

Introdução

A informação e o conhecimento passaram a exercer um novo e estratégico papel na sociedade atual, pois vivemos em

“...uma nova era, a da *economia digital*, em que o capital humano passa a ser mais importante que o capital tradicional. Nesta era da economia baseada mais no cérebro do que nos recursos físicos e materiais, as inovações e vantagens competitivas tornam-se efêmeras e transitórias em um menor espaço de tempo.” (TACHIZAWA, ANDRADE, p.132, 1999)

Sendo assim, procurou-se no desenvolvimento deste trabalho, construir uma ferramenta que colaborasse para facilitar a gestão das instituições de ensino e, ao mesmo tempo, fosse inovadora e altamente competitiva no mercado.

É evidente que as Tecnologias de Informação e Comunicação (TIC) estão, cada vez mais, presentes em todos os setores da sociedade e, com as instituições de ensino não poderia ser diferente. De acordo com Takeshy Tachizawa e Rui Andrade (1999),

“Está ocorrendo uma mudança na natureza das organizações e, o reflexo natural, nas IEs. A organização de antigamente já não funciona mais. A transformação organizacional, possibilitada pela informação, é necessária para se obter sucesso no novo ambiente. A nova organização é dinâmica e pode responder rapidamente a mudanças no mercado.... Os processos empresariais são simplificados em favor da produtividade e da qualidade. A nova organização é aberta e atua em rede.” (TACHIZAWA, ANDRADE, p.132,)

Após cumprir toda a formação superior em Computação e, estudar este campo sob diversas perspectivas, teve-se como objetivo, ao realizar este trabalho, a união do conhecimento construído com sua aplicação prática.

Para tal, foi proposto e implementado um sistema de informação que auxiliasse as instituições de ensino nos processos inerentes à sua gestão. Mais do que isso, o intuito foi elaborar uma ferramenta de apoio pedagógico, que fosse capaz de agregar valor aos canais de comunicação das instituições de ensino, estreitando a relação entre alunos – consumidores – e as demais partes envolvidas – fornecedores.

Por fim, “*No contexto desse novo paradigma de tecnologia, quanto mais tempo a instituição de ensino esperar para iniciar esta transição, maior será a exigência de investimentos e gastos, de longo e curto prazos.*”

O trabalho está organizado da seguinte forma:

O primeiro capítulo apresenta o projeto conceitual do Portal Acadêmico Innova. Nele são apresentados o objetivo do trabalho e a motivação para o desenvolvimento do mesmo. Também são descritas as funcionalidades do sistema nos seus diferentes módulos, bem como, propostas de

solução para implementação da ferramenta.

No segundo capítulo, são apresentadas atividades que compõem o projeto lógico do desenvolvimento do portal, como: a lista de eventos realizados pelo sistema e os diagramas da UML. Através deles, é possível descrever a estrutura do sistema e identificar seu comportamento. Nesta seção do trabalho, pode-se verificar também, o dicionário de objetos, que descreve a composição das tabelas do sistema e dos métodos.

Já no terceiro capítulo, o esquema físico do trabalho é apresentado, juntamente com as regras para a padronização do desenvolvimento do sistema. Este capítulo é composto pelo projeto lógico e físico de banco de dados.

No capítulo quatro, são descritas as tecnologias e metodologias, que foram utilizadas para o desenvolvimento e implementação do trabalho.

O capítulo seguinte, o quinto, apresenta os protótipos de tela do sistema, bem como, o cronograma do trabalho.

Por último, são realizadas as considerações finais e sugestões para trabalhos futuros, todos relacionados ao Portal Acadêmico Innova.

1. Projeto Conceitual

1.1. Objetivo

O presente trabalho tem por objetivo implementar um portal acadêmico para ser acessado via *web*, no intuito de agregar valor aos *websites* das instituições de ensino. A ferramenta conterá três grandes funcionalidades: Gerenciamento de Informações Acadêmicas para alunos, professores e funcionários, Mural de Estágios e ambiente para Ensino à Distância (EAD).

1.2. Motivação

As transformações da era da informação acabam por gerar uma demanda por facilidades, a fim de que os indivíduos se sintam cada vez mais à vontade para utilizar os aparatos tecnológicos que os cercam.

Visando atender às necessidades das instituições de ensino, agilizar e facilitar a comunicação entre elas e os alunos, o portal acadêmico justifica ser implementado, pois, nos dias atuais, há uma crescente demanda pela disponibilidade de informações educacionais e suas áreas afins.

O módulo de Gerenciamento de Informações Acadêmicas permite ao aluno acessar diversos dados que lhes concernem, como por exemplo, grades dos cursos, ementas das disciplinas, pré-inscrição em disciplinas, datas das avaliações e suas respectivas notas, calendário acadêmico, dentre outras. Já os professores, poderão publicar materiais para auxiliar os alunos, agendar as datas das provas e outras avaliações, lançar as notas e a frequência dos alunos, emitir relatórios gerenciais, dentre outras.

O estágio é uma importante etapa na vida acadêmica dos discentes, pois possibilita aplicar de fato, o conhecimento construído em sala de aula. Esta é a fase em que um aluno inicia sua carreira profissional. Há inúmeras empresas oferecendo oportunidades de estágio, cada uma delas requisitando perfis diferentes para atuação em diversas áreas. Sendo assim, as vagas acabam por ter, de certa forma, uma identidade com a qual os alunos se identificam.

Existem diversos *websites* que disponibilizam oportunidades de emprego e atuam como intermediadores no processo de admissão de profissionais. Entretanto, muitos deles não oferecem vagas de estágio. O módulo de Mural de Estágios vem para facilitar este processo e auxiliar os futuros profissionais, encurtando a distância entre eles e o mercado. Hoje para que os alunos possam ter acesso a estas oportunidades, eles devem se cadastrar nos diversos *sites* existentes e, dessa forma, acabam tendo suas informações redundantes em diversos locais.

Ensino à distância (EAD), segundo Jose Manuel Moran (2002), *“É ensino/aprendizagem onde professores e alunos não estão normalmente juntos, fisicamente, mas podem estar conectados, interligados por tecnologias, principalmente as telemáticas, como a Internet”*. Esta facilidade do portal acadêmico visa oferecer uma forma de aumentar a capacitação não somente dos alunos, como também de outros profissionais, pois lá existirão cursos de extensão, cursos rápidos, fóruns de discussão, provas de certificação de empresas conveniadas, dentre outras mais, facilitando cada vez mais o aprendizado.

O ambiente de EAD colabora para o processo de formação continuada, que, também de acordo com Jose Manuel Moran (2002), *“...se dá no processo de formação constante, de aprender sempre, de aprender em serviço, juntando teoria e prática, refletindo sobre a própria experiência, ampliando-a com novas informações e relações”*.

1.3. Descrição

Pela análise, a partir de questionários aplicados aos alunos, professores e funcionários da Uni La Salle e de pesquisas realizadas em diferentes *websites*, de diversas instituições de ensino, o trabalho será desenvolvido com intuito de agilizar e facilitar a comunicação entre os alunos, professores, funcionários e as instituições, bem como auxiliar os profissionais de ensino na tomada de decisão.

A fim de facilitar a implementação e entendimento também por parte dos usuários, o portal está segmentado em módulos e contém três grandes funcionalidades: Gerenciamento de Informações Acadêmicas, Mural de Estágios e Ambiente de Ensino à Distância (EAD).

1.3.1. Módulo de Gerenciamento de Informações Acadêmicas

O módulo de informações acadêmicas contém dados inerentes aos alunos, professores, funcionários, cursos, grades, disciplinas, para fins de elaboração da estrutura que compõe uma instituição de ensino. A instituição deve ser cadastrada no sistema com dados como: nome, CNPJ e endereço completo.

Para o sistema, dos alunos é importante saber informações básicas, tais como: seu nome completo, CPF, documento de identidade (número, UF, data de emissão, órgão emissor), endereço completo, telefones residencial e celular, sexo, data de nascimento, e-mail e matrícula. A secretaria acadêmica é responsável pela alimentação do sistema com estas informações.

A universidade possui diversos cursos de graduação, que devem ser cadastrados através de seus nomes, titulação, habilitação, duração em semestres e código. Após o curso ser registrado, deve

ser informada em seguida a sua grade curricular, composta de dados tais como: nome, código, descrição e situação, pois uma grade pode ser modificada ao longo da existência do curso, isto é, disciplinas podem ser adicionadas ou excluídas fazendo com que novas grades sejam geradas.

A grade curricular de um curso é composta de diversas disciplinas. Assim, uma disciplina deve possuir um código identificador, nome, carga horária, tipo, isto é, se é obrigatória ou optativa, seus pré-requisitos caso haja algum, quantidade de créditos e situação, pois ao longo da existência de um curso as disciplinas podem ser adicionadas ou retiradas das grades. Sendo assim, após uma disciplina ser cadastrada, ela passa a constar como disponível para ser associada a um determinado curso. Caso a instituição queira retirar esta disciplina de uma grade, ela passa a constar como bloqueada no sistema, podendo ser descontinuada ou reativada. Para que uma disciplina possa ser associada a uma grade, sua ementa deve ser cadastrada no sistema.

A ementa de uma disciplina deve ser informada com seu nome, código e situação. Depois de informada, a ementa passa a ficar com sua situação em análise, isto é, aguardando aprovação da coordenação do curso para que ela possa entrar em vigor. Todas as ementas que estão em análise podem ser aceitas ou não, ou então simplesmente podem ser canceladas. Quando aceitas, as ementas constam como ativas no sistema podendo estar ou não disponíveis para consulta pelos alunos. Elas também podem ser bloqueadas caso não sejam mais utilizadas e passam a constar como inativas, podendo ser excluídas se desejado.

Cada uma das ementas é composta por um ou mais conteúdos, os quais devem possuir também um código identificador e nome.

Definida a estrutura de um curso, ou seja, ter associado a ele uma grade composta de diversas disciplinas; e estas por sua vez cadastradas com suas respectivas ementas e conteúdos; poderão ser feitas matrículas no mesmo.

Quando um aluno é matriculado em um curso de uma instituição de ensino, são registradas sua data de ingresso, observações, caso necessárias, e sua situação. Pois, um discente pode estar matriculado em um determinado curso da instituição, pode trancar sua matrícula, ou simplesmente cancelá-la se desejado. Um aluno pode estar associado a uma ou mais turmas dependendo das disciplinas às quais estiver inscrito. Através do portal acadêmico, no módulo dos alunos, os discentes podem realizar diversas pré-inscrições nas disciplinas existentes caso estejam aptos a cursá-las.

Quando um aluno realiza uma pré-inscrição através do portal acadêmico, o sistema gera um número identificador referente ao pedido, a data de solicitação e a sua situação. O aluno pode cancelá-la ou então deverá aguardar o resultado de uma análise feita pelo coordenador em cima da requisição. Esta análise é compreendida na verificação do número de vagas e quantidade de

inscrições, se os inscritos possuem os pré-requisitos necessários para cursar a disciplina desejada, etc. Um aluno pode estar inscrito em uma disciplina, também pode trancar sua inscrição ou simplesmente cancelá-la.

Como consequência de uma inscrição em disciplina, o discente é associado a uma turma e, sendo assim, é importante o sistema saber informações como: código, semestre, ano, turno, dia da semana, horário, local e sala de aula.

No Gerenciamento de Informações Acadêmicas, os alunos têm à sua disposição diversos tipos de consultas, dentre elas estão: as datas das avaliações cadastradas pelos professores nas várias disciplinas, e as notas, caso as mesmas estejam disponíveis para serem consultadas. Um professor, ao cadastrar uma nota, pode ou não disponibilizá-la para consulta, ou então pode excluí-la caso a tenha cadastrado de forma incorreta. Os alunos podem solicitar revisão da prova quando julgarem necessário. Neste caso, a nota do aluno é bloqueada e passa a constar com situação em análise. O professor faz a revisão da prova do aluno em questão e depois disponibiliza a nota no portal novamente para ser consultada. Os discentes também podem verificar informações sobre: as grades curriculares dos cursos ao quais estiverem inscritos, as grades de disciplinas que compõem um determinado semestre, a grade de horários das disciplinas do semestre atual e as ementas de cada disciplina do curso.

Quando um discente necessita comprovar que está matriculado a um curso em uma instituição de ensino, geralmente ele recorre ao setor de registro acadêmico e solicita uma declaração de curso. Este processo leva alguns dias para ser concretizado. Através do portal acadêmico, o aluno tem a facilidade de poder imprimir sua declaração de curso.

Assim como os alunos, os professores da instituição, de acordo com seu perfil de acesso, podem realizar diversas tarefas no módulo de Gerenciamento de Informações Acadêmicas. Para o sistema, é importante saber dos professores informações como: seu nome completo, CPF, documento de identidade (número, UF, data de emissão, órgão emissor), endereço completo, telefones residencial e celular, sexo, data de nascimento, e-mail e matrícula.

Um professor pode estar alocado em diversas turmas em função da(s) disciplina(s) ministrada(s) por ele. Ao final de cada dia de aula, ele pode lançar através do portal acadêmico, a frequência dos alunos e o conteúdo ministrado naquele dia. São registrados junto ao conteúdo: a data atual, o próprio conteúdo e comentários, se necessário. Esta função colabora para tornar o acompanhamento da coordenação, responsável por cada curso, simples e objetivo.

Com o intuito de auxiliar os alunos durante a construção de conhecimento nas diversas disciplinas, os professores, também podem disponibilizar materiais de apoio para os estudantes. Cada material tem cadastrado seu código de identificação, nome, descrição, tipo, isto é, se é um

artigo, exercício, apresentação em *slides*, textos, etc e são registradas sua data de cadastramento no sistema e situação. Ao cadastrar um material ele consta no sistema como pendente e pode ser disponibilizado para consulta ou *download*. O arquivo disponível também pode ser bloqueado, fazendo com que o mesmo não esteja mais visível para os estudantes. Caso o professor queira reutilizá-lo, ele pode desbloqueá-lo e o material pode ser consultado novamente. Junto ao material, os docentes têm a oportunidade de associar comentários a ele. Os materiais podem ser excluídos a qualquer momento se assim desejado.

O agendamento de avaliações pelo portal acadêmico será uma das formas de formalizar o planejamento do professor com suas turmas. Dessa forma, ao cadastrar uma avaliação, o sistema armazenará seu tipo, se é uma prova, resumo, apresentação, etc, peso e a data do agendamento.

Uma vez aplicadas as avaliações, estas são corrigidas e suas notas podem ser registradas no sistema. Os docentes podem publicar cada nota selecionando a turma, a disciplina, a avaliação e o aluno. Após essa seleção, o professor indica qual a nota do referido aluno e diz se ela está ou não disponível para consulta. Caso o aluno solicite revisão de prova, a referida nota é bloqueada. Após a revisão da prova realizada pelo docente o mesmo divulga a nova nota do aluno, fazendo com que ela seja desbloqueada e esteja novamente disponível para consulta.

As instituições de ensino possuem diversos recursos como *datashows*, laboratórios de informática, auditórios, retro-projetores, etc, para serem utilizados como ferramentas de auxílio ao trabalho dos professores. No portal acadêmico os docentes podem solicitar a alocação de um determinado recurso em uma data e horário. A secretaria acadêmica é responsável por verificar a disponibilidade e registrar a alocação do mesmo.

Para tornar a navegação do módulo de Gerenciamento de Informações Acadêmicas mais fácil, seus usuários têm à disposição um calendário semanal informando-os sobre as atividades que estão cadastradas para eles em uma determinada data. Também estão disponíveis um quadro de avisos e uma lista de eventos, o que colabora como um meio de divulgação, além dos tradicionais cartazes e posters.

1.3.1.1. Consultas Parametrizadas - Geração de Relatórios

Ainda no módulo de Gerenciamento de Informações Acadêmicas, os professores, coordenadores e funcionários da instituição têm à sua disposição a possibilidade de gerar diversos relatórios gerenciais parametrizados.

As consultas parametrizadas possibilitam que os usuários possam gerar relatórios analíticos, ou seja, sob diferentes perspectivas de informações sumarizadas. A geração acontece em detrimento às escolhas das informações desejadas. Ainda há a possibilidade da realização de filtros, no intuito

de restringir a quantidade de dados.

A informação armazenada pode ser comparada com um cubo, isto é, ela pode ser analisada sob diversas perspectivas.

As consultas parametrizadas possibilitam aos usuários levantar informações como, por exemplo, se o fato de um aluno ter tirado uma nota baixa em uma avaliação, tenha sido em detrimento à sua baixa frequência em uma determinada disciplina. Também é possível verificar a evasão de alunos de um determinado curso, identificando qual o período possui a maior taxa. Para os professores, também é possível extrair informações a respeito das notas e frequências dos alunos por eles lançadas.

Esses relatórios colaboram para que a coordenação de cada curso e a direção da instituição possam identificar pontos importantes para auxiliar no processo de tomada de decisão. Também servem para contribuir no processo de melhoria continuada na qualidade do ensino e infra-estrutura oferecidos na instituição. Um ponto importante a ser destacado é o fato de que, com as informações obtidas através desses relatórios, poderiam ser feitas campanhas de marketing direcionadas.

1.3.2. Mural de Estágios

O módulo de Mural de Estágios é um importante facilitador para auxiliar os alunos no processo de inserção no mercado de trabalho. Ele também colabora para facilitar a comunicação entre os estudantes, as instituições de ensino e as empresas. Isto faz com que exista uma maior e melhor integração entre as partes envolvidas neste processo, podendo criar meios para a elaboração de convênios entre as diversas empresas interessadas e as instituições de ensino.

Neste módulo, não somente funcionários, mas também os professores poderão disponibilizar vagas no portal.

Atualmente, observa-se que diversos professores, além do ensino acadêmico, mantêm contato com empresas, as quais normalmente disponibilizam vagas de estágio e gostariam de divulgar este processo seletivo para seus alunos. A divulgação desta informação acaba ficando restrita. Através do portal acadêmico, os professores podem disponibilizar as oportunidades existentes, podendo dessa forma atingir não somente o público próximo a eles, como também os demais alunos.

No momento em que os discentes realizarem seu primeiro acesso a este módulo, eles devem informar se estão ou não interessados em receber informações por e-mail as diversas oportunidades de estágio divulgadas. Caso não queiram receber estas informações, eles continuam podendo consultar, a qualquer momento, as vagas cadastradas.

Ao cadastrar uma vaga, deve ser associada a ela a empresa que está oferecendo a

oportunidade. Caso a empresa não esteja cadastrada no sistema, ela tem ser informada pela secretaria de registro acadêmico com seus dados como: código, nome, CNPJ, endereço completo, telefones para contato e *website*. Os professores podem somente cadastrar vagas de estágio e caso haja a necessidade de informar uma nova empresa, o mesmo deve encaminhar um *e-mail* ao setor responsável por tal procedimento. Feito o registro da empresa no sistema, ela passa a constar como disponível para ser associada a uma vaga. As empresas também podem ser bloqueadas no sistema e excluídas a qualquer momento. As operações de bloqueio e exclusão das empresas são de responsabilidade de quem as cadastra.

Para cadastrar uma vaga, dados como: código, data, nome, descrição, quantidade de vagas oferecidas, pré-requisitos caso haja algum e situação deverão ser informados. As vagas devem ser associadas a um curso específico, de modo que somente os alunos pertencentes a ele possam visualizá-la. Ao ser cadastrada, a vaga passa a constar como disponível, podendo ser consultada pelos estudantes. Se necessário ela pode ter seu estado alterado para indisponível. As vagas, podem ser excluídas a qualquer momento.

1.3.3. Ensino à Distância (EAD)

O ensino à distância vem sendo muito utilizado em diversas instituições de ensino e muitas delas já oferecem cursos de graduação e pós-graduação à distância, disciplinas *online*, cursos de férias, entre outros mais. Este tipo de ensino colabora para o processo de formação continuada e visa aumentar a capacitação profissional de seus alunos.

Para disponibilizar um ambiente propício para o EAD, o portal acadêmico será integrado ao *software moodle*. Entretanto, o controle de acesso a este ambiente é feito através do portal, isto é, para que os alunos possam ter acesso aos cursos disponíveis, eles devem ser previamente cadastrados no sistema. Dessa forma, não há a necessidade de efetuar cadastros de usuários no *moodle*.

A aplicação *moodle* (*Modular Object-Oriented Dynamic Learning Environment*) começou a ser desenvolvida na década de 90 por Martin Dougiamas e teve sua primeira versão divulgada ao mundo em 2002. Influenciado pelo construcionismo social, Martin desenvolveu o *moodle*. Segundo ele esta abordagem “...*não só trata a aprendizagem como uma atividade social, mas focaliza a atenção na aprendizagem que acontece enquanto construímos ativamente artefatos (como textos, por exemplo), para que outros vejam ou utilizem.*”

O *moodle* é um software *opensource* e de acordo com seu criador a filosofia é “...*mantê-lo Aberto e Gratuito. Tenho a profunda convicção da importância do acesso irrestrito à educação e do ensino enriquecido (empowered teaching); e o Moodle é a principal forma em que eu posso*

contribuir para a realização desses ideais”.

A ferramenta é de fácil interação permitindo aos usuários a criação de diversos cursos e administração dos mesmos. Dentre as funcionalidades disponíveis estão: criação de cursos, inscrição nos mesmos, divulgação de notas e materiais, fórum de discussão, chat, dentre outros. Tudo isso “customizável” e de fácil manipulação.

Para auxiliar na composição deste trabalho visitamos o *moodle* de algumas universidades, tais como: da Universidade Federal Fluminense (UFF) e do Instituto Superior de Educação La Salle (Unilasalle), do qual tivemos a oportunidade de usufruir como usuários (alunos).

1.3.4. Informações Gerais

O sistema possui todo um apoio quanto à sua administração, ou seja, controle de perfis de acesso, usuários, permissões, auditoria de navegabilidade, auditoria de registros das operações realizadas no banco de dados, entre outras funcionalidades mais.

Na intenção de oferecer um alto nível de segurança e integridade, os usuários, quando cadastrados, recebem um *e-mail* de boas vindas contendo informações pertinentes àquele determinado usuário. Os dados informados na correspondência eletrônica são: o nome de usuário (*login*) e uma senha inicial, aleatória, que é gerada automaticamente pelo sistema. Além destas informações o usuário também recebe endereço eletrônico (*link*) que, obrigatoriamente, deve ser acessado, para que o sistema possa validar e ativar a conta do usuário em questão.

Após acessar o *link*, o usuário é direcionado para o Portal Acadêmico Innova. Lá ele deve informar seus dados – nome de usuário e senha – para realizar seu primeiro acesso. No seu primeiro acesso, o usuário recebe uma mensagem na tela inicial do portal informando-o para alterar sua senha, a fim de que não precise memorizar a senha gerada pelo sistema que, por sua vez, é complexa.

Colaborando para aumentar o nível de segurança para seus usuários, o sistema assegura que, se uma sessão do sistema for deixada aberta e a mesma estiver inativa por um determinado período (ajustável de acordo com as necessidades do cliente), automaticamente o cliente/usuário será desconectado.

Como forma de restringir o acesso de seus usuários a determinados módulos e funcionalidades, o sistema possui perfis de acesso, a fim de que eles possam navegar e buscar informações conforme os graus de sigilo implementados. O cadastramento destes perfis consiste em definir um conjunto de regras e permissões, que determinam quais funções cada usuário pode realizar dentro de cada módulo implementado. Os responsáveis pela definição e cadastramento de cada perfil são os administradores do sistema. Após definidos os perfis de acesso, os usuários

deverão ser cadastrados no sistema, cuja função também é de cunho dos administradores.

Caso haja algum usuário que necessite de uma permissão que não tenha sido contemplada em seu perfil, o sistema também permite a atribuição de mais funções em tempo de execução, ou seja, não há a necessidade que um usuário se desconecte do sistema e conecte novamente para que a permissão extra seja atribuída.

Para que o ambiente do portal acadêmico possa estar disponível, um administrador de banco de dados *Database Administrator* (DBA) pode carregar as tabelas iniciais do sistema.

Se existir algum outro sistema na instituição de ensino que contenha os dados inerentes aos alunos, professores e funcionários, o DBA carrega estes dados no ambiente do portal acadêmico. Caso contrário, a secretaria de registro acadêmico fica com a incumbência de alimentar o sistema com essas informações.

O sistema possui dois tipos distintos de auditoria. O primeiro, tem seus registros guardados automaticamente em função da navegação através dos módulos do sistema. Por ele é possível identificar quais partes do sistema são as mais acessadas pelos usuários, qual módulo um determinado usuário está acessando em um certo momento, além de outras facilidades. No segundo, a medida em que informações forem inseridas, alteradas ou excluídas do sistema, registros serão criados em tabelas específicas de auditoria do sistema.

A funcionalidade de auditoria será realizada pela aplicação através de gatilhos (*triggers*) da aplicação. Estes por sua vez são altamente portáteis, garantindo uma independência de sistema gerenciador de banco de dados (SGBD). Isso faz com que a aplicação não fique restrita a um determinado SGBD. Um outro fator que colabora para que as *triggers* sejam de responsabilidade da aplicação é diminuir a sobrecarga no banco de dados.

1.4. Requisitos do Sistema

A seguir serão listados os requisitos iniciais levantados para o sistema:

- Carga Inicial das Tabelas;
- Carga das Tabelas das Consultas Parametrizadas das Avaliações;
- Carga das Tabelas das Consultas Parametrizadas das Frequências;
- Cadastro dos Usuários;
- Cadastro de Perfis de Acesso;
- Cadastro da Instituição de Ensino;
- Cadastro dos Cursos da Instituição de Ensino;
- Cadastro das Grades Curriculares dos Cursos;

- Cadastro das Disciplinas de um Curso;
- Cadastro das Ementas das Disciplinas;
- Cadastro dos Conteúdos Programáticos das Disciplinas;
- Cadastro das Turmas das Disciplinas;
- Cadastro dos Alunos;
- Cadastro dos Professores;
- Cadastro de Vagas de Estágio;
- Cadastro das Empresas;
- Cadastro dos Recursos da Instituição;
- Alocação dos Recursos da Instituição;
- Cadastro de Locais da Instituição;
- Cadastro das Salas dos Locais da Instituição;
- Agendamento de Avaliações;
- Disponibilização de Materiais;
- Lançamento de Frequência dos Alunos nas suas Turmas e Disciplinas;
- Lançamento de Notas das Avaliações;
- Lançamento de Conteúdo Ministrado;
- Associação de Descrição aos Materiais Disponibilizados;
- Solicitar Alocação de Recurso;
- Consultas Parametrizadas das Avaliações;
- Consultas Parametrizadas das Frequências;
- Consulta de Notas das Avaliações;
- Realizar Pré-inscrição nas Disciplinas;
- Solicitar Declaração de Curso;
- Consulta de Datas das Avaliações;
- Consulta de Grade Curricular das Disciplinas de um Curso;
Consulta de Grade das Disciplinas por Semestre;
- Consulta de Grade de Horários das Disciplinas em um Semestre;
- Consulta de Ementas Disciplinares;
- Consulta de Conteúdos Programáticos Ministrados;
- Cadastro de Pré-Requisitos das Disciplinas;

- Aprovação de Pré-Inscrição em Disciplina;
- Cadastro de Divulgação de Eventos/Avisos da Instituição;
- Acesso do Sistema via *Web*;

1.5. Propostas de solução para o portal acadêmico:

As propostas não estão direcionadas para o desenvolvimento de um novo portal para as instituições de ensino, mas sim para agregar valor aos seus *websites*. A seguir serão apresentadas três propostas de solução para a implementação do portal acadêmico:

1.5.1. Primeira alternativa – portal desenvolvido em linguagem de programação Java com Sistema Gerenciador de Banco de Dados (SGBD) Microsoft SQL Server 2005:

Esta alternativa baseia-se em uma programação voltada para a *web*, utilizando a linguagem de programação Java, com SGBD SQL Server 2005 desenvolvido pela Microsoft.

Todos os usuários envolvidos no processo poderão ter acesso ao portal, justamente pelo fato de o mesmo estar disponível na internet. A linguagem Java, criada pela *Sun (Stanford University Network) Microsystems*, é uma das linguagens de programação mais utilizadas no mundo. Ela utiliza para o desenvolvimento, os conceitos e metodologias de orientação a objeto, é multi-plataforma, isto é, funciona independentemente do sistema operacional, é altamente portátil, dentre outras características mais. Neste caso, o tempo de desenvolvimento do trabalho seria um pouco extenso devido à grande complexidade da linguagem.

O SGBD Microsoft SQL Server 2005 é uma plataforma abrangente de banco de dados, que fornece recursos de gerenciamento de dados de classe empresarial com ferramentas de BI (*Business Intelligence*). É uma ferramenta altamente robusta utilizada por diversas empresas de diferentes portes no mercado. O objetivo de todo SGBD é trazer para si a responsabilidade de gerenciar o acesso, manipulação e organização dos dados.

Requisitos Atendidos:

- Carga Inicial das Tabelas;
- Carga das Tabelas das Consultas Parametrizadas das Avaliações;
- Carga das Tabelas das Consultas Parametrizadas das Frequências;
- Cadastro dos Usuários;

- Cadastro de Perfis de Acesso;
- Cadastro da Instituição de Ensino;
- Cadastro dos Cursos da Instituição de Ensino;
- Cadastro das Grades Curriculares dos Cursos;
- Cadastro das Disciplinas de um Curso;
- Cadastro das Ementas das Disciplinas;
- Cadastro dos Conteúdos Programáticos das Disciplinas;
- Cadastro das Turmas das Disciplinas;
- Cadastro dos Alunos;
- Cadastro dos Professores;
- Cadastro de Vagas de Estágio;
- Cadastro das Empresas;
- Cadastro dos Recursos da Instituição;
- Alocação dos Recursos da Instituição;
- Cadastro de Locais da Instituição;
- Cadastro das Salas dos Locais da Instituição;
- Agendamento de Avaliações;
- Disponibilização de Materiais;
- Lançamento de Frequência dos Alunos nas suas Turmas e Disciplinas;
- Lançamento de Notas das Avaliações;
- Lançamento de Conteúdo Ministrado;
- Associação de Descrição aos Materiais Disponibilizados;
- Solicitar Alocação de Recurso;
- Consultas Parametrizadas das Avaliações;
- Consultas Parametrizadas das Frequências;
- Consulta de Notas das Avaliações;
- Realizar Pré-inscrição nas Disciplinas;
- Solicitar Declaração de Curso;
- Consulta de Datas das Avaliações;
- Consulta de Grade Curricular das Disciplinas de um Curso;
- Consulta de Grade das Disciplinas por Semestre;

- Consulta de Grade de Horários das Disciplinas em um Semestre;
- Consulta de Ementas Disciplinares;
- Consulta de Conteúdos Programáticos Ministrados;
- Cadastro de Pré-Requisitos das Disciplinas;
- Aprovação de Pré-Inscrição em Disciplina;
- Cadastro de Divulgação de Eventos/Avisos da Instituição;
- Acesso do Sistema via *Web*;

Data Prevista:

- Início: 20/08/2008
- Fim: 20/02/2010
- Tempo Previsto: 18 meses

Fases	Analista de Sistema	DBA	Programador	Analista de Teste	Web Designer
Levantamento de requisitos	40 dias				
Análise	35 dias				
Projeto Lógico	33 dias	33 dias			
Carga e atualização das tabelas		20 dias	30 dias		
Implementação		45 dias	70 dias	35 dias	70 dias
Testes			40 dias	20 dias	25 dias
Implantação e Aceite			24 dias	12 dias	15 dias
TOTAL	108 dias	98 dias	164 dias	67 dias	110 dias

Tabela 1 – Fases do Sistema referente à primeira alternativa de implementação

Quantidade de horas trabalhadas:

- Analista de Sistema: 8h/dia
- Administrador de banco de dados: 8h/dia
- Programador: 8h/dia
- Analista de Teste: 8h/dia
- *Web Designer*: 8h/dia

Valor dos Salários/Hora

- Analista de Sistema: R\$ 80,00
- Administrador de banco de dados: R\$ 120,00
- Programador: R\$ 70,00
- Analista de Teste: R\$ 60,00
- *Web Designer*: R\$ 50,00

O tempo estimado para construção da solução são de 547 dias.

Custo Previsto:

Item	Quantidade	Valor Unitário	Valor Total
Servidor: Intel Xeon Quad Core 2.0Ghz, 4Gb Memória DDR2, 2xHD SATA II 250Gb.	01 Unidade	R\$ 3.970,00	R\$ 3.970,00
TOTAL			R\$ 3.970,00

Tabela 2 – Custo de hardware previsto referente à primeira alternativa de implementação

Item	Licença (qtd)	Valor Unitário	Valor
Eclipse	03	R\$ 0,00	R\$ 0,00
Microsoft Windows 2003 Server Standard Edition	01	R\$ 3.235,00	R\$ 3.235,00
Microsoft SQL Server 2005 Enterprise Edition	01	R\$ 14.446,00	R\$ 14.446,00
TOTAL			R\$ 17.681,00

Tabela 3 – Custo de software previsto referente à primeira alternativa de implementação

Profissional	Qtd. Horas	Valor/Hora	Valor Total
Analista de Sistema	864	R\$ 80,00	R\$ 69.120,00
Administrador de Banco de Dados	784	R\$ 120,00	R\$ 94.080,00
Programador	1312	R\$ 70,00	R\$ 91.840,00
Analista de Teste	536	R\$ 60,00	R\$ 32.160,00
Web Designer	880	R\$ 50,00	R\$ 44.000,00
TOTAL			R\$ 331.200,00

Tabela 4 – Custo de peopleware previsto referente à primeira alternativa de implementação

1.5.2. Segunda alternativa – portal desenvolvido em linguagem de programação PHP com Sistema Gerenciador de Banco de Dados (SGBD) MySQL:

A linguagem de programação PHP é uma linguagem interpretada, livre e muito utilizada para gerar conteúdo dinâmico na Internet. Também trabalha sob os conceitos e metodologias da orientação a objetos e é altamente modularizada, o que a torna uma boa opção para instalação e utilização em servidores *web*. Seus tipos de dados, sintaxe e funções são muito parecidos com os da linguagem C e C++. Também pode ser embarcada no código HTML.

O MySQL é um dos SGBD mais populares do mundo e é de fácil integração com aplicações desenvolvidas em PHP. Além disso, os requerimentos de *hardware* para o funcionamento dele são baixos, é multi-plataforma, isto é, pode ser instalado e utilizado em diversos sistemas operacionais (praticamente todos), é altamente portátil e possui um bom desempenho e estabilidade.

O melhor desempenho dessa combinação se dá através de um servidor Linux e para atender a todos os requisitos do trabalho, o desenvolvimento do portal acadêmico irá durar 16 (dezesesseis) meses.

