EDU.TEC Revista Científica Digital da Faetec
ISSN: 1984-2007

Smart Web Developer: Uma proposta de desenvolvimento de aplicações web através de ferramenta CASE, integração e engenharia reversa em PHP.

Vinícius Marques da Silva Ferreira[1,2], Alfredo Nazareno Pereira Boente[1,2], Ricardo Marciano dos Santos[2], Kilmer Pereira Boente[1,2]

[1] Universidade Federal do Rio de Janeiro - LAMAE/NCE/UFRJ, Av. Athos da Silveira 274, sala E-1008, CCMN, Cidade Universitária, Rio de Janeiro, Brasil, CEP: 21.941-916

[2] Faculdades de Educação Tecnológica do Estado do Rio de Janeiro - FAETERJ Duque de Caxias, Rua Almirante Cochrane, s/n, Santa Lúcia, Duque de Caxias, Rio de Janeiro, Brasil, CEP: 25.271-000

profvmarques@gmail.com, kilmer_pereira@yahoo.com.br, richackerbr@gmail.com

Resumo. *Este artigo enfatiza uma proposta de melhoria no processo de desenvolvimento de aplicações web, o qual está baseado nos conceitos da engenharia reversa e padrões da programação orientada a objeto. A pesquisa tem caráter exploratório, descritivo e qualitativo sendo realizada junto à equipe de desenvolvimento de software de certa fundação. Deste modo constitui-se o produto de software Smart Web Developer a partir da pesquisa do presente trabalho, na qual se refere à melhoria de produção de aplicações voltadas para web, com desenvolvimento de um novo recurso que utiliza como conceito a engenharia reversa, a biblioteca mPDF, cujo foco é gerar relatórios dinâmicos, conversão dos dados acessados pela linguagem PHP (Hypertext Preprocessor) em formato PDF (Portable Document Format) e utilização do SGBD Mysql Server, integrando o modelo MVC da aplicação de software web gerado, promovendo assim a produção, aperfeiçoamento, qualidade e redução no tempo de desenvolvimento de aplicações de produtos de software web.*

Palavras-Chave*:Ferramenta CASE, Engenharia Reversa, Web Developer*

1. Introdução

No atual mercado altamente competitivo e globalizado, os produtos de software são bastante utilizados pelas empresas, organizações e instituições dos mais diversos setores, que envolvem desde aplicações triviais até sistemas críticos.

O desenvolvimento de software tem sido uma atividade de grande importância na sociedade contemporânea, pois segundo Mecenas e Oliveira (2005), a produção de software deixou de ser, há algum tempo, uma atividade baseada apenas na intuição ou na experiência dos desenvolvedores. A crescente utilidade de computadores nas mais diversas áreas do conhecimento humano tem gerado um grande acréscimo na demanda

por soluções computadorizadas. Desta maneira as empresas, organizações e instituições conseguem direcionar seus focos, áreas de atuação, elaboração de novos produtos, oferta de novos serviços etc., com o intuito de atender aos desejos dos clientes.

O processo de desenvolvimento de produtos de software tem sido objeto de inúmeros estudos há mais de três décadas, numa tentativa de derivar modelos que possibilitem o gerenciamento das fases de produção de software que apresentem a qualidade desejada por seus clientes (Pressman, 2011).

Prezando pela melhoria, qualidade dos produtos e o aumento da produtividade no processo de desenvolvimento de produtos de software, temos a engenharia de software que enfatiza o cuidado de aspectos relacionados ao estabelecimento de métodos, processos, ferramentas, técnicas e ambientes de suporte ao desenvolvimento de software.

Na proporção em que os métodos de desenvolvimento se tornam mais sofisticados, a complexidade da administração do processo de software aumenta. As limitações do homem diante da gestão do grande volume de informações e etapas sistêmicas que englobam a aplicação dos métodos demandam ferramentas de apoio automatizadas.

Pressman (2011) propõe que o uso de metodologias para desenvolvimento de software seja o primeiro passo para obtenção da qualidade de processos e de produtos de software.

Neste contexto, o Smart Web Developer, é uma ferramenta de inteligência computacional baseada nos padrões da programação orientada a objeto e conceitos da engenharia reversa, tem com objetivo auxiliar o desenvolvimento de aplicações voltadas para web.

Este artigo está organizado em quatro seções. A seção 1 faz introdução contextualizada do problema. A seção 2 trata da abordagem teórica do trabalho, enfatizando os aspectos subjacentes, bem como apresentando e aplicando conceitos da engenharia de produção, utilização de ferramentas CASE, padrões de programação orientada a objeto e qualidade na produção de software. A metodologia está descrita na seção 3. Finalmente os resultados e conclusões estão na seção 4.

2. Abordagem Teórica

2.1. Engenharia reversa

O termo “engenharia reversa” teve sua origem na análise do hardware, em que a prática de extração de projetos de produtos concluídos é comum, sendo aplicada para a melhoria desses produtos e análise de produtos de competidores (Chikofsky, 1990).

Nesse viés a engenharia reversa pode ser definida como o processo de desenvolvimento de um conjunto de especificações para um sistema de hardware complexo através do exame ordenado dos componentes do sistema (Rekoff, 1985).

Tratando-se de software esse processo é bastante utilizado para se ter um entendimento do mesmo no nível de desenvolvimento. Portanto, define-se engenharia reversa para um software como o processo de análise para identificar seus componentes e inter-relacionamentos e criar representações do mesmo em outra forma ou num nível mais alto de abstração (Chikofsky, 1990).

