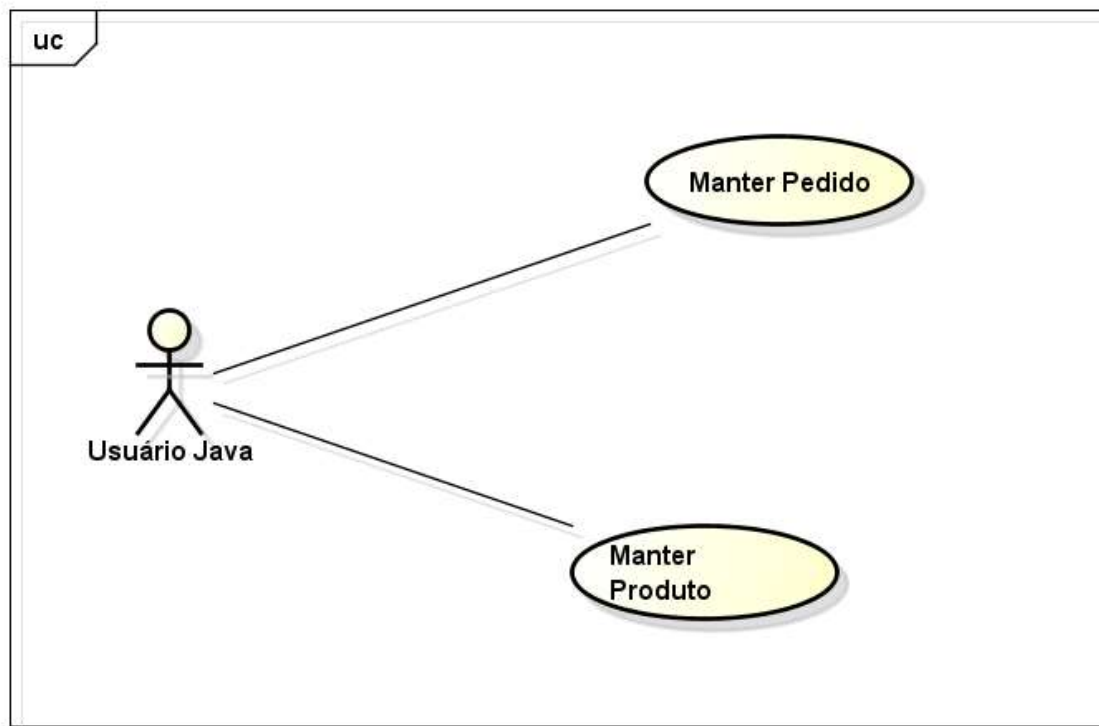


powered by Astah

Figura 1: Caso de Uso do Sistema Completo

## 6.1 JAVA

O diagrama a seguir mostra uma breve descrição das funcionalidades do sistema web que será desenvolvido.

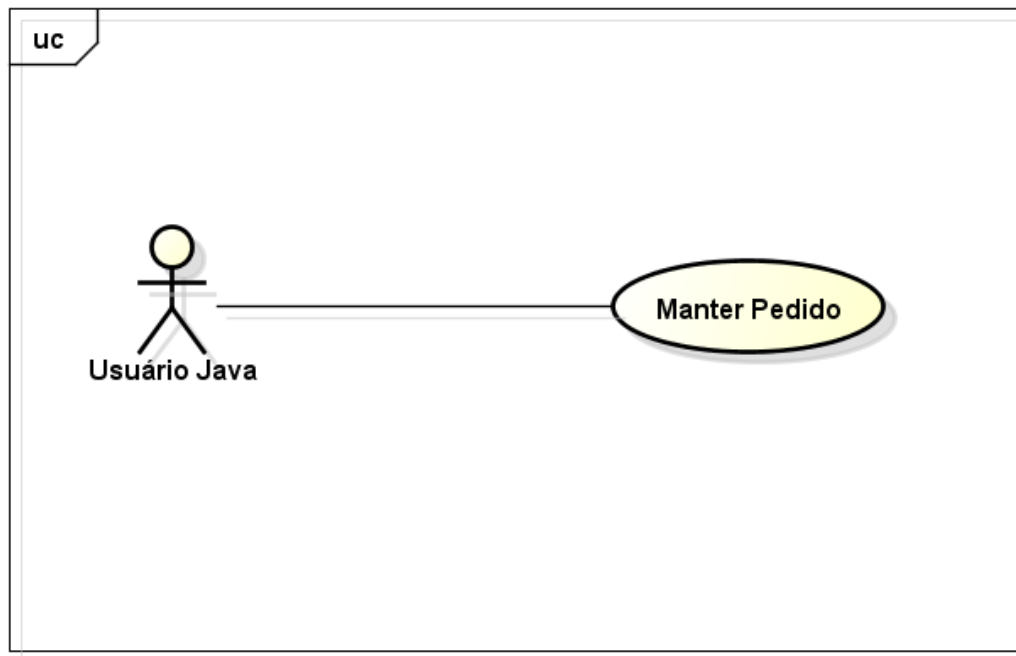


powered by Astah

Figura 2: Caso de Uso do sistema web em Java.

## Narrativa do U.C:

### Manter Pedido



powered by Astah

Figura 3: Caso de Uso de manter pedido.

### Finalidade:

- Permite o administrador gerenciar os pedidos que foi recebido.

### Fluxo principal:

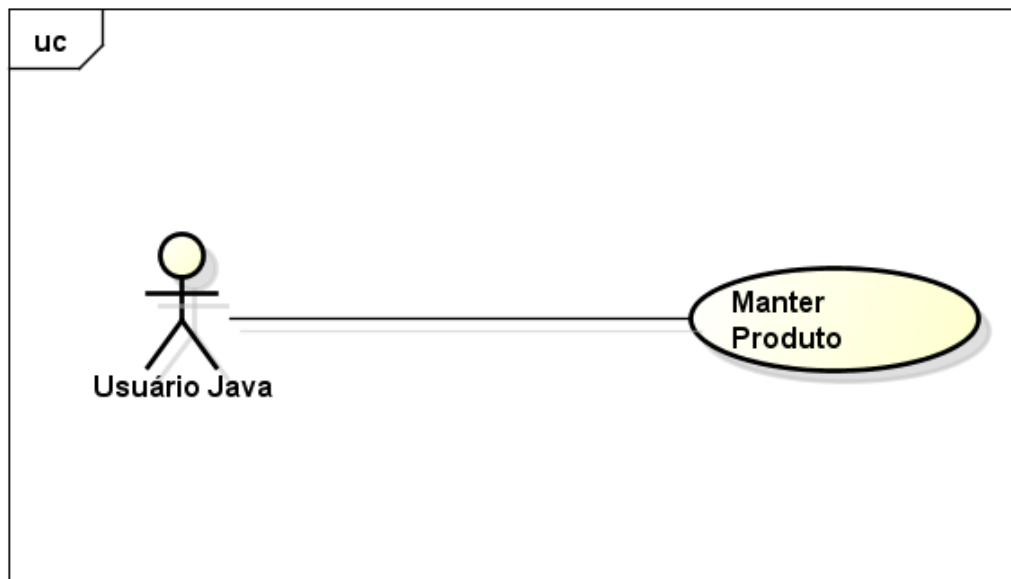
- a. O sistema fornece uma tabela com todos os pedidos recebidos recentemente.
- b. O administrador, em um campo ao lado da tabela, informa o cliente que seu pedido está pronto passando o Id do pedido **[E1]**.

### Fluxo de Exceção:

#### E1. Envio do Pedido:

- a. O sistema ao informar o cliente sobre o envio do pedido faz a exclusão do pedido no banco de dados.

Manter Produto:



powered by Astah

Figura 4: Caso de Uso de manter produto

#### Finalidade:

- Permite que o administrador do sistema cadastre os produtos de seus clientes

#### Fluxo principal:

- a. O administrador pode selecionar a opção cadastrar, alterar [A1] ou excluir [A2] produto.
- b. O sistema solicita os dados necessários para realizar determinada operação.
- c. O administrador fornece os dados [A3].
- d. O sistema verifica se os dados necessários foram fornecidos e em seguida confirma a operação [E1].

**Fluxo Alternativo:****A1. Alterar cadastro:**

- O sistema fornece uma lista dos produtos cadastrados para realizar a alteração.

**A2. Excluir cadastro:**

- O sistema fornece uma lista dos produtos cadastrados para realizar a exclusão.

**A3. Cancelar Cadastro:**

- O sistema retorna ao passo 'a' do fluxo principal.

**Fluxo de Exceção:****E1. Dados incompletos:**

- a. O sistema verifica se os dados estão completos.
- b. O sistema retorna ao passo 'c' do Fluxo Principal e informa os dados que faltam para completar o cadastro.

## 6.2 ANDROID

Este diagrama mostra de maneira simples quais as funcionalidades da aplicação para o Android.

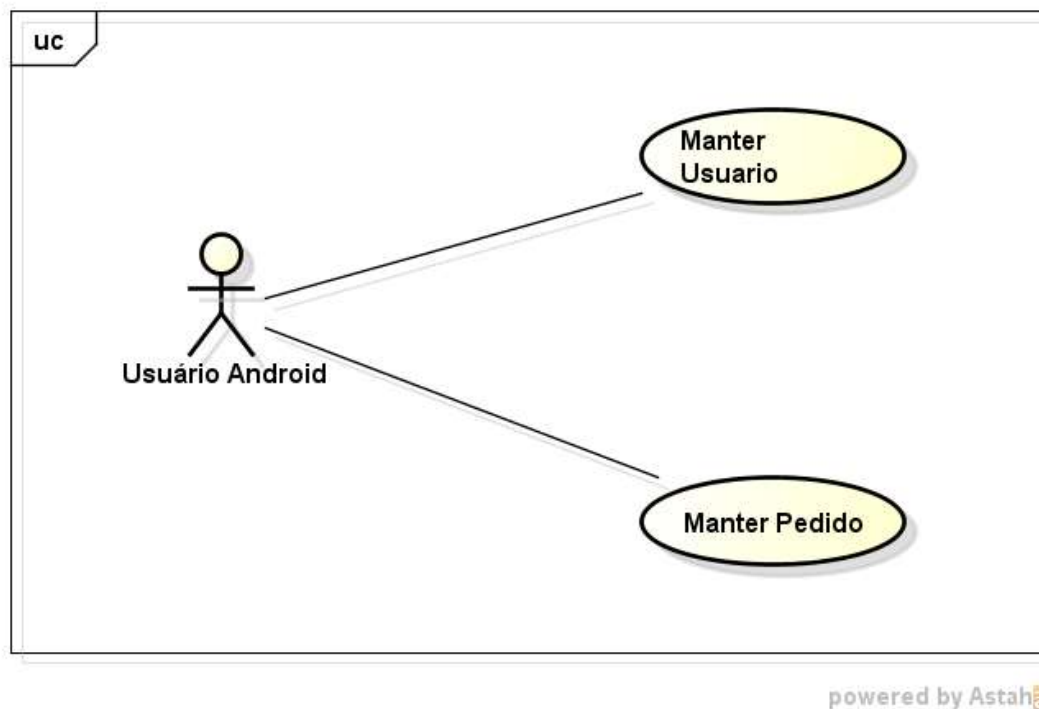
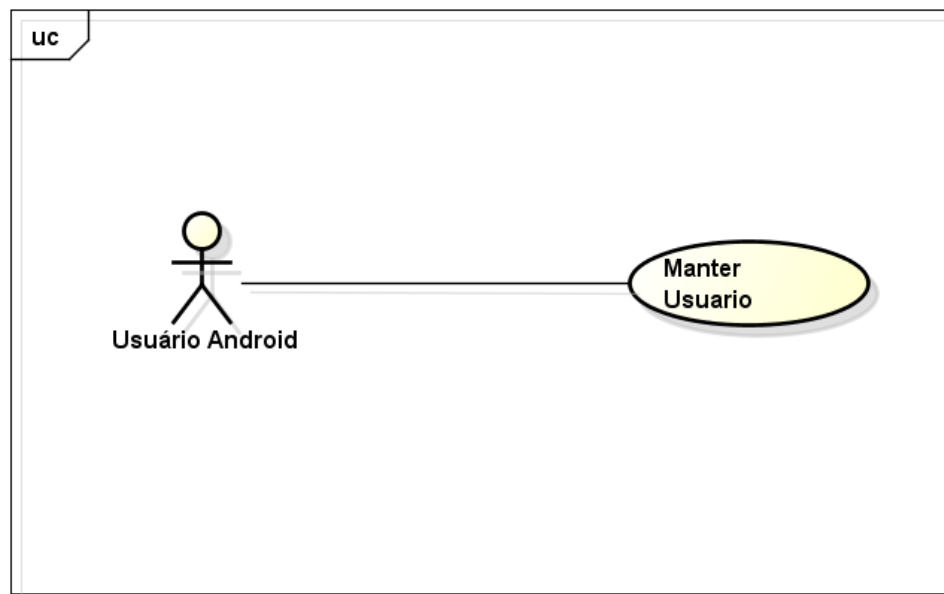


