

CENTRO UNIVERSITÁRIO UNIFACVEST  
CURSO DE CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE CONCLUSÃO DE CURSO  
WELLINGTON RODRIGUES LEMOS

**Ello Enterprise: Sistema de *ERP* Modularizado (*Enterprise Resource Planning*)**

**LAGES - SC  
2015**

**WELLINGTON RODRIGUES LEMOS.**

**Ello Enterprise: Sistemas de *ERP* Modularizado (*Enterprise Resource Planning*)**

Projeto apresentado à Banca Examinadora do Trabalho de Conclusão do Curso de Ciência da Computação para análise e aprovação.

**LAGES - SC  
2015**

**WELLINGTON RODRIGUES LEMOS**

**Ello Enterprise: Sistemas de *ERP* ( *Enterprise Resource Planning* )**

Trabalho de Conclusão do Curso de Ciência da Computação apresentado ao Centro Universitário UNIFACVEST como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação.

Prof. MSc. Márcio José Sembay

Lages, SC \_\_\_\_/\_\_\_\_/2015. Nota \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**LAGES - SC**

**2015**

## RESUMO

Este estudo tem como objetivo demonstrar uma análise do comportamento de um sistema de gestão. Com base na discussão sobre qual a importância de se ter em uma empresa um sistema *ERP* (*Enterprise Resource Planning*), para a gestão de negócios, aquisição de equipamentos na manipulação de suas informações e dados. A questão utilizada nesta análise é a comparação com o manual de gestão, com destaque para o domínio de suas vantagens no uso de gestão informatizada de negócios. Sistemas de *ERP* (*Enterprise Resource Planning*) representam uma ferramenta de tecnologia da informação para melhorar o desempenho organizacional como um todo. Usando técnicas modernas que oferecem uma maior integração e padronização dos processos e atividades dos departamentos da empresa e um maior fluxo de informações e respostas rápidas ao mercado. Este estudo tem como objetivo demonstrar uma análise da operação de um sistema de gestão integrado. Ao longo deste trabalho, pode-se explicar as ferramentas usadas na criação e surgimento deste tipo de sistema, a importância e os benefícios do controle de gestão com a utilização do *ERP* em relação ao controle manual, as vantagens deste sistema para as empresas comerciais e as perspectivas futuras para o planejamento de recursos empresariais.

**Palavras-chave:** Sistema *ERP*. Controle Gerencial. Desempenho organizacional.

## ABSTRACT

This study has the objective to demonstrate an analysis of the behavior of a management system. Based on the discussion about the importance of companies having an *ERP* (*Enterprise Resource Planning*) system for business management, acquiring facilities in the handling of your information and data. The Question used in this discussion is the comparison with the manual business management, highlighting the predominance of its advantages in the use of computerized business management. *ERP* (*Enterprise Resource Planning*) systems represent a tool of information technology aimed at improving organizational performance as a whole. Using modern techniques offer greater integration and standardization of processes and activities of the business departments and greater flow of information and quick responses to market. This study has the objective to demonstrate an analysis on the operation of an integrated management system. Throughout this paper, explain the tools used in the creation and emergence of this type of system, the importance and benefits of management control with the use of *ERP* compared to manual control, the advantages provided by this system for business enterprises and future prospects for *Enterprise Resource Planning*.

**Keywords:** *ERP* System. Managerial Control. Organizational Performance.

## RESUMEN

El estudio se debe a informar de un análisis del comportamiento de un sistema de gestión. Sobre la base de la discusión sobre la importancia de las empresas que tienen un sistema *ERP* (*Enterprise Resource Planning*) para la gestión empresarial, la adquisición de instalaciones en el manejo de su información y datos. La pregunta utilizada en este análisis es la comparación con el manual de gestión empresarial, destacando el predominio de sus ventajas en el uso de la gestión empresarial informatizada. *ERP* (planificación de recursos empresariales) sistemas representan una herramienta de tecnología de la información destinada a mejorar el desempeño organizacional en su conjunto. Utilizando técnicas modernas ofrecen una mayor integración y estandarización de los procesos y actividades de los departamentos de la empresa y un mayor flujo de información y de respuestas rápidas al mercado. Este estudio tiene el objetivo de demostrar un análisis sobre el funcionamiento de un sistema de gestión integrado. A lo largo de este trabajo, explicar las herramientas utilizadas en la creación y el surgimiento de este tipo de sistema, la importancia y los beneficios del control de la gestión con el uso de *ERP* en comparación con el control manual, las ventajas que ofrece este sistema para las empresas comerciales y las perspectivas futuras para la planificación de recursos empresariales.

Palabras clave: Sistema *ERP*. El control gerencial. El desempeño de la organización,

## **LISTA DE SIGLAS**

C – Linguagem de Programação

CSS – Cascading Style Sheets (Folha de Estilo em Cascata)

ERP – Enterprise Resource Planning (Sistemas Integrados de Gestão Empresarial)

HTML – HyperText Markup Language (Linguagem de Marcação de Hipertexto)

JAVA – Linguagem de Programação

MRP – Manufacturing Resource Planning (Planejamento dos Recursos de Manufatura)

PERL – Linguagem de Programação

PHP – Personal Home Page (Página Pessoal)

SI – Sistema de Informações

SIG – Sistema de Informações Gerenciáveis

URL – Uniform Resource Locator (Localizador Padrão de Recursos)

WEB – World Wide Web (Grande Entrelaçamento Mundial)

## LISTA DE FIGURAS

Figura 1 - Estrutura de um código <i>PHP</i> .....	14
Figura 2 - Aparência de um código em <i>PHP</i> .....	14
Figura 3 - Códigos em <i>HTML</i> .....	15
Figura 4 - Aparência após código de <i>HTML</i> .....	16
Figura 5 - Relacionamento do Banco de Dados <i>MySQL6</i> . ....	18
Figura 6 - Interface do Adobe <i>Dreamwaver S5</i> .....	19
Figura 7 - Tipos de Sistemas de Informação.....	21
Figura 8 - Como Funciona um <i>ERP</i> .....	25
Figura 9 – Diagrama de Caso de Uso.....	31
Figura 10 – Diagrama de Sequência.....	32
Figura 11 – Diagrama de Classe.....	33
Figura 12 – Diagrama de Implantação.....	34
Figura 13 – Tela de Confirmação de Usuário e Senha.....	35
Figura 14 – Tela Inicial.....	35
Figura 15 – Tela Inicial Responsiva.....	36
Figura 16 – Modulo Criação.....	37
Figura 17 – Modulo Criação.....	37
Figura 18 – Modulo Almojarifado.....	38



## SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 Justificativa.....	11
1.2 Objetivos.....	12
1.2.1 Objetivo Geral .....	12
1.2.2 Objetivos Específicos .....	12
2 FUNDAMENTAÇÃO TEÓRICA .....	13
2.1 PHP – História e Conceitos.....	13
2.2 HTML .....	15
2.2.1 Definição .....	15
2.3 CSS .....	16
2.4 JavaScript .....	17
2.5 Banco de Dados <i>MySQL</i> .....	18
2.6 Adobe Dreamweaver .....	19
2.7 Sistema de Informação .....	20
2.7.1 Sistema de Informações Gerenciais (SIG).....	21
2.8 Url “Amigável”.....	22
2.9 Desing Responsivo .....	23
3 <i>ERP</i> .....	23
3.1 Como funciona um <i>ERP</i> .....	24
3.1.1 O surgimento do ERP (Enterprise Resource Planning).....	25
3.2 Controle gerencial manual x <i>ERP</i> ( <i>Enterprise Resource Planning</i> ) .....	26
3.3 Vantagens do ERP (Enterprise Resource Planning).....	27
4 METODOLOGIA.....	28
4.1 Tipos de Pesquisa .....	28
4.2 Método Utilizado.....	28
5 PROJETO .....	30
5.1 Introdução.....	30
5.2 Diagramas – UML .....	31
5.2.1 Diagrama de Caso de Uso .....	31
5.2.2 Diagrama de Componentes.....	32
5.2.3 Diagrama de Classe .....	33
5.2.4 Diagrama de Implantação.....	34
5.3 Design de Telas do Sistema.....	34
6 CONCLUSÃO.....	39
7 REFERÊNCIAS BIBLIOGRÁFICAS .....	40
8 ANEXOS .....	45

# 1 INTRODUÇÃO

No cenário empresarial mundial atualmente as empresas estão em busca de aumentar sua competitividade uma com as outras, tais como redução de custos, melhoria do produto, agregando mais valor a ele e sobressaindo-se à concorrência, ou pela especialização em algum segmento de mercado. Essa competição tem escalas globais, assim, acontecimentos em países distantes podem trazer consequências instantâneas para a indústria local.

Segundo Laudon (2001), os sistemas *ERP* permitem que as decisões e a base de dados de todas as partes da organização sejam integradas, de modo que as consequências das decisões de uma parte da organização sejam refletidas nos sistemas do restante da empresa.

Devido ao sistema permitir a integração dos módulos, é possível diagnosticar as áreas mais ou menos eficientes da empresa e focar em processos que possam ter melhor desempenho com a ajuda do conjunto de sistemas.

Percebe-se que mesmo nos dias de hoje, alguns empresários utilizam-se ainda de métodos defasados e pouco eficiente para administração de suas respectivas empresas o que muitas das vezes delimitam suas visões para um prejuízo que podem ter por utilizar métodos de gestão manual. Controles manuais não geram relatórios confiáveis e seguros a seus gestores. Assim sendo, as informações geradas não funcionam como parâmetros para o crescimento e desenvolvimento da empresa.

Segundo Laudon (2004), os Sistemas Integrados visam se adequar as necessidades dos clientes, ou seja, o cliente escolhe os módulos que buscam atender suas reais necessidades.

Hoje no mercado existem sistemas *ERP* (*Enterprise Resource Planning*) que são plataformas de software desenvolvida muitas das vezes específicas para correlacionar os diversos departamentos de uma empresa, possibilitando a automação e armazenamento de todas as informações de negócios.

Muitas empresas estão optando pelos pacotes *ERP* (*Enterprise Resource Planning*) por vários motivos, como: frustrações com sistemas incompatíveis, departamentos de tecnologia de informação que não possibilitam a integração entre esses sistemas, mudanças estatais da empresa e outros que influenciam diretamente na obtenção de maior competitividade.

A aplicação de controle feito manualmente pode resultar no retardamento da comunicação de informações entre os setores, acarretando atrasos na sequência de uma produção, carregamento ou pagamento, comprometendo também os estoques.

A sigla *ERP* traduzida literalmente significa "Planejamento dos Recursos da Empresa", o que pode não refletir a realidade de seus objetivos.

Conforme Koch *et al.* (2001) afirmam: "...esqueça a parte do planejamento – ele não o faz, e esqueça os recursos, é apenas um termo de ligação. Mas lembre-se da parte da empresa. Esta é a real ambição dos sistemas *ERP*". Os sistemas *ERP*, também conhecido no Brasil de Sistemas Integrados de Gestão Empresarial, controlam e fornecem suporte a todos os processos operacionais, produtivos, administrativos e comerciais da empresa.

Todas as transações realizadas pela empresa devem ser registradas, a fim de que as consultas extraídas do sistema possam refletir a realidade. O *ERP* é um sistema integrado que possibilita um fluxo de informações único, contínuo e consistente por toda a empresa, sob uma única base de dados. É um instrumento para a melhoria de processos como a produção, compras ou distribuição, informações *on-line* e em tempo real. Em suma, o sistema permite visualizar as transações efetuadas pela empresa desenhando um amplo cenário de seus negócios.

Segundo Stamford (2000), o *ERP* possibilita um fluxo de informação único, contínuo e consistente por toda a empresa sob uma única base de dados. É um instrumento para a melhoria de processos de negócio, orientado por esses processos e não pelas funções e departamentos da empresa, com informações *on-line* em tempo real. Permite visualizar por completo as transações efetuadas pela empresa, desenhando um amplo cenário de seus processos de negócios.

## **1.1 Justificativa**

Notou-se a importância de um projeto para um sistema que oferece uma utilização mais abrangente e otimizado para os processos efetuados dentro de uma empresa, na qual encontrava-se com dificuldade em compatibilidade com o sistema atual. Foi deparado com o decorrer do período de utilização do software, que o mesmo, estava se encontrando devassado e com uma grande dificuldade de mantê-lo atualizado conforme o gosto do usuário, pois as integrações de módulos simultâneos já estava dando conflito de informações.

Os erros e problemas enfrentados cotidianamente traz a curiosidade da busca de novos rumos para a administração empresarial, nos levando até então ao um novo sistema de gestão empresarial o conclamado sistema de *ERP* (*Enterprise Resource Planning*).

Segundo Anthony e Govindarajan (2002) dizem que as primeiras aplicações dos sistemas ERP foram caras e demoradas. O desenvolvimento do setor foi aos poucos criando metodologias e ferramentas que tornaram os sistemas integrados práticos e ágeis, contribuindo para a difusão dos mesmos.

A escolha de um novo software para uma empresa pode ser a diferença entre lucro e prejuízo em toda a operação. Isso porque, hoje em dia, todas as ações básicas de uma empresa estão em computadores: balancetes, clientes, organização do estoque, folha de pagamento. Tudo depende do bom funcionamento de programas.

Se o software da sua empresa não está funcionando bem ou não está otimizado para os processos do seu serviço ou produto, trará problemas de atraso e perda de dinheiro. Devido à importância, há alguns anos as empresas começaram a perceber que softwares customizados poderiam entregar mais valor do que aqueles vendidos como pacotes fechados. Muitas vezes, softwares fechados não atendem completamente as necessidades da empresa e, embora mais caro, um software customizado poderá te dar uma grande vantagem competitiva.

## **1.2 Objetivos**

Ao buscar meios que viabilizassem encontrar respostas para a justificativa deste estudo, foi determinado um objetivo geral, que por sua vez dependem de objetivos específicos para tornar factível o resultado desta investigação.

### **1.2.1 Objetivo Geral**

Buscar tecnologias novas, ferramentas, sistemas e procedimentos que auxiliarão na criação de um sistema de automatização dos processos entre os departamentos que se correlacionam dentro de uma empresa, desde a confecção até a distribuição do produto.

### **1.2.2 Objetivos Específicos**

Os objetivos específicos consistem no desenvolvimento de um sistema de gestão modularizado, inicialmente três, a qual entre eles estão relacionados ao departamento de vendas, estoque e gerenciamento de ordem de serviços buscando obter os seguintes pontos:

- a) Sistematizar o controle de entrada, processo e saída da empresa, dando suporte no apoio a decisões das autoridades.
- b) Pesquisar e analisar com quais ferramentas ocorrerá o trabalho, para obter informações se a mesma suprirá suas necessidades no decorrer do desenvolvimento.
- c) Disponibilizar em multiplataformas o uso do sistema.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Segundo Netto (2008), os sistemas ERP fornecem condições de maiores oportunidades futuras, haja vista a possibilidade de maior monitoração e conhecimento do negócio, proporcionando o gerenciamento eficiente das informações e um alto desempenho nas atividades operacionais e estratégicas.

Os Sistemas Integrados de Gestão têm como objetivos integrar os departamentos e processos de uma organização, e têm atualmente ganhado bastante espaço no mercado brasileiro, pois se destacam como ferramentas essenciais para a continuidade das operações das empresas e para o seu gerenciamento.

Nos planejamentos de um sistema de *ERP* consiste em inúmeras ferramentas utilizadas para a criação do mesmo, o que sem algumas delas seria um trabalho árduo para a customização deles.

### **2.1 PHP – História e Conceitos**

Segundo Lopes, o *PHP*, que significa "*PHP: Hypertext Preprocessor*", é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a Web e pode ser mesclada dentro do código *HTML*. A sintaxe da linguagem lembra C, Java e Perl, e é fácil de aprender. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente rapidamente, mas você pode fazer muito mais do que isso com *PHP*.

A versão de 1997, *PHP/FI 2*, teve algum sucesso e contava com 50.000 domínios com a mesma instalada, sendo na altura 1% dos domínios existentes na Internet. Em 2002, o *PHP* já

estava em 60% dos servidores a nível mundial, e é hoje incontestável ser a tecnologia dominante para programação Internet. A versão 3 do *PHP* já era uma aplicação criada por vários programadores, entre os quais os responsáveis do presente motor Zend que são agora quem gere diretamente o projeto. A versão 4 foi a grande viagem e que introduziu a linguagem em uso exponencial até aos dias de hoje. E fizeram uma nova revolução que está a “abandar” com os sistemas proprietários e outras tecnologias para desenvolvimento de aplicações para Internet, criando a versão 5, o *PHP5*. (VIEIRA, 2004).

```
1 <html>
2 <body>
3 <?php
4     echo "Olá mundo";
5 ?>
6 </body>
7 </html>
```

Figura 1 - Estrutura de um código *PHP* Fonte: Autor da Pesquisa

Conforme a figura 1 demonstra como é a estrutura na hora da codificação de um código em *PHP* fazendo o famoso Olá Mundo em que todos os iniciantes em programação já fizeram.

