

ERIKA UTIAMA

**MODELAGEM DE UMA FERRAMENTA DE INTERAÇÃO PARA O
ENSINO A DISTÂNCIA**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências para a obtenção do título de Bacharel em Ciência da Computação.

Orientador

Prof. André Luiz Zambalde

LAVRAS
MINAS GERAIS - BRASIL
2003

ERIKA UTIAMA

**MODELAGEM DE UMA FERRAMENTA DE INTERAÇÃO PARA
ENSINO A DISTÂNCIA**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências para a obtenção do título de Bacharel em Ciência da Computação.

APROVADA em ___ de _____ de 200_

Prof. Rêmulo Maia Alves

Prof. André Luiz Zambalde
UFLA
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL

DEDICATÓRIA

Dedico esta monografia
aos meus queridos pais,
Kikue Utiana e
Yochiaki Utiana

AGRADECIMENTOS

Primeiramente minha infinita gratidão a Deus, por tudo que me proporcionou até a presente data. Me dá forças para continuar a enfrentar e superar os obstáculos com os quais me deparo diariamente.

Agradeço eternamente à minha querida família que é e sempre será meu grande porto seguro, que durante toda minha vida me apóia e incentiva, principalmente no decorrer destes anos acadêmicos.

Ao meu orientador, o professor André Luiz Zambalde, por sua disponibilidade e atenção em auxiliar-me durante o projeto, esclarecendo dúvidas, direcionando o trabalho, enriquecendo meus conhecimentos e dando respaldo ao meu crescimento profissional.

Em especial, ao caro colega Roosevelt Mairink dos Santos Júnior, por seus esclarecimentos e sua prontidão; seu auxílio foi de vital importância.

Aos queridos amigos e colegas do curso de Ciência da Computação (formandos e os demais) pelas alegrias e experiências dos momentos convvidos, nesta jornada de quatro anos, sempre serão lembrados com carinho; especialmente os inesquecíveis: Ricardo, Adriana, Carmen, Flávia, Paulo, Muzamba, Geovane, Júlio Teodoro, Alexandre, Flávio Alves, Matheus ...

À minha família postiça, o apartamento 303, tanto as moradoras antigas quanto as atuais, pelos momentos de amizade, alegrias, conversas e dificuldades.

Aos amigos e pessoas que mesmo distantes se preocuparam comigo e sempre torceram por meu sucesso.

A todos àqueles que de alguma forma contribuíram para a concretização desta monografia.

RESUMO

Modelagem de uma ferramenta de interação para Ensino a Distância

A proposta deste trabalho é apresentar a modelagem de uma ferramenta de Ensino a Distância, que sirva como documentação e através da análise dos diagramas e modelos obtidos, consiga-se compreender o sistema como um todo e possibilite a tomada de decisões mais precisas e em menor tempo, caso seja necessário efetuar alterações nesta aplicação.

Palavras-chave: modelagem, documentação, diagramas e modelos.

ABSTRACT

Modelling of an interaction tool for Teaching the Distance

The proposal of this work is to present the modelling of a tool of Distance Education, that serves as documentation and through the analysis of the diagrams and obtained models, be gotten to understand the system as a whole and make possible the of more exacts decisions and in smaller time, in case it is necessary to make alterations in this application.

Keywords: modelling, documentation, diagrams and models.

Orientador: André Luiz Zambalde - DCC - UFLA.

ÍNDICE

1. Introdução.....	1
1.1. Objetivos.....	4
1.2. Justificativa.....	5
1.3. Organização do texto	5
2. Referencial Teórico.....	7
2.1. Ensino a Distância (EAD)	7
2.1.1. Histórico do Ensino a Distância.....	7
2.1.2. Vantagens e desvantagens do EAD	8
2.2. Modelos de EAD.....	9
2.2.1. O ambiente EAD.....	11
2.2.3. Ferramentas Web usadas no EAD	12
2.3 Desenvolvimento de software para EAD	13
2.3.1. Desenvolvimento de interfaces para ambientes EAD via Web	13
2.3.3. Por que modelar?	14
2.3.2. Engenharia Reversa	15
2.3.2. Arquitetura para desenvolvimento de aplicações Web em três camada	16
2.3.4. Casos de uso	18
2.3.5. OOHDM	20
2.3.5.1. Hipertexto e hipermídia: conceitos básicos:	20
2.3.5.2 A modelagem conceitual em OOHDM	23
2.3.5.3. Projeto de Navegação.....	24
2.3.5.4. Interface Abstrata	24
2.3.5.5. Implementação.....	25
2.3.6. DFD	25
2.3.6.1. Componentes de um DFD.....	26
2.3.6.2. DFD com níveis.....	28
2.3.7. Modelagem de dados	28
2.7.2. Modelo ER (Entidade-Relacionamento)	30
2.7.3. Modelo Relacional.....	31
3. Metodologia.....	33
4. Resultados e Discussão.....	35
4.1. Visão geral do ambiente de ensino	35
4.2. A modelagem do sistema.....	37
4.2.1. Camada de usuário.....	37
4.2.2. Camada de regra de negócio	41

4.2.3. Camada de dados	42
5. Conclusões e propostas para trabalhos futuros	50
6. Bibliografia.....	52
ANEXO A: Diagramas de <i>use-cases</i>	56
ANEXO B: Descrição dos <i>use-case</i>	61
ANEXO C: Modelagem OOADM.....	94
ANEXO D: Diagramas de fluxo de dado (DFD)	103

LISTA DE FIGURAS

Figura 01: Modelo <i>broadcast</i>	9
Figura 02: Versão virtual da sala de aula tradicional.....	10
Figura 03: Modelo ambiente virtual.....	10
Figura 04: Processo de Engenharia Progressiva X Engenharia Reversa.....	16
Figura 05 : Arquitetura cliente-servidor em aplicações web em três camadas ...	18
Figura 6.1: Ator.....	19
Figura 6.2: Caso de Uso.....	19
Figura 07: Atividades do OOADM.....	23
Figura 08: Simbologia básica de um DFD	26
Figura 09: Visão geral dos projetos de um banco de dados	30
Figura 10: Visão geral do processo de mapeamento entre os modelos ER Relacional	32
Figura 11: Diagrama de <i>use-case</i> de cursos do administrador.....	38
Figura 12: Modelagem navegacional da página principal.....	39
Figura 13: Modelagem navegacional coordenação_cursos	40
Figura 14: DFD da autenticação do administrador	41
Figura 15: DFD de cadastro de curso	42
Figura 16: Modelo E-R	49

LISTA DE TABELAS

Tabela 01: Histórico do EAD no Brasil.....	8
Tabela 02: Administration do banco de dados	43
Tabela 03: Cursos do banco de dados.....	43
Tabela 04: Secretárias do banco de dados	44
Tabela 05: Usuários do banco de dados.....	44
Tabela 06: Bibliolinks do banco de dados.....	45
Tabela 07: Turmas do banco de dados.....	45
Tabela 08: Alunos_turmas do banco de dados.....	45
Tabela 09: Fóruns do banco de dados.....	46
Tabela 10: Fóruns_disciplinas do banco de dados	46
Tabela 11: Fóruns_respostas do banco de dados	46
Tabela 12: Bibliolinks do banco de dados.....	47
Tabela 13: Notas do banco de dados.....	47
Tabela 14: Chat_users do banco de dados.....	47
Tabela 15: Chat_messages do banco de dados.....	48

1. Introdução

Atualmente, com o advento da Internet vem-se alterando alguns conceitos de difusão e de gestão da informação. A Web caminha rapidamente para se tornar o grande repositório que armazenará todo tipo de informação que for tornada pública. Permitindo não só que se agilize o processo de acesso a documentos textuais, mas abrangendo gráficos, imagens, fotografias, sons e vídeo. Dada essa facilidade, surge a sociedade do conhecimento, na qual obter informação não tem sido mais o problema principal, mas sim como geri-la. Através dos inúmeros recursos oferecidos pela Web, a rotina de vida humana vem sendo modificada. A maioria das pessoas se conecta a rede para desfrutar da facilidade de se comunicar com outros indivíduos no mundo todo, para formar grupos de trabalho ou para se divertir.

A configuração do novo cenário de popularização do uso do computador e da Internet no nosso cotidiano, tem refletido na educação, em particular, no ensino a distância (EAD) e na sua forma de atuar.

"O ensino a distância via Internet vence as barreiras do espaço e do tempo e proporciona o acesso às informações mais distantes e com uma rapidez incrível. Constitui também uma forma concreta da democratização do saber, permitindo atingir um contingente da população que se encontra excluído da forma tradicional de ensino pela dificuldade de acesso ou disponibilidade de tempo." (Carvalho, 2002)

O Ambiente do EAD é o local virtual onde o aluno pode encontrar o conteúdo a ser assimilado, além de propiciar a interação entre ele e outros alunos e/ou professores. Além disso, o pessoal administrativo do curso deve ser capaz de controlar aspectos como avaliações, acessos ao ambiente, matrículas entre outros. (Carvalho, 2002)

A fim de que o suporte ao aprendizado em EAD seja positivo, faz-se necessário uma comunicação bilateral entre alunos-professores e entre alunos-alunos, tanto síncrona quanto assincronamente, baseada em tecnologias Web, como: fórum de discussão, *e-mail*, *chat*, vídeoconferência, lista de discussão. Estas permitem que se crie um elevado grau de motivação no estudante e estimule-o à constante e intensa troca e cooperação de todos com todos, possibilitando assim, uma aprendizagem colaborativa.

O objetivo deste trabalho é apresentar a análise e modelagem da ferramenta de interação para Ensino a Distância utilizado em cursos de *Lato Sensu* da Universidade Federal de Lavras, apresentando sua estrutura e proporcionando uma visão do funcionamento interno, através dos modelos a serem gerados.

Segundo Rumbaugh(1994), modelo é uma visão simplificada de um sistema. Ele mostra os elementos essenciais do sistema de uma perspectiva específica e oculta os detalhes não essenciais, sendo utilizado para: auxiliar na compreensão de sistemas complexos, formar uma base para implementação, além de comunicar decisões sem ambigüidade.

A modelagem de sistemas pode ser justificada com base em dados obtidos em pesquisas de algumas empresas de desenvolvimento que gastam aproximadamente 70% de suas atividades, normalmente assim distribuídas:

- 12% para manutenção corretiva para eliminar erros flagrantes;
- 9% para manutenção preventiva para eliminar erros potenciais;
- 49% para manutenção adaptativa, ou seja, incorporação de novos desejos de usuários e ajustes pela dinâmica do sistema. (Barbieri, 1994)

Atualmente, como estão sendo desenvolvidas aplicações cada vez mais complexas, e a maior parte delas não apresenta nenhuma documentação, incrementa-se consideravelmente os números desta pesquisa; revelando assim, o grande gargalo na área de desenvolvimento de *softwares*, que se encontra

exatamente no tempo, no esforço e no dinheiro despendidos na manutenção de *softwares* ao invés do próprio desenvolvimento.

Devido ao fato de que muitos dos sistemas desenvolvidos é de autoria de programadores que possuem prazos reduzidos para dedicar-se às modelagens, poderia-se pensar que o problema estaria solucionado, caso houvesse tempo hábil para se trabalhar sobre requisitos dos clientes e posterior modelagem do sistema: isto seria realmente bastante proveitoso a partir do momento que essa realidade fosse concretizada. Contudo, como resolver a questão dos *softwares* já desenvolvidos a certo tempo e que não possuem documentação ?

Neste contexto, pode-se visualizar uma área em ascendente crescimento, a Engenharia Reversa, que consiste em seguir o fluxo contrário ao dos princípios da Engenharia Progressiva, isto é, parte-se de um sistema pronto para tentar chegar-se ao seu nível de abstração máximo. Isto significa que tenta-se compreender o código-fonte de um sistema desenvolvido a partir da observação do funcionamento da interface, da resposta do sistema a determinadas interações com a interface e se possível entrevistas com os próprios desenvolvedores. Só após se esgotarem todas as possibilidades é que se parte para a etapa de estudo do código-fonte. A partir desse momento, o analista deverá documentar tudo que achar relevante e passar a utilizar diversas modelagens e diagramas que conseguir melhor se adaptar ao tipo de programa desenvolvido ou da plataforma que é disponibilizada.

A Engenharia Reversa possui duas categorias que a abrangem: a redocumentação e a recuperação de *software*. A redocumentação busca ter uma visão geral de funcionamento do sistema, enquanto a recuperação visa entender o sistema como um todo, tentando descobrir o que e como ele faz as coisas. A Engenharia Reversa pode possibilitar inúmeras alternativas para sistemas que estão totalmente fora dos escopo do cliente.

Com a modelagem, a partir da Engenharia Reversa, consegue-se respaldo suficiente para se compreender as funcionalidades do sistema de uma forma que não gere ambigüidade, além de se tomar decisões mais rápidas e acertadas quanto à alterações. Além disto, numa empresa de desenvolvimento, serve como histórico no caso de algum *software* que seja desenvolvido posteriormente e apresente semelhanças, podendo haver um aproveitamento e algumas adaptações da modelagem de um sistema já projetado, permitindo que não se parta do ponto inicial.

Modelar é preciso! Esquecer ou omiti-la é sinônimo de prejuízo financeiro, perda de tempo, retrabalho e situações bastante indesejáveis.

Ao término do projeto pretende-se recuperar informações que auxiliem o entendimento do sistema desenvolvido e agilizem a manutenção deste, além de resultar numa documentação que poderá ser freqüentemente consultada e alterada se necessário; tornando-se um "guia" do *software* modelado reversamente.

1.1. Objetivos

- Modelar o sistema de EAD utilizado nos cursos de Pós-Graduação *Lato Sensu* da UFLA, a fim de se ter uma real visão da estrutura e das funcionalidades presentes no mesmo;
- Proporcionar o entendimento do código-fonte do sistema para que se evite perda de tempo no estudo do mesmo, além de permitir que um programador altere com mais facilidade e precisão partes ou o sistema inteiro;
- Gerar documentação da modelagem do sistema desenvolvido, de tal forma, que possa sanar as exigências dos itens anteriores;
- Ao término da modelagem, verificar possíveis erros de implementação, falta de alguma funcionalidade e ainda sugerir possíveis alterações que venham acrescentar aspectos positivos ao sistema;

- Estudar e avaliar criteriosamente, as características do EAD atualmente, bem como, seus pontos positivos e negativos, a adaptação do professor e do aluno no ambiente educacional e o impacto ocasionado na sociedade com a disseminação deste recurso instrucional.

1.2. Justificativa

A modelagem do sistema de interação para auxílio ao Ensino a Distância proposta neste trabalho, refere-se ao sistema desenvolvido por Roosevelt Mairink dos Santos Júnior, que foi disponibilizado nas *homepages* dos cursos de Pós-Graduação *Lato Sensu*: Informática em Educação (IED) e Administração em Sistemas de Informação (ASI), cujas sedes administrativas situam-se na Universidade Federal de Lavras, em Minas Gerais. Pretende-se que a modelagem desenvolvida sirva como documentação para que os usuários do sistema tenham uma visão de sua estruturação e funcionamento, além de agilizar o entendimento do sistema por desenvolvedores, auxiliando em possíveis manutenções do mesmo.

1.3. Organização do texto

Este trabalho está estruturado em seis capítulos. No Capítulo 1, faz-se a introdução do problema a ser desenvolvido ao longo dos demais capítulos e também são descritos os objetivos referentes ao assunto abordado. No Capítulo 2, apresenta-se uma cuidadosa revisão bibliográfica sobre o Ensino a Distância, sua evolução histórica, suas vantagens e desvantagens, o ensino *on-line*, ainda aborda aspectos relativos às ferramentas empregadas no desenvolvimento de ambientes e recursos tecnológicos apropriados ao EAD, além de dar grande enfoque na definição, estrutura e funcionamento de todos os modelos utilizados na modelagem do sistema. No Capítulo 3, enfoca-se a metodologia aplicada

no planejamento das ações que convergirão para a solução do problema. No Capítulo 4, apresenta-se os resultados e as discussões obtidos, através da modelagem da ferramenta de interação para o Ensino a Distância proposta. No Capítulo 5, são descritas conclusões e considerações finais a respeito do trabalho. No Capítulo 6, são enumeradas as referências bibliográficas que serviram de suporte ao desenvolvimento deste texto.

2. Referencial Teórico

2.1. Ensino a Distância (EAD)

Afinal, o que vem a ser exatamente EAD ?

"Considera-se ensino a distância (EAD) qualquer forma de execução de atividades de ensino sem que haja a proximidade física constante entre professores e alunos. A comunicação bidirecional é realizada por meio de algum recurso tecnológico intermediário, como: textos, impressos, telefone, fax, televisão, radiodifusão, fitas, CD-ROM e/ou ambientes computacionais." (Zambalde, 2001)

"Educação a Distância é uma forma de ensino que possibilita a auto-aprendizagem, com a mediação de recursos didáticos sistematicamente organizados, apresentados em diferentes suportes de informação, utilizados isoladamente ou combinados, e veiculados pelos diversos meios de comunicação." (LDBE, 1996)

2.1.1. Histórico do Ensino a Distância

A evolução histórica do Ensino a Distância no Brasil pode ser dividida em três gerações, conforme a Tabela 01:

Geração	Início (aproximado)	Característica	Recursos disponíveis
1ª Geração	1940	Ensino por correspondência	Material impresso: cartas e livros
2ª Geração	1970	Meios de comunicação de massa	Rádio, televisão, vídeo-aulas, audiocassetes, sistemas de telefonia
3ª Geração	1990	Uso de computadores	<i>Softwares</i> educativos, Internet, vídeo-conferência

Tabela 01: Histórico do EAD no Brasil

Fonte: (Santos Júnior, 2002)

2.1.2. Vantagens e desvantagens do EAD

Segundo Arnaldo Santos, citado por Santos Junior(2002), existem algumas vantagens e desvantagens relevantes do EAD que devem ser observadas.

As principais vantagens defendidas são relativas a disponibilidade e ritmos de estudos diferenciados, através da eliminação de barreiras de tempo e espaço, o que fomenta a aquisição autônoma e contínua por novos conhecimentos, além de estimular trabalho em grupo e incentivar a experiência no uso de novas Tecnologias de Informação e Comunicação (TIC's).

Em contrapartida, as desvantagens referem-se à falta de convívio social proporcionado pelo ensino presencial, de conhecimento tecnológico prévio e de equipes de suporte multidisciplinares, o que pode ocasionar falta de motivação e determinação; além de exigir elevados investimentos iniciais e abranger apenas cursos de escopo genérico e de menor componente prático,

sugestionando assim, a falta de credibilidade por parte de alguns setores da sociedade.

Portanto, o EAD só pode ser visto como vantajoso se os argumentos de flexibilidade, acessibilidade, racionalização de recursos, interatividade e outros elementos supra mencionados, permitirem obter iguais ou melhores resultados pedagógicos, comparados com a formação tradicional.

2.2. Modelos de EAD

De acordo com Valente(2000), existem três modelos de EAD, via Internet: o broadcast, a versão virtual da sala de aula tradicional e o ambiente virtual de construção de conhecimento.

No modelo *broadcast* a plataforma disponibiliza a informação para o aprendiz, via telemática não existe a interação entre aprendiz e professor. Dá-se ênfase em materiais instrucionais e devem ser utilizados bons recursos de interatividade para permitir a transmissão contida no material.

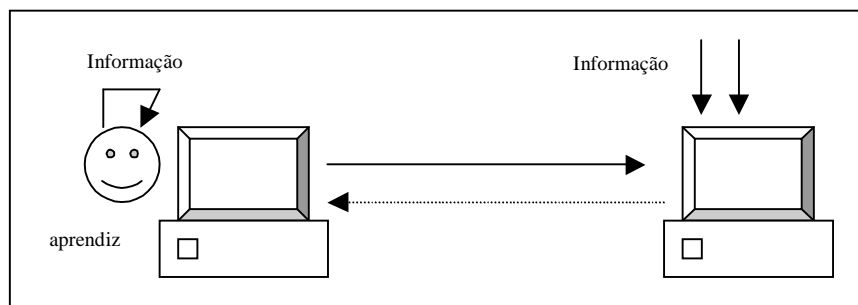


Figura 01: Modelo *broadcast*

Fonte: (Valente, 2000)

A versão virtual da sala de aula tradicional é um modelo baseado no sistema tradicional de ensino, centrado no professor e há pouca interação do professor com o aprendiz, podendo ser realizada via *e-mail* ou lista de discussão.

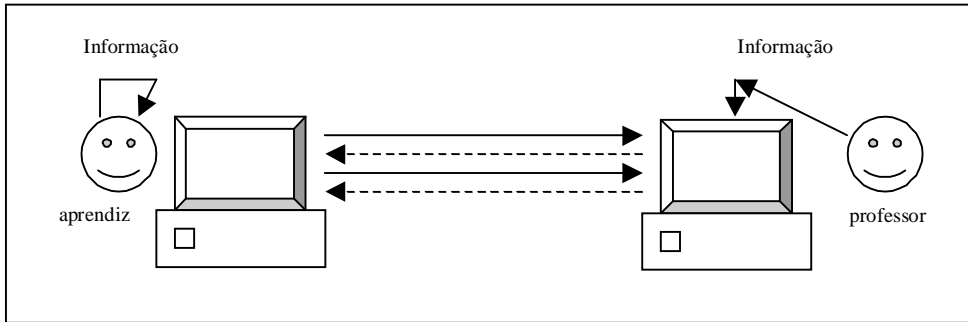


Figura 02: Versão virtual da sala de aula tradicional

Fonte: (Valente, 2000)

No modelo de ambiente virtual para construção de conhecimento existe a combinação de atividades presenciais e via plataforma. O aprendiz resolve as atividades em seu contexto e recebe suporte via plataforma, simulando a situação de "estar junto" virtualmente. A aprendizagem é gerada através da construção pedagógica do conhecimento, no qual toda ação realizada pelo aluno reflete num resultado. O *design* participativo das ferramentas disponibilizadas deve envolver os usuários.

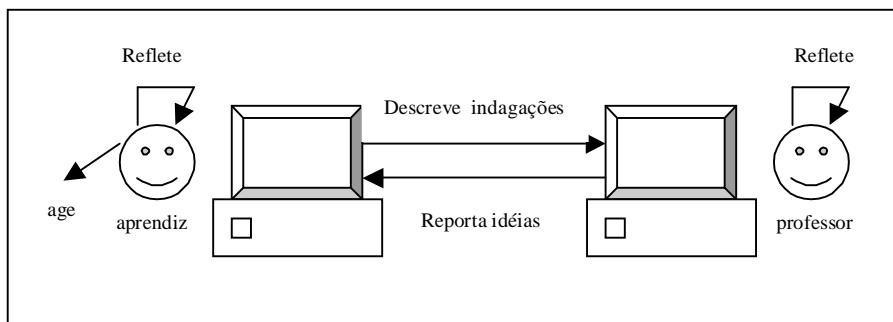


Figura 03: Modelo ambiente virtual

Fonte: (Valente, 2000)

2.2.1. O ambiente EAD

O software de suporte às atividades realizadas em um ambiente *on-line* leva os agentes do processo educacional a interagirem de uma forma colaborativa em um espaço comum. Tais espaços de colaboração, suportados por software, são conhecidos como ambientes virtuais de aprendizagem ou "Learning Management Systems (LSM)". (Hall, 2000) As ferramentas que compõem esses ambientes virtuais de aprendizagem estão organizadas em três grupos básicos: autoria, administração e uso dos alunos. Fundamentalmente, o que diferencia os diversos ambientes virtuais de aprendizagem existentes, são as funcionalidades presentes em cada grupo que compõe o ambiente. No conjunto de autoria tem-se um número grande de ferramentas para edição e inclusão de textos em diversos formatos (HTML, PDF, etc.), slides ou transparências, áudio, vídeo e animações. Além disso, elas também possibilitam ao professor definir as cores, o padrão das páginas e quais recursos de comunicação poderão ser usados durante o curso. Em todos os ambientes, as ferramentas de autoria estão disponíveis apenas para o professor, e para seus auxiliares. O grupo referente a administração inclui ferramentas que facilitam o gerenciamento do curso e fornecem certas informações a respeito do mesmo para o professor. Com isso, ele pode controlar o período de inscrição no curso, verificar todos os grupos de discussão criados, acessar alguns gráficos estatísticos gerados a partir de dados de arquivos de *log*, etc. Algumas ferramentas permitem ao professor acompanhar o acesso dos alunos ao curso, suas notas e alguns outros relatórios que, em tese, devem auxiliá-lo a acompanhar o desenvolvimento de seus alunos. O conjunto de recursos disponíveis para os alunos inclui ferramentas para comunicação, avaliação automática, pesquisa em glossários, anotações, criação de páginas pessoais e acompanhamento de notas.

2.2.3. Ferramentas Web usadas no EAD

Para que o suporte ao aprendizado em EAD seja positivo, faz-se necessário a construção de ambientes cada vez mais interativos e de fácil manuseio. Deve-se suprir as necessidades dos usuários com ferramentas de fácil utilização, amplamente disponíveis que garantam a qualidade na aprendizagem. Tais como as ferramentas síncronas de comunicação que exigem a participação dos estudantes e professores em eventos marcados, com horários específicos, para que ocorram, por exemplo, *chats*, videoconferências ou audioconferências através da Internet; e ferramentas assíncronas, que independem de tempo e lugar, como listas de discussão por correio eletrônico, fórum de discussão e as trocas de trabalhos através da rede. (Oliveira, 2001)

Através de interações **síncronas** ou **assíncronas** utilizando a Internet como dispositivo de mediação entre os vários intervenientes espalhados geograficamente, existe a possibilidade de aulas colaborativas, fazendo surgir assim as **Comunidades Virtuais** com proliferação de **escolas virtuais, universidades virtuais, institutos virtuais, turmas virtuais** com cursos e conteúdos acessíveis via *World Wide Web (WWW)*, onde existe a possibilidade de cada estudante interagir com o professor e com as bases de conhecimento computadorizadas, assim como é possível interagir também com outros estudantes, possibilitando uma aprendizagem colaborativa. Esta abordagem colaborativa em comunidades virtuais oferece o modelo mais adequado à educação *on-line*, visto que, pode envolver o estudante, mantendo sua motivação e estimulando-o a constante e intensa troca e cooperação de todos com todos. Portanto: "aprender é a meta e motivação é a base".

2.3 Desenvolvimento de software para EAD

2.3.1. Desenvolvimento de interfaces para ambientes EAD via Web

Andrade(2002), afirma que, utilizar a Web como uma tecnologia habilitada a apoiar um programa de ensino não é uma tarefa muito fácil, por isso, requer uma organização, de modo a possibilitar não só a apresentação dos conteúdos como também a avaliação dos esforços individuais dos estudantes, e a obtenção de resultados que permitam a avaliação da classe como um todo. Deste modo, o projeto e desenvolvimento de sistema Web para EAD deve ser sistemático, (constituindo dos estágios habituais de projeto, desenvolvimento, evolução e revisão) e permitir interatividade entre professores, alunos e o ambiente de aprendizado. Conseguir projetar interfaces de boa qualidade é um processo que envolve inúmeras variáveis, visto que as necessidades e a diversidade de público-alvo diferem de aplicação para aplicação.

Gui Bonsiepe(1997), citado por Andrade(2002), afirma que a interface é um meio que: pode frustrar e irritar; facilitar ou dificultar a aprendizagem; ser divertida ou chata; revelar relações entre informações ou deixá-las confusas; abrir ou excluir possibilidades de ação efetiva-instrumental ou comunicativa.

Muitas das interfaces de ambientes existentes apresentam problemas na arquitetura da informação, de organização de elementos visuais, de interatividade e de funcionalidade. (Andrade, 2002) Estes problemas nas interfaces dificultam o aprendizado dos ambientes de EAD, prejudicam a memorização, retardam o desenvolvimento de uma tarefa, causam insatisfação subjetiva e aumentam a incidência de erros por parte dos usuários. Por isso, Winckler (2001), citado por Andrade(2002), declara que: quando a usabilidade é aplicada durante o processo de

concepção da interface vários problemas podem ser detectados e possibilita-se correções do tipo: redução do tempo de acesso à informação, tornar informações facilmente disponíveis aos usuários e evitar a frustração do usuário.

