

MICHAEL FERNANDES RODRIGUES

SISTEMA DE CAMISETAS PERSONALIZADAS

BRASÍLIA - DF

DEZEMBRO - 2011

MICHAEL FERNANDES RODRIGUES

SISTEMA DE CAMISETAS PERSONALIZADAS

Trabalho de Conclusão de curso apresentado como requisito parcial ao Curso de Sistemas de Informação da Faculdade Fortium, para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em \_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_\_\_

Orientador:

Prof. Esp. Leonardo Felix

BANCA EXAMINADORA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Professor Orientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Professor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Professor

BRASÍLIA-DF

2011

|  |
| --- |
| Michael Fernandes Rodrigues.  SISTEMA DE CAMISETAS PERSONALIZADAS / Michael Fernandes Rodrigues – Brasília: 2011. XXX p.  **Projeto Final de Graduação –** Faculdade FORTIUM, Curso de Sistemas de Informação.  Orientado por:  Prof. Esp. Leonardo Felix  1. Projeto de Graduação. I. SISTEMA DE CAMISETAS PERSONALIZADAS |

DEDICATÓRIA

Dedico este trabalho à minha família, aos professores com os quais tive a oportunidade de expandir meu aprendizado, por todo o apoio prestado e pela motivação que foi fundamental para a conclusão deste.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por possibilitar a realização desse sonho. Em segundo lugar agradecer a minha família, que esteve comigo nos momentos de dificuldade e aos professores e amigos, pelo apoio e compreensão dispensados.

**RESUMO**

Este trabalho de conclusão de curso propõe a criação de um sistema de camisetas personalizadas. O sistema tem por objetivo fornecer parâmetros para o bom funcionamento e controle dos processos de fabricação e personalização de camisetas. A motivação para a realização deste trabalho é tornar o processo de aquisição e fabricação de camisetas personalizadas mais preciso e confiável de forma a garantir sua qualidade. A parte teórica do projeto foi estabelecida para garantir esta qualidade desde a elaboração até a implantação do produto de software. O sistema apresentado é implementado na linguagem PHP (versão 5.3.8). A escolha de alguns frameworks foram vitais para a evolução do projeto, são eles: ZendFramework (versão 1.11) responsável pela camada de controle e integração com as demais camadas, Extjs (versão 2.1) responsável pela camada de visão e o Doctrine (versão 1.2) responsável pela camada de modelo onde é sub-dividida em outras três camadas (Value Object, ORM e Persiste ou DAO).

**Palavras-chave**: metodologias, sistema, camisetas, personalizadas, fabricação, aquisição, processos, framework, extjs, doctrine, zendfrawork, value object, ORM, Persiste, DAO.

**ABSTRACT**

This conclusion of course work proposes the creation of a system of custom shirts. The system aims to provide parameters for the proper functioning and control of manufacturing processes and custom shirts. The motivation for this work is to make the procurement and manufacturing of custom t-shirts more accurate and reliable to ensure its quality. The theoretical part of the project was established to ensure this quality from development to deployment of the software product. This scheme is implemented in PHP (version 5.3.8). The choice of some frameworks are vital to the evolution of design, they are: ZendFramework (version 1.11) responsible for the control layer and integration with the other layers, ExtJs (version 2.1) responsible for the view layer and Doctrine (version 1.2) responsible model where the layer is further sub-divided into three layers (Value Object, DAO or ORM and persists).

**Keywords:** methodology, systems, t-shirts, custom manufacturing, purchasing, process, framework, extjs, doctrine, zendfrawork, value object, ORM, persists, DAO.

**LISTA DE ILUSTRAÇÕES**

Figura 01:

Figura 02:

Figura 03:

Figura 04:

Figura 05:

Figura 06:

Figura 07:

Figura 08:

Figura 09:

Figura 10:

Figura 11:

Figura 12:

**LISTA DE TABELAS**

Tabela 01:

Tabela 02:

Tabela 03:

Tabela 04:

Tabela 05:

Tabela 06:

Tabela 07:

Tabela 08:

Tabela 10:

Tabela 11:

**LISTA DE ABREVIATURAS**

**API**: Application Programming Interface

**ASF**: Apache Software Foundation.

**CGI**: Common Gateway Interface.

**CSS**: Cascading Style Sheets.

**CTO**: Chief Technology Officer

**DQL**: Doctrine Query Language.

**HTTP**: Hypertext Transfer Protocol.

**IRUP**: IBM Rational Unified Process.

**MD5**: Message-Digest Algorithm.

**MER**: Modelo de Entidade e Relacionamento.

**MPL**: Mozilla Public License.

**MVC**: Model View Controller.

**MVCC**: MultiVersion Concurrency Control.

**NCSA** : National Center for Supercomputing Applications.

**OO**: Orientação a objetos.

**PERL**: Practical Extraction and Report Language.

**PHP**: Hypertext Processor.

**POO**: Programação Orientada a Objetos.

**RDBMS**: Relational Data Base Management System.

**SCP**: Sistema de Camisetas Personalizadas.

**SGBD**: Sistema Gerenciador de Banco de Dados.

**SPL**: Sun Public License.

**SQL**: Structured Query Language.

**SSH**: Secure Shell.

**SSL**: Secure Sockets Layer.

**TI**: Tecnologia da informação.

**TLS**: Transport Layer Security.

**TOAST**: The Oversized-Attribute Storage Technique.

**UCB**: University of California at Berkeley

**UML**: Linguagem de Modelagem Unificada.

**W3C**: World Wide Web Consortium.

**WAL**: Write-Ahead Log.

**WEB**: World Wide Web.

**XHTML**: EXtensible HyperText Markup Language.

**XML**: EXtensible Markup Language.

**YUI**: Yahoo User Interface.

SUMÁRIO

[1. INTRODUÇÃO 15](#_Toc308709479)

[1.1. MOTIVAÇÃO 15](#_Toc308709480)

[1.2. OBJETIVOS DO SISTEMA 16](#_Toc308709481)

[1.2.1. GERAL 16](#_Toc308709482)

[1.2.2. ESPECÍFICOS 16](#_Toc308709483)

[1.3. ESTRUTURA DO TRABALHO DE CONCLUSÃO 16](#_Toc308709484)

[2. PESQUISA E MÉTODOS 18](#_Toc308709485)

[2.1. SISTEMA PROPOSTO E SISTEMAS SIMILARES 18](#_Toc308709486)

[2.2. DESENVOLVIMENTO DE SOFTWARE ORIENTADO A OBJETOS 20](#_Toc308709487)

[2.2.1. PRINCÍPIOS DA ORIENTAÇÃO A OBJETOS 21](#_Toc308709488)

[2.2.2. ANÁLISE ORIENTADA A OBJETOS 23](#_Toc308709489)

[2.2.3. PROJETO ORIENTADO A OBJETOS 24](#_Toc308709490)

[2.2.4. PROGRAMAÇÃO ORIENTADA A OBJETOS 25](#_Toc308709491)

[2.2.5. LINGUAGEM PHP 26](#_Toc308709492)

[2.2.6. BANCO DE DADOS POSTGRESQL 28](#_Toc308709493)

[2.2.7. CSS 31](#_Toc308709494)

[2.2.8. XHTML 32](#_Toc308709495)

[2.2.9. IRUP 33](#_Toc308709496)

[2.2.10. JAVASCRIPT 35](#_Toc308709497)

[2.2.11. UML 35](#_Toc308709498)

[2.2.12. MVC 38](#_Toc308709499)

[2.2.13. W3C 39](#_Toc308709500)

[2.2.14. SERVIDOR WEB (APACHE) 41](#_Toc308709501)

[2.2.15. FERRAMENTAS 42](#_Toc308709502)

[2.2.15.1. NETBEANS 7.0.1 42](#_Toc308709503)

[2.2.15.2. POWER DESIGNER 15.0 43](#_Toc308709504)

[2.2.15.3. APACHE JMETER 2.3 43](#_Toc308709505)

[2.2.15.4. ACUNETIX WEB SECURITY SCANNER 7 43](#_Toc308709506)

[2.2.15.5. TORTOISESVN 1.6 44](#_Toc308709507)

[2.2.15.6. JCROP 0.9 44](#_Toc308709508)

[2.2.16. FRAMEWORKS 46](#_Toc308709509)

[2.2.16.1. EXTJS 2.0 46](#_Toc308709510)

[2.2.16.2. DOCTRINE 1.2 48](#_Toc308709511)

[2.2.16.3. ZEND FRAMEWORK 1.11 49](#_Toc308709512)

[3. ANÁLISE FUNCIONAL 51](#_Toc308709513)

[3.1. REQUISITOS DO NEGÓCIO 51](#_Toc308709514)

[3.1.1. PAGAMENTO DOS PEDIDOS 51](#_Toc308709515)

[3.1.2. RELATÓRIOS 51](#_Toc308709516)

[3.1.3. CONTROLE DE ACESSO 51](#_Toc308709517)

[3.1.4. MANUTENÇÃO 52](#_Toc308709518)

[3.1.5. CLIENTES 52](#_Toc308709519)

[3.1.6. CONFECÇÃO 52](#_Toc308709520)

[3.1.7. POSTAGEM DE PEDIDO 52](#_Toc308709521)

[3.2. ANÁLISE DAS NECESSIDADES E PROBLEMAS DIAGNOSTICADOS 52](#_Toc308709522)

[3.3. BENEFÍCIOS ESPERADOS 52](#_Toc308709523)

[3.4. ANÁLISE DE RISCOS E PLANO DE CONTINGÊNCIA 54](#_Toc308709524)

[3.5. PLANO DE CRONOGRAMA DO PROJETO 54](#_Toc308709525)

[4. PROJETO LÓGICO E FÍSICO 55](#_Toc308709526)

[4.1. REQUISITOS DO USUÁRIO 55](#_Toc308709527)

[4.1.2 REQUISITOS DE INTERFACE EXTERNA 55](#_Toc308709528)

[4.1.3 REQUISITOS DE INTERFACE COM AMBIENTE DE HARDWARE 55](#_Toc308709529)

[4.1.4 REQUISITOS DE INTERFACE COM AMBIENTE DE SOFTWARE 56](#_Toc308709530)

[4.1.5 REQUISITOS DE EXECUÇÃO 56](#_Toc308709531)

[4.1.6 REQUISITOS GENÉRICOS 56](#_Toc308709532)

[4.1.7 REQUISITOS INVERSOS 56](#_Toc308709533)

[4.1.8 REQUISITOS DE INTERFACE 56](#_Toc308709534)

[4.2. MODELAGEM FUNCIONAL E DE DADOS 56](#_Toc308709535)

[4.2.1. MODELO ENTIDADE RELACIONAMENTO – MODELO CONCEITUAL 57](#_Toc308709536)

[4.2.2. MODELO ENTIDADE RELACIONAMENTO - MODELO FÍSICO 57](#_Toc308709537)

[4.2.3. DICIONÁRIO DE DADOS – MODELO FÍSICO 57](#_Toc308709538)

[5. MODELAGEM UML 58](#_Toc308709539)

[5.1. DIAGRAMA DE CASO DE USO 58](#_Toc308709540)

[5.2. DIAGRAMA DE ATIVIDADES 58](#_Toc308709541)

[5.3. DIAGRAMA DE SEQÜÊNCIA 58](#_Toc308709542)

[5.4. DIAGRAMA DE IMPLANTAÇÃO 58](#_Toc308709543)

[5.5. DIAGRAMA DE CLASSES 58](#_Toc308709544)

[6. ARQUITETURA COMPUTACIONAL E CUSTO 59](#_Toc308709545)

[6.1. DIAGRAMA DE EXECUÇÃO E/OU COMPONENTES 59](#_Toc308709546)

[6.2. FORMAS DE ARMAZENAMENTO E DISPONIBILIZAÇÃO DOS DADOS 59](#_Toc308709547)

[6.3. DIAGRAMA DA REDE 60](#_Toc308709548)

[6.4. SOFTWARES BÁSICOS E DE APOIO UTILIZADOS 61](#_Toc308709549)

[6.5. DETALHAMENTO DOS HARDWARES NECESSÁRIOS 61](#_Toc308709550)

[6.5.1 HARDWARE DO CLIENTE 61](#_Toc308709551)

[6.5.2. HARDWARE DO SERVIDOR 61](#_Toc308709552)

[6.6. DETALHAMENTO DE CUSTO 62](#_Toc308709553)

[7. SEGURANÇA 64](#_Toc308709554)

[7.1. MECANISMOS DE SEGURANÇA DA LINGUAGEM DE PROGRAMAÇÃO E SGDB. 64](#_Toc308709555)

[7.2. MECANISMOS DE SEGURANÇA E PRIVACIDADE DE DADOS. 65](#_Toc308709556)

[7.3. MECANISMO DE SEGURANÇA DE EQUIPAMENTOS E INSTALAÇÕES. 66](#_Toc308709557)

[8. TESTES E IMPLEMENTAÇÃO 67](#_Toc308709558)

[8.1. TESTES DE QUALIDADE 67](#_Toc308709559)

[8.1.1. REQUISITOS DE INTERFACE 67](#_Toc308709560)

[8.1.2. ROTEIROS DE TESTE 67](#_Toc308709561)

[8.1.3. EVIDÊNCIA DE TESTE 67](#_Toc308709562)

[8.2. TESTES DE STRESS 67](#_Toc308709563)

[8.3. TESTES DE SEGURANÇA 67](#_Toc308709564)

[8.4. IMPLANTAÇÃO 67](#_Toc308709565)

[8.4.1. PLANO DE IMPLANTAÇÃO 67](#_Toc308709566)

[8.4.2. MANUAL DE IMPLANTAÇÃO 69](#_Toc308709567)

[9. CONCLUSÃO 70](#_Toc308709568)

[10. REFERÊNCIAS BIBLIOGRÁFICAS 71](#_Toc308709569)

1. **INTRODUÇÃO**

Nos dias de hoje a necessidade de automatizar tarefas do dia-a-dia é cada vez mais comum, desta forma nota-se a oportunidade de desenvolver uma solução tecnológica para as empresas de camisetas personalizadas.