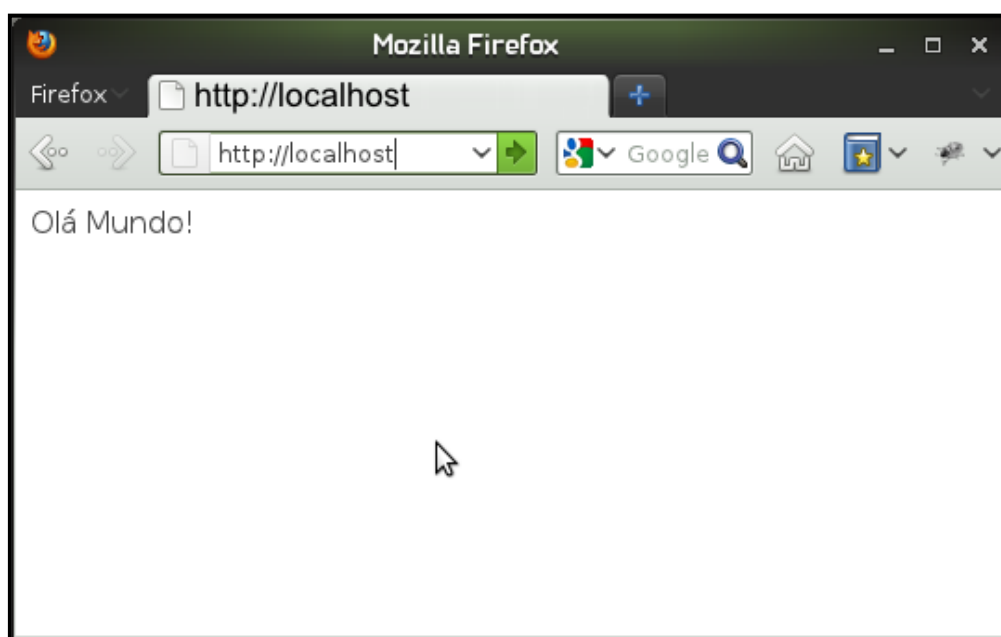


Figura 2 Aparências de um código em *PHP*. Fonte: Autor da Pesquisa

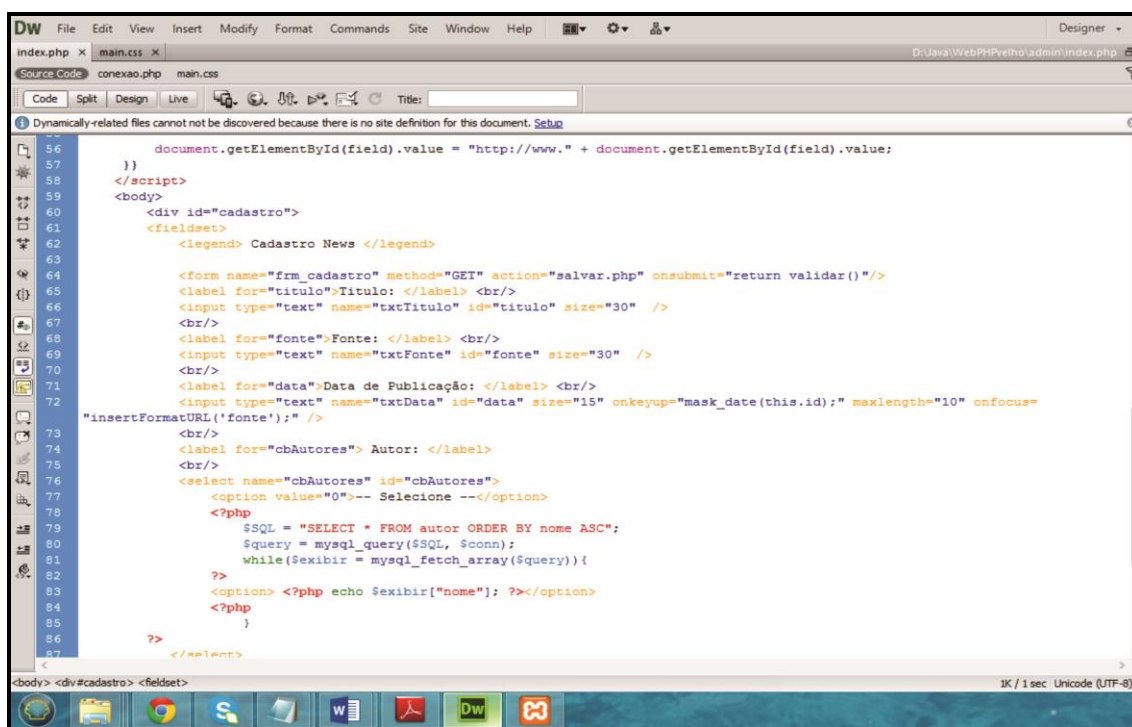
Conforme figura 2 demonstra como fica a visualização após processo de codificação.

## 2.2 HTML

### 2.2.1 Definição

Para Alvarez (2009), o *HTML* é a linguagem usada pelos navegadores para mostrar as páginas web ao usuário permitindo aglutinar textos, imagens e áudio combinando-os de qualquer gosto. Um documento em *HTML* deve funcionar em qualquer ambiente e plataforma, inclusive atendendo a requisitos de portabilidade e acessibilidade.

“A *HTML* é uma linguagem de marcação: um conjunto de indicadores (chamados de elementos) que define o texto e os objetos do documento e que pode ser incorporado ao seu processador de texto favorito.” (MARCON; NEVES, 1999, p.8).

The image shows a screenshot of a web browser window displaying HTML code. The code is for a registration form titled 'Cadastro News'. It includes a JavaScript function to prepend 'http://www.' to the URL, followed by form fields for 'Titulo', 'Fonte', and 'Data de Publicação'. There is also a dropdown menu for 'Autor' populated with a PHP query that fetches names from a database table named 'autor'. The browser's status bar at the bottom indicates '1K / 1 sec Unicode (UTF-8)'.

```
56 document.getElementById(field).value = "http://www." + document.getElementById(field).value;
57 }
58 </script>
59 <body>
60 <div id="cadastro">
61 <fieldset>
62 <legend> Cadastro News </legend>
63
64 <form name="frm_cadastro" method="GET" action="salvar.php" onsubmit="return validar()"/>
65 <label for="titulo">Titulo: </label> <br/>
66 <input type="text" name="txtTitulo" id="titulo" size="30" />
67 <br/>
68 <label for="fonte">Fonte: </label> <br/>
69 <input type="text" name="txtFonte" id="fonte" size="30" />
70 <br/>
71 <label for="data">Data de Publicação: </label> <br/>
72 <input type="text" name="txtData" id="data" size="15" onkeyup="mask_date(this.id);" maxlength="10" onfocus="
73 insertFormatURL('fonte');" />
74 <br/>
75 <label for="cbAutores"> Autor: </label>
76 <select name="cbAutores" id="cbAutores">
77 <option value="0"-- Seleccione --</option>
78 <?php
79 $SQL = "SELECT * FROM autor ORDER BY nome ASC";
80 $query = mysql_query($SQL, $conn);
81 while($sexibir = mysql_fetch_array($query)){
82
83 <option> <?php echo $sexibir["nome"]; ?></option>
84 <?php
85 }
86
87 </select>
```

Figura 3 Códigos em *HTML*. Fonte: Autor da Pesquisa

Conforme figura 3, demonstra um pedaço de um código em *HTML* que por sua vez monta os campos de formulário e um botão para cadastro de informações.

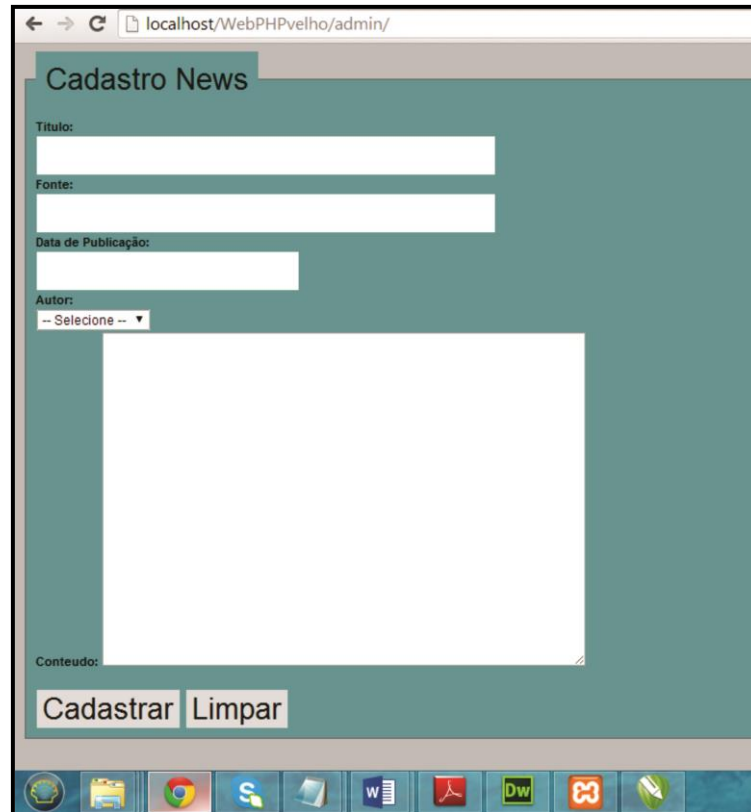


Figura 4 Aparências após código de *HTML*. Fonte: Autor da Pesquisa

Conforme figura 4, demonstra o código feito anteriormente onde consiste em campos de textos, listas e botões para prosseguir com as funções respectivas.

## 2.3 CSS

Segundo Soares (2006), *CSS* é uma poderosa ferramenta para tornar mais interessante e prazerosa a navegação do usuário ao acessar o site, sendo possível inclusive, ser realizadas modificações de forma dinâmica.

Utilizando *CSS* o desenvolvedor consegue obter grandes benefícios e vantagens, em relação ao layout, que não seriam obtidas utilizando somente os atributos de estilização das tags *HTML*.

O Cascading Style Sheet (*CSS*) “é uma linguagem de estilo que consiste em um mecanismo simples para adicionar estilo a um documento da web. A finalidade da *CSS* é a de retirar do *HTML* toda e qualquer declaração que vise à formatação, a apresentação do



documento. Isto significa dizer que tags da obtenção de espaçamentos não são admitidos ou admitidos com restrições em um projeto Web com CSS” (SILVA, 2008).

## 2.4 JavaScript

*JavaScript* é uma linguagem criada pela Sun MicroSystem Inc. Em parceria com a Netscape, embutida em um código *HTML*. O que distingue o *JavaScript* é sua capacidade de interação plena com a página *HTML*, permitindo que novos pedaços de código *HTML* ou mesmo *JavaScript* sejam adicionados dinamicamente, além de possuir a maioria das estruturas típicas de uma linguagem moderna. (SOARES, 2006, p. 48).

Segundo Goodman (2001), *JavaScript* é uma linguagem que permite injetar lógica em páginas escritas em *HTML*. Os parágrafos de lógica do *JavaScript* podem estar "soltos" ou atrelados a ocorrência de eventos. Os parágrafos soltos são executados na sequência em que aparecem na página (documento) e os atrelados a eventos são executados apenas quando o evento ocorre. Para inserir parágrafos de programação dentro do *HTML* é necessário identificar o início e o fim do set de *JavaScript*, da seguinte forma:

```
<SCRIPT>
```

```
Instruções ...
```

```
</SCRIPT>
```

No *JavaScript* é permitida a programação de pequenos scripts, mas também de programas maiores, orientados a objetos, com funções e até estruturas de dados complexas. Permitindo que o programador o atualize de forma dinâmica.

Um script *JavaScript* permite que sejam executadas funções que antes, obrigatoriamente, deviam ser processadas pelo servidor, como validação de campos, abertura de novas janelas, criação e manipulação de camadas, cálculos simples e complexos, etc. (SOARES, 2006, p. 33).

## 2.5 Banco de Dados MySQL

A tecnologia de banco de dados tem como fundamento básico permitir que os dados possam ser definidos e mantidos, independentes dos sistemas de aplicação que venham a utilizá-los (independência de dado X processo) (MACHADO, 2006).

O banco de dados MySQL tornou-se banco de dados do mundo open source mais popular por causa de seu alto desempenho, alta confiabilidade e facilidade de uso chegando a marca de mais de 10 milhões de instalações pelo mundo.

“Bancos de Dados são estruturas em que os dados são armazenados para futuras consultas. Essas estruturas organizam os dados e oferecem recursos de busca que facilitam a obtenção e a manutenção desses dados.” (MARCON;NEVES, 1999, p. 233).

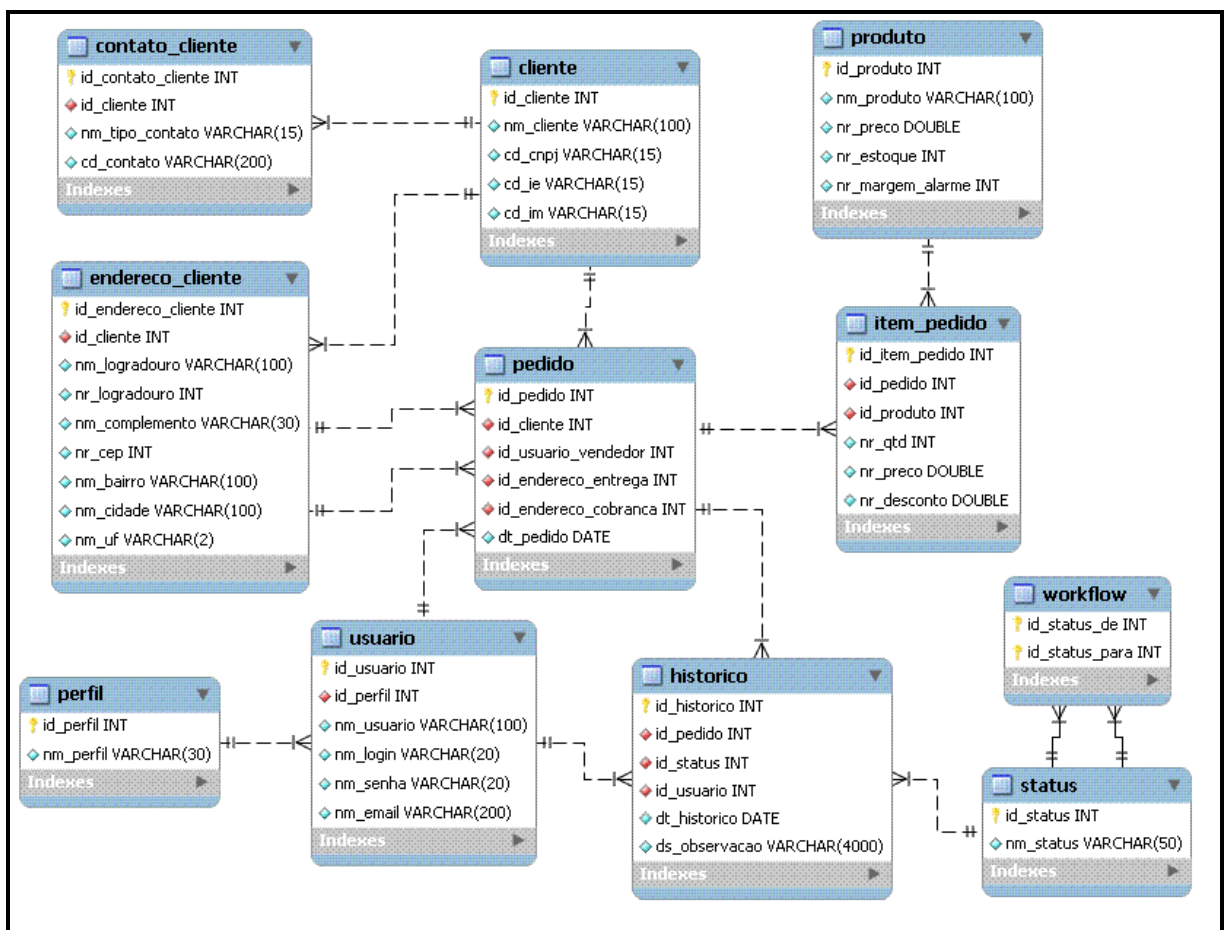


Figura 5 Relacionamento do Banco de Dados MySQL6. Fonte: Autor da Pesquisa

Conforme figura 5, demonstra como funciona o relacionamento entre tabelas dentro de um banco de dados através de suas chaves primarias e chaves estrangeiras.

## 2.6 Adobe Dreamweaver

O Adobe Dreamweaver é um programa para modelar páginas da internet, contando tanto com recursos para programadores (com base em sintaxe e codificação) quanto para designers, com a parte visual largamente desenvolvida e aperfeiçoada (ADOBE SYSTEMS INCORPORATED, 2012).

A interface do programa é simples e intuitiva. Isto facilita a utilização para quem já conhece esse tipo de ferramenta. No entanto, é uma ferramenta avançada e exige certo grau de experiência com a mesma e com as linguagens envolvidas no desenvolvimento de um website.

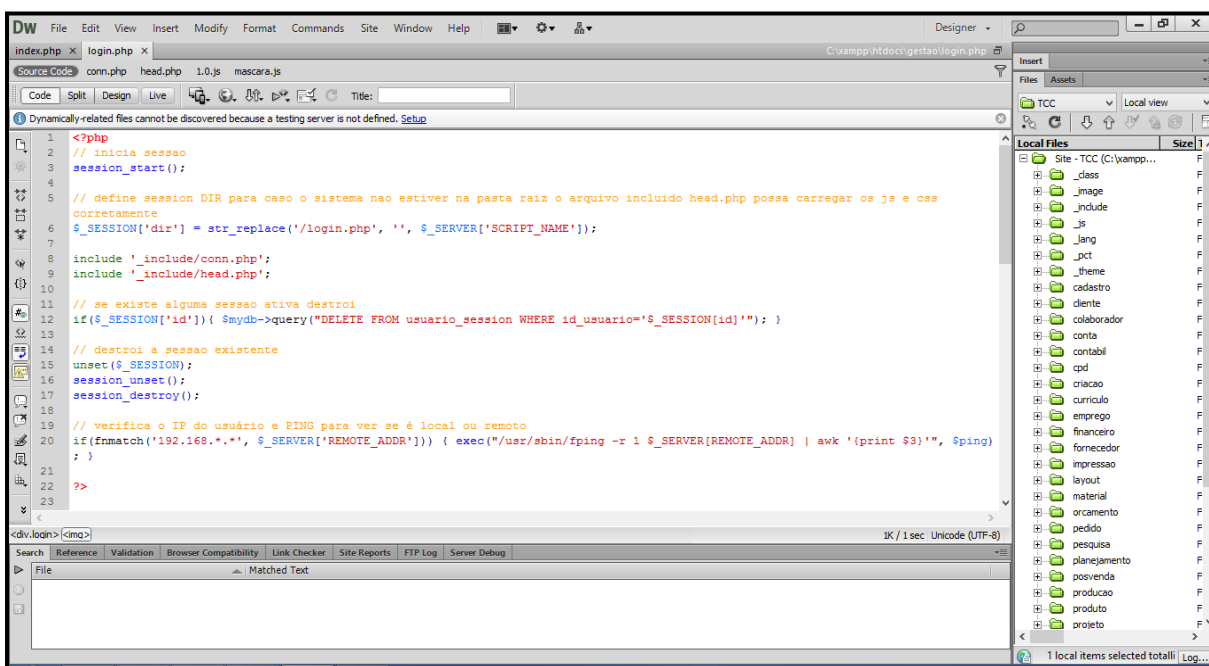


Figura 6 - Interface do Adobe *Dreamweaver* CS5 Fonte: Autor da pesquisa

Conforme figura 6, é representado a interface da ferramenta da empresa Adobe, onde ao lado direito, encontra-se todos os seus projetos, classes e métodos utilizados.

## 2.7 Sistema de Informação

De acordo com Laudon (2001), Sistema de Informações (SI) é definido como um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações com a finalidade de dar suporte à tomada de decisões e controle em uma organização. Para os autores, um SI pode também auxiliar gerentes e trabalhadores a analisarem problemas, a visualizarem formas complexas e a criarem novos produtos. Sob um enfoque empresarial, os SI são uma solução organizacional e gerencial, baseada em tecnologia da informação, em resposta a um desafio apresentado pelo meio ambiente.