Figura 5: Caso de Uso do sistema móvel.

**Narrativa do U.C:**

Manter Usuário:



powered by Astah

Figura 6: Caso de Uso de manter usuário.

**Finalidade:**

- Permite que o usuário se cadastre no sistema com suas informações.

**Fluxo principal:**

- a. O usuário pode escolher a opção de cadastrar, alterar **[A1]** ou excluir **[A2]**.
- b. O sistema solicita os dados necessários para realizar determinada operação.
- c. O usuário fornece seus dados.
- d. O sistema verifica se os dados necessários foram fornecidos e em seguida confirma a operação **[E1]**.



**Fluxo Alternativo:****A1. Alterar cadastro:**

- O sistema fornece ao usuário seus dados cadastrados para que ele realize a alteração.

**A2. Excluir cadastro:**

- O sistema fornece uma opção para que o usuário possa excluir seu cadastro.

**A3. Cancelar Cadastro:**

- O sistema retorna ao passo 'a' do fluxo principal.

**Fluxo de Exceção:****E1. Dados incompletos:**

- a. O sistema verifica se os dados estão completos.
- b. O sistema retorna ao passo 'c' do Fluxo Principal e informa os dados que faltam para completar o cadastro.

Manter Pedido:

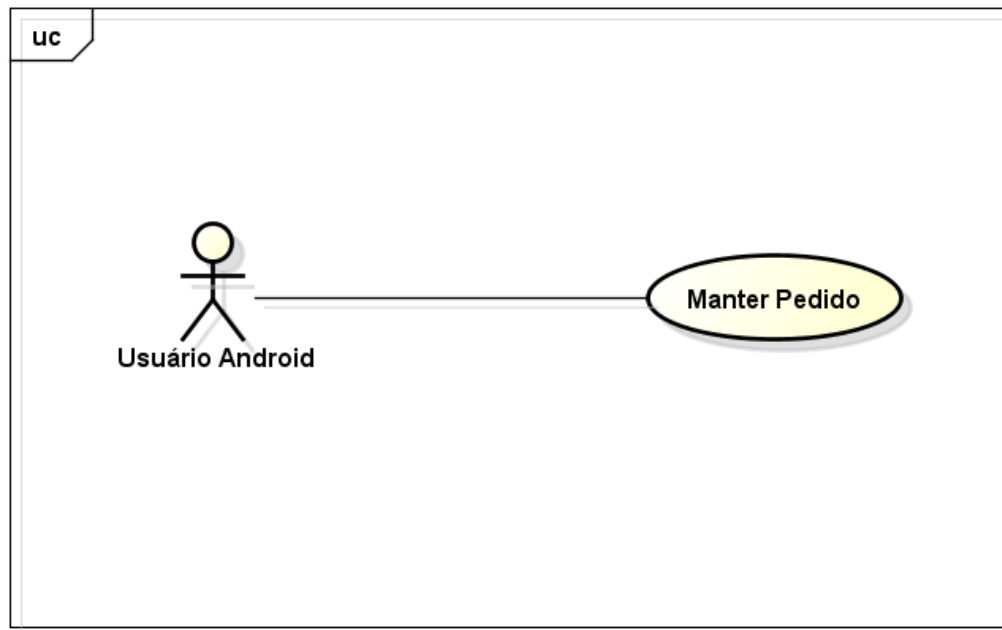


Figura 7: Caso de Uso de manter pedido.

**Finalidade:**

- Permite o usuário realize os pedidos de determinados produtos.

**Fluxo Principal:**

- a. O sistema solicita os dados necessários para realizar determinada operação.
- b. O usuário fornece os dados para realizar o seu pedido.
- c. O sistema verifica se os dados necessários do pedido foram fornecidos e em seguida confirma a operação **[E1]**.

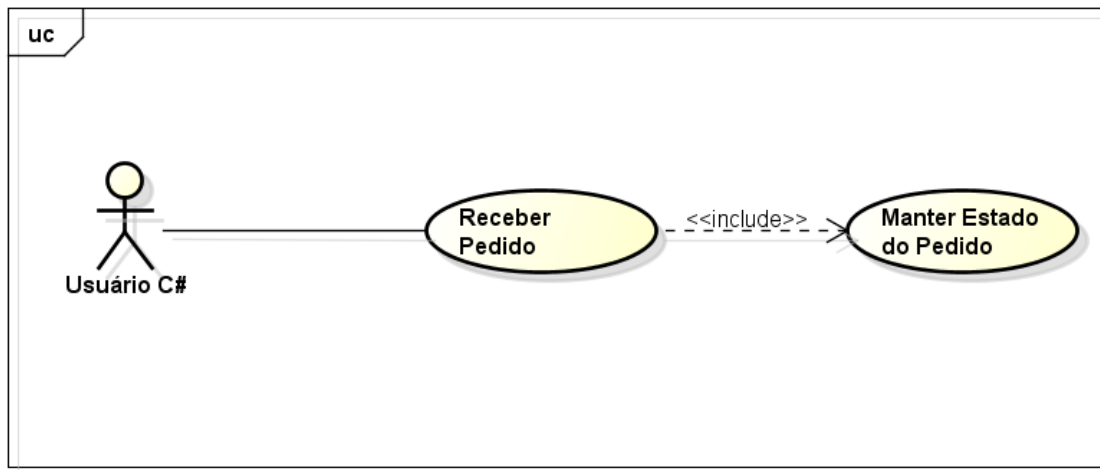
**Fluxo de Exceção:**

**E1. Dados incompletos:**

O sistema verifica se foi feito algum pedido, se foi ele pede uma confirmação do pedido. Senão, ele volta para a tela principal.

### 6.3 C#

O diagrama à seguir é uma simples representação que mostra a funcionalidade do sistema desktop.

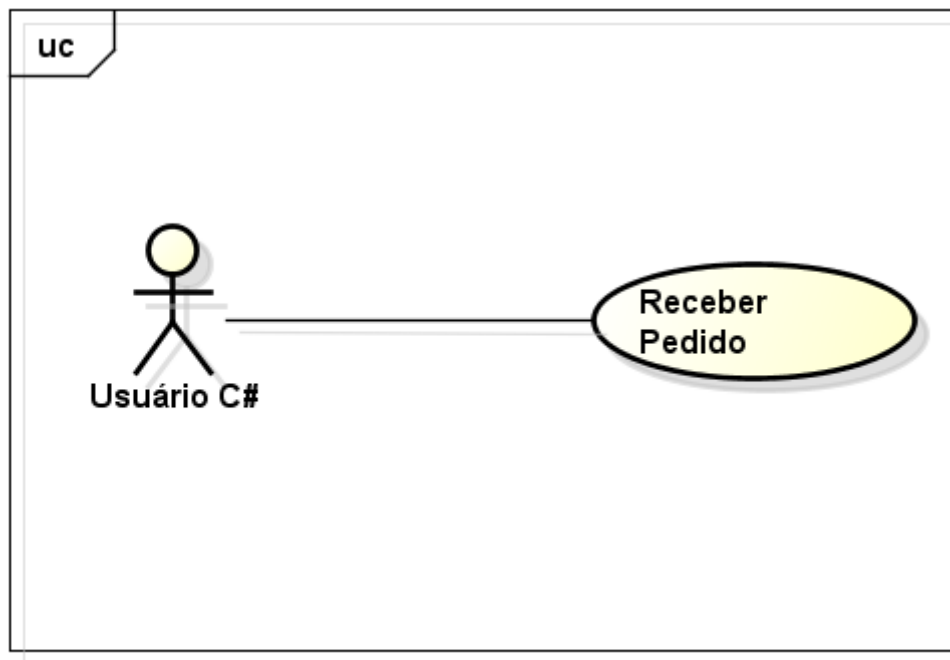


powered by Astah

Figura 8: Caso de Uso do Sistema Web em C#

**Narrativa de U.C:**

Receber e Registrar Pedido:



powered by Astah

Figura 9: Caso de Uso de receber pedido.

**Finalidade:**

- Permite o sistema receber o pedido feito através de um sistema móvel.

**Fluxo Principal:**

- a. O sistema recebe um pedido feito por um cliente através do aplicativo móvel.

Informar Envio do Pedido:

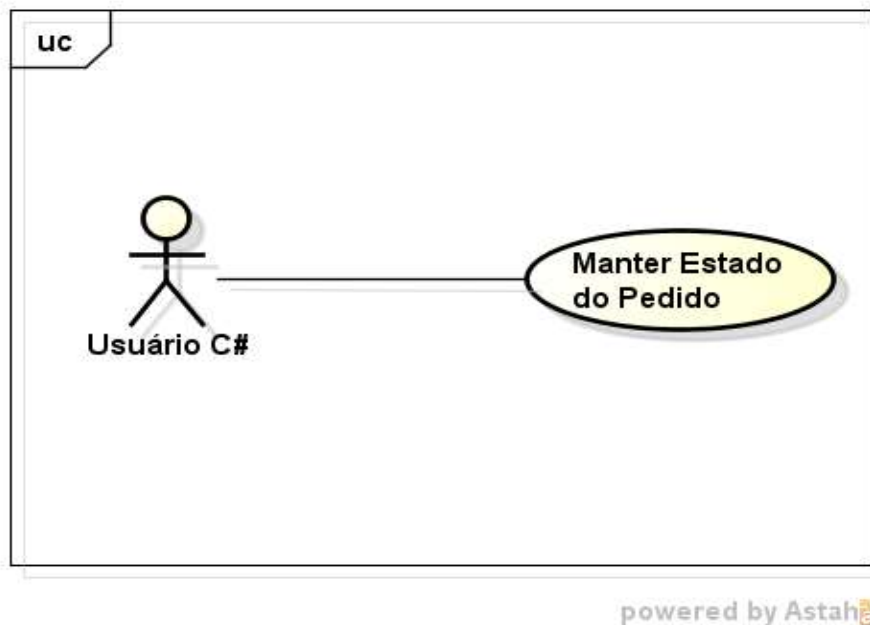


Figura 10: Caso de Uso de manter estado do Pedido

Finalidade:

- Permite que o usuário do sistema informe o estado do pedido, se ainda esta em processo ou se está pronto.