Um dos grandes entraves encontrados, é que muitos dos ambientes não passam de uma rede de páginas de hipertextos/hiperdokumentos estáticos e não aproveitam todas as funcionalidades e características de aplicações hipermídia, como de fato deveriam ser. Uma tentativa de saná-lo é preparar os textos disponibilizados para se adequar ao meio, sendo enriquecidos por estruturas de hipertexto dinâmicas, anotações, comentários, glossários, mapas de navegação, referências (links) para outros textos igualmente disponíveis, que possam servir como discussões ou complementos dos textos originais, desta forma, a eficácia de EAD aumenta consideravelmente.

2.3.3. Por que modelar?

Um modelo é uma visão simplificada de um sistema. Ele mostra os elementos essenciais do sistema de uma perspectiva específica e oculta os detalhes não essenciais (abstração). (Rumbaugh, 1994)

Os modelos são importantes para:

- ajudando na compreensão de sistemas complexos;
- explorando e comparando alternativas de *design* a um baixo custo;
- formando uma base para implementação;
- capturando requisitos com precisão ;
- comunicando decisões sem ambigüidade.

Uma modelagem visual usa de notações de *design* gráficas e textuais, semanticamente ricas, para capturar *designs* de *software*. Uma notação como a UML (*Unified Modeling Language*), por exemplo, permite que o nível de

abstração seja aumentado, enquanto mantém sintaxe e semântica rígidas. A linguagem de modelagem é muito importante porque a comunicação entre as partes (usuários, analistas, programadores, etc) é feita através da interpretação dos elementos criados usando essa notação (diagramas de classe, por exemplo), fornecendo assim, uma base não ambígua para a implementação.

2.3.2. Engenharia Reversa

Atualmente, houve um visível aumento no desenvolvimento de *softwares*, associado a isto, existe a grande problemática de pouca ou nenhuma documentação ou ainda, documentações errôneas referentes aos mesmos. Visto que, Rugaber(1992) citado por Feltrin(1999), considera que o maior esforço, tempo e custos empregados no desenvolvimento de software são gastos no entendimento e posterior manutenção dos sistemas. Na tentativa de sanar ou melhorar tal deficiência a Engenharia Reversa vem obtendo resultados bastante positivos.

Segundo Costa(1997), a Engenharia Reversa é uma técnica utilizada para recuperar informações a partir dos documentos do software relativos ao produto ou código-fonte, visando a obtenção de sua representação em um nível mais alto de abstração.

A Engenharia Reversa segue o sentido oposto ao da Engenharia Progressiva. A Engenharia Progressiva segue a seqüência que vai desde os requisitos, passa pelo projeto até implementação, tendo o sistema como resultado do processo de desenvolvimento. Em contrapartida, na Engenharia Reversa, o sistema geralmente é o ponto inicial do processo (Chikofsky; Cross II, 1990), citado por Feltrin(1999), parte-se de um baixo nível de abstração para um alto nível de abstração. A figura compara as diferenças de sentido de fluxo durante o ciclo de vida.

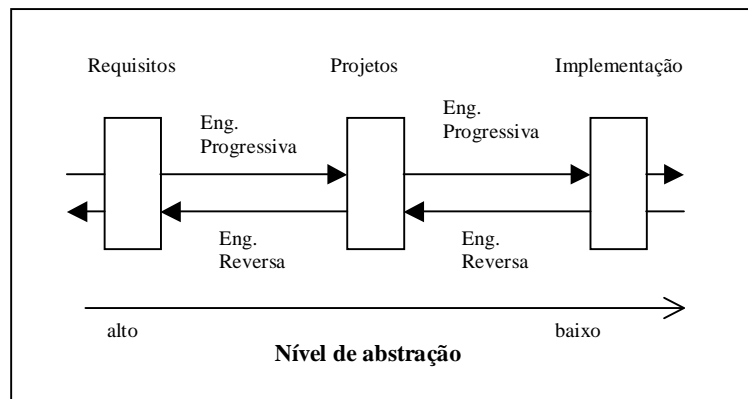


Figura 04: Processo de Engenharia Progressiva X Engenharia Reversa

Fonte: (Feltrin, 1999)

As duas importantes categorias englobadas pela Engenharia Reversa são: a redocumentação (visualização de código-fonte) e a recuperação de projeto (entendimento de programa). Ao término deste processo, são produzidos de forma automática, documentos que ajudam a aumentar o entendimento geral de sistemas de *software*, facilitando o reuso, manutenção, teste e controle de qualidade do *software*.

2.3.2. Arquitetura para desenvolvimento de aplicações Web em três camadas

Segundo Microsoft(2001) citado por Ângelo(2001), o desenvolvimento em três camadas, ou *three tiered*, é uma nova tecnologia que ganhou impulso com o advento da Internet e baseia-se no conceito de tratar qualquer processamento em três camadas distintas:

- A primeira é a **camada de usuário** (*User Service*) ou camada de apresentação, ou seja, o que o usuário observa em sua tela, recebendo resultados

do processamento ou requisitando que seja processada alguma informação. Há um enfoque no ambiente visual e de *layout* da aplicação que permita a interação do usuário com o sistema (*front-end*) do projeto através de uma interface amigável que facilite a navegação;

- A segunda camada, a **camada de regra de negócios** (*Business Service*), encapsula todas as regras que definem o funcionamento (comportamento interno) da aplicação na interação correta com a interface. Por intermédio de scripts *server-sides*¹ é criada a interface para referência aos objetos e componentes que são utilizados na conexão e tratamento das regras e dados da aplicação;
- A terceira é a camada dados (*Data Service*), responsável por recuperar os dados solicitados existentes no servidor, através de todas as *store procedures*² e regras de negócio acomodadas nesta camada, que gerenciam o banco de dados. Os dados podem ser registrados tanto em base de dados, quanto em *e-mails*, planilhas, arquivos, ou qualquer outra fonte.

A Figura 5, fornece uma visão geral do funcionamento de aplicações Web em três camadas na arquitetura cliente-servidor.

Esta tecnologia garante a estruturas com estruturas complexas, confiabilidade, interoperabilidade, flexibilidade, liberdade e independência da interface do usuário, além de permitir a reutilização das camadas de regras do negócio através das regras e procedimentos já implementados, nesse caso, dos componentes e também das regras estabelecidas no banco de dados por meio das *store procedures* utilizadas pelo sistema. (Ângelo, 2001)

¹ Scripts server-sides: *scripts* escritos em qualquer linguagem *script* que são interpretados pelo servidor web.

² Store procedure: Regra de negócio implementada diretamente do Sistema Gerenciador de Banco de Dados.

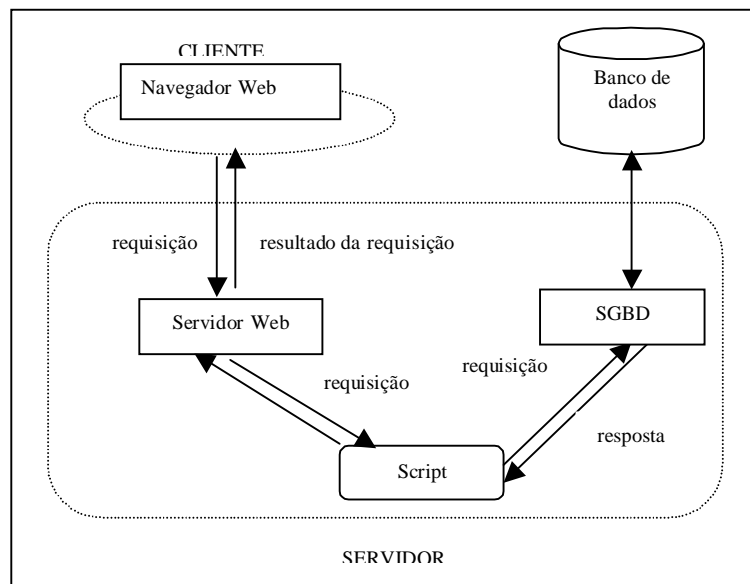


Figura 05 : Arquitetura cliente-servidor em aplicações web em três camadas

Fonte: (Muniz, 2002)

2.3.4. Casos de uso

Atualmente, uma das maiores fontes de fracasso é desenvolver *softwares* de qualidade, mas que não satisfazem aos desejos dos usuários. Um recurso bastante utilizado para se prevenir tal situação é transformar os requisitos do cliente em *use case*, sendo este, o elo de ligação entre o processo de desenvolvimento e as necessidades reais do cliente.

Cardoso(2003), define ator como: "qualquer pessoa ou sistema externo (SE) que tenha interação com o sistema que está em desenvolvimento", e *use case* (Caso de Uso) como: "um conjunto de funcionalidades de um sistema, representado por fluxos de eventos iniciados por um ator e apresentando um resultado de valor a um ator".

Normalmente, o nome de um Caso de Uso deve representar uma ação do sistema. Casos de Uso ajudam a entender os requisitos de um sistema, visto que, são usados para capturar comportamentos relevantes do sistema, sem a preocupação de especificar como os mesmos serão implementados. (Almeida, 2001).

As figuras representam um ator e um Caso de Uso utilizados em modelagens, respectivamente:

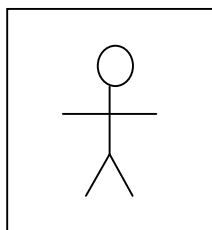


Figura 6.1: Ator

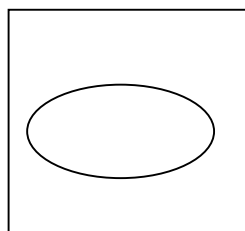


Figura 6.2: Caso de Uso

A especificação de Caso de Uso deve considerar:

- como e quando o Caso de Uso inicia ou termina
- as interações com os atores envolvidos
- a seqüência normal de execução
- as seqüências alternativas ou de exceção

Um Diagrama de Casos de Uso apresenta um conjunto de casos de uso, atores e suas relações, que tem como objetivo capturar as funcionalidades de um sistema de acordo com a visão de seus usuários.

Os Diagramas de Casos de Uso são utilizados para modelar o contexto de um sistema, identificando os atores e seus papéis na interação com o sistema, além dos requisitos de um sistema, especificando o que o sistema deve fazer sem, no entanto, se preocupar com sua implementação. Eles ainda são importantes para validar a arquitetura do sistema e identificar e gerar casos de teste.

Para Almeida(2001), cada Diagrama de Casos de Uso representa graficamente uma visão parcial do Sistema. Por sua vez, o conjunto de Diagramas de Casos de Uso forma a visão de Casos de Uso completa do Sistema, representando assim, uma visão externa ao sistema, que auxilia na identificação e especificação do conjunto de classes e suas interações para atingir o propósito do sistema.

2.3.5. OOADM

2.3.5.1. Hipertexto e hipermídia: conceitos básicos:

Faz-se necessário o conhecimento de alguns conceitos navegacionais essenciais para um melhor entendimento das estruturas usadas nos ambientes educacionais *on-line*.

Hipertexto: sistema ou aplicativo que permite criar, manter e manipular trechos de informações (textos e gráficos) interligados de forma não-sequencial ou não-linear; D'ipólito(1989), citado por Oliveira(2002).

Multimídia: combinação de texto, gráficos, desenhos, imagens, áudio e vídeo, em um conjunto ou apresentação exclusiva de computador; Vaughan(1994), citado por Oliveira(2002).

Hipermídia: junção dos tipos de dados da multimídia com os mecanismos e semânticas dos hipertextos; D'ipólito(1989), citado por Oliveira(2002).

Nós: trechos de informação correspondentes a uma parte de um documento de hipermídia ou de hipertexto. (Oliveira, 2002)

Elos: ligações entre dois nós. A ativação de um elo invoca o nó de destino de seu relacionamento. (Oliveira, 2002)

Hiperdocumento: rede de nós e elos. Uma aplicação pode ser composta por um ou mais hiperdocumentos relacionados. (Oliveira, 2002)

Âncoras: estrutura de ativação de um elo (normalmente botões). (Oliveira, 2002)

Estruturas de acesso: menus ou índices hierárquicos que permitem ao usuário o desvio do fluxo principal de idéias ou ainda roteiros, podendo ser pré-definidos ou não; Schwabe(1993), citado por Oliveira(2002).

Autoria em Ponto Grande: projeto da estrutura conceitual da hiperbase, ou seja, etapas envolvidas na modelagem estrutural e conceitual da aplicação tomadas de forma abstrata, em que detalhes de implementação são considerados. (Oliveira, 2002)

Autoria em Ponto Pequeno: etapas que envolvem a preocupação de como apresentar o modelo obtido pela Autoria em Ponto Grande, ou seja, como cada nó será apresentado e como os elos poderão ser acessados. (Oliveira, 2002)

O desenvolvimento de ambientes de EAD via Web pode se beneficiar das metodologias existentes para o desenvolvimento de aplicações hipermídias. Entre estas metodologias destacam-se: HDM (*Hypertext Design Model*), o RMM (*Relationship Mangement Methodology*), o OOHDM(*Object-Oriented Hypermidia Design Model*) e o EHDM (*Educational Hyperdocuments Design Method*). Contudo, o HDM e o RMM não são apropriados a domínios dinâmicos, como sistemas de tomada de decisão, ambientes de engenharia de programas e aplicações educacionais, optando-se assim, por utilizar o OOHDM criado por Schwabe & Rossi, como solução alternativa às falhas identificadas nos demais modelos. (Oliveira, 2002)

O OOHDM dá enfoque a modelagem de documentos hipermídia interativos baseados na utilização de classes e orientação a objetos, visando desta forma, ampliar o reuso de arquiteturas de *software*.

A seguir são descritas algumas das conveniências em associar orientação a objetos ao desenvolvimento de aplicações hipermídia, segundo à crescente tendência defendida por diversos autores da área, Magalhães(2002):

- são utilizadas as mesmas primitivas de modelagem (objetos, classes), simplificando a transição de uma atividade para outra.

- ao longo do processo utiliza-se os mesmos mecanismos de abstração, isto é, agregação, classificação e generalização/especialização.

- devido ao fato de os objetos serem artefatos reativos, podem ser construídas aplicações sofisticadas baseados em hipermídia, definindo-se padrões de comportamento e de comunicação entre objetos.

- há poderosos formalismos, já existentes para especificar a estrutura, o comportamento e as relações dos objetos e podem facilmente ser adaptados ao campo da hipermídia.

- Aplicações projetadas e construídas em torno de objetos tendem a ser mais robustas e fáceis de modificar, tanto pelo uso de polimorfismo e herança como por composição.

- Fornecem primitivas de modelagem de alto nível na forma de padrões de projeto que podem ser utilizadas sem alterações, ou modificadas de acordo com as necessidades do projetista.

- Construir novas aplicações reutilizando componentes existentes é altamente viável quando os componentes são descritos como objetos.

O OOHDM é um processo constituído de quatro atividades distintas: a Modelagem Conceitual, o Projeto da Navegação, o Projeto de Interface Abstrata e a Implementação.

A Figura 7, descreve um esboço das etapas ocorridas durante a modelagem OOHDM.

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

Figura 07: Atividades do OOADM

Fonte: (Magalhães, 2002)

2.3.5.2 A modelagem conceitual em OOADM

De acordo com Cunha(2002) citado por Magalhães(2002), a modelagem conceitual propõe a construção de um esquema contendo classes, objetos, relacionamentos e subsistemas existentes para o domínio especificado (mapa da aplicação). A descrição de classes segue a notação da orientação a objetos, mas seus atributos podem ser multi-tipados, representando assim diferentes perspectivas da mesma entidade real. A utilização de Agregação e Generalização/Especialização acrescentam poder de abstração ao sistema. Também é bastante recomendável o uso de Cartões de Classes e

Relacionamentos , similares aos cartões CRC, para auxiliar na tomada de decisões, tanto "para frente" como "para trás", a serem feitas no decorrer da documentação de modelagem do projeto.

Segundo Zambalde(1999), ao se concentrar nos aspectos conceituais, abstraindo da parte de apresentação, o projetista permite que um mesmo projeto lógico seja utilizado para implementação em diversas plataformas de *hardware* e *software*.

2.3.5.3. Projeto de Navegação

Os aplicativos de hipermídia são projetados para realizar navegação através de um espaço de informação. Dentro do Projeto Navegacional, estabelecesse as estratégias de navegação, bem como as visões que um determinado usuário terá ao navegar pela aplicação hipermídia. As estruturas de acesso (menus, índices hierárquicos) e os contextos definidos nesta etapa levam em conta os tipos de tarefas que aplicação suporta. A semântica de navegação é definida no mínimo em termos de estabelecimento de possíveis caminhos de navegação. (Zambalde, 1999)

2.3.5.4. Interface Abstrata

Nesta fase define-se a aplicação dinâmica com o qual o usuário irá interagir. Todos os objetos que foram definidos em etapas anteriores deverão ser mapeados em objetos perceptíveis (uso de *software*) que estarão disponíveis ao usuário e como eles se comportarão em termos de ações efetuadas durante sua navegação pela hiperbase.

2.3.5.5. Implementação

Nesta ultima etapa do OOHDM, o sistema hipermídia a ser executado é gerado após o mapeamento do design abstrato da interface em objetos de interface concretos. (Zambalde, 1999)

2.3.6. DFD

“Um Diagrama de Fluxo de Dados (DFD) é uma ferramenta de modelagem que nos permite imaginar um sistema como uma rede de processos funcionais, interligados por 'dutos' e 'tanques de armazenamento de dados’”. (Yourdon(1998), citado por Kléber(2001))

O DFD faz o mapeamento de todas as funções e dos dados utilizados para executá-la, especificando como estes dados devem entrar, sair ou ser armazenados, ou seja: descrevendo o fluxo de informações sob o ponto de vista das funções que o sistema executa. Fornece apenas uma **visão do sistema, a visão orientada para funções** : mostra o que o sistema faz e não como é feito.

Faz-se necessário ressaltar que o DFD mostra o **Fluxo dos Dados** e não o fluxo de controle do sistema, retrata a situação de ponto de vista dos dados. Portanto, controles, como iterações e decisões não aparecem nos DFDs.

O diagrama de fluxo de dados é sem dúvida uma ferramenta de importância fundamental para a análise e desenvolvimento de sistemas em uma linguagem, pois é nele que se pode desenhar e projetar um sistema estruturado antes do seu desenvolvimento, e com o diagrama fica muito mais simples fazer o banco de dados, imaginar como serão as telas do sistema e o seu processamento, em síntese, tem-se uma visão perfeita do sistema de forma simplificada e compreensiva.

2.3.6.1. Componentes de um DFD

Os DFD baseiam-se no princípio de que a funcionalidade de um sistema pode ser representada como uma rede que combina 4 componentes fundamentais, ilustrados na Figura 8:

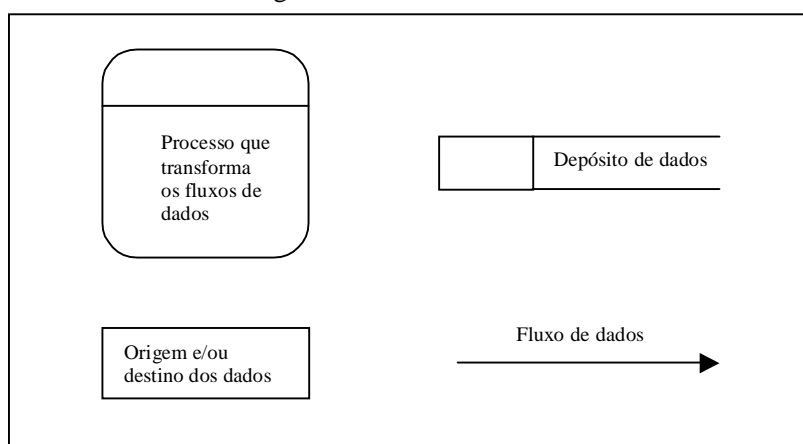


Figura 08: Simbologia básica de um DFD

Fonte: (Yourdon, 1998)

Processo - O principal componente de um DFD, sinônimos mais conhecidos são: Bolha, Função ou Transformação.

“Um processo é uma transformação dos fluxos de dados de entrada em fluxo de dados de saída”. (Yourdon(1998), citado por Kléber(2001)).

Os processos representam as diversas funções individuais que o sistema executa, mostrando as transformações dos dados (entradas em saídas). É representado por um quadrado com cantos arredondados, em seu interior descreve-se o processo que está sendo realizado. O processo deve conter sempre uma denominação indicando o que faz e ser descrito com uma única sentença simples (verbo + objeto).

Entidades Externas - também conhecidos como “terminadores”, geralmente, são classes lógicas, de atividades e ou pessoas que interagem com o sistema sendo fontes ou destinos das informações(criadores e consumidores de dados) para fora do sistema. Exemplo: cliente, banco, fornecedores, etc... .

Como as entidades são externas, os fluxos que interligam os terminadores aos processos representam a interface entre o sistema e o ambiente.

Qualquer relacionamento entre entidades externas não será mostrado no DFD.

Fluxo de dados - Modelam a passagem de dados, mostrando o movimento de fragmentos ou de pacotes de informações que fluem de um ponto a outro do sistema. Indicam sempre uma direção e pode ser bidirecional indicando um diálogo.

São representados por uma seta que entra ou sai de um processo indicando o sentido do fluxo, sendo nomeados com o nome do fluxo associado.

O Fluxo de dados não leva em conta os aspectos de procedimento, ou seja, não é possível saber, por exemplo, quando ou em que seqüência os pacotes transitam.

Depósitos de Dados - são os "armazéns" (arquivos ou tabelas de um banco de dados) que guardam dados e informações entre os vários processos.

Os depósitos são necessários para atender um requisito básico do usuário (armazenar informações históricas) ou para suprir algum aspecto prático de implementação do sistema.

Representados graficamente por um par de linhas paralelas, fechadas apenas de um lado por duas outras linhas, bem próximas, formando um pequeno quadrado do lado esquerdo. Nesse quadrado coloca-se uma referência numérica arbitrária para o depósito, antecedida pela letra "D" e, no espaço restante, coloca-se o nome atribuído ao banco de informações, que deve ser aquele usado no dia-a-dia do usuário.

Os depósitos são interligados aos processo por meio de fluxo de dados. Um fluxo que parte de um depósito é interpretado como uma leitura, enquanto, um fluxo para um depósito é interpretado como uma escrita, ou quando novos pacotes são acrescentados/retirados.

2.3.6.2. DFD com níveis

Em sistemas de alguma complexidade, os DFD são organizados em níveis. Um DFD de um dado nível da hierarquia contém uma explicação funcional mais detalhada de um processo na hierarquia superior. A decomposição consiste na subdivisão de um DFD de alto nível numa hierarquia de DFD.

O DFD de nível mais alto consiste de uma única bolha, representando o sistema inteiro, os fluxos de dados mostram as interfaces entre o sistema e as entidades externas. Esse DFD especial é conhecido como **diagrama de contexto**.

O DFD imediatamente abaixo do diagrama de contexto é conhecido como nível 0. Ele representa a visão de mais alto nível das principais funções do sistema bem como as principais interfaces entre essas funções.

2.3.7. Modelagem de dados

Modelagem de Dados é uma atividade desenvolvida em fases variadas do processo metodológico de desenvolvimento de sistemas, com a finalidade de garimpar informações para a obtenção do modelo de dados.

O Modelo de Dados é a representação gráfica e textual das estruturas, dos operadores e das regras que definem dados. As estruturas são formadas de: entidades ou blocos conceituais, relacionamentos e atributos. As regras são asserções que regulam o funcionamento dessas estruturas. Os operadores são

comandos que permitem a manipulação das estruturas segundo as regras estabelecidas.

Para Barbieri(1994), "os dados devidamente modelados representam a mais preciosa matéria-prima para a confecção de sistemas estáveis e de uso efetivo."

Por isso, os dados devem apresentar requisitos básicos de documentação, como o modelo de dados, cuja finalidade é garantir a permanência de sua verdade semântica e de seu entendimento ao longo do tempo. Apesar de ser um dos quesitos mais importantes, é também, um dos mais negligenciados atualmente pelas empresas.

Todo projeto de banco de dados deve conter os seguintes modelos:

1. Modelo Conceitual de Dados - MCD

- É aquele em que os objetos, suas características e relacionamentos têm a representação fiel ao **ambiente observado**, independentemente das limitações quaisquer impostas por tecnologias ou implementação (SGBD), ou seja, é a ferramenta para a representação da realidade num alto-nível de abstração. Ele produz um esquema mais estável do que o esquema lógico. O MCD é mais fácil de ser entendido pelos usuários, além de ser livre de considerações de eficiência e armazenamento.

2. Modelo Lógico de Dados - MLD

- É aquele em que os objetos, suas características e relacionamentos têm a representação de acordo com as **regras da implementação** e do tipo de tecnologia utilizada (SGBD), ou seja, é uma ferramenta para a descrição de estruturas de dados numa forma passível de ser processada por computador. Ele produz um esquema sujeito a eventuais reestruturações, para garantir a eficiência das consultas. O MLD é fácil de ser entendido pelos usuários.



Figura 09: Visão geral dos projetos de um banco de dados

3. Modelo Físico - MFD

- É aquele em que a representação dos objetos é feita sob o foco do nível físico de implementação das ocorrências, ou instâncias das entidades e seus relacionamentos, ou seja, é uma coleção estática de estruturas sintáticas e/ou gráficas, construída a partir de um modelo de dados, que descreve uma porção específica da realidade. O Conhecimento do modo físico de implementação das **estruturas de dados** é o ponto básico para este modelo. Tem foco nos requisitos de desempenho: tempo de resposta, throughput, utilização de espaço.

2.7.2. Modelo ER (Entidade-Relacionamento)

O Modelo Conceitual de Dados é representado por um gráfico denominado DIAGRAMA ENTIDADES X RELACIONAMENTOS, proposto por P.P.Chen. Trata-se de um diagrama que detalha as associações existentes entre as entidades de Dados e utiliza componentes semânticos próprios.

A terminologia básica do Modelo Entidade x Relacionamento é apresentada a seguir:

ENTIDADE: são os componentes físicos e abstratos utilizados pela empresa, sobre os quais são armazenados dados;

ATRIBUTO: corresponde à representação de propriedades de uma Entidade. Um atributo não tem vida própria ou independente. Existe apenas para caracterizar uma Entidade;

OCORRÊNCIA: conjunto de atributos de uma determinada Entidade;

RELACIONAMENTO: é uma correspondência entre duas entidades. Uma associação entre dois conjuntos de dados (entidades);

IDENTIFICADOR ou ATRIBUTO DETERMINANTE: um atributo ou uma coleção de atributos que determina de modo único uma Ocorrência de Entidade;

GRAU DE RELACIONAMENTO: o número de entidades que participam da associação;

CLASSE DE RELACIONAMENTO ou CARDINALIDADE: quantas ocorrências de cada entidade estão envolvidas no relacionamento. Pode ser:

- 1:1 (um para um)
- 1:N (um para muitos)
- M:N (muitos para muitos)

2.7.3. Modelo Relacional

No Modelo Lógico (relacional) o conjunto de dados são representados por **tabelas** de valores bidimensionais denominadas **relações**.

As **relações** são organizadas em **linhas** e **colunas**.

Cada elemento de um conjunto de dados é representado por uma linha da relação (instância).

Os valores de uma coluna são retirados de um conjunto de valores denominado de **domínio** da coluna.

Uma coluna, ou concatenação de colunas, de uma relação é denominada **chave**, se para uma linha qualquer da relação não existe uma outra linha com o mesmo valor nessa coluna (ou nessa concatenação de colunas).

Freqüentemente é de praxe utilizar a modelagem ER para realizar o mapeamento para o modelo Relacional, contudo o inverso (Relacional/ER) pode ser um pouco mais complexo e necessitar do uso da Engenharia Reversa.

