

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**RODRIGO ROSA DA ROCHA**

**ETAPAS DE METODOLOGIA DE DESENVOLVIMENTO DE  
SOFTWARE ÁGIL BASEADA EM PROGRAMAÇÃO EXTREMA**

**CRICIÚMA, JUNHO DE 2008**

**RODRIGO ROSA DA ROCHA**

**ETAPAS DE METODOLOGIA DE DESENVOLVIMENTO DE  
SOFTWARE ÁGIL BASEADA EM PROGRAMAÇÃO EXTREMA**

Trabalho de Conclusão de Curso apresentado  
para obtenção do Grau de Bacharel em Ciência da  
Computação da Universidade do Extremo Sul  
Catarinense.

Orientadora: Prof<sup>a</sup>. MSc. Ana Cláudia Garcia  
Barbosa.

**CRICIÚMA, JUNHO DE 2008**

Dedico este trabalho aos meus familiares,  
namorada e amigos, que fazem parte da  
minha vida e do meu dia-a-dia.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, que me deu o dom da vida, que está presente em todos os momentos de minha vida e que me deu forças para elaborar este trabalho.

Também devo agradecer aos meus pais, Valdir e Arlete, que me trouxeram ao mundo, me ensinaram a viver com dignidade e me deram a oportunidade de desfrutar de muitos momentos especiais.

À minha irmã Morgana que está fora do país, mas dentro do coração. Agradeço também a minha namorada Ana Paula, que me deu forças, e me aconselhou na hora certa, enfim, contribuiu para o meu êxito; compreendendo minhas ausências, compartilhando meus ideais e incentivando-me sempre.

Aos meus amigos, sem distinção, que participam da minha vida e que tem um lugar especial em meu coração! Valeu pessoal!

Aos mestres, em especial a minha orientadora Ana Cláudia Garcia Barbosa , que me acompanharam durante este tempo que estive na universidade.

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo”.

José de Alencar

## RESUMO

Este trabalho apresenta uma metodologia ágil para desenvolvimento de software, que ainda é pouco utilizada na engenharia de software. Esta metodologia ágil proporciona uma maior agilidade na realização do projeto, visando um contato mais próximo ao cliente e diminuindo a burocracia necessária em metodologias tradicionais. A metodologia ágil aplicada neste trabalho foi a Programação Extrema (XP). A aplicação trata um protótipo de um *site* com ambiente virtual para a instituição de ensino Colégio Cocal. O *site* com ambiente foi desenvolvido, após estudos dos conceitos do relacionamento entre colégio e aluno, além de várias etapas da XP, que foram possíveis ser aplicadas. Ao final do estudo, pode-se observar que a XP como metodologia ágil pode tornar o desenvolvimento de projeto mais interativo, diminuindo a distância entre cliente e desenvolvedor, objetivando o *feedback* rápido, reduzindo custos com mudanças no decorrer do projeto. Com a aplicação de etapas de XP no desenvolvimento do projeto notou-se que realmente a diferença básica entre as metodologias tradicionais e ágeis se dá na maneira como o cliente participa do desenvolvimento, aumentando as chances do produto final estar de acordo com as expectativas do cliente.

**Palavras-chaves:** Engenharia de Software, Gerenciamento de Projeto, Metodologia Ágil, Programação Extrema (XP).

## ABSTRACT

This present academic work shows an agile methodology for software development, that is still not much used in software engineering. This methodology provide a best agility on project development process, reaching a closer contact to the client and reducing the bureaucracy necessary in traditional methodologies. The agile methodology applied on this work was Extreme Programming (XP). The application handles a website prototype with a virtual environment interface for the learning institution Colégio Cocal. The website's environment was developed after an intense analysis about the concepts of the student-institution relationship, also many XP stages that were possibly applied on it. At the final of this analysis, was observed that the XP, as an agile methodology, can make the project development process more interactive, reducing the distance between client and developer, objectifying a fast feedback, reducing costs with alterations during the development process. Applying stages of XP on the project development, make possible to notice that the basic difference between traditional and agile methodologies is the way how the client takes part in the development process, increasing the chances of getting a final product that is according with the client expectancies.

**Key words:** Software Engineering, Project Management, Agile Methodology, Extreme Programming (XP).

## LISTA DE ILUSTRAÇÕES

Figura 1. Modelo Cascata .....	12
Figura 2. Modelo Prototipação.....	13
Figura 3. Modelo Espiral.....	14
Figura 4. Custo de modificações na engenharia de software tradicional.....	20
Figura 5. Custo de modificações na programação extrema.....	20
Figura 6. Cartão de história do cliente.....	51
Figura 7. Etapas do planejamento.....	51
Figura 8. Cartão de história preenchido pelo cliente.....	52
Figura 9. Cartão de tarefa.....	53
Figura 10. Mural de tarefas.....	53
Figura 11. Tela <i>login</i> e senha do ambiente virtual.....	54
Figura 12. Página principal do <i>site</i> .....	55
Figura 13. Código de sessão do <i>site</i> .....	56

## LISTA DE SIGLAS

**HTML**     *HyperText Markup Language*

**MA**        **Metodologia Ágil**

**PHP**       *Hipertext PreProcessor*

**SQL**       *Structured Query Language*

**USP**       **Universidade de São Paulo**

**XP**        *Extreme Programming*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 OBJETIVO GERAL.....	14
1.2 OBJETIVO ESPECÍFICO.....	14
1.3 JUSTIFICATIVA.....	14
1.4 ESTRUTURA DO TRABALHO.....	15
<b>2 ENGENHARIA DE SOFTWARE TRADICIONAL.....</b>	<b>17</b>
2.1 MODELO CASCATA.....	20
2.2 PROTOTIPAÇÃO .....	21
2.3 MODELO ESPIRAL.....	22
<b>3 MODELAGEM ÁGIL.....</b>	<b>24</b>
3.1 MODELADORES ÁGEIS.....	26
3.2 MODELO ÁGIL.....	26
<b>4 PROGRAMAÇÃO EXTREMA.....</b>	<b>28</b>
4.1 VALORES DA PROGRAMAÇÃO EXTREMA.....	31
<b>4.1.1 Feedback.....</b>	<b>31</b>
<b>4.1.2 Comunicação.....</b>	<b>32</b>
<b>4.1.3 Simplicidade.....</b>	<b>32</b>
<b>4.1.4 Coragem.....</b>	<b>33</b>
4.2 CICLO DE VIDA DA PROGRAMAÇÃO EXTREMA.....	35
<b>4.2.1 Exploração.....</b>	<b>36</b>
<b>4.2.2 Planejamento.....</b>	<b>36</b>
<b>4.2.3 Iterações para liberação.....</b>	<b>36</b>
<b>4.2.4 Produção.....</b>	<b>37</b>
<b>4.2.5 Manutenção.....</b>	<b>37</b>

<b>4.2.6 Morte.....</b>	<b>38</b>
4.3 PAPÉIS DA PROGRAMAÇÃO EXTREMA.....	38
4.4 QUANDO NÃO USAR A PROGRAMAÇÃO EXTREMA.....	39
<b>5 TRABALHOS CORRELATOS.....</b>	<b>40</b>
5.1 AVALIAÇÃO DE PROCESSOS DE SOFTWARE: UM ESTUDO DE CASO EM XP E ICONIX.....	40
5.2 CARACTERIZAÇÃO DE MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE.....	40
5.3 USO EFICAZ DE MÉTRICAS EM MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE.....	41
5.4 ESTUDO ANALÍTICO DA ENGENHARIA DE REQUISITOS ATUAL E A DE PROCESSOS ÁGEIS (XP).....	41
<b>6 ETAPAS DE METODOLOGIA ÁGIL, UTILIZANDAS NA CONSTRUÇÃO DO SITE COM AMBIENTE VIRTUAL.....</b>	<b>43</b>
6.1 DEFININDO PARTES ENVOLVIDAS.....	44
6.2 VALORES APLICADOS.....	45
<b>6.2.1 Feedback.....</b>	<b>45</b>
<b>6.2.2 Comunicação.....</b>	<b>46</b>
<b>6.1.3 Simplicidade.....</b>	<b>46</b>
<b>6.1.4 Coragem.....</b>	<b>46</b>
6.3 CICLO DE VIDA.....	47
<b>6.3.1 Fase de Exploração.....</b>	<b>47</b>
<b>6.3.2 Fase de Planejamento.....</b>	<b>49</b>
<b>6.3.3 Fase de Iterações para liberação.....</b>	<b>50</b>
<b>6.3.4 Fase de Produção.....</b>	<b>52</b>
<b>6.3.5 Fase de Manutenção.....</b>	<b>52</b>
<b>6.3.6 Fase de Morte.....</b>	<b>53</b>
6.4 REFATORAÇÃO.....	54

CONCLUSÃO.....	57
REFERÊNCIAS.....	61
BIBLIOGRAFIA COMPLEMENTAR.....	62
APÊNDICE A - DIAGRAMA CASO DE USO: VISITANTE.....	63
APÊNDICE B - DIAGRAMA CASO DE USO: ADMINISTRADOR.....	64
APÊNDICE C - DIAGRAMA CASO DE USO: ALUNO.....	65
APÊNDICE D - DIAGRAMA CASO DE USO: PROFESSORES.....	66
APÊNDICE E - <i>INTERFACE DO SITE LINK PROFESSORES</i> .....	67
APÊNDICE F - <i>INTERFACE DO SITE LINK TURMAS</i> .....	68
APÊNDICE G - <i>INTERFACE DO SITE LINK CONTATO</i> .....	69

# 1 INTRODUÇÃO

Devido ao avanço tecnológico ocorrido nos últimos anos, notou-se a necessidade das empresas se informatizarem, isso porque, uma empresa que atualmente não é informatizada perde muito em tempo, não conseguindo acompanhar seus concorrentes.

Visando este mercado competitivo as fábricas de softwares estão sempre se atualizando para se sustentar no mercado e melhor atender seus clientes.

Segundo Pressman (2006) há algum tempo o software superou o hardware em importância computacional. Sendo usado para dirigir um negócio, controlar um produto ou capacitar um sistema, oferecendo maiores oportunidades, o que diferencia uma empresa no mercado.

As empresas fabricantes de softwares contam com diferentes formas para construí-los. A forma de desenvolvimento mais usada atualmente é a baseada na engenharia de software tradicional ou clássica. Esta utiliza como base para a construção, alguns estilos de processos de softwares, sendo que os mais conhecidos e aplicados são o processo em cascata e em espiral (TELES, 2006).

Conforme Braude (2005) o processo em cascata é lógico, relativamente fácil de entender e gerenciar. Entretanto, ele é mais um ideal ou uma linha-base do que um processo que expresse a realidade. Normalmente, este processo de software avança pelas fases em cascata de forma linear. O mesmo não ocorre no processo espiral, pois neste processo a linha de trabalho é modificada várias vezes, e cada passagem (iteração) pela cascata se produz um produto intermediário mais capaz do que o anterior, até que o sistema final seja produzido.

Apesar da engenharia de software tradicional implementar práticas de desenvolvimento organizadas e documentadas, criou-se um novo método de

desenvolvimento que identifica os problemas existentes na aplicação dos processos convencionais, tais como: a complexidade, dificuldade de manutenção, alto custo, diferenças entre o produto desenvolvido e as necessidades do usuário. Diante destes problemas, originaram-se soluções possíveis, que consistem em métodos conhecidos como modelagem ágil (BONA, 2002).

Segundo Ambler (2004) a Modelagem Ágil (MA) é uma metodologia baseada na prática para modelagem e documentação eficazes de sistemas baseados em software, ou seja, métodos implementados através do computador, que possuam um processo com um conjunto de práticas guiadas por princípios e valores para profissionais de software aplicar no seu dia-a-dia.

Várias metodologias ágeis podem ser citadas, mas as de maior visibilidade hoje, são os processos baseados na Programação Extrema(XP) e no Scrum. Sanches (2007) escreve que Scrum é uma metodologia ágil para gerência de projetos. Ela é baseada em ciclos de trinta dias chamados *Sprints*, onde se trabalha para alcançar objetivos bem definidos. Estes objetivos são representados no *Product Backlog*, uma lista de tarefas a fazer, que é constantemente atualizada e repriorizada.

A programação extrema, que terá maior foco neste trabalho, é uma metodologia ágil voltada para equipes de pequeno e médio porte, que desenvolvem sistemas que estão em constante mudança, mas sempre de forma científica. Alguns pontos diferenciam a programação extrema das outras metodologias, como: *feedback* rápido, simplicidade presumida, mudanças incrementais, aceitação das mudanças e alta qualidade (BECK, 2004).

Neste contexto, a problemática em questão, baseia-se na necessidade de se aplicar etapas de metodologia ágil, baseado em programação extrema, disponibilizando para empresas de pequeno e médio porte do ramo de desenvolvimento de software uma

avaliação sobre esta metodologia que poderão utilizar na escolha de um método adequado para aplicarem na criação de seus sistemas.

