

LÉLIO DOS SANTOS FALCÃO

**Sistema para Gerenciamento dos Conteúdos Programáticos do DRA/UFPel: um
Estudo de Caso de Desenvolvimento Baseado em UML**

Trabalho acadêmico apresentado ao Curso de Ciência da Computação da Universidade Federal de Pelotas, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Prof. Gil Carlos Rodrigues Medeiros, Msc.

Co-Orientador:

Prof. Carlos Michel Betemps, Msc.

Capão do Leão, março de 2007

Dados de catalogação na fonte:
Ubirajara Buddin Cruz – CRB-10/901
Biblioteca de Ciência & Tecnologia - UFPel

F178s Falcão, Lélío dos Santos
Sistema para gerenciamento dos conteúdos programáticos do DRA/UFPel : um estudo de caso de desenvolvimento baseado em UML / Lélío dos Santos Falcão ; orientador Gil Carlos Rodrigues Medeiros ; co-orientador Carlos Michel Betemps. – Pelotas, 2007. – 99f. il. - Monografia (Conclusão de curso). Curso de Bacharelado em Ciência da Computação. Departamento de Informática. Instituto de Física e Matemática. Universidade Federal de Pelotas. Pelotas, 2007.

1.Informática. 2.Engenharia de software. 3.Linguagens de programação. 4.Unified Modeling Language. 5.UML. 6.Workflow. 7.Extensão para aplicações web. I.Medeiros, Gil Carlos Rodrigues. II.Betemps, Carlos Michel. III.Título.

CDD: 005.74

**Sistema para Gerenciamento dos Conteúdos Programáticos do DRA/UFPel: um
Estudo de Caso de Desenvolvimento Baseado em UML**

por

Lélio dos Santos Falcão

Monografia defendida pelo autor e aprovada em 29 de junho de 2006
pela banca examinadora composta pelos professores abaixo relacionados.

Banca Examinadora:

Prof. Gil Carlos Rodrigues Medeiros, Msc.

Prof. Carlos Michel Betemps, Msc.

Profa. Eliane da Silva Alcoforado Diniz, Msc.

Profa. Tatiana Aires Tavares, Dra

Agradecimentos

Ao Prof. Gil, que sempre exigiu o máximo de mim;

Ao Prof. Michel, que nunca desistiu de mim;

Ao Prof. Campani, cujas disciplinas de teoria eu jamais poderei esquecer;

Aos demais professores do Curso, por suas contribuições à minha formação acadêmica, profissional e intelectual;

Aos meus amigos, que nunca me abandonaram e sempre me deram forças pra continuar;

Aos meus companheiros das longas jornadas de programação e estudo;

Aos meus amigos Jairo e Juliana, que me prestaram inestimável ajuda para que este trabalho fosse concluído, bem como pelo seu apoio.

Aos meus colegas de curso que, numa ocasião ou outra, contribuíram comigo, de uma forma ou de outra;

Aos meus avós, Lélío, Claudete e Ilka, que tem me apoiado e auxiliado sobremaneira nestes últimos anos;

Aos meus irmãos, que me apoiaram;

Ao meu pai e a minha mãe que, sem eles, nada disto seria possível;

E, principalmente, à minha mulher Carolina e ao meu filho Lélío, pois é para eles que eu dedico este trabalho.

Muito obrigado.

Resumo

Uma das maiores dificuldades para a criação de sistemas computacionais de qualidade é o processo de desenvolvimento. Considerando que sistemas diferentes podem necessitar de processos diferentes (MARKS, 2002), e considerando que muitas pessoas podem ter dificuldade em utilizar métodos iterativos e incrementais (SCOTT, 2003), desenvolveu-se um processo baseado em metodologias iterativa e de cascata, visando proporcionar um processo que minimize as deficiências de ambas as metodologias, utilizando UML, uma linguagem de modelagem amplamente aceita.

Como estudo de caso de tal processo híbrido, utilizou-se um sistema de gerenciamento de Programas de Disciplinas da Universidade Federal de Pelotas, concebido para um ambiente *Web*.

Aproveitando este estudo de caso, também foi feito um estudo da Extensão para Aplicações *Web*, de Conallen (2002), e um estudo sobre *Workflow*, pois o mesmo se fez necessário, dado o modelo de gerenciamento de Programas de Disciplinas.

Abstract

One major difficulty to creating quality computing systems is the development process. Taking into account that different systems may need different processes and many people may find using iterative methods difficult, a process based on waterfall and iterative methodologies was produced. This process uses UML, a broadly accepted modeling language, and attempts to minimize the flaws of both methodologies.

As study case of this hybrid process, a Universidade Federal de Pelotas' Syllabi Manager System based on Internet environment was used.

In fact, applying such a study case brought the need to study workflow and Conallen's Web Application Extension, given the Syllabi manager's model.

Lista de Figuras

Figura 1 – Exemplo de diagrama utilizando a WAE	22
Figura 2 – Uma Simplificação dos Processos e Modelos Recomendados, de Larman	24
Figura 3 – Diagrama de atividades do processo de aprovação de um Programa de Disciplina	34
Figura 4 – Casos de Uso	39
Figura 5 – Caso de Uso: Aprovar Programa de Disciplina	40
Figura 6 – Modelo Conceitual	41
Figura 7 – Rascunho da Arquitetura do sistema	42
Figura 8 – Diagrama de Seqüência: Aprovar Programa de Disciplina	43
Figura 9 – Contrato da operação: Aprovar Programa de Disciplina	44
Figura 10 – Diagrama de Estado: Aprovar Programa de Disciplina	45
Figura 11 – Caso de Uso Real: Aprovar Programa de Disciplina	46
Figura 12 – Diagrama de Interação: Aprovar Programa de Disciplina	46
Figura 13 – Diagrama de Classes de Projeto no 1º Ciclo de Desenvolvimento	47
Figura 14 – Arquitetura refinada para o 1º ciclo de desenvolvimento	48
Figura 15 – Diagrama de Artefatos <i>Web</i> e Navegação	49
Figura 16 – Sincronia de artefatos para o 2º Ciclo	50
Figura 17 – Novos Casos de Uso Atualizar Programa de Disciplina e Enviar Memorando	51
Figura 18 – Diagrama de Implantação Física	51
Figura 19 – Diagrama de Implantação Lógica	52
Figura 20 – Processo de Trabalho	59
Figura 21 – Caso de Uso: Consultar Programa	67
Figura 22 – Caso de Uso: Imprimir Programa	68
Figura 23 – Caso de Uso: Consultar Situação do Programa	68

Figura 24 – Caso de Uso: Incluir Programa de Disciplina	69
Figura 25 – Caso de Uso: Alterar Programa de Disciplina	70
Figura 26 – Diagrama de Seqüência: Consultar Programa de Disciplina	71
Figura 27 – Diagrama de Seqüência: Incluir Programa de Disciplina	72
Figura 28 – Contrato da operação: Consultar Programa	72
Figura 29 – Contrato da operação: Incluir Programa de Disciplina	73
Figura 30 – Caso de Uso Real: Consultar Programa	74
Figura 31 – Caso de Uso Real: Incluir Programa de Disciplina	74
Figura 32 – Janela do índice	74
Figura 33 – Janela de Pesquisa	75
Figura 34 – Janela do índice de Disciplinas para aprovar	75
Figura 35 – Janela de Inserção	76
Figura 36 – Janela de Aprovar	77
Figura 37 – Janela de Consultar Programa	78
Figura 38 – Diagrama de Comunicação: Consultar Programa de Disciplina	78
Figura 39 – Diagrama de Comunicação: Incluir Programa de Disciplina	79
Figura 40 – Refinamento de Caso de Uso: Imprime Programa	80
Figura 41 – Diagrama de Seqüência: Imprimir Programa de Disciplina	80
Figura 42 – Diagrama de Seqüência: Consultar Situação do Programa	81
Figura 43 – Contrato da operação: Imprimir Programa	81
Figura 44 – Contrato da operação: Consulta Situação do Programa	81
Figura 45 – Caso de Uso Real: Imprimir Programa	82
Figura 46 – Caso de Uso Real: Consulta Situação do Programa	82
Figura 47 – Janela da Pesquisa da Situação do Programa	83
Figura 48 – Janela das alterações na Situação do Programa	83
Figura 49 – Diagrama de Comunicação: Imprimir Programa	84
Figura 50 – Diagrama de Comunicação: Consultar situação do Programa	84
Figura 51 – Diagrama de Classes de Projeto para o 2º Ciclo	85
Figura 52 – Refinamento do Diagrama de Artefatos Web e Navegação, para o 2º ciclo de desenvolvimento	86
Figura 53 – Refinamento do Caso de Uso: Alterar Programa de Disciplina	87
Figura 54 – Caso de Uso: Atualizar Programa de Disciplina	88
Figura 55 – Caso de Uso: Enviar Memorando	88
Figura 56 – Caso de Uso: Incluir Programa de Disciplina	89

Figura 57 – Caso de Uso: Aprovar Programa de Disciplina	89
Figura 58 – Diagrama de Seqüência: Alterar Programa de Disciplina	90
Figura 59 – Diagrama de Seqüência: Enviar Memorando	90
Figura 60 – Diagrama de Seqüência: Atualizar Programa de Disciplina	91
Figura 61 – Contrato da Operação: Alterar Programa de Disciplina	91
Figura 62 – Contrato da Operação: Enviar Memorando	92
Figura 63 – Contrato da Operação: Atualizar Programa de Disciplina	92
Figura 64 – Caso de Uso Real: Alterar Programa de Disciplina	93
Figura 65 – Caso de Uso Real: Enviar Memorando	93
Figura 66 – Caso de Uso Real: Atualizar Programa de Disciplina	94
Figura 67 – Tela de Índice Atualizada	94
Figura 68 – Janela de inserir memorando	95
Figura 69 – Janela do índice dos Programas para alteração	95
Figura 70 – Porção inferior da janela de alteração de um Programa de Disciplina	96
Figura 71 – Diagrama de Comunicação: Atualizar Programa de Disciplina	96
Figura 72 – Diagrama de Comunicação: Alterar Programa de Disciplina	97
Figura 73 – Diagrama de Comunicação: Enviar Memorando	97
Figura 74 – Definição do Diagrama de Classes para o 3º Ciclo	98
Figura 75 – Refinamento do Diagrama de Artefatos Web e Navegação, para o 3o Ciclo	98

Lista de Tabelas

Tabela 1 – Funções básicas	38
Tabela 2 – Funções acessórias.....	38
Tabela 3 – Funções Básicas - adendo no 3º Ciclo de Desenvolvimento	50
Tabela 4 - Funções acessórias - adendo no 3o Ciclo de Desenvolvimento	50
Tabela 5 – Glossário (até o final da fase Planejar e Elaborar).....	63
Tabela 6 – Glossário	65

Lista de Abreviaturas

CGI –	Common Gateway Interface – Interface de Gateway Comum
COCEPE –	Conselho Coordenador do Ensino, Pesquisa e Extensão
CSS –	Cascading Style Sheet – Folha de Estilos em Cascata
DRA –	Departamento de Registros Acadêmicos
HTML –	Hyper Text Mark-up Language (Linguagem Marcada de Hipertexto)
OMT –	Object Modeling Technique – Técnica de Modelagem de Objeto
OOSE –	Object Oriented Software Engineering – Engenharia de Software Orientada a Objeto
PERL –	Practical Extraction and Report Language – Linguagem de Extração e Relatório Práticos
PHP –	Personal Home Page – Processador de Página Pessoal (atualmente, Hypertext Preprocessor – Pré-processador de Hipertexto)
PMR –	Processos e Modelos Recomendados
UFPEl –	Universidade Federal de Pelotas
UML –	Unified Modeling Language – Linguagem de Modelagem Unificada
URL –	Universal Resource Locator – Localizador de Recursos Universal
WAE –	Web Application Extension – Extensão para Aplicações Web
WfMC –	The Workflow Management Coalition
WfMS –	Workflow Management System – Sistema de Gerenciamento de Workflow

Sumário

1	Introdução	13
2	UML	16
2.1	Utilizando UML para descrever uma aplicação para Internet.....	17
2.1.1	Modelando Páginas <i>Web</i>	17
3	Processo de Desenvolvimento	23
3.1	Um processo recomendado	23
3.1.1	Planejar e Elaborar	24
3.1.2	Construir	27
3.1.3	Implantar.....	30
3.2	Manutenção	31
4	Estudo de Caso.....	32
4.1	O processo atual	33
4.2	O Modelo Proposto para o Processo de Gerência.....	35
5	Modelagem da Ferramenta Proposta	37
5.1	Planejar e Elaborar (1)	37
5.2	Construir (2)	42
5.2.1	Ciclo de Desenvolvimento 1 (2.1)	42
5.2.2	Ciclo de Desenvolvimento 2 (2.2)	49
5.2.3	Ciclo de Desenvolvimento 3 (2.3)	50
5.3	Implantar (3).....	51
6	Conclusões	53
	Referências	55
	Apêndice A – Workflow	57
	Apêndice B – Glossário inicial do sistema	63
	Apêndice C – Glossário Completo	65
	Apêndice D – Planejar e Elaborar.....	67

Apêndice E – 1º Ciclo de Desenvolvimento	71
Apêndice F – 2º Ciclo de Desenvolvimento	80
Apêndice G – 3º Ciclo de Desenvolvimento	87

1 Introdução

A parte mais difícil do desenvolvimento de um sistema é a modelagem do mesmo (KRUCHTEN, 2004). Logo, é imprescindível que se utilizem técnicas corretas para se facilitar este desenvolvimento, e também para que o sistema, quando completado, atenda às necessidades do cliente.

Uma vez que os sistemas de computação se tornam cada vez mais complexos, é de extrema importância que se utilizem ferramentas e técnicas adequadas, que possam facilitar o entendimento global do sistema, que facilite o reuso dos componentes, facilite a coordenação do desenvolvimento, facilite a evolução do sistema, que atenda às (reais) necessidades do cliente. A *Unified Modeling Language* (UML – Linguagem de Modelagem Unificada), em conjunto com um processo iterativo e incremental de desenvolvimento provêm estas facilidades (SCOTT, 2003).

Entretanto, meramente dizer que técnicas avançadas de desenvolvimento (tais como o desenvolvimento iterativo) vão automaticamente causar uma melhoria no desenvolvimento é um erro, segundo Dan Marks (2002). Ele diz que ambos os métodos, iterativo ou cascata possuem várias falhas, assim como pontos positivos, e ressalta que o modelo iterativo é melhor para analisar novos conceitos, enquanto que um processo em cascata pode ser melhor para sistemas menores e bem compreendidos. Além disto, ele preconiza que métodos híbridos podem ser a melhor solução para algumas aplicações. Além disto, Scott (2003) diz que muitas pessoas acham o Processo Unificado (um processo iterativo) bastante complexo.

Considerando as prerrogativas apresentadas, este trabalho fundamenta-se na experiência de desenvolver e aplicar um processo híbrido baseado em metodologias iterativa/incremental e de cascata. Visando uma possível melhoria no sistema acadêmico do Departamento de Registros Acadêmicos (DRA) da Universidade Federal de Pelotas (UFPel), decidiu-se testar tal processo

desenvolvendo uma aplicação para agilizar as transações burocráticas referentes ao processo de aprovação de Conteúdos Programáticos.

Dado o modo como este processo burocrático ocorre, esta também é uma ótima oportunidade de estudar e aplicar formas de gerenciamento de informações e aprovação de documentos entre vários participantes, de modo similar às técnicas de *Workflow* (BETEMPS, 2000), sendo esta uma ferramenta de Tecnologia de Informação bastante atual, embora consolidada, com mais de 10 anos de desenvolvimento (WfMC - THE WORKFLOW MANAGEMENT COALITION, 2004).

1.1 Objetivos

Este trabalho tem como objetivo, portanto, descrever um processo de desenvolvimento de software, baseado nas informações apresentadas por Larman (2000), que por sua vez baseou-se em vários outros trabalhos (JACOBSON, 1992; WIRFS-BROCK, 1993; BOOCH, 1994; COLEMAN, 1994; MARTIN; ODELL, 1995; RUMBAUGH, 1997 apud LARMAN, 2000) e utiliza largamente UML.

Como objetivos secundários, para a demonstração do processo e como estudo de caso, desenvolve-se uma ferramenta para mostrar uma maneira de informatizar o processo de aprovação dos Programas de Disciplinas da UFPel. De forma complementar, o desenvolvimento desta ferramenta deve implementar técnicas similares às de *Workflow* (GEORGAKOPOULOS; HORNICK; SHETH, 1995; ALLEN, 2000; BETEMPS, 2000; PRIOR, 2003), na forma de um *Workflow* estático e embutido no sistema, para o gerenciamento do processo de aprovação, que controlará a troca de informações.

Tal ferramenta – concebida para controlar todo o processo de aprovação, desde a concepção das Disciplinas até sua aprovação final e conseqüente armazenamento nos bancos de dados, aos cuidados do DRA – é implementada na linguagem PHP¹, uma linguagem voltada para o desenvolvimento de páginas pessoais *Web* executadas no servidor. Desta forma, se obtém a flexibilidade que a Internet permite aos usuários, os quais podem, então, utilizar a ferramenta em

¹ PHP era a sigla, originalmente, de *Personal Home Page*. Atualmente os desenvolvedores a utilizam como um acrônimo recursivo de *Hypertext Preprocessor*.

qualquer computador com acesso a Internet. Surge aqui, então, a necessidade de se estudar extensões de UML para Internet, tais como a sugerida por Conallen (2002).

1.2 Desenvolvimento do texto

Para desenvolver este estudo, no capítulo 2, caracteriza-se a extensibilidade da UML e apresenta-se uma extensão para modelar aplicações para a WEB. No capítulo 3, é mostrado um processo e modelos recomendados por Larman (2000), originando-se então, um processo de desenvolvimento de software sugerido para o desenvolvimento de uma aplicação, utilizando UML. O capítulo 4 mostra o sistema de aprovação de Conteúdos Programáticos da UFPel e outros dados referentes à implementação da aplicação proposta como estudo de caso. O capítulo 5 apresenta o processo de modelagem da ferramenta, baseado nas técnicas estudadas. O capítulo 6 conclui este trabalho com as considerações do autor sobre a experiência desenvolvida.

Após, são incluídos Apêndices, contendo algumas noções de *Workflow*, diagramas e outros artefatos utilizados durante o capítulo 5.

2 Extensão da UML para WEB

Os criadores da UML dizem que esta “é uma linguagem destinada a visualizar, especificar, construir e documentar os artefatos de um sistema complexo de software” (BOOCH; JACOBSON; RUMBAUGH, 2005).

Para este fim, conforme seus criadores, a UML utiliza-se principalmente de abstração. Considerando que fazendo isto se pode chegar a um melhor entendimento e comunicação, sugerindo que cada modelo pode ser expresso em diferentes níveis de fidelidade, todo sistema complexo é melhor abordado através de um pequeno grupo de modelos relativamente independentes e são mais expressivos aqueles modelos que mais se aproximam da realidade.

O desenvolvimento de modelos de sistemas de software para a indústria deve preceder a construção ou a renovação dos mesmos para que haja uma boa comunicação entre as várias partes do sistema, partes estas que são decomposições do sistema global. Como estas partes são feitas por várias equipes, é imprescindível que sejam utilizadas boas técnicas de modelagem, para assegurar o sucesso do projeto. A UML é uma resposta bem definida e amplamente aceita para esta necessidade, como uma linguagem de modelagem visual para construir sistemas orientados a objetos e baseados em componentes (BOOCH; JACOBSON; RUMBAUGH, 2005).

Booch, Rumbaugh e Jacobson observam ainda que aplicações diferentes utilizam formas diferentes de referenciar seus componentes, que a informática é uma ciência que avança rapidamente e que linguagens, metodologias e visões estão surgindo e sendo modificadas constantemente. Como consequência, a UML também define mecanismos de extensibilidade, na forma de estereótipos, valores atribuídos e restrições que podem ser aplicados aos modelos comuns da UML. Uma destas extensões é proposta por Conallen (2002), visando a utilização de UML para a modelagem de páginas para a Internet (*Web*).

2.1 Utilizando UML para descrever uma aplicação para Internet

Sendo uma página para WEB uma aplicação e, sendo a maioria dos programas (*scripts* e Java) executados na máquina do usuário baseados em objetos, é natural que, ao se pensar na modelagem de uma aplicação *Web*, se pense em UML (CONALLEN; 1999).

Feita esta associação básica, a analogia mais natural que se tem entre um artefato *Web* real e um elemento de modelagem é: *hyperlinks* seriam Associações, visto que associam uma página *Web* a outra; as páginas, em si, seriam Classes; considerando as páginas como sendo Classes, então os *scripts* que estão numa página são seus Métodos e quaisquer variáveis nesta página, sendo “globais” à mesma, seriam seus Atributos.

Aí surge um problema nesta classificação: há *scripts* que rodam no servidor², preparando a página para ser exibida, e há *scripts* que rodam no cliente³. Deste modo, torna-se difícil verificar quais atributos estão ativos no servidor e quais são ativos na página que irá interagir com o usuário. Então, ainda segundo Conallen (1999; 2002), as páginas são melhores modeladas como componentes do sistema e o simples mapeamento de uma página *Web* para uma Classe UML não vai tornar o sistema mais compreensível ou, analisando pelo outro lado, para que se compreenda uma aplicação *Web* modelada em UML, é necessário que a modelagem não seja simplesmente uma tradução dos componentes *Web* em artefatos UML.

2.1.1 Modelando Páginas *Web*

Como visto, componentes de uma página *Web* podem ser modelados em componentes de UML. Entretanto, as páginas, compiladas ou de *script*, se comportam de maneiras diferentes no servidor e no cliente – no servidor, a página tem acesso a bancos de dados, sistemas de arquivos, etc, enquanto no cliente a página pode acessar componentes de JavaScript, ActiveX e acessar até outras

² O servidor é o computador onde ficam armazenadas as páginas *Web*, onde são executados os *scripts* CGI, PHP, entre outros.

³ O cliente é o computador do usuário, onde são executados os *scripts* JavaScript.

páginas *Web* que o cliente estiver utilizando (CONALLEN, 1999). Conallen, então, sugere que algumas providências sejam tomadas.

A melhor alternativa para se evitar esta confusão é modelar as páginas *Web* no servidor e no cliente separadamente, utilizando respectivamente os estereótipos «página do servidor» e «página do cliente», separando assim os atributos e operações ativas em cada caso. Conallen coloca também um estereótipo «constrói» na relação entre a página do servidor e a do cliente, pois pode ser dito que a página do servidor constrói páginas do cliente.

A seguir, tem-se uma lista das definições da *Web Application Extension* (WAE – Extensão para Aplicações Web) que Conallen (2002) define e que são utilizadas neste trabalho:

Endereço Virtual

Item associado	Pacote
Descrição	Este é o endereço utilizado para resolver chamadas <i>Web</i> do tipo <a> e formulários. Este pacote, então, contém todas as páginas e arquivos que ficam no mesmo endereço (diretório).
Ícone	
Restrições	Nenhuma
Valores atribuídos	Podem conter o endereço a que se refere.

Página Dinâmica

Item associado	Componente
Descrição	Uma página <i>Web</i> que precisa ser compilada pelo servidor e produz dinamicamente seu conteúdo.
Ícone	
Restrições	Realiza apenas uma única «página do servidor».
Valores atribuídos	Nenhum

Página de Estilos

Item associado	Componente
Descrição	Uma página <i>Web</i> que é requisitada por outra página, e que contém seus estilos, ou ainda a definição de estilos de uma página <i>Web</i> . Usualmente, utiliza a tecnologia <i>cascading style sheet</i> (CSS – folha de estilos em cascata)
Ícone	
Restrições	Podem ter relações apenas com objetos no servidor
Valores atribuídos	Motor do <i>Script</i> : Ou a linguagem ou o motor que deve ser usado para executar ou interpretar esta página (JavaScript, PERL, PHP, etc)

Página Estática

Item associado
Descrição
Ícone

Componente
Esta é a página requerida pelo navegador (*browser*) *Web*.



Restrições
Valores atribuídos

Podem realizar apenas uma única «página do cliente».
Nenhum.

Página do Servidor

Item associado
Descrição

Classe
Uma página *Web* que precisa ser compilada pelo servidor; tem acesso a bancos de dados, sistemas, lógica de negócios. As operações de seus objetos representam as funções do *script* e seus atributos representam as variáveis que são do escopo do servidor.
Ela constrói uma «página do cliente».

Ícone



Restrições
Valores atribuídos

Podem ter relacionamentos simples apenas com recursos do servidor
Nenhum

Página do Cliente

Item associado
Descrição

Classe
Uma instância de uma página do cliente é uma página *Web* formatada HTML, que é uma mistura de dados, apresentação e até lógica. São lidas pelos navegadores e pode conter *scripts* interpretados por estes. Podem ter associações com outras páginas de cliente ou de servidor.

Ícone



Restrições
Valores atribuídos

Nenhuma
TagTítulo: O título da página como mostrado no navegador;
TagBase: A URL base para derreferenciar URLs relativas;
TagCorpo: O conjunto de atributos para a *tag* *<body>*, que define o fundo da página e os atributos padrão do texto. E todos os atributos que são atribuídos dentro da seção *<HEAD>* de uma página HTML.
Além disto, é importante salientar que as funções são mapeadas para as operações da classe e os atributos destas funções são mapeados para os Atributos da classe, bem como a navegação entre uma página e outra (geralmente através da *tag* *<A>*) são feitas através de associações para outras classes.

Formulário

Item associado
Descrição

Classe

Uma classe estereotipada como um «formulário» é uma coleção de campos a serem preenchidos numa página do cliente. Uma classe formulário é mapeada diretamente para a *tag* <form> de HTML. Os atributos desta classe representam os campos de entrada do formulário de HTML, e seus tipos são definidos pelos elementos HTML (texto, botão de opção, radio, etc).

Um «formulário» não possui operações, pois não podem ser encapsuladas num formulário. Quaisquer operações que interajam com o formulário são de propriedade da página que o contém.

Um formulário compõe uma página e se relaciona com uma «página do servidor» através do relacionamento «submete».

Ícone



Restrições
Valores atribuídos

Nenhuma

O método – *GET* ou *POST* – usado para submeter os dados à URL que agirá.

Página de quadros

Item associado
Descrição

Classe

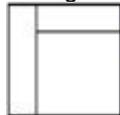
Uma página de quadros é uma página que contém múltiplas páginas *Web*. A área retangular visível é dividida em quadros retangulares menores. Cada quadro pode ser associado através de um «alvo» nominado, embora não necessariamente.

Pode conter (agregar) páginas *Web* ou outra(s) página(s) de quadros.

Uma classe estereotipada página de quadros mapeia diretamente para uma página *Web* de quadros e a *tag* de HTML <frame>.

Como a página de quadros é também uma «página do cliente», também pode ter operações e atributos, mas estes serão ativados apenas se o navegador não puder interpretar os quadros.

Ícone



Restrições
Valores atribuídos

Deve conter pelo menos uma «página do cliente» ou um «alvo».

Linhas: o valor do atributo linhas do *tag* <frameset> de HTML. Delimita a altura das linhas;

Colunas: o valor do atributo colunas do *tag* <frameset> de HTML. Delimita a largura das colunas.

Alvo

Item associado
Descrição

Classe

É um compartimento nomeado de uma janela de navegador em que as páginas podem ser exibidas. O nome da classe estereotipada é o nome do alvo. Tipicamente, um alvo é um quadro definido por uma página de quadros, mas também pode ser uma nova instância do navegador ou outra janela.

As associações «link alvejado» especificam em qual quadro uma página *Web* deve ser exibida.

Ícone



Restrições
Valores atribuídos

Nenhum

Linha: o número da linha em que a página deve aparecer;

Coluna: o número da coluna em que a página deve aparecer.

Link

Item associado	Associação
Descrição	Um <i>link</i> é um apontador de uma página do cliente para outra página, seja do cliente ou não.
Ícone	Nenhum
Restrições	Nenhum
Valores atribuídos	Parâmetros: Uma lista com os nomes dos parâmetros que devem ser passados com a requisição para a página associada. Pode-se referenciar estes valores tanto por uma classe de associação estereotipada «parâmetros do link», ou simplesmente como um valor atribuído.

Link Alvejado

Item associado	Associação enésima
Descrição	Similar à associação «link», uma associação « <i>link</i> alvejado» é uma associação que aponta para outra página, seja do cliente ou não. Entretanto, a associação «link alvejado» especifica que a página solicitada seja exibida em algum quadro específico de uma «página de quadros». Esta associação não é diretamente mapeada do código HTML <i>target</i> . Ao invés disto ele é associado a uma página do cliente através de uma associação «link» e, por sua vez, esta associado a um alvo e a uma página, fazendo a associação entre eles.
Ícone	
Restrições	Deve referenciar a exatamente um «alvo» e também deve ter uma associação direcional a uma «página do cliente», uma «página do servidor» ou a uma «página de quadros».
Valores atribuídos	Nenhum.

Submete

Item associado	Associação
Descrição	Uma associação «submete» indica para qual página os campos de um formulário serão enviados.
Ícone	Nenhum
Restrições	Nenhuma
Valores atribuídos	Nenhum.

Constrói

Item associado	Associação
Descrição	A associação «constrói» é uma relação especial que liga as páginas do servidor às páginas do cliente. Páginas do servidor existem apenas no servidor <i>Web</i> . Elas são usadas para construir as páginas do cliente. Esta associação identifica qual página do servidor é responsável pela criação de uma página do cliente. Esta é uma relação direcional, uma vez que a página do cliente não tem o conhecimento de como surgiu.
Ícone	Nenhum
Restrições	Nenhum
Valores atribuídos	Nenhum

Página de Estilos

Item associado	Classe
Descrição	Uma página <i>Web</i> que é requisitada por outra página, e que contém seus estilos, ou ainda a definição de estilos de uma página <i>Web</i> . Usualmente, utiliza a tecnologia CSS
Ícone	
Restrições	Podem ter relações apenas com objetos no servidor
Valores atribuídos	Motor do <i>Script</i> : Ou a linguagem ou o motor que deve ser usado para executar ou interpretar esta página (JavaScript, PERL, PHP, etc)

src

Item associado	Associação
Descrição	Uma associação «src» é diretamente mapeada para a <i>tag</i> <src>
Ícone	Nenhum
Restrições	Nenhum
Valores atribuídos	Nenhum

Um exemplo de como estes diversos itens se associam pode ser visto na

Fig. 1.