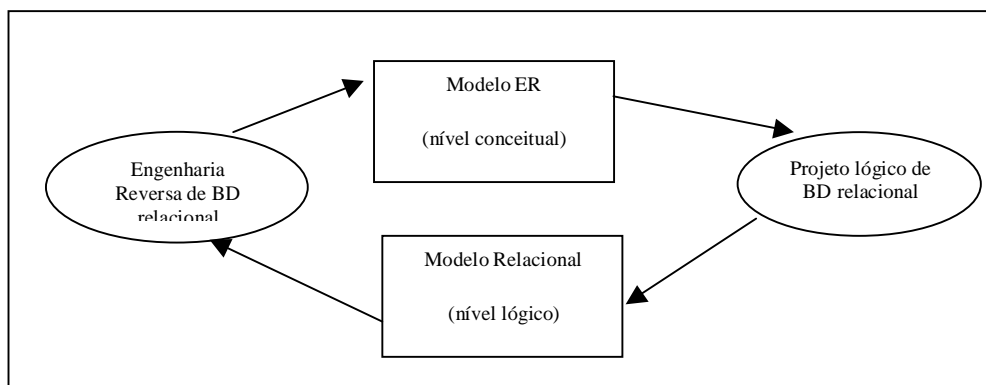


Figura 10: Visão geral do processo de mapeamento entre os modelos ER Relacional

3. Metodologia

O presente trabalho foi desenvolvido no âmbito das dependências do Departamento de Ciência da Computação da Universidade Federal de Lavras.

A princípio houve um estudo bibliográfico a respeito do assunto a ser desenvolvido, neste caso, sobre modelagem de sistemas, os tipos de modelos existentes, os modelos viáveis e os não tão viáveis, projetos e estudos já realizados, a fim de obter um maior embasamento teórico que oferece respaldo para partir para a modelagem em si.

Também foi possível ter contato com o desenvolvedor do sistema em questão, que pôde mostrar a aplicação em funcionamento e posteriormente procurou esclarecer quaisquer dúvidas a respeito de funcionalidades implementadas. Obter esta visão do desenvolvedor auxiliou bastante o início da modelagem, não permitindo se abrissem brechas para ambigüidades.

A abordagem adotada utilizou recursos da área de Engenharia Reversa. Como o próprio nome sugere, trata-se de uma Engenharia de Software com a direção do fluxo de ciclo de vida oposto ao da Engenharia Progressiva. Seu objetivo é partir do todo, isto é, do sistema já implementado para suas partes, tentando ao seu término conseguir mapear todas as funcionalidades existentes na aplicação, de tal forma, que algum analista ou programador consiga facilmente entender o que foi implementado e como pode-se alterá-lo com mais facilidade.

Optou-se por utilizar uma modelagem baseada na arquitetura cliente-servidor para desenvolvimento de aplicações Web em três camadas, por abranger toda a implementação: a interface, os códigos e a base de dados.

Para modelar a interface foram usados diagramas de *use-case*, que possibilitam rapidamente visualizar as opções disponíveis ao manuseio para cada usuário; além disso também se utilizou a modelagem OOHDM navegacional, com o intuito de tentar mapear os caminhos e as funcionalidades que vão se disponibilizando ao mesmo a partir de uma determinada opção. A modelagem OOHDM foi realizada no aplicativo ARGOUML, versão 1.4 (disponível em <http://arguml.tigris.org>), por se tratar de um *software* livre e fornecer suporte a esta modelagem.

A modelagem que busca o entendimento do comportamento de suas funções em resposta ao manuseio da interface, foi proposto o uso de Diagramas de Fluxos de Dados, que permitem visualizar funções do sistema, suas entradas e saídas e seu relacionamento com a base de dados.

Finalmente, na camada de dados, optou-se por descrever a estrutura do banco de dados utilizando o modelo relacional e o modelo entidade-relacionamento.

Paralelamente, à modelagem iniciou-se a redação da presente monografia, apresentando o que foi estudado, quais os resultados obtidos e as conclusões alcançadas ao término do desenvolvimento do projeto.

4. Resultados e Discussão

4.1. Visão geral do ambiente de ensino

O sistema ao qual se aplicou a modelagem foi desenvolvido por Roosevelt Mairink dos Santos Júnior. Trata-se de um ambiente de ensino a distância para cursos de Pós-Graduação *Lato Sensu* da Universidade Federal de Lavras, implementado em PHP e MySQL e executado no servidor Apache³.

O ambiente educacional projetado, apresenta dois sub-ambientes: a Coordenação e a Sala de Aula, tendo como usuários: o administrador, a secretária, o professor e o aluno.

Antes de executar qualquer funcionalidade disponível no ambiente educacional, o usuário deverá se autenticar no sistema digitando seu *login* e senha. Somente assim, será possível identificar quem é o usuário em questão e quais os acessos que lhe serão permitidos.

O sub-ambiente Coordenação, engloba toda a parte gerencial do sistema envolvendo cadastramento, alteração, consulta e exclusão dos módulos: cursos, secretárias, professores, disciplinas, turmas e alunos. Este sub-ambiente é gerenciado pelo administrador (único usuário que possui permissão de realizar todas as atividades gerenciais em todos os cursos) e pela secretária (usuário que só terá permissão de gerenciar funções relativas a professores, disciplinas, turmas e alunos do curso ao qual está associado).

Devido a dependência existente entre os módulos, é necessário que haja uma seqüência na inserção de seus dados, a fim de se evitar retorno de erros. Por exemplo, não seria possível cadastrar uma disciplina sem antes ter disponível um curso e um professor. A ordem correta de inserção deverá ser: curso, secretária, professor, disciplina, turma e aluno.

³ PHP (*Hypertext Preprocessor*), MySQL e Apache são *softwares* livres bastante seguros e eficientes.

No sub-ambiente Sala de Aula, podem ser disponibilizados aos professores e alunos cinco módulos: Alunos, Bibliolinks, Fórum de Discussões, Notas e Sala de Reuniões. Este sub-ambiente disponibiliza informações relevantes do aluno e ferramentas que possibilitam a interação entre professores-alunos e alunos-alunos gerando assim, uma aprendizagem colaborativa .

Em Alunos, o professor obtém a relação dos alunos matriculados em uma turma, podendo enviar *e-mail* a todos os alunos da turma ou a algum aluno específico; o aluno visualizará as informações de sua turma e também poderá postar *e-mail*.

Em Bibliolinks, serão permitidos ao professor inserir, remover, alterar e visualizar bibliolinks (bibliografia ou *link*) de uma disciplina, ao aluno não existe a possibilidade de efetuar remoção de bibliolinks, as demais opções podem se efetuadas.

Em Fórum de Discussões, há um fórum para cada disciplina, no qual alunos e professores podem postar algum tópico (dúvidas ou sugestões) e também é possível respondê-los (o fundo da resposta do professor se encontra numa cor diferenciada). Também é permitido ao professor remover tópicos e respostas que verificar não serem relevantes.

Em Notas, o professor poderá inserir, alterar ou consultar as notas de uma disciplina para os alunos de uma turma, enquanto o aluno poderá consultar todas as suas notas.

Em Sala de Reuniões, é disponibilizada uma ferramenta de bate-papo para que os alunos e professores possam se comunicar sincronamente para tratar de assuntos relacionados às disciplinas ou simplesmente para trocarem idéias informais. É permitido o envio de mensagens reservadas ou para todos os presentes na sala.

4.2. A modelagem do sistema

A descrição da estrutura do sistema permite uma visão geral do ambiente como um todo e as possíveis interações de seus usuários. Contudo, isto é insuficiente para documentar um *software*, por isso, é proposta a modelagem da aplicação mencionada. Esta modelagem segue os princípios da arquitetura para desenvolvimento em três camadas de aplicações Web, contendo: a camada do usuário (interface), camada de regras de negócio (*scripts servers*) e camada de dados (banco de dados). É importante ressaltar que os modelos e diagramas apresentados abrangem o âmbito do administrador (usuário que possui todas as funcionalidades do sub-ambiente Coordenação) e do professor (usuário que engloba todas as funcionalidades referentes à(s) disciplina(s) que ministra no sub-ambiente Sala de Aula).

Por se tratar de um sistema bastante complexo, a modelagem gerada se torna bastante extensa, portanto somente alguns exemplos estão incluídos no corpo da monografia, os demais se encontram na seção de Anexos.

4.2.1. Camada de usuário

Nesta camada, serão encontrados os modelos que definem como o usuário interage com o sistema através de sua interface.

Primeiramente, visualiza-se um diagrama de *use-case* que indica quais opções podem ser realizadas pelo usuário na manipulação da interface e a seguir, é apresentada a descrição um *use case*.

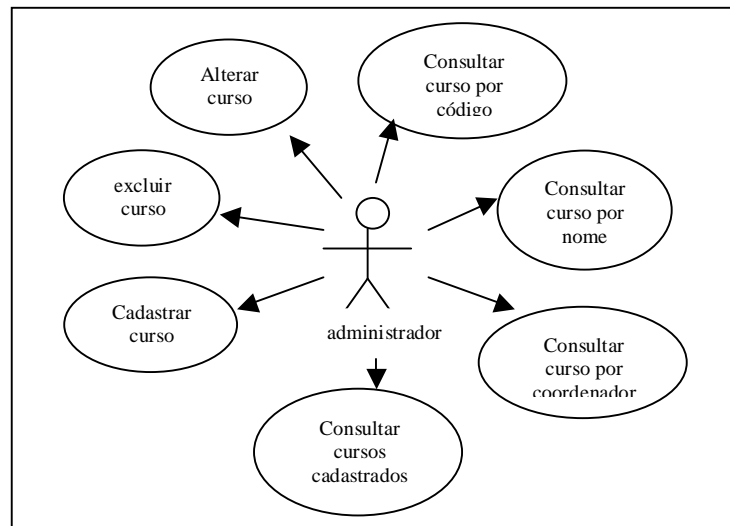


Figura 11: Diagrama de *use-case* de cursos do administrador

Use case: Cadastrar curso

O administrador poderá cadastrar um novo curso no sistema.

Ator: Administrador

Entradas e pré condições: Digitar os dados do curso.

Saídas e pós condições: Curso incluído no banco de dados.

Sequência de operações:

- 1- O administrador digita todos os dados do curso;
- 2- O sistema verifica se o curso não está cadastrado;
- 3- Se ele estiver cadastrado (já foi cadastrado anteriormente), fim;
- 4- Caso contrário, o sistema insere os dados do curso na tabela de cursos.

As figura 12 e 13, ilustram exemplos de modelagem OOHDH navegacional, cuja finalidade é mapear o caminho que o usuário poderá seguir a partir de uma opção acessada em uma determinada página.

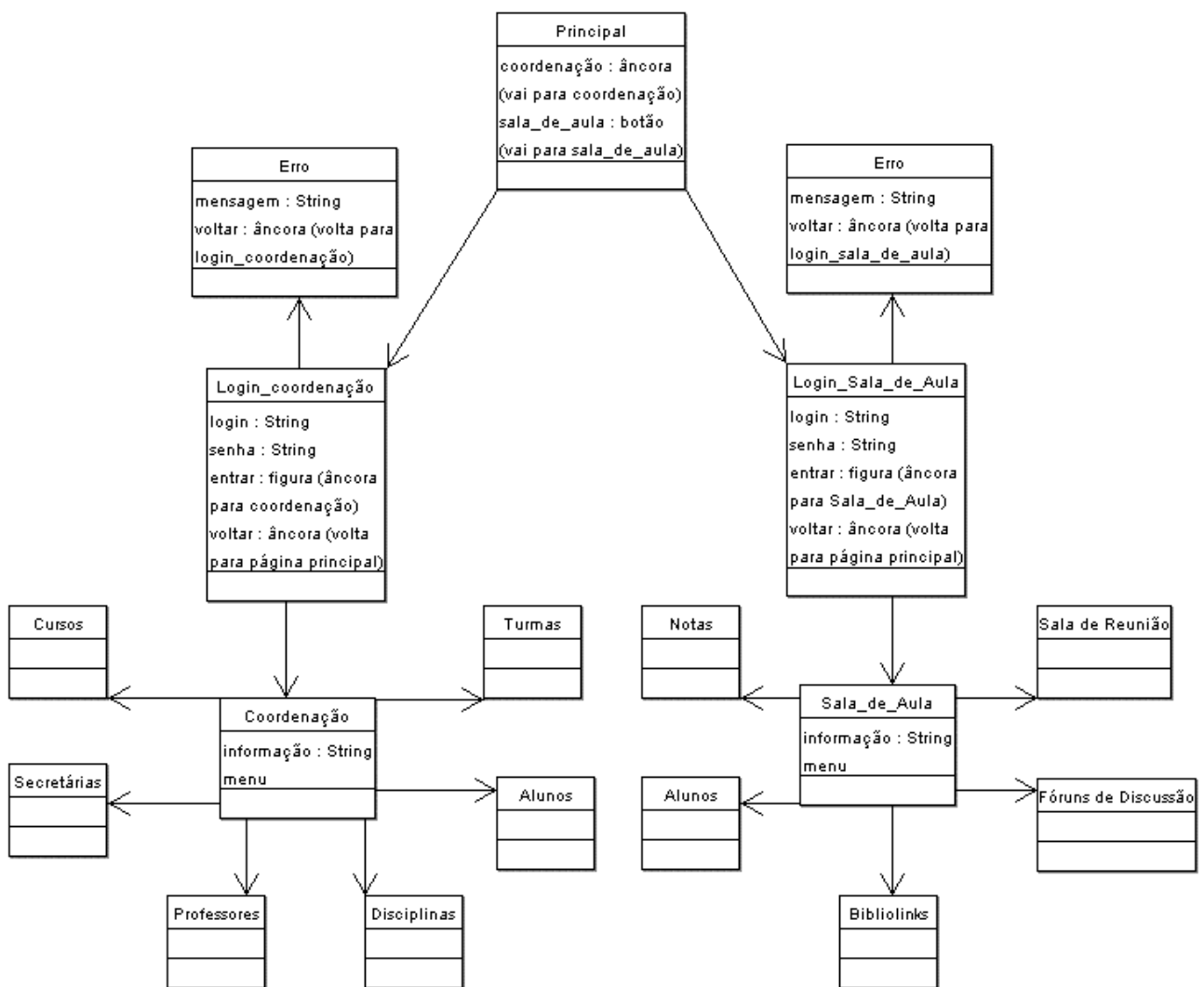


Figura 12: Modelagem navegacional da página principal

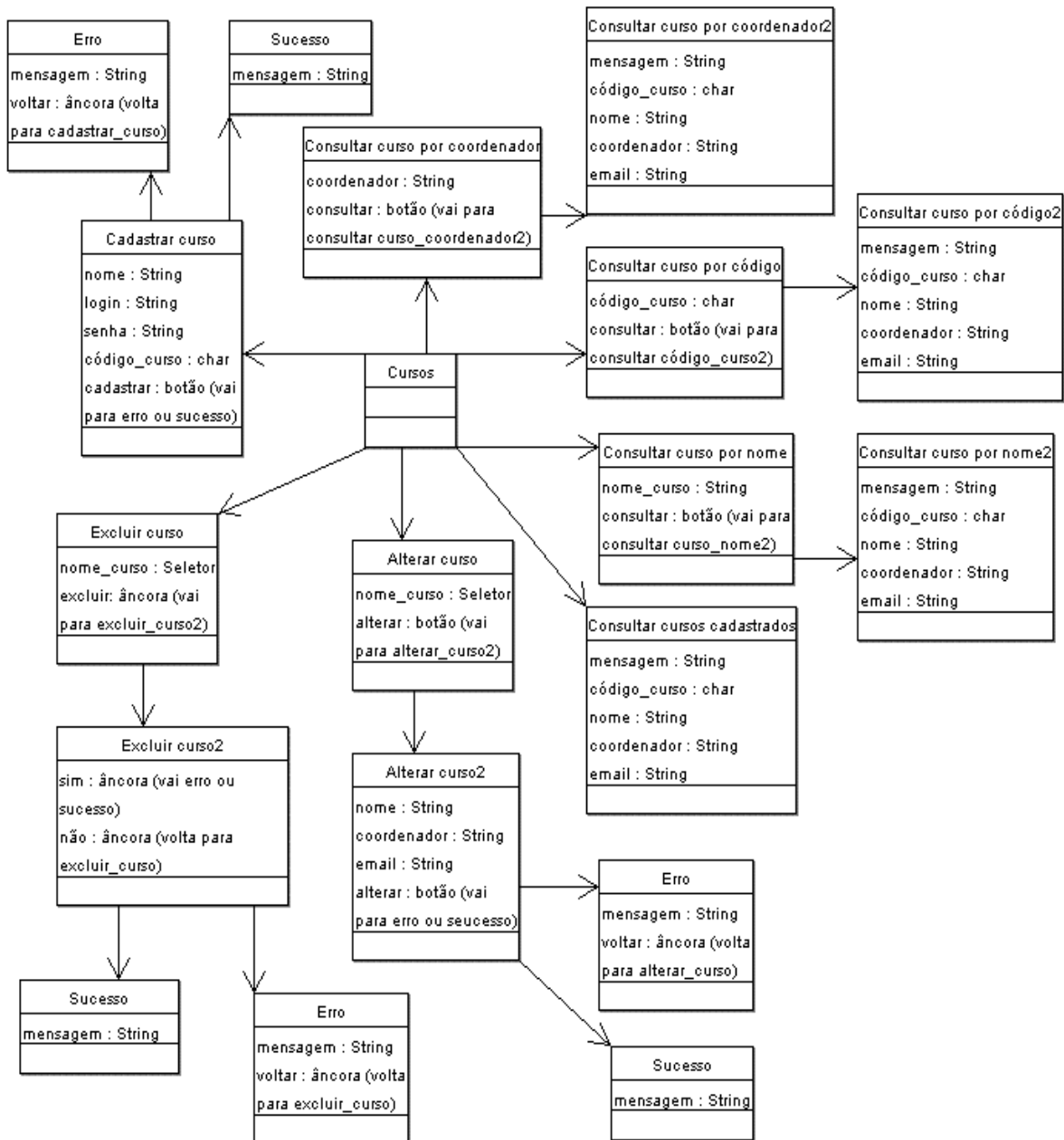


Figura 13: Modelagem navegacional coordenação_cursos

4.2.2. Camada de regra de negócio

No escopo desta camada é definido o funcionamento do sistema, isto é, como serão processados os dados manipulados na interface. Esta etapa não é visualizada pelo usuário do sistema.

O modelo proposto para descrever o comportamento da aplicação é o Diagrama de Fluxo de Dados (DFD), que determina quais são as funções e como estas interagem entre si e também permitem a visualização dos tipos de entradas e saídas associados a cada função.

As entidades externas (usuários) são representadas por retângulos e os fluxos de dados por setas (caso a seta aponte para direita, representa entrada de dados e se a seta apontar para a esquerda, representa dados retornados). Os processos (manipulam e transformam dados) são representados por retângulos de borda arredondada. Os depósitos de dados (local de armazenamento das informações) são representados por retângulos abertos.

Segue alguns exemplos de DFD modelados:

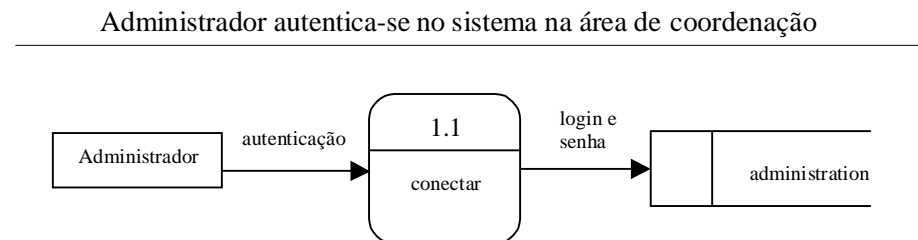


Figura 14: DFD da autenticação do administrador

Administrador cadastra novo curso no sistema

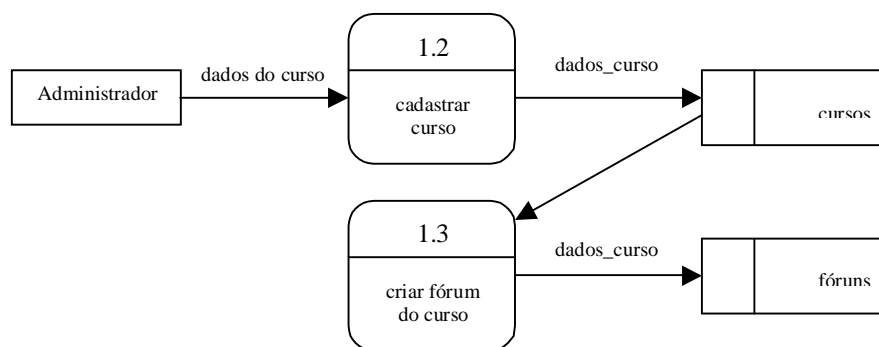


Figura 15: DFD de cadastro de curso

4.2.3. Camada de dados

Num sistema tão complexo como o que se está modelando, deve-se garantir a integridade, segurança e disponibilidade dos dados manipulados armazenando-os numa base de dados confiável. O banco de dados que armazena as informações necessárias ao funcionamento correto do sistema implementado foi criado com o nome de **faepe** e está alocado na máquina local (*localhost*), na qual o sistema está sendo executado.

É indispensável que se tenha uma noção da estrutura conceitual e lógica do banco de dados gerado a partir do código fornecido. Isto é melhor verificado através do modelo E-R (entidade-relacionamento) e do modelo relacional que são descritos a seguir.

Obtida a modelagem E-R consegue-se efetuar o mapeamento para o dicionário de dados. Sendo que, a base de dados é constituída pelas seguintes tabelas: *administration*, *cursos*, *secretarias*, *usuarios*, *disciplinas*, *turmas*, *alunos_turmas*, *foruns*, *foruns_disciplinas*, *foruns_respostas*, *bibliolinks*, *chat_users*, *chat_messages*, *notas*.

Os atributos (login, senha) da tabela administration serão utilizados para identificar e autenticar o administrador (usuário *root*) do sistema, sendo permitido a este o controle total da parte gerencial do sistema.

Os dados referentes ao único administrador são inseridos logo após a criação do banco de dados, antes que o sistema seja manipulado por outros usuários; devido a isso é que não existe chave-primária.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Administration	login	varchar	15	-----
	senha	varchar	50	

Tabela 02: Administration do banco de dados

Os atributos da tabela de cursos definirão os cursos que serão criados e disponibilizados pelo administrador. A chave-primária desta tabela é o atributo codigo, o atributo tem_secretaria deveser ser setado para 1 se houver secretaria associada, senão deve-se setá-lo em 0.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Cursos	codigo	char	3	codigo
	nome	varchar	100	
	coordenador	varchar	100	
	email	varchar	50	
	tem_secretaria	char	1	

Tabela 03: Cursos do banco de dados

Os atributos da tabela secretarias definirão as secretárias que serão criadas e disponibilizadas pelo administrador. Estas serão responsáveis por determinado curso e terão direito a realizar várias funções da parte gerencial. Sua chave-primária é composta pela associação do codigo_secretaria com o codigo_curso.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Secretarias	codigo_secretaria	int	3	codigo_secretaria, codigo_curso
	codigo_curso	char	3	
	login	varchar	15	
	senha	varchar	50	
	nome	char	100	

Tabela 04: Secretárias do banco de dados

Os atributos da tabela usuarios definirão os usuários (professores ou alunos) que serão criados e disponibilizados pelo administrador e/ou pela secretária. Estes serão os usuários de um determinado curso e terão acesso à sala de aula do sistema. Caso seja professor o atributo tipo deverá ser setado para 1, se for aluno será setado para 0.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Usuarios	codigo_identificacao	int	3	codigo_identificacao
	tipo	char	1	
	login	varchar	15	
	senha	varchar	50	
	codigo_curso	char	3	
	nome	varchar	100	
	cidade	varchar	50	
	estado	varchar	2	
	email	varchar	50	

Tabela 05: Usuários do banco de dados

Os atributos da tabela disciplinas definirão as disciplinas que serão criadas e disponibilizadas em cada curso pelo administrador e/ou pela secretária. Para toda disciplina deve ser associado um professor correspondente. Em caso afirmativo, os atributos lancou_nota e tem_bibliolink deverão ser setados para 1, em caso negativo, serão setados para 0.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Disciplinas	codigo_disciplina	int	3	codigo_disciplina, codigo_professor
	codigo_curso	char	3	
	codigo_professor	int	3	
	nome	varchar	100	
	lançou_nota	char	1	
	tem_bibliolink	char	1	

Tabela 06: Bibliolinks do banco de dados

Os atributos da tabela turmas definirão as turmas que serão criadas e disponibilizadas para cada curso pelo administrador e/ou pela secretária. No caso de uma turma estar ativada deverá ser setada para 1, em caso contrário, será setada para 0.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Turmas	codigo_turma	int	3	codigo_turma, codigo_curso,
	codigo_curso	char	3	
	ativada	char	1	

Tabela 07: Turmas do banco de dados

Os atributos da tabela alunos_turmas definirão os alunos que serão criados e disponibilizados em cada turma pelo administrador e/ou pela secretária.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Alunos_turmas	codigo_turma	char	4	codigo_turma, codigo_identificacao
	codigo_identificacao	int	3	

Tabela 08: Alunos_turmas do banco de dados

Os atributos da tabela fóruns definirão os fóruns que serão criados e disponibilizados ao se associar determinado curso a eles.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Foruns	codigo_forum	int	3	codigo_forum, codigo_curso
	codigo_curso	char	3	

Tabela 09: Fóruns do banco de dados

Os atributos da tabela fóruns_disciplinas definirão os tópicos (normalmente perguntas dos alunos) que serão criados e disponibilizados em relação a determinada disciplina no fórum de discussões.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Foruns_disciplinas	codigo_topico	int	11	codigo_topico, codigo_disciplina
	codigo_disciplina	int	3	
	codigo_forum	int	3	
	codigo_autor	int	3	
	assunto	varchar	100	
	mensagem	text		
	data	datetime		
	visitas	char	3	

Tabela 10: Fóruns_disciplinas do banco de dados

Os atributos da tabela fóruns_respostas definirão as respostas aos tópicos correspondentes do fórum de discussões.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Foruns_respostas	codigo_resposta	int	11	codigo_disciplina, codigo_professor
	codigo_topico	int	11	
	codigo_disciplina	int	3	
	codigo_forum	int	3	
	codigo_autor	varchar	3	
	assunto	varchar	50	
	resposta	text		

Tabela 11: Fóruns_respostas do banco de dados

Os atributos da tabela bibliolinks definirão os links ou bibliografias que serão postados em relação a determinada disciplina. Caso seja link, o atributo deverá ser setado para 1, se for bibliografia, será setado para 0.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Bibliolinks	codigo_bibliolink	int	3	codigo_bibliolink, codigo_disciplina
	codigo_disciplina	int	3	
	tipo_bibliolink	char	1	
	assunto	varchar	50	
	url	text		
	bibliografia	text		
	descricao	text		

Tabela 12: Bibliolinks do banco de dados

Os atributos da tabela notas definirão as notas dos aluno referentes a determinada disciplina.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
Notas	codigo_identificacao	int	3	codigo_turma, codigo_curso
	codigo_disciplina	int	3	
	nota	varchar	5	

Tabela 13: Notas do banco de dados

Os atributos da tabela chat_users definirão quais os usuários que estarão participando da Sala de Reuniões.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
chat_users	ID	int	11	ID
	logintime	datetime		
	user	varchar	18	
	IP	varchar	150	
	status	varchar	10	

Tabela 14: Chat_users do banco de dados

Os atributos da tabela chat_messages definirão as mensagens impressas na tela que serão enviadas pelos usuários enquanto estiverem participando da Sala de Reuniões.

Tabela	Atributo	Tipo	Tamanho	Chave-primária
chat_users	ID	int	5	ID
	time	datetime		
	user	varchar	18	
	msgtext	text		
	target	text		

Tabela 15: Chat_messages do banco de dados

Na figura 15, pode-se observar a modelagem E-R (Entidade-Relacionamento) obtida para o banco de dados da aplicação em estudo.