Requisitos Atendidos:

- Carga Inicial das Tabelas;
- Carga das Tabelas das Consultas Parametrizadas das Avaliações;
- Carga das Tabelas das Consultas Parametrizadas das Frequências;
- Cadastro dos Usuários;
- Cadastro de Perfis de Acesso;
- Cadastro da Instituição de Ensino;
- Cadastro dos Cursos da Instituição de Ensino;
- Cadastro das Grades Curriculares dos Cursos;
- Cadastro das Disciplinas de um Curso;
- Cadastro das Ementas das Disciplinas;
- Cadastro dos Conteúdos Programáticos das Disciplinas;
- Cadastro das Turmas das Disciplinas;
- Cadastro dos Alunos;
- Cadastro dos Professores;
- Cadastro de Vagas de Estágio;

- Cadastro das Empresas;
- Cadastro dos Recursos da Instituição;
- Alocação dos Recursos da Instituição;
- Cadastro de Locais da Instituição;
- Cadastro das Salas dos Locais da Instituição;
- Agendamento de Avaliações;
- Disponibilização de Materiais;
- Lançamento de Frequência dos Alunos nas suas Turmas e Disciplinas;
- Lançamento de Notas das Avaliações;
- Lançamento de Conteúdo Ministrado;
- Associação de Descrição aos Materiais Disponibilizados;
- Solicitar Alocação de Recurso;
- Consultas Parametrizadas das Avaliações;
- Consultas Parametrizadas das Frequências;
- Consulta de Notas das Avaliações;
- Realizar Pré-inscrição nas Disciplinas;
- Solicitar Declaração de Curso;
- Consulta de Datas das Avaliações;
- Consulta de Grade Curricular das Disciplinas de um Curso;
- Consulta de Grade das Disciplinas por Semestre;
- Consulta de Grade de Horários das Disciplinas em um Semestre;
- Consulta de Ementas Disciplinares;
- Consulta de Conteúdos Programáticos Ministrados;
- Cadastro de Pré-Requisitos das Disciplinas;
- Aprovação de Pré-Inscrição em Disciplina;
- Cadastro de Divulgação de Eventos/Avisos da Instituição;
- Acesso do Sistema via *Web*;

Data Prevista:

- Início: 20/08/2008
- Fim: 20/12/2009
- Tempo Previsto: 16 meses

Fases	Analista de Sistema	DBA	Programador	Analista de Teste	Web Designer
Levantamento de requisitos	40 dias				
Análise	35 dias				
Projeto Lógico	33 dias	30 dias			
Carga e atualização das tabelas		15 dias	30 dias	10 dias	
Implementação		15 dias	60 dias	10 dias	70 dias
Testes			40 dias	30 dias	25 dias
Implantação e Aceite			15 dias	8 dias	15 dias
TOTAL	108 dias	60 dias	145 dias	58 dias	110 dias

Tabela 5 - Fases do Sistema referente à segunda alternativa de implementação

Quantidade de horas trabalhadas:

- Analista de Sistema: 8 horas/dia
- Administrador de banco de dados: 8 horas/dia
- Programador: 8 horas/dia
- Analista de Teste: 8h/dia
- *Web Designer*: 8 horas/dia

Valor dos Salários/Hora:

- Analista de Sistema: R\$ 80,00
- Administrador de banco de dados: R\$ 90,00
- Programador: R\$ 60,00
- Analista de Teste: R\$ 60,00
- *Web Designer*: R\$ 50,00

O tempo estimado para construção da solução são de 481 dias.

Custo Previsto:

Item	Quantidade	Valor Unitário	Valor Total
Servidor: Intel Xeon Dual Core 1.8Ghz, 2Gb Memória DDR2, 2xHD SATA II 250Gb	01 Unidade	R\$ 2.500,00	R\$ 2.500,00
TOTAL			R\$ 2.500,00

Tabela 6 – Custo de hardware previsto referente à segunda alternativa de implementação

Item	Licença (qtd)	Valor Unitário	Valor
Ubuntu Server Edition 8.04	01	R\$ 0,00	R\$ 0,00
MySQL	01	R\$ 0,00	R\$ 0,00
Zend	01	R\$ 0,00	R\$ 0,00
TOTAL			R\$ 0,00

Tabela 7 – Custo de software previsto referente à segunda alternativa de implementação

Profissional	Qtd. horas	Valor/Hora	Valor Total
Analista de Sistema	864	R\$80,00	R\$ 69.120,00
Adm. de Banco de Dados	480	R\$90,00	R\$ 43.200,00
Programador	1160	R\$60,00	R\$ 69.600,00
Analista de Teste	464	R\$60,00	R\$ 27.840,00
Web Designer	880	R\$50,00	R\$ 44.000,00
TOTAL			R\$ 253.760,00

Tabela 8 – Custo de peopleware previsto referente à segunda alternativa de implementação

1.5.3. Terceira alternativa – portal desenvolvido em linguagem de programação Ruby, utilizando o *framework* Ruby on Rails e com Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL 8.3:

Esta alternativa sugere a implementação do portal acadêmico desenvolvido em linguagem de programação Ruby, utilizando o *framework* Ruby on Rails com o SGBD PostgreSQL 8.3.

Ruby é uma linguagem de programação interpretada, com tipagem dinâmica e forte, orientada a objetos que foi profundamente influenciada por linguagens como Perl, Python e Smalltalk. O grande diferencial do Ruby é o fato de ser uma linguagem humana, que preza pela legibilidade e clareza do código. Ela é extremamente simples e possui uma orientação a objetos clara e eficiente. Assim como o PHP, é *opensource* e multiplataforma, entretanto apresenta melhor desempenho em sistemas operacionais Linux.

Ruby on Rails é um *framework* recente que surgiu em meados de 2002, tendo apresentado um crescimento exponencial em sua utilização. O mesmo também preza por conceitos de agilidade e produtividade, chegando a ser vinte vezes mais produtivo que alternativas em outras linguagens.

O SGBD apresentado nesta alternativa de solução é o PostgreSQL. Assim como a linguagem de programação utilizada, ele é *opensource*, gratuito e, da mesma forma como o Oracle, também é relacional-objeto, isto é, além de implementar características de um SGBD relacional, também apresenta características de orientação a objetos, como herança e tipos personalizados.

No segmento deste tipo de sistema, o PostgreSQL é reconhecido por ser robusto, ter alta integridade, confiabilidade e estabilidade. A versão utilizada no desenvolvimento desta solução é a 8.3, a última disponível em seu *website*.

Requisitos Atendidos:

- Carga Inicial das Tabelas;
- Carga das Tabelas das Consultas Parametrizadas das Avaliações;
- Carga das Tabelas das Consultas Parametrizadas das Frequências;
- Cadastro dos Usuários;
- Cadastro de Perfis de Acesso;
- Cadastro da Instituição de Ensino;
- Cadastro dos Cursos da Instituição de Ensino;
- Cadastro das Grades Curriculares dos Cursos;
- Cadastro das Disciplinas de um Curso;

- Cadastro das Ementas das Disciplinas;
- Cadastro dos Conteúdos Programáticos das Disciplinas;
- Cadastro das Turmas das Disciplinas;
- Cadastro dos Alunos;
- Cadastro dos Professores;
- Cadastro de Vagas de Estágio;
- Cadastro das Empresas;
- Cadastro dos Recursos da Instituição;
- Alocação dos Recursos da Instituição;
- Cadastro de Locais da Instituição;
- Cadastro das Salas dos Locais da Instituição;
- Agendamento de Avaliações;
- Disponibilização de Materiais;
- Lançamento de Frequência dos Alunos nas suas Turmas e Disciplinas;
- Lançamento de Notas das Avaliações;
- Lançamento de Conteúdo Ministrado;
- Associação de Descrição aos Materiais Disponibilizados;
- Solicitar Alocação de Recurso;
- Consultas Parametrizadas das Avaliações;
- Consultas Parametrizadas das Frequências;
- Consulta de Notas das Avaliações;
- Realizar Pré-inscrição nas Disciplinas;
- Solicitar Declaração de Curso;
- Consulta de Datas das Avaliações;
- Consulta de Grade Curricular das Disciplinas de um Curso;
- Consulta de Grade das Disciplinas por Semestre;
- Consulta de Grade de Horários das Disciplinas em um Semestre;
- Consulta de Ementas Disciplinares;
- Consulta de Conteúdos Programáticos Ministrados;
- Cadastro de Pré-Requisitos das Disciplinas;
- Aprovação de Pré-Inscrição em Disciplina;

- Cadastro de Divulgação de Eventos/Avisos da Instituição;
- Acesso do Sistema via *Web*;

Data Prevista:

- Início: 20/08/2008
- Fim: 20/09/2009
- Tempo Previsto: 13 meses

Fases	Analista de Sistema	DBA	Programador	Analista de Teste	Web Designer
Levantamento de requisitos	40 dias				
Análise	35 dias				
Projeto Lógico	33 dias	25 dias			
Carga e atualização das tabelas		10 dias	10 dias	10	
Implementação		10 dias	30 dias	12	70 dias
Testes			40 dias	20	25 dias
Implantação e Aceite			24 dias	8	15 dias
TOTAL	108 dias	45 dias	104 dias	50 dias	90 dias

Tabela 9 – Fases do Sistema referente à terceira alternativa de implementação

Horas trabalhadas/dia:

- Analista de Sistema: 8 horas/dia
- Administrador de banco de dados: 8 horas/dia
- Programador: 8 horas/dia
- Analista de Teste: 8 horas/dia
- *Web Designer*: 8 horas/dia

Valor dos Salários/Hora:

- Analista de Sistema: R\$ 80,00
- Administrador de banco de dados: R\$ 100,00
- Programador: R\$ 80,00
- Analista de Teste: R\$ 60,00
- *Web Designer*: R\$ 50,00

O tempo estimado para construção da solução é de aproximadamente 397 dias.

Custo Previsto:

Item	Quantidade	Valor Unitário	Valor Total
Servidor para a aplicação: Intel Xeon Quad Core 2.5Ghz, 4Gb Memória DDR3, 2xHD SATA II 250Gb	01 Unidade	R\$ 4.200,00	R\$ 4.200,00
TOTAL			R\$ 4.200,00

Tabela 10 – Custo de hardware previsto referente à terceira alternativa de implementação

Item	Licença (qtd)	Valor Unitário	Valor
Ubuntu Server Edition 8.04	01	R\$ 0,00	R\$ 0,00
Oracle 10g XE	01	R\$ 0,00	R\$ 0,00
Ruby on Rails	01	R\$ 0,00	R\$ 0,00
TOTAL			R\$ 0,00

Tabela 11 – Custo de software previsto referente à terceira alternativa de implementação

Profissional	Qtd horas	Valor/Hora	Valor Total
Analista de Sistema	864	R\$80,00	R\$ 69.120,00
Adm. de Banco de Dados	360	R\$100,00	R\$ 36.000,00
Programador	832	R\$80,00	R\$ 66.560,00
Analista de Teste	400	R\$60,00	R\$ 24.000,00
Web Designer	720	R\$50,00	R\$ 36.000,00
TOTAL			R\$ 231.680,00

Tabela 12 – Custo de peopleware previsto referente à terceira alternativa de implementação

1.6. Proposta para implementação do ambiente

Dentre as alternativas apresentadas anteriormente, utiliza-se para a implementação deste trabalho a terceira alternativa. Como já explanado anteriormente, a proposta de solução utiliza a linguagem de programação *Ruby* com o *framework Ruby on Rails* e Sistema Gerenciador de Banco de Dados PostgreSQL 8.3.

Esta alternativa foi escolhida em detrimento das outras, pois apresenta o melhor tempo de implantação do sistema, bem como une o que há de mais moderno nas melhores práticas de desenvolvimento de sistema, com a robustez das tecnologias já existentes.

A linguagem Ruby foi projetada tanto para a programação em grande escala quanto para codificação rápida, permitindo suporte a orientação a objetos simples e prático.

O framework Ruby on Rails utiliza a metodologia de boas práticas no seu desenvolvimento, isto é, sempre que algo é criado, como um *model* ou um *controller*, automaticamente é gerado uma série de testes que depois podem ser melhorados. Possui também vários *plugins* para desenvolvimento, tais como: *Test Driven Development* (TDD) e *Behavior Driven Development* (BDD) que se integram perfeitamente ao *framework*.

Um dos conceitos modernos que o *framework Ruby on Rails* contempla é o *Convention Over Configuration* (COC). No dia-a-dia da programação, utiliza-se convenções, em geral, para facilitar o entendimento e a manutenção por parte de outros desenvolvedores. Levando isto em consideração e sabendo que o tempo gasto para criar arquivos de configuração em alguns frameworks de outras linguagens é extremamente alto, decidiu-se adotar esse conceito de desenvolvimento.

Outro conceito implementado pelo *Rails* é o *Don't Repeat Yourself* (DRY). Que consiste em uma técnica para definir nomes, propriedades e códigos em um lugar somente e reaproveitar essas informações em outros lugares. Assim como na Orientação a Objetos este conceito prega a reutilização de código de forma bastante prática e eficiente. Desta forma permite um desenvolvimento rápido, reduz o tempo de codificação e diminui consideravelmente a quantidade de erros no código.

Esta alternativa utiliza o Sistema Gerenciador de Banco de Dados PostgreSQL que é reconhecido por sua robustez, integridade, confiabilidade e estabilidade. A versão utilizada no desenvolvimento deste trabalho é a 8.3.

Além dos conceitos apresentados acima, outros também são levados em consideração nas diversas etapas de desenvolvimento da aplicação. Com isso será possível oferecer um produto moderno, confiável, em um prazo curto e com o custo final de investimento atraente e diferenciado.

2. Projeto Lógico

2.1. Lista de Eventos:

Abaixo segue a lista de eventos do sistema:

No.	Evento	Ação	Responsável
1	DBA realiza carga inicial das tabelas no sistema	Realizar Carga Inicial das Tabelas	DBA
2	DBA realiza carga das tabelas de consultas parametrizadas das avaliações	Realizar Carga das Tabelas de Consulta das Avaliações	DBA
3	DBA realiza carga das tabelas de consultas parametrizadas das frequências	Realizar Carga das Tabelas de Consulta das Frequências	DBA
4	Administrador cadastra usuários	Cadastrar Usuários	Administrador do sistema
5	Administrador cadastra perfis de usuários,	Cadastrar Perfis de Usuários	Administrador do sistema
6	Secretaria cadastra instituição de ensino	Cadastrar Instituição de Ensino	Secretaria Acadêmica
7	Secretaria cadastra curso	Cadastrar Curso	Secretaria Acadêmica
8	Secretaria cadastra grade	Cadastrar Grade	Secretaria Acadêmica
9	Secretaria cadastra disciplina	Cadastrar Disciplina	Secretaria Acadêmica
10	Secretaria cadastra ementa	Cadastrar Ementa	Secretaria Acadêmica
11	Secretaria cadastra conteúdo programático	Cadastrar Conteúdo Programático	Secretaria Acadêmica
12	Secretaria cadastra turma	Cadastrar Turma	Secretaria Acadêmica
13	Secretaria cadastra alunos	Cadastrar Alunos	Secretaria Acadêmica
14	Secretaria cadastra professores	Cadastrar Professores	Secretaria Acadêmica
15	Secretaria / Professor cadastra vaga de estágio	Cadastrar Vaga Estágio	Secretaria Acadêmica e Professor
16	Secretaria / Professor cadastra empresa	Cadastrar Empresa	Secretaria Acadêmica e Professor
17	Secretaria cadastra recursos da instituição	Cadastrar Recursos	Secretaria Acadêmica

18	Secretaria aloca recurso para professor	Alocar Recurso	Secretaria Acadêmica
19	Secretaria cadastra local	Cadastrar Local	Secretaria Acadêmica
20	Secretaria cadastra sala	Cadastrar Sala	Secretaria Acadêmica
21	Professor agenda avaliação	Agendar Avaliação	Professor
22	Professor disponibiliza material	Disponibilizar material	Professor
23	Professor lança frequência	Lançar frequência	Professor
24	Professor lança notas	Lançar notas	Professor
25	Professor lança conteúdo ministrado	Lançar Conteúdo	Professor
26	Professor associa descrição sobre o material disponibilizado	Associar Comentário	Professor
27	Professor solicita alocação de recurso	Solicitar Recurso	Professor
28	Professor / Coordenador realiza consultas parametrizadas das avaliações dos alunos.	Realizar Consulta Parametrizada das Avaliações	Professor e Coordenador
29	Professor / Coordenador realiza consultas parametrizadas da frequência dos alunos.	Realizar Consulta Parametrizada da Frequência	Professor e Coordenador
30	Aluno consulta notas avaliação	Consultar Notas	Aluno
31	Aluno realiza pré-inscrição em disciplina	Realizar pré-inscrição	Aluno
32	Aluno solicita declaração de curso	Solicitar Declaração	Aluno
33	Aluno consulta datas das avaliações	Consultar Datas Avaliação	Aluno
34	Aluno consulta grade curricular	Consultar Grade Curricular	Aluno
35	Aluno consulta grade disciplinas semestrais	Consultar Grade Semestral	Aluno
36	Aluno consulta grade de horários das disciplinas	Consultar Grade de Horários	Aluno
37	Aluno consulta ementas disciplinares	Consultar Ementas	Aluno
38	Coordenador consulta conteúdos programáticos ministrados.	Consultar Conteúdos	Coordenador
39	Coordenador cadastra pré-requisitos das disciplinas	Cadastrar Pré-requisitos	Coordenador
40	Coordenador aprova pré-inscrição	Aprovar Pré-Inscrição	Secretaria Acadêmica
41	Coordenador / Professor / Secretaria acadêmica cadastra divulgação	Cadastrar Divulgação	Coordenador / Professor / Secretaria Acadêmica

Tabela 13 – Lista de Eventos do Sistema

2.2. Diagramas de Classe Conceitual

2.2.1. Gerenciamento de Informações Acadêmicas

2.2.2. Mural de Estágios

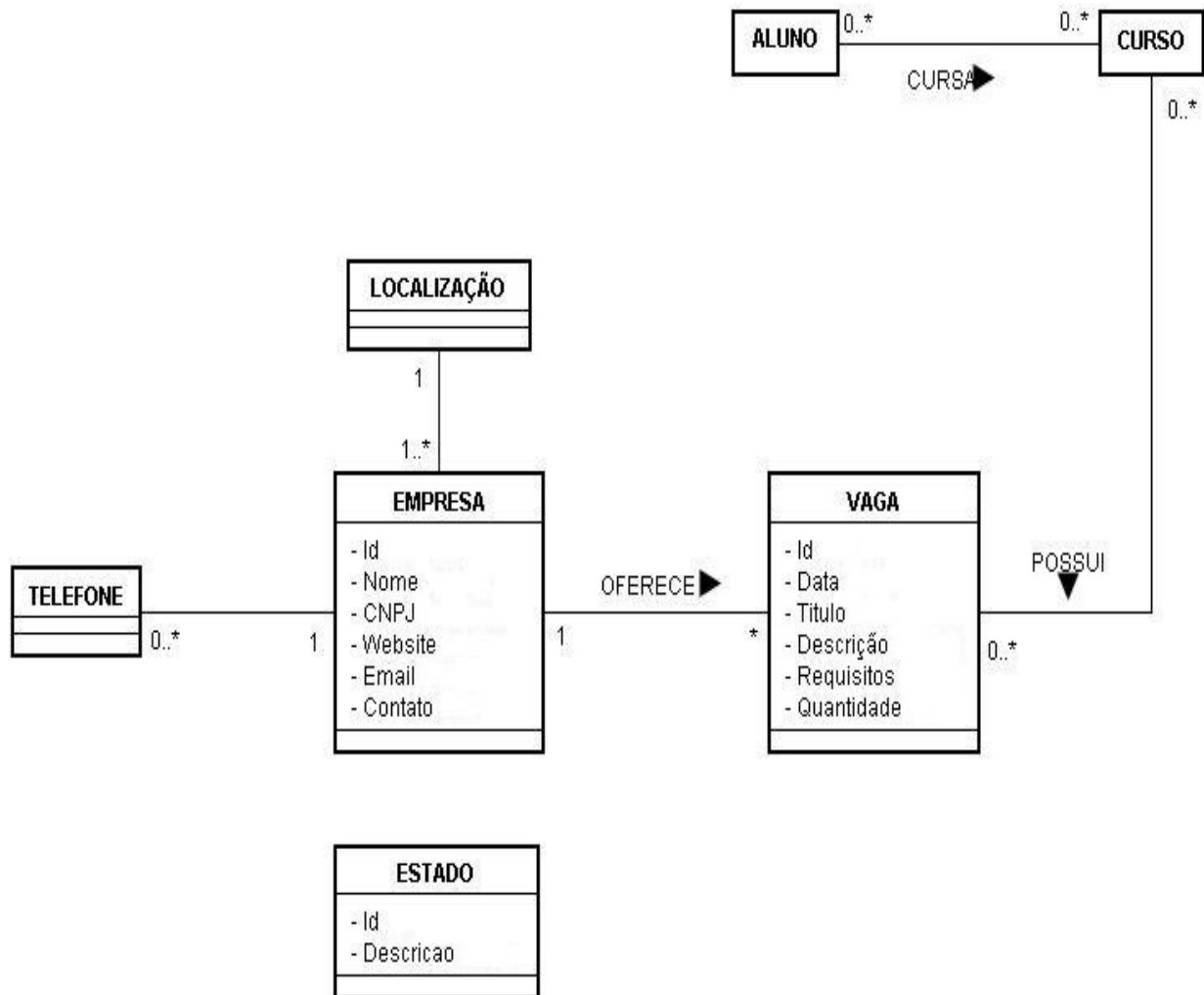


Figura 2 – Diagrama de Classe Conceitual – Mural de Estágios

2.2.3) Consultas Parametrizadas – Avaliações

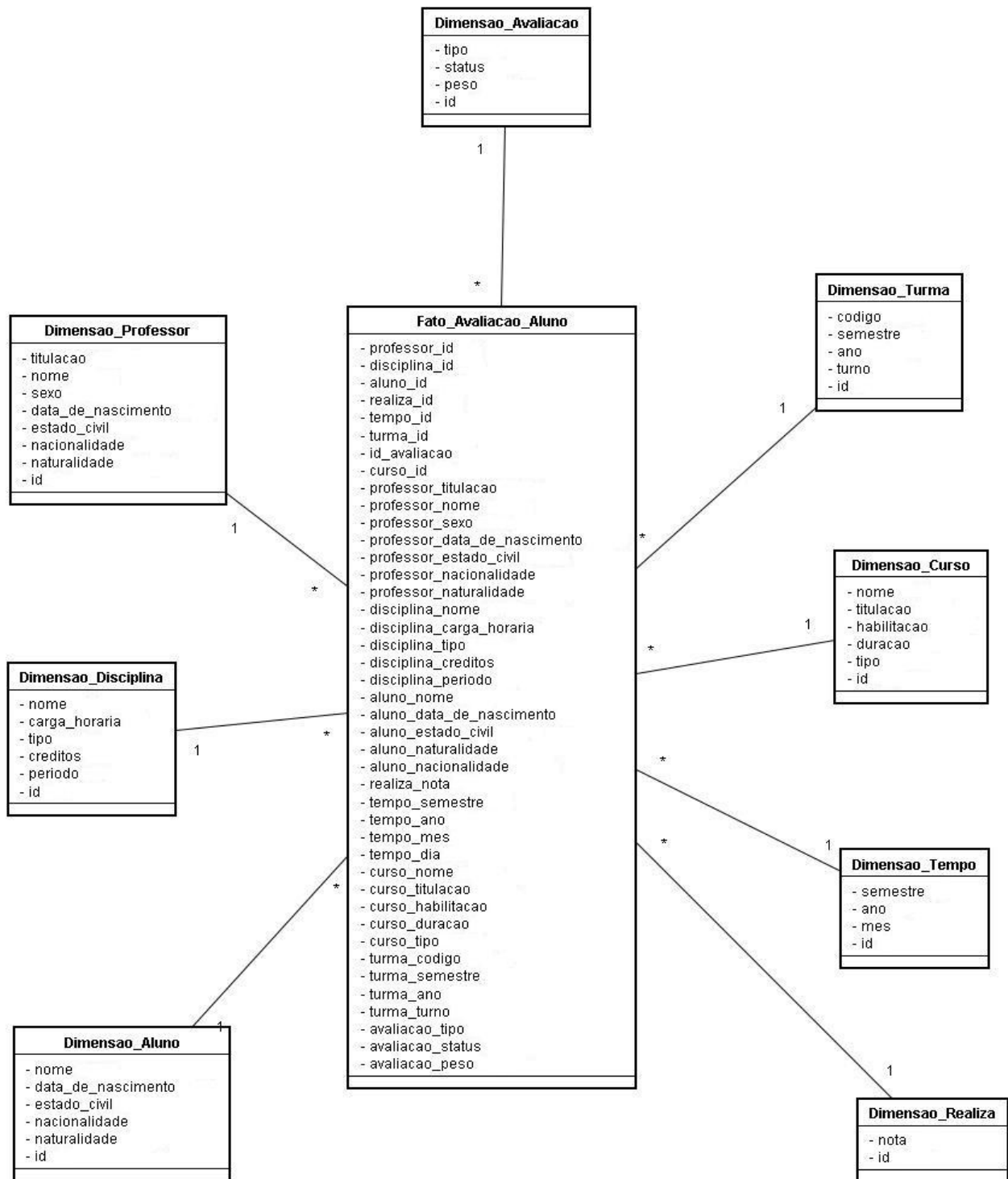


Figura 3 – Diagrama de Classe Conceitual – Consultas Parametrizadas das Avaliações

2.2.4. Consultas Parametrizadas – Frequências

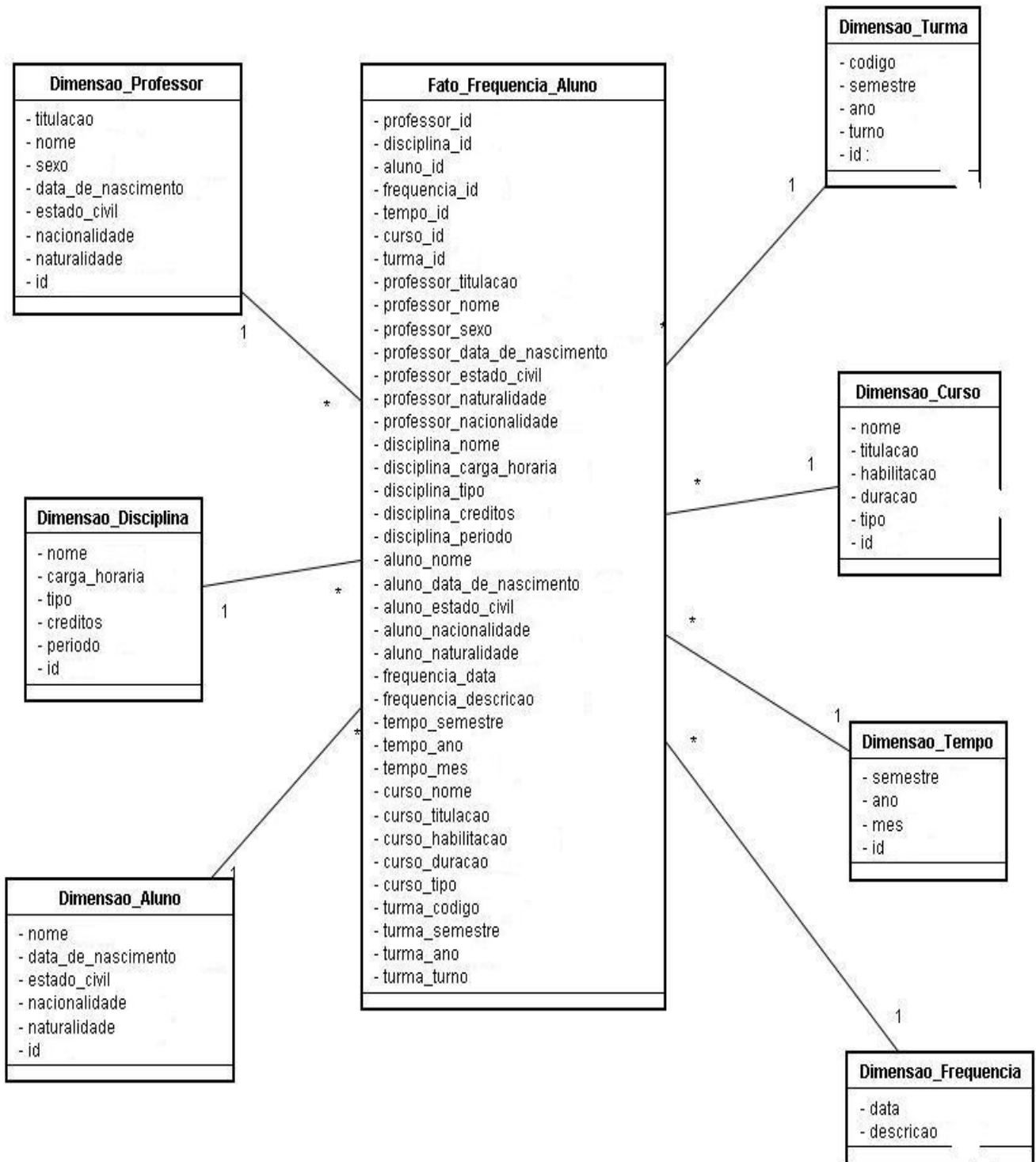


Figura 4 – Diagrama de Classe Conceitual – Consultas Parametrizadas das Frequências

2.3. Diagramas de Casos de Uso

2.3.1. Ator – Secretaria Acadêmica

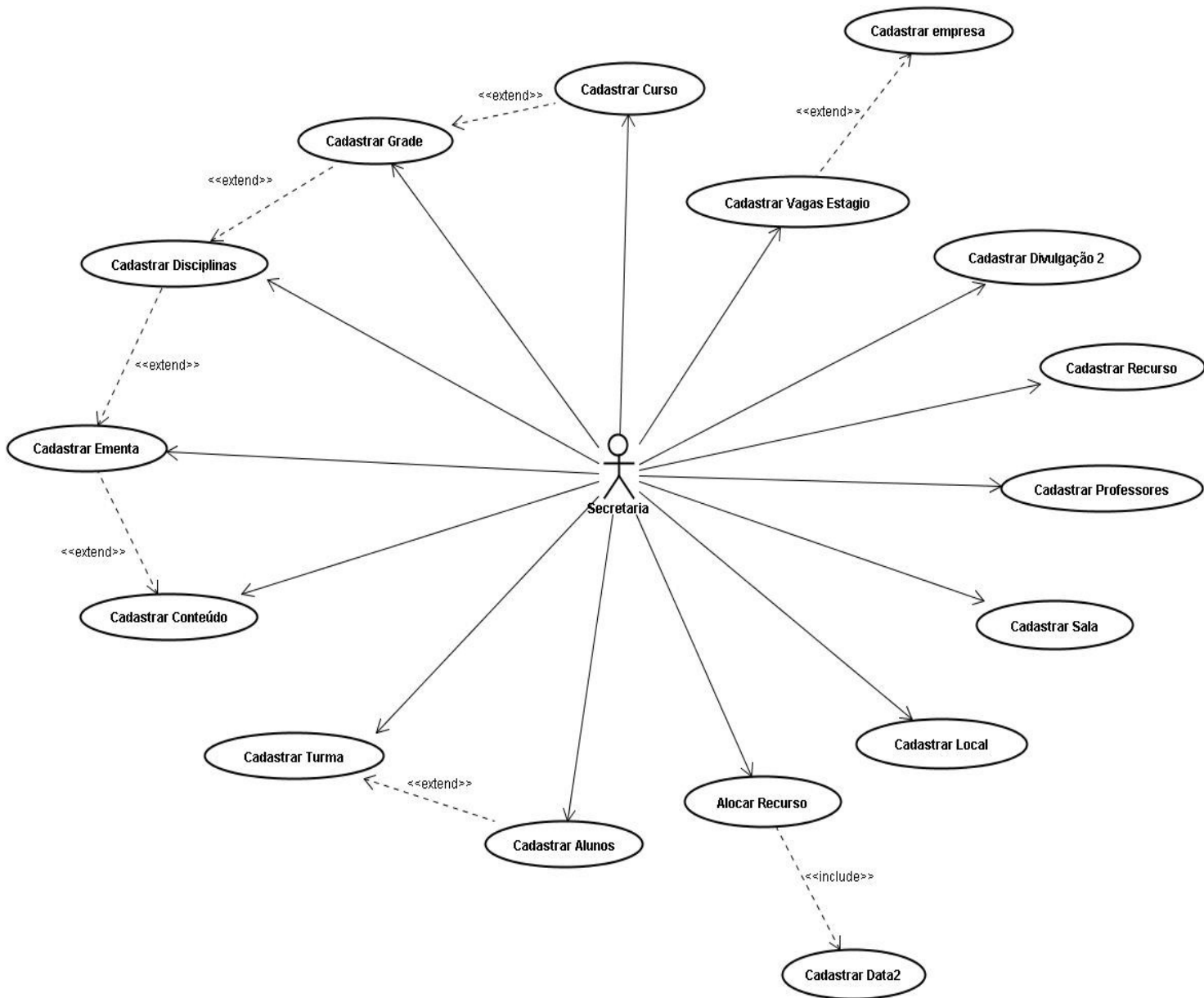


Figura 5 – Diagrama de Casos de Uso – Secretaria Acadêmica

2.3.2) Atores – Administrador, Aluno e DBA

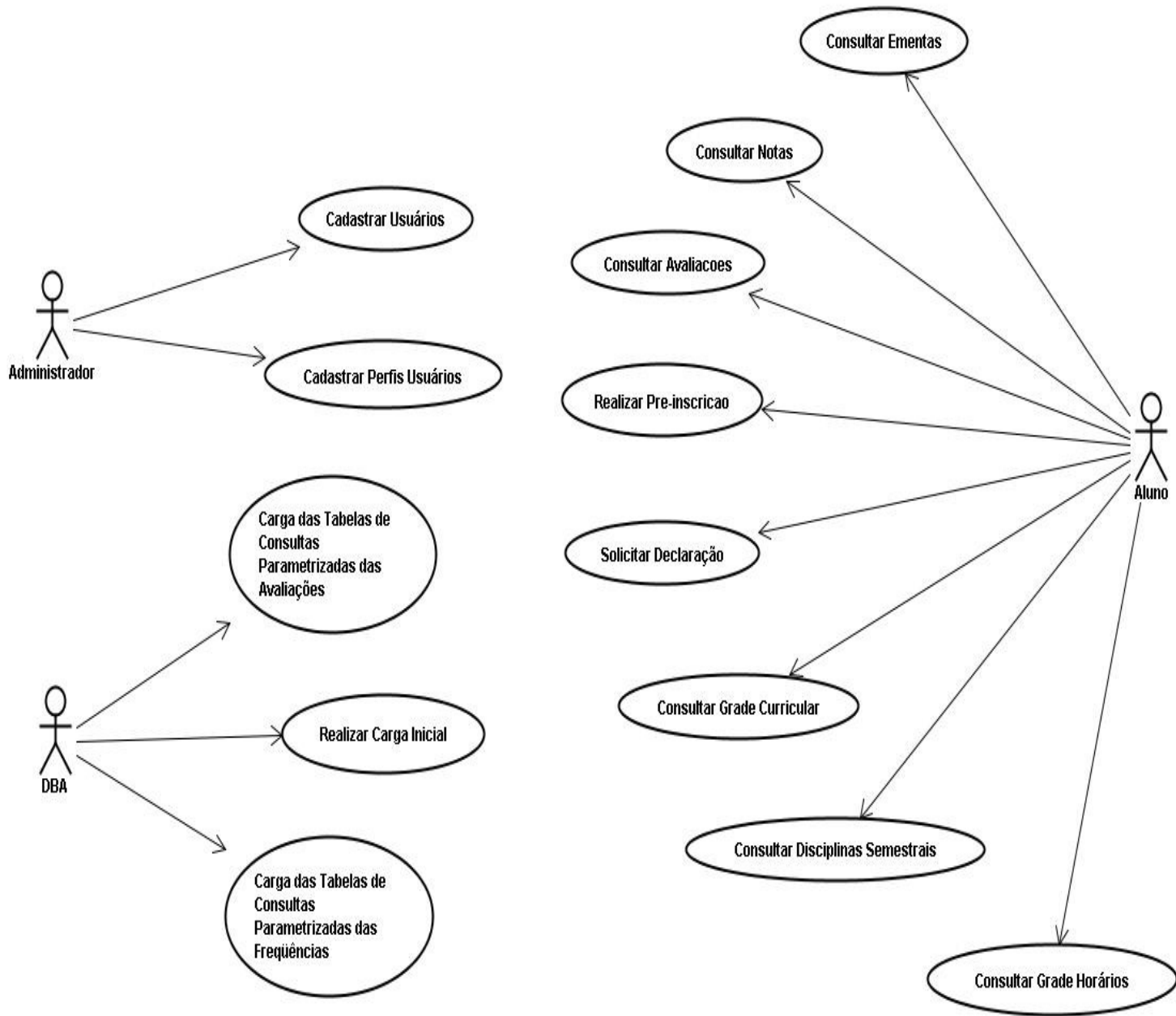


Figura 6 – Diagrama de Casos de Uso – Administrador, Aluno e DBA

2.3.3. Atores – Professor e Coordenador



Figura 7 – Diagrama de Casos de Uso – Professor e Coordenador

2.4. Descrição dos Casos de Uso

2.4.1. Evento 1

Nome: DBA realiza carga inicial das tabelas no sistema

Ator: DBA

Ação: Realizar a carga inicial das tabelas no sistema

Descrição: Este evento é responsável por alimentar o sistema com dados oriundos de outros sistemas que uma instituição de ensino pode possuir contendo informações cadastrais de alunos, professores, funcionários, etc.