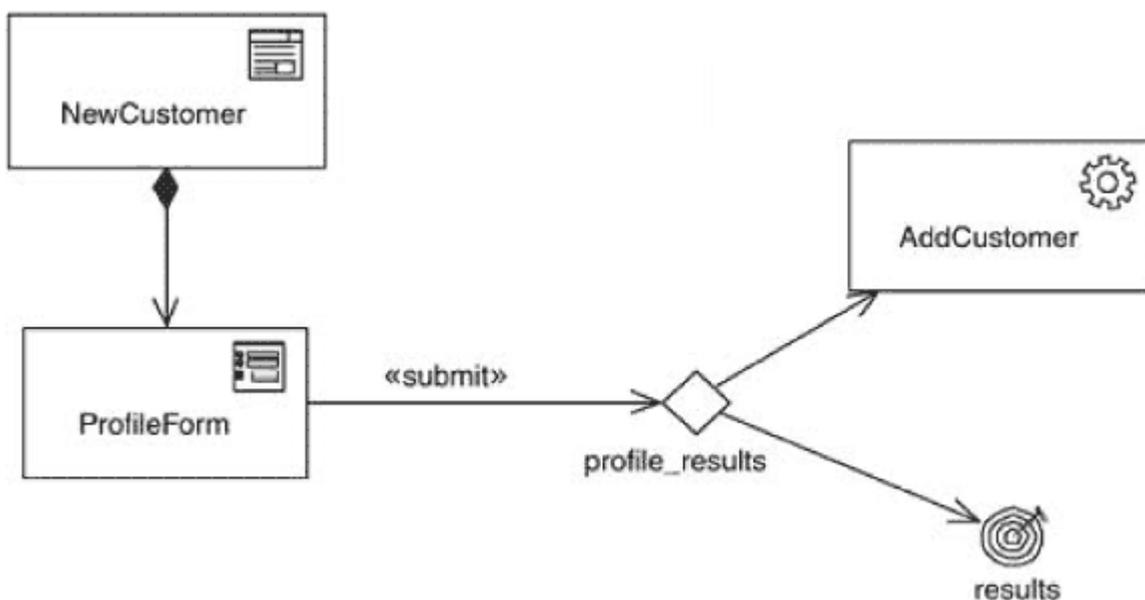


Figura 1 – Exemplo de diagrama utilizando a WAE

Fonte: Conallen, 2002

3 Processo de Desenvolvimento

A UML não define nenhum processo, podendo ser utilizada em diversos processos de desenvolvimento de software, principalmente para poder aumentar a aceitação da linguagem e por poder se ajustar aos mais diferentes processos, uma vez que cada equipe, problema, ferramenta, etc, pode necessitar de processos bastante diferentes (LARMAN, 2000; BOOCH; JACOBSON; RUMBAUGH, 2005).

Larman, então, apresenta processos e modelos recomendados (PMR) para um processo de desenvolvimento em seu livro (Larman, 2000), baseado em vários autores e técnicas já conhecidas, incluindo-se Booch (1994 apud LARMAN, 2000), OOSE (JACOBSON, 1992 apud LARMAN, 2000), OMT (RUMBAUGH, 1991 apud LARMAN, 2000) e, principalmente, na UML, a qual é utilizada neste processo.

Também é importante considerar a manutenção de um sistema, ao se desenvolver o mesmo. Mais adiante, este capítulo trata brevemente sobre manutenção em sistemas.

3.1 Um processo recomendado

O processo pode se dividir em três fases, que se subdividem em passos menores, que continuam a se dividir em tarefas mais simples (Larman, 2000). Este processo é exibido na Fig. 2, e cada tarefa é comentada em detalhes a seguir.

1. Planejar e Elaborar
 - 1.1. Definir rascunho do plano;
 - 1.2. Criar relatório de investigação preliminar;
 - 1.3. Definir requisitos;
 - 1.4. Registrar termos no glossário (Esta tarefa inicia aqui);
 - 1.5. Implementar protótipo (opcional, não necessariamente nesta ordem);
 - 1.6. Definir Casos de Usos (alto nível e essenciais);
 - 1.7. Definir rascunhos do modelo conceitual (pode ser postergado);
 - 1.8. Definir rascunho da arquitetura do sistema (não necessariamente neste ponto mas, assim que começar, é feito continuamente);
 - 1.9. Aperfeiçoar plano;
2. Construir
 - 2.1. Ciclo de Desenvolvimento 1
 - 2.1.1. Refinar o Plano;
 - 2.1.2. Sincronizar Artefatos;
 - 2.1.3. Analisar
 - 2.1.3.1. Definir Casos de Uso Essenciais (se ainda não foi feito);
 - 2.1.3.2. Refinar Diagramas de Casos de Uso;
 - 2.1.3.3. Refinar o modelo conceitual;
 - 2.1.3.4. Refinar o glossário (continuamente);
 - 2.1.3.5. Definir Diagramas de Seqüência;
 - 2.1.3.6. Definir contratos da operação;
 - 2.1.3.7. Definir Diagramas de Estado (opcional);
 - 2.1.4. Projetar
 - 2.1.4.1. Definir Casos de Uso reais;
 - 2.1.4.2. Definir Interfaces de Usuários e “StoryBoards” (em paralelo com os Casos de Uso Reais);
 - 2.1.4.3. Definir Diagramas de Interação;
 - 2.1.4.4. Definir Diagramas de Classes de Projeto (em paralelo com os Diagramas de Interação);
 - 2.1.4.5. Refinar a arquitetura do sistema (não necessariamente nesta ordem);
 - 2.1.5. Programar;
 - 2.1.6. Testar;
 - 2.2. Ciclo de Desenvolvimento n
3. Implantar

Figura 2 – Uma Simplificação dos Processos e Modelos Recomendados por Larman

3.1.1 Planejar e Elaborar

Inicia-se o processo de desenvolvimento verificando a aplicação a ser desenvolvida. Nesta etapa, Planejar e Elaborar, são feitos o estudo da necessidade da aplicação e os esboços iniciais do projeto.

3.1.1.1 Definir rascunho do plano

A primeira tarefa é definir um rascunho do plano, que consiste em verificar requisitos básicos, definir um cronograma, verificar recursos e definir um orçamento preliminar com base nas informações coletadas.

Note que “requisitos básicos” significa exatamente isto: nesta tarefa apenas se percebe qual a utilidade geral do sistema (por exemplo: vendas, controle de estoque), algumas utilidades necessárias ao cliente específico e talvez algumas

funcionalidades que o cliente ache que poderão ser o diferencial. Idealmente, se deveria colher cerca de metade da funcionalidade do sistema nesta etapa, deixando a cargo de consultas futuras a possibilidade de se incorporar novas funcionalidades ao sistema. Os demais trabalhos a serem realizados nesta tarefa devem levar em consideração que estes requisitos são incompletos, o que acaba fazendo com que a experiência da equipe contribua para a avaliação destes dados.

3.1.1.2 *Criar relatório de investigação preliminar*

A seguir, deve-se fazer um relatório de investigação preliminar, fazendo-se as justificativas para a execução do plano do sistema, buscando-se alternativas e verificando se realmente há a necessidade da implementação.

3.1.1.3 *Definir requisitos*

Então, ao se decidir por implementar o sistema, deve-se colher os requisitos do mesmo junto ao cliente. Nesta etapa, deve-se colher o máximo de requisitos, de tal modo que seja possível definir de antemão todos os Ciclos de Desenvolvimento. Apesar de estes requisitos poderem ser alterados nas tarefas posteriores, com a alteração ou exclusão dos já definidos e a inclusão de novos requisitos, ter uma idéia clara de como abordar o desenvolvimento no sistema é o que se deseja.

Para a análise e facilidade de consulta, Larman (2000) sugere que se façam tabelas com as funções que o sistema deve exercer.

Cada tabela deve possuir um nome, de acordo com suas funções, sejam estas funções básicas, sejam elas funções acessórias, de gerência, de negócio, etc.

Esta tabela contém três campos, um para o número de referência (o “índice” do requisito), um para a descrição da função e um terceiro para identificar a importância da função, que pode ser:

Evidente – Deve ser implementada e o usuário deve saber que é executada;

Ocultas – Deve ser implementada e o usuário não deve saber que é executada;

Enfeite/Decoração – Pode ser ou não implementado, e não altera muito a aplicação.

3.1.1.4 *Registrar termos no glossário*

Será utilizado, neste trabalho, uma versão semelhante à apresentada por Larman (2000), sendo uma tabela contendo o nome do item e sua respectiva descrição.

O glossário é uma lista com vários nomes utilizados na implementação, logo, é de grande importância que se inicie a registrar termos nele assim que forem surgindo. Estes nomes podem ser atributos de uma Classe, a própria Classe, pode ser um Caso de Uso, um Ator, um relacionamento, etc.

3.1.1.5 *Implementar protótipo*

A seguir, pode-se implementar um protótipo. Esta tarefa consiste em fazer um protótipo do sistema, para auxiliar na compreensão do problema e detectar possíveis problemas e requisitos do sistema.

3.1.1.6 *Definir Casos de Usos (alto nível e essenciais)*

Depois de se definir os requisitos, Larman (2000) sugere, para a descrição dos Casos de Uso, uma tabela contendo o nome do Caso de Uso, os Atores envolvidos, a finalidade do Caso de Uso, bem como um resumo do mesmo, o seu Tipo (que será visto a seguir), quais requisitos ele atende, a Seqüência Típica de Eventos e as Seqüências alternativas.

Os Tipos de um caso de uso podem ser:

1. Quanto à importância, podem ser **primários** (são os usos comuns principais do sistema), **secundários** (casos menos importantes ou mais raros) ou **opcionais** (podem não ser considerados);
2. Quanto à descrição, podem ser **reais** (com os nomes dos botões, tabelas e campos da aplicação real) ou **essenciais** (apenas descrevendo as ações, livres dos termos da aplicação). Na prática, a maioria dos Casos de Uso pode ser descrita mesclando-se a terminologia real com a essencial, pois há várias formas de descrever os casos de uso, deixando uma “zona cinza” onde os casos de uso poderiam ser ditos como reais ou essenciais, dependendo do autor.

3.1.1.7 *Definir rascunhos do modelo conceitual*

As últimas tarefas do planejamento incluem esboçar um Diagrama de Classes contendo os relacionamentos entre os diversos componentes conceituais do sistema. Neste Diagrama, se utiliza apenas alguns atributos necessários à compreensão do domínio, os relacionamentos e a multiplicidade destes relacionamentos.

3.1.1.8 *Definir rascunho da arquitetura do sistema*

Ao mesmo tempo, é definido um primeiro esboço do Diagrama de Arquitetura. Este esboço é refinado à medida que se avança na implementação da aplicação.

3.1.1.9 *Aperfeiçoar o plano*

Então, no final da fase de planejar e elaborar, de posse de todas as informações coletadas, pode-se fazer uma melhor estimativa de tempo e orçamento. Também, neste momento, se elabora um esboço de como distribuir os Casos de Uso pelos diversos Ciclos de Desenvolvimento, verificando a melhor alternativa de abordá-los, priorizando aqueles que foram detectados como Primários durante a tarefa 1.6, passando então a atacar os Secundários e, por fim, abordando os de Enfeite/Decoração. É importante salientar que uma boa abordagem do problema não envolve atacar todos os casos de uso de uma mesma prioridade ao mesmo tempo, mas dividi-los entre os diversos ciclos de desenvolvimento, podendo até mesmo dividir um único caso de uso em mais de um ciclo, facilitando assim a exploração da solução deste (LARMAN, 2000; SCOTT, 2003).

Esta não é uma escolha imutável, é apenas uma idéia geral: à medida que novos Casos de Uso e/ou funções surgem ou são detectados, é possível inserir novos Ciclos de Desenvolvimento entre os que já foram definidos anteriormente, ou alterar a ordem em que eles foram previstos (CONALLEN, 2004).

3.1.2 **Construir**

A segunda fase do processo de desenvolvimento é a construção. Nesta fase, depois de ter sido feita a análise da aplicação, a mesma é mais detalhadamente analisada, projetada, programada e testada, gradualmente.

É definido um primeiro esboço de um diagrama da arquitetura do sistema, tipicamente na forma de um Diagrama de Componentes, mas poderia também ser utilizado um Diagrama de Implantação. Este esboço é refinado à medida que se avança na implementação da aplicação.

3.1.2.1 *Ciclo de Desenvolvimento 1 (ou 1º Ciclo de Desenvolvimento)*

No primeiro Ciclo de Desenvolvimento são implementadas as funções mais importantes do sistema, detectadas na fase anterior, através dos casos de uso. Para isto, devem ser observadas as seguintes tarefas:

a) Refinar o Plano – Esta é a tarefa onde se deve verificar, junto ao cliente, se os artefatos implementados até o momento, os requisitos e outras funcionalidades do sistema estão de acordo com as expectativas do cliente e, se não estiverem, corrigi-las no modelo.

a) Sincronizar Artefatos – Aqui, adequam-se os diagramas aos artefatos implementados, os quais invariavelmente começam a divergir dos diagramas definidos anteriormente (Esta etapa só tem validade a partir do 2º Ciclo de Desenvolvimento, ou quando o protótipo é implementado e percebe-se, desde então, divergências com os diagramas rascunhados no início do projeto);

b) Analisar – Então, verifica-se os conceitos que serão abordados pelo ciclo em questão com mais detalhamento, iniciando-se por **Definir Casos de Uso Essenciais**, que ainda não tenham sido definidos na etapa descrita anteriormente (3.5.1.6). Também, neste ponto, pode-se verificar se há mais algum caso de uso ou requisito que deva ser considerado no sistema e que poderá ou não ser abordado neste Ciclo.

Após, deve-se **Refinar os Diagramas de Caso de Uso**, aprimorando os que já foram criados e que serão utilizados neste ciclo, adequando-os à medida que se desenvolve o sistema. Também é desejável que se crie Casos de Uso mais próximos dos reais.

A seguir, **Refina-se o Modelo Conceitual**, tornando-o mais próximo do que será abordado no ciclo de desenvolvimento.

Enquanto isto, deve-se continuamente **Refinar o Glossário**, à medida que novos artefatos vão sendo implementados e novos conceitos introduzidos.

Deve-se **Definir Diagramas de Seqüência**, para ilustrar como os atores e os processos interagem entre si; Uma vez que os processos são definidos sem muita clareza (do ponto de vista de implementação), estes diagramas são úteis para visualizar a ordem em que os objetos e os atores trocam mensagens e quais os eventos são desencadeados a partir destas mensagens, bem como ter uma idéia geral da natureza destas mensagens.

De posse dos Diagramas de Seqüência, pode-se **Definir Contratos da Operação**, para acrescentar ainda mais detalhes aos diagramas, uma vez que nestes não são detalhadas as funcionalidades associadas às operações invocadas, nem tampouco são detalhadas as pré e pós-condições de um Caso de Uso. Larman (2000) descreve um contrato como segue:

- Um contrato possui um nome, na forma de um nome de operação e seus parâmetros, suas responsabilidades, seu tipo (conceito, sistema ou interface), as referências cruzadas, notas (do projeto, algoritmos, etc) exceções, saída e as pré e pós-condições;

- Sugere que se identifique as operações a partir dos Diagramas de Seqüência e se crie um contrato para cada operação, então se comece a descrevê-los a partir de suas responsabilidades e suas pós-condições, utilizando, entre outras categorias, a criação/destruição de instâncias, a modificação de atributos e as associações formadas e desfeitas.

Definidos os contratos, se necessário, **Define-se Diagramas de Estado** para os Casos de Uso analisados.

c) Projetar – Uma vez terminada a fase de Análise, passa-se ao passo de Projeto para que se possa desenvolver alguns diagramas e modelos que servem para a definição das classes e interfaces que deverão ser implementadas no *software*.

As primeiras tarefas deste passo são **Definir Casos de Uso reais** e **Definir Interfaces de Usuários e “Storyboards”**, mostrando como serão as telas do software, com seus respectivos campos e botões, e especificando no Caso de Uso correspondente a qual item da tela a operação faz referência, além de exibir, passo a passo, como se executa cada Caso de Uso..

Então, **Definem-se Diagramas de Interação** (utilizando Diagramas de Comunicação) com base nos Casos de Uso percebidos anteriormente. Deve ser feito ao mesmo tempo em que Define Diagramas de Classes, constando as operações que cada Classe deverá possuir/oferecer, podendo ser descritas na íntegra, os atributos e seus tipos correspondentes e também a visibilidade destas operações e atributos. Apesar de bastante semelhantes aos diagramas de seqüência criados na tarefa 2.1.2.5, nesta etapa as operações são descritas, enquanto que naquela tarefa se concentrava apenas em perceber as relações entre os diversos objetos do sistema.

Ao mesmo tempo em que se definem os diagramas de interação, **Definem-se Diagramas de Classes de Projeto**, quando se percebe os atributos e operações de cada classe.

A qualquer momento desta fase, pode-se **Refinar a arquitetura do sistema**, aplicando as alterações encontradas durante o Ciclo de Desenvolvimento.

d) Programar – Nesta fase é feito o mapeamento dos artefatos do projeto para o código que resultará no sistema de computação. Apesar de parecer um trabalho mecânico, deve-se sempre ter em mente que a modelagem de um sistema é uma maneira de fazer com que o sistema fique compreensível para as pessoas. Logo, muitas diferenças existirão entre o modelo e o código (CONALLEN, 2002). Além disto, algumas alterações podem acontecer, especialmente em virtude das limitações das linguagens utilizadas na programação, acarretando uma ou outra alteração no modelo (que será corrigido posteriormente).

e) Testar – Assim que os modelos são implementados, os mesmos devem ser testados, um a um. Eventualmente, devem ser corrigidos, causando alteração do projeto, bem como uma reconstrução de alguns componentes.

3.1.2.2 *Ciclo de Desenvolvimento n*

Uma vez terminada a implementação das operações básicas do sistema, passa-se à implementação das operações que implementam casos de uso secundários e opcionais e das funções acessórias. Obviamente, é possível destacar dentre estas operações aquelas que forem mais importantes, delegando-as para Ciclos anteriores, deixando as outras funcionalidades para futuros Ciclos.

Nestes Ciclos seguintes também se podem refinar algumas funcionalidades implementadas nos Ciclos anteriores, ampliando-lhes a operacionalidade ou acrescentando-lhes adereços.

Note que, à medida que o modelo torna-se mais complexo, pode ser necessário incluir no projeto alguns itens de agrupamento da UML (pacotes), para tornar mais fácil a divisão das diversas áreas abordadas pelo sistema.

Além disto, à medida que o sistema vai sendo modelado de forma incremental, novos Casos de Uso podem ser detectados e deverão ser estudados/modelados nos Ciclos de Desenvolvimento posteriores, na ordem que a equipe achar mais conveniente, utilizando os mesmos parâmetros comentados na seção 3.1.1.9.

3.1.3 **Implantar**

Após o sistema ter sido concluído, passa-se à implantação do mesmo, integrando-o aos outros sistemas já existentes, substituindo os antigos sistemas, ou mesmo colocando-o em um arquivo auto-instalador, além de, possivelmente, escrever manuais de uso e/ou treinar os usuários para utilizar o sistema.

3.2 Manutenção

É importante que se tenha em mente que, assim que um sistema for posto à disposição dos clientes (ou seja, após sua implantação), o mesmo estará sujeito a falhas, quer por descoberta dos clientes, quer por estar constantemente envelhecendo (SOMMERVILLE, 2003). Então, uma vez que o sistema precisará, necessariamente de manutenção, é importante que se utilize uma técnica que permita que partes do sistema sejam trocadas sem afetar as outras, tais como num paradigma de programação orientada a objetos.

Sommerville (2003) cita três tipos diferentes de manutenção:

- **Manutenção para reparar os defeitos no *software***, que são os mais simples de se resolver, dado que é necessário corrigir, em geral, apenas a programação;

- **Manutenção para adaptar o *software* a um ambiente operacional diferente**, que ocorre em geral em função de troca de hardware ou sistema operacional diferente e;

- **Manutenção para fazer acréscimos à funcionalidade do sistema ou modificá-la**, que é a mais difícil manutenção, uma vez que, em geral, o acréscimo ou modificação dos requisitos modificam, principalmente, o modelo do sistema, que deve ser remodelado.

Em geral, os processos de manutenção devem iniciar logo após a implementação de um sistema, cada um tendo uma duração aproximadamente igual ao de uma iteração de desenvolvimento, bem como são, basicamente, tratados do mesmo modo (SCOTT, 2003; SOMMERVILLE, 2003). Quanto mais tempo um sistema permanece em funcionamento, mais difíceis (e mais demorados) vão se tornando os ciclos de manutenção. Isto ocorre por causa de defeitos na manutenção que, invariavelmente, ocorrem, e delimitam a vida útil de um sistema.

4 Estudo de Caso

Neste capítulo, é apresentado o caso a ser analisado e desenvolvido com base na UML, no processo de desenvolvimento de *software* adaptado de Larman (2000) e nas técnicas similares a *Workflow* (algumas informações sobre *workflow* podem ser encontradas no Apêndice A). O caso a ser estudado (escolhido) é uma proposta de informatização do processo de aprovação e manutenção dos Conteúdos Programáticos de Disciplinas da UFPel via *Web*.

O processo descrito no Capítulo 3 é empregado para a construção da ferramenta que controlará todo o processo de aprovação, desde a concepção das Disciplinas, quer seja pelo Departamento, quer seja pelo Colegiado de Curso, até sua aprovação final e conseqüente armazenamento nos bancos de dados aos cuidados do DRA.

Segundo o Regimento da Universidade Federal de Pelotas (2007):

Art. 172 – Disciplina é um conjunto de estudos e atividades de um setor definido de conhecimentos, correspondente a um programa a ser desenvolvido, no máximo, em um período letivo e com carga mínima de 2 (dois) créditos.

Parágrafo Único: Exceptuam-se as disciplinas optativas dos cursos de graduação, as disciplinas da pós-graduação e a Prática Desportiva.

O Conselho Coordenador do Ensino, Pesquisa e Extensão (COCEPE) é o órgão responsável pela aprovação e criação dos Currículos, sendo portanto o órgão responsável pela aprovação das Disciplinas. Quem requer a aprovação dos Currículos são os Colegiados dos respectivos Cursos.

Qualquer alteração no Currículo de um Curso requer um processo para a aprovação destas alterações. Portanto, qualquer alteração de uma Disciplina, deverá passar pela aprovação dos órgãos envolvidos. Sempre que a alteração diz respeito

à concepção da Disciplina, os reflexos sobre a concepção dos Currículos são significativos, exigindo uma avaliação mais detalhada e passando pela aprovação de todos os níveis regimentais para aprovação de um Currículo. Alterações feitas com uma grande regularidade, relativas à forma de execução, tais como a substituição de professores que ministrarão as aulas, a metodologia empregada por eles, bem como o cronograma destas atividades não geram a necessidade de alteração curricular.

4.1 O processo atual

O atual processo de inclusão, alteração ou eliminação de uma Disciplina é feito através de reuniões dos órgãos colegiados envolvidos, a partir das quais uma proposta original é avaliada e enviada, de uma instância à outra, acompanhada do ofício de deferimento da mesma, até que a Disciplina seja totalmente aceita e oficialmente integrada ao(s) currículo(s).

Para iniciar o processo de inclusão/alteração de uma Disciplina, é necessário haver um desejo do Departamento em questão, ou mesmo do Colegiado do Curso ao qual esta Disciplina faça parte do Currículo. Note-se que a alteração pode se dar pelos mais variados motivos, como alteração da carga horária (e a respectiva mudança nos créditos da Disciplina), alteração do Conteúdo Programático, alteração do nome, extinção ou criação de uma Disciplina, divisão em duas Disciplinas, ou fusão de duas Disciplinas, etc.

Independentemente da origem da solicitação, o processo deve se iniciar no Departamento ao qual a Disciplina é vinculada, quer seja por vontade do próprio Departamento, quer seja através da comunicação a este Departamento do desejo do Colegiado de iniciar o processo.

É feita, então, uma reunião de Departamento, onde são discutidos todos os pormenores da Disciplina, encaminhando-se, ao final, uma proposta, na forma de um Programa de Disciplina, ao(s) Colegiado(s) do(s) Curso(s) ao(s) qual(is) esta Disciplina faça/fará parte.

Esta proposta dentro do Colegiado deverá ser votada e, em caso de aprovação, deverá ser enviada ao COCEPE para aprovação final. Caso não seja aprovada no Colegiado, ela pode ser alterada, voltando ao Departamento de onde se originou, para que haja apreciação das alterações e posterior aprovação.

No caso de uma revisão do Currículo, o Colegiado enviará ao COCEPE toda a lista de Disciplinas, para que possam ser aprovadas.

No COCEPE, em caso de indeferimento, o Programa de Disciplina deverá ser devolvido para o Colegiado, que deverá remeter ao Departamento, para possíveis correções, ou seja, reiniciar o processo. Em caso de aprovação, a Disciplina é incluída, sendo a mesma encaminhada ao DRA para armazenamento e futuras consultas.

Esta seqüência de aprovação é detalhada na Fig. 3, utilizando-se um Diagrama de Atividades da UML (BOOCH; JACOBSON; RUMBAUGH, 2005) para descrevê-la. Pode-se notar o fluxo do trabalho de aprovação, o que nos remete diretamente a pensar em *workflow* (“fluxo de trabalho”).

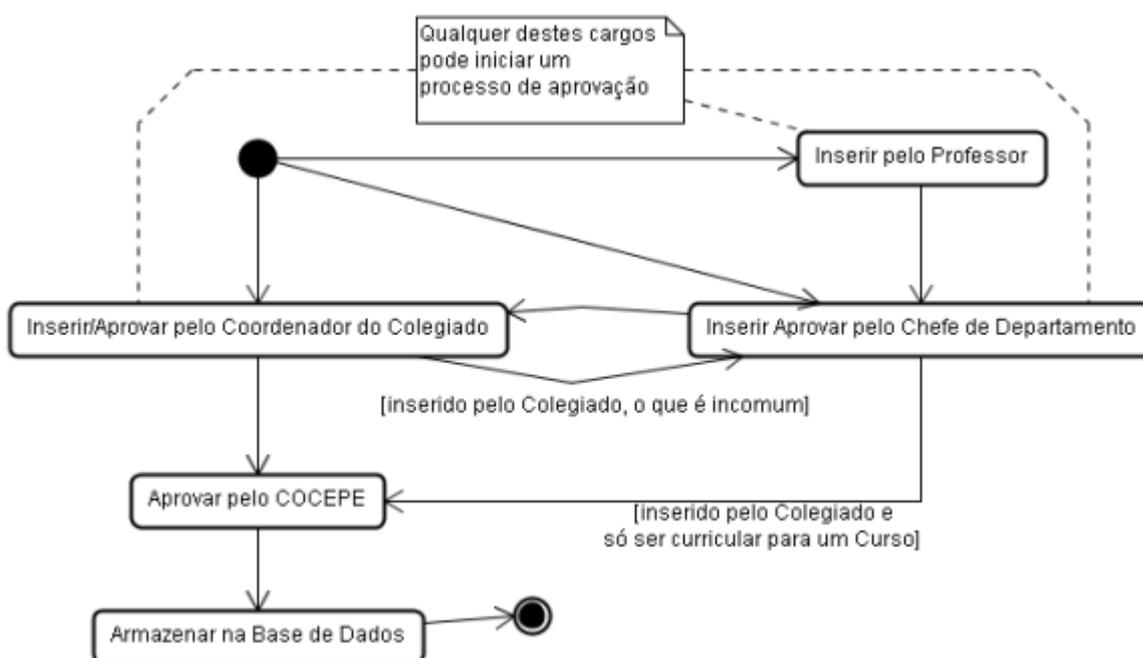


Figura 3 – Diagrama de atividades do processo de aprovação de um Programa de Disciplina

Entretanto, estes procedimentos são gerais e não padronizados. Departamentos, Colegiados, Institutos e Faculdades distintos têm maneiras diferentes de aprovar Ementas/Programas, ora incluindo, no processo, informações não pertinentes ao órgão a que se está encaminhando o documento, ora excluindo algumas informações importantes, uma vez que existem informações que são pertinentes à aprovação e outras pertinentes ao armazenamento dos Conteúdos Programáticos no DRA.

4.2 O Modelo Proposto para o Processo de Gerência

Neste trabalho, é proposta a criação de uma padronização, tanto na forma do documento, como na resolução das ações, para um melhor e mais rápido andamento das tramitações, mantendo o fluxo de atividades do processo.

Entre os elementos sujeitos à padronização, inclui-se:

a) Padronização da metodologia de aprovação dos Programas de Disciplinas;

b) Padronização do modelo de documento impresso, a ser entregue aos alunos;

c) Base de dados padronizada, constando todos os dados pertinentes à aprovação e ao armazenamento final da Disciplina, contendo, entre outros, os campos para:

- Departamento;
- Nome que a caracterize a Disciplina;
- Objetivo;
- Uma carga horária definida;
- Um conteúdo programático (os tópicos a serem abordados durante o período letivo);
- Uma bibliografia sugerida, incluindo o livro-texto;
- Cursos nos quais é ministrada;
- Código;

d) Fluxo de aprovação, com controle de visibilidade associado ao nível hierárquico do usuário, ocultando informações não pertinentes a cada nível hierárquico de aprovação e permitindo o acompanhamento do andamento dos processos.

No novo modelo, é desejável não alterar a seqüência de aprovação, tal como foi percebida na seção 4.1. Entretanto, propõe-se também incluir a possibilidade de que o DRA insira as Disciplinas diretamente na base de dados das Disciplinas aprovadas. Isto é necessário, pois existem muitos Programas de Disciplinas armazenados ainda em papel no referido Departamento, os quais não necessitam ser submetidos ao processo de aprovação. Além disso, também é possível que existam alguns Programas de Disciplinas em tramitação no período de implantação do sistema, que estarão tramitando ainda em papel.

Para a gerência deste fluxo de aprovação, pode ser aplicado um modelo de *Workflow* Administrativo, pois percebe-se que a Definição do Processo será pouco alterada ao longo do tempo, e que a comunicação entre os membros da equipe é fundamental assim como a velocidade em que esta comunicação ocorre.

