

HUMBERTO AFONSO NUNES DE MELO

**SISTEMA DE GESTÃO:  
PARA EMPRESAS DO RAMO DE  
ALIMENTAÇÃO.**

Brasil, Diamantina-MG

12/08/2016, versão-3.0.1



HUMBERTO AFONSO NUNES DE MELO

**SISTEMA DE GESTÃO:  
PARA EMPRESAS DO RAMO DE ALIMENTAÇÃO.**

Monografia de projeto final de graduação apresentada ao curso Bacharelado em Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito parcial para obtenção de graduação no curso bacharelado de Sistemas de Informação. Sob o título "SISTEMA DE GESTÃO: PARA EMPRESAS DO RAMO DE ALIMENTAÇÃO." defendida por Humberto Afonso Nunes de Melo, Diamantina, Minas Gerais.

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

FACULDADE DE CIÊNCIAS EXATAS

CURSO DE SISTEMAS DE INFORMAÇÃO

Orientador: Pof. Dr. Alessandro Vivas Andrade

Brasil, Diamantina-MG

12/08/2016, versão-3.0.1

---

HUMBERTO AFONSO NUNES DE MELO  
SISTEMA DE GESTÃO:  
PARA EMPRESAS DO RAMO DE ALIMENTAÇÃO./ HUMBERTO AFONSO  
NUNES DE MELO. – Brasil, Diamantina-MG, 12/08/2016, versão-3.0.1-

71 p.

Orientador: Pof. Dr. Alessandro Vivas Andrade

Monografia (Graduação) – UNIVERSIDADE FEDERAL DOS VALES DO JE-  
QUITINHONHA E MUCURI  
FACULDADE DE CIÊNCIAS EXATAS  
CURSO DE SISTEMAS DE INFORMAÇÃO, 12/08/2016, versão-3.0.1.

1. Sistemas de Informação. 2. Gerenciamento de restaurantes. I. Alessandro  
Vivas Andrade. II. Universidade Federal dos Vales do Jequitinhonha e Mucuri-  
UFVJM-. III. Faculdade de Ciências Exatas- FACET-.

---

HUMBERTO AFONSO NUNES DE MELO

## **SISTEMA DE GESTÃO: PARA EMPRESAS DO RAMO DE ALIMENTAÇÃO.**

Monografia de projeto final de graduação apresentada ao curso Bacharelado em Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito parcial para obtenção de graduação no curso bacharelado de Sistemas de Informação. Sob o título "SISTEMA DE GESTÃO: PARA EMPRESAS DO RAMO DE ALIMENTAÇÃO." defendida por Humberto Afonso Nunes de Melo, Diamantina, Minas Gerais.

Trabalho aprovado. Brasil, Diamantina-MG, 12 de Agosto de 2016:

---

**Pof. Dr. Alessandro Vivas Andrade**  
Orientador

---

**Prof. Dr. Luciana Pereira de Assis**  
Convidado 1

---

**Prof. Ms. Raphael Santin**  
Convidado 2

Brasil, Diamantina-MG  
12/08/2016, versão-3.0.1



*Este trabalho é dedicado às crianças adultas que,  
quando pequenas, sonharam em se tornar cientistas.*





# Agradecimentos

Agradeço a Deus pela vida, pela saúde e pela família que ele me fez nascer.

Agradeço ao amor incondicional dos meus pais, Sostenes Pedro Nunes de Melo e Rosane Afonso Franco de Melo, aos meus irmãos Túlio e Larrissa pelo apoio e por sempre me apoiarem e estarem do meu lado, a minha namorada Camila que esteve presente nos momentos especiais.

Agradeço aos meus Avós e Avôs, pelos ensinamentos e valores passados a mim como exemplo de vida.

Agradeço a todos os meus professores que me ensinaram durante essa caminhada o significado de estudar.



*“Um dia, quando olhares para trás,  
verás que os dias mais belos foram aqueles em que lutaste.”  
(Sigmund Freud.)*



# Resumo

Os sistemas de informação são as ferramentas mais utilizadas nos dias de hoje, estando presente de diversas formas e complexidades de acordo com a sua especificidade em que fosse realizado sua concepção e aplicação. Visto que a maioria dos restaurantes não possuem um sistema de informação especializado para sua demanda, percebeu-se a necessidade da criação de uma ferramenta que otimizasse o atendimento ao cliente e maior controle financeiro deste tipo de ambiente de trabalho. O sistema utiliza-se de *Framework* de desenvolvimento responsivo BootsTrap, JQuery para criação das iterações de interfaces, HTML, PHP e SGBD Mysql para criação do sistema e interação destes elementos, caracterizando uma proposta de desenvolvimento de um *Software* com alto potencial para aplicação de testes em um restaurante.

**Palavras-chaves:** Sistemas de informação, desenvolvimento de sistemas gerencias, PHP, MySql, Bootstrap, JQuery, Web, ERP.



# Abstract

*Information systems are the tools most commonly used today, and is present in various forms and complexity according to their specificity that you were conducted their design and implementation. Since most restaurants do not have a specialized information system to your demand, we realized the need to create a tool that optimize customer service and greater financial control of this type of work environment. The system makes use of textit Framework Bootstrap responsive development, jQuery for creating iterations of interfaces, HTML, PHP and SGBD MySQL for system creation and interaction of these elements, featuring a proposal to develop a textit Software with high potential for application testing in a restaurant.*

**Key-words:** *Information System, System Devolepment Management, PHP, WEB,Bootstrap, jQuery, ERP.*





# Lista de ilustrações

Figura 1 – Modelo cascata fonte, Sommerville (2007) . . . . .	29
Figura 2 – Modelo incremental Fonte: Peters(2000, p.42) . . . . .	30
Figura 3 – Modelo prototipação Fonte: Peters(2000, p.42) . . . . .	30
Figura 4 – Scrum ciclo único de trabalho Fonte: Sbok Guide (2013, p.8) . . . . .	31
Figura 5 – Orientação a objetos Fonte: Apostila k19 (2013) . . . . .	33
Figura 6 – Caso de uso Administrador . . . . .	47
Figura 7 – Caso de uso do usuário . . . . .	48
Figura 8 – Teste 1 . . . . .	65
Figura 9 – Teste 2 . . . . .	66
Figura 10 – Item com adicional monetário. . . . .	67
Figura 11 – Cadastro de cliente. . . . .	67
Figura 12 – Histórico de Cardápios. . . . .	68
Figura 13 – Montar cardápio. . . . .	68
Figura 14 – Item sem adicional monetário. . . . .	69
Figura 15 – Contas a Pagar/Pagas . . . . .	69
Figura 16 – Contas a Receber/Recebidas. . . . .	69
Figura 17 – Fazer pedidos. . . . .	70
Figura 18 – Lista de pedidos. . . . .	70
Figura 19 – Ordem de serviço de um pedido. . . . .	70
Figura 20 – Modelo lógico do Banco de dados. . . . .	71



# Lista de tabelas

Tabela 1 – Cronologia Html.Fonte: W3schools: adaptada pelo autor. . . . .	40
Tabela 2 – Product Backlog . . . . .	49
Tabela 3 – Sprint Banco de Dados . . . . .	50
Tabela 4 – Desenvolvimento do Módulo do Administrador. . . . .	50
Tabela 5 – Desenvolvimento do Módulo do Empresa. . . . .	51
Tabela 6 – Sprint Desenvolvimento do Módulo do Cliente . . . . .	51
Tabela 7 – Sprint Integração de Módulos . . . . .	51



# Lista de abreviaturas e siglas

CRUD	Create Read Update e Delete
MVC	Model Viewer Controller
PHP	HyperText PreProcessor
CSS	Cascading Style Sheet
HTML	Hypertext Markup Language
XHTML	Extensible Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language
MYSQL	My Structured Query Language
SGBD	Sistema gerenciador de banco de dados
RDBMS	Relational Database Management System
RF	Requisitos Funcionais
RNF	Requisitos não Funcionais
API	Application Programming Interface
ERP	Enterprise Resource Planning



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>23</b>
1.1	Objetivos	24
1.2	Objetivos Gerais	24
1.3	Objetivos específicos	24
1.4	Justificativa	25
1.5	Organização da Dissertação	25
<b>2</b>	<b>REFERENCIAIS TEÓRICOS</b>	<b>27</b>
2.1	Planejamento dos Recursos da Empresa ERP	28
2.2	Engenharia de <i>Software</i>	29
2.3	Ciclo de Vida de Desenvolvimento	29
2.3.1	Modelo Cascata	29
2.3.2	Modelo Incremental	30
2.3.3	Prototipação	30
2.4	<b>Métodologia Ágil SCRUM</b>	<b>31</b>
2.4.1	Funcionamento do SCRUM	31
2.5	<b>Model Viewer Controller (MVC)</b>	<b>32</b>
2.6	<b>Programação Orientada a Objetos</b>	<b>33</b>
2.6.1	Objetos	34
2.6.2	Classes	34
2.6.3	Encapsulamento	34
2.6.4	Herança	35
2.6.5	Polimorfismo	35
2.6.6	Métodos	35
2.6.7	Atributos	35
2.7	<b>Sistema gerenciador de banco de dados</b>	<b>36</b>
2.8	<b>Linguagem de Modelagem Unificada (UML)</b>	<b>37</b>
2.8.1	Casos de Uso	37
2.9	<b>Teste de <i>Software</i></b>	<b>37</b>
2.9.1	Teste Caixa Branca	37
2.9.2	Teste Caixa Preta	37
<b>3</b>	<b>FERRAMENTAS UTILIZADAS</b>	<b>39</b>
3.1	<b>Hyper Text Markup Language (HTML)</b>	<b>39</b>
3.2	<b>Cascading Style Sheet CSS 3.0</b>	<b>40</b>
3.3	<b>Javascript</b>	<b>40</b>

3.4	Hipertext PreProcessor PHP . . . . .	41
3.5	Bootstrap 3.0 . . . . .	42
3.6	jQuery . . . . .	42
3.7	Ajax . . . . .	43
3.8	Adobe <i>Dreamweaver</i> . . . . .	43
4	<b>SISTEMA DESENVOLVIDO . . . . .</b>	<b>45</b>
4.1	Necessidades e problemas do proprietário do restaurante . . . . .	45
4.2	Marketing limitado . . . . .	45
4.3	Requisitos funcionais e não funcionais do sistema . . . . .	46
4.4	Caso de uso do Administrador . . . . .	47
4.5	Caso de uso do cliente . . . . .	48
4.6	Product Backlog . . . . .	49
4.6.1	Sprint Backlogs . . . . .	50
4.6.2	Sprint 1 – Desenvolvimento do Banco de Dados . . . . .	50
4.6.3	Sprint 2 –Desenvolvimento do Módulo do Administrador . . . . .	50
4.6.4	Sprint 3 –Desenvolvimento do Módulo da Empresa . . . . .	51
4.6.5	Sprint 4 – Desenvolvimento do Módulo do Cliente . . . . .	51
4.6.6	Sprint 5 – Integração entre Módulos . . . . .	51
5	<b>TESTES REALIZADOS . . . . .</b>	<b>53</b>
5.1	Verificação Validação e Testes . . . . .	53
5.1.1	Realização dos testes . . . . .	53
6	<b>CONCLUSÕES . . . . .</b>	<b>55</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>57</b>
	<b>APÊNDICES . . . . .</b>	<b>61</b>
	<b>APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO DE QUALIDADE . . . . .</b>	<b>63</b>
	<b>APÊNDICE B – QUADRO DE TESTES . . . . .</b>	<b>65</b>
	<b>ANEXO A – ANEXO DOS PRINTS DO SISTEMA . . . . .</b>	<b>67</b>
	<b>ANEXO B – MODELO LÓGICO BANCO DE DADOS . . . . .</b>	<b>71</b>



# 1 Introdução

Os sistemas de informação estão presentes atualmente ao redor do mundo como uma tecnologia única em torno da qual orbitam outras tecnologias (PRESSMAN, 2006). Essa tecnologia única se relaciona diretamente a outras tecnologias e seu avanço tende a tornar avanços em outras tecnologias dependentes possíveis. PRESSMAN (2006, p. -1) se refere a esse fenômeno como “lei das consequências não pretendidas” e destaca que o software de computadores assumiu um papel tão importante que seria impossível de ser previsto por seus criadores na década de 50. A evolução tecnológica que ocorreu após a década de 40 tornou possível a criação de tecnologias que possibilitaram enormes avanços tecnológicos e podemos dizer que o avanço dos sistemas computacionais tornaram muitas dessas invenções possíveis (FILHO, 1994).