**Fluxo Principal:**

- a. O usuário altera o estado do pedido.

## 7 DIAGRAMA DE CLASSE

Este diagrama mostra o relacionamento de todas as classes no sistema completo.

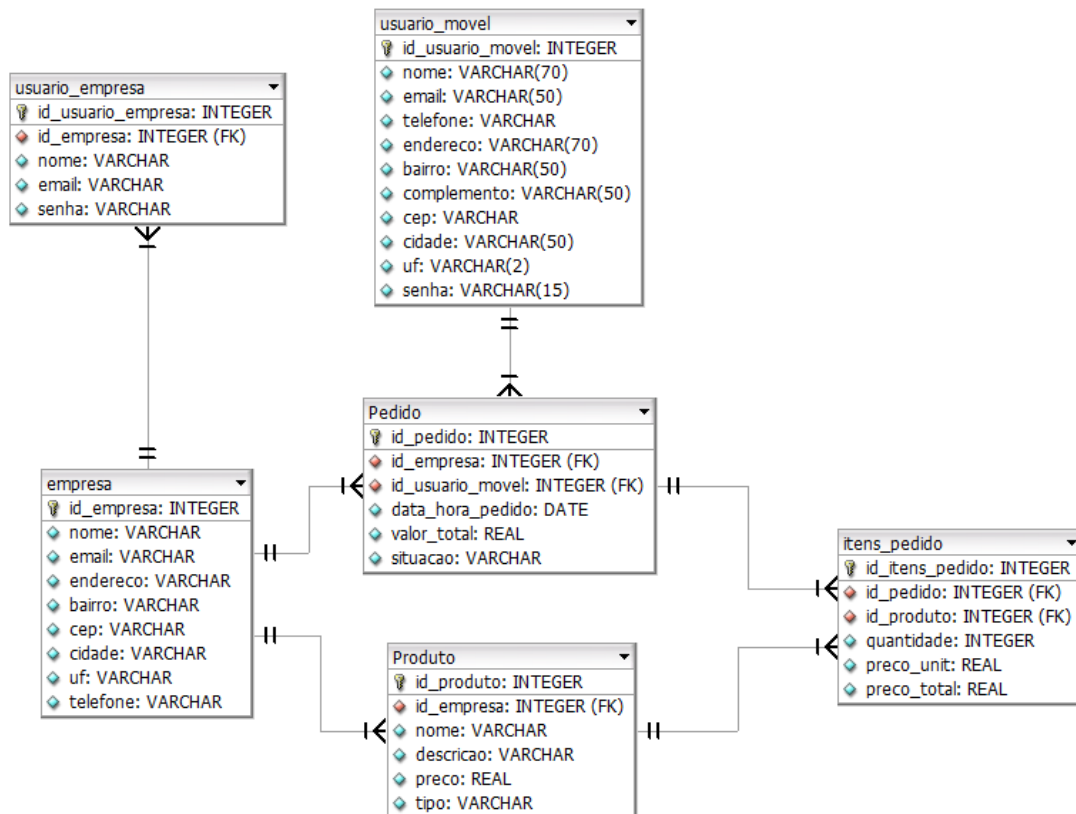


Figura 11: Diagrama de Classe do Sistema de Controle de Pedido

## 7.1 JAVA

O sistema web em Java possui a classe `usuario_empresa` para conter os dados do administrador, a classe `empresa` para ter acesso a seus próprios dados e a classe `produto` para os produtos que a empresa comercializa.

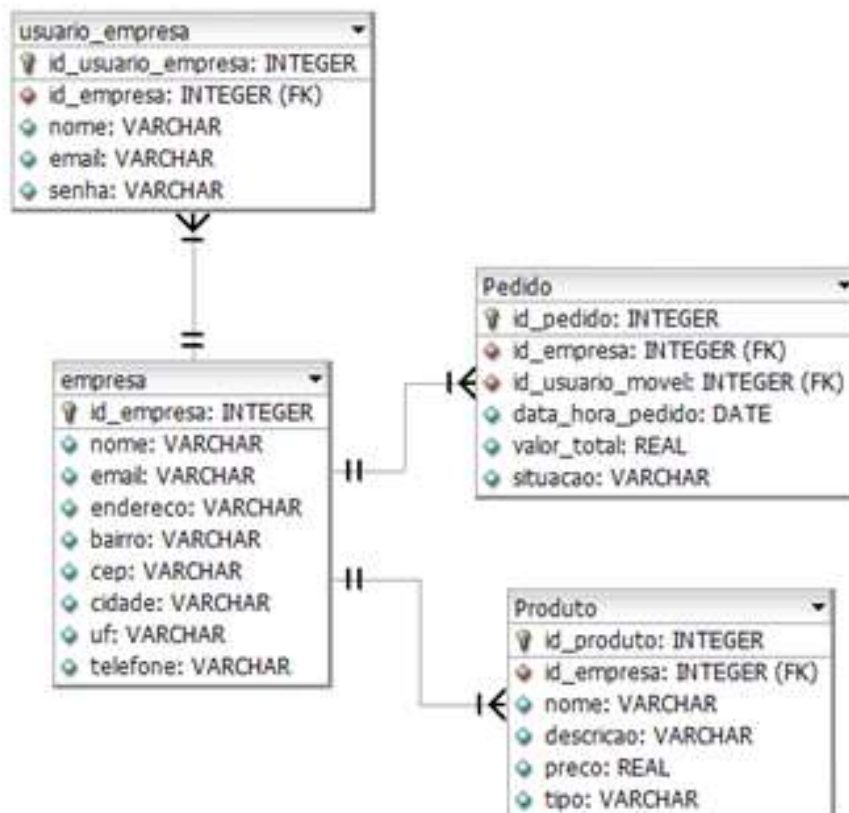


Figura 12: Diagrama de Classe do Sistema Web em Java

## 7.2 ANDROID

O sistema Móvel possui a classe UsuarioMovel para o representar o usuário, a classe Produto para os produtos para que possa fazer o pedido utilizando a classe Pedido e a ItensPedido para adicionar vários produtos no pedido.

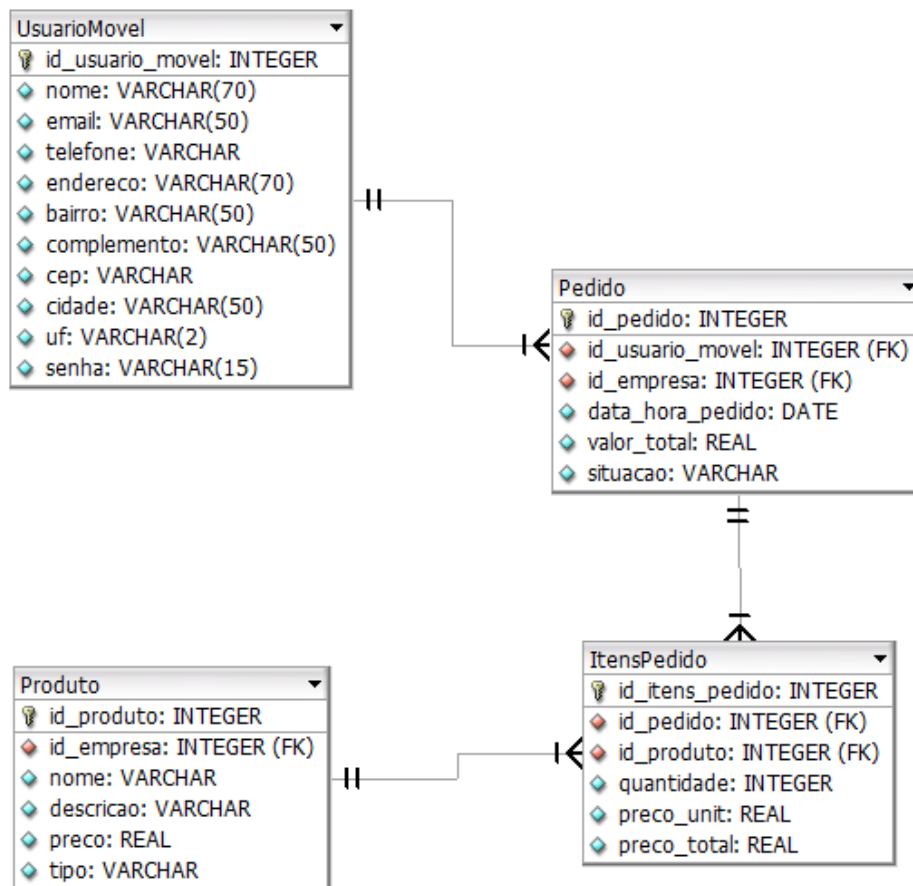


Figura 13: Diagrama de Classe do Aplicativo Móvel



### 7.3 C#

O sistema web em C# irá utilizar as mesmas classes do Sistema Móvel, menos a classe UsuarioMovel, pois ela controlara apenas a parte de de pedido que utilizara as tabelas Pedido, ItensPedido para identificar os itens pedido pelo cliente e a tabela Produto para poder ter os dados de cada produto.