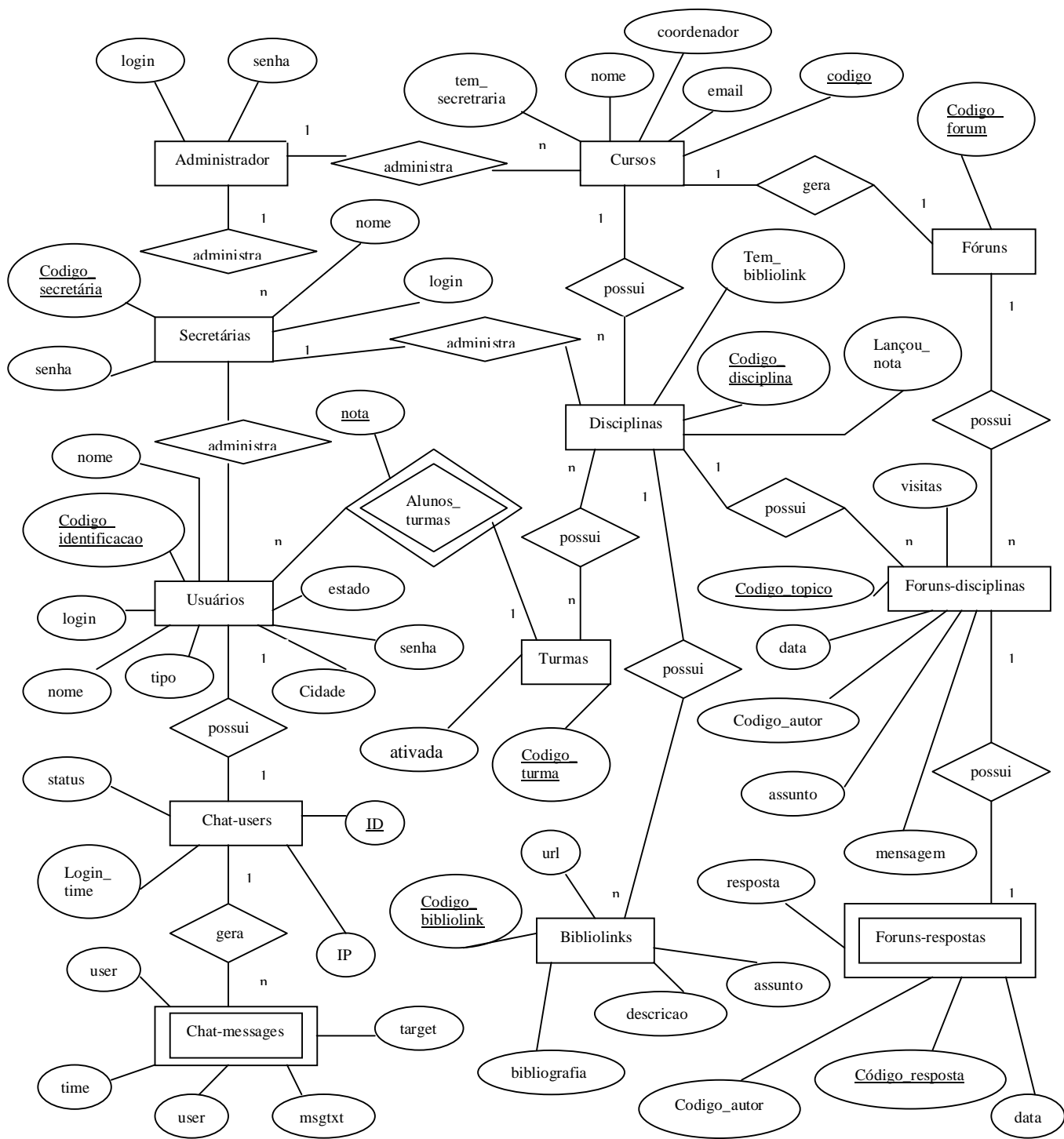


Figura 16: Modelo E-R

5. Conclusões e propostas para trabalhos futuros

No presente trabalho foi realizada a modelagem de uma ferramenta de interação ao Ensino a Distância, após extenso estudo bibliográfico e análise minuciosa do código-fonte e da observação das respostas fornecidas através da interação com a interface do sistema.

No decorrer da modelagem pôde-se averiguar que é necessário fazer-se um estudo dos modelos existentes para que os escolhidos melhor se adequem às características do sistema a ser modelado, como orientação a objetos ou a funções, desenvolvimento para Web.

Muitas vezes, um modelo proposto não consegue se adequar perfeitamente ao sistema projetado, sendo necessário adaptá-lo às suas necessidades, como o que ocorreu com a modelagem OOHDm nesta modelagem. Apesar da modelagem OOHDm ser orientada a objetos e o sistema também, foi preciso que se fizesse a modelagem navegacional baseada em páginas; desta forma conseguiu-se uma melhor visualização dos possíveis caminhos a serem seguidos por um usuário que acesse o sistema, e não uma visão apenas num nível macro.

Iniciar uma modelagem de um sistema já pronto (técnicas de Engenharia Reversa) pode apresentar vantagens de se ter exatamente noção de como a aplicação funciona, contudo, pode-se chegar à conclusão de que a mesma não esteja funcionando exatamente como se deveria e que as alterações a serem realizadas serão tão ou mais custosas do que se partisse de um modelo planejado, além de poderem ser constatadas "gambiarras" no código que solucionam parcialmente os problemas ou códigos que pudessem ser otimizados. Portanto, modelar é preciso (tanto antes ou depois do sistema pronto), para se economizem esforços e tempo com manutenção e no desenvolvimento de *software*.

É importante ressaltar que por se tratar de um ambiente educacional, no qual os usuários que o acessem possam não ter uma afinidade muito grande com recursos computacionais, é que a interface baseou-se em princípios de usabilidade que facilitassem sua manipulação e fornecessem mensagens de erro ou de informação se necessário, além de conter inúmeras ilustrações que são mais facilmente associadas e assimiladas pelo usuário.

A seguir são expostas algumas propostas para trabalhos futuros:

- Finalizar a descrição dos modelos propostos neste trabalho para os usuários secretária e aluno, apesar de suas funcionalidades já estarem englobadas nas do administrador e do professor respectivamente;
- Modelar um fluxograma ou um diagrama de atividade para cada função do sistema, visto que as modelagens propostas não ofereceram suporte a representação de desvios, *loops* e até mesmo como a dependência entre as funções é tratada;
- Na parte de Sala de Aula do ambiente educacional poderia ser implementada a disponibilização de material complementar das disciplinas para os alunos, no qual o professor faria um upload destes arquivos, possibilitando um maior enriquecimento ao curso;
- Disponibilizar um sistema de ajuda aos usuários, visto que muitos deles possam não apresentar afinidade com as tecnologias Web disponibilizadas.

6. Bibliografia

ALMEIDA, A. ; DAROLT, R.; **Pesquisa e desenvolvimento em UML**; 2001. Projeto de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade do Sul de Santa Catarina, Araranguá, SC.

ANDRADE, L.; **Desenvolvimento de Interfaces Ergonômicas para Ambientes de Ensino a Distância** - Congresso 2002

Disponível em: <<http://educarecursosonline.pro/artigos/>>.

Acessado em abril de 2003.

ÂNGELO, F.D., **Acesso ao sistema acadêmico via web** - artigo publicado em 2001.

BARBIERI, C., **Modelagem de dados**, 3ª edição - Editora IBPI Press, 1994.

CARDOSO, C., **UML na prática - do problema ao sistema** - Editora Ciência Moderna, 2003.

CARVALHO, C.A., **Ambiente de Ensino a Distância via Internet para o agronegócio: modelagem e implementação**, 2002 - Monografia (Curso de Especialização em Gestão da Informação no Agronegócio) - Universidade Federal de Juiz de Fora, MG.

CHIKOFSKY, E.J.; CROSS II, J.H., **Reverse Engineering and Design Recovery: A Taxonomy**. IEE Software, v.7, 1999.

COSTA, R.M., **Um método de Engenharia Reversa para Auxiliar a Manutenção de Software**, 1997. Dissertação de mestrado, ICMC–USP, São Carlos - SP.

CUNHA, R.M.C.M., **Autoria em Hiperfídia: O modelo OOADM Aplicado Ao Ensino de Linguagens de Programação**, UFLA, 2002.

D'IPOLITO, C., **Hipertexto: uma visão geral**. Rio de Janeiro: COPPE / Sistemas/ UFRJ, 1989 (Boletim Técnico, 197)

FELTRIN, V.D., **Engenharia Reversa de Software por meio de Hipertextos**; 1999. Dissertação (Mestrado em Ciência - Área de Ciências de Computação e Matemática Computacional) - Universidade de São Paulo, SP.

HALL, B., *Learning Management System: How to Choose the Right System for your Organization*

Disponível em: <<http://www.brandon-hall.com>>.

Acessado em maio de 2003.

KLÉBER, R., **Análise e projeto estruturado de sistemas**; 2001- Universidade Federal do Rio Grande do Sul, Departamento de Informática e Matemática Aplicada

LDBE, **Lei de Diretrizes e Bases da Educação Nacional** - Lei nº 9394 - 20 dez. 1996, Brasília, MEC, 1996.

MAGALHÃES, G.S., **Autoria em Hipermídia: o modelo OOHDM aplicado à gestão de eventos**; 2002. Monografia (Bacharelado em Ciência da Computação) - Universidade Federal de Lavras, MG.

MICROSOFT, **Site oficial da Microsoft**.

Disponível em: <<http://www.microsoft.com>>.

Acessado em outubro de 2001.

MUNIZ, M. C. M., **Proposta de um Framework de Gerenciamento Eletrônico de Documentos via Web**. Lavras: UFLA, MG, 2002. Monografia-Graduação em Ciência da Computação.

OLIVEIRA, T. M. P. ; **Interatividade na Educação a Distância**, 2001. Tese (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, SC

OLIVEIRA, R., Zambalde, A.L.; Alves, R.M.; Gomes, S.A.C, **Uso do modelo OOHDM para a construção de uma aplicação de ensino voltada para o setor agropecuário** - Revista Brasileira de Agroinformática, v.4, n.1, p.28-46, 2002.

RUGABER, S. **Program Comprehension for Reverse Engineering**. *In: AAAI Workshop on AI and Automated Program Understanding*, San Jose, California, July 1992.

RUMBAUGH,J.; BLAHA, M.; PREMELARNI,W.; EDDY, F.; LORENSEN, W, **Modelagem e Projetos baseados em objetos**; 1994 - Editora Campus.

SANTOS JUNIOR, R. M. ; **Implementação de um Modelo Informacional para Avaliações no Ensino a Distância via Web**; 2002. Monografia (Bacharelado em Ciência da Computação) - Universidade Federal de Lavras, Lavras, MG

SCHWABE, D., **Autoria em Hipermídia**. Versão preliminar. Rio de Janeiro, PUC, 1993.

VALENTE, J.A., **Plataforma para EAD via Internet: considerações técnicas e pedagógicas**, 2002 - Departamento Multimeios e NIED - Unicamp, CEd, PUC-SP.

YOURDON. **Análise Estruturada Moderna**. Editora Campus, 1998

ZAMBALDE, A. L., ALVES, R.M. e LOPES, M.A., **Modelagem, autoria e análise de usabilidade de aplicação hipermídia direcionada ao setor agropecuário**, UFLA, 1999.

ZAMBALDE, A.L.; ALVES, R.M., **Ensino a Distância via Internet - Conteúdo e Interface**, Lavras, MG, Junho, 2001.

ANEXO A: Diagramas de *use-cases*

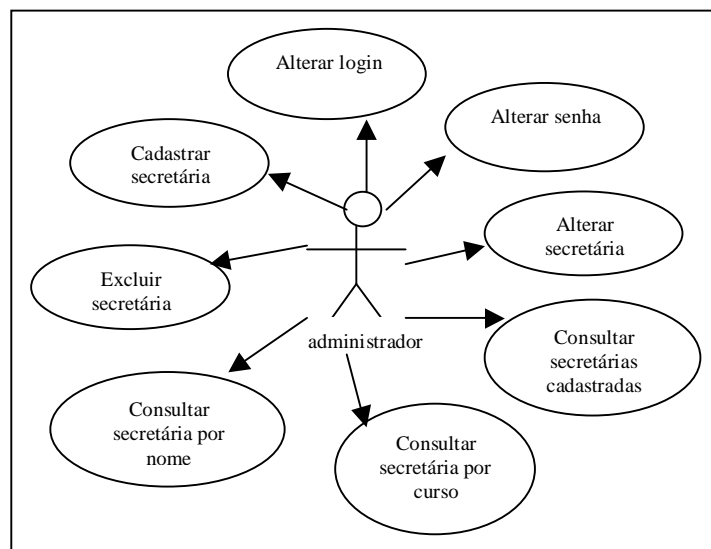


Diagrama de *use-case* secretária

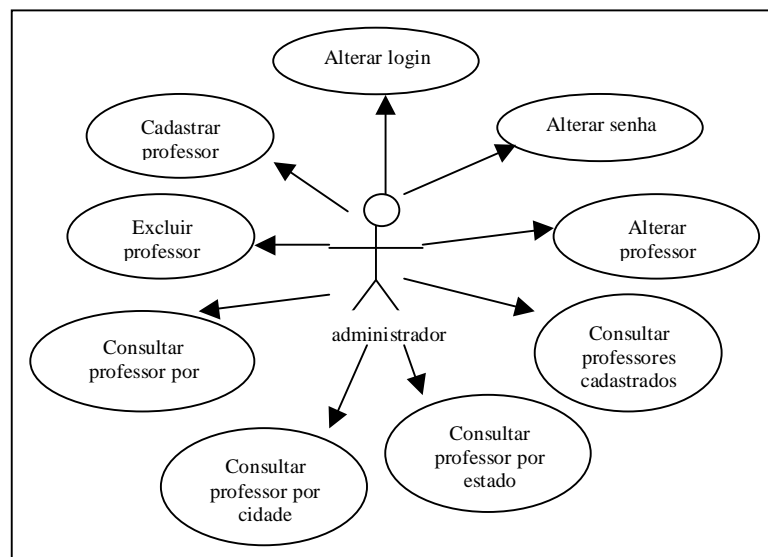


Diagrama de *use-case* professor

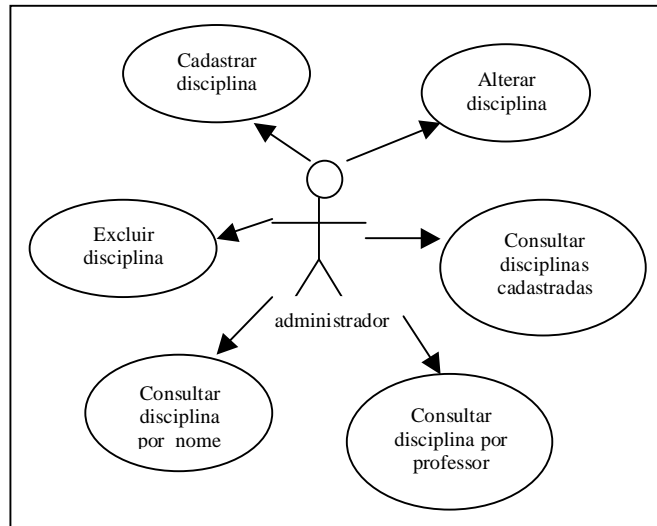


Diagrama *use-case* disciplina

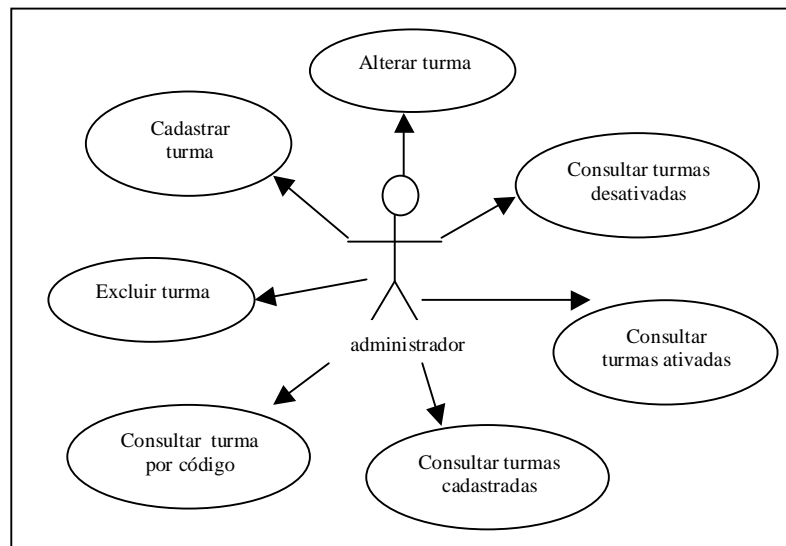


Diagrama *use-case* turma

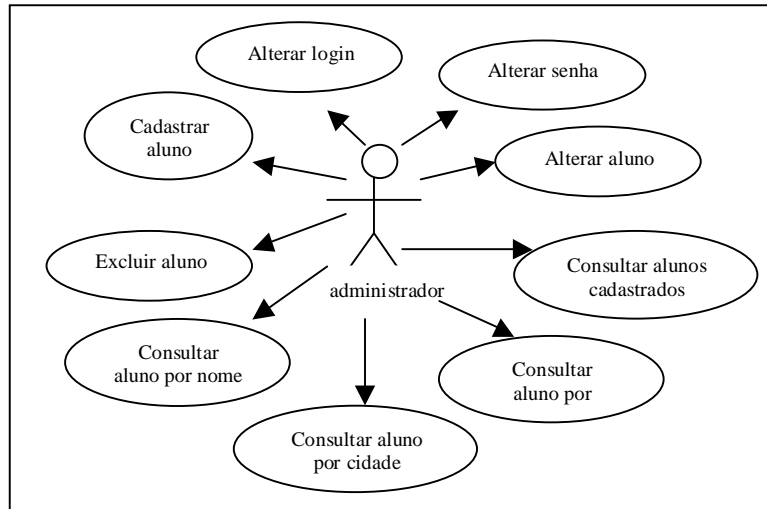


Diagrama de *use-case* aluno

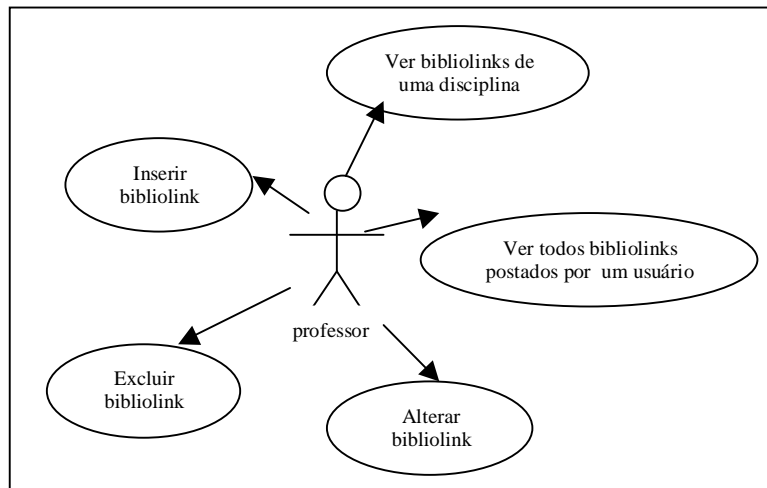


Diagrama de *use-case* bibliolink

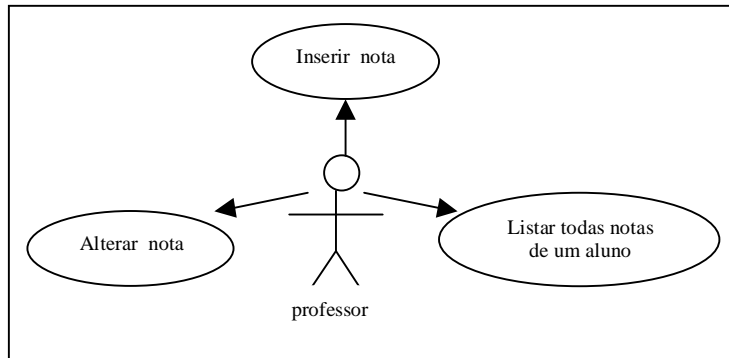


Diagrama de *use-case* nota

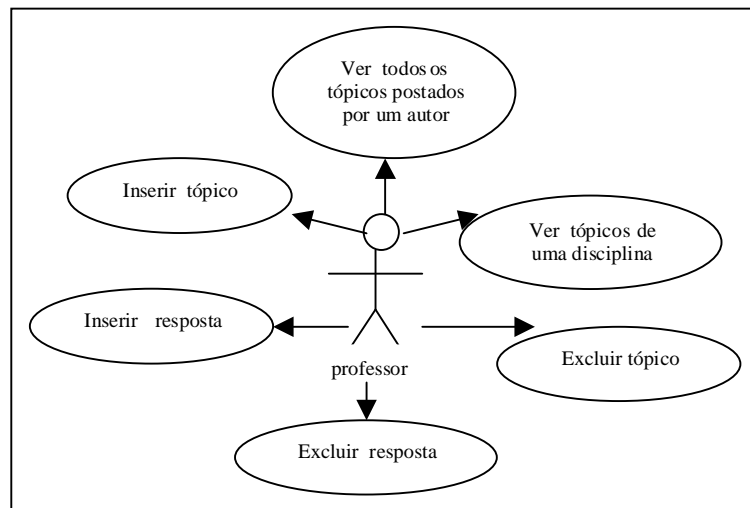


Diagrama de *use-case* fórum

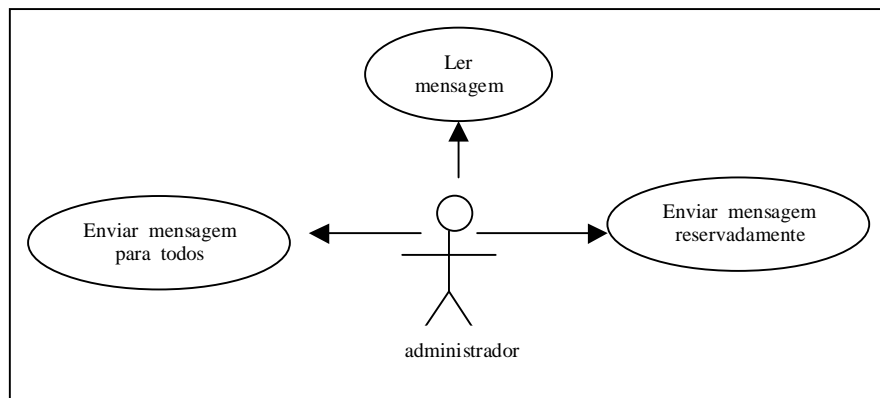


Diagrama de *use-case chat*

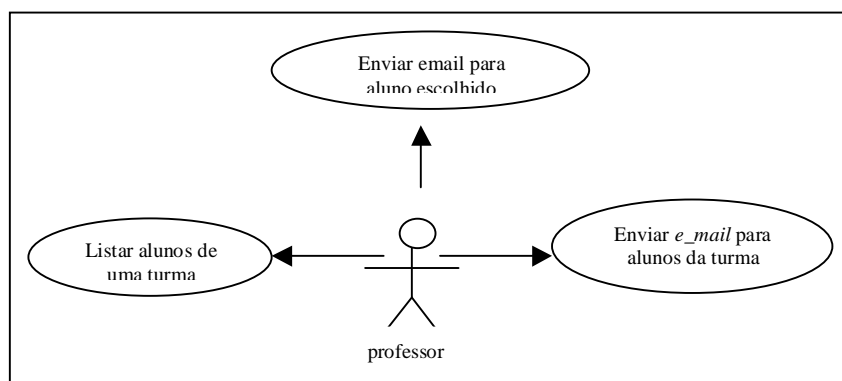


Diagrama de *use-case alunos*

ANEXO B: Descrição dos *use-case*

Use-case: Cadastrar curso

O administrador poderá cadastrar um novo curso no sistema.

Ator: Administrador

Entradas e pré condições: Digitar os dados do curso.

Saídas e pós condições: Curso incluído no banco de dados.

Sequência de operações:

- 1- O administrador digita todos os dados do curso;
- 2- O sistema verifica se o curso não está cadastrado;
- 3- Se ele estiver cadastrado (já foi cadastrado anteriormente), fim;
- 4- Caso contrário, o sistema insere os dados do curso na tabela de cursos.

Use-case: Consultar curso por código

O administrador verifica através do código de um curso, se este está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador

Entradas e pré condições: Digitar o código do curso.

Saídas e pós condições: Dados do curso exibidos.

Sequência de operações:

- 1- O administrador digita o código do curso no sistema;
- 2- O sistema procura o curso na tabela de Cursos;
- 3- Se o curso pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do curso.

Use-case: Consultar curso por coordenador

O administrador verifica através do nome do coordenador de um curso, se este está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador

Entradas e pré condições: Digitar o nome do coordenador do curso.

Saídas e pós condições: Dados do curso exibidos.

Sequência de operações:

- 1- O administrador digita o nome do coordenador do curso no sistema;
- 2- O sistema procura o curso na tabela de Cursos;
- 3- Se o curso pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do curso.

Use-case: Consultar curso por nome

O administrador verifica através do nome do curso, se este está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador

Entradas e pré condições: Digitar o nome do curso.

Saídas e pós condições: Dados do curso exibidos.

Sequência de operações:

- 1- O administrador digita o nome do curso no sistema;
- 2- O sistema procura o curso na tabela de Cursos;
- 3- Se o curso pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do curso.

Use-case: Consultar todos os cursos cadastrados

O administrador verifica quais os estão cursos cadastrados no sistema e retorna seus dados.

Ator: Administrador

Entradas e pré condições: Dados dos cursos disponíveis.

Saídas e pós condições: Dados dos cursos cadastrados exibidos.

Sequência de operações:

- 1- O administrador solicita ao sistema quais os cursos que se encontram armazenados no banco de dados;
- 2- O sistema procura os cursos inseridos na tabela de Cursos;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados de todos os cursos cadastrados.

<i>Use-case: Alterar dados do curso</i>
--

O administrador modifica os dados de um curso

Ator: Administrador

Entradas e pré condições: O curso deve ser cadastrado.

Saídas e pós condições: Dados do curso são atualizados.

Sequência de operações:

- 1- O administrador escolhe o curso a ser alterado;
- 2- O administrador digita os novos dados do cliente;
- 3- O sistema procura o curso na tabela de Cursos;
- 4- Se ele existir, o sistema envia os novos dados para o tabela de Cursos, atualizando o banco de dados;
- 5- Caso contrário, fim.

<i>Use-case: Excluir curso</i>

O Administrador remove um curso

Ator: Administrador

Entradas e pré condições: Escolher o código do curso.

Saídas e pós condições: Curso excluído.

Sequência de operações:

- 1- O administrador seleciona o código do curso a ser removido;
- 2- O sistema procura este curso na tabela de Cursos;
- 3- Se o curso existir, o sistema exclui este curso do banco de dados;
- 4- Caso contrário, fim.

<i>Use-case: Cadastrar secretária</i>
--

O administrador poderá cadastrar uma nova secretária no sistema.

Ator: Administrador

Entradas e pré condições: Cursos cadastrados sem secretária e fornecer os dados da secretária.

Saídas e pós condições: Secretária incluída no banco de dados.

Sequência de operações:

- 1- O administrador digita todos os dados da secretária;
- 2- O sistema verifica se existem cursos cadastrados sem secretária;
- 3- Se existir, o sistema disponibiliza os códigos destes cursos;
 - 3.1- O administrador escolhe a qual curso associar a secretária;
 - 3.2- O sistema inclui a nova secretária na tabela de Secretárias do banco de dados.
- 4- Caso contrário, fim.

<i>Use-case: Consultar curso por nome</i>
--

O administrador verifica através do nome de uma secretária, se esta está cadastrada no sistema, em caso positivo, retorna seus dados.

Ator: Administrador

Entradas e pré condições: Digitar o nome da secretária.

Saídas e pós condições: Dados da secretária exibidos.

Sequência de operações:

- 1- O administrador digita o nome da secretária no sistema;
- 2- O sistema procura a secretária na tabela de Secretárias;
- 3- Se a secretária pesquisada não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados da secretária.

Use-case: Consultar secretária por curso

O administrador verifica através do nome do curso se a secretária está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador

Entradas e pré condições: Digitar o nome do curso.

Saídas e pós condições: Dados da secretária exibidos.

Sequência de operações:

- 1- O administrador digita o nome do curso no sistema;
- 2- O sistema procura o curso indicado;
- 3- Se o curso pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados da secretária relacionados ao curso informado.

Use-case: Consultar todas as secretárias cadastradas

O administrador verifica quais as secretárias que estão cadastradas no sistema e retorna seus dados.