Esse avanço é percebido em tecnologias avançadas, mas pode e deve também ser usado pelo público em geral, de forma a facilitar diversas tarefas do cotidiano. Facilitar a vida das pessoas executando tarefas repetitivas é uma das finalidades dos sistemas de informação, mas nem todas as pessoas conseguem ter essa percepção (FILHO, 1994). Esse avanço é perceptível nos equipamentos que são usados no cotidiano, como *smartphones*, computadores pessoais, *tablets* e automóveis, por exemplo, mas alguns setores do comércio ainda não se beneficiaram de tais avanços. Pequenos comerciantes ainda não se sensibilizaram dos benefícios que seu negócio pode ter com o uso de um bom sistema de informação, outros simplesmente não tem acesso a esse tipo de serviço devido ao alto custo de um sistema desenvolvido especificamente para o seu negócio (FEITOSA; GARCIA, 2016).

O setor alimentício é um dos setores que apresenta baixo uso de sistemas de informação para a gestão do negócio. Essa informação pode ser ainda mais negativa considerando que a falta do uso de tecnologias prejudica a conquista e manutenção de uma posição vantajosa por uma empresa. Os pequenos negócios, que não dispõem de recursos financeiros para desfrutar das tecnologias disponíveis têm a sua competitividade prejudicada, pois o uso de tais tecnologias propiciam a oferta de produtos mais baratos e seus processos mais eficientes e eficazes (MARTINS; FERNANDES, 2016; PEREIRA; VIEIRA; FONSECA, 2016).

O compartilhamento de recursos por empresas é uma oportunidade para os pequenos negócios de ter acesso a ferramentas estratégicas, como sistemas de computação a um custo bastante reduzido. Esse modelo de negócio também é uma oportunidade a ser explorada pelos desenvolvedores, que podem aumentar seu número de clientes, oferecendo um serviço que seja acessível, barato e que traga benefícios para os clientes (FEITOSA; GARCIA, 2016). Sistemas como esse operam geralmente na internet, uma vez que facilita o acesso e

a portabilidade desses sistemas. Além disso, a demanda por Internet e um espaço *online* para comunicação entre as pessoas é um negócio rentável para alguns e divertido para todos e o potencial que a *Web* pode alcançar é indeterminado (FILHO, 2003).

O trabalho tem como ideia principal implementar uma plataforma *Web* para agregar vários restaurantes que poderão utilizar o sistema tanto para gerir seu negócio quanto para atender seus clientes através dos serviços disponibilizados. Esse sistema deve apresentar algumas características específicas como boa usabilidade e entendimento de suas funcionalidades, de maneira que possa ser utilizado por um público heterogêneo, trazendo benefícios tanto para os restaurantes, quanto para os clientes.

## 1.1 Objetivos

### 1.2 Objetivos Gerais

Realizar a implementação de um sistema de gestão para empresas no ramo de alimentação, disponibilizando a ferramenta para uso coletivo, mas específico para cada empresa do ramo, a priori foi desenvolvido para atingir os restaurantes.

### 1.3 Objetivos específicos

A fim da realização de estudos, foi colocado a seguinte questão principal de pesquisa:

Como obter um atendimento mais rápido ao se realizar um pedido em um estabelecimento de entregas por demanda utilizando um sistema de informação?

Para definição dos objetivos específicos nesse trabalho foi levado em consideração a forma de atendimento dos estabelecimentos que realizam o serviço de tele-entrega de marmitas, ou atendam por demandas de pedidos mensais, levando em consideração, destacam-se como objetivos específicos:

- Gestão financeira.
- Gestão de clientes.
- Gestão de pedidos.
- Desenvolvimento de um sistema Web.
- Interação do usuário e estilos de interação.

## 1.4 Justificativa

Percebeu-se a necessidade de um sistema de gestão empresarial, visto que a maioria das pessoas só utilizam-se de métodos tradicionais de controle financeiro para registrar as contas e comunicar-se com os clientes, ou não utilizam de nenhuma tecnologia para facilitar o controle e conhecimento das pessoas e do mercado de trabalho que estão trabalhando.

Tendo em vista que o atendimento ao cliente deve ser mais eficaz, o uso da Internet e o mundo digital estão cada vez mais evoluídos para que realizem esse tipo de serviço, criou-se a ideia para desenvolvimento e implementação da plataforma de multi usuários aderindo a todos que não possuem essa tecnologia, gerando um maior conforto nas operações cotidianas das empresas.

## 1.5 Organização da Dissertação

Além da introdução a dissertação compreende dos seguintes capítulos:

Capítulo 2: Referencial teórico: Consiste em realizar uma revisão dos trabalhos já existentes sobre o tema abordado, que pode ser em livros, teses, monografias, enciclopédias, artigos, filmes, mídias eletrônicas e outros materiais cientificamente confiáveis.

Capítulo 3: Consiste em apresentar as linguagens de programação as interfaces dos Usuário as ferramentas Web para desenvolvimento do sistema.

Capítulo 4: Projeto de *Software*: Serão discutidas como foram feitas as principais implementações desse trabalho, como foi desenvolvido para realizar o gerenciamento de um restaurante On-line.

Capítulo 5: Resultados e testes: Apresentação das ferramentas de *Hardware* utilizadas para desenvolvimento do sistema, as telas do sistema e testes aplicados.

Capítulo 6: Considerações finais à cerca do trabalho e propostas para trabalhos futuros.

Capítulo 7:Referências bibliográficas.



## 2 Referenciais Teóricos

A palavra sistema pode ser definida como um conjunto de subsistemas que se comunicam interagem entre si com um objetivo em comum. De acordo com [Bertalanffy \(1975\)](#):

Sistema é uma entidade que tem a capacidade de manter um certo grau de organização em face de mudanças internas ou externas, composto de um conjunto de elementos, em interação, segundo determinadas leis, para atingir um objetivo específico.

Os dados podem ser definidos como símbolos e fatos brutos que representam os eventos que ocorrem em uma determinada organização. Por sua vez, a informação consiste em dados que são apresentados de forma que permitem a utilização e interpretação por seres humanos ([LAUDON; LAUDON, 2005](#), p. -4).

A informação é algo muito importante nos dias de hoje, podendo ser processada armazenada em diversas formas e tamanhos e em vários dispositivos diferentes. Antigamente era muito difícil obter informação a respeito de determinado lugar, ou sobre alguma pessoa, a comunicação era lenta, as tecnologias voltadas para comunicação não eram acessíveis a todos, as pessoas eram mais desinformadas com questões de Internet e sistemas Web pois não havia acesso as informações igual é disponibilizada nos dias de hoje por esses veículos de comunicação digital.

Podemos dizer que a informação é o dado mais trabalhado podendo ser usado para tomar decisões. A informação é algo valioso, quando reunida e classificada se tornam dados se transformam em conhecimento e este é um valor que não tem como determinar o preço. Segundo [RODRIGUES \(1999\)](#):

Pode-se dizer que, a partir de 1985, a informação passou a ser utilizada, mais orientadamente, como recurso estratégico. A diferença entre dado e informação pode se dizer, pelo motivo do dado, ser um elemento que mantém a sua forma bruta (sons, imagens, vídeos, texto), ou seja, ele sozinho não levará a compreender determinada situação.

A informação seria um tipo de dado que apoia a tomada de decisão, o dado seria qualquer elemento menor que em sua forma primária, sem associação com outros dados, ou estruturação e não fornece informações sobre o sistema.

Os sistemas de informação são ferramentas capazes de fazer com que esses dados sejam convertidos e as informações sejam tratadas de uma forma mais segura, informações com maior transparência aos usuários ([OLIVEIRA, 2007](#)).

Temos também a engenharia de *Software* aplicada nos sistemas, , define a Engenharia de *Software* como aplicação sistemática, disciplinada com abordagem quantitativa para desenvolvimento, operações e manutenções de *Software*, demonstrando que para conseguir desenvolver um sistema não é simplesmente codificar, são várias as etapas para se alcançar o produto final de forma que haja qualidade na concepção do *Software*.

De acordo com (REZENDE, D.A.; ABREU, 2000, p.-97):

A informação nos dias de hoje tem um valor altamente significativo e pode representar poder para quem a possui, seja pessoa, seja instituição. Ela possui seu valor, pois está presente em todas as atividades que envolvem pessoas, processos, sistemas, tecnologias, etc. Toda informação que é transmitida pela internet, é regida pela política de segurança de informação, para cada tipo de ambiente Web em que se utilizam os dados de usuários para realização de serviços Web.

Os softwares ou programas de computador são conjuntos de instruções específicas escritas por um programador, que fazem a interface de operação entre o usuário que opera a máquina e o hardware, ou seja, dispositivos físicos que compõem o computador (PAUL; DEITEL, 2010).

As instruções são escritas em forma de texto que, depois de construído ou compilado, ao ser executado, deve atender de forma satisfatória às demandas daqueles que usam o programa. De maneira diferente de outras criações humanas, o sendo um sistema lógico, não possui propriedades físicas (PRESSMAN, 2006).

O código fonte é constituído pelos comandos escritos pelos programadores e podem ser usados para gerar novos programas executáveis. Devido a essas características os *Software* podem ser facilmente entregues ao cliente sob a forma de um arquivo digital, que é executado, ou apenas na forma de código que será usado para compilar um novo *Software*. Por ser um processo bastante complexo, é preciso utilizar alguma forma de estruturação das equipes de desenvolvimentos, de forma a assegurar um trabalho uniforme, boa comunicação e ainda garantia a qualidade (FILHO, 2003).

## 2.1 Planejamento dos Recursos da Empresa ERP

O Planejamento dos Recursos da Empresa (*Enterprise Resource Planning, o ERP*) são *Software* responsáveis por auxiliar a gestão empresarial em pontos estratégicos da empresa, aplicando novas tecnologias as necessidades rotineiras da empresa. Existem diversos sistemas ERP que auxiliam nas ações do fluxo de caixa, apuração de Impostos, Gestão de Pessoas, assim como na organização de inventário de estoque, controle de ponto dos funcionários, controle do maquinário da fábrica. trabalho administrativo e operacional realizados pela empresa (ERP, 2016).

## 2.2 Engenharia de *Software*

A Engenharia de *Software* é um processo utilizado para o desenvolvimento e modelagem de *Software*, que tem como alvo principal a descrição detalhada dos processos utilizados na concepção de *Software*. Ela pode ser definida um processo de concepção de *Software* dividido em etapas que, quando integradas, formam o *Software*. Cada modelo de processo representa um processo a partir de uma perspectiva particular, de uma maneira que proporciona apenas informações parciais sobre o processo (SOMMERVILLE, 2007; FILHO, 2003).

## 2.3 Ciclo de Vida de Desenvolvimento

O ciclo de vida de um *Software* pode variar bastante, mas durante toda sua existência ele nunca estará definitivamente pronto, embora possa estar pronto para o uso. Nesse caso, durante toda a sua existência o *Software* receberá manutenção e/ou melhorias (REZENDE, 2006).

A modelagem de processos de *Software* alcançou sua fase de maturidade com base no conhecimento e experiência adquirida na produção de grande escala de projetos. Existem vários modelos de ciclo de vida para desenvolvimento de *Software*. Dentre eles os mais conhecidos são os modelos em cascata, incremental, prototipação e espiral (PETERS; PEDRYCZ, 2000; SOMMERVILLE, 2007; FILHO, 2003).

### 2.3.1 Modelo Cascata

O modelo cascata é o mais conhecido como modelo de ciclo de vida de *Software*. Ele tem essas cinco fases denominada de uma forma diferente, como mostra a Figura 1 (SOMMERVILLE, 2007).