EDU.TEC Revista Científica Digital da Faetec
ISSN: 1984-2007

Abstração pode ser definida como a tarefa de utilizar apenas assuntos relevantes ao propósito em questão (Oxford, 1986). Dois conceitos podem ser derivados, como descrito a seguir e ilustrado na figura 1.

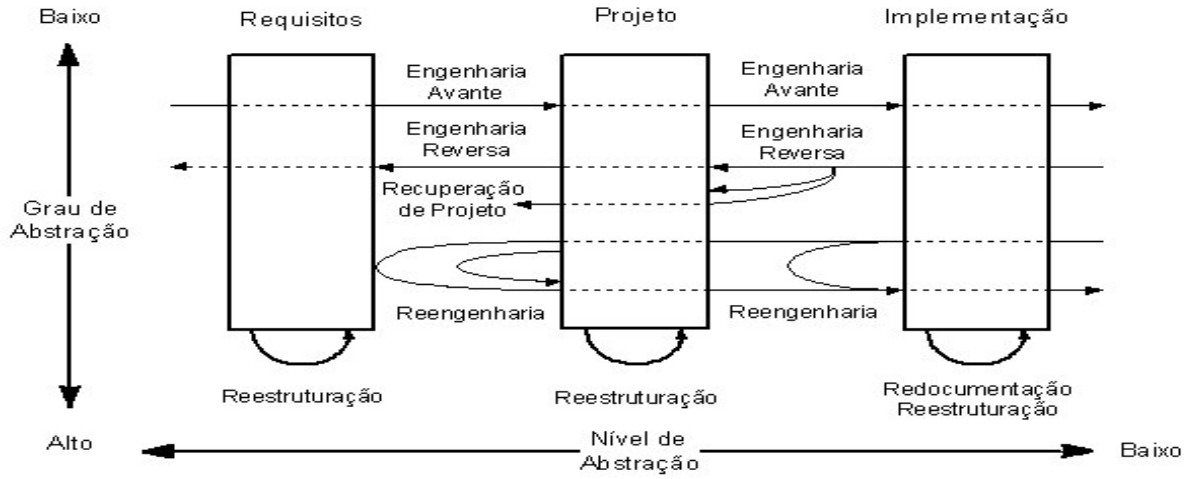

Figura 1. Nível e grau de abstração do ciclo de vida do software.

Nível de Abstração: ocorre entre as fases do ciclo de vida do software. Quanto mais próximo da fase implementação, menor é o nível de abstração e quanto mais próximo se estiver da fase de especificação de requisitos, maior o nível de abstração;

Grau de Abstração: ocorre dentro de uma mesma etapa do ciclo de vida do software. Se as informações forem escassas em detalhes, o grau de abstração é alto. Se as informações forem bastante detalhadas, é baixo o grau de abstração (Chikofsky, 1990). O mesmo autor afirma que existem duas categorias de engenharia reversa: a redocumentação e a recuperação de projeto. A redocumentação pode ser também denominada como visualização de código, sendo esta, utilizada na criação de representações a partir de informações obtidas apenas da exploração analítica do código fonte, com o objetivo de recuperar a documentação do software.

A recuperação de projeto, também conhecida como entendimento de programa, utiliza além da análise do código, o entendimento do domínio das informações relativo ao software e as deduções, com a finalidade de obterem-se informações com um nível mais alto de abstração.

Para que as questões técnicas relativas ao processo de engenharia reversa sejam tratadas de forma mais efetiva, o processo deve tornar-se maduro, capaz de ser repetido, e passível de evolução de modo que cada vez mais passos possam ser executados por ferramentas automatizadas (Wong et al., 2000).

2.2. Ferramentas CASE

Ferramentas CASE (Computer Assisted / Aided Software Engeneering) são aplicativos computacionais que suportam uma ou mais atividades do processo de desenvolvimento de software. O CASE ou Engenharia-análise de sistemas apoiada-assistida por computador é uma ferramenta ou conjunto de técnicas facilitadoras de desenvolvimento de software moderno (Resende, 2006). A introdução de tais ferramentas visa a melhoraria do aceleramento da produtividade do processo de produção e qualidade do software. As ferramentas CASE podem ser divididas em :

(a) horizontais - disponibilizam serviços para serem utilizados durante todo o processo de software, tais como gerenciamento de versões, suporte à documentação e configurações;

(b) verticais - Utéis em fases específicas no processo de desenvolvimento de produtos de software, tais como levantamento, análise de requisitos e teste de software.

Entretanto, ferramentas CASE também podem ser classificadas segundo o conjunto de serviços principais que estas disponibilizam. Um serviço é uma ação efetuada pelo computador que é de interesse do desenvolvedor (Schefström, 1993).

Acerca de ferramentas CASE é de suma importância mencionar o conceito do "Triângulo para o sucesso", este conceito segundo Terry Quatrani (1999), para um projeto bem sucedido é necessário conhecer bem três coisas: Notação, Processo e Ferramenta, conforme ilustrado na figura 2.

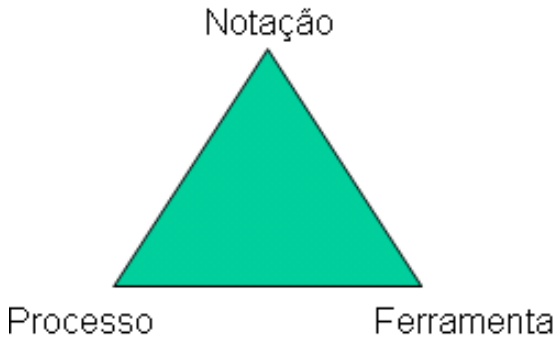

Figura 2 - Triângulo para o sucesso.