Como forma de obter resultados para a avaliação do trabalho, serão aplicadas algumas etapas da metodologia ágil em um processo de desenvolvimento de um *site* com ambiente virtual para um colégio particular.

## 1.1 OBJETIVO GERAL

Avaliar etapas de metodologia de desenvolvimento de softwares ágeis, baseada em programação extrema, para empresas de pequeno e médio porte.

## 1.2 OBJETIVOS ESPECÍFICOS

Esta pesquisa tem como objetivos específicos os itens relacionados abaixo:

- a) analisar os principais processos de desenvolvimento de software;
- b) compreender a aplicação de modelagens ágeis;
- c) entender a Programação Extrema (XP);
- d) identificar as vantagens e desvantagens da modelagem ágil por meio da programação extrema.

## 1.3 JUSTIFICATIVA

A situação atual de alguns processos de desenvolvimento de softwares está abaixo de um nível satisfatório. Os problemas estão se tornando cada vez mais constantes. Isso ocorre por algumas razões, entre elas podem-se citar os constantes

atrasos na entrega dos softwares, assim como gastos acima dos programados no decorrer do processo, entre outros (AMBLER, 2004).

Segundo Pressman (2006) a coleta de dados sobre o processo de desenvolvimento de software é baixa, por isto não se têm uma bibliografia confiável sobre a eficácia de novas ferramentas, métodos e padrões.

Os métodos ágeis surgiram como tentativa de melhorar o desenvolvimento de softwares, algumas novas metodologias que os aplicam foram criadas, entre elas a Programação Extrema.

A Programação Extrema possui algumas práticas que já estão sendo usadas por muitos desenvolvedores de softwares, isso porque nenhuma das idéias da XP é nova, suas técnicas foram comprovadas por décadas. As principais inovações da XP consistem em aplicar práticas tradicionais agrupadas, garantindo que elas sejam realmente praticadas (BECK, 2004).

Nota-se que estas metodologias de desenvolvimento de software ainda necessitam de maiores estudos, para que sejam melhores utilizadas. Diante das situações expostas, justifica-se a realização desse estudo que envolve as principais metodologias aplicadas na elaboração de softwares. E para melhor avaliação será produzida uma aplicação utilizando etapas de programação extrema que é uma metodologia ágil, em um projeto para assim poder obter parâmetros que permitam levantar pontos positivos e negativos desta metodologia diante das ditas tradicionais.

#### 1.4 ESTRUTURA DO TRABALHO

O presente trabalho de pesquisa tem como objetivo o estudo da metodologia ágil de desenvolvimento de software denominada Programação Extrema (XP), verificando a funcionalidade de suas etapas para aplicação em um projeto de construção

de um *site* com ambiente virtual para uma entidade educacional. Para tanto, o referente trabalho está disposto em seis capítulos, os quais incluem a introdução, do desenvolvimento trabalho e a conclusão.

No segundo capítulo descreve-se a engenharia de software tradicional assim como suas principais metodologias de desenvolvimento, demonstrando basicamente seu funcionamento. O terceiro capítulo tem como objetivo explicar a metodologia ágil que é um método de desenvolvimento relativamente novo dentro da engenharia de software.

O quarto capítulo apresenta de forma mais detalhada a Programação Extrema, que se trata de uma metodologia ágil com maior visibilidade no mercado de produção de softwares, relatando seu funcionamento, os valores por ela defendidos, seu ciclo de vida e papéis realizados por seus usuários, e também quando não se deve usar esta metodologia. No capítulo seguinte é feito um breve relato sobre os trabalhos correlatos do assunto por este tratado.

O sexto e último capítulo, trata da aplicação do trabalho desenvolvido onde foi criado um *site* com ambiente virtual aplicando etapas de metodologia ágil de Programação Extrema, gerando assim os resultados para avaliar esta aplicação. Logo a seguir têm-se os apêndices do trabalho mostrando os casos de uso da aplicação.

## **2 ENGENHARIA DE SOFTWARE TRADICIONAL**

Atualmente a informática é importante para pessoas e empresas. Isso acontece porque com a evolução que a informática apresentou nas últimas décadas, tornou-se cada vez mais necessário seu uso para manter as pessoas e empresas evoluindo e usando ao máximo a tecnologia oferecida.

Segundo Pressman (2006) o software já há algum tempo ultrapassou o hardware. Isso porque o software hoje já consegue extrair muito do potencial do hardware. Agora a principal meta das empresas fabricantes de software é melhorar a qualidade e reduzir os custos.

As empresas que produzem softwares estão relacionadas com a tecnologia de informação e são hoje um dos principais seguimentos que mantém o continuo crescimento da economia mundial. Apesar deste crescimento a construção de softwares passa por uma série de reformulações nas técnicas e metodologias usadas para seu desenvolvimento.

A engenharia de software engloba todos estes processos de produção de software, é a partir dela que se pode produzir com uma maior qualidade, confiabilidade sempre visando menor custo, usando menos tempo.

O software está passando por uma série de problemas e já é possível prever uma crise na área em um horizonte aproximado. Esta crise está sendo criada por uma falta de planejamento por parte dos gerentes e desenvolvedores que não acompanharam o desenvolvimento dos computadores. Os principais problemas levantados pelos próprios gerentes de projeto de desenvolvimento de software estão relacionados a estimativas erradas de custo para a construção de um software, assim também como datas de entregas imprecisas e variação de produtividade dos desenvolvedores (PETERS; WITOLD, 2001).

Estes problemas levantados causam uma série de novas dificuldades para a construção de softwares como:

- a) falta de tempo dedicado para a coleta de dados sobre o processo de desenvolvimento;
- b) insatisfação do cliente com o sistema entregue;
- c) falta de comunicação entre os desenvolvedores e clientes;
- d) qualidade do software suspeito;
- e) manutenção cara do sistema.

Problemas assim são difíceis de solucionar, mas o primeiro passo já foi tomado que é o reconhecimento dos problemas, agora é localizar as possíveis causas para encontrar as soluções.

A engenharia de software se baseia em três elementos fundamentais: métodos, ferramentas e procedimentos. Estes elementos ajudam os profissionais na construção de software de alta qualidade minimizando os problemas já citados (PRESSMAN, 2006).

São os métodos da engenharia de software que introduzem critérios para se iniciar o processo de criação de um software. Os métodos possuem tarefas como o planejamento, estimativa de projeto, análise de requisitos, projeto de estrutura de dados, arquitetura de programa de algoritmo de processamento, codificação, teste e manutenção.

As ferramentas têm como principal função dar auxílio aos métodos já citados. A partir do momento que há informações criadas pela integração de ferramentas diferentes, diz-se que está estabelecido um sistema de suporte ao desenvolvimento, também chamado de ferramenta Computer Aided Software Engineering (CASE). O CASE combina software, hardware e o banco de dados, contendo informações sobre análise, projeto, codificação e testes (PRESSMAN, 2006).

Os procedimentos da engenharia de softwares são responsáveis pela ligação entre os métodos e as ferramentas, possibilitando o desenvolvimento ordenado do software.

O conjunto de métodos, ferramentas e procedimentos são chamados na engenharia de software de paradigmas ou processo de desenvolvimento de software.

Sommerville (2003) diz que apesar dos processos de desenvolvimento de softwares serem capazes de produzir todo o software, na maioria das vezes é usado para modificar e expandir sistemas já existentes.

Algumas atividades do processo de software são consideradas fundamentais, sendo assim comum em todos os processos. São eles:

- a) projeto e implementação de software;
- b) validação de software;
- c) evolução de software.

Na verdade não existe um processo de desenvolvimento de software ideal, mas sim métodos de trabalho para melhorar as empresas que os utilizam.

Uma forma de se extrair melhores resultados dos processos de software é fazer uma padronização dos softwares utilizados pela empresa, diminuindo o tempo de treinamento fazendo que o processo utilizado seja mais econômico.

A engenharia de software tradicional possui alguns modelos de processos já padronizados. Os processos que são mais amplamente utilizados por empresas de pequeno e médio porte são os processos em cascata, prototipação e espiral (SOMMERVILLE, 2003).

## 2.1 MODELO CASCATA

Este processo de desenvolvimento de software trabalha de forma seqüencial, e evolui por meio da análise, projeto, codificação, teste e manutenção. O modelo em cascata envolve algumas atividades, como:

- a) análise e engenharia de sistema;
- b) análise de requisitos de software;
- c) projeto;
- d) codificação;
- e) teste;
- f) manutenção.

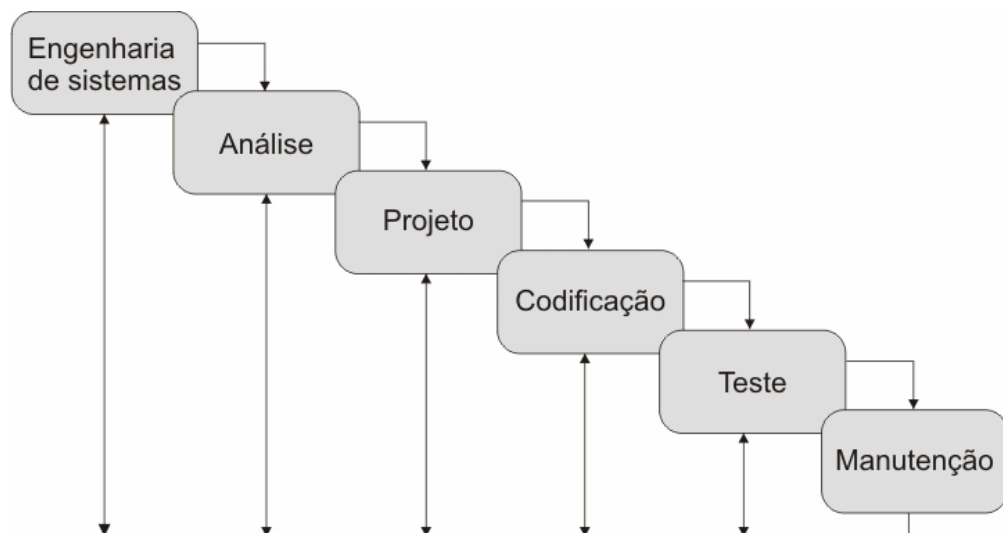


Figura 1. Modelo cascata  
Fonte: PRESSMAN, R. S. (2006)

O modelo de processo em cascata é o mais antigo e mais utilizado da engenharia de software. É usado em projetos de grande porte e quando não se sabe as funcionalidades e requisitos que o sistema terá de forma detalhada. Apesar de sua grande utilização alguns pontos negativos estão sendo levantados a respeito desse

modelo, como: os projetos nem sempre ocorrem de forma sequencial, pois problemas com iterações ocorrem a qualquer momento. A obtenção de informações passadas pelo cliente as vezes não são facilmente entendidas, dificultando a aplicação desse modelo. Neste modelo o resultado ou a entrega da primeira versão acontece na parte final do projeto, se um erro muito grave for encontrado na parte final do projeto pode colocar em risco todo o sistema (PFLEEGER, 2004).

Apesar dessas falhas encontradas, este modelo de desenvolvimento de software é tido como base por vários outros, pois tem padrões bem definidos.

## 2.2 PROTOTIPAÇÃO

O processo de prototipação é uma forma utilizada quando se necessita de um modelo do software que será implementado. Este modelo pode ser produzido de três formas diferentes:

- a) protótipo em papel, que mostra a interação homem-máquina, facilitando o entendimento do usuário;
- b) protótipo do trabalho a ser feito, implementando uma função exigida do software;
- c) protótipo de um programa existente, este executa parte ou total das funções desejadas, possuindo funções que ainda serão melhoradas.

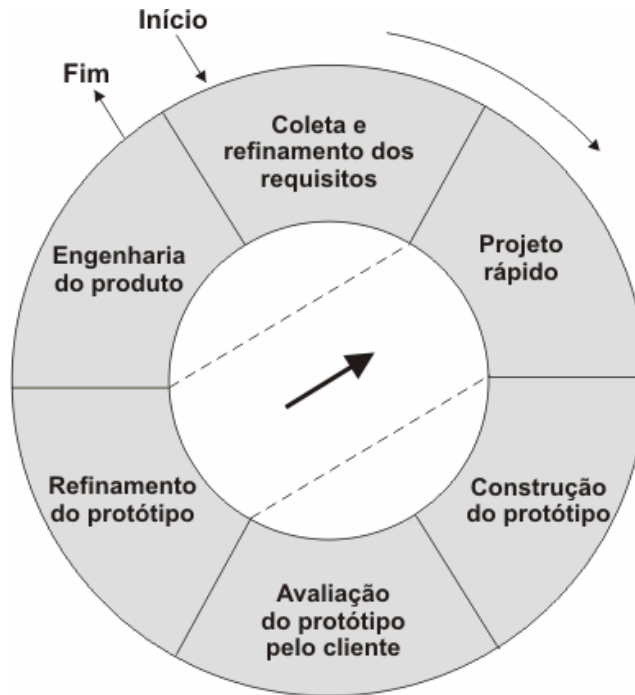


Figura 2. Modelo prototipação  
 Fonte: PRESSMAN, R. S. (2006)

Como mostra a Figura 2 este modelo contém basicamente todo o processo de construção de um software completo, mas com a diferença que todos seus passos são feitos de maneira mais superficial, sem suas funcionalidades completas.