Assim, tão logo uma Disciplina seja deferida, os próximos níveis hierárquicos da aprovação poderão perceber o deferimento e dar continuidade ao processo. Em caso de indeferimento, os níveis hierárquicos inferiores também o perceberão imediatamente, bem como a causa do indeferimento, podendo fazer as alterações necessárias para o rápido re-encaminhamento.

Na proposta, também, é permitido que professores escrevam e submetam as Disciplinas, como forma de agilizar o processo de submissão da mesma. Neste caso, a proposta é enviada ao Departamento, para aprovação e eventual correção.

5 Modelagem da Ferramenta Proposta

Neste capítulo define-se o projeto da ferramenta proposta no Capítulo 4, utilizando a UML, estendida de acordo com as sugestões de Conallen (1999) e seguindo o processo descrito no Capítulo 3.

Para tanto, este Capítulo é dividido de acordo com a numeração utilizada na Fig. 2, para uma maior facilidade de leitura. Em cada início de fase, bem como nas etapas de Ciclo de Desenvolvimento, são utilizadas as numerações de sub-capítulos. Nas demais etapas e tarefas, a numeração é utilizada no corpo do texto. Esta numeração é indicada entre parênteses, logo após o nome da tarefa mencionada ou no corpo do texto.

A implementação do modelo desenvolvido, realizada de acordo com o processo utilizado, é feita em PHP e HTML (*Hyper Text Mark-up Language* – Linguagem Marcada de Hipertexto) e descreve uma implementação de *Workflow* (alguns conceitos de *workflow* são descritos no Apêndice A), à medida que são definidos controle de fluxo baseados nos conceitos desta tecnologia para a tramitação do processo de cadastramento/atualização dos objetos de trabalho, bem como utiliza ferramentas que fazem com que o fluxo seja encaminhado de um nível a outro e que todo o fluxo possa ser acompanhado via registro de mudanças no sistema.

5.1 Planejar e Elaborar (1)

Ao iniciar a tarefa 1.1 (Definir rascunho do plano), percebe-se que os requisitos estão descritos, em sua maioria, no Capítulo 4 (os requisitos omitidos estão presentes na Fig.6). Também se definiu que o cronograma do trabalho teria a mesma duração deste trabalho. As informações sobre recursos limitaram-se a captar informações acerca da tecnologia utilizada no servidor em uso, concluindo-se que o mesmo utiliza PHP e banco de dados MySQL.

A segunda tarefa, 1.2 (Criar relatório de investigação preliminar), limita-se a definir que a tecnologia será implementada ao invés de adquirida.

Então, ao se Definir Requisitos (1.3), verificaram-se as tarefas anteriores, entrevistas e modelos existentes (tais como os itens a, b, c e d da seção 4.2), resultando na tab. 1 e na tab. 2.

Tabela 1 – Funções básicas

Ref #	Função	Categoria
R1.1	Incluir um Programa de Disciplina	Evidente
R1.2	Salvar um Programa de Disciplina na base de Dados	Oculto
R1.3	Aprovar um Programa de Disciplina para que ele possa seguir sua tramitação rumo à aprovação final	Evidente
R1.4	Imprimir um Programa de Disciplina	Evidente
R1.5	Reprovar um Programa de Disciplina, para que ele possa ser revisto pelo Colegiado e/ou Departamento responsáveis, conforme o caso de reprovação	Evidente
R1.6	Solicitar o andamento da tramitação	Evidente / Oculto

Tabela 2 – Funções acessórias

Ref #	Função	Categoria
R2.1	Consultar o registro da tramitação de um Programa de Disciplina	Evidente
R2.2	Alterar um Programa de Disciplina, para evitar digitar novamente um Programa rejeitado, a fim de tentar novamente sua aprovação junto às instâncias superiores	Evidente
R2.3	Permitir que professores incluam Programas de Disciplinas, diminuindo o trabalho dos Colegiados e dos Departamentos	Enfeite / Decoração

Após definirem-se os Requisitos do sistema, procura-se por todos os termos utilizados em todos os campos e tabelas, a fim de Registrar os Termos no Glossário (1.4). Esta tarefa foi feita à medida que prosseguia o projeto. O Apêndice B contém uma tabela com todos os termos utilizados nesta primeira fase.

O protótipo implementado neste ponto (1.5), utilizando a linguagem não-dinâmica HTML, foi desenvolvido meramente para apresentar a visualização das telas básicas do sistema. As telas criadas no protótipo podem ser verificadas nas fases seguintes. Então, foram Definidos os Casos de Uso (1.6) e os atores relacionados com eles, utilizando a notação UML (Fig. 4).

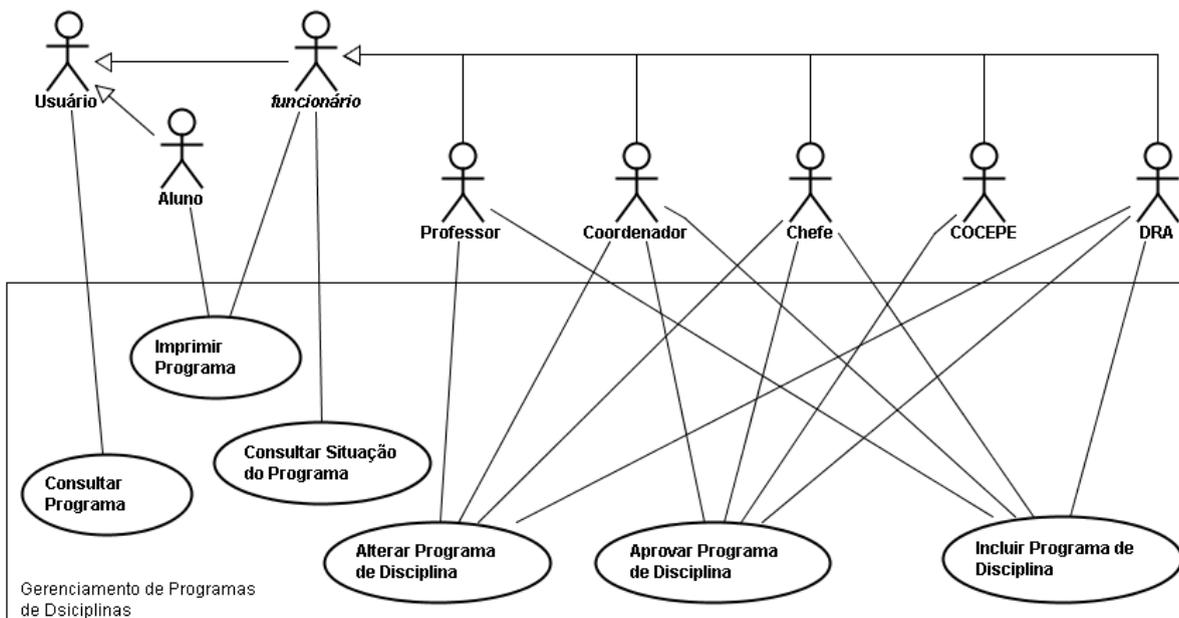


Figura 4 – Casos de Uso

Depois, estes Casos de Uso são exibidos textualmente, utilizando o modelo sugerido por Larman (2000), com a descrição de suas seqüências típicas e alternativas de eventos. Como exemplo, utiliza-se o Caso de Uso Aprovar Programa de Disciplina que, conforme se observa na Fig. 5, pode-se perceber quais os atores podem realizar este Caso de Uso, qual a finalidade do mesmo, bem como uma visão geral de sua funcionalidade. Também te-se o tipo deste Caso de Uso e o número de referência dos requisitos colhidos. Ao final, tem-se uma seqüência típica dos eventos deste Caso, seguida das seqüências alternativas, relacionando a numeração utilizada na seqüência típica e a alteração que causa a diferença de seqüência, bem como a própria alteração. Uma vez que a maioria dos Casos de Uso pode ser descrita mesclando-se a terminologia real com a essencial, para este trabalho optou-se por definir como essencial todo aquele que não seja totalmente real, ou seja, todo aquele que não exhibe referência à aplicação em termos de telas.

Os demais Casos de Uso, bem como o desenvolvimento deles, são descritos no Apêndice D.

Caso de Uso:	Aprovar Programa de Disciplina	
Atores:	Chefe, Coordenador, COCEPE ou DRA	
Finalidade:	Deferir ou indeferir um Programa de Disciplina, dando continuidade ao trabalho.	
Visão Geral:	O usuário acessa o sistema, escolhe a opção Aprovar Programa de Disciplina, clica na Disciplina que deseja trabalhar e a aprova.	
Tipo:	1. Primário 2. Essencial	
Referências:	<i>Funções:</i> R1.3, R1.5, R1.6	
Cruzadas:	<i>Casos de Uso:</i> Incluir Programa de Disciplina, Alterar Programa de Disciplina	
Seqüência Típica de Eventos		
	Ação do Ator	Resposta do Sistema
	1. O usuário seleciona a opção Aprovar Programa de Disciplina.	2. O sistema consulta a base de dados e exibe uma lista de todos os Programas de Disciplinas que requerem a atenção do usuário.
	3. O usuário seleciona um dos Programas, clicando sobre ele.	4. O sistema exibe uma página formatada <i>Web</i> com o Programa da Disciplina selecionada, contendo campos que podem ser selecionados para o caso do usuário selecionar a opção Reprovar.
	5. O usuário seleciona a opção Aprovar.	6. O sistema salva nas bases de dados as alterações necessárias.
Seqüências alternativas		
Linha 4: o Ator é um DRA. O sistema exibe uma página formatada com o campo do código da Disciplina passível de alteração.		
Linha 5: o Ator é um Coordenador, seleciona todos, alguns ou nenhum dos campos selecionáveis e seleciona a opção Reprovar. O sistema reduz o nível de aprovação a zero e envia um e-mail ao usuário que inseriu o Programa avisando que o mesmo foi reprovado e necessita alteração, avisando por quais motivos o mesmo foi reprovado pelo Coordenador.		
Linha 5: o Ator é um Chefe, seleciona todos, alguns ou nenhum dos campos selecionáveis e seleciona a opção Reprovar. O sistema reduz o nível de aprovação a zero e envia um e-mail ao usuário que inseriu o Programa avisando que o mesmo foi reprovado e necessita alteração, avisando por quais motivos o mesmo foi reprovado pelo Chefe.		
Linha 5: o Ator é um COCEPE, seleciona todos, alguns ou nenhum dos campos selecionáveis e seleciona a opção Reprovar. O sistema reduz o nível de aprovação a zero e envia um e-mail ao usuário que inseriu o Programa avisando que o mesmo foi reprovado e necessita alteração, avisando por quais motivos o mesmo foi reprovado pelo COCEPE.		
Linha 5: o Ator é um DRA. O usuário verifica o código da Disciplina e altera-o ou não, antes de selecionar a opção Aprovar.		
Linha 6: o Ator é um Coordenador. O Programa de Disciplina é salvo na base de com a Disciplina aprovada pelo Curso do Coordenador.e é enviado um e-mail ao COCEPE, caso o Programa já tenha sido aprovado pelo Chefe e pelos demais Coordenadores dos Cursos em que a Disciplina é ministrada.		
Linha 6: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe e é enviado um e-mail aos Coordenadores responsáveis que ainda não tenham aprovado a Disciplina, ou ao COCEPE, caso todos os Coordenadores já tenham aprovado a Disciplina.		
Linha 6: o Ator é um COCEPE. O Programa de Disciplina é salvo na base de dados com o nível de aprovação COCEPE e é enviado um e-mail ao DRA para que o código da Disciplina seja aprovado/alterado.		
Linha 6: o Ator é um DRA. O Programa de Disciplina é salvo na base de dados definitiva e é apagado da base de dados de aprovação.		

Figura 5 – Caso de Uso: Aprovar Programa de Disciplina

Baseado nas fontes obtidas pôde-se Definir um Rascunho do Modelo Conceitual (1.7) contendo alguns objetos, atributos e operações, que se acreditou que seriam melhor compreendidos agrupados conforme se observa na Fig. 6 e, a

partir deste, um Rascunho para a Arquitetura do Sistema (1.8), conforme pode se observar na Fig. 7.

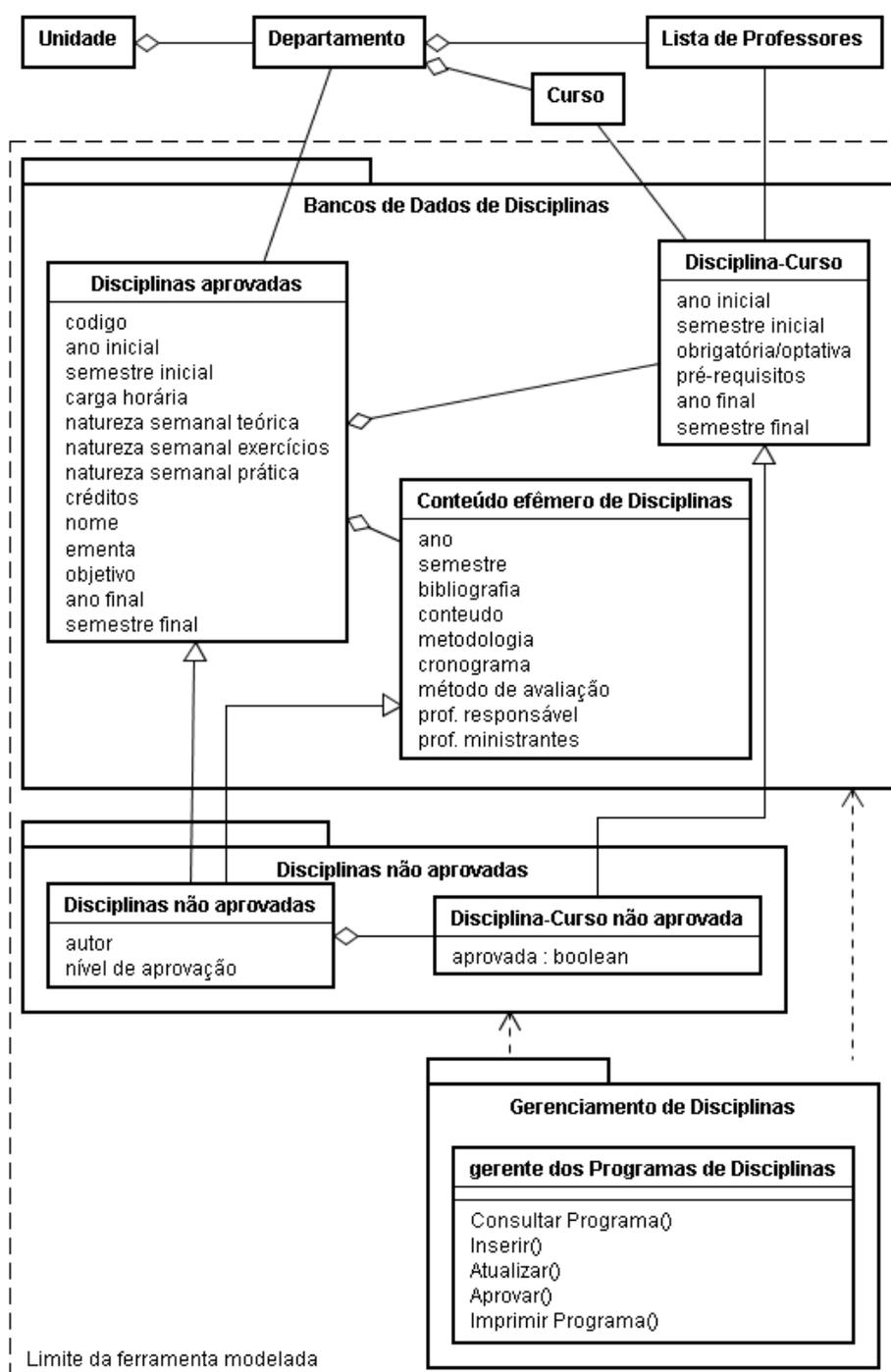


Figura 6 – Modelo Conceitual

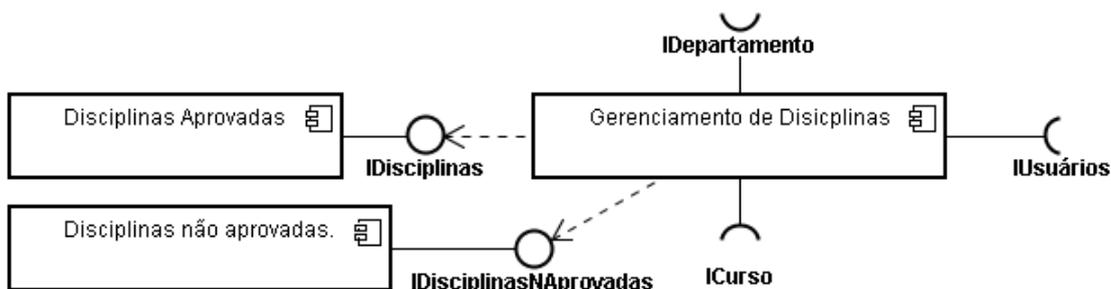


Figura 7 – Rascunho da Arquitetura do sistema

5.2 Construir (2)

Para a construção da ferramenta escolhe-se utilizar 3 ciclos de desenvolvimento, seguindo a estratégia sugerida por Larman (2000) de abordar primeiro as funções mais relevantes ao sistema e, a cada Ciclo, abordar funções menos relevantes ou que, de outra forma, poderiam ser postergadas.

5.2.1 Ciclo de Desenvolvimento 1 (2.1)

Ao se iniciar o primeiro Ciclo de Desenvolvimento, verificaram-se os requisitos e modelos descritos como um Refinamento do Plano (2.1.1) e não houveram sugestões de alteração. Então, ao se perceber que não havia nada implementado e os artefatos não precisaram ser sincronizados (2.1.2).

A seguir, Analisando este 1º Ciclo (2.1.3), percebe-se estar abordando os seguintes casos de uso:

- ? Consultar Programa;
- ? Incluir Programa de Disciplina;
- ? Aprovar Programa de Disciplina.

Decidiu-se, neste momento, não abordar o Caso de Uso Imprimir Programa neste ciclo, cujo desenvolvimento não interferiria em nenhum outro processo, e que foi verificado que demandaria um tempo de execução demasiado longo, uma vez que a linguagem de programação ainda não era dominada e que poderia sê-lo durante este primeiro ciclo.

Neste momento, ao se verificar a existência de novos Casos de Uso a serem Definidos (2.1.3.1), não se percebeu nenhum Caso de Uso novo. Da mesma forma, os Diagramas de Caso de Uso não necessitaram Refinamento (2.1.3.2) neste ponto, da mesma forma que o Modelo Conceitual (2.1.3.3) também não sofreu alteração.

O Glossário, assim como os Casos de Uso, não necessitou Refinamento (2.1.3.4).

A seguir, definem-se os Diagramas de Seqüência (2.1.3.5) do Caso de Uso aprovar programa de disciplina, exibindo as interações entre o ator que pode interagir neste caso de uso, bem como a diferença de interação dependendo de qual ator específico atua, conforme exibido nas notas (Fig. 8). Os demais casos de uso podem ser vistos no Apêndice E.

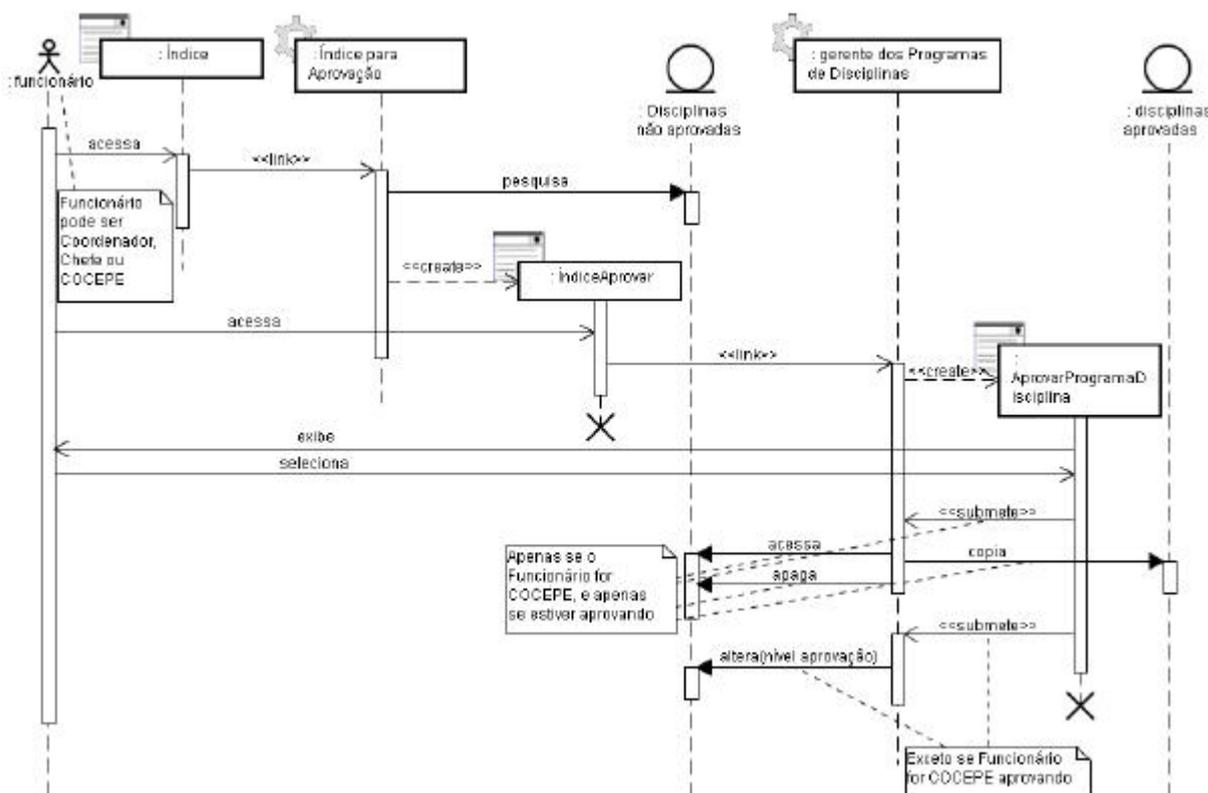


Figura 8 – Diagrama de Seqüência: Aprovar Programa de Disciplina

Definidos os Diagramas de Seqüência, torna-se possível Definir os Contratos da Operação (2.1.3.6) deste caso de uso (Fig. 9). Neste contrato, podemos ver o nome desta operação, bem como quais os parâmetros são necessários a ela, o que se pretende com ela, quais os requisitos e Casos de Uso ela atende, o que acontece ao final desta operação, o que é necessário que o ator tenha para iniciar esta operação, bem como uma lista de alterações no sistema ocorrem, exibindo também as associações que são criadas com esta operação.

Contrato	
Nome:	Aprovar Programa de Disciplina (usuario; protocolo)
Responsabilidades:	Promover o andamento da aprovação, seja aprovando o Programa da Disciplina na instância, para que a mesma dê mais um passo rumo à aprovação, seja reprovando a Disciplina, fazendo-a retornar a um nível de Aprovação mais baixo para modificação.
Tipo:	Sistema
Referências Cruzadas:	Funções do Sistema: R1.3, R1.5, R1.6 Casos de Uso: Incluir Programa de Disciplina, Alterar Programa de Disciplina
Notas:	
Exceções:	
Saída:	Se o nível hierárquico do usuário não for o de aprovação final, envia uma mensagem para o(s) usuário(s) responsável(is) pelo próximo nível de aprovação ou pelo anterior.
Pré-condições:	Há um Programa de Disciplina esperando a aprovação deste usuário.
Pós-condições:	Se o usuário não possuir nível final de aprovação e pressionar o botão de aprovar, envia uma mensagem para o(s) usuário(s) responsável(is) pelo próximo nível de aprovação (<i>formada uma associação</i>); Se o usuário pressionar o botão de reprovar, envia uma mensagem padronizada para o usuário responsável pela inserção da Disciplina, descrevendo todos os campos que levaram à reprovação; Se o usuário possuir nível final de aprovação e pressionar o botão de aprovar, o Programa de Disciplina é transferido da Base de Dados das Disciplinas não Aprovadas para a dos Programas de Disciplinas; Qualquer que seja o caso, é criada uma linha relatando a ação do usuário em Log das Disciplinas (<i>criada uma instância</i>)

Figura 9 – Contrato da operação: Aprovar Programa de Disciplina

A seguir, analisa-se as alterações que a operação Aprovar Programa de Disciplina realiza no sistema e, então, torna-se necessária uma Definição do Diagrama de Estado (2.1.3.7), mostrado na Fig. 10. Este diagrama, semelhante ao exibido na Fig. 3, é bastante expandido e ilustra como o *workflow* embutido funciona no sistema, através dos efeitos de entrada e da própria manipulação do estado da operação.

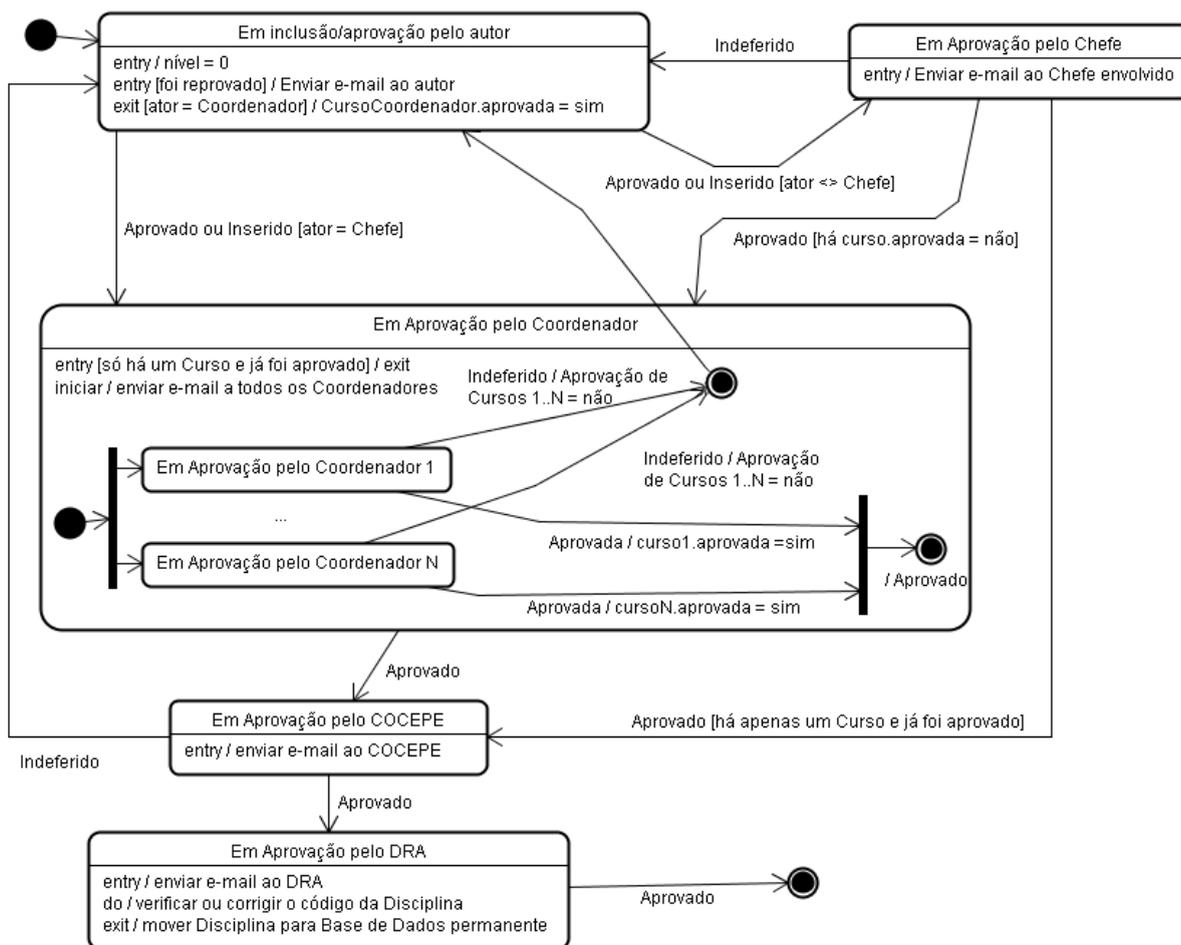


Figura 10 – Diagrama de Estado: Aprovar Programa de Disciplina

Ao final deste passo de análise, passa-se então a Projetar (2.1.4) os Casos analisados. As primeiras tarefas deste passo são Definir Casos de Uso Reais (2.1.4.1) e Definir Interfaces de Usuários e “*Storyboards*” (2.1.4.2) os quais, para evitar repetir os Casos de Uso da tarefa 1.6 na íntegra, optou-se por exibir apenas seus nomes e as alterações efetuadas. Então, a Fig. 11 exibe o caso de uso aprovar programa de disciplina, agora de um modo mais real, como pode se notar pela alteração da seqüência típica de eventos, que agora relata quais campos devem ser preenchidos, quais os botões devem ser pressionados, e a quais as telas estes botões e outros campos se referem. Para uma melhor leitura do texto, as telas de interface são exibidas todas no Apêndice E.

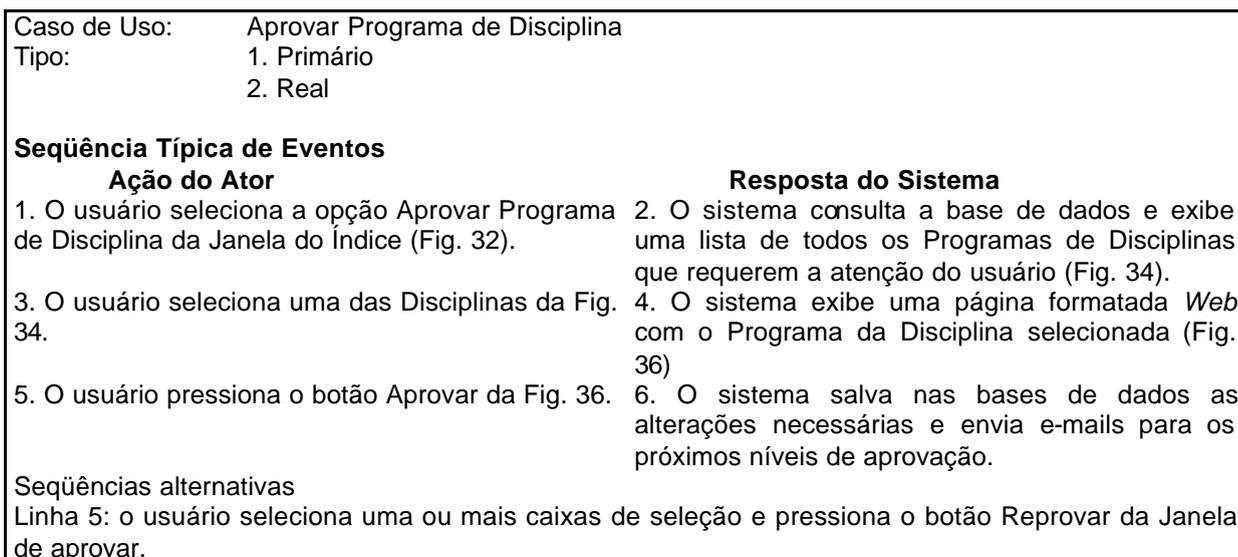


Figura 11 – Caso de Uso Real: Aprovar Programa de Disciplina

A seguir, Define-se Diagramas de Interação (2.1.4.3), conforme se observa na Fig. 12, com detalhes sobre as diversas operações e seus parâmetros.

