

JANAISE FRANCISCO SICKA

**Analise do Esforço de Desenvolvimento de
Aplicações GUI e WEB.**

Vol. 1

Santa Catarina

2009

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

Janaise Francisco Sicka

**Analise do Esforço de Desenvolvimento de
Aplicações GUI e WEB.**

Trabalho de conclusão de curso submetido à Universidade Estadual de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação

Salvador Antonio dos Santos
Orientador

Joinville, junho 2009

Analise do Esforço de Desenvolvimento de Aplicações GUI e WEB.

Janaise Francisco Sicka

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação Integral do CCT/UEDESC.

Banca Examinadora

Salvador Antonio dos Santos, MSc.
Professor Orientador

Débora Cabral Nazário, MSc.

Edson Murakami, Dr

Joinville, 10 de junho de 2009.

RESUMO

Com a evolução da internet e da tecnologia *web* surge uma nova geração de sistemas de informação para negócios. Entre suas funções estão as de interação entre clientes e fornecedores. Muitas propostas de modelos e metodologias de desenvolvimento de sistemas para *web* surgiram. O objetivo dessas metodologias é ajudar o desenvolvedor a pensar de forma estruturada, mantendo a comunicação entre a equipe de desenvolvimento e apoio do sistema. Uma vez que a engenharia de *software* tradicional não absorve as tendências da *web* surge a engenharia da *web* e as metodologias Hiperemídia, com o objetivo de disciplinar o processo de desenvolvimento de sistemas *web*. O sistema de informação com interface gráfica com o usuário (GUI – *graphical user interface*) dá ao usuário uma aparência familiar com o sistema, que resulta no aumento do conforto do usuário na utilização do sistema, e manuseio mais produtivo, além de reduzir o tempo de aprendizagem. Porém, com a tecnologia *web*, o sistema de informação tradicional (*desktop*) tende a migrar sua estrutura para *web*. Este trabalho tem o objetivo de comparar os esforços de implementação entre estruturas GUI e *Web*, através da medição de esforço hora-homem durante o desenvolvimento do sistema. Desta forma será possível dar as empresas um parâmetro de consumo de recurso no desenvolvimento de sistemas de tecnologia *web* consome. Isso ajudará nas tomadas de decisões sobre o investimento da evolução de seus sistemas para *web*. Para chegar a esse resultado foi desenvolvido o sistema Questionário, em estrutura GUI e com estrutura *Web*. O sistema possui apenas a função de inclusão de dados no banco de dados, ou seja, um sistema simples. As medições de hora-homem foram realizadas durante as fases de implementação do sistema, e foram obtidos os resultados: 25:56 horas de implementação para estrutura GUI e 31:24 hora para estrutura *web*.

Palavras-chave: sistemas de informação, metodologia de desenvolvimento *web*, internet, tecnologia *web*.

ABSTRACT

With the development of internet and web technology is a new generation of information systems for business. Among its functions are those of interaction between customers and suppliers. Many proposals of models and methodologies for developing systems for web emerged. The purpose of these methodologies is to help the developer to think in a structured way, maintaining communication between the team and support the development of the system. Since no traditional software engineering trends absorbs web comes to engineering Web Hypermedia and methodologies with the objective of the disciplinary process of developing web systems. The information system with graphical user interface (GUI - graphical user interface) gives users a familiar look to the system, which results in increasing the comfort of the user in using the system, and handling more productive, and reduce the time of learning. However, with web technology, the traditional information system (desktop) tends to migrate to the web structure. This work aims to compare the efforts to implement structures between GUI and Web, by measuring stress-hours / man during the development of the system. This way you can give companies a measure of resource consumption of the development of web technology consumes. This will help in decision making on investment of changes in their systems to web. To reach this result the system was developed questionnaire in structure to structure Web GUI and the system has only the function of including data in the database, or a simple system. Measurements of time / man were made during the implementation of the system, and the results were obtained: 25:56 hours deployment structure for GUI and 31:24 hours for web structure.

Key words: information systems, methods of development, internet, web technology.

LISTA DE FIGURAS

Figura 1: Modelo em Cascata de processo da Web Engineering (LEITE, 2002 apud CECHELERO, D.; VOLPI, M. M , 2004).....	13
Figura 2: Modelo Espiral de processo da <i>Web Engineering</i> (PRESSMAN, 2002).	14
Figura 3:Modelo Evolucionário para o processo de Engenharia da <i>Web</i> . Baseado em (MURUGESAN; GINIGE, 2005).	17
Figura 4: Processo <i>Extreme Programming</i> . Baseado em (PRESSMAN, 2006).	22
Figura 5: primitivas do RMDM (COSTAGLIOLA, G.; FERRUCCI, F.; FRANCESE, 2002)	25
Figura 6: Diagrama do RMDM, baseado em Costagliola; Ferrucci e Francese (2002) ..	26
Figura 7: Parte de modelo E-R do Projeto Envolve-se. Baseado em (CERI, S.; FRATERNALI, P.; PARABOSCHI, S, 1999).....	29
Figura 8: Sistema de Informação. Adaptado de (O'BRIEN, 2002).	32
Figura 9: Funções de um sistema de informação (LAUDON; LAUDON, 2004).....	33
Figura 10: Estrutura Cliente Servidor adaptado (LAUDON; LAUDON, 2004).	40
Figura 11: Estrutura P2P.....	40
Figura 12: Processo de Contagem de pontos de Função adaptado de (VAZQUEZ; SIMÕES; ALBERT, 2003).	56
Figura 13 – Comparação de esforço hora-homem.....	65

LISTA DE TABELAS

Tabela 1: Tipos de manutenção (GINIGE, 2002 apud KAPPEL, 2006)	18
Tabela 2: Tabela de Características da Metodologia OOHDM, (ROSSI, 1996).	28
Tabela 3: Proposta de relacionamento entre Fatores versus Processos de Desenvolvimento <i>Web</i>	45
Tabela 4: Proposta de relacionamento entre Fatores versus Sistemas de Informação <i>Web</i>	46
Tabela 5: Igualdade de requisitos do relacionamento entre Processos e Fatores.	47
Tabela 6: Igualdade de requisitos do relacionamento entre Sistema de Informação e Fatores.	48
Tabela 7: Proposta de relacionamento entre Sistemas de Informação <i>Web</i> versus Processos de Desenvolvimento <i>Web</i>	49
Tabela 8: Tabela de Complexidade do Tipo de Dado (ROCHA, 2007)	58
Tabela 9: Contagem de funções de tipo de dados	59
Tabela 10: Tabela de Complexidade para EE (ROCHA, 2007)	59
Tabela 11: Tabela de Complexidade para SE e CE (ROCHA, 2007)	59
Tabela 12: Contagem de funções de tipo de dados e transação	60
Tabela 13: Tabela de Conversão de Complexidade x UFP. Baseado em (ROCHA, 2007)	60
Tabela 14: Dados para contagem de função não ajustadas	61
Tabela 15: Estimativa de horas de trabalho	63

LISTA DE ABREVIATURAS

AIE	Arquivo de Interface Externa
ALI	Arquivo Lógico Interno
AR	Arquivos Referenciados
B2B	<i>Business to Business</i>
B2C	Business to Commerce
BDES	Grupo de Banco de Dados e Engenharia de Software
CCT	Centro de Ciências Tecnológicas
CD-R	<i>Compact Disk Recordable</i>
CD-ROM	<i>Compact Disk Read-Only Memory</i>
CD-RW	<i>CD-ReWritable</i>
CE	Consulta Externa
CMM	<i>Capability Maturity Level</i>
CNC	<i>Computer Numeric Control</i>
CPD	Centro de Processamento de Dados
EE	Entrada Externa
E-R	Entidade e Relacionamento
ERP	<i>Enterprise Resource Planning</i>
GUI	<i>Graphical user interface</i>
H/PF	Hora por Ponto de Função
HTML	<i>Hiper Text Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
JSF	<i>Java Server Faces</i>
I/O	<i>Input/Output</i>
MS-DOS	<i>Microsoft Disk Operating System</i>
O.O.	Orientado a Objeto
OOHDM	<i>Object-Oriented Hypermedia Design Model</i>
P2P	<i>peer-to-peer</i>
PC's	Computadores Pessoais
RMM	<i>Relationship Management Methodology</i>
RMDM	<i>Relationship Management Data Methodology</i>
RUP	<i>Rational Unified Process</i>

SE	Saída Externa
SPICE	<i>Software Improvement and Capability dEtermination</i>
TCC-II	Trabalho de Conclusão de Curso - II
TCP/IP	<i>Transmission Contrl Protocol/Internet Protocol</i>
TD	Tipo de dados
TGI	Yotal de grau de influencia
UDESC	Universidade do Estado de Santa Catarina
URL	<i>Uniform Resource Locator</i>
UML	Linguagem de Modelagem Unificada
VFA	Valor do Fator de Ajuste
XP	<i>Extreme Programming</i>
WebMI	<i>Web Modeling Language</i>
WIMP	<i>Windows, Icons, Menus and Pointers</i>
www	<i>World Wide Web</i>

SUMÁRIO

1. INTRODUÇÃO	6
1.1 OBJETIVOS	9
1.1.1 OBJETIVO GERAL	9
1.1.2 OBJETIVOS ESPECÍFICOS	9
1.2 ESTRUTURA DO TRABALHO	9
2. PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE PARA WEB	11
2.1 PROCESSOS DE ENGENHARIA DA WEB	11
2.1.1 MODELO EM CASCATA	12
2.1.2 MODELO EM ESPIRAL	13
2.1.3 MODELO EVOLUCIONÁRIO	17
2.2 PROCESSOS ÁGEIS	20
2.2.1 EXTREME PROGRAMMING (XP)	22
2.2.2 SCRUM	23
2.3 MODELAGEM HIPERMÍDIA	24
2.3.1 RMM (RELATIONSHIP MANAGEMENT METHODOLOGY)	24
2.3.2 OOHDM (OBJECT-ORIENTED HYPERMEDIA DESIGN MODEL)	27
2.3.3 WEBML (WEB MODELING LANGUAGE)	28
2.3.3.1 O MODELO DE ESTRUTURA	29
2.3.3.2 O MODELO DE HIPERTEXTO	29
2.3.3.3 O MODELO DE APRESENTAÇÃO	30
2.3.3.4 O MODELO DE PERSONALIZAÇÃO	30
3. SISTEMAS DE INFORMAÇÃO E SUA TAXONOMIA	31
3.1 DEFINIÇÃO DOS SISTEMAS DE INFORMAÇÃO	31
3.2 SISTEMAS DE INFORMAÇÃO CIENTÍFICOS	33
3.3 SISTEMAS DE INFORMAÇÃO COMERCIAL	34
3.4 FERRAMENTAS	35
3.5 SISTEMAS DE INFORMAÇÃO EMBARCADOS	35
3.6 SISTEMAS DE INFORMAÇÃO WEB	36
4. SISTEMAS DE INFORMAÇÃO E SEUS PROCESSOS DE DESENVOLVIMENTO	38
4.1 FATORES DE UM SISTEMA DE INFORMAÇÃO	38
4.1.2 HARDWARE	38
4.1.2.1 MEIO DE ARMAZENAMENTO	38
4.1.2.2 LOCAL DE PROCESSAMENTO	39

4.1.3 SOFTWARE	40
4.1.3.1 SOFTWARE BÁSICO	41
4.1.4 REQUISITOS	42
4.1.4.1 TEMPO DE RESPOSTA	42
4.1.4.2 REQUISITO VOLATILIDADE	42
4.1.5 PARADIGMA DE MODELAGEM DE SOFTWARE	42
4.1.6 ERGONOMIA	43
4.1.7 USUÁRIO	43
4.2 PROCESSOS VERSUS FATORES	44
4.3 SISTEMAS DE INFORMAÇÃO VERSUS FATORES	45
4.4 ADEQUAÇÃO DOS PROCESSOS AOS SISTEMAS	47
5. TRABALHOS CORRELATOS	50
5.1 MIGRAÇÃO DA PLATAFORMA COBOL PARA WEB	50
5.2 APLICAÇÃO DE MÉTRICAS DE PONTO DE FUNÇÃO EM SISTEMAS WEB	51
5.3 MIGRAÇÃO DE APLICAÇÕES WIMP PARA AMBIENTE WEB	53
6. ESTUDO DE CASO	55
6.1 PROJETO DE EXTENSÃO “ENVOLVA-SE”	55
6.2 PONTOS DE FUNÇÃO	56
6.2.1 TIPO DE CONTAGEM	57
6.2.2 ESCOPO DA CONTAGEM E FRONTEIRA DA APLICAÇÃO	57
6.2.3 FUNÇÃO DO TIPO DE DADO	58
6.2.4 FUNÇÃO DO TIPO TRANSAÇÃO	59
6.2.5 CONTAGEM DOS PONTOS DE FUNÇÃO NÃO-AJUSTADOS (UFP)	60
6.2.6 VALOR DO FATOR DE AJUSTE	61
6.2.7 CALCULO DO NÚMERO DOS PONTOS DE FUNÇÃO AJUSTADO	62
6.3 RESULTADO DO PONTO DE FUNÇÃO	62
6.4 SISTEMAS DE INFORMAÇÃO DE INTERFACE GRÁFICA COM O USUÁRIO (GUI)	63
6.5 SISTEMA DE INFORMAÇÃO WEB	64
6.6 RESULTADOS	65
7. CONCLUSÃO	66
REFERÊNCIAS	68
APÊNDICES	73
APÊNDICE A – PLANO DE TRABALHO DE CONCLUSÃO DE CURSO	73
APÊNDICE B – LEVANTAMENTO DE REQUISITOS DO SISTEMA QUESTIONÁRIO DO PROJETO DE EXTENSÃO ENVOLVA-SE	80

APÊNDICE C – QUESTIONÁRIO IMPRESSO DO SISTEMA QUESTIONÁRIO DO PROJETO DE EXTENSÃO ENVOLVA-SE	87
APÊNDICE D – DOCUMENTAÇÃO DO PROCESSO ESPIRAL PARA SISTEMA COM INTERFACE GUI	89
APÊNDICE E – DOCUMENTAÇÃO DA METODOLOGIA OOHDM PARA SISTEMA WEB	91

1. Introdução

A Internet e a tecnologia *Web* foram desenvolvidas com o objetivo de divulgar trabalhos científicos, aos poucos elas evoluíram, deixando de apresentar interfaces pouco sofisticadas e de difícil manuseio, e assim modificaram a sua finalidade inicial. Dentre as mudanças sofridas pela tecnologia *Web*, uma grande evolução se deu quando os usuários tiveram a possibilidade de interagir com a *web*. Deixou de funcionar como um simples repositório de conteúdos estáticos, e passou a funcionar como uma interface de diversos sistemas de informações dinâmicos. Também aumentou a interatividade, semelhante ao de uma aplicação de *desktop*, com mais velocidade e agilidade no manuseio das informações (GABRIELI et al, 2007) (SCHMITT; SCHMITT, 2007) (ZANETI JR, 2003).

Surge então, uma nova geração de negócios, com atividades que envolvem a interação entre clientes e fornecedores, sendo possível aproveitar os benefícios dessa tecnologia. Essa evolução da tecnologia da informação e da comunicação tornou-se um fator muito importante na atual gestão de empresas, atingindo suas diferentes áreas. Nesse cenário pode ser citado o comércio eletrônico, principalmente o negócio-para-consumidor (*B2C*) e o negócio-para-negócio (*B2B*) (LAUDON; LAUDON, 2000; O'BRIEN, 2003 apud GABRIELI et al, 2007), (ZANETI JR, 2003 apud SCHMITT, SCHMITT; 2007).

Apesar do rápido desenvolvimento da tecnologia *web*, pouca coisa mudou em relação ao desenvolvimento de suas aplicações. Muitas organizações têm sucesso, outras apresentam falhas ou tendências a elas, e fracassam antes de alcançar suas primeiras metas. A principal razão destes problemas está no desenvolvimento de aplicações *web* que continuam improvisadas, sem suporte de metodologias ou processos disciplinados. Isso proporciona o surgimento de grande quantidade de sistemas mal desenvolvidos e que podem gerar muitas falhas, colocando em risco a confiança na internet. Para a construção de sistemas, e aplicações na *web* é preciso que os desenvolvedores adotem abordagens disciplinadas de Engenharia da *Web* (COSTAGLIOLA; FERRUCCI; FRANCESE, 2002) (MURUGESAN;GINIGE, 2005) (PRESSMAN, 2002).

Muitas propostas de modelos e metodologias de engenharia *Web*/Hipermissão foram desenvolvidas. Elas têm por objetivo ajudar o desenvolvedor a pensar de

forma estruturada, com o foco em aspectos como *links* de estrutura e navegação, além de manter a comunicação entre programadores, analistas, administrador do *software* e usuário (COSTAGLIOLA; FERRUCCI; FRANCESE, 2002).

Gabrielli et al (2007) diz que os esforços para possibilitar que os sistemas de informação usufruam os benefícios da internet são grandes. Pois os sistemas de informação baseados na *web* podem ser acessados através de um navegador por várias pessoas simultaneamente, geograficamente distantes, diferentemente dos sistemas tradicionais, que para serem acessados necessitam de uma estação em que este esteja instalado. Zaneti Jr (2003), diz que um sistema de informação baseado na *web*, pode trocar informações com qualquer outro sistema disponível na *web*, independente da organização que desenvolve ou processa os dados. Isso que torna a tecnologia *Web* uma infra-estrutura de comunicação dos sistemas entre si, e comunicação de usuários com sistemas.

Uma das definições mais conhecida de sistemas de informação é de Laudon e Laudon (2004). Eles dizem que um sistema de informação é um conjunto de elementos relacionados à coleta, processo, armazenamento e distribuição de informação, com o objetivo de ajudar no processo de organização e tomada de decisão. Existem três atividades em um sistema de informação que produzem as informações necessárias para gerência das organizações, elas são: entrada, processamento e saída.

Autores como Pressmann (1999, apud ZANETI JR, 2003) apresentam três gerações de sistemas de informação, a primeira possui o processamento *off-line* e através de cartões perfurados. Os autores Sordi e Marinho (2007) também definem com primeira geração os sistemas *batch*. A Segunda geração é marcada pelo processamento *on-line* que surgiu para suprir as deficiências da geração anterior, seja dos cartões ou *batch*. E a terceira geração é denominada como cliente-servidor, surgiu a partir da grande troca de informações entre os sistemas integradas ao banco de dados.

O objetivo deste trabalho está na caracterização dos sistemas baseados na *web*, nas suas metodologias de desenvolvimento e na implementação de um sistema com estrutura GUI (*Graphical User Interface*) e *Web*, para comparar os esforços de implementação. Para isso, foram estudados processos de desenvolvimento, elaborada uma taxonomia de aplicações *web*, e correlacionados aos processos através de fatores dos sistemas de informações e processos de

desenvolvimento. Após esse estudo foi desenvolvida a aplicação “Questionário” do Projeto de Extensão ENVOLVA-SE, em duas versões para comparação dos esforços. Uma com estrutura GUI com processo mais adequado às características do sistema de informação desktop, implementado em Java através dos componentes Swing. E outra versão do mesmo sistema em estrutura Web, também com um processo adaptado conforme suas características, e desenvolvido também em Java para web com JSF (Java Server Faces). Foram feitas medidas de ponto de função e esforço (hora-homem) em cada caso. Com essas medidas foi possível comparar o gasto na migração de sistemas de informação com estrutura GUI para Web.

Tais migrações na verdade sempre consistiram em re-engenharia, ou seja, praticamente todo o software é normalmente re-escrito. Isto tem um gasto de consumo. As empresas que desenvolvem software têm uma historia de consumo de recurso para o desenvolvimento de sistemas em GUI. Não possuem histórico de desenvolvimento de sistemas web. A contribuição deste trabalho, dará as empresas um parametro de consumo de recurso que a tecnologia web consome, comparando a tecnologia GUI. Isso facilitará as empresas nas suas decisões sobre o investimento na evolução de seus sistemas para web.

O projeto de Extensão ENVOLVA-SE é coordenado pela Professora Jurema Iara Reis Belli, do Departamento de Ciências Básicas Sociais. O Questionário é utilizado para coletar informações socioeconômicas e culturais dos acadêmicos da UDESC-Joinville, tendo por objetivo traçar um perfil do campus, e assim possibilitar o oferecimento de atividades extra-curriculares. Estrategicamente, os alunos que respondem a esse questionário são os bolsistas cadastrados no SOE (Setor de Orientação ao Estudante) e os calouros. Atualmente esse questionário é disponibilizado em formato impresso e a coleta das informações é feita manualmente em planilhas de Excel, o que justifica a restrição de acesso dos acadêmicos a esse documento e dificulta a análise dos dados. Com um sistema *on-line*, é possível atingir grande parte da comunidade acadêmica, gerando análises mais precisas e atividades de interesse aos acadêmicos.

1.1 Objetivos

1.1.1 Objetivo Geral

Fazer uma análise comparando esforços de implementação de aplicações de estrutura GUI e Web. A diferença do esforço entre o desenvolvimento GUI e Web, ajudará os desenvolvedores de software a planejarem seus investimentos, uma vez que atuais aplicações GUI sofrem pressão para usarem tecnologia Web.

1.1.2 Objetivos Específicos

- 1) Modelagem Orientada a Objeto para aplicação GUI e *Web*;
- 2) Processo de Desenvolvimento GUI e *Web*;
- 3) Tecnologia de Implementação GUI e Web

1.2 Estrutura do Trabalho

Este trabalho está estruturado em seis capítulos, da seguinte forma:

No capítulo um é apresentada a introdução os objetivos gerais e específicos deste trabalho.

No capítulo dois, é descrito o estudo dos processos de Engenharia de *Software* para *web*, processos ágeis e das metodologias de Hipermídia para o desenvolvimento de sistemas *web*.

No capítulo três, é apresentado o estudo das características que identificam os sistemas de informação até a chegada da *web*, também o desenvolvimento de uma taxonomia desses sistemas.

No capítulo quatro, é realizado o estudo de alguns fatores relevantes para os sistemas de informação, também é elaborado um relacionamento entre os processos e metodologias estudadas no capítulo 2 e a taxonomia dos sistemas de informação apresentada no capítulo 3.

No capítulo cinco, é apresentado os trabalhos relacionados a este.

No capítulo seis é apresentado o sistema Questionário e os resultados das implementações do sistema em estrutura GUI e Web.

No Capítulo sete, é apresentada a conclusão do trabalho, indicando os resultados obtidos, também as sugestões de novas propostas de estudos posteriores a este trabalho.

2. Processos de Desenvolvimento de *Software* para *Web*

O objetivo deste capítulo é apresentar as diferentes formas de desenvolvimento dos *softwares* para *web*. Ele é dividido em três partes, esta divisão se dá pela literatura encontrada. A primeira parte é a Engenharia de *Software* para *Web*, é uma adaptação dos processos de Engenharia de *Software* tradicionais, representados pelos processos Cascata, Espiral e Evolucionário.

A segunda é o Processo Ágil é um manifesto a favor de um desenvolvimento de *software* dinâmico e rápido, representados pelos processos *Extreme Programming* (XP) e *Scrum*. O terceiro, por sua vez é a Modelagem Hipermédia, que são formas estruturadas de desenvolver um sistema multimídia, seja ele um *site* ou não, representados pelas metodologias OOHDM (*Object-Oriented Hypermedia Design Model*), RMM (*Relationship Management Methodology*), e WebML (*Web Modeling Language*).

Com esse estudo, será possível, no capítulo 4, elaborar o relacionamento entre os sistemas de informação e seus processos e métodos de desenvolvimento.

2.1 Processos de Engenharia da *Web*

A engenharia de *software* tradicional tornou-se insuficiente para o desenvolvimento de sistemas e aplicações *web*, por não se preocupar com aspectos importantes para tais sistemas. Dentre eles podemos citar aspectos de usabilidade, interface e também de estrutura de fácil percepção aos usuários. É reconhecida então, a necessidade de desenvolver uma engenharia de *web* que utiliza metodologias de Hipermédia e juntas, propor uma aproximação de um desenvolvimento sistemático e disciplinado (COSTAGLIOLA; FERRUCCI; FRANCESE, 2002).

Com o objetivo de desenvolver sistemas e aplicações *web* de alta qualidade, minimizando os riscos de falhas e melhorando a manutenção, surge a *Web Engineering* (Engenharia da *Web*), o que possibilita o sucesso no desenvolvimento de diversificadas e complexas aplicações (MURUGESAN ; GINIGE, 2005).