Uma proposta de classificação é apresentada no quadro 1. Esta tabela proporciona a observação do amplo aspecto de ferramentas CASE existente, apesar de ser comum referência a ferramentas CASE como ferramentas específicas para projeto e análise de software (Pressman, 2011).

Atividades	Exemplos
Planejamentos	Foundation, Interactive Engineering Workbench, Information Engineering Facility;
Gerenciamento de Projetos	SuperProject, Microsoft Project, MacProject II, ESTIMATES;
Especificação de Requisitos	CORE, RMS/PC, R-Trace;
Especificação Formal de Sistemas	CADIZ, OBJ;
Documentação	Interleaf, Page Maker (Aldus);
Comunicação	Utilitários do Unix, Microsoft mail;

Controle de Qualidade	Controle de Qualidade Q/Auditor, Auditor;
Gerenciamento de Versões	SCCS do Unix, PVCS ;
Análise e Projeto de Software	JSD, SADT, HOOD, PC Case, OMT;
Projeto e Desenvolvimento de Interfaces	Interviews, Lucas Film;
Programação	Turbo X's, Anna.

Quadro 1. Classificação de ferramentas CASE

A elaboração de ferramentas CASE é como a construção de um software para domínio detalhado. Apesar dos possíveis serviços disponibilizados por essas ferramentas serem vastos, há características comuns. Dado que as ferramentas CASE, geralmente auxiliam em atividades específicas, um projeto carece de várias destas para conseguir suprir todo o processo de software. Tais ferramentas podem ser vista como um módulo que mantém um conjunto de serviços. Estes módulos devem ser combinados para disponibilizar novas funcionalidades. Portanto, além do problema de compartilhamento de dados, o uso de inúmeras ferramentas isoladas pode trazer situações de sobreposição de funcionalidades.

A integração de ferramentas CASE é um campo de estudo muito explorado atualmente. A partir destes estudos, modelos de integração que representam os aspectos essenciais a serem tratados vêm sendo produzidos. Um dos mais conhecidos é o modelo de três dimensões (Schefström, 1989) apresentado na figura 3, onde se pode concluir que os aspetos fundamentais de integração são vistos em três níveis dimensionais: apresentação, dados e controle.

A apresentação representa a igualdade do estilo e comportamento da interface das ferramentas. Isto engloba uso uniforme de janelas e menus, bem como mesmo comportamento dos eventos semelhantes. Um dos princípios básicos do projeto de ferramentas é a separação da interface do código da aplicação (Schefström, 1993). Este fato permite o desenvolvimento da interface independentemente do código da aplicação, fácil às alterações na interface sem influenciar na padronização, código e concentração de investimentos.

A integração por dados tem o objetivo de prover formas de compartilhar os dados manipulados pelas ferramentas, evitando-se a diversidade e multiplicação, de estruturas e serviços. O compartilhamento dá-se através de dados persistentes. Utilizando um repositório de dados comum, por exemplo, um gerenciador de banco de dados, como o Mysql server onde todas as ferramentas armazenam seus dados.