Ator: Administrador

Entradas e pré condições: Dados das secretárias disponíveis.

Saídas e pós condições: Dados das secretárias cadastradas exibidos.

Sequência de operações:

- 1- O administrador solicita ao sistema todas as secretárias armazenadas no banco de dados;

- 2- O sistema procura as secretárias inseridas na tabela de Secretárias;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados de todas as secretárias cadastradas.

Use-case: Alterar login da secretária

O administrador modifica o login de uma secretária

Ator: Administrador

Entradas e pré condições: A secretária deve ser cadastrada e deve ser fornecido o login.

Saídas e pós condições: Login da secretária será atualizado.

Sequência de operações:

- 1- O administrador digita o login da secretária a ser alterado;
- 2- O administrador digita o novo login;
- 3- O sistema procura o login informado na tabela de Secretárias;
- 4- Se ele existir, o sistema envia o novo login para o tabela de Secretárias, atualizando o banco de dados;
- 5- Caso contrário, fim.

Use-case: Alterar senha da secretária

O administrador modifica a senha de uma secretária

Ator: Administrador

Entradas e pré condições: A secretária deve ser cadastrada e deve ser fornecida a senha.

Saídas e pós condições: Senha da secretária será atualizada.

Sequência de operações:

- 1- O administrador digita a senha da secretária a ser alterada;
- 2- O administrador digita a nova senha e a confirma redigitando-a;
- 3- O sistema procura a senha informada na tabela de Secretárias;

- 4- Se ele existir, o sistema envia a nova senha para o tabela de Secretárias, atualizando o banco de dados;
- 5- Caso contrário, fim.

<i>Use-case: Alterar dados da secretária</i>

O administrador modifica os dados de uma secretária

Ator: Administrador

Entradas e pré condições: A secretária deve ser cadastrada.

Saídas e pós condições: Dados da secretária serão atualizados.

Sequência de operações:

- 1- O administrador escolhe o código da secretária a ser alterada;
- 2- O sistema verifica se a secretária indicada pode ser alterada (caso ainda existam cursos sem secretária);
- 3- Se for possível a alteração, o administrador fornece os novos dados da secretária, atualizando-a no banco de dados;
- 4- Caso contrário, fim.

<i>Use-case: Excluir secretária</i>
--

O Administrador remove uma secretária.

Ator: Administrador

Entradas e pré condições: Escolher o código da secretária.

Saídas e pós condições: Secretária excluída.

Sequência de operações:

- 1- O administrador seleciona o código da secretária a ser removida;
- 2- O sistema procura esta secretária na tabela de Secretárias;
- 3- Se a secretária existir, o sistema exclui esta secretária do banco de dados;
- 4- Caso contrário, fim.

Use-case: Cadastrar professor

O administrador ou a secretária poderá cadastrar um novo professor para determinado curso no sistema.

Ator: Administrador e Secretária

Entradas e pré condições: Curso estar cadastrado e fornecer os dados do professor.

Saídas e pós condições: Professor incluído no banco de dados.

Sequência de operações:

- 1- O administrador ou a secretária digita todos os dados do professor;
- 2- O sistema verifica se o curso está cadastrado;
- 3- Se estiver, o sistema inclui o novo professor na tabela de Professores do banco de dados com os respectivos dados.
- 4- Caso contrário, fim.

Use-case: Consultar professor por nome

O administrador ou a secretária verifica através do nome de um professor, se este está cadastrada no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o nome do professor.

Saídas e pós condições: Dados do professor exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o nome do professor no sistema;
- 2- O sistema procura o professor na tabela de Professores;
- 3- Se o professor pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do professor.

Use-case: Consultar professor por cidade

O administrador ou a secretária verifica através do nome de uma cidade se professor está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o nome da cidade.

Saídas e pós condições: Dados do professor exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o nome da cidade no sistema;
- 2- O sistema procura o professor correspondente;
- 3- Se o professor pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do professor relacionados à cidade informada.

Use-case: Consultar professor por estado

O administrador ou a secretária verifica através do nome de um estado se professor está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o nome do estado.

Saídas e pós condições: Dados do professor exibidos.

Sequência de operações:

- 1- O administrador ou a secretária seleciona o nome do estado no sistema;
- 2- O sistema procura o professor correspondente;
- 3- Se o professor pesquisado não existir na tabela ou se a tabela estiver vazia, fim;

4- Caso contrário, o sistema exibirá os dados do professor relacionados ao estado informado.

Use-case: Consultar todos os professores cadastrados

O administrador ou a secretária verifica quais os professores que estão cadastrados no sistema e retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Dados dos professores disponíveis.

Saídas e pós condições: Dados dos professores cadastrados exibidos.

Sequência de operações:

- 1- O administrador ou a secretária solicita ao sistema todos os professores armazenados no banco de dados;
- 2- O sistema procura os professores inseridos na tabela de Professores;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados de todos os professores cadastrados.

Use-case: Alterar login do professor

O administrador ou a secretária modifica o login de um professor

Ator: Administrador ou Secretária

Entradas e pré condições: O professor deve ser cadastrado e deve ser fornecido o login.

Saídas e pós condições: Login do professor será atualizado.

Sequência de operações:

- 1- O administrador ou a secretária digita o login do professor a ser alterado;
- 2- O administrador ou a secretária digita o novo login;
- 3- O sistema procura o login informado na tabela de Professores;

- 4- Se ele existir, o sistema envia o novo login para o tabela de Professores, atualizando o banco de dados;
- 5- Caso contrário, fim.

<i>Use-case: Alterar senha do professor</i>
--

O administrador ou a secretária modifica a senha de um professor

Ator: Administrador ou Secretária

Entradas e pré condições: O professor deve ser cadastrado e deve ser fornecida a senha.

Saídas e pós condições: Senha do professor será atualizada.

Sequência de operações:

- 1- O administrador ou a secretária digita a senha do professor a ser alterado;
- 2- O administrador ou a secretária digita a nova senha e a confirma redigitando-a;
- 3- O sistema procura a senha informada na tabela de Professores;
- 4- Se ele existir, o sistema envia a nova senha para o tabela de Professores, atualizando o banco de dados;
- 5- Caso contrário, fim.

<i>Use-case: Alterar dados do professor</i>
--

O administrador ou a secretária modifica os dados de um professor

Ator: Administrador ou Secretária

Entradas e pré condições: O professor deve ser cadastrado.

Saídas e pós condições: Dados do professor serão atualizados.

Sequência de operações:

- 1- O administrador ou a secretária escolhe o código do professor a ser alterado;
- 2- O sistema verifica se o professor está cadastrado;

- 3- Em caso afirmativo, o administrador ou a secretária fornece os novos dados do professor a serem alterados;
- 3.1- O sistema atualiza o banco de dados com as novas informações referentes ao professor informado;
- 4- Caso contrário, fim.

Use-case: Excluir professor

O Administrador ou a secretária remove um professor.

Ator: Administrador ou Secretária

Entradas e pré condições: Escolher o código do professor.

Saídas e pós condições: Professor excluído.

Sequência de operações:

- 1- O administrador ou a secretária seleciona o código do professor a ser removido;
- 2- O sistema procura este professor na tabela de Professores;
- 3- Se o professor existir, o sistema exclui este professor do banco de dados;
- 4- Caso contrário, fim.

Use-case: Cadastrar disciplina

O administrador ou a secretária poderá cadastrar uma nova disciplina para determinado curso no sistema.

Ator: Administrador e Secretária

Entradas e pré condições: Curso e professor devem estar cadastrados e fornecer os dados da disciplina.

Saídas e pós condições: Disciplina incluída no banco de dados.

Sequência de operações:

- 1- O administrador ou a secretária fornece os dados da disciplina;
- 2- O sistema verifica se o curso e o professor estão cadastrados;

3- Se estiverem, o sistema inclui a nova disciplina na tabela de Disciplinas do banco de dados com os respectivos dados.

4- Caso contrário, fim.

Use-case: Consultar disciplina por nome

O administrador ou a secretária verifica através do nome de uma disciplina, se esta está cadastrada no sistema, em caso positivo, retoma seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o nome da disciplina.

Saídas e pós condições: Dados da disciplina exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o nome da disciplina no sistema;
- 2- O sistema procura a disciplina na tabela de Disciplinas;
- 3- Se a disciplina pesquisada não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados da disciplina.

Use-case: Consultar disciplina por professor

O administrador ou a secretária verifica através do nome de um professor se a disciplina está cadastrada no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o nome do professor.

Saídas e pós condições: Dados da disciplina exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o nome do professor no sistema;
- 2- O sistema procura a disciplina correspondente;

3- Se a disciplina pesquisada não existir na tabela ou se a tabela estiver vazia, fim;

4- Caso contrário, o sistema exibirá os dados da disciplina relacionados ao professor informado.

Use-case: Consultar todas as disciplinas cadastradas

O administrador ou a secretária verifica quais as disciplinas que estão cadastradas no sistema e retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Fornecer o código de um curso e disciplinas devem estar cadastradas.

Saídas e pós condições: Dados das disciplinas cadastradas exibidos.

Sequência de operações:

- 1- O administrador ou a secretária solicita ao sistema todas as disciplinas armazenadas no banco de dados;
- 2- O sistema procura as disciplinas inseridas na tabela de Disciplinas;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados de todas as professoras cadastradas.

Use-case: Alterar dados da disciplina

O administrador ou a secretária modifica os dados de uma disciplina.

Ator: Administrador ou Secretária

Entradas e pré condições: O curso e a disciplina devem ser cadastrados.

Saídas e pós condições: Dados da disciplina serão atualizados.

Sequência de operações:

- 1- O administrador ou a secretária escolhe o nome da disciplina a ser alterada;
- 2- O sistema verifica se a está disciplina cadastrada;

- 3- Em caso afirmativo, o administrador ou a secretária fornece os novos dados da disciplina a serem alterados;
- 3.1- O sistema atualiza o banco de dados com as novas informações referentes a disciplina informada;
- 4- Caso contrário, fim.

Use-case: Excluir disciplina

O Administrador ou a secretária remove uma disciplina.

Ator: Administrador ou Secretária

Entradas e pré condições: Escolher o nome da disciplina.

Saídas e pós condições: Disciplina excluída.

Sequência de operações:

- 1- O administrador ou a secretária seleciona o nome da disciplina a ser removida;
- 2- O sistema procura esta disciplina na tabela de Disciplinas;
- 3- Se a disciplina existir, o sistema exclui esta disciplina do banco de dados;
- 4- Caso contrário, fim.

Use-case: Cadastrar turma

O administrador ou a secretária poderá cadastrar uma nova turma para determinado curso no sistema.

Ator: Administrador e Secretária

Entradas e pré condições: Curso deve estar cadastrado e fornecer os dados da turma.

Saídas e pós condições: Turma incluída no banco de dados.

Sequência de operações:

- 1- O administrador ou a secretária fornece os dados da turma;
- 2- O sistema verifica se o curso está cadastrados;

3- Se estiver, o sistema inclui a nova turma na tabela de Turmas do banco de dados com os respectivos dados.

4- Caso contrário, fim.

Use-case: Consultar turma por código

O administrador ou a secretária verifica através do código de uma disciplina, se esta está cadastrada no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o código da turma.

Saídas e pós condições: Dados da turma exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o código da turma no sistema;
- 2- O sistema procura a turma na tabela de Turmas;
- 3- Se a turma pesquisada não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados da turma.

Use-case: Consultar turmas ativadas

O administrador ou a secretária verifica quais das turmas cadastradas no sistema que estão ativadas, retornando-as.

Ator: Administrador ou Secretária

Entradas e pré condições: O curso e a turma estarem cadastrados.

Saídas e pós condições: Dados da turma exibidos.

Sequência de operações:

- 1- O administrador ou a secretária pesquisa as turmas que estão ativadas no sistema;
- 2- O sistema procura a turma correspondente ao código do curso;
- 3- Se a tabela não possuir turmas ativadas ou estiver vazia, fim;

4- Caso contrário, o sistema exibirá os dados destas turmas.

Use-case: Consultar turmas desativadas

O administrador ou a secretária verifica através quais turmas cadastradas no sistema estão desativadas, retornando seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: O curso e a turma estarem cadastrados.

Saídas e pós condições: Dados da turma exibidos.

Sequência de operações:

- 1- O administrador ou a secretária pesquisa as turmas que estão desativadas no sistema;
- 2- O sistema procura a turma correspondente ao código do curso;
- 3- Se a tabela não possuir turmas desativadas ou estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados destas turmas.

Use-case: Consultar todas as turmas cadastradas

O administrador ou a secretária verifica quais as turmas que estão cadastradas no sistema e retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Fornecer o código de um curso e as turmas devem estar cadastradas.

Saídas e pós condições: Dados das turmas cadastradas exibidos.

Sequência de operações:

- 1- O administrador ou a secretária solicita ao sistema todas as turmas armazenadas no banco de dados;
- 2- O sistema procura as turmas inseridas na tabela de Turmas;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados de todas as turmas cadastradas.

Use-case: Alterar dados da turma

O administrador ou a secretária modifica os dados de uma disciplina.

Ator: Administrador ou Secretária

Entradas e pré condições: O curso e a disciplina devem ser cadastrados.

Saídas e pós condições: Dados da disciplina serão atualizados.

Sequência de operações:

- 1- O administrador ou a secretária escolhem o código da turma a ser alterada;
- 2- O sistema verifica se a turma está cadastrada;
- 3- Em caso afirmativo, o administrador ou a secretária fornece os novos dados da turma a serem alterados;
- 3.1- O sistema atualiza o banco de dados com as novas informações referentes a turma informada;
- 4- Caso contrário, fim.

Use-case: Excluir turma

O Administrador ou a secretária remove uma turma.

Ator: Administrador ou Secretária

Entradas e pré condições: A turma deve estar cadastrada e escolher o nome da turma.

Saídas e pós condições: Turma excluída.

Sequência de operações:

- 1- O administrador ou a secretária seleciona o nome da turma a ser removida;
- 2- O sistema procura esta disciplina na tabela de Disciplinas;
- 3- Se a disciplina existir, o sistema exclui esta disciplina do banco de dados;
- 4- Caso contrário, fim.

Use-case: Cadastrar aluno

O administrador ou a secretária poderá cadastrar um novo aluno no sistema.

Ator: Administrador e Secretária

Entradas e pré condições: Curso e turma devem estar cadastrados e fornecer os dados do aluno.

Saídas e pós condições: Aluno incluído no banco de dados.

Sequência de operações:

- 1- O administrador ou a secretária fornece os dados do aluno;
- 2- O sistema verifica se o curso e a turma estão cadastrados;
- 3- Se estiverem, o sistema inclui o novo aluno na tabela de Alunos do banco de dados com seus respectivos dados.
- 4- Caso contrário, fim.

Use-case: Consultar aluno por nome

O administrador ou a secretária verifica através do nome de um aluno, se este está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o nome do aluno.

Saídas e pós condições: Dados do aluno exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o nome do aluno no sistema;
- 2- O sistema procura o aluno na tabela de Alunos;
- 3- Se o aluno pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do aluno.

Use-case: Consultar aluno por cidade

O administrador ou a secretária verifica através do nome de uma cidade, se um aluno está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o nome de uma cidade.

Saídas e pós condições: Dados do aluno exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o nome da cidade no sistema;
- 2- O sistema procura o aluno correspondente;
- 3- Se o aluno pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do aluno.

Use-case: Consultar aluno por estado

O administrador ou a secretária verifica através do nome de um estado, se um aluno está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Escolher o nome de um estado.

Saídas e pós condições: Dados do aluno exibidos.

Sequência de operações:

- 1- O administrador ou a secretária escolhe um estado;
- 2- O sistema procura o aluno correspondente;
- 3- Se o aluno pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do aluno.

Use-case: Consultar aluno por turma

O administrador ou a secretária verifica através do código de uma turma, se um aluno está cadastrado no sistema, em caso positivo, retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Digitar o código da turma.

Saídas e pós condições: Dados do aluno exibidos.

Sequência de operações:

- 1- O administrador ou a secretária digita o código da turma;
- 2- O sistema procura o aluno correspondente;
- 3- Se o aluno pesquisado não existir na tabela ou se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados do aluno.

<i>Use-case: Consultar todos os alunos cadastrados</i>

O administrador ou a secretária verifica quais os alunos que estão cadastradas no sistema e retorna seus dados.

Ator: Administrador ou Secretária

Entradas e pré condições: Fornecer o código de um curso.

Saídas e pós condições: Dados dos alunos cadastrados exibidos.

Sequência de operações:

- 1- O administrador ou a secretária solicita ao sistema todos os alunos de um curso;
- 2- O sistema procura os alunos inseridos na tabela de Alunos;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá os dados de todos os alunos cadastrados.

<i>Use-case: Alterar login do aluno</i>
--

O administrador ou a secretária modifica o login de um aluno.

Ator: Administrador ou Secretária

Entradas e pré condições: O aluno deve ser cadastrado e deve ser fornecida o novo login.

Saídas e pós condições: Login do aluno será atualizado.

Sequência de operações:

- 1- O administrador ou a secretária digita o login do aluno a ser alterado;
- 2- O administrador ou a secretária digita o novo login;
- 3- O sistema procura o login informado na tabela de Professores;
- 4- Se ele existir, o sistema envia o novo login para o tabela de Professores, atualizando o banco de dados;

Caso contrário, fim.

<i>Use-case: Alterar senha do aluno</i>
--

O administrador ou a secretária modifica a senha de um aluno.

Ator: Administrador ou Secretária

Entradas e pré condições: O aluno deve ser cadastrado e deve ser fornecida a senha.

Saídas e pós condições: Senha do aluno será atualizada.

Sequência de operações:

- 1- O administrador ou a secretária digita a senha do aluno a ser alterado;
- 2- O administrador ou a secretária digita a nova senha e a confirma redigitando-a;
- 3- O sistema procura a senha informada na tabela de Alunos;
- 4- Se ele existir, o sistema envia a nova senha para o tabela de Turmas, atualizando o banco de dados;
- 5- Caso contrário, fim.

<i>Use-case: Alterar dados do aluno</i>
--

O administrador ou a secretária modifica os dados de um aluno.

Ator: Administrador ou Secretária

Entradas e pré condições: O aluno deve estar cadastrado.

Saídas e pós condições: Dados do aluno serão atualizados.

Sequência de operações:

1. O administrador ou a secretária escolhem o código do aluno a ser alterado;
2. O sistema verifica se o aluno está cadastrado;
3. Em caso afirmativo, o administrador ou a secretária fornece os novos dados do aluno a serem alterados;
- 3.1- O sistema atualiza o banco de dados com as novas informações referentes ao aluno informado;
- 4- Caso contrário, fim.

<i>Use-case: Excluir aluno</i>

O Administrador ou a secretária remove um aluno.

Ator: Administrador ou Secretária

Entradas e pré condições: O aluno deve estar cadastrado e escolher o código do aluno.

Saídas e pós condições: Aluno excluído.

Sequência de operações:

- 1- O administrador ou a secretária seleciona o código do curso a ser removido;
- 2- O sistema procura este aluno na tabela de Alunos;
- 3- Se o aluno existir, o sistema exclui este aluno do banco de dados;
- 4- Caso contrário, fim.

<i>Use-case: Inserir bibliolink</i>
--

O professor ou o aluno insere um bibliolink (link ou bibliografia) em uma disciplina.

Ator: Professor ou aluno

Entradas e pré condições: A disciplina deve estar cadastrada e os dados do bibliolink devem ser fornecidos.

Saídas e pós condições: Bibliolink inserido em determinada disciplina.

Sequência de operações:

- 1- O professor ou o aluno seleciona o código da disciplina que irá inserir o bibliolink;
- 2- O sistema procura esta disciplina na tabela de Disciplinas;
- 3- Se a disciplina existir,
 - 3.1- O professor ou o aluno escolhe o tipo do bibliolink a inserir;
 - 3.2- Se escolher link digita todos os dados do link;
 - 3.3 - Caso escolha bibliografia digita todos os dados da bibliografia;
 - 3.4- O sistema inclui o novo bibliolink na tabela de Bibliolinks do banco de dados;
- 1- Caso contrário, fim

<i>Use-case: Listar bibliolinks de uma disciplina</i>

O professor ou o aluno consulta todos os bibliolinks inseridos em uma disciplina e retorna seus dados.

Ator: Professor ou aluno

Entradas e pré condições: A disciplina deve estar cadastrada e os dados dos bibliolinks devem estar disponíveis.

Saídas e pós condições: Dados dos bibliolinks de determinada disciplina exibidos.

Sequência de operações:

- 1- O professor ou o aluno seleciona o código da disciplina que irá consultar os bibliolinks;
- 2- O sistema procura esta disciplina na tabela de Disciplinas;
- 3- Se a disciplina existir,
 - 3.1- O professor ou o aluno solicita ao sistema todos os bibliolinks daquela disciplina;
 - 3.2 - O sistema procura os bibliolink inseridos na tabela de Bibliolinks;
 - 3.3- Se a tabela estiver vazia, fim;

3.4- Caso contrário, o sistema exibirá todos os dados dos links e das bibliografias relacionados à disciplina solicitada;

4- Caso contrário, fim.

Use-case: Listar bibliolinks cadastrados

O professor ou o aluno consulta todos os bibliolinks inseridos e retorna seus dados.

Ator: Professor ou aluno

Entradas e pré condições: Os dados dos bibliolinks devem estar disponíveis.

Saídas e pós condições: Dados dos bibliolinks exibidos.

Sequência de operações:

- 1- O professor ou o aluno solicita ao sistema todos os bibliolinks armazenados em um banco de dados;
- 2- O sistema procura os bibliolinks inseridos na tabela de Bibliolinks;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá todos os dados dos links e das bibliografias cadastradas;

Use-case: Alterar bibliolink

O professor modifica os dados de um bibliolink.

Ator: Professor ou aluno

Entradas e pré condições: O bibliolink deve estar cadastrado e os novos dados devem ser fornecidos.

Saídas e pós condições: Dados do bibliolink serão atualizados.

Sequência de operações:

- 1- O professor escolhe o código do bibliolink a ser alterado;
- 2- O sistema procura este bibliolink na tabela de Bibliolinks;
- 3- Se o bibliolink existir,

3.1- O professor digita os novos dados do bibliolink, atualizando-o no banco de dados;

4- Caso contrário, fim.

Use-case: Excluir bibliolink

O professor remove um bibliolink.

Ator: Professor

Entradas e pré condições: O bibliolink deve estar cadastrado e selecionar o código do bibliolink.

Saídas e pós condições: Bibliolink excluído.

Sequência de operações:

- 1- O professor seleciona o código do bibliolink a ser removido;
- 2- O sistema procura este bibliolink na tabela de Bibliolinks;
- 3- Se o bibliolink existir, o sistema exclui este bibliolink do banco de dados;
- 4- Caso contrário, fim.

Use-case: Inserir notas

O professor poderá inserir as notas de determinada disciplina de todos os alunos de uma turma.

Ator: Professor

Entradas e pré condições: Disciplina deve estar cadastrada, alunos devem estar cadastrados em uma turma e fornecer notas dos alunos.

Saídas e pós condições: Notas dos alunos incluídas no banco de dados.

Sequência de operações:

- 1- O professor seleciona o código da disciplina na qual irá inserir as notas;
- 2- O sistema procura a disciplina na tabela de Disciplinas;
- 3- Caso a disciplina exista,

- 3.1- O professor digita as notas de todos os alunos em frente de seus nomes e códigos de identificação;
- 3.2- O sistema faz a conversão das notas se necessário;
- 3.3- O sistema inclui as notas dos alunos na tabela de Notas do banco de dados;
- 4- Caso contrário, fim.

Use-case: Listar notas

O professor consulta as notas de um aluno ou o aluno lista todas as suas notas

Ator: Professor ou aluno

Entradas e pré condições: O aluno e suas respectivas notas devem estar cadastrados e as notas devem estar disponíveis.

Saídas e pós condições: Notas do aluno exibidas.

Sequência de operações:

- 1- O professor solicita ao sistema todas as notas de uma aluno armazenadas no banco de dados ou o aluno solicita ao sistema todas as suas notas armazenadas no banco de dados;
- 2- O sistema procura as notas do aluno inseridas na tabela de Notas;
- 3- Se a tabela estiver vazia, fim;
- 4- Caso contrário, o sistema exibirá todas os notas relativas ao aluno informado;

Use-case: Alterar nota

O professor poderá modificar a nota de uma disciplina de determinado aluno de uma turma, caso seja necessário.

Ator: Professor

Entradas e pré condições: Disciplina e nota cadastradas e fornecer nova nota.

Saídas e pós condições: Nota do aluno será atualizada.

Sequência de operações:

- 1- O professor seleciona o código da disciplina que irá alterar a nota;

- 2- O sistema procura esta disciplina na tabela de Disciplinas;
- 3- Se a disciplina existir,
 - 3.1- O professor digita a nova nota na frente do nome do aluno e de seu código de identificação, atualizando o banco de dados;
- 4- Caso contrário, fim.

Use-case: Listar alunos

O Professor ou o aluno consulta todos os dados dos alunos de uma turma, podendo enviar *e-mail* para determinado aluno ou para a turma inteira.

Ator: Professor ou aluno

Entradas e pré condições: Alunos devem estar cadastrados em uma turma e os dados dos alunos devem estar disponíveis.

Saídas e pós condições: Dados dos alunos de uma turma exibidos e *e-mail* enviado.

Sequência de operações:

- 1- O professor ou o aluno solicita ao sistema os alunos cadastrados em uma turma;
- 2- O sistema procura todos os alunos inseridos na referida turma;
- 3- Se a turma existir,
- 4- O sistema exibirá os dados de todos os alunos da turma informada;
- 5 - O professor o aluno posta um *e-mail* para determinado aluno ou para todos os alunos de uma turma, se houver necessidade;
- 5.1- O sistema direciona o *e-mail* ao(s) seu(s) destinatário(s).

Use-case: Listar fórum

O professor ou o aluno poderá visualizar o fórum de discussão de determinada disciplina.

Ator: Professor e aluno

Entradas e pré condições: Disciplina estar cadastrada e dados do fórum disponíveis.

Saídas e pós condições: Dados do fórum de discussão de uma disciplina exibidos.

Sequência de operações:

- 1- O professor seleciona o código da disciplina que retornará o fórum correspondente;
- 2- O sistema procura esta disciplina na tabela de Disciplinas;
- 3- Se a disciplina existir,
 - 3.1- O sistema exibe os dados (tópicos e respostas) inseridos no fórum desta disciplina;
- 4- Caso contrário, fim.

<i>Use-case:</i> Ler tópico

O professor ou o aluno poderá visualizar um tópico de um fórum de discussão de determinada disciplina.

Ator: Professor e aluno

Entradas e pré condições: Tópico estar inserido e dados do tópico disponíveis.

Saídas e pós condições: Dados do tópico do fórum de discussão exibidos.

Sequência de operações:

- 1- O professor ou o aluno acessa o link para o tópico que deseja visualizar;
- 2- O sistema procura o tópico na tabela de Fóruns_disciplinas e respostas associadas a ele na tabela de Fóruns_respostas;
- 3- Se o tópico existir, o sistema exibe todos os dados do tópico informado;
 - 3.1- Se existirem respostas associadas, o sistema exibe todos os dados destas respostas ;
- 4- Caso contrário, fim.

Use-case: **Inserir tópico**

O professor ou o aluno poderá inserir um novo tópico no fórum de discussão de determinada disciplina.

Ator: Professor e aluno

Entradas e pré condições: Disciplina estar cadastrada e fornecer dados do tópico.

Saídas e pós condições: Tópico inserido no banco de dados.

Sequência de operações:

- 1- O professor seleciona o código da disciplina que retornará o fórum correspondente;
- 2- O sistema procura esta disciplina na tabela de Disciplinas;
- 3- Se a disciplina existir,
 - 3.1- O professor ou o aluno digita todos os dados do tópico;
 - 3.2- O sistema inclui o novo tópico na tabela de Fóruns_disciplinas do banco de dados;
- 4- Caso contrário, fim.

Use-case: **Responder tópico**

O professor ou o aluno poderá responder um tópico no fórum de discussão de determinada disciplina.

Ator: Professor e aluno

Entradas e pré condições: Tópico estar cadastrado e fornecer dados da resposta.