Diante da crescente procura das pessoas por camisetas personalizadas, nota-se a necessidade de criação um meio de possibilite que estas pessoas expressem suas vontades e que personalizarem as suas próprias camisetas da forma que quiserem através de um web site elegante, funcional, confiável e de fácil navegação.

* 1. **MOTIVAÇÃO**

O software que será desenvolvido neste projeto visa promover o bom atendimento aos clientes que utilizaram os serviços oferecidos pelas empresas de estampas personalizáveis em camisetas.

Um dos fatos que trouxe a motivação para desenvolvimento deste projeto foi o fato de que, hoje no mercado de camisetas personalizadas não existe nenhum sistema que possa ser adquirido (comprado e baixado gratuitamente na internet). Um sistema que possibilite a integração de uma empresa de camisetas com os seus clientes, a ponto de permitir que os clientes personalizem suas camisetas e sejam notificados do processo de produção das mesmas.

Com a integração do fluxo de produção das camisetas ao sistema informatizado, o tempo de resposta das solicitações dos clientes será mais rápido e controle da produção unitária será possível assim como os relatórios gerenciais acerca da produção das camisetas.

A solução apresentada irá facilitar as atividades do setor financeiro das empresas de estampas personalizáveis em camisetas através dos relatórios de produção dos funcionários.

* 1. **OBJETIVOS DO SISTEMA**
     1. **GERAL**

O Sistema de Camisetas Personalizadas automatiza todo o processo de fabricação de camisetas com estampas personalizadas.

* + 1. **ESPECÍFICOS**
* Autenticação.
* Cadastro do cliente.
* Galeria de camisetas.
* Cadastro de profissionais.
* Consulta de produtos.
* Confecção de produtos.
* Pagamento de produtos.
* Despachar produtos
* Geração de relatórios
  1. **ESTRUTURA DO TRABALHO DE CONCLUSÃO**

No capítulo 2 (dois) serão abordados os embasamentos tecnológicos utilizados neste projeto.

No capítulo 3 (três) serão abordados os itens que compõem a análise funcional do sistema de Camisetas Personalizadas, assim como os requisitos específicos e os benefícios esperados com este sistema.

No capítulo 4 (quatro) serão apresentadas as descrições dos requisitos dos usuários do Sistema de Camisetas Personalizadas, assim como a abordagem dos requisitos de interface externa, requisitos de interface com ambiente de hardware, requisitos de interface com ambiente de software, requisitos de execução.

No capítulo 5 (cinco) será apresentada a modelagem UML do Sistema de Camisetas Personalizadas. Através dos Diagramas de Caso de Uso, Diagramas de Atividade, Diagramas de Seqüência, Diagrama de Implantação e Diagrama de Classes todos eles construídos a partir dos requisitos levantados.

No capítulo 6 (seis) serão abordados a arquitetura computacional e o custo do produto deste projeto.

No capítulo 7 (sete) serão apresentadas as sugestões de segurança lógica e física para Sistema de Camisetas Personalizadas.

No capítulo 8 (oito) será apresentada a implementação e os testes do Sistema de Camisetas Personalizadas.

No capítulo 9 (nove) será apresentada a conclusão do projeto.

No capítulo 10 (dez) serão apresentadas as referências bibliográficas que irão compor este projeto.

1. **PESQUISA E MÉTODOS**

Este capítulo descreve o conteúdo teórico sobre as ferramentas e tecnologias envolvidas no desenvolvimento deste projeto.

* 1. **SISTEMA PROPOSTO E SISTEMAS SIMILARES**

O Sistema de Camisetas Personalizadassurge a partir da oportunidade de gerenciar o processo de fabricação de camisetas personalizadas utilizando a plataforma web para a integração das atividades envolvidas neste processo.

Considerando que as empresas de estampas em camisetas na maioria das vezes não possuem um sistema informatizado para gerenciar suas atividades internas assim como a integração de um módulo (página web) para que os seus clientes possam expressar sua criatividade e solicitar a confecção de seus pedidos de forma fácil, rápida e segura; é proposto um sistema especifico para esta área que ira proporcionar aos seus usuários uma sistematização semelhante à de suas rotinas e atividades diárias no processo de fabricação das camisetas personalizadas.

O Sistema de Camisetas Personalizadas inicialmente é dividido em dois módulos públicos e privado.

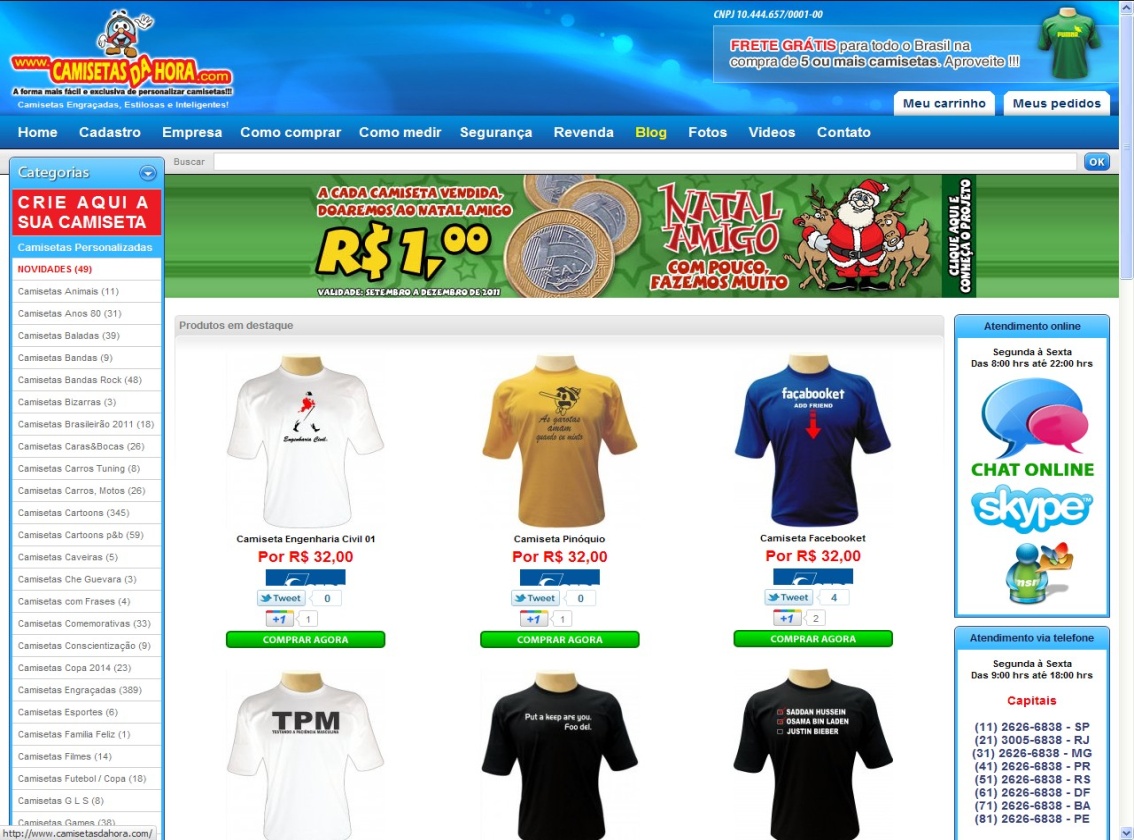
**Módulo Público (Frontend):** É a interface com os clientes ou visitantes do portal. Os recursos deste módulo são a navegação pela galeria de camisetas personalizadas e o painel de controle do cliente que comporta todas as funcionalidades atribuídas aos clientes.

**Módulo Privado (Backend):** É a interface com os funcionários da empresa, onde as funcionalidades serão apresentadas de acordo com o perfil do profissional autenticado. Os perfis autorizados para acessar estes módulos são: Administrador, Conferidor de Pagamentos, Confeccionador e Despachante.

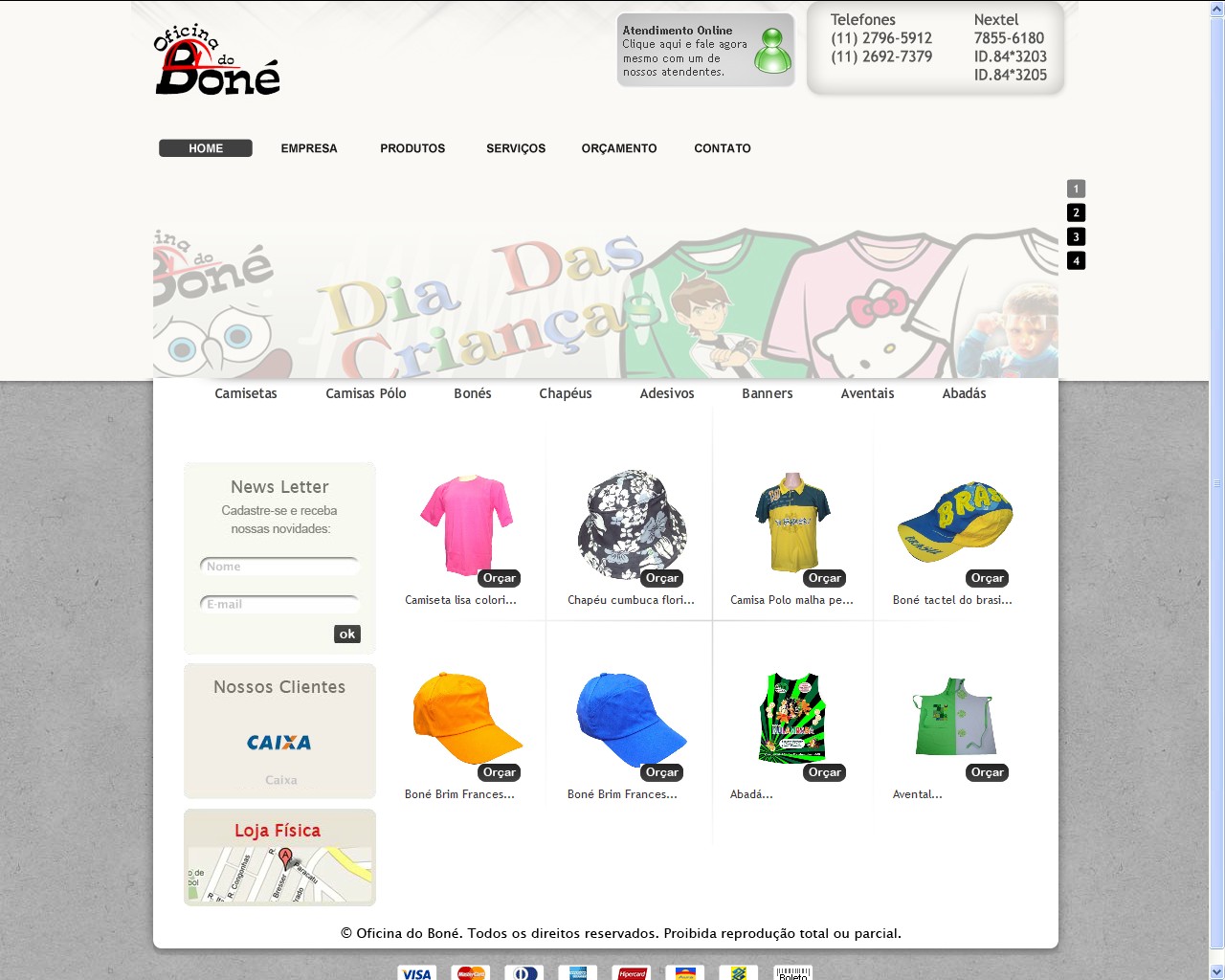
Quanto aos Sistemas Similares não existe nenhuma solução deste tipo que possa ser adquirida, ou seja, na verdade o que existe no mercado são serviços que utilizam de sistemas semelhantes, mas que não são comercializados e nem distribuídos.