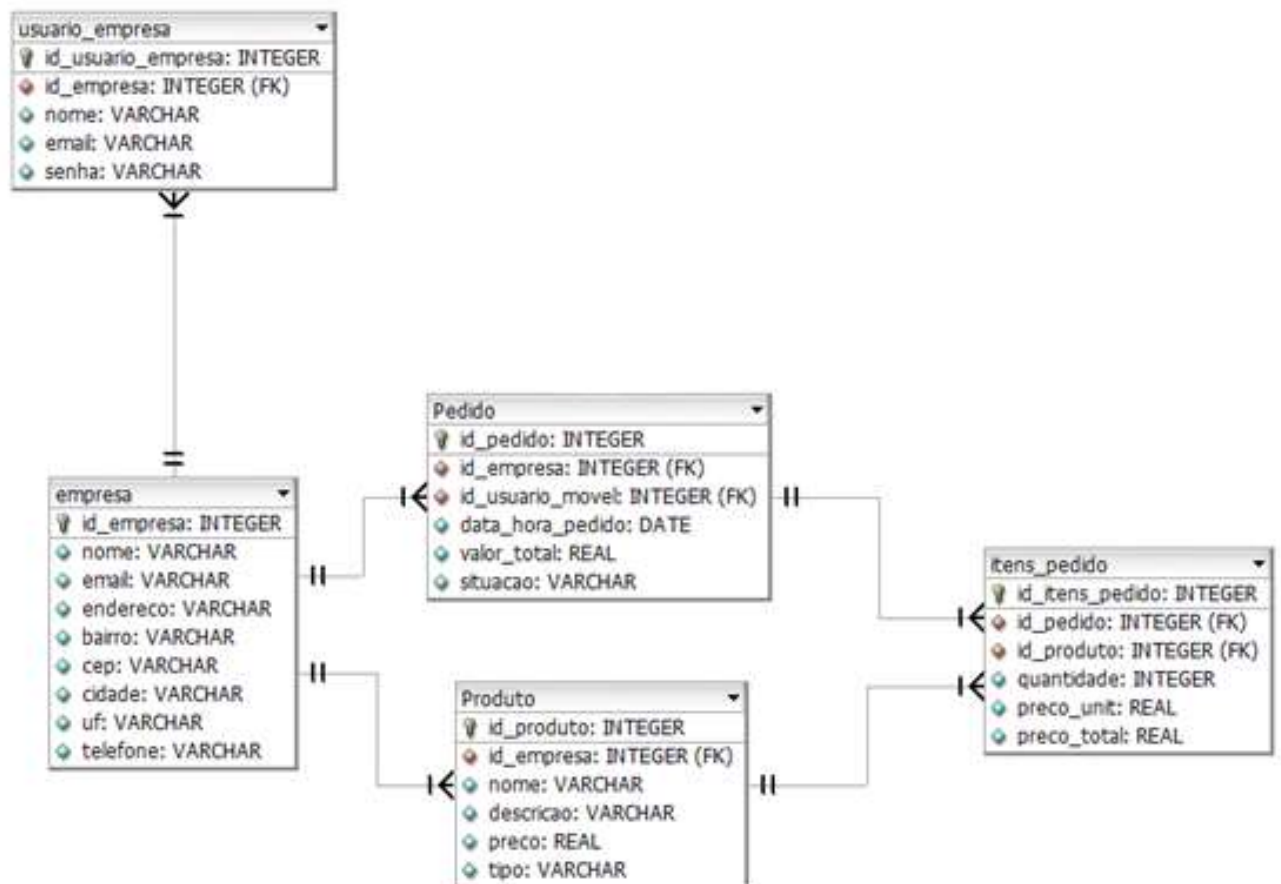


Figura 14: Diagrama de Classe do Sistema Web em C#

## 8 DIAGRAMA DE ENTIDADE E RELACIONAMENTO – DER

Esse diagrama representa o Sistema de Controle de Pedido completo, ou seja, o relacionamento das tabelas sem diferenciar o relacionamento de cada aplicação composta pelo sistema.

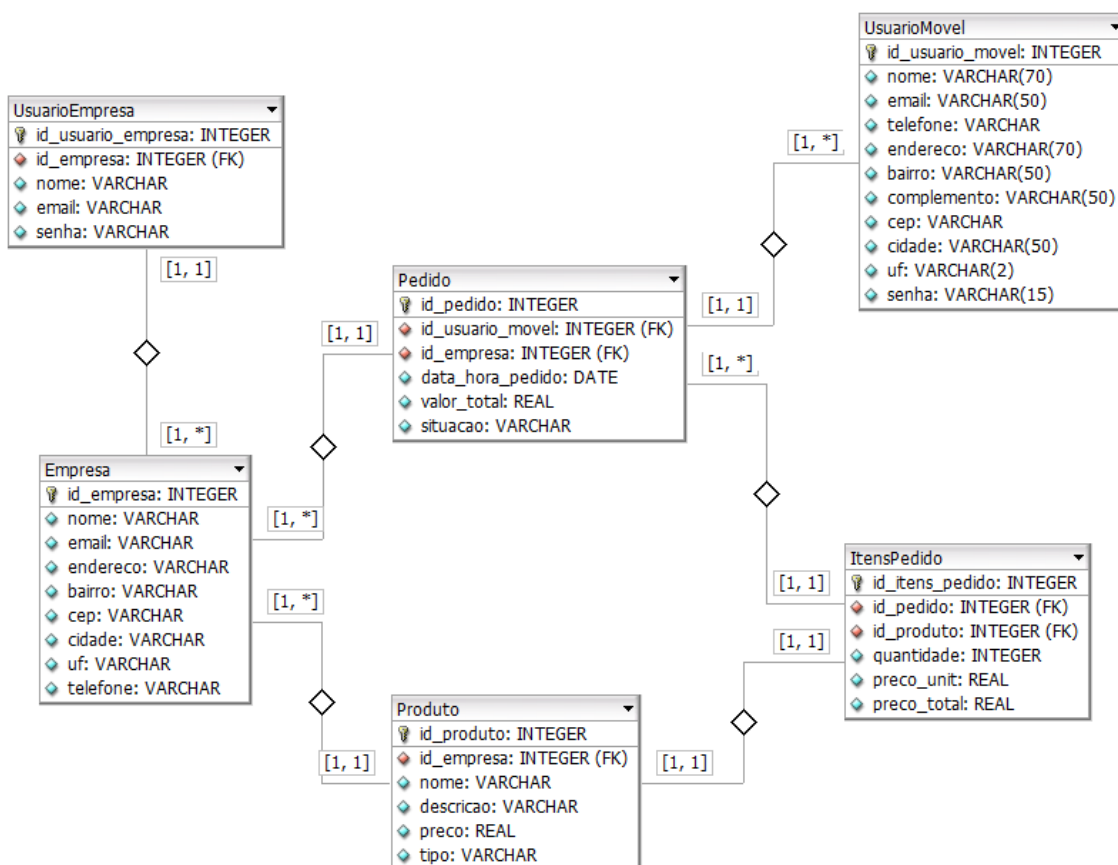


Figura 15: Diagrama de Entidade e Relacionamento do sistema completo.

## 8.1 JAVA

Esse diagrama representa o relacionamento entre as tabelas que a aplicação web em Java irá utilizar. A tabela UsuarioEmpresa contém os dados do administrador, a tabela Empresa para conter os dados da empresa e a tabela produtos para os produtos oferecidos pela empresa.

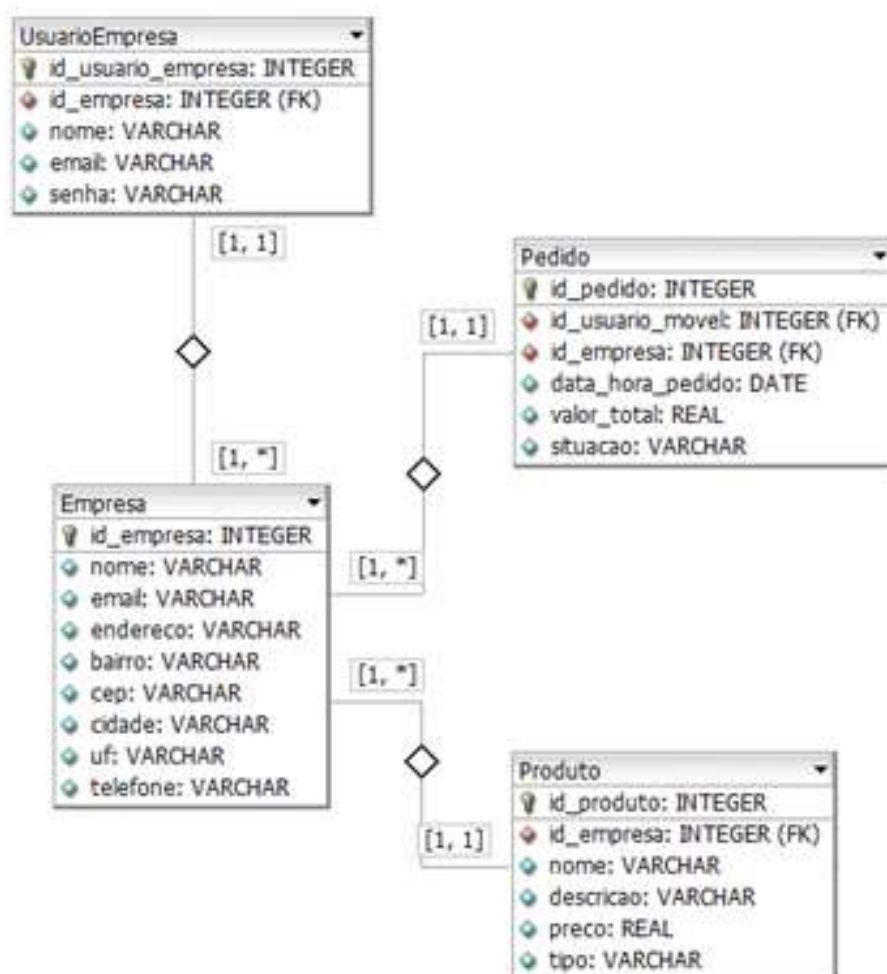


Figura 16: Diagrama de Entidade e Relacionamento da aplicação Web em Java.

## 8.2 ANDROID

Esse diagrama representa o relacionamento das tabelas na aplicação móvel. A tabela de UsuarioMovel é para os dados do usuário da aplicação, as tabelas Pedido, Produto e ItensPedido é para a controlar o pedido que o usuário irá fazer contendo cada produto desejado.

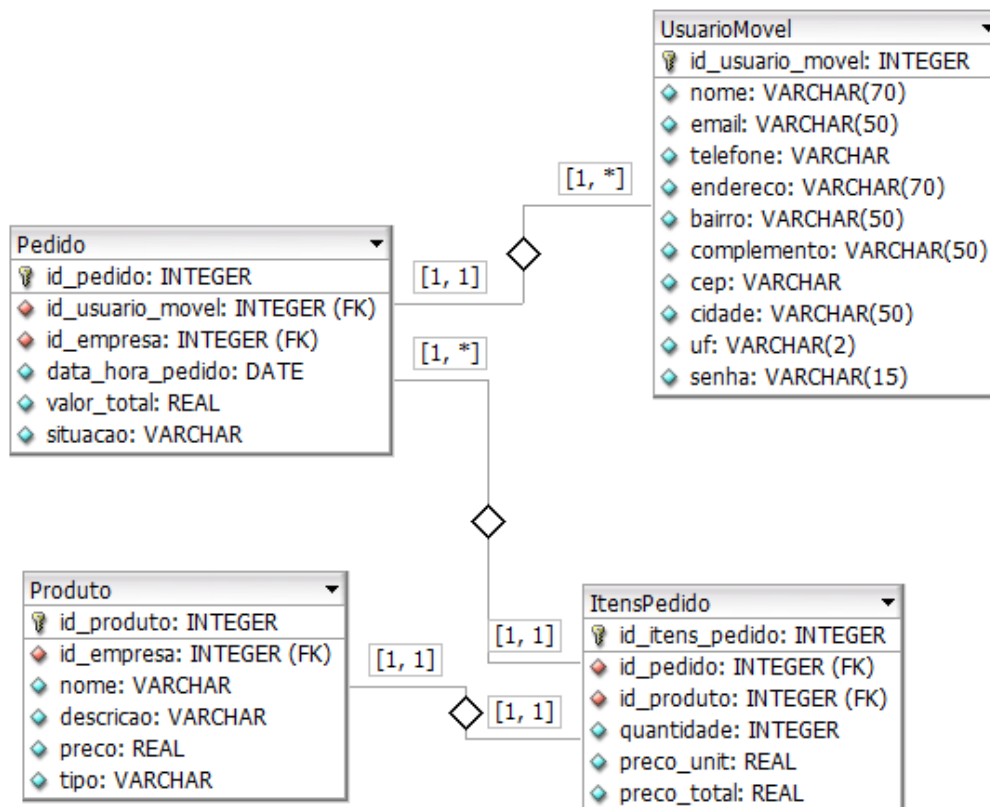


Figura 17: Diagrama de Entidade e Relacionamento na aplicação móvel

### 8.3 C#

Esse diagrama representa o relacionamento entre as tabelas que a aplicação web em C# irá utilizar. Sua finalidade é controlar as classes Pedido, Produto e ItensPedido que contém os dados dos produtos recebido em cada pedido.