Um processo de desenvolvimento de uma aplicação *web*, deve estabelecer etapas de atividades de desenvolvimento. Estas, por sua vez devem estar dispostas

de forma clara, para que os desenvolvedores possam segui-las disciplinarmente. Também se faz necessário um cronograma de atividades, para ter um tempo estimado de desenvolvimento entre as fases de concepção e de implementação (GINIGE, A.; MURUGESAN, S., 2001).

A seguir serão apresentados os modelos de processos de engenharia da *web*, que mais sofreram alterações dos modelos tradicionais e foram desenvolvidos para atender as necessidades desses sistemas. Os modelos em Cascatas e o Espiral são adaptações da engenharia de *software* tradicional (CECHELERO, D.; VOLPI, M. M, 2004). Já o modelo Evolucionário foi baseado nas experiências que Murugesan e Ginigi (2005) tiveram na construção de aplicações e sistemas *web*.

2.1.1 Modelo em Cascata

Na figura 1 é apresentado o processo em Cascata. Suas atividades de desenvolvimento são estruturadas em forma de cascata de etapas. Como o próprio nome sugere, a saída de uma atividade é à entrada da próxima. Uma atividade só pode começar a partir do momento que a anterior seja finalizada. Também descreve a separação das etapas de definição, análise e design da implementação, que é a principal etapa no desenvolvimento *web* (CECHELERO, D.; VOLPI, M. M , 2004).

As fases do modelo em Cascata são adaptadas da engenharia de *software* tradicional, também possuem novas atividades como a primeira atividade, definição do problema e operação. A segunda atividade diz respeito a análise e especificação dos requisitos que como na engenharia tradicional tem o objetivo de identificar os requisitos do sistema, através de questionamento aos usuários. A terceira também é uma nova atividade que foca ao *design* do site, a quarta por sua vez, diz respeito a implementação e testes, que tem por objetivo de verificar se o site atende a necessidade do cliente. A quinta fase, possui um teste completo do sistema para garantir que os requisitos foram implementados com sucesso. A ultima fase diz respeito a toda manutenção que o sistema pode ter durante sua vida útil (SOMMERVILLE, 2003).

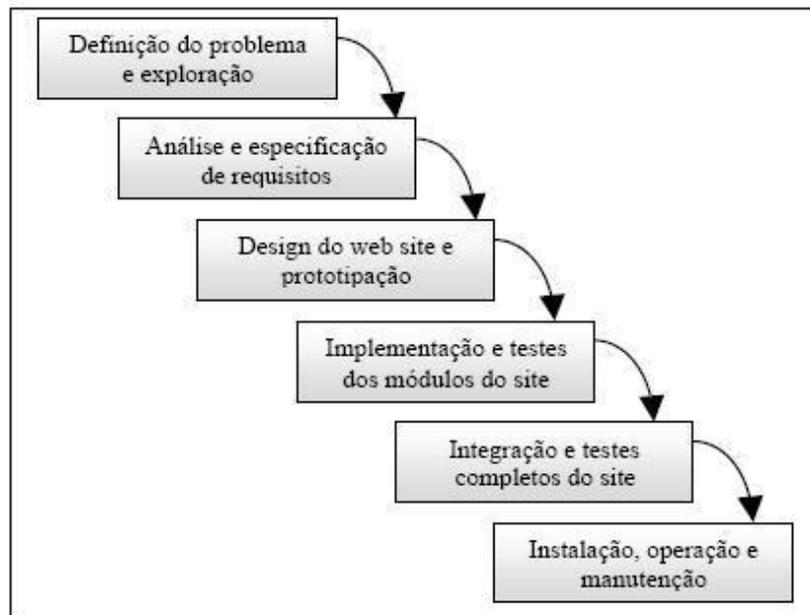


Figura 1: Modelo em Cascata de processo da Web Engineering (LEITE, 2002 apud CECHELERO, D.; VOLPI, M. M , 2004).

O modelo em cascata é dito como um modelo crítico, pelo fato de sua linearidade. (LEITE, 2002 apud CECHELERO, D.; VOLPI, M. M , 2004). Na Engenharia de software tradicional, Sommerville (2003) diz que o maior problema desse modelo, é a sua inflexibilidade na divisão de projetos em etapas distintas, ou seja, não é possível atender a requisitos que sofrem constantemente evolução.

2.1.2 Modelo em Espiral

O imediatismo e a grande evolução das aplicações *web* pedem um modelo interativo e incremental, como o espiral, pois eles produzem várias versões de sistemas baseados na *web* em um curto espaço de tempo. Os primeiros protótipos podem ser modelos manuscritos e durante as ultimas iterações, as versões estarão cada vez mais completas (PRESSMAN, 2002).

Para aspectos de distribuição como internet, intranet ou extranet, se faz necessário dar importância a questões como a diversidade de usuários, e a arquitetura da aplicação. Sistemas e aplicações *web* geralmente são guiadas pelo conteúdo juntamente com a estética, então é provável que a utilização de atividades paralelas seja incorporada dentro de um processo de engenharia para *web*, podendo

ainda envolver equipes de desenvolvimento técnicos e não técnicos (PRESSMAN, 2002).

Pressman (2002) se preocupa com a migração dos sistemas da *web* estáticas para as dinâmicas, ressaltando a importância de ser aplicado princípios rígidos de gestão e de engenharia, e sugere um modelo espiral de processo para *Web Engineering* apresentado pela figura 2.

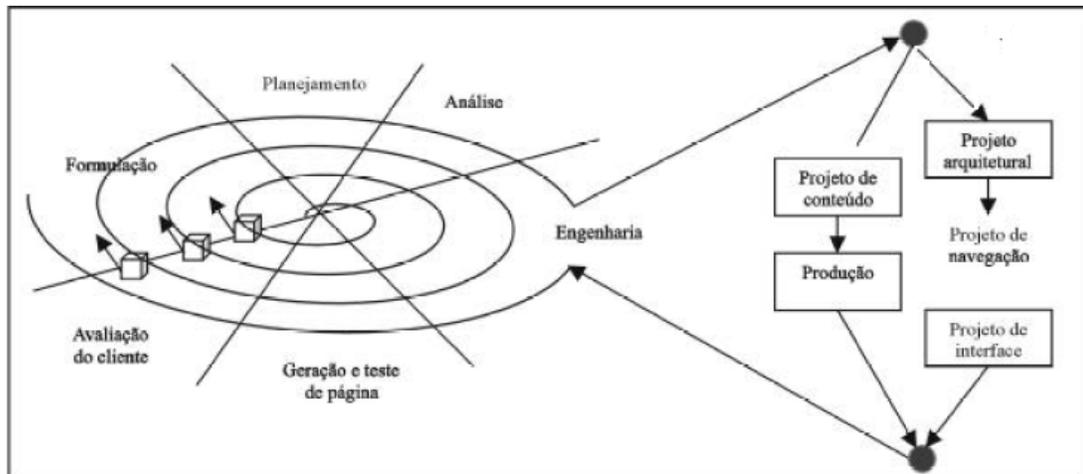


Figura 2: Modelo Espiral de processo da *Web Engineering* (PRESSMAN, 2002).

O modelo espiral sugerido por Pressman (2002) apresenta as seguintes atividades que serão descritas a seguir: formulação, planejamento, análise, engenharia, geração e teste de páginas e avaliação do cliente.

- **Formulação:** Essa é a primeira atividade do processo, nela são realizadas atividades de identificação de metas e objetivos da aplicação baseada em *web*, juntamente com os clientes, e também é formulado o primeiro escopo.
- **Planejamento:** Suas tarefas estão relacionadas ao levantamento do custo estimado para a construção do projeto, é feita uma avaliação dos riscos e define dois tipos de cronogramas, o primeiro de granularidade fina para o incremento inicial e o outro de granularidade grossa para os subseqüentes.
- **Análise:** São estabelecidos requisitos técnicos e gráficos (interface), também especifica itens de conteúdo.

- **Engenharia:** Nessa etapa as atividades estão relacionadas aos projetos: de estrutura, de navegação e de interface. Porém possui duas tarefas paralelas: projeto de conteúdo e produção, elas tem por objetivo projetar, produzir e quando necessário adquirir conteúdos gráficos, de texto, áudio e outras mídias que podem fazer parte da aplicação.
- **Geração e testes das páginas:** São atividades que necessitam de ferramentas para a criação das aplicações ou sistemas, é nesse momento que é feita a integração entre as arquiteturas definidas fase de engenharia para a produção das páginas. O teste exercita a navegação do sistema e verifica se a aplicação possui algum tipo de erro, ajudando a garantir a qualidade operacional do produto.
- **Avaliação do cliente:** É feita a cada incremento do processo, é nesse momento que são solicitadas as mudanças, que serão feitas nos próximos incrementos.

As etapas de formulação e de análise representam uma seqüência de atividades. A primeira dá início a identificação de um conjunto de metas e a objetivos comuns entre clientes e desenvolvedores. Já a análise como é uma atividade técnica, finaliza essa seqüência identificando os requisitos de dados, funcionalidades e comportamentos de uma aplicação *web* (PRESSMAN, 2002).

Na fase de engenharia são incorporadas etapas que dizem respeito à construção de projetos. Essas etapas foram destacadas, pois existem uma grande cobrança do imediatismo e da constante evolução que força o engenheiro de *web* estabelecer projetos que solucione de forma rápida o problema do negócio. A grande dificuldade dessa etapa está em garantir que a resolução proposta pelo engenheiro da *web*, não comprometa a capacidade de evolução do sistema ao longo do tempo (PRESSMAN, 2002).

Ao iniciar o processo a equipe de desenvolvimento movimenta a espiral. O primeiro circuito pode resultar na especificação do produto, e as subseqüentes são utilizadas para desenvolver protótipos e assim progressivamente versões mais completas (PRESSMAN, 2002).

As etapas de definição de projetos são:

- projetos arquiteturais: possui três modelos a serem definidos: o linear, o hierárquico e o em rede;
- projeto de navegação: é a definição dos caminhos que usuário poderá ter acesso ao conteúdo e as serviços oferecidos pela aplicação *web*; e
- projeto de interface: uma vez que é com a interface que o usuário terá seu primeiro contato, por esse motivo é importante lembrar que uma interface bem projetada melhora a percepção do usuário e aos serviços oferecidos, não é necessário ser algo extraordinário, porém deve ser bem estruturado e ergonomicamente sólido (PRESSMAN, 2002).

A filosofia adotada por Pressman (2002) para a fase de teste, é de exercitar o sistema a fim de encontrar erros. Para engenheiros da *web* encontrar erros é um grande desafio, visto que aplicações *web* residem em redes e distribuídas em diversos tipos de sistemas operacionais, *browsers*, plataformas de *hardware* e protocolos de comunicação. Porém para aplicações *web* é adotada uma abordagem de princípios básicos de teste, aplicando uma estratégia táticas que são recomendadas para sistemas orientados a objetos, que são as seguintes:

- o modelo de conteúdo é revisado com o intuito de descobrir erros;
- o modelo de projeto é revisado para descobrir erros navegacionais;
- componentes de processamentos pré-selecionados são dispostos a testes de unidade;
- após a construção da arquitetura é feito teste de integração;
- assim que as aplicações *web* são montadas é feito um teste em relação a funcionalidade e a entrada de conteúdo;
- é feito teste de compatibilidade para várias configurações de ambientes;
- é elaborado um teste para usuários que são monitorados.

Pressman (2002) diz que as principais vantagens desse modelo é o fato de ser ótimo para curtos prazos de desenvolvimento e também é bastante fácil de lidar com requisitos que possam sofrer alterações ou que ainda não existem.

2.1.3 Modelo Evolucionário

Baseado nas experiências que tiveram na construção de aplicações e sistemas *web*, Murugesan e Ginige (2005) recomendam um processo evolucionário para a engenharia da *web*, como mostrado na figura 3.

Os passos do modelo evolucionário do processo de engenharia da *web* são descritos a seguir (GINIGE, 2002 apud KAPPEL, 2006), (MURUGESAN; GINIGE, 2005 apud KAPPEL, 2006) e (MURUGESAN et al., 1999 apud KAPPEL, 2006):

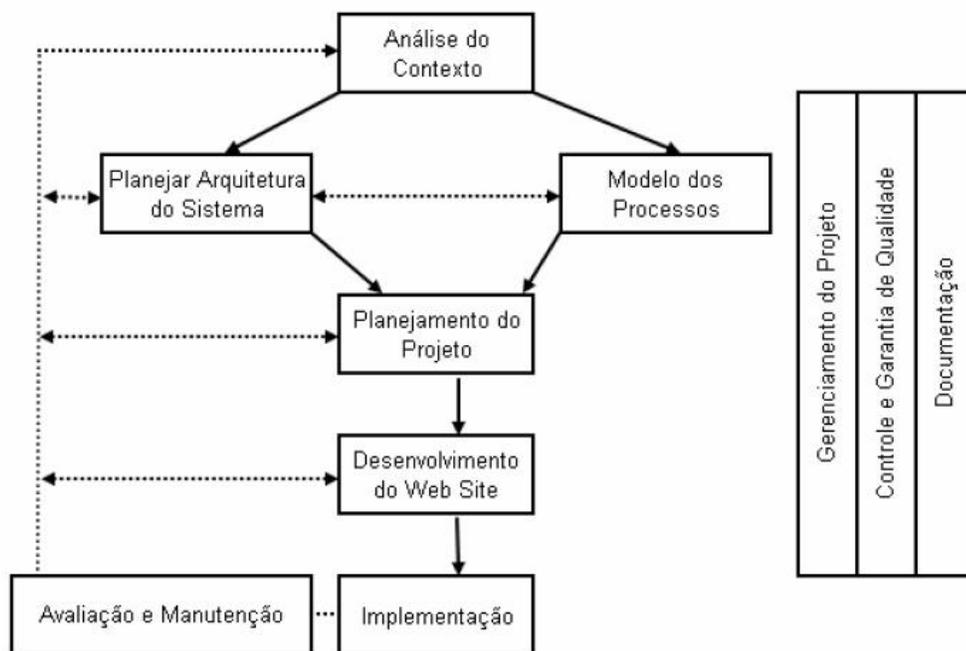


Figura 3: Modelo Evolucionário para o processo de Engenharia da *Web*. Baseado em (MURUGESAN; GINIGE, 2005).

- **Análise do Contexto:** A análise de contexto tem por objetivo identificar os principais usuários e suas necessidades, determinar as funcionalidades que serão projetadas, determinar as informações do sistemas, identificar requisitos não técnico como: *layout* e desempenho, e fazer uma estimativa da demanda do público alvo.
- **Planejar Arquitetura do Sistema:** Nessa etapa são utilizadas as informações obtidas na fase anterior para desenvolver as arquiteturas: física do sistema que são as ligações entre servidores, banco de dados, e outras necessárias; do aplicativo que determina um mapa de

módulos e funções disponíveis; e da aplicação que identifica os módulos de programas e a base de dados definidas pela arquitetura de aplicativos.

- **Modelo dos Processos:** As atividades dessa etapa consistem em uma análise detalhada dos requisitos funcionais, *layout*, testes e implementação, também se faz necessário obter o endereçamento dos requisitos não técnicos que foram determinados pela análise de projetos e dessa forma é especificado um grupo de subprojeto ou subprocesso que deverão ser desenvolvidos e implementados.
- **Planejamento do Projeto:** Trata-se da determinação de um cronograma de atividades a serem desenvolvidas, prazos e recursos necessários, baseados no planejamento dos subprojetos.
- **Desenvolvimento do Web Site:** Nesta etapa é feita a criação da interface que determinará quais informações serão apresentadas e a forma de apresentação, também é construído o *Web Site*.
- **Implementação:** Consiste em disponibilizar para a utilização *online* e também a permissão de acesso as informações necessárias aos administradores.
- **Avaliação e Manutenção:** A avaliação é feita a partir dos retornos dados pelos clientes e usuários, a manutenção pode ser classificada em 3 tipos que são descritas na tabela 1 : manutenção do conteúdo; manutenção do sistema que pode ser do tipo corretiva, preventiva, de aperfeiçoamento e adaptativa; e manutenção de *hardware*.

Tabela 1: Tipos de manutenção (GINIGE, 2002 apud KAPPEL, 2006)

Tipos	Sub-classes	Descrições
Manutenção de Conteúdo		Foco da manutenção das informações armazenadas no sistema.
Manutenção de Sistema	Corretiva	Atua em ações corretivas realizadas em decorrência de falhas ocorridas no sistema.
	Preventiva	Execução de ações em caráter preventivo para evitar a ocorrência de falhas no sistema.
	Adaptativa	Busca adaptar o sistema às novas necessidades e requisitos do usuário.
	Aperfeiçoamento	Atua na melhoria das funcionalidades do sistema através de atualizações e melhorias de desempenho.
Manutenção de Hardware		Atua na manutenção dos equipamentos físicos que compõem o sistema.

- **Gerenciamento do Projeto, Documentação e Controle e Garantia de Qualidade:** O Gerenciamento de Projetos são para garantir que os principais processos e atividades trabalhem em harmonia no desenvolvimento do sistema e aplicação *web*. A Documentação e Controle e Garantia de Qualidade são atividades importantes no processo de desenvolvimento, pois possuem métodos e técnicas para a realização bem sucedida da engenharia.

Murugesan e Ginige (2005) ressaltam que na fase de análise de contexto, que extrai os principais objetivos do sistema e a real necessidade dos usuário ou organização, as exigências estabelecidas podem sofrer alterações ou evoluções, mesmo durante o desenvolvimento do processo ou após a sua distribuição na rede. Por esse motivo os autores recomendam fortemente um breve estudo das operações que serão aplicadas no desenvolvimento de sistemas *web*.

A grande preocupação com a análise do contexto é pelo fato que se ela extrair os requisitos e objetivos do sistema de forma correta, pode minimizar ou eliminar os principais problemas que tanto denigrem o desenvolvimento de sistemas *web*. Muitos profissionais da área negligenciam essa etapa do desenvolvimento e em etapas mais avançadas, encontrar problemas e tem dificuldades em corrigi-los (MURUGESAN; GINIGE, 2005).

As vantagens apresentadas por esse modelo de processo são vastas, pois ele ajuda o entendimento do contexto da aplicação e a identificar os requisitos, facilita a comunicação entre os vários membros envolvidos no desenvolvimento da aplicação, tem um contínuo suporte de manutenção e evolução, ainda facilita o gerenciamento da informação e ajuda na gestão da complexidade em diversos tipos de processos de desenvolvimento (MURUGESAN; GINIGE, 2005).

Os modelos de processo de Cascata, Espiral e Evolucionário, nasceram a partir dos processos da engenharia de *software* tradicional e das experiências de desenvolvimento de sistemas *web*. O modelo em cascata, apesar de seus pontos críticos, ainda é muito utilizado no desenvolvimento de sistemas de informação, seja eles *web* ou não. O modelo Espiral é muito apropriado, ao desenvolvimento de sistemas *web*, pois através dos protótipos, que o próprio processo desenvolve, essa adaptação a mudanças é garantida. Já o modelo evolucionário é adequado quando se trata de sites *web* ou portais, as atividades de Gerenciamento do Projeto,

Documentação e Controle e Garantia de Qualidade, que são realizadas de forma paralela com todas as outras atividades, garantem a qualidade da evolução do site.

2.2 Processos Ágeis

Os métodos de engenharia de *software* tradicional sempre foram compostos por conjuntos de atividades. A partir da década de 90 surgiram novos métodos de desenvolvimento de *software*, com a finalidade dos processos adaptarem-se a mudanças, tendo como apoio a equipe de trabalho (FAGUNDES; DETERS; SANTOS, 2008).

Os processos ágeis surgiram oficialmente no ano de 2001 quando a “Aliança Ágil”, grupo composto por 17 pessoas entre desenvolvedores, produtores e consultores de *software*, assinou um manifesto para o desenvolvimento ágil de *software*. O grupo declarou estar descobrindo melhores maneiras de desenvolvimento de *software*, valorizando aspectos como: indivíduos e interações, *software* funcionando, a colaboração entre clientes e a resposta a modificações (PRESSMAN, 2006).

Pressman (2006), diz que a engenharia de *software* ágil foi desenvolvida para superar as deficiências apresentadas pela engenharia de *software* tradicional. Apesar de apresentar importantes benefícios de desenvolvimento, a Engenharia Ágil não é aplicável a todos os projetos e situação. Diz não ser contrária a engenharia de *software* tradicional e pode ser aplicada como uma filosofia a todo trabalho de *software*.

A engenharia de *software* ágil é definida por Pressman (2006), como uma junção de filosofia e diretrizes. A filosofia abrange a satisfação ao cliente, à rápida entrega do produto ao cliente, pequenas equipes que trabalham motivadas, métodos informais, produtos de trabalhos de engenharia de *software* mínimos e simplicidade no desenvolvimento. E as diretrizes de desenvolvimento ressaltam a entrega em contraposição a análise, projeto e a comunicação entre desenvolvedores e clientes.

Os 17 notáveis que formam a “Aliança Ágil”, descrevem doze princípios definindo os critérios que um método de desenvolvimento deve adequar-se para alcançar a agilidade (PRESSMAN,2006) (AMBLER, 2004 apud FAGUNDES; DETERS; SANTOS, 2008) (BECK, et. al. 2001 apud FAGUNDES; DETERS; SANTOS, 2008):

- A prioridade está na satisfação do cliente, por meio de entregas contínuas de software;
- Aproveitar das freqüentes modificações para a competitividade do cliente;
- Freqüentes, atualizações do sistema;
- Os desenvolvedores e profissionais de negócios, devem trabalhar juntos durante todo o processo;
- Equipe de trabalho motivada, com um ambiente de apoio necessário;
- Constante troca de informações do grupo;
- Garantir o funcionamento do software;
- Desenvolvimento sustentável. Os envolvidos no processo devem manter um ritmo constante;
- Atenção dos envolvidos deve ser constante;
- Simplicidade é essencial para minimizar a quantidade de trabalho;
- Equipes auto-organizadas produzem as melhores arquiteturas, requisitos e projetos;
- A equipe frequentemente deve refletir sobre como tornar-se mais ativa.

A agilidade pode ser aplicada em todo processo de *software*. Porém é necessário que o processo seja de fácil adaptação das tarefas pela equipe de trabalho (PRESSMAN, 2006).

A adaptação é uma característica marcante nos processos ágeis, pois Pressman (2006), diz que todo processo ágil deve ser adaptável, mas de forma incremental. Uma equipe de desenvolvimento ágil, necessita de um *feedback* do cliente, para que as adaptações incrementais sejam realizadas conforme a sua necessidade. Ou seja, as adaptações e incrementos devem ser entregues em curtos períodos, para que o cliente possa avaliar as modificações regularmente, fornecendo o *feedback* de forma implícita e influenciando as adaptações do sistema.

Na seqüência deste trabalho serão apresentados alguns dos modelos de processos ágeis. Existem vários, porem foram escolhidos dois deles, para representar esse modelo de desenvolvimento de software para web, o Extreme Programming (XP) e o *Scrum*. Também é importante perceber a filosofia que envolve os processos ágeis, o respeito em relação ao manifesto para o desenvolvimento ágil e os princípios apresentado anteriormente.

2.2.1 Extreme Programming (XP)

A metodologia XP tem como uma de suas características, suprir as necessidades de pequenas e médias equipes de trabalhos, que desenvolvem *softwares* com requisitos vagos e de rápidas mudanças (SOARES). Ele tem uma abordagem orientada a objetos e inclui um conjunto de quatro regras práticas: planejamento, projeto, codificações e teste. A figura 4 apresenta o conjunto de regras e algumas tarefas extras que estão associadas a cada atividade (PRESSMAN, 2006).

A primeira atividade é a de planejamento, tem por objetivo identificar as prioridades do projeto (SOARES). Pressman (2006), explica que nesta fase é criado um conjunto de história contada pelo cliente, onde são definidas características e funcionalidades do sistema. E ressalta a importância de cliente e equipe XP trabalharem juntos na decisão de junção das histórias para as versões seguintes.

A simplicidade é o que caracteriza a atividade de projeto (segunda atividade), preocupando-se com os requisitos atuais e não com os futuros. Pois estes devem ser feitos apenas quando for necessário (SOARES).