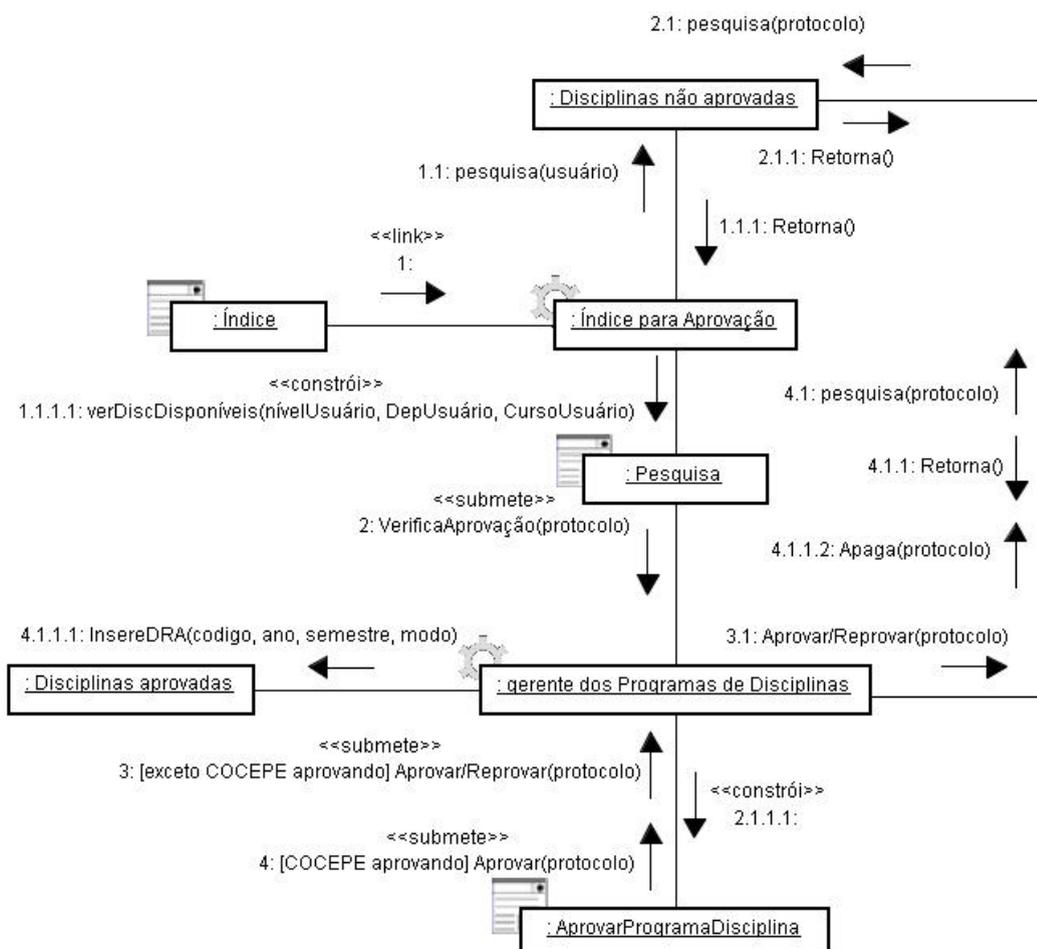


Figura 12 – Diagrama de Comunicação: Aprovar Programa de Disciplina

Ao mesmo tempo em que se definem os diagramas de interação, também se definem os Diagramas de Classes de Projeto (2.1.4.4), com as classes percebidas, suas operações e atributos, conforme a Fig. 13.

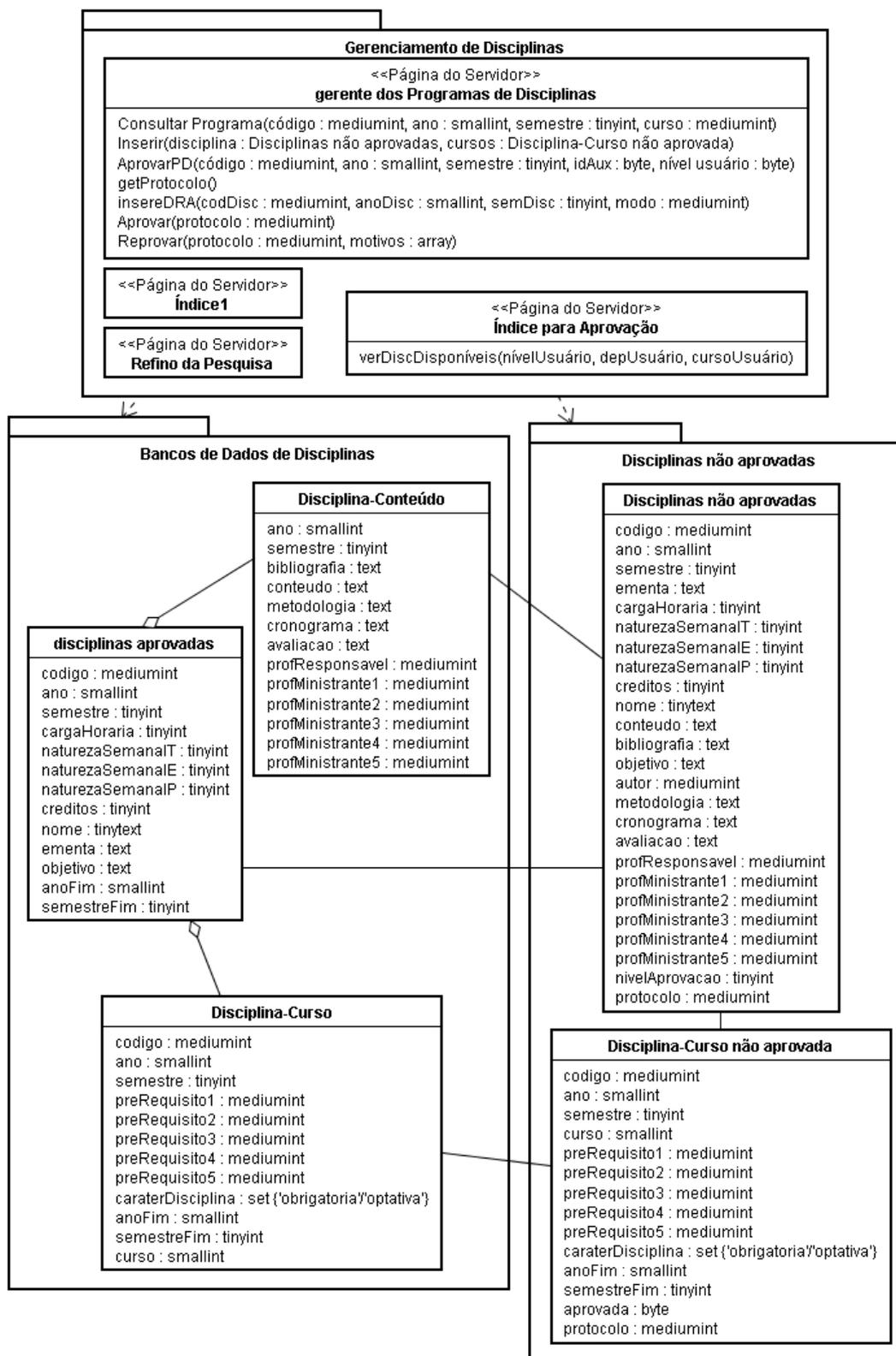


Figura 13 – Diagrama de Classes de Projeto no 1º Ciclo de Desenvolvimento

Assim que os casos de uso reais são feitos, Refina-se a Arquitetura do Sistema (2.1.4.5), conforme se observa na Fig. 14, onde os componentes do sistema são refinados e pode-se perceber seus componentes internos, e na Fig. 15, onde é exibido um diagrama dos artefatos *Web* (componentes), mostrando quais classes estes geram, bem como se pode perceber algumas funcionalidades destas classes, tais como formulários e outras navegações possíveis dentro do sistema.

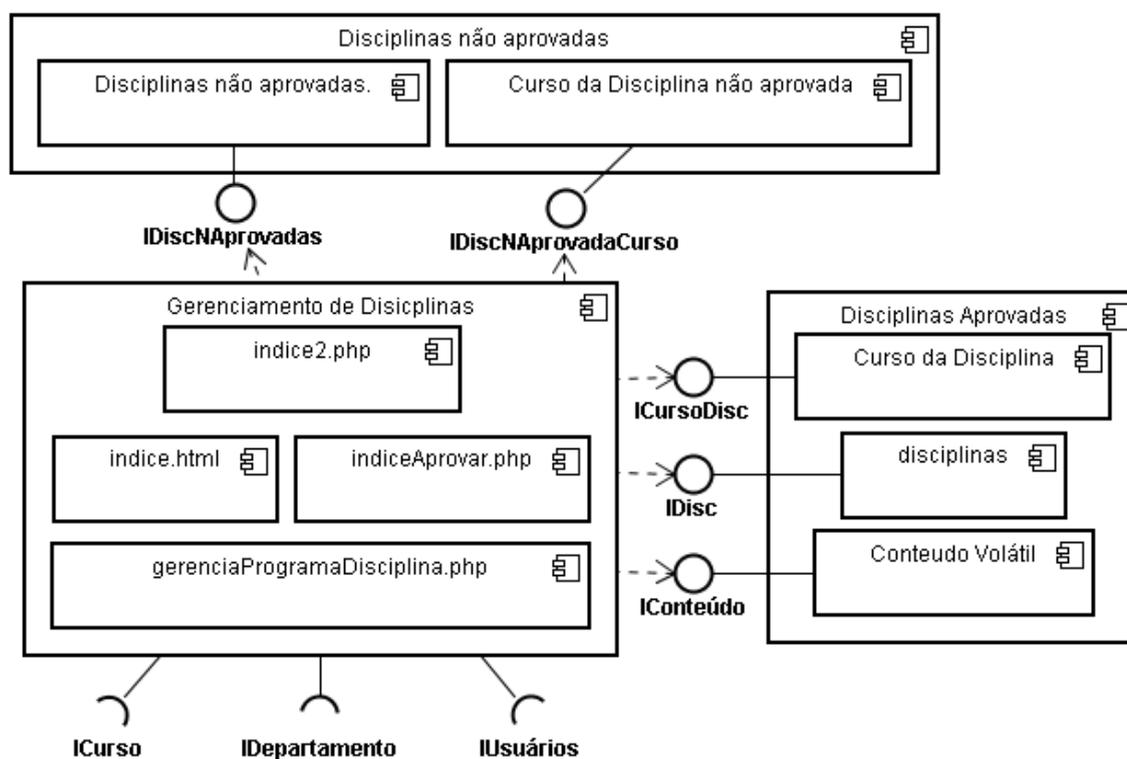


Figura 14 – Arquitetura refinada para o 1º ciclo de desenvolvimento

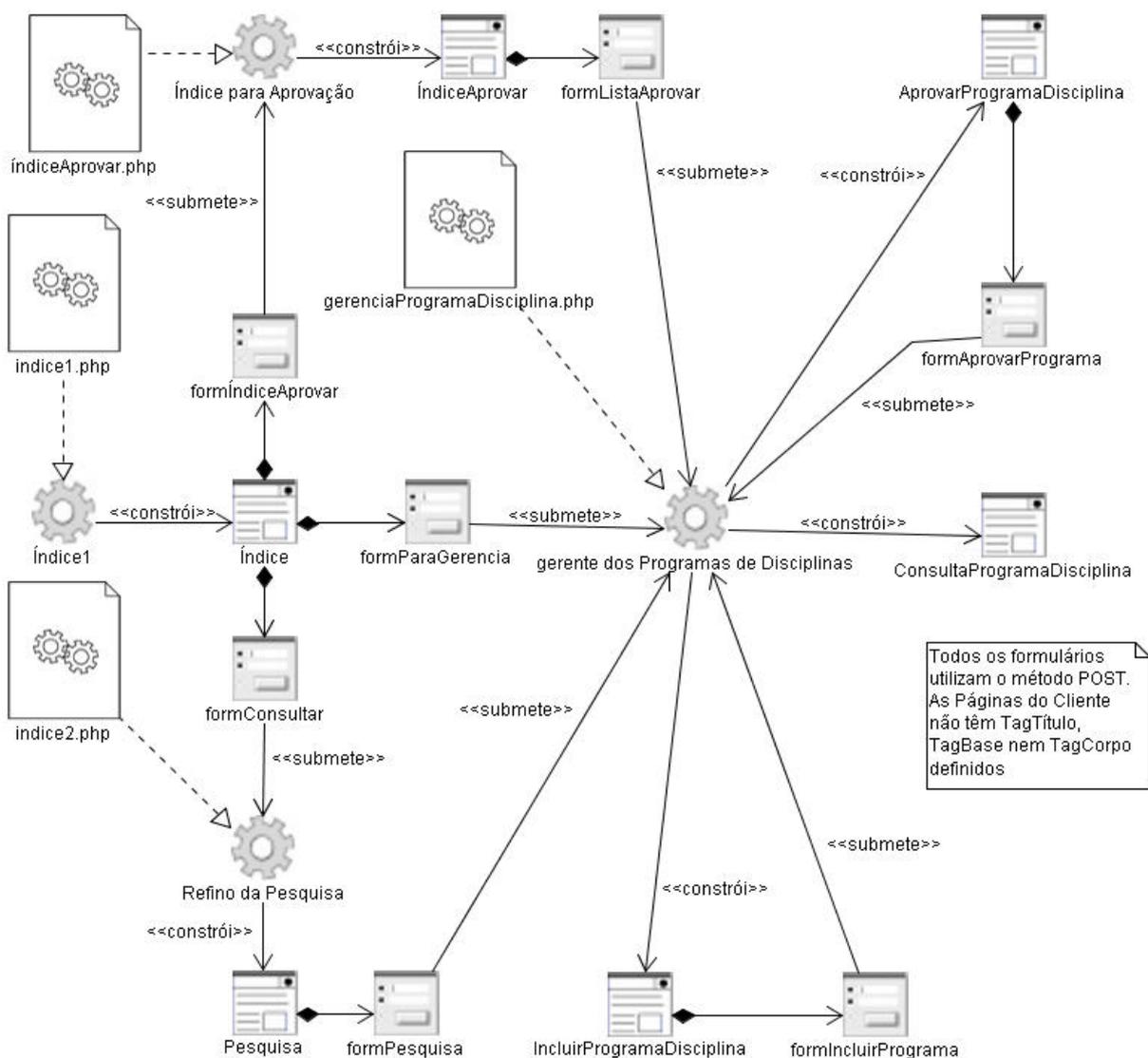


Figura 15 – Diagrama de Artefatos Web e Navegação

No final deste primeiro ciclo de desenvolvimento, Programou-se (2.1.5) e Testou-se (2.1.6) um modelo, salvo como Ciclo de Desenvolvimento 1.zip, contendo os arquivos gerenciaProgramaDisciplina.php, index1.php, indice2.php e indiceAprovar.php.

5.2.2 Ciclo de Desenvolvimento 2 (2.2)

Neste segundo Ciclo de Desenvolvimento, escolheu-se desenvolver os Casos de Uso Primários que não foram abordados no 1º Ciclo, quer por sua dificuldade de implementação ou por acreditar que eles seriam melhor compreendidos e implementados após o 1º Ciclo.

Não houve necessidade de se Refinar o Plano (2.2.1), uma vez que o cliente estava satisfeito com os artefatos desenvolvidos até o momento.

Então, houve apenas um pequeno acréscimo na classe gerente dos Programas de Disciplina que, a fim de Sincronia (2.2.2), tal acréscimo é exibido na Fig. 16, exibindo a nova operação desta classe.

gerente dos Programas de Disciplinas
- selectRow(tabela : int, codDisc : int, anoDisc : int, semDisc : int, idAux : int) : {Return= varia} {versão 1.2}

Figura 16 – Sincronia de artefatos para o 2º Ciclo

Então, partiu-se para a tarefa de Analisar (2.2.3), onde se decidiu abordar os casos de Uso:

- Imprimir Programa;
- Consultar Situação do Programa.

(O restante deste ciclo de desenvolvimento é exibido no Apêndice F)

5.2.3 Ciclo de Desenvolvimento 3 (2.3)

Neste Ciclo, todos os Casos de Uso restantes são examinados.

Ao se Refinar o Plano (2.3.1), o cliente definiu novos requisitos, exibidos na tab. 3 e na tab. 4, referentes à tramitação burocrática do sistema de aprovação, que necessitam de vias assinadas.

Tabela 3 – Funções Básicas - adendo no 3º Ciclo de Desenvolvimento

Ref #	Função	Categoria
R1.7	Emitir memorandos quando um Programa de Disciplina é inserido, alterado, aprovado ou rejeitado.	Evidente
R1.8	Permitir que funcionários possam atualizar as Disciplinas utilizando as antigas versões como base	Evidente

Tabela 4 - Funções acessórias - adendo no 3o Ciclo de Desenvolvimento

Ref #	Função	Categoria
R2.4	Permitir que o COCEPE altere os modelos de memorandos enviados.	Evidente

Como não se percebeu necessidade de Sincronizar os Artefatos (2.3.2) nesta etapa, passou-se à Análise (2.3.3), onde se percebeu o seguinte caso de uso:

- Alterar Programa de Disciplina.

Além deste, e baseado nos novos requisitos apresentados neste ciclo, criaram-se o caso de uso de Atualizar Programa de Disciplina e Enviar Memorando, exibidos na Fig. 17 (2.3.3.1). A descrição textual deste caso de uso e as alterações

que se perceberam necessárias nos casos de uso Incluir Programa de Disciplina e Aprovar Programa de Disciplina em decorrência destes novos requisitos, e Alterar Programa de Disciplina, em decorrência destes novos requisitos e por outras considerações percebidas no decorrer dos testes dos Ciclos anteriores são encontradas no Apêndice G, juntamente com o restante deste ciclo de desenvolvimento.

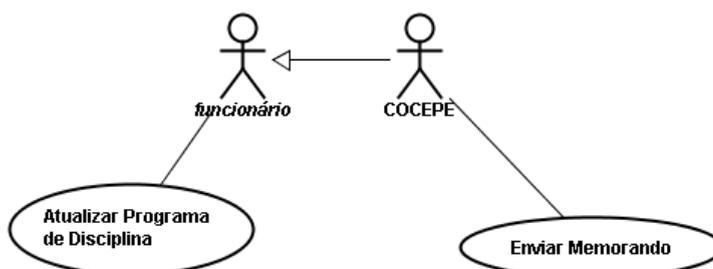


Figura 17 – Novos Casos de Uso Atualizar Programa de Disciplina e Enviar Memorando

5.3 Implantar (3)

O modelo construído foi salvo como final.zip, na sua versão final e contendo todas os arquivos contidos no arquivo Ciclo de Desenvolvimento 3.zip, pronto para ser instalado no servidor (Fig. 18), ajustar a referência às bases de dados existentes (Fig. 19) e que foram criadas para fins de teste, ajustar a aparência das páginas utilizando padrões de cores, fontes, tabelas, figuras e outros itens, e então começar o treinamento dos usuários.

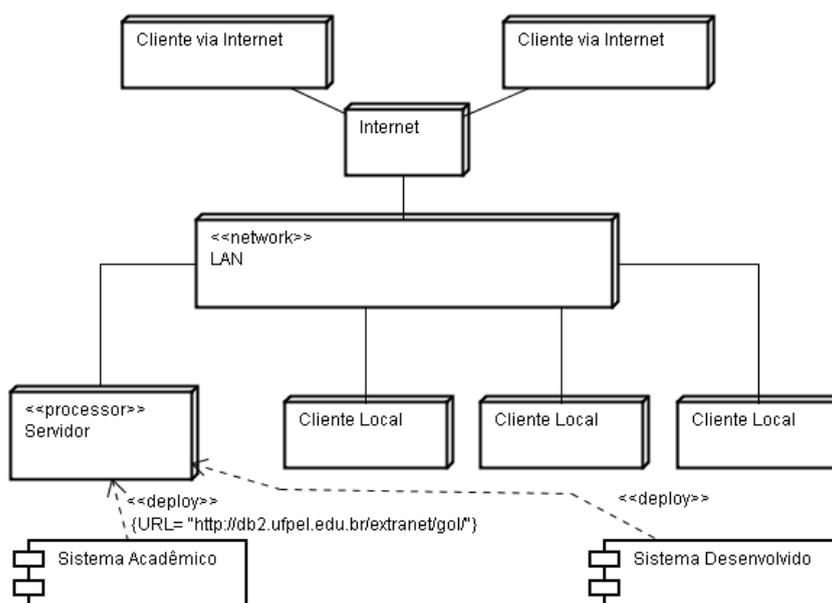


Figura 18 – Diagrama de Implantação Física

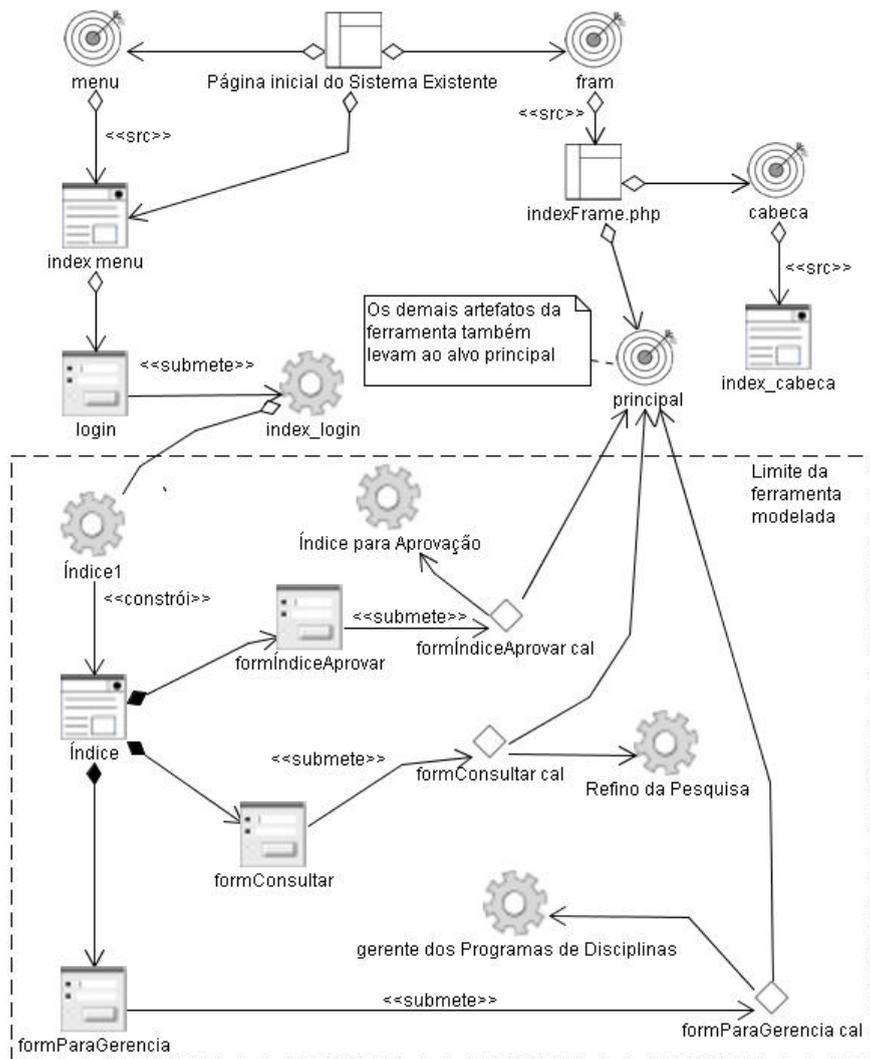


Figura 19 – Diagrama de Implantação Lógica

6 Conclusões

Concluídos todos os processos de pesquisa, especificação e modelagem, tornou-se possível destacar diversas considerações importantes acerca de todos estes tópicos.

Conforme se utilizou os processos e métodos recomendados de Larman (2000), verificou-se a riqueza de detalhes captada nas fases anteriores, antes mesmo da fase de Construir, cuja penúltima tarefa, dentre tantas realizadas ao longo de seis passos, é efetivamente escrever o código fonte. Isto possibilitou uma melhor compreensão não apenas do projeto a ser abordado, mas também da forma como o mesmo deveria ser abordado. Também, dado o constante refinamento conceitual, em até três refinamentos sucessivos, é possível melhor estruturar e compreender o sistema e as necessidades da construção.

Assim, com um processo de desenvolvimento claramente detalhado e aplicado tão rigorosamente quanto necessário, é possível criar um sistema de aplicação bastante documentado e detalhado, não apenas em sua versão final, mas em cada etapa do planejamento. Em processos de desenvolvimento de sistemas mais complexos, provavelmente as vantagens do uso deste processo possam ser mais claramente percebidas, o que poderia ser testado em um trabalho futuro.

Entretanto, acredita-se que o passo Refinar o Plano (2.1.1) poderia ser melhor utilizado ficando por último em cada fase, após o Teste (2.1.6). Deste modo, teria-se a apreciação do cliente após a programação e teste, logo antes de verificar se o mesmo está pronto para a Implantação, e se manteria a posição de consultar o cliente logo antes de um novo Ciclo de Desenvolvimento.

Com relação ao caso estudado, percebe-se, de imediato, a grande diminuição da circulação de papéis, uma vez que apenas memorandos são passados de setor a setor, anunciando formalmente o encaminhamento de documentos digitais, uma vez que a dependência de assinaturas ainda é importante

no processo burocrático institucional e a mudança, mesmo para um processo totalmente digital, deve ser gradual.

Outra vantagem é que o acompanhamento da aprovação das disciplinas via *Web*, através da ferramenta, permite que haja uma maior facilidade de cobrança por parte dos agentes interessados, contribuindo significativamente para o aumento da eficiência do *Workflow*.

Como trabalho futuro, sugere-se um estudo sobre a implantação de um WfMS capaz de gerenciar este processo, utilizando uma linguagem voltada para a *Web*. Também seria interessante viabilizar a distribuição desta ferramenta a outras instituições de ensino, uma vez que um WfMS poderia suprir às necessidades de cada instituição.

Sugere-se, também, o estudo sobre o uso das ferramentas comerciais de WfMS existentes, de modo a estudar um meio de utilizá-las no sistema de aprovação de Programas de Disciplinas.

Referências

- ALLEN, Rob. Workflow: An Introduction. In: **The workflow handbook**. USA: Open Image System Inc., 2000. p.15-38.
- BETEMPS, Carlos Michel. **Análise e estudo de sistemas de gestão de workflow na internet**. 2000. 58f. Trabalho Individual - Faculdade de Computação Programa de Pós Graduação em Computação / Universidade Federal do Rio Grande do Sul, Porto Alegre.
- BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivar. **UML, Guia do Usuário**. 2 ed. Rio de Janeiro: Elsevier, 2005. 474p.
- CONALLEN, Jim. **Modeling web application architectures with UML**. USA: Rational Software White Paper, 1999.
- CONALLEN, Jim. **Building web application with UML second edition**. USA: Addison Wesley, 2002. 496 p.
- FOWLER, Martin; SCOTT, Kendall. **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. 2.ed. Porto Alegre: Bookman, 2000.
- GEORGAKOPOULOS, Diimitrios, HORNICK, Mark e SHETH, Amit. **An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure**. Boston: Kluwer Academic Publishers, 1995. 34f. Disponível em: <<http://citeseer.ist.psu.edu/>> Acesso em: fev. 2004.
- KRUCHTEN, Phillipe. Foreword. In: **Applying UML and Patterns: an introduction to object oriented analysis and design and iterative development**, 3rd edition. US: Prentice Hall, 2005. 736p.
- LARMAN, Craig. **Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos**. Porto Alegre: Bookman, 2000.
- MARKS, Dan. **Development Methodologies Compared**. Disponível em: <http://www.ncycles.com/N_Cycles_Methodologies.pdf> Acesso em: Jun. de 2007.
- PRIOR, Carol. Workflow and process management In: **THE WORKFLOW Handbook** 2003. Future Strategies Inc., Lighthouse Point, FL, USA, 2003. p. 17-25.
- RUSINCKIEWICZ, Marek e SHETH, Amit. Specification and Execution of Transactional Workflows. In: **Modern Database Systems: The Object Model, Interoperability, and Beyond**. USA: Addison-Wesley, 1995. p. 592-620.
- SCOTT, Kendall. **O processo unificado explicado**. Porto Alegre: Bookman, 2003. 160p.

SOMMERVILLE, Ian. **Software Engineering**, 6th ed. Harlow, Inglaterra: Addison-Wesley, 2001. 693 pg.

UNIVERSIDADE FEDERAL DE PELOTAS. Administração. Conselho Universitário. Regimento Geral da Universidade Disponível em <<http://www.ufpel.tche.br/consun/regimento.php>>. Acesso em fev. 2007.

WfMC - The Workflow Management Coalition. Disponível em <<http://www.wfmc.org/index.html>>. Acesso em out. 2004.

Apêndice A – Workflow

Desde o início da industrialização, a humanidade começou a utilizar o conceito de *Workflow*, que nada mais é que a organização do fluxo do trabalho. Esta organização pode ser física (manufatura) ou, mais modernamente, por sistemas de informação (por exemplo, sistemas computadorizados) (GEORGAKOPOULOS; HORNICK; SHETH, 1995; ALLEN, 2000; PRIOR, 2003).