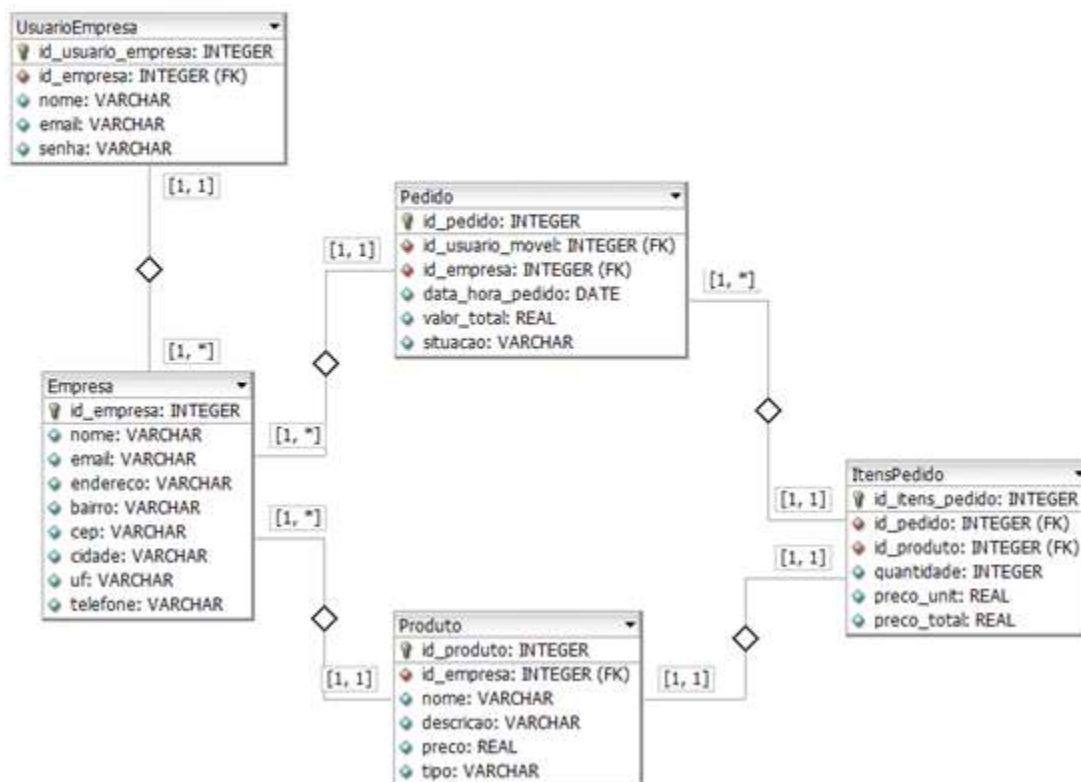


Figura 18: Diagrama de Entidade e Relacionamento na aplicação Web em C#

## 9 DIAGRAMAS DE ATIVIDADE

### 9.1 JAVA

Esse diagrama mostra as principais atividades que o Web Site desenvolvido em Java irá executar durante seu funcionamento. Ele descreve as seguintes atividades: cadastrar e manter o cadastro de todos os produtos e manter o estado do pedido, em que se o estado do pedido passar para “PRONTO”, o administrador irá informar ao cliente que seu pedido está sendo enviado.

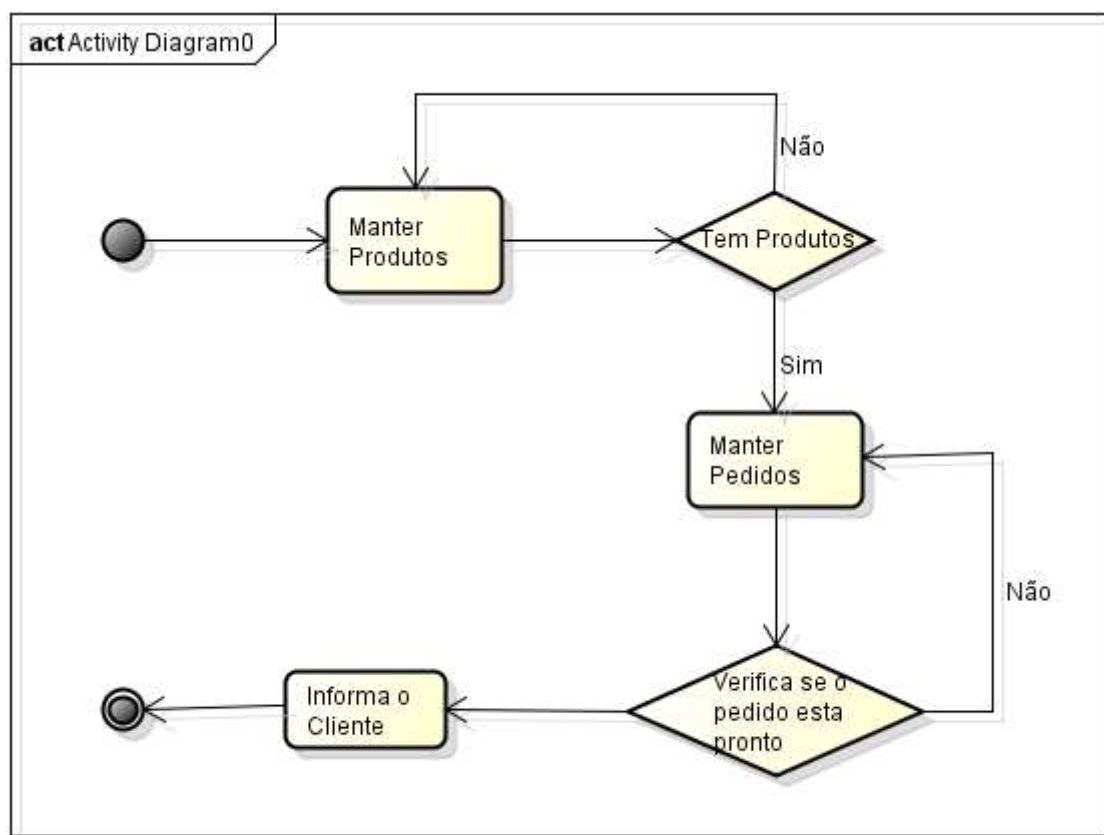


Figura 19: Diagrama de atividade do Web Site em Java

## 9.2 ANDROID

Este diagrama representa as principais atividades da aplicação móvel. Ela inicia com o cadastro do usuário da aplicação, após ter efetuado o seu cadastro ele poderá fazer seu pedido, mas primeiro ele terá que escolher os itens a ser pedido e em seguida poderá confirmar o seu pedido.

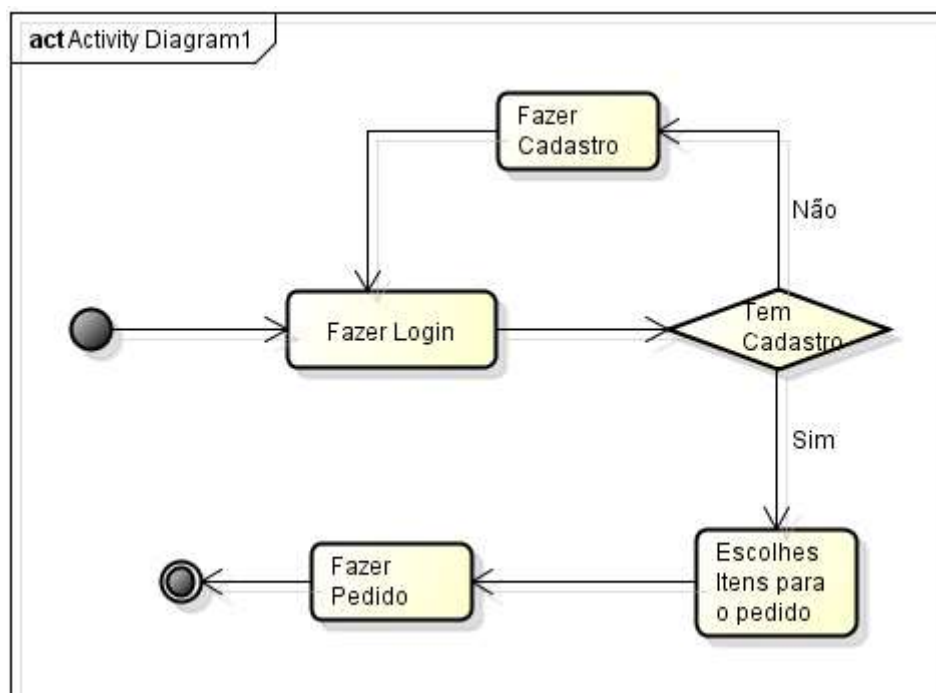


Figura 20: Diagrama de Atividade da Aplicação Móvel

### 9.3 C#

Como os outros diagramas, este descreve as principais atividade do Web Site em C#. Ele será utilizado para verificar cada item que um pedido possui. As atividade tem principio na hora em que o seu usuario verifica se o pedido está pronto. Tendo confirmado, o usuário altera o estado do pedido para “PRONTO”.

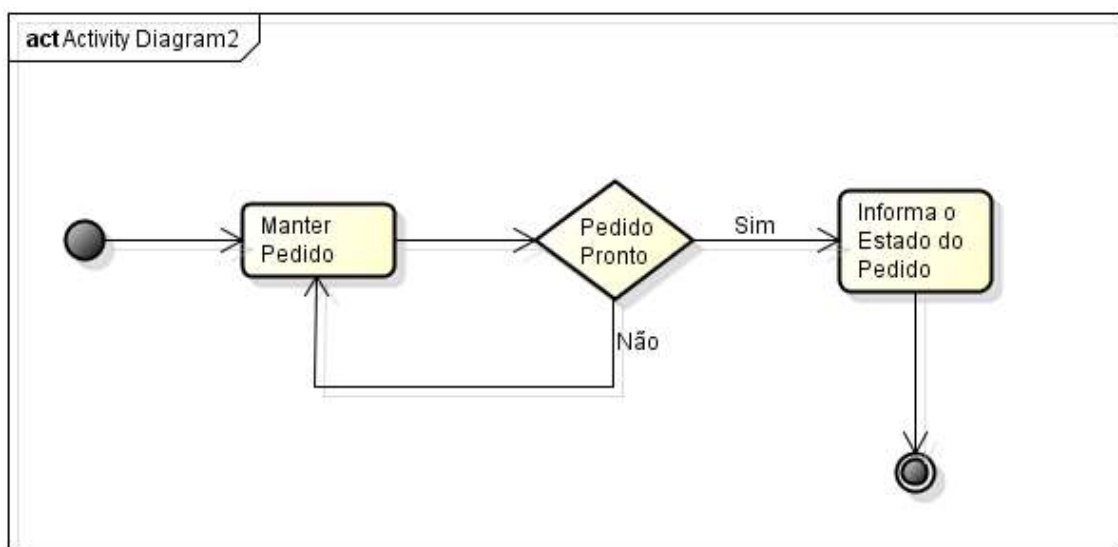


Figura 21: Diagrama de Atividade do Web Site em C#



## 10 DIAGRAMA DE SEQUÊNCIA

### 10.1 JAVA

Os diagramas a seguir irão mostrar as principais atividades e as sequencias em que ela serão executadas no Web Site em Java.