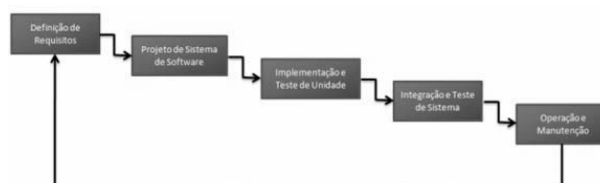


Figura 1 – Modelo cascata fonte, Sommerville (2007)

O processo se inicia a partir da definição de requisitos, nesta etapa do modelo cascata serão definidos quais são os requisitos do sistema, ou seja o que o sistema deverá possuir de funcionalidade, qual o objetivo quer chegar com esses requisitos. A segunda etapa é o projeto do sistema de *Software* onde definimos a arquitetura, tanto no aspecto *Software*, com os requisitos funcionais, quanto no aspecto hardware, os requisitos não funcionais. Um projeto bem feito dependerá da coleta de requisitos feita anteriormente

com qualidade para gerar um projeto de qualidade. A terceira etapa do processo envolve a implementação e teste de unidade sendo que o próprio nome já diz, é a etapa em que se implementa o *Software* realizando teste de caixa branca e caixa preta, para ver se esta atendendo a sua finalidade. A quarta etapa é a parte do suporte técnico vendido ao cliente onde se corrigem erros não encontrados na etapa anterior do processo de concepção do sistema (SOMMERVILLE, 2007).

### 2.3.2 Modelo Incremental

O modelo incremental foi desenvolvido pensando em melhorar o modelo cascata neste modelo os requisitos e especificações são primeiramente identificados, as demais etapas são executadas em cada momento que é necessário o lançamento de uma nova versão do *Software*. A figura abaixo demonstra o processo incremental (PETERS; PEDRYCZ, 2000).

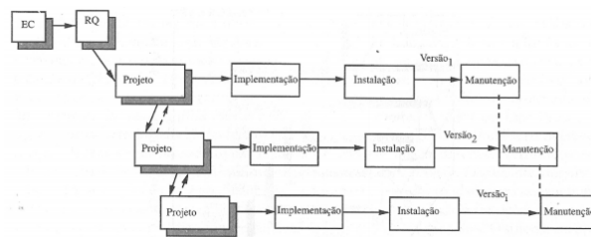


Figura 2 – Modelo incremental Fonte: Peters(2000, p.42)

### 2.3.3 Prototipação

Este modelo de prototipação surgiu com o intuito de possibilitar um desenvolvimento rápido de produtos de *Software*. A prototipação descreve a situação de estudo mais clara para o cliente, quando não tem precisão na descrição do problema. Quando o cliente define apenas um conjunto de objetivos gerais para o *Software*, mas não identifica requisitos de entrada ou saídas bem detalhadas, a prototipação transcreve perfeitamente o cenário servindo também para mostrar ao cliente como ficará o *Software* desejado com base nos requisitos e objetivos (PETERS; PEDRYCZ, 2000).

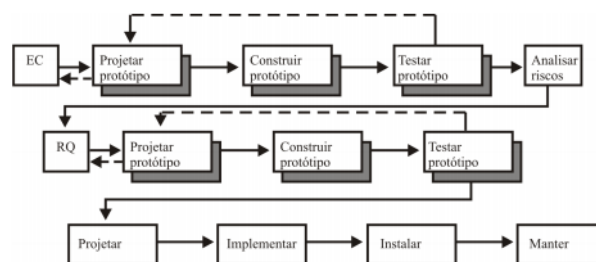


Figura 3 – Modelo prototipação Fonte: Peters(2000, p.42)



## 2.4 Metodologia Ágil SCRUM

O SCRUM é um Framework que está estruturado de tal maneira que ele suporta produtos e o desenvolvimento de serviços em todos os tipos de indústrias e em qualquer tipo de obra, independentemente da sua complexidade (SCRUMSTUDY, 2013).

Um dos pontos fortes para Scrum reside nele uso de equipes auto-organizadas e capacitadas multifuncionais que dividem o seu trabalho em ciclos de trabalho curto concentrado Sprints called (SCRUMSTUDY, 2013).

O processo produz incrementos contínuos no *Software* que podem ser inspecionados, testados, documentados e expandidos, pois os testes e documentação são realizados à medida que o *Software* é construído. Dessa forma o cliente fica mais próximo do processo de desenvolvimento e as equipes podem reagir melhor a cada nova mudança proposta (PRESSMAN, 2006).

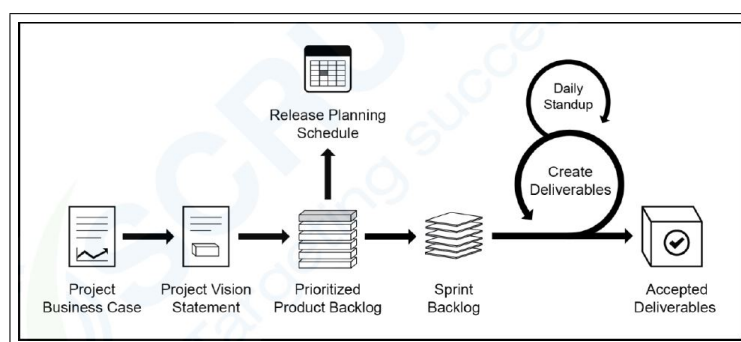


Figura 4 – Scrum ciclo único de trabalho Fonte: Sbok Guide (2013, p.8)

O ciclo de etapas do SCRUM começa na primeira reunião com os Stakeholders, os interessados no projeto, e durante essa reunião é criado o Project Vision Statement. O proprietário do produto desenvolve as ideias junto com a equipe que vai classificando a importância das ações de negócios que devem ocorrer de acordo com as necessidades dos usuários, e assim cria-se o Product Backlog. Cada ação é chamada de Sprint, o primeiro Sprint realizado é para planejar as reuniões e definir os pontos de maior prioridade para dar início as atividades da equipe (SCRUMSTUDY, 2013; SCHWABER, 2004).

### 2.4.1 Funcionamento do SCRUM

O SCRUM baseia-se em diferentes papéis, de acordo com a participação do indivíduo no processo. Entender e definir os papéis e responsabilidades no Scrum é muito importante para garantir o sucesso da implementação do Scrum, os papéis do SCRUM podem se enquadrar em duas categorias: core rules e non-core rules (SCRUMSTUDY, 2013; SCHWABER, 2004).

**Core Roles:** São os papéis que são obrigatórios para requisição de produção de produtos ou serviços. Quem assume essa responsabilidade está totalmente envolvido para que a iteração do projeto seja concluída com sucesso como um todo. Estão incluídos nessas Regras os papéis:

- **Product Owner:** É a pessoa responsável por agregar o máximo de credibilidade nos valores do negócio, representa a voz do cliente.
- **Scrum master:** É o facilitador que garante que a equipe que está realizando o Scrum consiga prover uma condução do ambiente para que seja completado o projeto com sucesso.
- **Scrum Team:** É a equipe constituída entre 6 a 10 pessoas que são responsáveis por entender os aspectos levantados em reuniões pelo Product Owner, produzir o produto ou serviço, e entregar o projeto.

**Non-core Roles:** São os papéis que não são obrigatoriamente exigidos pelo projeto Scrum, podendo incluir membros que estão interessados no projeto. Eles possuem um projeto não formal no time do projeto e podem interagir com a equipe, não sendo responsáveis pelo sucesso do projeto. Estão incluídos nessas Regras os papéis:

- **Stakeholder(s):** É um termo coletivo que inclui os clientes, usuários e patrocinadores, que frequentemente se interagem com a equipe principal do Scrum, e influenciam o projeto durante sua etapa de desenvolvimento.
- **Chief Product Owner:** É um papel importante quando se trata de várias equipes de Scrum, responsável por facilitar o trabalho, manutenção e justificativas de negócios criadas pelo projeto.
- **Chief Scrum Master:** Responsável por coordenar as atividades relatadas do Scrum em larga escala, necessitando de múltiplas equipes que trabalham em paralelo.

## 2.5 Model Viewer Controller (MVC)

O modelo de padrão arquitetural Model Viewer Controller (MVC), é uma forma de quebrar uma aplicação, ou até mesmo um pedaço de uma aplicação, em três partes: o model(modelo), viewer (visão), e controller (controlador) (KRASNER; POPE et al., 1988).

O controlador, é o interpretador das funcionalidades requisitadas pelo usuário ao se utilizar os periféricos do computador o teclado e mouse, essas ações requisitadas são enviadas para o modelo seguindo para a visualização das alterações desejadas (DEV MEDIA, 2016).

O modelo gerencia um ou mais elementos de dados, respondendo as requisições de mudança de estado. O modelo sabe o que a aplicação quer realizar, pois ele sabe quem deve fazer cada ação.

A visão é a parte visual que chega ao usuário, utilizando de linguagens de programações Web para sua alta performance e maior usabilidade das funcionalidades. A visão não sabe interpretar o que acontece, simplesmente exhibe os dados que são passados do modelo e do controle.

Destaca-se que o padrão arquitetural MVC, tem como característica a divisão dos processos e dos códigos, facilitando uma reutilização de código. Pode ser comparada a uma programação orientada a objetos devido sua forma de divisão em classes, e herança de códigos que definem estruturalmente os processos e como eles se interagem e são exibidos na tela para utilização do sistema.

Existem frameworks que aderem ao padrão arquitetural MVC, destacando-se: JSF, Struts 1 e Struts 2, Spring MVC, Play Framework, Tapestry.

## 2.6 Programação Orientada a Objetos

Um modelo de programação ou paradigma de programação é um conjunto de princípios, ideias, conceitos e abstrações utilizadas para o desenvolvimento de uma aplicação. Devemos escolher o modelo de programação mais propício para desenvolvermos as necessidades das aplicações propostas. Podemos ter como exemplo a figura abaixo:([K19 Treinamentos, 2013](#)).



Figura 5 – Orientação a objetos Fonte: Apostila k19 (2013)

Os modelos de programação orientados objetos possuem vantagens do desenvolvimento do *Software* descrevendo quais os tipos de sistemas que se beneficiam com esse desenvolvimento.([K19 Treinamentos, 2013](#)).

1. Sistemas com grande quantidade de funcionalidades desenvolvidos por uma equipe.
2. Sistemas que serão utilizados por um longo período de tempo e sofrerão alterações constantes.

### 2.6.1 Objetos

Objetos são representações das entidades do mundo real, para uma linguagem de programação, será a forma de descrever a sua forma “digital”, objeto é criado (instanciado) a partir de uma classe, para que seja possível utilizar-se dos atributos e métodos especificados na classe que representa o objeto (TONSIG, SERGIO LUIZ, 2007). Alguns exemplos de entidades:

- **Entidade Física:** Bicicleta, prédios, skate, carro, caminhão e etc.
- **Entidade Conceitual:** Nome, processo químico, matrícula, endereço, sexo, e etc.
- **Entidade de *Software*:** Lista encadeada, arquivo, classes, objetos, métodos e etc.

O próprio objeto deve realizar operações de consulta ou alteração dos valores de seus atributos. Essas operações são definidas nos métodos do objeto.(TONSIG, SERGIO LUIZ, 2007).

### 2.6.2 Classes

As classes são modelos estruturais que possuem características semelhantes, que podem ser modeladas e especificadas em uma estrutura. Tomando como exemplo uma classe “funcionários”, esta terá as características gerais de um funcionário. Cada funcionário é uma instância (objeto) da classe “funcionários” (PRESSMAN, 2006).

As classes são estruturas de objetos, essa estrutura tem como característica os atributos, ou seja, os atributos não definem o comportamento da classe, definem sua estrutura. De acordo com Santos (2003), Classe é um modelo e todos os seus objetos têm os mesmos atributos (embora esses atributos possam ter valores diferentes) e os mesmos métodos. As classes são estruturas de objetos, essa estrutura tem como característica os atributos, ou seja, os atributos não definem o comportamento da classe, definem sua estrutura. Em resumo, Classe é um modelo e todos os seus objetos têm os mesmos atributos, embora esses atributos possam ter valores diferentes e os mesmos métodos.