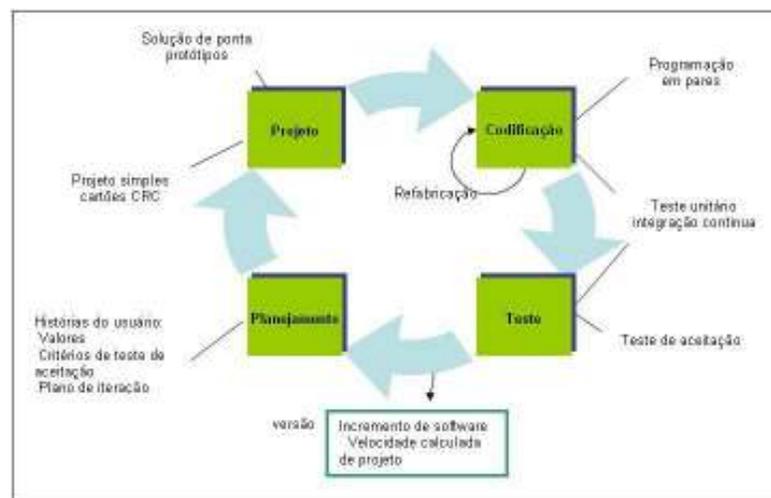


Figura 4: Processo *Extreme Programming*. Baseado em (PRESSMAN, 2006).

Após as histórias serem desenvolvidas e a etapa de pré-projetos efetuada, não recomendável que a equipe avance para os códigos de desenvolvimento. O ideal é desenvolver uma série de testes unitários, com o objetivo de exercitar as

histórias e preparar o desenvolvedor, para que possa estar focado na implementação dos requisitos atuais, sendo esta a terceira atividade (PRESSMAN, 2006). Outra característica desta atividade é a programação em pares, onde duas pessoas trabalham juntas em uma estação de trabalho. Enquanto um realiza a implementação do código, o outro acompanha rigorosamente, para detectar possíveis erros sintáticos e semânticos. A grande vantagem da programação em dupla é dos desenvolvedores estarem trocando experiências e assim aprendendo um com o outro (PRESSMAN, 2006) (SOARES).

A atividade de testes (quarta atividade) já foi previamente comentada na codificação. Eles devem ser criados usando uma estrutura que os realizem automaticamente, dessa forma sempre que há uma modificação, os testes poderão ser executados. Os testes de integração e de validação podem ser executados diariamente, o que garante o contínuo progresso ou o alerta de erros nas modificações. Os testes realizados pelo cliente, também são importantes, pois eles focalizam características e funcionalidades cotidianas (PRESSMAN, 2006).

2.2.2 Scrum

Os princípios do *Scrum* são consistentes ao do manifesto: utilizam pequenas equipes de trabalho; processo deve ser adaptável, tanto em técnicas quanto em negócio; freqüentes incrementos de software no processo; divisão de tarefas entre as equipes de trabalho; a medida que o produto for desenvolvido, testes e documentações também devem estar prontas; e finalmente fornece o produto 'pronto' (PRESSMAN, 2006). Tem como foco desenvolver uma forma de trabalho entre os membros da equipe, com o objetivo de produzir um *software* de forma flexível para ambientes instáveis, ou seja, de constante mudança (SOARES).

O *Scrum* trabalha com um conjunto de padrões, que tem o objetivo de apresentar resultados dentro do prazo estipulado, geralmente curtos. Cada padrão define um conjunto de atividade de desenvolvimento: Pendência, *Sprint*, Reuniões *Scrum* e Demos (PRESSMAN, 2006).

As pendências são listas de requisitos ou características dos sistemas priorizadas, itens podem ser incluídos a qualquer momento, o gerente avalia e indica a prioridade. Os *Sprint* são unidades de trabalhos que tem o objetivo de satisfazer requisitos definidos nas pendências com prazos predefinidos. As reuniões *Scrum*

são realizadas pela equipe de trabalho e cada membro deve responder três perguntas: O que vocês fez desde a ultima reunião de equipe?; Que obstáculo você está encontrando? O que você planeja realizar até a próxima reunião de equipe? Essas reuniões têm o objetivo de descobrir problemas o mais rápido possível. As Demos, é a forma de demonstração e avaliação dos novos implementos dos sistemas ao cliente, através de atualizações e testes (PRESSMAN, 2006).

Os processos ágeis possuem metodologias e características semelhantes, uma vez que surgiram oficialmente através do manifesto que originou a “Aliança Ágil”. Ambos processos apresentados neste trabalho, XP e *Scrum*, trabalham em equipe com o objetivo de garantir um desenvolvimento ágil e com qualidade. O que difere são as atividades que compõem cada uma delas, mas todas tendem a constante atualizações do sistema.

2.3 Modelagem Hipermídia

Organizações, indústrias, comércios e até mesmo pessoas físicas disponibilizam uma grande quantidade de informação na internet. Porém, o desenvolvimento desses sites, na sua grande maioria, é improvisado e o resultado é um produto de má qualidade e com estrutura “espaguete”, ou seja, não existe estrutura, o que torna difícil a manutenção e a atualização (COSTAGLIOLA, G.; FERRUCCI, F.; FRANCESE, R, 2002).

Muitas propostas de modelos e metodologias de engenharia *Web*/Hipermídia foram desenvolvidas. Elas têm por objetivo ajudar o desenvolvedor pensar de forma estruturada, com o foco em aspectos específicos como os *links* de estrutura e navegação, além de manter a comunicação entre programadores, analistas, administrador do software e usuário (COSTAGLIOLA, G.; FERRUCCI, F.; FRANCESE, R, 2002).

A seguir serão apresentadas três metodologias de Hipermídia utilizadas para o desenvolvimento de sistemas para *web*: OOHDM, RMM e *WebML*.

2.3.1 RMM (*Relationship Management Methodology*)

RMM é uma metodologia para o desenvolvimento de aplicações hipermídia. É definido como um veículo para gerenciar o relacionamento entre objetos de

informação, e pode ser melhor aplicado em sistemas que apresentam características de uma estrutura regular do domínio de interesse, como classes de objetos, relacionamentos definidos entre classes e objetos que as instanciam (ISAKOWITZ, STOHR e BALASUBRANANIAN ,1995 apud ALVARES, P, 2001).

Segundo Isakowitz, Stohr e Balasubrananian (1995 apud COSTAGLIOLA, G.; FERRUCCI, F.; FRANCESE, 2002), o RMM tem por objetivo facilitar o desenvolvimento de sites fazendo a integração com a base de dados e tarefas do sistema de informação. Com essa peculiaridade é possível obter a separação dos dados, da estrutura e da interface do usuário. Também utiliza o RMDM (*Relationship Management Data Methodology*) como modelo de dados, que é baseado no modelo Entidade – Relacionamento (E-R). O RMDM é uma linguagem que descreve objetos de informação e o modelo navegacional de aplicações hipermídia. Essa linguagem descreve primitivas de domínio e de acesso, que são apresentadas na figura 5.

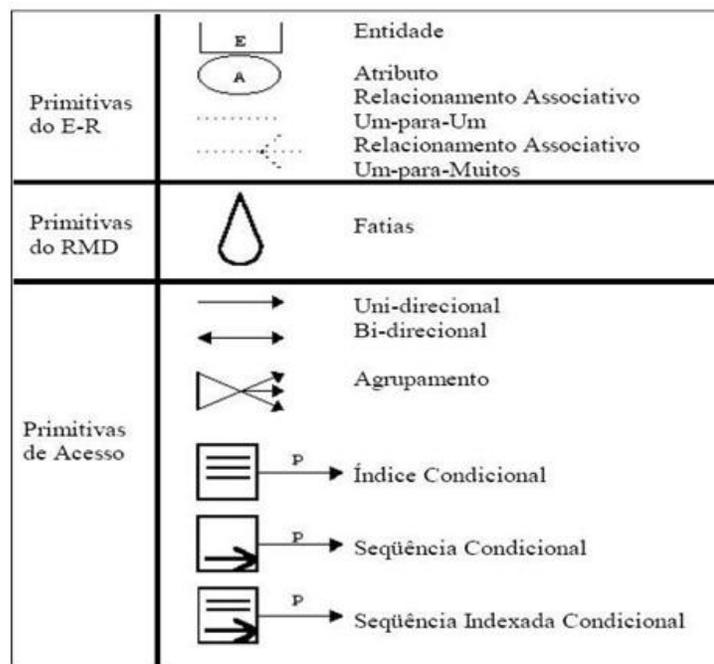


Figura 5: primitivas do RMDM (COSTAGLIOLA, G.; FERRUCCI, F.; FRANCESE, 2002)

A modelagem RMM possui um ciclo de vida completo e dividido em sete etapas de viabilidade. Essas etapas são denominadas como: análise de requisitos, projeto E-R, projeto de fatias, projeto navegacional, projeto de interface com o usuário, construção, teste e avaliação. Ao final de cada fase são gerados alguns produtos

como resultados da especificação. (ISAKOWITZ, STOHR e BALASUBRANIAN, 1995 apud SEVERO, 2001)

Durante as fases iniciais são desenvolvidos os documentos de possibilidade, requisitos e definição do hardware. Com o documento de possibilidades é avaliada a necessidade, objetivo, análise de mercado e um levantamento das mídias que serão utilizadas na aplicação, e os canais de distribuição e custo/benefício (ISAKOWITZ, STOHR e BALASUBRANIAN, 1995 apud SEVERO, 2001).

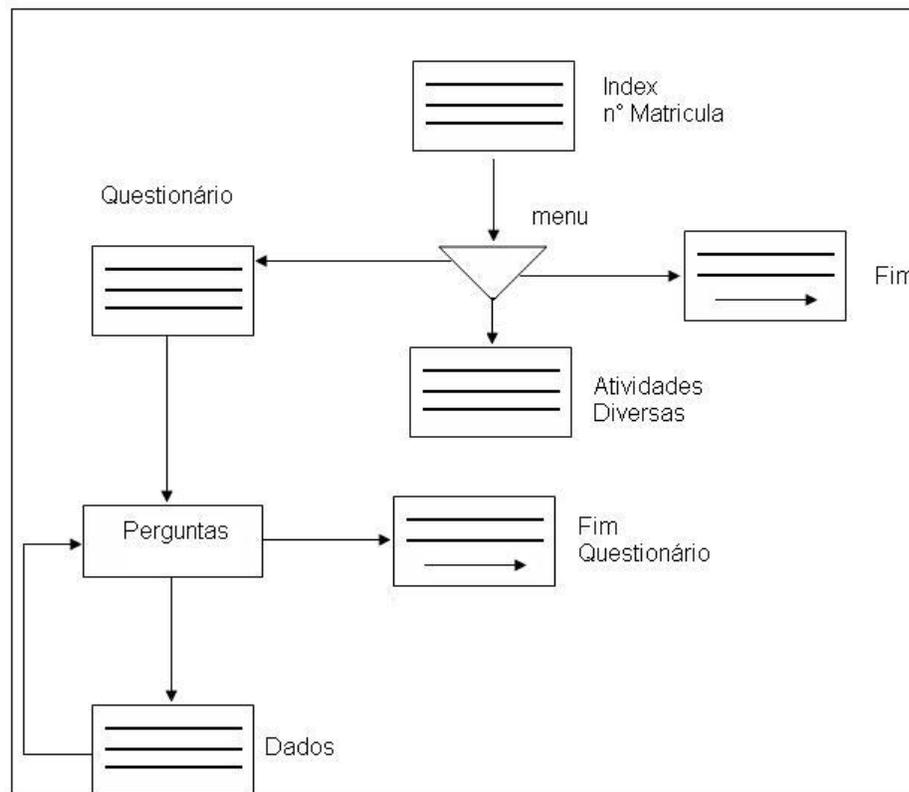


Figura 6: Diagrama do RMDM, baseado em Costagliola; Ferrucci e Francese (2002)

Isakowitz, Stohr e Balasubranian (1995 apud SEVERO, 2001) dão ênfase nas seguintes etapas: projeto E-R, projeto de fatias e projeto navegacional. A fase de projeto E-R, modela o domínio da aplicação através das entidades e relacionamentos gerados pelo diagrama de entidade e relacionamento. Na fase seguinte, projeto de fatias, no diagrama da fase anterior é acrescentado um modelo que especifica as partes de uma informação, denominando o diagrama E-R+. E na fase de projeto navegacional é especificado um modelo de navegação da aplicação, definindo as estruturas de acesso das informações. Ao fim dessa fase é gerado o

diagrama RMDM que é apresentado na figura 6, com algumas características do Questionário do projeto ENVOLVA-SE, já citado na introdução deste trabalho, que foi desenvolvido ao final deste trabalho.

As fases de projeto de interface com o usuário, construção, teste e avaliação, seguem as seguintes instruções respectivamente: para cada objeto do diagrama RMDM é criado um layout, projetar a forma que cada elemento de navegação será implementado, construir e testar (ISAKOWITZ, STOHR e BALASUBRANANIAN, 1995 apud ALVARES, P, 2001). Lembrando que esse modelo de projeto não é linear, pois existem um *feedback* durante as etapas de desenvolvimento (ISAKOWITZ, STOHR e BALASUBRANANIAN, 1995 apud SEVERO, 2001).

2.3.2 OOHDM (*Object-Oriented Hypermedia Design Model*)

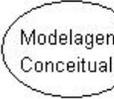
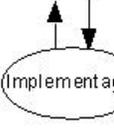
O OOHDM (Metodologia de Desenvolvimento de Hipermídia Orientado a Objeto) é uma metodologia que tem o processo de desenvolvimento de uma aplicação hipermídia dividido em quatro partes: modelagem de domínio ou conceitual, projeto navegacional, projeto da interface abstrata e implementação. Elas foram desenvolvidas em uma mistura de estilos iterativos e incrementais de desenvolvimento, e em cada etapa um modelo é construído (ROSSI, 1996).

A tabela 2 apresenta as características das quatro partes de desenvolvimento da modelagem OOHDM. As setas representam a evolução natural do projeto. Rossi (1996) apresenta a descrição das quatro partes: modelagem conceitual, projeto de navegação, projeto da interface abstrata e implementação.

Durante a modelagem conceitual é construído um modelo de domínio que utiliza princípios conhecidos de modelagem orientada a objeto adicionados a algumas primitivas como perceptivas de atributos e sub-sistemas. O modelo dessa etapa é um esquema de classes e objetos. Na metodologia do OOHDM, uma aplicação é tem uma visão navegacional do modelo conceitual. É no projeto navegacional a construção dessa visão, que considera os diferentes usuários do sistema, e o conjunto de tarefas que deverão realizar. Para um projeto é possível desenvolver diferentes modelos navegacionais, dando diferentes visões da mesma aplicação. Na estrutura navegacional de um hipermídia são definidas as classes navegacionais, que mostram a visão escolhida sobre o domínio da aplicação. Durante essa etapa também é definido o conjunto de objetos navegacionais

acessíveis em cada momento. Um protótipo de interface é construído durante o projeto de interface abstrata. É neste protótipo que será estipulado os objetos que serão vistos pelos usuários. Com a separação do projeto navegacional e da interface abstrata é possível construir diferentes interfaces para a mesma navegação, adequando-se as variações e as preferências do usuário ou da tecnologia. E finalmente, após definidos os modelos navegacionais e de interface, é possível implementar a aplicação de desenvolvimento hipermídia (ROSSI, 1996)

Tabela 2: Tabela de Características da Metodologia OOHDM, (ROSSI, 1996).

Atividades	Produtos	Mecanismos	Interesses do Projeto
	Classes, sub-sistemas, relacionamentos, perspectivas de atributos	Classificação, composição, generalização e especialização	Modelagem da semântica do domínio de aplicação
	Nós, elos, estruturas de acesso, contextos de navegação, transformações navegacionais	Mapeamento entre objetos conceituais e de navegação. Padrões de navegação para a descrição da estrutura geral da aplicação.	Leva em conta o perfil do usuário e a tarefa; ênfase em aspectos cognitivos e arquiteturais.
	Objetos de interface abstrata, reações a eventos externos, transformações de interface	Mapeamento entre objetos de navegação e objetos de interface.	Modelagem de objetos perceptíveis, implementa metáforas escolhidas. Descrição de interface para objetos navegacionais
	Aplicação em execução	Aqueles fornecidos pelo ambiente alvo	Desempenho, completitude

Costagliola; Ferrucci e Francese (2002) dizem que a metodologia OOHDM melhora a sustentabilidade e a reutilização, em função da separação das quatro etapas e capacidade de abstrair características orientas a objetos.

2.3.3 WebMI (Web Modeling Language)

A *WebMI* é uma linguagem de modelagem para *web*. Representa aplicações *web* com alto nível, com conceitos de representação gráfica, e de fácil adaptação a ferramentas CASE e de geração a código automático devido associação com a sintaxe XML. Também está direcionada a atender todas as partes de

desenvolvimento do processo de criação de uma aplicação. A especificação para a construção de um *site* através da modelagem *WebMI* consiste em quatro partes: modelo de estrutura, modelo de hipertexto, modelo de apresentação, e modelo personalizado (CERI, S.; FRATERNALI, P.; PARABOSCHI, S, 1999).

2.3.3.1 O modelo de estrutura

O modelo de estrutura tem por objetivo especificar os dados de uma aplicação, de maneira formal, intuitiva e prática. A *WebMI* não propõem uma linguagem de modelagem de dados, e é compatível com as notações do modelo E-R como mostra a figura 7 e do diagrama de classes da UML (CERI, S.; FRATERNALI, P.; PARABOSCHI, S, 1999).

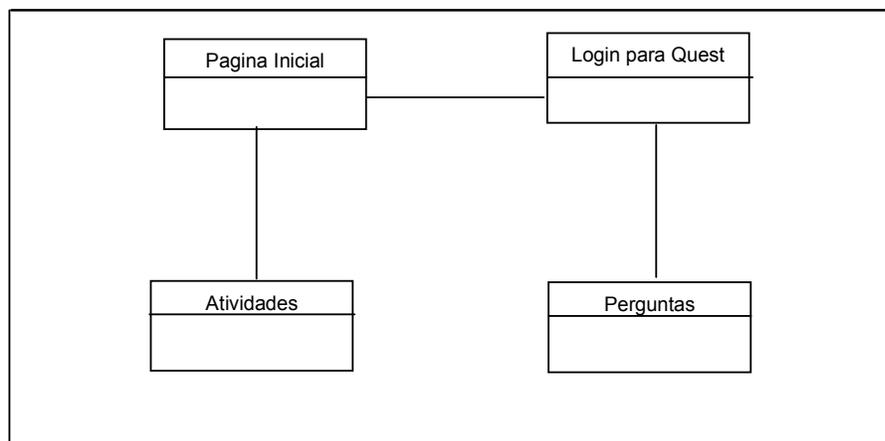


Figura 7: Parte de modelo E-R do Projeto Envolve-se. Baseado em (CERI, S.; FRATERNALI, P.; PARABOSCHI, S, 1999).

2.3.3.2 O modelo de Hipertexto

O modelo Hipertexto é a descrição de um ou mais hipertexto, que podem conter dentro do site. Cada hipertexto define uma visão do site. Essa visão está dividida em dois sub-modelos: modelo de composição e o de navegação. O modelo de composição é a especificação quais objetos irão compor o hipertexto da *website*, que é a unidade de conteúdo ou a página. Em uma possível implementação a unidade de conteúdo poderia ser arquivos HTML e as páginas *frames* HTML, que são responsáveis por organizar estes arquivos. Ainda existem seis tipos de unidade

de conteúdo: dados, multi-dado, índices, filtro, unidade de rolagem e de direção. O modelo de navegação expressa a forma em que as páginas e as unidades de conteúdo são ligadas com o hipertexto. É essa ligação que faz com que o usuário interaja com a aplicação, disponibilizando comandos e especificando caminhos entre as páginas (CERI, S.; FRATERNALI, P.; PARABOSCHI, S, 1999).

2.3.3.3 O modelo de Apresentação

O modelo de apresentação tem por objetivo melhorar o visual das páginas, se preocupando com o *layout* e a aparência gráfica das páginas, independente de *browser*. A apresentação pode ser de dois tipos, páginas específicas ou genéricas. As páginas específicas descrevem o *layout* independente do conteúdo, já as páginas genéricas são modelos já pré-definidos e depende da especificação do conteúdo (CERI, S.; FRATERNALI, P.; PARABOSCHI, S, 1999).

2.3.3.4 O modelo de personalização

O modelo de personalização permite criar páginas e estilos de apresentação e acordo com o perfil dos usuários. Nesse modelo são coletados dados do usuário, a fim de classificar os diferentes perfis de usuários e grupos de usuários, e é formado por dois processos complementares: personalização declarativa e procedural. A primeira define conteúdo e conceitos específicos que compõem o perfil dos usuários, dessa forma o sistema identifica o perfil e apresenta os elementos relevantes. Na personalização procedural é definida as regras de negócio, que coleta as informações específicas sobre os usuários (CERI, S.; FRATERNALI, P.; PARABOSCHI, S, 1999).

Nas metodologias apresentadas neste trabalho é encontrada a preocupação com a estrutura navegacional e de interface do usuário. Pois elas foram desenvolvidas, para todo e qualquer tipo de sistema de informação que se preocupa com a forma simples de navegação e entendimento da interface de que o usuário sinta-se a vontade em utilizá-lo.

3. Sistemas de Informação e sua Taxonomia

Os sistemas de informação tornaram-se ferramentas de grande influência ao sucesso de empresas e organizações (O'BRIEN, 2002). Nas últimas três décadas, eles sofreram importantes mudanças, passando de ferramentas de estratégia e gerenciamento para um elemento capaz de transformar a estrutura organizacional de uma empresa (KAPPEL, 2006). Essas mudanças estão relacionadas ao surgimento de novas tecnologias e aplicações para atender as exigências do ambiente (ALBERTIN, 2001). Com esse contexto empresarial e de mudanças, este capítulo é voltado inteiramente ao estudo dos sistemas de informações.

Após apresentar a taxonomia dos sistemas de informações e correlacioná-la com métodos e técnicas de desenvolvimento de *software* para *web*, apresentados no capítulo anterior. Neste capítulo serão apresentadas definições de sistemas de informação, e um breve histórico da sua evolução. Este tem como objetivo caracterizar os tipos de sistemas de informação que podem é encontrado como os científicos, comercial, ferramentas, embarcados e os baseados na *web*, essas denominações são extraídas das experiências profissionais e didáticas do orientador deste trabalho Professor e Mestre Salvador Antonio dos Santos. Dessa forma é definido a Taxonomia como uma classificação científica, segundo o dicionário online Priberam.

3.1 Definição dos Sistemas de Informação

Na literatura é possível encontrar várias definições de Sistemas de Informação. Autores como Hirschhein, Klein, Lyytienem (1995 apud Zaneti, 2003) possuem duas perspectiva em relação aos sistemas de informação, uma relacionada à função e outra à estrutura. A Funcional refere-se aos aspectos técnicos como gravar, armazenar e distribuir os dados. E a estrutural diz respeito às aspectos como pessoas envolvidas, processos, dados, modelos, tecnologia e linguagem formalizada. Porém nas duas perspectivas é possível perceber a influência de pessoas expressando sua capacidade e contribuições para que os sistemas de informações atinjam suas expectativas (ZANETI, 2003).

Outra definição diz que os sistemas de informações, são sistemas técnicos capaz de realizar atividades racionais, implicando em um aumento de eficiência, redução de custos e economia de tempo (WAEMA e WALSHAM, 1994 apud KAPEL, 2006). O'Brien (2002) define sistemas de informação como um conjunto formado de pessoas organizadas, *hardware*, *software*, rede de comunicação e recursos de banco de dados, que tem a função de coletar, transformar e compartilhar as informações, conforme apresenta a figura 8.

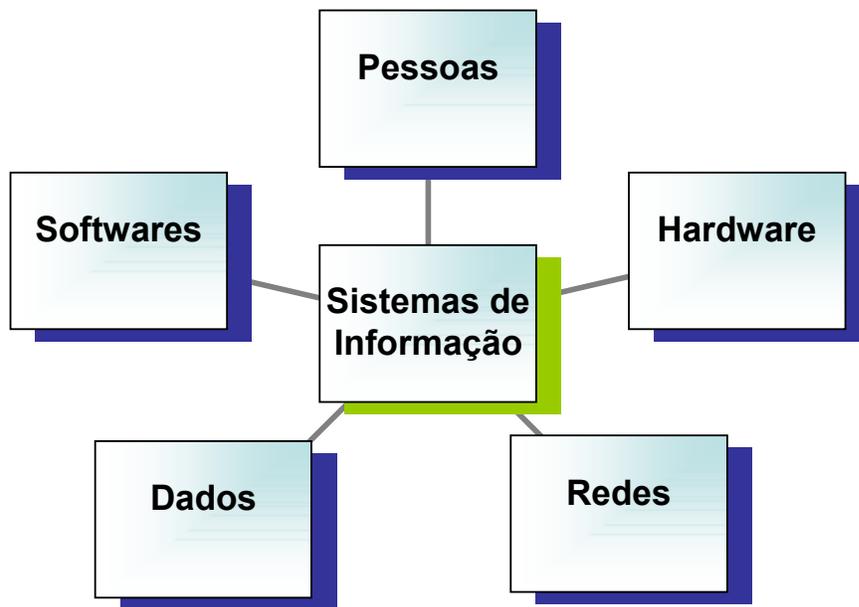


Figura 8: Sistema de Informação. Adaptado de (O'BRIEN, 2002).