O processo de prototipação serve como uma espécie de filtro para os requisitos do sistema, geralmente este primeiro protótipo produzido é descartado, com a geração do software real (PRESSMAN, 2006).

### 2.3 MODELO ESPIRAL

O processo de espiral é na verdade a combinação entre o modelo em cascata e a prototipação acrescentando a ele a análise de riscos. Este modelo é dividido em quatro partes: planejamento, análise de riscos, engenharia e avaliação feita pelo cliente. Este modelo é mais bem entendido na Figura 3, onde se mostra que a evolução na figura ocorre de dentro para fora.

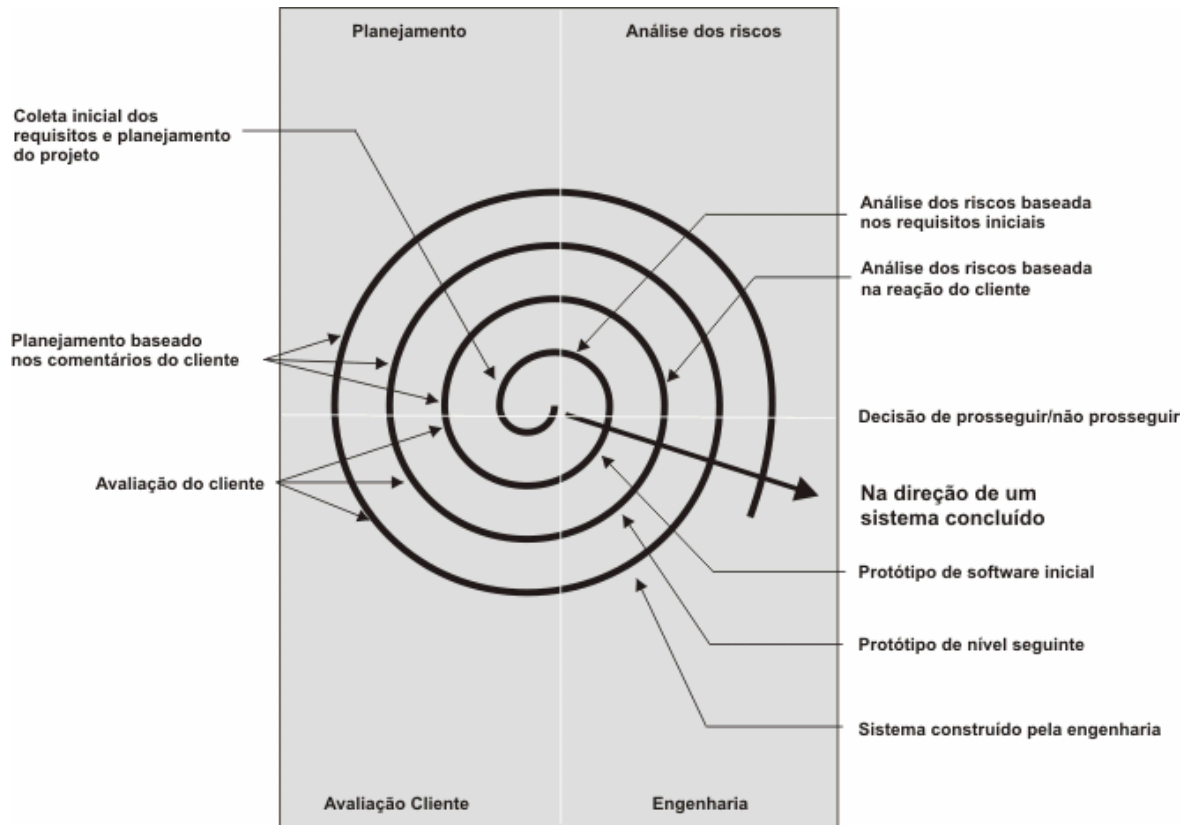


Figura 3. Modelo espiral  
 Fonte: PRESSMAN, R. S. (2006)

O modelo espiral é o mais realista dentro da engenharia de software. Ele usa uma abordagem evolucionária. Até mesmo por ser um modelo relativamente novo, ainda não é muito utilizado, precisando de mais tempo de aplicação para sua consolidação (PRESSMAN, 2006).

### 3 MODELAGEM ÁGIL

A modelagem ágil é uma metodologia de construção de softwares, ela firma seus conceitos em princípios úteis para o trabalho dos profissionais de software. Ao contrário dos outros modelos de software que trabalha de forma prescritiva a modelagem ágil funciona por meio de conselhos para que a programação se torne mais eficiente (AMBLER, 2004).

Seus princípios são baseados no manifesto Agile Alliance que foi criado por 17 metodologistas que formam a Agile Software Development. Este manifesto foi criado para aumentar os meios estruturados de desenvolvimento ágil.

Assim, como em todos os modelos de processo de desenvolvimento de software, os métodos ágeis também possuem alguns princípios, dentre os quais se destacam:

- a) a prioridade é satisfazer o cliente por meio da entrega contínua e rápida de uma versão do software com valor agregado;
- b) alterações sobre os requisitos são bem vindas , mesmo que ela ocorra tarde, pois isto dá uma vantagem competitiva ao cliente;
- c) entrega de software freqüente, a cada semana ou a cada mês, sempre visando o menor tempo possível;
- d) os especialistas no negócio e os desenvolvedores trabalham juntos diariamente no projeto;
- e) manter uma equipe motivada fornecendo ambiente e confiança necessários para o trabalho a ser executado;
- f) a maneira mais eficiente de comunicação da equipe é através de uma conversa face-a-face;
- g) a simplicidade é essencial;
- h) software funcionando é a medida primária de progresso;

- i) a atenção contínua a excelência técnica e ao bom projeto melhoram a agilidade;
- k) as melhores arquiteturas , requisitos e projetos emergem de equipes auto-organizadas;
- l) a equipe reflete, a intervalos regulares, sobre como se tornar mais efetivo, então se ajusta e otimiza o seu comportamento;
- m) promover desenvolvimento sustentável. Todos *stakeholders* devem ser capazes de manter um ritmo constante indefinidamente.

A modelagem ágil fornece novas mudanças na maneira de desenvolver software dentre as quais as mais usadas são:

- a) indivíduos e iterações valem mais que processos e ferramentas;
- b) um software funcionando vale mais que documentação extensa;
- c) a colaboração do cliente vale mais que negociação de contrato;
- d) responder as mudanças vale mais que seguir um plano estático;

Segundo Ambler (2004), a modelagem ágil (MA) adota valores usados na programação extrema, que é uma metodologia ágil. Estes valores são: comunicação, simplicidade, *feedback* e coragem, mas além desses, a MA aplica um quinto valor que julga ser um dos mais importantes que é a humildade.

A comunicação é importante, pois por meio dela as pessoas envolvidas no projeto conseguem um maior entrosamento. Muitas soluções simples são mais eficazes e atendem os requisitos do sistema, facilitando o *feedback* do cliente com o sistema. A coragem é usada na tomada de decisões e a humildade para aceitar que nunca se sabe tudo, que outras pessoas podem acrescentar maiores conhecimentos ajudando a desenvolver o sistema (AMBLER, 2004).

### 3.1 MODELADORES ÁGEIS

De acordo com Ambler (2004) os modeladores ágeis são todos que aplicam e seguem os valores da modelagem ágil, isso na fase do projeto, os que usam de métodos ágeis em todo o desenvolvimento do sistema são chamados de desenvolvedores ágeis.

### 3.2 MODELO ÁGIL

Segundo Ambler (2004) o modelo ágil é tido como mais compreensivo, para os clientes, mais simples, suficiente, preciso e consistente, que pode ajudar a valorizar o processo de construção de um bom projeto. O modelo ágil possui alguns traços que são bem próprios de sua aplicação.

A modelagem é compreendida por todos os envolvidos no projeto, isto é importante, pois tanto usuários como desenvolvedores, precisam compreender os assuntos para com o qual respondem, e um modelo que possui uma maior simplicidade, facilita e muito todo o trabalho (AMBLER, 2004).

A maioria dos projetos não consegue alcançar 100% de precisão, mesmo assim os projetos não são descartados, pois a urgência do cliente faz com que ele aceite o sistema mesmo que ele não esteja perfeito. A modelagem ágil procura chegar à perfeição mesmo sabendo que isso na maioria dos projetos é impossível. Os modelos ágeis são suficientemente consistentes e detalhados, se adequando à necessidade de cada projeto (AMBLER, 2004).

Um modelo para ser considerado satisfatório necessita que ele agregue valor positivo ao projeto, e este é um benefício do uso da MA. Assim, a MA consegue realizar as atividades para a qual foi criada.

Os métodos ágeis podem ser considerados novos no contexto da engenharia de software, pois datam de 2001 e são ainda mais recentes no Brasil. Porém, várias iniciativas estão sendo desenvolvidas numa tentativa de consolidar estes métodos na área da engenharia de software. Um exemplo são as universidades renomadas no Brasil como a Universidade de São Paulo (USP) que mantém grupos de estudo desenvolvendo a metodologia ágil, e em maior escala sobre a metodologia XP (programação extrema).

## 4 PROGRAMAÇÃO EXTREMA

A Programação Extrema (XP) foi desenvolvida por Kent Beck, dono e presidente da First Class Software Inc, onde seus dois maiores interesses são *patterns* (padrões) e Programação Extrema. A XP foi concebida a partir da idéia que desenvolver software é difícil, e desenvolver software de qualidade no prazo combinado é muito mais. Para funcionar, é preciso o uso disciplinado de práticas-modelo adicionais (TELES, 2006).

A XP foi criada para suprir os problemas que foram levantados por programadores que trabalham com o método tradicional que é geralmente maçante e de longa duração.

Uma das principais características da XP é a agilidade gerada no projeto e programação do sistema. Ele trata os projetos de maneira mais eficiente, diminuindo os riscos, mas isso tudo sem deixar de lado a parte científica dos projetos. A XP está diretamente relacionada com a tecnologia de informação, preocupando-se mais com as necessidades do cliente, que na maioria das vezes necessita de resultados mais rápidos (TELES, 2006).

“*Extreme* por que enfatiza o uso extremo de práticas que se mostram funcionais, como revisões de código, teste, simplicidade, ciclos curtos e iterativos.” (BECK, 2004).

Segundo Beck (2004) a XP aborda as necessidades específicas do desenvolvimento de software conduzido por equipes pequenas diante de requisitos vagos e instáveis. A adoção da XP desafia o modo tradicional, incluindo os custos das modificações em um software que na engenharia tradicional obrigatoriamente aumentam ao longo do tempo, como mostra a Figura 4.

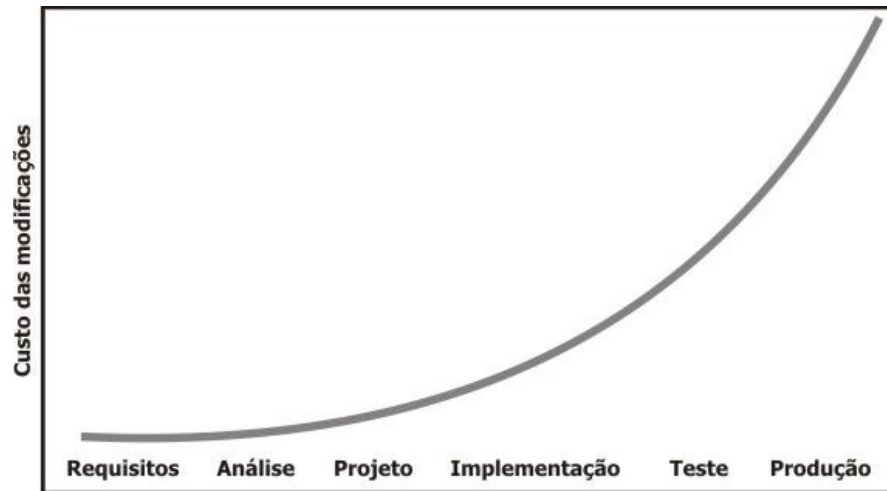


Figura 4. Custo de modificações na engenharia de software tradicional  
 Fonte: KENT, B. (2004)

Para que a aplicação da XP seja possível a curva de custo de modificações deve se manter de forma mais achatada, isto porque se os gastos com modificações não fossem tão altos, os desenvolvedores por mais tradicionais que fossem aguardariam mais para tomar decisões importantes e até mesmo implementariam somente o necessário, pois funcionalidades futuras talvez não se fizessem necessárias. A Figura 5 mostra este gráfico de curva ideal para XP.