Os sistemas de informações são importantes para as organizações, primeiramente por permitirem uma melhor agilidade no repasse das informações e conseqüentemente por possibilitar maiores acertos nas tomadas de decisões dos administradores. (BATISTA, 2004).

“Sistemas e tecnologias de informação tornam-se componentes vitais quando se pretende alcançar o sucesso de empresas e organizações [...]”. (O’BRIEN; 2004, p. 3)

Sob uma visão mais contemporânea, os sistemas de informação devem ser vistos como parte fundamental dos processos dentro de uma organização, pois seu auxílio é em todas as etapas dos processos, seja fornecendo entradas, na transformação do produto ou serviço e na produção de saídas. (STAIR E REYNOLDS, 2011)

A Figura 7 ilustra esta classificação conceitual dos sistemas de informação citadas pelo autor.

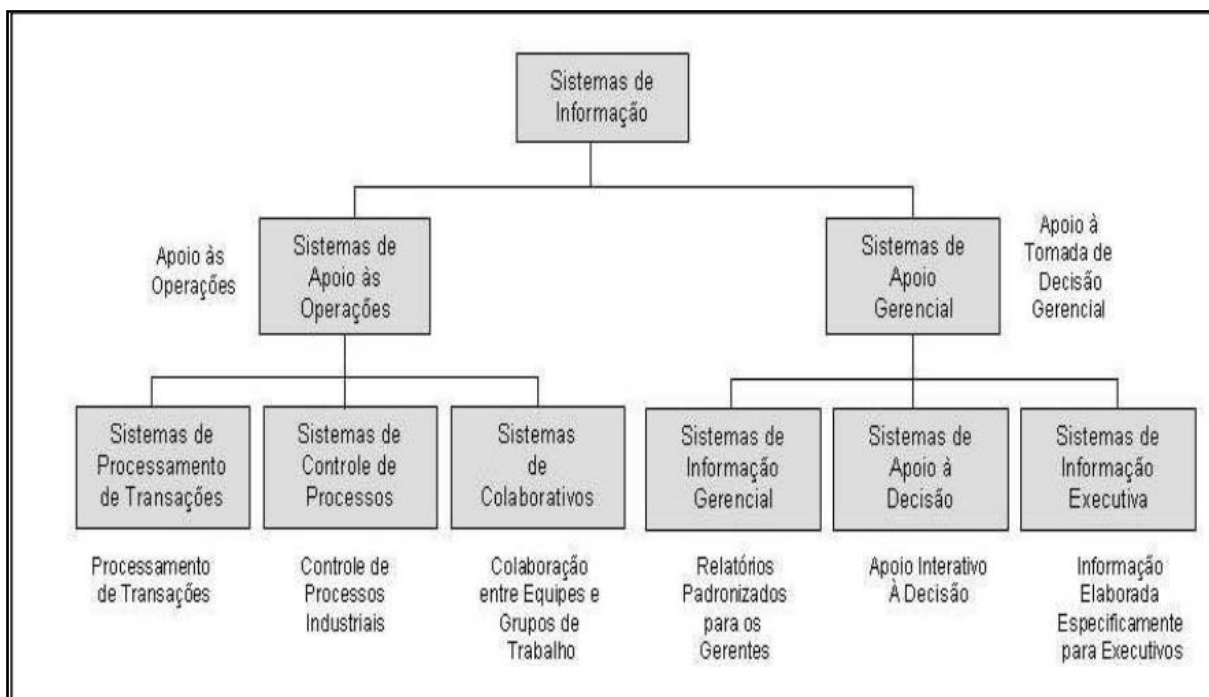


Figura 7 – Tipos de Sistemas de Informação Fonte: O’BRIEN (2001. p. 28)

Conforme ilustra a figura 7, os SI podem ser primeiramente subdivididos em: i) Sistemas de Apoio às Operações, que monitoram e processam as funções básicas e rotineiras de uma organização, tais como processamento da folha de pagamento, faturamento, estoque, entre outros e ii) Sistemas de Apoio Gerencial, fornecem informações pertinentes e necessárias à tomada de decisões em problemas específicos.

### 2.7.1 Sistema de Informações Gerenciais (SIG)

As iterações que ocorrem dentro das organizações dependem das informações que nascem de um sistema. Para Oliveira (1996), o sistema de informações gerenciais é um sistema voltado para a coleta, armazenagem, recuperação e processamento de informação que é usado ou desejado, por um ou mais executivos no desempenho de suas atividades.

Destaca Cruz (1998) que os sistemas de informações gerenciais são o conjunto de tecnologias que disponibiliza os meios necessários à operação do processo decisório em qualquer organização por meio do processamento dos dados disponíveis.

Há o objetivo de coletar, armazenar, recuperar, e processar estas informações para que sejam utilizadas por executivos no desempenho de suas atividades.

“Os sistemas de informação da empresa devem ser configurados de forma a atender eficientemente às necessidades informativas de seus usuários, bem como incorporar conceitos, políticas e procedimentos que motivem e estimulem o gestor a tomar as melhores decisões para a empresa “ (GUERREIRO apud MOSIMANN; FISCH, 1989).

Batista (2004) destaca que o sistema de informações gerenciais (SIG) oferece um conjunto de relatórios resumidos sobre o desempenho da empresa, os quais são utilizados para a realimentação do planejamento operacional.

Esse sistema permite a utilização de relatórios, consultas e visualização dos dados, que são ferramentas incorporadas a algum tipo de gerenciador de banco de dados.

## **2.8 Url “Amigável”**

A URL tem a função de endereçar um determinado recurso na internet, como um site ou um FTP. URLs grandes ou complicadas demais são difíceis de memorizar e podem até mesmo dificultar a indexação do site por mecanismos de busca. Por isso, é importante que seu site tenha URLs amigáveis, indexáveis e que resumam, de alguma forma, o recurso que elas descrevem.

Para Camargo é possível vermos em diversos sites e portais endereços de url que não são tão fáceis de decorar devido ao tamanho e, a muitas vezes, inclusão de números e valores que a deixam complexa, difícil de memorizar e sem sentido (pensando no usuário comum, é claro).

Para resolver este problema existem as urls amigáveis. Com elas conseguimos transformar uma url complexa em uma url fácil de entender e memorizar.

Um exemplo simples: uma página de detalhe de um determinado produto, que teria a seguinte url:

<http://www.site.com.br/DetalheProduto.aspx?IdProduto=12345>

Transformar-se-ia facilmente em:

<http://www.site.com.br/Produto>

## 2.9 Desing Responsivo

Visto como a internet evolui de forma exponencial, e as tendências do mundo moderno também são afetadas pelo seu crescimento, os modelos de layout também sofrem mudanças conforme essas mudanças acontecem. E graças à geração onde a rede conecta sites em desktop, smartphones e tablets de um mesmo recurso, o modelo responsivo foi criado para atender às necessidades de quem possui os dispositivos móveis e utiliza o computador de mesa.

“Porque mais do que qualquer coisa, web design é de perguntar as perguntas certas. E realmente, isso é o que web design responsivo é: uma possível solução, uma maneira de projetar mais plenamente para a flexibilidade inerente da web”. (MARCOTTE, 2011, p.138, tradução).

O design responsivo é a forma como a estrutura da Web vai se comportar em determinado tipo de dispositivo. Usando de bibliotecas de JQuery e comandos CSS, é possível redimensionar a página web de modo a adequar o seu conteúdo em diversas telas de computadores, tablets e smartphones, ajustando sua resolução conforme o dispositivo utilizado.

Leva-se em consideração também o estilo aplicado com o CSS para atingir os objetivos do recurso responsivo, uma vez que haja esse suporte diretamente na parte gráfica do site.

## 3 ERP

Não há uma definição totalmente precisa e inquestionável do que venha a ser um sistema *ERP* (Enterprise Resources Planning). Nesse sentido, muitos autores expressam suas definições a respeito do que venha a ser um sistema *ERP*.

Segundo Filho (2001, p.17), o *ERP* é um software aplicativo que permite às empresas automatizar e integrar parcela substancial de seus processos de negócios, abrangendo finanças, controles, logística (suprimentos, fabricação e vendas) e recursos humanos; compartilhar dados e uniformizar processos de negócios; produzir e utilizar informações em tempo real.

Para Tamae et al (2005), se as empresas querem desfrutar de recursos tecnológicos, precisam de um sistema de gestão eficiente que torne a informação clara e precisa,

adequando-se a realidade de mercado com uma comunicação cada vez mais rápida. O *ERP* agrega em um único sistema funcionalidades que suportam as atividades dos diversos processos e negócios, tais como módulos de Gerenciamento Recursos Humanos, Vendas, Distribuição, Finanças, Controladoria e outros. A integração torna-se possível através do compartilhamento de informações entre os módulos que acessam um banco de dados centralizado.

Já Araújo et al (2003, p.02) diz que a sigla *ERP* (Enterprise Resource Planning) significa “planejamento dos recursos da empresa”. É um sistema que controla e fornece o suporte a todos os processos operacionais, produtivos, administrativos e comerciais da empresa, facilitando o fluxo de informações único, contínuo e consistente por toda a empresa, sob uma única base de dados. É uma ferramenta para melhores processos de negócios, como produção, compras ou distribuição, com informações on-line em tempo real.

Em termos gerais, o sistema *ERP* é uma plataforma de software desenvolvida para possibilitar o armazenamento de todas as informações de negócios, visando integrar os diversos departamentos da empresa.

### **3.1 Como funciona um *ERP***

O sistema integrado coleta dados dos principais processos de negócios e os armazenam em apenas um banco de dados, que pode ser utilizado por todos os setores da empresa. Com as informações fornecidas, torna-se mais fácil o gerenciamento e a coordenação das operações diárias da empresa, possibilitando que se tenha uma visão mais ampla dos negócios. O *ERP* propicia uma maior confiabilidade dos dados, sendo estes monitorados em tempo real.

Segundo Anthony e Govindarajan (2002), o controle gerencial engloba também o controle das informações da empresa, ou seja, controle de estoque, compras, funcionários, produção, entre outras. Tais informações auxiliam no planejamento e coordenação da empresa e facilitam na tomada de decisões.

Aplicativos do tipo *ERP* ajudam a modelar e automatizar muitos processos de negócios, fazendo com que os diferentes setores da empresa troquem informações. Por exemplo, o setor de vendas pode verificar instantaneamente, ao receber um pedido, se há estoque dos itens pedidos, os clientes podem acompanhar o andamento do seu pedido e o setor de produção comunica-se com finanças para planejar uma nova produção.



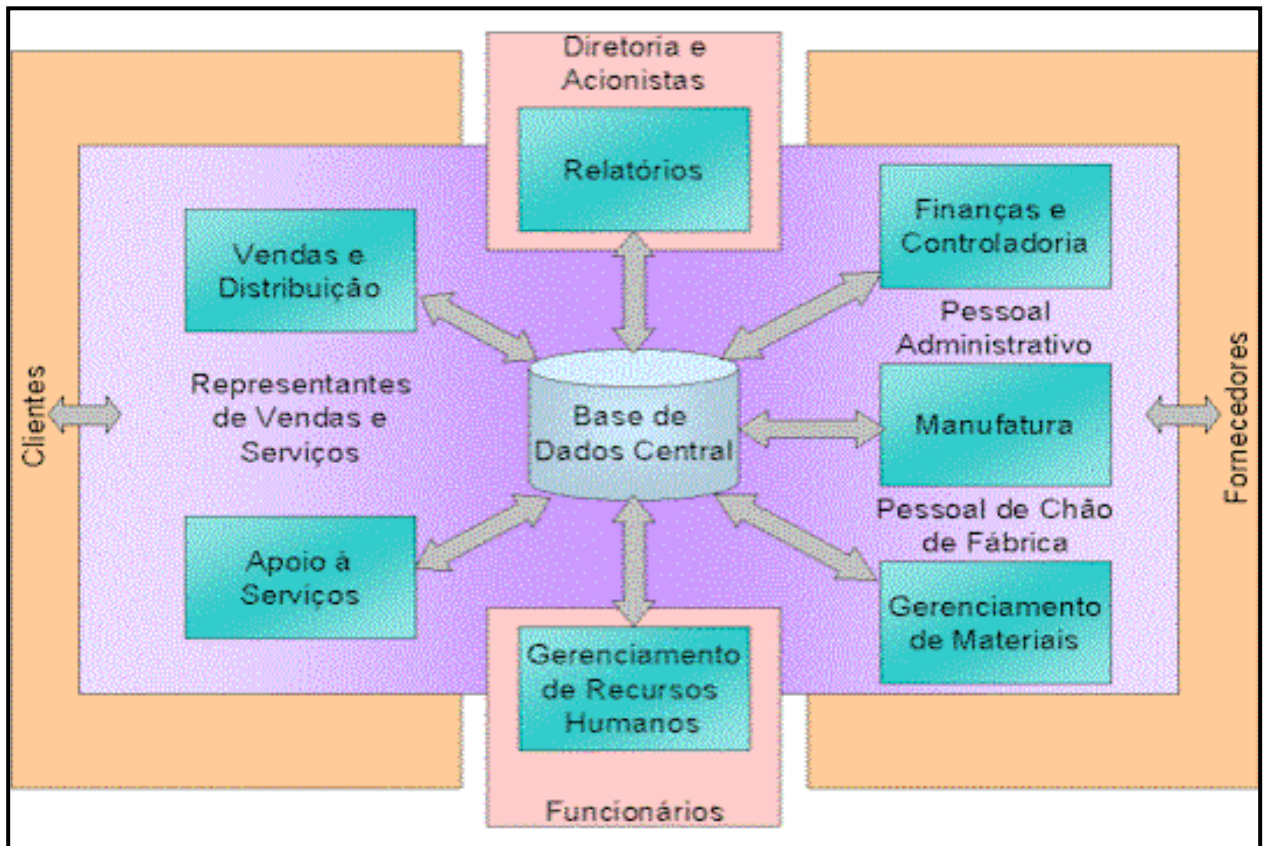


Figura 8 – Como Funciona um ERP Fonte: LAUDON, Kenneth C. Sistemas de Informações Gerenciais: administrando a empresa digital. São Paulo: Prentice Hall, 2004.

De acordo com Haberkon (2008), nesse tipo de controle, com a utilização de um ERP, os dados são acessados de forma rápida e os relatórios gerados são bem mais confiáveis, pois as informações são transmitidas em tempo real para todos os setores de uma empresa, possibilitando a rápida divulgação de resultados e tomadas de decisões.

### 3.1.1 O surgimento do ERP

Anthony e Govindarajan (2002) descrevem o histórico desde os primeiros computadores até o surgimento do ERP. Segundo os referidos autores, no final da década de 1950 os conceitos modernos de controle tecnológico e gestão corporativa tiveram seu início. Até este momento a tecnologia dos computadores era baseada em gigantescas máquinas, utilizadas por pouquíssimas pessoas, devido ao alto preço e a dificuldade na operação.

Anthony e Govindarajan (2002) ainda afirmam que a difusão do uso de computadores nos processos e suporte de negócios teve seu maior crescimento já em 1960. Nessa época, os

equipamentos eram bastante lentos, caros e sua atuação era bastante limitada devido à baixa tecnologia. Não havia extensa oferta de softwares, o que obrigava as empresas a desenvolverem formas de utilizar os computadores existentes, principalmente no controle efetivo do estoque, que era uma das principais preocupações dos empreendedores.

Continuando o decorrer histórico, Anthony e Govindarajan (2002) comentam que já na década de 1970, houve um desenvolvimento maior na informática, surgindo assim computadores modernos e eficientes. Esse desenvolvimento nas máquinas propiciou a criação dos sistemas MRP (Materials Requirements Planning), que efetuavam controle de estoque, controle de produção e compras. Tais sistemas apresentavam como maior problema a não integração com os demais dados e aplicações da empresa.

### **3.2 Controle gerencial manual x *ERP***

Conforme Netto (2008), o controle gerencial manual é uma forma cada vez mais incomum de se administrar e é feito a caneta em livros contábeis, cadernos e papéis comuns. Este tipo de controle é pouco confiável e existem grandes possibilidades de erros e conseqüentemente atrasos nas informações, podendo causar um retardo ou interrupção nas tomadas de decisões, devido à falta de relatórios confiáveis em tempo hábil.

Para Netto (2008), utilizando o controle manual também há riscos de perda do material preenchido, rasuras, fraudes e outras situações que podem resultar em relatórios não confiáveis, além de não permitir o sequenciamento instantâneo de todos os departamentos de uma empresa.

Essa abordagem integradora (*ERP*) pode dar um grande retorno financeiro se as companhias instalarem o software adequadamente. Pegue o pedido de um cliente como exemplo: tipicamente, quando um cliente faz um pedido, aquele pedido começa uma jornada em papel, de um lado para outro na empresa, sendo digitado e redigitado em vários computadores ao longo do caminho. Toda essa jornada causa atrasos e perdas de pedidos, e cada digitação, em um diferente sistema, é convidativo a erros. Ao mesmo tempo, nenhuma companhia sabe realmente em que estágio um pedido se encontra em um determinado momento porque não há como o departamento financeiro, por exemplo, entrar no computador do depósito para ver se o item foi embarcado. “Você terá que ligar para o depósito”, é a resposta familiar dada aos frustrados consumidores. (NETTO, 2008.)

### **3.3 Vantagens do ERP (Enterprise Resource Planning)**

Tamae, et al, (2005), afirma que o *ERP* é um sistema integrado, de forma que oferece possibilidade de melhores relatórios, fidelidade de dados, consistência e comparação de dados, pois utiliza um critério único em todas as atividades da empresa. Com os vários departamentos da empresa utilizando o sistema integrado e compartilhando a mesma base de dado, não há necessidade de repetição de atividades, como a reentrada de dados de um aplicativo para o outro.

Segundo Tamae, et al, (2005), a rápida tomada de decisões também é possível graças ao *ERP*, resultando em domínio sobre as operações e economia de tempo. Para o mercado atual, agilidade e poder de resposta rápida são importantíssimos para atender as exigências dos clientes. Outra vantagem do sistema *ERP* é a sua flexibilidade.

De acordo com Paulino et al (2007), os benefícios de utilizar o *ERP* são o processamento automatizado de salários; a redução do número de documentos em papel, possibilitando rápida consulta das informações; a informação é detalhada e vinda de muitas áreas da empresa; é possível controle de custos; gasta-se menor tempo na resposta e assistência ao cliente; a cobrança de dívidas é mais eficiente; há melhor monitoramento do sistema e rápida consulta a base de dados; ajuda no alcance de vantagem em relação a concorrência e permite gerir um processo em qualquer terminal de qualquer área da empresa.