### 2.6.3 Encapsulamento

A Programação orientadas objetos utiliza-se desta “habilidade” para encapsular(esconder) o código, a lógica de programação, escondem o segredo, um exemplo seria o controle remoto de um rádio, é capaz de realizar ações a distância e os botões realizam ações independentes que haja uma falha em algum botão não compromete a funcionalidade dos outros botões. O conceito que abrange à liberação ou restrição de acesso aos atributos e métodos de uma classe, utilizando a propriedade de visibilidade que emprega os chamados “modificadores de acesso” (private, project e public) (TONSIG, SERGIO LUIZ, 2007).

### 2.6.4 Herança

As linguagens de programação que utilizam a orientação a objetos possuem essa característica das classes capazes de ceder seus atributos e métodos para reutilização de códigos para criação de novas classes, assim economiza-se tempo e serviço. Tradicionalmente conhecida como relacionamento “é – um”. Quando ocorre esse relacionamento é denominado de “superclasse” a que define o comportamento, e - subclasse a que herda o relacionamento. As subclasses podem realizar tarefas como adicionar novas operações, redefinir operações existentes, e até mesmo remover uma operação (Dall’Oglio, Pablo, 2015).

Pode chamar de classe filha aquela que herda as características de outra (classe mãe). Herança é uma forma de reutilização de códigos. Por exemplo, que se tenha uma classe “pessoa” e também uma classe “funcionários”. Na classe “pessoa” se define alguns atributos como nome, endereço, telefone e alguns métodos como, por exemplo, cadastrar e consultar. Se a classe “funcionários” tiver como característica tudo que já foi definido para “pessoa”, então, a classe “funcionários” é uma sub-classe de “pessoa” (“pessoa” é a classe mãe). Herança, implica dizer que é tudo aquilo que foi definido na classe mãe está também disponível (se faz presente) na classe filha (TONSIG, SERGIO LUIZ, 2007).

### 2.6.5 Polimorfismo

Polimorfismo é o nome é dado para as funções que possuem operandos que podem ser tratados com vários tipos diferentes, uma linguagem fortemente tipada você precisa declarar qual o tipo de objeto a variável irá referenciar. É a Capacidade de respostas diferentes a uma mesma forma de chamada do método. Fato que pode ocorrer pela sobreposição do método”, isto é, a classe mãe possui uma definição e procedimento de um método que, na classe filha o mesmo não é utilizado tal como foi herdado, em função de seus procedimentos terem sido redefinidos (PRESSMAN, 2006).

### 2.6.6 Métodos

Os métodos são as funcionalidades e ações que o objeto pode oferecer. No caso de um objeto “funcionário”, por exemplo, algumas possíveis funcionalidades seriam: admissão, promoção, suspensão, alteração, demissão, afastamento (TONSIG, SERGIO LUIZ, 2007).

### 2.6.7 Atributos

Os atributos são as características próprias de um determinado objeto. Considere o objeto “pessoa”. Pessoa pode ter nome, endereço, telefone, altura, idade, religião, cor dos olhos etc. Todas estas características específicas de pessoa são chamadas de atributos (NETO, 2007).

## 2.7 Sistema gerenciador de banco de dados

Os programas desenvolvidos para computadores de hoje em dia estão todos utilizando dessa tecnologia para uma maior segurança e confiabilidade na hora de armazenar seus documentos .

Os sistemas gerenciador de banco de dados(SGBD), é uma coleção de grupos de informações que estão relacionadas entre si de alguma maneira. Os grupos lógicos de informações podem incluir categorias como dados do clientes, informações sobre um pedido e informações sobre os produtos (Date, Christopher J, 2004).

São os responsáveis por facilitar a configuração construindo e manipulando dados que serão armazenados nos banco de dados. Alguns dos sistemas gerenciadores de banco de dados são gratuitos e outros são pagos mais todos são eficientes e de grande valia o conhecimento para o profissional na área de sistemas :

1. Oracle<sup>1</sup>
2. iDB2<sup>2</sup>
3. SQL-Server<sup>3</sup>
4. Informix<sup>4</sup>
5. MYSQL.<sup>5</sup>
6. FirebirdSQL<sup>6</sup>

Neste trabalho abordaremos apenas o MYSQL. O MySQL está publicamente disponível desde 1996, mas tem uma história de desenvolvimento que remonta a 1979.

O MySQL é um sistema de gerenciamento de banco de dados relacional (relational database management system -RDBMS) poderoso e muito rápido. Um banco de dados permite armazenar, pesquisar, classificar e recuperar dados de forma eficiente. O servidor do MySQL controla o acesso aos dados para assegurar que vários usuários possam trabalhar com os dados ao mesmo tempo, fornecer acesso rápido aos dados e assegurar que somente usuários autorizados obtenham acesso. Portanto, o MySQL é um servidor multiusuário e multithreadado (ou multithreaded). Ele utiliza SQL (Structured Query Language), a linguagem de consulta padrão de banco de dados em todo o mundo.(LUKE; WELLING, 2005). O MySQL está disponível sob a licença Open Source, contanto que cumpra os termos de licença.

<sup>1</sup> <http://www.oracle.com/br/database/index.html>

<sup>2</sup> <http://www.ibm.com/analytics/us/en/technology/db2/>

<sup>3</sup> <http://www.microsoft.com/en-us/cloud-platform/sql-server>

<sup>4</sup> <http://www-01.ibm.com/software/data/informix/>

<sup>5</sup> <https://www.mysql.com/>

<sup>6</sup> <http://firebirdsql.org/>

## 2.8 Linguagem de Modelagem Unificada (UML)

### 2.8.1 Casos de Uso

Os casos de usos são escritos pela linguagem UML (Unified Modeling Language ou Linguagem de Modelagem Unificada), é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de orientação a Objetos. (GUEDES, 2011).

No diagrama de caso de uso está representando os requisitos do sistema referindo-se, as tarefas ou funcionalidades que foram identificados dentro da análise de requisitos que podem ser necessários no desenvolvimento do *Software*. (GUEDES, 2011).

Para facilitar a escrita dos casos de uso é importante observar detalhadamente o cenário. Pode-se pedir para que o usuário descreva alguns cenários na forma de exemplos situados em casos de uso de cada ação, mostrando o ator usando o sistema para realizar o caso de uso em questão. (COCKBURN, 2005).

## 2.9 Teste de *Software*

Testes de software tem a função de avaliar e validar o sistema desenvolvido, assegurando ao desenvolvedor tanto que suas funções, realizem as tarefas que precisam ser executadas, quanto para validar o sistema do ponto de vista do cliente (PRESSMAN; TRAVIESO, 2002; FILHO, 2003; SOMMERVILLE, 2007). Partindo-se do princípio que o sistema é baseado em código e é esse código que estaremos testando, podemos dividir os testes em:

### 2.9.1 Teste Caixa Branca

Técnica utilizada para trabalhar diretamente sobre o código fonte do módulo de *Software* em teste, para avaliar aspectos tais como: teste de fluxo de dados, teste de caminhos lógicos, teste de condição e teste de ciclos. (SOMMERVILLE, 2007).

### 2.9.2 Teste Caixa Preta

Para a realização dessa técnica, não é necessário que o testador conheça o conteúdo do código fonte. Basta que seja informado os valores esperados de entrada e saída de uma função. Com essa técnica é possível realizar testes do tipo (SOMMERVILLE, 2007; PRESSMAN, 2006) :

- **Teste unitário:** funções que testam outras funções primárias do sistema de forma automática, informando uma entrada e avaliando o que é retornado.

- **Teste de integração:** avalia se os módulos do sistema estão bem integrados.
- **Teste de sistema:** avalia a integração do sistema como um todo.
- **Teste de aceitação:** se as funções primárias do sistema estão sendo atendidas e se as mesmas atendem às necessidades do usuário.



## 3 Ferramentas Utilizadas

Neste trabalho foram utilizadas no desenvolvimento do *Software* as seguintes linguagens de programação: HTML5; CSS 3; JavaScript; PHP 5; e MySQL. Foram também utilizados os frameworks: Bootstrap 3.0 e jQuery 1.11. Para a engenharia de *Software* do projeto foi utilizado a metodologia Scrum. O arquitetura de projeto (design pattern), o Model Viewer Controller (MVC) foi utilizado para facilitar a escrita, manutenção, aprimoramento e melhorias dos códigos.

### 3.1 Hyper Text Markup Language (HTML)

O HTML<sup>1</sup> significa linguagem de marcação de texto, baseada em tags, que descrevem o comportamento das páginas da internet, são essas 'marcações' ou tags que realizam as interações vistas pelos usuários que estão navegando na internet. O HTML atual está em sua versão 5.0 com várias novidades e inovações que facilitam o uso das tags para um melhor desenvolvimento de aplicações, levando em conta a maior compatibilidade entre os *browser*.

O HTML 'é um conjunto de tags de marcação os documentos HTML são descritos por *tags* HTML Cada *tag* HTML descreve o conteúdo do documento diferente, cada uma com sua funcionalidade'. A seguir apresentação das principais *tags* do HTML (W3SCHOOLS, 2016).

- **HTML** – `<html> </html>`: *Tag* é responsável por definir o início e fim de um documento HTML. Logo deve ser aberta no início do documento e fechado no seu final. Todas as demais tags devem ser escritas dentro da tag.
- **HEAD** – `<head></head>`: *textit*Tag que define o início e o fim do cabeçalho do documento.
- **BODY** – `<body></body>`: *textit*Tag onde devem ser inseridos os conteúdos do site que deverão ficar visíveis ao usuários quando a página for renderizada no navegador. Somente com essas três tags já podemos ter uma estrutura HTML básica. Todas as linguagens descritas são utilizadas em conjunto para criação de elementos WEB.

A tabela 1 demonstra cronologia do HTML.

---

<sup>1</sup> <http://www.w3.org/html/>

Ano	Versão
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML	2014

Tabela 1 – Cronologia Html. Fonte: W3schools: adaptada pelo autor.

## 3.2 Cascading Style Sheet CSS 3.0

O CSS (Cascading Style Sheets)<sup>2</sup> é uma linguagem de estilos utilizada para definir a apresentação de documentos. Ao invés de colocar a formatação dentro do documento, o desenvolvedor cria um link para um arquivo que contém os estilos, procedendo de forma idêntica para todas as páginas do site. Quando o programador desejar alterar a aparência do site basta modificar apenas o arquivo com o link respectivo. Segundo DAMIANI (2006) “O CSS veio para botar cada coisa no seu lugar por ser uma linguagem própria de formatação.”

As unidades de medidas utilizadas são (ZEMEL, 2014) :

- **Ems (em):** O em representa uma unidade escalável. A unidade 1em representa o tamanho real do elemento pai. Por exemplo, uma fonte 12pt representa 1em. O em é escalável, assim 2em seria igual 24pt ou 0.75em igual a 8pt.
- **Pixel (px):** Esta é a unidade fixa mais utilizada no CSS. O pixel é um ponto indivisível para a tela de exibição. Os desenvolvedores costumam utilizar esta unidade pelo fato de ser considerada de pixel perfect, sendo assim, um layout pode ser representado identicamente ao layout do projeto.
- **Ponto (point):** Geralmente utilizados em CSS de impressão. Um ponto representa 1/72 polegadas e assim como pixel também são indivisíveis.
- **Porcentagem(%):** (Esta unidade é muito parecido com em porém existem algumas diferenças como, 12pt representa 100(por cento) na unidade de porcentagem. Assim, em dispositivos móveis o texto consegue ser escalável.

## 3.3 Javascript

Javascript<sup>3</sup> é uma linguagem de programação, interpretada e com recursos de orientação a objetos. O Javascript é executado do lado do cliente e permite que algum

<sup>2</sup> <https://www.w3.org/Style/CSS/>

<sup>3</sup> <https://www.javascript.com/>

conteúdo executável seja incluído em páginas Web. Isso permite a uma página Web não ficar restrita ao HTML estático, além disso, é possível desenvolver programas que interajam com o usuário, controlem o navegador e criem conteúdo HTML dinamicamente (FLANAGAN, 2004).

Sintaticamente, o JavaScript assemelha-se às linguagens C, C++ e Java, contendo elementos como: instrução IF, laço while e os operadores. Entretanto, o JavaScript é uma linguagem não-tipada, ou seja, as variáveis não precisam ter um tipo especificado.