Curso Normal:

- 1.O DBA seleciona a opção de importação de dados.
- 2.O DBA seleciona a localização do arquivo a ser importado.
- 3.Sistema verifica a estrutura dos dados que serão importados pelo sistema.
- 4.O sistema solicita confirmação da operação.
- 5.O DBA confirma a operação.
- 6.Sistema insere os dados nas respectivas tabelas.
- 7.Sistema finaliza operação emitindo mensagem “Operação Efetuada com Sucesso”.

Curso Alternativo:

- 1.Após o passo 1, caso o DBA cancele a operação, o sistema emitirá a seguinte mensagem: “Importação de dados cancelada” e encerrará a operação.
- 2.Após o passo 2, no caso de a estrutura do arquivo importado não se enquadrar nas necessidades do sistema, será emitida uma mensagem de: “Impossível importar arquivo de dados” e a operação será encerrada.
- 3.Após o passo 3, se os dados que serão importados forem inválidos, o sistema emite a seguinte mensagem: “Dados Inválidos!” e cancela a operação.
- 4.Após o passo 4, o DBA cancela a operação, o sistema emite a seguinte mensagem: “Operação Cancelada pelo Usuário!” e cancela a operação.

2.4.2. Evento 2

Nome: DBA realiza carga das tabelas das consultas parametrizadas das avaliações

Ator: DBA

Ação: Realizar a carga inicial das tabelas das consultas parametrizadas das avaliações.

Descrição: Este evento é responsável por alimentar as tabelas que serão utilizadas como fonte de informações das diversas consultas parametrizadas das avaliações dos alunos.

Curso Normal:

- 1.O DBA seleciona a opção de carga de dados.
- 2.O usuário seleciona a opção de avaliações.
- 3.Usuário confirma a carga dos dados.
- 4.Sistema realiza a carga das tabelas de dimensão e da tabela fato das avaliações.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, o sistema emitirá a mensagem: “Operação cancelada!” e encerrará a operação.
- 2.Após o passo 3, no caso de os dados informados pelo usuário não forem confirmados, o sistema informará que: “Os dados informados não foram confirmados. Operação cancelada!” e a operação será cancelada.

2.4.3. Evento 3

Nome: DBA realiza carga das tabelas de consultas parametrizadas das frequências

Ator: DBA

Ação: Realizar a carga das tabelas das consultas parametrizadas das frequências.

Descrição: Este evento é responsável por alimentar as tabelas que serão utilizadas como fonte de informações das diversas consultas parametrizadas das frequências dos alunos.

Curso Normal:

- 1.O DBA seleciona a opção de carga de dados.
- 2.O usuário seleciona a opção de frequência.
- 3.Usuario confirma a carga dos dados.
- 4.Sistema realiza a carga das tabelas de dimensão e da tabela fato das frequências dos alunos.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, o sistema emitirá a mensagem: “Operação cancelada!” e encerrará a operação.
- 2.Após o passo 3, no caso de os dados informados pelo usuário não forem confirmados, o sistema informará que: “Os dados informados não foram confirmados. Operação cancelada!” e a operação será cancelada.

2.4.4. Evento 4

Nome: Administrador cadastra usuários do sistema.

Ator: Administrador

Ação: Cadastrar usuários do portal

Descrição: Este evento tem por objetivo realizar o cadastramento dos diversos usuários do portal acadêmico.

Pré-condição: Um aluno, ou professor, ou funcionário e um perfil para associar ao usuário devem estar previamente cadastrados no sistema.

Curso Normal:

- 1.O administrador seleciona a opção de usuários.
- 2.O usuário seleciona a opção de cadastrar usuário.
- 3.O usuário seleciona o perfil a ser associado ao usuário
- 4.Usuário preenche as informações do cadastramento.
- 5.O sistema realiza a validação dos dados.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.O usuário finaliza a operação e o sistema confirma o cadastramento emitindo a mensagem: “Usuário cadastrado com sucesso.

Curso Alternativo:

- 1.Após o passo 1, 2, 3 ou 4, caso o usuário cancele a operação, o sistema exibirá a seguinte mensagem: “Operação cancelada pelo usuário!” e encerrará a operação.
- 2.Após o passo 5, caso no sistema já exista um registro do usuário, será emitida uma mensagem: “O usuário informado já existe no sistema” e os campos informados serão limpos.
- 3.Após o passo 5, caso o sistema não valide os dados informados pelo usuário, será exibida a mensagem: “Os dados informados não foram validados” e a operação será cancelada.
- 4.Após o passo 6, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.5. Evento 5

Nome: Administrador cadastra os perfis para usuários

Ator: Administrador

Ação: Cadastrar perfis de usuários

Descrição: O evento em questão tem o objetivo realizar o cadastramento dos diversos perfis de acesso para os usuários do portal acadêmico.

Curso Normal:

- 1.Usuário seleciona a opção de cadastramento de perfis.
- 2.O usuário informa o nome do perfil e seleciona as permissões relativas ao mesmo.
- 3.Sistema realiza validação dos dados.
- 4.O sistema exibe uma tela contendo os dados informados.
- 5.Sistema finaliza operação emitindo mensagem: “Novo perfil adicionado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, o sistema emitirá uma mensagem: “Operação de cadastramento cancelada”.
- 2.Após o passo 3, no caso de o perfil de usuário já estar cadastrado, será emitida uma mensagem: “Perfil de usuário já existente no sistema” e os campos informados serão limpos.
- 3.Após o passo 4, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.6. Evento 6

Nome: Secretaria Cadastra Instituição de Ensino

Ator: Secretaria acadêmica

Ação: Cadastrar Instituição de ensino

Descrição: Este evento tem o objetivo de informar os dados cadastrais de uma instituição de ensino no portal acadêmico

Curso Normal:

- 1.Usuário seleciona a opção de cadastramento de Instituição de Ensino.
- 2.O usuário preenche os dados da Instituição.
- 3.O sistema verifica se os dados são válidos.
- 4.O sistema exibe uma tela contendo os dados informados.
- 5.O usuário finaliza a operação e o sistema exibe a mensagem: “Instituição Cadastrada com Sucesso”.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, será emitida pelo sistema uma mensagem: “Operação de cadastramento cancelada” e encerrará a operação.
- 2.Após o passo 3, se no banco de dados já existir um registro com as informações preenchidas pelo usuário, será emitida uma mensagem “A instituição já se encontrada cadastrada no sistema” e os campos informados serão limpos.
- 3.Após o passo 3, caso os dados informados pelo usuário não sejam validados pelo sistema, será exibida uma mensagem: “Os dados informados estão incorretos” e os campos preenchidos serão limpos.
- 4.Após o passo 4, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.7. Evento 7

Nome: Secretaria Cadastra Curso

Ator: Secretaria acadêmica

Ação: Cadastrar os cursos os cursos da instituição

Descrição: O evento em questão tem por objetivo realizar o cadastramento dos diversos cursos da instituição de ensino.

Pré-condição: Ter uma instituição de ensino previamente cadastrada no sistema.

Curso Normal:

- 1.Usuário seleciona a opção cadastrar curso.
- 2.O usuário informa os dados do curso.
- 3.O sistema realiza a validação dos dados.
- 4.O sistema exibe uma tela contendo os dados informados.
- 5.O usuário finaliza a operação e o sistema emite a mensagem: “Curso Cadastrado com Sucesso”

Curso Alternativo:

- 1.Após o passo 1 ou 2, se o usuário cancelar a operação, o sistema emitirá uma mensagem: “Operação de cadastramento cancelada” e encerrará a operação.
- 2.Após o passo 3, se os dados preenchidos pelo usuário não forem validados pelo sistema, será exibida a mensagem: “Os dados informados estão incorretos” e os campos informados pelo usuário serão limpos.
- 3.Apos o passo 3, se o curso cadastrado já existir no sistema, uma mensagem de “O curso informado já está cadastrado” será emitida e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 4, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.8. Evento 8

Nome: Secretaria cadastra grades curriculares

Ator: Secretaria acadêmica

Ação: Cadastrar grades curriculares

Descrição: Neste evento o usuário informa ao sistema os dados inerentes a uma grade curricular de um curso.

Pré-Condição: Ter um curso previamente cadastrado no sistema para que a grade possa ser associada.

Curso Normal:

- 1.Usuário seleciona a opção cadastrar grade de disciplinas.
- 2.O usuário seleciona o curso ao qual deseja associar à grade.
- 3.Os dados inerentes à grade são informados.
- 4.O sistema faz a validação das informações.
- 5.O sistema exibe uma tela contendo os dados informados.
6. O usuário finaliza a operação e o sistema exibe a mensagem: “Grade curricular cadastrada com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, o sistema emitirá uma mensagem: “Operação de cadastramento cancelada” e encerrará a operação.
- 2.Após o passo 4, se no sistema já houver sido cadastrada uma grade com os dados informados, o sistema irá emitir uma mensagem: “A grade informada já consta no sistema” e a operação de cadastramento será encerrada.
- 3.Após o passo 4, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.9. Evento 9

Nome: Secretaria acadêmica cadastra disciplina

Ator: Secretaria acadêmica

Ação: Cadastrar Disciplinas

Descrição: Este evento tem por objetivo alimentar o sistema com as informações cadastrais relacionadas às diversas disciplinas dos cursos da instituição.

Pré-Condição: Ter um curso e uma grade previamente cadastrados no sistema.

Curso Normal:

- 1.O usuário seleciona a opção de cadastramento de disciplina.
- 2.O usuário associa um curso e uma grade ao cadastramento da disciplina.
- 3.O usuário preenche as informações cadastrais da disciplina.
- 4.Sistema faz a validação dos dados informados pelo usuário.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema emite a mensagem: “Operação de cadastramento de disciplina efetuada com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, o sistema emitirá uma mensagem: “Operação de cadastramento cancelada” e encerrará a operação.
- 2.Após o passo 4, se no sistema já existir a referida disciplina cadastrada, o mesmo irá exibir uma mensagem “A disciplina informada já consta no sistema” e limpará os dados informados.
- 3.Após o passo 4, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.10. Evento 10

Nome: Secretaria acadêmica cadastra ementas disciplinares

Ator: Secretaria acadêmica

Ação: Cadastrar ementas disciplinares

Descrição: Este evento tem o objetivo de cadastrar as ementas referentes às várias disciplinas.

Pré-Condição: Ter um curso, uma grade e uma disciplina previamente cadastrada no sistema.

Curso Normal:

- 1.O usuário seleciona a opção cadastrar ementas.
- 2.Usuário seleciona o curso, a grade e a disciplina aos quais a ementa será associada.
- 3.O usuário informa os dados cadastrais da ementa.
- 4.Sistema faz a validação das informações.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema emite a mensagem: “Ementa cadastrada. Aguardando aprovação da coordenação”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, o sistema emitirá então uma mensagem: “Operação de cadastramento cancelada” e a operação será encerrada.
- 2.Após o passo 4, se no sistema já existir a ementa informada, será emitida uma mensagem “Ementa já existente no sistema.” e os campos informados pelo usuário serão limpos.
- 3.Após o passo 4, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.11. Evento 11

Nome: Secretaria acadêmica cadastra conteúdo programático da ementa disciplinar

Ator: Secretaria acadêmica

Ação: Cadastrar conteúdo programático

Descrição: O evento em questão é responsável pelo cadastramento de conteúdos programáticos das disciplinas no portal acadêmico.

Pré-Condição: Ter uma ementa, uma disciplina, uma grade e um curso previamente cadastrados.

Curso Normal:

- 1.Usuário seleciona a opção de cadastrar conteúdo programático.
- 2.O usuário seleciona o curso, a grade, a disciplina e a ementa aos quais o conteúdo programático será associado.
- 3.Usuário preenche os dados cadastrais do conteúdo programático.
- 4.Sistema realiza a verificação dos dados informados.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema confirma a operação de cadastramento de conteúdo programático emitindo a mensagem: “Cadastramento de conteúdo realizado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, se o usuário cancelar a operação, o sistema irá emitir uma mensagem: “Operação de cadastramento cancelada” e encerrará a operação.
- 2.Após o passo 4, caso já exista cadastrado no sistema um conteúdo igual, o sistema exibirá uma mensagem: “O conteúdo programático informado já existe no sistema”.
- 3.Após o passo 4, se os dados informados pelo usuário não forem validados pelo sistema, uma mensagem de: “Os dados informados estão incorretos” e os campos informados pelo usuário serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.12. Evento 12

Nome: Secretaria acadêmica cadastra turma

Ator: Secretaria acadêmica

Ação: Cadastrar Turmas

Descrição: Este evento tem por objetivo o cadastramento das várias turmas dos variados cursos de uma instituição

Pré-Condição: Devem ser cadastrados previamente no sistema, uma disciplina e um curso.

Curso Normal:

- 1.Usuário seleciona a opção de cadastramento de turma.
- 2.O usuário associa a turma à disciplina e a um curso.
- 3.Usuário informa os dados da turma (local, sala, dia e horário).
- 4.Sistema realiza validação das informações.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema confirma o cadastramento exibindo uma mensagem: “Operação de cadastramento de turma realizada com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, o sistema irá emitir uma mensagem: “Operação de cadastramento cancelada” e a operação será encerrada.
- 2.Após o passo 4, caso no banco de dados já exista um registro igual ao informado, o sistema exibirá uma mensagem “A turma já existe no sistema” e os campos preenchidos pelo usuário serão limpos.
- 3.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.13. Evento 13

Nome: Secretaria acadêmica cadastra alunos

Ator: Secretaria acadêmica

Ação: Cadastrar Alunos

Descrição: Esse evento tem a responsabilidade de realizar o cadastramento de alunos

Pré-Condição: Devem ser cadastrados previamente no sistema, uma disciplina, um curso e uma turma.

Curso Normal:

- 1.Usuário seleciona a opção de cadastramento de alunos.
- 2.Usuário preenche os dados do aluno.
- 3.O usuário seleciona a disciplina, a turma e o curso aos quais irá associar o aluno.
- 4.Sistema faz a validação dos dados.
- 5.O usuário associa o aluno a uma turma.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.O usuário finaliza a operação e o sistema confirma o cadastramento exibindo uma mensagem: “O cadastro do aluno foi realizado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, uma mensagem de: “Cadastramento do aluno cancelado” será emitida e a operação será cancelada.
- 2.Após o passo 4, caso já exista o registro do aluno no banco de dados, o sistema irá emitir uma mensagem: “O aluno já está cadastrado no sistema” e os campos preenchidos pelo usuário serão limpos.
- 3.Após o passo 4, se os dados digitados pelo usuário não forem validados, o sistema emitirá uma mensagem: “Os dados informados estão incorretos” e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 6, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.14. Evento 14

Nome: Secretaria acadêmica cadastra professores

Ator: Secretaria acadêmica

Ação: Cadastrar Professores

Descrição: O evento em questão tem o objetivo de realizar o cadastramento de professores.

Curso Normal:

- 1.Usuário seleciona a opção de cadastramento de professores.
- 2.O usuário associa o professor a uma disciplina.
- 3.Usuário preenche as informações do professor.
- 4.Sistema faz a validação dos dados.
- 5.O sistema exibe uma tela com os dados informados.
- 6.O usuário finaliza a operação e o sistema confirma o cadastramento exibindo uma mensagem: “O cadastro do professor foi realizado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, uma mensagem de: “Cadastramento do professor cancelado” será emitida e a operação será cancelada.
- 2.Após o passo 4, caso já exista o registro do professor no banco de dados, o Sistema irá emitir uma mensagem: “O professor já possui um registro no sistema” e os campos preenchidos pelo usuário serão limpos.
- 3.Após o passo 4, se os dados digitados pelo usuário não forem validados, o sistema emitirá uma mensagem: “Os dados informados estão incorretos” e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.15. Evento 15

Nome: Secretaria acadêmica ou professor cadastra vaga de estágio

Ator: Secretaria acadêmica / Professor

Ação: Cadastrar Vaga de Estágio

Descrição: Este evento tem o objetivo de fazer o cadastramento de vagas de estágio pelos professores ou funcionários da instituição de ensino.

Curso Normal:

- 1.Usuário seleciona a opção de cadastramento de vagas de estágio.
- 2.O usuário associa a vaga a uma empresa.
- 3.O usuário associa a vaga a um curso da instituição.
- 4.Usuário preenche os dados da vaga de estágio.
- 5.Sistema faz a validação das informações.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.O usuário finaliza a operação e o sistema confirma o cadastramento da vaga emitindo a mensagem: “A vaga de estágio foi cadastrada com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2, 3 ou 4, caso o usuário cancele a operação, uma mensagem de: “Agendamento de avaliação cancelado” é emitida e a operação de será encerrada.
- 2.Após o passo 2, caso a empresa a ser associada não esteja cadastrada no sistema, o usuário pode cadastrá-la no momento do cadastramento da vaga e, para tal, ele é encaminhado para o Evento 16.
- 3.Após o passo 5, caso o sistema não valide os dados informados pelo usuário, uma mensagem informando-o que: “Os dados informados estão incorretos” será emitida e os campos preenchidos serão limpos.
- 4.Após o passo 5, caso no sistema já exista um registro da vaga em questão, a mensagem: “Esta vaga já foi cadastrada no sistema” e a operação será cancelada.
- 5.Após o passo 6, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.16. Evento 16

Nome: Secretaria acadêmica / Professor cadastra empresa

Ator: Secretaria acadêmica e Professor

Ação: Cadastrar Empresa

Descrição: Este evento tem por objetivo possibilitar o cadastramento de empresas para que vagas de estágio possam ser associadas a ela.

Curso Normal:

- 1.Usuário preenche os dados da empresa
- 2.Sistema realiza validação dos dados.
- 3.O sistema exibe uma tela contendo os dados informados.
- 4.O usuário finaliza a operação e o sistema emite a mensagem: “A empresa foi cadastrada com sucesso”.
- 5.O usuário é encaminhado ao evento 15 para finalizar o cadastramento da vaga de estágio.

Curso Alternativo:

- 1.Após o passo 1, caso o usuário cancele a operação, uma mensagem de: “Cadastramento de empresa cancelado” será emitida e as operações de cadastramento da empresa e da vaga de estágio serão encerradas.
- 2.Após o passo 2, caso o sistema não valide as informações passadas pelo usuário, será emitida uma mensagem informando-o que: “Os dados informados estão incorretos” e os campos informados pelo usuário serão limpos.
- 3.Após o passo 2, no caso de já existir no sistema um registro da empresa informada, a mensagem: “A empresa informada já consta no sistema” será emitida e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 3, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.17. Evento 17

Nome: Secretaria cadastra recursos da instituição

Ator: Secretaria acadêmica

Ação: Cadastrar Recurso

Descrição: Este evento tem por objetivo realizar o cadastramento dos recursos disponíveis na instituição de ensino.

Curso Normal:

- 1.Usuário seleciona opção de cadastramento de recursos.
- 2.O usuário informa o tipo do recurso.
- 3.Usuário preenche os dados inerentes ao recurso.
- 4.Sistema realiza validação dos dados.
- 5.O sistema exibe uma tela contendo os dados informados..
- 6.O usuário finaliza a operação e o sistema confirma o cadastramento exibindo a mensagem: “O recurso foi cadastrado com sucesso no sistema”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, uma mensagem de: “Cadastramento de recurso cancelado” será emitida e a operação de será encerrada.
- 2.Após o passo 4, caso o sistema não valide as informações passadas pelo usuário, será emitida uma mensagem informando-o que: “Os dados informados estão incorretos” e os campos informados pelo usuário serão limpos.
- 3.Após o passo 4, se no sistema já constar um registro com as mesmas características do informado pelo usuário, uma mensagem de: “Este recurso já se encontra cadastrado no sistema” será emitida e os dados informados pelo usuário serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.18. Evento 18

Nome: Secretaria acadêmica aloca recursos para os professores

Ator: Secretaria acadêmica

Ação: Alocar Recursos

Descrição: Este evento tem por objetivo possibilitar a secretaria acadêmica alocar recursos das instituições de ensino para os professores.

Pré-condição: Os professores e recursos da instituição devem ser previamente cadastrados no sistema.

Curso Normal:

- 1.Usuário seleciona opção de alocação de recursos.
- 2.O usuário seleciona um recurso e o associa a um professor.
- 3.O usuário seleciona uma data para alocar o recurso.
- 4.Usuário preenche as informações da alocação do recurso.
- 5.Sistema realiza validação dos dados.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.O usuário finaliza a operação e o sistema emite mensagem: “Recurso alocado com sucesso para o professor”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, uma mensagem de: “Alocação de recurso cancelada” será emitida e a operação de será encerrada.
- 2.Após o passo 4, se os dados informados pelo usuário não forem validados pelo sistema, a mensagem: “Os dados informados estão incorretos” será emitida e os campos preenchidos serão limpos.
- 3.Após o passo 4, caso no sistema já exista um registro de uma alocação de um determinado recurso para um professor na data selecionada pelo usuário, o sistema emitirá uma mensagem informando que: “O recurso solicitado já se encontra alocado” e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 4, se as informações preenchidas pelo usuário não forem validadas, o sistema enviará uma mensagem: “Os dados informados estão incorretos” e os campos preenchidos pelo usuário serão limpos.

5. Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.19. Evento 19

Nome: Secretaria cadastra local da instituição.

Ator: Secretaria acadêmica

Ação: Cadastrar Local

Descrição: Este evento tem o objetivo de a secretaria acadêmica realizar o cadastramento dos locais da instituição de ensino (blocos, prédios, campus, etc)..

Curso Normal:

1. Usuário seleciona opção de cadastramento de locais
2. O usuário informa os dados do local.
3. Sistema realiza validação dos dados.
4. O sistema exibe uma tela contendo os dados informados.
5. O usuário finaliza a operação e o sistema exibe a mensagem: “O local foi cadastrado com sucesso”.

Curso Alternativo:

1. Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de: “Operação de cadastramento cancelada” será emitida e a operação de será encerrada.
2. Após o passo 3, caso no sistema já exista um registro do local informado, o sistema emitirá uma mensagem informando que: “Já existe um local com as mesmas informações no sistema” e os campos preenchidos pelo usuário serão limpos.
3. Após o passo 3, se as informações preenchidas pelo usuário não forem validadas, o sistema enviará uma mensagem: “Os dados informados estão incorretos” e os campos preenchidos pelo usuário serão limpos.
4. Após o passo 4, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.20. Evento 20

Nome: Secretaria cadastra sala de aula de um local da instituição.

Ator: Secretaria acadêmica

Ação: Cadastrar Sala

Descrição: Este evento tem o objetivo de a secretaria acadêmica realizar o cadastramento das salas de aula dos locais da instituição de ensino.

Pré-Condição: Para que uma sala de aula possa ser cadastrada, no sistema deve previamente existir um registro de um local.

Curso Normal:

- 1.Usuário seleciona opção de cadastramento de salas de aula
- 2.O usuário associa a sala de aula a um local
- 3.O usuário informa os dados da sala de aula.
- 4.Sistema realiza validação dos dados.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema confirma o cadastramento emitindo a mensagem: “A sala de aula foi cadastrada com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, uma mensagem de: “Operação de cadastramento cancelada” será emitida e a operação de será encerrada.
- 2.Após o passo 4, caso no sistema já exista um registro da sala de aula informada, o sistema emitirá uma mensagem informando que: “Já existe um registro desta sala de aula no sistema” e os campos preenchidos pelo usuário serão limpos.
- 3.Após o passo 4, se as informações preenchidas pelo usuário não forem validadas, o sistema enviará uma mensagem: “Os dados informados estão incorretos” e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.21. Evento 21

Nome: Professor agenda avaliação

Ator: Professor

Ação: Agendar avaliação

Descrição: Este evento tem o objetivo de professores realizarem o agendamento de avaliações institucionais.

Pré-condição: Uma disciplina e uma turma devem estar previamente cadastradas e um professor deve estar associado a ela para que a avaliação possa ser agendada.

Curso Normal:

- 1.Usuário seleciona opção de agendamento de avaliação
- 2.O usuário associa a avaliação à uma turma e à uma disciplina.
- 3.O usuário seleciona a data da avaliação.
- 4.Usuário preenche os dados inerentes à avaliação.
- 5.Sistema realiza validação dos dados.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.O usuário finaliza a operação e o sistema exibe a mensagem: “Avaliação agendada com sucesso para a data selecionada”.

Curso Alternativo:

- 1.Após o passo 1, 2, 3 ou 4, caso o usuário cancele a operação, uma mensagem de: “Agendamento de avaliação cancelado” será emitida e a operação de será encerrada.
- 2.Após o passo 5, caso no sistema já exista um registro de uma avaliação da disciplina na data selecionada pelo professor, o sistema emitirá uma mensagem informando que: “Já existe uma avaliação agendada para esta data” e os campos preenchidos pelo usuário serão limpos.
- 3.Após o passo 5, se as informações preenchidas pelo usuário não forem válidas, o sistema enviará uma mensagem: “Os dados informados estão incorretos” e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 6, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.22. Evento 22

Nome: Professor disponibiliza material para os alunos

Ator: Professor

Ação: Disponibilizar Material

Descrição: Este evento tem por objetivo realizar a disponibilização de materiais pelos professores para os alunos com o intuito de auxiliá-los em seus estudos.

Pré-condição: Deverão ser previamente cadastrados no sistema uma disciplina e uma turma.

Curso Normal:

- 1.Usuário seleciona a opção de disponibilização de materiais.
- 2.O usuário associa o material à uma disciplina e à uma turma.
- 3.Usuário preenche os dados para a disponibilização do material.
- 4.Sistema faz a validação dos dados.
- 5.O usuário seleciona o arquivo a ser disponibilizado.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.Sistema faz o *upload* do arquivo.
- 8.Sistema finaliza a operação emitindo a mensagem: “O upload do arquivo foi realizado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2, 3 ou 4, caso o usuário desista da disponibilização do arquivo e cancele a operação, o sistema emitirá uma mensagem: “Operação cancelada” e cancelará a operação.
- 2.Após o passo 3, caso já exista cadastrado um arquivo com as mesmas características, será emitida uma mensagem: “Não foi possível prosseguir, pois o arquivo informado já existe no sistema” e a os campos preenchidos pelo usuário serão limpos.
- 3.Após o passo 3, caso os dados informados pelo usuário não forem validados pelo sistema, a mensagem: “Os dados informados estão incorretos” será exibida e os campos preenchidos pelo usuário serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.23. Evento 23

Nome: Professor lança frequência dos alunos

Ator: Professor

Ação: Lançar Frequência

Descrição: O evento em questão tem o objetivo de fazer com que os professores possam lançar no portal acadêmico a frequência dos seus alunos em uma determinada disciplina.

Pré-condição: Deverão ser previamente cadastrados no sistema uma disciplina, uma turma e um aluno.

Curso Normal:

- 1.Usuário seleciona opção de lançamento de frequência.
- 2.O usuário seleciona a disciplina e a turma.
- 3.Enquanto houver alunos, o usuário preenche os dados do lançamento..
- 4.Sistema realiza a validação das informações.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema exibe a mensagem: “O lançamento da presença dos alunos foi realizado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário desista de realizar o lançamento no sistema, uma mensagem de: “Lançamento de frequência cancelado” será exibida e a operação será encerrada.
- 2.Após o passo 4, se no sistema já existir registro de lançamento de frequência para a data selecionada, o sistema emitirá a mensagem: “Frequência para a data selecionada já lançada no sistema. Operação encerrada” e encerrará a operação.
- 3.Após o passo 4, caso os dados digitados pelo usuário não sejam validados pelo sistema, será emitida uma mensagem informando-o que: “Os dados preenchidos estão incorretos” e os campos informados pelo usuário serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.24. Evento 24

Nome: Professor lança notas das avaliações dos alunos

Ator: Professor

Ação: Lançar Notas Avaliação

Descrição: Este evento tem o objetivo de fazer com que os professores possam lançar no portal acadêmico as notas das avaliações de seus alunos.

Pré-condição: Deverão ser previamente cadastrados no sistema uma disciplina e uma turma.

Curso Normal:

- 1.Usuário seleciona opção de lançamento de notas.
- 2.O usuário seleciona a turma, disciplina e a avaliação.
- 3.Enquanto houver alunos, o usuário preenche os dados do lançamento.
- 4.O usuário seleciona o status da nota.
- 5.Sistema realiza a validação das informações.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.O usuário finaliza a operação e o sistema emite a mensagem: “O lançamento da nota do aluno foi realizado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2, 3 ou 4, caso o usuário cancele a operação, será emitida uma mensagem: “Lançamento de notas cancelado” e a encerrará a operação.
- 2.Após o passo 4, caso o sistema não valide os dados informados pelo usuário, a mensagem: “Os dados informados estão incorretos” será emitida e os campos informados pelo usuário serão limpos.
- 3.Após o passo 4, no caso de já existir um lançamento de nota para a determinada avaliação e para o determinado aluno, o sistema exibirá a mensagem: “Nota já lançada no sistema” e os campos informados pelo usuário serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.25. Evento 25

Nome: Professor lança conteúdo ministrado

Ator: Professor

Ação: Lançar Conteúdo Ministrado

Descrição: Este evento tem por objetivo fazer com que professores possam lançar através do portal acadêmico o conteúdo ministrado por eles após um dia de aula.

Pré-condição: Para o conteúdo ministrador possa ser lançado, o professor deve estar associado a uma turma e a uma disciplina.

Curso Normal:

- 1.Usuário seleciona opção de lançamento de conteúdo.
- 2.O usuário seleciona a disciplina e turma.
- 3.Usuário preenche os dados do lançamento de conteúdo.
- 4.Sistema faz a validação dos dados.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema exibe a mensagem: “Conteúdo lançado com sucesso”

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, será emitida uma mensagem: “Lançamento de conteúdo cancelado” e a encerrará a operação.
- 2.Após o passo 4, caso o sistema não valide as informações preenchidas pelo usuário, será emitida uma mensagem informando-o que: “Os dados informados estão incorretos” e os campos informados serão limpos.
- 3.Após o passo 4, caso no banco de dados já exista um registro deste conteúdo, será emitida uma mensagem informando que: “O conteúdo lançado já está cadastrado no sistema” e os campos informados serão limpos.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.26. Evento 26

Nome: Professor lança comentário do material disponibilizado

Ator: Professor

Ação: Lançar comentário

Descrição: O evento em questão tem o objetivo de possibilitar aos professores que cadastrem comentários sobre os conteúdos lançados ou materiais disponibilizados.

Pré-condição: Para que uma descrição possa ser lançada no sistema, um material deve ser previamente disponibilizado.

Curso Normal:

- 1.Usuário seleciona opção de lançamento de comentário.
- 2.O usuário seleciona a turma e a disciplina para as quais o material foi disponibilizado.
- 3.O usuário associa o comentário a um material ou conteúdo.
- 4.Usuário preenche os dados do comentário a ser lançado.
- 5.Sistema faz a validação dos dados.
- 6.O sistema exibe uma tela contendo os dados informados.
- 7.O usuário finaliza a operação e o sistema exibe a mensagem: “Comentário lançado com sucesso”

Curso Alternativo:

- 1.Após o passo 1, 2, 3 ou 4, caso o usuário cancele a operação, será emitida uma mensagem: “Lançamento de comentário cancelado” e a encerrará a operação.
- 2.Após o passo 5, caso o sistema não valide as informações preenchidas pelo usuário, será emitida uma mensagem informando-o que: “Os dados informados estão incorretos” e os campos informados serão limpos.
- 3.Após o passo 5, caso no sistema já exista um registro deste comentário, será emitida uma mensagem informando que: “O comentário lançado já está cadastrado no sistema” e os campos informados serão limpos.
- 4.Após o passo 6, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.27. Evento 27

Nome: Professor solicita alocação de recursos

Ator: Professor

Ação: Solicitar Recurso

Descrição: Este evento tem por objetivo possibilitar aos professores que realizem a alocação dos recursos existentes na instituição de ensino.

Pré-condição: Para que o professor possa realizar essa solicitação, os recursos da instituição deverão ser previamente cadastrados no sistema.

Curso Normal:

- 1.Usuário seleciona opção de alocação de recursos.
- 2.O usuário seleciona o recurso a ser alocado.
- 3.Usuário seleciona a data da alocação.
- 4.Sistema realiza a validação dos dados.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.O usuário finaliza a operação e o sistema exibe a mensagem: “A solicitação foi concluída com sucesso e encaminhada para a secretaria”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, será emitida uma mensagem: “Solicitação de recurso cancelada” e encerrará a operação.
- 2.Após o passo 4, no caso de o sistema não validar os dados informados pelo usuário, uma mensagem de : “Os dados informados estão incorretos” será exibida e os parâmetros passados serão limpos.
- 3.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.4.28. Evento 28

Nome: Professor / Coordenador realiza consultas parametrizadas das avaliações dos alunos.

Ator: Professor / Coordenador

Ação: Realizar Consulta Parametrizada das Avaliações

Descrição: Este evento tem por objetivo possibilitar aos professores e/ou coordenadores a realização de consultas parametrizadas acerca das avaliações dos alunos..

Curso Normal:

- 1.O usuário seleciona a opção de geração de relatórios.
- 2.O usuário seleciona a opção avaliações.
- 3.Usuário preenche os dados da consulta.
- 4.O sistema exibe uma tela contendo os dados informados..
- 5.Sistema exibe em tela a consulta parametrizada.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, será emitida uma mensagem: “Operação cancelada pelo usuário” e encerrará a operação.

2.4.29. Evento 29

Nome: Professor / Coordenador realiza consultas parametrizadas da frequência dos alunos.

Ator: Professor / Coordenador

Ação: Realizar Consulta Parametrizada da Frequência

Descrição: Este evento tem por objetivo possibilitar aos professores e/ou coordenadores a realização de consultas parametrizadas acerca da frequência dos alunos.

Curso Normal:

- 1.O usuário seleciona a opção de geração de relatórios.
- 2.O usuário seleciona a opção frequência.
- 3.Usuário preenche os dados da consulta.
- 4.O sistema exibe uma tela contendo os dados informados.
- 5.Sistema exibe em tela a consulta parametrizada.

Curso Alternativo:

1. Após o passo 1, 2 ou 3, caso o usuário cancele a operação, será emitida uma mensagem: “Operação cancelada pelo usuário” e encerrará a operação.

2.4.30. Evento 30

Nome: Aluno consulta notas das diversas avaliações

Ator: Aluno

Ação: Consultar Nota Avaliação

Descrição: O evento tem por objetivo possibilitar aos alunos que consultem suas notas das diversas avaliações através do portal acadêmico.

Pré-condição: Para que os alunos possam consultar suas notas, as mesmas devem ser previamente lançadas pelos professores.

Curso Normal:

1. Usuário seleciona opção de consulta de notas.
2. O usuário seleciona a disciplina, a turma e a avaliação.
3. Sistema realiza consulta da nota.
4. Sistema finaliza a operação exibindo em tela a disciplina selecionada e a respectiva nota.

Curso Alternativo:

1. Após o passo 1 ou 2, caso o usuário cancele a operação, será emitida uma mensagem: “Consulta cancelada pelo usuário” e a encerrará a operação.
2. Após o passo 3, caso a nota da disciplina selecionada ainda não tenha sido lançada, o sistema emitirá uma mensagem informando ao usuário que: “Ainda não há notas disponíveis para esta disciplina” e os campos informados pelo usuário serão limpos.

2.4.31. Evento 31

Nome: Aluno realiza pré-inscrição em disciplina

Ator: Aluno

Ação: Realizar Pré-inscrição

Descrição: O evento tem o objetivo de oferecer aos alunos a possibilidade de fazer diversas pré-inscrições nas disciplinas que eles estejam aptos a cursar

Pré-condição: No sistema deverão estar previamente cadastradas uma turma e uma disciplina e caso haja algum pré-requisito para alguma disciplina selecionada, ele deve também estar previamente cadastrado e o requerente deve ter o mesmo atendido.