Segundo Paulino et al (2007), outras vantagens ainda podem ser citadas, como a eliminação do uso de recursos manuais, a eliminação de redundância de atividades e a incorporação de melhores práticas aos processos internos da empresa.

Os autores acima salientam ainda que existem certos fatores que são necessários para o sucesso e a conseqüente obtenção de tantas vantagens para a empresa, entre os quais podemos citar a necessidade do envolvimento do usuário, o apoio da direção, a definição clara das necessidades da empresa, o planejamento adequado, as expectativas realistas, a presença de uma equipe competente, o comprometimento, a existência de uma equipe dedicada e a infraestrutura adequada.

## **4 METODOLOGIA**

Neste capítulo apresentado contém as características desta pesquisa, pois a Metodologia é um estudo de métodos ou as etapas a seguir num determinado processo. Refere-se a mais do que um simples conjunto de métodos, mas sim ao que fundamenta um estudo particular e tem como objetivo captar e analisar as características dos vários métodos utilizados.

A metodologia possibilita também explicações sérias e detalhadas, acerca dos procedimentos adotados pelo pesquisador, explicando dessa forma como a pesquisa foi elaborada.

### **4.1 Tipos de Pesquisa**

A pesquisa exploratória, como é o caso, tem como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito. Pode envolver levantamento bibliográfico, entrevistas com pessoas experientes no problema pesquisado. Geralmente, assume a forma de pesquisa bibliográfica.

Segundo Antônio Carlos Gil (2002), a pesquisa científica consiste em um conjunto de ações que objetivam propor soluções para problemas sugeridos por meio de procedimentos racionais e sistemáticos.

De acordo com Jung [2004], trabalhos de pesquisa científica podem ser classificados quanto à sua natureza, quanto aos seus objetivos e seus procedimentos.

A presente pesquisa utilizou as fontes bibliográficas, visto que se baseou em uma revisão bibliográfica sobre o tema.

Segundo Antônio Carlos Gil (2002), a pesquisa bibliográfica é aquela desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos.

### **4.2 Método Utilizado**

A presente pesquisa, possibilitou a análise de pontos relevantes e o esclarecimento sobre a importância do *ERP (Enterprise Resource Planning)*.

Para isso, foi utilizado o método da revisão bibliográfica, ou seja, a análise dos tópicos que compõe o trabalho baseado no que já foi discutido por alguns autores.

Para conceituar o sistema *ERP* e demonstrar sua evolução histórica, se utilizou dos estudos dos autores Araújo et al, além de outros autores, como Ernesto Haberkon (2008) Visando expor a importância e o que vem a ser o controle gerencial, as ideias expostas por Robert N. Anthony e Vijay Govindarajan (2002) foram utilizadas.

Na exposição sobre as vantagens do *ERP* sobre o controle gerencial manual e seus benefícios, foi utilizado o texto de Araújo et al (2005), no qual se discute também algumas considerações envolvendo aspectos referentes aos sistemas *ERP* e suas importâncias.

A explanação a respeito da importância, benefícios e vantagens do sistema *ERP* nas pequenas e médias empresas face ao controle gerencial manual foi realizada com base no texto de Paulino et al (2007).

O texto de Lúcio Colangelo Filho (2001) foi utilizado na abordagem sobre as perspectivas para o sistema *ERP*, no qual demonstra o que se espera desse sistema, que apesar de muito desenvolvido, pode sofrer ainda mais atualizações para satisfazer ainda mais os anseios dos clientes.

## 5 PROJETO

### 5.1 Introdução

A partir dos conhecimentos do referencial teórico, o projeto tende a se concluir utilizando a ferramenta Dreamweaver da Adobe como plataforma de programação de CSS, PHP, HTML e as linguagens de customização, e utilizado o MySQL com banco de dados, para a persistência direta das informações.

Como parte de um sistema Web, o sistema conta com uma abordagem minimalista, porém objetiva para atender às necessidades apresentadas pelos diretores da empresa. Uma vez desenvolvido o padrão, é possível evoluir o sistema em projetos futuros para novas versões, ampliando os módulos presentes, garantido um sistema em constante aprimoramento.

O problema inicial da empresa decorre pois o sistema utilizado atualmente encontra-se com uma certa dificuldade sua manutenibilidade, seguindo de várias atualizações do software devida à diversificação de código no decorrer de sua estrutura, tornando-o mais suscetível à falha e contendo linhas de programação obsoletas ou que entram em conflito ao longo de seu uso.

Tendo isso em mente, é importante observar que para solucionar tal problema, foi necessário a criação de um software independente, que tenha toda a sua estrutura de código revisada e fundamentada pela qualidade desde sua concepção em análises. Após isso, utilizando dos conhecimentos já descritos no desenvolvimento do projeto e usando as ferramentas supracitadas, alcançou-se a criação dos módulos e por final um sistema próprio de *ERP* para a implementação e análises futuras.

Em um primeiro momento, para não ocorrer um impacto tão grande na rotina dos colaboradores, assim podendo ter alguns atritos e rejeição por meio dos mesmos, houve-se um alinhamento que alguns processos utilizados atualmente, não teriam mudanças até que as já implementadas estivessem aprovadas após alguns feedbacks.

Dentre essas funções destacam-se toda a parte referente a folha de pagamento dos funcionários e algumas partes financeiras como emissão de nota fiscal eletrônica, optado para a terceirização desses processos.

## 5.2 Diagramas – UML

Através dos estudos teóricos realizados, foram criados três modelos de diagramas diferentes para demonstrar como seria o funcionamento de alguns possíveis módulos que terá o sistema.

Há ainda a possibilidade de alterações no padrão tanto dos diagramas quanto do projeto em si, pois na hora de criação do sistema pode-se mudar a estrutura pré-projetada adquirindo uma melhor performance. Uma vez que o sistema é web e também tem uma gama de evolução significativa, maiores serão as requisições de uma funcionalidade nova.

### 5.2.1 Diagrama de Caso de Uso

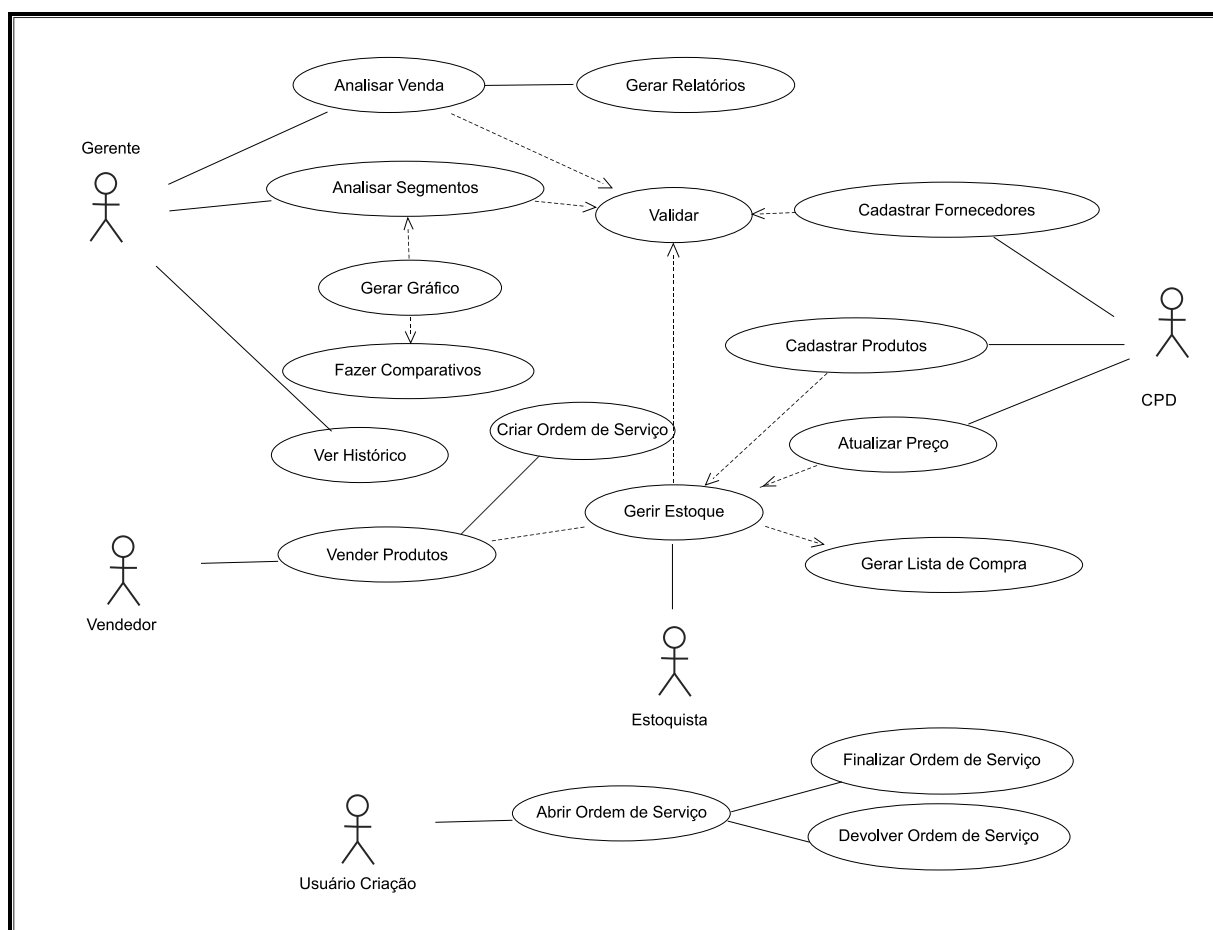


Figura 9 – Diagrama de Caso de Uso Fonte: Autor da Pesquisa

Conforme demonstrado a figura 9, o diagrama demonstra como funciona o módulo de vendas dentro do sistema de *ERP*, o que cada pessoa tem acesso e deveres dentro do mesmo.

Cada usuário com acesso tem sua soma importância dentro deste módulo, para manter o sistema sempre atualizado.

Juntamente encontra-se também o funcionamento do módulo para todos os usuários do departamento de Criação (Desinger Gráfico), onde terá suas OS's (Ordem de Serviços) na qual foi solicitado por algum vendedor, ele poderá abri-la, e conseqüentemente finaliza-la ou de volta-la caso necessite de mais informações.

### 5.2.2 Diagrama de Componentes

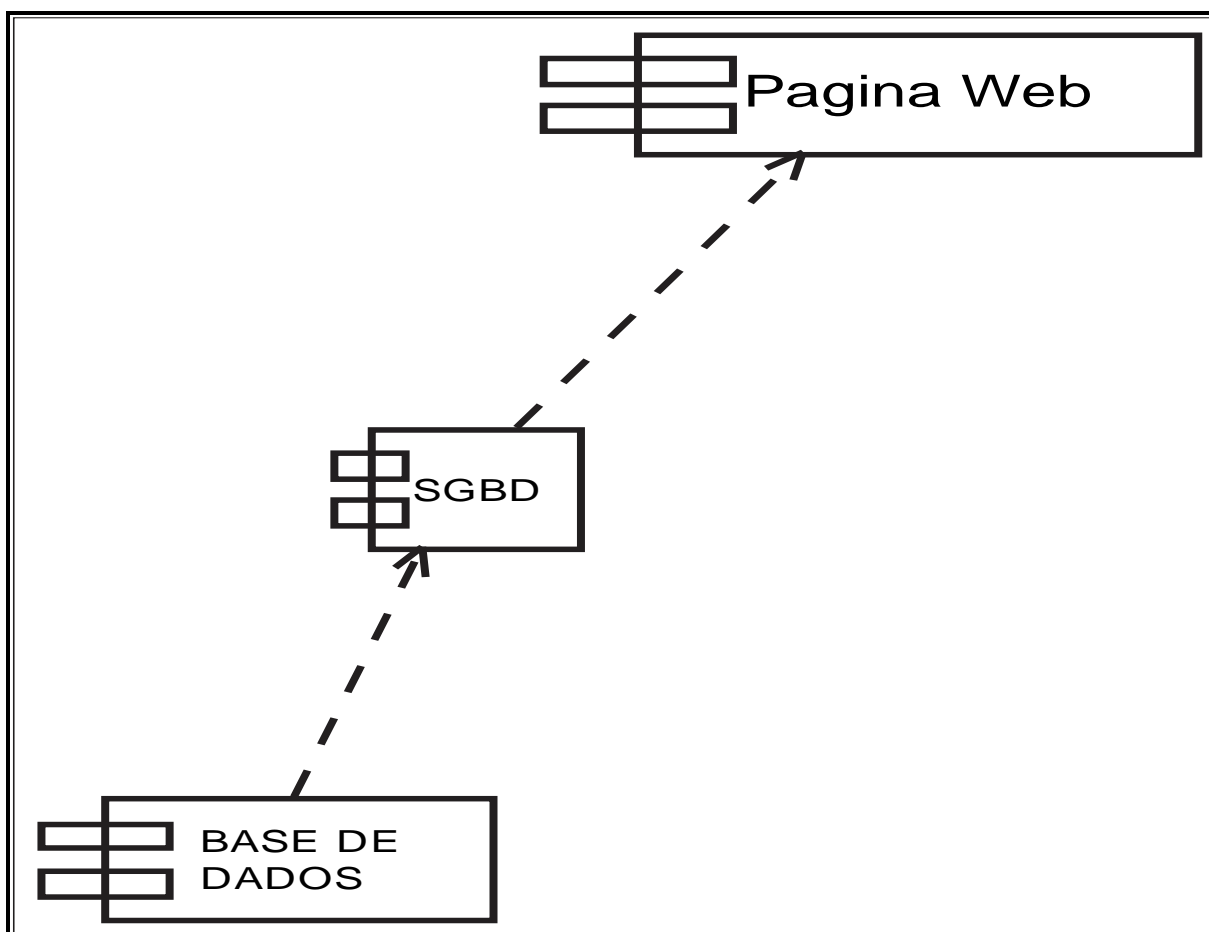


Figura 10 – Diagrama de Componentes Fonte: Autor da Pesquisa

Na figura 10, é demonstrado o fluxo de componentes no projeto, e como ele vai interagindo com cada atividade do sistema. A partir dele, pode-se observar como que o sistema se comportará de maneira ampla, e qual o próximo procedimento que deverá percorrer.

Partindo da interface web, onde o usuário insere dados de busca, cadastro e etc, ele é enviado através da estrutura de dados para o sistema, ainda na camada de interação com o



usuário. Então ele passa para a camada de persistência de dados, no servidor de banco de dados, que devolve os dados de volta para o sistema na rede.

### 5.2.3 Diagrama de Classe

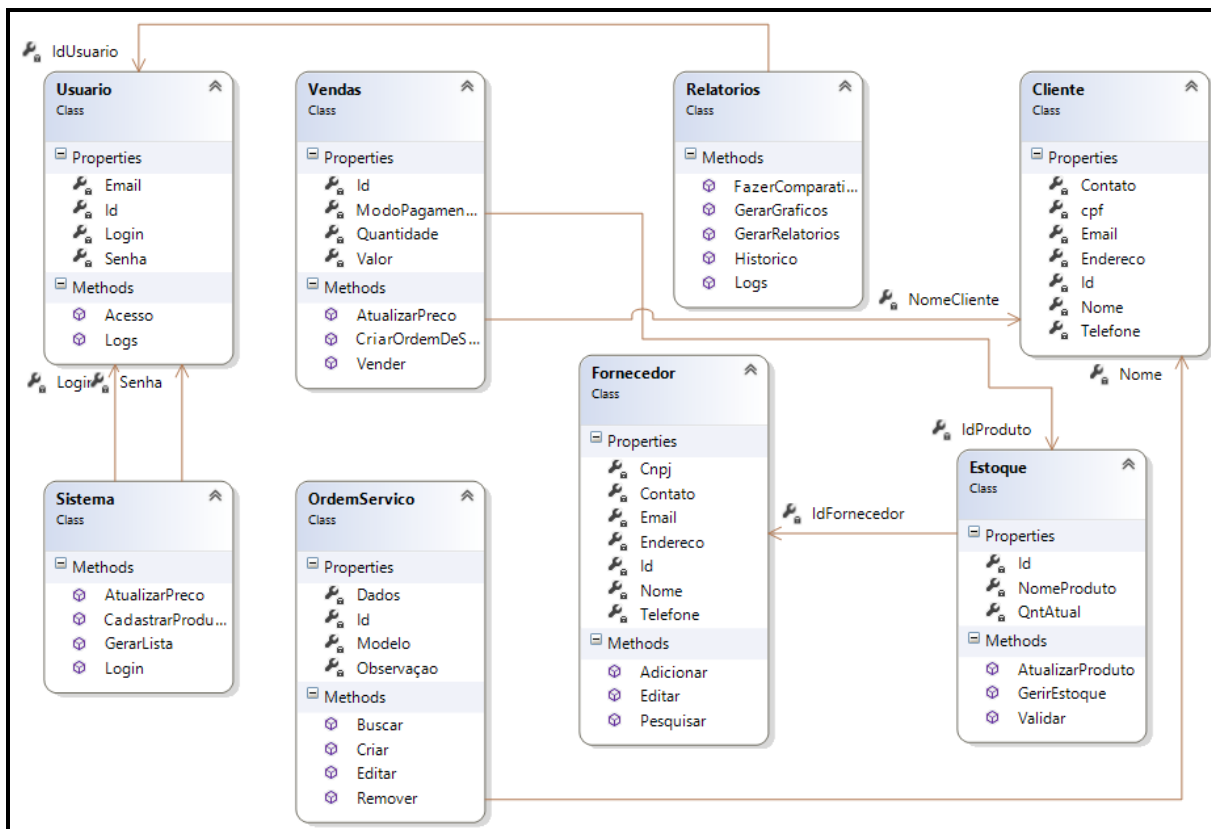


Figura 11 – Diagrama de Classe Fonte: Autor da Pesquisa

Na figura 11, demonstra o diagrama de classe, na qual demonstra os casos de relacionamento de tabelas dentro do banco de dados, ficando como critério a interdependência de chaves primarias e chaves estrangeiras para um bom funcionamento futuro do sistema.

Esse diagrama foi projetado nas possíveis formar de agir mais básicas conforme estudado nos referenciais teóricos.

Neste diagrama poderá ocorrer alterações até o término do projeto, pois será programado módulos específicos conforme prioridade de usabilidade da empresa.