Um dos equívocos mais comuns que se comete sobre JavaScript é sugerir que seja o Java, a linguagem de programação da Sun Microsystems. Exceto por alguma semelhança sintática e pelo fato de que tanto Java como JavaScript podem fornecer conteúdo executável em navegadores Web, as duas linguagens não tem nenhuma relação. Segundo Flanagan (2004), “A semelhança de nomes é puramente uma manobra de marketing, no início a linguagem foi chamada de LiveScript; seu nome mudou para Javascript no último minuto.”

JavaScript e Java formam, porém, uma boa equipe. As duas linguagens possuem diferentes recursos. JavaScript consegue controlar o comportamento e o conteúdo de um navegador, mas não consegue desempenhar operações de rede. Java não tem nenhum controle sobre o navegador como um todo, mas oferece recursos gráficos e recursos de rede. JavaScript do lado cliente é capaz de interagir e controlar applets Java embutidos em páginas Web. Nesse sentido, JavaScript pode realmente empregar Java em Scripts (FLANAGAN, 2004).

## 3.4 Hipertext PreProcessor PHP

O PHP (Hipertext PreProcessor)<sup>4</sup> é uma linguagem de programação utilizada para criar sites dinâmicos. Aplicações em PHP são geradas em excelente performance, e automaticamente, pelo servidor. O usuário não tem acesso ao código PHP. Esse tipo de detalhe é muito importante quando estamos lidando, por exemplo, com senhas e acesso a banco de dados (NIEDERAUER, JULIANO, 2007).

O PHP é uma linguagem de script, executada no lado do servidor. Podemos coletar dados de formulários, gerar páginas com conteúdo dinâmico, enviar e receber dados, gerar xml a partir de dados coletados em banco de dados entre outras funções. Scripts no lado do servidor (server-side) é a principal área de atuação do PHP. Para implementá-lo, precisamos apenas de um interpretador PHP, um servidor Web e um browser. Basta então rodar o servidor Web conectado a um PHP instalado. É possível acessar os resultados do programa PHP com um browser, visualizando a página PHP por meio de um servidor Web.

---

<sup>4</sup> <http://php.net/>

O PHP pode ser utilizado na maioria dos sistemas operacionais, como Linux, variantes Unix (Solaris, OpenBSD), Microsoft Windows, etc. Ele também é suportado pela maioria dos servidores Web atuais, incluindo Apache, Microsoft Internet Information Server (IIS), Personal Web Server e muitos outros (MORAZ, 2005).

Com o PHP não estamos limitados a gerar HTML. As funcionalidades do PHP incluem gerações de imagem, arquivos PDF e animações em Flash criadas dinamicamente. Podemos, ainda, gerar padrões de texto, como XML, e envia-los via HTTP para o usuário. Com o PHP é possível se conectar numa variedade de banco de dados. Entre os bancos suportados pelo PHP, destacam-se MySQL, Oracle (OCI7 e OCI8), InterBase e PostgreSQL.

Para Luke e Welling (2005): “O Protocolo HTTP não tem nenhuma maneira predefinida de manter o estado entre duas transações”, ou seja, o HTTP não fornece uma maneira de dizer se as duas solicitações vêm do mesmo usuário. Com isso a segurança da informação fica comprometida, podendo disponibilizar dados indevidos a um determinado usuário do sistema. Para resolver o problema de segurança na transação cliente/servidor o PHP utiliza o controle de sessão, onde é gerado um ID único de sessão, um número criptograficamente aleatório. Esse ID de sessão é gerado pelo PHP e armazenado no lado cliente. A idéia do controle de sessão é monitorar um usuário durante uma única sessão em um Web site. Podendo facilmente suportar o login de um usuário e mostrar o conteúdo de acordo com seu nível de autorização.

### 3.5 Bootstrap 3.0

O *Bootstrap*<sup>5</sup> é um *Framework* fundamental para implementação da versão 3.0 do sistema, sistema altamente responsivo, dinâmico e de altíssima usabilidade, com ferramentas atuais da Web, utilizando-se de recursos disponibilizados pelo *Framework* constituindo essas características (BOOTSTRAP, 2016).

### 3.6 jQuery

jQuery<sup>6</sup> é uma biblioteca JavaScript que possibilita uma maneira rápida, pequena e rica em recursos para tornar os links mais dinâmicos, eventos como passagem de dados por HTML, documentos, animações decorrentes de comandos no sistema, utilizam desta habilidade. Integrado ao Ajax uma API fácil de usar que funciona através de uma multiplicidade de navegadores. Com uma combinação de versatilidade e capacidade de extensão, jQuery mudou a maneira de milhões de pessoas escreverem JavaScript. O lema “*write less do more*”, descreve perfeitamente a desabilidade da ferramenta (JQUERY, 2016).

---

<sup>5</sup> <http://getbootstrap.com.br/>

<sup>6</sup> <http://jquery.com/>

## 3.7 Ajax

Ajax<sup>7</sup> método responsável por fazer requisições assíncronas ao servidor web. Muito utilizado nos Websites atuais, ao se deparar na situação de carregar apenas o comando exigido da página. AJAX é a arte de trocar dados com um servidor, podendo ocorrer (W3SCHOOLS, 2016).

- Atualizar uma página web sem recarregar a página.
- Receber dados de um servidor depois que a página for carregada.
- Enviar dados para um servidor em Background.

## 3.8 Adobe *Dreamweaver*

O Adobe *Dreamweaver*<sup>8</sup> foi a plataforma de desenvolvimento de interface utilizada para o desenvolvimento do trabalho. O Adobe *Dreamweaver* é desenvolvido pela ADOBE, possuindo licença para utilização e versão gratuita para estudantes ou interessados em conhecer a ferramenta (DREAMWEAVER, 2016).

---

<sup>7</sup> <http://developer.mozilla.org/pt-BR/docs/AJAX>

<sup>8</sup> <http://www.adobe.com/br/products/dreamweaver.html>



## 4 Sistema Desenvolvido

Depois de algumas reuniões informais realizadas em restaurantes de diversos lugares pode-se destacar que a maioria dos casos analisados possuíam parte do processo administrativo escritos e realizados de forma antiga. Este processo antigo, gera um certo atraso em relação a gestão de contas e visualização do fluxo de clientes durante o funcionamento do estabelecimento.

### 4.1 Necessidades e problemas do proprietário do restaurante

Os restaurantes necessitam de um registro mais rápido dos pedidos que são realizados durante o horário de funcionamento, e como são registrados a mão, há uma demora e uma chance de erro humano prejudicar o andamento do fechamento do caixa diário, e no final do dia não há um descritivo de qual tipo de alimento se consome mais e nem a quantidade em números de pedidos, surgindo a necessidade da implantação de um sistema de gerenciamento.

Outro ponto que pode-se destacar nesse tipo de mercado, é o erro da troca de pedidos ou esquecer o que o cliente pediu para ser adicionado ao pedido, algo simples mais que pode gerar prejuízo, tanto para o proprietário, quanto para o cliente que está sendo atendido. É importante frisar que o restaurante quer vender o máximo de pedidos ao final do dia, quando focado em satisfazer a clientela aumenta tanto suas chances de vendas, quanto o índice de fidelidade.

O uso de telefones em geral são os meios mais tradicionais que se existe para comunicação, mas devido ao fato das pessoas estarem mais conectadas a Internet, esse costume está se migrando para utilização de aplicativos capazes de enviar mensagens, vídeos, imagens, de uma forma muito mais rápida e integra para as pessoas que se utilizam desse benefício. A forma de como as pessoas irão utilizar do serviço, faz com que aproxime mais o cliente do proprietário, pois com o tempo é possível traçar quais são os pratos mais requisitados pelos clientes, qual o perfil de cada uma das pessoas que se alimentam no restaurante.

### 4.2 Marketing limitado

Devido a inclusão digital, é possível atingir um público maior que o estabelecimento suporta, utilizando-se das ferramentas web com o desenvolvimento de um Web site, cujo o usuário possa ter contato com o restaurante, ficar por dentro do que acontece e fazer

parte da comunidade de usuários através do cadastro no sistema Web, para realização de pedidos e demais funcionalidades do sistema.

O Web site e sistema são meios de veiculação de propaganda. A utilização do espaço Web, é uma plataforma de marketing , pois nenhum outro lugar você consegue divulgação 24h disponível ao mundo todo. As pessoas podem acompanhar a programação do restaurante, não precisando apenas de saber no dia, podem se antecipar, ou até mesmo agendar os dias que lhe for de mais interesse para almoçar ou jantar de acordo com o cardápio.

### 4.3 Requisitos funcionais e não funcionais do sistema

**Requisitos funcionais:** descrevem explicitamente as funcionalidades e serviços do sistema. A documentação é feita a partir das entradas do sistema e como o sistema deve reagir a essas entradas, qual é o comportamento do sistema, o que o sistema não deve fazer.

- **RF 1:** O sistema deve oferecer telas apropriadas para o usuário ler o que está armazenado.
- **RF 2:** Aparência do sistema: O sistema deve ser de fácil manuseio e aprendizado.
- **RF 3:** Adicionar pedidos: Incluir, alterar e excluir pedidos.
- **RF 4:** Adicionar cardápio: Incluir, alterar e excluir itens do cardápio.
- **RF 5:** Adicionar usuários: Incluir, alterar e excluir usuários.
- **RF 6:** Gerar ordem de pedido.
- **RF 7:** Gerar relatório de clientes.
- **RF 8:** Gerar relatório de fluxo de caixa.
- **RF 9:** Gerar relatório de vendas.

**Requisitos não funcionais:** Requisitos não funcionais do sistema incluem características como escalabilidade, interoperabilidade, facilidade de manutenção, desempenho, portabilidade e segurança da informação.

- **RNF 0:** O sistema deve prever procedimentos de segurança que garantam a integridade dos dados, que permita o controle e acesso para o público a que se destina, permitindo restringir o acesso aos dados disponibilizados de acordo com o tipo de cadastro realizado no sistema, cliente e administrador.



- **RNF 1:** O sistema deve ter um interface de interação com o usuário.
- **RNF 2:** O sistema deve apresentar usabilidade.
- **RNF 3:** O sistema deve ser desenvolvido utilizando-se da linguagem de programação HTML 5.0,Php, Css 3.0, JavaScript.
- **RNF 4:** O sistema de banco de dados deve-se utilizar language o sistema gerenciador de banco de dados MySQL.
- **RNF 5:** O sistema utilizara de Frameworks para desenvolvimento de estilo de páginas, destacando-se Bootstrap 3.0 e JQuery 1.11.
- **RNF 6:** O Sistema disponibilizara um manual de utilização do usuário.
- **RNF 7:** O sistema deverá permitir a gestão de usuários e de perfis.

## 4.4 Caso de uso do Administrador

A figura abaixo demonstra o processo de ações disponíveis ao administrador do sistema: Os artifícios utilizados pelo administrador são simples podendo realizar, criação, leitura, alterações e exclusões dos dados no sistema.

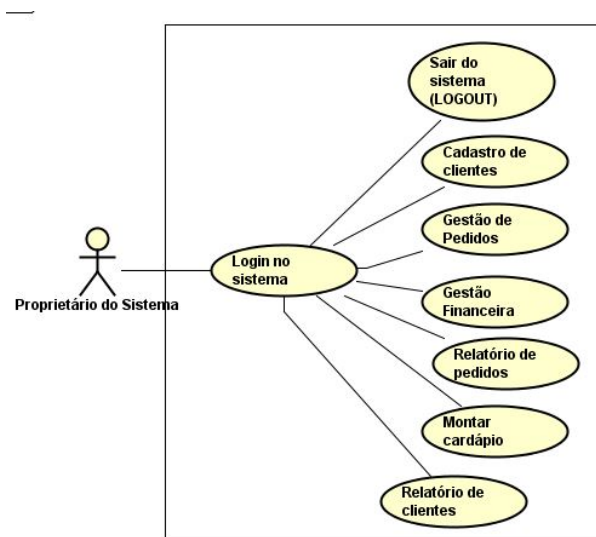


Figura 6 – Caso de uso Administrador

- **Login/Logout do sistema:** Para ter acesso ao sistema e desconectar-se.
- **Cadastro de clientes:** Opção de criação, leitura, alterações e exclusões dos dados no sistema de acordo com cada cliente.