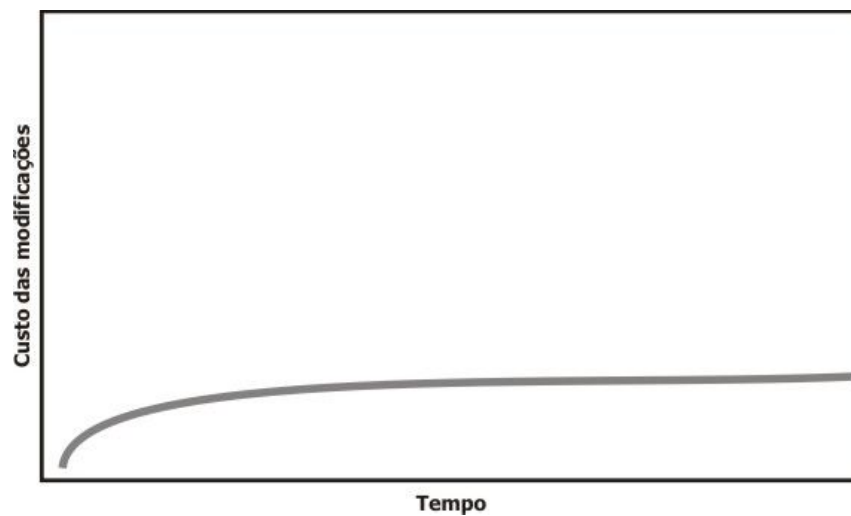


Figura 5. Custo de modificações na programação extrema  
 Fonte: KENT, B. (2004)

A XP defende a não especialização dos membros do time, ou seja, todos participam das atividades, em pares e no desenvolvimento de infra-estruturas e *frameworks* durante o desenvolvimento da aplicação, e a comunicação face a face ou por meio de testes eficientes e código cuidadosamente escrito.

Beck (2004) diz que a XP trata também de variáveis de negócio que são muito importantes para os clientes. Estas variáveis são tratadas já na fase do projeto, e são eles: o custo, o tempo, a qualidade e o escopo. O projeto implementado para ser considerado satisfatório, teria que apresentar um equilíbrio entre estas quatro variáveis.

O alto custo na elaboração de um software talvez o tornasse inviável para o cliente, mas por outro lado pouco investimento pode acarretar em um projeto mal estruturado que não irá solucionar os problemas do negócio do cliente.

Na visão de tempo, também se encontram dúvidas, pois muito tempo despendido pode trazer maior qualidade ao software que está sendo produzido, assim como desinteresse por parte da equipe de trabalho e maior custo.

Na variável qualidade a incerteza pode estar presente, pois se diminuir o nível de exigência de qualidade o tempo despendido seria menor conseqüentemente uma baixa nos custos, mais um software com pouca qualidade pode ter suas funcionalidades comprometidas (BECK, 2004).

O escopo delimita as atividades a serem produzidas em determinada parte do projeto, quanto menor o escopo maior qualidade pode ser aplicada em menos tempo gerando menores custos, mas isto desde que o escopo contemple as necessidades do negócio do cliente. O escopo é a variável mais importante dentro da XP isso porque ele pode interferir em todas as outras variáveis, sendo também o resultado das três primeiras (BECK, 2004).

## 4.1 VALORES DA PROGRAMAÇÃO EXTREMA

Teles (2006) diz que a XP se organiza por meio de um conjunto de valores e práticas, sempre visando fornecer ao cliente o maior retorno possível no investimento feito na aquisição de um software.

Estes valores fundamentais para a XP são *feedback*, comunicação, simplicidade e coragem.

### 4.1.1 *Feedback*

O *feedback* funciona na XP como uma espécie de ajuda contínua e mútua, onde o cliente na medida em que vai entendendo melhor o funcionamento do software passa novas informações para os desenvolvedores de como deseja o funcionamento do sistema.

Esta alimentação também ocorre do desenvolvedor para o cliente, isto acontece quando se mostram para ele os riscos técnicos, alternativas de design entre outras.

A XP trata o cliente como um constante produtor e consumidor de informações, estas informações que ele produz são passadas para os desenvolvedores que rapidamente, as usam para realizar modificações no sistema e repassar novamente para o cliente. Na medida em que esse processo ocorre aumenta a perspectiva para que o software atenda as necessidades do cliente (TELES, 2006).

### **4.1.2 Comunicação**

A comunicação é muito importante em um grupo de trabalho, onde a troca de informações é imprescindível para o bom funcionamento de um projeto. A comunicação dentro de um grupo de trabalho pode ocorrer de várias formas, através de conversas face-a-face, telefone, e-mail, mensagens instantâneas e outros. A XP acredita que a comunicação face-a-face ainda supera todas as outras, pois o entendimento acontece de forma imediata, isso por usar vários outros elementos da comunicação, como o conteúdo da fala, expressão, gestos, tom da voz, entre outros.

Enquanto na engenharia tradicional a comunicação ocorre na maioria das vezes em forma de documentos escritos, na XP estes documentos são usados para descrever o que já foi realizado e não o que ainda deve ser feito (TELES, 2006).

### **4.1.3 Simplicidade**

A simplicidade na XP está presente nas ações dos desenvolvedores que precisa ser suficientemente simples para obter um maior entendimento dos outros membros da equipe e principalmente do cliente.

Desenvolvendo o projeto e o software de uma forma mais simples evita-se criar funcionalidades desnecessárias, diminuindo o re-trabalho, pois o trabalho estará focado no que realmente o cliente necessita naquele momento (TELES, 2006).

#### 4.1.4 Coragem

A coragem na XP se apresenta na medida em que os programadores precisam dela para contrariar tudo o que se apresentou até agora em termos de modelagem de construção de software, utilizando novas premissas como:

- a) desenvolver o software de forma incremental;
- b) manter o sistema simples;
- c) permitir que o cliente priorize as funcionalidades;
- d) fazer os desenvolvedores trabalharem em par;
- e) investir tempo em *refactoring*;
- f) investir tempo em testes automatizados;
- g) estimar as histórias na presença do cliente;
- h) expor o código a todos os membros da equipe
- i) integrar o sistema diversas vezes ao dia;
- j) adotar um ritmo sustentável;
- k) abrir mão de documentações que servem como defesa;
- l) propor contratos de escopo variável;
- m) propor a adoção de um processo novo.

Estes pontos levantados ajudam o programador a continuar trabalhando de forma extrema, sempre visando o acréscimo de rendimento para o cliente e seu negócio (TELES, 2006).

Desenvolvimento de software de maneira incremental ocorre assim como no ciclo de vida em espiral, onde partes do sistema que já foram feitas são repassadas, mas sempre com o acréscimo de novas funcionalidades usando o *feedback* extraído do cliente. Mesmo que para isso tenha que alterar partes do sistema que já estejam funcionando.

Uma das práticas da XP é a programação em par, por meio dela pode-se facilitar estas alterações, diminuindo a taxa de erros, pois a revisão se torna constante.

Esta prática por trabalhar com dois programadores em uma mesma máquina pode passar a idéia ao cliente de que a um desperdício de mão de obra, aumentando o custo, pois se imagina que talvez um programador fizesse o mesmo trabalho, mas a XP mostra que isso não é verdade.

Os desenvolvedores que utilizam a XP visam deixar o código o mais simples possível, criando funcionalidades que serão usados pelos clientes no presente. A equipe que usa a XP tem coragem para afirmar que podem adicionar novas funções com o sistema em andamento, sem colocar em risco o projeto.

As funções e prioridades do software serão definidas por meio de histórias criadas pelas necessidades do cliente, são eles que definem o que e quando deve ser feito, mesmo que as prioridades não sigam as mesmas idéias do ponto de vista técnico.

Outro ponto levantado pela XP é a *refactoring*, que nada mais é do que melhora constante da qualidade do software. Alguns desenvolvedores tratam a *refactoring* como sendo um desperdício de tempo, que pode em alguns casos acabar atrasando o desenvolvimento do software, é preciso ter coragem para aplicar esta prática e provar que ela ajuda a aumentar a qualidade do software.

Os testes não são desperdícios de tempo como comumente são entendidos pelos programadores tradicionais. Um aumento de tempo gasto pode até vir a ocorrer em curto prazo, mas isso muda quando se avaliam os resultados futuros onde os testes ajudam a diminuir os erros e a diagnosticá-los o mais rápido possível.

As estimativas de tempo utilizado para realizar as histórias criadas pelo cliente devem ser feitas da maneira mais precisa possível, a XP trata o cliente de forma aberta e honesta deixando ele sempre a par de tudo que acontece no projeto de desenvolvimento.

Outra forma de utilização da coragem na XP é a exposição do código para todos os membros da equipe, portanto o código passa a ser coletivo, avaliado e manipulado por outros programadores. O desenvolvedor precisa estar aberto a sugestões e críticas.

A integração do sistema acontece constantemente na XP, com esta integração surgem riscos que algumas partes do sistema parem de funcionar corretamente, um modo de diminuir estes erros é a aplicação do procedimento de testes.

Na parte que diz respeito à documentação a XP também se distingue da tradicional. Como já foi dito anteriormente, ela conduz o projeto de forma mais aberta, não necessitando tanto dessa parte totalmente burocrática para fazer sua defesa, pois do início ao fim do projeto os clientes estão integrados com os outros membros, estando por dentro de tudo que está acontecendo. Mas isso não significa que a XP não usa de documentação, mas utiliza somente quando se faz necessário.

Na XP os contratos de escopo de desenvolvimento são totalmente variáveis, pois assim pode-se extrair e já utilizar o *feedback* do cliente. É preciso ter coragem para aplicar essa idéia nova de contrato.

Por se tratar de um processo de desenvolvimento novo é necessária novamente coragem para aplicá-lo, isso por ser diferente de tudo que já se fez até hoje em termos de desenvolvimento de software.

#### 4.2 CICLO DE VIDA DA PROGRAMAÇÃO EXTREMA

O ciclo de vida da XP consiste de seis fases: exploração, planejamento, iterações para liberação, produção, manutenção e morte.

### **4.2.1 Exploração**

A XP possui como característica a iteração com o cliente, é na fase de exploração que ele participa ativamente, pois tem a possibilidade de descrever o que deseja que seja incluído na primeira liberação, isto é feito através de cartões escritos pelo usuário para o programador. Estes cartões possuem histórias que descrevem as funções para serem adicionadas ao programa, muito similar aos casos de uso tradicionais, com a diferença que é escrita pelos próprios clientes em forma de texto. Enquanto isso, a equipe de projeto se familiariza com as ferramentas, tecnologias e práticas que estarão usando no projeto. A tecnologia a ser usada será testada e as possibilidades de arquitetura são exploradas construindo-se um protótipo do sistema.

Essa fase de exploração leva entre algumas semanas a poucos meses, dependendo do conhecimento que os programadores têm da tecnologia (BECK, 2004).

### **4.2.2 Planejamento**

O Planejamento é uma das fases mais curtas do processo XP, e costuma durar apenas alguns dias. É na fase de planejamento que as histórias escritas pelos clientes nos cartões são avaliadas para ver se estão de acordo com as funções que irão constar na primeira liberação. Os programadores primeiro estimam quanto esforço cada história irá consumir e então é feita a programação (BECK, 2004).

### **4.2.3 Iterações para liberação**

A fase de iterações para liberação inclui várias modificações no sistema antes de sua primeira liberação. A programação que foi feita durante a fase de

planejamento é dividida em um número de iterações sendo que cada uma leva de algumas semanas para ser implementada. A primeira iteração cria um sistema base com as principais funcionalidades. Isso é conseguido selecionando-se as histórias que vão dar base para que a estrutura do sistema todo seja implementada. O cliente decide as histórias que vão ser selecionadas e implementadas em cada iteração. No final da última iteração, o sistema está pronto para a produção (BECK, 2004).

#### **4.2.4 Produção**

É na fase de produção onde ocorrem testes extras no desempenho do sistema para que não seja encaminhado com problemas para o cliente. Seguindo as práticas da XP, nesta fase ainda são aceitas novas mudanças, onde se decide se elas serão incluídas no software. Durante essa fase, as iterações precisam ser reduzidas de três para uma semana. As idéias e sugestões são documentadas para implementações futuras durante a fase de manutenção (BECK, 2004).

#### **4.2.5 Manutenção**

Após a primeira liberação para o cliente, o projeto XP prevê que seja feito ao mesmo tempo novas iterações no sistema, dando também assistência ao sistema que já está em uso. Para isso, a fase de manutenção requer um esforço também para as tarefas de suporte ao cliente. Assim, a velocidade de desenvolvimento deve desacelerar depois que o sistema está em produção. Nesta fase se faz necessário um maior número de pessoas na equipe para suprir a demanda, já que são feitas duas funções (BECK, 2004).

#### 4.2.5 Morte

A fase de morte acontece quando o cliente não tem mais histórias para serem implementadas. Isso requer que o sistema satisfaça suas necessidades em outros pontos (por exemplo, aumentando o desempenho e a reusabilidade). Esse é o momento em que a documentação necessária do sistema é finalmente realizada e mais nenhuma modificação na arquitetura, no projeto ou no código são feitas. A morte pode ocorrer também caso o sistema não esteja exibindo as saídas corretas, ou se ele se tornou muito caro para desenvolvimento adicional (BECK, 2004).