#### 10.1.1 Manter Pedido

Esse diagrama mostra as sequencias de atividades que serão necessárias para a execução da atividade manter pedido. O administrador tem a tarefa de comunicar ao usuário que o seu pedido, ao ficar pronto, está sendo enviado.

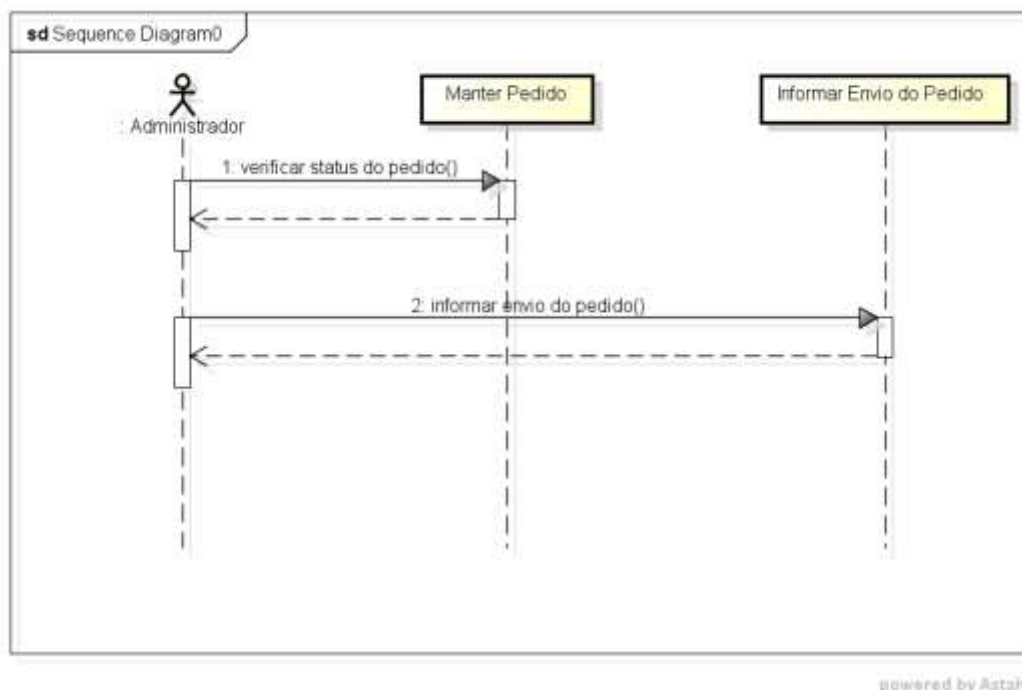
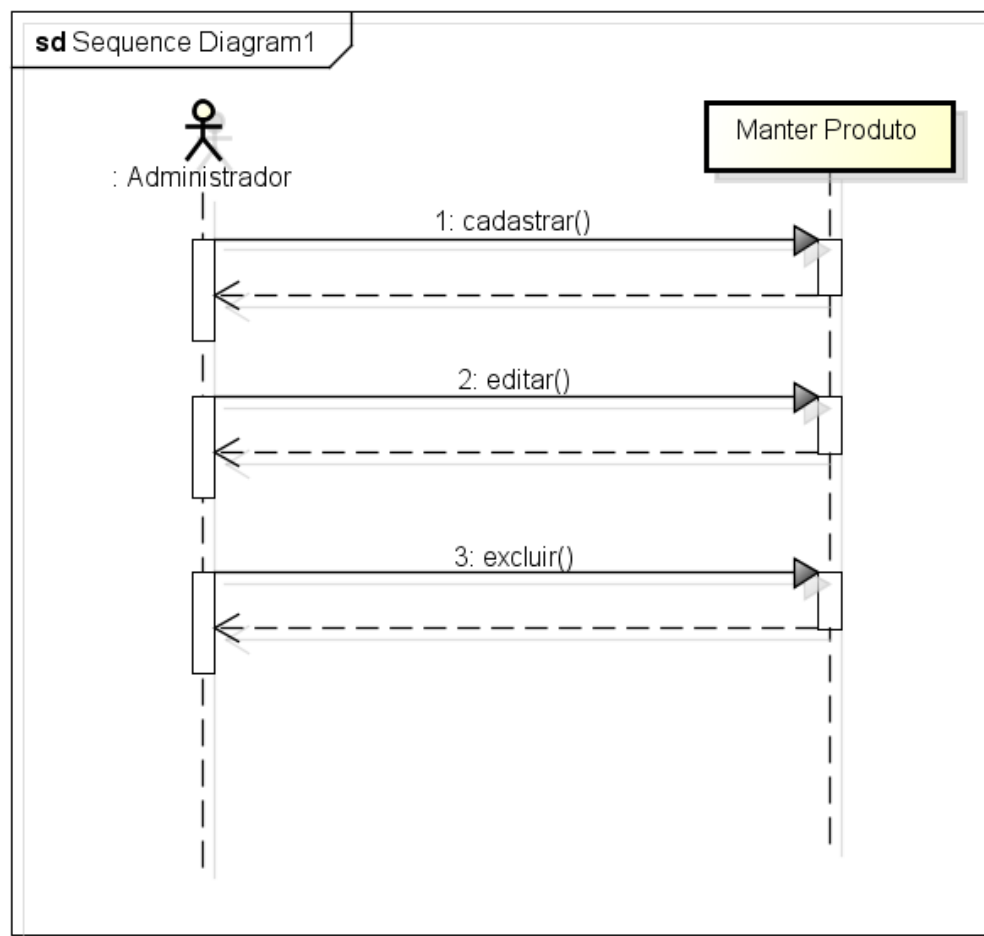


Figura 22: Diagrama de Sequência do Web Service em Java - Manter Pedido

### 10.1.2 Manter Produto

Essa é a sequencia de atividade que o administrador tem que fazer ao manter o cadastro de seus produtos. Aqui fala que ele pode cadastras um novo produto, editar seu produto, quando for necessário fazer alguma alteração em seus dados e excluir ele.



powered by Astah

Figura 23: Diagrama de Sequência do Web Site em Java - Manter Produto

## 10.2 ANDROID

Os diagramas a seguir mostrarão também as principais atividades na aplicação móvel e descrever a sua sequência de execução.

### 10.2.1 Cadastro Usuário Móvel

Esse diagrama descreve a sequência em que será executada a atividade de cadastro do usuário móvel. O usuário fornecera os dados necessários para efetuar o seu cadastro, e assim que confirmado, o aplicativo móvel manda um arquivo em formato JSON para o Web Service para que seus dados sejam gravados no Banco de Dados.

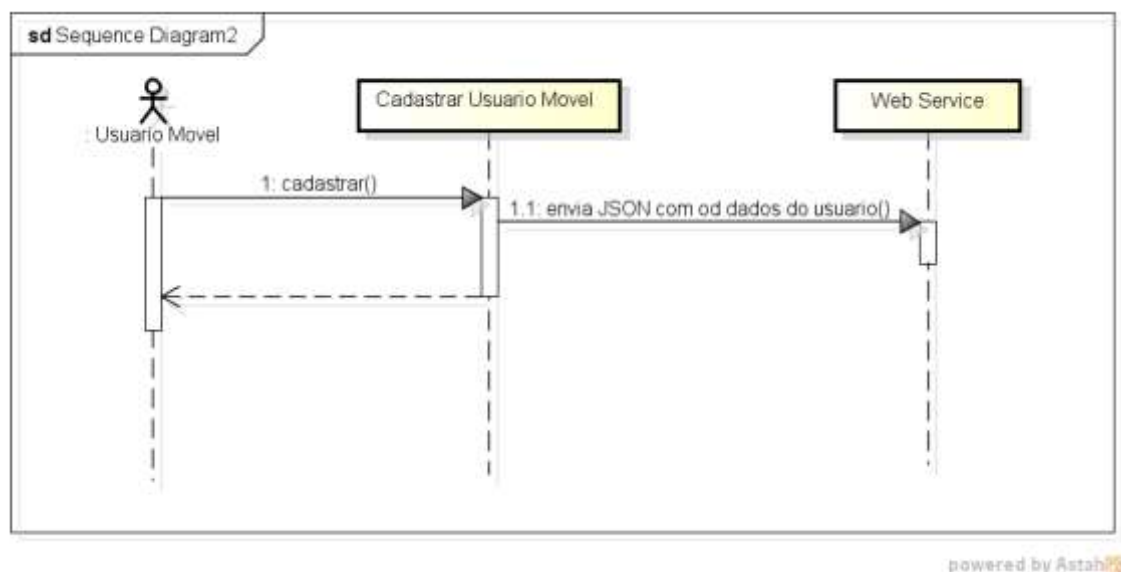


Figura 24: Diagrama de sequência da Aplicação Móvel - Cadastro de Usuário Móvel

### 10.2.2 Manter Pedido

Aqui mostra a sequencia em que a atividade de manter pedido é executada. O usuário seleciona cada produto que ele queira adicionar no seu pedido. Selecionando todos, ele confirma o seu pedido fazendo com que o aplicativo móvel mande para o Web Service um arquivo JSON com cada item que seu pedido possui, podendo então gravar no Banco de Dados o seu pedido junto com o itens do pedido.

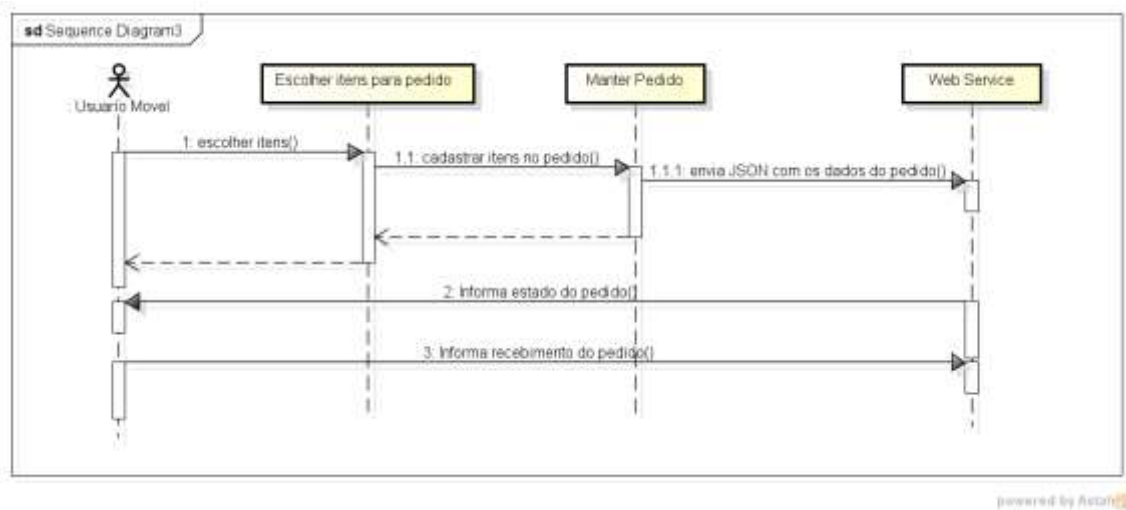


Figura 25: Diagrama de Sequência da Aplicação Móvel - Manter Pedido

## 10.3 C#

### 10.3.1 Manter Pedido

Esse ultimo diagrama mostra a sequencia de execucao das atividades de manter o pedido no Web Site em C#. Quando o pedido do cliente ficar pronto o administrador informa no sistema o estado atual do pedido, automaticamente a aplicacao manda um arquivo JSON para o Web Service com o novo estado do pedido.