- **Relatório de clientes:** Gerar relatórios de clientes, demonstrando seus dados cadastrais.
- **Relatório de pedidos:** Gerar relatório de todos os pedidos realizados no período indicado.
- **Gestão financeiras:** Gerar relatório descritivo dos dados financeiros da empresa, assim como adicional de valores específicos ao pedido.
- **Gestão de Pedidos:** Acompanha e realiza pedidos do cardápio disponível no dia.
- **Montar Cardápio:** O administrador realiza a montagem dos itens disponíveis para o cardápio do dia.

## 4.5 Caso de uso do cliente

Os artificios utilizados pelo cliente são simples podendo realizar, criação, leitura, alterações e exclusões dos pedidos realizados e de seus dados pessoais.

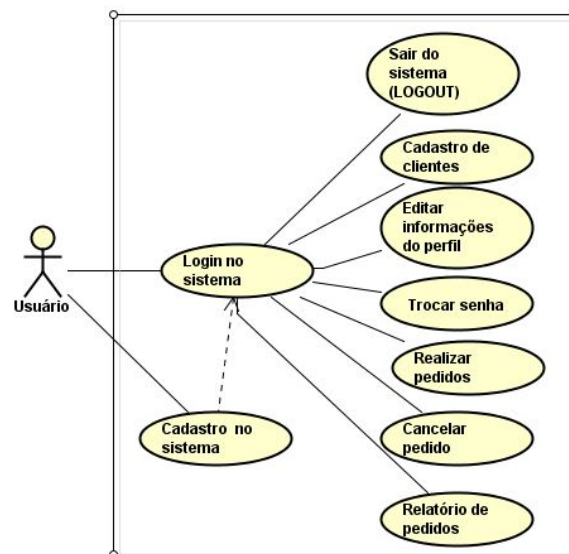


Figura 7 – Caso de uso do usuário

- **Login/Logout do sistema:** Para ter acesso ao sistema e desconectar-se.
- **Cadastro de clientes:** Opção de criação do cadastro no sistema, leitura, alterações e exclusões dos dados pessoais no sistema.
- **Editar informações do perfil:** Para um maior controle administrativo, foi desenvolvido este caso de uso para o próprio cliente ter autonomia sobre seus dados e mantê-los organizados.

- **Trocar senha:** Edição de senha disponível para realização de mudanças e segurança.
- **Realizar pedido:** O usuário pode realizar o pedido on-line, basta ter cadastro no sistema.
- **Cancelar pedido:** O usuário pode realizar o cancelamento do pedido, de acordo com as normas especificadas do restaurante.
- **Relatório de pedidos:** O usuário podera exigir um relatório de todos os seus pedidos a qualquer momento.

Estes casos de usos descritos acima são responsáveis por conter a informação que será vinculada no sistema, sendo a base para criação dos produtos de backlog apresentados na seção seguinte.

## 4.6 Product Backlog

O seguinte Product Backlog foi gerado apartir dos casos de usos da tabela 2:

Procedimentos	Desenvolvimento	E.I	Importância	Demonstrativo
Sprint 1	Banco de Dados	10	Alta	Análise de requisitos.
Sprint-2	Módulo Administrador	21	Alta	Gerenciar perfil Administrador
Sprint-3	Módulo da Empresa	19	Alta	Gerenciar perfil empresa
Sprint-4	Módulo de Cliente	10	Alta	Gerenciar perfil clientes
Sprint-5	Integração entre Módulos	10	Alta	Integração entre Módulos

Tabela 2 – Product Backlog

Legenda das tabelas: E.I: Estimativa Inicial em dias.

### 4.6.1 Sprint Backlogs

A partir das técnicas descritas, foram desenvolvidos cinco etapas para realização do Sprint Backlog do sistema de gerenciamento para restaurantes. Legenda: E.I: Estimativa Inicial em dias.

### 4.6.2 Sprint 1 – Desenvolvimento do Banco de Dados

Após a realização dos levantamentos dos requisitos, foi elaborada construção do banco de dados em etapas descrito na tabela 3:

Procedimentos	Desenvolvimento	Importância	E.I
1	Diagrama de Entidade e Relacionamento (D.E.R)	Alta	4
2	Tabelas	Média	3
3	Chaves estrangeiras BD	Média	3

Tabela 3 – Sprint Banco de Dados

Com os parâmetros alinhados, o Sprint 1 toma formato do banco de dados, é o corpo do projeto, indicando como as variáveis se interrelacionam através de conceitos do processo de desenvolvimento de *Software*, todos os dados do sistema devem ser representados nessa etapa do projeto, uma base de dados bem definida evita vários problemas e facilita a lógica de implementação do sistema.

As tabelas, chaves estrangeiras e os relacionamentos serão realizados a partir dos dados coletados com os interessados.

### 4.6.3 Sprint 2 –Desenvolvimento do Módulo do Administrador

Esta etapa do projeto consiste na elaboração do módulo administrador, a partir dos dados coletados em reuniões,foi desenvolvido o sprint em tempo hábil e de acordo com a tabela 4:

Procedimentos	Desenvolvimento	Importância	E.I
1	Módulo Administrador.	Alta	3
2	Sistema de login e logout Administrador	Alta	1
3	CRUD do Cadastro de produtos	Alta	4
4	CRUD Cardápio	Média	2
5	CRUD Clientes	Média	1
6	CRUD Financeiro	Média	1
7	CRUD Pedidos	Média	1
8	CRUD Relatórios	Média	1

Tabela 4 – Desenvolvimento do Módulo do Administrador.

#### 4.6.4 Sprint 3 –Desenvolvimento do Módulo da Empresa

Nesta etapa do Sprint, o desenvolvimento do módulo empresa ocorreu da seguinte forma de acordo com a tabela 5:

Procedimentos	Desenvolvimento	Importância	E.I
1	Histórico de Pedidos Finalizados.	Alta	2
2	Fluxo de Caixa	Média	2
3	Relatório de Pedidos	Média	2
4	Relatório Financeiro	Média	2
5	Relatório de vendas	Média	2

Tabela 5 – Desenvolvimento do Módulo do Empresa.

#### 4.6.5 Sprint 4 – Desenvolvimento do Módulo do Cliente

Módulo desenvolvido para gerenciar as demandas dos clientes representado no product backlog 4, sua implementação de acordo com a tabela 6:

Procedimentos	Desenvolvimento	Importância	E.I
1	Página do cliente	Alta	2
2	Gerenciamento de informações pessoais e senhas.	Média	4
3	Relatório de Pedidos	Média	4

Tabela 6 – Sprint Desenvolvimento do Módulo do Cliente

#### 4.6.6 Sprint 5 – Integração entre Módulos

Nesta etapa do projeto é etapa em que realizamos a integração entre os módulos do sistema desenvolvido,e devemos seguir os sprints para que não ocorra atrasos no processo de desenvolvimento de *Software*. Após todos os itens formalizados, podemos prosseguir para próxima etapa que é a de validação e testes do sistema. O sprint 5 foi conduzido de acordo com a tabela 7:

Procedimentos	Ação	Importância	E.I
1	Operações e cadastro de Administrador no sistema	Alta	2
2	Operações e cadastro de cliente no sistema	Média	1
3	Teste localhost de usabilidade do sistema	Média	7

Tabela 7 – Sprint Integração de Módulos



# 5 Testes Realizados

## 5.1 Verificação Validação e Testes

Há dois tipos de testes que podem ser usados em estágios diferentes no processo de *Software*: teste da validação e o teste de defeitos. O primeiro tem como objetivo avaliar se o *Software* realiza aquilo que o cliente deseja, ou seja, se atende os requisitos. O teste de defeitos, por sua vez, consiste em encontrar defeitos no sistema em vez de simular seu uso operacional. (SOMMERVILLE, 2007).

Os recursos utilizados para a execução dos testes, incluem:

- **Hardware – Um computador com configuração:** AMD- A8-4500M Apu with Radeon(tm) HDGraphics 1,90Ghz, 4Gb de Ram, Sistema Operacional windows 8 64-bits.
- **Sistema gerenciador de banco de dado - SGBD:** Mysql.
- **Recursos humanos:** Treinamento de habilidades específicas das pessoas envolvidas nos testes.

### 5.1.1 Realização dos testes

Durante o desenvolvimento do projeto foram realizados testes Caixa Preta e Caixa Branca. Os testes Caixa Branca foram realizados manualmente pelo desenvolvedor, depurando os erros que apareciam durante o processo de desenvolvimento. Não foi utilizada nenhuma ferramenta específica para tais testes, visto que as dimensões do sistema e o fato de se ter somente um desenvolvedor possibilitou que o rastreo de erros fosse feito manualmente, com ajuda apenas da IDE e do depurador do Navegador Web.

Os testes Caixa Preta em sua maioria foram realizados também pelo desenvolvedor, mas algumas pessoas foram convidadas para utilização e teste do sistema. Assim, a usabilidade pôde ser avaliada em conjunto com os testes Caixa Preta. De acordo com os erros encontrados as correções necessárias foram feitas e os testes foram novamente executados.

Maiores detalhes sobre os testes Caixa Branca e Caixa Preta estão descritos nos quadros de testes que se encontram no anexo B, deste trabalho.





## 6 Conclusões

Os desafios e obstáculos apareceram, durante o desenvolvimento do projeto, visto que os prazos foram cumpridos, as metas alcançadas e o produto final atende as especificações descritas podendo ser utilizado como um protótipo de aplicação experimental.

Ao término desse projeto de *Software*, foi possível observar o quanto as tecnologias fazem parte do nosso cotidiano, visto que a globalização e as inovações tecnológicas agregam as necessidades do ser humano de interagir e gerir dados na *Web*, sendo esta uma maneira de cortar desvios naturalmente criados ao se realizar pedidos em um determinado estabelecimento. A metodologia utilizada geralmente pelos comerciantes para atendimento de pedidos é o telefone, porém gera muito estresse entre funcionários e clientes dependendo do pedido do cliente. Com o desenvolvimento dessa ferramenta o pedido já vem padronizado, podendo ser alterado pelo usuário evitando falhas humanas, na descrição do pedido.

Este estudo pode ser aprofundado, ampliando suas funcionalidades e diversificando em vários tipos de setores que tenham como fundamento a realização de pedidos. Melhorias futuras podem incluir a inclusão de módulos com novas funcionalidades para o usuário e a criação de um sistema mobile, o que pode permitir ao sistema alcançar um público maior. Além disso, a realização de estudos sobre ergonomia, interação humano-computador podem aprimorar as interfaces, oferecendo maior potencial de usabilidade do sistema.



# Referências

- BERTALANFFY, L. V. *Teoria geral dos sistemas*. [S.l.]: Vozes, 1975. v. 351. Citado na página 27.
- BOOTSTRAP. *Bootstrap Oficial* [<http://getbootstrap.com/>]. 2016. Disponível em: <<http://getbootstrap.com/>>. Citado na página 42.
- COCKBURN, A. *Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software*. Bookman Editora, 2005. ISBN 978-85-7780-019-3. Disponível em: <<https://books.google.com.br/books?id=gbBRo8CxmFUC>>. Citado na página 37.
- Dall'Oglio, Pablo. *Php-programando com orientacao a objetos*. [S.l.]: Novatec Editora, 2015. Citado na página 35.
- DAMIANI, E. *Javascript: O guia de consulta rápida*. 3 ed. ed. São Paulo: Novatec, 2006. ISBN 978-85-7522-150-1. Citado na página 40.
- Date, Christopher J. *Introdução a sistemas de bancos de dados*. [S.l.]: Elsevier Brasil, 2004. Citado na página 36.
- DEVMEDIA. *DevMedia Oficial* [<http://devmedia.com.br/>]. 2016. Disponível em: <<http://devmedia.com.br/>>. Citado na página 32.
- DREAMWEAVER. *Dreamweaver Oficial* [<http://adobe.com/>]. 2016. Disponível em: <<http://adobe.com/Dreamweaver>>. Citado na página 43.
- ERP, P. *O que é ERP? Oficial* [<http://portalerp.com/>]. 2016. Disponível em: <<http://portalerp.com/erp/5-entenda-erp>>. Citado na página 28.
- FEITOSA, D. de L.; GARCIA, L. S. Sistemas de reputação: Um estudo sobre confiança e reputação no comércio eletrônico brasileiro/reputation systems: A study on trust and reputation in brazilian electronic commerce. *Revista de Administração Contemporânea*, Associação Nacional de Pós-Graduação e Pesquisa em Administração, v. 20, n. 1, p. 84, 2016. Citado na página 23.
- FILHO, M. P. d. C. Os sistemas de informação e as modernas tendências da tecnologia e dos negócios. *Revista de Administração de Empresas*, SciELO Brasil, v. 34, n. 6, p. 33-45, 1994. Citado na página 23.
- FILHO, W. de P. P. *Engenharia de software*. [S.l.: s.n.], 2003. v. 2. Citado 4 vezes nas páginas 24, 28, 29 e 37.
- FLANAGAN, D. *JavaScript o guia definitivo*. Porto Alegre: Bookman, 2004. OCLC: 709388290. ISBN 978-85-363-0475-5. Citado na página 41.
- GUEDES, G. T. *UML 2 - Uma Abordagem Prática - 2ª Edição - Novatec Editora*. 2. ed. São Paulo (SP): Novatec, 2011. ISBN 978-85-7522-281-2. Disponível em: <[http://novatec.com.br/livros/abordagem\\_p/](http://novatec.com.br/livros/abordagem_p/)>. Citado na página 37.