Curso Normal:

- 1.Usuário seleciona opção de pré-inscrição em disciplina.
- 2.Enquanto houver disciplinas, o usuário poderá selecioná-las.
- 3.Sistema faz a validação dos dados.
- 4.Usuário confirma os dados informados.
- 5.Sistema finaliza a operação e emite uma mensagem informando: “Sua pré-inscrição foi realizada com sucesso. Para concretizar sua inscrição, compareça ao setor de registro acadêmico.”

Curso Alternativo:

- 1.Após o passo 1 ou 2, se o usuário cancelar a operação, uma mensagem de “Pré-inscrição cancelada” será emitida e a operação será encerrada.
- 2.Após o passo 3, caso o aluno não esteja apto quaisquer das disciplinas selecionadas, o sistema informará ao usuário através da mensagem: “Você ainda não pode cursar esta disciplina. Pré-requisitos pendentes” e os campos informados pelo usuário serão limpos.
- 3.Após o passo 3, no caso de já existir um registro de que uma pré-inscrição para uma determinada disciplina tenha sido feito, a mensagem: “Você já fez uma pré-inscrição para essa disciplina” e os campos informados serão limpos.
- 4.Após o passo 4, se o usuário não confirmar as informações preenchidas, o sistema emitirá uma mensagem informando-o que: “Os dados informados não foram confirmados. Operação de lançamento de comentário encerrada” o a operação será cancelada.

2.4.32. Evento 32

Nome: Aluno solicita declaração de curso

Ator: Aluno

Ação: Solicitar Declaração de Curso

Descrição: Este evento tem o objetivo de possibilitar os alunos imprimirem o documento que comprova que estão matriculados em um curso da instituição de ensino.

Pré-condição: O aluno deve estar devidamente matriculado em um curso.

Curso Normal:

- 1.Usuário seleciona opção de impressão de documentos.
- 2.O usuário seleciona a declaração de curso.
- 3.Sistema faz a validação dos dados.
- 4.Usuário confirma os dados informados.
- 5.Sistema finaliza a operação e emite uma mensagem informando: “Impressão realizada com sucesso”

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de “Impressão de documento cancelada” será emitida e a operação será encerrada.
- 2.Após o passo 3, caso o aluno não tenha renovado sua matrícula, o sistema informará ao usuário através da mensagem: “Sua matrícula ainda não foi renovada. Procure o setor de registro acadêmico” e a operação de impressão será encerrada.

2.4.33. Evento 33

Nome: Aluno consulta datas das avaliações cadastradas

Ator: Aluno

Ação: Consultar Datas Avaliações

Descrição: Este evento tem por objetivo fazer com que os alunos possam consultar através do portal acadêmico as datas das avaliações cadastradas pelos professores nas diversas disciplinas.

Pré-condição: Uma avaliação deve ser previamente cadastrada no sistema.

Curso Normal:

- 1.Usuário seleciona opção de consultar datas das avaliações.
- 2.O usuário seleciona a turma e a disciplina.
- 3.Sistema faz a validação dos dados.
- 4.Sistema finaliza a operação e exibe em tela a(s) avaliação(ões) cadastradas de acordo com a parametrização feitas pelo usuário.

Curso Alternativo:

- 1.Após o passo 1 ou 2, se o usuário cancelar a operação, uma mensagem de “Consulta de notas cancelada” será emitida e a operação será encerrada.
- 2.Após o passo 3, caso o sistema não valide as informações digitadas pelo usuário, uma mensagem informando-o que: “Os dados não foram validados” será exibida e os campos preenchidos pelo usuário serão limpos.

2.4.34. Evento 34

Nome: Aluno consulta grade curricular de disciplinas

Ator: Aluno

Ação: Consultar Grade Curricular

Descrição: O evento tem por objetivo fornecer aos alunos a possibilidade de consultar a grade curricular de disciplinas que compõem o curso ao qual estiver inscrito.

Pré-condição: A grade de disciplinas deve estar previamente cadastrada no sistema juntamente com suas respectivas disciplinas.

Curso Normal:

- 1.Usuário seleciona opção de consulta de grade.
- 2.O usuário seleciona a opção de grade curricular do curso.
- 3.Sistema finaliza operação exibindo em tela a grade de disciplinas que compõem o curso ao qual o aluno está associado.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de “Operação cancelada” será emitida e a operação será encerrada

2.4.35. Evento 35

Nome: Aluno consulta grade de disciplinas semestrais

Ator: Aluno

Ação: Consultar Grade de Disciplinas Semestrais

Descrição: O evento tem por objetivo fornecer aos alunos a possibilidade de consultar a grade de disciplinas do curso separadas por semestre.

Pré-condição: A grade de disciplinas deve estar previamente cadastrada no sistema juntamente com suas respectivas disciplinas.

Curso Normal:

- 1.Usuário seleciona opção de consulta de grade.
- 2.O usuário seleciona a opção de grade de disciplinas por semestre.
- 3.Sistema finaliza operação exibindo em tela a grade de disciplinas que compõem o curso ao qual o aluno está associado separadas por semestre.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de “Operação cancelada” será emitida e a operação será encerrada.

2.4.36. Evento 36

Nome: Aluno consulta grade de horários das disciplinas às quais estiver inscrito

Ator: Aluno

Ação: Consultar Grade de Horários

Descrição: O evento tem por objetivo fornecer aos alunos a possibilidade de consultar a grade de horários das disciplinas às quais estiver inscrito.

Pré-condição: Uma turma e a grade de disciplinas devem estar previamente cadastradas no sistema juntamente com suas respectivas disciplinas.

Curso Normal:

- 1.Usuário seleciona opção de consulta de grade.
- 2.O usuário seleciona a opção de grade de horários.
- 3.Sistema finaliza operação exibindo em tela a grade de horários das disciplinas às quais o aluno estiver inscrito.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de “Operação cancelada” será emitida e a operação será encerrada

2.4.37. Evento 37

Nome: Aluno consulta ementa de disciplina

Ator: Aluno

Ação: Consultar Ementas das Disciplinas

Descrição: O evento tem por objetivo possibilitar que os alunos consultem as ementas referentes às disciplinas que eles estejam cursando.

Pré-condição: As ementas disciplinares devem estar previamente cadastradas.

Curso Normal:

- 1.Usuário seleciona opção de consultar ementas das disciplinas.
- 2.Usuário seleciona o curso, a grade e a disciplina a ser consultada.
- 3.Sistema finaliza operação exibindo na tela a ementa da disciplina selecionada.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de “Operação cancelada” será emitida e a operação será encerrada.

2.4.38. Evento 38

Nome: Coordenador consulta conteúdos ministrados

Ator: Coordenador

Ação: Consultar Conteúdos

Descrição: O evento em questão tem por objetivo permitir que os coordenadores possam consultar os conteúdos ministrados cadastrados no sistema pelos professores.

Pré-condição: Para que os coordenadores possam consultar os conteúdos, os mesmos devem ser previamente cadastrados pelos professores.

Curso Normal:

- 1.Usuário seleciona opção de consultar conteúdos ministrados
- 2.O usuário seleciona a data, a turma, a disciplina e o professor.
- 3.Sistema realiza validação dos dados.
- 4.Sistema finaliza a operação exibindo na tela o nome do professor, a disciplina e o conteúdo cadastrado pelo professor.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de “Consulta cancelada” será emitida e a operação será encerrada.
- 2.Após o passo 3, caso não haja nenhum registro no sistema de acordo com os parâmetros selecionados pelo usuário, o sistema emitirá uma mensagem informando-o que: “Não há conteúdos cadastrados para os dados selecionados”.

2.4.39. Evento 39

Nome: Coordenador cadastra pré-requisitos das disciplinas

Ator: Coordenador

Ação: Cadastrar Pré-requisitos

Descrição: Este evento tem por objetivo possibilitar o cadastramento dos pré-requisitos das disciplinas pelos coordenadores dos vários cursos das instituições de ensino.

Curso Normal:

- 1.Usuário seleciona opção de cadastramento de pré-requisitos.
- 2.O usuário seleciona o curso, a grade e a disciplina.
- 3.O usuário associa o pré-requisito à disciplina selecionada.
- 4.Sistema realiza a validação dos dados.
- 5.Sistema finaliza a operação e exibe mensagem: “O pré-requisito foi cadastrado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, uma mensagem de “Operação de cadastramento de pré-requisito cancelada” será emitida e a operação será encerrada.
- 2.Após o passo 4, caso os dados informados pelo usuário não sejam validados pelo sistema, a mensagem: “Os dados informados estão incorretos” e os campos preenchidos serão limpos.
- 3.Após o passo 4, caso no sistema já exista um registro do pré-requisito informado pelo usuário, uma mensagem informando-o que: “Já existe um pré-requisito associado a esta disciplina” será emitida e os campos preenchidos serão limpos.

2.4.40. Evento 40

Nome: Coordenador aprova pré-inscrição em disciplina

Ator: Coordenador

Ação: Aprovar Pré-Inscrição

Descrição: Este evento tem por objetivo possibilitar a aprovação de uma pré-inscrição em uma disciplina.

Pré-condição: Uma pré-inscrição em uma disciplina deve ser previamente cadastrada no sistema bem como os pré-requisitos da referida disciplina devem ser atendidos, caso exista algum.

Curso Normal:

- 1.Usuário seleciona opção de aprovar pré-inscrição.
- 2.Enquanto houver pré-inscrições, o usuário poderá aprová-las.
- 3.Sistema finaliza a operação informando: “O aluno foi inscrito na(s) disciplina(s) selecionada(s)”.

Curso Alternativo:

- 1.Após o passo 1 ou 2, caso o usuário cancele a operação, uma mensagem de “Operação cancelada!” será emitida e a operação será encerrada.

2.4.41. Evento 41

Nome: Coordenador ou Professor ou Secretaria acadêmica cadastra divulgação

Ator: Coordenador / Professor / Secretaria Acadêmico

Ação: Cadastrar Divulgação

Descrição: Este evento tem por objetivo possibilitar o cadastramento de eventos/avisos da instituição de ensino.

Pré-condição: Um curso deve estar previamente cadastrado no sistema.

Curso Normal:

- 1.Usuário seleciona opção de cadastramento de divulgação
- 2.O usuário seleciona o tipo de divulgação e associa um curso à divulgação.
- 3.O usuário preenche os dados da divulgação.
- 4.Sistema realiza a validação dos dados.
- 5.O sistema exibe uma tela contendo os dados informados.
- 6.Sistema finaliza a operação informando que: “O evento/aviso foi cadastrado com sucesso”.

Curso Alternativo:

- 1.Após o passo 1, 2 ou 3, caso o usuário cancele a operação, uma mensagem de “Operação cancelada!” será emitida e a operação será encerrada.
- 2.Após o passo 4, caso no sistema já exista um registro de divulgação igual ao informado, o sistema irá emitir a mensagem: “Este evento/aviso já existe no sistema!” e os campos informados pelo usuário serão limpos.
- 3.Após o passo 4, se os dados informados pelo usuário não forem validados, o sistema emitirá a mensagem: “Os dados informados não foram validados. Operação cancelada” e a operação será encerrada.
- 4.Após o passo 5, caso o usuário tenha preenchido alguma informação de forma equivocada, ele tem a possibilidade de, posteriormente, editar o cadastro realizado, corrigindo as informações.