Laudon e Laudon (2004) definem sistemas de informação como um conjunto, porém de elementos relacionados à coleta, processo, armazenamento e distribuição de informação, com o objetivo de ajudar no processo de organização e tomada de decisão. Existem três atividades em um sistema de informação que produzem as informações necessárias para gerencia das organizações, elas são: entrada, processamento e saída.

A entrada tem função de capturar os dados da organização. O processamento transforma esses dados em informação significativa. E a saída repassa essas informações processadas para o usuário. Também se faz necessário um *feedback*, que é uma novo dado de entrada, disponibilizado para um membro da organização, com o objetivo de ajudar avaliar e corrigir a entrada de dados, como apresenta a figura 9 (LAUDON; LAUDON, 2004).

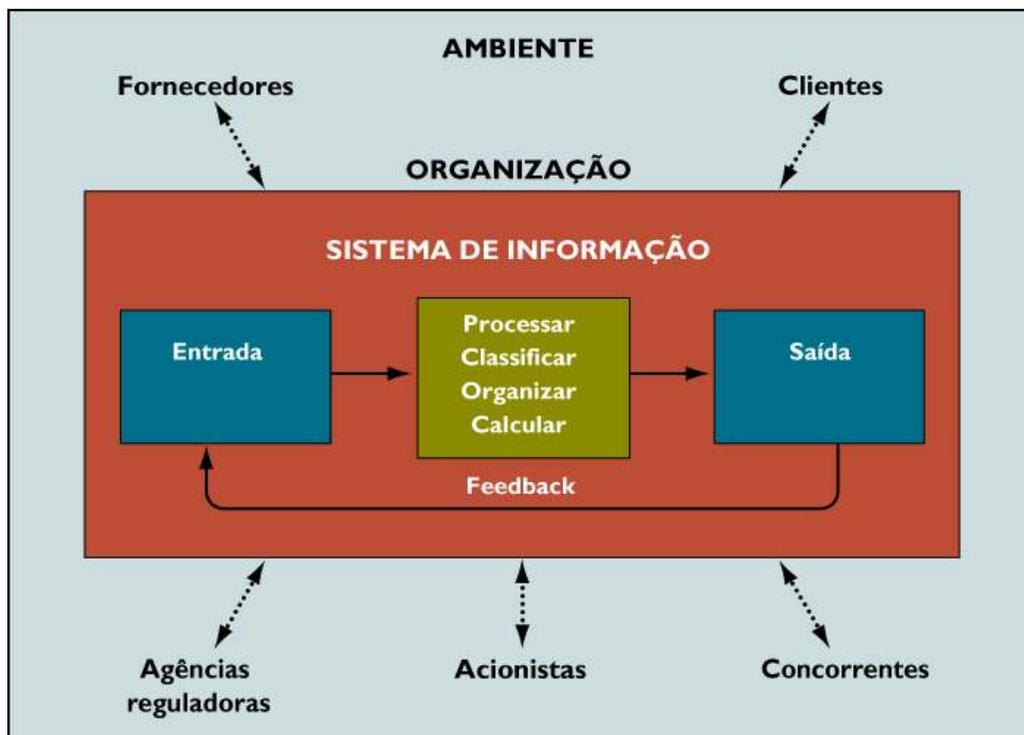


Figura 9: Funções de um sistema de informação (LAUDON; LAUDON, 2004).

Desta forma, é apresentada as definições de Sistemas de Informação. Nas seções que seguem, serão apresentados os tipos de sistemas de informação, propondo assim a taxonomia dos sistemas de informação que é composta pelos sistemas científicos, comerciais, ferramentas de administração e engenharia, os sistemas embarcados e os sistemas de informação baseados na *web*.

3.2 Sistemas de Informação Científicos

Nos anos 40 os computadores tinham funções científicas, com o objetivo de realizar cálculos de seno, co-seno e outros problemas matemáticos (GUEDES; SILVA, 2006). A principal característica está na velocidade de processamento, ou seja, as informações eram inseridas nas grandes máquinas, que possuíam cerca de 20.000 válvulas, realizava o processamento dos dados e apresentavam as informações ao usuário. Neste momento da história, não existiam qualquer tipo de sistema operacional ou interação com o usuário, por esse motivo alguns computadores comunicavam-se com os usuários através de luzes coloridas.

Os sistemas científicos evoluíram com a tecnologia dos computadores, ainda é possível realizar grandes cálculos de âmbito científico. Porém com sistemas de informação que possui interatividade com o usuário, entre outros aspectos.

3.3 Sistemas de Informação Comercial

Os sistemas comerciais possuem três fases de evolução, os sistemas comercial, comercial integrado e comercial interativo. Neste momento será descrito cada um deles e suas principais características que surgiram com a evolução.

Os sistemas de informação comercial foram originados a partir dos sistemas científicos, como já citado anteriormente. Essa primeira fase foi caracterizada pela velocidade de I/O (entrada e saída) de dados, isto é, as informações eram repassadas aos computadores, via cartão perfurado, processadas e apresentada o resultado para o usuário, sendo a novidade da época as grandes impressoras.

A segunda fase apresenta os sistemas comerciais integrados, que além de ter velocidade de processamento, velocidade de I/O, surge à necessidade de utilizar um banco de dados. O seu objetivo era centralizar as informações para que estas não ficassem perdidas.

A ultima fase surgiu a partir da necessidade de interatividade entre máquina e usuário, por esse motivo chamamos de sistema comercial interativo. Neste momento os sistemas de informação conquistaram grandes avanços tecnológicos, como o aumento da capacidade do banco de dados, a interatividade com o usuário, que no inicio era apenas em interface caractere e posteriormente interface gráfica, encontrada atualmente. Outra grande evolução surgiu através das redes de comunicação e seus protocolos, que permitem o compartilhamento de informações e o acesso ao banco de dados por vários usuários simultaneamente, como acontece hoje nos ERP II.

A principal característica desse tipo de sistemas informação está na entrada de saída de dados (I/O). Os sistemas comerciais possuem um grande volume de entrada e saída de dados, porem pouco tempo de processamento como nos sistemas científicos.

3.4 Ferramentas

A partir dos sistemas científicos e comerciais, surge um novo tipo de sistemas de informação, que iremos chamar de Ferramentas, pois auxiliam o usuário em seu trabalho cotidiano. Os sistemas científicos serviram de base para as demais linhas de sistemas de informação, permitindo que elas herdassem a característica de velocidade de processamento. Dessa forma as ferramentas de administração e de engenharia permaneceram com esta característica, o que é essencial principalmente para cálculos complexos da engenharia. Também os sistemas comerciais possuem grande participação nas características desse sistema de informação, principalmente em relação a administração de negócios.

As ferramentas de Administração, que iremos chamá-los de sistemas de escritórios, como os *Microsoft Office*, possuem maior quantidade de características dos sistemas comerciais do que científicos, que ajuda nas tomadas de decisão e organização das empresas. Já as ferramentas de Engenharia é o inverso, possuem características comerciais, porém as que prevalecem são as científicas, principalmente pela quantidade de processamento dos cálculos.

Apesar de essas ferramentas terem origem dos sistemas científicos e comerciais, novas tecnologias foram implementadas a elas, como a interface gráfica e mais processamento. Mais tarde ganha interatividade com o usuário, ou seja, a troca de informação entre máquina e usuários através de interfaces caractere ou gráfica. Temos como exemplo de ferramentas de engenharia os sistemas *AutoCad*, que exigem processamento, interface gráfica para a elaboração de desenhos e a interatividade com usuário.

3.5 Sistemas de Informação Embarcados

A linha dos sistemas embarcados, também é originada dos sistemas científicos, e permanece com suas características. Uma vez que na sua atual evolução, chegamos aos equipamentos de automação industrial e robótica. A evolução desse tipo de equipamento teve três fases os *softwares* embarcados, *softwares* atuantes e *softwares* de automação industrial.

Os softwares embarcados também partilham da herança deixada pelos sistemas científicos, à velocidade de processamento, tendo essa como sua principal característica. Estes são geralmente instalados em equipamentos industriais, que

tem por objetivo o auxílio tecnológico na produção de chão de fábrica, como nos equipamento CNC (*Computer Numeric Control*).

Nos *softwares* atuantes, segunda fase de evolução dessa linha de sistemas de informação, foi adicionada como característica a velocidade de I/O. Essa nova característica tem a finalidade de facilitar a entrada de dados, em interfaces gráficas, e por esse motivo esse sistema é denominado como atuante. Pois era necessário um sistema bastante sofisticado para que realizar as funções de um sistema de informação: entrada dos dados, processamento e a saída do resultado.

A última evolução é denominada como sistemas de automação industrial, esses tem como características a implantação de banco de dados e de redes de comunicação. Essas duas características tornaram-se necessárias para que sejam simuladas as ações que devem ser realizadas até chegar ao produto final.

3.6 Sistemas de Informação Web

A internet e a tecnologia *web*, foram criadas como uma forma de divulgar o conhecimento científico, com o passar dos anos elas foram evoluíram e modificaram hábitos. Com a evolução a *web* foi transformada em uma poderosa ferramenta em diversas áreas empresariais (SCHMITT; SCHMITT, 2007).

A tecnologia *web* funciona utilizando o paradigma cliente-servidor. Ou seja, seu processamento é dividido entre clientes e servidores, como o próprio nome sugere. Os clientes requisitam uma informação ao servidor, e este repassa ao cliente o resultado da solicitação (ZANETTI JR, 2003).

Zanetti Jr (2003) define a tecnologia *web* como um sistema de padrões, que é composto por um padrão de endereçamento, de comunicação e de estrutura de informação. O padrão de endereçamento é denominado como URL, ele é único e pode ser localizado em qualquer lugar. O padrão de comunicação é bastante conhecido através do padrão HTTP, é uma linguagem que permite a solicitação e a recuperação de recursos da *web*, quando solicitados. E o padrão de estrutura das informações, este é baseado na linguagem de marcação denominada HTML, ela tem por objetivo definir os elementos de visualização das informações.

Neste momento será apresentada a evolução desse conjunto de padrões, conhecido como *web*. Iniciamos com as páginas *web* que foram as pioneiras na apresentação de informação na internet. Seguimos com os *sites*, até chegar aos sistemas de comércio eletrônico e seus derivados.

As páginas *web* são apresentadas ao usuário, através das solicitações feitas para o servidor. Essa solicitação é realizada no momento em que o usuário informa a URL, então é a comunicação entre o navegador e o servidor *web*. Este por sua vez, recupera as informações do endereço, e apresenta ao usuário. As páginas podem conter em seu conteúdo recursos como, botões, figuras ou textos, também elas eram capazes de ligar-se a outras páginas. Pois o navegador é responsável em transformar a ativação dos recursos em solicitação de páginas (ZANETTI JR, 2003).

Apesar das páginas *web* poderem ser ligadas entre si, existem páginas que são agrupadas em conjuntos de informações correspondentes, e ficam armazenadas em um mesmo servidor. Esses conjuntos são chamados de *sites web* (ZANETTI JR, 2003).

O comércio eletrônico é um sistema de informação que interliga compradores e vendedores para a troca de informações, produtos, serviços e pagamentos. Um dos benefícios apresentados pelo comércio eletrônico, está nos baixos custos para a realização de transações de mercados. Além de permitir aos compradores e vendedores realizarem suas transações, de maneira digital e independente da localização (BAKOS, 1998 apud LAUDON; LAUDON, 2004).

Por ser um sistema de informação bastante complexo, os *e-commerce*, de um modo geral, tem a necessidade de um servidor que funcione de forma dedicada. Pois esses sistemas possuem funções específicas, como: a exibição de produtos e preços através de vitrines ou catálogos; projetar carrinhos de compras, possibilitando a coleta de dados do produto desejado pelo cliente; ofertar condições de entrega; conectar os sistemas de processamento dos pagamentos e mostrar estoques rastreando a expedição (LAUDON; LAUDON, 2004).

4. Sistemas de Informação e seus Processos de Desenvolvimento

Neste capítulo será realizado o co-relacionamento entre os sistemas de Informações e seus processos de desenvolvimento, ambos estudados nos capítulos anteriores. Para este co-relacionamento seja realizado é preciso estudar os fatores que caracterizam um sistema de informação, esses fatores foram escolhidos pelo orientador deste trabalho, professor e mestre Salvador Antonio dos Santos, conforme sua experiência profissional e didática dos sistemas de informação.

4.1 Fatores de um Sistema de Informação

Os sistemas de informações possuem vários fatores de caracterização. Neste momento serão apresentados alguns fatores e suas evoluções como os *hardwares*, *softwares*, requisitos de tempo e volatilidade, paradigma de modelagem, ergonomia e usuário. O objetivo de estudar esses fatores está em posteriormente, ainda neste capítulo, realizar o co-relacionamento entre os sistemas de informações e os processos de desenvolvimento através desses fatores.

4.1.2 Hardware

Laudon e Laudon (2004) possuem definição semelhante a Stair e Reynolds (2002) referente aos *hardwares* como um conjunto de periféricos ou equipamentos físicos usados para a entrada, processamento e saída de um sistema de informação. Como um dos fatores escolhidos para co-relacionar com os processos de desenvolvimento e os sistemas de informação, serão apresentados os meios de armazenamento como cartões, fitas e discos magnéticos e o local de processamento, centralizado ou distribuído.

4.1.2.1 Meio de Armazenamento

Na década de 50, surgiu a primeira geração tecnológica de sistemas de informação, que foi marcada pela utilização de cartões perfurados para o armazenamento e processamento off-line (PRESS, 1999 apud ZANETI JR, 2003).

O processamento dos cartões funcionava da seguinte forma: após a perfuração, os cartões eram entregues aos programadores que tinham acesso aos

computadores. Em alguns casos eram necessários era necessário carregar cartões referentes a programas auxiliares. Isso implicava em trabalho dobrado e tempo perdido na espera da resposta do processamento. Para solucionar esse problema, foi desenvolvido um novo sistema de processamento (GUEDES; SILVA, 2006).

A fita magnética é outra forma de armazenamento de informações, apesar de antiga foi utilizada por muito tempo. Era capaz de gravar grandes quantidades de dados, que precisam ser acessados com rapidez, mas não instantaneamente. O sistema de armazenamento é seqüencial, ou seja, para buscar uma informação é necessário que a fita seja lida desde o início até encontrar o registro desejado (LAUDON; LAUDON, 2004).

Após as fitas magnéticas, surgiram os discos magnéticos, que podem ser flexíveis ou rígidos. Os discos flexíveis são removíveis, portáteis e utilizados PC's, porém com capacidade e velocidade de acesso muito inferior ao discos rígidos. Estes por sua vez, permitem o acesso direto aos registros individuais, independente a ordem em que foram registrados originalmente. Surgiram também os discos ópticos, também conhecidos como discos compactados CD-ROM (*Compact Disk Read-Only Memory*), ainda muito utilizado, possui tecnologia a laser de armazenamento, o que permite uma grande quantidade de dados armazenados em forma compactada. No primeiro momento os CD-ROM eram apenas para leitura, ou seja, nenhum dado novo pode ser gravado sobre ele, conhecido como CD-R (*Compact Disk Recordable*). Mas logo surgiu o CD-RW (*CD-ReWritable*), criado para permitir que usuários além de armazenar grandes quantidades de dados, possam atualizá-los também (LAUDON; LAUDON, 2004).

4.1.2.2 Local de Processamento

Ao contrario de um processamento centralizado, que utiliza apenas um computador para executar suas tarefas, o processamento distribuído utiliza vários computadores conectados a um servidor em uma rede de comunicação executando processos (LAUDON; LAUDON, 2004). Neste trabalho é citado rapidamente dois tipos de arquiteturas: cliente/servidor, P2P (*peer-to-peer*)

Na figura 10 apresenta a arquitetura Cliente-Servidor, que possui dois computadores, distintos e interligados em rede, um configurado como Servidor e outro como cliente. A maquina cliente requisita uma tarefa ou dado armazenado no

servidor, este por sua vez, processa a solicitação do cliente e envia a resposta para o mesmo (LAUDON; LAUDON, 2004).

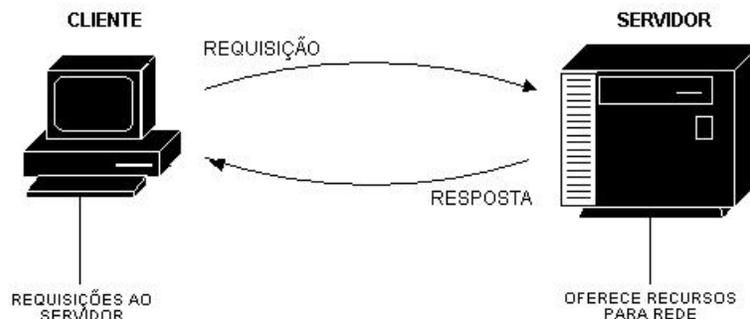


Figura 10: Estrutura Cliente Servidor adaptado (LAUDON; LAUDON, 2004).

A arquitetura P2P utiliza a rede de computadores, dando funções de clientes e servidores a um mesmo computador. Dessa forma possibilita o compartilhamento de informações ou mídias simultâneas entre vários computadores, como descritas na figura 11. Sua principal característica está no processamento em computadores individuais, que trabalham em conjunto sem a presença de servidores ou uma central de controle (LAUDON; LAUDON, 2004).

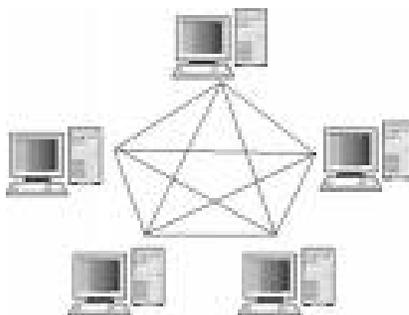


Figura 11: Estrutura P2P.

4.1.3 Software

Software é definido como um “programa para computador que possibilita a operação de equipamentos” (STAIR, R. M; REYNOLDS, 2002). Dentro desta definição, podemos encontrar os *softwares* básicos, que são dito como um conjunto de programas que tem a função de coordenar as atividades e funções de *hardware* e

demais programas encontrados no computador (LAUDON; LAUDON, 2004) (STAIR, R. M; REYNOLDS, 2002).

Neste trabalho também é apresentado como um dos fatores que compõem um sistema de informação os *softwares* básicos, que serão representados pelos sistemas operacionais e o *browser* que faz a conexão com a internet.

4.1.3.1 Software Básico

Laudon e Laudon (2004) definem sistemas operacionais como um gerente geral dos sistemas do computador. Ou seja, acompanha cada tarefa do computador e usuário, também verificar tentativas de acessos não autorizados ao sistema.

Os primeiros computadores a utilizar os MS-DOS (*Microsoft Disk Operating System*), foram os compatíveis aos IBM nos anos 80. Esse sistema operacional é monousuário, processa uma tarefa por vez e é baseado em linhas de comando. Eram muito utilizados em computadores relativamente lentos e que utilizavam sistema de Disquete. Com o passar do tempo, foi necessário um novo sistema operacional que executa-se várias tarefas ao mesmo tempo, denominados como multitarefa (STAIR, R. M; REYNOLDS, 2002).

As primeiras versões do sistema operacional *Windows* eram carregadas após do DOS, que fornecia muitos serviços básicos de sistemas operacionais. Porém o *windows* adicionava um ambiente gráfico e de multitarefas cooperativas entre os dois sistemas operacionais. Os atuais sistemas operacionais utilizam uma interface gráfica ao usuário, também conhecida como GUI, que faz o uso de ícones, botões, barras e caixas para execução de tarefas (LAUDON; LAUDON, 2004) (STAIR, R. M; REYNOLDS, 2002).

Para o acesso a internet, existe outro tipo de sistemas operacional, conhecido como Navegador *Web* ou *Browser*. Laudon e Laudon (2004), dizem que os navegadores são ferramentas projetadas para apresentar páginas *web* e também outros recursos da internet. Esses navegadores podem exibir e apresentar além das tradicionais informações textuais, também informações gráficas como áudios e vídeo. Também permite clicar em botões ou palavras iluminadas na tela para estabelecer links com outros sites.

4.1.4 Requisitos

No desenvolvimento de *software*, é necessário elaborar um documento de requisitos. Nele são registrados todos os requisitos funcionais e de qualidade do *software*, também a capacidade do produto, os recursos disponíveis, os benefícios e critérios de aceitação (TURINE; MASIERO, 1996).

Para o estudo dos fatores de *softwares*, neste trabalho vamos apresentar dois requisitos, de tempo de resposta e volatilidade, para junto com os demais correlacionar com os processos de desenvolvimento e com a taxonomia dos sistemas de informação.

4.1.4.1 Tempo de Resposta

O comportamento de um requisito de tempo é definido como parte do software que confirmam o tempo de resposta do sistema em relação ao processamento e a velocidade de execução das funções. Este requisito garante a eficiência, sendo este um requisito de qualidade. Por exemplo, Turine e Masiero (1996), propõem que os processos de inserção, alteração e exclusão não devem exceder três segundos.

4.1.4.2 Requisito Volatilidade

A volatilidade é definida como uma mudança nos requisitos ao longo do tempo. Este é um dos problemas que são encontrados durante a definição dos requisitos, no processo de desenvolvimento de *software*. Outros problemas também são encontrados na definição do escopo e entendimento (PRESSMAN,2002).

4.1.5 Paradigma de modelagem de software

Durante muitos anos o temo Orientado a Objeto (O.O.), era usado para caracterizar uma abordagem de desenvolvimento. Porém hoje este paradigma tem por característica a fácil manutenção, pois sua estrutura é desmembrada. Isso faz com que modificações sejam bem aceitas pelos sistemas, não causando resultados indesejáveis. Além de fácil adaptação e ampliação do sistema, uma vez que este pode ser montado através de subsistemas ou funções (PRESSMAN,2002).

A modelagem orientada a objeto utiliza uma classe para encapsular atributos necessários para descrever o conteúdo ou comportamento de uma entidade. Os

atributos por sua vez, são encontrados em métodos que podem manipular os dados. São paradigmas como esses que garantem a qualidade e facilidade de desenvolvimento, em sistema de informação que está em constante evolução, como os baseados na *web*.

4.1.6 Ergonomia

A forma de interagir com o usuário é um termo que engloba todas as formas de comunicação entre usuários e sistemas de informação (Preece et al., 1994; Shneiderman, 1998 apud SOUZA, 1999). Para o fator que denominamos como “ícone”, será apresentado o conceito de WIMP (*Windows, Icons, Menus and Pointers*) e entendemos como “mídia”, toda forma de comunicação com o usuário, além de janelas, ícone e menus, como por exemplo, o vídeo e áudio.

O WIMP, que chamaremos de “ícones”, é um estilo de interface através de componentes de iteração virtual, também conhecida como *widgets*. É implementado através do auxílio de interfaces gráficas, que permitem a construção de janelas e controles de entrada de dados, através do teclado, *mouse*. Essa forma de iteração com o usuário é uma junção da tecnologia *hardware* e *software*, com conceitos de janelas e *widgets* que permite o desenvolvimento de vários estilos (SOUZA, 1999).

4.1.7 Usuário

Nos anos 50 os computadores passaram a funcionar com transistores, e quem operava essas grandes máquinas eram os profissionais do CPD (centro de processamento de dados) (GUEDES; SILVA, 2006), que eram altamente qualificados para essa função. Chamaremos esses profissionais de “Técnico em TI”.

Sabe-se que com o desenvolvimento de computadores pessoais (PC's), mais usuários passaram a ter contato com essa máquina. Porém como tudo era novidade, principalmente os sistemas operacionais que possuíam interfaces bastante rudimentares. Era necessário que os usuários tivessem no mínimo um treinamento para executar algumas atividades, outras por sua vez mais complexas, não bastava apenas um treinamento, exigiam uma especialização em determinadas atividades.

Com a total popularização dos PC's, e as diversas vantagens que um computador traz ao homem na atualidade, qualquer usuário é capaz de utilizar um

computador, mesmo que nunca tenha recebido um treinamento, a este chamamos de “usuário casual”.

4.2 Processos versus Fatores

Depois de descrito todos os fatores que são importantes em um sistema de informação, propomos um relacionamento entre os fatores apresentados e processos e metodologias de desenvolvimento de sistemas de informação *web*, já estudados no capítulo 2. Este relacionamento será utilizado para definição do relacionamento entre processos de desenvolvimento e sistemas de informação *web*.