EDU.TEC Revista Científica Digital da Faetec
ISSN: 1984-2007

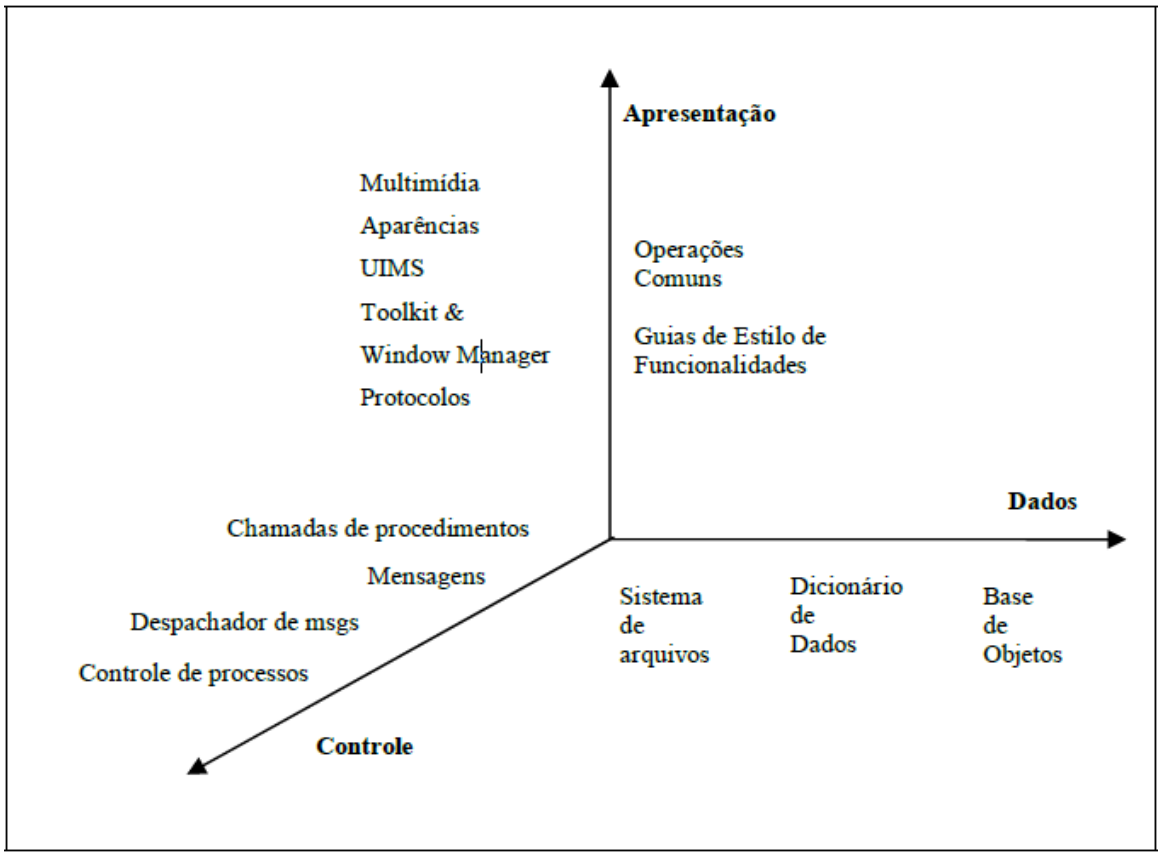

Figura 3. Modelo de três dimensões de integração de ferramentas CASE.

A integração por controle oferece mecanismos de comunicação entre ferramentas independente do compartilhamento de dados. As ações executadas pelas ferramentas são comunicadas às outras através de sinais de controle. A intenção desse mecanismo é ser mais flexível, obtendo integração sem sacrificar a independência da ferramenta.

As pesquisas nessa área vêm sendo intensificadas dando ênfase no trabalho cooperativo, visando facilitar a comunicação à distância entre usuários.

2.3. Smart Web Developer

A ferramenta CASE Smart Web Developer (SWD) baseia-se nos conceitos da engenharia reversa e padrões da programação orientada a objeto. É uma poderosa ferramenta que aumenta a produtividade do desenvolvimento de produtos de software web, reduzindo assim o tempo de produção de uma nova aplicação de software. Executado diretamente via browser[1], permite o desenvolvimento colaborativo.

[1] Browser – Navegador de internet.

O SWD é essencialmente constituído da tecnologia server-side PHP[2], que independe de seu funcionamento em plataformas distintas dos mais variados sistemas operacionais utilizados no mercado.

Suporta um dos principais sistemas gerenciadores de banco de dados do mercado que é o *Mysql Server*, construindo um código flexível independente. Elaborado dentro dos moldes do padrão de desenvolvimento de produtos de software *Model-View-Control* (MVC). A utilização dos padrões de projetos proporciona a excitação ou redução do re-projeto. A utilização de Padrões de Projeto em sistemas orientados a objetos torna estes sistemas mais flexíveis e reutilizáveis (Gamma et al., 1995).

Entre as vantagens de se utilizar padrões de projeto no desenvolvimento de produtos de software pode-se citar: aumento de produtividade, uniformidade na estrutura do software, incremento da padronização no desenvolvimento de software, aplicação imediata por outros desenvolvedores, redução da complexidade do sistema (Prieto, 2001).

Visando separar o modelo de sua representação, à implementação de SWD é baseada no padrão de projeto MVC[3], conforme ilustra a figura 4.

Figura 4. Componentes do padrão MVC.

Além do padrão de desenvolvimento supracitado, o SWD dispõe das tecnologias XML[4], Javascript[5], CSS[6], bibliotecas mPDF[7] e FPDF[8] que são responsáveis pelos

[2] PHP - é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na internet.
[3] MVC - é um modelo de desenvolvimento de Software, atualmente considerado uma "arquitetura padrão" utilizada na Engenharia de Software.
[4] XML - tem por objetivo trazer flexibilidade as aplicações de software voltadas para a web.

[5] Javascript - é uma linguagem de programação baseada na linguagem de programação ECMAScript padronizada pela Ecma international nas especificações ECMA-262[2] e ISO/IEC 16262 e é atualmente a principal linguagem para programação client-side em navegadores web.
[6] CSS (Cascading Style Sheets) - permite a separação da estrutura lógica da aparência da página.
7 mPDF – é uma classe PHP que auxilia no processo de converter páginas web e gera relatórios no formato PDF.
8 FPDF - é uma biblioteca PHP que auxilia no processo de gerar relatórios voltados para aplicações de softwares.

módulos de relatórios no formato PDF[9] das aplicações que sofrem o processo da engenharia reversa do SWD.

A partir da análise da figura 5, pode-se perceber que o SWD utiliza-se da estrutura do banco de dados Mysql Server, precisamente da sua estrutura denominada Entidade Relacionamento (ER), onde ocorre o processo de engenharia reversa executado pelo SWD, e em seguida é elaborada a aplicação web devidamente codificada.