2.5. Diagramas de Transição de Estados

2.5.1. Classe ARQUIVO

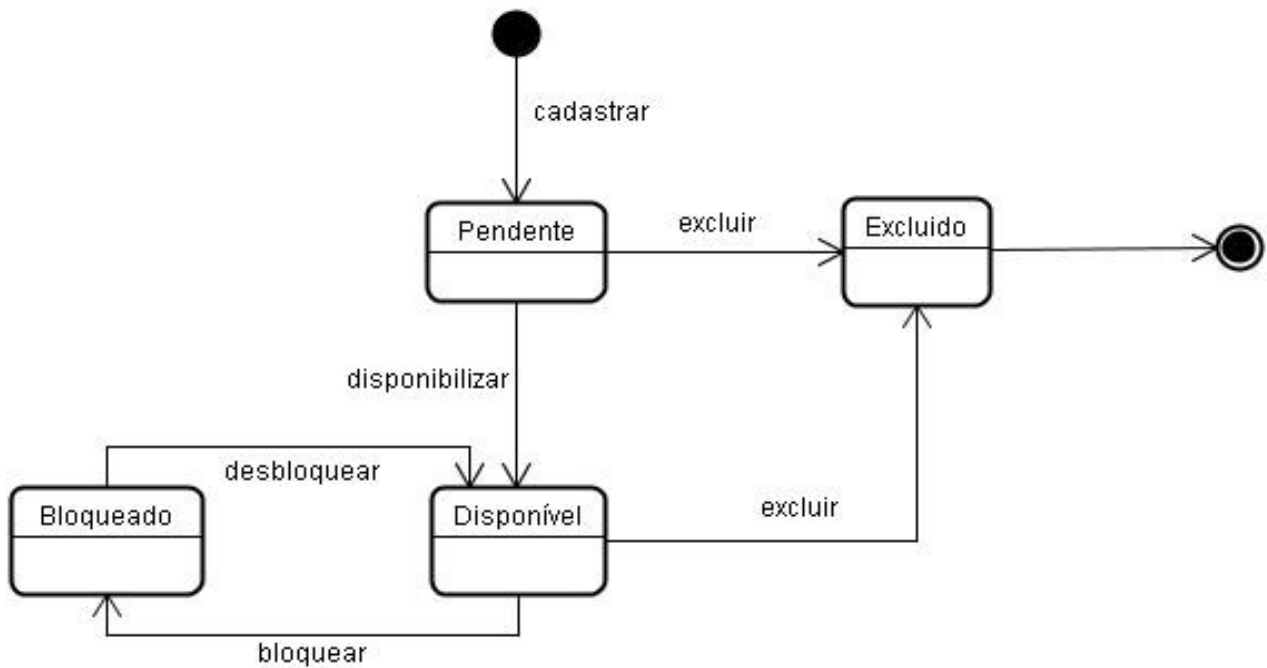


Figura 8 – Diagrama de Transição de Estados – Classe ARQUIVO

2.5.2. Classe EMENTA

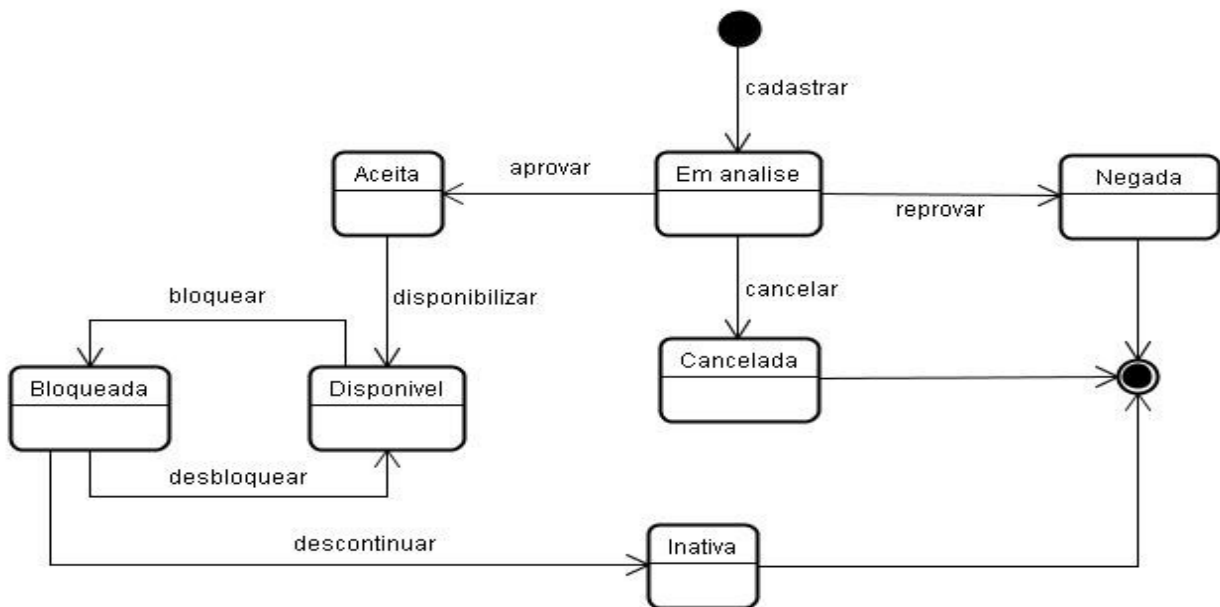


Figura 9 – Diagrama de Transição de Estados – Classe EMENTA

2.5.3. Classe REALIZA

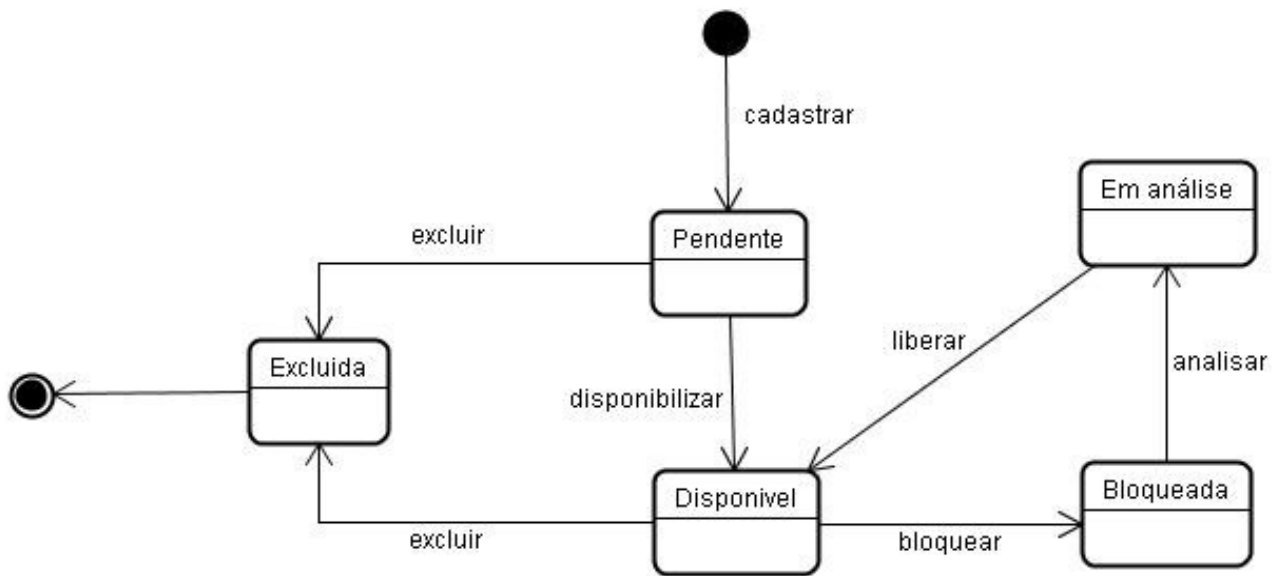


Figura 10 – Diagrama de Transição de Estados – Classe REALIZA

2.5.4. Classes ALUNO

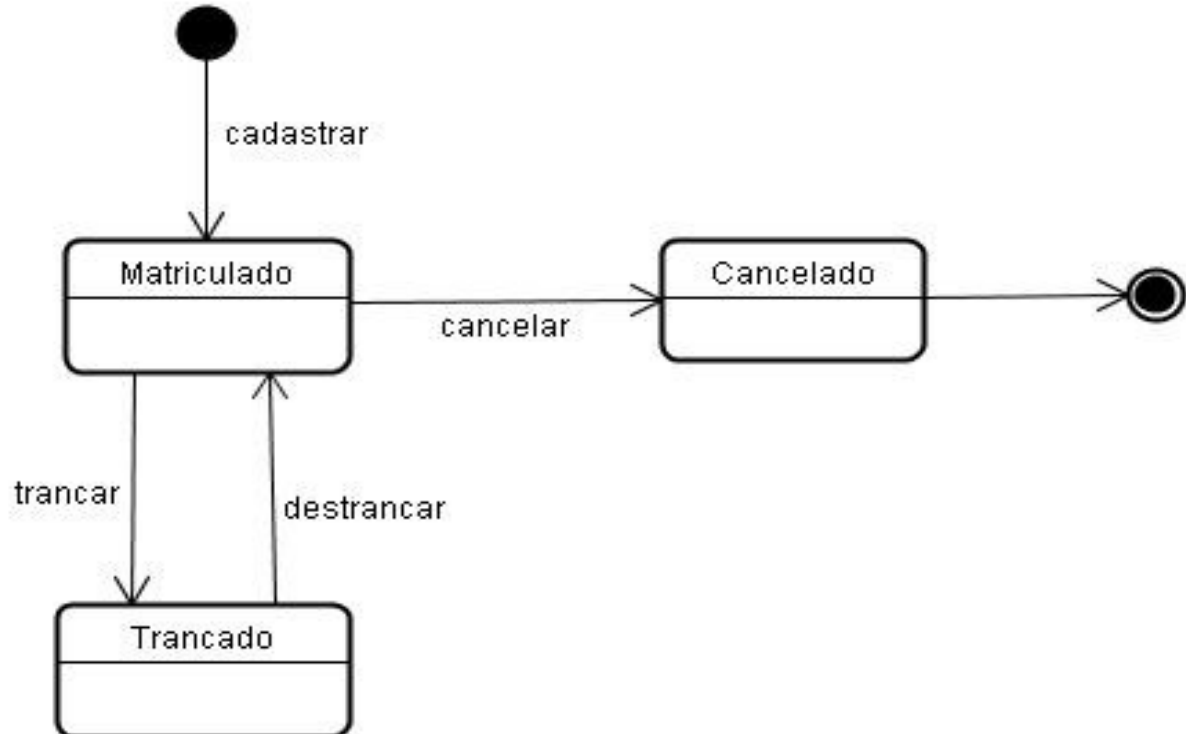


Figura 11 – Diagrama de Transição de Estados – Classe ALUNO

2.5.5. Classes PRÉ-INSCRIÇÃO

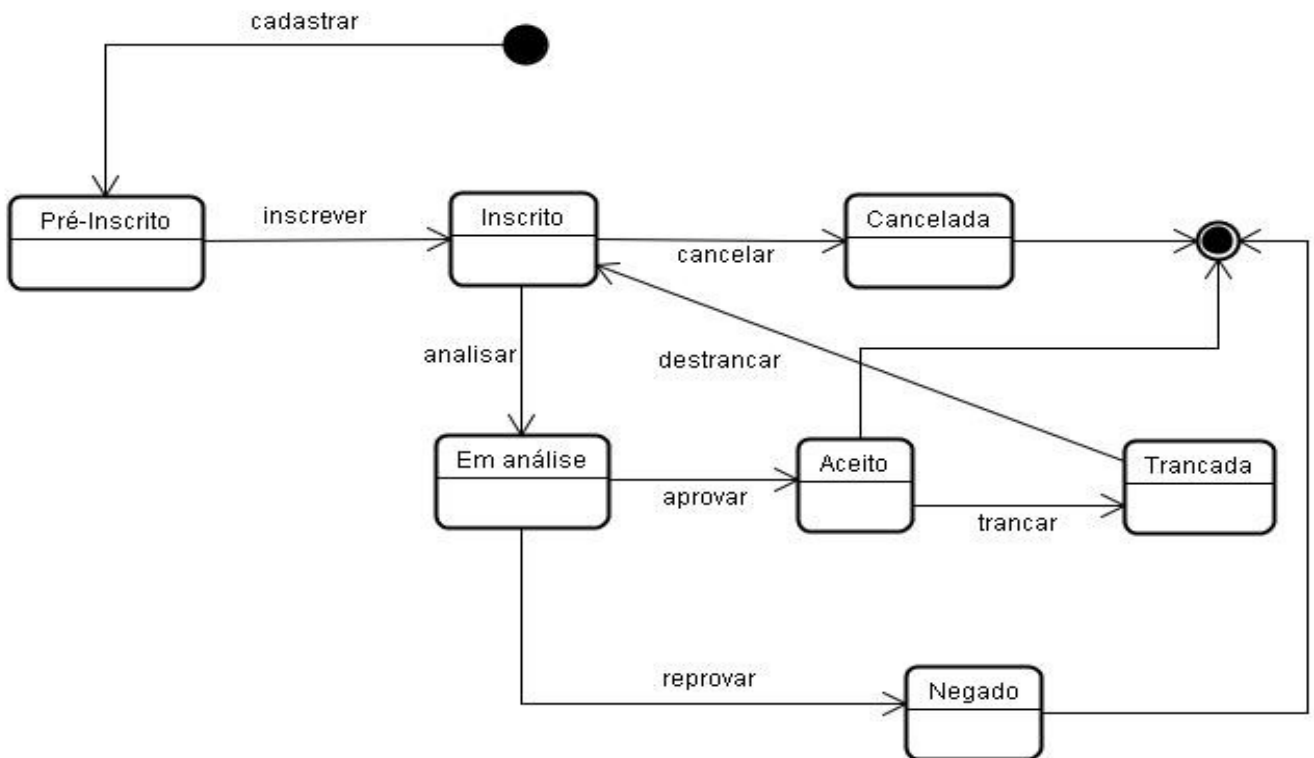


Figura 12 – Diagrama de Transição de Estados – Classe PRÉ-INSCRIÇÃO

2.5.6. Classe ESTÁGIO

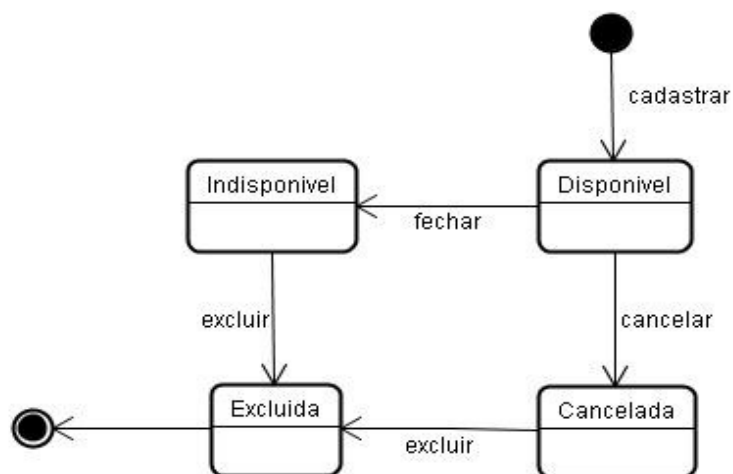


Figura 13 – Diagrama de Transição de Estados – Classe ESTÁGIO

2.5.7. Classe GRADE

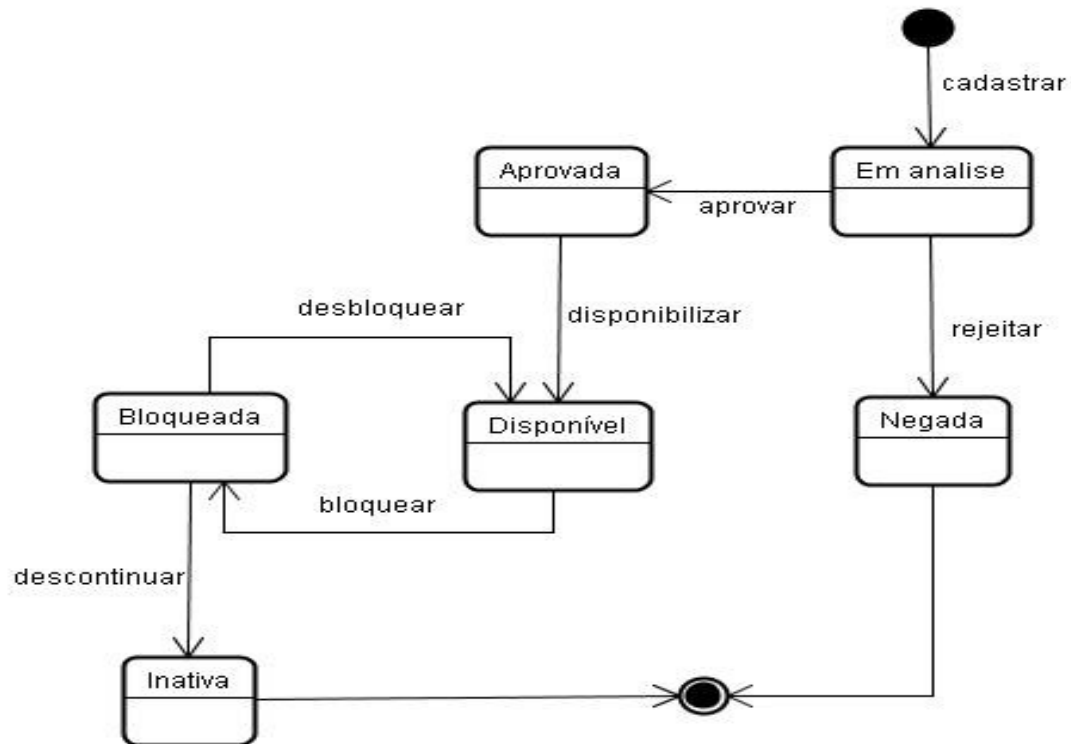


Figura 14 – Diagrama de Transição de Estados – Classe GRADE

2.5.8. Classe DISCIPLINA

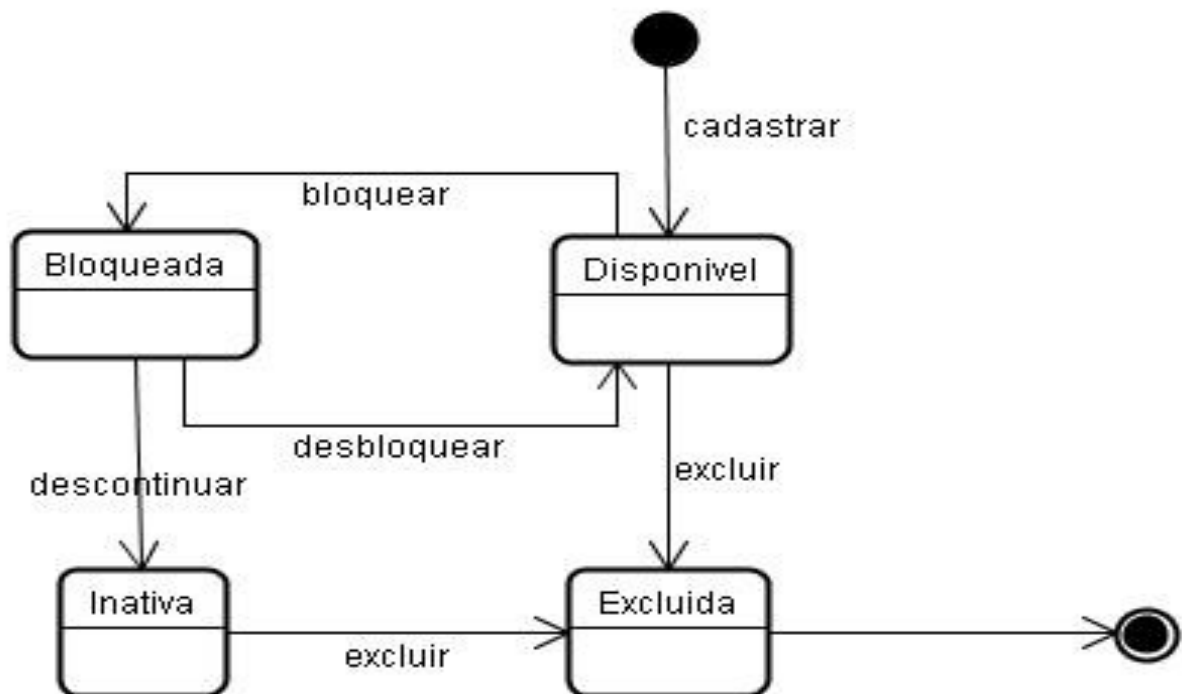


Figura 15 – Diagrama de Transição de Estados – Classe DISCIPLINA

2.5.9. Classe EMPRESA

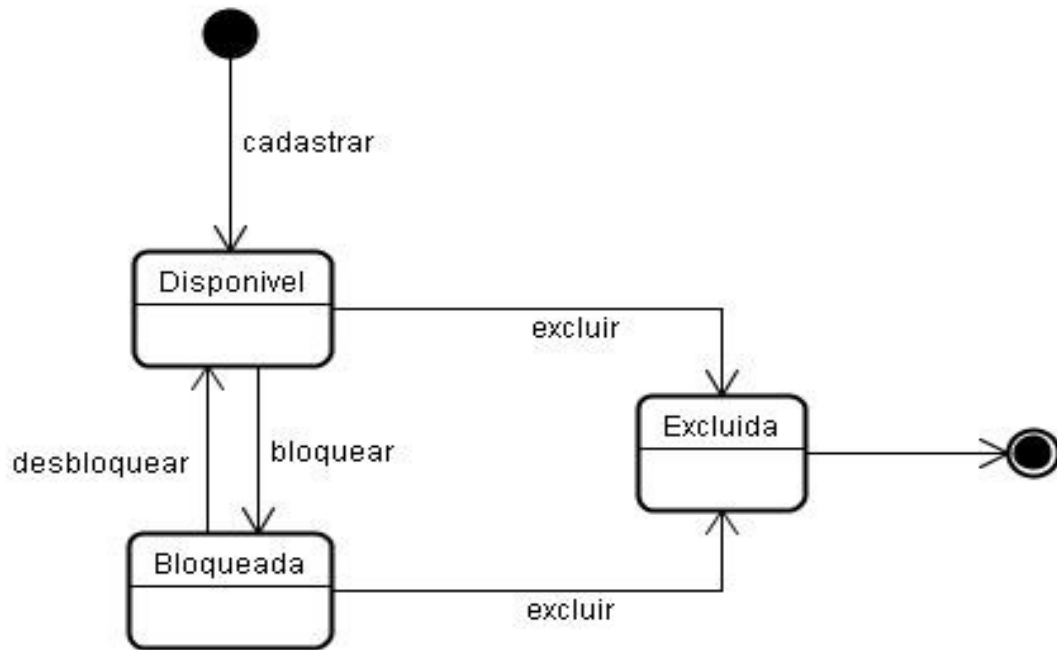


Figura 16 – Diagrama de Transição de Estados – Classe EMPRESA

2.5.10. Classe AVALIAÇÃO

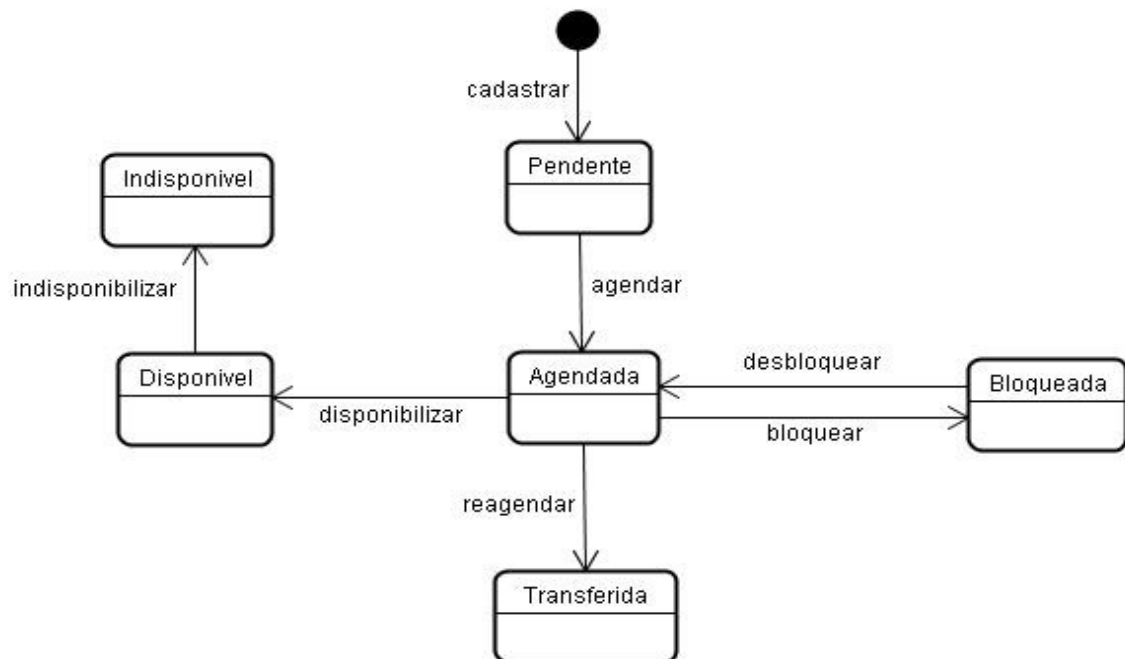


Figura 17 – Diagrama de Transição de Estados – Classe AVALIAÇÃO

2.5.11. Classe INSCRIÇÃO

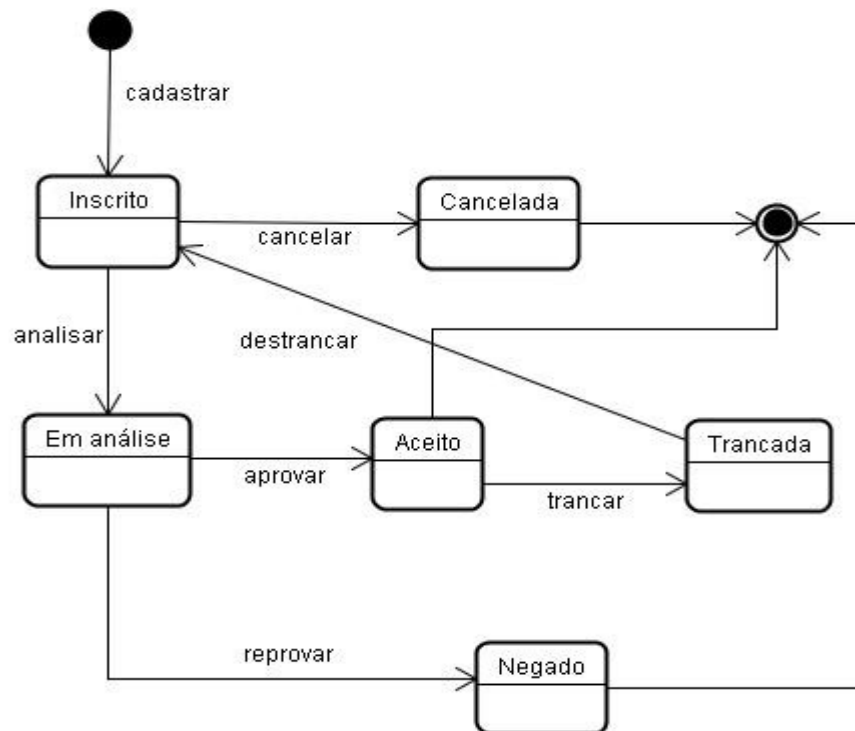


Figura 18 – Diagrama de Transição de Estados – Classe INSCRIÇÃO

2.5.12. Classe USUÁRIO

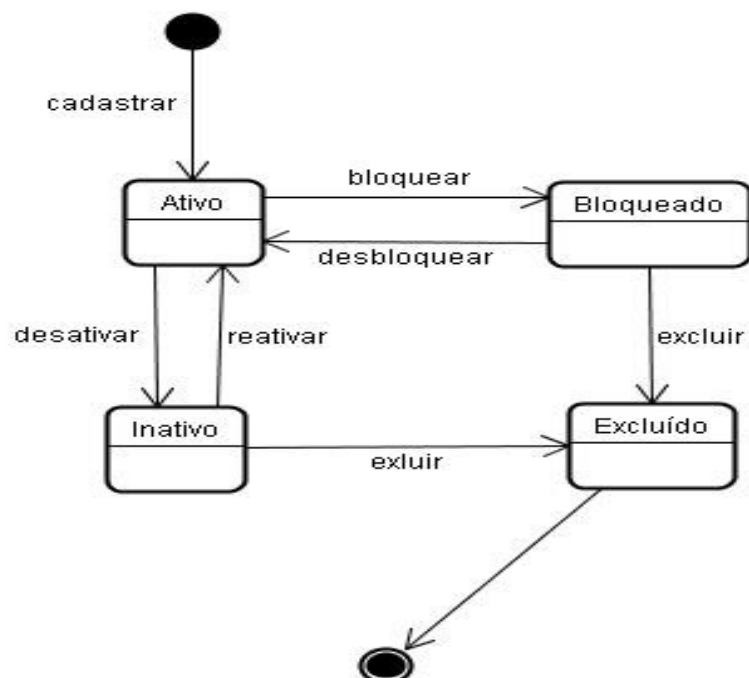


Figura 19 – Diagrama de Transição de Estados – Classe USUÁRIO

2.6. Diagramas de Seqüência

2.6.1. Carga Inicial das Tabelas

Figura 20 – Diagrama de Seqüência – Evento 1

2.6.2. Carga das Tabelas de Consultas Parametrizadas das Avaliações

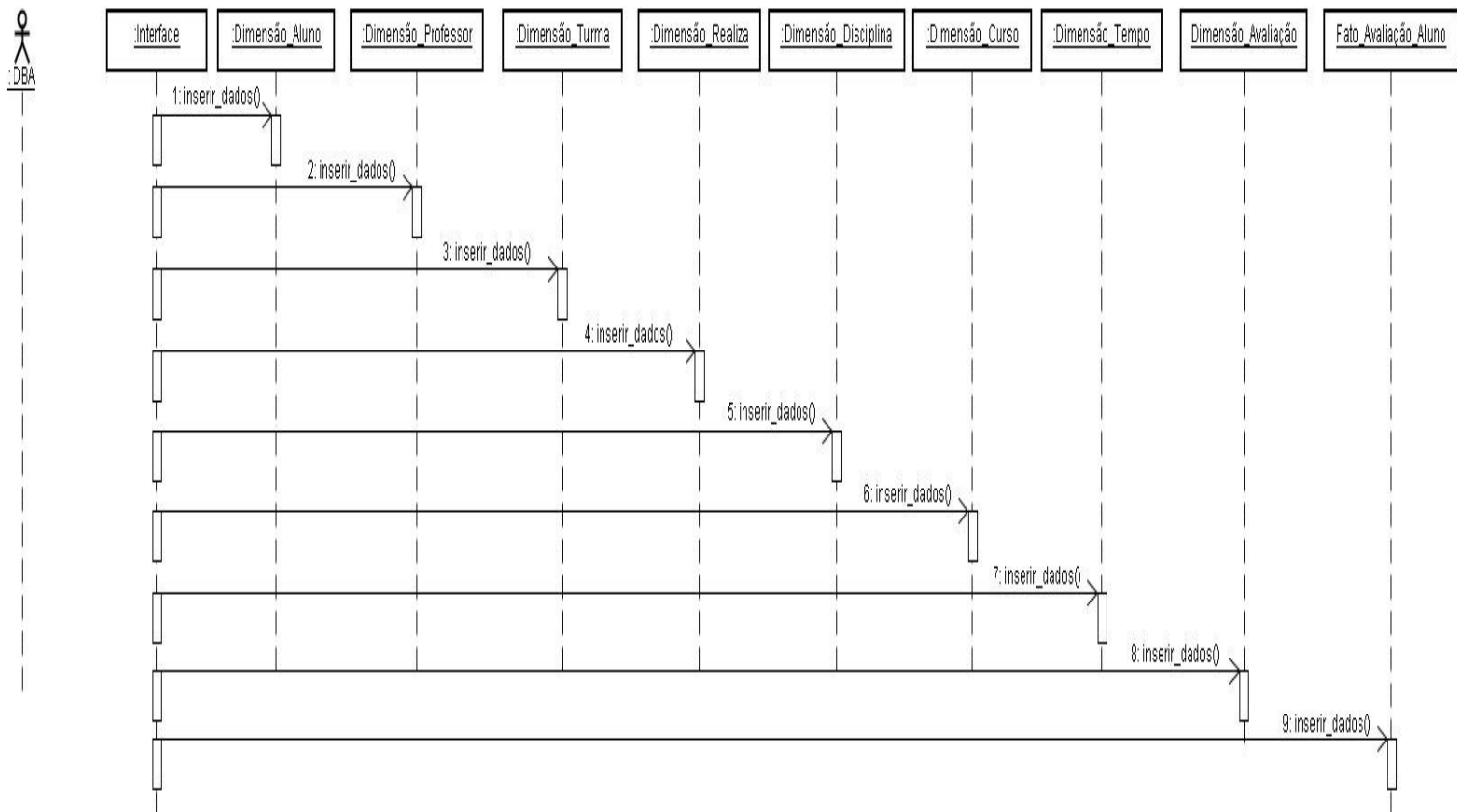


Figura 21 – Diagrama de Seqüência – Evento 2

2.6.3. Carga das Tabelas de Consultas Parametrizadas das Frequências

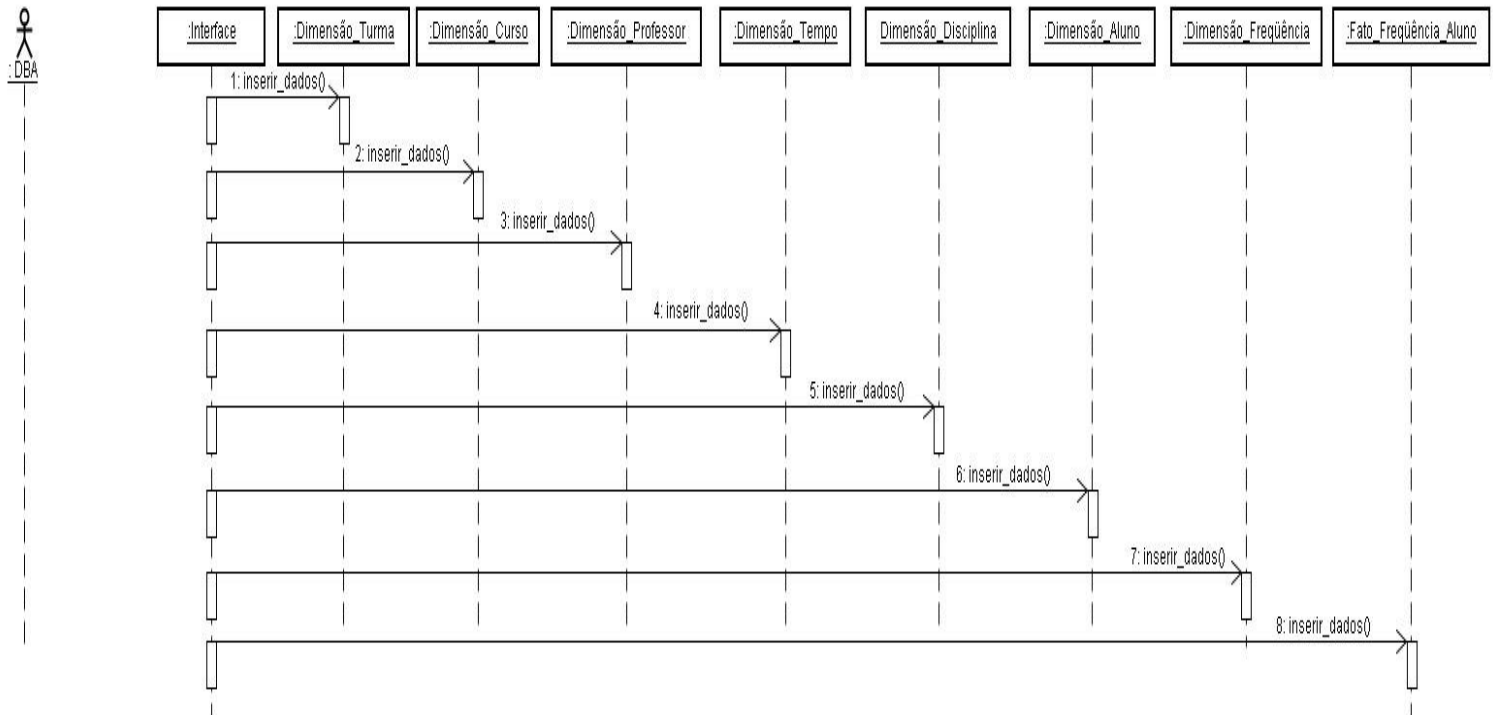


Figura 22 – Diagrama de Seqüência – Evento 3

2.6.4. Cadastrar Usuários

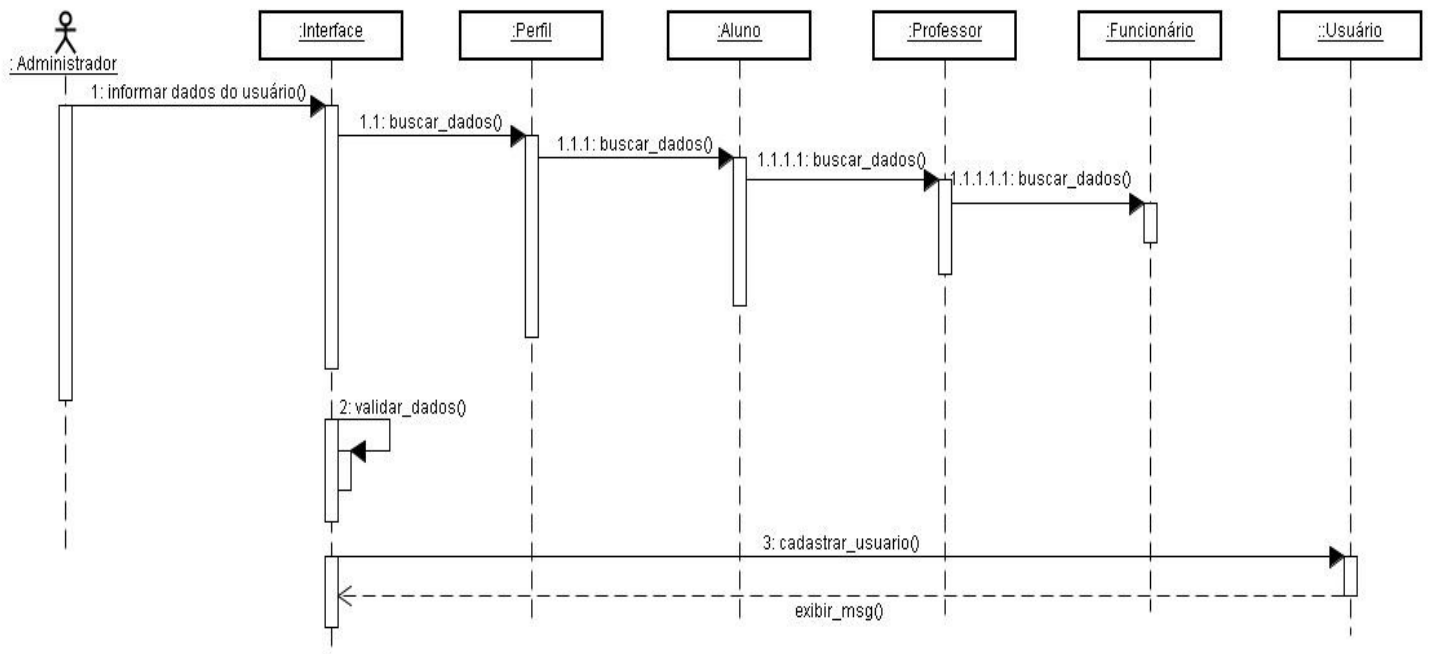


Figura 23 – Diagrama de Seqüência – Evento 4

2.6.5. Cadastrar Perfis dos Usuários

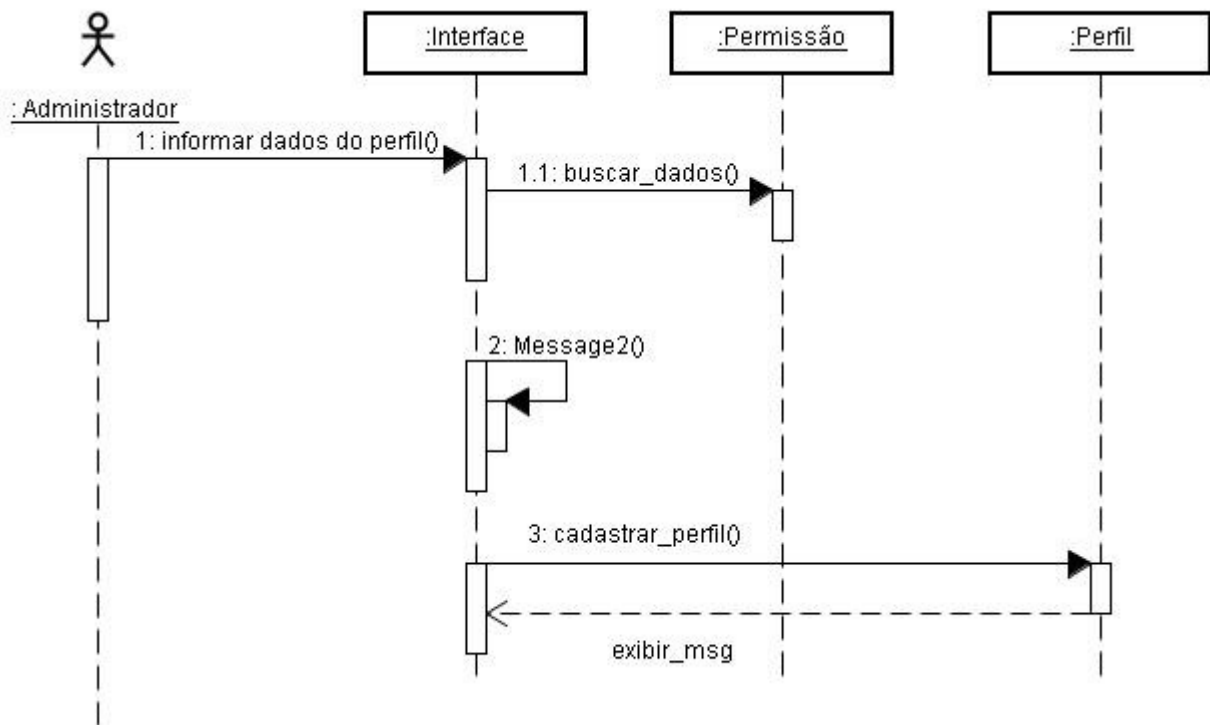


Figura 24 – Diagrama de Seqüência – Evento 5

2.6.6. Cadastrar Instituição de Ensino

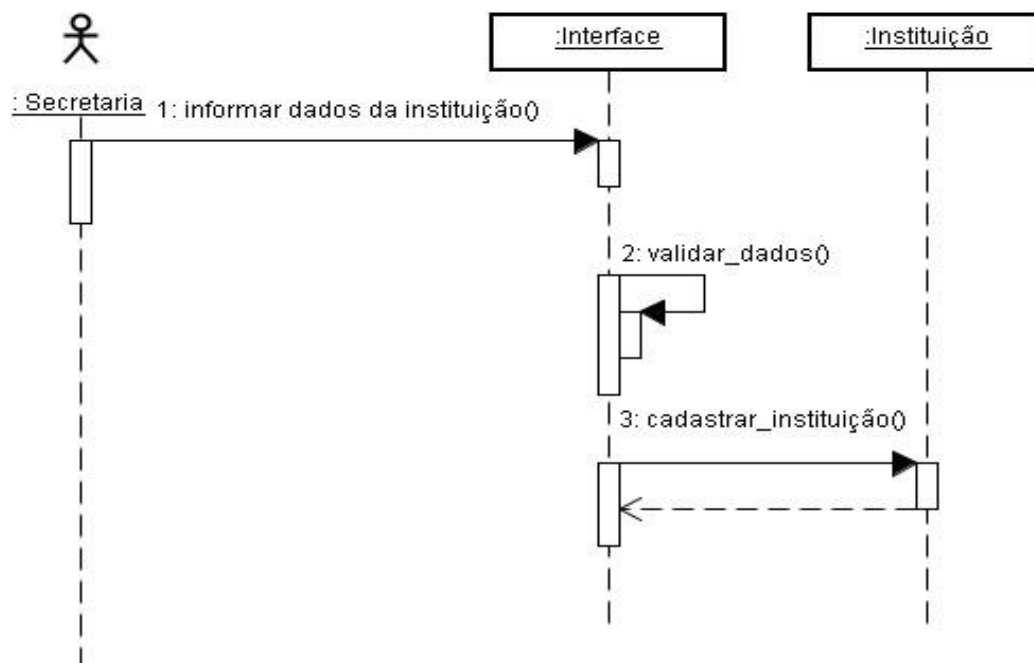


Figura 25 – Diagrama de Seqüência – Evento 6

2.6.7. Cadastrar Curso

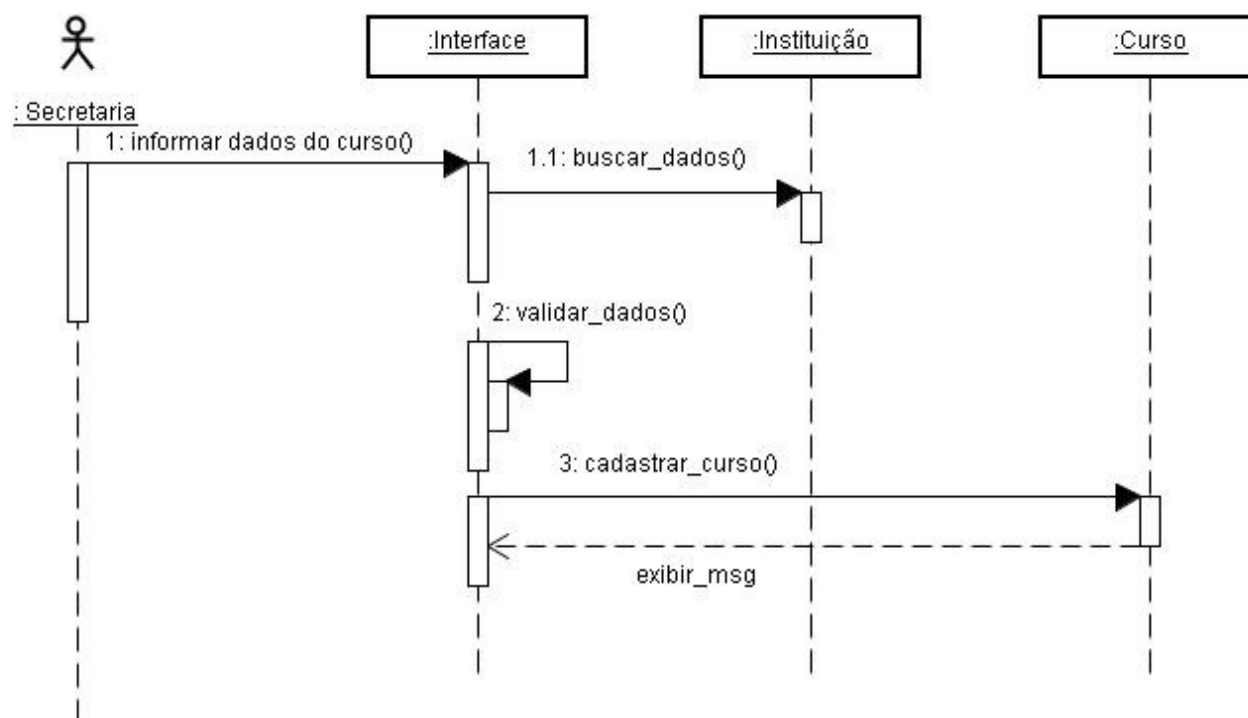