Para o relacionamento apresentado na tabela 3, são definidos os fatores da seguinte forma:

- M.A. – Meio de Armazenamento – Cartões, Fitas e Discos;
- L.P. – Local de Processamento – Centralizado (Centr) ou Distribuído (Distr);
- S.O – Sistema Operacional – Caractere, Gráfico, *Browser*
- T. Resp – Tempo de Resposta – Super Rápido (SR), Interativo e Tempo Real (TR);
- R. Volátil – Volatilidade – Requisito Volátil e não Volátil (N Volatil);
- Pdgma – Paradigma de modelagem de Software – O.O.;
- Ergonomia – Ícone e Mídia;
- Usuário – Técnico em TI (TTI), Treinado, Especialista (Esp.) e Casual.

A proposta deste relacionamento é baseada nas características dos processos de desenvolvimento que apresentam os fatores citados. O processo em Casta apresenta como característica a linearidade de seu processo, que pede a definição de requisitos de forma rígida, pois esse processo não permite evolução. Então este não possui volatilidade, a ergonomia aplicada nesse processo é a de ícones que permite ao usuário acessar o sistema através do teclado ou mouse, e para desenvolver um sistema com esse processo é necessário que o usuário seja treinado, devido a sua complexidade.

Os processos Espiral e Evolucionário assim como os processos ágeis XP e *Scrum*, e também as metodologias de hipermídia RMM, OOHDM e WebML, permitem a evolução ou alteração de requisitos durante o processo de

desenvolvimento, então essas metodologias apresentam requisitos voláteis. Apenas nas metodologias Hiperímídia permitem mídias, são apropriadas para usuários casuais, e é indicado o paradigma Orientado a Objeto, pela sua constante mudança de informações. Os demais processos, assim como o Cascata tem sua ergonomia voltada a ícones e necessita de usuários treinados.

Tabela 3: Proposta de relacionamento entre Fatores versus Processos de Desenvolvimento *Web*.

	M. A	L. P	S.O	R. Volátil	T. Resp	Ergonomia	Usuário	Pdgma
Cascata				N Volátil		Ícones	Treinado	
Espiral				Volátil		Ícones	Treinado	
Evolucionário				Volátil		Ícones	Treinado	
XP				Volátil		Ícones	Treinado	O.O
Scrum				Volátil		Ícones	Treinado	O.O
RMM				Volátil		Mídia	Casual	O.O.
OOHDM				Volátil		Mídia	Casual	O.O.
WebML				Volátil		Mídia	casual	O.O.

4.3 Sistemas de Informação versus Fatores

Da mesma forma que na seção anterior, na tabela 4 será apresentado o relacionado os fatores relevantes a um sistema de informação entre os próprios sistemas de informação baseados na *web*. Para na próxima seção apresentar a conclusão deste capítulo, o co-relacionamento entre os sistemas de informação e seus processos de desenvolvimento.

Os sistemas científicos têm como característica, a baixa entrada e saída de dados e muito processamento, por esse motivo o tempo de resposta deve ser super rápido (SR). A volatilidade é progressiva (progres), ou seja, a evolução acontece, mas não é constante e o usuário desse sistema deve ser um especialista (esp) devido a complexidade das informações que serão processadas. Os sistemas Cae/ Cad/ Cam/ Case, são ferramentas de auxílio a engenharia, que tem o objetivo de realizar cálculos e gráficos complexos. Devido a sua complexidade possui alguns fatores em comum com os sistemas científicos, como a volatilidade progressiva e um usuário especialista. Porém não exige um tempo de resposta rápido e pode ser desenvolvido através do paradigma Orientado a Objeto.

Tabela 4: Proposta de relacionamento entre Fatores versus Sistemas de Informação Web.

	M. A	L. P	S.O	R. Volátil	T. Resp	Ergonomia	Usuário	Pdgm
Científico				Progres	SR		Esp	
ERP II				N. Volátil	Interativo	Ícones	Treinado	O.O.
Sistemas Escritório				Volátil			Treinado	O.O.
Cae/ Cad/ Cam/ Case				Progres			Esp	O.O.
Automação		Centr	Espec	N. Volátil	TR		Treinado	O.O
Sistema Web		Distri	Browser	Volátil		Mídia	Casual	O.O.

Os ERP II são sistemas comerciais, que possuem seus requisitos muito bem definidos, e tem como característica muita entrada e saída de dados e pouco processamento. Com essas duas é possível perceber que este é um sistema que apresenta requisito Não volátil, ou seja, seus requisitos não mudam, sua ergonomia dá acesso ao usuário através de ícones. Por ser um sistema comercial com inúmeras funções, é necessário que o usuário tenha treinamento para manuseá-lo, e sua forma de modelagem é Orientada a Objeto, para facilitar a manutenção do mesmo.

As ferramentas de Administração que chamamos de “Sistemas para escritórios”, têm como característica tem como função auxiliar na organização e tomada de decisões, o que caracteriza um sistema comercial, porém com a complexidade de um sistema Científico. Essa mistura faz com que esse sistema seja volátil para adaptação das necessidades do usuário, e que este seja treinado para usufruir de seus benefícios. E também utilizam paradigma orientado a objetos, facilitando o desenvolvimento e a manutenção como os demais sistemas.

Um sistema embarcado como de automação, tem característica peculiar, por ser um sistema que auxilia no desenvolvimento de produtos entre outras funções. Eles são encontrados em equipamentos específicos em nosso dia a dia, como por exemplo, em uma cafeteira, microondas, também em fabricas como os robôs de solda e corte. Este sistema exige que seu local de processamento seja centralizado, uma vez que podemos encontrá-los dentro de outros produtos eletro-eletrônico. Por esse motivo o sistema operacional deve ser específico (espec), além de apresentar requisito não volátil, necessita de processamento com resposta em tempo real. E

assim como todo sistema complexo, o usuário deve ser treinado e a forma de desenvolvimento Orientada a Objetos.

Os sistemas *web*, são caracterizados pela sua distribuição cliente/servidor e pela rápida mudança de requisito e preocupa-se com a visualização dos sites pelos usuários e também com sua navegação. Os fatores que abrange esses sistemas são bastante específicos, como o local de processamento que é distribuído (Distri), o sistema operacional é substituído pelo *browser*, pois ele é próprio para a comunicação das páginas *web* entre a internet. Também possui requisitos voláteis como já citado, o que pede um desenvolvimento Orientado a Objeto. A ergonomia é voltada a mídia, podendo assim possuir vídeo e áudio, para melhor relação como usuário, uma vez que este pode ser casual, qualquer pessoa é capaz de manusear.

4.4 Adequação dos Processos aos Sistemas

Com os relacionamentos apresentados a cima, Fatores versus Processos e Fatores versus Sistemas de informações. Na tabela 5 é apresentada uma divisão de cores entre igualdades de requisitos do relacionamento entre Processos de Desenvolvimento e Fatores. A cor amarela representa o processo em cascata que possui requisito não volátil, ergonomia de ícones e usuário treinados. Para cor cinza é apresentada os processos Espiral, Evolucionário, XP e Scrum que apresenta requisitos de volatilidade, ergonomia de ícones e usuários treinados. Por fim, a cor roxa apresenta os processos RMM, OOHDM e WebMI que identificam requisitos de volatilidade, ergonomia de mídia, usuário casual e paradigma Orientado a Objeto.

Tabela 5: Igualdade de requisitos do relacionamento entre Processos e Fatores.

	M. A	L. P	S.O	R. Volátil	T. Resp	Ergonomia	Usuário	Pdgm
Cascata				N Volátil		Ícones	Treinado	
Espiral				Volátil		Ícones	Treinado	
Evolucionário				Volátil		Ícones	Treinado	
XP				Volátil		Ícones	Treinado	
Scrum				Volátil		Ícones	Treinado	
RMM				Volátil		Mídia	Casual	O.O.
OOHDM				Volátil		Mídia	Casual	O.O.
WebML				Volátil		Mídia	casual	O.O.

Uma vez identificados os requisitos dos processos de desenvolvimento de um sistema de informação, a tabela 6 identifica os requisitos dos sistemas de informações, igualando as cores da tabela 5, onde são aprestadas as coincidência de requisitos. A cor branca apresenta os sistemas de informação Científicos e CAE/CAD/CAM/CASE que possuem em comum os requisitos de volatilidade progressivos e usuários experientes. A cor amarela define os sistemas ERP II e Automação que em comum apresentam requisitos de não volatilidade, usuários treinados e paradigma orientado a objeto. A cor cinza propõe o sistema para escritório com requisito de volatilidade, usuário treinado e paradigma orientado a objeto. E a cor roxa o sistema web, com requisitos de volatilidade, usuário casual e paradigma orientado a objeto.

Tabela 6: Igualdade de requisitos do relacionamento entre Sistema de Informação e Fatores.

	M. A	L. P	S.O	R. Volátil	T. Resp	Ergonomia	Usuário	Pdgma
Científico				Progres	SR		Esp	
ERP II				N. Volátil	Interativo	Ícones	Treinado	O.O.
Sistemas Escritório				Volátil			Treinado	O.O.
Cae/ Cad/ Cam/ Case				Progres			Esp	O.O.
Automação		Centr	Espec	N. Volátil	TR		Treinado	O.O.
Sistema Web		Distri	Browser	Volátil		Mídia	Casual	O.O.

A tabela 7 mostra o relacionamento das coincidências de cores e fatores entre as tabelas 5 e 6. Os sistemas de informação ERP II e automação possuem a indicação do processo em cascata, pois ambos possuem requisitos de não volatilidade, ergonomia de ícones e usuários treinados. O Sistema de Escritório pode ser desenvolvido por diversos tipos de processos de desenvolvimento, como o Espiral, Evolucionário, XP e *Scrum*, todos possuem requisitos de volatilidade, usuários treinados e paradigma orientado a objeto. Assim como os Sistemas *Web*, podem ser desenvolvidos pelas metodologias hipermídia, RMM, OOHD, WebML que possuem fatores de volatilidade, usuários casuais e paradigma orientada a objeto.

Tabela 7: Proposta de relacionamento entre Sistemas de Informação *Web* versus Processos de Desenvolvimento *Web*.

	Científico	ERP II	Sist. de Escritório	Case	Automação	Sistema Web
Cascata		X			X	
Espiral			X			
Evolucionário			X			
XP			X			
Scrum			X			
RMM						X
OOHDM						X
WebML						X

Apenas os sistemas Científicos e as ferramentas de Engenharia, representada pelo CASE, não apresentam relacionamento entre os processos de desenvolvimento com os fatores escolhidos para este trabalho.

Na continuidade deste trabalho, será apresentado o desenvolvimento o Sistema Questionário em duas arquiteturas, uma para interface gráfica e outra Web. O sistema de interface gráfica será desenvolvido com o processo Espiral, pois este sistema é de uma aplicação com características semelhantes à de um sistema para escritório. A escolha desse modelo teve dois motivos, o primeiro é o relacionamento apresentado na tabela 7 que foi estudado durante este trabalho. O segundo motivo está relacionado com o conhecimento já obtido, durante a disciplina de Engenharia de Software, do processo Espiral. Já para o desenvolvimento do sistema em arquitetura Web, a metodologia escolhida foi a OOHDM. A escolha dessa modelagem também foi influenciada pelos estudos que realizados até a demonstração da tabela 7. Entre os modelos RMM, OOHDM e WebML, foi levada em consideração a simplicidade de aplicação da metodologia no desenvolvimento do sistema.

5. Trabalhos Correlatos

Neste capítulo são apresentados alguns trabalhos relacionados à migração de um sistema de informações na estrutura GUI para estrutura Web. Esses trabalhos não apresentam os esforços obtidos na migração do sistema informação, eles tratam apenas questões como linguagem, re-engenharia, novas ferramentas e pontos de função.

Apesar da tecnologia web se desenvolver de forma rápida, não foram encontrados trabalhos que medissem o esforço de migração de sistemas de informação para *web*. Dentre os trabalhos pesquisados, os assuntos abordados sobre a migração de sistemas de interface gráfica para *web* foram: a escolha da melhor linguagem, aplicação do ponto de função em sistemas web e reengenharia. Nas próximas seções serão apresentados esses trabalhos.

5.1 Migração da plataforma COBOL para Web

Franca e Silva (2007) propuseram a migração de um sistema de informação de plataforma COBOL para um ambiente Web. O ERP/SIG é um sistema comercial que faz a integração de dados em módulos como: emissão de notas fiscais, faturas, controle financeiro entre outros. Dentre os objetivos está à avaliação de uma proposta de integração de uma base de dados no formato ISAM (Método de Acesso Seqüencial e Indexado) através da construção de *middlewares*.

O principal objetivo da migração da plataforma COBOL para *web* é o desenvolvimento de um novo sistema com a utilização de um sistema gerenciador de banco de dados com visão integral do sistema e que mantenha todas as funcionalidades já existentes, em uma nova arquitetura que permita atualizações mais rápidas e seguras (FRANCA; SILVA, 2007).

As linguagens de desenvolvimento web estudadas pela equipe responsável pelo ERP/SIG foram *Python*, Java e Ruby. A escolha foi feita por *Python* por possuir características como: linguagem naturalmente orientada a objeto é multi-plataforma, roda nos mais utilizados sistemas operacionais, entre eles Palm, *Handhelds* e celulares da Nokia. Além de possuir licença compatível com GPL (*General Public License*) e seus direitos são de propriedade da *Python Software Foundation*. O

banco de dados escolhido foi o MySQL, pela experiência da equipe com esse SGBD (Sistema de Gerenciamento de Banco de Dados) (FRANCA; SILVA, 2007).

Estima-se que o tempo de migração total do produto seja entorno de 2 a 3 anos. Então o planejamento é de uma migração gradativa, pois minimizará o impacto da transição no que diz respeito ao cliente e ao ambiente operacional. Enquanto a migração do sistema não for completada, o dois sistemas ERP/SIG CORBA e ERP/SIG *Python*, deverão funcionar de forma harmoniosa, sem gerar problemas. (FRANCA; SILVA, 2007).

5.2 Aplicação de métricas de ponto de função em sistemas Web.

A proposta de Drach (2005) é estudar as características computacionais de plataforma Web que ajudem a avaliar o grau de adequação da Análise do Ponto de Função nesse ambiente. As diferenças entre um sistema convencional e aplicações Web são grandes, não apenas pela volatilidade de requisitos, mas pela existência do processo *authoring*. Esse processo corresponde às atividades de desenho, publicação de conteúdos estáticos, componentes multimídias, página HTML, folhas de estilo, animações e script para camada de apresentação. Essas atividades podem demandar um esforço considerável no total do projeto, e não é contabilizado no método de contagem tradicional.

Algumas características da Web fazem esta uma plataforma particular. Dentre elas podemos citar a forte predominância do aspecto visual, também possui uma arquitetura escalar, o fornecimento de acesso a múltiplas aplicações Web. As soluções para tipos de função de negócio disponibilizadas são cada vez mais sofisticadas, podendo ser uma coleção de páginas HTML estáticas ou aplicações online complexas, como plataformas de desenvolvimento J2EE e *dotNet* por exemplo, além das distintas tecnologias existentes como HTML, XML, Java entre outras. Outras características relevantes dizem respeito ao produto como: Confiabilidade, Usabilidade, Segurança, Disponibilidade, Escalabilidade, e Manutenibilidade (DRACH,2005).

Drach (2005) apresenta três métricas: o *Web Point* e *Web Objects*, que são voltadas para aplicações Web e o Pontos por Caso de Uso, que apesar de não ser exclusiva da Web, traz resultados satisfatórios. O *Web Point* baseia-se no número e complexidade das páginas HTML, e possui os seguintes passos: (1) Determinar o

número de paginas; (2) Determinar o grau de complexidade de cada pagina; e(3) Determinar o numero de *Web Point*. Esse método de contagem pode ser utilizado em conjunto com dados de produtividade e assim determinar o esforço de desenvolvimento ou melhorias na análise de requisitos, projeto, construção teste e implementação de paginas HTML.

O método *Web Objectcs* estende a contagem de Ponto de Função tradicional e acrescenta quatro grupos de elementos funcionais (*Links*, Componentes multimídia, *Scripts* e Blocos de construção). Possui os seguintes passos: (1) identificar e contar o número de elementos funcionais; (2) identificar e contar o número de operandos; (3) identificar e contar o numero de operações únicas executadas sobre os operadores identificados; (4) Aplicar peso correspondente conforme os grupos (Componentes multimídias, *Scripts*, bloco de construção e *Links*); e (5) Calcular o total de *Web Object*. O numero de *Web Objects* pode ser utilizado em equações do modelo paramétrico de estimativa WEBMO para o cálculo de esforço e duração do projeto (DRACH,2005).

O método de Pontos de Caso de uso não é uma contagem própria para sistemas web, mas obtém um resultado satisfatório em termos de erros relativos. Possui os seguintes passos para obtenção da estimativa: (1) Contar atores; (2) Contar Caso de uso; (3) Calcular o peso total Não ajustado; (4) Calcular o fator de ajuste; e (5) Calcular o Ponto de Caso de Uso Ajustado. Este método possui algumas desvantagens, a principal é a inexistência de padrões universais para a construção do caso de uso (DRACH,2005).

Os métodos do IFPUG FPA, Pontos de Caso de Uso e *Web Points* foram aplicados em um sistema de plataforma web. Esses métodos foram escolhidos pela facilidade de aplicação. Com os resultados obtidos dos métodos foi possível realizar uma comparação entre os métodos e o caso real. Chegou-se ao seguinte resultado: o *Web Point* não pé um método adequado para aplicações Web complexas, com funcionalidades além de paginas HTML de conteúdos estático, pois registrou uma discrepância de 63% entre o valor obtido com esse método e o caso real. O método FPA apresentou uma diferença de 14% entre este método e caso real, ainda é considerada uma faixa aceitável. A métrica do Ponto de Função por Caso de Usos apresentou-se inferior ao método PFA, como uma diferença de 33%. Assim a estimativa deste método não pode ser descartada, principalmente pela facilidade de

utilização. Contudo o método PFA mostrou-se suficiente para a contagem da maioria das funções presentes em um sistema Web. (DRACH, 2005).

5.3 Migração de aplicações WIMP para ambiente Web

Martins (2003) propôs uma metodologia para migrar sistemas legados com características de um sistema WIMP (*Windows, Icon, Menu, Pointing device*) para a plataforma web. O processo proposto para a migração possui duas estratégias: a elaboração de modelos de classes conceituais da aplicação e o tratamento dado a interface do usuário. O processo é baseado em técnicas e métodos de engenharia reversa, que tem o objetivo de obter abstrações por meio de análise estáticas e dinâmicas da aplicação.

A Reengenharia de software é a análise e alteração do sistema para reconstruí-lo em uma nova forma. Ela envolve duas etapas, a engenharia reversa e a engenharia avante, mas também pode incluir a reestruturação, reprojetar e a reimplementação. Engenharia reversa possui duas características, a redocumentação e recuperação do projeto. Ou seja, é um processo que analisa um sistema existente, identifica seus componentes e representa-os em um nível de abstração mais alto (MARTINS, 2003).

O processo proposto por Martins (2003) engloba duas atividades principais: o conhecimento do sistema original (engenharia reversa) e a elaboração de modelos de projeto (reprojeto), para depois implementar o novo sistema (engenharia avante). O processo de migração possui quatro atividades distintas: Compreender, Reprojetar, Reimplantar e Implantar. Este processo não é completo para a migração de um sistema, mas ajuda o desenvolvedor na construção dos modelos de requisitos funcionais e no projeto de interfaces. A engenharia avante não faz parte da proposta do autor, pois varias ferramentas existentes e também métodos podem auxiliar a realizar essa tarefa.

Em uma visão geral, o processo de migração é definido em quatro atividades: (1) Recuperar abstrações; (2) Construir casos de uso para Reengenharia; (3) Gerar modelo conceitual de Sistemas; e (4) Mapear interfaces WIMP para Web. O sistema utilizado como caso de uso, é uma aplicação cliente/servidor, implementada em Delphi e acessa tabelas locais no formato Paradox (MARTINS, 2003).

Após o desenvolvimento da migração do sistema legado para web, utilizando o processo de reengenharia proposto, é visto que o protótipo atingiu seus objetivos. Principalmente em relação a migração de interfaces que provou ser possível quanto a geração automática de código HTML, já os objetos visuais da interface do tipo WIMP são mapeados diretamente no código de paginas Web (MARTINS, 2003)

6. Estudo de Caso

Como já citado anteriormente neste trabalho, o estudo de caso foi o desenvolvimento um sistema de informação para o projeto de Extensão ENVOLVA-SE, coordenado pela Professora Jurema Iara Reis Belli do Departamento de Ciências Básicas Sociais, da UDESC campus CCT.

Essa aplicação é um questionário, e foi desenvolvida em duas metodologias distintas, a primeira com estrutura de interface gráfica com usuário (GUI - *Graphical user interface*) e a outra para tecnologia web com JSF (*JavaServer Faces*). O sistema Questionário será desenvolvido conforme o anexo B, apenas até a inserção de dados fornecida pelo usuário no banco de dados, devido ao prazo estabelecido a este trabalho.

A grande maioria dos sistemas de informação é desenvolvida com estrutura GUI, porém a tecnologia web está crescendo e solicita que os tradicionais sistemas de informação migrem para essa nova realidade. Este trabalho irá disponibilizar informações de comparação entre essas estruturas através do ponto de função e o esforço (hora-homem), e ajudará as empresas do ramo a tomar decisões sobre essa migração.

6.1 Projeto de Extensão “ENVOLVA-SE”

O Projeto de Extensão Envolve-se, tem como objetivo beneficiar a comunidade acadêmica com cultura, lazer e socialização. Hoje uma das suas principais atividades é o “Dia da Cultura”, um dia com apresentações de dança, teatro, corais, filmes e outras atrações, o projeto também promove aulas de teatro em parceria com a Casa da Cultura de Joinville. Com a intenção de obter um perfil da comunidade acadêmica, e melhor atender as necessidades desse público, o projeto Envolve-se desenvolveu uma pesquisa com questões econômicas e culturais.

O questionário é composto por dezoito (18) perguntas de múltipla escolha. Elas são divididas em três partes, moradia, família e vida acadêmica. Este se encontra em formato impresso, no Anexo C é possível encontrar uma cópia. Atualmente muitos acadêmicos não têm acesso a ele, assim não podem colaborar

com a pesquisa. Com a disposição do questionário via *web*, estima-se que é possível atingir uma amostra maior de acadêmicos e ter resultados mais precisos.

6.2 Pontos de Função

O propósito da contagem de pontos de função é fornecer uma resposta a um problema de negocio. Dessa forma o propósito pode ser: determinar algumas premissas para o processo de contagem; determinar o tipo de contagem: projeto de desenvolvimento, projeto de melhoria ou aplicação; determinar o escopo da contagem, que pode abranger parte, uma ou mais aplicações; afeta o posicionamento da fronteira da aplicação; e determina o nível do detalhe da contagem. O escopo de processo de contagem e suas relações de dependências são mostrados na figura 12 (VAZQUEZ; SIMÕES; ALBERT, 2003).

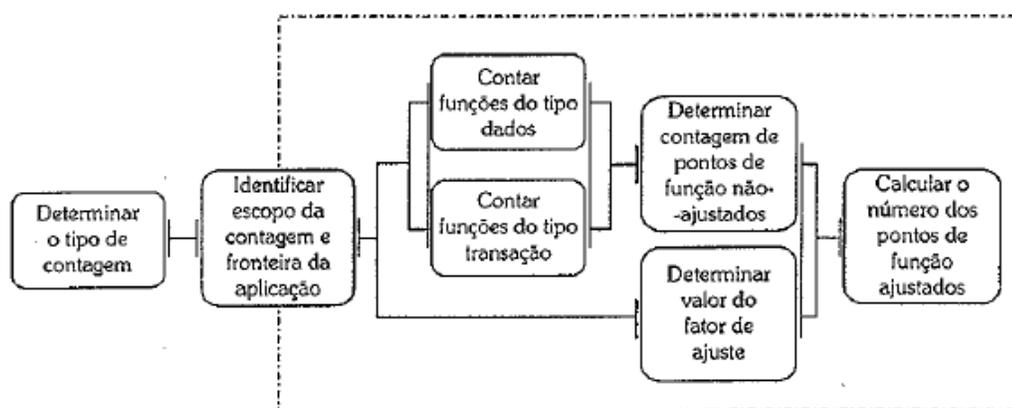


Figura 12: Processo de Contagem de pontos de Função adaptado de (VAZQUEZ; SIMÕES; ALBERT, 2003).