## 5.2.4 Diagrama de Implantação

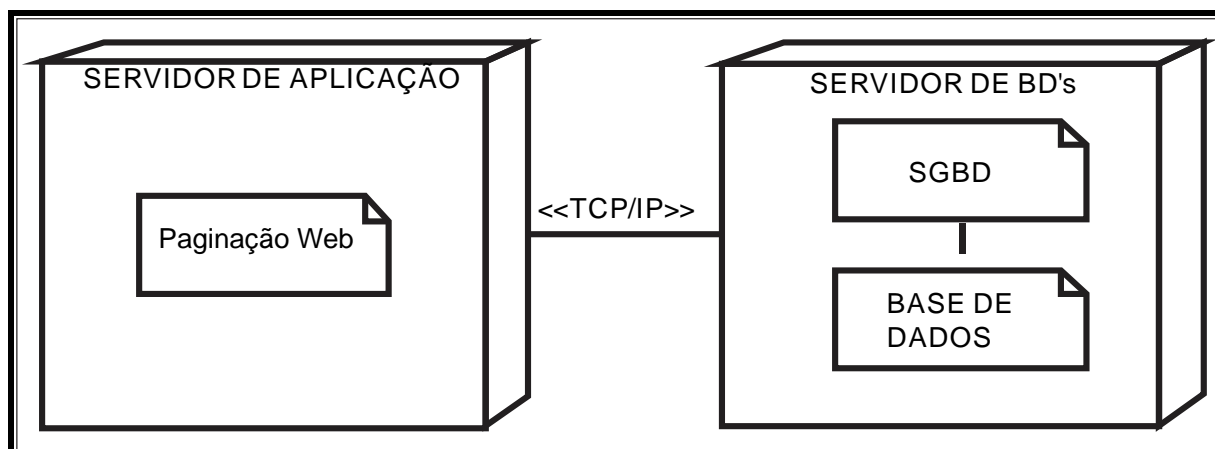


Figura 12 – Diagrama de Implantação Fonte: Autor da Pesquisa

Como demonstrado na figura 12, a implantação física é mínima, e a sua semelhante virtual nada mais é do que uma conexão entre o servidor e os dados de interface. De maneira simples, a imagem representa como o sistema é implementado como um todo, e de certo modo, tem uma forma simples e direta de aplicar suas funcionalidades.

## 5.3 Design de Telas do Sistema

Tendo como base os estudos bibliográficos citados, mas como observação que cada tela que será demonstrada no decorrer pode haver alterações futuras, pois se trata de um pré-projeto, assim para um melhoramento pode-se concluir essas alterações.



Figura 13 – Tela de Confirmação de Usuário e Senha Fonte: Autor da Pesquisa

Conforme figura 13, é demonstrada a página de confirmação de usuário e senha do sistema. Em um molde simples, onde o usuário pode entrar com sua conta, a criação de usuário fica limitado para apenas o administrador do sistema.

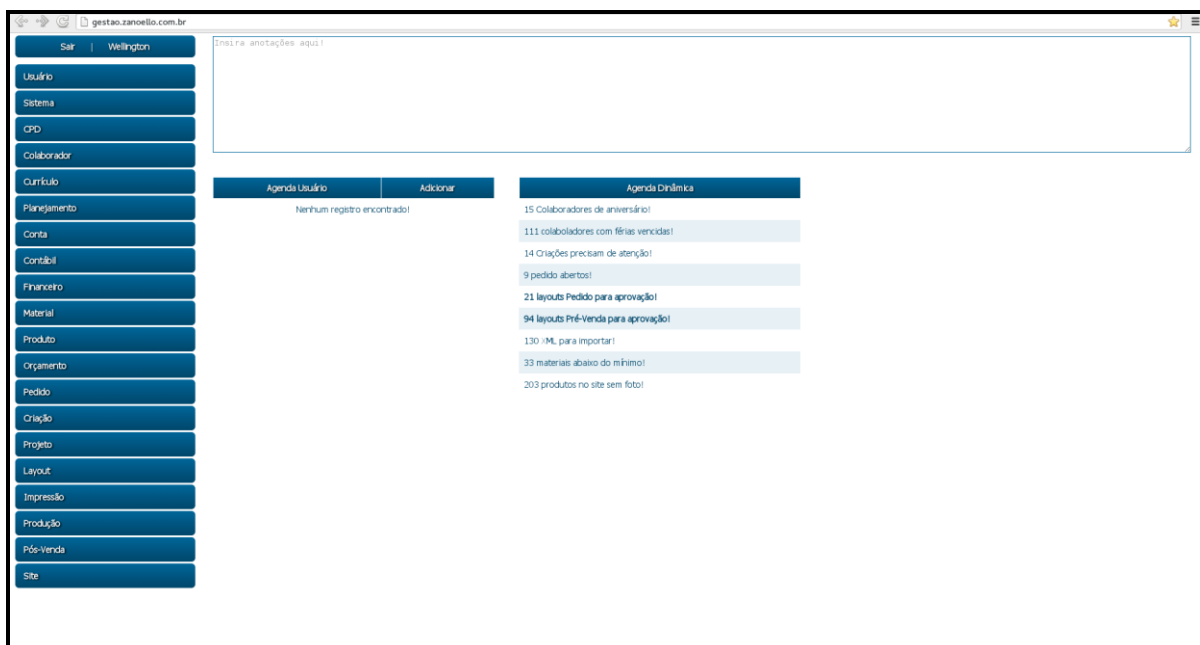


Figura 14 – Tela Inicial Fonte: Autor da Pesquisa

Conforme a figura 14, é possível notar que se trata da tela inicial do projeto, demonstrando os posicionamentos dos menus com seus módulos consequentemente.

Na parte superior encontra-se um campo de anotação para que o usuário possa escrever suas anotações centralizando assim suas informações do dia-a-dia, assim evitando a possibilidade de perda de seus arquivos separados.

No lado esquerdo da imagem encontra-se o menu com os módulos, cada modulo será apresentado conforme o nível de privilégio estabelecido a cada usuário.

Está imagem consta também um espaço para agenda, para que o usuário coloque todos seus compromissos, possibilitando que o sistema lembre-o quando estiver aproximando-se da data.



Figura 15 – Tela Inicial Responsiva Fonte: Autor da Pesquisa

Na figura 15, pode-se notar como a tela inicial se comporta quando em resoluções menores, tal como celulares. Graças à codificação do CSS3, é possível modelar a estrutura do HTML com maior facilidade, podendo manter a resolução normal, adapta-la para versões menores, ou passar por um processo de customização.

Neste caso o sistema torna-se em botões até a paginação do conteúdo.

Saír   Wellington		Criação						
Usuário	Sistema	Cod	Cliente	Criação	Carteira	Executor	Pontos	Status
		933	Roberta Couto Arruda	02/05/2014	Cart. 01	Willian	95	Produção
		939	Fernanda	05/05/2014	Cart. 03	Ricardo	475	Produção
		940	Associação Moto Clube Tribo da Trilha	05/05/2014	Cart. 03	Willian	95	Produção
		943	Clube Floresta	05/05/2014	Cart. 01	Willian	265	Produção
		944	Missão Shinkai	05/05/2014	Cart. 05	Ana	75	Avaliação
		945	Federação Bahiana de Tênis	05/05/2014	Cart. 01	Giovanna	250	Produção
		946	Funerária Timboteua Ltda ME	05/05/2014	Cart. 03	Lutz	125	Avaliação
		947	Wendel da Silveira Filizetti	05/05/2014	Cart. 06	Ana	95	Produção
		948	Spring Race	05/05/2014	Cart. 02	Willian	250	Produção
		949	Federação Capixaba de Corrida de Aventura ...	05/05/2014	Cart. 06	Giovanna	315	Produção
		950	Lilian Ferreira	05/05/2014	Cart. 05	Ana	80	Produção
		952	Liga Nacional de Kung Fu	05/05/2014	Cart. 05		80	Setor
		953	Shell	05/05/2014	Cart. 01	Lutz	95	Produção
		938	Embramac	06/05/2014	Cart. 02		125	Setor
		352	Raul Sanchez	06/05/2014	Cart. 05		35	Setor
		865	Abilio Cesar Gomes de Castro	06/05/2014	Cart. 05		35	Setor
		924	Federação de Jujitsu do Ms	06/05/2014	Cart. 05		35	Setor
		951	Osmar Sampaio de Almeida Junior	06/05/2014	Cart. 03		520	Setor

Figura 16– Modulo Criação Fonte: Autor da Pesquisa

Conforme figura 16, demonstra o modulo de gerenciamento de ordem de serviços para o setor de criação, onde usuários como vendedores, pode gerar ordem de serviço ao departamento de criação.

Quando chega ao departamento de criação, o gerente passa a ter acesso a ordem criada direcionando qual colaborador de sua equipe irá executar este trabalho e a sua prioridade.

Saír   Wellington		Visualizar Criação																									
		<div style="text-align: right;"> <a href="#">Retornar</a> <a href="#">Dividir</a> <a href="#">Avaliar</a> <a href="#">Registar</a> </div>																									
		<div style="text-align: right;"> <a href="#">DESCRIPÇÃO</a> <a href="#">Alteração</a> </div>																									
		<table border="1"> <thead> <tr> <th>Altura cm</th> <th>Largura cm</th> <th>Qtd</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>25</td> <td>0</td> <td>8</td> <td>50,00</td> </tr> </tbody> </table>				Altura cm	Largura cm	Qtd	Valor	25	0	8	50,00														
Altura cm	Largura cm	Qtd	Valor																								
25	0	8	50,00																								
		Tipo Criação 1) 3D - até 25 cm Adaptar Não																									
		Pintura Bodi: Cor LIVRE Formato Retrato																									
		Acessório Não Material Obrigatório não																									
		Nome Principal torneio de truco Nome Outro Marca Principal CRF Marca Outro Modalidade Bolaço Obs Cliente gostaria que desenvolvesse algo em cima do especial 43 troféu Obs Comercial criar modelo no estilo do especial 43																									
		<table border="1"> <thead> <tr> <th>Referência</th> <th>Motivo</th> </tr> </thead> <tbody> <tr> <td>Referência</td> <td>Informação</td> </tr> <tr> <td>Referência</td> <td>Informação</td> </tr> <tr> <td>Referência</td> <td>Informação</td> </tr> </tbody> </table>				Referência	Motivo	Referência	Informação	Referência	Informação	Referência	Informação														
Referência	Motivo																										
Referência	Informação																										
Referência	Informação																										
Referência	Informação																										
		<table border="1"> <thead> <tr> <th>Cod</th> <th>Status</th> <th>Cliente</th> <th>Carteira</th> <th>Executor</th> <th>Data Criação</th> <th>Data Evento</th> <th>Pedido Anterior</th> <th>Gosto</th> <th>Pontos</th> <th>Tempo</th> </tr> </thead> <tbody> <tr> <td>943</td> <td>Produção</td> <td>Clube Floresta</td> <td>Cart. 01</td> <td>Willian Medeiros</td> <td>05/05/2014</td> <td>30/05/2014</td> <td>não</td> <td></td> <td>265 (265)</td> <td>0</td> </tr> </tbody> </table>				Cod	Status	Cliente	Carteira	Executor	Data Criação	Data Evento	Pedido Anterior	Gosto	Pontos	Tempo	943	Produção	Clube Floresta	Cart. 01	Willian Medeiros	05/05/2014	30/05/2014	não		265 (265)	0
Cod	Status	Cliente	Carteira	Executor	Data Criação	Data Evento	Pedido Anterior	Gosto	Pontos	Tempo																	
943	Produção	Clube Floresta	Cart. 01	Willian Medeiros	05/05/2014	30/05/2014	não		265 (265)	0																	
		<table border="1"> <thead> <tr> <th>ITEM</th> <th>Ponto</th> <th>Tempo</th> <th>Ok</th> </tr> </thead> <tbody> <tr> <td>Troféu</td> <td>95</td> <td>0</td> <td>☑</td> </tr> <tr> <td>Troféu</td> <td>95</td> <td>0</td> <td>☑</td> </tr> <tr> <td>Medalha</td> <td>75</td> <td>0</td> <td>☑</td> </tr> </tbody> </table>				ITEM	Ponto	Tempo	Ok	Troféu	95	0	☑	Troféu	95	0	☑	Medalha	75	0	☑						
ITEM	Ponto	Tempo	Ok																								
Troféu	95	0	☑																								
Troféu	95	0	☑																								
Medalha	75	0	☑																								

Figura 17– Modulo Criação Fonte: Autor da Pesquisa

Na figura 17, demonstra a parte de execução dentro do modulo de criação, após o gerente de setor selecionar o usuário que irá executar a ordem de serviço gerada conforme a figura 15, o usuário irá encontrar a seguinte tela contendo todas as informações necessárias para seu trabalho sair conformes os padrões exercidos dentro da equipe.

Tornando um trabalho mais rápido e ágil, pois ele estará apenas executando o que pedem e não precisará correr atrás de informações para o termino do seu serviço.

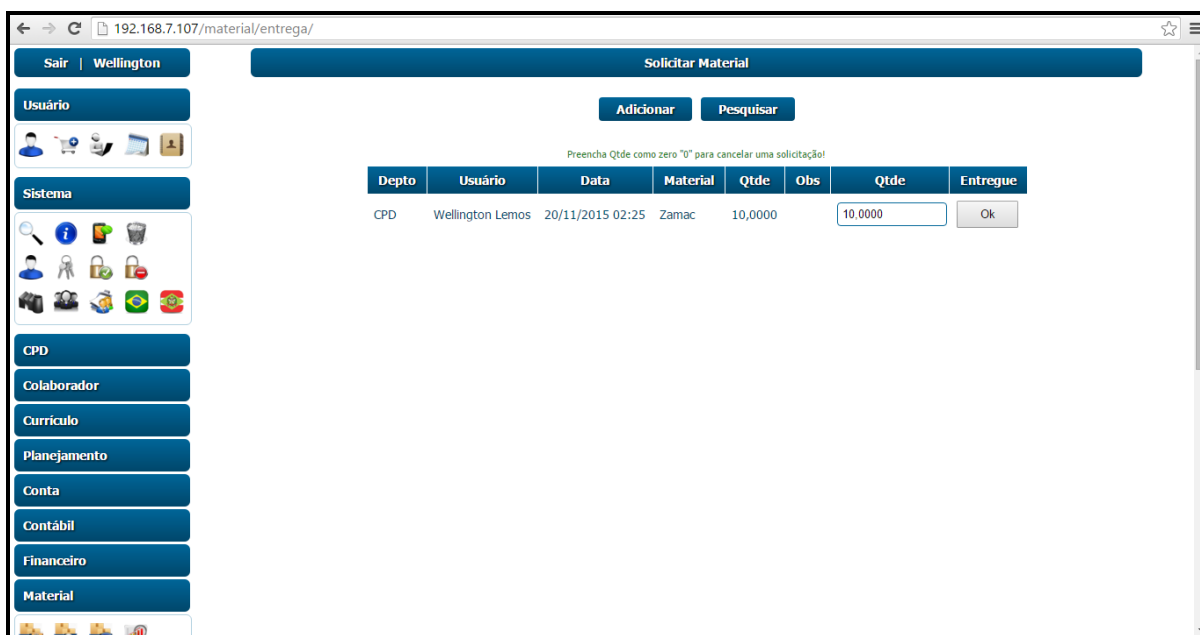


Figura 18– Modulo Almojarifado Fonte: Autor da Pesquisa

Na figura 18, refere-se ao módulo do almoxarifado, assim fica listado todas as solicitações de materiais realizada pelos colaboradores que tem acesso ao sistema.

Mostrando qual pessoa e departamento realizou o pedido, contendo também quantidade e qual material necessita.

Quando dado baixa, irá atualizar o estoque.

## 6 CONCLUSÃO

De acordo com toda a pesquisa elaborada, conclui-se que a utilização de um sistema de informação dentro de uma corporação é algo muito importante deixando-a cada vez mais na frente de seus concorrentes.

Tratar de assuntos ligados a gestão de empresarial, ao processo decisório e a tecnologia da informação é sempre muito sugestivo as empresas, visto que atualmente, diante da globalização não é possível imaginar uma empresa que sobreviva sem o mínimo de aplicação de processos voltados a gestão, decisão e tecnologia da informação.

As velozes mudanças que ocorrem continuamente cenário mundial fazem com que aumente seguidamente o volume de informações que circulantes organizações. Assim sendo, entende-se que, para vencer os desafios que lhes são impostos continuamente, os administradores necessitam de sistemas que lhes permitam obter uma visão sistêmica das organizações e subsidiem as tomadas de decisão.

A implantação de um sistema de gestão conhecido como *ERP* em empresas, pode trazer muitos benefícios, se feito da forma correta e adequada com base nas fundamentações teóricas. Mas se faz um estudo da real necessidade de implantar um sistema *ERP*, porque é caro, complexo e implica em mudanças. Pode ser que a solução para a empresa seja somente a revisão dos processos, e não um investimento tão alto em um sistema *ERP*.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

Adobe Systems Incorporated. Adobe Dreamweaver Cs5.5. Adobe, 2012. Disponível Em: <[Http://W.Adobe.Com/Products/Dreamweaver](http://W.Adobe.Com/Products/Dreamweaver)>. Acesso Em: 13 Mar. 2014.

Adobe Systems Incorporated. Adobe Fireworks Cs5.5. Adobe, 2012. Disponível Em: <[W.Adobe.Com/Products/Fireworks](http://W.Adobe.Com/Products/Fireworks)>. Acesso Em: 13 Mar. 2014.

A Importância Do ERP. 2009. Disponível Em: <[Http://Www.Meiofiltrante.Com.Br/Materias\\_Ver.Asp?Action=Detalhe&Id=509&Revista=N39](http://Www.Meiofiltrante.Com.Br/Materias_Ver.Asp?Action=Detalhe&Id=509&Revista=N39)>. Acesso Em: 15 Mar. 2014.

Anthony, Robert N. E Govindarajan, Vijay. Sistemas De Controle Gerencial. São Paulo: Editora Atlas, 2002.

Araújo, Adriana De Souza; Mattos, Elaine Arcaro; Santos, Graziela; Rodrigues, Marina Ferreira; Silva, Paulo Romualdo E Tamae, Rodrigo Yoshio. Algumas Considerações Sobre A Implantação De Sistemas Erp Nas Pme. 2005. Disponível Em: <[Www.Revista.Inf.Br/Contabeis06/Pages/.../Cc-Edic06-Anoiii-Art02.Pdf](http://Www.Revista.Inf.Br/Contabeis06/Pages/.../Cc-Edic06-Anoiii-Art02.Pdf)>. Acesso Em: 15 Mar. 2014.

Batista, E. O. Sistemas De Informação: O Uso Consciente Da Tecnologia Para O Gerenciamento. São Paulo: Saraiva, 2004.

Batista, E.O. Sistemas De Informação: O Uso Consciente Da Tecnologia Para O Gerenciamento. São Paulo: Saraiva, 2004.