Figura 26 – Diagrama de Seqüência – Evento 7

2.6.8. Cadastrar Grade

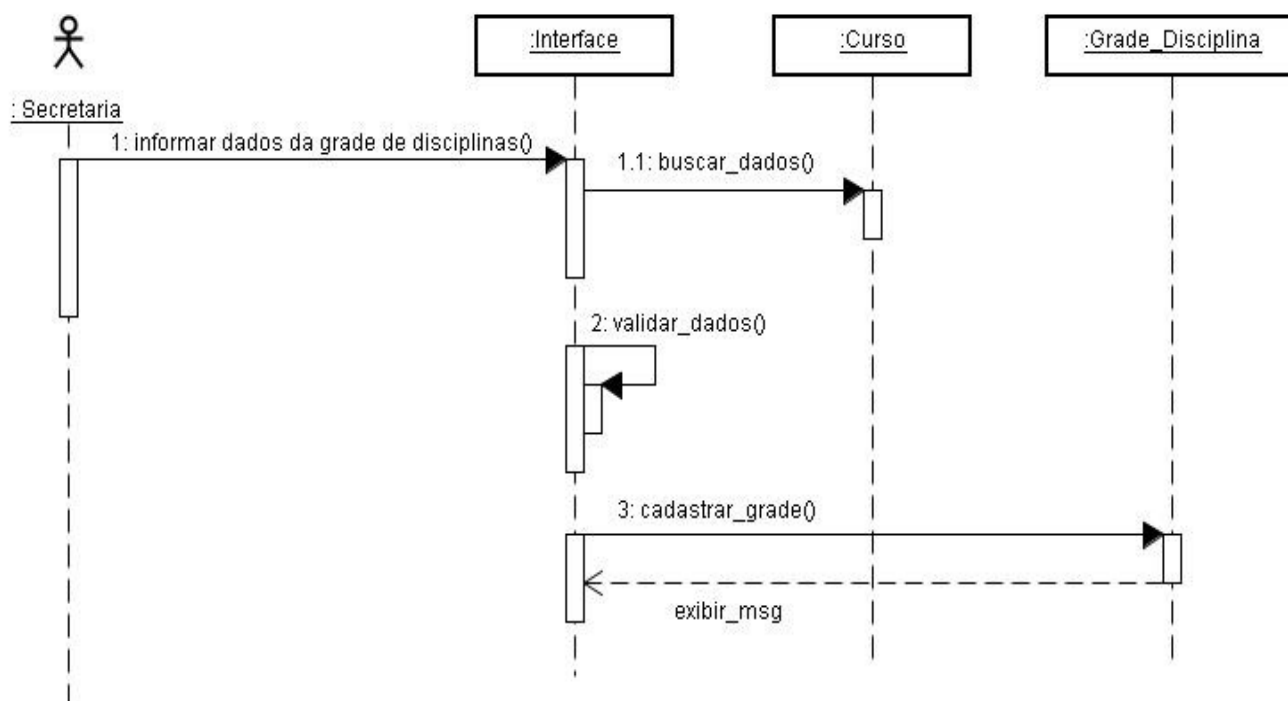


Figura 27 – Diagrama de Seqüência – Evento 8

2.6.9. Cadastrar Disciplina

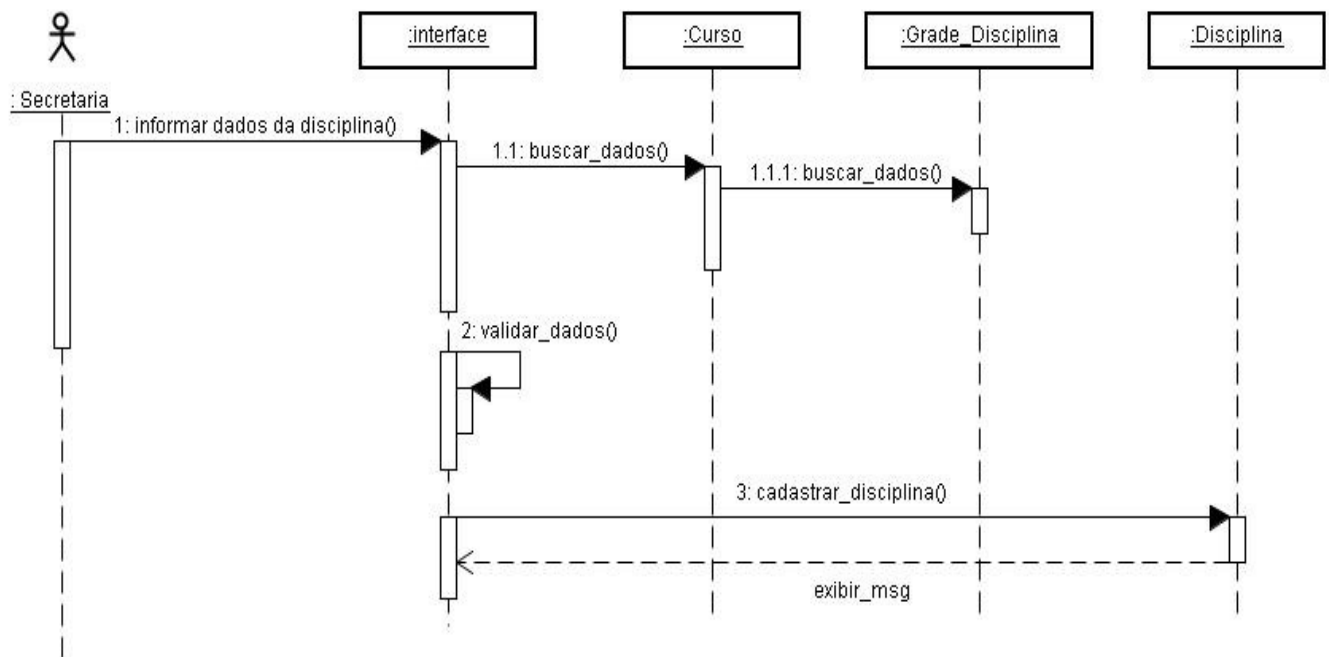


Figura 28 – Diagrama de Seqüência – Evento 9

2.6.10. Cadastrar Ementa

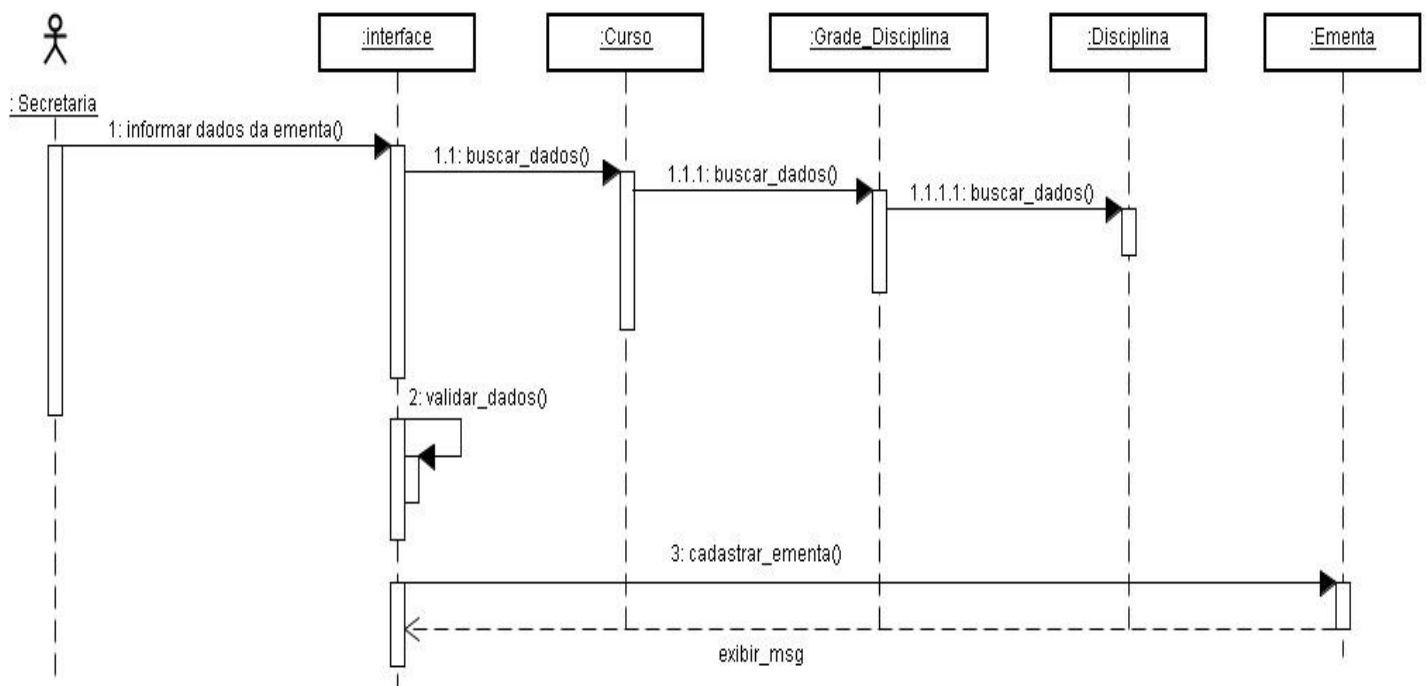


Figura 29 – Diagrama de Seqüência – Evento 10

2.6.11. Cadastrar Conteúdo

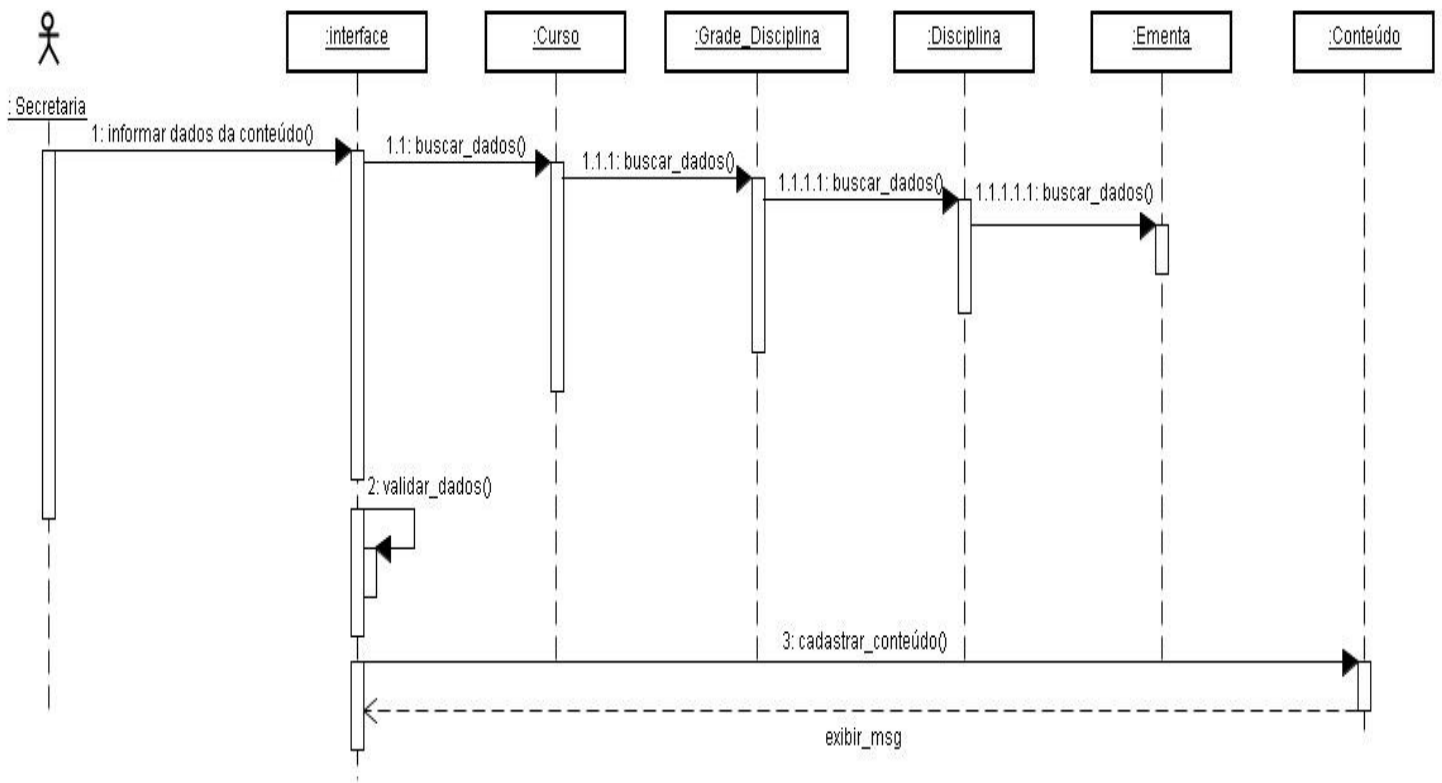


Figura 30 – Diagrama de Seqüência – Evento 11

2.6.12. Cadastrar Turma

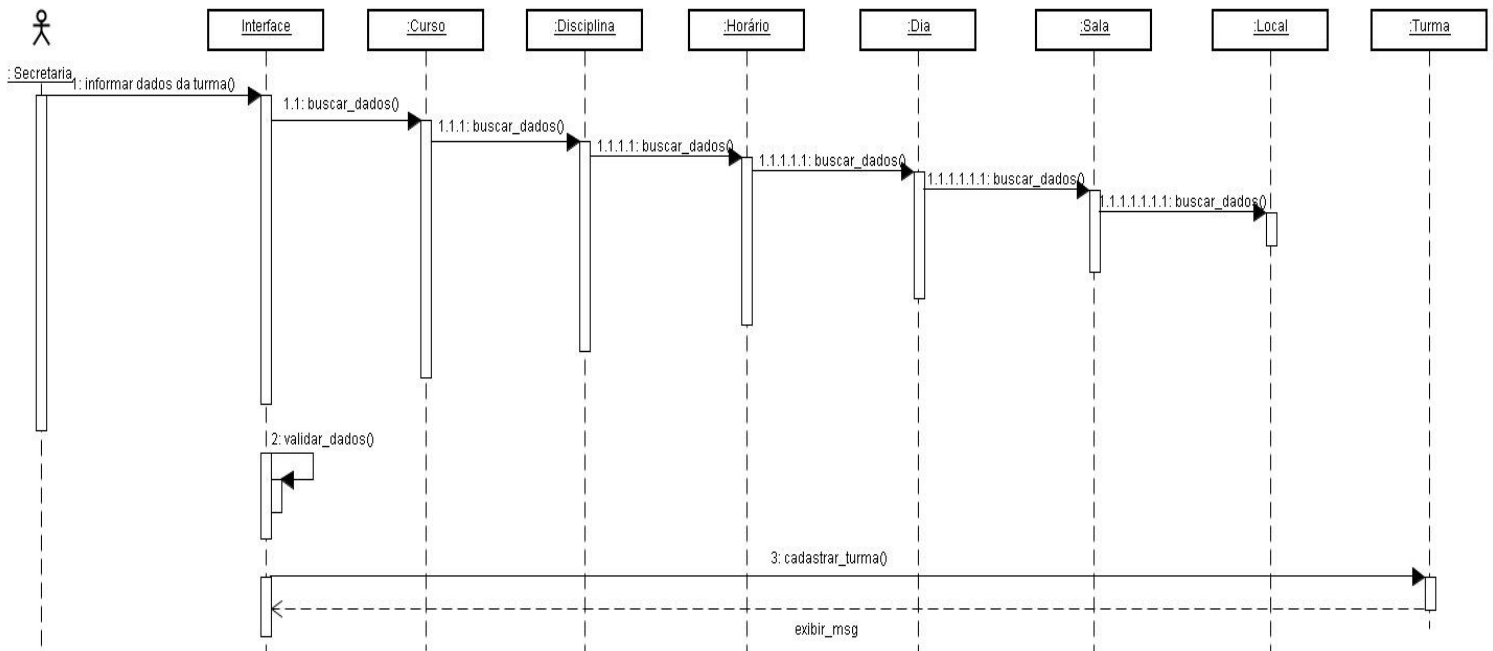


Figura 31 – Diagrama de Seqüência – Evento 12

2.6.13. Cadastrar Aluno

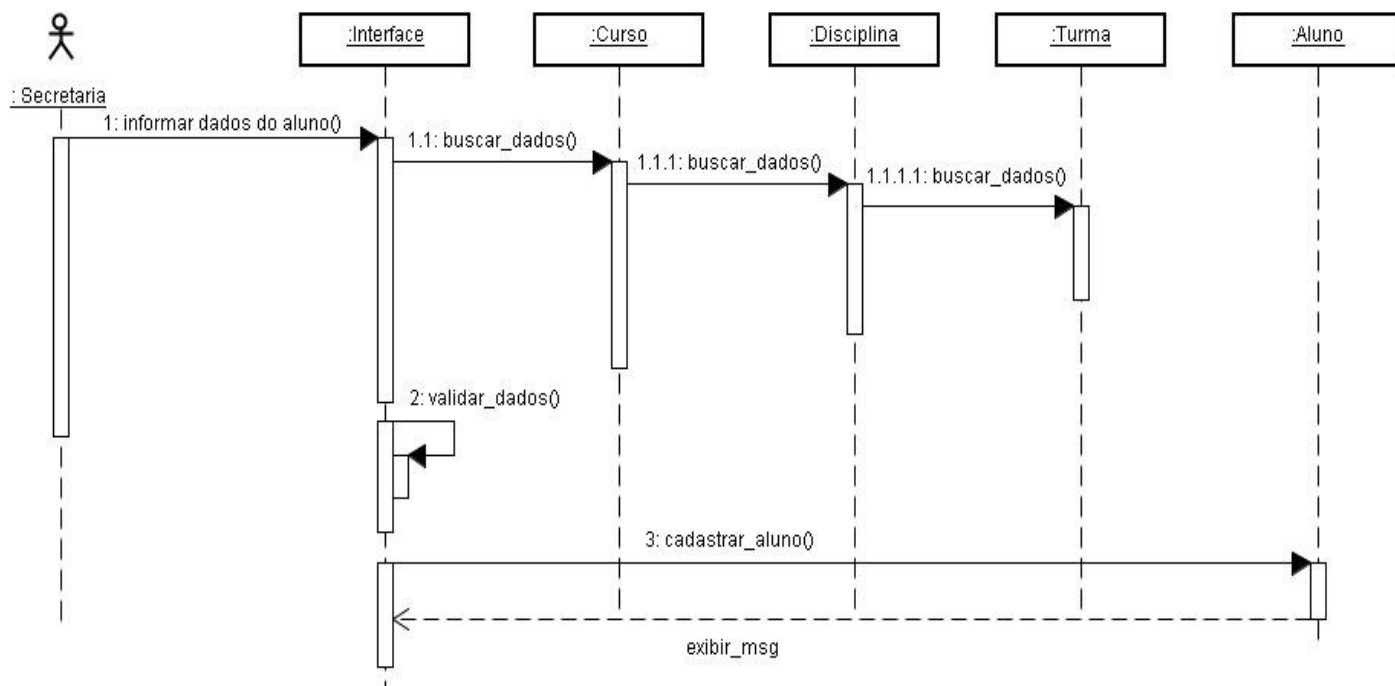


Figura 32 – Diagrama de Seqüência – Evento 13

2.6.14. Cadastrar Professor

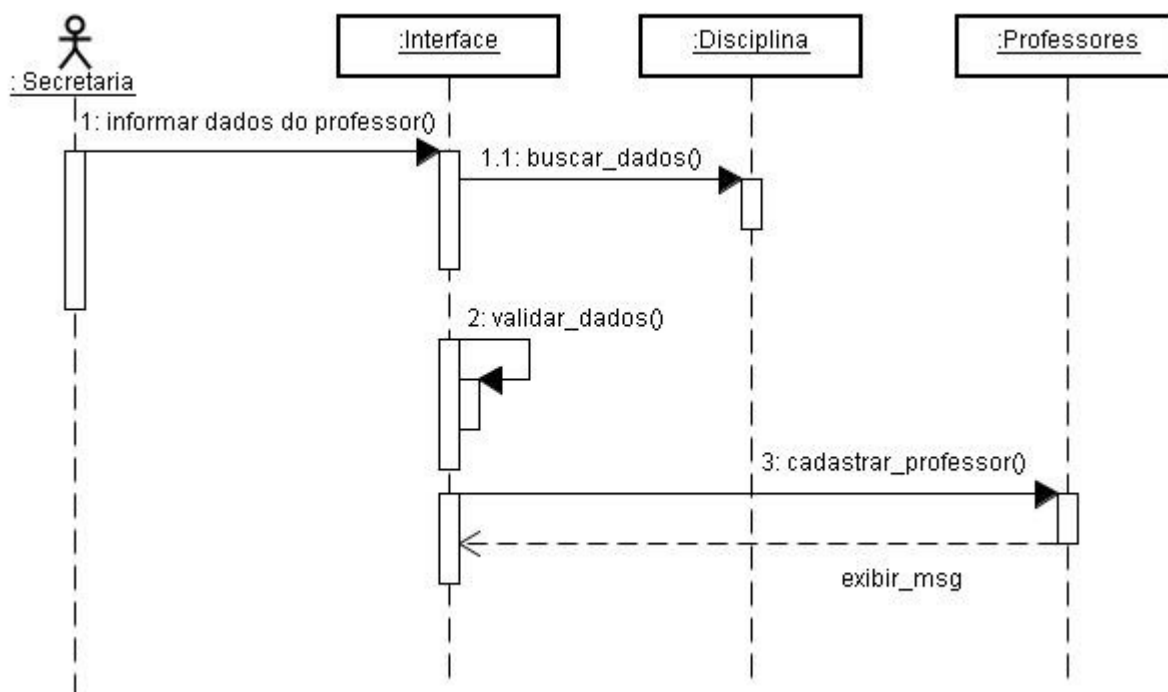


Figura 33 – Diagrama de Seqüência – Evento 14

2.6.15. Cadastrar Vaga de Estágio

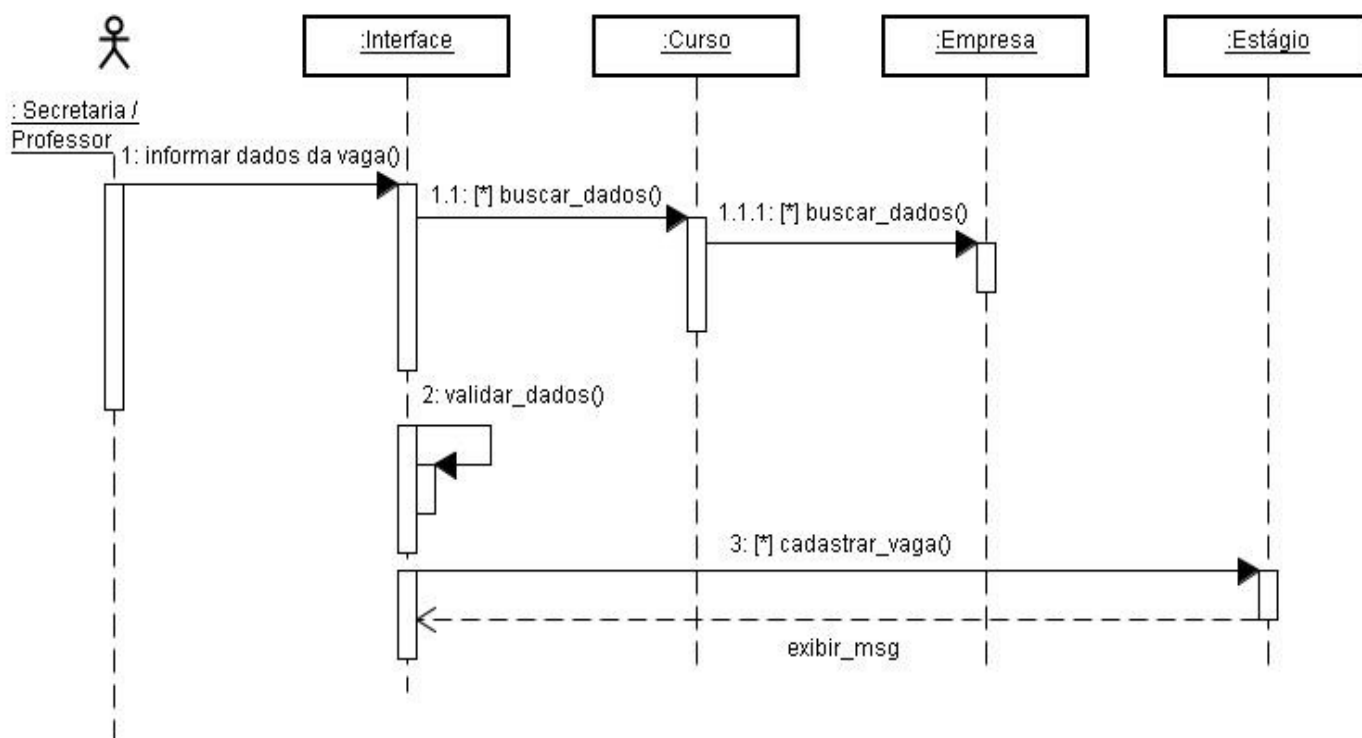


Figura 34 – Diagrama de Seqüência – Evento 15

2.6.16. Cadastrar Empresa

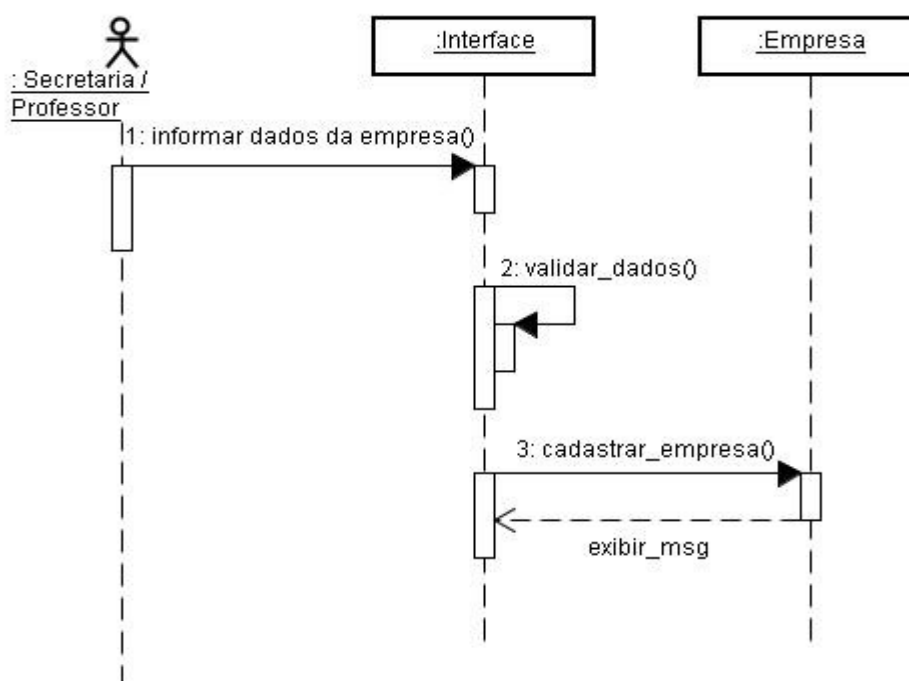


Figura 35 – Diagrama de Seqüência – Evento 16

2.6.17. Cadastrar Recursos

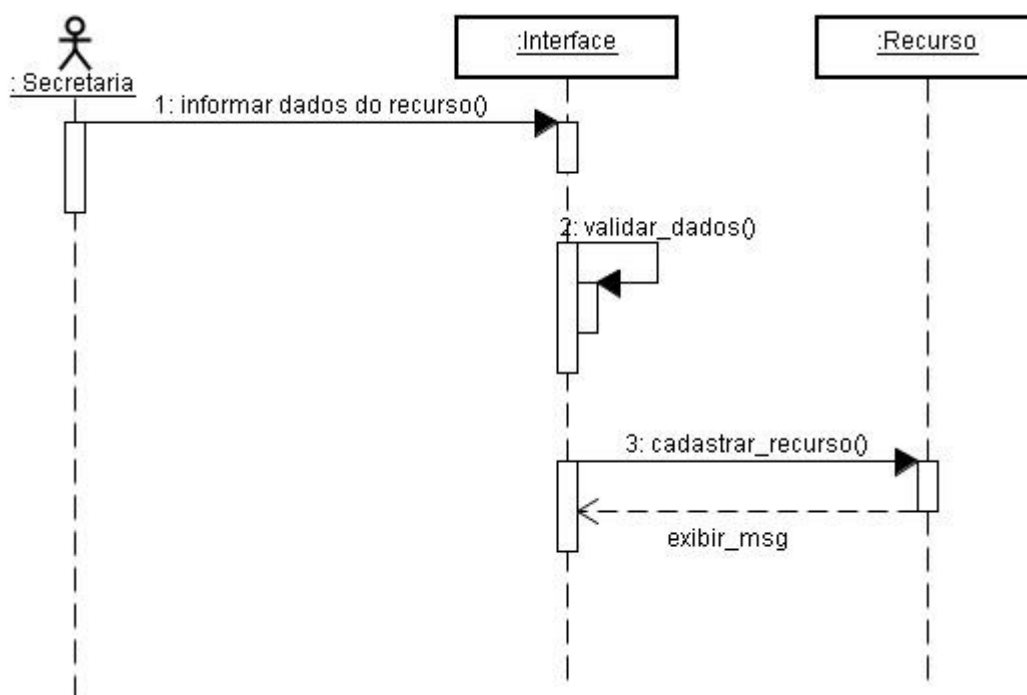


Figura 36 – Diagrama de Seqüência – Evento 17

2.6.18. Alocar Recursos

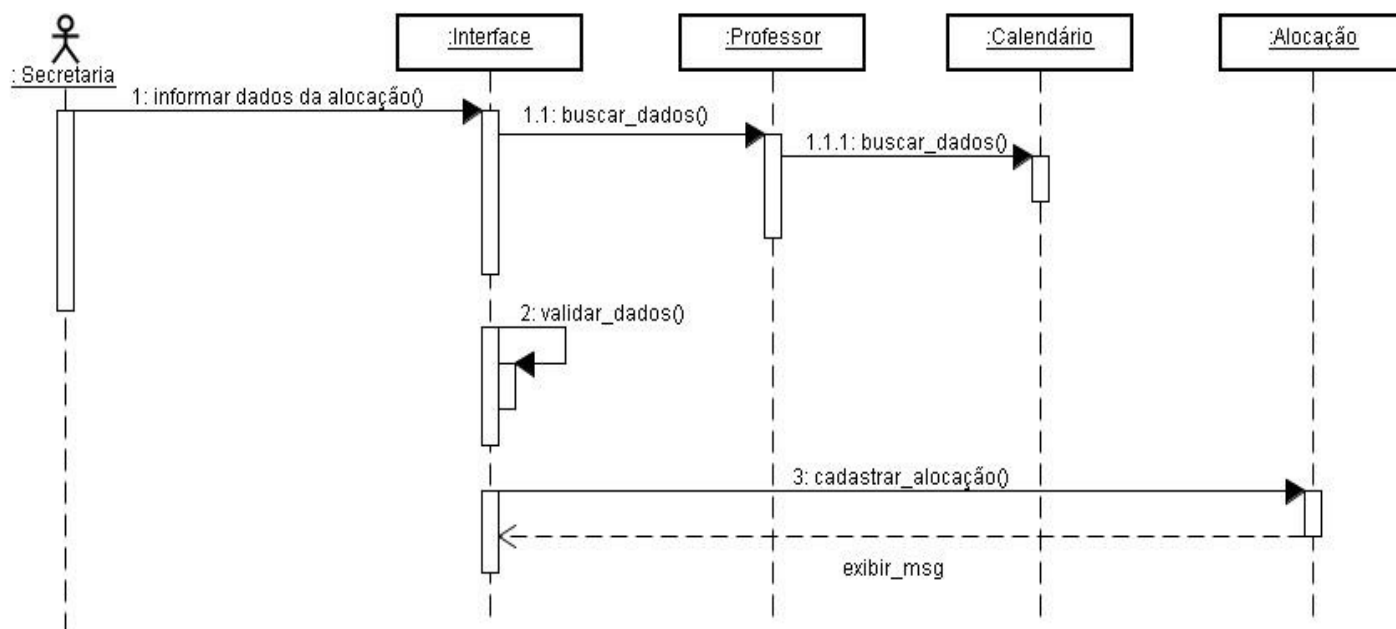


Figura 37 – Diagrama de Seqüência – Evento 18

2.6.19. Cadastrar Local

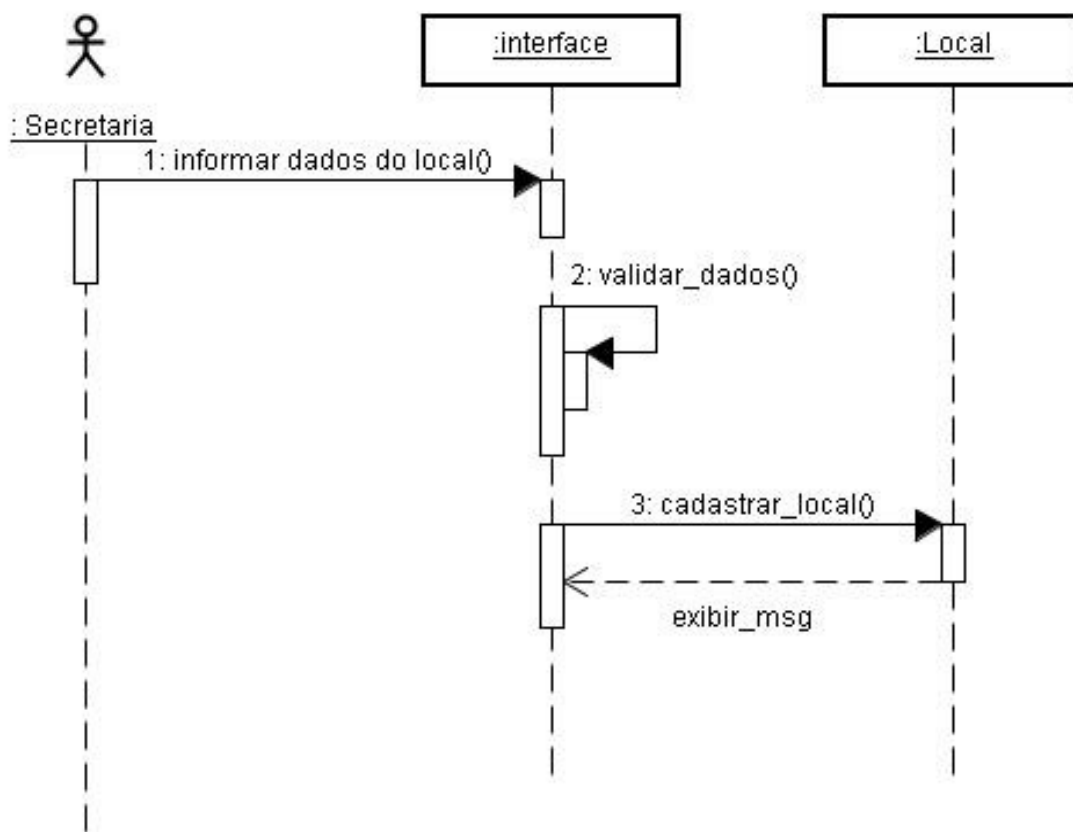


Figura 38 – Diagrama de Seqüência – Evento 19

2.6.20. Cadastrar Sala

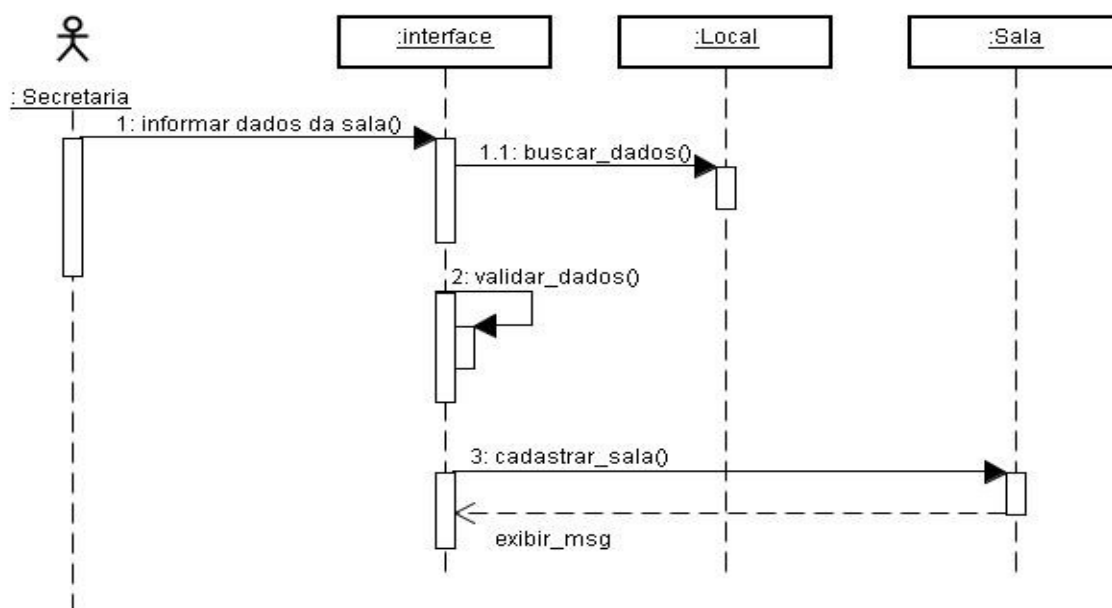


Figura 39 – Diagrama de Seqüência – Evento 20

2.6.21. Agendar Avaliação

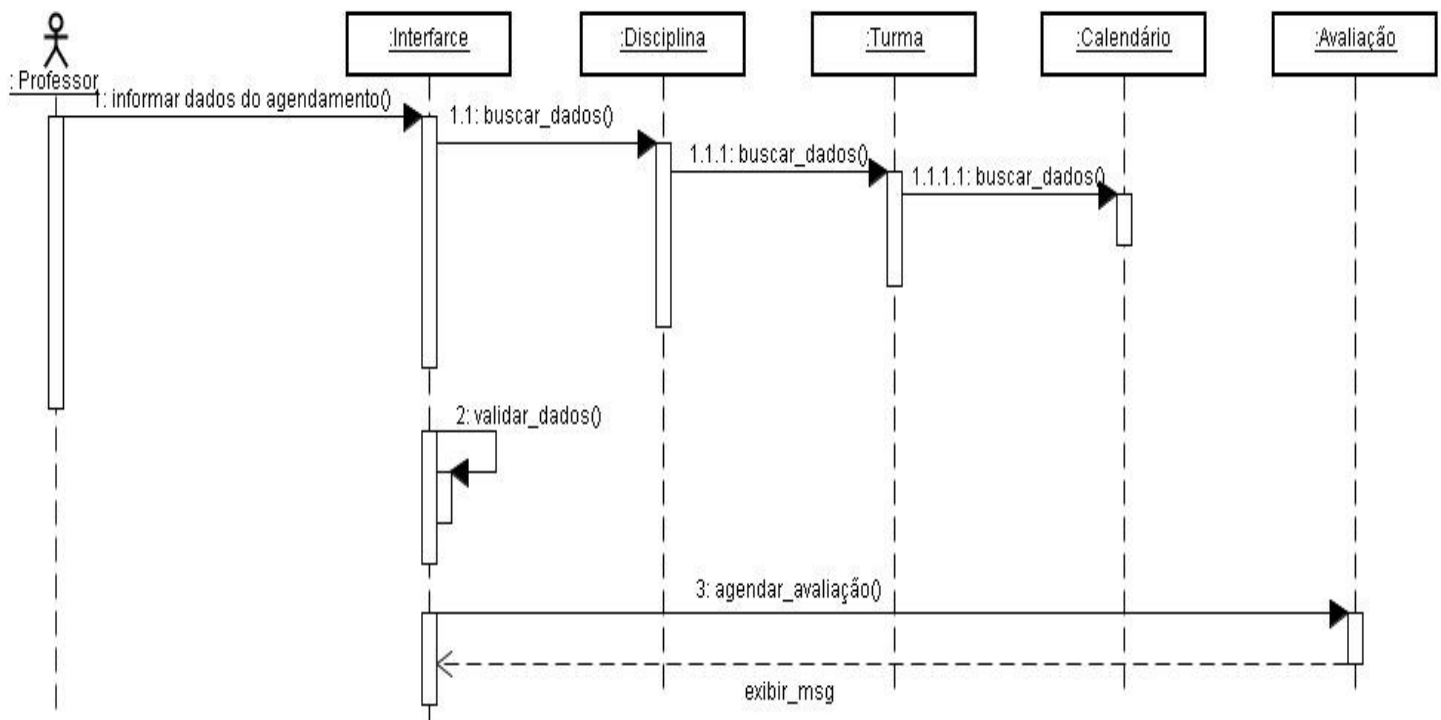


Figura 40 – Diagrama de Seqüência – Evento 21

2.6.22. Disponibilizar Materiais

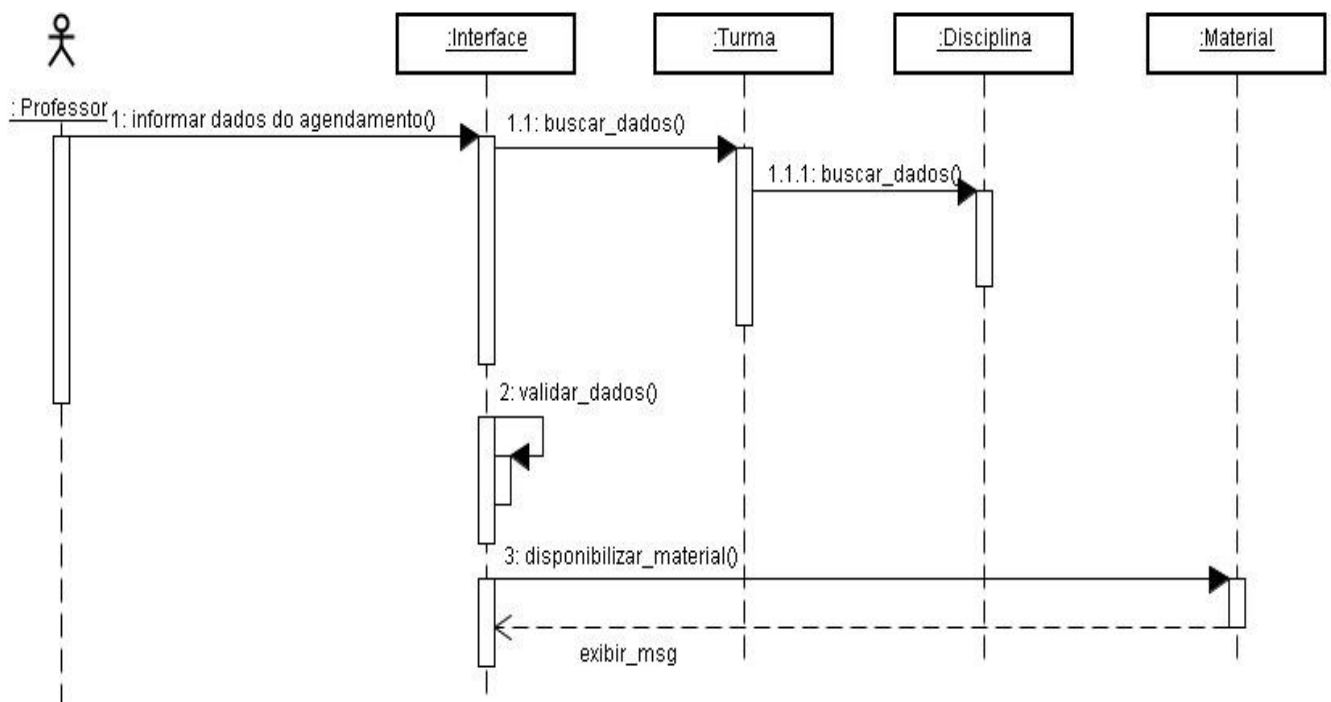


Figura 41– Diagrama de Seqüência – Evento 22

2.6.23. Lançar Frequência

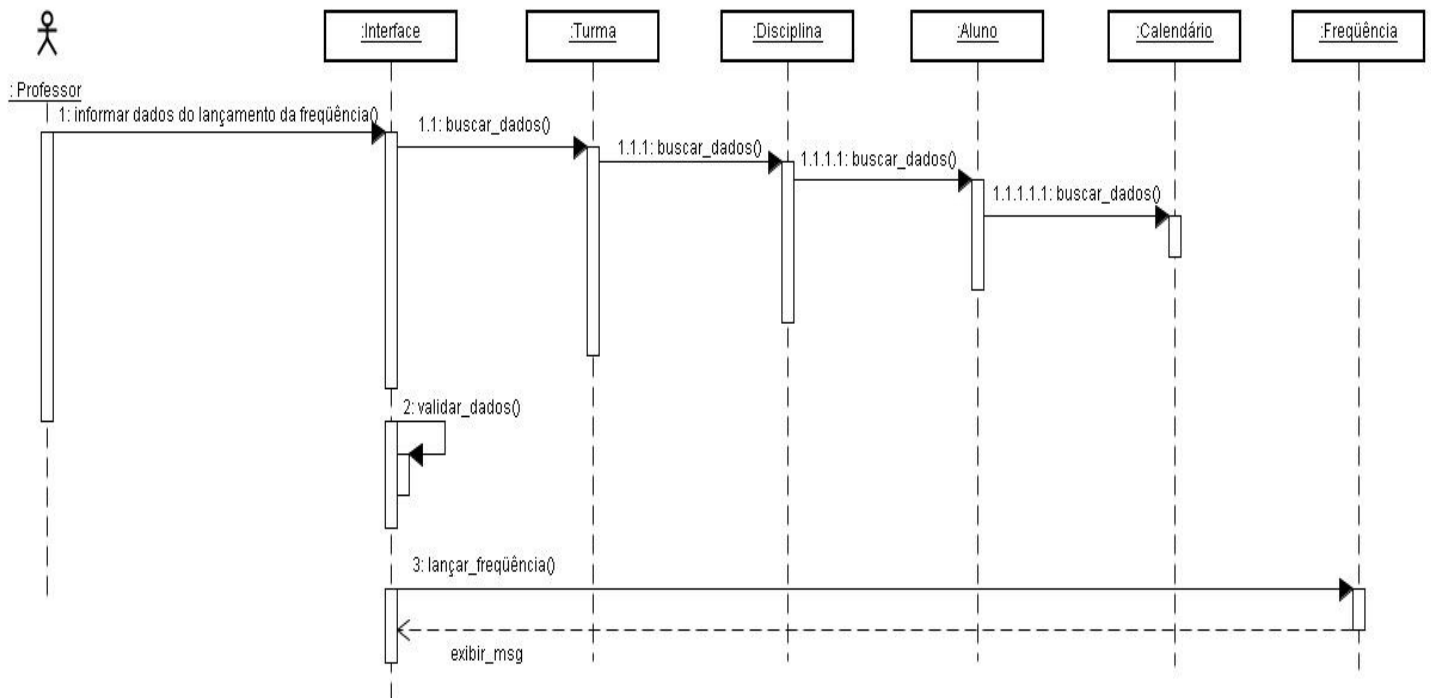


Figura 42 – Diagrama de Seqüência – Evento 23

2.6.24. Lançar Notas

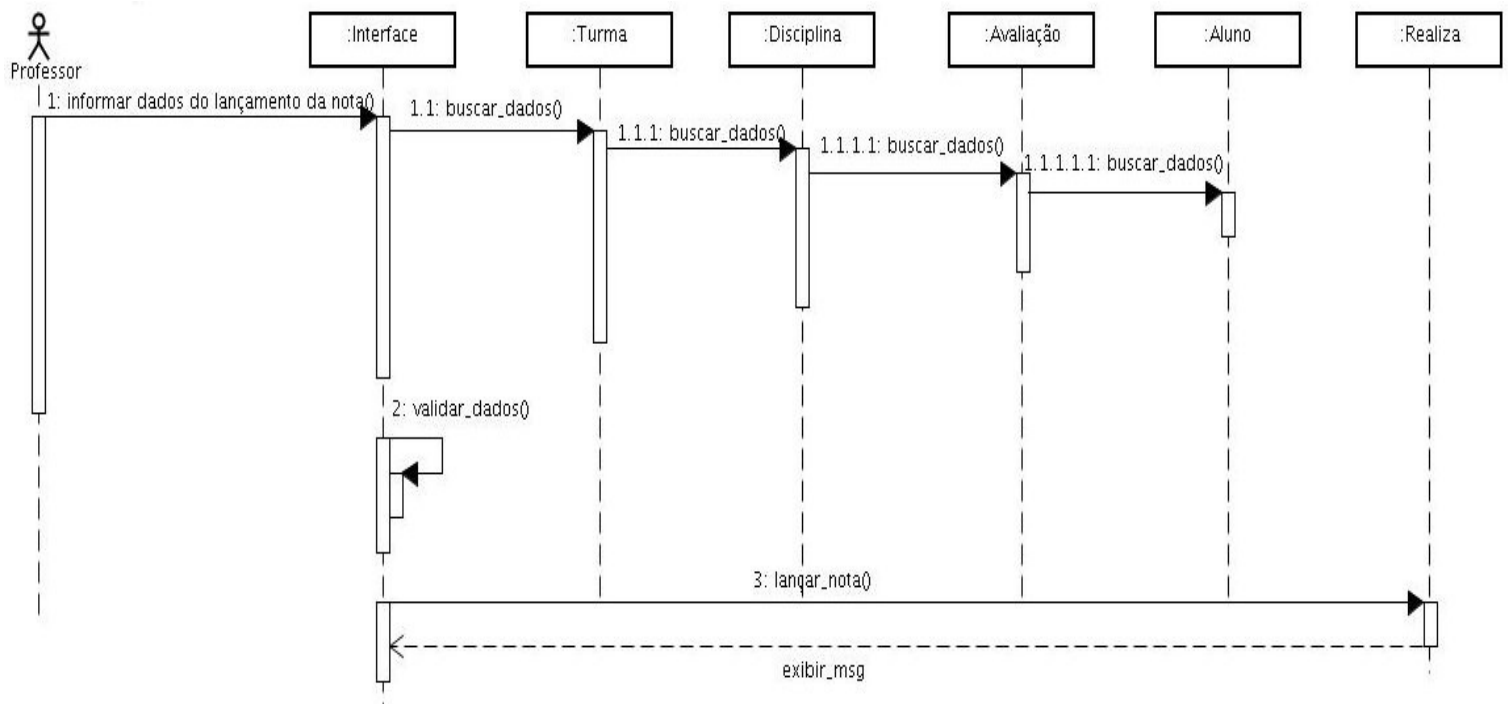


Figura 43 – Diagrama de Seqüência – Evento 24

2.6.25. Lançar Conteúdo Ministrado

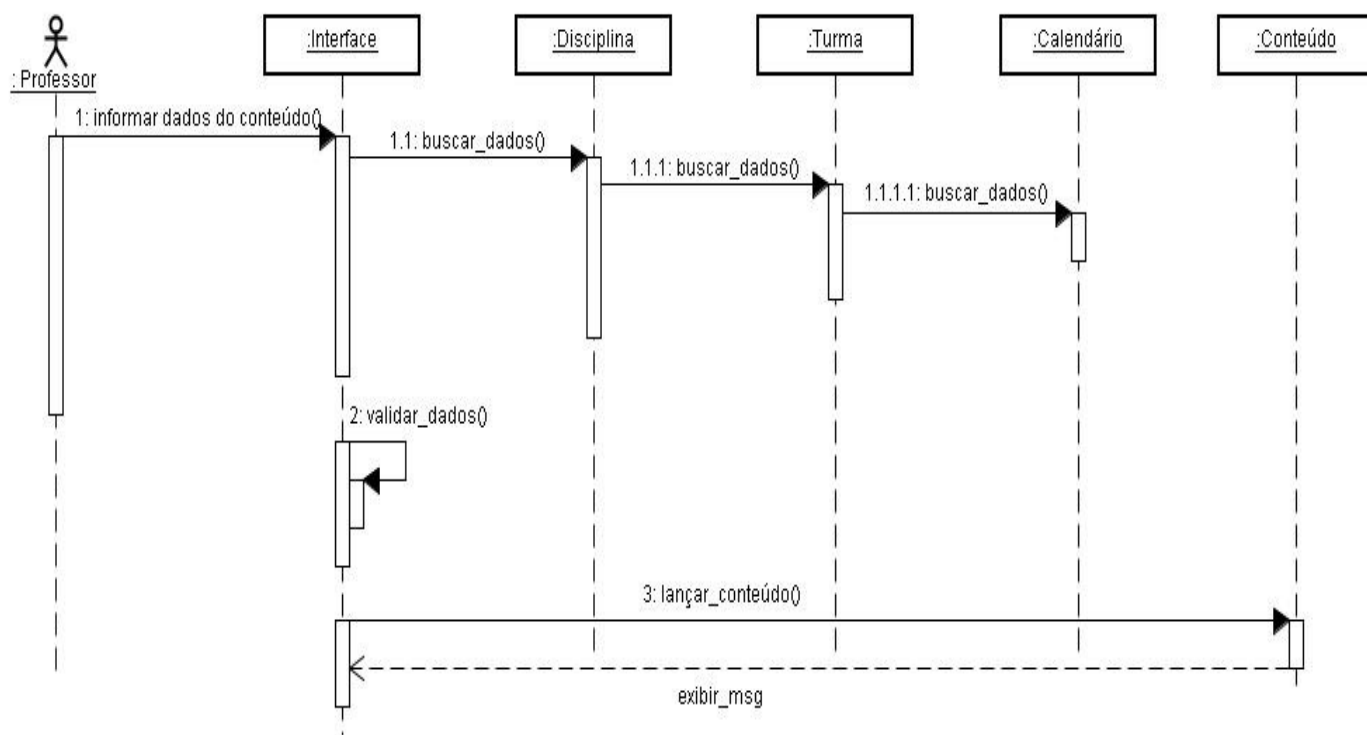


Figura 44 – Diagrama de Seqüência – Evento 25

2.6.26. Associar Comentário ao Material

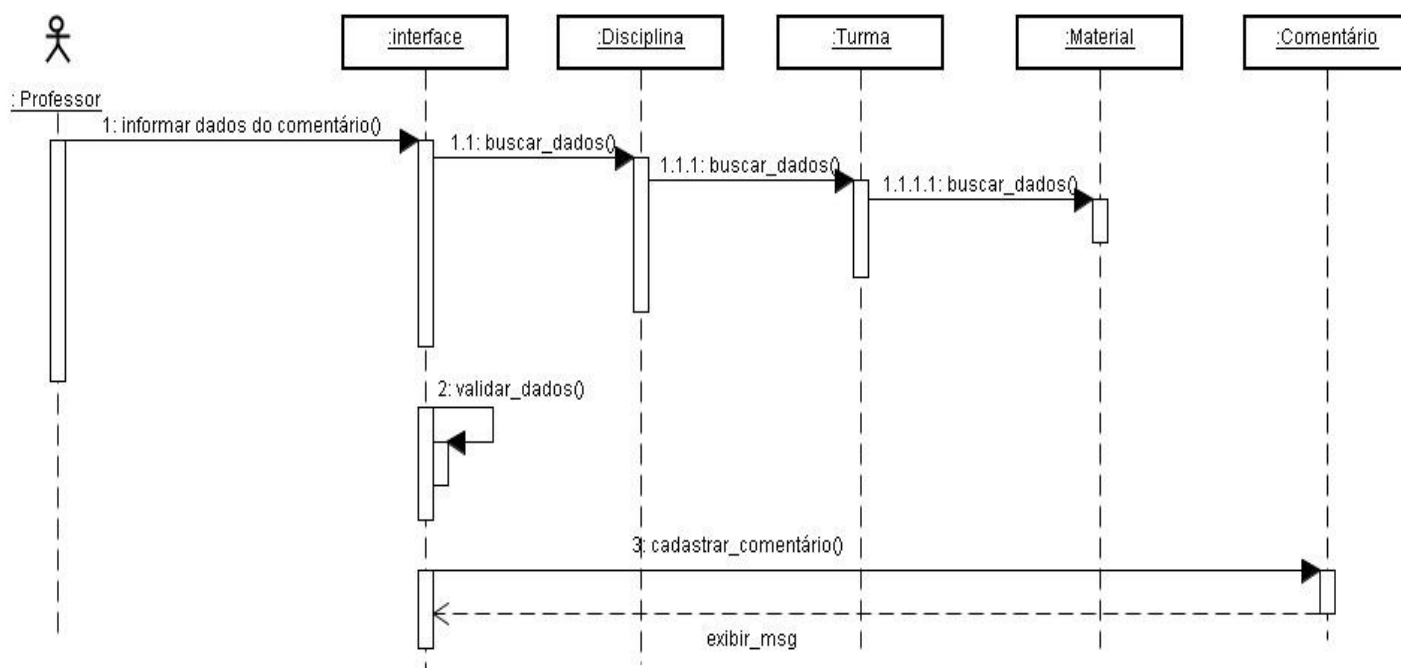


Figura 45 – Diagrama de Seqüência – Evento 26

2.6.27. Professor Solicita Alocação de Recurso

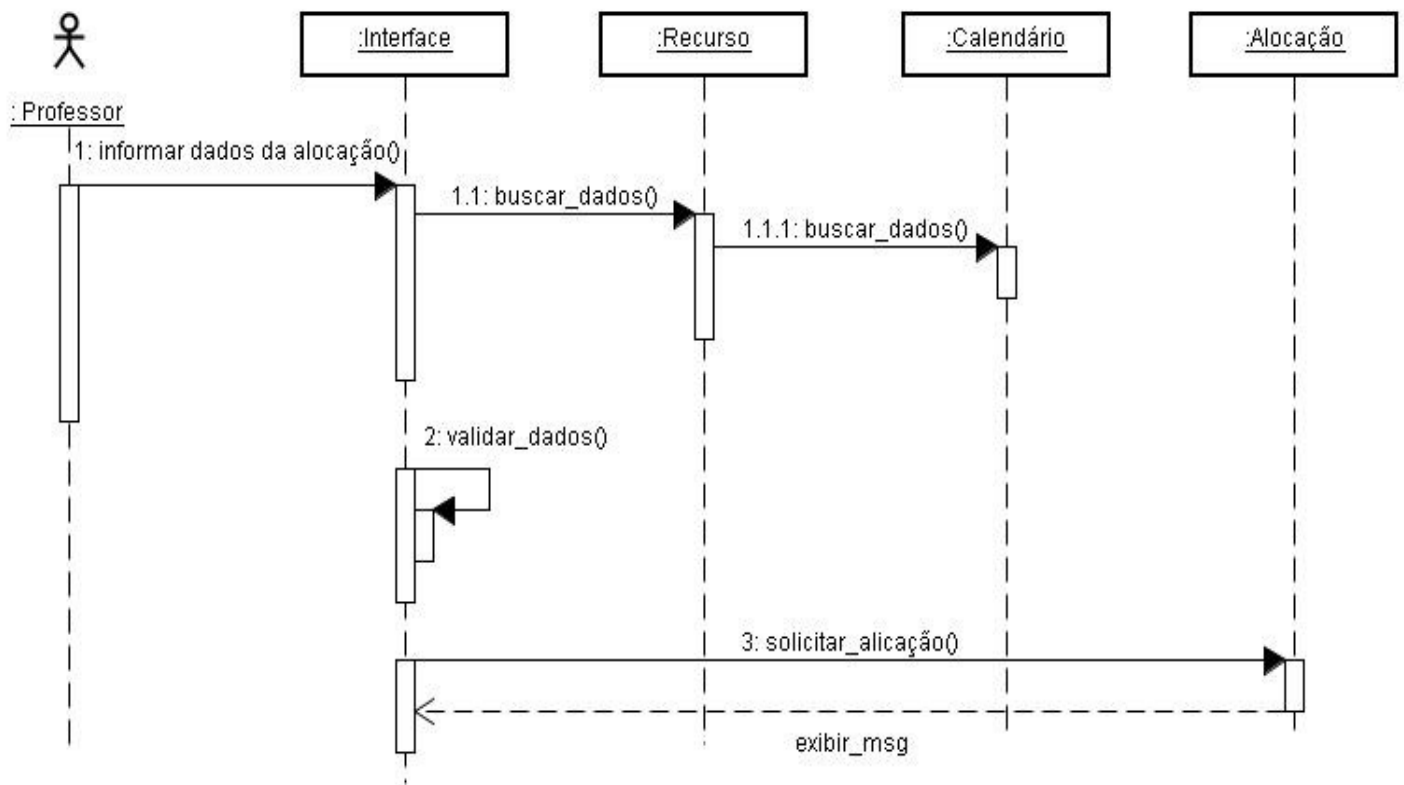


Figura 46 – Diagrama de Seqüência – Evento 27

2.6.28. Realizar Consultas Parametrizadas das Avaliações