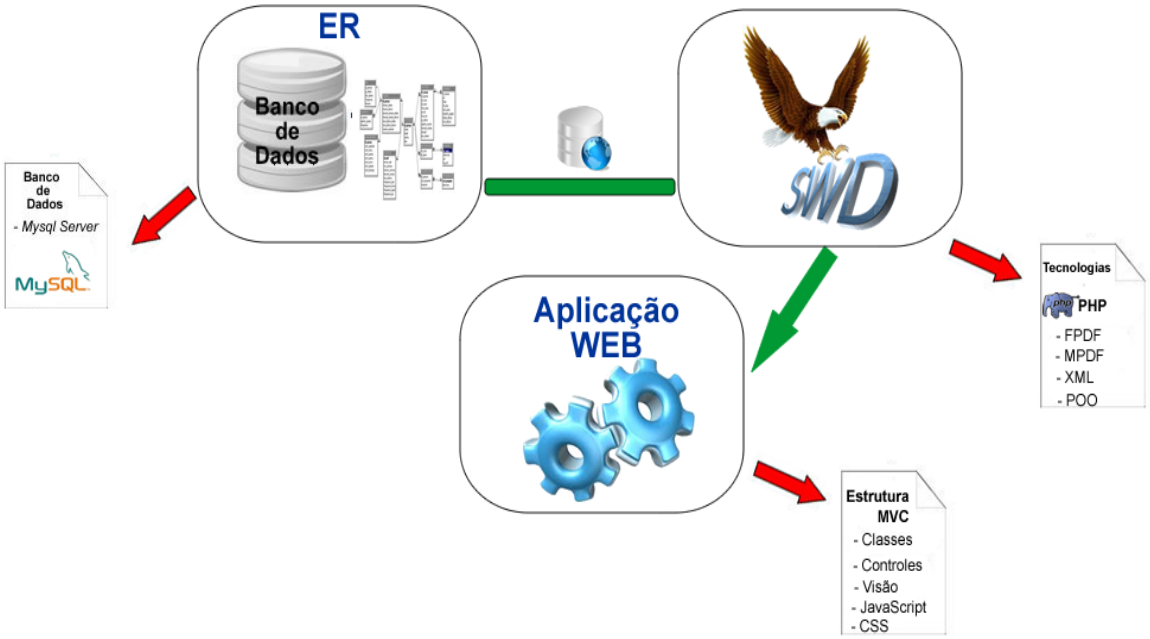

Figura 5. Fluxo de funcionamento do SWD.

A estrutura do SWD, conforme ilustrada na figura 6, segue o padrão de notação, permitindo, portanto, que outros desenvolvedores que estão sendo inseridos no processo de desenvolvimento do mesmo, tenham o entendimento e a clareza quanto à estrutura e a codificação de sua engenharia.

Neste viés, procura-se obter produtos de software voltados para a Web desenvolvidos com a devida qualidade de software requerida, pois de acordo com Boente, Oliveira e Alves (2008), a qualidade de software não pode ser avaliada isoladamente.

No desenvolvimento de produtos de software, um método pobre ou a ausência de uma metodologia pode ser a causa da baixa qualidade. A avaliação da qualidade está diretamente relacionada com a qualidade de processos e metodologias utilizadas no desenvolvimento do produto de software.

9 PDF - é um formato de arquivo, desenvolvido pela Adobe Systems em 1993, para representar documentos de maneira independente do aplicativo, do hardware e do sistema operacional usados para criá-los. Um arquivo PDF pode descrever documentos que contenham texto, gráficos e imagens num formato independente de dispositivo e resolução.

EDU.TEC Revista Científica Digital da Faetec
ISSN: 1984-2007

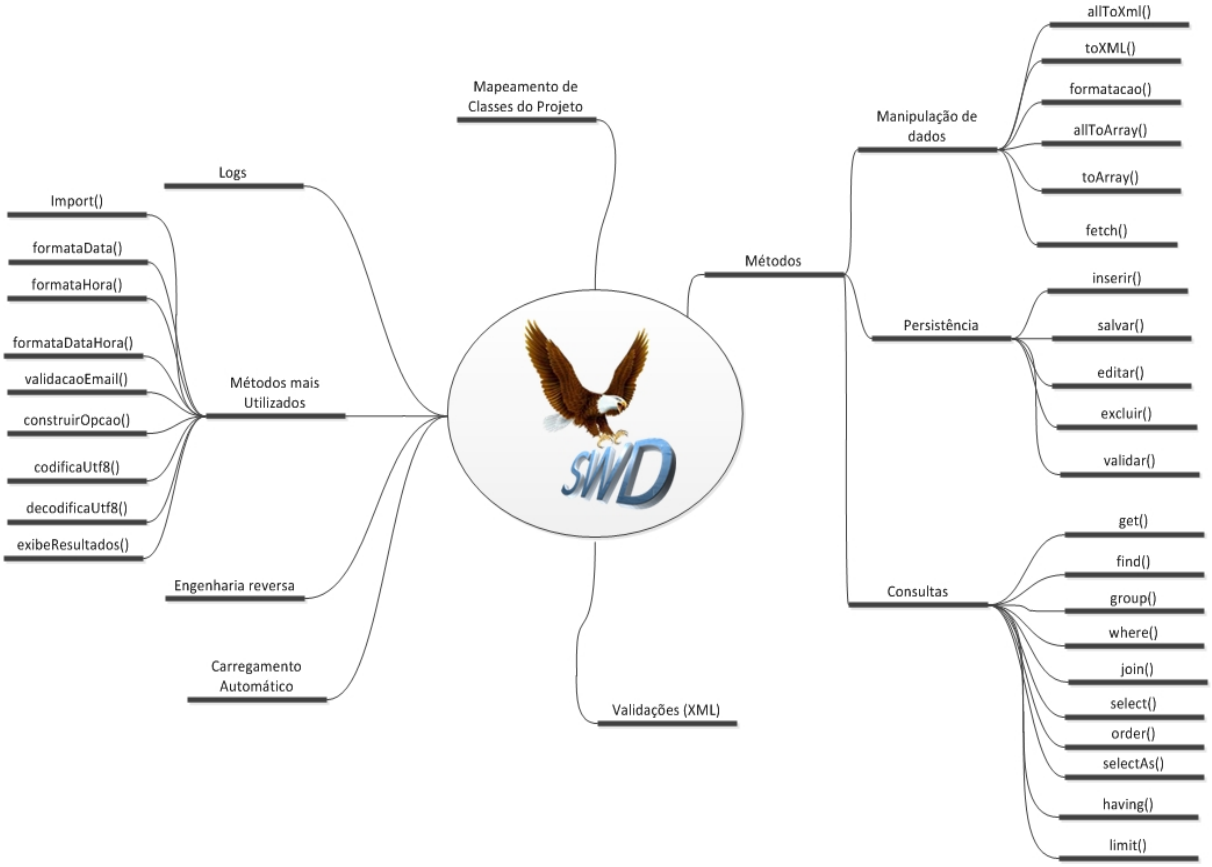

Figura 6. Estrutura do SWD.