- JQUERY. *Jquery Oficial* [<http://www.jquery.com/>]. 2016. Disponível em: <<http://jquery.com/>>. Citado na página 42.
- K19 Treinamentos. Orientação a objetos em Java. In: . [S.l.]: K19 Treinamentos, 2013. Citado na página 33.
- KRASNER, G. E.; POPE, S. T. et al. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, v. 1, n. 3, p. 26–49, 1988. Citado na página 32.
- LAUDON, K. C.; LAUDON, J. P. *Sistemas de informação gerenciais: administrando a empresa digital*. [S.l.]: Pearson Prentice Hall, 2005. Citado na página 27.
- LUKE; WELLING. *PHP e MySQL*. Rio de Janeiro: Elsevier, 2005. OCLC: 298931090. ISBN 978-85-352-1714-8. Citado 2 vezes nas páginas 36 e 42.
- MARTINS, G. B.; FERNANDES, M. Planejamento e gestão de restaurantes. *UNILUS Ensino e Pesquisa*, v. 13, n. 30, p. 283, 2016. Citado na página 23.
- MORAZ, E. *Treinamento prático em PHP*. São Paulo: Digerati Books, 2005. OCLC: 230882070. ISBN 978-85-7702-013-3. Citado na página 42.
- NETO, P. *PostgreSQL: técnicas avançadas versões open source 7.x e 8.x., soluções para desenvolvedores e administradores de bancos de dados*. 4ª. ed. São Paulo (SP): Érica, 2007. (Bancos de dados). ISBN 978-85-7194-948-5. Disponível em: <<http://www.saraiva.com.br/postgresql-tecnicas-avancadas-versoes-open-source-7x-serie-banco-de-dados-125517.html>>. Citado na página 35.
- NIEDERAUER, JULIANO. *Web interativa com Ajax e PHP*. [S.l.]: Novatec Editora, 2007. Citado na página 41.
- OLIVEIRA, D. d. P. R. d. *Sistemas de informações gerenciais estratégicas, táticas, operacionais*. São Paulo: Atlas, 2007. OCLC: 212884508. ISBN 978-85-224-4613-1. Citado na página 27.
- PAUL; DEITEL, H. *Java como programar*. São Paulo (SP): Pearson Prentice Hall, 2010. 00000. ISBN 978-85-7605-563-1 85-7605-563-5. Citado na página 28.
- PEREIRA, J. d. S.; VIEIRA, R. B.; FONSECA, K. Z. Características administrativas dos restaurantes self service de santo antônio de jesus–bahia. *Revista Saúde e Desenvolvimento*, v. 8, n. 4, p. 36–51, 2016. Citado na página 23.
- PETERS, J. F.; PEDRYCZ, W. *Software engineering: an engineering approach*. New York: John Wiley, 2000. (Worldwide series in computer science). ISBN 978-0-471-18964-0. Citado 2 vezes nas páginas 29 e 30.
- PRESSMAN, R. S. *Engenharia de software*. 6 e.d. ed. São Paulo: McGraw-Hill, 2006. OCLC: 69936802. ISBN 978-85-86804-57-1. Citado 5 vezes nas páginas 23, 28, 31, 34 e 35.
- PRESSMAN, R. S. *Engenharia de Software 6a edição*. 6 e.d. ed. São Paulo: McGraw-Hill, 2006. 720 p. ISBN 8586804576. Citado 2 vezes nas páginas 23 e 37.
- PRESSMAN, R. S.; TRAVIESO, M. M. G. *Engenharia de software*. Rio de Janeiro: McGraw-Hill, 2002. OCLC: 55912185. ISBN 978-85-86804-25-0. Citado na página 37.

- REZENDE, D. A. *Engenharia de software e sistemas de informação*. Rio de Janeiro: Brasport, 2006. OCLC: 71311792. ISBN 978-85-7452-215-9. Citado na página 29.
- REZENDE, D.A.; ABREU, A. *Tecnologia da Informação : Aplicada a Sistemas de Informação Empresariais*. Sao Paulo: Atlas, 2000. 311P. p. ISBN 85-224-2705-4. Citado na página 28.
- RODRIGUES, L. C. Impactos dos sistemas de informação. In: . Blumenau-SC: [s.n.], 1999. ISBN 04-711-8964-2. Citado na página 27.
- SANTOS, R. *Introdução à programação orientada a objetos usando Java*. [S.l.]: Elsevier, 2003. Citado na página 34.
- SCHWABER, K. *Agile project management with Scrum*. [S.l.]: Microsoft press, 2004. Citado na página 31.
- SCRUMSTUDY. A guide to the Scrum Body of Knowledge. In: . [S.l.]: SCRUMstudy, 2013. Citado na página 31.
- SOMMERVILLE, I. *Engenharia de software*. 8 ed.. ed. São Paulo: Pearson : Wesley, 2007. OCLC: 817223088. ISBN 978-85-88639-28-7. Citado 4 vezes nas páginas 29, 30, 37 e 53.
- TONSIG, SERGIO LUIZ. *PHP COM AJAX NA WEB 2.0: COM EXEMPLOS MUITO PRATICOS!* [S.l.]: CIENCIA MODERNA, 2007. ISBN 978-85-7393-638-4. Citado 2 vezes nas páginas 34 e 35.
- W3SCHOOLS. *HTML Tutorial* [<http://www.w3schools.com/>]. 2016. Disponível em: <<http://www.w3schools.com/html/default.asp>>. Citado 2 vezes nas páginas 39 e 43.
- ZEMEL, T. *WEB DESIGN RESPONSIVO: PAGINAS ADAPTAVEIS PARA TODOS OS DISPOSITIVOS*. São Paulo: CASA DO CODIGO, 2014. ISBN 978-85-66250-07-7. Disponível em: <<https://books.google.com.br/books?id=dv6LnQEACAAJ>>. Citado na página 40.



# Apêndices





# APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO DE QUALIDADE

## 1. Usabilidade

1.1. Existe ambiguidade da lingua portuguesa apresentados na interface?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

1.2. Qual a frequência de erros ortográficos apresentados na interface?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

1.3. O usuário reconhece com facilidade no protótipo o conceito lógico e sua aplicabilidade?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

1.4. O usuário aprende com facilidade a aplicação?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

1.5. O usuário opera e controla cada operação do *Software* com facilidade?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

## 2. Funcionalidade

2.1. A interface mantém uma padronização própria em relação a configuração de janela?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

2.2. A interface mantém uma padronização própria em relação a formatação de ícones?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

2.3. As funções implementadas no protótipo estão todas implementadas corretamente?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

2.4. As funções verificadas no protótipo geram resultados corretos ou conforme o esperado?

( )Excelente ( )Bom ( )Regular ( )Insuficiente ( )Não se Aplica

2.5. O protótipo evita acesso não autorizado, acidental ou deliberado?

Excelente Bom Regular Insuficiente Não se Aplica

### 3. Confiabilidade

3.1. Qual a frequência de falhas por defeitos que o protótipo apresenta?

Excelente Bom Regular Insuficiente Não se Aplica

3.2. O protótipo mantém um nível de desempenho especificado nos casos de falhas no *Software* ou violação nas interfaces especificadas?

Excelente Bom Regular Insuficiente Não se Aplica

### 4. Eficiência

4.1. É satisfatório o tempo de resposta do protótipo, o tempo de processamento e velocidade na execução de suas funções?

Excelente Bom Regular Insuficiente Não se Aplica

4.2. É satisfatória a quantidade de recursos usados pelo protótipo e a duração de seu uso na execução de suas funções?

Excelente Bom Regular Insuficiente Não se Aplica

### 5. Aspectos Visuais

5.1. As telas apresentam somente informações necessárias e utilizáveis, sensíveis ao contexto?

Excelente Bom Regular Insuficiente Não se Aplica

5.2. As telas apresentam contrastes e cores, facilitando a leitura?

Excelente Bom Regular Insuficiente Não se Aplica

5.3. As telas exibem as mensagens com bom aspecto visual, utilizando, com moderação, negrito, itálico e sublinhado?

Excelente Bom Regular Insuficiente Não se Aplica

## APÊNDICE B – Quadro de testes

Caso de Teste	
Número do teste	1
Objetivo	Verificar consistência do Banco de dados
Entradas	Inserção dos dados.
Resultado esperados	Inserção de dados corretamente. Visualização das integrações de tabelas e dados inseridos.
Dependências	Não se aplica
Procedimentos de teste	
Preparação	Definir base de dados a ser testada, realizar Backup e testar.
Inicialização	Execução do banco de dados
Execução	Inserção, alteração de dados. Visualização dos dados inseridos.
Reorganização	Restaurar dados para o inicial.
Interrupção	Não se aplica
Recursos Específicos	Qualquer SGBD para Mysql
AVALIAÇÃO DOS CASOS DE TESTE	
Responsável	Humberto Afonso Nunes de Melo
Resultados obtidos	Banco de dados com sucesso

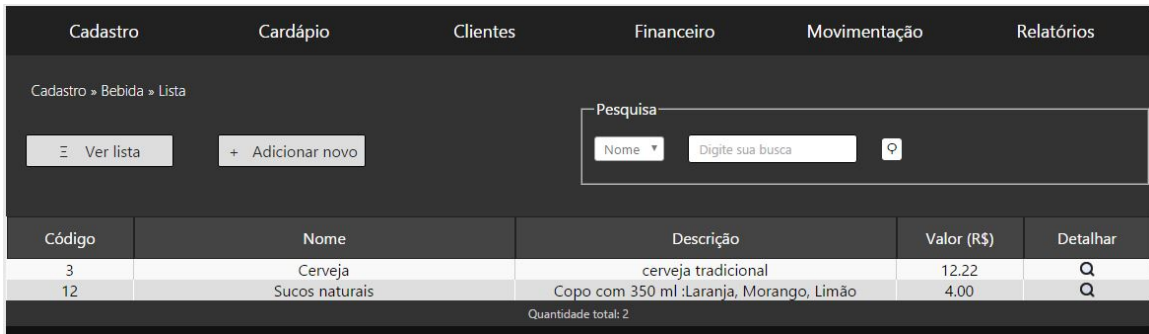
Figura 8 – Teste 1

Caso de Teste	
Número do teste	2
Objetivo	Verificar Layout de telas
Entradas	Interfaces apresentadas ao usuário
Resultado esperados	Usuabilidade do usuário aplicável
Dependências	Não se aplica
Procedimentos de teste	
Preparação	Interfaces dispostas em ordem proposta pelo sistema
Inicialização	Execução do sistema.
Execução	Iniciar o sistema, para visualizar as interfaces
Reorganização	Caso haja erros reportar e corrigir.
Interrupção	Não se aplica
Recursos Específicos	IDE Dreamweaver
AVALIAÇÃO DOS CASOS DE TESTE	
Responsável	Humberto Afonso Nunes de Melo
Resultados obtidos	Interfaces com sucesso

Figura 9 – Teste 2

# ANEXO A – Anexo dos prints do sistema

As principais telas do sistema estão sendo apresentadas pois o foco é demonstrar as diferentes funcionalidades e interação do usuário com o sistema.