Saídas e pós condições: Resposta inserida no banco de dados.

Sequência de operações:

- 1- O professor ou o aluno digita os dados da resposta correspondentes a um tópico lido ;

2- O sistema insere a nova resposta na tabela de Fóruns_respostas do banco de dados;

3- O sistema exibe os dados da resposta inserida e do tópico correspondente.

Use-case: Excluir resposta

O professor poderá excluir um tópico do fórum de discussão de determinada disciplina, se considerar necessário.

Ator: Professor

Entradas e pré condições: Resposta estar cadastrada e escolher resposta.

Saídas e pós condições: Resposta removida.

Sequência de operações:

- 1- O professor seleciona resposta que deseja excluir;
- 2- O sistema procura esta resposta na tabela de Fórum_respostas;
- 3- Se a resposta existir, o sistema exclui esta resposta do banco de dados;
- 4- Caso contrário, fim.

Use-case: Excluir tópico

O professor poderá excluir um tópico do fórum de discussão de determinada disciplina, se considerar necessário.

Ator: Professor

Entradas e pré condições: Tópico estar cadastrado e escolher dados do tópico.

Saídas e pós condições: Tópico removido.

Sequência de operações:

- 1- O professor seleciona o tópico que deseja excluir;
- 2- O sistema procura este tópico na tabela Fóruns_disciplinas;
- 3- Se o tópico existir, o sistema exclui o tópico e as respostas associadas a ele do banco de dados.
- 5- Caso contrário, fim.

Use-case: ler mensagem

O professor ou o aluno visualiza mensagens enviadas na sala de reuniões

Ator: Professor ou aluno

Entradas e pré condições: Professor ou aluno estar logado no *chat*.

Saídas e pós condições: Mensagem visualizada pelo professor ou aluno.

Sequência de operações:

- 1- O professor ou o aluno acessam a sala do *chat*;
- 2- O sistema cria novo usuário com seu respectivo login;
- 3- O sistema disponibiliza na tela as mensagens enviadas durante a sessão.

Use-case: responder mensagem

O professor ou o aluno responde mensagens enviadas na sala de reuniões

Ator: Professor ou aluno

Entradas e pré condições: Professor ou aluno estar logado no *chat*.

Saídas e pós condições: Mensagem respondida pelo professor ou aluno.

Sequência de operações:

- 1- O professor ou o aluno visualiza as mensagens enviadas durante sessão;
- 2- O professor ou o aluno responde a uma mensagem;
- 3- O sistema disponibiliza na tela a resposta enviada pelo professor ou aluno.

Use-case: responder reservadamente

O professor ou o aluno responde mensagens reservadamente a determinado usuário na sala de reuniões

Ator: Professor ou aluno

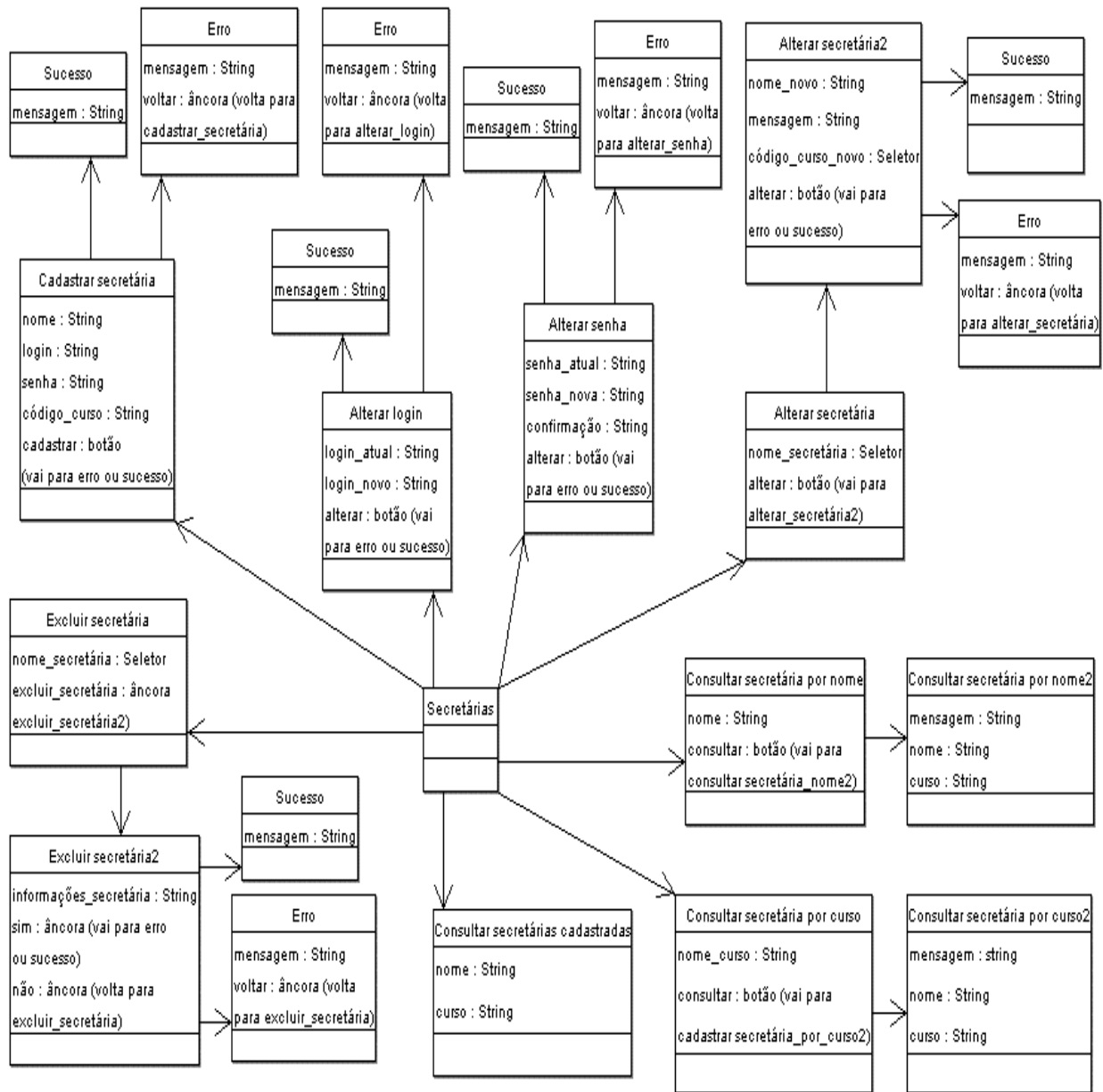
Entradas e pré condições: Professor ou aluno estar logado no *chat*.

Saídas e pós condições: Mensagem enviada reservadamente pelo professor ou aluno.

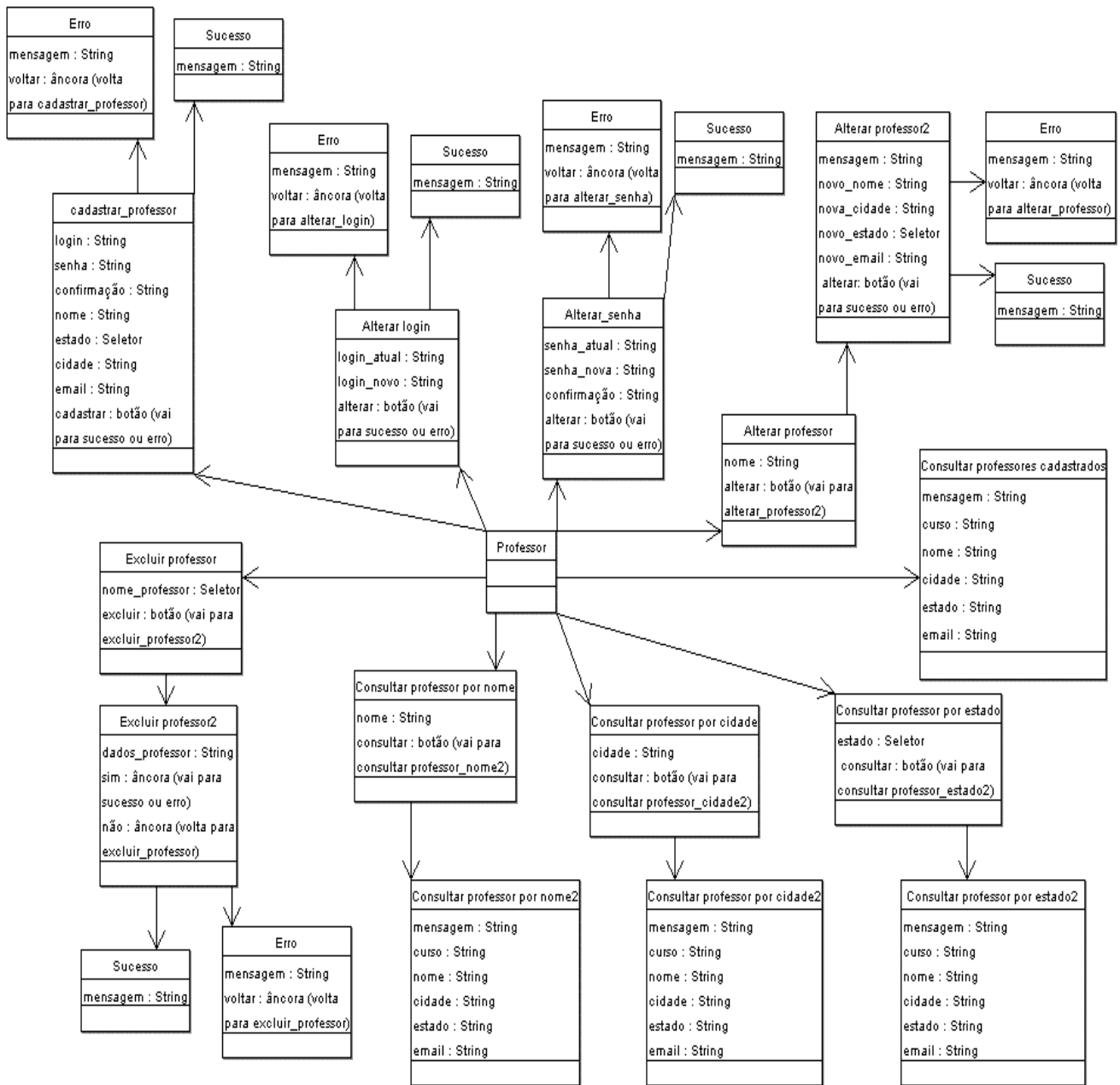
Sequência de operações:

- 1- O professor ou o aluno visualiza as mensagens enviadas durante sessão;
- 2- O professor ou o aluno escolhe determinado usuário do *chat* e lhe responde reservadamente uma mensagem;
- 3- O sistema disponibiliza na tela a resposta enviada somente para o destinatário e para o remetente.

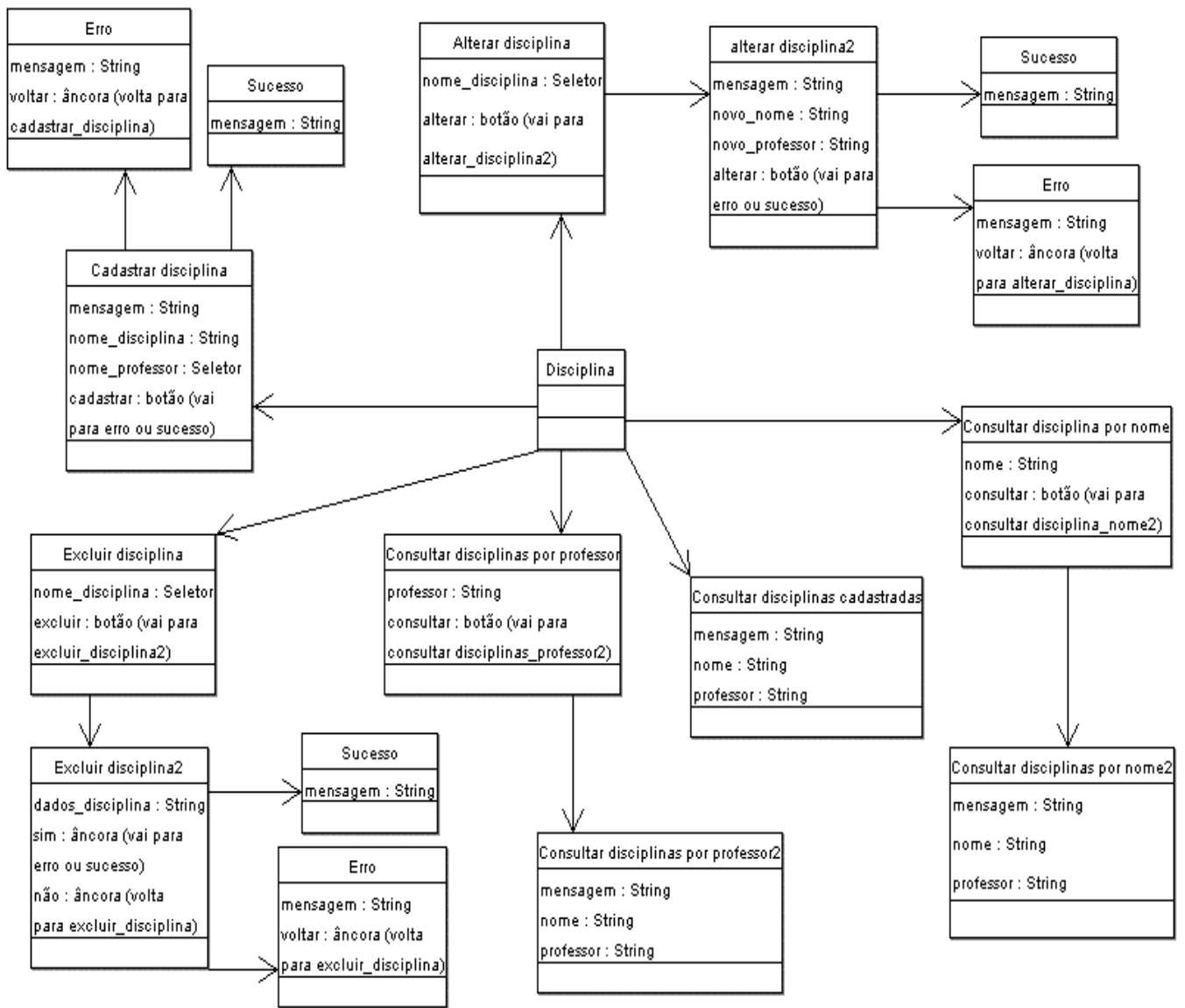
ANEXO C: Modelagem OOHDM



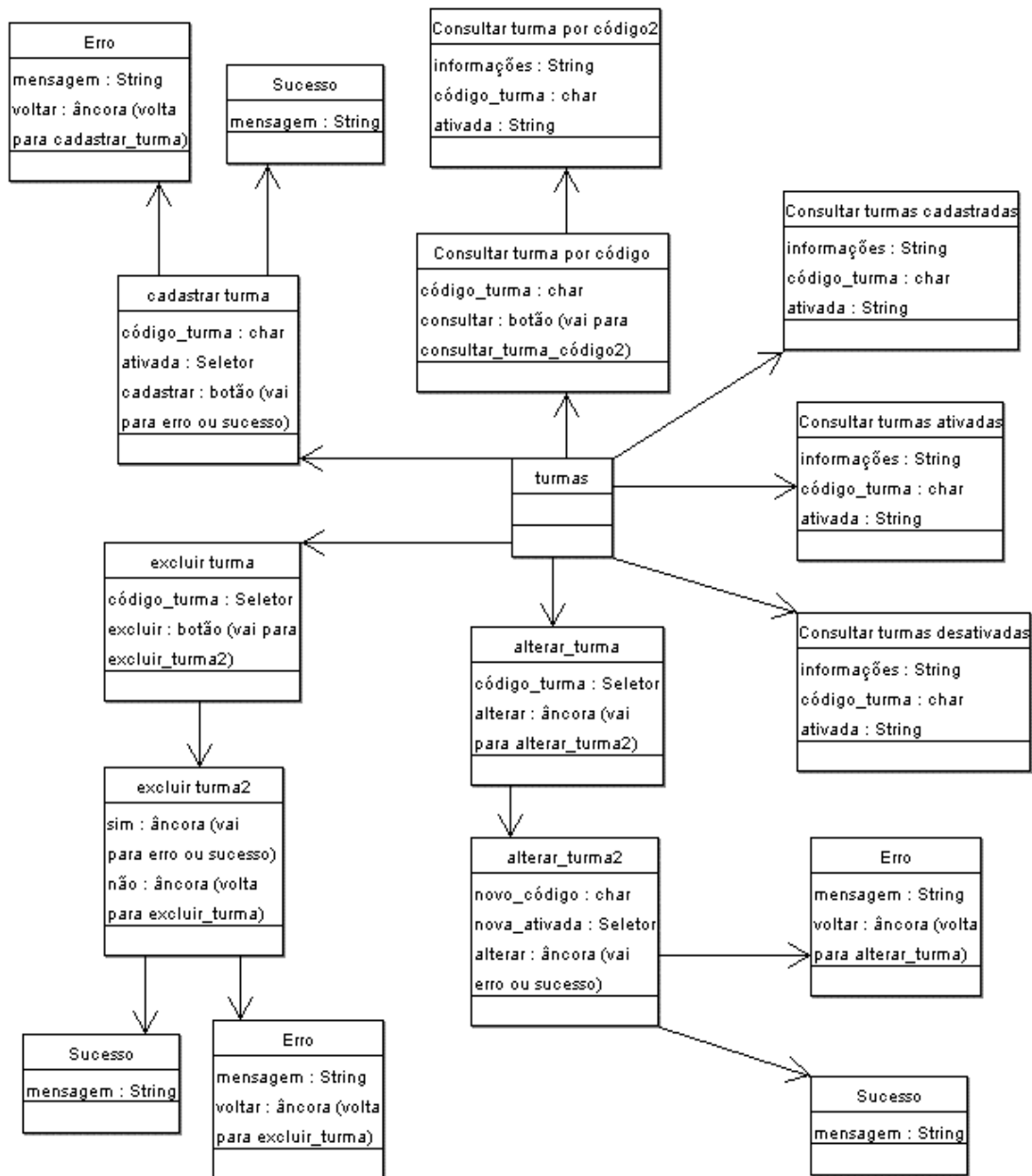
Modelagem navegacional de coordenação-secretária



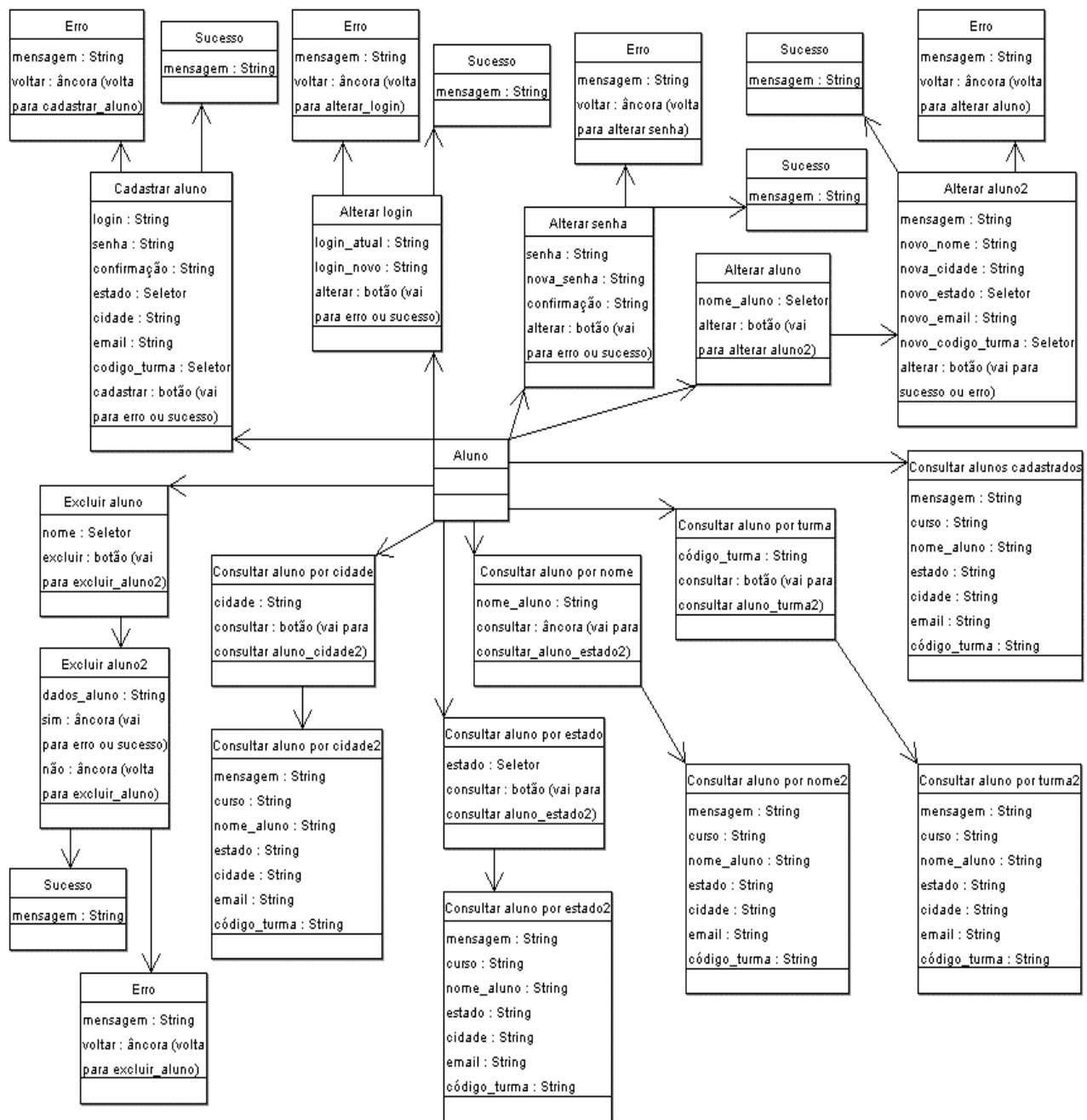
Modelagem navegacional de coordenação-professor.



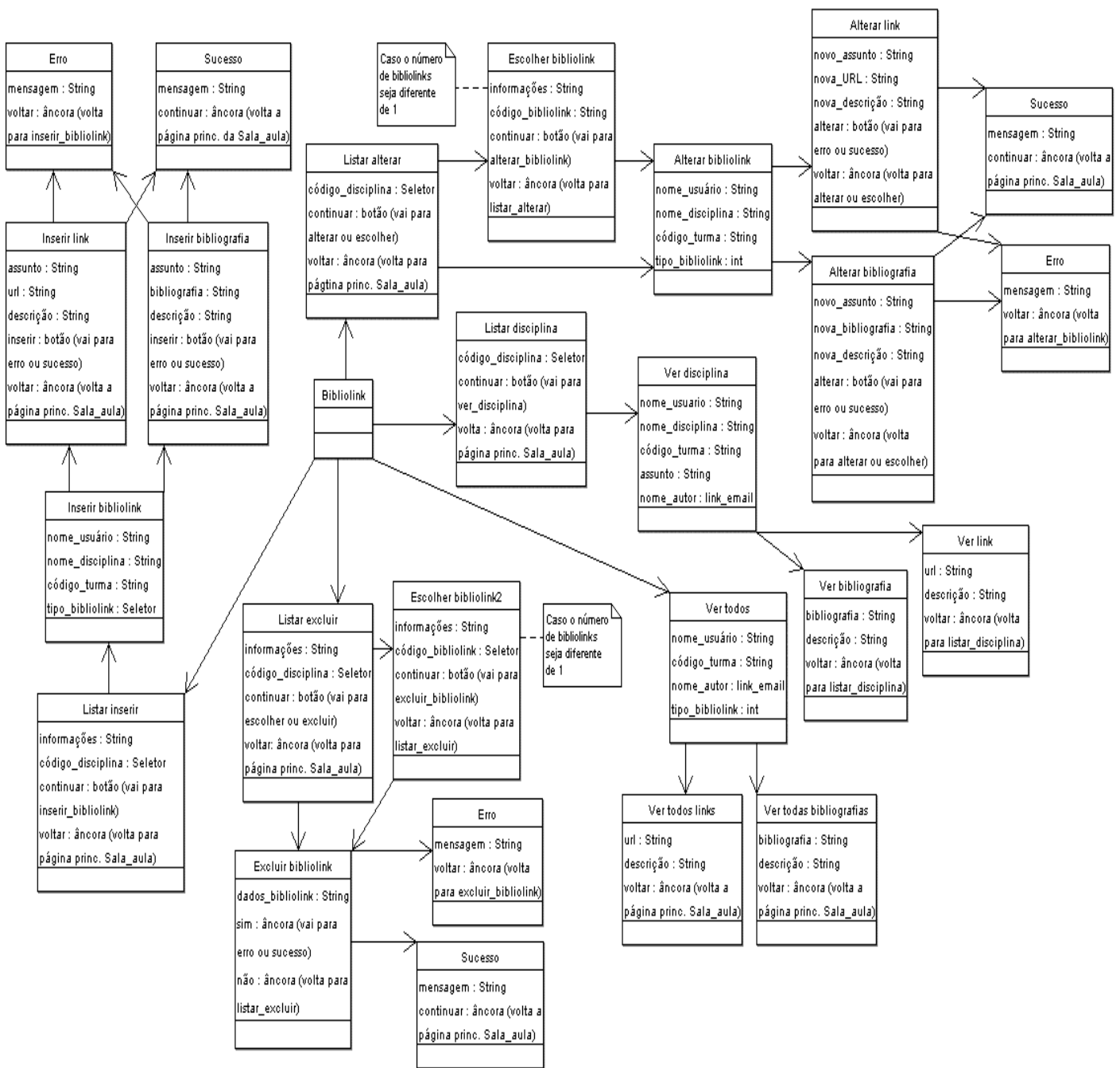
Modelagem navegacional de coordenação_disciplina



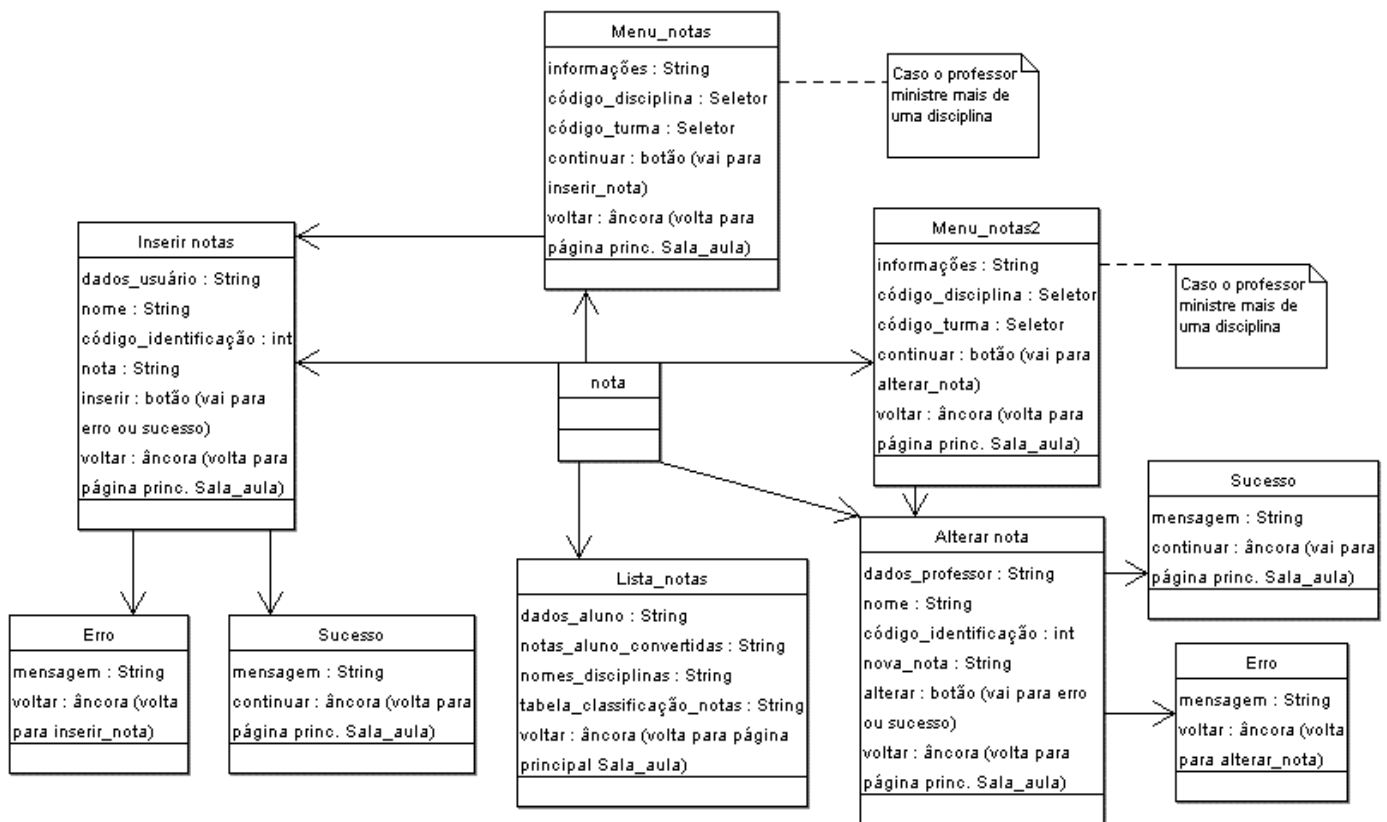
Modelagem navegacional de coordenação-turma