#### 4.3 PAPÉIS DA PROGRAMAÇÃO EXTREMA

A XP possui vários papéis que são executados por diferentes pessoas, cada uma possuindo sua finalidade, mais todos são de extrema importância para o bom funcionamento do projeto (BECK, 2004):

- a) **programador:** é ele quem faz o código do programa, sempre tentando fazer o código mais simples para melhor entendimento dos usuários e de outros programadores, um bom programador tem uma boa comunicação com todos os membros da equipe, visando o sucesso do projeto;
- b) **cliente:** é o cliente quem diz como quer que o programa funcione, passando para o programador quais serão os requisitos básicos do programa e qual sua prioridade na hora da implementação;
- c) **verificador:** o verificador é quem ajuda o cliente a fazer os testes no programa, transmitindo os resultados para os programadores;
- d) **fiscal:** o papel do fiscal é dar as respostas se o processo da

- XP está evoluindo de maneira satisfatória de acordo com o que foi planejado ou se será necessário alguma mudança no decorrer do processo;
- e) **técnico:** é quem mais entende o processo da XP, conhece todos os papéis, por isso tem a função de auxiliar os membros da equipe que necessitam de algum tipo de ajuda;
  - f) **consultor:** possui o conhecimento técnico específico, e por isso pode ajudar a equipe para resolver problemas específicos. É o membro mais externo da equipe;
  - g) **gerente:** é quem toma as decisões, mais sempre ouvindo a opinião da equipe, para diminuir as dificuldades no processo.

Necessariamente não se precisa ter todos estes papéis dentro de uma equipe de XP, mas alguns como: programador, cliente, treinador e verificador, são essenciais.

#### 4.4 QUANDO NÃO USAR A PROGRAMAÇÃO EXTREMA

De acordo com Beck (2004) em alguns projetos a XP não é indicada. Uma das principais dificuldades que tornam a XP inviável é a cultura de negócio das empresas que impõem barreiras para algo novo, mesmo antes de conhecê-la. Outro ponto importante é o tamanho do projeto, pois equipes com mais de dez programadores torna a comunicação entre eles e também a iteração muito mais difícil.

A curva de custos com modificações quando não é constante dificulta a aplicação da XP ela age de forma que as modificações são constantes durante a produção do software.

Projetos onde os programadores estão separados sejam em salas, andares ou até mesmo geograficamente distantes também não funciona com a XP, isto acaba com a integração da equipe diminuindo a comunicação face-a-face.

## **5 TRABALHOS CORRELATOS**

No decorrer dos estudos feitos para esse trabalho de pesquisa, foram analisados alguns trabalhos científicos que possuíam propósitos relacionados com essa pesquisa, mas com foco em objetivos diferentes. Alguns trabalhos envolvendo esta metodologia relativamente nova, que é a programação extrema, serão descritos a seguir.

### **5.1 AVALIAÇÃO DE PROCESSOS DE SOFTWARE: UM ESTUDO DE CASO EM XP E ICONIX**

Dissertação de Cristina Bona apresentada ao programa de Pós-Graduação em Engenharia de Produção, em 2002, pela Universidade Federal de Santa Catarina localizada em Florianópolis no estado de Santa Catarina.

Esta dissertação tem como objetivo principal realizar um estudo comparativo entre os modelos de desenvolvimento ágil de sistemas: o XP e o ICONIX, onde são examinados simultaneamente os recursos de ambos os processos, determinando quais características devem ser inerentes a um processo de software que garanta a produtividade e qualidade (BONA, 2002). Este trabalho pode ser encontrado no endereço: <http://teses.eps.ufsc.br/defesa/pdf/10816.pdf>.

### **5.2 CARACTERIZAÇÃO DE MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE**

Artigo científico de José Fortuna Abrantes e Guilherme Horta Travassos apresentado no Workshop em Desenvolvimento Rápido de Aplicações (WDRA) em 2006, a Universidade Federal do Rio de Janeiro localizada no Rio de Janeiro, Capital.

O objetivo deste artigo é avaliar quais as características de agilidade presentes nos métodos ágeis de desenvolvimento de software. Um protocolo de pesquisa foi formalizado e executado para conduzir uma revisão sistemática de literatura. Os dados obtidos foram analisados e uma proposta para caracterização básica de métodos ágeis é apresentada.

Este artigo pode ser encontrado no endereço:  
<http://reuse.cos.ufrj.br/wdra2007/images/artigos/30188.pdf>

### 5.3 USO EFICAZ DE MÉTRICAS EM MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE

Dissertação de Danilo Toshiaki Sato para obtenção do grau de Mestre em Ciência da Computação, em 2007, pelo Instituto de Matemática e Estatística da Universidade de São Paulo, localizada em São Paulo Capital.

O objetivo principal deste trabalho foi realizar um estudo de métricas utilizando métodos ágeis de desenvolvimento de software. Um estudo de caso da aplicação de XP em sete projetos diferentes. Por fim este trabalho classifica os sete projetos em relação aos termos e categorias propostas pela programação extrema. Esta dissertação está disponível no endereço:  
<http://grenoble.ime.usp.br/~gold/orientados/dissertacaoDaniloSato.pdf>

### 5.4 ESTUDO ANALÍTICO DA ENGENHARIA DE REQUISITOS ATUAL E A DE PROCESSOS AGEIS (XP)

Trabalho de Conclusão de Curso de Mario Domingos Castra Maranhão Wolf e Mauricio Aldenos Souza dos Santos para a obtenção do grau de Bacharel em

Ciência da Computação, em 2004, ao Centro de Ciência Exatas e Tecnologia da Universidade da Amazônia, localizada em Belém, no estado do Pará.

O objetivo deste trabalho é estabelecer uma relação entre XP e casos de uso, mostrando prós e contras. Outro objetivo é estudar os tipos de técnicas, ferramentas e processos atuais, no âmbito da engenharia de requisitos, enfatizando as praticadas nos processos XP e RUP.

Este trabalho está disponível no endereço:

<http://www.cci.unama.br/margalho/portaltcc/tcc2004/mario&mauricio.pdf>

## **6 ETAPAS DE METODOLOGIA ÁGIL, UTILIZADAS NA CONSTRUÇÃO DO SITE COM AMBIENTE VIRTUAL**

Por meio da revisão bibliográfica feita sobre as metodologias tradicionais de desenvolvimento dentro da engenharia de software, notou-se que esta apresenta fragilidades, como prazo de entrega impreciso, custo de mudança crescente, além de produção de funcionalidades que em alguns casos nunca serão usadas pelo cliente.

Estes estudos sobre as metodologias tradicionais ocorreram apenas para ter uma forma de diferenciá-la da metodologia ágil, mas isso sem fazer comparações mais aprofundadas referente a pesquisa, pois ela aplica somente a metodologia ágil e não a tradicional.

A metodologia ágil de desenvolvimento foi criada com o intuito de tentar diminuir ou até mesmo acabar com fragilidades das metodologias tradicionais. Mas por serem relativamente novas, estas metodologias ainda são questionadas por desenvolvedores que usam a metodologia tradicional, que acreditam ser necessários maiores estudos para que estas metodologias sejam tratadas como um modo confiável de desenvolver e projetar software.

Os desenvolvedores que testaram à metodologia ágil mostram na maioria das vezes bons resultados, salientando a necessidade de mudar a cultura dos desenvolvedores que estão estagnados sem aceitar a existência de possíveis métodos com uma melhor forma de trabalhar e obter resultados mais satisfatórios do que os atuais.

A metodologia ágil estudada com maior intensidade neste trabalho foi a Programação Extrema. Suas etapas se adaptaram muito bem a aplicação. Desta forma elas foram utilizadas para se obter os resultados no projeto.

Neste capítulo, serão apresentadas etapas de metodologia ágil (XP) utilizadas na aplicação do trabalho. Pesquisas e estudos foram realizados para a elaboração desta aplicação.

O tema desta aplicação trata do desenvolvimento de um *site* com ambiente virtual para um colégio localizado na cidade de Cocal do Sul, Santa Catarina. A seguir será mostrado as partes envolvidas neste processo, assim como as etapas de XP que foram utilizadas na execução do projeto, por meio do ciclo de vida e dos valores por ela sugeridos.

## 6.1 DEFININDO PARTES ENVOLVIDAS

Seguindo as diretrizes passadas pelo cliente, pode-se levantar com clareza as pessoas que estariam envolvidas com o *site*, definindo assim seus atores. Os quatro atores que podem acessar o *site* e interagir com suas funcionalidades estão definidos nos seguintes perfis:

- a) **Visitante:** usuários com menor nível de permissões, só poderá acessar *links* como: o colégio, turmas, professores e contato, que são os *links* básicos do *site*;
- b) **Aluno:** usuário que além dos *links* básicos do *site* também pode interagir com o *link* de ambiente virtual, onde pode consultar e alterar seus dados pessoais, além de visualizar suas disciplinas cursadas, podendo ver as datas de provas e trabalhos, ver suas notas e baixar o material de aula;
- c) **Professor:** usuários previamente cadastrados pelo administrador que podem acessar o *link* de ambiente virtual para colocar as datas de provas e trabalhos, notas dos alunos, material de aula e também editar seus

próprios dados pessoais;

- d) **Administrador:** usuário com mais alto grau de responsabilidade e privilégios dentro do *site*, podendo cadastrar os usuários e editar qualquer *link* do *site*.

Visando o maior entendimento das atividades operacionais de cada ator foi elaborado os diagramas de caso de uso para cada um deles, que podem ser vistos nos apêndices A,B,C e D do trabalho.

A seguir têm-se os valores da XP aplicados no desenvolvimento da aplicação do trabalho.

## 6.2 VALORES APLICADOS

Podem-se explicar estes valores como sendo a forma de agir do desenvolvedor, como se porta frente o cliente e ao projeto, mudando a cultura com que os desenvolvedores tradicionais estão acostumados a demonstrar. A seguir se apresenta cada um destes valores aplicados no desenvolvimento.

### 6.2.1 *Feedback*

Nesta aplicação o *feedback* ocorreu em todo o processo de desenvolvimento do projeto, pois o cliente e o desenvolvedor estavam em constante contato, havendo uma troca de informações necessárias para o melhor funcionamento do software, neste caso do *site*.

Esta troca de informações foi de extrema importância para que o *site* fosse construído de acordo com as expectativas do cliente, mostrando assim como o *feedback* é imprescindível para a metodologia XP.

## **6.2.2 Comunicação**

Dentro do projeto em questão, a comunicação não aconteceu na sua totalidade como sugere a XP, pois a aplicação foi desenvolvida basicamente por um único desenvolvedor, tendo em alguns pontos o acompanhamento de uma pessoa disponibilizada pelo cliente. Mas por outro lado, a comunicação com o cliente que também é uma premissa da metodologia aplicada, teve como facilitador deste valor, a disponibilidade de um funcionário capacitado do colégio para comunicação face-a-face, com o qual se pôde esclarecer as dúvidas que surgiram no decorrer do processo.

## **6.2.3 Simplicidade**

Este valor de simplicidade foi aplicado muitas vezes no decorrer do projeto, pois sempre se procurou, por parte do desenvolvedor, a forma mais simples de realizar as iterações do software, sempre colocando à frente as funcionalidades que o cliente primeiro necessitou.

## **6.2.4 Coragem**

A coragem aconteceu desde o início do projeto, onde uma nova metodologia foi aplicada mesmo contrariando a forma de projetar e desenvolver software mais conhecida e utilizada na região, permitindo que o cliente participasse de forma atuante inclusive priorizando as funcionalidades do software.

## 6.3 CICLO DE VIDA

O ciclo de vida acompanhou, na medida do possível, as etapas da XP, sendo às vezes necessário adaptação para a aplicação decorrente.

### 6.3.1 Fase de exploração

A fase de exploração nesta aplicação começou a partir do momento em que houve a primeira reunião para a apresentação dos objetivos do trabalho. A diretoria do colégio, por sua vez, mostrou-se muito interessada em usufruir da aplicação, e através dela ter um diferencial frente a maioria dos colégios da região que ainda não utilizam tal tecnologia.

Logo após o aceite do colégio, partiu-se para o levantamento dos requisitos pelo cliente (colégio) que teve um tempo de aproximadamente três semanas para avaliar o que achava pertinente que o *site* contemplasse.

Neste mesmo intervalo de tempo foi feito um estudo da linguagem e do banco de dados que seriam utilizados na aplicação. As tecnologias que mais se adaptaram as necessidades do cliente conforme avaliações foram à linguagem php, a utilização de html e javascript e o banco de dados *mysql*.

O colégio por sua vez, disponibilizou o professor de informática da instituição para servir como cliente atuante no projeto e também como par, já que a metodologia de programação extrema utiliza programação pareada.

O *site* tem como público alvo os alunos, professores, pais, administração do colégio e visitantes em geral, sendo que a administração, professores e alunos terão acesso ao ambiente virtual, que é o grande diferencial do *site* em questão.