Segue abaixo alguns *printscreens* (telas) de serviços com o propósito similar ao do Sistema de Camisetas Personalizadas:

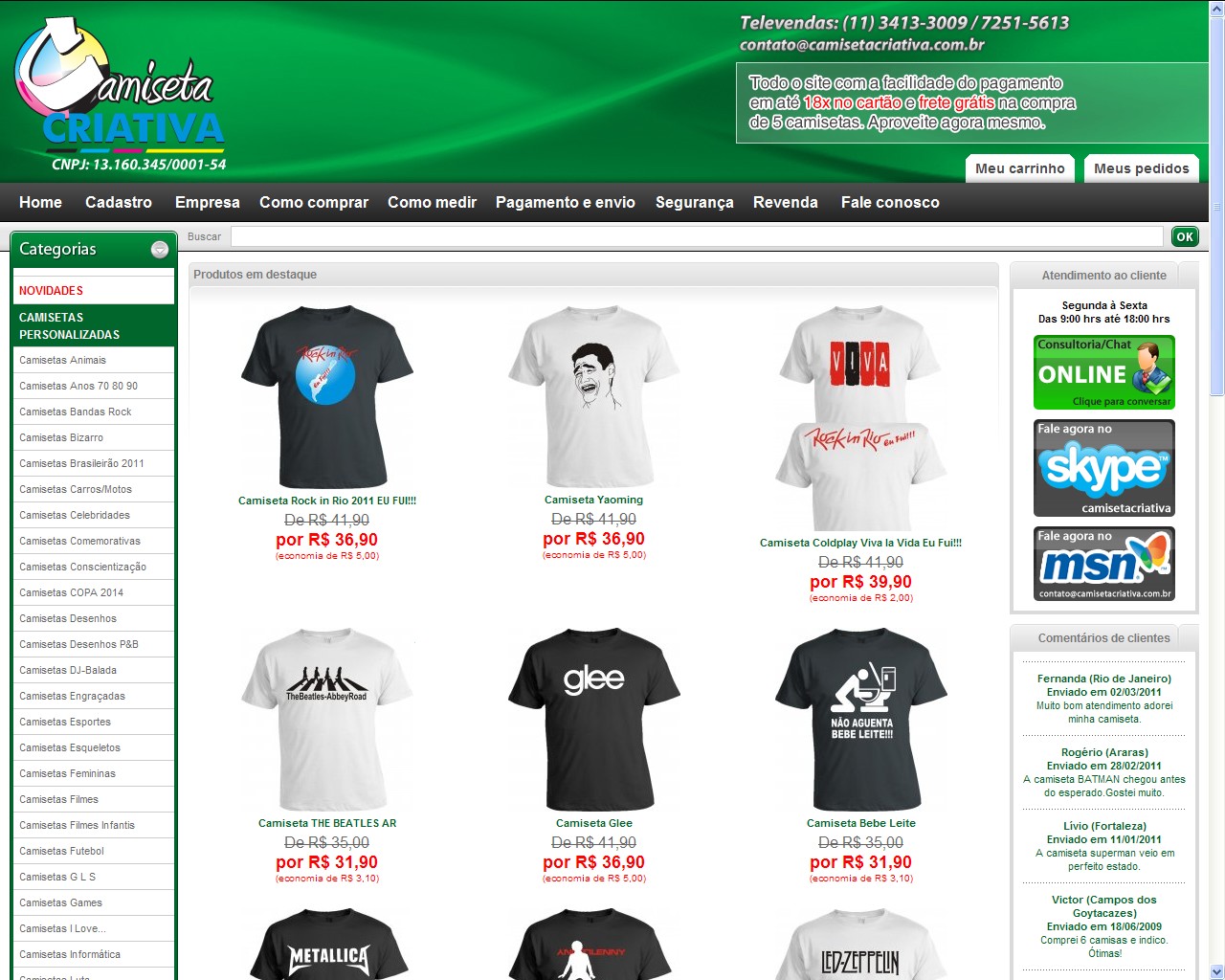
Camisetas da Hora



Oficina do Boné



Camiseta Criativa



Portal Transpaper



* 1. **DESENVOLVIMENTO DE SOFTWARE ORIENTADO A OBJETOS**
     1. **PRINCÍPIOS DA ORIENTAÇÃO A OBJETOS**

Alan Kay foi o criador do termo Programação orientada a Objetos e autor da linguagem programação Smalltalk. Algumas idéias da Programação Orientada a Objetos já eram utilizadas antes do Smalltalk. O Simula 67 foi criado por Ole Johan Dahl e Kristen Nygaard em 1967 e foi a primeira linguagem de programação a utilizar Programação Orientada a Objetos. Se observarmos bem os paradigmas da Programação Orientada a Objetos são bem antigos, mas só de uns anos pra cá vêem ganhando maior reconhecimento de grandes empresas do ramo de desenvolvimento de softwares. As linguagens que utilizam os artifícios da Programação Orientada a Objetos são diversas, as mais modernas são: C#, C++, Java, PHP, Python entre outras (David, 2007).

Segundo Horstmann, o projetista de software deve organizar a programação em um conjunto de classes que se relacionam. Cada classe deve ser especificada com precisão, listando tanto suas responsabilidades como seus relacionamentos com outras classes do sistema (Horstmann, 2006).

*“A programação orientada a objetos consiste em uma metodologia de desenvolvimento de software para gerenciamento de problemas cada vez mais complexos que não poderiam ser solucionados com antigas técnicas com foco meramente na codificação do software (Eckel, 2006)”.*

Já outras linguagens são consideradas mais puras, ou seja, obrigam a utilização de programação orientada a objetos, sendo construídas do zero focando-se nos aspectos da programação orientada a objetos, um exemplo disso e a linguagem Smalltalk.

De acordo com Horstmann, um objeto ou instancia de uma classe pode armazenar informações que é o resultado de operações previas. Essa informação pode determinar como o objetivo vai se comportar no futuro. O conjunto de toda a informação armazenada pelo objeto corresponde ao estado de objeto. O estado do objeto pode se alterar ao longo do tempo, mas somente quando uma operação que implica mudança de estado e executada (Horstmann, 2006).

Grande parte das linguagens de programação adota a Programação Orientada a Objetos de forma parcial, ou seja, sedem espaço para antigos modelos procedurais (Estruturada) de programação. Como o PHP, por exemplo, que pode ser utilizado orientado a objetos ou na forma de programação estruturada já que o PHP não obriga o desenvolvedor a programar somente orientado a objetos (Horstmann, 2006).

* + 1. **ANÁLISE ORIENTADA A OBJETOS**

Segundo Betencourt, a análise orientada a objetos consiste na criação de objetos que representam o problema a ser resolvido, desta forma as classes se relacionam e interagem umas com as outras utilizando os métodos, atributos e os mecanismos de comunicação destes objetos. Para se fazer a análise orientada a objetos é necessário fazer uma descrição das características dos objetos utilizados para descrever o problema (Betencourt, 2010).

*“A análise enfatiza uma investigação do problema e dos requisitos, em vez de uma solução, por exemplo, se desejamos um novo sistema online de comercialização, como ele será usado? Quais são as suas funções?”. (Larman ,2007).*

De acordo com Larman, a analise é um termo de significado amplo melhor qualificado como analise de requisitos (uma investigação dos requisitos) análise orientada a objetos (uma investigação dos objetos do domínio) (Larman, 2007).

Segundo Betencourt, os princípios básicos da analise orientada a objetos são: A função é descrita, o comportamento é representado, o domínio da informação é modelado, os modelos de dados funcionais e comportamentais são separados a fim de expor maiores detalhes, os modelos iniciais representam a essência do problema, e os últimos modelos expõem detalhes da implementação (Betencourt, 2010).

Segundo Larman, O projeto enfatiza uma solução conceitual (em software ou hardware) que satisfaça os requisitos e não sua implementação. Uma descrição de um esquema de banco de dados e objetos de software é um bom exemplo. Idéias de projeto excluem freqüentemente detalhes de baixo nível ou “óbvios” – óbvios para os consumidores visados. Em ultima instancia, projetos podem ser implementados e a implementação (como por exemplo, o código) expressa o verdadeiro e completo projeto realizado (Larmam, 2007).

* + 1. **PROJETO ORIENTADO A OBJETOS**

Betencourt diz que, o projeto orientado a objetos transforma o modelo criado na analise orientada a objetos em um modelo que se apresenta como um documento para a construção do projeto de software. O projeto orientado a objetos necessita de uma arquitetura de software com multicamadas, da especificação de subsistemas que exerçam funções críticas e que forneçam uma infra-estrutura de apoio ao projeto. A descrição dos objetos que formam os blocos do sistema, assim como, a descrição dos mecanismos de comunicação dos objetos que possibilitam a transição das informações entre as camadas também são essenciais para um projeto orientado a objetos (Betencourt, 2010).

A arquitetura Orientada a Objetos visa maximizar o reuso dos objetos isso com objetivo de aumentar a qualidade e diminuir o tempo de desenvolvimento do produto. Para isso os arquitetos necessitam da documentação gerada pelo projeto orientado a objetos estes artefatos são essências para a continuação do projeto.

Segundo Larman, o projeto orientado a objetos se preocupa com a definição de objetos de software e suas responsabilidades e colaborações. Uma notação comum para ilustrar essas colaborações e o diagrama de sequência (uma espécie do diagrama de interação da UML). Ele mostra o fluxo de mensagens entre os objetos de software e, assim, a invocação de métodos (Larman, 2007).

Um projeto orientado a objetos é dividido em quatro partes, cada uma dessas camadas é responsável por áreas especificas do projeto (Horstmann, 2006).

A Camada de subsistemas é a camada responsável por conter a representação de cada um dos subsistemas, essa camada permite ao software responder aos requisitos definidos pelos clientes e implementar a infra-estrutura para o suporte técnico dos requisitos. (Betencourt, 2010)

A Camada de classes e objetos esta camada tem a função de conter a representação de cada um dos objetos desta forma as hierarquias das classes são definidas nesta camada, com isso é possível a construir o software utilizando generalizações e especializações cada vez mais precisas (Betencourt, 2010).

A Camada de mensagens contém os detalhes do projeto orientado a objetos, estes detalhes possibilitam que cada objeto se comunique entre seus colaboradores. Sendo assim a camada de mensagens estabeleça as interfaces internas e externas do software (Betencourt, 2010).

Camada de responsabilidades é responsável por manter as estruturas de dados e os algoritmos dos atributos e métodos e todos os objetos do projeto orientado a objetos (Horstmann, 2006).

* + 1. **PROGRAMAÇÃO ORIENTADA A OBJETOS**

De acordo com Milane de padronização para expressar instruções para um computador. Conjunto de regras semânticas e sintáticas usadas para definir um programa de computador. Ela permite que permita que o programador sobre quais dados o computador via atuar (Milane, 2010).

Segundo Linden, as linguagens orientadas a objetos (LOO) atingem estes objetivos principais, permitindo grande reusabilidade de software e facilitando o desenvolvimento de sistemas corretos que atinjam as especificações desejadas. (Linden, 2008).

Segundo Milane a orientação a objeto, também é conhecida como POO (Programação Orientada a Objeto) é um modelo de desenvolvimento de software, que organiza as informações utilizadas como parte de informações maiores conhecidas como objetos (Milane, 2010).

Milane diz que, o objetivo da programação orientada a objetos é melhor identificar os conjuntos de objetos responsáveis por detalhar um produto de software. Este princípio de programação consiste em relacionamentos e trocas de mensagens entre os objetos. Na programação orientada a objetos um conjunto de classes é implementado. Essas classes são responsáveis por definir os comportamentos dos objetos isso através de métodos e atributos. (Milane, 2010).

* + 1. **LINGUAGEM PHP**

Segundo Dall`Oglio, o PHP é uma das linguagens mais utilizadas no mundo. Sua popularidade deve-se à facilidade em criar aplicações dinâmicas com suporte à maioria dos bancos de dados existentes e ao conjunto de funções que, por meio de uma estrutura flexível de programação, permitem desde a criação de simples portais até complexas aplicações de negócio. O uso da orientação a objetos juntamente com o emprego de boas práticas de programação nos possibilita manter um ritmo sustentável no desenvolvimento de aplicações (Dall`Oglio, 2007).

*“PHP é uma linguagem script de programação disponível de graça sob licença open source e utilizada principalmente em servidores web Linux como alternativa ao ASP da Microsoft. (LUKE,2005)”.*

Antes da descrição do Hypertext Preprocessor (PHP), é interessante abordar o passado desta linguagem. Anteriormente, as informações eram encontradas na internet através de ferramentas Gopher que hoje são insuficientes. O Gopher reinou sozinho até a tela e o primeiro browser, Mosaic, que tinha a capacidade de exibir texto e imagens, mostrando um novo caminho para formatar a informação contida nos servidores de uma forma mais agradável e ilustrada, mas a informação era estática.

Mas havia ainda a necessidade de páginas dinâmicas, dai então, surgiu o CGI (Common Gateway Interface) que são programas executados a pedido do navegador e a saída destes programas é enviada de volta para o navegador. Para exibir a data em uma página, era necessário executar um programa CGI no servidor, este programa irá ler a data do sistema e informações informando ao navegador que cuidava da tela. Esta dupla HTML (Hypertext Markup Language) / CGI durou até fins de 1998, início de 1999, e é usado hoje.   
Programas CGI foram escritas, principalmente, com o C e PERL (*Practical Extraction and Language Report*), causando uma série de problemas nos servidores. Se um site recebeu 15 pedidos por segundo, e executa um programa CGI em cada uma dessas visitas, 15 casos foram abertos pelo segundo, o que poderia fazer com que o servidor para passar mais tempo a gerir esses processos do que servindo páginas, segundo a qual havia sido projetado.

Para resolver o problema surgiu uma grande idéia para inserir os comandos que seriam o programa CGI na página HTML, para que o servidor enviando esta página, execute o bloco de comando, o mesmo processo. Então, para mostrar a data em nossa página, basta inserir, em algum ponto desta página, o comando para que a data foi inserida. Hoje podemos usar a linguagem de comandos, que é gratuita, portátil, aberta e escalável, que é PHP.

* + 1. **BANCO DE DADOS POSTGRESQL**

Dadas as suas características poderosas e avançadas, você pode se perguntar como uma peça valiosa de software passou a ser tanto livre e de código aberto. Como acontece com muitos outros importantes projetos de código aberto, a resposta começa na Universidade da Califórnia em Berkeley (UCB) (POSTGRES, 2010).