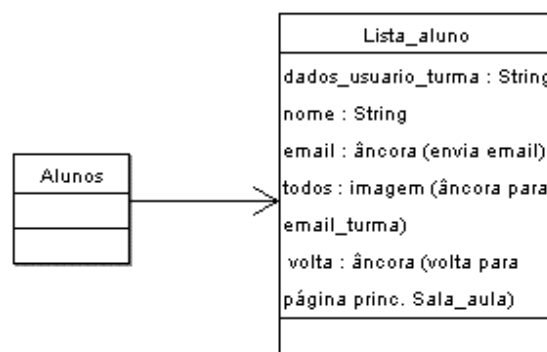
Modelagem navegacional coordenação-aluno



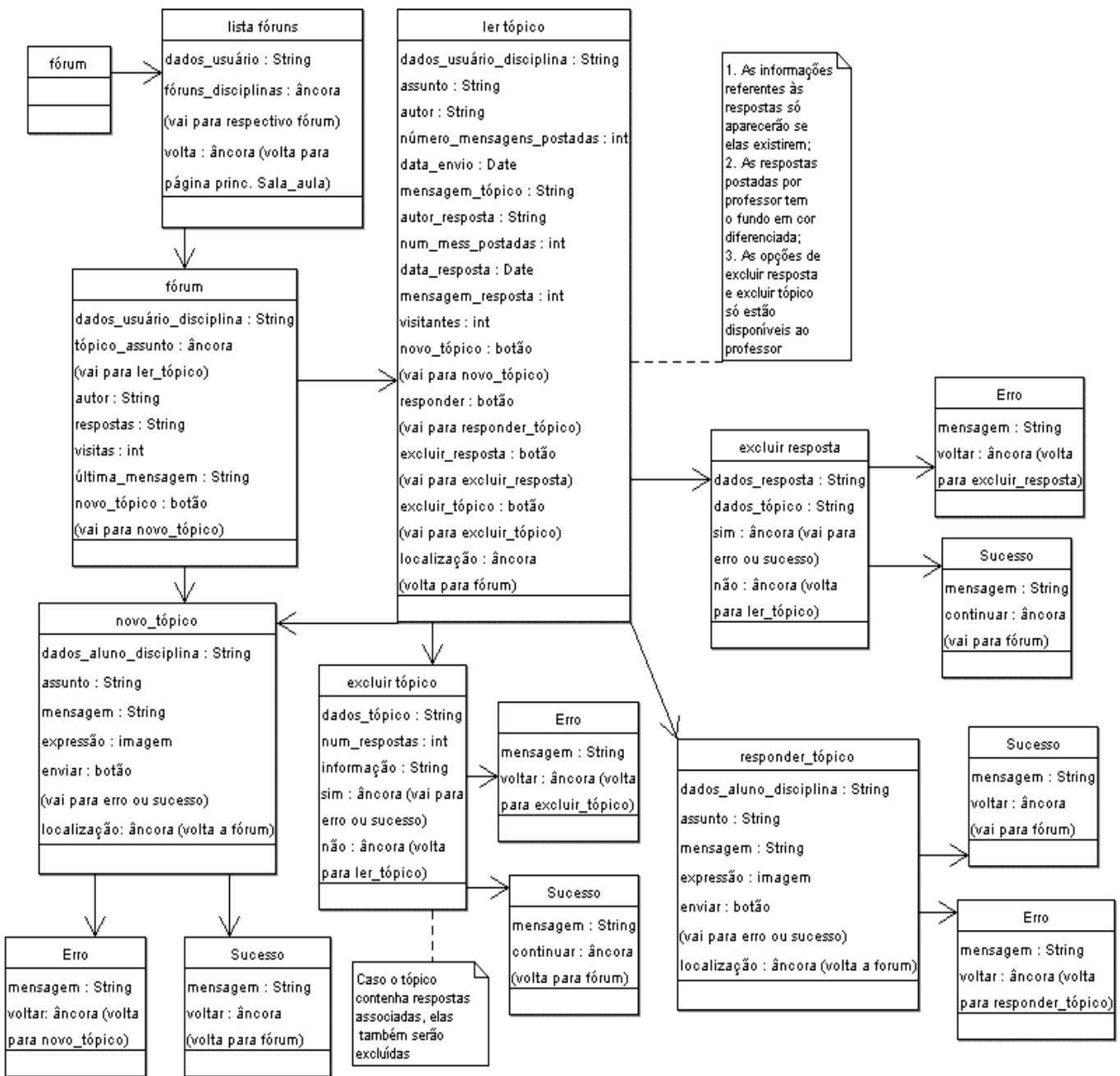
Modelagem navegacional salaAula_bibliolink



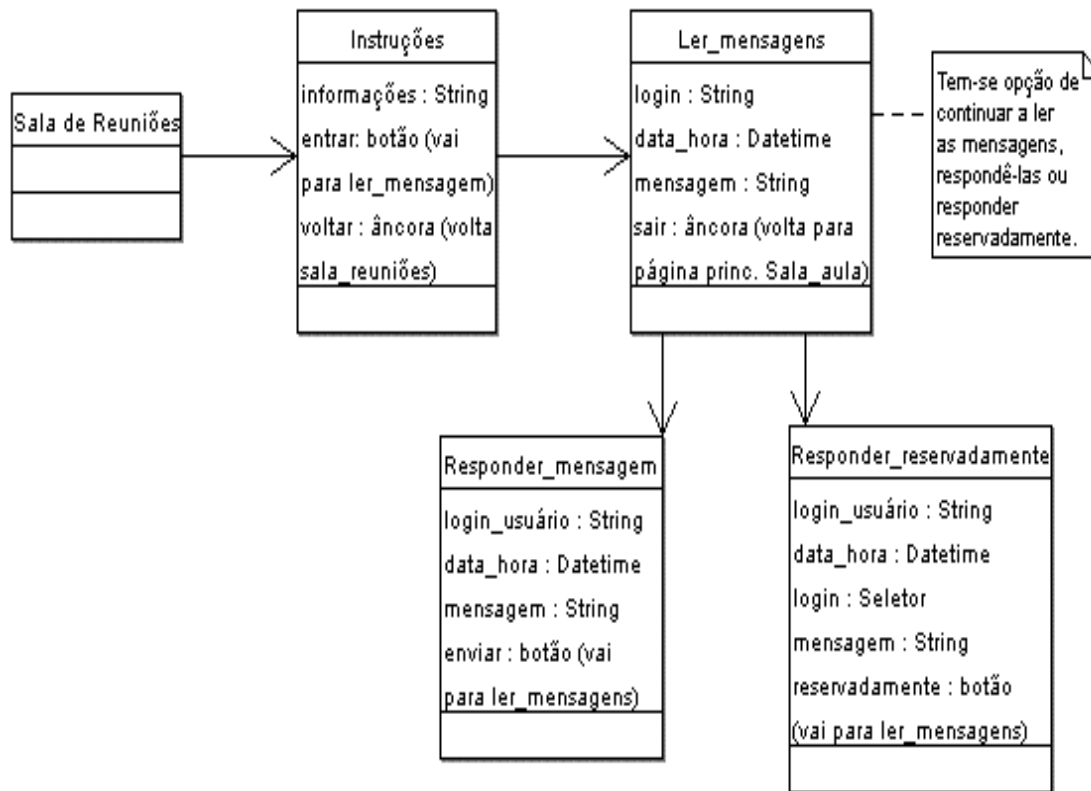
Modelagem navegacional salaAula_nota



Modelagem navegacional salaAula_lista_aluno



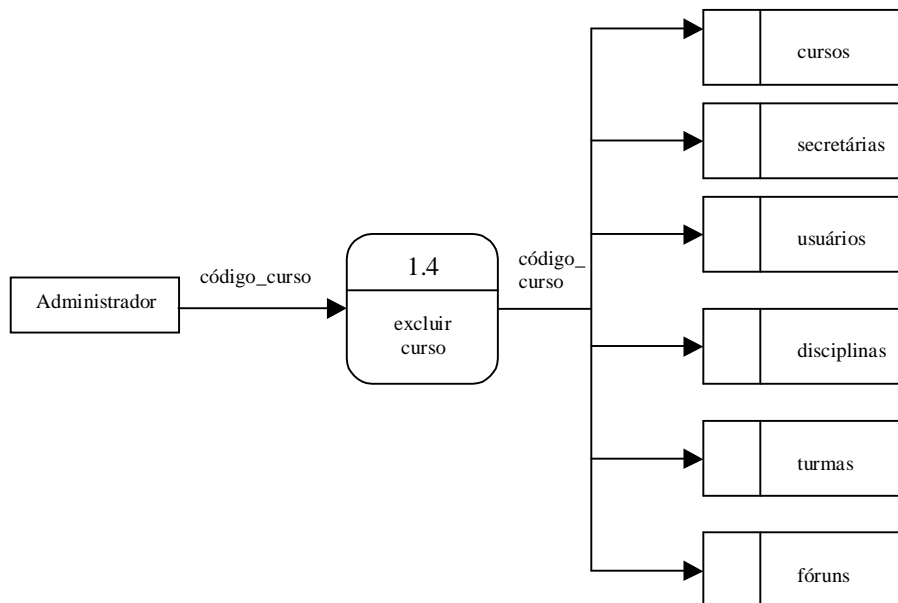
Modelagem navegacional salaAula_fórum



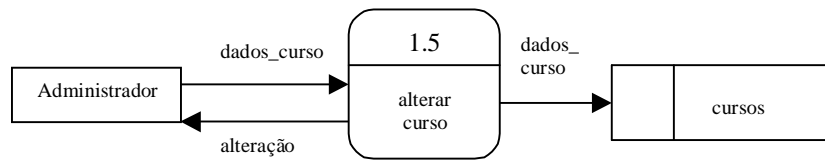
Modelagem navegacional salaAula_salaReuniões

ANEXO D: Diagramas de fluxo de dado (DFD)

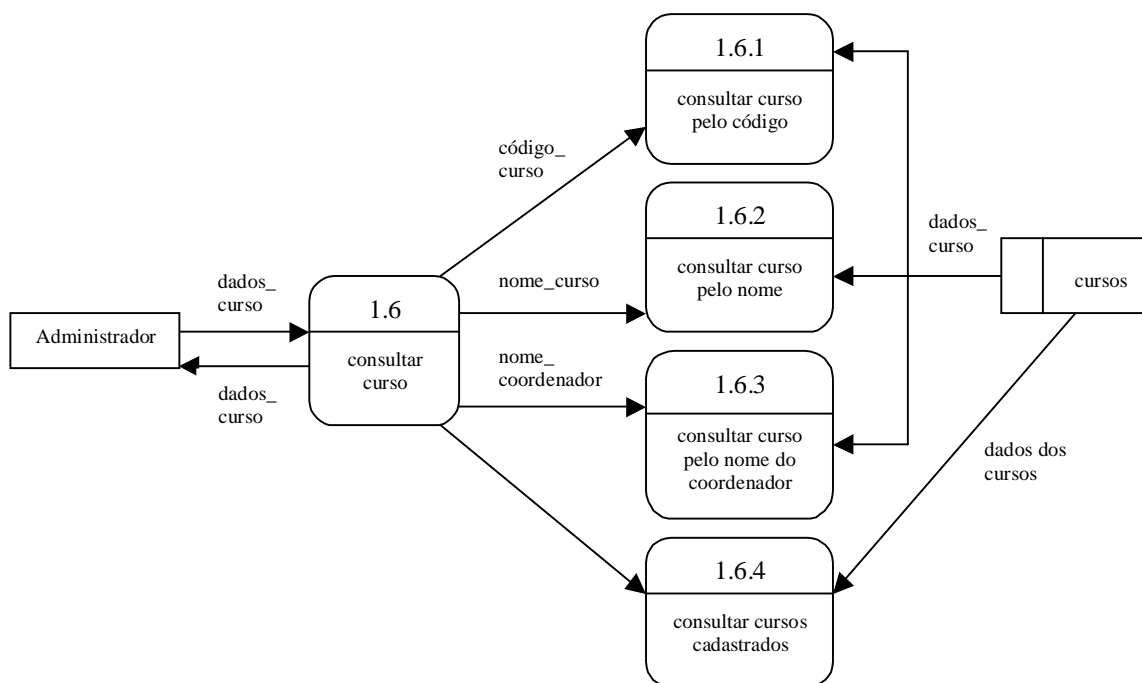
Administrador exclui um curso do sistema, retirando juntamente todos os vínculos das outras dependências



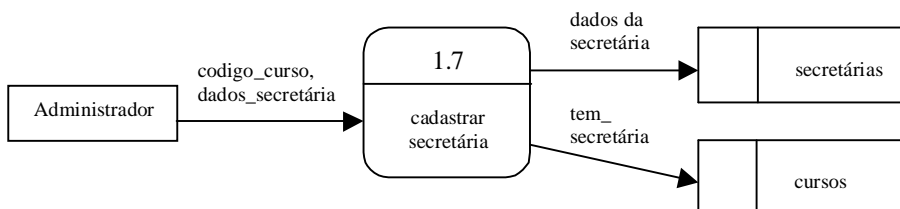
Administrador altera dados de um curso do sistema



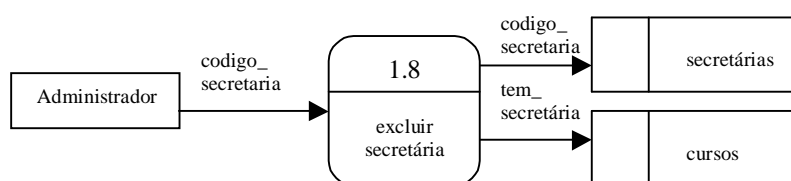
Administrador faz consultas a fim de acessar os dados referentes ao(s) curso(s). Estas podem variar de acordo com a opção escolhida.



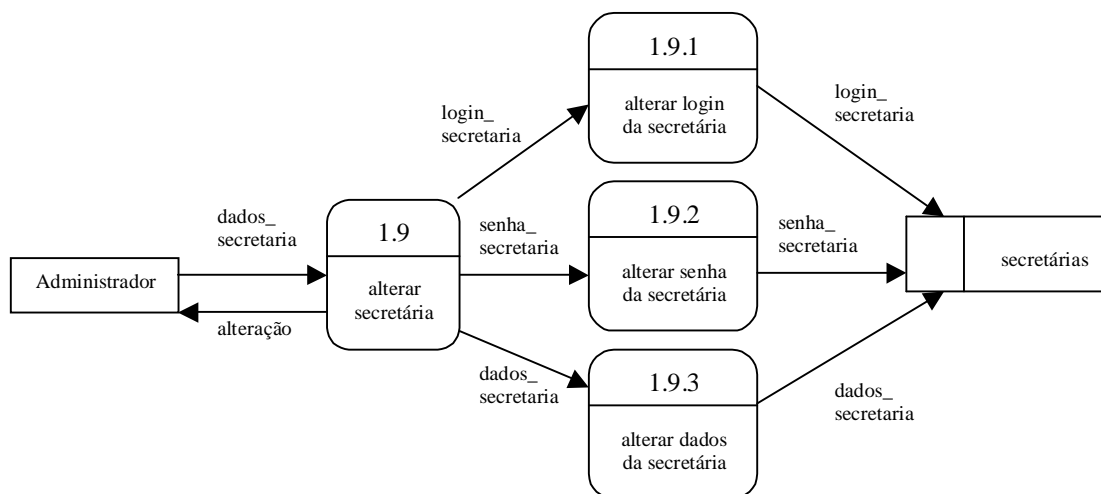
Administrador cadastra nova secretária para determinado curso no sistema



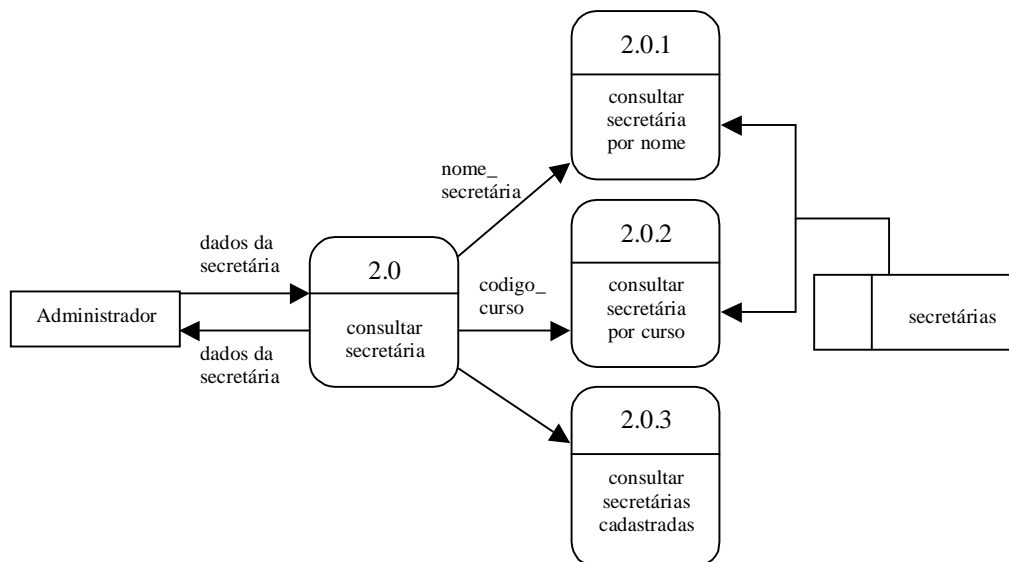
Administrador exclui secretária do sistema, desassociando-a do curso que estava vinculada.



Administrador pode alterar dado(s) da secretária, caso seja necessário.

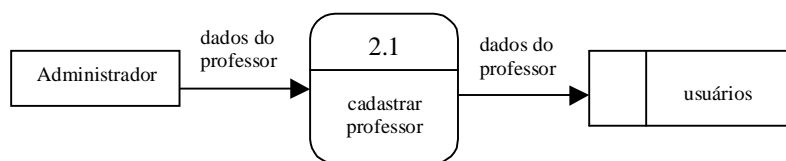


Administrador consulta informações referentes a(s) secretária(s) de um (todos os) curso(s)

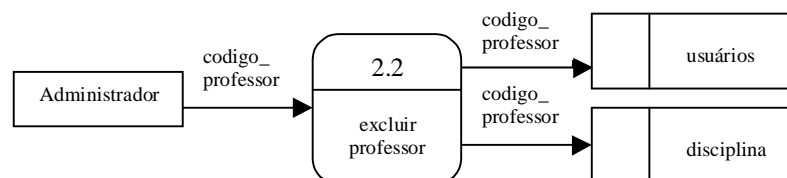


É importante ressaltar que a secretária poderá realizar as mesmas funções de gerenciamento do sistema que o administrador executa, logo a partir das funções referentes ao professor.

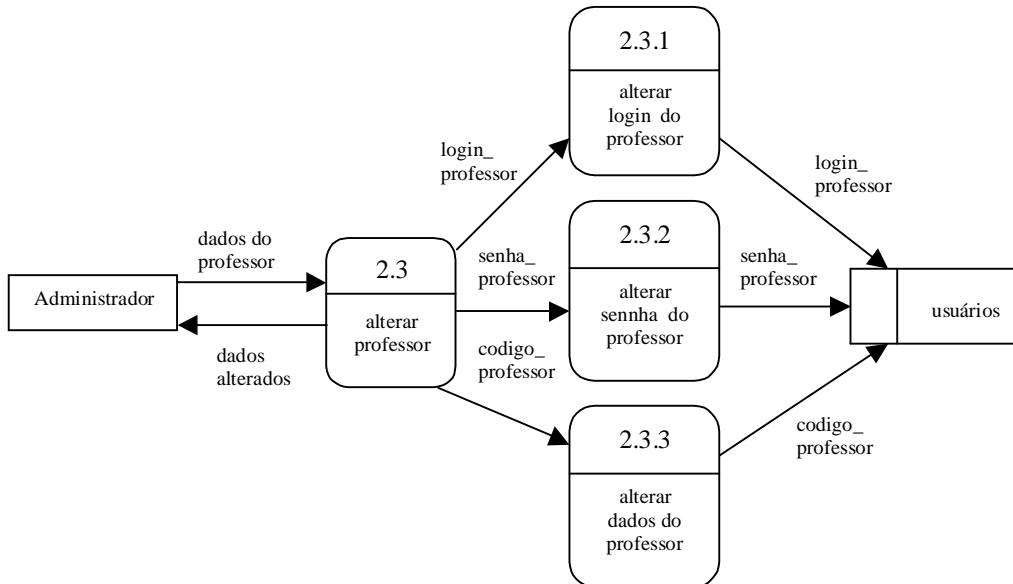
Administrador cadastra um novo professor no sistema.



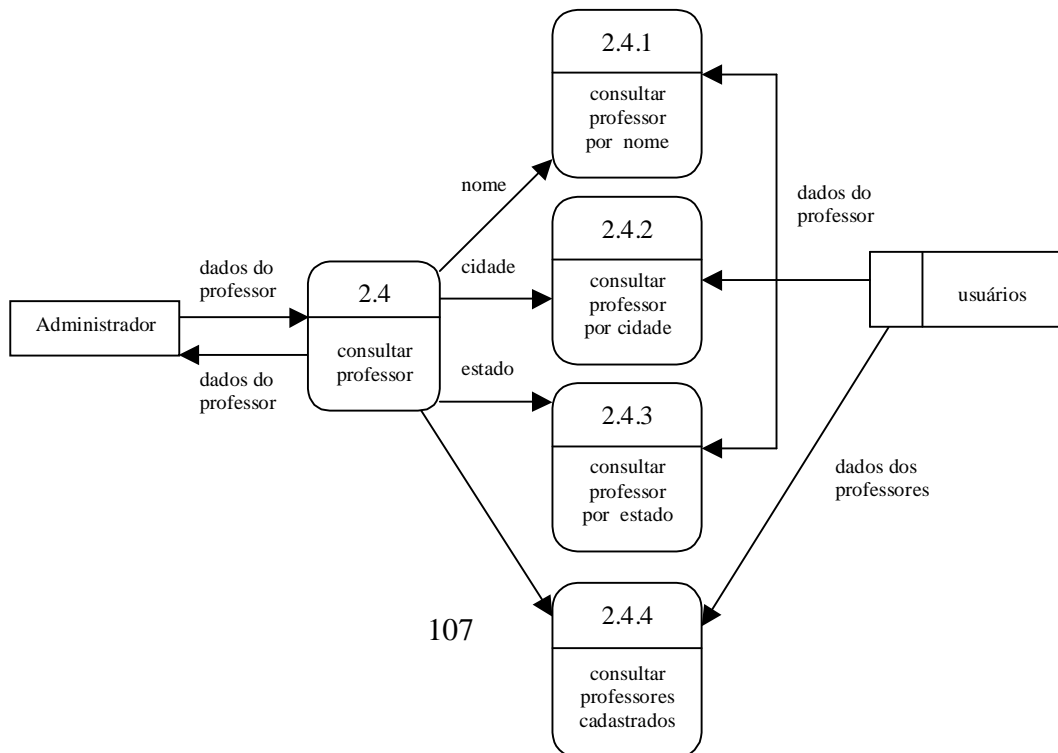
Administrador exclui um professor do sistema, desassociando-o da disciplina que ministrou.



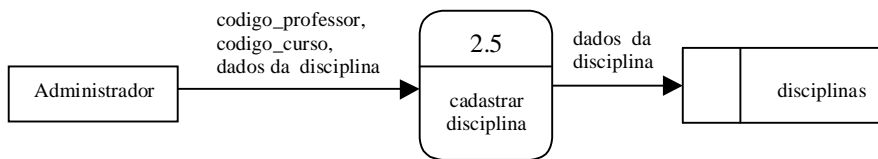
Administrador pode altera dado(s) do professor, se houver necessidade.



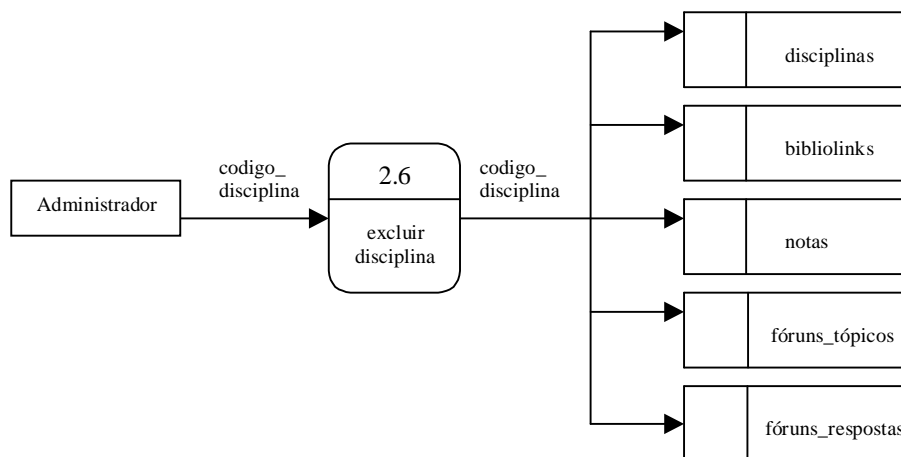
Administrador consulta informações referentes a um a um (todos os) professores cadastrado(s) no sistema.



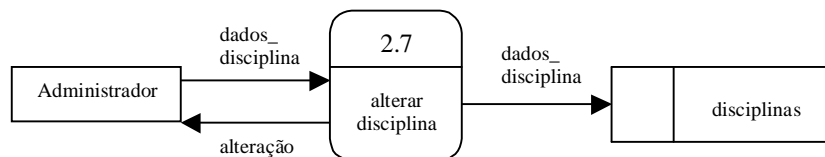
Administrador cadastra uma nova disciplina num curso, associando-a a um professor.