Cadastro Cardápio Clientes Financeiro Movimentação Relatórios

Cadastro » Bebida » Lista

Ver lista Adicionar novo

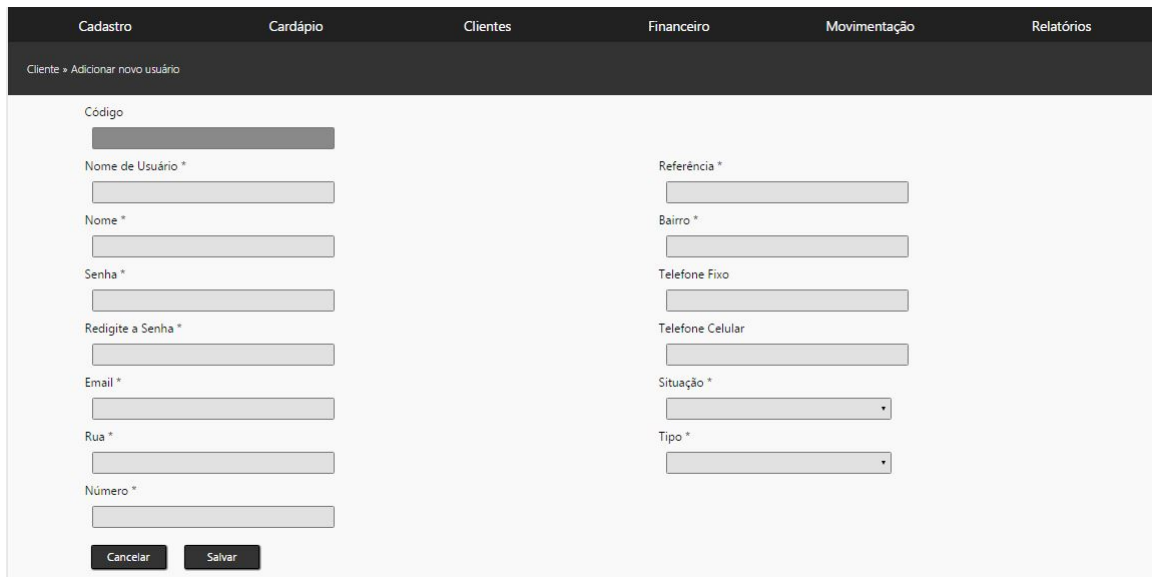
Pesquisa

Nome Digite sua busca

Código	Nome	Descrição	Valor (R\$)	Detalhar
3	Cerveja	cerveja tradicional	12.22	Q
12	Sucos naturais	Copo com 350 ml :Laranja, Morango, Limão	4.00	Q

Quantidade total: 2

Figura 10 – Item com adicional monetário.



Cadastro Cardápio Clientes Financeiro Movimentação Relatórios

Cliente » Adicionar novo usuário

Código

Nome de Usuário \*

Nome \*

Senha \*

Redigite a Senha \*

Email \*

Rua \*

Número \*

Referência \*

Bairro \*

Telefone Fixo

Telefone Celular

Situação \*

Tipo \*

Cancelar Salvar

Figura 11 – Cadastro de cliente.

Cadastro    Cardápio    Clientes    Financeiro    Movimentação    Relatórios							
Cardápio » Histórico de Cardápios							
Pesquisa							
Data ▾ Digite sua busca 🔍							
Cód.	Data	Arroz	Carne	Complemento	Feijão	Fruta	Salada
1	24/06/2015	+ Arroz Branco	+ Contra -Filé	+ Salada extra	+ feijão carioca	+ Maça	+ salada
2	30/06/2015	+ Arroz Branco	+ Contra -Filé	+ Salada extra	+ feijão carioca	+ Maça	+ salada
3	23/03/2016	+ Arroz Branco	+ Contra -Filé	+ Salada extra	+ feijão carioca + Feijão roxo	+ Maça	
4	02/05/2016	+ Arroz Branco	+ Contra -Filé	+ Salada extra			+ salada
5	16/05/2016	+ Arroz carreteiroa	+ Contra -Filé		+ feijão carioca + Feijão roxo	+ Maça + Abacaxi	+ salada
7	17/05/2016	+ Arroz Branco + Arroz carreteiroa	+ Contra -Filé + Picanha	+ Salada extra	+ feijão carioca + Feijão roxo	+ Maça + Abacaxi	+ salada + Salpicão
8	27/05/2016	+ arroz com milho	+ Picanha	+ Salada extra	+ Feijão roxo	+ Maça	+ salada
Quantidade total: 7							

Figura 12 – Histórico de Cardápios.

Data

06/06/2016

**Arroz**

- Arroz Branco
- Arroz carreteiroa
- arroz com milho

**Carne**

- Contra -Filé
- Picanha

**Complemento**

- Salada extra

**Feijao**

- feijão carioca
- Feijão roxo

**Fruta**

- Maça
- Abacaxi

**Salada**

- salada
- Salpicão

Cancelar    Salvar

Figura 13 – Montar cardápio.

Código	Nome	Descrição	Detalhar
2	Arroz Branco	temperado ao alho e sal.	Q
10	Arroz carreteiroa	Arroz branco com milho ervilha, ervas finas e toque do chefe	Q
18	arroz com milho	arroz com milho	Q

Quantidade total: 3

Figura 14 – Item sem adicional monetário.

Código	Pagar para	Descrição	Data de Lançamento	Data de Vencimento	Valor (R\$)	Forma de Pagamento	Situação	Pagar conta	Detalhar
1	Cordeiro	Presuntos, diversos	16/05/2016	16/05/2016	100.00	CartaoCredito	Pago		Q

Quantidade total: 1

Figura 15 – Contas a Pagar/Pagas

Código	Nome Cliente	Tipo da Conta	Data Lançamento	Data de Vencimento	Valor (R\$)	Situação	Lançar Recebimento	Detalhar
5	Humberto	Mensalidade	16/05/2016	23/06/2016	250.00	Aberto		Q
1	Admin	Pedido	24/06/2015	24/06/2015	47.47	Recebido		Q
2	Humberto	Mensalidade	23/03/2016	23/03/2016	250.00	Recebido		Q
3	Humberto	Mensalidade	23/03/2016	23/04/2016	250.00	Recebido		Q
4	Humberto	Mensalidade	23/03/2016	23/05/2016	250.00	Recebido		Q

Quantidade total: 5

Figura 16 – Contas a Receber/Recebidas.

Código

Usuário \*

Admin

Data do Pedido \*

06/06/2016

Escolher itens do pedido

Valor

10

Cancelar Salvar

Data do Cardápio

06/06/2016

**Arroz**

Arroz carreteiroa

**Bebida**

Cerveja

Sucos naturais

**Carne**

Contra -Filé

**Complemento**

Salada extra

**Feijao**

Feijão roxo

**Fruta**

Maça

**Outros**

Doces

Chocollate

Adicional de carne

**Salada**

salada

Figura 17 – Fazer pedidos.

Cadastro Cardápio Clientes Financeiro Movimentação Relatórios

Movimentação » Lista de Pedidos

Ver lista Adicionar novo

Pesquisa

Data Digite sua busca

Cód.	Data	Cliente	Rua	Número	Bairro	Referência	Valor R\$	Visualizar	Detalhar
12	06/06/2016	Admin	Admin	Admin	Admin	Admin	42.22		

Quantidade total: 1

Figura 18 – Lista de pedidos.

Código do Pedido: <b>12</b>	Data do Pedido: <b>06/06/2016</b>	Nome de Usuário do Cliente: <b>Admin</b>	Valor: <b>42.22</b> R\$
Nome do cliente: <b>Admin</b>	Rua: <b>Admin</b> Número: <b>Admin</b>	Bairro: <b>Admin</b> Referência: <b>Admin</b>	Telefone Celular: <b>Admin</b> Telefone Fixo: <b>Admin</b>

Tipo do item	Nome do item	Valor
Bebida	Cerveja	12.22
Carne	Contra -Filé	---
Complemento	Salada extra	---
Outros	Doces	20.00
Salada	salada	---
Arroz	Arroz carreteiroa	---
----	<b>Valor total</b>	<b>42.22</b>

Figura 19 – Ordem de serviço de um pedido.



# ANEXO B – Modelo lógico Banco de Dados

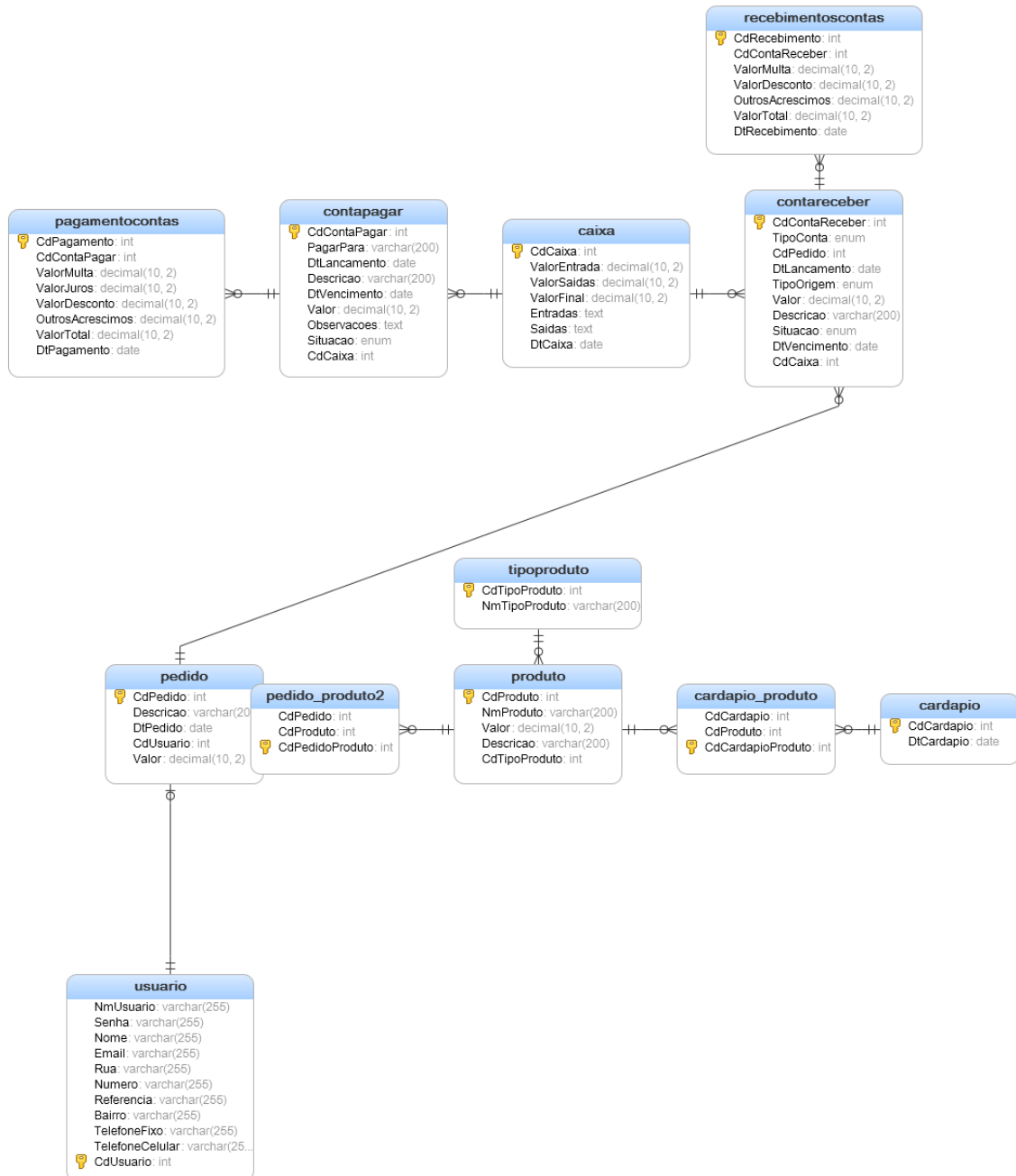


Figura 20 – Modelo lógico do Banco de dados.