De acordo com o site oficial, o SGBD PostgreSQL, originalmente chamado de Postgres, foi criado na UCB por um professor de ciência da computação chamado Michael Stonebraker, que passou a se tornar o CTO da *Informix Corporation*. *Stonebraker Postgres* começou em 1986 com um projeto de acompanhamento ao seu antecessor, Ingres, agora propriedade da Computer Associates (POSTGRES, 2010).

Segundo o site oficial do PostGreSql, o nome de Postgres, foi retirado do seu antecessor Ingres, desenvolvido a partir de 1977 a 1985, tinha sido um exercício de criação de um sistema de banco de dados de acordo com a teoria clássica RDBMS. Postgres, desenvolvido entre 1986-1994, foi um projeto destinado a abrir novos caminhos em conceitos de banco de dados, tais como a exploração de "objeto relacional" tecnologias. (POSTGRES, 2010).

Stonebraker e seus alunos de pós-graduação desenvolveram ativamente o Postgres por oito anos. Durante esse tempo, o Postgres introduziu regras, procedimentos, tempo de viagem, tipos extensíveis como índices, objetos relacionais e conceitos. O Postgres foi posteriormente comercializado para se tornar Illustra que mais tarde foi comprada pela Informix e integrada em seu Servidor Universal. Informix foi comprada pela IBM em 2001 para um bilhão de dólares (POSTGRES, 2010).

Em 1995, dois PHDs alunos do laboratório de Stonebraker, Yu e Jolly Chen Andrew, substituíram a linguagem de consulta PostQuel do Postgres por um subconjunto estendido do SQL. A partir dai mudaram o sistema para Postgres95 (POSTGRES, 2010).

Em 1996, o Postgres95 saiu da academia e começou uma nova vida no mundo open source, quando um grupo de desenvolvedores dedicados fora de Berkeley viu a promessa do sistema, e se dedicaram ao seu desenvolvimento continuado. Contribuindo enormes quantidades de tempo, habilidades, trabalho e conhecimento técnico, esse grupo global de desenvolvimento radicalmente transformado Postgres. Ao longo dos próximos oito anos, eles trouxeram consistência e uniformidade à base de código, testes de regressão detalhados foram criados para a garantia de qualidade, criar listas de discussão para relatórios de bugs, erros corrigidos inúmeros, acrescentou novas funcionalidades incríveis, e arredondado para fora do sistema, preenchimento de lacunas vários como a documentação para desenvolvedores e usuários (POSTGRES, 2010).

Segundo o site postgres.org, o resultado de seu trabalho foi uma nova base de dados que *garnered* uma reputação de estabilidade rocha sólida. Com o início da sua nova vida no mundo do código aberto, com muitos novos recursos e aprimoramentos, o sistema de banco de dados teve seu nome atual: PostgreSQL (POSTGRES, 2010).

O SGBD PostgreSQL começou na versão 6.0, dando crédito a seus muitos anos de desenvolvimento anterior. Com a ajuda de centenas de desenvolvedores de todo o mundo, o sistema foi modificado e aprimorado em quase todas as áreas. Ao longo dos próximos quatro anos (versões 6,0-7,0) importantes melhorias e novas funcionalidades foram implementadas (POSTGRES, 2010).

O postgres.org diz que, o controle multi-versão concorrência (MVCC) foi umas das implementações que melhoram e muito o PostgreSQL. O nível de bloqueio de tabela foi substituído por um sofisticado sistema de controle de concorrência multi-versão, que permite aos leitores continuar a ler dados consistentes durante a atividade de escrita, possibilitando backups online (backups quentes) enquanto o banco está em execução (POSTGRES, 2010).

De acordo com o site oficial do SGBD PostgreSQL, importantes características SQL foram incluídas, muitas melhorias SQL foram feitas incluindo subconsultas, padrões, restrições, chaves primárias, chaves estrangeiras, identificadores entre aspas, a coerção de tipo literal string, fundição, tipo e binário e de entrada inteiro hexadecimal entre outros e novos tipos nativos foram adicionados, incluindo uma ampla gama de tipos de data / hora e adicional tipos geométricos (POSTGRES, 2010).

Quanto ao desempenho, grandes aumentos de velocidade e desempenho na casa dos 20-40% foram feitas, e tempo de arranque do backend foi reduzida em 80%.   
 Os quatro anos seguintes (versões 7,0-7,4) trouxeram o Write-Ahead Log (WAL), esquemas SQL, consultas preparadas, junções externas, consultas complexas, SQL92 juntar sintaxe, TOAST, suporte IPv6, o esquema do padrão SQL informações, de texto completo indexação, auto-vácuo, Perl / Python / TCL linguagens procedurais, suporte SSL melhorado, uma revisão otimizada, informações estatísticas do banco de dados, acrescentou tabela de segurança, funções e melhorias de registro e melhorias significativas de velocidade, entre outras coisas. Um pequeno grau de desenvolvimento intensivo do PostgreSQL se reflete em suas notas de lançamento (POSTGRES, 2010).

“*O PostgreSQL é um poderoso sistema gerenciador de banco de dados objeto-relacional de código aberto.  Tem mais de 15 anos de desenvolvimento ativo e uma arquitetura que comprovadamente ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões.* (2010,www.postgresql.org.br) ”.

Hoje, a base de usuários do PostgreSQL é maior do que nunca e inclui um grupo considerável de grandes empresas que a utilizam em ambientes exigentes. Algumas destas empresas como a Fujitsu Afilias e fizeram contribuições significativas para o desenvolvimento do PostgreSQL. E, fiel às suas raízes, ela continua a melhorar em ambos os sofisticação e performance, agora mais do que nunca. Versão 8.0 é a estréia de longa PostgreSQL é aguardado no mercado de banco de dados da empresa, trazendo recursos como tablespaces, Java procedimentos armazenados, ponto no tempo de recuperação, e transações aninhadas (savepoints) (POSTGRES, 2010).

Muitas organizações, agências governamentais e empresas utilizam o PostgreSQL que pode ser encontrados em organizações e empresas internacionais como: ADP, CISCO, NTT Data, NOAA, Research In Motion, O Serviço Florestal dos Estados Unidos e a Sociedade Americana de Química. Atualmente, é difícil encontrar uma grande empresa ou agência do governo que não está usando PostgreSQL em pelo menos um departamento (POSTGRES, 2010).

Segundo a site do PostgresSQL, as principais características do PostgreSQL 8.2 são: suporte a Stored Procedures, Triggers, possibilidade de configuração de cursores, possui tipo de dados para moeda (“money”), uso de domains, possui recuperação no tempo (point-in-time), views atualizáveis, transações incluindo *savepoints*, portável (roda em *Linux, Unix (AIX, BSD, HP-UX, SGI, IRIX, Mac OS X, Solaris, Tru64*) e *Windows*), possui plataforma de administração e desenvolvimento (*pgAdmin*). Além disso, um recurso importante e de extrema necessidade é o tipo de bloqueio adotado. O bloqueio serve para impedir que duas transações simultâneas acessem o mesmo recurso ou dado ao mesmo tempo. O *PostgreSQL* utiliza o “*optimistic locking*”. Resumidamente, permite que dados possam ler do banco e escrever sem utilizar explicitamente um bloqueio. Isto permite isolar a leitura da escrita. Caso o “*optimistic locking*” não funcione adequadamente, pode-se utilizar explicitamente o bloqueio de tabela ou linha (POSTGRES, 2010).

* + 1. **CSS**

O CSS Cascading Style Sheets ou folhas de estilo é uma linguagem para formatação de documentos web. O CSS pode combinar descrições de estilo e técnicas diferentes, ao qual é dado nome de cascata, já que muitas folhas de estilo se juntam a uma única folha para resultarem no resultado final (Silva, 2007).

Segundo o Site Maujor.com, CSS é a sigla de *Cascading Style Sheet*. CSS é definido com folha de estilo em cascata, é um mecanismo simples para adicionar fontes, espaçamentos, formatações de tabela, cores em outros em documentos web (Silva, 2007).

De acordo com o site Maujor.com, a vantagem do CSS é a possibilidade de separar o HTML que é uma linguagem de marcação da apresentação visual do site. Ou seja, a função do HTML é de apenas estruturar e marcar o conteúdo, ficando por conta do CSS todas as outras responsabilidades de apresentação visual do documento web (Silva, 2007).

O CSS é utilizado em sistemas web para facilitar o desenvolvimento e a manutenção da parte visual do mesmo, por exemplo, a alteração de um template (tema) de um sistema para promover a acessibilidade ou até mesmo a reestruturação das paginas do sistema (Silva, 2007).

O Maujor.com diz também que o HTML é responsável por marca e estrutura textos, imagens, parágrafos, formulários, botões, links, cabeçalhos e demais elementos do documento web enquanto o CSS define as cores, posicionamento dos componentes na tela, estilos de linhas, estilos de bordas e os demais componentes visuais que possam existir (Silva, 2007).

Segundo o site Marjor.com, ao utilizar as técnicas do CSS corretamente ao invés de utilizar atributos dentro de *tags* HTML a manutenção deste documento web se torna mais fácil (Silva, 2007).

* + 1. **XHTML**

De acordo com Silva, o XHTML é uma recomendação do W3C. O XHTML igual ao XML não permite erros de sintaxe no corpo do código diferente do HTML que por sua tente a ter seus erros ignorados pelos browsers (Silva, 2007).

*“XHTML é a abreviação para* ***eXtensible Hyper Markup Language****, grafia em inglês que, no jargão da internet, foi traduzido para Linguagem Extensível de Marcação para Hipertexto e que se destina a escrever documentos web com a funcionalidade adicional de ser compatível com as aplicações XML.” (Silva, 2007).*

Exemplo de Codificação XHTML:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; />

<title>Titulo da Pagina</title>

</head>

<body>

</body>

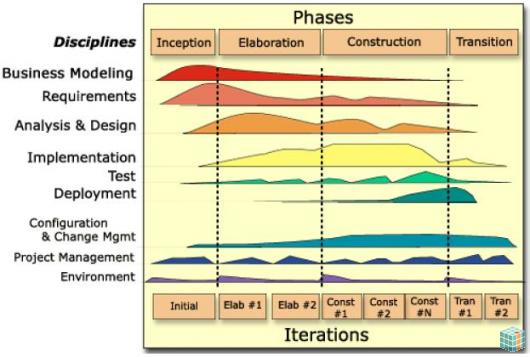
</html>

* + 1. **IRUP**

Segundo Fowler, IRUP é uma estrutura de processos fornecidos pela (*Rational Unified Process*) fornece um vocabulário e uma vaga estrutura sobre processos define o que e como fazer ele e basicamente uns processos interativos (Fowler, 2005).

Assim com mencionado no site www.wthreex.com, O Rational Unified Process® (também chamado de processo RUP®) é um processo de engenharia de software.  Ele oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento.  Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis (Fowler, 2005).

O IRUP é uma estrutura de processos iterativos para permitir a produção de software adequando à UML para suprir a necessidade do cliente/usuário, onde serão definidas as tarefas com eficiência de cada iteração. (BOOCH, 2005; MARTIM, 2005).



Figura

Fonte

O RUP é inicialmente dividido em quatro fases: iniciação, elaboração, construção e transição. E também e composto de nove disciplinas presentes em quase todas as fases citadas essas disciplinas são: modelagem de negócios, requisitos, analise e designer, implementação, teste, implantação, gerencia de configuração e mudança gerenciamento de projeto e por fim Ambiente (FOWLER, 2005).

Concepção: É nesta fase do projeto que se estabelece a visão, escopo e o plano inicial para o projeto, ou seja, nesse momento ocorrem as reuniões para o levantamento dos requisitos do sistema assim como a troca de informações que estarão presentes em todo o ciclo do projeto (FOWLER, 2005).

Elaboração: Esta fase prever o uma maior envolvimento de toda equipe para se discutir a análise do domínio do problema, da arquitetura, do desenvolvimento do plano do projeto assim como a elaboração dos artefatos do projeto que irão auxiliar em sua construção. Nesta fase também se tem a necessidade de avaliar, eliminar ou amenizar alguns os riscos do projeto (FOWLER, 2005).

Construção: Nesta fase se constrói um produto completo, para que essa fase ocorra sem grandes imprevistos é necessário que as duas fases anteriores interajam corretamente. E assim em seguida se conclua a implementação e os testes do software (FOWLER, 2005).

Transição: E por fim a fase da transição que é o momento em que o sistema será disponibilizado aos clientes para demonstração, correção e ajustes para a finalização do sistema que, será substituído pelo sistema de produção (FOWLER, 2005).

No Desenvolvimento deste projeto foram utilizados alguns artefatos do IRUP como: plano de risco, documento de visão do projeto e cronograma de desenvolvimento do projeto. Quanto às faces do processo IRUP foram utilizadas na medida do possível já que o projeto é relativamente pequeno (FOWLER, 2005).

* + 1. **JAVASCRIPT**

O JavaScript foi desenvolvido por Brendan Eich da Netscape sob o nome de *Mocha*, e passou a ser chamado de *LiveScript* por algum tempo, e depois de alguns anos finalmente passou a ser chamado de *JavaScript como e conhecido hoje*. Em setembro de 1995 o LiveScript teve o lançamento oficial do seu beta para os navegadores Netscape 2.0 e em dezembro de 1995 teve seu nome mudado em um anúncio conjunto com a Sun Microsystems.