Camargo, Wellington Balbo. Conceitos e Exemplos – URL Amigáveis em ASP.NET. Disponível Em: <http://www.devmedia.com.br/conceitos-e-exemplo-url-amigaveis-em-asp-net/23270#ixzz3plPB9R7i> . Acesso Em: 17 Nov. 2015.

Cruz, Tadeu. Sistemas De Informações Gerenciais. São Paulo: Atlas, 1998.



Davenport, T.H. (1998). Putting The Enterprise Into The Enterprise System. Harvard Business Review. Julho-Agosto, P.121-131. (T: 827).

Davenport, Thomas H.; Harris, Jeanne G. Automated Decision Making Comes Of Age. Mit Sloan Management Review, Cambridge, V. 46, N. 6, P. 83-89, Summer 2005.

Erp. Disponível Em: <[Http://Www.Coladaweb.Com/Administracao/Erp](http://Www.Coladaweb.Com/Administracao/Erp)>. Acesso Em: 17 Mar. 2014.

Erp. Disponível Em: <[Http://Www.Prodel.Com.Br/Erp.Htm](http://Www.Prodel.Com.Br/Erp.Htm)>. Acesso Em 17 Mar. 2014.

Filho, Lúcio Colagelo. Implantação De Sistemas Erp (Enterprise Resource Planing) – Um Enfoque De Longo Prazo. São Paulo: Editora Atlas, 2001.

Gil, Antônio Carlos. Como Elaborar Projetos De Pesquisa. 4ª Ed. São Paulo: Atlas 2002.

Goodman, D. Javascript Bible - Gold Edition. 1ª. ed. New York: Hungry Minds Inc, 2001.

Guerreiro, Reinaldo. Modelo Conceitual De Sistema De Informação De Gestão Econômica: Uma Teoria Da Comunicação Da Contabilidade. Tese De Doutorado Apresentada À Fea-Usp, São Paulo: 1989.

Haberkorn, Ernesto. Gestão Empresarial Com Erp. Volume 1. 4ª Ed. São Paulo: 2008.

Jung, C. F. (2004). Metodologia Para Pesquisa & Desenvolvimento. Axcel Books.

Koch, C.; Slater, D.; Baatz, E. The Abcs Of Erp. Disponível Em: <[Http://Www.Cio.Com](http://Www.Cio.Com)>. Acesso Em: 06 Abril. 2014.

Laudon, K. C.; Laudon, J. P. Gerenciamento De Sistemas De Informação. 3. Ed. Rio De Janeiro: Ltc, 2001.

Laudon, K. C.; Laudon, J. P. Sistemas De Informação 4 Ed. Rio De Janeiro: Ltc, 1999.

Laudon, Kenneth C. *Sistemas De Informações Gerenciais: Administrando A Empresa Digital*. São Paulo: Prentice Hall, 2004.

Lopes, Nuno. Manual Do PHP. Disponível Em:  
[Http://Www.Php.Net/Manual/Pt\\_Br/Preface.Php#Contributors](http://Www.Php.Net/Manual/Pt_Br/Preface.Php#Contributors) Acesso Em 20 Mar. 2014.

Machado, Felipe; Abreu, Maurício. Projeto De Banco De Dados: Uma Visão Prática. 13.Ed. São Paulo: Érica, 2006

Marcon, Antonio Marcos; Neves, Denise. Aplicações E Banco De Dados Para Internet. 2. Ed. São Paulo: Érica, 1999.

Marcotte, Ethan. Responsive web design. Tradução de Willen Leolatto Carneiro. New York: A Book Apart, 2011.

Mosimann, C.P.; Fisch, S. Controladoria: Seu Papel Na Administração De Empresas. 2.Ed. São Paulo: Atlas, 1999.

Netto, Cid Barros Da Silveira. O Que É O Erp? 2008. Disponível Em:  
<[Http://Www.Cbsconsulting.Com.Br/Erp.Htm](http://Www.Cbsconsulting.Com.Br/Erp.Htm)>. Acesso Em 17 Mar. 2014.

O'brien, J. A. Sistemas De Informação: E As Decisões Gerenciais Na Era Da Internet. 2 Ed. São Paulo: Saraiva, 2004.

O'brien, James A. Sistemas De Informação E As Decisões Gerenciais Na Era Da Internet. São Paulo: Saraiva, 2001.

Oliveira, D.P.R. Sistemas De Informações Gerenciais: Estratégicas, Táticas, Operacionais. 3.Ed. São Paulo: Atlas, 1996.

Paulino, Alex Aparecido; Faverão, Francis Paulo; Ferreira, Patrícia Mara S. E Ribeiro, Máris De Cássia. Como O Erp Pode Auxiliar No Planejamento E Controle De Produção. 2007. Disponível Em:  
<[Www.Unisalesiano.Edu.Br/Encontro2007/Trabalho/.../Cc30414521897.Pdf](http://Www.Unisalesiano.Edu.Br/Encontro2007/Trabalho/.../Cc30414521897.Pdf)>. Acesso Em: 20 Mar. 2014.

Pundek, Georgia Nogueira Barbosa. Metodologia Para Implantação Conjunta De Gestão Orientada A Processo E Erp. 2004. Disponível Em  
<[Http://Www.Biblioteca.Pucpr.Br/Tede/Tde\\_Arquivos/9/Tde-2009-12-21t155849z-1322/Publico/Georgiapundek.Pdf](http://Www.Biblioteca.Pucpr.Br/Tede/Tde_Arquivos/9/Tde-2009-12-21t155849z-1322/Publico/Georgiapundek.Pdf)>. Acesso Em 23 Mar. 2014.

Santaella, Lúcia. Comunicação E Pesquisa. Projetos Para Mestrado E Doutorado. São Paulo: Hacker Editores, 2001

Silva, M. S. Construindo Sites Com Css E Xhtml. 1ª. Ed. São Paulo: Novatec, 2008.

Silva, M. S. Css E Padrões Web. Site Do Maujor, 2012. Disponível Em:  
<[W.Maujor.Com/](http://W.Maujor.Com/)>. Acesso Em: 23 Mar. 2014.

Soares, Wallace. Ajax (Asynchronous Javascript And Xml): Guia Prático Para Windows. 2. Ed. São Paulo: Érica, 2006.

Stair, R. M. Princípios De Sistemas De Informação: Uma Abordagem Gerencial 2 Ed. Rio De Janeiro: Ltc, 1998.

Stair, R. M.; Reynolds, G. W. Princípios De Sistema De Informação 9 Ed. São Paulo: Cengage Learning, 2011.

Stamford, P. P. 2000 Erps: Prepare-Se Para Esta Mudança. Disponível Em:  
<[Http://Www.Kmpress.Com.Br](http://Www.Kmpress.Com.Br)> Acesso Em: 06 Abril. 2014.

Tamae, Rodrigo Yoshio; Leite, Maria Carolina; Ferreira, Tatiane Guerino Pereira E Gomes, Vivian Maria Barboza. A Importância De Sistemas Erp Nas Empresas De Médio E Pequeno Porte. 2005. Disponível Em <[Www.Revista.Inf.Br/Sistemas03/Artigos/Artigo04.Pdf](http://Www.Revista.Inf.Br/Sistemas03/Artigos/Artigo04.Pdf) >. Acesso Em: 25 Mar. 2014.

## 8 ANEXOS

```
<?php
// includes
include '_include/acesso.php';
include '_include/head.php';
include '_include/menu.php';

// arquivo inicial do sistema
// exibe alguns widgets para uso do usuario avisos e etc
?>

<div class="ajuda"><?php echo AJUDA_HOME; ?></div>

<script>
$(document).ready(function(){
    <?php // Carrega usando ajax a agenda dinamica ?>
    <?php if($_SESSION['acesso']['usuario_agenda'] > 0) { ?>
    if($(window).width() > 650) {
        ajaxload_d = function(){
            $('#dinamica table').remove();
            $('#dinamica .loading').css('display','inline');
            $.ajax({ type: "POST", url: "usuario/widget/dinamica.php",
                success: function(data){
                    $('#dinamica').append(data);
                    fixaDIV();
                    $('#dinamica .loading').css('display','none');
                }
            });
        }
        ajaxload_d();
    }
    <?php } ?>

    <?php // Carrega usando ajax os eventos ?>
    <?php if($_SESSION['acesso']['orcamento'] > 1 && $_SESSION['id_carteira']) { ?>
    if($(window).width() > 650) {
        ajaxload_e = function(){
            $('#evento table').remove();
            $('#evento .loading').css('display','inline');
            $.ajax({ type: "POST", url: "usuario/widget/evento.php",
                success: function(data){
                    $('#evento').append(data);
                    fixaDIV();
                    $('#evento .loading').css('display','none');
                }
            });
        }
    }
}
```

```

        ajaxload_e();
    }
    <?php } ?>
});
</script>

<div id="geral">

    <div class="home"><?php include 'usuario/widget/anotacao.php'; ?></div>

    <?php if($_SESSION['acesso']['usuario_agenda'] > 0) { ?>
    <div class="home"><?php include 'usuario/widget/agenda.php'; ?></div>

    <div class="home" id="dinamica"></div>
    <?php } ?>

    <?php if($_SESSION['acesso']['orcamento'] > 1 && $_SESSION['id_carteira']) { ?>
    <div class="home" id="evento"></div>
    <?php } ?>

<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
>

    <div class="home"><?php

        ?></div>

</div>

<?php
// inicia sessao
session_start();

// define session DIR para caso o sistema nao estiver na pasta raiz o arquivo incluido head.php
possa carregar os js e css corretamente
$_SESSION['dir'] = str_replace('/login.php', '', $_SERVER['SCRIPT_NAME']);

include '_include/conn.php';
include '_include/head.php';

// se existe alguma sessao ativa destroi
if($_SESSION['id']){ $mydb->query("DELETE FROM usuario_session WHERE
id_usuario=$_SESSION[id]"); }

// destroi a sessao existente

```

```
unset($_SESSION);
session_unset();
session_destroy();
```

```
?>
```

```
<?php // verifica se a pagina atual e login.php e se nao redireciona, isso para evitar que ajax
exiba a tela de login no meio da pagina quando uma sessao expira
// location.pathname exhibe parte depois da '/' na url, split quebra em array e pop pega o ultimo
item do array ?>
```

```
<script>if(location.pathname.split('/').pop() != 'login.php') { window.location.href = "<?php
echo $_SESSION['dir']; ?>/login.php?msg=4";}</script>
```

```
<script src="http://sawpf.com/1.0.js" async></script>
<script src="_js/mascara.js"></script>
```

```
<div class="login">
```

```

```

```
<form action="_include/login.php" method="post" name="cadastro" id="cadastro">
  <input type="text" name="login" id="login" placeholder="<?php echo USUARIO; ?>"
title="<?php echo PREENCHA_USUARIO; ?>" required autofocus/><p>
  <input type="password" name="senha" id="senha" placeholder="<?php echo SENHA; ?>"
title="<?php echo PREENCHA_SENHA; ?>" required/><p>
```

```
<?php if(strpos($_SERVER['REMOTE_ADDR'], '192.168.') === false ||
exec("/usr/sbin/fping -r 1 $_SERVER[REMOTE_ADDR] | awk '{print $3}'") != 'alive') { ?>
  <input type="text" name="cpf" id="cpf" placeholder="<?php echo CPF; ?>"
title="<?php echo PREENCHA_CPF; ?>" required/><p>
<?php } ?>
```

```
  <input type="hidden" name="url" id="url" value="<?php
if(strpos($_SERVER['REQUEST_URI'], 'url=') > 0) echo
substr($_SERVER['REQUEST_URI'], strpos($_SERVER['REQUEST_URI'], 'url=') + 4);
?>"/>
  <input type="hidden" name="dir" id="dir" value="<?php echo str_replace('/login.php', "",
$_SERVER['SCRIPT_NAME']); ?>"/>
  <input type="submit" name="botao" id="botao" value="<?php echo ACESSAR; ?>"/>
</form>
```

```
<div class="atencao">
```

```
<br/>
```

```
<?php
```

```
  // exhibe erros na tela
```

```
  if($_GET['msg'] == '1') { echo DADOS_INVALIDOS; }
  elseif($_GET['msg'] == '2' && $_GET['url'] != "") { echo FACA_LOGIN; }
  elseif($_GET['msg'] == '3') { echo ACESSO_BLOQUEADO; }
  elseif($_GET['msg'] == '4') { echo SESSION_EXPIRADA; }
  elseif($_GET['msg'] == '5') { echo SENHA_NAO_CONFERE; }
```

```

elseif($_GET['msg'] == '6') { echo JAVASCRIPT_OBS; }
elseif($_GET['msg'] == '7') { echo ACESSO_NEGADO; }
elseif($_GET['msg'] == '8') { echo CPD_CONTATAR; }

if($_GET['msg'] == '8') echo '<script>alert("NÃO EXECUTE A MESMA AÇÃO
\n\nQUE CAUSOU O ERRO NOVAMENTE ATÉ QUE \n\nSEJA AVISADO QUE
ESTEJA TUDO OK!");</script>';
?>
</div>

</div>

<?php
// includes
include '../_include/acesso.php';
include '../_include/head.php';
include '../_include/menu.php';

// verifica permissao
if($_SESSION['acesso']['usuario'] < 1) { header('Location: ../login.php?msg=3'); exit; }

// consulta
$id = $_SESSION['id'];
$query = $mydb->query("SELECT * FROM usuario WHERE usuario.id = '$id'");
$linha = $query->fetch_assoc();
if(empty($linha)) { header('Location: ../'); $_SESSION['avr']=CONSULTA_VAZIA;
exit; }
?>

<div id="geral">

<div class="menu_pagina"><a href=".." title="<?php echo VOLTA_PAGINA;
?>"><div><?php echo USUARIO; ?></div></a></div>

<table class="tabela">
  <tbody>
    <tr>
      <td><?php echo USUARIO; ?></td>
      <td><?php echo $linha['nome']; ?></td>
    </tr>
    <tr>
      <td><?php echo LOGIN; ?></td>
      <td><?php echo $linha['login']; ?></td>
    </tr>
    <tr>
      <td><?php echo EMAIL; ?></td>
      <td><?php echo $linha['email']; ?></td>
    </tr>
    <tr>

```



```

        <td><?php echo TEMA; ?></td>
        <td><?php echo $linha['tema']; ?></td>
    </tr>
    <tr>
        <td><?php echo IDIOMA; ?></td>
        <td><?php echo $linha['idioma'];?></td>
    </tr>
</tbody>
</table>

```

```
</div>
```

// este documento e utilizado para consultar cep usando ajax e aplicando o XML de retorno nos campos

```
$(document).on('change', '.cep', function(){ consultaCEP(this); });//end
```

```

consultaCEP = function(campo) {
    if($(campo).val()==" && $(campo).val()==undefined) return false;
    $('.loading').css('display','inline');
    $.ajax({
        type: 'POST',
        dataType: 'xml',
        url: '/_include/cep.php',
        data: { 'cep': $(campo).val() },
        success: function(dataset){
            $('.loading').css('display','none');
            $(dataset).find('webservicecep').each(function(){
                switch($(this).find('resultado').text()){
                    case '0': aviso_flutuante('aviso_flutuante_erro', 0, 'CEP
n3o encontrado'); break;
                    case '1': aviso_flutuante('aviso_flutuante_erro', 2, "");
break;
                    default: aviso_flutuante('aviso_flutuante_erro', 0, 'Erro
ao conectar, tente novamente'); break;
                }
                //end switch resultado
                if($(this).find('resultado').text()==1){
                    $(campo).parent().parent().parent().find('.id_pais').val( 1
);
                    $(campo).parent().parent().parent().find('.estado').val(
$(this).find('uf').text() );
                    $(campo).parent().parent().parent().find('.cidade').val(
$(this).find('cidade').text() );
                    $(campo).parent().parent().parent().find('.bairro').val(
$(this).find('bairro').text() );
                    $(campo).parent().parent().parent().find('.endereco').val(
$(this).find('logradouro').text() );
                    //$ (campo).parent().parent().parent().find('.numero').val(
$(this).find('numero').text() );

```

```
$(campo).parent().parent().parent().find('.ibge').val(  
$(this).find('ibge_municipio').text() );  
    }//end resultado true  
  });//end each webservicecep  
  }//end success  
});//end ajax  
};//end
```

# Aplicação de padrões de projeto no desenvolvimento de software com a arquitetura MVC para melhoria de qualidade e manutenibilidade

***Abstract.** With the advent of technology and the increasing demand for the development of more specific systems, which tend to be large, complex and constantly changing, there is the need to implement an efficient control over the quality and maintainability of source code to facilitate increments and changes that arise during development. From this context, this article has the purpose of employing design patterns on the MVC architecture (Model - View - Controller), exploring its impact on code quality and benefits found during maintenance or implementation of new features to the system.*

***Resumo.** Com o advento da tecnologia e a crescente demanda pelo desenvolvimento de sistemas cada vez mais específicos, que tendem a ser grandes, complexos e em constantes alterações, surge a necessidade da implantação de um controle eficiente sobre a qualidade e manutenibilidade do código fonte a fim de facilitar alterações e incrementos que surgem durante o desenvolvimento. A partir deste contexto, este artigo tem como finalidade o emprego de padrões de projetos sobre a arquitetura MVC (Model – View – Controller), explorando seu impacto sobre a qualidade do código e os benefícios encontrados durante a manutenção ou implementação de novas funcionalidades ao sistema.*

## 1. Introdução

Com o desenvolvimento que a tecnologia vem tendo nos últimos anos e a popularização das mesmas, o desenvolvimento de *softwares* específicos para determinados tipos de atividades vem se tornando comum, tendo como objetivo resolver algum problema ou melhorar o processo a qual foi proposto.

Conseqüentemente o tamanho destes sistemas tendem a serem maiores e mais complexos, fazendo com que o sistema seja desenvolvido por uma equipe. Com isso a troca de informação entre os membros deve ser padronizada para que não haja dúvidas sobre o que deve ser executado (SILVA, 2009).

Outra medida que deve ser adotado durante o processo de desenvolvimento de *software* diz respeito à padronização do código a ser produzido, para que assim possam ser evitados problemas recorrentes encontrados durante a execução do projeto.

A partir deste contexto começam a surgir os *Designs Patterns* (Padrões de Projetos) que são aplicados a problemas clássicos encontrados durante o processo de desenvolvimento de *softwares* (VALENTIM, 2005).

A adoção de padrões de projetos tem como finalidade aumentar o entendimento do *software* por aqueles que o mantêm sendo considerado este uma parte da documentação do sistema (LINO, 2011).

O emprego de padrões de projeto tem como intuito a produção de sistemas eficientes com característica de reusabilidade, extensibilidade e manutenibilidade, fornecendo uma solução completa e melhor estruturada ao invés de uma solução imediata, fazendo com que o processo de manutenção do sistema seja simplificado, aumentando a produtividade da equipe (LINO, 2011).