Figura 26: Diagrama de Sequência do Web Site em C# - Manter Pedido

## 11 CRONOGRAMA

<b>Mês/Ano</b>	<b>Atividade</b>
Novembro/2013	Pesquisas de desenvolvimento Levantamento de necessidades Levantamento de requisitos
Dezembro/2013	Análise dos requisitos Gerenciamento dos requisitos
Janeiro/2014	Diagrama de Casos de Uso Diagrama de Classe Diagrama de Atividade Diagrama de entidade relacionamento Diagrama de Sequência
Fevereiro/2014	Análise dos diagramas Levantamento de acessórios necessários Análise dos requisitos
Março/2014	Criação do Banco de Dados Implementação
Abril/2014	Implementação
Maió/2014	Implementação
Junho/2014	Implementação
Julho/2014	Implementação Testes finais
Agosto/2014	Instalação do sistema Conclusão

## 12 TELAS DO SISTEMA

### 12.1 APLICAÇÃO JAVA

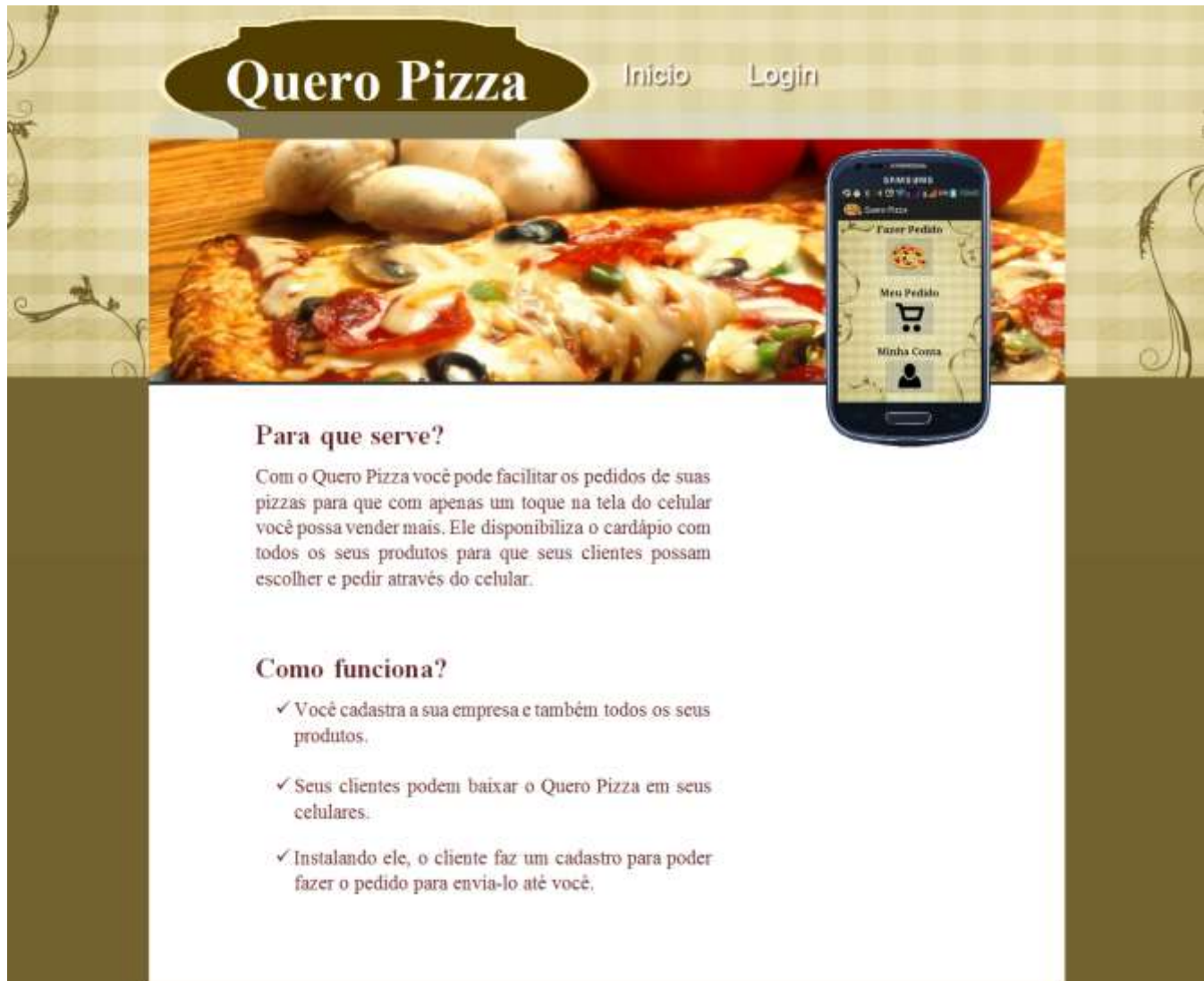


Figura 27: Tela inicial da aplicação Java



Figura 28: Tela de Login



Figura 29: Tela de Pedidos



# Quero Pizza

## Produtos

» » Painel de Controle » Produtos » Meus Dados » Sair

### Pizzas

(1 of 1)

Nome	Descrição	Preço (R\$)		
CHEFE	Calabresa moída e cobertura de catupiry	19,55		
COREANA	Calabresa, mussarela, tomate fatiado e salinha	20,55		
FLORENÇA	Presunto, calabresa, vinagrete, bacon e cebola	20,55		
INDONESIA	Calabresa moída, pimentão, mussarela e salinha	19,55		
JAGUARE	Frango, escarola, mussarela e rodela de calabresa	20,55		
MADRI	Presunto, catupiry, cebola e salinha	20,55		
PIZZA TOP	Molho, mussarela, lombo, calabresa, frango, presunto	21,55		
POLONESA	Calabresa, ovos, mussarela e salinha	19,55		

(1 of 1)

### Bebidas

Cadastrar mais produtos

Figura 30: Tela de Produtos

The image shows a web application interface for 'Quero Pizza'. At the top, there is a dark brown banner with the text 'Quero Pizza' in a white, serif font. Below this is a dark brown header bar with the word 'Produtos' in white. A breadcrumb trail is visible: 'Painel de Controle > Produtos > Meus Dados > Sair'. A navigation menu contains three items: 'Pizzas', 'Bebidas', and 'Cadastrar mais produtos', with the last one highlighted in blue. The main form area is a light beige box with the following fields: 'Nome:' with a text input; 'Tipo:' with a dropdown menu showing 'Pizza' and 'Bebida'; 'Descrição:' with a large text area; and 'Preço:' with a text input. At the bottom of the form are two buttons: 'Cancelar' and 'Confirmar'.

Figura 31: Tela de Cadastro de Produtos

## 12.2 APLICAÇÃO ANDROID



Figura 32: Tela Inicial



The image shows a mobile application interface for a pizza business named "Quero Pizza". The screen is titled "Cadastro" (Registration) and features a decorative background with a checkered pattern and floral motifs. The registration form consists of several fields, each with a label and an asterisk indicating it is required:

- Nome \***: Input field with placeholder text "Nome".
- Email \***: Input field with placeholder text "Email".
- Telefone \***: Input field with placeholder text "Telefone".
- Endereço \***: Input field with placeholder text "Endereço".
- Bairro \***: Input field with placeholder text "Bairro".
- Complemento**: Input field with placeholder text "Complemento".
- CEP \***: Input field with placeholder text "CEP".
- Cidade \***: Input field with placeholder text "Cidade".
- UF \***: Input field with placeholder text "UF".
- Senha \***: Input field with placeholder text "Senha".
- Confirmar Senha \***: Input field with placeholder text "Confirmar Senha".

At the bottom of the screen, there are two buttons: a grey button with a black "X" icon and a grey button with a black checkmark icon.

Figura 33: Tela de cadastro de Usuário



Figura 34: Tela de Opções do Usuário



Figura 35: Tipos de Produto



Figura 36: Lista de Produto

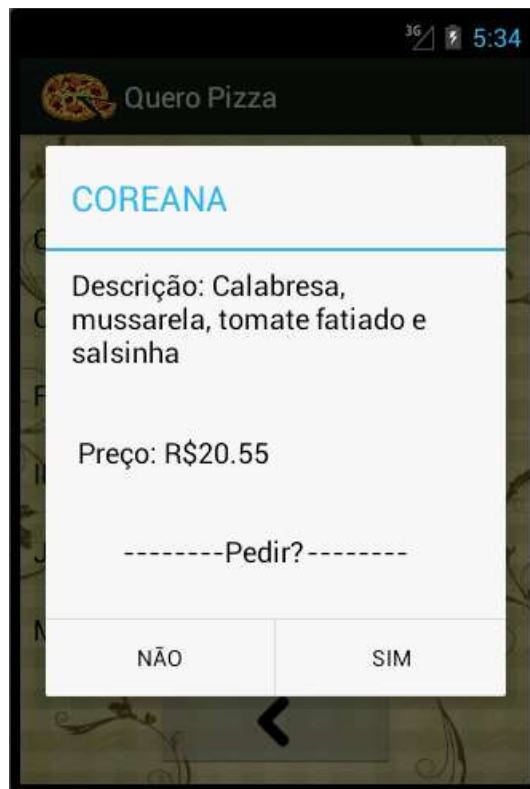


Figura 37: Descrição do Produto



## 12.3 APLICAÇÃO C#



Figura 38: Tela de Login



Figura 39: Tela de Pedidos

## 13 LINHAS DE CÓDIGO

### 13.1 JAVA

Método REST para cadastro de usuário móvel pelo celular. Recebe um JSON com todos os dados, converte para o objeto e grava no banco de dados.

```
44  /**
45   * Cadastra um novo UsuarioMovel
46   *
47   * @param userDTO
48   * @return
49   * @throws Throwable
50   */
51  @POST
52  @Path("/newusermov")
53  @Produces({ "application/json" })
54  public Response newUserMov(UsuarioMovelDTO userDTO) throws Throwable {
55      Log.info(logger, MessageLog.USUARIO_MOVEL_REST_NEW);
56      try {
57          initServices();
58          UsuarioMovel usuario = new UsuarioMovel();
59          usuario.setNome(userDTO.getNome());
60          usuario.setEmail(userDTO.getEmail());
61          usuario.setTelefone(userDTO.getTelefone());
62          usuario.setEndereco(userDTO.getEndereco());
63          usuario.setBairro(userDTO.getBairro());
64          usuario.setComplemento(userDTO.getComplemento());
65          usuario.setCidade(userDTO.getCidade());
66          usuario.setCep(userDTO.getCep());
67          usuario.setUf(userDTO.getUf());
68          usuario.setSenha(userDTO.getSenha());
69
70          // Cadastro um novo usuario
71          usuarioService.inserir(usuario);
72
73          UsuarioMovelConverter userConv = new UsuarioMovelConverter(logger);
74          UsuarioMovelDTO result = userConv.entityToDTO(usuario);
75
76          return response(result, false);
77      } catch (RestConverterException e) {
78          return response(StatusRest.INTERNAL_SERVER_ERROR);
79      }
80  }
```