3. Metodologia

O desenvolvimento desta pesquisa deu-se, a partir de um levantamento dentro do enfoque teórico bibliográfico, baseando-se em um conhecimento pré-concebido através de informações pesquisadas que estivessem voltadas para área de desenvolvimento de produtos de software.

O presente trabalho apresenta uma proposta de inovação em ferramentas CASE que tenham suporte a tecnologia PHP, através de melhorias de persistência, encapsulamento de dados, tendo como modelo a UML[10] e padrões de desenvolvimento MVC.

Entretanto, para inovação e aperfeiçoamento desta ferramenta CASE, realizaram-se pesquisas literárias, pesquisas comparativas baseadas nas principais ferramentas da atualidade que disponham de tecnologia de código aberto com suporte a PHP. Além de ter um cunho bibliográfico, a pesquisa é caracterizada também como pesquisa de laboratório, pois permite a criação de uma ferramenta baseada na engenharia reversa e engenharia de software.

[10] UML (*Unified Modeling Language*) - permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica.

4. Resultados e Conclusões

Foi desenvolvida a modelagem da ferramenta CASE, Smart Web Developer, baseada na linguagem de modelagem unificada (UML), através dos diagramas de casos de uso, do modelo de classes, do diagrama de sequência, do diagrama de atividade, do diagrama de gráfico de estado e do modelo de implementação.

Neste trabalho estão contemplados os diagramas de casos de uso, diagramas de classes (referente ao modelo de classes do projeto), componentes e implantação (referentes ao modelo de implementação do projeto) e o modelo de entidade relacionamento referente ao mesmo.

A figura 7 ilustra o diagrama de casos de uso, que apresenta o levantamento de requisitos para a elaboração da ferramenta proposta.

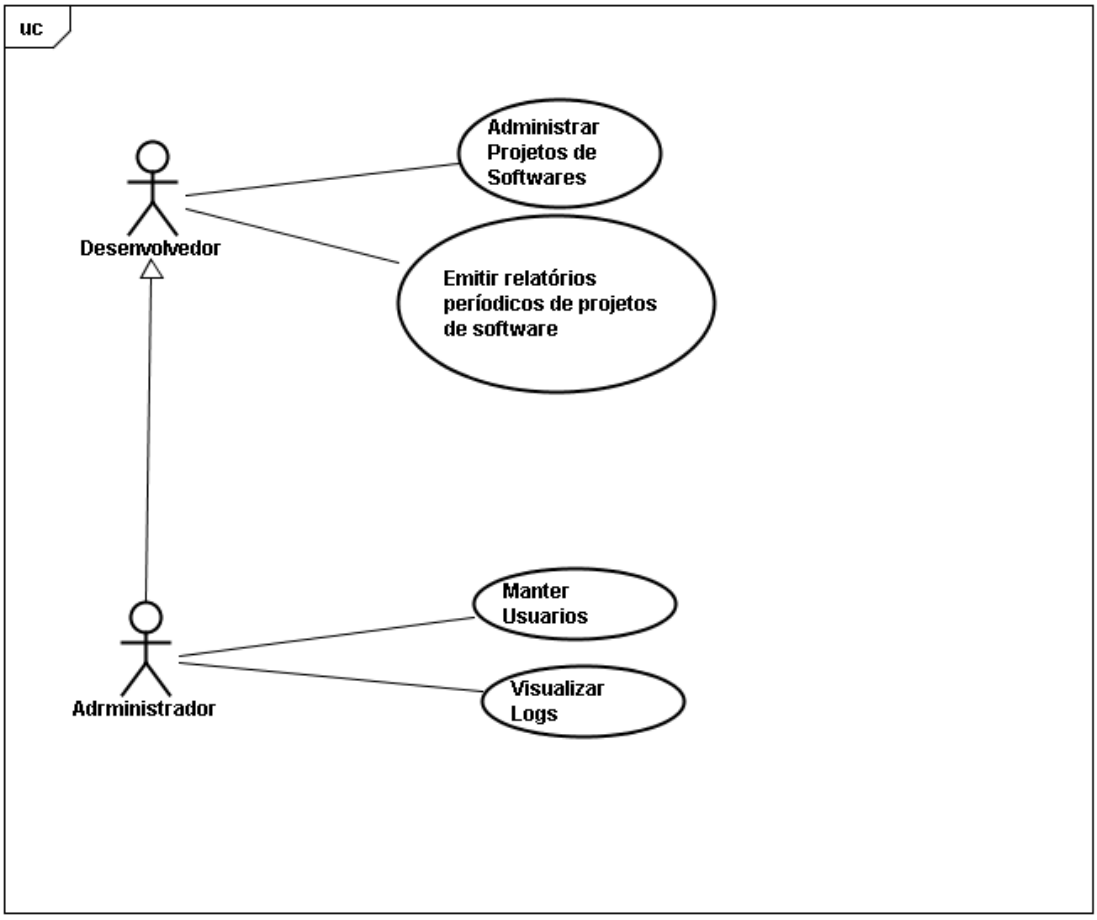

Figura 7. Diagrama de Caso de Uso

EDU.TEC Revista Científica Digital da Faetec
ISSN: 1984-2007

A partir da ilustração da figura 7, pode-se observar os casos de usos da ferramenta SWD, que traduz todo o fluxo de iteração dos atores (Administrador e Desenvolvedor). Com o diagrama de classes (ver ilustração da figura 8) temos uma visão detalhada da estrutura do SWD.