Por se concentrar na rotina e nos conceitos de regras, atributos e procedimentos da tarefa, e aplicando-se a tecnologia da informática, é possível criar sistemas que implementem as regras definidas por pessoas para a automatização e melhor controle de certas tarefas. Estas regras são chamadas de **processo de trabalho**; são as definições em alto nível do processo a ser realizado (GEORGAKOPOULOS; HORNICK; SHETH, 1995).

Estes processos de negócio são, então, implementados no sistema para serem realizados. Estes processos implementados são chamados de **processo de informação** e são baseadas em bancos de dados, transações e outras tarefas parcialmente automatizadas, enquanto aquela que é executada por seres humanos, sem interferência do computador é chamada de **processo material** (como exemplo neste caso, pode-se citar a entrega num serviço de tele-entrega).

Desta forma, é possível que se faça uma reengenharia dos processos, adaptando-os, melhorando-os e até mesmo criando novos, visando melhorar a qualidade dos produtos, a eficiência dos processos, a satisfação do cliente, entre outros. Para que esta reengenharia seja feita, são necessárias a **reengenharia dos processos de informação** e a **reengenharia dos processos de negócio**, sendo que uma é complementar à outra e servem de *feedback* mútuo.

Workflow está bastante relacionado a esta reengenharia, uma vez que trata, conceitualmente, dos processos de negócio e descreve os sistemas envolvidos nos processos de informação, provendo descrições dos processos relevantes e o rápido

acesso ao (re)design e à (re)implementação dos sistemas de informação e de negócio, à medida que estes vão sendo implementados e alterados (GEORGAKOPOULOS; HORNICK; SHETH, 1995; ALLEN, 2000; PRIOR, 2003).

A seguir, são apresentadas algumas terminologias utilizadas em *Workflow*, bem como a caracterização dos tipos de Sistemas de *Workflow* existentes no mercado atual.

A.1 Definições de *Workflow* Terminologia

Apesar dos esforços de WfMC – *The Workflow Management Coalition* (2004) há pouco consenso no que se refere ao uso da palavra “*Workflow*”: as pessoas a utilizam para definir um processo de negócio, especificação de um processo, *software* que implementa e automatiza um processo ou simplesmente o *software* que serve de suporte para a coordenação e a colaboração de pessoas que implementam um processo. Há, ainda, outras definições que separam *Workflow* das aplicações. Por exemplo, Rusinkiewicz (RUSINCKIEWICZ; SHETH, 1995) define *Workflow* como:

[...] as atividades envolvendo a execução coordenada de múltiplas tarefas desenvolvidas por entidades de processamento diferentes. Uma tarefa define algum trabalho a ser feito e pode ser especificada de várias maneiras, incluindo uma descrição textual num arquivo ou uma mensagem de correio eletrônico, um formulário, ou um programa de computador. Uma entidade de processamento que realiza uma tarefa pode ser uma pessoa ou um sistema de software (por exemplo, um correio eletrônico, um programa aplicativo, um sistema de gerenciamento de banco de dados). A especificação de um *workflow* envolve a descrição destes aspectos e das suas tarefas constituintes (e das entidades de processamento que as executam) que são relevantes ao controle e à coordenação de sua execução. Também requer a especificação dos relacionamentos (dependências) entre as tarefas e seus requerimentos de execução. Isto pode ser especificado utilizando uma variedade de paradigmas de software (por exemplo, regras, restrições ou programas).

A definição atual de *Workflow*, segundo a WfMC (2004), e que data de 1996, é a seguinte:

A automação de um processo de negócio, no todo ou em parte, durante o qual documentos, informação ou tarefas são passadas de um participante a outro para agir, de acordo com um número de regras procedurais.

Quanto ao Sistema Gerenciador de *Workflow* (WfMS – *Workflow Management System*), Allen (2000) simplifica dizendo que é o *software* que gerencia o *Workflow*.

Outras definições importantes do *Workflows* são as Definições de Processos e os Processos de Trabalho, que são as regras procedurais que descrevem o processo num nível médio e alto, respectivamente.

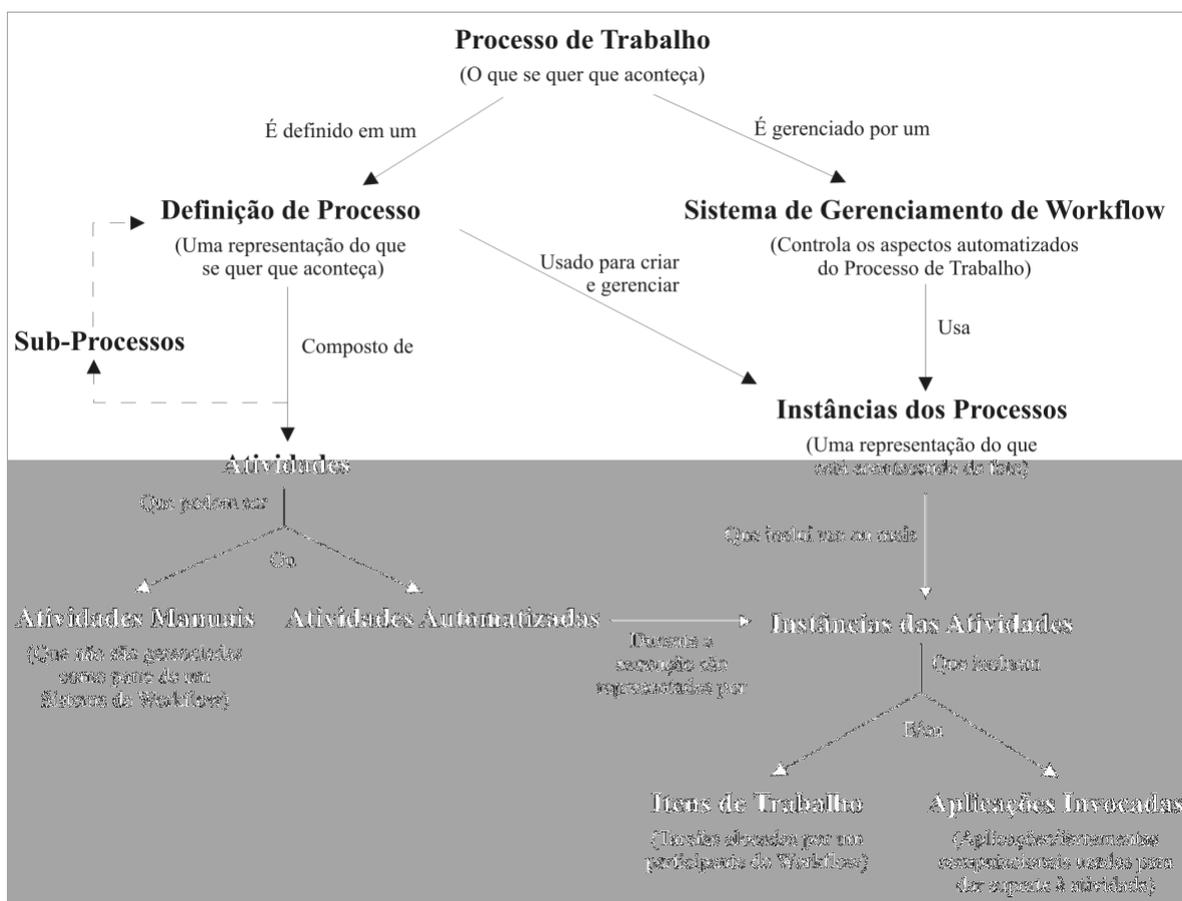


Figura 20 – Processo de Trabalho

Fonte: ALLEN, 2000.

A Fig. 20 apresenta um paralelo entre os **processos materiais** (ao lado esquerdo da figura) e os **processos de informação** (ao lado direito). Observa-se que Definição de Processo e Atividades estão do lado esquerdo, pois são definidas por seres humanos.

A.2 Caracterização (Tipos) de *Workflows*

Pode-se caracterizar *Workflow* de algumas maneiras, dentre as quais as citadas por Georgakopoulos, Hornick e Sheth (1995), mas também discutidas no documento da WfMC (ALLEN, 2000).

A caracterização de *Workflows* exibida a seguir leva em consideração a utilização do *Workflow* em razão da liberdade de se criar e alterar as Definições de Processos e da quantidade necessária de processamento em relação à quantidade de atividade humana envolvida no processo, o que afeta diretamente a produtividade (a produtividade sendo proporcional à porcentagem de processamento da tarefa).

A.2.1 *Workflow Ad-Hoc*

Sistemas de *Workflow Ad-Hoc* destinam-se, tipicamente, a pequenos grupos para tarefas igualmente pequenas e rápidas, onde há muito mais esforço humano do que de automação, tais como coordenação, colaboração e co-decisão. Portanto, estes sistemas permitem que os usuários criem e alterem as Definições de Processos rápida e facilmente, para que estes se adaptem às novas circunstâncias assim que estas se apresentem, tornando-as tão especializadas para o caso específico que praticamente cada Definição de Processo servirá para uma única tarefa (GEORGAKOPOULOS; HORNICK; SHETH, 1995; ALLEN, 2000).

Estes sistemas maximizam a flexibilidade em áreas onde a segurança e a produtividade não são muito importantes. É possível, também, que hajam falhas no *software*, sem que isso comprometa a produtividade pois, em geral, são acessados apenas para verificar se alguma determinada tarefa já foi concluída pelo grupo.

A.2.2 *Workflow Administrativo*

A característica mais importante no *Workflow Administrativo* é a facilidade de definir processos, pois os mesmos são repetitivos e constantes, praticamente mecânicos (GEORGAKOPOULOS; HORNICK; SHETH, 1995; ALLEN, 2000).

As Definições de Processos são, normalmente, feitas através de formulários e, se uma tarefa for muito complexa para ser descrita, é necessário utilizar outro programa.

Este tipo de sistema é empregado quando é mais importante a flexibilidade do que a produtividade, apesar de que, em Sistemas de *Workflow* Administrativos, o envio de mensagens entre os participantes é feito ativamente, em geral, através do uso de e-mails, enquanto que nos sistemas ad-hoc esta comunicação é feita passivamente.

A.2.3 Workflow Colaborativo

Workflows Colaborativos foram feitos para grupos de trabalho, não importando se for um grupo pequeno, de uma meia dúzia de colegas de uma mesma sala, ou um grupo imenso e disperso, como uma grande comunidade de pesquisadores ao redor do mundo. Ele se aplica quando a comunicação é vital para o empreendimento, o que é cada vez mais constante no mundo de hoje. Nestes sistemas, a produtividade não é muito importante, sendo as Definições de Processos muito menos rígidas e com a capacidade de serem alteradas freqüentemente (ALLEN, 2000).

Workflows Colaborativos são, muitas vezes, chamados de *Groupware*. Entretanto, nem todos os *Groupwares* são *Workflows* (videoconferências, *bulletin boards*, por exemplo, não o são).

A.2.4 Workflow de Produção

Workflows de Produção são aqueles que primam pela automação do serviço, tentando minimizar a interferência humana, sendo esta feita apenas nos casos excepcionais (aqueles que estão fora dos padrões de tolerância pré-determinados) (GEORGAKOPOULOS; HORNICK; SHETH, 1995; ALLEN, 2000).

Devido à grande quantidade de Sistemas de *Workflows* de Produção existentes no mercado, é possível caracterizá-los ainda mais, através da funcionalidade de seus motores, ou seja, do software que, ativamente, gerencia o sistema.

A.2.1.1 Motores de Workflows Autônomos

Um motor de *Workflow* autônomo é um WfMS que não requer o apoio de nenhum outro *software*, à exceção de sistemas gerenciadores de bancos de dados e

um agente de intermediação para a troca de mensagens (ALLEN, 2000). Eles chamam outros utilitários em tempo de execução e as informações relevantes ao *workflows* são passadas entre os usuários.

WfMS Autônomos são utilizados para concatenar diversas aplicações que, apesar de necessárias (ou até vitais) ao funcionamento de um Processo, sem o gerenciamento de um WfMS Autônomo não passariam de aplicações isoladas.

Por trabalhar com diversas aplicações diferentes, cada qual com suas próprias estruturas de dados e especificações de interface, pode ser uma tarefa bastante complexa programar um WfMS Autônomo.

A.2.1.2 Workflow Embarcado

WfMS Embarcado é aquele que não funciona sem estar ligado ao seu ambiente. Ele é percebido através de outros softwares, uma vez que seus componentes apenas gerenciam o fluxo de trabalho, através do controle da seqüência de funções das aplicações e do gerenciamento de filas, e auxiliam na resolução de exceções. Normalmente, estes componentes são intercambiáveis, funcionando em diversas aplicações, uma vez que as interfaces são padronizadas (ALLEN, 2000).

Apêndice B – Glossário inicial do sistema

Tabela 5 – Glossário (até o final da fase Planejar e Elaborar)

Termo	Comentários
Alterar Programa de Disciplina	A ação de um Professor, Chefe ou Coordenador para modificar um programa de Disciplina.
Aluno	Um usuário, autenticado, que estuda na UFPel e está subordinado a um Curso.
Ano	O ano em que a Disciplina foi inserida no currículo.
Ano Final	Ano a partir do qual os dados não são mais válidos, porque a Disciplina foi extinta, ou alterada ou excluída do currículo de um curso.
Aprovada	Define se uma Disciplina foi aprovada ou não por uma determinada Entidade, durante o Fluxo de Aprovação.
aprovada	Indica se um Programa de Disciplina está ou não aprovado pelo Coordenador do Curso a que será ministrada.
Aprovar Programa de Disciplina	A ação de um Chefe, Coordenador ou COCEPE para indicar a aceitação ou rejeição de um Programa de Disciplina.
Responsável	O usuário que iniciou um processo de aprovação de uma Disciplina.
Bibliografia	A bibliografia sugerida para uma Disciplina.
Carga Horária	A Carga Horária da Disciplina.
Chefe	Um usuário que detém poder de decisão num Departamento.
COCEPE	Um usuário do COCEPE.
Código da Disciplina	O código de uma determinada Disciplina.
Consultar Programa	A ação de um cliente consultar um Programa de Disciplina aprovado.
Consultar Situação do Programa	A ação de um funcionário consultar o nível atual de um Programa de Disciplina
Conteúdo efêmero de Disciplina	O setor de um Programa de Disciplina que contém informações voláteis. Podem ser substituídos a cada novo semestre.
Conteúdo Programático	O Conteúdo a ser ministrado em uma Disciplina, ao longo do período letivo.
Coordenador	Um usuário que detém poder de decisão num Colegiado.
Créditos	A quantidade de créditos recebidos pelo aluno ao término da Disciplina.
Cronograma	O cronograma que o professor seguirá para desenvolver o conteúdo.
Curso	Um Curso da UFPel, que contém seu número de código.
Departamento	Um Departamento da UFPel, que contém seu número de código.
Disciplina-Curso	O setor de um Programa de Disciplina que varia de acordo com os diversos Cursos em que a mesma é ministrada.
Disciplina-Curso não aprovada	O setor de um Programa de Disciplina ainda não aprovada que varia de acordo com os diversos Cursos em que a mesma é ministrada.
Disciplinas aprovadas	Os Programas de Disciplinas que estão aprovados, sejam eles atuais ou que já tenham caducado.
Disciplinas não-aprovadas	Programas de Disciplinas que estão em processo de aprovação
DRA	Um usuário do DRA.
Ementa	A ementa de uma Disciplina.
Fluxo de Aprovação	O nível de aprovação atual da Disciplina.
Funcionário	Um usuário do sistema, cadastrado e autenticado (“logged in”) para utilizá-lo.

Termo	Comentários
gerente dos Programas de Disciplinas	Gerencia o processo de aprovação, consulta, inserção e aprovação dos Programas de Disciplinas.
Imprimir Programa	A ação de um cliente imprimir um Programa de Disciplina aprovado.
Incluir Programa de Disciplina	A ação de um Professor, Chefe, Coordenador ou DRA incluir um novo programa de Disciplina.
Lista de Professores	Os Professores que trabalham na UFPel e são subordinados a um Departamento. Eles têm um número de código.
Método de Avaliação	O método de avaliação que será utilizado na Disciplina.
Metodologia	Relaciona os procedimentos didáticos que serão utilizados ao longo da Disciplina.
Natureza Semanal Exercícios	Número de aulas semanais destinadas a exercícios que serão ministradas na Disciplina.
Natureza Semanal Prática	Número de aulas práticas semanais que serão ministradas na Disciplina.
Natureza Semanal Teórica	Número de aulas teóricas semanais que serão ministradas na Disciplina.
Nível de Aprovação	O estado em que o Programa de Disciplina se encontra, no processo de aprovação
Nome da Disciplina	O nome de uma determinada Disciplina.
Objetivo	As habilidades e os conhecimentos que se deseja desenvolver nos alunos com a Disciplina.
Obrigatória / Optativa	Caráter da Disciplina no Currículo de um Curso.
Pré-Requisitos	Disciplinas necessárias para a matrícula numa Disciplina, ou carga horária ou créditos mínimos necessários para esta matrícula.
Prof. Responsável Professor	O professor Responsável pela Disciplina. Um usuário que atua num Departamento e/ou Curso mas não tem o poder de decisões deste(s) órgão(s).
Profs. Ministrantes	Os professores que ministram a Disciplina. Podem ser apenas 1 ou vários. O Prof. Responsável pode ou não estar incluído nesta lista, e pode ser o único professor a ministrar, ou pode não ministrar aulas, também.
Programa de Disciplina	Um Programa de Disciplina, esteja ele em processo de aprovação ou já aprovado.
Semestre	O semestre específico em que a Disciplina foi inserida no currículo.
Semestre Final	Semestre do ano em que os dados não são mais válidos, seja porque a Disciplina foi extinta, alterada ou excluída do currículo de um curso.
Unidade	Uma divisão da UFPel, que engloba Departamentos.
Usuário	Uma pessoa qualquer, não cadastrada no sistema ou que ainda não tenha entrado no sistema.

Apêndice C – Glossário Completo

Tabela 6 – Glossário

Termo	Comentários
Alterar Programa de Disciplina	A ação de um Professor, Chefe ou Coordenador para modificar um programa de Disciplina.
Aluno	Um usuário, autenticado, que estuda na UFPel e está subordinado a um Curso.
Ano	O ano em que a Disciplina foi inserida no currículo.
Ano Final	Ano a partir do qual os dados não são mais válidos, porque a Disciplina foi extinta, ou alterada ou excluída do currículo de um curso.
Aprovada	Define se uma Disciplina foi aprovada ou não por uma determinada Entidade, durante o Fluxo de Aprovação.
aprovada	Indica se um Programa de Disciplina está ou não aprovado pelo Coordenador do Curso a que será ministrada.
Aprovar Programa de Disciplina	A ação de um Chefe, Coordenador ou COCEPE para indicar a aceitação ou rejeição de um Programa de Disciplina.
Atualizar Programa de Disciplina	A ação de um Professor, Coordenador ou Chefe atualizar um Programa de Disciplina, tornando-o compatível com o semestre em questão.
Bibliografia	A bibliografia sugerida para uma Disciplina.
Carga Horária	A Carga Horária da Disciplina.
Chefe	Um usuário que detém poder de decisão num Departamento.
COCEPE	Um usuário do COCEPE.
Código da Disciplina	O código de uma determinada Disciplina.
Consultar Programa	A ação de um cliente consultar um Programa de Disciplina aprovado.
Consultar Situação do Programa	A ação de um funcionário consultar o nível atual de um Programa de Disciplina
Conteúdo efêmero de Disciplina	O setor de um Programa de Disciplina que contém informações voláteis. Podem ser substituídos a cada novo semestre.
Conteúdo Programático	O Conteúdo a ser ministrado em uma Disciplina, ao longo do período letivo.
Coordenador	Um usuário que detém poder de decisão num Colegiado.
Créditos	A quantidade de créditos recebidos pelo aluno ao término da Disciplina.
Cronograma	O cronograma que o professor seguirá para desenvolver o conteúdo.
Curso	Um Curso da UFPel, que contém seu número de código.
Departamento	Um Departamento da UFPel, que contém seu número de código.
Disciplina-Curso	O setor de um Programa de Disciplina que varia de acordo com os diversos Cursos em que a mesma é ministrada.
Disciplina-Curso não aprovada	O setor de um Programa de Disciplina ainda não aprovada que varia de acordo com os diversos Cursos em que a mesma é ministrada.
Disciplinas aprovadas	Os Programas de Disciplinas que estão aprovados, sejam eles atuais ou que já tenham caducado.
Disciplinas não-aprovadas	Programas de Disciplinas que estão em processo de aprovação
DRA	Um usuário do DRA.
Ementa	A ementa de uma Disciplina.

Termo	Comentários
Enviar Memorando	A ação de um COCEPE para enviar um modelo de memorando, utilizado a fim de dar continuidade ao processo de aprovação.
Fluxo de Aprovação	O nível de aprovação atual da Disciplina.
Funcionário gerente dos Programas de Disciplinas	Um usuário do sistema, cadastrado e autenticado ("logged in") para utilizá-lo. Gerencia o processo de aprovação, consulta, inserção e aprovação dos Programas de Disciplinas.
Imprimir Programa	A ação de um cliente imprimir um Programa de Disciplina aprovado.
Incluir Programa de Disciplina	A ação de um Professor, Chefe, Coordenador ou DRA incluir um novo programa de Disciplina.
Lista de Professores	Os Professores que trabalham na UFPel e são subordinados a um Departamento. Eles têm um número de código.
Método de Avaliação	O método de avaliação que será utilizado na Disciplina.
Metodologia	Relaciona os procedimentos didáticos que serão utilizados ao longo da Disciplina.
Natureza Semanal Exercícios	Número de aulas semanais destinadas a exercícios que serão ministradas na Disciplina.
Natureza Semanal Prática	Número de aulas práticas semanais que serão ministradas na Disciplina.
Natureza Semanal Teórica	Número de aulas teóricas semanais que serão ministradas na Disciplina.
Nível de Aprovação	O estado em que o Programa de Disciplina se encontra, no processo de aprovação
Nome da Disciplina	O nome de uma determinada Disciplina.
Objetivo	As habilidades e os conhecimentos que se deseja desenvolver nos alunos com a Disciplina.
Obrigatória / Optativa	Caráter da Disciplina no Currículo de um Curso.
Pré-Requisitos	Disciplinas necessárias para a matrícula numa Disciplina, ou carga horária ou créditos mínimos necessários para esta matrícula.
Prof. Responsável Professor	O professor Responsável pela Disciplina. Um usuário que atua num Departamento e/ou Curso mas não tem o poder de decisões deste(s) órgão(s).
Profs. Ministrantes	Os professores que ministram a Disciplina. Podem ser apenas 1 ou vários. O Prof. Responsável pode ou não estar incluído nesta lista, e pode ser o único professor a ministrar, ou pode não ministrar aulas, também.
Programa de Disciplina	Um Programa de Disciplina, esteja ele em processo de aprovação ou já aprovado.
Responsável	O usuário que está responsável por um processo de aprovação de uma Disciplina.
Semestre	O semestre específico em que a Disciplina foi inserida no currículo.
Semestre Final	Semestre do ano em que os dados não são mais válidos, seja porque a Disciplina foi extinta, alterada ou excluída do currículo de um curso.
Unidade	Uma divisão da UFPel, que engloba Departamentos.
Usuário	Uma pessoa qualquer, não cadastrada no sistema ou que ainda não tenha entrado no sistema.

Apêndice D – Planejar e Elaborar

São definidos, nesta tarefa (1.6), os casos de uso Consultar Programa (Fig. 21), Imprimir Programa (Fig. 22), Consultar Situação do Programa (Fig. 23), Incluir Programa de Disciplina (Fig. 24) e Alterar Programa de Disciplina (Fig. 25). Pode-se observar, em cada um deles, os atores envolvidos, a finalidade geral de cada um bem como um resumo de sua funcionalidade. A seguir são exibidos os tipos e as referências cruzadas, relacionando-os com os requisitos coletados na tarefa 1.3.

Por último, temos a seqüência típica de eventos, com as ações e respostas do ator e do sistema, bem como as seqüências alternativas, onde é relatada qual a alteração necessária e em qual linha, bem como qual a variação que esta alteração acarreta.

Caso de Uso:	Consultar Programa
Atores:	Cliente
Finalidade:	Consultar o Programa de uma Disciplina, de modo que possa ser visualizado na tela.
Visão Geral:	O cliente indica o código, ano, semestre e curso da Disciplina cujo Programa de Disciplina quer consultar, seleciona a opção de Consultar Programa e o sistema exibe uma página <i>Web</i> com o Programa de Disciplina.
Tipo:	1. Primário 2. Essencial
Referências Cruzadas:	<i>Funções</i> : R1.4
Seqüência Típica de Eventos	
Ação do Ator	
Resposta do Sistema	
1. O cliente digita o código da Disciplina, o semestre, o ano e o curso desejado (em que a Disciplina teria sido ministrada) e seleciona a opção Consultar Programa.	2. O sistema consulta a base de dados dos Programas pelo código da Disciplina, que tenha sido ministrada no Curso escolhido, e a dupla semestre/ano igual ou anterior ao escolhido.
4. O cliente verifica os dados exibidos.	3. O sistema exibe uma página formatada <i>Web</i> com o único resultado obtido, exibindo apenas alguns dados relevantes, tais como código, nome e objetivos da Disciplina, entre outros.
Seqüências alternativas:	
Linha 3: o Ator é um Aluno do Curso em questão ou um outro Funcionário. O sistema exibe uma página formatada <i>Web</i> com todos os campos.	

Figura 21 – Caso de Uso: Consultar Programa

Caso de Uso:	Imprimir Programa
Atores:	Cliente
Finalidade:	Consultar o Programa de uma Disciplina, de modo que possa ser impresso.
Visão Geral:	O cliente indica o código, ano, semestre e curso da Disciplina cujo Programa de Disciplina quer consultar, seleciona a opção de Imprimir Programa e o sistema exibe uma página pronta para ser impressa.
Tipo:	1. Primário 2. Essencial
Referências	<i>Funções:</i> R1.4
Cruzadas:	<i>Casos de Uso:</i> Consulta Programa
Seqüência Típica de Eventos	
Ação do Ator	Resposta do Sistema
1. O cliente digita o código da Disciplina, o semestre, o ano e o curso desejado (em que a Disciplina teria sido ministrada) e seleciona a opção Imprimir Programa.	2. O sistema consulta a base de dados dos Programas pelo código da Disciplina, que tenha sido ministrada no Curso escolhido, e a dupla semestre/ano igual ou anterior ao escolhido.
4. O cliente verifica os dados exibidos e imprime a página.	3. O sistema exibe uma página para impressão com o único resultado obtido, exibindo apenas alguns dados relevantes, tais como código, nome e objetivos da Disciplina, entre outros.
Seqüências alternativas:	
Linha 3: o Ator é um Aluno do Curso em questão ou um outro Funcionário. O sistema exibe uma página formatada <i>Web</i> com todos os campos.	

Figura 22 – Caso de Uso: Imprimir Programa

Caso de Uso:	Consultar Situação do Programa
Atores:	Funcionário
Finalidade:	Verificar a situação atual do Programa, bem como os estágios pelo qual o programa passou, caso não aprovado ainda.
Visão Geral:	O funcionário acessa o sistema, indica o código, ano e semestre da Disciplina cuja situação quer consultar, seleciona a opção de Consultar Situação do Programa e o sistema exibe a situação atual.
Tipo:	1. Primário 2. Essencial
Referências	<i>Funções:</i> R2.1
Cruzadas:	
Seqüência Típica de Eventos	
Ação do Ator	Resposta do Sistema
1. O funcionário digita o código da Disciplina, o semestre e o ano desejado (em que a Disciplina teria sido ministrada) e seleciona a opção Consultar Situação do Programa.	2. O sistema consulta a base de dados dos Programas pelo código da Disciplina e a dupla semestre/ano igual ao escolhido, começando pelos Programas de Disciplinas não aprovados.
	3. O sistema consulta a base de dados dos relatórios dos Programas pelo código da Disciplina.
5. O funcionário verifica os dados exibidos.	4. O sistema exibe uma página formatada <i>Web</i> com a lista dos resultados obtidos.

Figura 23 – Caso de Uso: Consultar Situação do Programa

Caso de Uso:	Incluir Programa de Disciplina	
Atores:	Professor, Chefe, Coordenador, DRA	
Finalidade:	Incluir um novo Programa de Disciplina na base de dados	
Visão Geral:	O usuário seleciona a opção de Incluir Programa de Disciplina, preenche todos os campos do formulário e seleciona a opção de avançar.	
Tipo:	1. Primário 2. Essencial	
Referências:	<i>Funções:</i> R1.1, R1.2, R1.6, R2.3	
Cruzadas:	<i>Casos de Uso:</i> Aprovar Programa de Disciplina, Alterar Programa de Disciplina	
Seqüência Típica de Eventos		
	Ação do Ator	Resposta do Sistema
	1. O usuário seleciona a opção de Incluir Programa de Disciplina.	2. O sistema exibe um formulário com todos os campos em branco, para ser preenchido.
	3. O usuário preenche todos os campos obrigatórios (o que exclui o campo do código da Disciplina, de ano e semestre finais, de pré-requisitos e de professores ministrantes, bem como a multiplicidade de cursos) e seleciona a opção de avançar.	4. O sistema salva os campos na base de dados dos Programas.
	Seqüências alternativas	
	Linha 4: o Ator não preenche o campo de Código da Disciplina: O Programa é salvo na base de dados com um código padrão, dependente do código do Departamento.	
	Linha 4: o Ator é um Professor. O Programa de Disciplina é salvo na base de dados com o nível de aprovação 0.	
	Linha 4: o Ator é um Coordenador. O Programa de Disciplina é salvo na base de dados com a Disciplina aprovada pelo Curso do Coordenador.	
	Linha 4: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe.	
	Linha 4: o Ator é um COCEPE ou DRA. O Programa de Disciplina é copiado para a base de dados definitiva e é apagado da base de dados de aprovação.	