Os requisitos funcionais são a base para o cálculo dos pontos de função não ajustados, e alguns requisitos não funcionais são integrantes das características gerais do sistema utilizadas na fase de determinação do fator de ajuste utilizada no cálculo do número de pontos de função ajustados (VAZQUEZ; SIMÕES; ALBERT, 2003).

Os requisitos necessários para a contagem dos pontos de função do sistema QUESTIONÁRIO é encontrado no Anexo B. Como base nessa documentação é apresentada e aplicada as sete (7) etapas da contagem nas próximas seções.

6.2.1 Tipo de Contagem

A primeira etapa da contagem do ponto de função é definir o tipo de contagem, que pode ser de três tipos: contagem de projeto de desenvolvimento, contagem de um projeto de melhorias ou contagem de uma aplicação.

A contagem de projetos de desenvolvimento mede as funcionalidades fornecidas aos usuários pela primeira instalação da aplicação. A contagem de projetos de melhoria mede a modificação de projetos já existentes. E a contagem de aplicação mede uma aplicação já desenvolvida, com o objetivo de fornecer um valor para as funcionalidades disponíveis (VAZQUEZ; SIMÕES; ALBERT, 2003).

Para o sistema Questionário do projeto Envolve-se a contagem será a de um projeto de desenvolvimento, pois trata-se do desenvolvimento de um novo sistema.

6.2.2 Escopo da Contagem e Fronteira da Aplicação

A segunda etapa refere-se à identificação do escopo da contagem e da fronteira da Aplicação. O escopo da contagem é determinado pelo objetivo da contagem, ou seja, é ela quem vai especificar o que a contagem vai abranger. O escopo pode variar desde todas as funcionalidades disponíveis; apenas funcionalidades utilizadas pelo usuário; ou ainda em funcionalidades específicas (VAZQUEZ; SIMÕES; ALBERT, 2003).

Já a fronteira da aplicação é o que divide o software que será medido da interação do usuário. Esta fronteira pode ser também uma aplicação que interage com o sistema. A identificação da fronteira de aplicação é dada a partir de três regras: a fronteira é definida a partir da visão do usuário, a fronteira entre aplicações devem ser definidas com base nas regras do negócio e não em considerações técnicas, e por fim, a fronteira inicial já estabelecida para uma aplicação não deve ser baseada pelo escopo de contagem (VAZQUEZ; SIMÕES; ALBERT, 2003).

A definição desta etapa para o sistema Questionário do projeto Envolve-se segue da seguinte forma: o escopo da contagem será de todas as funcionalidades do sistema e a fronteira será a interface de comunicação com o usuário, visto que é um sistema pequeno e não interage com outro tipo de fronteira.

6.2.3 Função do Tipo de Dado

As funções do tipo de dados representam as funções dos usuários, ou seja, os dados fornecidos a ele que podem ser internos ou externos, classificados com ALI (Arquivo Lógico Interno) e AIE (Arquivo de Interface Externa) (VAZQUEZ; SIMÕES; ALBERT, 2003).

O ALI representa os dados disponibilizados pelo usuário que são mantidos dentro do sistema a ser contado, como por exemplo, um banco de dados utilizados pela aplicação. E o AEI representa os dados que são mantidos fora do sistema que será contado, como um banco de dados utilizados por outra aplicação (VAZQUEZ; SIMÕES; ALBERT, 2003).

Com a identificação do ALI e do AEI é possível classificar a complexidade de cada tipo de dado através da contagem dos elementos: Arquivos Referenciados (AR) e Tipo de Dados (TD). O AR são ALI's ou AIE's, em um subgrupo único de elementos de dados, não repetidos e reconhecidos pelo usuário. E o TD é um campo único não repetido e reconhecido pelo usuário (ROCHA, 2007).

Após a contagem do AR e TD, é possível encontrar a complexidade funcional na tabela 8, que classifica as funções de Tipo de Dados em Baixa, Média e Alta (ROCHA, 2007).

Tabela 8: Tabela de Complexidade do Tipo de Dado (ROCHA, 2007)

AR - Arquivos Referenciados	TD - Tipo de Dados		
	1 a 19	20 a 50	51 ou mais
1	Baixa	Baixa	Média
2 a 5	Baixa	Média	Alta
Maior 5	Média	Alta	Alta

A Função de Tipo de Dados é aplicada no sistema Questionário, através da tabela 9, que descreve duas funcionalidades do sistema: Acadêmico, o usuário fornece as informações de matrícula e curso que será armazenado no banco de dados da aplicação; e, Questionário, o usuário fornece as respostas do questionário para armazenar no banco de dados da aplicação. Essas funcionalidades e os seus tipos de dados podem ser observados na camada de persistência do anexo B. Visto que as funcionalidades representam a utilização do banco de dados do sistema, ambas recebem a classificação de ALI.

Tabela 9: Contagem de funções de tipo de dados

Descrição de Funcionalidade	Tipo de Dados	AR	TD	Complexidade
Acadêmico	ALI	1	2	Baixa
Questionário	ALI	1	19	Baixa

6.2.4 Função do Tipo Transação

A quarta etapa do processo de contagem do ponto de função refere-se a contagem da função do tipo de transação. A função de tipo transação representa as funcionalidades de processamento de dados que são classificadas com: Entrada Externa (EE), Saída Externa (SE) e Consulta Externa (CE) (VAZQUEZ; SIMÕES; ALBERT, 2003).

A Entrada Externa é definida por uma informação fornecida pelo usuário ao sistema, através das ações de incluir, editar ou excluir cadastro. A Saída Externa representa as informações que serão geradas a partir do sistema para o usuário, como a geração de relatórios. E a Consulta Externa fornecem informações cadastrais armazenadas no banco de dados, através de consulta de cadastros (ROCHA, 2007).

Para cada uma das funcionalidades do tipo de função de transação existe a identificação de complexidade. Na tabela 10 é apresentada a complexidade do elemento EE e na tabela 11 é apresentada a complexidade dos elementos SE e CE (ROCHA, 2007).

Tabela 10: Tabela de Complexidade para EE (ROCHA, 2007)

AR / TD	1 a 4	5 a 15	16 ou mais
0 a 1	Baixa	Baixa	Média
2	Baixa	Média	Alta
Maior 2	Média	Alta	Alta

Tabela 11: Tabela de Complexidade para SE e CE (ROCHA, 2007)

AR / TD	1 a 5	6 a 19	20 ou mais
0 a 1	Baixa	Baixa	Média
2 a 3	Baixa	Média	Alta
Maior 3	Média	Alta	Alta

Com a apresentação das tabelas de complexidades dos elementos da função de tipo de transação é possível completar a contagem das funções de tipo de dados e de transação do sistema Questionário. Na tabela 12 é apresentada a inclusão da coluna de tipo de transação e a funcionalidade Registrar Dados do Acadêmico e Registrar Resposta do Questionário, que pode ser observadas no caso de uso da documentação de requisitos do Anexo B. Estas novas funcionalidades descrevem a inclusão dos dados através das respostas fornecidas pelo usuário, e ambas possuem classificação do tipo de transação como EE e complexidade Baixa e Média respectivamente. Outros tipos de transação não foram registrados, pois conforme os requisitos apresentados no anexo B, o sistema compreende apenas a inclusão de dados, ou seja, não possui edição e consultas.

Tabela 12: Contagem de funções de tipo de dados e transação

Descrição de Funcionalidade	Tipo de Dados	Tipo de Transação	AR	TD	Complexidade
Acadêmico	ALI		1	2	Baixa
Questionário	ALI		1	19	Baixa
Registrar Dados do Acadêmico		EE	1	2	Baixa
Registrar Resposta do Questionário		EE	1	19	Média

6.2.5 Contagem dos Pontos de Função Não-Ajustados (UFP)

A contagem do ponto de função não-ajustado é baseada no grau de complexidade dos tipos de funções de acordo com o grau de complexidade. Na tabela 13 é apresentada a conversão do grau de complexidade em pontos de função não ajustados.

Tabela 13: Tabela de Conversão de Complexidade x UFP. Baseado em (ROCHA, 2007)

Função	Complexidade		
	Simples	Média	Alta
ALI	7	10	15
AIE	5	7	10
EE	3	4	6
SE	4	5	7
CE	3	4	6

Após a identificação dos tipos de dados, complexidade e do ponto dos pontos de função, é possível completar a tabela de pontos de função não ajustados do sistema Questionário. Na tabela 14 são apresentados os elementos relevantes para a contagem total de ponto de função não ajustado, que possui um total de 21 pontos de função.

Tabela 14: Dados para contagem de função não ajustadas

Descrição de Funcionalidade	Tipo de Dados	AR	TD	Complexidade	Ponto de Função
Acadêmico	ALI	1	2	Baixa	7 PF
Questionário	ALI	1	19	Baixa	7 PF
Registrar Dados do Acadêmico	EE	1	2	Baixa	3 PF
Registrar Resposta do Questionário	EE	1	19	Média	4 PF
Total UFP					21 PF

6.2.6 Valor do Fator de Ajuste

A sexta etapa do processo de contagem do ponto de função é determinar o valor do fator de ajuste. Essa etapa tem o objetivo de ajustar os pontos de função não ajustadas em mais ou menos (+/-) 35%, através da fórmula $VAF = (TDI \times 0,01) + 0,65$, onde VAF (*Value Adjustment Factor*) é baseado nas características gerais do sistemas e TDI (*Total Degree of Influence*) é a somatória dos níveis de influencia das características gerais (VAZQUEZ; SIMÕES; ALBERT, 2003).

O TDI é o nível de influência de 14 características gerais do sistema, para cada uma das influencias são atribuídos os seguintes valores: 0– Nenhuma influencia; 1- Influencia mínima; 2- Influencia moderada; 3- Influencia média; 4- Influencia significativa; e 5- Grande Influencia (ROCHA, 2007). A atribuição dos valores de influencia se dá através das características técnicas do local de desenvolvimento do sistema.

O sistema Questionário do projeto Envolve-se, foi desenvolvido no laboratório BDES (Grupo de Banco de Dados e Engenharia de Software) da UDESC-Joinville. O laboratório é ligado a rede desta universidade, que possui uma demanda aproximada de mil e quinhentos (1.500) computadores. A maquina utilizada no desenvolvido do sistema possui as seguintes características: Fabricante Dell,

modelo Vostro 220s, processador Intel Core 2Duo CPU 2.66GHz, memória RAM de 4,00 GB, sistema operacional de 32 bits (*Windows Vista – Home Basic*).

Após conhecer o cenário de desenvolvimento, é possível atribuir as influencias nas 14 características:

1- Comunicação de dados	:2	8- Atualização on-line	:0
2- Processamento distribuído	:0	9- Processamento Complexo	:0
3- Performance	:1	10- Reutilização de código	:0
4- Utilização de Equipamento	:0	11- Facilidade de implementação	:3
5- Volume de transação	:2	12- Facilidade operacional	:3
6- Entrada de dados on-line	:5	13- Múltiplos locais	:0
7- Eficiência do usuário final	:0	14- Facilidade de mudanças	:1

O resultado do TDI é 17, que representa o grau de influencia. Então o resultado do valor do fator de ajuste é: $VFA = (TGI \times 0,01) + 0,65 = 0,82$.

6.2.7 Calculo do Número dos Pontos de Função Ajustado

A última etapa dessa contagem de ponto de função é o calculo dos pontos de função ajustados, que possui a seguinte formula: $FP = UFP \times VAF$ (VAZQUEZ; SIMÕES; ALBERT, 2003).

Com a multiplicação entre o ponto de função não ajustado e o valor do fator de ajuste é chegado no valor de 17,22 pontos de função para o sistema Questionário do projeto Envolve-se.

6.3 Resultado do Ponto de Função

Com base no valor do Ponto de Função ajustado é possível fazer uma estimativa de quantidade de horas em média que será necessário para desenvolver o sistema. Para isso é necessário fazer uma estimativa de produtividade do desenvolvimento do sistema.

O desenvolvimento deste sistema será na linguagem Java, tanto para *desktop* quanto para *web*. O grupo BFPUG (2009) apresenta uma produtividade em média de 15 a 20 horas por ponto de função para a linguagem Java. Considerando a

inexperiência do desenvolvedor deste sistema, será estimado uma produtividade de 20 H/PF. Assim tem-se a tabela 15 que apresenta a estimativa de trabalho para o sistema Questionário.

Tabela 15: Estimativa de horas de trabalho

Produtividade Equipe (Java)	20 H por PF
Prazo: 17,22 * Produtividade	344:40 Horas

O resultado apresentado na tabela 15 refere-se ao desenvolvimento completo de um ciclo de vida de um sistema. Com essa estimativa é possível prever que o sistema questionário poderá ser desenvolvido em aproximadamente 45 dias, trabalhando oito (8) horas diárias. Porém neste trabalho será realizado apenas a atividade de implementação do sistema Questionário, isso ocorrerá devido ao tempo estimado para conclusão deste trabalho. Nas próximas seções serão apresentadas os resultados obtidos na implementação do sistema.

6.4 Sistemas de Informação de Interface Gráfica com o Usuário (GUI)

O sistema Questionário foi desenvolvido com estrutura GUI na linguagem Java utilizando componentes *Swing*, e como ferramenta de desenvolvimento foi utilizado o *NetBeans* 6.5.1, linguagem e ferramenta de domínio do programador. Durante o desenvolvimento deste sistema foram realizadas medições de esforço hora-homem e nesta seção serão apresentados os resultados obtidos.

O Sistema Questionário com estrutura GUI foi desenvolvido através do processo espiral, estudado no capítulo 2 deste trabalho. Este processo apresenta seis (6) etapas, sendo a primeira a de formulação, que identifica as metas e objetivos da aplicação. A segunda etapa é a de planejamento, nela é realizado o levantamento de custo para o desenvolvimento do projeto. A terceira diz respeito a análise de requisitos técnicos e gráficos. A quarta etapa trata da engenharia de estrutura, navegação e interface do projeto. A quinta etapa é a geração do código e por fim, a sexta etapa é a avaliação do cliente. Todas as etapas são apresentadas no Anexo D deste trabalho.

As medições de esforço hora-homem realizadas durante o desenvolvimento do sistema Questionário tratam-se apenas da fase de geração de código. As medições foram realizadas em forma de tabela, nela foi registrada a data, hora inicial

e hora final do período de desenvolvimento. Ao final do desenvolvimento foram somadas as horas de trabalho e chegou-se ao resultado de 25:56 horas de desenvolvimento.

6.5 Sistema de Informação WEB

O sistema Questionário do projeto Envolve-se foi desenvolvido também na plataforma web, utilizando a linguagem JSF (*JavaServe Faces*) e como ferramenta de desenvolvimento foi utilizado o *NetBeans* 6.5.1. Durante o desenvolvimento de sistema também foram coletadas informações de esforço hora-homem e o resultado será apresentado nesta seção.

Para o desenvolvimento do sistema Questionário foi utilizando a metodologia de desenvolvimento OOADM como processo de desenvolvimento, já estudado no capítulo 2 deste trabalho. Esta metodologia é dividida em quatro partes, a primeira parte é a modelagem conceitual que define as classes e objetos do sistema. A segunda parte é projeto navegacional que define a forma de navegação do usuário no sistema. A terceira parte diz respeito ao projeto de interface, que é a construção de um protótipo da interface do sistema, nela será definida os componentes utilizados e a disposição deles. Por fim, a quarta parte trata da implementação do sistema. Assim como no sistema de estrutura GUI a medição de esforço hora-homem foi realizado apenas na etapa de implementação. Todas as etapas desse modelo de desenvolvimento poderão ser observadas no anexo E deste trabalho.

A linguagem JSF foi escolhida pelo fato de fazer parte da tecnologia Java, e possuir o mesmo paradigma Orientado a Objeto para realizar a comparação entre o desenvolvimento do sistema, porém não é uma linguagem de domínio do programador. Essa tecnologia incorpora características do *framework* MVC (*Model-View-Controller*) para Web e de um modelo de interfaces gráficas baseada em eventos. O padrão MVC tem o objetivo de dividir a aplicação em três camadas: modelo, visualização e controle. O modelo é responsável por representar os objetos de negócio. A visualização diz respeito às interfaces com o usuário. E a camada de controle faz a ligação entre o modelo e a visualização (PITANGA, 2004).

Assim como no sistema de estrutura GUI, as medições do sistema em estrutura web também foram realizadas em forma de tabela, nela foi registrada a data, hora inicial e hora final do período de desenvolvimento. Ao final do

desenvolvimento foram somadas às horas de trabalho e chegou-se ao resultado de 31:24 horas de desenvolvimento

6.6 Resultados

Conforme descrito nas seções anteriores, as medições de implementação do sistema Questionário em estrutura GUI e Web, foram realizadas em forma de tabela. Nela foi anotada a data, hora inicial e final do período de desenvolvimento. Com as somatórias dessas horas foram obtido os seguintes resultados: 25:56 horas de implementação para o sistema com estrutura GUI e 31:24 horas de implementação de desenvolvimento em estrutura Web. Esses dados podem ser observados no figura 13.

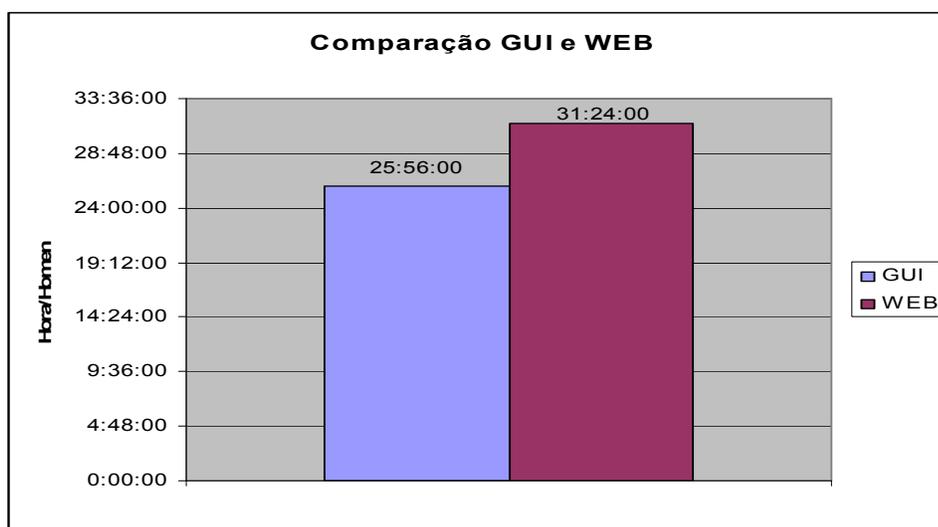


Figura 13 – Comparação de esforço hora-homem

Na figura 13 é perceber a diferença de 6 horas e 32 minutos de implementação entre a estrutura GUI e Web. Esta diferença representa 13,3 % a mais na implementação do sistema Web.

7. Conclusão

O advento dos computadores possibilitou a automação dos sistemas de informação. Estes sistemas variam de acordo com sua aplicabilidade. Há sistemas usados exclusivamente para resolver problemas científicos. Outros são usados como ferramentas de produtividade das pessoas, como os sistemas CAE/CAD/CAM e outros são usados para administração de empresas. Um dos objetivos deste trabalho é justamente apresentar uma taxonomia de tais sistemas e isso foi feito.

Obviamente estes sistemas precisam ser desenvolvidos. Para desenvolvê-los é preciso haver processos onde cada passo da construção de tais sistemas é definida e detalhada. A bibliografia apresenta muitos processos. Estes processos foram descritos neste trabalho. A questão é qual processo é mais adequado para o desenvolvimento de determinado sistema? Esta pergunta também foi respondida neste trabalho.

A grande dificuldade de chegar a esta conclusão se deu em função dos sistemas Web. Há sistemas que surgiram em função da tecnologia Web, como os B2B. Outros, anteriores, atualmente estão usando esta tecnologia. Na verdade são sistemas tradicionais baseados em Web. De qualquer maneira esta dificuldade foi contornada e chegou-se a uma conclusão de qual sistema seja ele “Web puro” ou baseado em Web, deve usar determinado processo de desenvolvimento.

Após a identificação dos processos de desenvolvimento para os sistemas GUI e Web, foram pesquisados alguns trabalhos correlatos. O mais interessante se deu ao fato de não se encontrar trabalhos que comparassem os esforços de desenvolvimento entre tecnologias distintas (GUI e Web). Os trabalhos apresentavam apenas dados de sucesso de migração de plataforma, a utilização de métricas de ponto de função em sistemas web, e o trabalho mais próximo foi à proposta de um processo de reengenharia de um sistema WIMP para plataforma Web.

Por fim foi realizado o estudo de caso, onde foi desenvolvido o sistema “Questionário” do Projeto ENVOLVA-SE. Tal sistema foi desenvolvido através de dois processos diferentes para estruturas diferentes: o sistema GUI foi desenvolvido com o auxílio do processo Espiral na linguagem Java utilizando componentes Swings e sistema Web teve o auxílio da metodologia hipermídia OOHDm na

linguagem JSF para web. Em ambos os desenvolvimentos foram coletados dados de esforço hora-homem durante a implementação, que chegou ao resultado de 25:56 horas para o sistema GUI e 31:24 horas para o sistema Web. Ou seja, a implementação de um sistema web consome 13,3% a mais de recurso.

Como o objetivo deste trabalho é medir o consumo de esforço na migração de um sistema GUI para Web, podemos observar nesse resultado que existe um investimento na geração de código a ser feito para essa migração. Este investimento representa 13,3 % a mais, comparando com o esforço de implementação de um sistema GUI.

É importante ressaltar que até o instante de desenvolver do sistema web, a tecnologia JSF era desconhecida para o programador, ou seja, o maior investimento para este trabalho trata-se do tempo gasto para o desenvolvedor aprender-se a nova linguagem. No caso de um programador com experiência na linguagem escolhida, o investimento tende a ser menor.

Um fator negativo se deu durante a implementação do sistema web, trata-se da interface com o usuário. Devido a falta de experiência do programador com a linguagem JSF, não foi possível utilizar o componente *jComboBox* como no sistema GUI para facilitar as respostas dos usuários. Dessa forma houve uma mudança de *layout* na interface do usuário, deixando de usar respostas prontas para digitar sua resposta em um *jLabel*. Por consequência, a busca de dados no banco, realizada pelo coordenador também fica comprometida.

Para trabalhos futuros sugerimos que seja desenvolvido novo relacionamento entre sistemas de informação e processos de informação. Porém com novos fatores para assim poder caracterizar o desenvolvimento dos sistemas científicos e de automação. Estes pedem novos fatores pela sua complexidade de processamento. Ainda como sugestão de trabalhos futuros, coletar dados de esforço hora-homem em um processo de reengenharia.

Referências

ALVARES, P. *WebPraxis - Um processo personalizado para projetos de desenvolvimento para Web*. Belo Horizonte, Brazil: Dissertação de curso de Pós Graduação em Ciência da Computação, 2001. Acessado em 08/11/2007. Disponível em: <<http://www.wppf.uaivip.com.br/pesquisa/DissertacaoPatricia.pdf>>

BFPUG (Brazilian Function Point Users Group). *Qual a produtividade Java?*. Acessado em 01/06/2009 . Disponível em: < <http://www.bfpug.com.br/> >

CECHELERO, D.; VOLPI, M. M. *Engenharia para aplicação web*. Revista de divulgação técnico-científica do ICPG, v. 2, n. 5, p. 49-54, abr-jun 2004. Acesso em 08/09/2007. Disponível em: <<http://www.icpg.com.br/hp/revista/index.php?rpauto=7>>.

CERI, S.; FRATERNALI, P.; PARABOSCHI, S. *Data-driven, one-to-one web site generation for data-intensive applications*. In: The VLDB Journal. [s.n.], 1999. p.615-626. Acessado em: 07/07/2009. Disponível em: <citeseer.ist.psu.edu/ceri99datadriven.html>.

COSTAGLIOLA, G.; FERRUCCI, F.; FRANCESE, R. *Web engineering: Models and methodologies for the design of hypermedia applications*. In: CHANG, S. K. (Ed.). *Hand-book of Software Engineering & Knowledge Engineering*. Italy: Emerging Technologies, 2002. v. 2, p. 181–199. Acessado em: 07/07/2009 Disponível em: <<ftp://cs.pitt.edu/chang/handbook/58b.pdf>>.