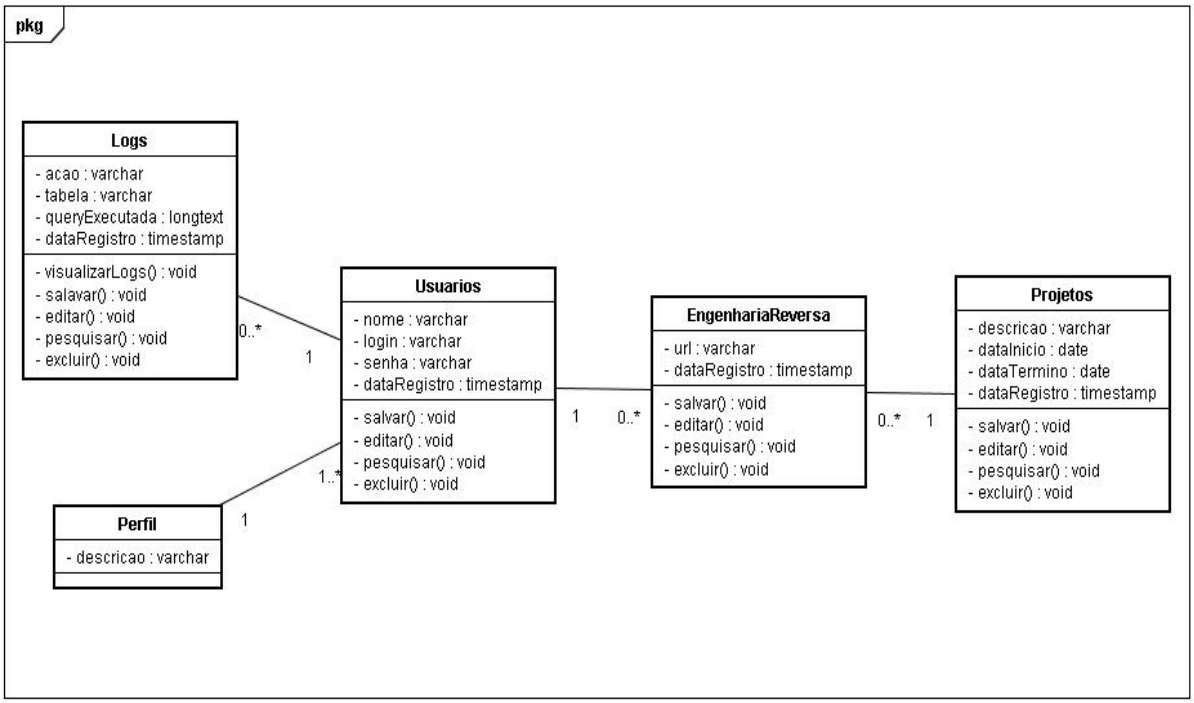

Figura 8. Diagrama de Classes

O modelo lógico representa um conjunto de objetos da realidade modelada sobre os quais se deseja manter informações no banco de dados. A seguir temos a ilustração da figura 9 que mostra a estrutura do banco de dados do SWD.

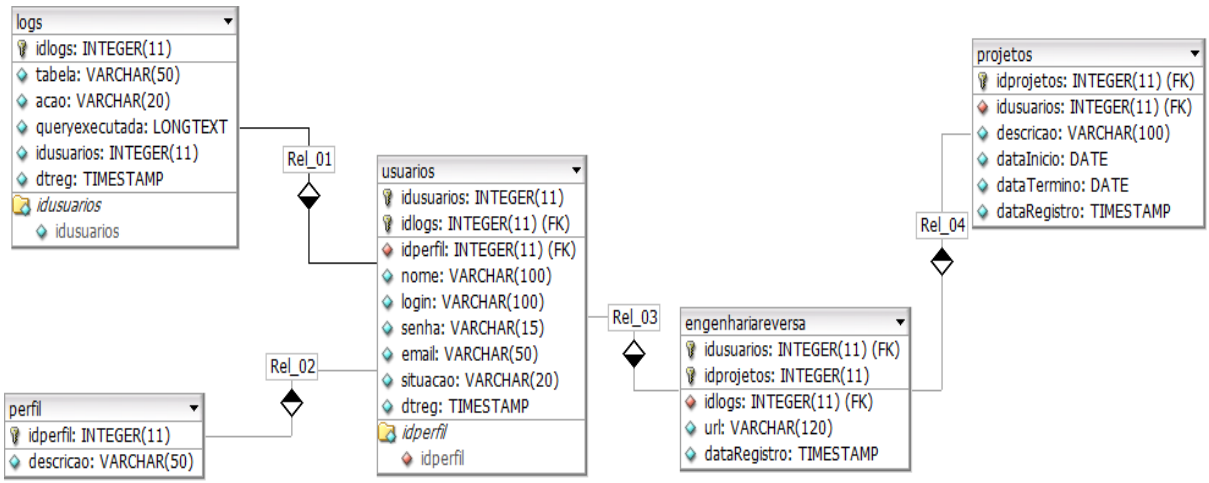

Figura 9. Modelo lógico de dados do SWD.