Figura 24 – Caso de Uso: Incluir Programa de Disciplina

Caso de Uso:	Alterar Programa de Disciplina	
Atores:	Professor, Chefe, Coordenador	
Finalidade:	Alterar um Programa de Disciplina existente na base de dados e que ainda não foi aprovada.	
Visão Geral:	O usuário acessa o sistema, indica o código, ano e semestre da Disciplina cujo Programa deseja alterar e seleciona a opção de Alterar Programa de Disciplina.	
Tipo:	1. Secundário 2. Essencial	
Referências:	<i>Funções:</i> R1.2, R2.2	
Cruzadas:	<i>Casos de Uso:</i> Incluir Programa de Disciplina, Aprovar Programa de Disciplina	
Seqüência Típica de Eventos		
	Ação do Ator	Resposta do Sistema
	1. O usuário digita o código da Disciplina e seleciona a opção Alterar Programa de Disciplina.	2. O sistema consulta a base de dados dos Programas pelo código da Disciplina na base de dados dos Programas de Disciplinas não aprovados.
		3. O sistema exibe um formulário com todos os campos preenchidos, para serem alterados.
	4. O usuário altera os campos que desejar e seleciona a opção de avançar.	5. O sistema salva os campos na base de dados dos Programas.
	Seqüências alternativas	
	Linha 3: o Ator não é um Chefe e não possui permissão de alterar o Programa. O sistema exibe a mensagem de que o Programa de Disciplina não pode ser alterado por este usuário.	
	Linha 3: o Ator é um Chefe e não possui permissão de alterar o programa. O sistema exibe uma página formatada Web com a Disciplina preenchida e um espaço para ser preenchido com o código do novo responsável pela inclusão da Disciplina.	
	Linha 5: o Ator é um Professor, responsável pela Disciplina. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Professor.	
	Linha 5: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe.	
	Linha 5: o Ator é um Chefe e não possui permissão de alterar o programa. As alterações relativas ao funcionário responsável pela Disciplina são salvos na base de dados.	
	Linha 5: o Ator é um DRA. As alterações são salvas na base de dados definitiva e a Disciplina é apagada da base de dados de aprovação	

Figura 25 – Caso de Uso: Alterar Programa de Disciplina

Apêndice E – 1º Ciclo de Desenvolvimento

Nesta tarefa (2.1.3.5), são exibidos os Diagramas de Seqüência dos casos de uso Consultar Programa de Disciplina (Fig. 26) e Incluir Programa de Disciplina (Fig. 27), onde se pode ver a interação do ator com o sistema e o relacionamento entre os diversos objetos do sistema.

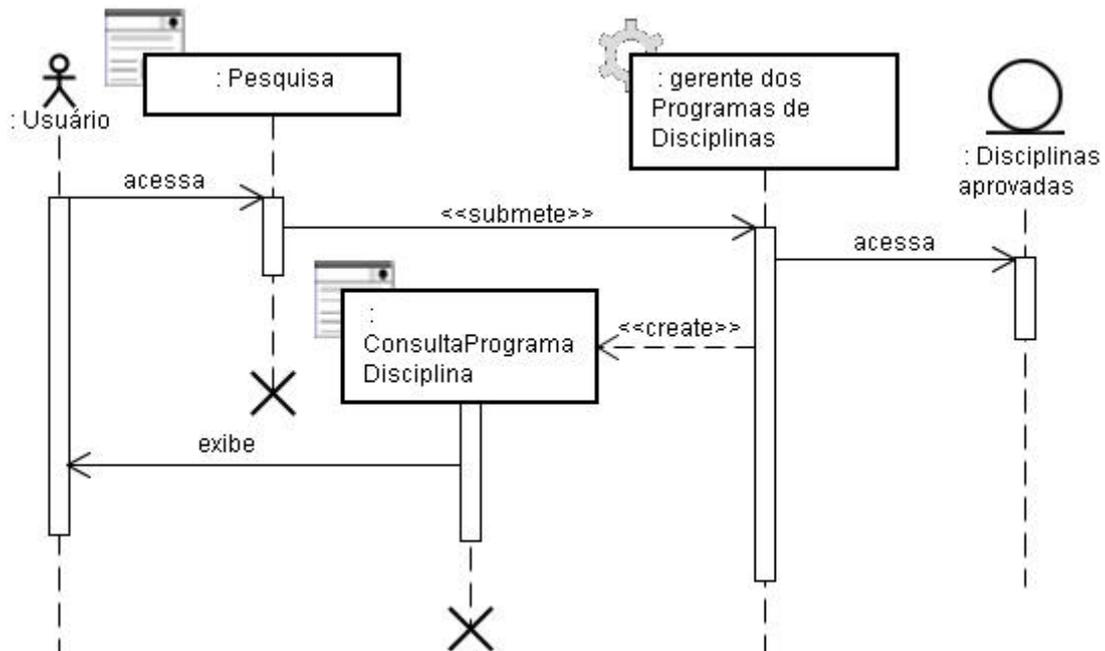


Figura 26 – Diagrama de Seqüência: Consultar Programa de Disciplina

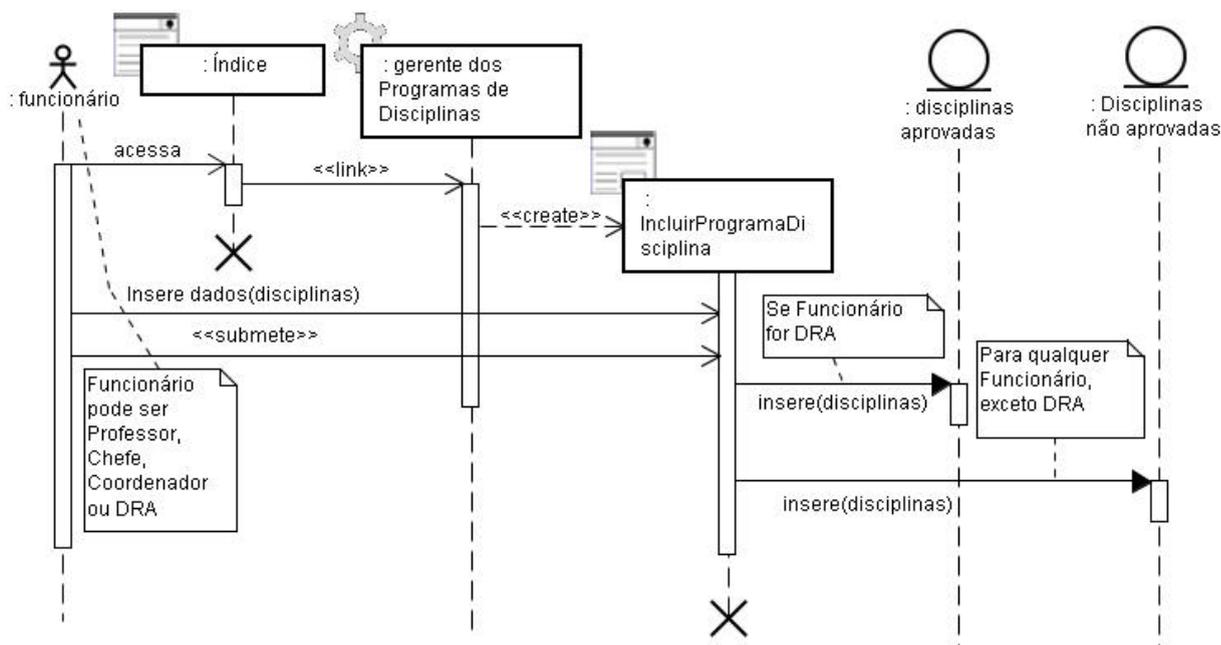


Figura 27 – Diagrama de Seqüência: Incluir Programa de Disciplina

De posse dos Diagramas de Seqüência, torna-se possível Definir os contratos das operações (2.1.3.6). Estes contratos exibem os parâmetros necessários à execução de cada operação, as responsabilidades de cada operação, as referências cruzadas com os casos de uso e os requisitos coletados, exceções à execução da operação, bem como as pré e pós condições (Fig. 28 e Fig. 29).

Contrato	
Nome:	Consultar Programa (código; semestre; ano; curso)
Responsabilidades:	Exibir o Programa da Disciplina com o código entrado, válido para o semestre/ano/curso também indicados; se curso não for indicado, exibirá todos os cursos válidos para o semestre/ano
Tipo:	Sistema
Referências Cruzadas:	Funções do Sistema: R1.4
Notas:	
Exceções:	Se não houver Disciplina a ser mostrada, exibir o erro em uma página com possibilidade de retorno.
Saída:	
Pré-condições:	O código da Disciplina é fornecido pelo cliente. Se a dupla semestre/ano não for indicada, usar a atual. Se o curso não for indicado, pesquisar todos.
Pós-condições:	Se existir uma Disciplina com o código igual ao indicado, uma dupla semestre/ano igual ou inferior a pesquisada, e um curso igual ao informado (ou no mínimo um curso) exibir o Programa de Disciplina na nova página Web; Se existir uma Disciplina com o código igual ao indicado, mas ela já estiver extinta para o curso indicado (ou todos, caso não seja indicado um curso), exibir o Programa de Disciplina na nova página Web, exibindo a dupla ano/semestre de inclusão e de exclusão. Se o usuário não estiver registrado no sistema, ou se for um Aluno consultando um Programa de um Curso que não for o seu próprio, o Programa de Disciplina referente à consulta aparecerá incompleto, contendo apenas alguns dados, tais como código, nome e objetivos da Disciplina, entre outros.

Figura 28 – Contrato da operação: Consultar Programa

Contrato	
Nome:	Incluir Programa de Disciplina (usuario; discNAprovadas; discNAprovadasCurso (5 vezes, excetuando-se código, ano e semestre, que não são informados) (excetuando-se protocolo, que não é informado))
Responsabilidades:	Incluir um Programa de Disciplina na Base de Dados dependente do nível hierárquico do usuário. Se o nível do usuário não for o de aprovação total, enviar mensagem para o nível hierárquico imediatamente superior.
Tipo:	Sistema
Referências Cruzadas:	Funções do Sistema: R1.1, R1.2, R1.6, R2.3 Casos de Uso: Aprovar Programa de Disciplina, Alterar Programa de Disciplina
Notas:	
Exceções:	Se faltar algum dado a ser informado, retornar à tela com a mensagem de erro. Se já houver uma Disciplina com o mesmo conjunto de código, ano e semestre, retornar à tela com mensagem de erro.
Saída:	Envia uma mensagem para o(s) usuário(s) responsável(is) pelo próximo nível de aprovação.
Pré-condições:	O usuário conhece todos os dados a informar e está autenticado pelo sistema.
Pós-condições:	Se todos os dados forem preenchidos e não houver problema com o conjunto código- semestre-ano, e o usuário não possuir nível final de aprovação, um novo Programa de Disciplina foi criado na base de dados das Disciplinas não Aprovadas (<i>criação de instância</i>); Se o usuário não possuir nível final de aprovação, uma mensagem é enviada ao nível seguinte (<i>criação de instância</i>); A mensagem é relacionada ao usuário (<i>formada uma associação</i>); Se todos os dados forem preenchidos e não houver problema com o conjunto código- semestre-ano, e o usuário possuir nível final de aprovação, um novo Programa de Disciplina foi criado (<i>criação de instância</i>) Qualquer que seja o caso, é criada uma linha relatando a ação do usuário em Log das Disciplinas (<i>criada uma instância</i>)

Figura 29 – Contrato da operação: Incluir Programa de Disciplina

Uma vez que nenhum destes casos de uso mudam o estado do sistema, passou-se ao passo seguinte e à tarefa seguinte, Definir Casos de Uso Reais (2.1.4.1), como vistos nas Fig. 30 e 31, exibindo os casos Consultar Programa e Incluir Programa de Disciplina, respectivamente. Nestes casos, para evitar repetição de informação, ocultou-se as informações repetidas dos casos de uso essenciais correspondentes.

Após estes, são exibidas as Interfaces de Usuários e “Storyboards” (2,1,4,2), exibindo as telas (Fig. 32, 33, 34, 35, 36 e 37) criadas na fase de prototipação e que são utilizadas e referenciadas na descrição dos casos de uso reais.

Caso de Uso:	Consultar Programa	
Tipo:	1. Primário 2. Real	
Seqüência Típica de Eventos		
Ação do Ator		Resposta do Sistema
1. Na Janela de Pesquisa (Fig. 33) o cliente digita o código da Disciplina no campo para o Código da Disciplina, o semestre no campo para o Semestre Cursado, o ano no campo para o Ano Cursado e o Curso em Curso desejados (em que a Disciplina teria sido ministrada) e seleciona o botão Consultar Programa.		2. O sistema consulta as bases de dados dos Programas pelo código da Disciplina, o Curso e a dupla semestre/ano igual ou anterior ao escolhido.
4. O cliente verifica os dados exibidos.		3. O sistema exibe uma página formatada Web com o único resultado obtido.

Figura 30 – Caso de Uso Real: Consultar Programa

Caso de Uso:	Incluir Programa de Disciplina	
Tipo:	1. Primário 2. Real	
Seqüência Típica de Eventos		
Ação do Ator		Resposta do Sistema
1. O usuário seleciona a opção de Inserir novo Programa de Disciplina na Janela do Índice (Fig. 32).		2. O sistema exibe um formulário com todos os campos em branco, para ser preenchido (Fig. 35).
3. O usuário preenche os campos da Tela de Inserção (Fig. 35), sendo os campos indicados em vermelho opcionais, e seleciona o botão Inserir.		4. O sistema salva os campos na base de dados das Disciplinas e envia e-mails para os próximos níveis de aprovação.

Figura 31 – Caso de Uso Real: Incluir Programa de Disciplina

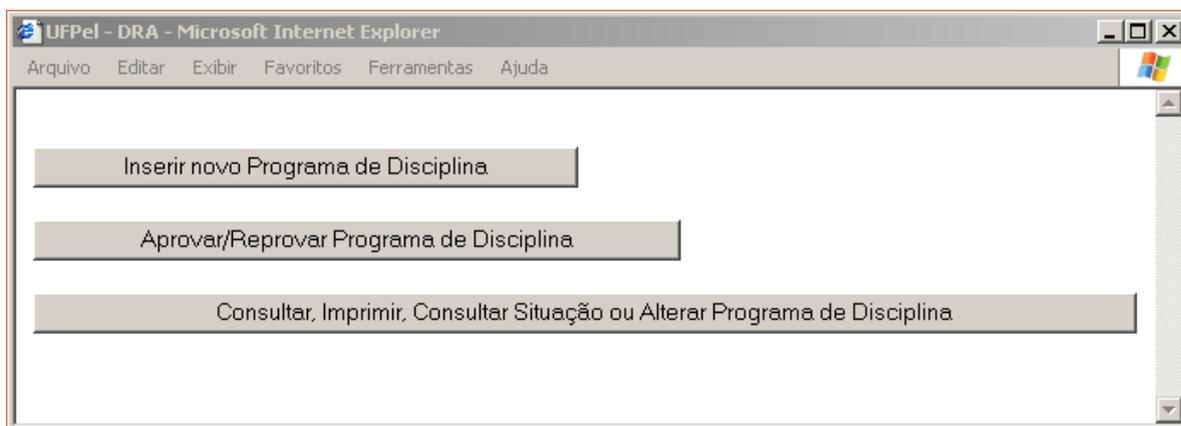
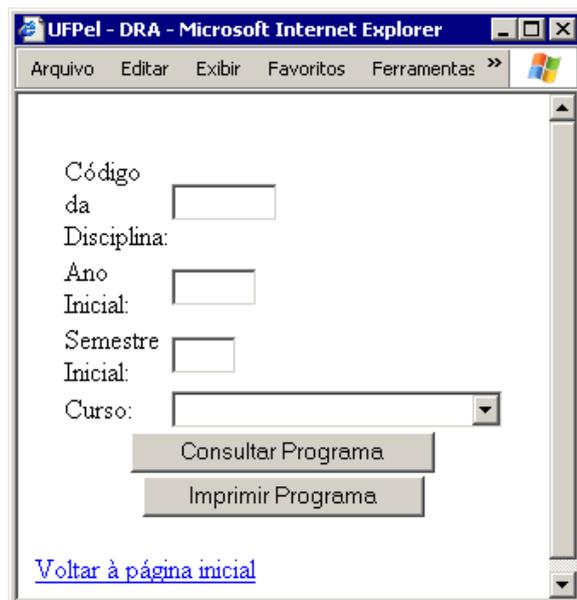


Figura 32 – Janela do índice



UFPeI - DRA - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas >>

Código da Disciplina:

Ano Inicial:

Semestre Inicial:

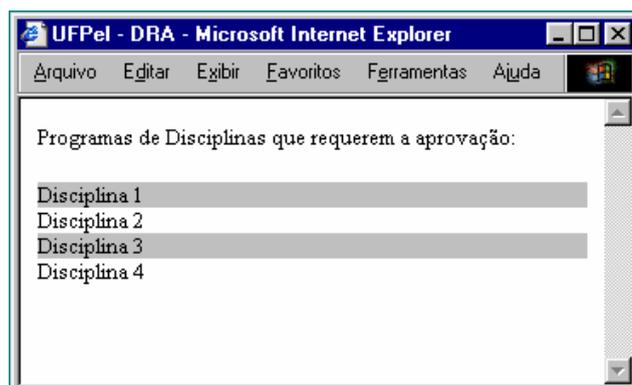
Curso:

Consultar Programa

Imprimir Programa

[Voltar à página inicial](#)

Figura 33 – Janela de Pesquisa



UFPeI - DRA - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Programas de Disciplinas que requerem a aprovação:

Disciplina 1

Disciplina 2

Disciplina 3

Disciplina 4

Figura 34 – Janela do índice de Disciplinas para aprovar

UPPel - DRA - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL DE PELOTAS
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE REGISTROS ACADÊMICOS

*- Os campos cujos nomes estejam em vermelho podem ser deixados em branco

Disciplina: Código:

Ano Inicial: Semestre Inicial: Carga Horária total: h Natureza Semanal: T -E -P Créditos:

Curso 1: Obrigatória Optativa Ano Final: Semestre Final:

Código dos pré-requisitos da Disciplina

Curso 2: Obrigatória Optativa Ano Final: Semestre Final:

Código dos pré-requisitos da Disciplina

Curso 3: Obrigatória Optativa Ano Final: Semestre Final:

Código dos pré-requisitos da Disciplina

Curso 4: Obrigatória Optativa Ano Final: Semestre Final:

Código dos pré-requisitos da Disciplina

Curso 5: Obrigatória Optativa Ano Final: Semestre Final:

Código dos pré-requisitos da Disciplina

Professor Responsável:

Professores Ministrantes:

Objetivos da Disciplina:

Ementa:

Conteúdo Programático:

Procedimentos Didáticos (Metodologia):

Desenvolvimento Programático (Cronograma):

Sistema de Avaliação:

Bibliografia:

Inserir [Cancelar](#)

Figura 35 – Janela de Inserção

UPPel - DRA - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL DE PELOTAS
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE REGISTROS ACADÊMICOS

*- Caso não concorde com algum campo da Disciplina, marque a caixa correspondente ao campo, antes de selecionar Reprovar.

Disciplina: XXXXXXXXXXXXXXXXXXXX Código: XXXXXXX

Ano Inicial: XXXX Semestre Inicial: X Carga Horária total: XXXX h Natureza Semanal: T: X -E: X -P: X Créditos: X

Curso 1 : XXXXXXX XXXX XXXXXXX (XX) Obrigatória

Código dos pré-requisitos da Disciplina

Professor Responsável: Prof. XXXXXXX

Professores Ministrantes:

Objetivos da Disciplina:
XXXXXXXXXXXX
XXXXXXXXXXXX

Ementa:
XXXXXXXXXXXX
XXXXXXXXXXXX

Conteúdo Programático:
XXXXXXXXXXXX
XXXXXXXXXXXX

Procedimentos Didáticos (Metodologia):
XXXXXXXXXXXX
XXXXXXXXXXXX

Desenvolvimento Programático (Cronograma):
XXXXXXXXXXXX
XXXXXXXXXXXX

Sistema de Avaliação:
XXXXXXXXXXXX
XXXXXXXXXXXX

Bibliografia:
XXXXXXXXXXXX
XXXXXXXXXXXX

Aprovar Cancelar Reprovar

Figura 36 – Janela de Aprovar

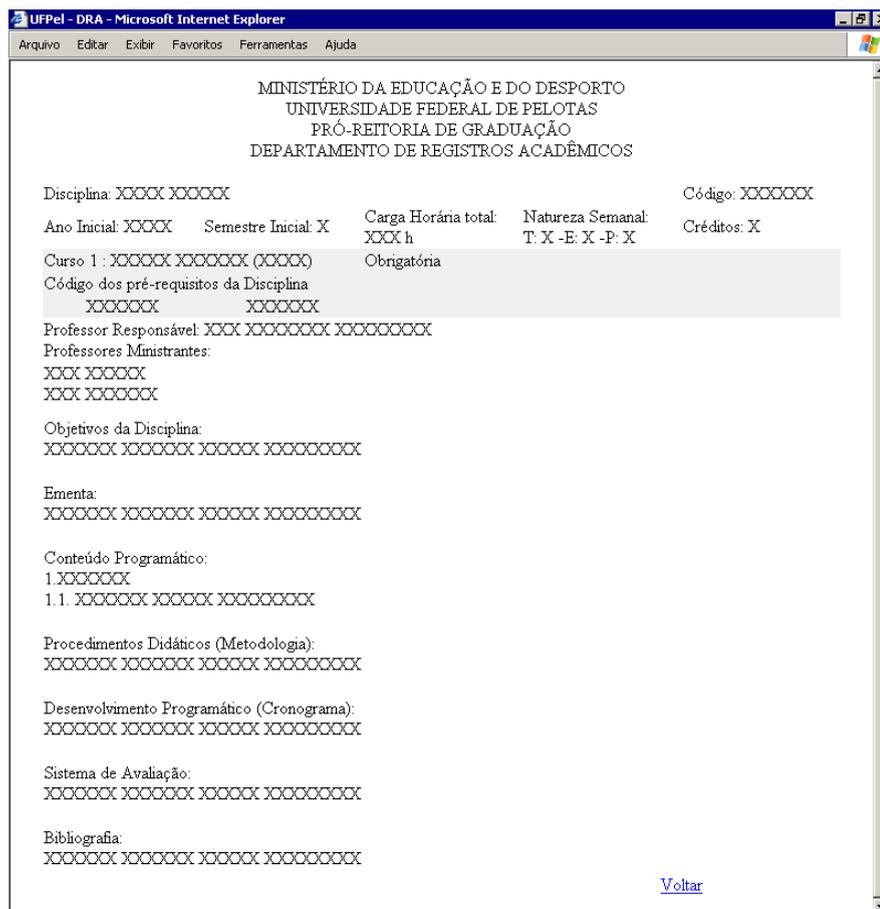


Figura 37 – Janela de Consultar Programa

Então, se elaboram Diagramas de Interação (2.1.4.3), onde se podem detectar as operações que as classes devem executar ao cumprir cada caso de uso, conforme se observa nas Fig. 38 e 39, exibindo os Diagramas de Comunicação dos casos de uso Consultar Programa de Disciplina e Incluir Programa de Disciplina..

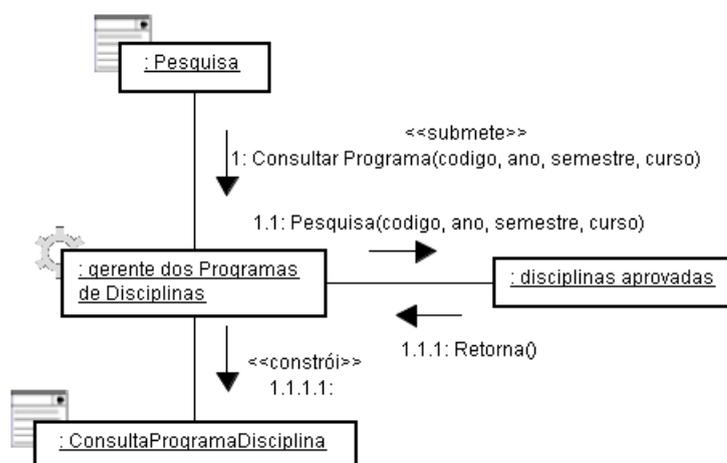


Figura 38 – Diagrama de Comunicação: Consultar Programa de Disciplina

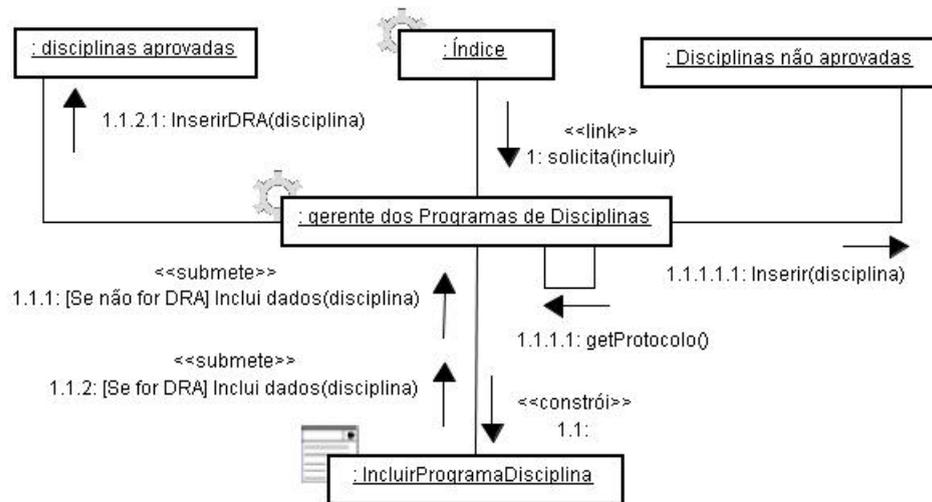


Figura 39 – Diagrama de Comunicação: Incluir Programa de Disciplina

Apêndice F – 2º Ciclo de Desenvolvimento

Continuando-se este Ciclo, Refinou-se o Diagrama de Caso de Uso (2.2.3.2) Imprimir Programa, o qual, para evitar-se ser repetido na íntegra, optou-se por exibir apenas seu nome e as alterações efetuadas (Fig. 40).

Caso de Uso: Imprimir Programa	
Seqüência Típica de Eventos	Resposta do Sistema
Ação do Ator	
1. O cliente digita o código da Disciplina, o semestre, o ano e o curso desejado (em que a Disciplina teria sido ministrada) e seleciona a opção Imprimir Programa.	2. O sistema consulta a base de dados dos Programas pelo código da Disciplina, que tenha sido ministrada no Curso escolhido, e a dupla semestre/ano igual ou anterior ao escolhido.
4. O cliente verifica os dados e seleciona a opção Imprimir Programa.	3. O sistema exibe uma página formatada Web com o único resultado obtido.
6. O cliente verifica os dados exibidos e imprime a página.	5. O sistema exibe uma página para impressão com o único resultado obtido.

Figura 40 – Refinamento de Caso de Uso: Imprime Programa

Uma vez que o Modelo Conceitual (2.2.3.3) não necessitou Refinamento, partiu-se para a Definição dos Diagramas de Seqüência (2.2.3.5), conforme podem ser vistos nas Fig. 41 e 42. A tarefa 2.2.3.4, Refinar o Glossário, não é exibida aqui. Ao invés disto, optou-se por exibir um Glossário com todas as alterações e adições feitas no modelo, incluído o 3º Ciclo de Desenvolvimento, no Apêndice C.

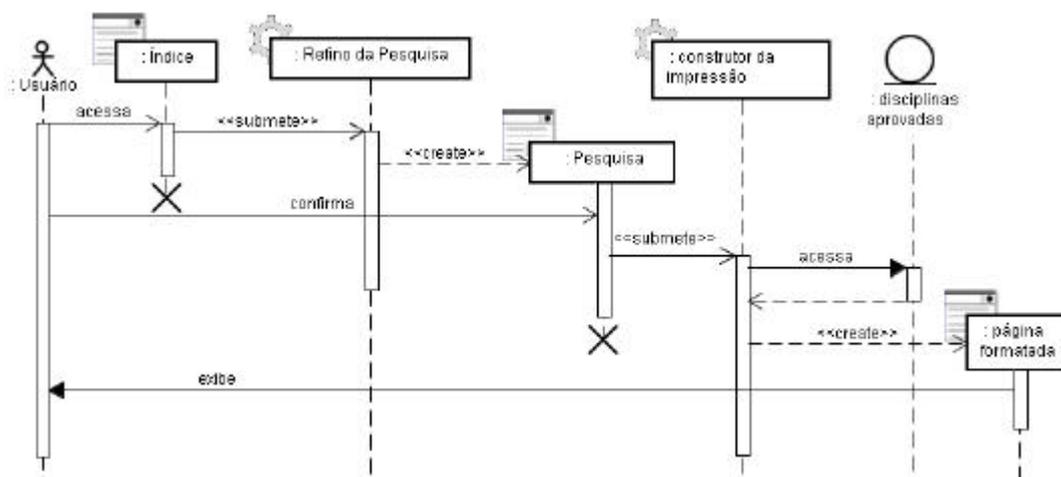


Figura 41 – Diagrama de Seqüência: Imprimir Programa de Disciplina

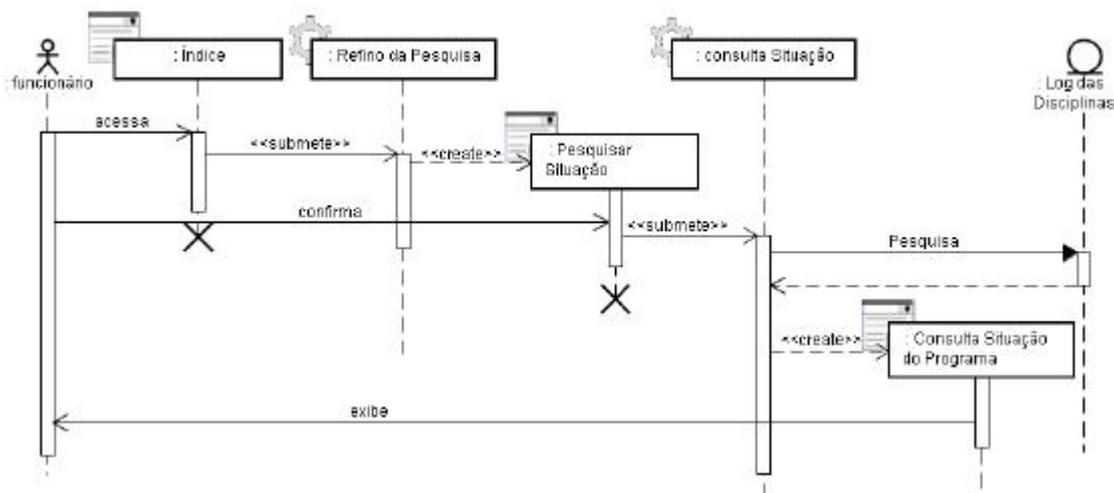


Figura 42 – Diagrama de Seqüência: Consultar Situação do Programa

Em seguida, Definiram-se os Contratos das Operações (2.2.3.6) Imprimir Programa e Consultar Situação do Programa, conforme se observa na Fig. 43 e na Fig. 44.