### 6.3.2 Fase de planejamento

De acordo com o ciclo de vida da Programação Extrema na segunda fase da construção do software acontece o planejamento, onde o cliente passa para os desenvolvedores através do cartão de história os requisitos que acredita ser necessário serem implementados. O planejamento seguido pelo desenvolvedor é mostrado na Figura 7.

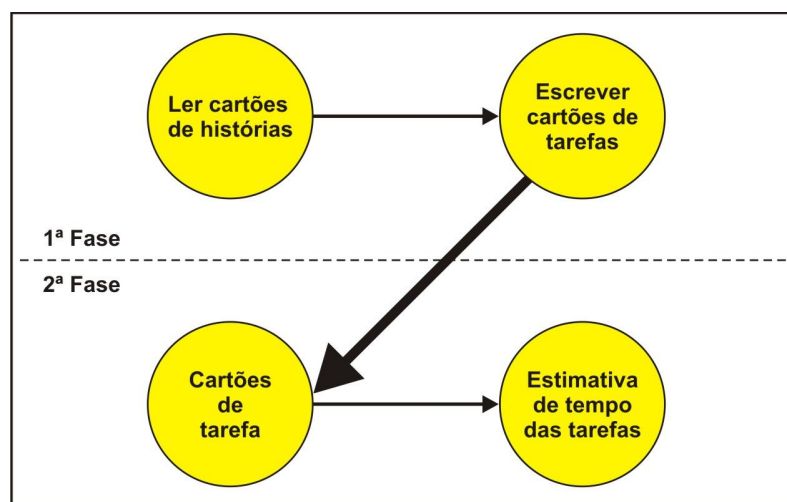


Figura 7. Etapas do planejamento

Na reunião seguinte o colégio apresentou os requisitos que achava necessário que estivessem contidos no *site*. Os pontos definidos para contemplar foram: uma apresentação do colégio, com histórico; relação do corpo docente, especificando suas formações; o currículo escolar que os alunos terão; principais eventos realizados pelo colégio; além de um mural de recados para pais e alunos e o mais importante da aplicação, um ambiente virtual.

Na Figura 8 mostram-se dois cartões de história já preenchidos pelo cliente que foram utilizados pelo desenvolvedor para elaborar os cartões de tarefas.

**Cartão de História - Prioridade 1**

Descrição: Criar um botão com as informações do colégio na primeira página, com uma foto de professor destaque do mês.

Data: 25 / 03 / 08

**Cartão de História - Prioridade 2**

Descrição: Criar uma janela para que o aluno possa entrar no ambiente virtual. Criar local onde ele possa alterar suas informações pessoais.

Data: 25 / 03 / 08

Figura 8. Cartão de história preenchido pelo cliente

### 6.3.3 Fase de iterações para liberação

Foi na fase de iterações para liberação que ocorreu o agrupamento e seleção das histórias de acordo com suas características e relevâncias, para assim poder formar o cartão de tarefas (Figura 9) respeitando o grau de prioridade que o cliente relatou.

**Cartão de Tarefa**

Data: \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

Tema: \_\_\_\_\_

Tarefa: \_\_\_\_\_

\_\_\_\_\_

Responsável: \_\_\_\_\_

Estimativa de Tempo: \_\_\_\_\_

Figura 9. Cartão de tarefa

Após os cartões de tarefas serem elaborados são repassados para o mural de tarefas onde estes cartões passam por três fases a realizar, em andamento e concluído.

A Figura 10 mostra uma foto do mural de tarefas em andamento.



Figura 10. Mural de tarefas

Ao final desta fase de agrupamento e seleção o software já tinha uma base para que passasse para a próxima fase, que é a produção propriamente dita.

### 6.3.4 Fase de produção

A partir da fase de produção já se possuía a estrutura básica do *site*, tendo alguns *links* funcionando conforme o desejo do cliente, sendo eles: apresentação do colégio, turmas, professores e contato, enquanto o *link* do ambiente virtual já estava em processo de construção tendo suas principais funcionalidades implementadas. Também foi nesta fase que se deu início aos testes de consistência das funções que já podiam ser operadas, como por exemplo, o cadastro de usuários (alunos, professores e administradores) do *site*, assim como o *login* e a senha de cada usuário. A tela de *login* e senha esta representada na Figura 11.



A screenshot of a login form titled "Ambiente Virtual". The form has a red header bar with the text "Ambiente Virtual" in white. Below the header, there are three input fields: "Código" (a text box), "Senha" (a text box), and "Grau/Série" (a dropdown menu showing "1ª série / Fundamental" with a downward arrow). Below the dropdown menu is a link that says "Esqueci a senha".

Figura 11. Tela *login* e senha do ambiente virtual

### 6.3.5 Fase de manutenção

Como o intuito do colégio, a princípio, era uma apresentação virtual da instituição, desta forma mais uma maneira de ser conhecido na sua região de atuação, foi disponibilizada para a primeira liberação dos *links* com menor esforço de

desenvolvimento, porém estes satisfizeram as necessidades a priori, podendo ser acessado pelos visitantes. A Figura 12 mostra a página principal do *site*.

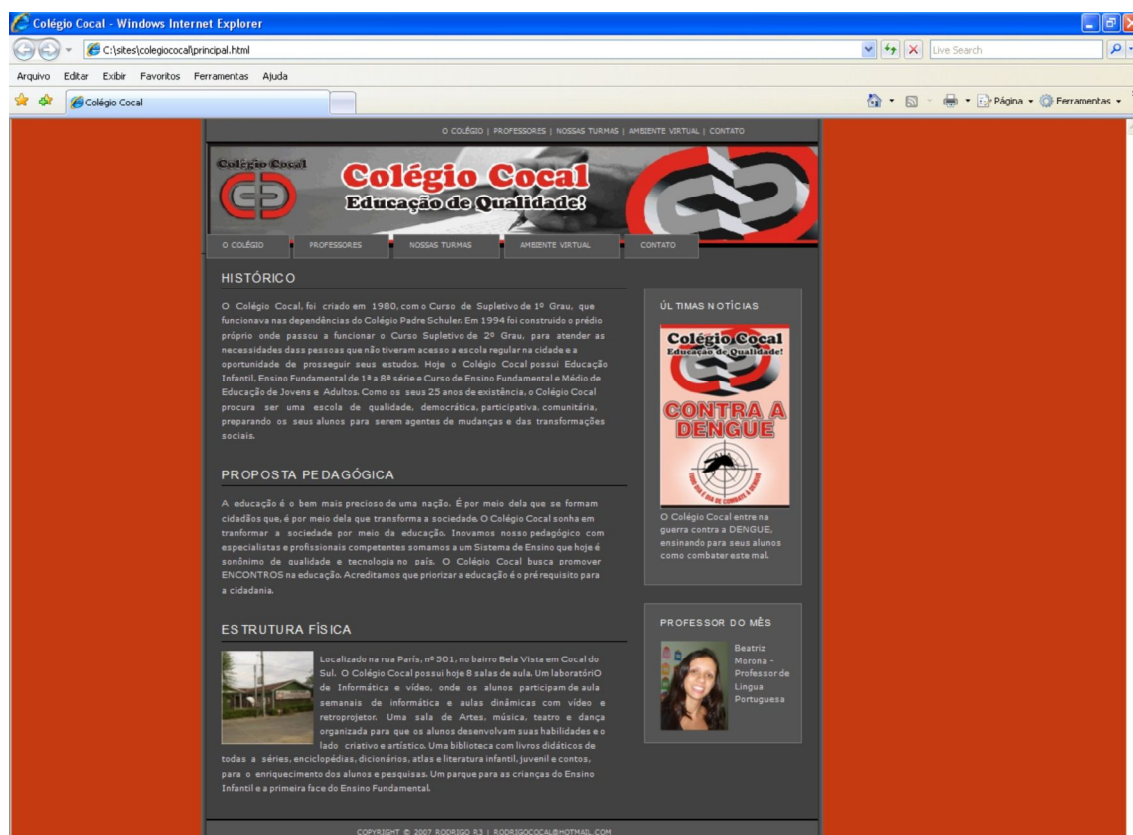


Figura 12. Página principal do *Site*

Enquanto isso, o *link* mais elaborado e que exigiu maiores estudos, que se trata do *link* de ambiente virtual continuava a ser implementado. Como os outros *links* se tratavam de iterações mais simples sua manutenção baseava-se em pequenas alterações.

### 6.3.6 Fase de morte

A última fase da Programação Extrema é a morte. É nesta fase que acontece a entrega final de todas as funcionalidades do sistema. Depois de tudo entregue começa a ser feito a segunda parte desta fase que se trata da documentação do sistema, pois a

XP prega ser mais simples documentar o que já foi feito do que algo a fazer, a maioria dos softwares sofre modificações incrementais no decorrer do processo de construção.

Nesta aplicação a última funcionalidade a ser entregue foi o *link* de ambiente virtual onde somente usuários pré-cadastrados podem acessar. O ambiente virtual serve como uma nova linha de comunicação entre a entidade de ensino e os alunos e professores, com ele pode-se ter acesso às notas de cada disciplina além de poder consultar datas de provas e trabalhos, fazer o *download* do material de aula e também postar os trabalhos.

Esta fase ainda não se encontra totalmente concluída, pois a parte de documentação ainda encontra-se em desenvolvimento.

#### 6.4 REFATORAÇÃO

Outra etapa bastante utilizada pela metodologia ágil de programação extrema (XP) é a refatoração, que se trata de modificações no código trazendo melhorias, mas sem alterar as funcionalidades do software.

Através da refatoração pode-se tornar o código mais simples, e a simplicidade é valor básico defendido pela XP. Para que a refatoração não se torne um problema dentro do ciclo do projeto, a XP recomenda testes constantes, pois mudanças podem causar inconsistência no código, por isso a XP trata os testes como algo muito importante dentro do processo, sejam estes testes automatizados ou manuais, em casos mais simples.

Em alguns casos específicos a refatoração é aconselhada, podendo-se citar o código duplicado, método longo, classe grande demais, lista de parâmetros longa e má endentação do código em questão.

Na construção do ambiente virtual foram usadas algumas refatorações, podendo citar-se a criação de sessões para segurança em cada arquivo do ambiente virtual. A princípio o código não continha esta parte de segurança, foi realizada a refatoração deste código, mas isso sem mudar a parte funcional do ambiente virtual. A parte do código refatorada citada está sendo mostrada na Figura 13.

```
session_start();
if(!isset($_SESSION['usuario']['login']))
{
    header("location:'admin/index.php'");
}
else
{
    $nome=$_SESSION['usuario']['login'];
}
```

Figura 13. Código de sessão do *site*

Para um desenvolvedor experiente, esta parte do código com sessões seria utilizada desde o início, mas como o desenvolvedor em questão está evoluindo junto com o projeto, isso foi notado após os primeiros testes, fazendo a refatoração logo que possível.

Outras refatorações também foram realizadas como a separação de código em funções para que o mesmo ficasse mais enxuto e de fácil localização, isto aconteceu no item cadastro, onde todas as funções foram separadas em arquivos, como: `item_cadastra`, `item_deleta`, `item_form_altera`, `item_inclui`, `item_lista`, `item_update`. Todas estas funções no início estavam localizadas em um único arquivo, e com a separação os códigos ficaram menores e fáceis de localizar.

Refatorações de pequeno porte como troca de nome de variáveis foram feitas constantemente no decorrer do desenvolvimento.

Estas refatorações têm como objetivo principal a aplicação de mais um valor da XP, a simplicidade, que torna o código mais fácil para que outro desenvolvedor possa entendê-lo, se for necessário.

Seguindo a tendência da informática que procura trazer cada vez mais interatividade entre homem e máquina, usufruindo dos benefícios que ela pode proporcionar para as pessoas e com o intuito de diminuir a exclusão digital que ainda assola nossa comunidade, a aplicação vem por meio do ambiente virtual trazer ao alcance dos alunos do Colégio Cocal mais uma forma de comunicação entre aluno e instituição de ensino.

A utilização de uma metodologia diferenciada e ainda pouco usada em nossa região para projetar e desenvolver uma aplicação ajuda a estender o leque de conhecimento e alternativas para desenvolvedores que acreditam ser possível ainda obter melhores resultados satisfazendo seus clientes.

## CONCLUSÃO

Na informática é muito importante sempre manter-se atualizado e atento às novas tecnologias disponíveis no mercado, sejam elas de hardware ou de software. Nas últimas décadas a evolução dos softwares vem superando as de hardware, tendo como principais focos de estudos e aprimoramento a engenharia de software, que viu surgir novas metodologias de desenvolvimento. A metodologia ágil é uma delas que há aproximadamente dez anos surgiu como um novo método de gerenciamento de projetos dentro da informática.

Hoje, estas metodologias ágeis ainda enfrentam um grau de rejeição considerável, mesmo assim as pesquisas sobre metodologias ágeis vêm aumentando. As metodologias ágeis possuem métodos bem específicos de organizar todo o processo de construção do software, isto para obter vantagens de qualidade e produtividade. Atualmente a metodologia ágil que possui maior visibilidade é a XP (Programação Extrema), isso por ser a mais documentada e pesquisada. A metodologia XP foi concebida a partir da idéia que desenvolver software é difícil, e desenvolver software de qualidade no prazo combinado é ainda mais difícil.