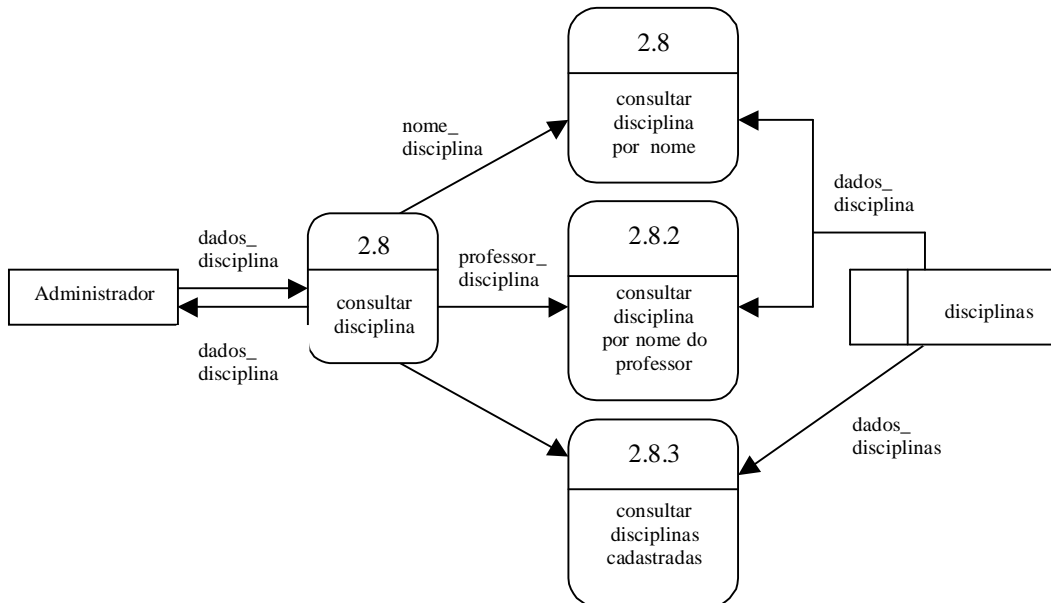
Administrador exclui uma disciplina do sistema, desabilitando as vinculações referentes a ela.



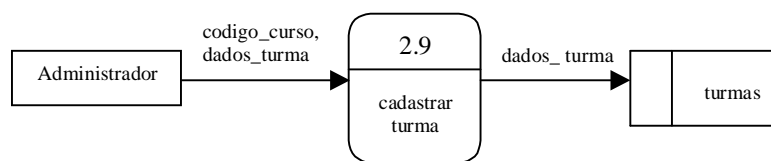
Administrador pode fazer alteração(ões) em determinada disciplina, se necessário.



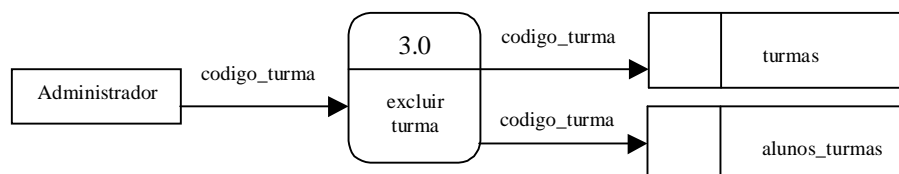
Administrador consulta informações referentes a uma (todas as) disciplina(s) cadastradas em um curso.



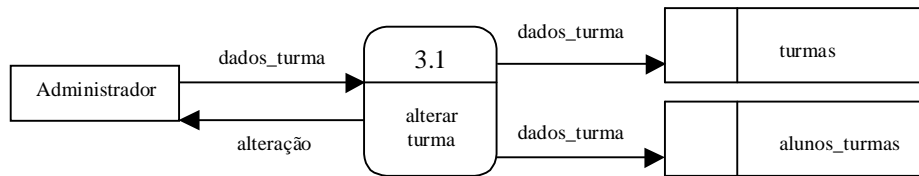
Administrador cadastra uma turma em determinado curso do sistema.



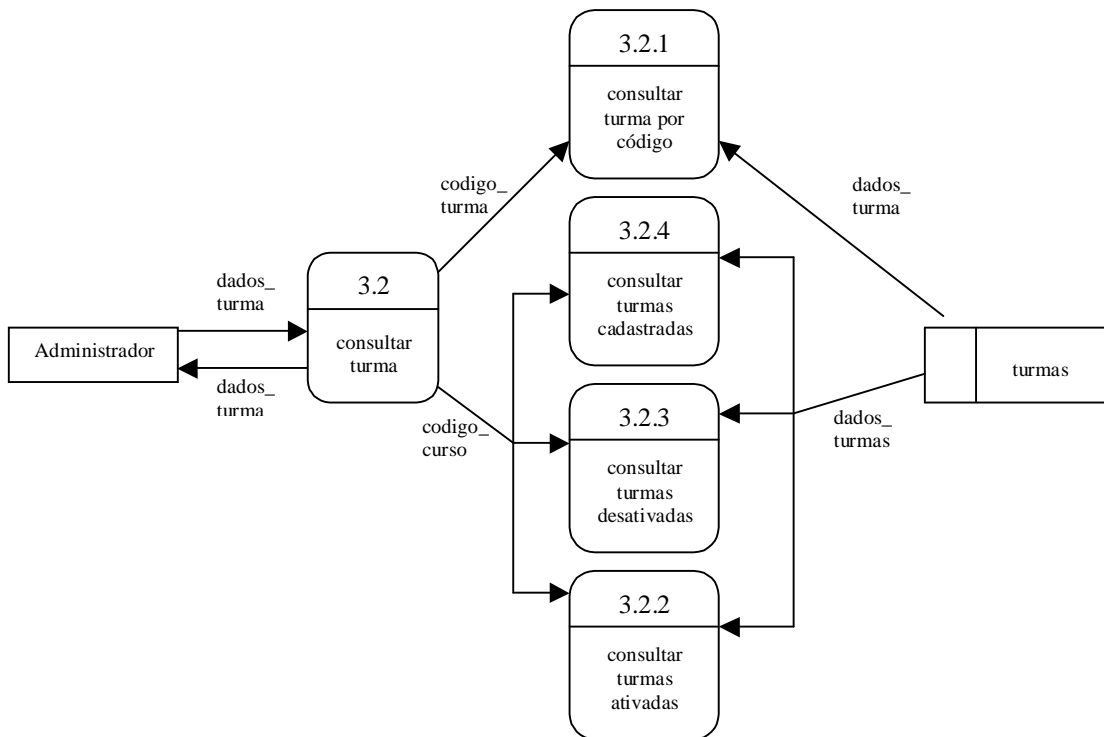
Administrador exclui uma turma do sistema.



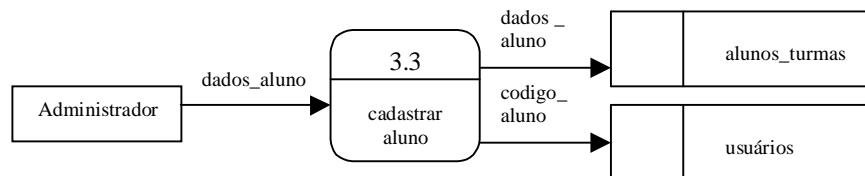
Administrador pode alterar dados relativos a uma turma se for necessário.



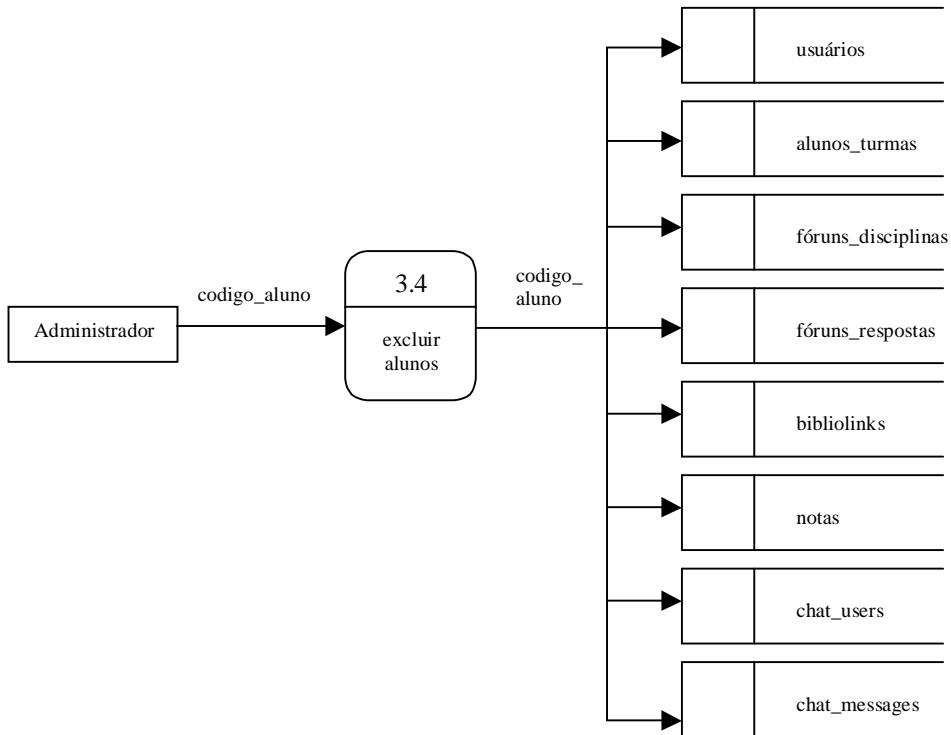
Administrador consulta informações de uma (todas as) turmas cadastradas no sistema.



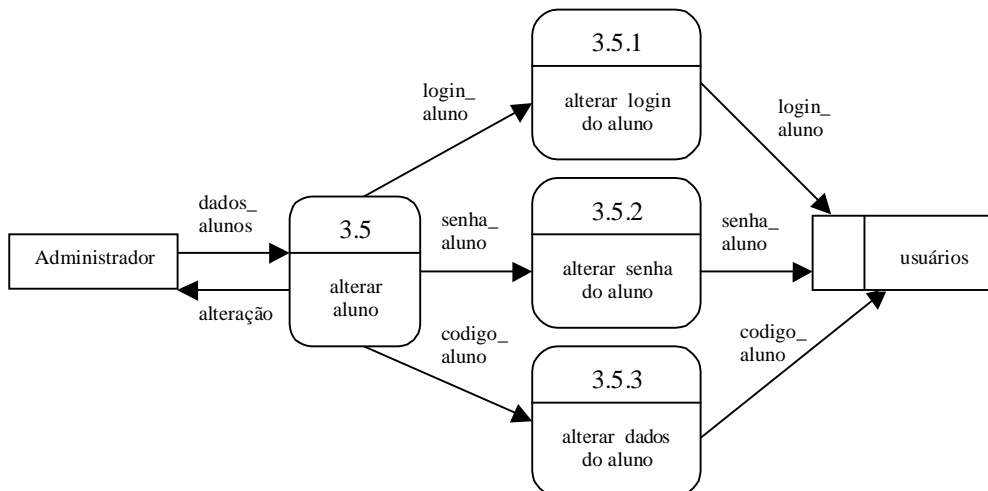
Administrador cadastra um novo aluno no sistema, associando-o a uma turma.



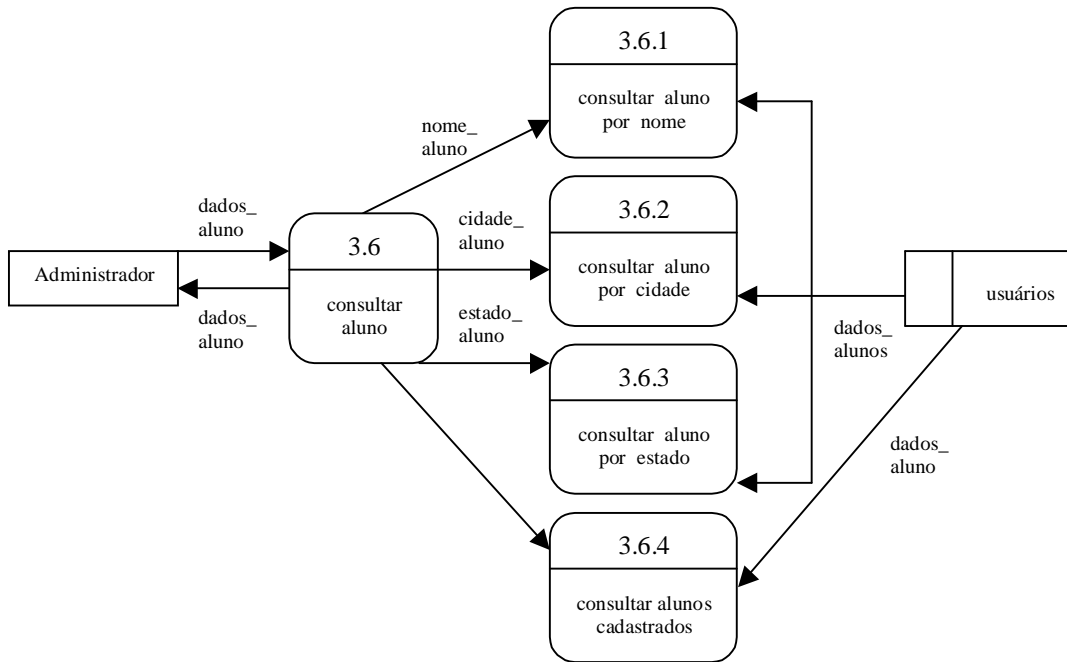
Administrador exclui um aluno do sistema, retirando juntamente todos os vínculos das outras dependências



Administrador pode alterar dado(s) de um aluno se for necessário

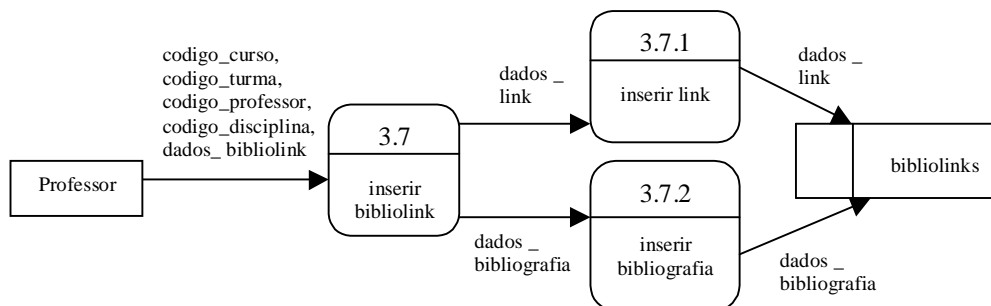


Administrador consulta informações de um (todos os) alunos em um curso.

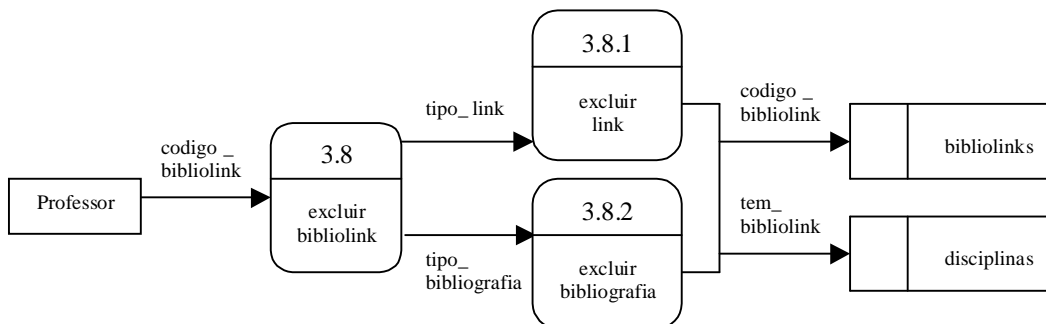


É importante ressaltar que o professor e o aluno basicamente realizam as mesmas funções, o professor conta com alguns permissões extras como alterar ou excluir alguns itens

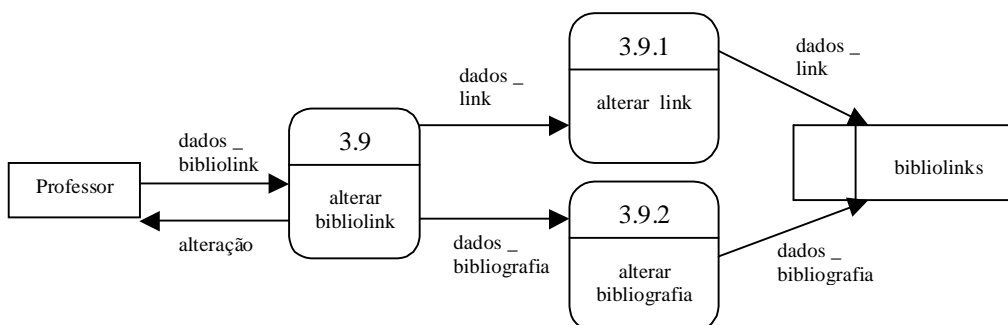
Professor insere um bibliolink (link ou bibliografia) no sistema.



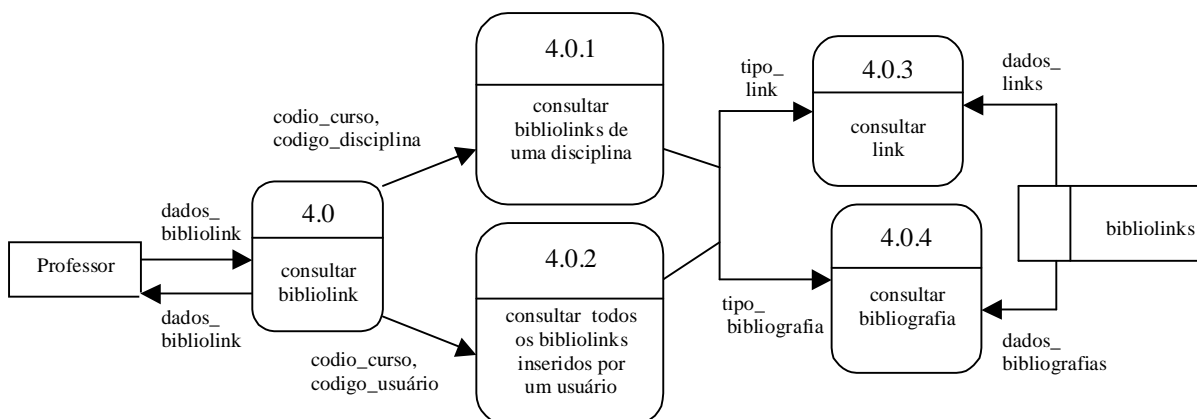
O professor pode excluir algum bibliolink que não parecer apropriado aos assuntos estudados em determinada disciplina. Desabilita-se o vínculo com disciplinas.



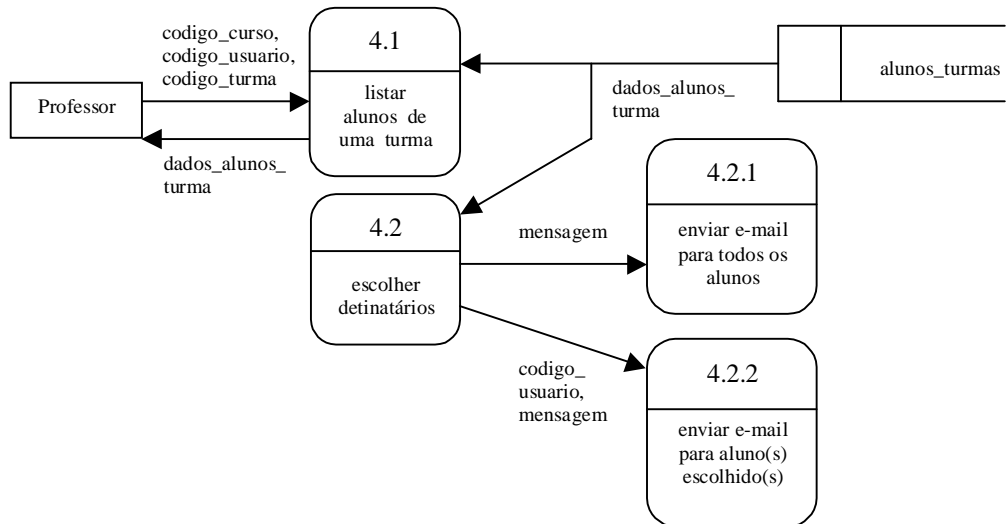
O professor pode alterar dado(s) de um bibliolink se for necessário.



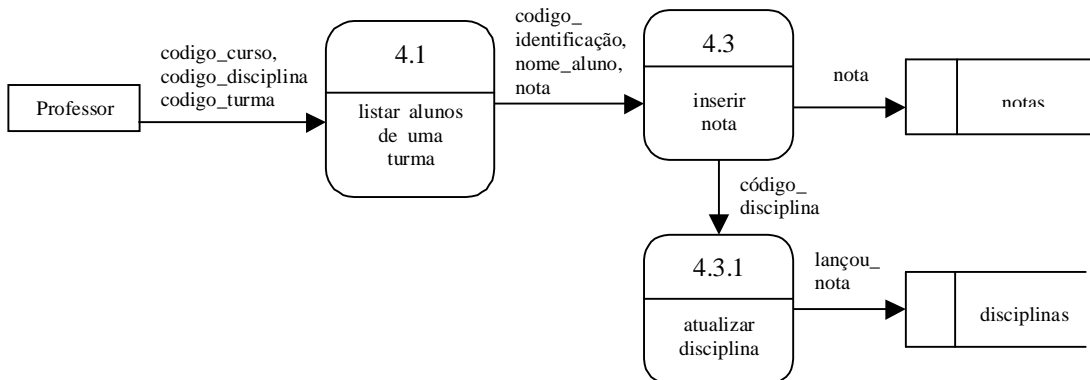
O professor pode consultar informações de um (todos os) bibliolinks cadastrado(s)



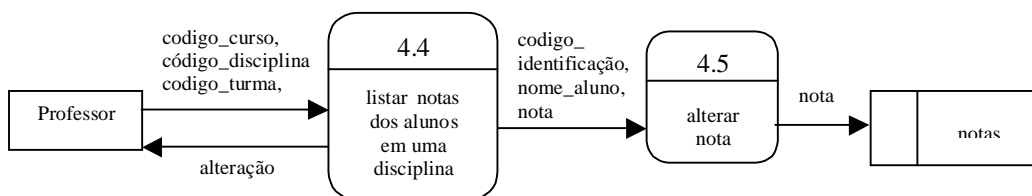
O professor pode consultar os alunos de uma turma para a qual ministra uma disciplina e enviar *e-mail* para algum aluno específico ou para a turma inteira



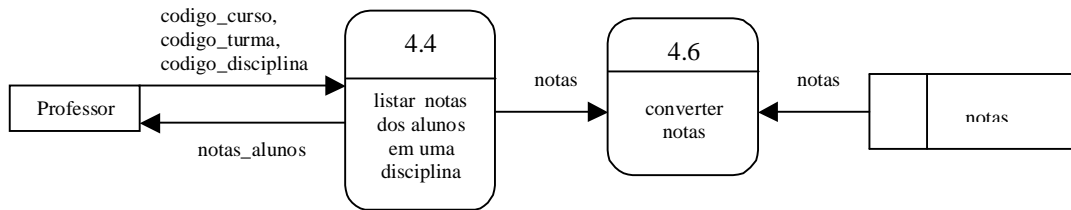
O professor pode inserir notas para os alunos de uma turma



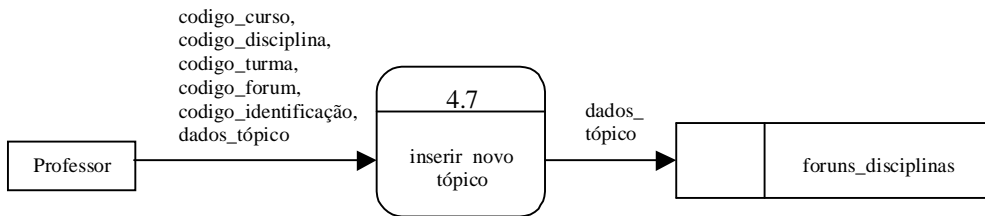
O professor pode alterar alguma nota de algum aluno se for necessário.



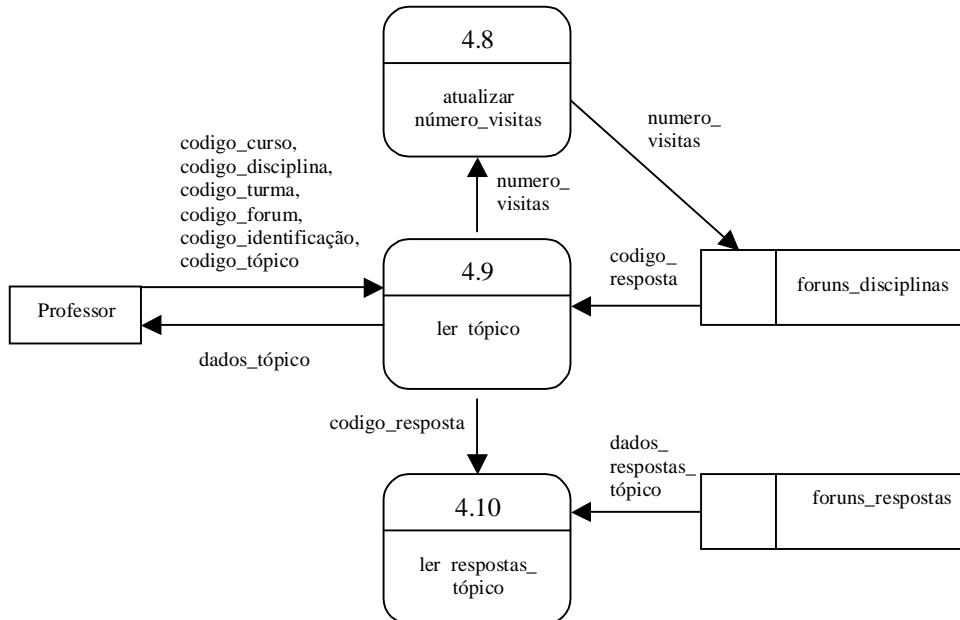
O professor consegue deixar as notas no padrão usado na instituição de ensino graças a conversão (numérica-alfabética) realizada pelo sistema.



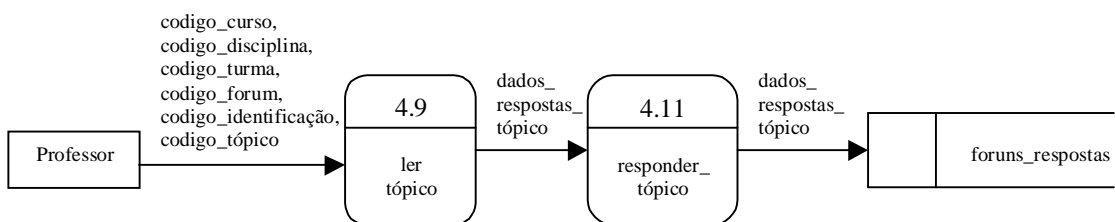
O professor pode inserir um novo tópico no fórum .



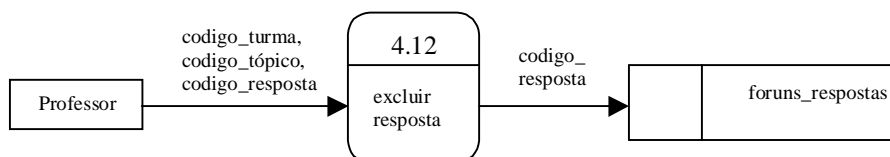
O professor pode ler um tópico do fórum e sua resposta (se houver)



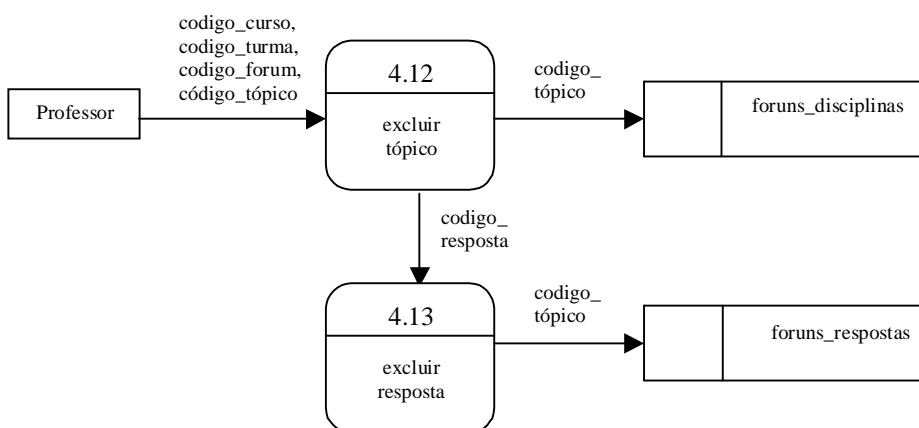
O professor pode responder a um tópicos quando achar conveniente.



O professor pode excluir uma resposta, caso a mesma distoe da discussão proposta.



O professor pode excluir um tópicos e as respostas associadas a ele, se achar que a discussão é desnecessária



O professor acessa a sala de reunião e troca mensagens síncronas com os alunos presentes