EDU.TEC Revista Científica Digital da Faetec
ISSN: 1984-2007

A partir daí aplica-se na prática o SWD para realizar tarefa de desenvolvimento de uma aplicação de software web conforme escopo da pesquisa realizada. Embora haja uma definição do escopo da pesquisa, e as ações dos módulos estejam descritos através dos diagramas de caso de uso e diagramas de classes, é necessário que seja visualizado o ambiente em que o SWD irá operar. Este é ilustrado no diagrama de implantação conforme mostra a figura 10.

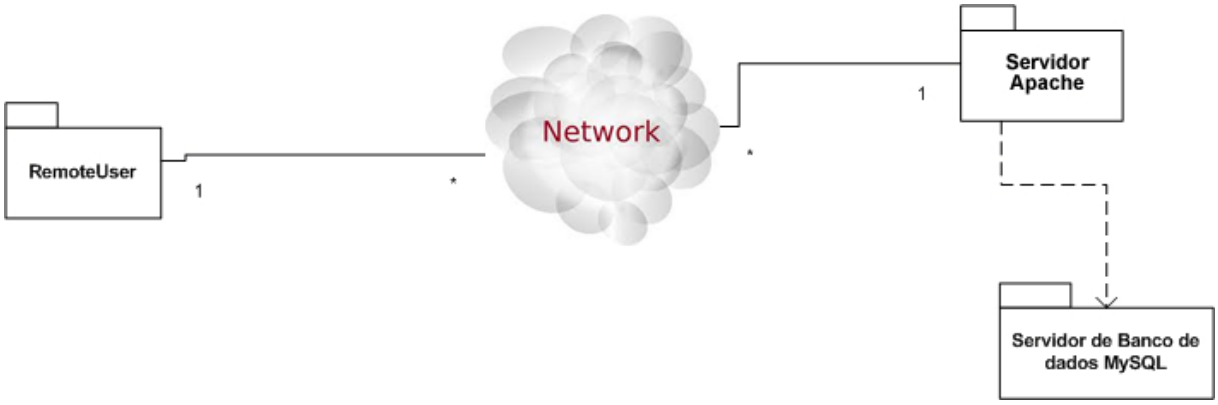

Figura 10. Diagrama de Implantação.

Este artigo apresentou as etapas iniciais de criação e comportamento do Smart Web Developer, ferramenta case baseada nos conceitos da engenharia reversa, engenharia de software e padrões da programação orientada a objeto. Trata-se de uma poderosa ferramenta que aumenta a produtividade do desenvolvimento de software web, reduzindo assim, o tempo de produção de uma nova aplicação de software, permitido o desenvolvimento padronizado e colaborativo.

Os diagramas apresentados modelam o funcionamento do Smart Web Developer, seu fluxo de dados, estrutura, caso de uso e o relacionamento do sistema com os atores e, principalmente o produto a ser gerado.

Hoje o SWD já se encontra em fase de implementação. Estima-se que em dezembro deste ano, sua primeira versão já estará disponível com distribuição gratuita para fim exclusivamente acadêmico. Depois da aprovação dessa versão inicial, a ferramenta case SWD estará pronta para ser distribuída e utilizada com objetivo de aumentar a produção, reduzir o tempo de desenvolvimento e padronização do processo de desenvolvimento de produtos de software.

Referências

BOENTE, A.N.P.; OLIVEIRA, F.S.G.; ALVES, J.C.N. (2008). **RUP como Metodologia de Desenvolvimento de Software para Obtenção da Qualidade de Software**. Anais: V Simpósio de Excelência em Gestão e Tecnologia, out.

GAMMA, E., HELM, R., JOHNSON, R., VLISSIDES, J., (1995). **Design Patterns – Element of Reusable of Object Oriented Software**. Editora Addison-Wesley.

MECENAS, I.; OLIVEIRA, V. (2005). **Qualidade de Software - Uma Metodologia para Homologação de Sistemas**. Rio de Janeiro: Alta Books.

PRESSMAN, R. (2011). **Engenharia de Software**. 8 ed. São Paulo: McGraw-Hill.

PRIETO, G. A., (2001). **Utilização de Padrões de Projeto de Software na Reengenharia de Sistemas**. Dissertação de Mestrado, Centro de Ciências Exatas e de Tecnologia, UFSCar.

QUATRANI, T. (1999). **Visual Modeling with Rational Rose 2000 and UML**. Second Edition. Editora: Addison Wesley.

REKOFF, M. G. Jr. (1985). **IEEE Trans. Systems, Man, and Cybernetics**. Volume março-abril, páginas 244-252.

RESENDE, D.A. (2006). **Engenharia de Software e Sistemas de Informação**. 3 ed. Rio de Janeiro: Brasport.

SCHEFSTRÖM, D. (1989). **Building a Highly Integrated Development Environment Using Preexisting Parts**, IFIP'89 San Francisco, CA, USA, North Holland, (Set.89).

SCHEFSTRÖM, D.; BROEK, G. (1993). **Tool Integration: Environments and Frameworks**, John Wiley & Sons Ltd., Inglaterra.

WONG, K. et al. (2000). **In Anthony Finkelstein**, ed. The Future of Software Engineering. Limerick, Ireland. ACM Press, Páginas 25-34. Páginas 47- 60.