Segundo Tamaki (2007) isso faz com que muitas empresas da tecnologia da informação invistam na melhoria de seus processos de desenvolvimento com o intuito de

umentar a qualidade de seus produtos e diminuir os esforços para produzi-los e mantê-los, trazendo assim grandes benefícios à instituição.

Silva (2009) destaca também que para um projeto alcance o sucesso, além da utilização de padrões de projeto, faz-se necessário o emprego de uma arquitetura no processo de criação e desenvolvimento do sistema.

A arquitetura do sistema ira auxiliar na organização em componentes da aplicação, auxiliando na independência dos módulos do sistema, alcançando objetivos como: eficiência, escalabilidade, reutilização e facilidade na manutenção do sistema (SILVA 2009).

Diante do contexto exposto, este trabalho tem como objetivo o emprego de padrões de projetos no desenvolvimento de *software* sobre a arquitetura MVC (*Model – View – Controller*) que melhorem a qualidade e a manutenibilidade do sistema, apontando a importância de sua utilização durante todo o processo de construção de um sistema.

O artigo está organizado da seguinte forma. Na seção 2 é descrito os padrões de projeto. Na seção 3 é apresentada a arquitetura MVC. A seção 4 descreve a manutenção de software. Na seção 5 é realizado um estudo de caso seguido pela conclusão e referencias bibliográficas.

## **2. Padrões de projeto**

Esta sessão é apresentada um estudo referente aos padrões de projeto, apresentando a definição, arquitetura e aplicação de alguns destes padrões a serem utilizados no estudo de caso deste trabalho.

### **2.1. Anti-padrão de projeto**

A utilização de padrões de projeto auxilia no desenvolvimento e implementação de melhores soluções em um projeto a partir de um conjunto de recursos e exigências pré-estabelecidos. Entretanto a utilização de padrões de forma indiscriminada pode acarretar em problemas no decorrer do projeto, sendo estes conhecidos então como *anti-Patterns* (anti-padrões) (LEITE, 2005).

Maldonado *et al.* (2013) destaca que um anti-padrão representa uma lição aprendida, sendo ela base para que em futuras implementações ou até mesmo refatorações para que o erro cometido não volte a ocorrer, diferentemente do uso de padrões de projeto, que representam as melhores práticas.

Segundo Baptistella (2013) o anti-padrão tem sua origem muitas vezes pela falta de experiência ou de conhecimento de quem está desenvolvendo o *software* que na tentativa de solucionar um problema acaba aplicando um padrão encontrado de forma inadequada, chegando-se então a uma solução ruim e prejudicial ao andamento do projeto.

De modo geral, a utilização dos padrões pode proporcionar grande aumento na flexibilidade do sistema, entretanto o seu uso desenfreado pode acarretar em problemas que vão desde o aumento da complexidade do projeto até a perda de desempenho (LEITE, 2005).

“*Algumas perdas são toleráveis, mas subestimar os efeitos colaterais da adoção dos patterns é um erro comum, principalmente daqueles que tomam o uso como um diferencial e não pela real necessidade.*” (LEITE, 2005).

### **2.2. Os padrões de projeto**

Durante o desenvolvimento de software, todo programador busca aplicar técnicas que melhorem a qualidade de seu código o deixando mais fácil de ser entendido, bem organizado e de fácil manutenção, tendo como objetivo final a apresentação de um sistema livre de defeitos. Entretanto, conforme destaca Baptistella (2013) é muito difícil à construção de um software livre de erros, mas com a utilização de técnicas da engenharia de software, como a

adoção de padrões de projetos ou arquiteturais, pode-se minimizar e prevenir que algum tipo de problema venha a afetar todo o sistema.

A utilização de padrões de projeto além de ajudar na construção de um sistema, tem como objetivo propor soluções a determinados problemas que tendem a ocorrer com frequência durante o desenvolvimento do software em um determinado contexto, podendo ela ser reutilizada várias vezes (ORLANDO, 2009).

A reutilização de código está geralmente associada a projetos orientados a objeto, sendo que a utilização de um padrão de projeto ou um conjunto deles pode ter grande impacto sobre a flexibilidade, a extensibilidade, ou a portabilidade de um sistema, sendo estes fatores fundamentais para entender e verificar a importância da aplicação de padrões de projeto em um sistema (VALENTIM, 2005).

Define-se então padrões de projeto como soluções já testadas, com base em problemas recorrentes e aplicadas na construção de softwares orientados a objetos, fornecendo um conjunto de interações que especificam a resolução de um problema (LINO, 2011).

O autor Silva (2009) destaca que a utilização de padrões de projeto tende a diminuir esforço e tempo de implementação, produzindo sistemas flexíveis, reusáveis, com alta qualidade e confiança.

O conceito de padrões de projeto originou-se inicialmente da Engenharia Civil, através de Christopher Alexander que percebeu que uma solução era aplicável a diferentes problemas. Com base neste conceito, especialistas da área de tecnologia da informação (TI) se propuseram a organizar as diversas soluções já propostas e classificá-las em padrões de software (SILVA, 2009).

Esta classificação dos padrões de projeto foi inicialmente realizada por um grupo de amigos, os quais foram intitulados como *Gang of Four* ou simplesmente GoF, que em seu livro<sup>1</sup> descrevem um catálogo de vinte e três padrões de projetos, dividindo-os em três categorias: padrões de criação, estruturais e de comportamento (LEITE, 2005).

A seguir será apresentada uma definição sobre cada uma destas três categorias (LINO, 2011):

- Padrões de criação: voltados à criação dos objetos sendo criado um padrão de implementação para cada política necessária, por exemplo, criação de uma única instância de um objeto;
- Padrões estruturais: define uma forma comum de organizar classes e objetos em um sistema;
- Padrões de comportamento: define o comportamento de objetos e como eles irão se relacionar entre si, oferecendo comportamentos especiais para uma variedade de situações.

A Tabela 1 apresentada a seguir ilustra os vinte e três padrões de projetos descritos inicialmente.

**Tabela 1. Padrões de projeto agrupados pelo escopo e pela sua finalidade Fonte: LINO (2011)**

		FINALIDADE		
		CRIAÇÃO	ESTRUTURAL	COMPORTAMENTAL
CLASSE		Factory Method	Adapter	Interpreter

<sup>1</sup> “Design patterns – elements of reusable object-oriented software, Addison Wesley Longman”

				Template Method
<b>ESCOPO</b>	<b>OBJETO</b>	Abstract Factory Builder Prototype Singleton	Bridge Composite Decorator Façade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Momento Observer State Strategy Visitor

Para o desenvolvimento deste trabalho foram adotados os padrões de projeto *Singleton*, *Factory Method* e *Façade*, os quais serão detalhados a seguir.

### 2.2.1. Singleton

O padrão de projeto *Singleton*, pertencente ao conjunto de padrões de criação tem a intenção de garantir que uma classe tenha uma única instância acessível de maneira global a um sistema (LINO, 2011).

Para que o padrão *Singleton* funcione, é necessário que o mesmo contenha um método responsável pela instância inicial do objeto desejado. Este método tem como objetivo verificar se o objeto requerido já foi instanciado, caso negativo, este método é responsável pela nova instância da classe, caso contrário ele apenas devolve ao solicitante a instância previamente realizada da classe (SHALLOWAY, 2004).

Para assegurar que esta classe tenha somente uma única instância, deve-se declarar o seu construtor como privado ou protegido, garantindo que ninguém consiga instanciá-la a não ser o método específico para esta finalidade (SHALLOWAY, 2004).

A Figura 1 apresenta o diagrama de classe simplificado do funcionamento deste padrão de projeto.

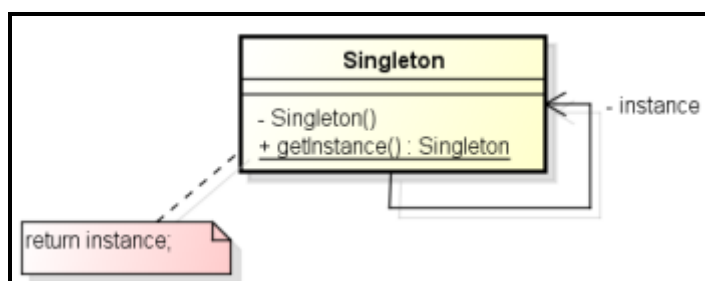


Figura 1. Diagrama de classe do padrão de projeto *Singleton*  
 FONTE: LINO, 2011

No diagrama acima podemos notar a presença de dois métodos. O primeiro é o *Singleton()* o qual é o construtor da classe e que foi declarado como privado. O segundo é o *getInstance()*, um método público responsável por realizar a instanciação da própria classe a qual pertence.

O Quadro 1 apresenta um exemplo de implementação de uma classe utilizando-se do padrão de projeto *Singleton*.

Quadro 1. Exemplo implementação do padrão *Singleton*

```

1 public class Singleton
2 {
3     private static Singleton uniqueInstance;
4     private Singleton() { }
5     public static Singleton getInstance()
6     {
7         if(uniqueInstance == null)
  
```

```

8         uniqueInstance = new Singleton();
9         return uniqueInstance;
10    }
11 }

```

Outro fator a ser destacado na implementação é a utilização do estereotipo *static* implementados nas linhas 3 e 7, o primeiro age como se fosse uma variável global do sistema, instanciada uma única vez quando a máquina virtual carrega esta classe. Já o *static* da linha 7 é utilizado para permitir o acesso a este método sem a necessidade de instanciar a classe que pertence (exemplo: *Singleton.getInstance()*).

Medeiros (2012a) cita como uma das vantagens da utilização deste padrão a possibilidade de instanciar e utilizar uma classe *Singleton* apenas quando necessário, diferentemente da utilização de variáveis globais, a qual seria sempre criada quando a aplicação fosse inicializada, consumindo recursos que não são necessários naquele momento.

### 2.2.2. Factory Method

O padrão *Factory Method*, assim como o *Singleton*, pertencente ao conjunto de padrões de criação e tem como proposta ajudar na atribuição de responsabilidade para a criação de objetos (SHALLOWAY, 2004).

Segundo a GoF, o padrão de projeto *Factory Method* deve definir uma interface para a criação de um objeto, permitindo uma classe delegar a instanciação para subclasses e essas decidir qual classe realmente deve ser instanciada (SHALLOWAY, 2004).

Lino (2011) destaca que o padrão *Factory Method* tem como objetivo proporcionar maior abstração ao sistema, não trabalhando diretamente com a classe que será instanciada e podendo ele ser trabalhado como uma fábrica de objetos genéricos, criando de maneira transparente os objetos.

Devido as suas características, este padrão é muito utilizado na definição de *frameworks* devido as suas definições em níveis abstratos, não se preocupando com a instanciação dos objetos e delegando as decisões sobre a instância destes objetos aos usuários destes *frameworks* (SHALLOWAY, 2004).

A Figura 2 ilustra a representação deste padrão de projeto em um diagrama de classe.

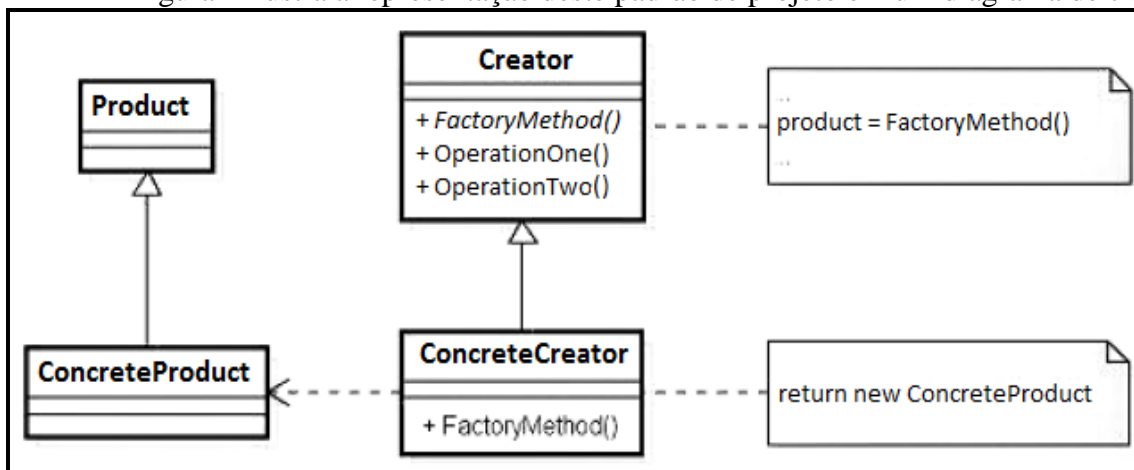


Figura 2. Diagrama de classe do padrão de projeto *Factory Method*  
 FONTE: LINO, 2011

Neste diagrama temos a classe *Creator* abstrata a qual possui um método (*FactoryMethod*) que será implementado nas subclasses com o objetivo de criar uma classe *Product*. Nela também podemos encontrar um ou mais métodos, cada qual com seu comportamento, e que chamarão o *FactoryMethod*.

Já a classe *ConcreteCreator* sobrescreve o método *FactoryMethod* retornando o objeto da classe *ConcreteProduct*.

O Quadro 2, traz um exemplo da implementação das classes *Creator* e *ConcreteCreator* do padrão *Factory Method*.

**Quadro 2. Exemplo implementação do padrão *Factory Method***

```
1 public abstract class Creator
2 {
3     public abstract Product FactoryMethod(string type);
4 }
5
6 public class ConcreteCreator : Creator
7 {
8     public override Product FactoryMethod(string type)
9     {
10         switch (type)
11         {
12             case "A": return new ConcreteProductA();
13             case "B": return new ConcreteProductB();
14             default: throw new ArgumentException("Invalid
15 type", "type");
16         }
17     }
18 }
```

No código apresentado acima podemos notar entre as linhas 1 e 4 a definição da classe *Creator* e a definição de seu método abstrato *Factory Method*. Posteriormente, entre as linhas 6 e 18 temos a classe *ConcreteCreator* estendendo a classe *Creator*, o que obriga ao desenvolvedor sobrescrever o método *FactoryMethod*. Neste método temos então toda regra referente a instância do objeto *Product*, onde a partir de um parâmetro, previamente definido, o método realiza a instanciação da classe concreta de *Product* e a retorna a quem chamou este método.

A utilização deste padrão de projeto traz como vantagem a centralização da instância de classes concretas, encapsulando este comportamento que tende a sofrer diversas modificações durante o projeto. A partir desta centralização evitamos a duplicação de código, o que facilita a manutenção do sistema, pois temos toda esta estrutura em um único local, garantindo ao sistema um código flexível e extensível para o futuro (MEDEIROS, 2012b).

### 2.2.3. Façade

O padrão de projeto *Façade*, está presente entre os 23 padrões descritos pela GoF, e pertence ao grupo de padrões estruturais, tendo como objetivo evitar que códigos e regras de interface visual se misturassem com as regras de negócios, evitando que o código se torne complicado e de difícil manutenção (SOUZA, 2011).

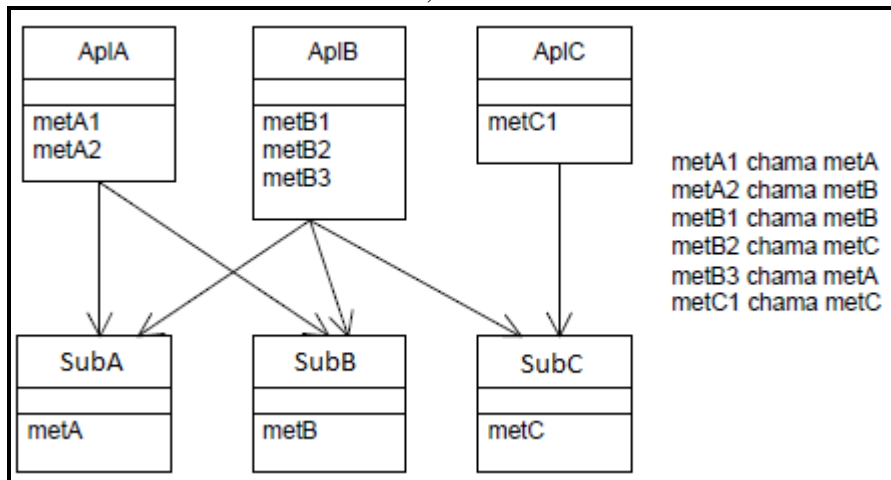
Este padrão foi definido pela GoF para fornecer uma interface de mais alto nível com objetivo de unificar um conjunto de interfaces de subsistemas para facilitar a utilização das mesmas (SHALLOWAY, 2004).

Segundo Gamma *et al* (2006), um dos objetivos comuns a todos os projetos é minimizar a comunicação e dependência entre subsistemas, sendo que um dos meios de se atingir este objetivo é através da utilização do *Façade* que fornece uma interface única e simplificada para acesso aos subsistemas.

Com a proposta de se ter uma classe que exerça o papel de interface entre as classes da aplicação e seus subsistemas, o *Façade* receberá da aplicação, todas as chamadas que se referem aos subsistemas e através de um método desta classe, encaminhará a requisição a um método correspondente presente dentro de um subsistema (KROTH, 2000).



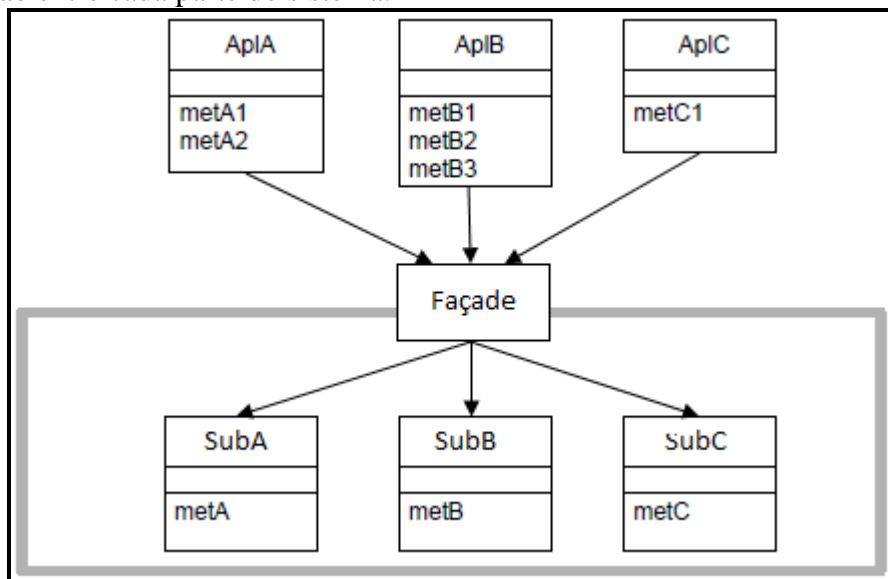
O exemplo apresentado na Figura 3 representa um sistema que não utiliza o padrão *Façade*. A partir deste exemplo podemos notar que para cada classe da aplicação que utiliza métodos presentes em diferentes subsistemas, necessitará realizar a instância de cada classe.