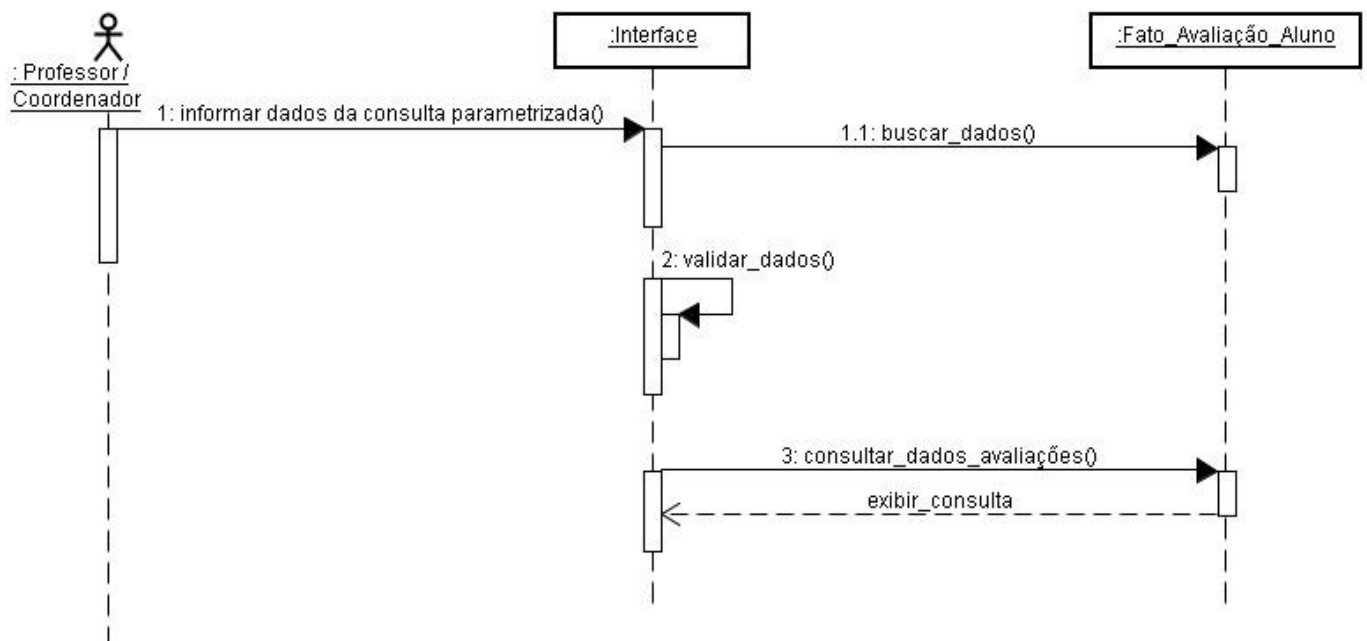


Figura 47 – Diagrama de Seqüência – Evento 28

2.6.29. Realizar Consultas Parametrizadas das Frequências

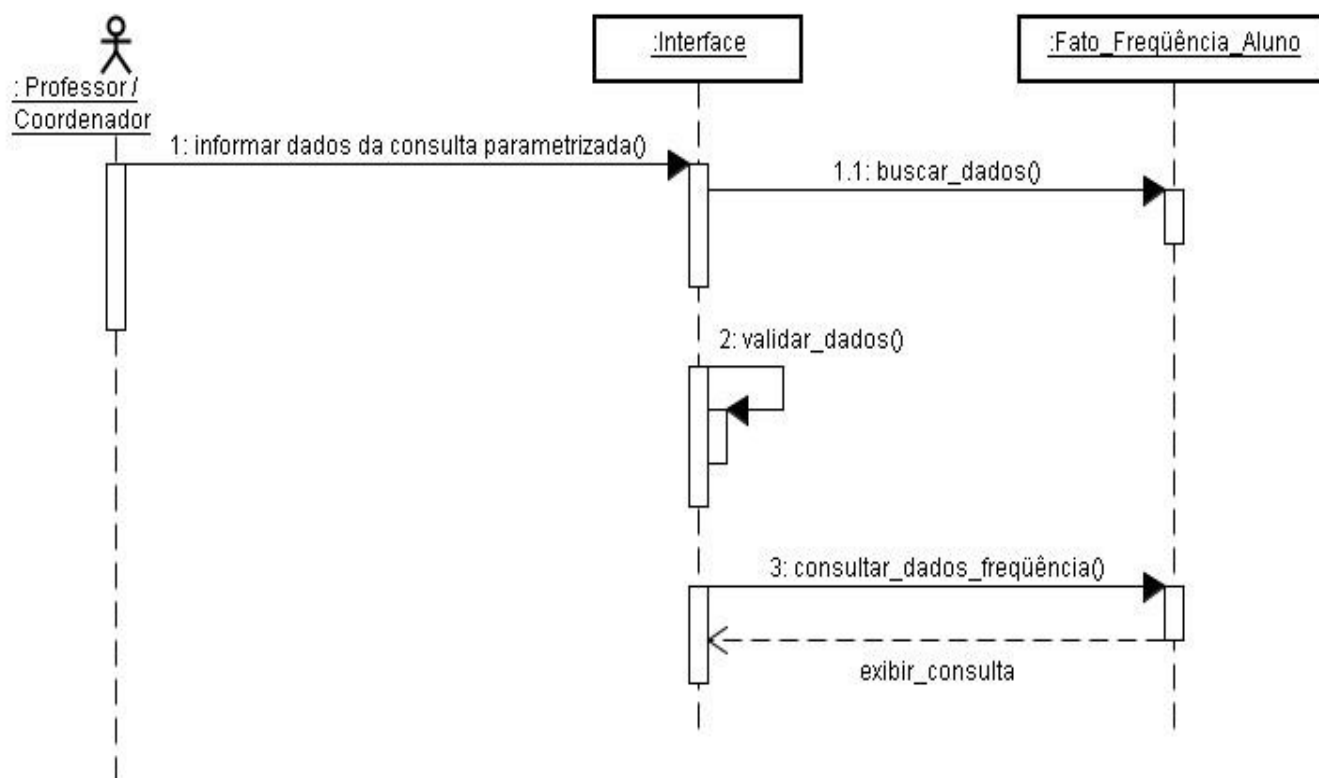


Figura 48 – Diagrama de Seqüência – Evento 29

2.6.30. Consultar Notas das Avaliações

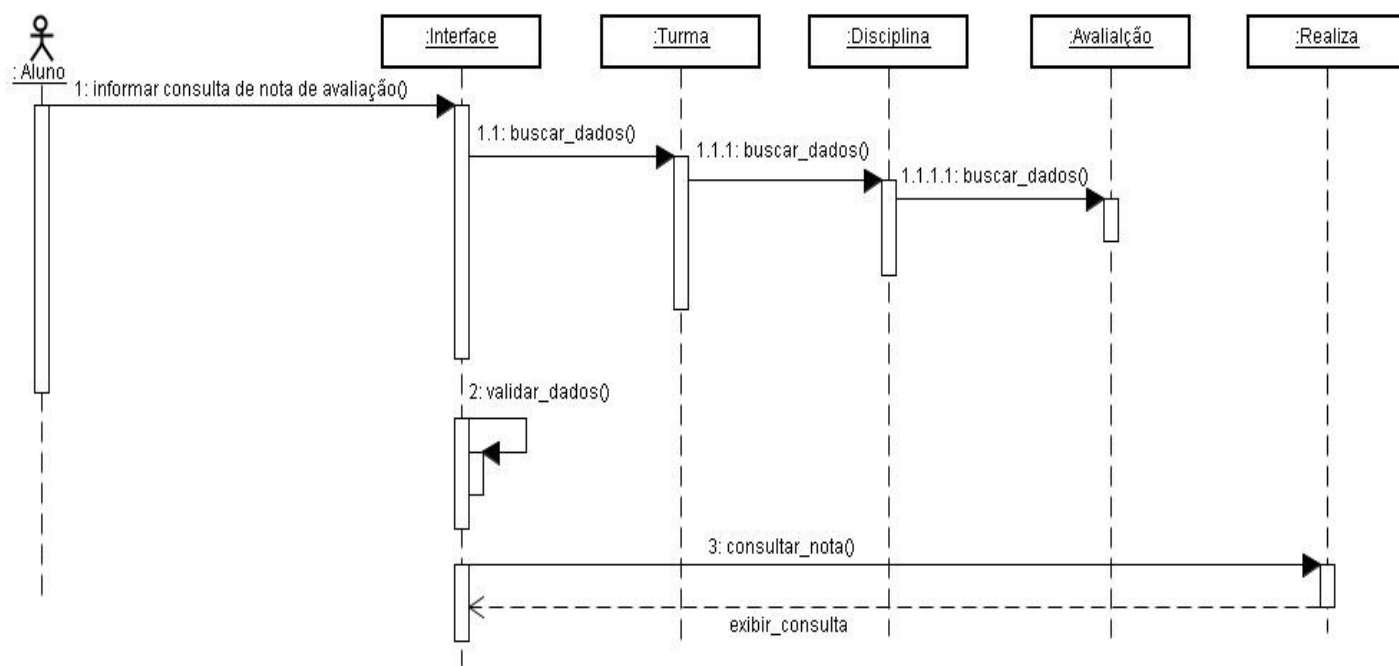


Figura 49 – Diagrama de Seqüência – Evento 30

2.6.31. Realizar Pré-Inscrição em Disciplina

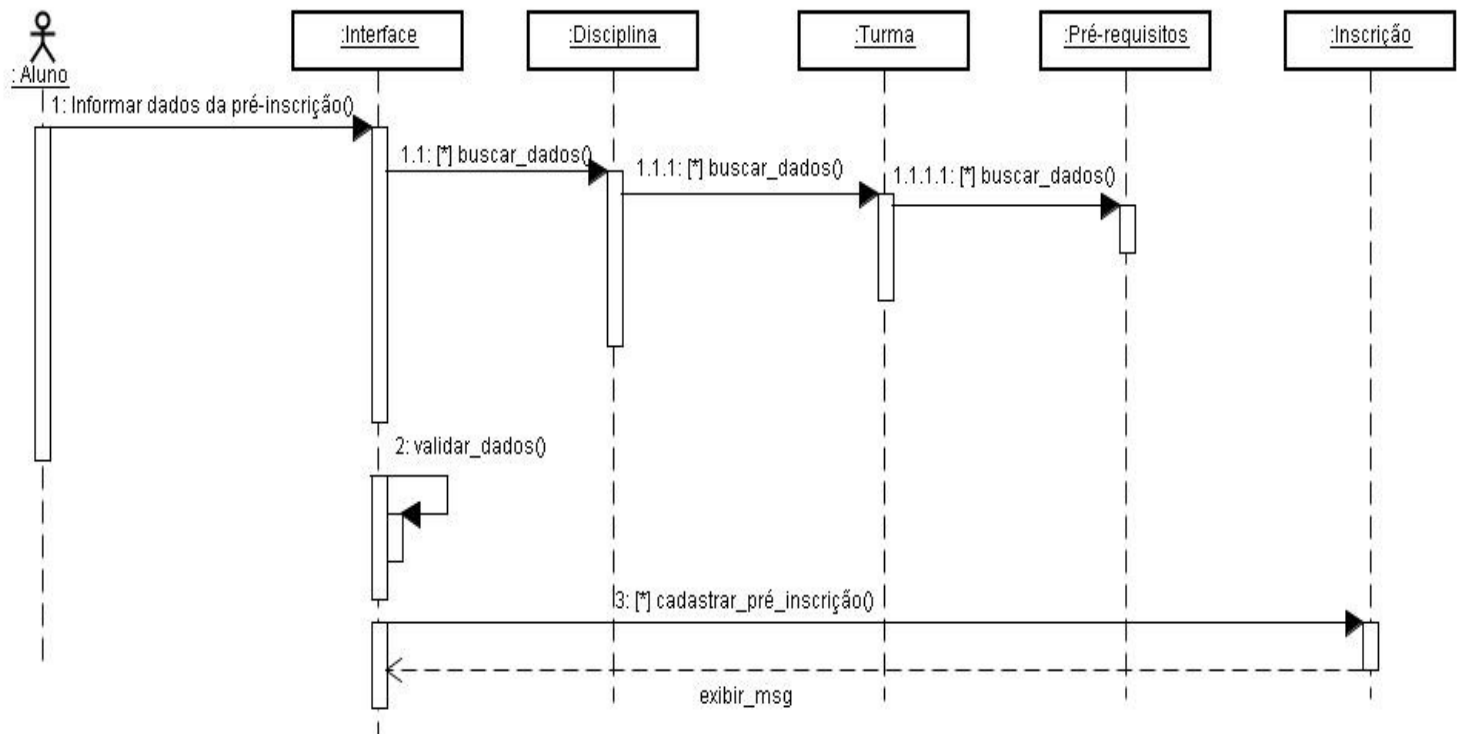


Figura 50 – Diagrama de Seqüência – Evento 31

2.6.32. Solicitar Declaração de Curso

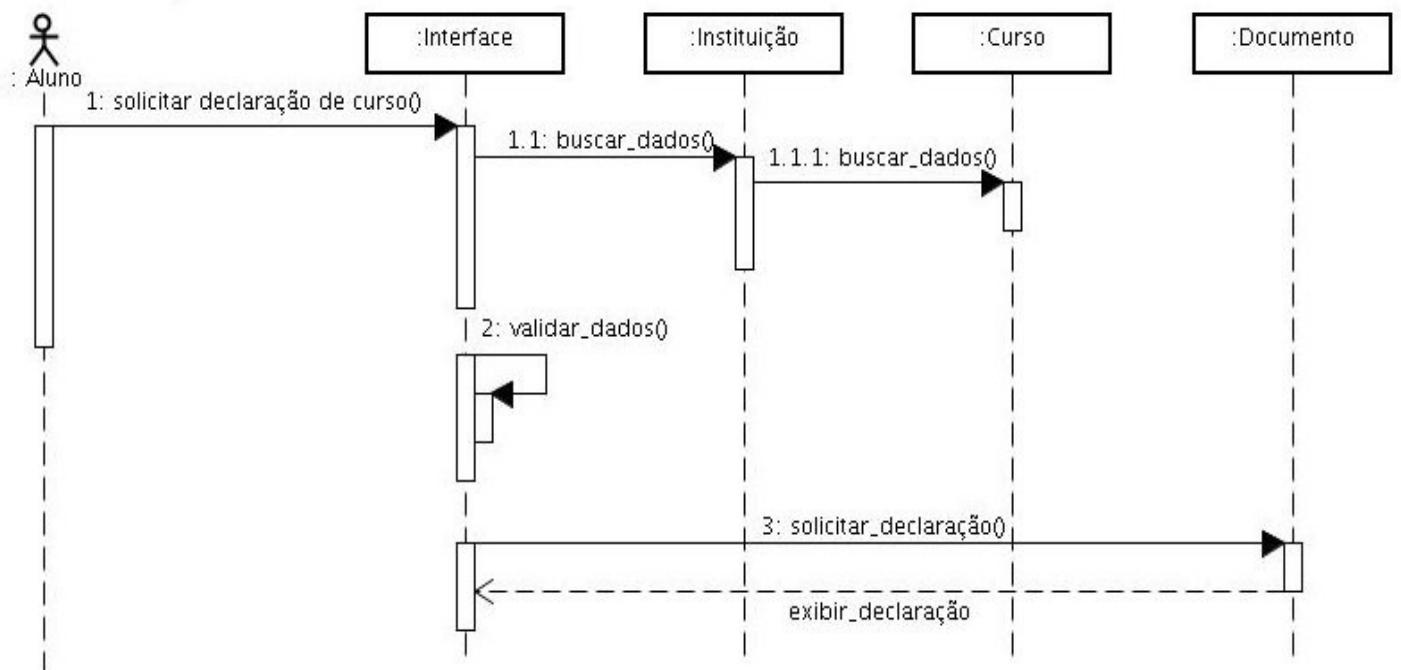


Figura 51 – Diagrama de Seqüência – Evento 32

2.6.33. Consultar Dados das Avaliações

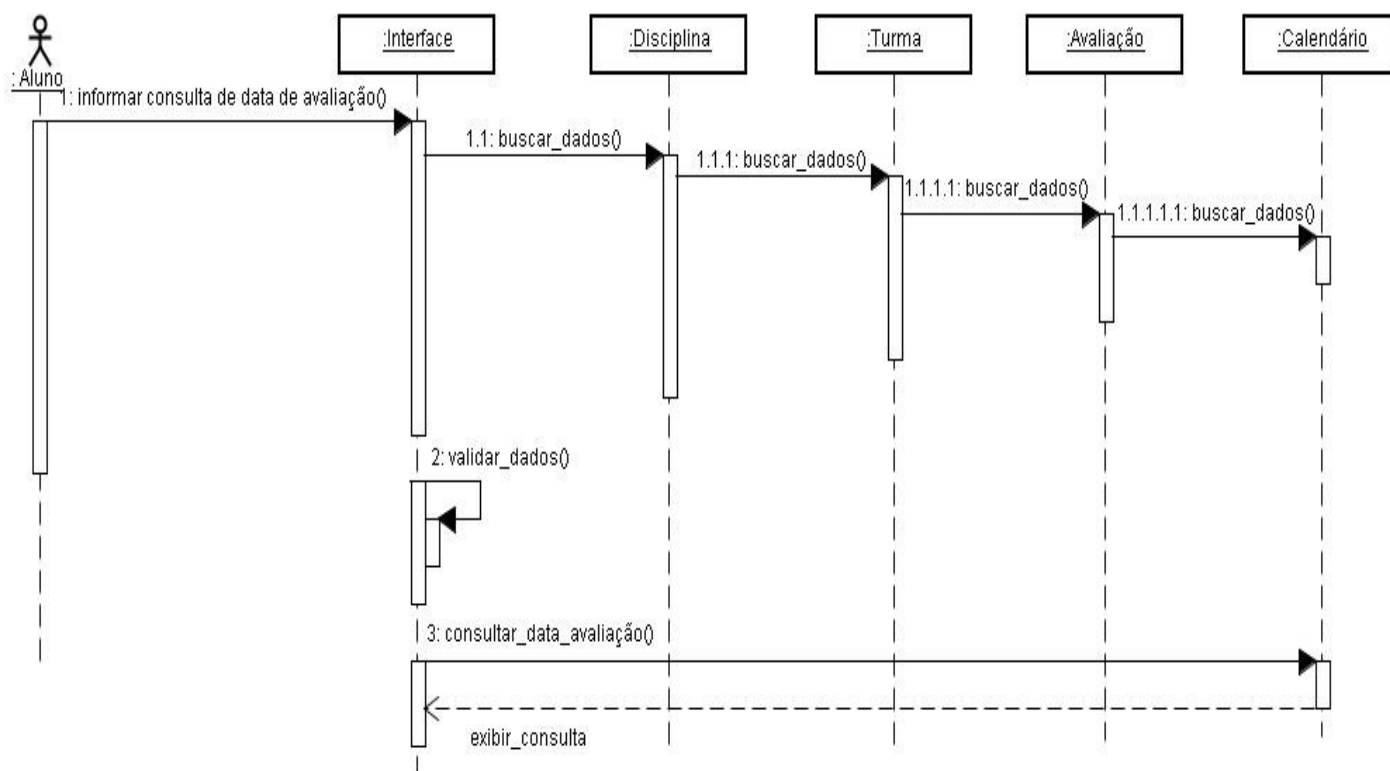


Figura 52 – Diagrama de Seqüência – Evento 33

2.6.34. Consultar Grade Curricular

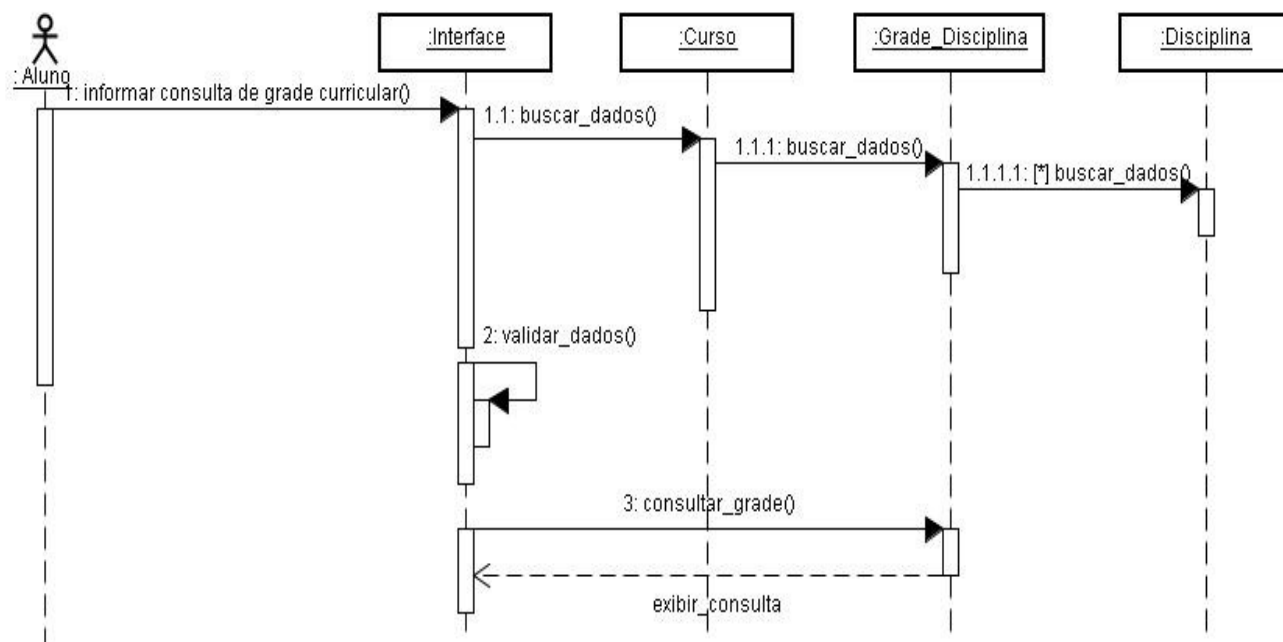


Figura 53 – Diagrama de Seqüência – Evento 34

2.6.35. Consultar Grade Semestral das Disciplinas

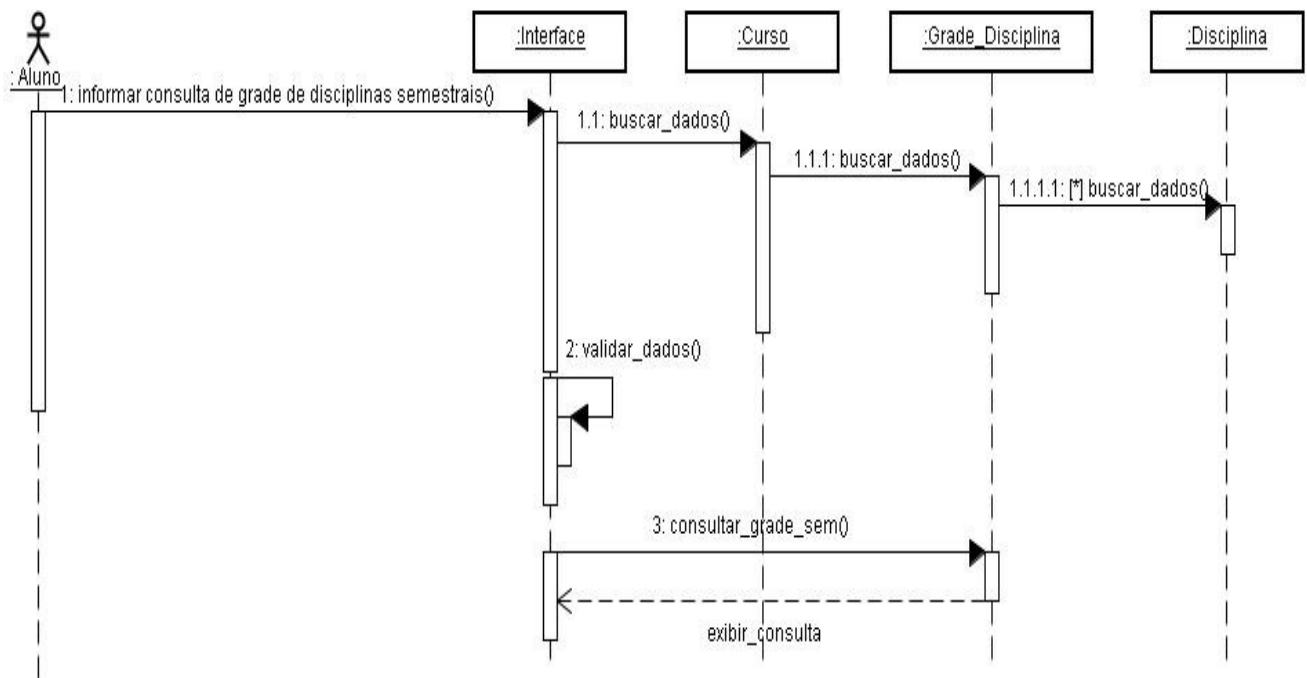


Figura 54 – Diagrama de Seqüência – Evento 35

2.6.36. Aluno Consulta Grade de Horários das Disciplinas

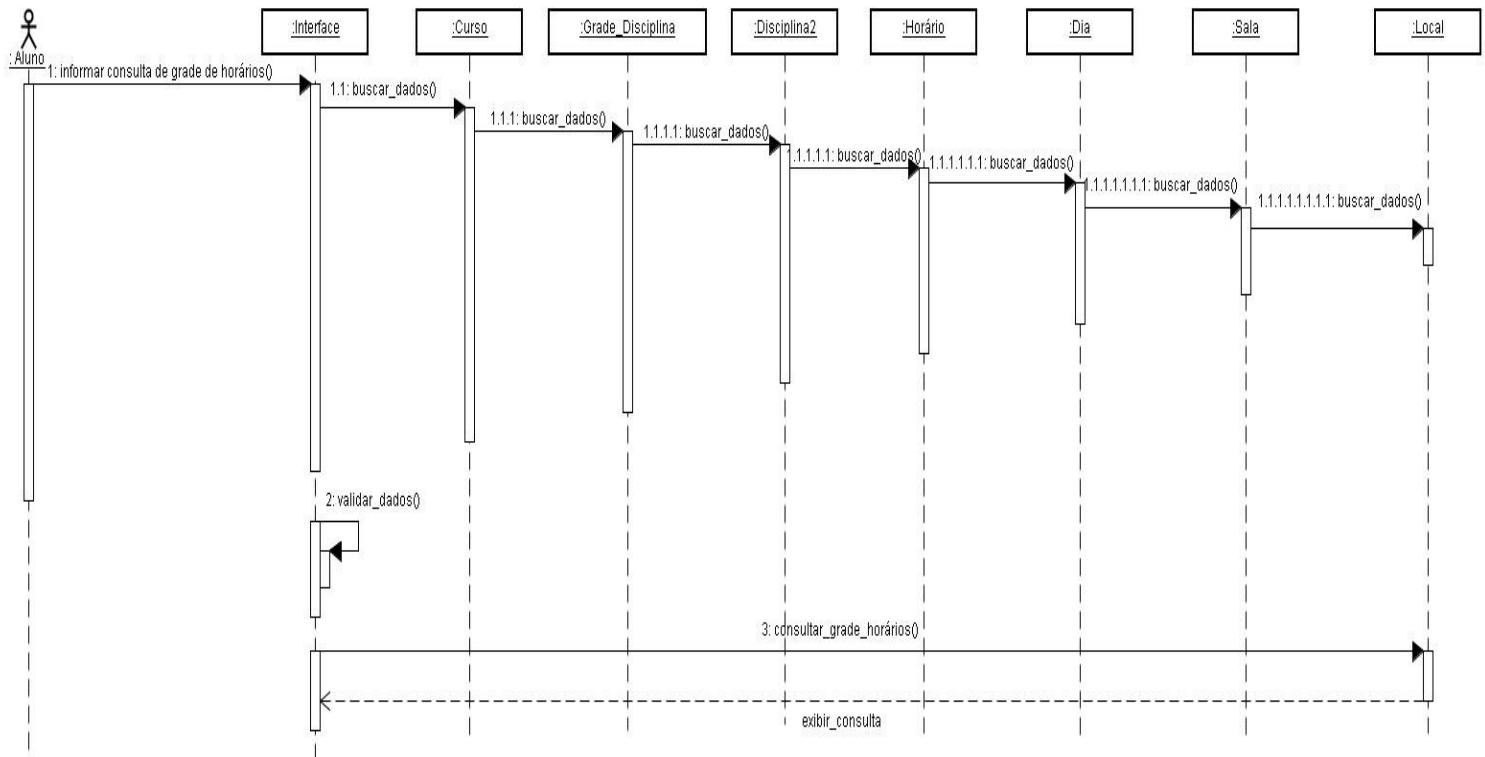


Figura 55 – Diagrama de Seqüência – Evento 36

2.6.37. Consultar Ementas Disciplinares

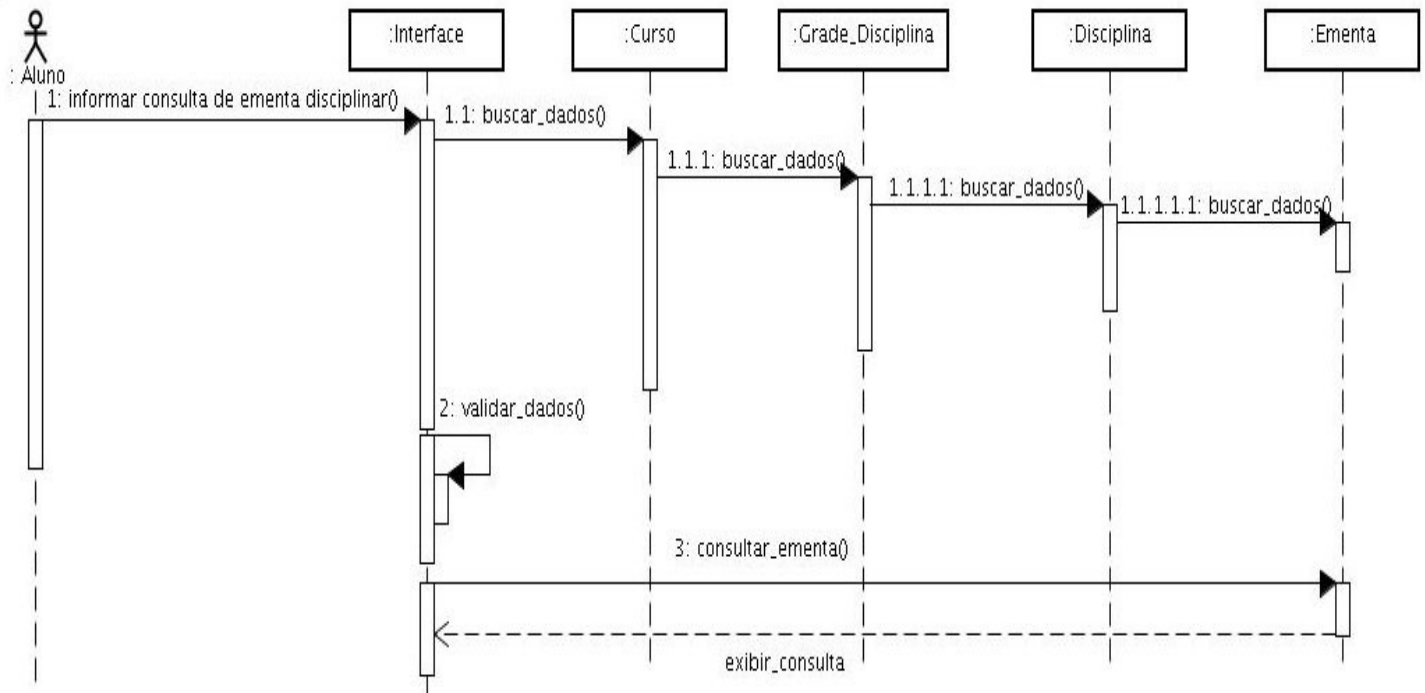


Figura 56 – Diagrama de Seqüência – Evento 37

2.6.38. Consultar Conteúdos Programáticos Ministrados

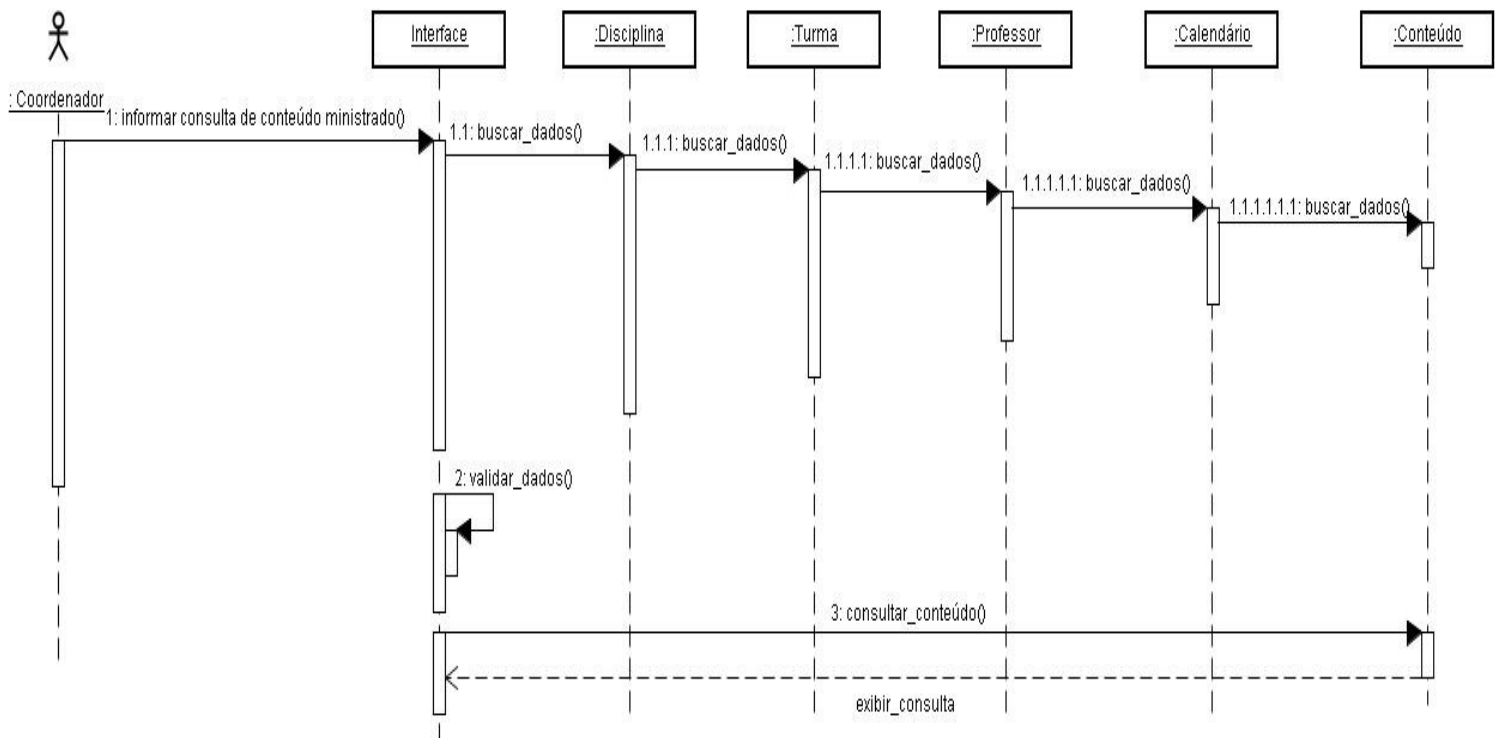


Figura 57 – Diagrama de Seqüência – Evento 38

2.6.39. Cadastrar Pré-Requisitos das Disciplinas

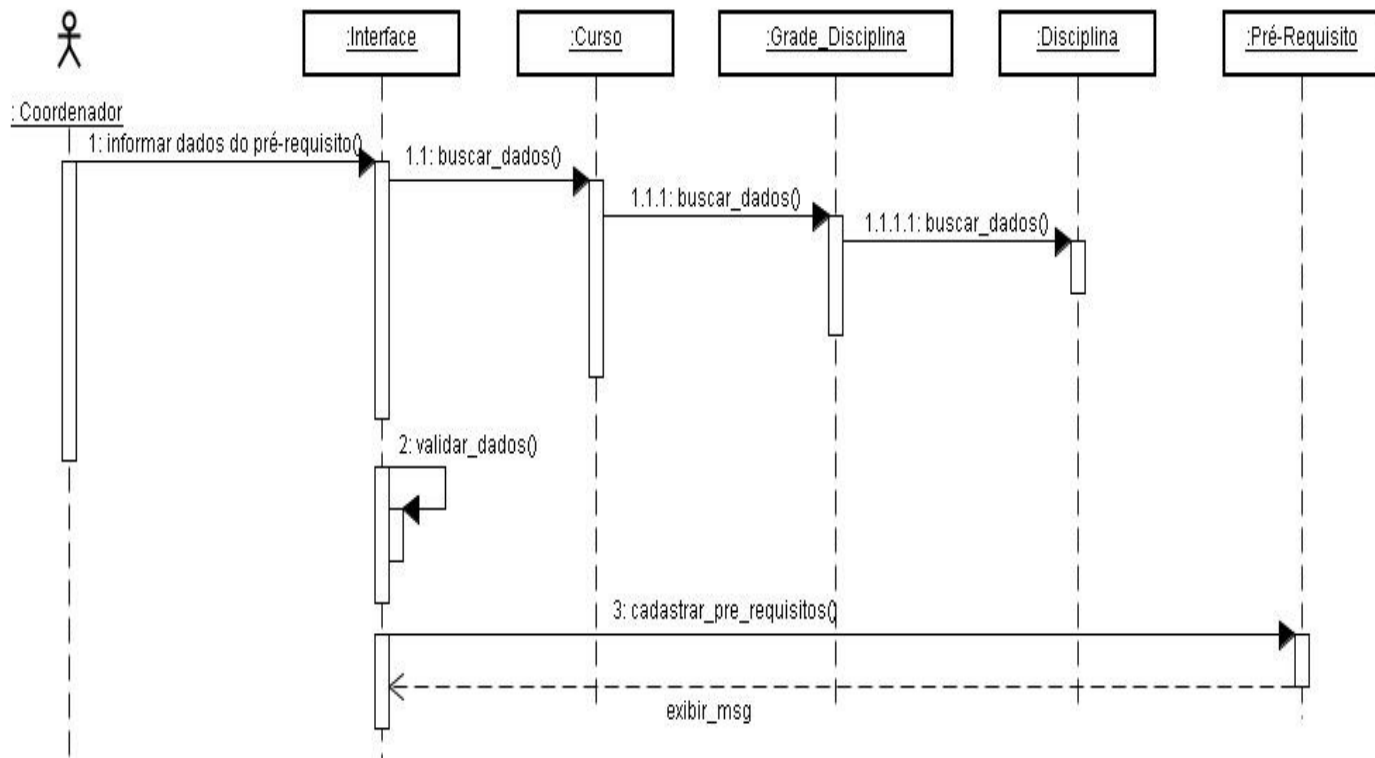


Figura 58 – Diagrama de Seqüência – Evento 39

2.6.40. Aprovar Pré-Inscrição

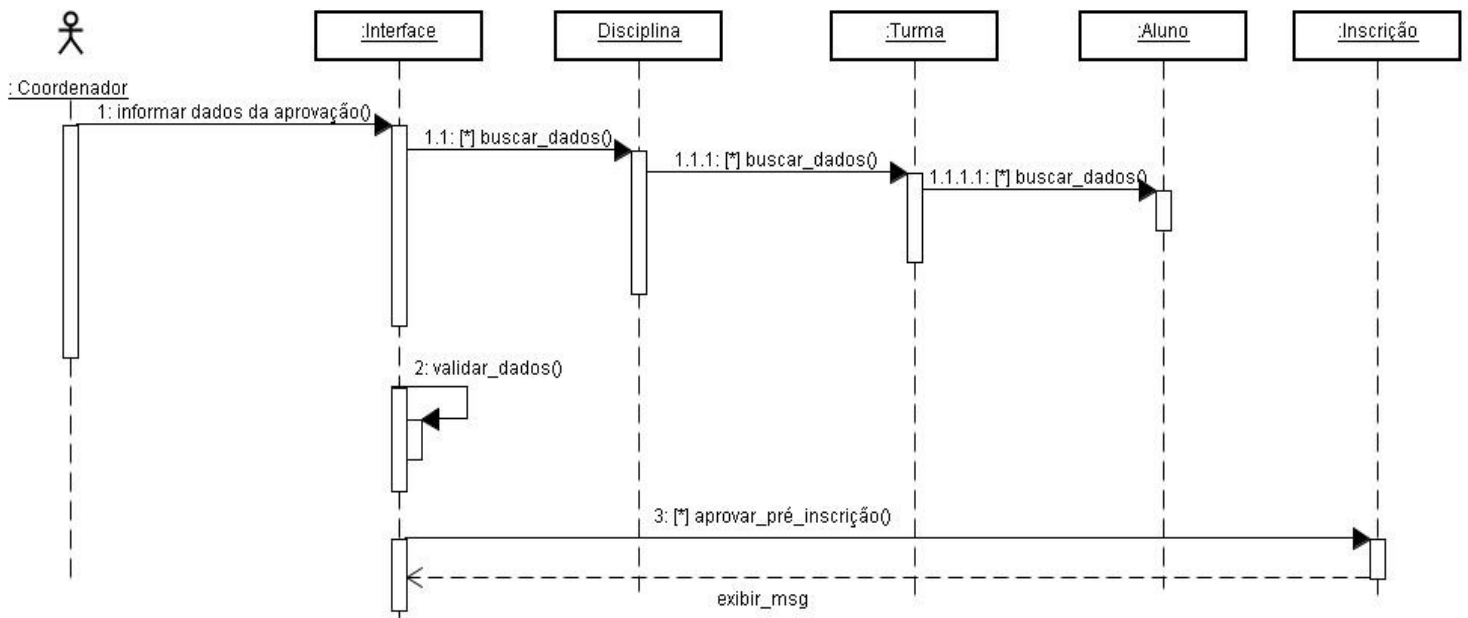


Figura 59 – Diagrama de Seqüência – Evento 40

2.6.41. Cadastrar Divulgação

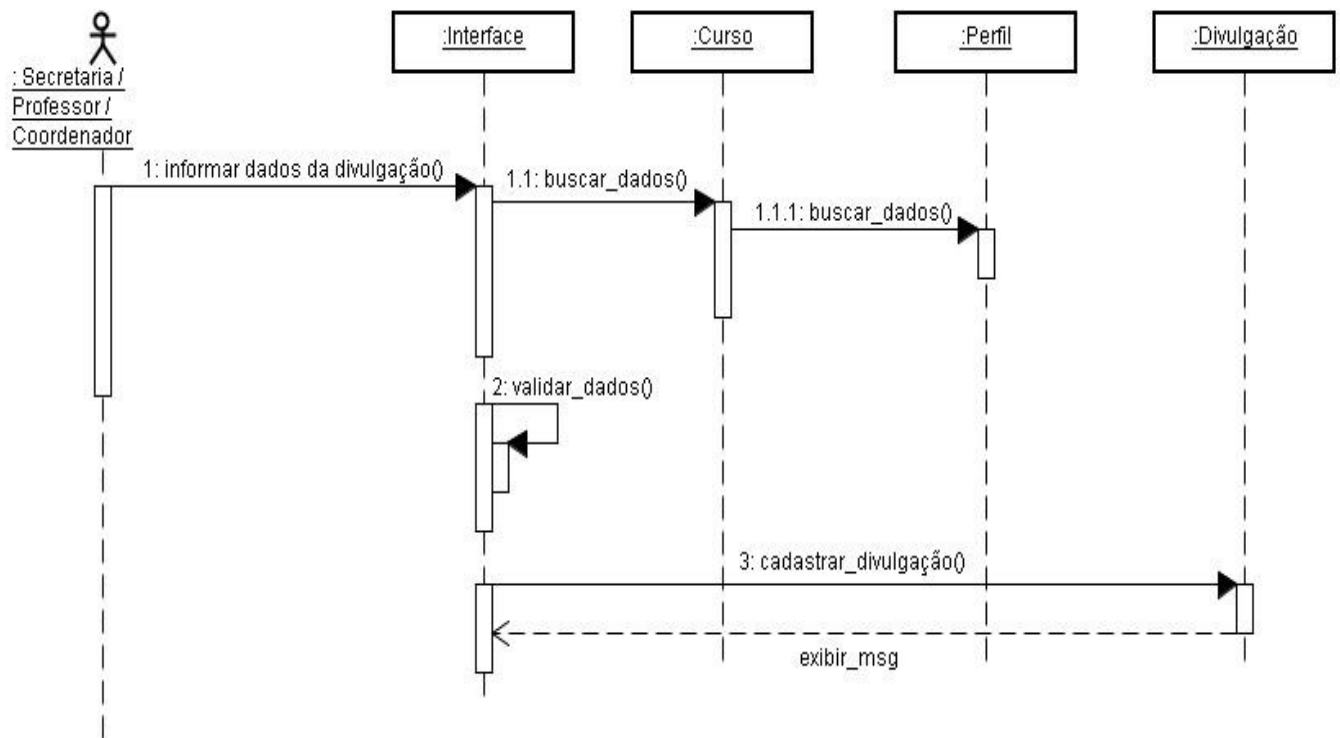


Figura 60 – Diagrama de Seqüência – Evento 41

2.7. Diagrama de Classe de Projeto

2.7.1. Gerenciamento de Informações Acadêmicas

***Figura 61** – Diagrama de Classe de Projeto – Gerenciamento de Informações Acadêmicas*

2.7.2. Mural de Estágios

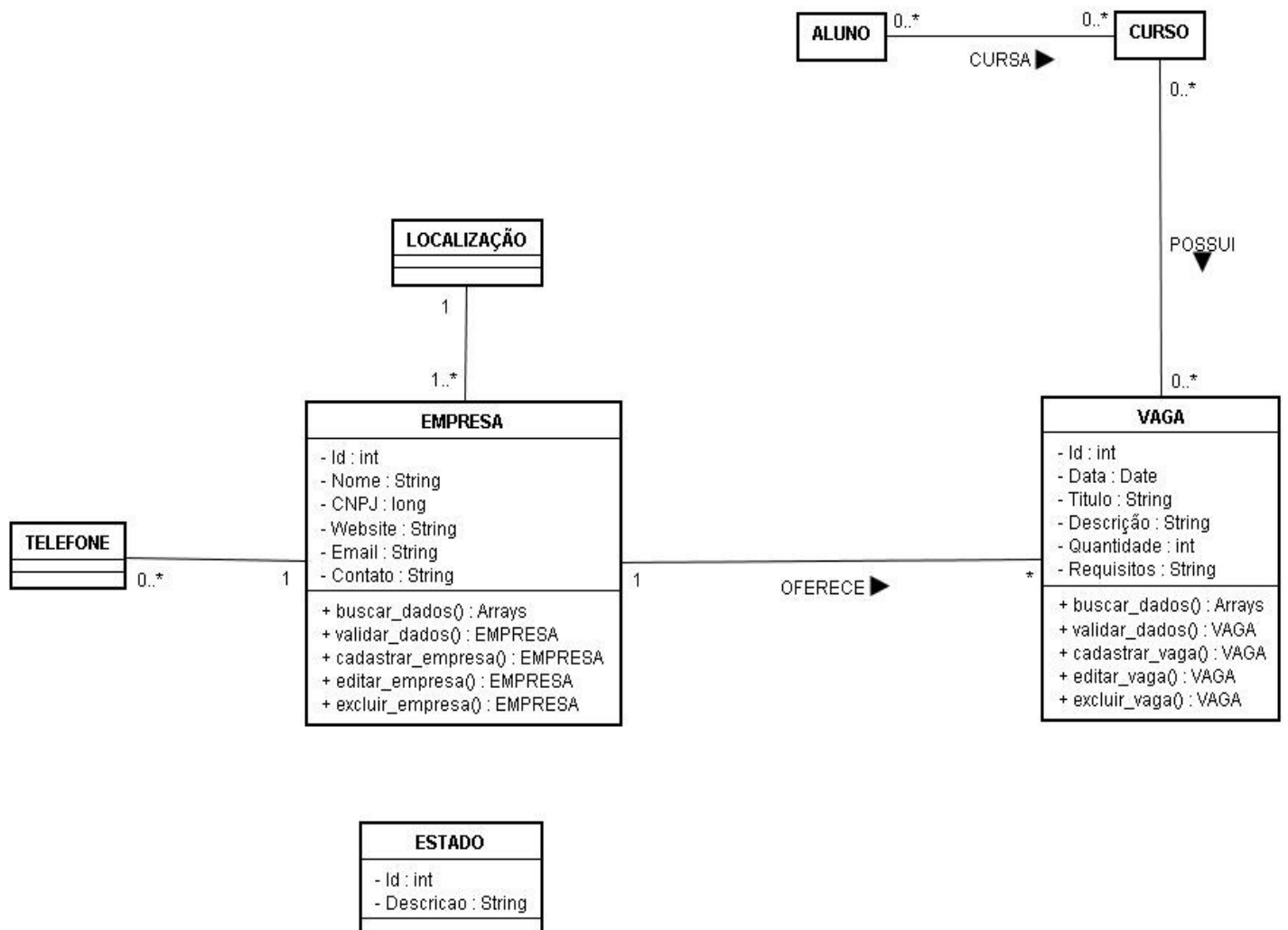


Figura 62 – Diagrama de Classe de Projeto – Mural de Estágios.

2.7.3. Consultas Parametrizadas – Avaliações

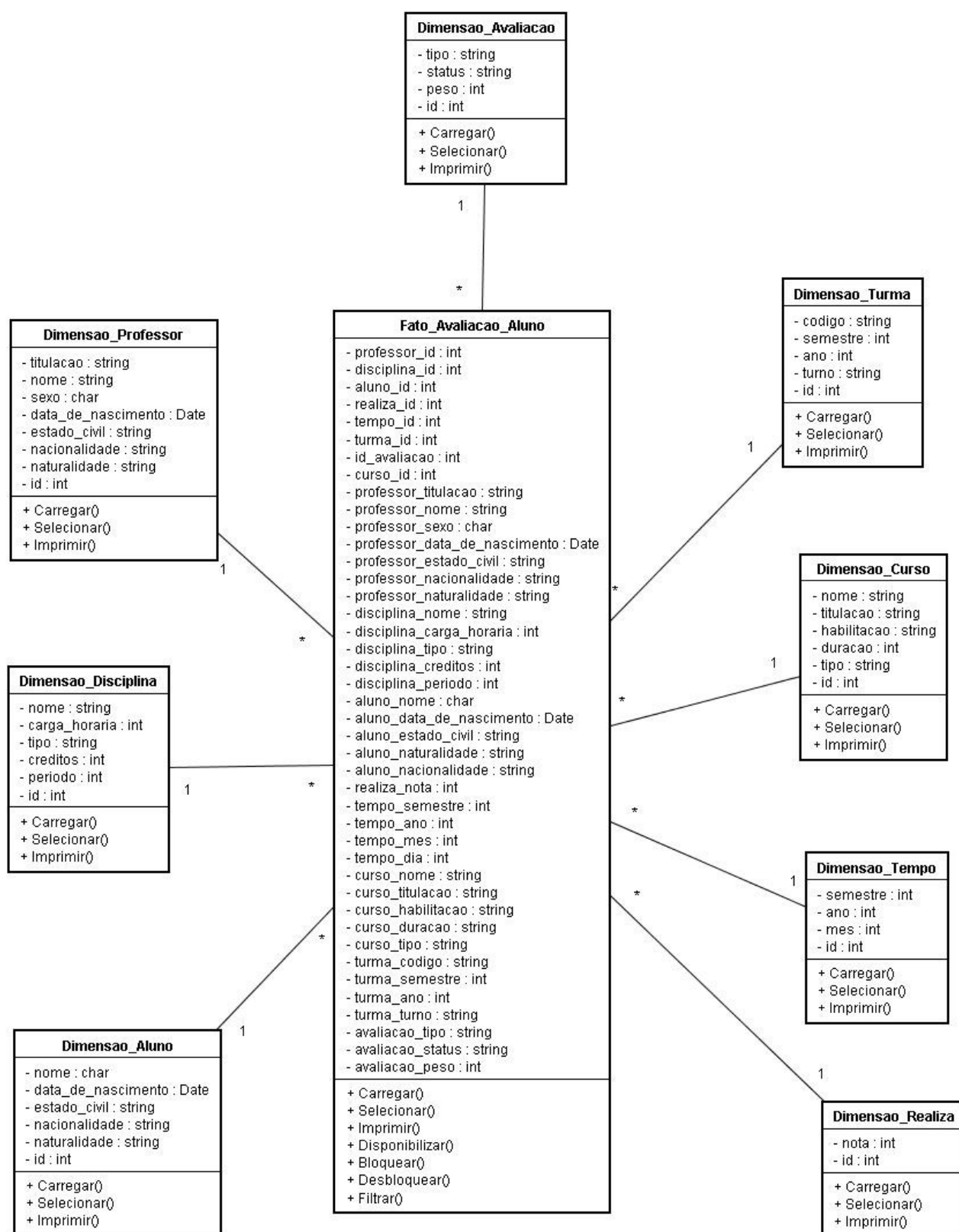


Figura 63 – Diagrama de Classe de Projeto – Consultas Parametrizadas das Avaliações.

2.7.4. Consultas Parametrizadas – Frequências

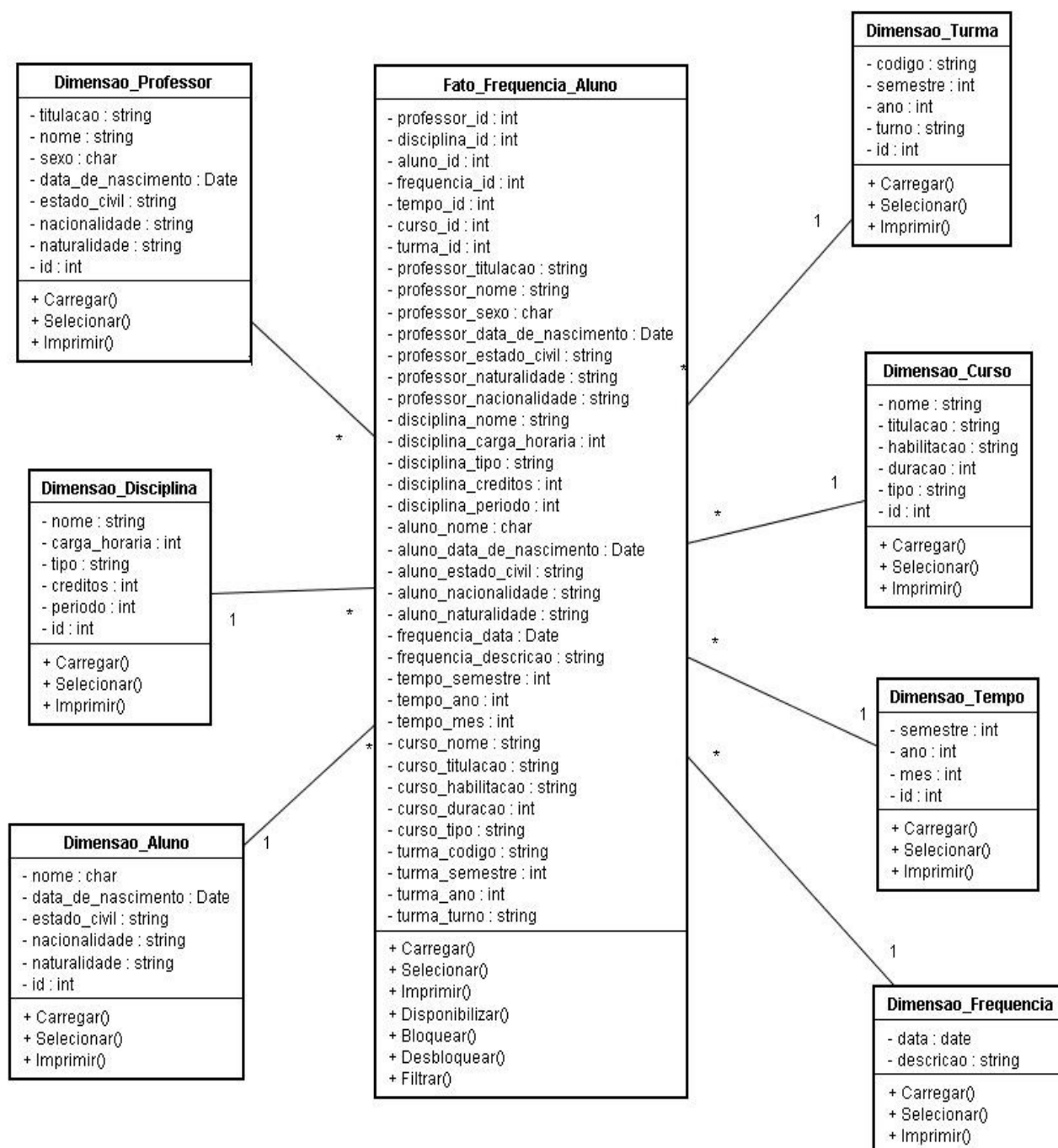


Figura 64 – Diagrama de Classe de Projeto – Consultas Parametrizadas das Frequências

2.8. Dicionário de Objetos