*“Netscape, depois de fazer seus navegadores compatíveis com os applets, começou a desenvolver uma linguagem de programação ao que chamou LiveScript que permitisse criar pequenos programas nas páginas e que fosse muito mais simples de utilizar que Java. De modo que o primeiro Javascript se chamou LiveScript, mas não durou muito esse nome, pois antes de lançar a primeira versão do produto se forjou uma aliança com Sun Microsystems, criador de Java, para desenvolver em conjunto essa nova linguagem.”(* *HTMLSTAFF,2006).*

De acordo com o que consta no site htmlstaff.org, a aliança fez com que Javascript se desenhara como um irmão pequeno de Java, somente útil dentro das páginas web e muito mais fácil de utilizar, de modo que qualquer pessoa, sem conhecimentos de programação pudesse aprofundar-se na linguagem e utilizá-la. Ademais, para programar Javascript não é necessário um kit de desenvolvimento, nem compilar os scripts, nem realizá-los em ficheiros externos ao código HTML, como ocorreria com os applets (HTMLSTAFF, 2006).

* + 1. **UML**

O UML é uma linguagem de modelagem unificada, ou seja, é uma linguagem visual para especificar, construir e documentar os artefatos de um sistema.

“O padrão UML indica que certos elementos normalmente são desenhados em determinados tipos de diagramas, mas isso não é uma regra.” (Fowler,2005).

De acordo com Larman, a palavra visual na definição é um ponto chave – UML é a notação diagramática padrão, de fato, para desenhar ou apresentar figuras (com algum texto) relacionadas a software – principalmente software OO (Larman, 2007).

Segundo Larmam a UML define vários perfis UML que especificam subconjuntos da notação para áreas de assunto comum, tais como diagramação de Enterprise JavaBeans (com o perfil UML EJB) (Larman, 2007).

*“A UML nasceu da unificação de muitas linguagens gráficas de modelagem orientadas ao objeto que floresceram ao final dos anos oitenta, início dos noventa. Desde sua aparição em 1997, ela fez com que essa torre de Babel fosse resolvida...” (Fowler ,2005).*

O diagrama de classe é responsável por descrever os tipos de objetos utilizados no sistema e os possíveis relacionamentos entre eles. O digrama de classe também tem a função de mostrar as propriedades e as operações de uma classe assim as suas restrições na maneira em que os objeto se comunicam (Fowler, 2005).

Diagramas de seqüência são os diagramas responsáveis por informar o comportamento de um determinado cenário. Nesses diagramas também é mostrado diversos exemplos de mensagens que trafegam as informações entre os objetos de um caso de uso especifico (Larman, 2007).

A divisão de componentes reutilizáveis e suas dependências internas de um sistema são de responsabilidade dos diagramas de componentes (Larman, 2007).

Outro diagrama muito utilizado no processo de modelagem dos sistemas é o diagrama de atividades. Os diagramas de atividades são utilizados para esclarecer os fluxos de controle ou de atividades em operações complexas em um caso de uso. Geralmente os digramas de Atividades são confundidos com fluxogramas, apesar de serem semelhantes à diferença principal é a possibilidade que os diagramas de atividades têm de suportar fluxos paralelos (Larman, 2007).

Os diagramas de casos de uso são técnicas de auxilio na captura de requisitos funcionais de um sistema. Sua função é descrever as interações entre os usuários e o sistema.

A UML possui varias ferramenta que auxiliam a modelagem do projeto, neste projeto foram aplicadas algumas destas ferramentas conhecidas como diagramas. Os diagramas utilizados foram: diagrama de caso de uso, diagrama de seqüência, diagrama de atividades, diagrama de redes e diagrama de implantação (Fowler, 2005).

* + 1. **MVC**

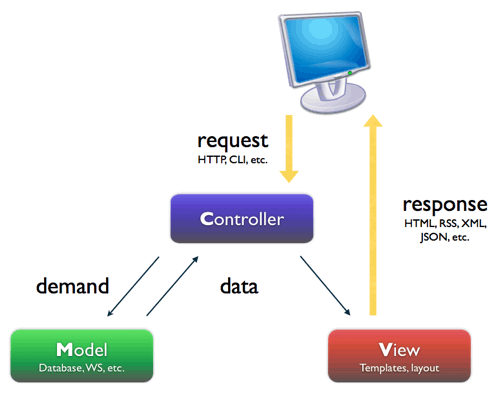
O MVC é um padrão de arquitetura de software que promove a separação em camadas as aplicações. Esta separação funciona separando a parte de lógica de negocio da parte de visão ou apresentação da aplicação assim como a parte de persistência com o banco de dados (OFICINANET, 2010).

O MVC se baseia no desenvolvimento de aplicações separadas por camadas, onde o mesmo defende que o modelo é um objeto que representa algumas informações sobre um domínio que na maioria das vezes representa uma entidade (tabela). Na orientação a objetos, o modelo é um objeto representa um modelo de dados de qualquer natureza (OFICINANET, 2010).

O MVC permite que os desenvolvedores possam testas, desenvolver e efetuarem manutenções isolando ambos.

O padrão MVC é constituído de três partes que são: modelos (*model*), controladores (*controllers*) e visões (*views*).

A um dos benefícios da separação da aplicação em três camadas é torná-las independentes e desacopláveis, além de possibilitar a reutilização do mesmo *model* em várias *views*. A Figura abaixo ilustra como é funciona a comunicação entre as três camadas do MVC.



Fonte: (http://1.bp.blogspot.com/-P8ZnMpKjpik/Tfn4Bf0gMyI/AAAAAAAAAW0/2E5buDgPHIg/s1600/mvc.png)

Figura 2

* + 1. **W3C**

De acordo com o site do W3C, o W3C cumpre sua missão com a criação de padrões e diretrizes para a Web. Desde 1994, o W3C publicou mais de 110 desses padrões, denominado recomendações do W3C. O W3C também se envolve em educação e divulgação, desenvolve softwares e atua como fórum aberto para discussões sobre a Web. Para que a Web atinja todo o seu potencial, as tecnologias mais fundamentais da Web precisam ser compatíveis entre si e permitir que todos os equipamentos e softwares usados para acessar a Web funcionem juntos. O W3C chama essa meta de “Interoperabilidade da Web”. Ao publicar padrões abertos (não-exclusivos) para línguas e protocolos da Web, o W3C procura evitar a fragmentação do mercado e, conseqüentemente, a fragmentação da Web (JACOBS, 2010).

*“O Consórcio World Wide Web (W3C) é um consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a Web. “(w3c.com, 2010).*

Segundo o site do W3C, Organizações do mundo todo e envolvidas em muitos campos diferentes se reúnem no W3C para participar de um fórum vendor-neutral (ou seja, que não é vinculado a nenhum fabricante) para criar padrões para a Web. Os Filiados do W3C e uma equipe de especialistas técnicos que trabalham em tempo integral levaram o W3C a ser reconhecido em nível internacional por suas contribuições para a Internet. Os Filiados do W3C (amostra de depoimentos), a equipe e os especialistas convidados trabalham juntos para desenvolver tecnologias que garantam que a Web continuará crescendo no futuro, acomodando a diversidade cada vez maior de pessoas, equipamentos e softwares (JACOBS, 2010).

Neste projeto foram aplicados os padrões de cores, fontes e disposição das informações para a plataforma web. O W3C diz que a padronização de cores e fontes visa facilitar a visibilidade e navegabilidade das aplicações web. Quanto a disponibilização de conteúdo deve ser feita de forma ordena seguinte uma linha de pensamento, por exemplo: A disposição dos menus deve ser feita na vertical ou horizontal de forma fixa.

As tonalidades das cores do sistema estão de comum acordo com as normas no W3C que orienta que cores fortes e fluorescentes não sejam utilizadas já que as mesmas causam cansaço visual e podendo causas ate ataques de epilepsia em pessoas com predisposição (JACOBS, 2010).

O site do W3C diz que as iniciativas globais do W3C também incluem a formação de laços com organizações nacionais, regionais e internacionais do mundo todo. Esses contatos ajudam o W3C a manter uma cultura de participação global no desenvolvimento da *World Wide Web*. O W3C trabalha em estreito contato principalmente com outras organizações que desenvolvem padrões para a Internet, a fim de possibilitar um progresso claro. O documento Worldwide Participation in the World Wide Web Consortium (Participação mundial no Consórcio World Wide Web) resume as iniciativas do W3C para ampliar seu impacto. Veja nossa página sobre relações internacionais para mais informações (JACOBS, 2010).

* + 1. **SERVIDOR WEB (APACHE)**

Segundo o site (Apache.org), o Servidor Apache é um software voltado ao processamento de script php, o Apache é o servidor web livre mais aceito e reconhecido no mundo nos dias de hoje (APACHE, 2011).

De acordo com o site oficial do Apache, o Apache foi criado em 1995 por Rob McCool funcionário da tão conhecida *National Center for Supercomputing Applications* (NCSA). Foi constatado em um pesquisar realizada em meados de 2007 que os servidores Apache representam aproximadamente 48% dos servidores web do mundo. Já em 2010 a utilização do Apache para servir aplicações web chegou a aproximadamente 55% de todos os sites disponíveis no mundo inteiro (APACHE, 2011).

Segundo o (Apache.org), Apache é considerado a principal tecnologia da ASF (Apache Software Foundation). O Apache Software Foundation é responsável por mais de 12 projetos que envolvem tecnologias de transmissão de dados via web, execução de aplicativos distribuídos e processamento de dados (APACHE, 2011).

Segundo o site (Apache.org), o servidor Apache é inteiramente compatível com protocolos HTTP. As funcionalidades do Apache são mantidas por meio de uma estrutura que utiliza módulos para separar e melhor gerenciar as funcionalidades. O Apache com a sua estrutura em módulos também possibilitam aos usuários que eles criem seus próprios módulos e os integrem ao Apache isso só é possível graça a API de desenvolvimento do Apache (APACHE, 2011).

Os servidores Apaches que existem hoje no mundo são executados em vários tipos de plataformas como, por exemplo: Windows, Novell Netware, Linux, Unix, Posix entre outras plataformas mais comuns (APACHE, 2011).

O servidor Apache possui um arquivo de configuração único que facilita e muito a vida dos desenvolvedores e arquitetos. O Apache utiliza o arquivo *httpd.conf* para armazenar parâmetros de configuração de seu ambiente (APACHE, 2011).

O Apache possui um modulo especifico para tratar as solicitações HTTPS este modulo chama-se *mod\_ssl*. Este módulo garante a segurança nas transações utilizando o protocolo HTTPS (APACHE, 2011).

O protocolo HTTPS utiliza uma camada de criptografia de dados que é transferido entre cliente e servidor, isso visa garantir mais segurança aos dados que trafegam pelas requisições. O SSL é certificado pela X509. X509 é uma das maiores certificadoras digitais do mundo (APACHE, 2011).

* + 1. **FERRAMENTAS**
       1. **NETBEANS 7.0.1**

Segundo o site oficial do NetBeans, o NetBeans é um ambiente de desenvolvimento desenvolvido em Java e que suporta a varias linguagem de programas como: C#, C++, PHP, JAVA em outras. O NetBeans é uma ferramentas que auxilia os desenvolvedores a escrever, compilar, depurar e instalar aplicações. O NetBeans foi arquitetado em forma de estrutura reutilizável que tem como objetivo facilitar o desenvolvimento e aumentar a produtividade pois junta em uma única aplicação todas as funcionalidades necessárias (NETBEANS, 2010).

*“O NetBeans fornece uma base sólida para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e API´s (Application Program Interface, um conjunto de rotinas, protocolos e ferramentas para a construção de aplicativos de software) além de uma documentação vasta inclusive em português bem organizada, tais ferramentas auxiliam o desenvolvedor de forma a escrever seu software de maneira mais rápida. A distribuição da ferramenta é realizada sob as condições da SPL (Sun Public License), uma variação da MPL (Mozilla Public License), esta licença tem como objetivo garantir a redistribuição de conhecimento à comunidade de desenvolvedores quando novas funcionalidades forem incorporadas à ferramenta.” (NETBEANS, 2010).*

* + - 1. **POWER DESIGNER 15.0**

De acordo com o site Sybase.com.br, o PowerDesigner é uma a ferramenta de modelagem de dados, permite que as empresas visualizem, analisem e manipulem mais facilmente metadados para uma arquitetura eficiente de informações corporativas (SYBASE, 2011).

O Sybase.com.br diz também que, o PowerDesigner para arquitetura corporativa também fornece uma abordagem controlada por modelo para o alinhamento de negócios e TI, o que facilita a implementação de informações eficientes e arquiteturas corporativas. Ele oferece técnicas poderosas de gerenciamento de análises, design e metadados para a empresa (SYBASE, 2011).

Segundo Sybase.com.br, o PowerDesigner combina várias técnicas de modelagem padrão (UML, Business Process Modeling e Data Modeling, líder do mercado) em conjunto com plataformas de desenvolvimento líderes como, por exemplo, .NET, WorkSpace, PowerBuilder, Java e Eclipse, para oferecer soluções de design de bancos de dados formais e análise de negócios. E funciona com mais de 60 sistemas de gerenciamento de bancos de dados relacionais (SYBASE, 2011).

* + - 1. **APACHE JMETER 2.3**

Apache JMeter pode ser usado para testar o desempenho tanto em recursos estáticos e dinâmicos (arquivos, Servlets, scripts Perl, Java Objects, Bases de Dados e Consultas, servidores FTP e muito mais). Ele pode ser usado para simular uma carga pesada em um servidor de rede, ou objeto para testar a sua força ou para analisar o desempenho global no âmbito de diferentes tipos de carga. Você pode usá-lo para fazer uma análise gráfica de desempenho ou para testar o seu servidor / script / comportamento do objeto sob carga pesada simultâneos. (*Traduzido por Google Tradutor*) (JMETER, 2011).