DRACH, Marcos D. *Aplicação de Métricas por Ponto de Função em sistemas baseados na Web*. Trabalho para título de mestrado. Universidade Estadual de Campinas-UNICAMP, 2005. Acessado em 01/06/2009. Disponível em: <<http://libdigi.unicamp.br/document/?code=vtls000359854>>

FALBO, R. A. *A experiência na definição de um processo padrão baseado no processo unificado*. Anais do II Simpósio Internacional de Melhoria de Processo de Software, SIMPROS'2000, pp. 63-74. São Paulo, Brasil, Setembro 2000. Disponível em <<http://www.inf.ufes.br/~falbo/download/pub/Simpros2000.pdf>>. Acessado em 07/11/2008.

FRANCA, Luis F.M.; SILVA, Leonardo L. *Uma Proposta de integração de sistemas para um ambiente de migração do ERP/SIG da plataforma COBOL para uma solução em ambiente WEB*. Centro Universitário do Belo Horizonte. Trabalho de conclusão de curso 2007. Disponível em <<http://www.leandroliberio.com.br/migracaocobol.pdf>>. Acessado em 10/07/2009

FRANCA, Luiz P.A.; STAA, Arndt Von; LUCENA, Carlos J. P. *Medição de software para pequenas empresas: uma solução baseada na Web*. PUC-RIO.MCC 32/98, outubro 1998. Disponível em: <ftp://ftp.inf.puc-rio.br/pub/docs/techreports/98_32_franca.pdf> Acessado em 08/06/2009.

GABRIELI, Leandro V. et al. *Uso de um Framework no Desenvolvimento de Sistemas de Informação Web*. XXVII Encontro Nacional de Engenharia de Produção, outubro 2007. Acessado em 07/03/2008. Disponível em <http://w3box.pro.br/arquivos/ENEGEP2007_TR640475_0336.pdf>

GUEDES, R. M.; SILVA, E. M. *Introdução ao uso do Linux*. Universidade do Estado do Rio de Janeiro. v. 4, jun-2006. Acessado em: 31/05/2008. Disponível em <<http://www.lee.eng.uerj.br/~elaine/introducao-ao-uso-do-linux.pdf>>.

GINIGE, A.; MURUGESAN, S. The Essence of Web Engineering: Managing the Diversity and Complexity of Web Application Development. IEEE MultiMedia, v. 8, n. 2, abr./jun. 2001. 22-25 p. Acessado em 05/04/2008. Disponível em: <<http://ieeexploreieee.org/Xplore/login.jsp?url=/iel5/93/19845/00917968.pdf?temp=x>>.

KAPPEL, A. M. *Uma abordagem para o gerenciamento do projeto de sistema de informação na web com características sazonais*. Porto Alegre. Escola de Engenharia Programa de Pós Graduação em Engenharia de Produção, 2006. Acessado em 08/09/2007. Disponível em: <<http://producao.ufrgs.br/arquivos/publicacoes/AlexandreKappel.pdf>>

LAUDON, Kenneth C.; LAUDON, Jane P. *Sistemas de Informação Gerenciais: administração a empresa digital*. 5. Ed. São Paulo: Pearson, Prentice Hall, 2004.

MARINHO, Fabiana. et al Uma proposta de um repositório de padrões de software integrado ao RUP. Permission is granted to copy for the SugarloafPloP 2003 Conference. Universidade Federal do Ceará. Disponível em <http://www.cin.ufpe.br/~sugarloafplop/articles/spa/spa_RepositorioPadroesSoftwareRUP.pdf>. Acessado em 10/11/2008.

MURUGESAN, S.; GINIGE, A. *Web Engineering: introduction and perspective*. 1–30 p. 2005. Acessado em 27/08/2007. Disponível em: <<http://www.idea-group.com/downloads/excerpts/SuhChapter1.pdf>>.

PITANGA, Talita. *JavaServer Faces: A mais nova tecnologia Java para desenvolvimento Web*. Tutorial do GUJ – Grupo de usuário Java. 2004. Disponível em: < <http://www.guj.com.br/content/articles/jsf/jsf.pdf> >. Acessado em: 02/06/2009.

PRESSMAN, R. S. *Engenharia de Software*. 5. ed. Rio de Janeiro: Mc Graw Hill, 2002.

Priberam Dicionário Online <http://www.priberam.pt/dlpo/definir_resultados.aspx>

ROCHA, Lidimon C. M. *Análise de Pontos de Função e sua importância para projeto de desenvolvimento de software*. Centro Universitário do Triângulo (UNITRI), 2007. Acessado em 14/04/2008. Disponível em <br.geocities.com/lidimoncristiano/ArtigoAPF.pdf >

ROSSI, G. *Um método orientado a objetos para o projeto de Aplicações Hiperfídia*. Rio de Janeiro, Brazil: Tese de Doutorado, 1996. Acessado em 08/10/2007. Disponível em:

<www.tecweb.inf.pucRio.br/oohdm/space/Publica%C3%A7%C3%B5es+sobre+o+m%C3%A9todo+propriamente+dito>

SCHONS, C. H. *O volume de Informações na internet e sua desorganização: reflexões e perspectivas*. Informação & Informação. V. 12. n.1. jan-jun 2007. Acessado em 10/06/2008. Disponível em: <<http://www.uel.br/revistas/informacao/include/getdoc.php?id=572&article=185&mode=pdf>>

SCHWABE, D. et al. Engineering Web applications for reuse. IEEE MultiMedia, v. 8, n. 1, p. 20-31, 2001. Acessado em 04/04/2008. Disponível em: <citeseer.ist.psu.edu/schwabe01engineering.html>.

SCHMITT, Rogério A.; SCHMITT Fátima A. B. S.. *Sistemas de Informação Baseados na Web*. Revista Eletrônica do CEBLUS, publicada em maio de 2007. Acessado em 01/03/2008. Disponível em: <http://www.cesblu.br/revista/artigos/Artigo_Pos_Web_IV_Rogério_Antonio_Schmitt.pdf>.

SEVERO, C. E. P. *Abordagens sobre especificação de hiperdocumentos*. REVISTA DO CCEI-Centro de Ciências da Economia e Informática, v. 5, n. 7, p. 36-42, 2001. Disponível em: <<http://www.urcamp.tche.br/ccei/revista7.pdf>>

SOARES, M. S. *Metodologias Ágeis Extreme Programming e Scrum para o desenvolvimento de software*. Universidade Presidente Antonio Carlos. Acessado em: 01/06/2008. Disponível em: <<http://www.inf.ufrgs.br/~zirbes/MaterialAula/Engenharia%20de%20SW%20N%202007/Medodologias%20de%20Desenvolvimento/Vant%20x%20Desvant%20Metodos%20Ageis.pdf>>

SOMMERVILLE, I. Engenharia de Software. 6. ed. [S.l.]: Addison Wesle, 2003.

SORDI, J.O.; MARINHO, B. L. *Integração entre Sistemas: Análise das Abordagens Praticadas pelas Corporações Brasileiras*. RBGN, v.9, n.23, p. 78-93, jan-abr. 2007. Acessado em 31/05/2008. Disponível em: <<http://200.169.97.103/seer/index.php/RBGN/article/viewFile/75/69>>.

SOUZA, C. S. et al. *Projeto de Interfaces de Usuários : Perspectivas Cognitiva e Semiótica*", Anais da Jornada de Atualização em Informática, XIX Congresso da Sociedade Brasileira de Computação. Rio de Janeiro: julho – 1999. Acessado em 09/07/2008. Disponível em: <http://www.dimap.ufrn.br/~jair/piu/JAI_Apostila.pdf>

STAIR, R. M; REYNOLDS, G. W. *Princípios de Sistemas de Informação uma abordagem gerencial*. 4.ed Rio de Janeiro: LTC, 2002.

TANENBAUM, A. S. *Sistemas Operacionais Modernos*. LTC, 1995.

TURINE, M. A. S; MASIERO, P. C. *Especificação de requisitos: uma introdução*. Instituto de Ciências Matemáticas de São Carlos. São Paulo, 1996. Acessado em 05/07/2008. Disponível em: <http://www2.unemat.br/rhycardo/download/engenharia_de_requisitos.pdf>.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme S.; ALBERT, Renato M.. *Análise de Pontos de Função, medição, estimativas e gerenciamento de projetos de software*. 1. ed. São Paulo: Editora Érica, 2003.

ZANETI JR, Luiz A. *Sistema de informação Baseados na Tecnologia Web: Um estudo sobre seu Desenvolvimento*. São Paulo, Brasil: Dissertação para obter titulação de Mestre em Administração – Universidade de São Paulo, 2003. Acessado em 01/03/2008. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/12/12139/tde-14082003-104928/>>.

APÊNDICES

Apêndice A – Plano de Trabalho de Conclusão de Curso

Plano de Trabalho de Conclusão de Curso

Analise do Esforço de Desenvolvimento de Aplicações GUI e WEB.

UDESC - Centro de Ciências Tecnológicas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação - Integral
Turma 2009/1 - Joinville – Santa Catarina

Janaise Francisco Sicka – janaisef@gmail.com

Orientador: Salvador A. dos Santos – salvador.santos@uol.com.br

Resumo – *A evolução da internet e a tecnologia web trouxeram grandes avanços para o desenvolvimento de sistemas de informação web. Muitas metodologias foram desenvolvidas para que os sistemas de informação possam usufruir dos benefícios oferecidos pela web. Esse trabalho tem o objetivo de caracterizar sistemas baseados na web, suas metodologias de desenvolvimento e implementação de um sistema com estrutura GUI e Web, também apresentar a comparação dos esforços de implementação ambas as estruturas.*

Palavras-chave: *sistemas de informação, metodologia de desenvolvimento web, internet, tecnologia web.*

1. Introdução e Justificativa

A Internet e a tecnologia Web foram desenvolvidas com o objetivo de divulgar trabalhos científicos, aos poucos elas evoluíram e modificaram a sua finalidade inicial (SCHMITT, SCHMITT; 2007). Dentre as mudanças sofridas pela tecnologia Web, Zaneti Jr (2003) diz que uma grande evolução se deu quando os usuários tiveram a permissão de interagir com a web, ou seja, deixou de funcionar como um repositório de acesso a conteúdos estáticos, passando a funcionar como uma interface de acessos a diversos sistemas de informações dinâmicos.

Surge então, uma nova geração de negócios, com atividades que envolvem a interação entre clientes e fornecedores, sendo possível aproveitar os benefícios dessa tecnologia (ZANETI JR, 2003 apud SCHMITT, SCHMITT; 2007). Essa evolução tecnológica de informação e comunicação tornou-se um fator muito importante na atual gestão de empresas atingindo suas diferentes áreas (LAUDON; LAUDON, 2000; O'BRIEN, 2003 apud GABRIELI et al, 2007).

Laudon e Laudon (2004) definem um sistema de informação como um conjunto de funções que coletam, processam, armazenam e distribuem informações, relacionando-se entre si, com o objetivo de apoiar a tomada de decisão, a coordenação e o controle de uma organização.

Um sistema de informação com de interface gráfica com o usuário (GUI – *graphical user interface*) fornece aparências e comportamentos diferenciados. O GUI dá ao usuário um nível básico de familiaridade com o sistema, assim reduz o tempo de aprendizagem e aumenta a habilidade de uso produtivo (DEITEL; DEITEL; LISBÔA, 2003). Atualmente o GUI tradicional (desktop) está sendo substituído pela *web*. Todas as vezes que a tecnologia de informática sofre um “*upgrade*”, as aplicações devem, ser ajustadas para acomodar tal “*upgrade*”. Assim, quando o terminal de vídeo foi incorporado à tecnologia, as aplicações que eram “*offline*” migraram para “*online*”. Com o advento do Sistema Operacional *Windows*, as aplicações com interface caracter, tiveram que migrar para GUI. Atualmente a nova tecnologia é a *web*. As aplicações devem então migrar de GUI para *web*.

Apesar do rápido desenvolvimento da tecnologia *web*, pouca coisa mudou em relação ao desenvolvimento de suas aplicações. Muitas organizações têm sucesso, outras apresentam falhas ou tendências a elas, e fracassam antes de alcançar suas primeiras metas. A principal razão destes problemas está no desenvolvimento de aplicações *web* que continuam improvisadas, sem suporte de metodologias ou processos disciplinados. Isso proporciona o surgimento de grande quantidade de sistemas mal desenvolvidos e que podem gerar muitas falhas, colocando em risco a confiança na internet. Para a construção de sistemas, e aplicações na *web* é preciso que os desenvolvedores adotem abordagens disciplinadas de Engenharia da *Web* (COSTAGLIOLA; FERRUCCI; FRANCESE, 2002) (MURUGESAN; GINIGE, 2005) (PRESSMAN, 2002).

É fato que a tradicional engenharia de Software se torna insuficiente para o desenvolvimento de aplicações *web*, por não abordar aspectos importantes como de

interface e estrutura de fácil percepção aos usuários. Então surge o reconhecimento da real necessidade de desenvolver uma engenharia de Web que utiliza metodologias de Hipermedia, e juntas propõem uma aproximação de um desenvolvimento sistemático e disciplinado (COSTAGLIOLA; FERRUCCI; FRANCESE, 2002).

Muitas propostas de modelos e metodologias de engenharia Web/Hipermedia foram desenvolvidas. Elas têm por objetivo ajudar o desenvolvedor pensar de forma estruturada, com o foco em aspectos como *links* de estrutura e navegação, além de manter a comunicação entre programadores, analistas, administrador do software e usuário (COSTAGLIOLA; FERRUCCI; FRANCESE, 2002).

Gabrielli et al (2007) diz que os esforços para possibilitar que os sistemas de informação usufruam os benefícios da internet são grandes, uma vez que os sistemas de informação baseados na *web* podem ser acessados através de um navegador por várias pessoas simultaneamente, geograficamente distantes, diferentemente dos sistemas tradicionais.

Como em qualquer engenharia, na engenharia de *software* é fundamental a medição do processo do software. A medição tem o objetivo de melhorar o *software*, avaliar a qualidade dos produtos do trabalho técnico e auxiliar a tomada de decisão tática. A contagem de ponto de função é uma técnica que define algumas abstrações que determinam o tamanho funcional do sistema. As funcionalidades a serem medidas trata-se do armazenamento e processamento de dados dos usuários. Cinco características das informações são necessárias para as medidas: quantidades de entradas do usuário, quantidades de saídas do usuário, número de consultas do usuário, quantidade de arquivos e quantidade de interfaces externas. (PRESSMAN, 2002) (VAZQUEZ; SIMÕES; ALBERT, 2003).

O objetivo deste trabalho está na caracterização dos sistemas baseados na *web*, nas suas metodologias de desenvolvimento e na implementação de um sistema com estrutura GUI (*Graphical User Interface*) e Web, para comparar os esforços de implementação. Para isso no TCC-I, foram estudados processos de desenvolvimento, elaborada uma taxonomia de aplicações *web*, e correlacionados aos processos através de fatores dos sistemas de informações e processos de desenvolvimento. E no TCC-II será desenvolvida uma aplicação em duas versões para comparação dos esforços. Uma com estrutura GUI com processo mais adequado às características do sistema de informação desktop, implementado em

Java através dos componentes Swing. E outra versão do mesmo sistema em estrutura Web, também com um processo adaptado conforme suas características, e desenvolvido também em Java para web com JSF (*Java Server Faces*). Serão feitas medidas de ponto de função e esforço (hora-homem) em cada caso. Com essas medidas será possível comparar o gasto na migração de sistemas de informação com estrutura GUI para *Web*.

Tais migrações na verdade sempre consistiram em re-engenharia, ou seja, praticamente todo o software é normalmente re-escrito. Isto tem um gasto de consumo. As empresas que desenvolvem software têm uma historia de consumo de recurso para o desenvolvimento de sistemas em GUI. Não possuem histórico de desenvolvimento de sistemas *web*. A contribuição deste trabalho, dará as empresas uma idéia de consumo de recurso que a tecnologia *web* consome, comparando a tecnologia GUI. Isso facilitará as empresas nas suas decisões sobre o investimento na evolução de seus sistemas para *web*.

2. Objetivos

Objetivo Geral: Fazer uma analise comparando esforços de implementação de aplicações de estrutura GUI e *Web*. A diferença do esforço entre o desenvolvimento GUI e *Web*, ajudará os desenvolvedores de software a planejarem seus investimentos, uma vez que atuais aplicações GUI sofrem pressão para usarem tecnologia *Web*.

Objetivos Específicos:

Modelagem Orientada a Objeto para aplicação GUI e *Web*;

Processo de Desenvolvimento GUI e *Web*;

Tecnologia de Implementação GUI e *Web*

3. Metodologia

As atividade a serem desenvolvidas são:

- 1- Estudo dos processos de desenvolvimento;
- 2- Estudo de aplicações e propor uma taxonomia;
- 3- Apresentação parcial de TCC I;
- 4- Co-relacionamento de taxonomia com processos;
- 5- Escolha e concepção de uma metodologia;
- 6- Apresentação final do TCC I;

As reuniões entre orientador e aluno serão semanais e presenciais, com duração de aproximadamente 1 (uma) hora, e quando solicitado pelo orientador será utilizado o correio eletrônico. Dessa forma o professor orientador fará o controle das atividades realizadas pelo orientado e o cumprimento de prazos e presenças nas reuniões.

7. Referências Bibliográficas

COSTAGLIOLA, G.; FERRUCCI, F.; FRANCESE, R. *Web engineering: Models and*

methodologies for the design of hypermedia applications. In: CHANG, S. K. (Ed.). Hand-book of Software Engineering & Knowledge Engineering. Italy: Emerging Technologies, 2002. v. 2, p. 181–199. Disponível em: <<ftp://cs.pitt.edu/chang/handbook/58b.pdf>>.

DEITEL, Harvey M.; DEITEL, Paul J; LISBÔA, Carlos Arthur Lang. Java: como programar. 4. ed. Porto Alegre: Bookman, 2003. 1386 p. ISBN 8536301236 (broch.)

GABRIELI, Leandro V. et al. *Uso de um Framework no Desenvolvimento de Sistemas de Informação Web*. XXVII Encontro Nacional de Engenharia de Produção, outubro 2007. Acessado em 07/03/2008. Disponível em <http://w3box.pro.br/arquivos/ENEGEP2007_TR640475_0336.pdf >

LAUDON, Kenneth C.; LAUDON, Jane P. *Sistemas de Informação Gerenciais: administração a empresa digital*. 5. Ed. São Paulo: Pearson, Prentice Hall, 2004.

MURUGESAN, S.; GINIGE, A. *Web Engineering: introduction and perspective*. 1–30 p. 2005. Acessado em 27/08/2007. Disponível em: <<http://www.idea-group.com/downloads/excerpts/SuhChapter1.pdf>>.

PRESSMAN, R. S. *Engenharia de Software*. 5. ed. Rio de Janeiro: Mc Graw Hill, 2002.

SCHMITT, Rogério A.; SCHMITT Fátima A. B. S.. *Sistemas de Informação Baseados na Web*. Revista Eletrônica do CEBLUS, publicada em maio de 2007. Acessado em 01/03/2008. Disponível em: <http://www.cesblu.br/revista/artigos/Artigo_Pos_Web_IV_Rogério_Antonio_Schmitt.pdf>.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme S.; ALBERT, Renato M.. *Análise de Pontos de Função, medição, estimativas e gerenciamento de projetos de software*. 1. ed. São Paulo: Editora Érica, 2003.

ZANETI JR, Luiz A. *Sistema de informação Baseados na Tecnologia Web: Um estudo sobre seu Desenvolvimento*. São Paulo, Brasil: Dissertação para obter titulação de Mestre em Administração – Universidade de São Paulo, 2003. Acessado em 01/03/2008. Disponível em: <<http://www.teses.usp.br/>>.

Salvador Antônio dos Santos

Janaise Francisco Sicka

Apêndice B – Levantamento de Requisitos do Sistema Questionário do Projeto de Extensão Envolve-se

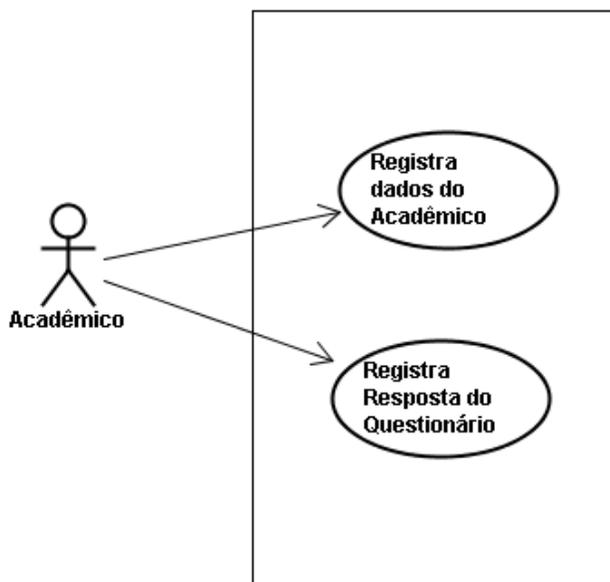
REQUISITOS

- [R01] O sistema deve ter dois acessos: acadêmico, com acesso apenas ao questionário e coordenador, com acesso ao cadastro de coordenador e as repostas do questionário.
- [R02] O sistema deve armazenar as informações do acadêmico.
- [R03] O sistema deve permitir que o coordenador visualize as informações do acadêmico, separados por curso.
- [R04] O sistema deve reconhecer o aluno pelo numero da matrícula.
- [R05] As informações do acadêmico devem ser substituídas sempre que uma matricula seja reconhecida.

RESTRICÇÕES

- [RN01] Acadêmico
Só possui acesso ao questionário.
- [RN02] Coordenador
Possui acesso irrestrito.
- [RN03] Acesso do coordenador
Deve cobrar usuário e senha.

DIAGRAMA DE CASO DE USO



ATORES

Acadêmico: É o usuário que irá informa sua matricula, curso e responder o questionário, alimentando o banco de dados

DETALHAMENTO DOS CASOS DE USO

[CSU01] Acesso ao Acadêmico

Sumário: Inserir

Ator Primário: Acadêmico

Precondição: Informar

Fluxo Principal:

1. Informar número da matricula e escolher curso;
2. Clicar no botão Entrar;
3. O sistema insere os dados no banco de dados;
4. Apresentar o questionário a ser respondido;

Fluxo de Exceção:

1. Nenhuma resposta poderá ficar em branco, caso isso aconteça o sistema deverá emitir uma mensagem.

[CSU02] Questionário Envolve-se

Sumário: Inserir

Ator Primário: Acadêmico

Precondição: Informar matricula e curso

Fluxo Principal:

1. Apresentar as perguntas do questionário, com espaço para o usuário responder.
2. O Acadêmico responde as perguntas
3. Ao clicar no botão enviar todas as repostas são armazenadas no banco de dados.
4. É apresentado a tela final com a mensagem Sucesso.