O grande ponto que dificulta uma maior difusão e aplicação da XP é cultural, pois desenvolvedores e projetistas na maioria dos casos ainda possuem uma visão conservadora, não aceitando novas metodologias. Segundo seus seguidores, para que a XP obtenha sucesso em sua implantação é preciso mudar a forma de pensar, com desenvolvedores agindo com mais simplicidade, visando maior entendimento do processo pelo cliente e com isso obtendo informações mais exatas das reais funcionalidades que o cliente necessita.

As metodologias tradicionais como já foram estudadas durante a disciplina de engenharia de software, já se possuía um conhecimento básico de seu processo de

execução, e isso aliado a maiores estudos bibliográficos pode-se entender com maior clareza seus pontos fortes e também suas fragilidades.

As metodologias ágeis também foram estudadas e com maior intensidade a XP, onde inclusive foi feito um curso sobre ela na universidade de São Paulo (USP), onde é bastante pesquisada e desenvolvida, sendo aplicada em diversos projetos da própria universidade. Assim notou-se que sua principal intenção é realmente suprir as necessidades que desenvolvedores insatisfeitos levantaram e relataram através do manifesto ágil.

A aplicação deste trabalho desde seu projeto até execução seguiu etapas da XP, aplicando inicialmente seus valores, que tratam do *feedback*, comunicação, simplicidade e a coragem. Apesar de estes valores serem aplicados pelo desenvolvedor, a participação do cliente foi essencial para que o projeto funcionasse como a XP prevê.

Mas a princípio como já se esperava o cliente não estava preparado para uma participação tão atuante. Foi neste momento que as primeiras dificuldades da aplicação de uma nova metodologia foram encontradas, isto porque para o cliente sua participação seria simplesmente relatar o que ele desejava em uma conversa breve e esperar o desenvolvimento do produto final para seu uso, tipicamente uma mentalidade usada nas metodologias tradicionais.

Porém, seguindo as etapas da XP, foi explicado ao cliente que se tratava de um projeto que usava uma nova metodologia e para melhor funcionamento e avaliação seria necessário um cliente atuante, pois se tratava também de um estudo científico, que visava obter resultados. Entendendo isso, o cliente (Colégio Cocal) disponibilizou uma pessoa para servir como cliente XP. Esta pessoa conhecia com maiores detalhes o que o colégio necessitava e esperava do projeto. Então a dificuldade inicial já estava superada, pois o projeto já tinha à sua disposição um cliente atuante.

Com esta experiência, pôde-se avaliar um ponto que a XP diz ser necessário aplicar, desenvolvedor com vontade de mudança. Mas esta vontade na verdade tem que partir dos dois lados envolvidos, desenvolvedores e cliente. Porque mesmo tendo desenvolvedores engajados em aplicar a XP isso não será possível se o cliente não quiser ou não colaborar. Então esta participação do cliente pode passar de um ponto positivo para negativo da XP.

Outro ponto que pôde ser levantado a respeito da participação do cliente são as vantagens do uso do *feedback* mais rápido, pois por meio deste pôde-se explicar a curva de custo e tempo, onde na XP é quase linear, enquanto nas metodologias tradicionais ela é muito mais acentuada. Isso se dá porque na metodologia ágil o cliente está em constante contato com o que está sendo feito, analisando a cada momento se é exatamente isso que ele deseja, desta forma os ciclos são menores, por isso o custo com mudança diminui, ou seja, gerando sempre uma integração contínua.

Com a experiência adquirida com a aplicação da XP neste trabalho pôde-se observar que ela funciona realmente de maneira satisfatória. Por esta razão pode ser usada em projetos de pequeno porte como trata este desenvolvimento. Nesta aplicação o cliente é representado por uma única pessoa que entende do negócio e pode passar informações pertinentes para o desenvolvedor.

No decorrer do desenvolvimento deste trabalho algumas dúvidas que dizem respeito ao bom funcionamento da XP foram surgindo. Uma destas dúvidas é como se portaria a metodologia XP em um projeto maior, onde, por exemplo, existem vários programas formando um grande sistema, onde vários clientes seriam necessários, e por isso o número de pessoas envolvidas aumentaria causando assim alguns desencontros nesta metodologia que age de forma mais enxuta. Acredita-se que neste caso um pouco mais de formalismo deixaria a própria equipe mais segura para desenvolver o sistema.

O ciclo de vida da aplicação, como já foi dito, seguiu os padrões da XP, e atendeu todas as necessidades de um bom projeto, com qualidade que satisfaz o cliente e obedecendo ao tempo de execução coerente com o tamanho do produto desenvolvido. Nas fases do ciclo de vida algumas dificuldades foram encontradas. Na exploração, por razão da necessidade do estudo de uma linguagem pouco conhecida pelo desenvolvedor, e também na fase de manutenção, que não foi tão utilizada até o presente momento, em virtude do tamanho da aplicação, não podendo se ter uma avaliação do seu uso na XP.

Este trabalho deixa como forma de contribuição para o curso e acadêmicos de ciência da computação, conceitos estudados ao longo do curso como: banco de dados, programação, além da aplicação de diretrizes da metodologia ágil de Programação Extrema.

Com o estudo e a implementação do trabalho proposto, pode-se aprender como funciona a aplicação de uma metodologia ágil em projeto de desenvolvimento, levantando vantagens e dificuldades encontradas, aprimorando assim o entendimento desta parte importante da engenharia de software que são as metodologias de desenvolvimento. Além disso, conceitos utilizados para a elaboração do ambiente virtual também foram assimilados, podendo-se perceber o quanto a internet e as aplicações *web* podem contribuir para uma melhor comunicação entre escola e aluno.

Este trabalho procurou aplicar as etapas da metodologia ágil de Programação Extrema em um projeto de construção de um *site* com ambiente virtual. Porém vários pontos ainda podem ser estudados e aprofundados no que diz respeito a aplicação de metodologia ágil, podendo aplicar em projetos maiores onde possam ter mais pessoas envolvidas relatando assim os resultados obtidos.

## REFERÊNCIA BIBLIOGRÁFICA

AMBLER, Scott W. **Modelagem ágil: práticas eficazes para a Programação eXtreme e o Processo Unificado**. Tradução: Acauan Fernandes. Porto Alegre: Bookman, 2004.

BECK, Kent. **Programação Extrema (XP) explicada: acolha as mudanças**. Tradução: Adriana P. S. Machado e Natália N. P. Lopes. Porto Alegre: Bookman, 2004.

BONA, Cristina. **Avaliação de processos de software: um estudo de caso em XP e Iconix**. 2002. 122f. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 2002.

BRAUDE, Eric. **Projeto de Software. Da programação à arquitetura: uma abordagem baseada em Java**. Tradução: Edson Furmankiewicz. São Paulo: Bookman, 2005.

KENDALL, Scott. **O processo unificado explicado**. Tradução: Ana M. de Alencar Price. Porto Alegre: Bookman, 2003.

PETTRES, James F.; PEDRYCZ, Wiltod. **Engenharia de Software**. Rio de Janeiro: Campus, 2001.

PFLEEGER, Shari L. **Engenharia de Software: teoria e prática**. São Paulo: Prentice Hall, 2004.

PRESSMAN, Roger S. **Engenharia de Software**. Tradução: José Carlos Barbosa dos Santos. São Paulo: Pearson Makron Books, 2006.

SATO, Danilo T. **Uso eficaz de métricas em métodos ágeis de desenvolvimento de software**. 2007. 155f. Dissertação (Mestrado em Ciência da Computação) – Universidade de São Paulo, São Paulo, 2007.

SOMMERVILLE, Iam. **Engenharia de Software**. São Paulo: Pearson, 2003.

TELES, Vinícius Manhães. **Extreme Programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. São Paulo: Novatec, 2006.

## BIBLIOGRÁFICA COMPLEMENTAR

FILHO, Wilson de Pádua Paula. **Engenharia de Software: fundamentos, métodos e padrões**. Rio de Janeiro: LTC, 2003.

FOWLER, Martin. **The New Methodology**. Disponível em:  
<<http://simplus.com.br/artigos/a-nova-metodologia/>>. Acesso em: 12 de Set. 2007.

GUSTAFSON, David A. **Engenharia de Software**. Porto Alegre: Bookman, 2003.

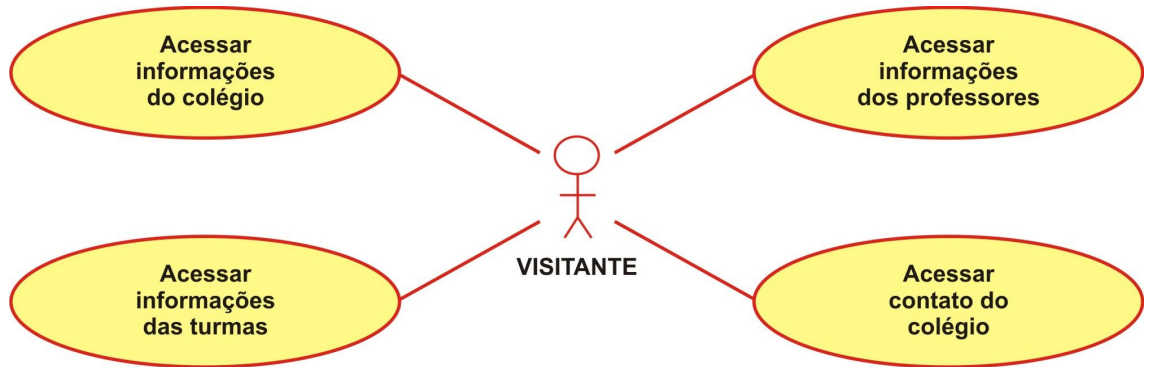
REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. Rio de Janeiro: Brasport, 2005.

SANCHEZ, Ivan. **Scrum em 2 minutos**. Disponível em:  
<<http://dojofloripa.wordpress.com/2007/02/07/scrum-em-2-minutos/>>. Acesso em: 20 de Ago. 2007.

SANCHEZ, Ivan. **XP é só o começo**. Disponível em:  
<<http://dojofloripa.wordpress.com/2007/08/18/xp-e-so-o-comeco/>>. Acesso em: 22 de Ago. 2007.

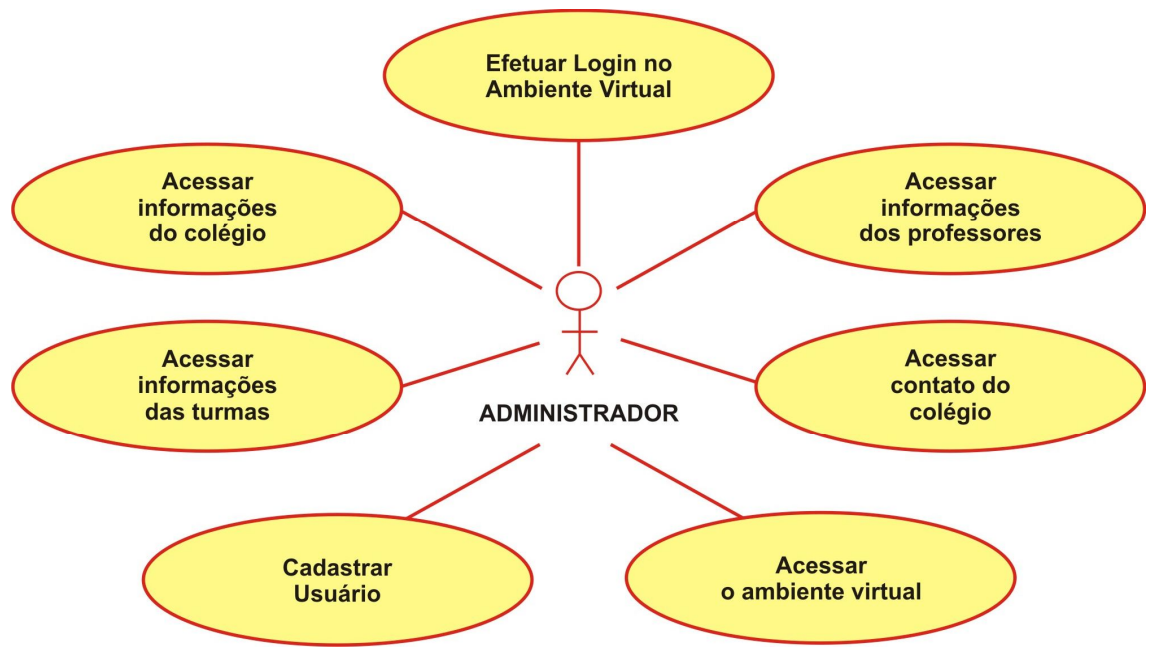
## APÊNDICE A – DIAGRAMA CASO DE USO: VISITANTE

Diagrama de caso de uso do perfil visitante.