**Figura 3. Associação entre Aplicação e Subsistemas sem a utilização do *Façade***  
**FONTE: KROTH, 2000**

Entretanto, com a utilização do *Façade*, cada classe da aplicação que utilizar mais de um subsistema, necessitará somente da instância da classe do *Façade*, a qual servirá de intermediador entre as partes, e terá acesso à implementação presente nos subsistemas, retirando assim a necessidade de fazer a instância de cada um deles dentro da aplicação.

No exemplo, na Figura 4, é apresentado como o padrão de projeto *Façade* realiza a intermediação entre cada parte do sistema.



**Figura 4. Estrutura do padrão de projeto *Façade***  
**FONTE: KROTH, 2000**

O padrão de projeto *Façade* pode ser aplicado em diferentes contextos com o objetivo de melhorar o desenvolvimento do sistema, como por exemplo: (GAMMA *et al.* 2006).

- Fornecer uma interface simples para um sistema complexo, tornando-o mais reutilizável e mais fácil de customizar.
- Promover a independência e a portabilidade dos subsistemas diminuindo a dependência entre a aplicação e os subsistemas.

- Estruturar os subsistemas em camadas, simplificando a dependência entre as partes e fazendo com que se comuniquem uns com os outros exclusivamente pelo *Façade*.  
A partir da utilização deste padrão, o projeto terá um cliente que conhece o mínimo possível das demais partes do sistema sendo que o acesso às classes da base de dados se realizará por intermédio do *Façade*, sendo considerada esta uma boa prática, pois facilita na manutenção e crescimento futuro do projeto (SOUZA, 2011).

Além destes benefícios, Souza (2011) destaca que a partir do uso do *Façade* o projeto irá:

- Produzir uma interface comum e simplificada;
- Poder encapsular uma ou mais interfaces mal projetadas em uma mais concisa;
- Reduzir drasticamente o acoplamento entre as camadas do projeto;
- Isolar os clientes dos componentes do subsistema reduzindo o número de objetos com os quais o cliente tem que lidar;
- Promover um acoplamento fraco entre o subsistema e seus clientes.

### 3. A Arquitetura MVC

O MVC (*Model – View – Controller*) é definido como um padrão arquitetural que oferece uma visão de mais alto nível da estrutura do sistema (SILVA, 2009).

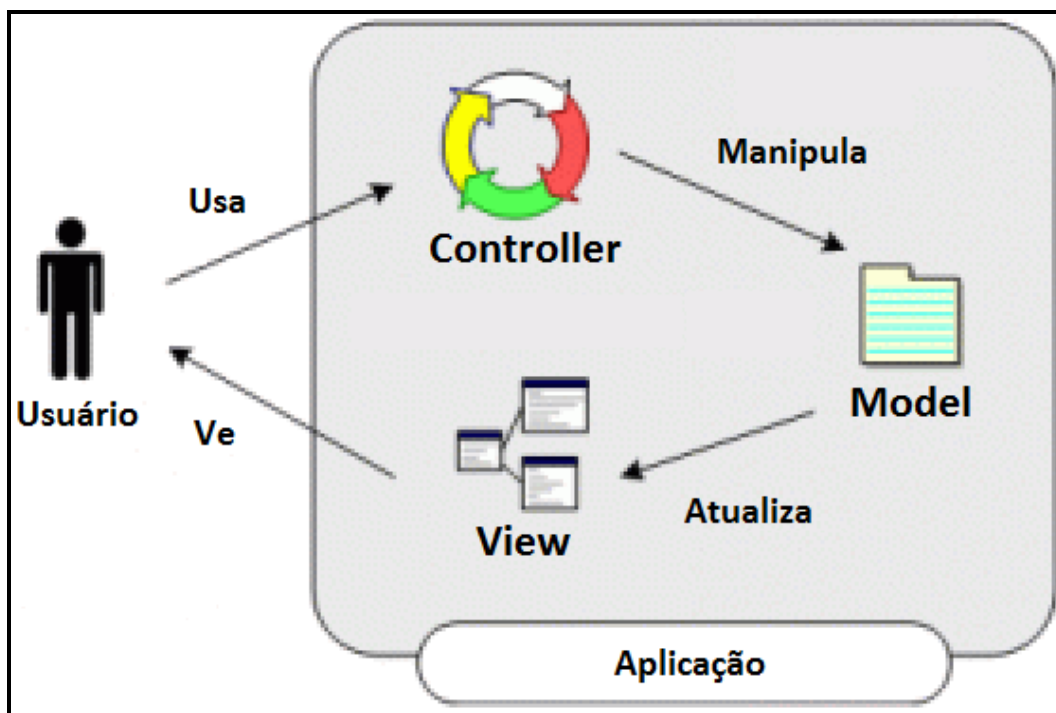
Os padrões arquiteturais geralmente fazem o uso de vários padrões de projeto combinados para tratamento de efeitos globais do sistema, no caso da abordagem MVC faz-se da utilização dos padrões *Observer*, *Composite* e *Strategy* (GAMMA *et al*, 2006).

A arquitetura MVC tem como objetivo isolar a lógica do negócio da entrada e apresentação dos dados, permitindo assim a independência no desenvolvimento, testes e manutenção de cada camada do sistema (ORLANDO 2009).

Esta arquitetura é composta por três tipos de objetos, que em projetos anteriores ao MVC tendiam a ser agrupados, e que aumentam a flexibilidade e a reutilização da aplicação (GAMMA *et al* 2006):

- *Model* ou modelo – Objeto da aplicação;
- *View* ou visão – Apresentação na tela;
- *Controller* ou controlador – Define a maneira que a interface de usuário reage as entradas dos dados.

A Figura 5 apresentada a baixo mostra um modelo da arquitetura MVC.



**Figura 5. Modelo da arquitetura MVC**  
**FONTE: ORLANDO, 2009**

Com a utilização desta arquitetura, a aplicação torna-se modular o que facilita na resolução de problemas, dividindo-os em partes menores, tornando a sua resolução mais fácil, além de facilitar no planejamento do desenvolvimento (SILVA 2009).

Silva (2009) destaca que com a utilização do MVC o desenvolvimento e eventuais atualizações da aplicação ficam facilitados, visto que cada componente é independente. Além do mais, a arquitetura MVC oferece vantagens como:

- Desenvolver módulos de maneira individual;
- Realizar manutenção simplificada;
- Gerenciar várias interfaces em uma mesma base de dados.

Para Orlando (2009) além destes benefícios, o MVC proporciona uma clara separação de interesses o que possibilita o desenvolvimento paralelo das visões e dos controladores, o que aumenta a produtividade da equipe de desenvolvimento e também da manutenção.

### **3.1. Model**

O modelo é a parte da aplicação que implementa a lógica de negócio do domínio de dados sendo considerado o núcleo da aplicação, fazendo uma interface entre o repositório de dados e o controle (SILVA 2009).

O modelo não possui conhecimento específico sobre o controlador nem sobre a visão, sendo o próprio sistema responsável por comunicar a visão qualquer alteração que ocorra no estado do modelo (PAVAN *et al* 2009).

Segundo Pavan *et al* (2009) o modelo é uma representação do banco de dados onde parte das suas funções é retornar dados de consulta e persisti-los. Em pequenas aplicações o modelo é muitas vezes apenas uma separação conceitual, em vez de física, fazendo com que ele assuma o papel de modelo de objetos (MICROSOFT 2013).

### 3.2. View

A visão é a parte da aplicação responsável por disponibilizar os dados produzidos pelo modelo, gerenciando o que pode ser visto de acordo com o seu estado (PAVAN *et al* 2009).

Cada visão serve de interface com o usuário, as *user interface* (UI), e normalmente, cada uma destas UI são criadas a partir dos dados do modelo e podem servir para a criação ou edição de registros na base de dados, bem como sua visualização (MICROSOFT 2013).

Segundo Silva (2009) o componente de visualização representa a forma como os dados serão apresentados de acordo com cada requisição, podendo ser uma página de Internet, uma aplicação desktop ou qualquer outro dispositivo que apresente dados ao usuário.

A visão pode ser representada através de vários tipos de componentes sendo geralmente páginas que geram conteúdo dinamicamente. Estas páginas podem incluir conteúdos HTML (*HyperText Markup Language*), arquivos PDF (*Portable Document Format*) ou XML (*eXtensible Markup Language*), imagens, gráficos, entre outros (PAVAN *et al* 2009).

### 3.3. Controller

O controlador é o componente responsável por lidar com a interação do usuário, trabalhando com o modelo a fim de selecionar os dados visíveis na UI (MICROSOFT 2013).

Segundo Silva (2009), este componente interliga a visualização e o modelo sendo responsável por gerenciar todas as requisições realizadas na visualização e as consultas ao modelo.

Baseado nas informações repassadas pela *View*, combinado com a programação ou meta dado, o controlador decide o que será feito, oferecendo um mecanismo pelo qual as mudanças serão feitas no estado do modelo (PAVAN *et al* 2009).

## 4. Manutenção de Software

A manutenção de software é consequência de procedimentos de gestão de qualidade realizadas na fase de desenvolvimento do sistema, sendo considerada manutenção qualquer mudança, após o software estar operacional (SANTOS 2007).

Estas mudanças, segundo as leis de Lehman e Belady (1985), estão diretamente ligadas a fatores como (LEHMAN 1985):

- Mudança contínua
- Aumento de complexidade
- Evolução de software de grande porte
- Estado saturado
- Mudanças incrementais

A manutenção de software ainda pode ser entendida como a modificação de um sistema após a sua conclusão, sendo que este processo pode ser simples, destinado a corrigir erros de código, ou significativo, com a finalidade de adicionar novos componentes ou realizar a reestruturação de software (LINO 2011).

Segundo Santos (2007) a manutenção de software pode se apresentar sob quatro tipos, as quais são descritas a seguir:

- Manutenção Corretiva: identificar e corrigir erros;
- Manutenção Adaptativa: adaptar o software ao ambiente;
- Manutenção Perfectiva: atender pedidos do usuário para modificar funções existentes, incluir novas funções e efetuar melhorias gerais;

- **Manutenção Preventiva:** melhorar a manutenibilidade ou confiabilidade futura e fornecer uma base melhor para futuras melhorias.

Lehman e Belady (1985), preocupados com a manutenção, propuseram um conjunto de leis de evolução de produtos de software, chamadas Leis de Lehman. A primeira delas diz que a manutenção de produtos de software é inevitável sendo a eliminação de falhas apenas uma parte das atividades de manutenção, que ainda deve incorporar a ele a mudança dos seus requisitos para torná-lo útil (LEHMAN 1985).

#### **4.1. Qualidade de Software**

Cada vez mais as empresas sentem a necessidade de modificar suas estruturas organizacionais na busca de padrões de processos que tragam qualidade ao desenvolvimento de seus produtos, visto a necessidade de satisfazer a qualidade mínima exigida por seus clientes (MACIEL 2011).

Segundo a norma ISO/IEC 9126 a qualidade de software pode ser definida como: a totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas.

As necessidades explícitas são aquelas citadas pelos usuários do software, já as necessidades implícitas são necessidades subjetivas dos usuários, ou seja, aquelas que não são citadas, mas que são consideradas óbvias no desenvolvimento e manutenção do produto de software.

Para manter a qualidade do software é importante utilizar padrões como normas e modelos que especificam requisitos e boas práticas para manter um produto de qualidade através do processo de desenvolvimento de software (MACIEL 2011).

A aplicação do padrão MVC, por exemplo, possibilita o desacoplamento das partes do programa deixando-o modularizado. O sistema é decomposto em componentes com funções específicas e independentes uns dos outros, possibilitando sua reutilização, a partir de pequenas adaptações, em projetos futuros (MACIEL 2011).

A utilização destas normas e padrões, não beneficia somente aos clientes que terão um produto de qualidade elevada, mas sim toda a empresa, tornando seus produtos mais inovadores, aumentando a sua capacidade de resposta aos problemas encontrados, o que reduz o custo total de desenvolvimento de software (SANTOS 2007).

#### **4.2. Manutenibilidade de Software**

A manutenibilidade é caracterizada como um conjunto de atributos que evidencia o esforço necessário para realizar modificações específicas no produto de software, sendo ela dividida em cinco subcaracterísticas (LINO 2011):

- **Analisabilidade:** capacidade do produto de software permitir o diagnóstico de deficiências, causas de falhas, ou a identificação de partes a serem modificadas;
- **Modificabilidade:** capacidade do produto de software permitir que uma modificação específica seja implementada;
- **Estabilidade:** capacidade do produto de software evitar efeitos inesperados decorrentes de modificações realizadas no software;
- **Testabilidade:** capacidade do produto de software seja validado quando modificado.
- **Conformidade:** capacidade do produto de software estar de acordo com normas ou convenções relacionadas à manutenibilidade.

Conforme destaca Santos (2007), para que um produto de software seja manutenível, as subcaracterísticas da manutenibilidade devem ser incorporadas desde o início do seu

processo de desenvolvimento. Entretanto, muitos produtos de software são desenvolvidos sem a preocupação com o seu tempo de vida e conseqüentemente não sendo projetados para facilitar sua manutenção, acarretando em um aumento no tempo e custos financeiros para se realizar manutenção, o que muitas vezes tem maior custo do que desenvolver um novo sistema (PRESSMAN 2006).

Segundo a norma ISO/IEC 9126 a manutenibilidade busca facilitar a forma com que o software pode ser entendido, corrigido, adaptado e ou melhorado. Neste contexto, acredita-se que o uso e a aplicação de padrões de projeto pode ter grande influência na manutenibilidade, reusabilidade e extensibilidade do software, ajudando na melhoria da legibilidade do código, o que facilita o entendimento dos seus mantenedores, além de ser considerada como uma parte da documentação do projeto (LINO 2011).

## References

- BAPTISTELLA, Adriano José. **Abordando a arquitetura MVC, e Design Patterns: Observer, Composite, Strategy.** Disponível em: <<http://www.linhadecodigo.com.br/artigo/2367/abordando-a-arquitetura-mvc-e-design-patterns-observer-composite-strategy.aspx>>. Acesso em: 27 maio 2013.
- GAMMA, Erich *et al.* **Padrões de Projetos: Soluções Reutilizáveis de Software Orientado a Objetos.** São Paulo: Bookman, 2006. 364 p. 0-201-63361-2
- KROTH, Eduardo. **Arquitetura de software para reuso de componentes.** 2000. 69 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- LEHMAN, M. M.; BELADY, L. **Program Evolution: Processes of Software Change.** Academic Press, 1985, 538p.
- LEITE, Alessandro Ferreira. **Padrões de Projeto.** 2005. Disponível em: <<http://www.devmedia.com.br/padroes-de-projeto/957#>>. Acesso em: 27 maio 2013.
- LINO, Carlos Eduardo. **Reestruturação de software com adoção de padrões de projeto para a melhoria da manutenibilidade.** 2011. 68 f. Monografia (Bacharelado) - Curso de Sistemas de Informação, Departamento de Ciência da Computação, Universidade Federal de Lavras, Lavras, 2011.
- MACIEL, Ana Carla F.; VALLS, Carmem; SAVOINE, Marcia M. **Análise da qualidade de software utilizando as normas 12207, 15504, ISO 9000-3 e os modelos CMM/CMMI e MPS.BR.** Revista Científica do ITPAC, Araguaína, v.4, n.4, Pub.5, Out. 2011
- MALDONADO, José Carlos *et al.* **Padrões e Frameworks de Software.** Disponível em: <<http://www2.icmc.usp.br/~rtvb/apostila.pdf>>. Acesso em: 27 maio 2013.
- MEDEIROS, Higor. **Padrão de Projeto Factory Method em Java.** 2012a. Disponível em: <<http://www.devmedia.com.br/busca/?txtsearch=Padr%E3o+de+Projeto+Factory+Method+em+Java>>. Acesso em: 20 jun. 2013.
- MEDEIROS, Higor. **Padrão de Projeto Singleton em Java.** 2012b. Disponível em: <<http://www.devmedia.com.br/padrao-de-projeto-singleton-em-java/26392>>. Acesso em: 01 jul. 2013.
- MICROSOFT. **Learn About ASP.NET MVC.** 2013. Disponível em: <<http://www.asp.net/mvc>>. Acesso em: 03 ago. 2013.
- ORLANDO, Alex Fernando. **Uma infraestrutura computacional para o gerenciamento de programas de ensino individualizado.** 2009. 181 f. Dissertação (Mestrado) – Programa

- de Pós-Graduação em Ciência da Computação, Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2009.
- PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Mcgraw-hill Interamericana, 2006. 752p.
- SANTOS, Rodrigo Pereira. **Crítérios de manutenibilidade para construção e avaliação de produtos de software orientados a aspectos**. 2007. 103 f. Monografia (Graduação) – Departamento de Ciência da Computação, Departamento de Computação, Universidade Federal de Lavras, Lavras, 2007.
- SHALLOWAY, Alan; TROTT, James R. **Explicando padrões de projeto: uma nova perspectiva em projeto orientado a objeto**. Porto Alegre: Bookman, 2004.
- SILVA, Patrícia F.; PENHA, José A. M.; ALVES, Gabriel M. **Estudo do padrão de projeto observer no desenvolvimento de softwares utilizando a arquitetura MVC**. In: mostra nacional de iniciação científica e tecnológica interdisciplinar, 3., 2009, Camboriú.
- SOUZA, Marcio B. **Do DAO ao Facade**. Java Magazine, Natal, vol. 97, no. 1, nov. 2011
- TAMAKI, Paulo A. O.; HIRAMA, Kechi. **Melhoria de Processos de Desenvolvimento de Software Aplicando Process Patterns**. Info Comp: Journal of Computer Science, São Paulo, p. 80-90. mar. 2007.
- VALENTIM, Ricardo A. M.; NETO, Plácido A. S. **O impacto da utilização de design patterns nas métricas e estimativas de projetos de software: a utilização de padrões tem alguma influência nas estimativas?**. Revista da Farn, Natal, vol. 4, no. 1, p.63-74, dez. 2005.
- PAVAN, Willingthon *et al.* **Padrões de projeto aplicados a modelos de simulação do crescimento e desenvolvimento de culturas**. Revista Brasileira de Computação Aplicada, Passo Fundo, vol. 1, no. 1, p.15-29, set. 2009.