* + - 1. **ACUNETIX WEB SECURITY SCANNER 7**

Segundo o site oficial do Acunetix Web Security Scanner 7, o Acunetix é uma ferramenta focada nos testes de segurança de web sites e sistemas web. O Acunetix Web Security Scanner 7 é composto de inúmeros módulos e procedimentos que são responsáveis por efetuar os testes de vulnerabilidade individualmente (ACUNETIX, 2011).

O Acunetix Web Security Scanner 7 procura vários tipos de falhas entre leves e críticas, entre os ataques que ele verifica estão: Cross Scripting, Ddos, sql injection e php injection. Esta ferramenta auxilia os desenvolvedores de sistema web na protecao contra ataques de hackers informando eventuais falhas de segurança (ACUNETIX, 2011).

* + - 1. **TORTOISESVN 1.6**

Segundo Alencar, o subversion é um sistema de controle de revisões geralmente usado para manter versionado códigos fontes (ALENCAR, 2011).

De acordo com Alencar, o TortoiseSVN é um implementação gráfica do subversion. O uso do TortoiseSVN provê: O versionamento de diretórios, histórico de versão, efetivações atômicas conhecidas com commits, versionamento de metadados, escolha de camadas de rede, manejo de dados consistente entre outros benefícios (ALENCAR, 2011).

* + - 1. **JCROP 0.9**

SegundoYogodoshi, o Jcrop é um plugin para jQuery que possibilita efetuar cortes em imagens de maneira simplificada (YOGODOSHI, 2008).

Jcrop é a maneira rápida e fácil de adicionar a funcionalidade de imagem de corte para sua aplicação web. Ele combina a facilidade de uso de um plugin jQuery típico com um poderoso motor de DHTML multi-plataforma de corte, que é fiel aos gráficos aplicativos de desktop conhecidos. (*Traduzido por Google Tradutor*) (DEEP LIQUID GROUP, 2011).

* + - 1. **ENTERPRISE ARCHITECT 7.5**

O Enterprise Architect, é um ambiente de modelagem UML. Baseasse no ciclo de desenvolvimento completo de produtos, com alto desempenho, ferramentas visuais para modelagem de negócios, engenharia de sistemas, arquitetura corporativa, gerenciamento de requisitos, projeto de software, geração de código, testes e muito mais. Uma ferramenta de ciclo de vida completo para integrar sua equipe e trazer a sua visão compartilhada para a vida. (*Traduzido por Google Tradutor*) (EA, 2011).

* + 1. **FRAMEWORKS**
       1. **EXTJS 2.0**

O EXTjs é uma framework desenvolvida na linguagem *javascript* de código livre, disponibilizado sob a licença LGPL.

O Extjs surgiu a partir da criação de uma extensão da tão conhecida framework YUI (Yahoo User Interface) e até então era conhecida como YUI-EXT e que logo dependia diretamente da Framework YUI. O reconhecimento da YUI-EXT foi muito grande entre os desenvolvedores já que esta extensão possuía inúmeras funcionalidades e componentes de interface super elegantes e funcionais. Devido o grande sucesso o YUI-EXT passou a ser independente do YUI e se transformou no ExtJs.

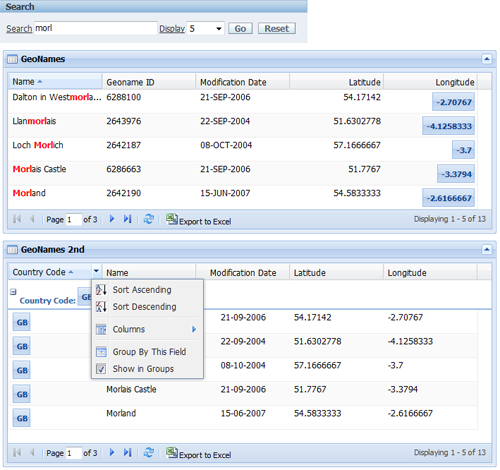


Figura 3

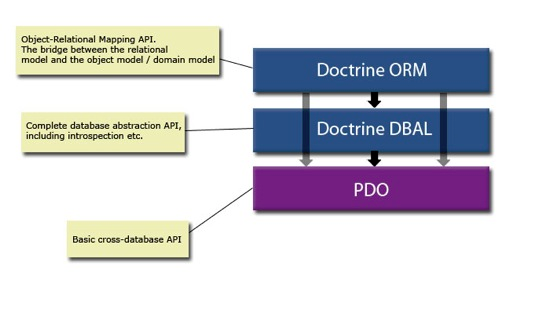
Fonte 2

O ExtJs desde que foi oficialmente lançado não teve muitas atualizações, agora o ExtJs possui um ext-base e que se tornou a base do framework e possui inúmeras funcionalidades que pode se esperar de um framework em javascript.

* + - 1. **DOCTRINE 1.2**

Segundo o site oficial Doctrime.com, O Doctrine é um conjunto de bibliotecas baseadas no mapeamento objeto-relacional de entidades de bancos de dados e desenvolvidas em linguagem PHP. Esse conjunto de bibliotecas que são focadas principalmente na prestação de serviços de persistência e funcionalidades relacionadas em banco de dados, ou seja, O Doctrine é uma Framework desenvolvida na linguagem PHP afim de auxiliar os desenvolvedores em operações com bancos de dados (Doctrine, 2011).

O Framework Doctrine oferece uma grande facilidade no desenvolvimento de aplicação na linguagem PHP utilizando bancos de dados. O Doctrime visa promover a transparência nas operações com bancos de dados já que o mesmo abstrair os recursos necessários para a manipulação das informações dos bancos de dados mais comuns (Doctrine, 2011).



Fonte 3

Figura 4

O Framework Doctrine possui vários recursos interessantes um deles tende a facilitar mais ainda a vida dos desenvolvedores PHP.O DQL (*Doctrine Query Language*) é um recurso que permite escrever consultas de bancos de dados em uma linguagem orientada a objetos própria do Doctrime. Essa linguagem inspira-se no Hibernate Query Language do Framework Hibernate que é voltado para a linguagem Java. Com este tipo de recurso é possível desenvolver aplicações poderosas e ao mesmo tempo manter a flexibilidade sem duplicações de códigos (Doctrine, 2011).

* + - 1. **ZEND FRAMEWORK 1.11**

De acordo com Lisboa, o maior esforço do desenvolvimento de sistemas não está na criação, mas na manutenção. As aplicações tornam-se cada vez mais complexas, e os requisitos dos clientes alteram-se várias vezes, antes de o projeto estar concluído. É preciso uma estrutura que permita a reutilização de código-fonte e o desenvolvimento simultâneo de partes do sistema, além de desvincular a aplicação do banco de dados, de forma que este possa ser trocado com o menor impacto possível. Zend Framework vem ao encontro desses problemas com a proposta de criar uma arquitetura flexível que permite o desenvolvimento de aplicações web MVC em PHP 5 com código reutilizável e mais fácil de manter, permitindo que os desenvolvedores concentrem-se nas regras de negócio do cliente. Este livro descreve passo a passo os fundamentos do framework por meio da criação de uma aplicação web completa (Lisboa, 2009).

O diagrama a seguir mostra o fluxo de trabalho, e as seguintes narrativas que descreve em detalhes as interações:

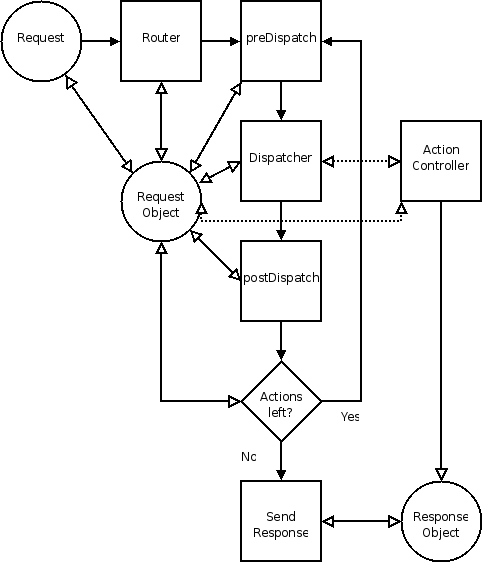


Figura 5

Fonte 4

ZendFramework é um conjunto de bibliotecas para o desenvolvimento de aplicações web com código fonte *OpenSource*.

O ZendFramework é baseado na implementação de aplicações web na linguagem PHP Orientado a Objetos. O principal objetivo do ZendFramework é facilitar e padronizar o desenvolvimento de aplicações web enquanto promove as boas praticas de programação já adotadas no universo de desenvolvimento da linguagem PHP (Lisboa, 2009).

O ZendFramework possui uma arquitetura padrão, mas também possibilita aos desenvolvedores ou arquitetos modificarem a mesma. A arquitetura do ZendFramework visa promover a reutilização de componentes sempre que possível. Outra vantagem do ZEndframework é a possibilidade de utilizar outros framework embutidas no seus escopo, como por exemplo: o ExtJs (Camadas de Visão) e o Doctrine (Camadas de Persistência) (Lisboa, 2009).

**3. ANÁLISE FUNCIONAL**

Este capítulo aborda os itens que compõem a análise funcional do Sistema de Camisetas Personalizadas, assim como os requisitos específicos e os benefícios esperados com este sistema.

**3.1. REQUISITOS DO NEGÓCIO**

Os requisitos deste projeto são baseados nas necessidades em um sistema que consiste em controlar o fluxo de produção de camisetas personalizadas de uma empresa de confecção.

**3.1.1. PAGAMENTO DOS PEDIDOS**

* Promover o controle do pagamento dos pedidos das camisetas personalizadas.
* Permitir que os clientes efetuem o pagamento de seus pedidos através de boleto bancário.

**3.1.2. RELATÓRIOS**

* Possibilitar que os Administradores do sistema extraiam relatórios gerenciais do sistema.
* Relatório de Produção Individual.
* Relatório de Produção Mensal.

**3.1.3. CONTROLE DE ACESSO**

* Permitir que os administradores do sistema alterem os dados dos clientes se necessário, incluído a desabilitarão do perfil em caso de abuso ou inadimplência do cliente.
* Disponibilizar aos administradores do sistema um módulo para se manter os perfis dos demais profissionais da empresa.

**3.1.4. MANUTENÇÃO**

* Possibilitar aos administradores do sistema a manutenção da galeria de produtos que serão ilustrados no site da empresa.

**3.1.5. CLIENTES**

* Informar o cliente através de email às tramitações de seus pedidos dentro da empresa de confecção.
* Facilitar a forma de aquisição das camisetas pelos clientes através de um site.
* Possibilitar que os clientes montem as estampas de suas camisetas de acordo com sua vontade.

**3.1.6. CONFECÇÃO**

* Construir uma sistemática que controle a atividade de confecção das camisetas personalizadas.

**3.1.7. POSTAGEM DE PEDIDO**

* Controlar o procedimento de envio dos pedidos aos clientes.

**3.2. ANÁLISE DAS NECESSIDADES E PROBLEMAS DIAGNOSTICADOS**

Foi diagnosticado que geralmente a necessidade de automatiza solicitações de confecção de camisetas personalizadas é grande. Com isso a tendência é desenvolver um sistema que atenda parcialmente este cenário. Um sistema informatizado facilita e as solicitações e também a metodologia de fabricação das referidas camisetas. Sendo assim este sistema torna-se a solução parcial ideal pra este problema e que será facilmente ampliado para atender necessidades que possam vir a existir.

**3.3. BENEFÍCIOS ESPERADOS**

Uma solução tecnologia para controlar as atividades em uma empresa de confecção de camisetas personalizadas traz grandes benefícios aos clientes, profissionais e donos da mesma. Espera-se que este projeto alcance um nível elevado de satisfação e benefícios aos seus utilizadores como:

* Uma diminuição significativa no tempo de atendimento dos pedidos dos clientes da empresa de confecção de camisetas.
* Aumentar a satisfação dos clientes possibilitando que os mesmo montem o layout de suas camisetas de acordo com suas necessidades.
* Centralizar as informações de clientes em um único banco de informações, para que quando necessário possam ser utilizadas e prol da divulgação de novos serviços e afins.
* Designar profissionais específicos para áreas especifica do processo de confecção dos pedidos dos clientes.
* Gerar relatórios gerenciais com informações úteis como, por exemplo: Nível de produção individual, Pedido Atendidos, Pedidos não Atendidos, Produtos mais vendidos entre outros.
* Possibilitar que o Administrador do sistema mantenha o site da empresa sempre atualizado, isso através da manutenção da galeria de produtos da empresa.
* Prestar satisfação ao cliente em tempo real à medida que o processo de confecção evolui.
* Aumentar a segurança da empresa tornando possível e obrigatório o pagamento dos pedidos via boleto bancário.
* Possibilitar que a confecção do pedido somente inicie quando o pagamento do mesmo for confirmado pelo profissional de conferencia de pagamento.
* Possibilitar o acesso aos administradores do sistema às informações dos profissionais da empresa.

**3.4. ANÁLISE DE RISCOS E PLANO DE CONTINGÊNCIA**

* *Vide Apêndice K.*

**3.5. PLANO DE CRONOGRAMA DO PROJETO**

* *Vide Apêndice J.*

**4. PROJETO LÓGICO E FÍSICO**

Este capítulo apresenta as descrições dos requisitos dos usuários do Sistema de Camisetas Personalizadas, assim como a abordagem dos requisitos de interface externa, requisitos de interface com ambiente de hardware, requisitos de interface com ambiente de software, requisitos de execução.