Fluxo de Exceção:

1. Nenhuma resposta poderá ficar em branco, caso isso aconteça o sistema deverá emitir uma mensagem.

MODELO CRC

Acadêmico (entidade)	
Responsabilidade	Colaboradores
1- Conhecer numero da matricula	Questionário
2- Conhecer curso	

Questionário (entidade)	
Responsabilidade	Colaboradores
1- Conhecer as respostas do acadêmico.	Acadêmico

DIAGRAMA DE CLASSES

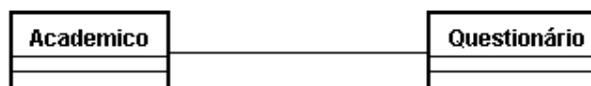
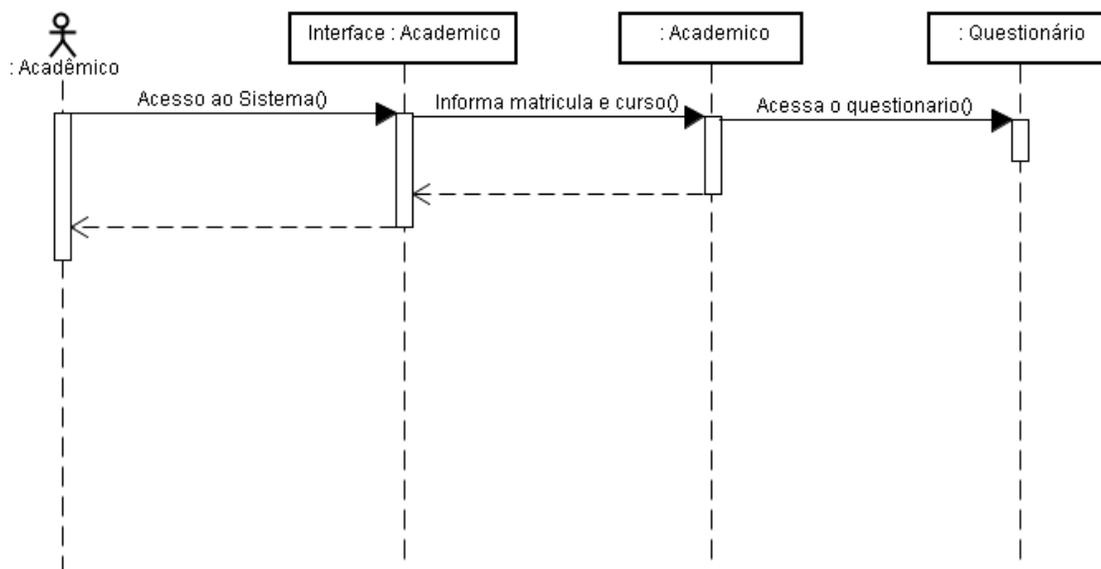
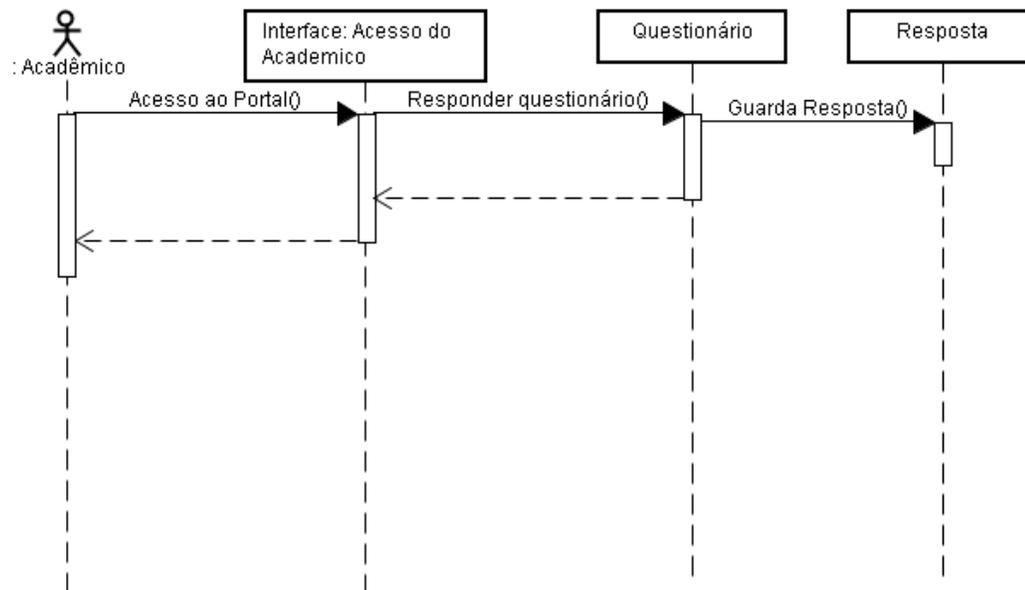


DIAGRAMA DE SEQÜÊNCIA

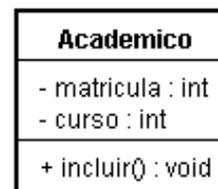
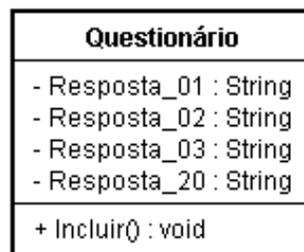
CSU 01 – Acesso ao Acadêmico



CSU 02 – Questionário Envolve-se



Detalhamento das Classes de Domínio



Telas GUI

Tela Principal do Sistema

Visualização do desenho [TelaAcademico]

Matricula :

Curso : C. Computação

Entrar

Tela do Questionário

Visualização do desenho [Questionario]

Familia

Onde mora : com pais ou parente

Estado Civil dos Pais : solteiros

Possui irmãos : 0

Formação da Mãe : 1º a 4º série incompleto

Formação do Pai : 1º a 4º série incompleto

Profissão da Mãe :

Profissão do Pai :

Acadêmico

O Ensino Fundamental foi em : Escola Particular

O Ensino Médio foi em : Escola Particular

Você fez pré vestibular : Sim

Quanto tempo está na Universidade : 6 meses - 1 ano

Como vem a Universidade : a pé

Onde faz a Alimentação : casa

Possui trabalho : Bolsa Trabalho/ Pesquisa/Extensão

Possui Plano de Saúde : Sim

Faz algum tratamento de Saúde : Sim

Pratica atividade esportiva, Qual :

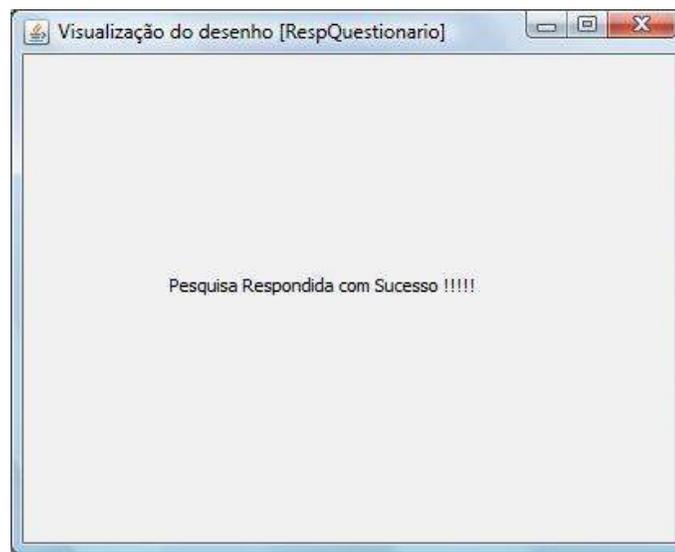
Possui algum Hobby, qual :

Qual sua religião : Católico

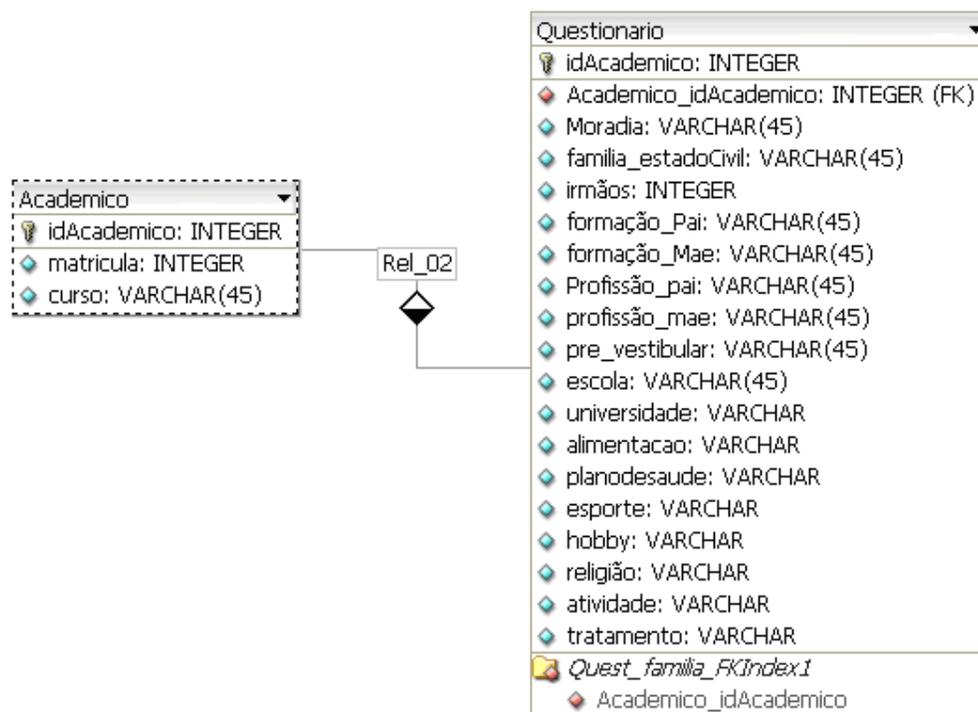
Você acha importante ter algum espaço dentro da universidade para atividades Espirituais : sim

Responder

Tela de Finalização



Camada de Persistência



Detalhamento das tabelas

Acadêmico

Item de dado	Atributo	Chave	Tipo	Edição
#idAcademico	#código do acadêmico	Primária	Auto Numeração	auto
matricula	matricula do acadêmico		Inteiro	999999999
curso	Nome do curso		varchar	VC[45]

Questionário

Item de dado	Atributo	Chave	Tipo	Edição
#Coordenador_idPesquisa	#codigo da pesquisa	Estrangeiro	Auto numeração	auto
#Academico_idAcademico	#codigo do acadêmico	Estrangeiro	AutoNumeração	Auto
Moradia	Resposta questionário		Varchar	Vc[45]
Família_estadoCivil	Resposta questionário		Varchar	Vc[45]
Irmãos	Resposta questionário		Varchar	Vc[45]
Formação_pai	Resposta questionário		Varchar	Vc[45]
Formação_mae	Resposta questionário		Varchar	Vc[45]
Profissão_pai	Resposta questionário		Varchar	Vc[45]
Profissão_mae	Resposta questionário		Varchar	Vc[45]
Pré_vestibular	Resposta questionário		Varchar	Vc[45]
Ensino_fundamental	Resposta questionário		Varchar	Vc[45]
Ensino_medio	Resposta questionário		Varchar	Vc[45]
Tempo_Universidade	Resposta questionário		Varchar	Vc[45]
Alimentação	Resposta questionário		Varchar	Vc[45]
Plano de saúde	Resposta questionário		Varchar	Vc[45]
Esporte	Resposta questionário		Varchar	Vc[45]
Hobby	Resposta questionário		Varchar	Vc[45]
Religião	Resposta questionário		Varchar	Vc[45]
Atividade	Resposta questionário		Varchar	Vc[45]
Tratamento	Resposta questionário		Varchar	Vc[45]

Apêndice C – Questionário Impresso do Sistema Questionário do Projeto de Extensão Envolve-se



PROJETO ENVOLVA-SE

Este questionário tem por objetivo traçar o perfil dos acadêmicos do Centro de Ciências Tecnológicas de Joinville (UDESC - Joinville) . As informações prestadas são de caráter confidencial, servindo, tão somente, para o diagnóstico e planejamento das atividades de ensino e Extensão deste Centro e do Projeto ENVOLVA-SE . Preencha com responsabilidade e exatidão.

Ressaltamos que as informações não interferirão na sua vida acadêmica, pelo contrario ajudarão a propor novas ações e metodologias de trabalho no Campus. Melhorando nossa qualidade de Vida e de Ensino.

Você deverá:

- optar por apenas uma resposta para cada pergunta;
- verificar se nenhuma pergunta deixou de ser respondida;
- é importante sua participação para o Crescimento do Centro.

Nº Matricula/CPF: _____

Data: ____/____/____

1- MORADIA

- Mora em: com pais ou parentes pensionato republica
 sozinho outro _____

Caso more em republica:

Quantos integrantes? _____

2- FAMILIA

- Seus pais são: solteiros casados separados
 falecidos pai mãe outro _____

- Tem irmãos? sim não

Se sim: Quantos? _____

• Formação do Pai

- 1° a 4° série incompleto 1° a 4° série completo ensino médio incompleto
 ensino médio completo superior incompleto superior completo

• Formação da Mãe

- 1° a 4° série incompleto 1° a 4° série completo ensino médio incompleto
 ensino médio completo superior incompleto superior completo

• Profissão dos pais:

Pai: _____

Mãe: _____

3- ACADÊMICO	
<ul style="list-style-type: none"> • Você frequenta escola Pública? <input type="checkbox"/> Ensino Fundamental <input type="checkbox"/> Ensino Médio 	<ul style="list-style-type: none"> ou Particular? <input type="checkbox"/> Ensino Fundamental <input type="checkbox"/> Ensino Médio
<ul style="list-style-type: none"> • Você fez cursinho pré-vestibular? <input type="checkbox"/> Sim 	<input type="checkbox"/> Não
<ul style="list-style-type: none"> • Quanto tempo está na Universidade? <input type="checkbox"/> menos de 6 meses <input type="checkbox"/> 6 meses - 1 ano <input type="checkbox"/> 1 ano - 1 ano e 6 meses <input type="checkbox"/> 1 ano e 6 meses - 2 anos <input type="checkbox"/> 2 anos – 2 anos e 6 meses <input type="checkbox"/> 2 anos e 6 meses - 3 anos <input type="checkbox"/> 3 anos – 3 anos e 6 meses <input type="checkbox"/> 3 anos e 6 meses - 4 anos <input type="checkbox"/> 4 anos – 4 anos e 6 meses <input type="checkbox"/> 4 anos e 6 meses - 5 anos <input type="checkbox"/> 5 anos – 5 anos e 6 meses <input type="checkbox"/> 5 anos e 6 meses - 6 anos <input type="checkbox"/> 6 anos – 6 anos e 6 meses <input type="checkbox"/> 6 anos e 6 meses - 7 anos <input type="checkbox"/> 7 anos – 7 anos e 6 meses <input type="checkbox"/> 7 anos e 6 meses - 8 anos <input type="checkbox"/> 8 anos – 8 anos e 6 meses <input type="checkbox"/> 8 anos e 6 meses - 9 anos <input type="checkbox"/> 9 anos – 9 anos e 6 meses <input type="checkbox"/> 9 anos e 6 meses - 10 anos <input type="checkbox"/> mais de 10 anos 	
<ul style="list-style-type: none"> • Como vem para Universidade? <input type="checkbox"/> a pé <input type="checkbox"/> bicicleta <input type="checkbox"/> ônibus <input type="checkbox"/> moto <input type="checkbox"/> carro <input type="checkbox"/> outro _____ 	
<ul style="list-style-type: none"> • Onde faz sua alimentação? <input type="checkbox"/> casa <input type="checkbox"/> lanchonete <input type="checkbox"/> R.U. <input type="checkbox"/> outro _____ 	
<ul style="list-style-type: none"> • Você tem algum tipo de trabalho? <input type="checkbox"/> sim <input type="checkbox"/> não Se sim: <input type="checkbox"/> bolsa trabalho/ pesquisa/ extensão <input type="checkbox"/> estágio <input type="checkbox"/> outro _____ Qual empresa? _____ 	
<ul style="list-style-type: none"> • Você tem plano de saúde ? <input type="checkbox"/> sim <input type="checkbox"/> não Se sim: Qual? _____ 	
<ul style="list-style-type: none"> • Faz algum tratamento de saúde? <input type="checkbox"/> sim <input type="checkbox"/> não Se sim: Qual? _____ 	
<ul style="list-style-type: none"> • Você pratica alguma atividade esportiva fora da Universidade? <input type="checkbox"/> sim <input type="checkbox"/> não Se sim: Qual? _____ 	
<ul style="list-style-type: none"> • Você tem algum hobby? (Ex: toca instrumento, pintura, possui banda, dança, artesanato, esportes radicais) <input type="checkbox"/> sim <input type="checkbox"/> não Se sim: Qual? _____ 	
<ul style="list-style-type: none"> • Qual sua religião? <input type="checkbox"/> católico <input type="checkbox"/> protestante/ evangélico <input type="checkbox"/> tenho minha fé independente de religião <input type="checkbox"/> outro _____ 	
<ul style="list-style-type: none"> • Você acha importante ter algum espaço dentro da Universidade para atividades espirituais? <input type="checkbox"/> sim <input type="checkbox"/> não 	

Apêndice D – Documentação do Processo Espiral para sistema com interface GUI

Modelo de Processo Espiral

1- Etapa de Formulação

- Objetivo: Realizar uma anamnese dos acadêmicos do campus da UDESC- Joinville, com o objetivo de traçar um perfil desta comunidade acadêmica.
- Meta: Atingir pelo menos 50% dos acadêmicos, para obter um resultado satisfatório.

2- Etapa de Planejamento

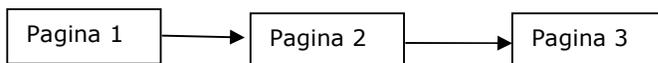
- Levantamento de custo: não haverá custo, pois a tecnologia, ferramenta de desenvolvimento são gratuitas e mão a obra será fornecida pela acadêmica que desenvolve este trabalho.

3- Etapa de Análise

O levantamento de requisitos é apresentado no anexo B deste trabalho.

4- Etapa de Engenharia

- Projeto de arquitetura: conforme a análise de requisito, o modelo de arquitetura que mais se adequa ao sistema é o linear, ou seqüencial.



- Projeto de navegação: A navegação do usuário será guiada por botões. A primeira tela possui o botão ENTRAR que leva o usuário a próxima tela, que possui o botão RESPONDER que finaliza a aplicação.
- Projeto de interface: As telas já foram definidas na análise de requisito do anexo B.

5- Etapa de Geração de testes das páginas

- Geração de código: a ferramenta utilizada para o desenvolvimento é o NetBean 6.5.1 , linguagem Java com componentes Swing.

- Teste: os testes são realizados pelo programador, verificando a inclusão de dados e requisitos que impedem prosseguir sem que todas as respostas sejam preenchidas.

6- Avaliação do Cliente

A avaliação do cliente será realizado após o termino deste trabalho pelo coordenador do projeto de extensão Envolve-se.

Apêndice E – Documentação da Metodologia OOHDM para sistema Web

Metodologia OOHDM

1- Modelagem Conceitual:

As classes e relacionamentos solicitados no modelo conceitual são apresentados no levantamento de requisitos do Anexo B.

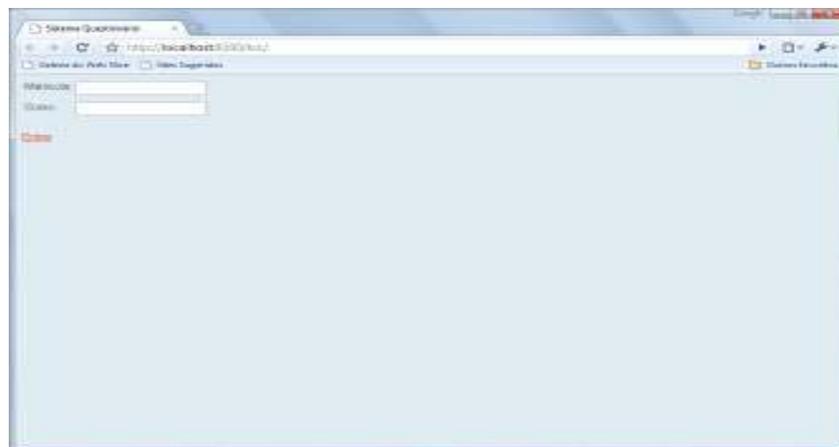
2- Projeto de Navegação:

O projeto navegacional considera as tarefas que o usuário deverá realizar. Como apresentado no levantamento de requisito do anexo B, o sistema possui apenas um (1) tipo de usuário, o acadêmico, este irá informar sua matrícula, curso e clicar no link ENTRAR, os dados serão armazenados no banco de dados e segue para a próxima página. A próxima página é onde será respondido todas as perguntas do questionário, e após será clicado no link RESPONDER, os dados serão novamente armazenados no banco de dados e o usuário será conduzido a pagina de finalização.

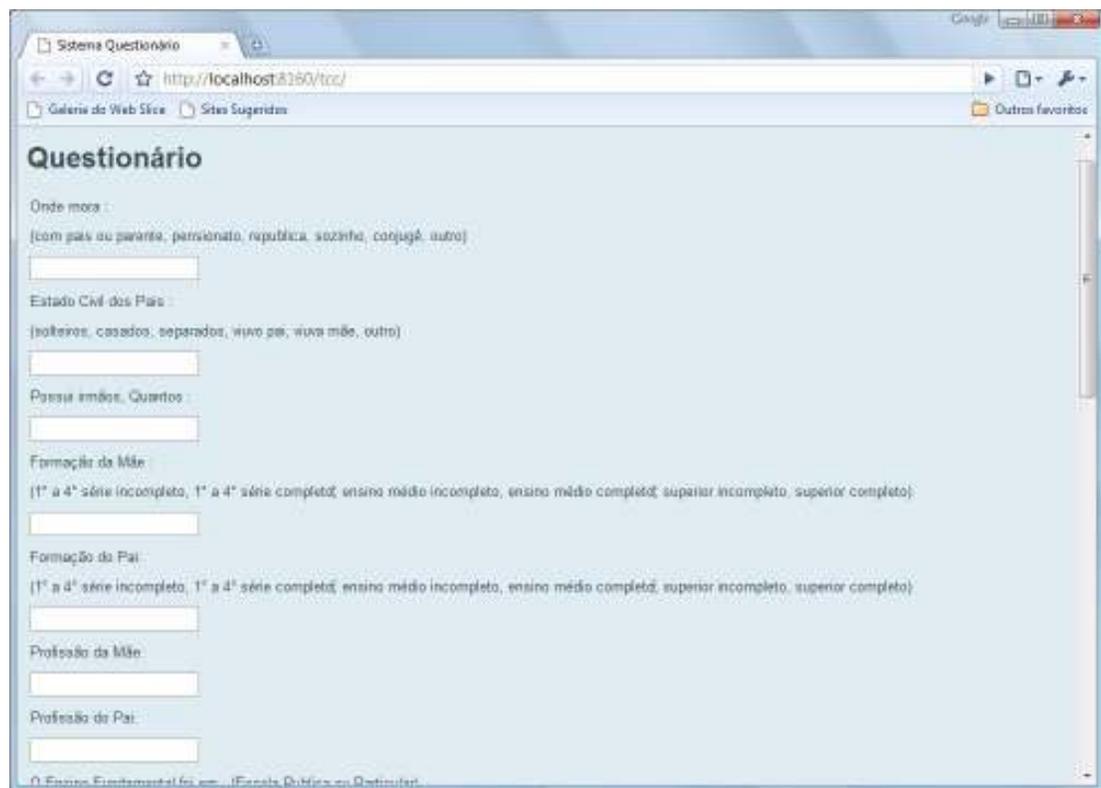
3- Projeto de interface:

Segue o protótipo da tela de apresentação:

Tela do Acadêmico:



Tela Questionário

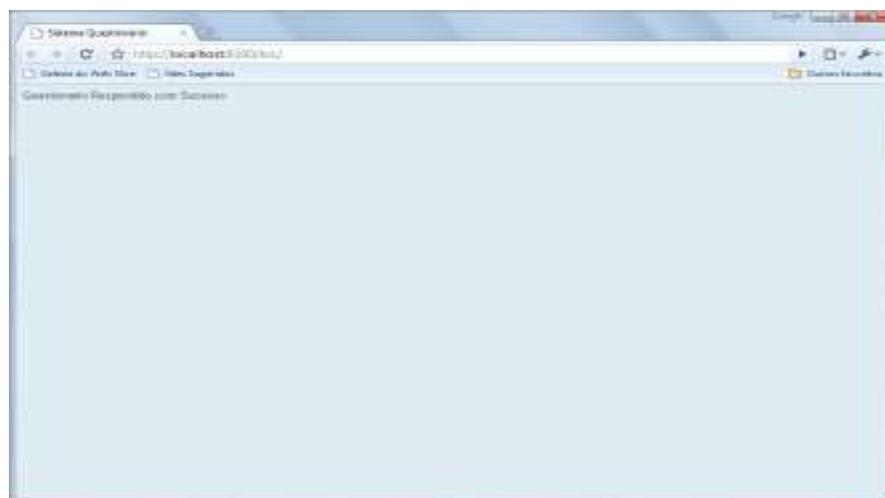


The screenshot shows a web browser window titled 'Sistema Questionário' with the address bar displaying 'http://localhost:8180/tcc/'. The page content is a questionnaire form with the following fields and labels:

- Questionário**
- Onde mora :** (com pais ou parente, pensionato, republica, sozinho, cônjuge, outro)
- Estado Civil dos Pais :** (solteiros, casados, separados, viúvo pai, viúva mãe, outro)
- Possui irmãos, Quantos :**
- Formação da Mãe :** (1ª a 4ª série incompleto, 1ª a 4ª série completo, ensino médio incompleto, ensino médio completo, superior incompleto, superior completo)
- Formação do Pai :** (1ª a 4ª série incompleto, 1ª a 4ª série completo, ensino médio incompleto, ensino médio completo, superior incompleto, superior completo)
- Profissão da Mãe :**
- Profissão do Pai :**

At the bottom of the page, there is a small footer: '© Escola Estadual de Ensino Médio - (Escola Duília de Oliveira)'. The browser's address bar also shows 'Galéria de Web Sites' and 'Sites Sugeridos'.

Tela Sucesso



4- Implementação :

A implementação é realizada na linguagem JSF, com a ferramenta NetBeans 6.5.1.