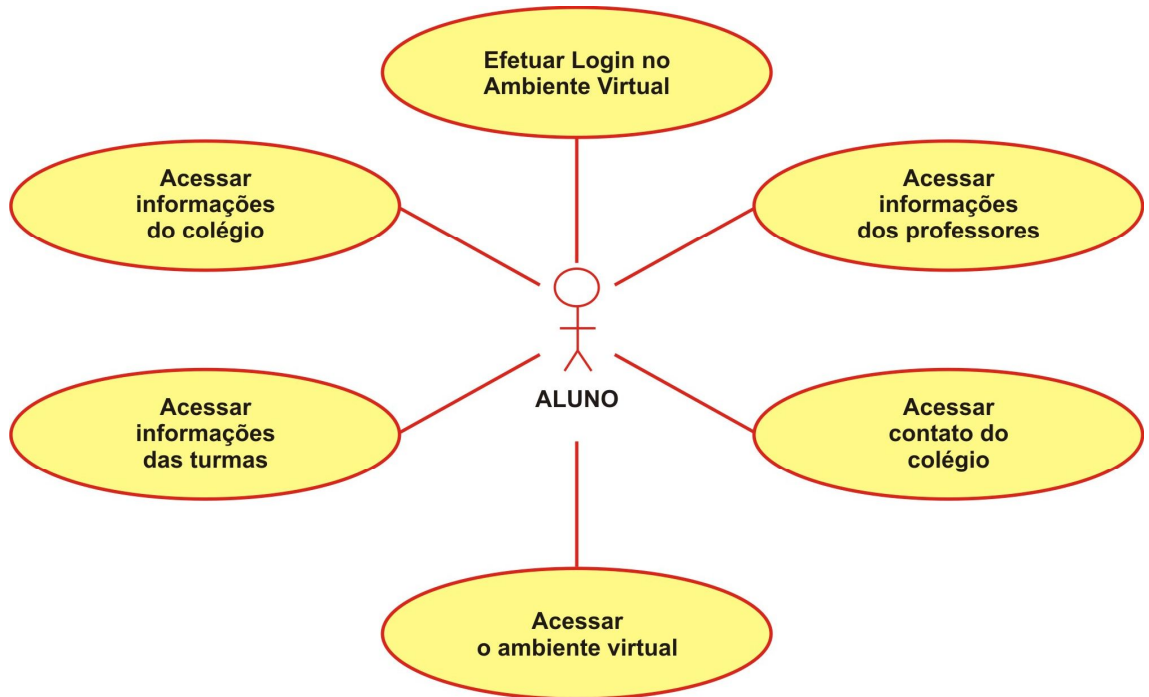
## APÊNDICE B – DIAGRAMA CASO DE USO: ADMINISTRADOR

Diagrama de caso de uso do perfil administrador.



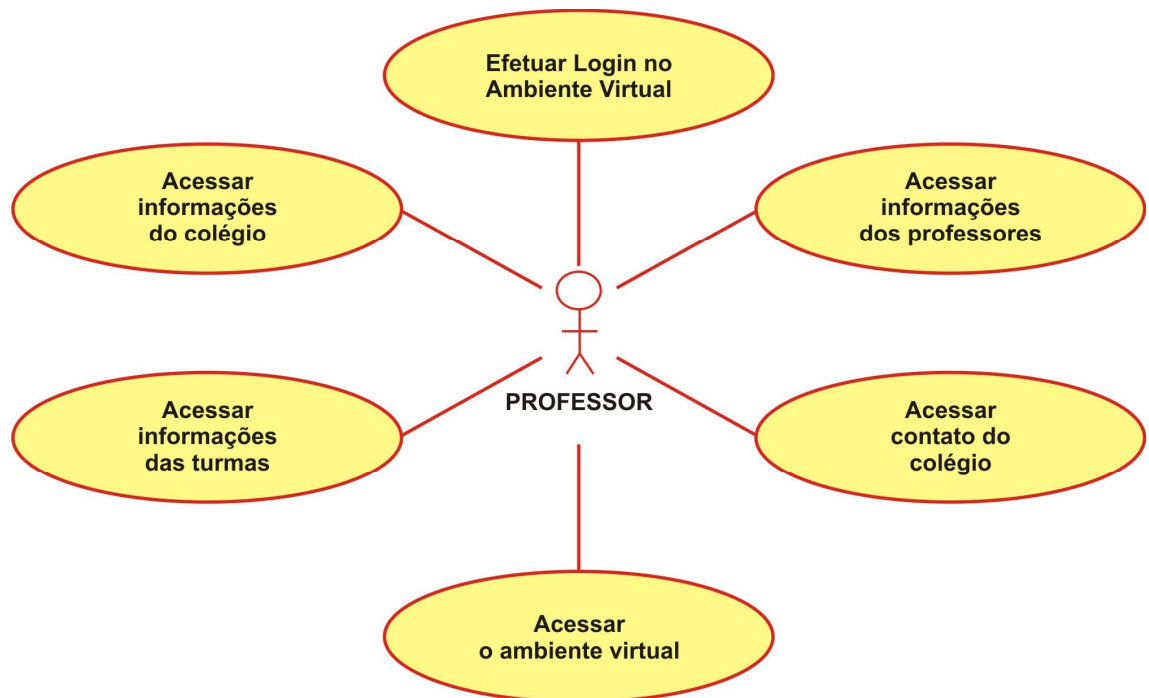
## APÊNDICE C – DIAGRAMA CASO DE USO: ALUNO

Diagrama de caso de uso do perfil aluno.

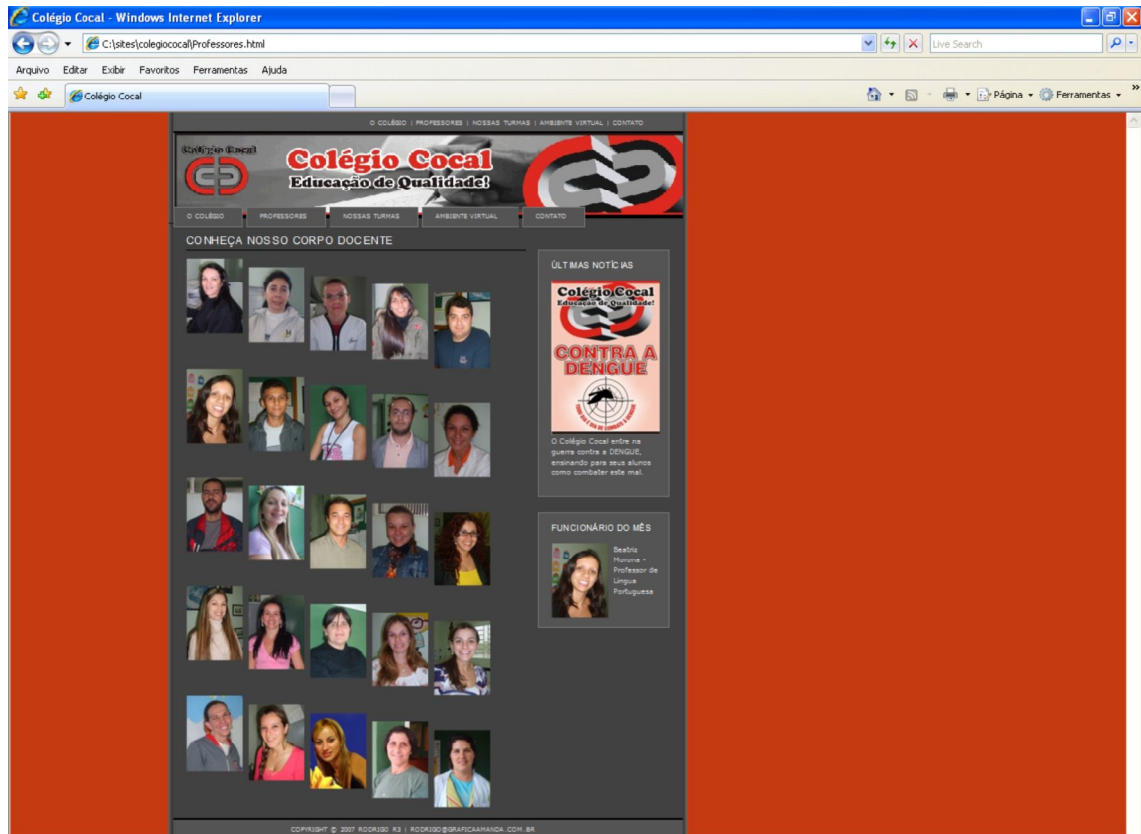


## APÊNDICE D – DIAGRAMA CASO DE USO: PROFESSORES

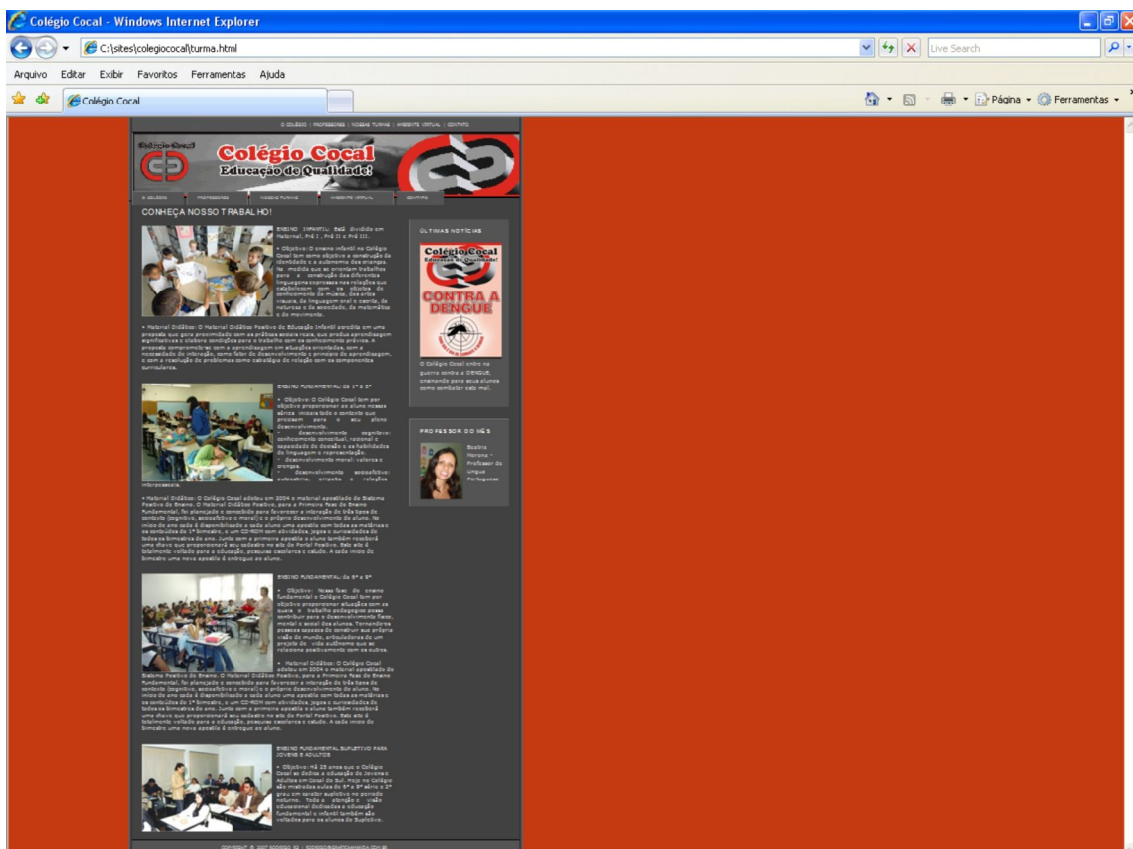
Diagrama de caso de uso do perfil professores.



## APÊNDICE E – INTERFACE DO SITE LINK PROFESSORES



# APÊNDICE F – INTERFACE DO SITE LINK TURMAS



## APÊNDICE G – INTERFACE DO SITE LINK CONTATO

The screenshot shows a Windows Internet Explorer browser window displaying the contact page of Colégio Cocal. The browser's address bar shows the local file path: C:\sites\colegiococal\Contato.html. The website header features the school's logo and navigation menu with options: O COLÉGIO, PROFESSORES, NOSSAS TURMAS, AMBIENTE VIRTUAL, and CONTATO. The main content area is titled "ENTRE EM CONTATO" and lists contact information: email (colegiococal@colegiococal.com.br), phone (48) 3447-6551, and address (Rua Paris, 260 - Bela Vista - Cep 88845-000 - Cocal do Sul - SC). It also provides a service schedule: 7:40 to 11:40 (matutino), 13:00 to 17:20 (vespertino), and 18:00 to 22:00 (noturno). To the right, there is a "ÚLTIMAS NOTÍCIAS" section with a poster for "CONTRA A DENGUE" and a "PROFESSOR DO MÊS" section featuring a photo of Beatriz Morona, Professor de Língua Portuguesa. The footer contains the copyright notice: COPYRIGHT © 2007 RODRIGO R3 | RODRIGO@GRAFICAAMANDA.COM.BR