**4.1. REQUISITOS DO USUÁRIO**

**4.1.1 REQUISITOS FUNCIONAIS**

* Manter Autenticação.
* Manter Perfil do Cliente.
* Manter Galeria de Produtos da Empresa.
* Manter Galeria de Produtos do Cliente.
* Manter Produtos do Cliente.
* Consultar Galeria de Produtos.
* Solicitar Confecção.
* Gerar Boleto Bancário para Pagamento.
* Conferir Pagamento
* Confeccionar Produtos
* Despachar Produtos
* Manter Clientes
* Gerar Relatórios
* Manter Profissionais

**4.1.2 REQUISITOS DE INTERFACE EXTERNA**

* Email para informar o status dos pedidos aos clientes.

**4.1.3 REQUISITOS DE INTERFACE COM AMBIENTE DE HARDWARE**

* Impressora laser ou jato de tinta para impressão de relatórios e boletos.
* Impressora de estampagem em tecidos.

**4.1.4 REQUISITOS DE INTERFACE COM AMBIENTE DE SOFTWARE**

* Software da impressora instalado e devidamente configurada nos computadores dos profissionais.
* Navegador Web compatível com Javascript 1.5 o maior.
* Leitor de arquivos PDF para a geração de relatórios.

**4.1.5 REQUISITOS DE EXECUÇÃO**

* Número de transações simultâneas: *Aproximadamente 10*.
* Quantidade de estações de trabalho: *Mínimo 5 (Modulo Privado)*.
* Quantidade de usuários simultâneos: *Aproximadamente 10.*

**4.1.6 REQUISITOS GENÉRICOS**

* Nenhum Requisito Genérico (Não se aplica)

**4.1.7 REQUISITOS INVERSOS**

* Não possibilita o calculo dinâmico do valor do frete dos pedidos.
* Não contempla controle financeiro avançado (apenas alguns relatórios).
* Não contempla estorno de valores pagos pelos clientes.
* Não possibilita o pagamento de pedidos via cartão de credito.
* Não possibilita a operação por deficientes visuais.
  + 1. **REQUISITOS DE INTERFACE**
* *Vide Apêndice M*

**4.2. MODELAGEM FUNCIONAL E DE DADOS**

Modelagem de dados é uma forma de especificar a estrutura de dados e de regras de negocio necessárias para dar suporte a um domínio de negocio de um sistema de informação. A modelagem de dados representa um conjunto de informações baseadas nas informações de negocio.

**4.2.1. MODELO ENTIDADE RELACIONAMENTO – MODELO CONCEITUAL**

* *Vide Apêndice G.*

**4.2.2. MODELO ENTIDADE RELACIONAMENTO - MODELO FÍSICO**

* *Vide Apêndice H.*

**4.2.3. DICIONÁRIO DE DADOS – MODELO FÍSICO**

* *Vide Apêndice I.*

# 5. MODELAGEM UML

Este capítulo apresenta a os diagramas específicos da modelam lógica do sistema de camisetas personalizadas. De acordo com os requisitos levantados os seguintes diagramas foram elaborados: diagrama de caso de uso, diagramas de atividade, diagramas de seqüência, diagrama de implantação e diagrama de classes.

**5.1. DIAGRAMA DE CASO DE USO**

* *Vide Apêndice C.*

**5.2. DIAGRAMA DE ATIVIDADES**

* *Vide Apêndice D.*

**5.3. DIAGRAMA DE SEQÜÊNCIA**

* *Vide Apêndice E.*

**5.4. DIAGRAMA DE IMPLANTAÇÃO**

* *Vide Apêndice F.*

**5.5. DIAGRAMA DE CLASSES**

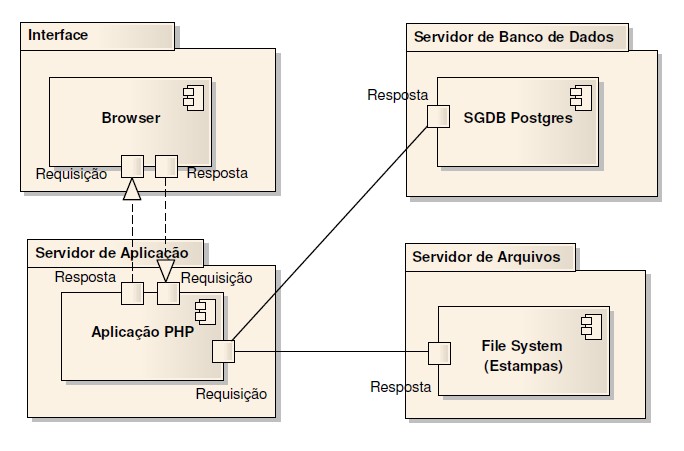
* *Vide Apêndice B.*

# 6. ARQUITETURA COMPUTACIONAL E CUSTO

Neste capítulo é apresentado a arquitetura computacional do Sistema de Camisetas Personalizadas e o custo envolvido para a construção deste projeto.

# 6.1. DIAGRAMA DE EXECUÇÃO E/OU COMPONENTES

O diagrama de componentes mostra a relação de interdependência e de comunicação entre seus relacionamentos por intermédio de interfaces. (MARTINS, 2007).



# 6.2. FORMAS DE ARMAZENAMENTO E DISPONIBILIZAÇÃO DOS DADOS

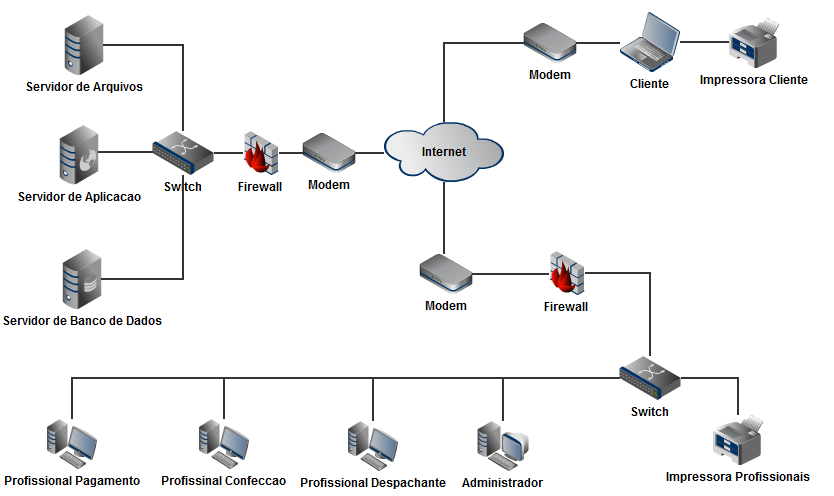
Recomenda-se que as imagens referentes às estampas do Sistema de Camisetas Personalizadas sejam armazenadas em um servidor de arquivos que não seja o servidor web e nem do servidor de banco de dados. O servidor web deve possuir um links simbólico ou pastas compartilhadas apontando para um servidor de arquivos.

Recomenda-se que o servidor de arquivos possua uma rotina de backups sendo ela uma rotina baseada em um *script shell* de cópia integral ou uma estrutura de redundância de dados com o padrão RAID.

É bastante recomendado que o banco de dados seja instalado em disponibilizado em um servidor dedicado. Este servidor também deve possui um sistema de backups automático e que de preferência seja baseado em LOGS do SGBD para que se necessário seja feita as restaurações do banco de dados.

O servidor de arquivos não necessita necessariamente de uma rotina de backups automáticos, o que é altamente recomendado é a utilização de uma solução para disponibilização interrupta do sistema e um balanceador de carga se necessário. O servidor web também deve possui interfaces de rede de alta velocidade.

# 6.3. DIAGRAMA DA REDE



# 6.4. SOFTWARES BÁSICOS E DE APOIO UTILIZADOS

* Firefox 6.0
* Chrome 11
* Opera 12
* Safari 5
* Internet Explorer 9.0
* PostGreSql 9.0
* Servidor de Web Apache 2.2
* Power Designer 15
* NetBeans 7.0
* Enterprise Architect 7.5

# 6.5. DETALHAMENTO DOS HARDWARES NECESSÁRIOS

* + 1. **HARDWARE DO CLIENTE**
* Requisitos Mínimos Recomendados:
* Processador Pentium III ou equivalente.
* Memória de 512 MB.
* Placa de rede de 10/100 MBPS.
* Monitor, mouse, teclado e impressora.

**6.5.2. HARDWARE DO SERVIDOR**

* Processador Intel Core 2 Duo 2.5 GHZ ou equivalente.
* Memória de 8 GB DDR3.
* Placa de rede 10/100/1000 MBPS
* Mouse e teclado.
  1. **DETALHAMENTO DE CUSTO**

O detalhamento do custo deste projeto tem como base os seguintes aspectos:

* Tabela de base salários de profissionais de TI (tecnologia da informação) foi retirada da página *http://info.abril.com.br/professional/salarios/* do site da revista info, *vide anexo A*.
* A média dos salários levando em consideração o valor pago por hora trabalhada vária de acordo com as atividades.
* O total de horas gastas na execução das atividades deste projeto baseasse na projeção e complexidade de cada atividade.
* O valor da hora trabalhada de cada profissional é obtida através da divisão do salário estabelecido na tabela (anexo A) por 30 e em seguida pela divisão por 8 (8hs = 1d) resultando então no valor da hora. Por exemplo. (*Analista de sistemas Jr. => 4.761,33 => 4.761,33/30/8 => 19,83 => ~20 reais*).
* O total do custo para cada atividade é o resultado da multiplicação das horas trabalhadas com o valor da hora de acordo com a atividade executada.
* A base salarial aplicada no detalhamento de custo do projeto é feita sobre os salários dos profissionais nos níveis Junior.
* Por fim se tem o total do custo do Sistema de Camisetas Personalizadas que é de **8713,5** (Oito mil setecentos e treze e cinqüenta centavos) reais assim como a tabela com os detalhes de casa atividade.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atividade | Custo Hora | Horas Gastas | Total Custo da Atividade | | |
| Conferir Pagamento | 11 reais/hora | 18 hs | | 198 reais |
| Confeccionar Pedido | 11 reais/hora | 29 hs | | 319 reais |
| Despachar Pedido | 11 reais/hora | 32 hs | | 352 reais |
| Manter Galeria Camisetas | 11 reais/hora | 25 hs | | 275 reais |
| Manter Cliente | 11 reais/hora | 17 hs | | 187 reais |
| Gerar Relatórios | 11 reais/hora | 10 hs | | 110 reais |
| Manter Profissional | 11 reais/hora | 25 hs | | 275 reais |
| Manter Autenticação | 11 reais/hora | 15 hs | | 165 reais |
| Consultar Galeria | 11 reais/hora | 22 hs | | 242 reais |
| Manter Perfil do Cliente | 11 reais/hora | 19,5 hs | | 214,5 reais |
| Manter Favoritos | 11 reais/hora | 13 hs | | 143 reais |
| Manter Carrinho | 11 reais/hora | 15 hs | | 165 reais |
| Definição da Estrutura | 16 reais/hora | 40 hs | | 640 reais |
| Definição do Modelo de Dados | 17 reais/hora | 34 hs | | 578 reais |
| Definição de Interfaces | 29 reais/hora | 50 hs | | 1450 reais |
| Documentação | 20 reais/hora | 170 hs | | 3400 reais |
| Total |  |  | | 8713,5 reais |

# 7. SEGURANÇA

O presente capítulo apresentará sugestões de segurança lógica e física para o Sistema de Camisetas Personalizadas.

**7.1. MECANISMOS DE SEGURANÇA DA LINGUAGEM DE PROGRAMAÇÃO E SGDB**.

O termo captcha vêem do inglês “Completely Automated Public Turing test to tell Computers and Humans Apart” e pode ser traduzido como “público completamente automatizado para diferenciação entre computadores e humanos”. O captcha tem a função de impedir que pessoas utilizem ferramentas para acessar de forma abusiva paginas web ou interfaces de sistemas. Os captchas também são conhecidos com anti-robôs já que dependendo do nível de complexidade do captcha torna-se impossível ser interpretado automaticamente por um computador. O captcha é usado na maioria das vezes em telas de autenticação de sistemas corporativos, blogs, site de download entre outros. Tudo isso no intuito de restringir o acesso das informações somente a humanos.

A linguagem de programação PHP possui alguns métodos de criptografia entre eles os que mais se destacam são: o MD5, o SHA1 e o Base64.

MD5 é a sigla de Message-Digest Algorithm 5 foi desenvolvido em 1991 por Ronald Rivest, o MD5 é um algoritmo de hash de 128 bits. O MD5 é utilizado por software para proteger senhas, verificar a integridade de arquivos entre milhares de aplicações. Uma aplicação bem conhecida é o md5sum que consiste em criar uma string *hash* de um arquivo.

O SHA ou Algoritmo de Hash Seguro é uma família de algoritmos, o algoritmo mais usado desta família é o SHA1. O SHA1 é utilizado em uma gama de aplicações e protocolos de segurança. Os protocolos TLS, SSL e SSH são exemplos de protocolos que utilizam a criptografia do SHA1, mas mesmo sendo tão reconhecido o SHA1 possui algumas vulnerabilidades comprovadas da mesma forma que o MD5.

O base64 se comparado ao MD5 e ao SHA1 é o mais simples, este método de codificação de dados e voltado para a transferência de informações pela internet, o seu funcionamento é constituído de 64 caracteres que são “de A até Z”, “de a até z” e “de 0 a 9” e o mesmo pode ser facilmente descriptografado.