Contrato	
Nome:	Imprimir Programa (codigo; ano; semestre)
Responsabilidades:	Imprimir um Programa de Disciplina que esteja na Base de Dados final, de acordo com o ano e semestre desejados.
Tipo:	Interface
Referências Cruzadas:	Funções do Sistema: R1.4 Casos de Uso: Consulta Programa
Notas:	
Exceções:	
Saída:	Uma página formatada pronta para ser impressa.
Pré-condições:	Há um Programa de Disciplina aprovado que date de um ano e semestre igual ou anterior ao pedido pelo cliente.
Pós-condições:	

Figura 43 – Contrato da operação: Imprimir Programa

Contrato	
Nome:	Consultar Situação do Programa (codigo; ano; semestre; idAux)
Responsabilidades:	Promover o acompanhamento do andamento da aprovação.
Tipo:	Sistema
Referências Cruzadas:	Funções do Sistema: R2.1 Casos de Uso: Incluir Programa de Disciplina, Alterar Programa de Disciplina, Aprovar Programa de Disciplina
Notas:	
Exceções:	
Saída:	Uma página formatada com o resultado da pesquisa, ordenado pela data.
Pré-condições:	O funcionário conhece o código da Disciplina, o ano e o semestre em que esta Disciplina deve ser primeiro ministrada
Pós-condições:	- Existe mais de uma Disciplina com o mesmo trio de dados e o usuário deve selecionar especificamente qual delas deseja, informando, transparentemente, o parâmetro idAux.

Figura 44 – Contrato da operação: Consulta Situação do Programa

Como nenhuma das operações abordadas neste ciclo alteram o estado do sistema, não foi preciso Definir Diagramas de Estado (2.2.3.7), podendo-se passar à fase de Projetar (2.2.4), Definindo os Casos de Uso Reais (2.2.4.1) abordados nesta etapa (Fig. 45 e Fig. 46). Como anteriormente, optou-se por exibir apenas seus nomes e as alterações efetuadas, a fim de não repetir os Casos de Uso da subseção 5.1.6 na íntegra.

Caso de Uso:	Imprimir Programa
Tipo:	1. Primário 2. Real
Seqüência Típica de Eventos	
Ação do Ator	Resposta do Sistema
1. Na Janela de Pesquisa (Fig. 47) o cliente digita o código da Disciplina no campo para o Código da Disciplina, o semestre no campo para o Semestre Cursado e o ano no campo para o Ano Cursado desejados (em que a Disciplina teria sido ministrada) e seleciona o botão Imprimir Programa.	2. O sistema consulta a base de dados dos Programas pelo código da Disciplina e a dupla semestre/ano igual ou anterior ao escolhido.
4. O cliente verifica os dados exibidos e imprime o programa, usando a função de impressão de seu cliente <i>Web</i> .	3. O sistema exibe uma página formatada com o único resultado obtido, exibindo o semestre e o ano informados pelo cliente.

Figura 45 – Caso de Uso Real: Imprimir Programa

Caso de Uso:	Consultar Situação do Programa
Tipo:	1. Primário 2. Real
Seqüência Típica de Eventos	
Ação do Ator	Resposta do Sistema
1. Na Janela de Pesquisa (Fig. 47) o funcionário digita o código da Disciplina no campo para o Código da Disciplina, o semestre no campo para o Semestre Cursado e o ano no campo para o Ano Cursado desejados (em que a Disciplina teria sido ministrada) e seleciona o botão Consultar Situação do Programa.	2. O sistema consulta a base de dados do Registro da Alteração da Situação dos Programas pelo código da Disciplina e a dupla semestre/ano igual ao escolhido.
4. O funcionário verifica os dados exibidos.	3. O sistema exibe uma página formatada (Fig. 48) com os resultados obtidos, agrupando-os por instância de Programa inserida.

Figura 46 – Caso de Uso Real: Consulta Situação do Programa

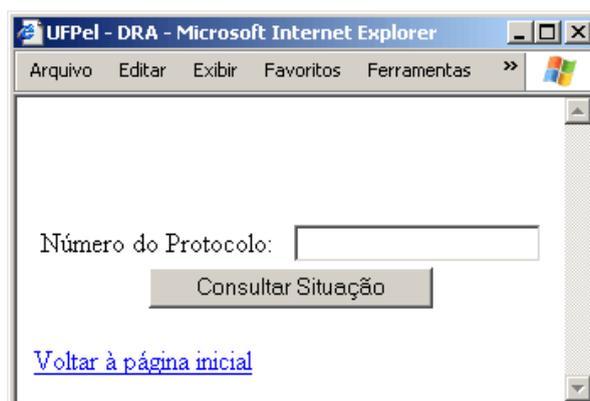


Figura 47 – Janela da Pesquisa da Situação do Programa

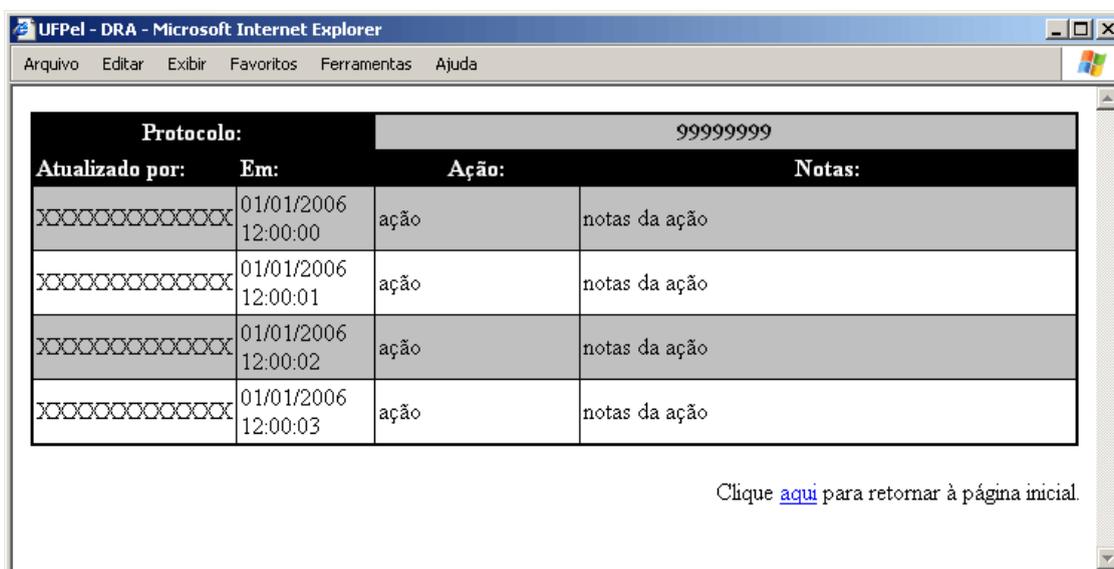


Figura 48 – Janela das alterações na Situação do Programa

A seguir, com base nos Casos de Uso, Definiram-se os Diagramas de Interação (2.2.3.3) (Fig. 49 e Fig. 50)

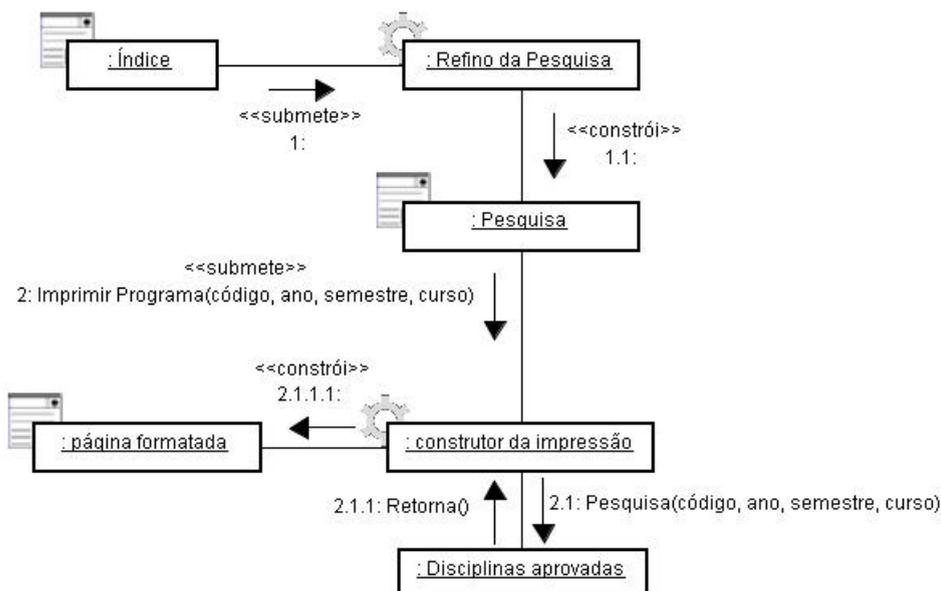


Figura 49 – Diagrama de Comunicação: Imprimir Programa

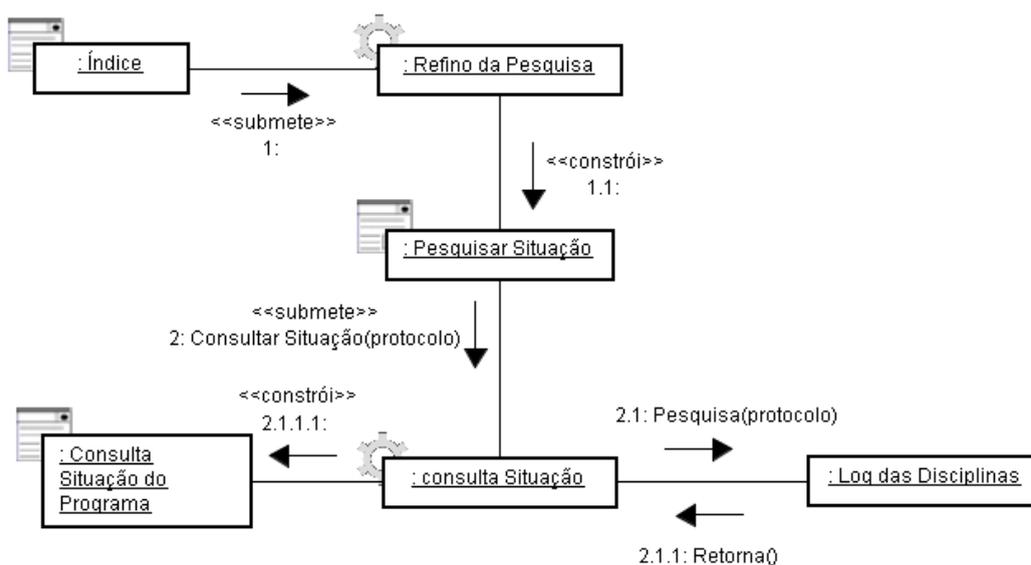


Figura 50 – Diagrama de Comunicação: Consultar situação do Programa

Devido às alterações no componente gerente dos Programas de Disciplina, as classes exibidas na Fig. 6 foram criadas ou alteradas, necessitando uma nova Definição do Diagrama de Classes (2.2.3.4), como se vê na Fig. 51.

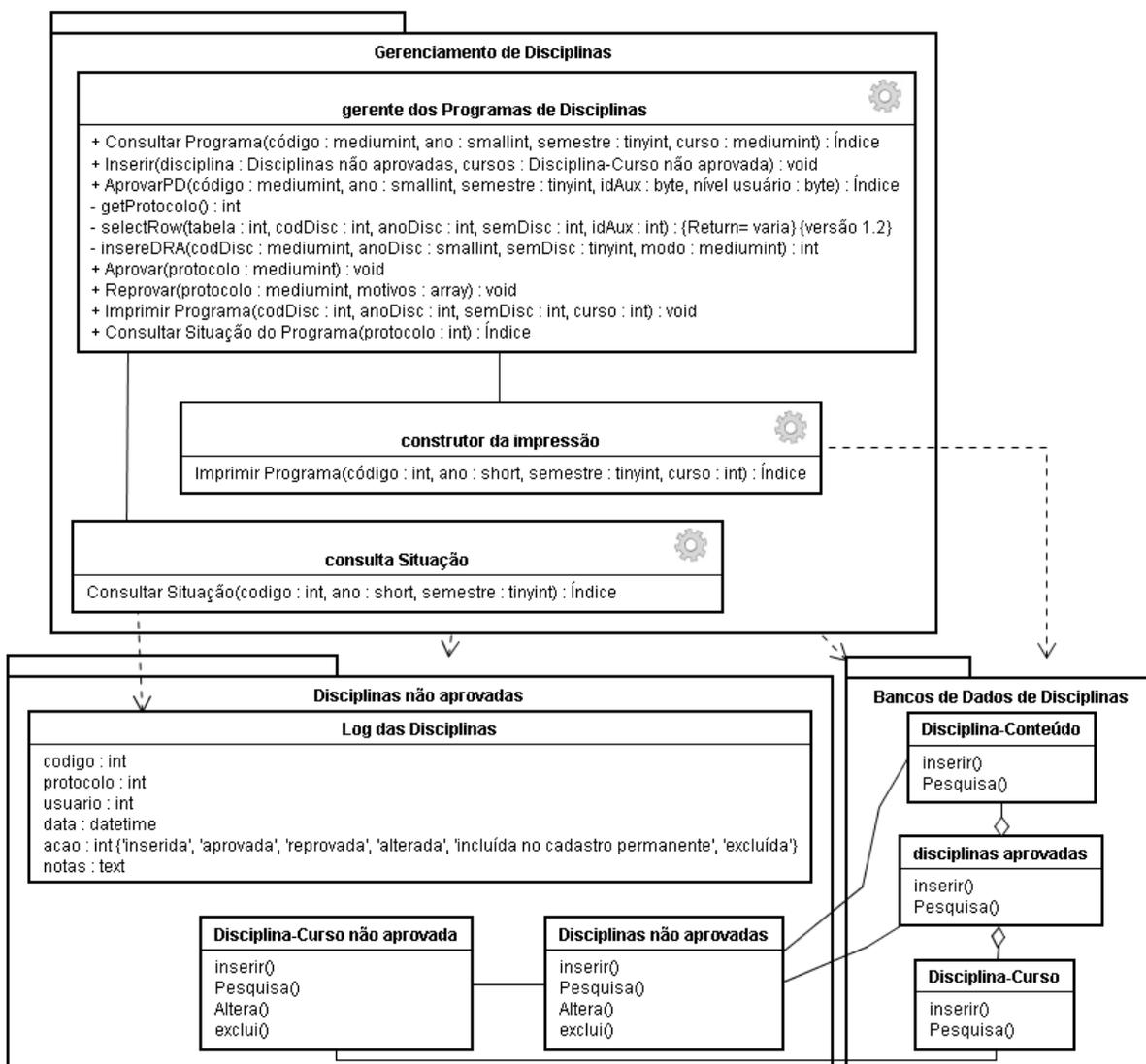


Figura 51 – Diagrama de Classes de Projeto para o 2º Ciclo

Ao se Refinar a Arquitetura do Sistema (2.2.4.5), optou-se por exibir na Fig 52 apenas os componentes que foram alterados a partir da arquitetura prevista para o 1º ciclo de desenvolvimento (Fig. 15), incluindo a alteração que houve na relação entre a «página do cliente» Pesquisa e a «página do servidor» gerente dos Programas de Disciplina.

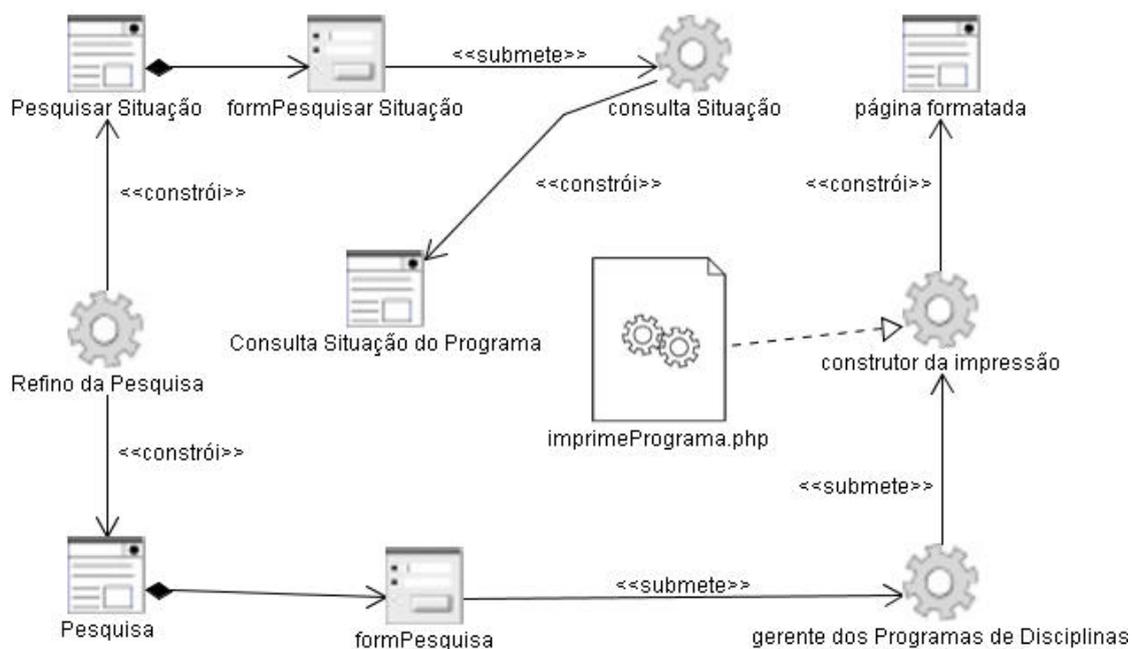


Figura 52 – Refinamento do Diagrama de Artefatos Web e Navegação, para o 2º ciclo de desenvolvimento

Após, o modelo Programado (2.2.4) foi Testado (2.2.5) e salvo como Ciclo de Desenvolvimento 2.zip, contendo os arquivos indice2.php, index1.php, gerenciaProgramaDisciplina.php, consultaSituacao.php, imprimePrograma.php e indiceAprovar.php.

Apêndice G – 3º Ciclo de Desenvolvimento

Continuando este Ciclo de Desenvolvimento, Caso de Uso Alterar Programa de Disciplina é alterado, conforme se vê na Fig. 53 e são inseridos os Casos de Uso Atualizar Programa de Disciplina (Fig. 54) e Enviar Memorando (Fig. 55). Também são exibidas as alterações realizadas nos Casos de Uso Incluir Programa de Disciplina (Fig. 56) e Aprovar Programa de Disciplina (Fig. 57).

Caso de Uso:	Alterar Programa de Disciplina	
Atores:	Professor, Chefe, Coordenador, DRA	
Visão Geral:	O usuário acessa o sistema, verifica uma Disciplina que contenha um ou mais campos que necessitem ser alterados e escolhe alterar a Disciplina	
Tipo:	1. Secundário 2. Essencial	
Seqüência Típica de Eventos		
	Ação do Ator	Resposta do Sistema
1.	O usuário seleciona a opção Aprovar Programa de Disciplina.	2. O sistema consulta a base de dados e exibe uma lista de todos os Programas de Disciplinas passíveis de alteração pelo usuário, incluindo o tipo de alteração (total, responsável ou código da Disciplina).
3.	O usuário seleciona um dos Programas, clicando sobre ele.	4. O sistema exibe uma página formatada <i>Web</i> com o Programa da Disciplina selecionada.
5.	O usuário seleciona a opção Alterar.	6. O sistema exibe um formulário com todos os campos preenchidos, para serem alterados.
7.	O usuário altera os campos que desejar e seleciona a opção de avançar.	8. O sistema salva os campos na base de dados dos Programas, reiniciando o Processo de Aprovação tendo como novo autor o usuário que efetuou as alterações.
Seqüências alternativas		
Linha 6: o Ator é um DRA: O único campo passível de alteração é o código da Disciplina.		
Linha 6: o Ator é um Chefe e o mesmo não é o autor da Disciplina: o único campo passível de alteração é o campo pra alterar o responsável pela Disciplina.		
Linha 8: o Ator é um Professor. O Programa de Disciplina é salvo na base de dados com o nível de aprovação de Professor.		
Linha 8: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe.		
Linha 8: o Ator é um Coordenador. O Programa de Disciplina é salvo na base de dados com a Disciplina aprovada pelo Curso do Coordenador.		
Linha 8: o Ator é um DRA. O Programa de Disciplina é salvo na base de e o Processo de Aprovação não é reiniciado.		

Figura 53 – Refinamento do Caso de Uso: Alterar Programa de Disciplina

Caso de Uso:	Atualizar Programa de Disciplina	
Atores:	Professor, Coordenador, COCEPE	
Finalidade:	Atualizar o conteúdo efêmero de um Programa de Disciplina.	
Visão Geral:	O funcionário escolhe qual o Programa de Disciplina vai atualizar, altera os campos que forem necessários e o envia.	
Tipo:	1. Secundário 2. Essencial	
Referências	<i>Funções:</i> R1.1, R1.2, R1.3, R1.6, R1.7, R1.8, R2.3	
Cruzadas:	Casos de Uso: Alterar Programa de Disciplina	
Seqüência Típica de Eventos		
	Ação do Ator	Resposta do Sistema
	1. O funcionário digita o código da Disciplina e seleciona a opção Atualizar Programa de Disciplina.	2. O sistema exibe um página formatada <i>Web</i> com o Programa de Disciplina escolhido, com alguns campos passíveis de alteração, devidamente preenchidos com as informações anteriores, além de campos para preencher com códigos de memorandos.
	3. O funcionário reescreve as informações que desejar, escreve os códigos dos memorandos e seleciona Atualizar.	4. O sistema salva as alterações na base de dados das disciplinas não aprovadas e exibe os memorandos a serem impressos e assinados.
	Seqüências alternativas:	
	Linha 4: o Ator é um Professor. O Programa de Disciplina é salvo na base de dados com o nível de aprovação de Professor. O sistema emite um memorando para ser assinado pelo Professor, endereçado ao Departamento do qual o Programa de Disciplina fará parte.	
	Linha 4: o Ator é um Coordenador. O Programa de Disciplina é salvo na base de dados com a Disciplina aprovada pelo Curso do Coordenador. O sistema emite um memorando para ser assinado pelo Coordenador, endereçado ao Departamento do qual o Programa de Disciplina fará parte.	
	Linha 4: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe. O sistema emite um memorando para cada Curso ao qual o Programa de Disciplina será utilizado no currículo. Este(s) memorando(s) deverá(ao) ser assinado(s) pelo Chefe.	

Figura 54 – Caso de Uso: Atualizar Programa de Disciplina

Caso de Uso:	Enviar Memorando	
Atores:	COCEPE	
Finalidade:	Enviar um novo modelo de memorando a ser impresso para ser enviado para as instâncias de aprovação.	
Visão Geral:	O COCEPE escreve o memorando, seguindo a linha geral indicada pelo sistema, acessa o sistema, seleciona o texto feito previamente e o envia para o banco de dados.	
Tipo:	1. Secundário 2. Essencial	
Referências	<i>Funções:</i> R1.1, R1.3, R1.5, R1.6, R1.7, R2.4	
Cruzadas:		
Seqüência Típica de Eventos		
	Ação do Ator	Resposta do Sistema
	1. O COCEPE escreve o modelo, baseado nas informações dadas pelo sistema, e o salva em seu computador.	
	2. O COCEPE acessa o sistema e escolhe a opção de Enviar Modelo de Memorando	3. O sistema exibe uma página formatada <i>Web</i> com a solicitação de envio de arquivo.
	4. O COCEPE indica o arquivo contendo o novo modelo de memorando, bem como qual a função deste memorando (seja Aprovação, Reprovação, etc.).	5. O sistema exibe uma página formatada <i>Web</i> informando que a operação foi bem sucedida.

Figura 55 – Caso de Uso: Enviar Memorando

Caso de Uso:	Incluir Programa de Disciplina
Referências:	<i>Funções:</i> R1.1, R1.2, R1.6, R1.7, R2.3
Cruzadas:	<i>Casos de Uso:</i> Aprovar Programa de Disciplina, Alterar Programa de Disciplina
Seqüências alternativas	
Linha 4: o Ator é um Professor. O Programa de Disciplina é salvo na base de dados com o nível de aprovação de Professor. O sistema emite um memorando para ser assinado pelo Professor, endereçado ao Departamento do qual o Programa de Disciplina fará parte.	
Linha 4: o Ator é um Coordenador. O Programa de Disciplina é salvo na base de dados com a Disciplina aprovada pelo Curso do Coordenador. O sistema emite um memorando para ser assinado pelo Coordenador, endereçado ao Departamento do qual o Programa de Disciplina fará parte.	
Linha 4: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe. O sistema emite um memorando para cada Curso ao qual o Programa de Disciplina será utilizado no currículo. Este(s) memorando(s) deverá(ao) ser assinado(s) pelo Chefe.	

Figura 56 – Caso de Uso: Incluir Programa de Disciplina

Caso de Uso:	Aprovar Programa de Disciplina
Referências:	<i>Funções:</i> R1.3, R1.5, R1.6, R1.7
Cruzadas:	<i>Casos de Uso:</i> Incluir Programa de Disciplina, Alterar Programa de Disciplina
Seqüência Típica de Eventos	
Ação do Ator	Resposta do Sistema
	4. O sistema exibe uma página formatada <i>Web</i> com o Programa da Disciplina selecionada, contendo campos que podem ser selecionados para o caso do usuário selecionar a opção Reprovar e um campo (para cada pessoa que deverá receber um memorando) para ser preenchido com um código do memorando a ser enviado.
5. O usuário preenche o(s) campo(s) relativos ao(s) memorando(s) e seleciona a opção Aprovar.	6. O sistema salva nas bases de dados as alterações necessárias e emite memorando(s).
Seqüências alternativas	
Linha 5: o Ator é um Coordenador, seleciona todos, alguns ou nenhum dos campos selecionáveis e seleciona a opção Reprovar. O sistema reduz o nível de aprovação a zero e envia um e-mail ao Responsável avisando que o mesmo foi reprovado e necessita alteração, avisando por quais motivos o mesmo foi reprovado pelo Coordenador. Também emite um memorando para ser impresso e assinado, destinado ao Responsável pelo Programa de Disciplina.	
Linha 5: o Ator é um Chefe, seleciona todos, alguns ou nenhum dos campos selecionáveis e seleciona a opção Reprovar. O sistema reduz o nível de aprovação a zero e envia um e-mail ao Responsável avisando que o mesmo foi reprovado e necessita alteração, avisando por quais motivos o mesmo foi reprovado pelo Chefe. Também emite um memorando para ser impresso e assinado, destinado ao Responsável pelo Programa de Disciplina.	
Linha 5: o Ator é um COCEPE, seleciona todos, alguns ou nenhum dos campos selecionáveis e seleciona a opção Reprovar. O sistema reduz o nível de aprovação a zero e envia um e-mail ao Responsável avisando que o mesmo foi reprovado e necessita alteração, avisando por quais motivos o mesmo foi reprovado pelo COCEPE. Também emite um memorando para ser impresso e assinado, destinado ao Responsável pelo Programa de Disciplina.	
Linha 6: o Ator é um Coordenador. O Programa de Disciplina é salvo na base de dados com a Disciplina aprovada pelo Curso do Coordenador. e é enviado um e-mail ao COCEPE, caso o Programa já tenha sido aprovado pelo Chefe e pelos demais Coordenadores dos Cursos em que a Disciplina é ministrada. Também emite um memorando para ser impresso e assinado, destinado ao COCEPE.	
Linha 6: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe e é enviado um e-mail aos Coordenadores responsáveis que ainda não tenham aprovado a Disciplina, ou ao COCEPE, caso todos os Coordenadores já tenham aprovado a Disciplina. Também emite um memorando para ser impresso e assinado, destinado aos Coordenadores que não tenham aprovado o Programa de Disciplina, ou ao COCEPE, caso todos os Coordenadores já tenham aprovado.	

Figura 57 – Caso de Uso: Aprovar Programa de Disciplina

Após esta tarefa, verificamos que estamos utilizando o mesmo Modelo Conceitual do ciclo anterior, que não necessitou Refinamento (2.3.3.3), e verificou-se, na tarefa de Refinar o Glossário (2.3.3.4), que a palavra “responsável” no lugar de “autor”, seria uma escolha muito mais adequada, uma vez que um usuário pode se tornar o responsável por um Programa de Disciplina mesmo não sendo seu autor. Nesta mesma tarefa, verificou-se, também, a necessidade da inclusão dos termos Atualizar Programa de Disciplina e Enviar Memorando. Desta forma, as alterações no glossário ficam representadas no Apêndice C.

A seguir, Definiram-se Diagramas de Seqüência (2.3.3.5), conforme se observa na Fig. 58, na Fig. 59 e Fig. 60.