Figura 40: Método Post REST para cadastro de Usuário Móvel

Método REST para consulta de usuário móvel, para fazer consulta dos dados do cliente durante o *login*. Recebe o e-mail do usuário para consultar no banco de dados.

```
82  /**
83   * Consulta um UsuarioMovel
84   *
85   * @param userDTO
86   * @return
87   * @throws Throwable
88   */
89  @GET
90  @Path("/getusermov")
91  @Produces({ "application/json" })
92  public Response getUserMov() throws Throwable {
93      Log.info(logger, MessageLog.USUARIO_MOVEL_REST_CONSULT);
94      try {
95          initServices();
96          UsuarioMovel usuario = new UsuarioMovel();
97
98          // Consulta o usuario
99          usuario = usuarioService.getById((long) 1);
100
101          UsuarioMovelConverter userConv = new UsuarioMovelConverter(logger);
102          UsuarioMovelDTO result = userConv.entityToDTO(usuario);
103
104          return response(result, false);
105      } catch (RestConverterException e) {
106          return response(StatusRest.INTERNAL_SERVER_ERROR);
107      }
108  }
```

Figura 41: Método REST para consulta de Usuário Móvel

Método REST para consulta de produtos, retorna um JSON com os dados de todos os produtos

```
83  /**
84   * Consulta o produto
85   *
86   * @param prodDTO
87   * @return
88   * @throws Throwable
89   */
90  @GET
91  @Path("/getprod/{tipo}")
92  @Produces({ "application/json" })
93  public List<Produto> getProd(@PathParam("tipo") String tipo) throws Throwable {
94      Log.info(logger, MessageLog.PRODUTO_REST_CONSULT);
95
96      initServices();
97      List<Produto> produtoList;
98      Produto produto = new Produto();
99
100     produto.setTipo(tipo);
101
102     // Consulta o Produto
103     produtoList = produtoService.getByCriteria(produto);
104
105     return produtoList;
106 }
```

Figura 42: Método REST para consulta de todos os produtos

## 13.2 ANDROID

Método que acessa o REST que de usuário móvel recebendo um JSON com os dados do usuário. Passa o e-mail por URL para que no REST a consulta seja pelo e-mail.

```
28  /**
29  * Busca os dados de UsuarioMovel no REST apartir de seu email para login
30  *
31  * @return UsuarioMovel
32  */
33  public UsuarioMovel getLogin(String email) {
34      String json = getRESTFileContent("http://10.4.1.19:80/pedidoonline-rest/usuariomovel/getusermovcriteria/"
35          + email);
36
37      Log.i(TAG, "Endereco: " + json);
38      UsuarioMovel usuario = new UsuarioMovel();
39      try {
40          JSONObject usuarioJson = new JSONObject(json);
41
42          usuario.setId(usuarioJson.getLong("id"));
43          usuario.setNome(usuarioJson.getString("nome"));
44          usuario.setEmail(usuarioJson.getString("email"));
45          usuario.setTelefone(usuarioJson.getString("telefone"));
46          usuario.setEndereco(usuarioJson.getString("endereco"));
47          usuario.setBairro(usuarioJson.getString("bairro"));
48          usuario.setComplemento(usuarioJson.getString("complemento"));
49          usuario.setCep(usuarioJson.getString("cep"));
50          usuario.setCidade(usuarioJson.getString("cidade"));
51          usuario.setUF(usuarioJson.getString("uf"));
52          usuario.setSenha(usuarioJson.getString("senha"));
53          return usuario;
54      } catch (JSONException e) {
55          Log.e("NGVL", "Erro ao fazer parsing do JSON", e);
56          return usuario = null;
57      }
58  }
```

Figura 43: Método de Acesso ao REST para consultar Usuário Móvel

Método que acessa o REST e recebe um JSON com os dados de todos os produtos referentes ao seu tipo.

```
30  /**
31  * Busca os dados de Produto no REST
32  *
33  * @return List<Produto>
34  * @throws JSONException
35  */
36  public List<Produto> getDados(String tipo) {
37      List<Produto> produtosList = new ArrayList<Produto>();
38      String json = getRESTFileContent("http://10.4.1.19:80/pedidoonline-rest/produto/getprod/" + tipo);
39      String jsonAux;
40      JSONArray JSONList;
41      try {
42          JSONList = new JSONArray(json);
43          int tamList = JSONList.length();
44          for (int i = 0; i < tamList; i++) {
45              Produto prod = new Produto();
46              prod.setId(JSONList.getJSONObject(i).getLong("idProduto"));
47              jsonAux = JSONList.getJSONObject(i).getString("idEmpresa");
48              prod.setEmpresa(convertJSONEmpresa(jsonAux));
49              prod.setNome(JSONList.getJSONObject(i).getString("nome"));
50              prod.setTipo(JSONList.getJSONObject(i).getString("tipo"));
51              prod.setDescricao(JSONList.getJSONObject(i).getString("descricao"));
52              prod.setPreco(JSONList.getJSONObject(i).getDouble("preco"));
53              produtosList.add(prod);
54          }
55      } catch (JSONException e) {
56          // TODO Auto-generated catch block
57          e.printStackTrace();
58      }
59      return produtosList;
60  }
```

Figura 44: Método de Acesso ao REST para consulta de produtos

### 13.3 C#

Método genérico que acessa o REST para consultas. Recebe uma URL como parâmetro.

```
//Acessa o RST e retorna um JSON.
public string GetJSONString(string url) {
    try
    {
        HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
        WebResponse response = request.GetResponse();
        using (Stream stream = response.GetResponseStream()) {
            StreamReader reader = new StreamReader( stream, Encoding.UTF8);
            return reader.ReadToEnd();
        }
    }
    catch (Exception )
    {
        return null;
    }
}
```

Figura 45: Método Genérico de acesso ao REST para consultas

## 14 CONCLUSÃO

Este trabalho teve como assunto principal a integração de software através de Web Services, que é uma tecnologia utilizada para realizar a comunicação e a integração de aplicações. Foi desenvolvido um sistema com três aplicativos, criados em plataformas distintas para que se pudessem aplicar os conceitos da tecnologia Web Service, fazendo com que cada um desses aplicativos se comunique um com os outros.

O desenvolvimento deste trabalho foi muito importante para o aprimoramento do conhecimento sobre o assunto, que me ajudará a ser um profissional com conhecimento e experiência para atuar na área de desenvolvimento.

Para trabalhos futuros pretende-se desenvolver uma aplicação para o sistema operacional Windows Phone, bem como melhorar o sistema desenvolvido em Java para que várias empresas possam utilizá-lo, ou seja, tornar o sistema multi-empresas, objetivando atingir novos clientes.



## REFERÊNCIAS BIBLIOGRÁFICAS

- IBGE. Ibge: uso de celular e internet cresceu mais de 100% no brasil em seis anos.. , 2013. Disponível em: <<http://www.tecmundo.com.br/brasil/39797-ibge-uso-de-celular-e-internet-cresceu-mais-de-100-no-brasil-em-seis-anos.htm>>. Acesso em: 28 oct. 2013.
- ENTENDENDO os webservices. , 2006. Disponível em: <<http://imasters.com.br/artigo/4245/>>. Acesso em: 28 oct. 2013.
- WEB SERVICES. construindo, disponibilizando e acessando web services via j2se e j2me. , 2010. Disponível em: <<http://javafree.uol.com.br/artigo/871485/Web-Services-Construindo-disponibilizando-e-acessando-Web-Services-via-J2SE-e-J2ME.html>>. Acesso em: 28 oct. 2013.
- UTILIZANDO web services no google android. , 2010. Dispon?vel em: <[http://www.cesar.org.br/site/files/file/WM23\\_Android\\_WebServices.pdf](http://www.cesar.org.br/site/files/file/WM23_Android_WebServices.pdf)>. Acesso em: 29 oct. 2013.
- A POPULARIZAÇÃO do smartphone. smartphone na sociedade. , 2010. Disponível em: <<http://smartphonenasociedade.blogspot.com.br/2011/06/popularizacao-do-smartphone.html>>. Acesso em: 29 oct. 2013.
- ALEXANDRE SAUDATE. *Soa aplicado integrando com web services e além*. São Paulo Sp: Casa do Código, 2011. Disponível em: <<http://www.casadocodigo.com.br/products/livro-soa-webservices>>. Acesso em: 30 oct. 2013.
- INDRUSIAK, Leandro Soares. Linguagem java. , 1996. Disponível em: <<http://www.cin.ufpe.br/~arfs/introjava.pdf>>. Acesso em: 26 dez. 2013.
- MENDES, Douglas Rocha. Programação java com ênfase em orientação a objetos. , 2010. Disponível em: <<https://www.novatec.com.br/livros/javaoo/capitulo9788575221761.pdf>>. Acesso em: 26 dez. 2013.
- POR DENTRO do primefaces 2.2 - artigo revista java magazine 93 leia mais em: por dentro do primefaces 2.2 - artigo revista java magazine 93. , 2012. Disponível em: <<http://www.devmedia.com.br/por-dentro-do-primefaces-2-2-artigo-revista-java-magazine-93/21651>>. Acesso em: 27 dez. 2013.
- BUENO, Kassia Jaqueline. O que é jsf (java server faces) ?. , 2013. Disponível em: <<http://fabrica.ms.senac.br/2013/06/o-que-e-jsf-java-server-faces/>>. Acesso em: 27 dez. 2013.
- JUNIOR, Prof. Pasteur Ottoni de Miranda. Enterprise java beans . , 2011. Disponível em: <<http://www.tesestec.com.br/pasteurjr/ejb.pdf>>. Acesso em: 26 dez. 2013.
- O QUE é uml?. , 2007. Disponível em: <<http://danielteofilo.wordpress.com/2007/10/18/o-que-e-uml/>>. Acesso em: 26 dez. 2013.
- MARTINEZ, Marina. Uml. , 2007. Disponível em: <<http://www.infoescola.com/engenharia-de-software/uml/>>. Acesso em: 27 dez. 2013.

- **DEVELOPERNS. Android developer tools. , 2009. Disponível em: <<http://developer.android.com/license.html>>. Acesso em: 27 dez. 2013.**
- **MONTEIRO, João Bosco. Google android. crie aplicações para celulares e tablets. São Paulo: Casa do Código, 2013.**
- **DEITEL, Paul; DAITEL, Abbey; DEITEL, Harvey. Android para programadores: Uma abordagem baseada em aplicativos. São Paulo: Bookman, 2012.**