O recurso register\_globals do PHP oferece grande risco a integridade do sistema. O register\_globals possibilita a criação de variáveis do PHP a partir do envio da requisição ao servidor que hospeda a aplicação, por exemplo:

Acessando a http://dominio.com.br/index.php?function=kill com o register\_globals habilitado, automaticamente o PHP irá instanciar uma nova variável com o nome de “function” contendo o valor “kill”.

Este recurso do PHP já é considerado obsoleto já que não través benefício algum pelo contrario representa um grande risco de segurança e integridade do ambiente.

O SQL Injection é uma técnica de invasão de sistemas geralmente aplicada contra sistemas web, mas pode ser utilizada em qualquer linguagem de programação. O SQL Injection ocorre quando o atacando malicioso consegue inserir um fragmento de SQL ou ate mesmo uma instrução inteira dentro de uma consulta isso através das entradas de dados vulneráveis no sistema.

# 7.2. MECANISMOS DE SEGURANÇA E PRIVACIDADE DE DADOS.

* Recomenda-se a utilização de um firewall para restringir os acessos ao servidor que o sistema estará instalado.
* Recomenda-se a utilização de um sistema operacional mais seguro. Como por exemplo: Linux.
* Recomenda-se que o servidor onde o sistema será instalado utilize um antivírus confiável e com rotina de atualização automática.
* Recomenda-se que as pessoas que irão manipular o servidor estejam devidamente credenciadas e autorizadas.

**7.3. MECANISMO DE SEGURANÇA DE EQUIPAMENTOS E INSTALAÇÕES.**

* Recomenda-se para assegurar à integridade física do ambiente onde será hospedado o Sistema de Camisetas Personalizadas a utilização de um controle de acesso à sala onde fica o servidor.
* Recomenda-se a utilização de no-breaks para impedir a interrupção do sistema.
* Recomenda-se a utilização de um sistema de prevenção e combate a incêndios.
* Recomenda-se a utilização de controle de temperatura na sala onde fica o servidor, assim como a utilização de climatizadores para evitar o superaquecimento do servidor.
* Recomenda-se a utilização de uma solução de backups passiva ou ativa para assegurar a recuperação de dados que eventualmente posam ser perdidos ou corrompidos.
* Recomenda-se atualização de links redundantes de conexão com a internet e com operadoras diferenciadas para garantir a disponibilização do sistema 24 horas por dia (24x7).

**8. TESTES E IMPLEMENTAÇÃO**

Este capítulo aborda os itens que compõem a análise funcional do Sistema de Camisetas Personalizadas, assim como os requisitos específicos e os benefícios esperados com este sistema.

**8.1. TESTES DE QUALIDADE**

**8.1.1. REQUISITOS DE INTERFACE**

* *Vide apêndice M.*

**8.1.2. ROTEIROS DE TESTE**

* *Vide apêndice N.*

**8.1.3. EVIDÊNCIA DE TESTE**

* *Vide apêndice O.*

**8.2. TESTES DE STRESS**

* *Vide apêndice P.*

**8.3. TESTES DE SEGURANÇA**

* *Vide apêndice Q.*

**8.4. IMPLANTAÇÃO**

**8.4.1. PLANO DE IMPLANTAÇÃO**

De acordo com o diagrama a abaixo o plano de implantação será dividido em 5 (cinco) partes. A implantação do Sistema de Camisetas Personalizadas é separada em 3 (três) cenários diferenciados que são eles: clientes, profissionais e aplicação.

Os clientes devem contar com dois componentes básicos que são um navegador web com conectividade com a internet e uma impressora, a impressora pode ser descartada caso o cliente opte por efetuar o pagamento do boleto pelo internet.

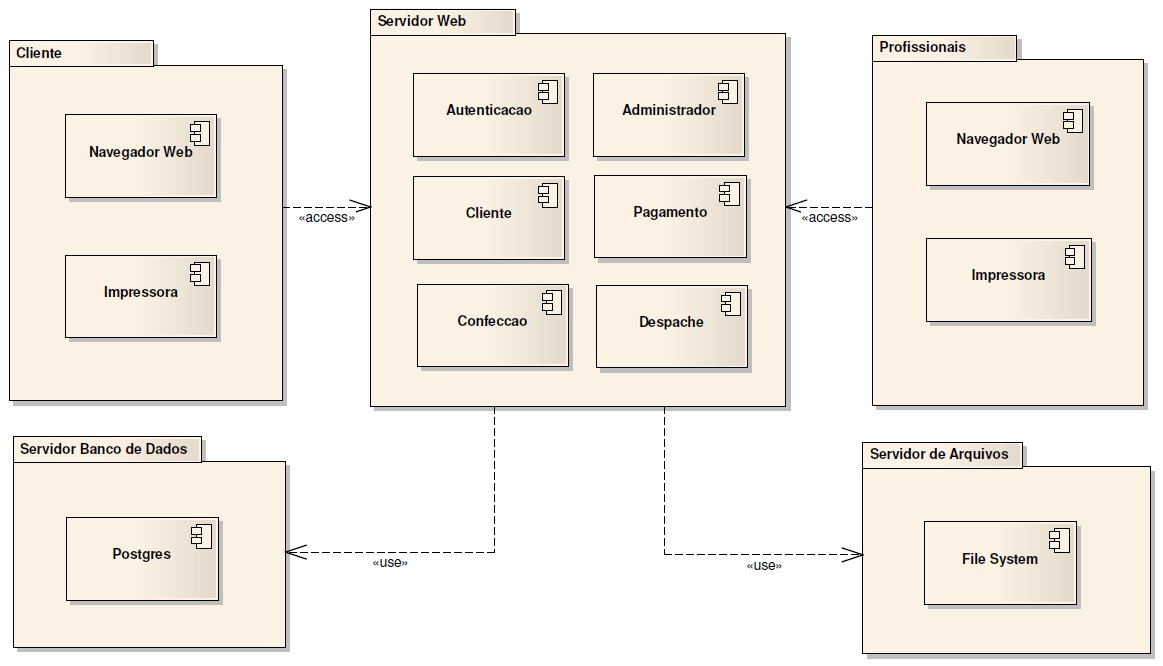
O browser (navegador web) será a interface do usuário com a aplicação, onde o mesmo terá acesso a todas as funcionalidades associadas ao seu perfil de ator (Cliente).

Os profissionais da empresa que irão usufruir do Sistema de Camisetas Personalizadas para a execução das tarefas relacionadas ao processo de fabricação de camisetas personalizas mapeadas no workflow do sistema, deve possuir no mínimo 5 (cinco) computadores desktop , 1 (uma) impressora jato de tinta ou laser para a impressão de relatórios e uma impressora de estamparia para a impressão das estampas das camisetas.

Quanto aos ambientes relacionados ao sistema a implantação deve seguir da seguinte forma:

* Um servidor deve comportar os arquivos básicos para o funcionamento da aplicação.
* Outro servidor deve ser o responsável por manter os arquivos de imagem das camisetas (estampas), a comunicação entre este servidor e o servidor de aplicação (servidor web) deve ser feita de maneira simbólica e transparente.
* Já o servidor de banco de dados recomenda-se que também seja disponibilizado em um maquina diferenciada.

Cumprindo com o proposto o esquema de implantação deve ser semelhante ao diagrama abaixo:



**8.4.2. MANUAL DE IMPLANTAÇÃO**

* *Vide apêndice R.*

1. **CONCLUSÃO**

A sdfasdfasdfasdfasdfasdfasdf asdf asdf asdf asdf asdf asdf asdf asdfa sdf.

# 10. REFERÊNCIAS BIBLIOGRÁFICAS

ACUNETIX. **Site Oficial do Acunetix Web Security Scanner**.Disponível em: <http:// www.acunetix.com/ > acessado em: 23 Out. 2011.

APACHE. **Site Oficial do Apache**.Disponível em: < http://www.apache.org/ > acessado em: 03 Fev. 2011.

JMETER. **Site Oficial do Apache JMeter**.Disponível em: < http:// jmeter.apache.org/ > acessado em: 21 Out. 2011.

BARNES, David. KOLLING, Michael. **Programação Orientada a Objetos com Java:** Introdução Prática usando o BLUEJ. São Paulo: Prentice Hall Brasil, 2007.

BETENCOURT, Paulo Ricardo B. **Análise Orientada a Objetos I**. Disponível em: < http://www.urisan.tche.br/~pbetencourt/engsoftII/analiseOO.pdf > acessado em: 19 Nov. 2010.

BETENCOURT, Paulo Ricardo B. **Análise Orientada a Objetos II**. Disponível em:<www.urisan.tche.br/~pbetencourt/engsoftII/**orienta**caoobjeto.doc */>* acessado em: 19 Nov. 2010.

BOOCH, Grady. RUMBAUGH, James. JACOBSON, Ivar. **UML**: Guia do usuário Rio de Janeiro: Elsevier, 2005.

DALL’OGLIO, Pablo. **PHP: Programando com Orientação a Objetos.** São Paulo: Novatec, 2007.

DAVID, Marcio Frayze. **Programação Orientada a Objetos: uma introdução**. Disponível em: < http://www.hardware.com.br/artigos/programacao-orientada-objetos > acessado em: 15 Nov. 2010.

DEEP LIQUID GROUP, deepliquid.com, Site Oficial do Jcrop. Disponível em: < http://deepliquid.com/content/Jcrop.html> acessado em: 19 Nov. 2010.

DOCTRINE. **Site Oficial do Framework Doctrine**: What's new in Doctrine 1.2.Disponível em: < http://www.doctrine-project.org/blog/what-s-new-in-doctrine-1-2

/ > acessado em: 25 Abr. 2011.

EA, Enterprise Architect. **Site Oficial do Enterprise Architect**.Disponível em: < http://www.sparxsystems.com.au/products/ea/index.html

/ > acessado em: 25 Abr. 2011.

EMILIANO, Pedro Henrique Mota. **Princípios da programação orientada a objetos**. Disponível em: < http://www.ph.eng.br/Programacao/principios-da-programacao-orientada-a-objetos.html /> acessado em: 18 Nov. 2010.

FOWLER, Martin. **UML Essencial**: Um breve guia para a linguagem padrão de modelagem de projetos. 3. edição. Porto Alegre: Bookman, 2005.

HORSTMANN, CAY. **Padrões e Projetos Orientados a Objetos.** 2. edição. São Paulo: Bookman, 2006.

HTMLSTAFF, htmlstaff.org. **Javascript**.Disponível em: <http://www. htmlstaff.org

> acessado em: 07 Mar. 2010.

INFO, Revista Exame. **Tabela de Salário dos profissionais de TI**. Disponível em: < http://info.abril.com.br/professional/salarios/*>* acessado em: 20 Out. 2011.

JACOBS, Ian. **O W3C desenvolve padrões e diretrizes para a Web.** Disponível em: <http://www.w3c.br/sobre> acessado em: 04 Set. 2010.

JHOHNYLAB. **Breve história do Javascript.** Disponível em: < http://johnylab.net/?id=75 > acessado em: 01 Jun. 2010.

LARMAN, Craig. **Utilizando UML e padrões**: Uma introdução à análise e aos projetos orientados a objetos e ao desenvolvimento iterativo. 3. Ed. Porto Alegre: Bookman, 2005.

LINDEN, Ricardo. KOLLING, Michael. **ALGORITMOS GENETICOS:** Uma importante ferramenta da inteligência computacional. Rio de Janeiro: Brasport, 2005.

LISBOA, Flávio Gomes da Silva. **ZEND FRAMEWORK:** Componentes poderosos para PHP. São Paulo: Novatec, 2009.

MILANI, André. **Construindo Aplicações Web Com PHP e MYSQL**. São Paulo: NovaTec, 2010.

NETBEANS. **Site Oficial do Netbeans**.Disponível em: <http://www.netbeans.org/ > acessado em: 23 Abr. 2010.

NETO, A. Pereira. **Postgresql: Técnicas avançadas: versões open source 7.x e 8.x:** soluções para desenvolvedores e administradores de bancos de dados. São Paulo: Érica, 2005.

OFICINADANET. **MVC:** O padrão de arquitetura de software.Disponível em: <http://www.oficinadanet.com.br/artigo/1687/mvc\_-\_o\_padrao\_de\_arquitetura\_de\_software

> acessado em: 15 Fev. 2011.

YOGODOSHI, Cayo Medeiros Aka. **Jcrop**: Recorte pedaços de imagens com ajuda do jQuery.Disponível em: <http://www.yogodoshi.com/blog/utilitarios/jcrop-recorte-pedacos-de-imagens-com-ajuda-do-jquery> acessado em: 19 Nov. 2010.

POSTGRESQL, Site Oficial do Postgresql. **History**. Disponível em: <http://www.postgresql.org/about/history/> acessado em: 30 Mai. 2010.

SYBASE. **Site Oficial do PowerDesigner**.Disponível em: < http://www.sybase.com.br/products/modelingdevelopment/powerdesigner

/ > acessado em: 23 Abr. 2011.

SILVA, Mauricio Samy. **Construindo sites com CSS e (X) HTML:** Sites controlados por folhas de estilo em cascata. São Paulo: Novatec, 2007.

SILVA, Mauricio Samy. **Tutorial:** CSS desde o inicio | Introdução.Disponível em: < http://www.maujor.com/tutorial/joe/cssjoe1.php> acessado em: 13 Mar. 2010.

ALENCAR, Walker. **Subversion & TortoiseSVN**.Disponível em: < http://blog.walkeralencar.com/archives/16> acessado em: 05 Nov. 2011.