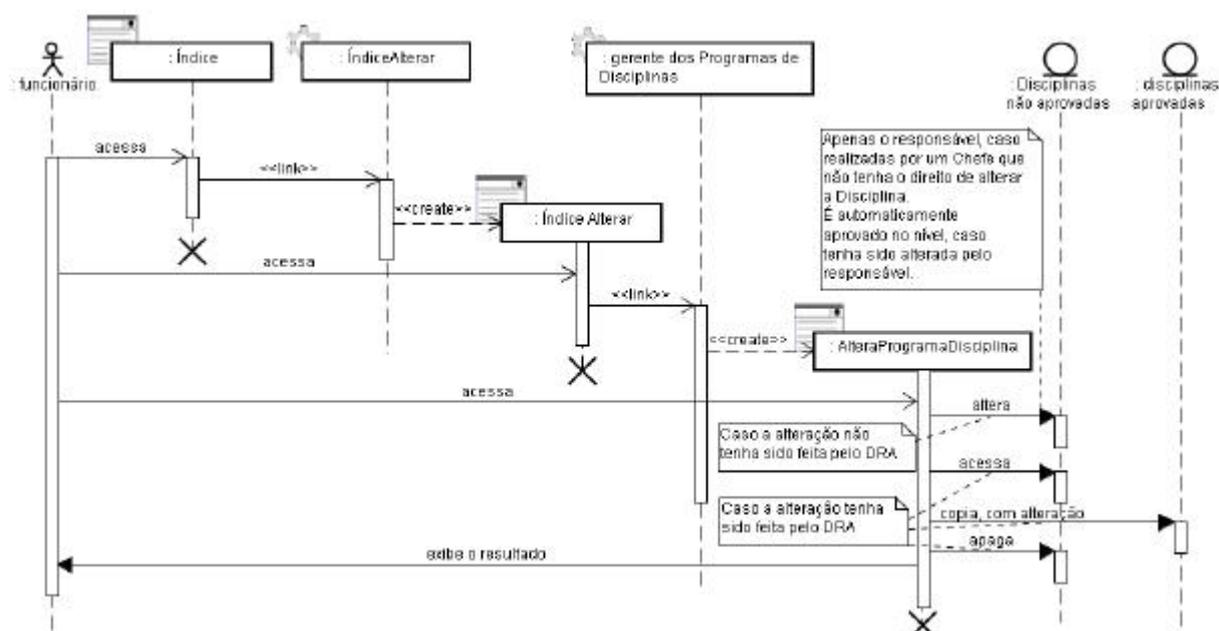


Figura 58 – Diagrama de Seqüência: Alterar Programa de Disciplina

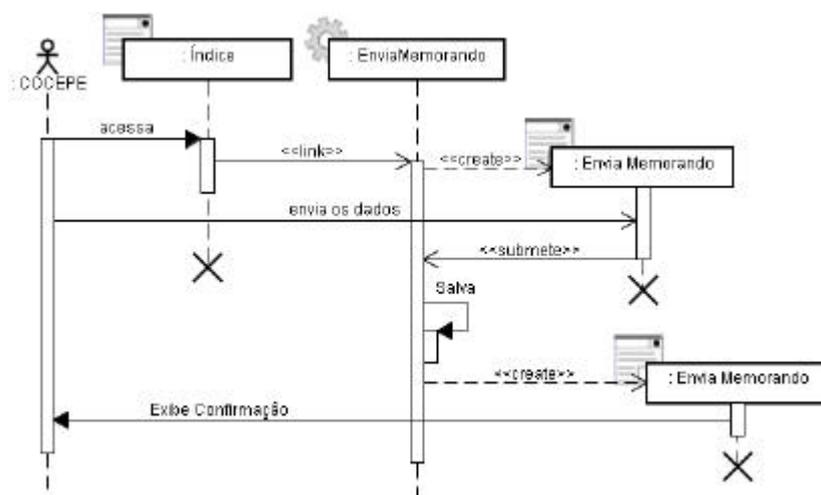


Figura 59 – Diagrama de Seqüência: Enviar Memorando

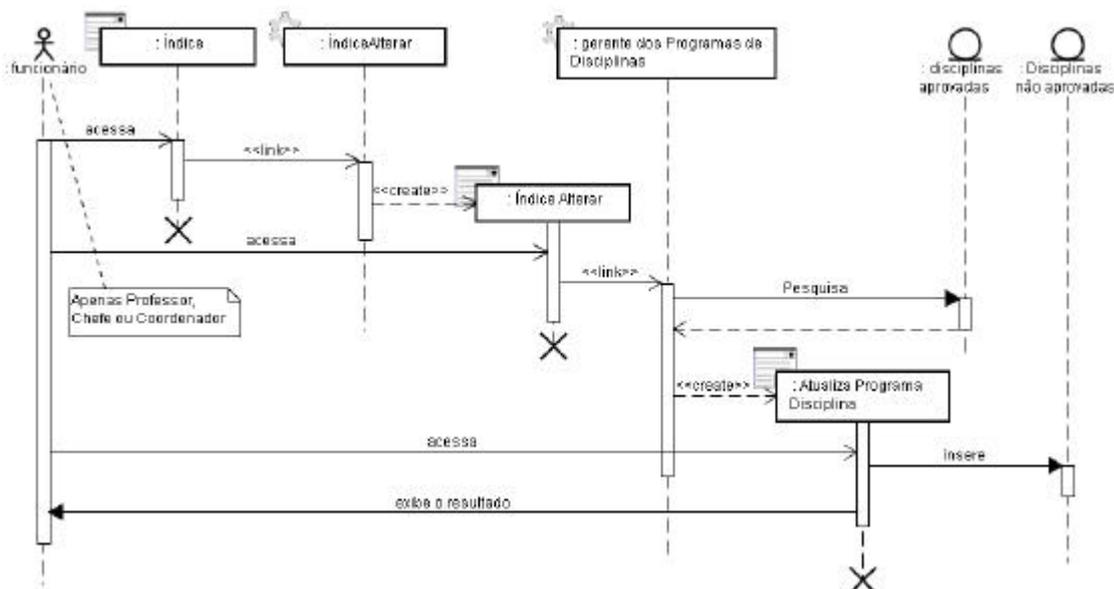


Figura 60 – Diagrama de Seqüência: Atualizar Programa de Disciplina

Após, definiram-se os Contratos das Operações (2.3.3.6) (Fig. 61, Fig. 62 e Fig. 63)

Contrato	
Nome:	Alterar Programa de Disciplina (protocolo, nívelAprovação, autor)
Responsabilidades:	Alterar um Programa de Disciplina que ainda não esteja Aprovado, ou o código de um Programa de Disciplina que já tenha sido aprovado pelo COCEPE.
Tipo:	Sistema
Referências	Funções: R1.2, R2.2
Cruzadas:	Casos de Uso: Incluir Programa de Disciplina, Aprovar Programa de Disciplina
Notas:	
Exceções:	
Saída:	Envia uma mensagem para o(s) usuário(s) responsável(is) pelo próximo nível de aprovação, caso tenha sido alterada pelo autor.
Pré-condições:	Há um programa de Disciplina que necessita alteração.
Pós-condições:	Se a Disciplina estiver reprovada, se todos os dados forem preenchidos e não houver problema com o conjunto código-semester-ano, e o usuário não possuir nível final de aprovação, um Programa de Disciplina foi alterado na base de dados das Disciplinas não Aprovadas; Se o usuário não possuir nível final de aprovação, uma mensagem é enviada ao nível seguinte (<i>criação de instância</i>); Se o usuário for um Chefe e a Disciplina não for de sua responsabilidade, um novo funcionário, lotado no Departamento do Chefe é o novo responsável pela Disciplina (<i>formada uma associação</i>); A mensagem é relacionada ao usuário (<i>formada uma associação</i>); Se todos os dados forem preenchidos e não houver problema com o conjunto código-semester-ano, e o usuário possuir nível final de aprovação, um novo Programa de Disciplina foi criado (<i>criação de instância</i>) Qualquer que seja o caso, é criada uma linha relatando a ação do usuário em Log das Disciplinas (<i>criada uma instância</i>)

Figura 61 – Contrato da Operação: Alterar Programa de Disciplina

Contrato	
Nome:	Enviar Memorando (arquivo, tipo)
Responsabilidades:	Alterar um modelo de memorando, de modo que o mesmo possa ser impresso toda vez que um Programa de Disciplina passe por um processo que exija a impressão deste documento
Tipo:	Sistema
Referências:	Funções: R1.1, R1.3, R1.5, R1.6
Cruzadas:	Casos de Uso: Enviar Memorando
Notas:	
Exceções:	
Saída:	Exibe uma mensagem informando o sucesso da operação.
Pré-condições:	O usuário tem um arquivo devidamente formatado, no formato .RTF, com as indicações necessárias para a interpretação do sistema.
Pós-condições:	O arquivo é salvo na base de dados para uso do sistema (<i>criação de uma instância</i>)

Figura 62 – Contrato da Operação: Enviar Memorando

Contrato	
Nome:	Atualizar Programa de Disciplina (usuario; discNAprovadas; discNAprovadasCurso (5 vezes, excetuando-se código, ano e semestre, que não são informados) (excetuando-se protocolo, que não é informado))
Responsabilidades:	Atualizar um Programa de Disciplina, inserindo um novo Programa de Disciplina na Base de Dados, dependente do nível hierárquico do usuário.
Tipo:	Sistema
Referências:	Funções do Sistema: R1.1, R1.2, R1.3, R1.6, R1.7, R1.8, R2.3
Cruzadas:	Casos de Uso: Alterar Programa de Disciplina, Incluir Programa de Disciplina
Notas:	
Exceções:	Se faltar algum dado a ser informado, retornar à tela com a mensagem de erro.
Saída:	Envia uma mensagem para o(s) usuário(s) responsável(is) pelo próximo nível de aprovação.
Pré-condições:	O usuário conhece todos os dados a informar e está autenticado pelo sistema.
Pós-condições:	Um Programa de Disciplina foi incluído na base de dados das Disciplinas não Aprovadas; Uma mensagem é enviada ao nível seguinte (<i>criação de instância</i>); A mensagem é relacionada ao usuário (<i>formada uma associação</i>); Qualquer que seja o caso, é criada uma linha relatando a ação do usuário em Log das Disciplinas (<i>criada uma instância</i>)

Figura 63 – Contrato da Operação: Atualizar Programa de Disciplina

Então, constatou-se que a operação Alterar Programa de Disciplina, abordada neste Ciclo de desenvolvimento, altera o estado do sistema da mesma forma que a operação Incluir, exceto quando o único campo alterado é o do responsável pela Disciplina. Sendo assim, não é necessário Definir-se um novo Diagrama de Estado (2.3.3.7) e sim, apenas utilizar o mesmo que está definido na Fig. 10.

Passa-se, a seguir, para a fase de Projetar (2.3.4), que inicia-se Definindo Casos de Uso Reais (2.3.4.1) que, para evitar repetir os mesmos descritos anteriormente nesta seção na íntegra, optou-se por exibir apenas seu nome e as alterações efetuadas, conforme se vê na Fig. 64, na Fig. 65 e na Fig. 66. As

Interfaces de Usuários (2.3.4.2), mencionadas nas figuras dos casos de uso, são exibidas na Fig. 67, Fig. 68, Fig. 69 e Fig. 70.

Caso de Uso:	Alterar Programa de Disciplina	
Tipo:	1. Secundário 2. Real	
Seqüência Típica de Eventos		
Ação do Ator		Resposta do Sistema
1. O usuário seleciona qual Disciplina deseja alterar, clicando no botão adequado da Janela do índice (Fig. 69).	2. O sistema consulta a base de dados dos Programas pelo código da Disciplina na base de dados dos Programas de Disciplinas não aprovados.	
	3. O sistema exibe um formulário com todos os campos preenchidos, para serem alterados (Fig. 70).	
4. O usuário altera os campos que desejar e seleciona a opção Submeter Alterações (Fig. 70).	5. O sistema salva os campos na base de dados dos Programas.	
Seqüências alternativas		
Linha 3: o Ator não é um Chefe e não possui permissão de alterar o Programa. O sistema exibe a mensagem de que o Programa de Disciplina não pode ser alterado por este usuário.		
Linha 3: o Ator é um Chefe e não possui permissão de alterar o programa. O sistema exibe uma página formatada Web com a Disciplina preenchida e um espaço para ser preenchido com o código do novo responsável pela inclusão da Disciplina.		
Linha 5: o Ator é um Professor, responsável pela Disciplina. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Professor.		
Linha 5: o Ator é um Chefe. O Programa de Disciplina é salvo na base de dados com o nível de aprovação Chefe.		
Linha 5: o Ator é um Chefe e não possui permissão de alterar o programa. As alterações relativas ao funcionário responsável pela Disciplina são salvos na base de dados.		
Linha 5: o Ator é um DRA. As alterações são salvas na base de dados definitiva e a Disciplina é apagada da base de dados de aprovação		

Figura 64 – Caso de Uso Real: Alterar Programa de Disciplina

Caso de Uso:	Enviar Memorando	
Tipo:	1. Secundário 2. Real	
Seqüência Típica de Eventos		
Ação do Ator		Resposta do Sistema
1. O COCEPE escreve o modelo, baseado nas informações dadas pelo sistema, e o salva em seu computador.	3. O sistema exibe uma página formatada Web com a solicitação de envio de arquivo (Fig. 68).	
2. O COCEPE acessa o sistema e escolhe a opção de Alterar Modelo de Memorando (Fig. 67).	5. O sistema exibe uma página formatada Web informando que a operação foi bem sucedida.	
4. O COCEPE indica o arquivo contendo o novo modelo de memorando, bem como qual a função deste memorando (Aprovação, Reprovação, etc), clicando no botão adequado, enviando o arquivo (Fig. 68).		

Figura 65 – Caso de Uso Real: Enviar Memorando

Caso de Uso:	Atualizar Programa de Disciplina	
Tipo:	1. Secundário 2. Real	
Seqüência Típica de Eventos		
Ação do Ator		Resposta do Sistema
1. O funcionário digita o código da Disciplina e seleciona a opção Atualizar Programa de Disciplina (Fig. 69).	2. O sistema exibe um página formatada <i>Web</i> com o Programa de Disciplina escolhido, com alguns campos passíveis de alteração, devidamente preenchidos com as informações anteriores, semelhante ao visto na Fig. 35 além de campos para preencher com códigos de memorandos,.	
3. O funcionário reescreve as informações que desejar, escreve os códigos dos memorandos e seleciona Inserir.	4. O sistema salva as alterações na base de dados das disciplinas não aprovadas e exibe os memorandos a serem impressos e assinados.	

Figura 66 – Caso de Uso Real: Atualizar Programa de Disciplina

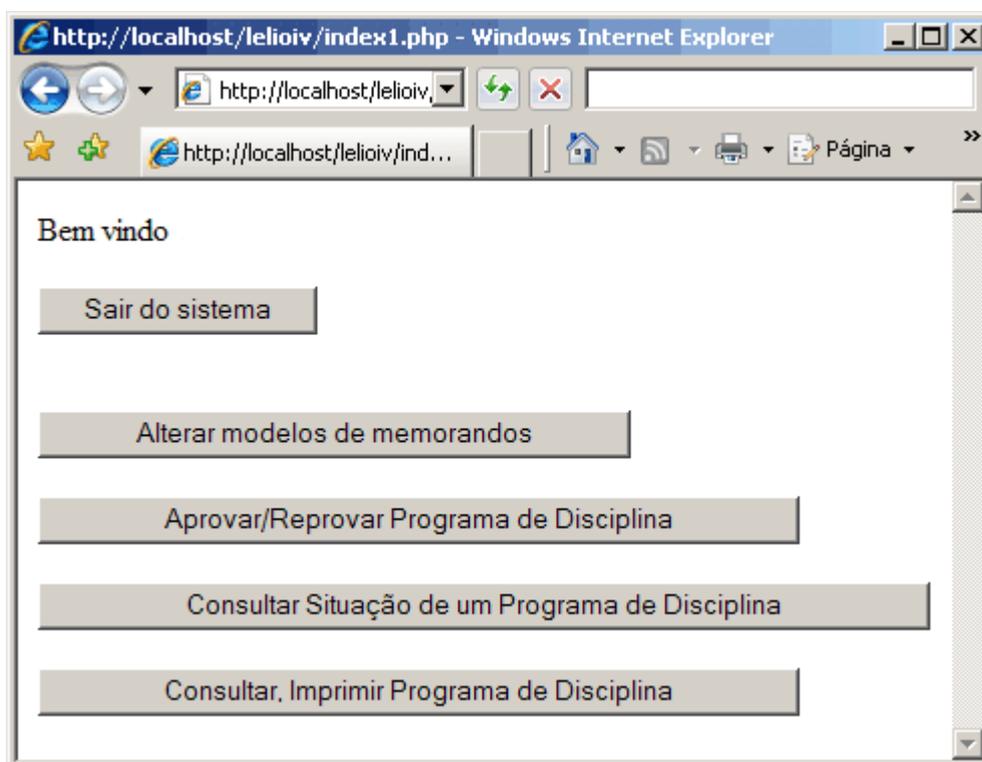


Figura 67 – Tela de Índice Atualizada

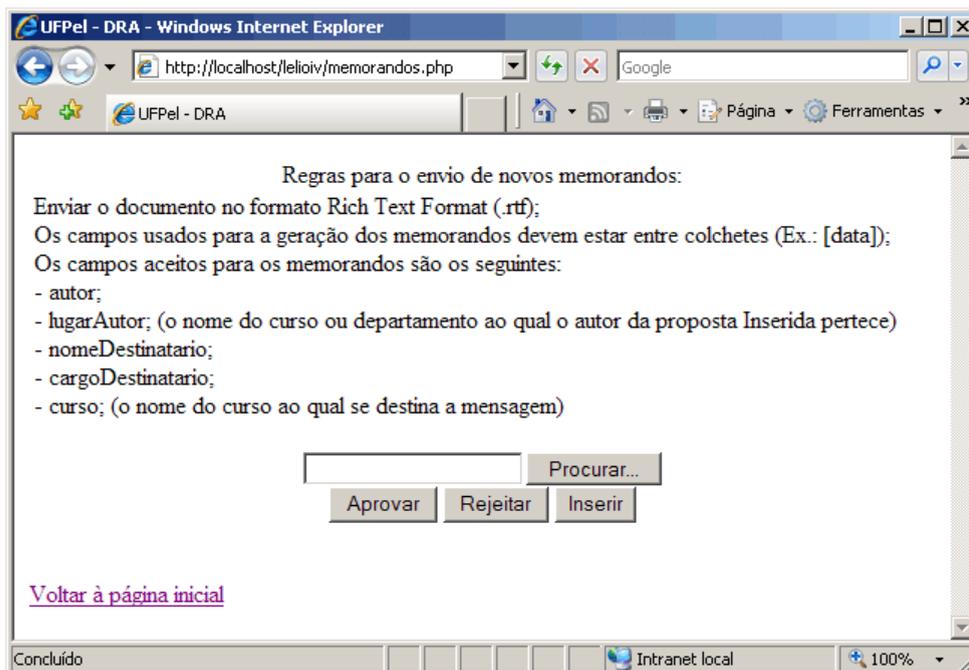


Figura 68 – Janela de inserir memorando

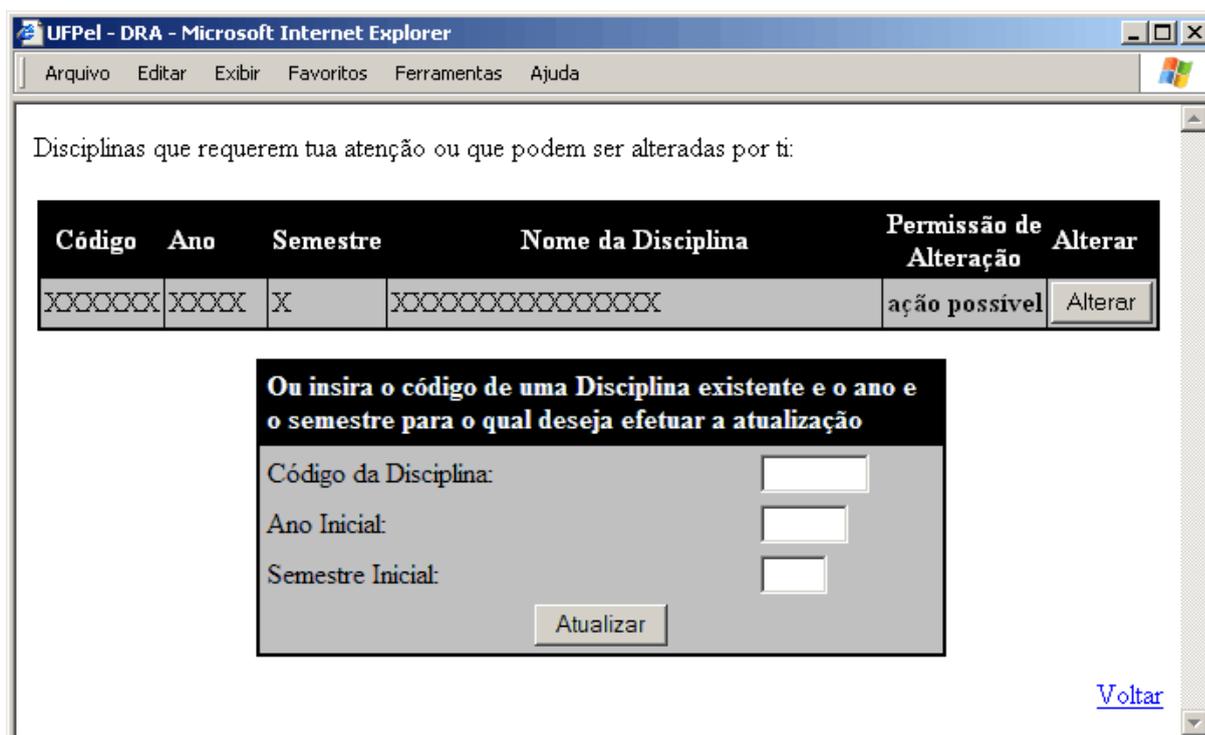


Figura 69 – Janela do índice dos Programas para alteração

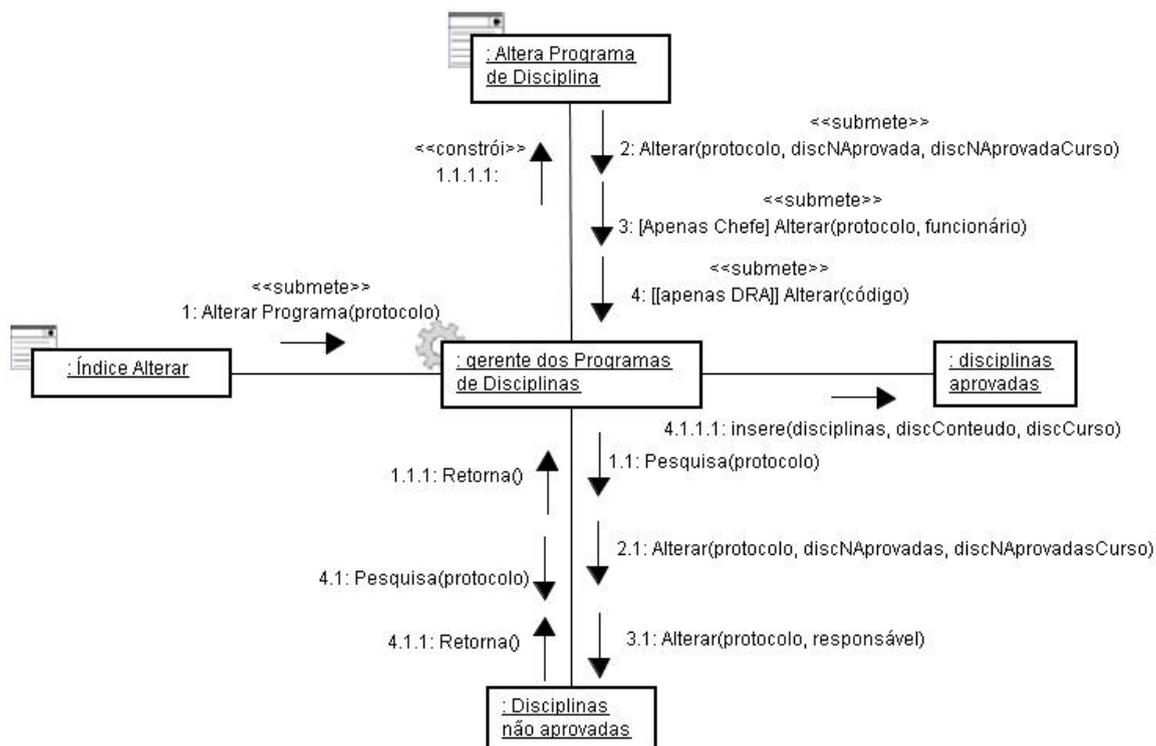


Figura 72 – Diagrama de Comunicação: Alterar Programa de Disciplina

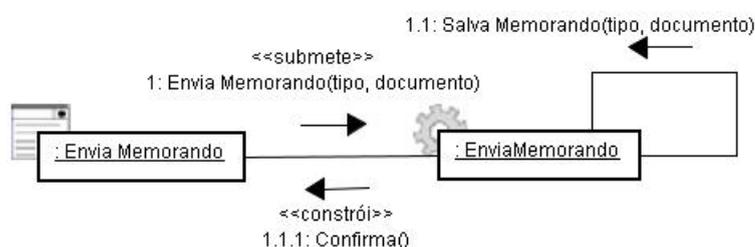


Figura 73 – Diagrama de Comunicação: Enviar Memorando

Devido às alterações no componente gerente dos Programas de Disciplina, foi necessária uma nova Definição de Diagramas de Classes (2.3.4.4), e as classes exibidas na Fig. 74 foram criadas ou alteradas. Para evitar repetição excessiva, exibe-se apenas as informações extras ou interessantes, suprimindo-se as demais, tais como as diversas «Páginas do Servidor» que fazem parte do pacote Gerenciamento de Disciplinas, ou os demais Pacotes do sistema, que são, também, relacionados com os itens exibidos aqui.

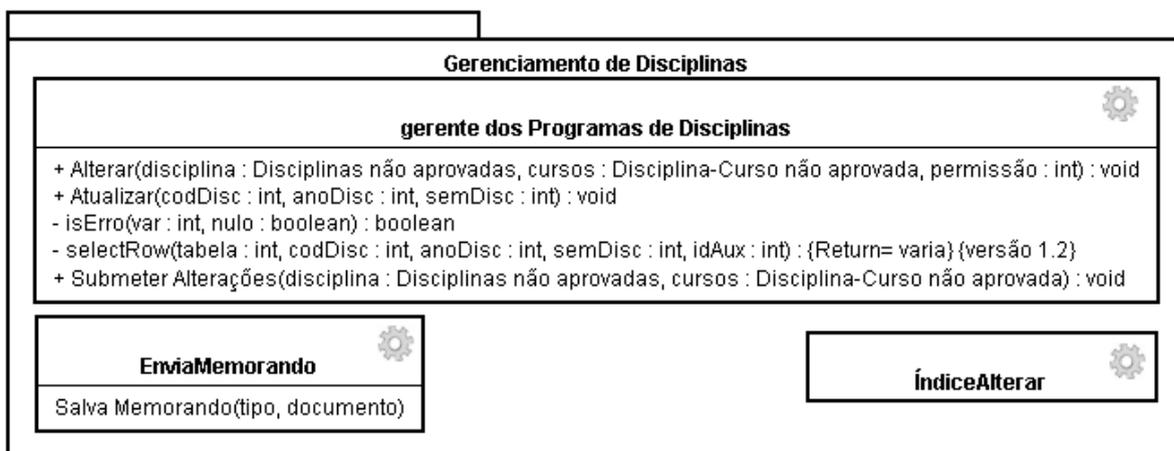


Figura 74 – Definição do Diagrama de Classes para o 3º Ciclo

A Fig. 75 exibe apenas os itens que foram alterados a partir da arquitetura prevista para o 1º ciclo de desenvolvimento, incluindo a alteração que houve na relação entre as páginas do cliente Pesquisa e Altera Programa de Disciplina, ao se Refinar a Arquitetura do Sistema (2.3.4.5).

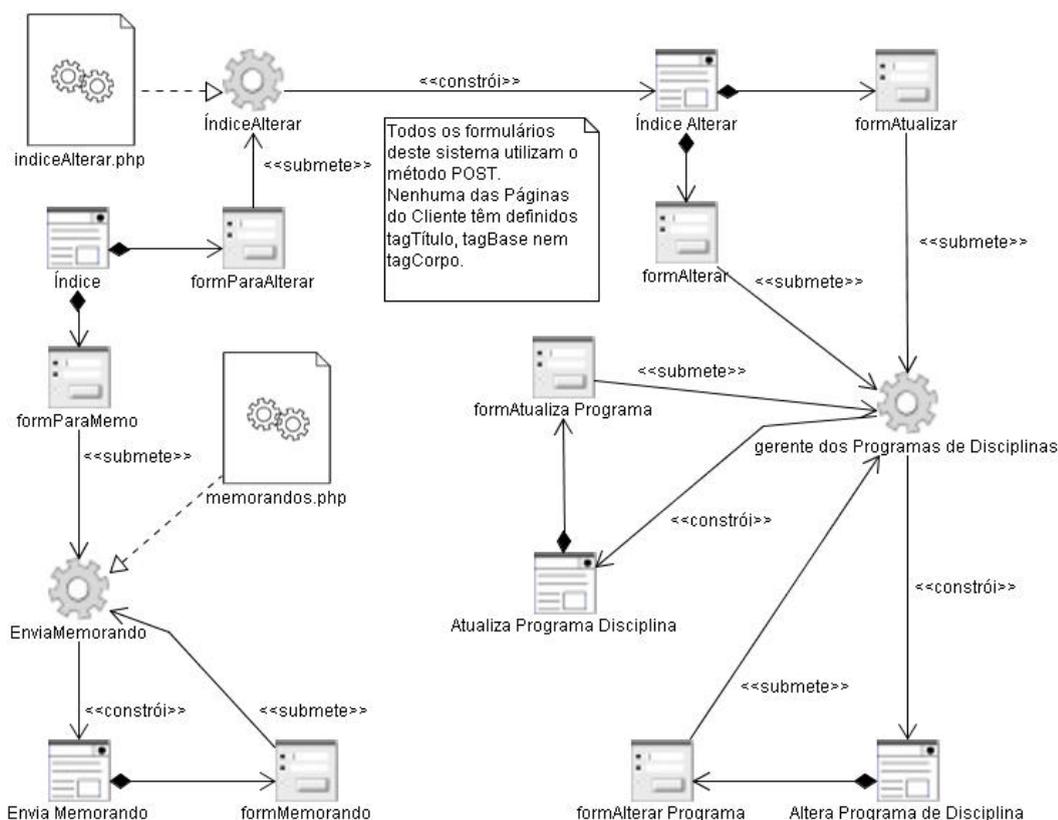


Figura 75 – Refinamento do Diagrama de Artefatos Web e Navegação, para o 3o Ciclo

Após, o modelo Programado (2.3.4) foi Testado (2.3.5) e salvo como Ciclo de Desenvolvimento 3.zip, contendo os arquivos indice2.php, index1.php,

gerenciaProgramaDisciplina.php, consultaSituacao.php, imprimePrograma.php, indiceAprovar.php, indiceAlterar.php e memorandos.php.