O dicionário de dados se relaciona com os objetos do nível de abstração físico, de implementação e do banco de dados. Abaixo seguem as descrições das tabelas que compõem o sistema.

2.8.1. Gerenciamento de Informações Acadêmicas

2.8.1.1. Users (Usuários)

Descrição: Classe que define os atributos e métodos do usuário.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
users_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
users_type	Campo que descreve o tipo do usuário	varchar2	20	Sim
login	Campo que descreve o código para entrar na aplicação	varchar2	50	Sim
password	Campo que descreve a senha para entrar na aplicação	varchar2	100	Sim
name	Campo que descreve o nome	varchar2	50	Sim
registration	Campo que descreve a matrícula	varchar2	20	Não
cpf	Campo que descreve o CPF	number	11	Sim
email	Campo que descreve o e-mail	varchar2	80	Não
sex	Campo que descreve sexo	char	1	Sim
date_of_birth	Campo que descreve a data de nascimento	date	-	Sim
father_name	Campo que descreve o nome do pai	varchar2	50	Não
mother_name	Campo que descreve nome da mãe	varchar2	50	Não
marital_status	Campo que descreve o estado Civil	varchar2	20	Sim
nationality	Campo que descreve a nacionalidade	varchar2	20	Sim
place_of_birth	Campo que descreve a naturalidade	varchar2	20	Sim
title	Campo que descreve a titulação	varchar2	50	Não
newsletter	Campo que descreve o recebimento de informativos	boolean	-	Não
job_status	Campo que descreve a situação empregatícia	varchar2	20	Não
workplace	Campo que descreve o local de trabalho	varchar2	20	Não
department	Campo que descreve o setor no qual está alocado	varchar2	50	Não

last_login	Campo que descreve o último acesso do usuário	date	-	Não
locations_id (FK)	Campo que descreve o identificador de uma localização	number	-	Sim
roles_id (FK)	Campo que descreve o identificador de um perfil	number	-	Sim
identities_id (FK)	Campo que descreve o identificador de uma identidade	number	=	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim

Tabela 14 – Dicionário de Objetos – Tabela Users – Atributos

Métodos

Nome	Descrição
create_users	Método responsável pela criação de um usuário.
edit_users	Método responsável pela edição dos dados de um usuário.
delete_users	Método responsável pela exclusão dos dados de um usuário.
validate_data	Método responsável em validar os dados de um usuário.
find_data	Método responsável em buscar os dados de um usuário.

Tabela 15 – Dicionário de Objetos – Tabela Users – Métodos

2.8.1.2. Roles (Regras)

Descrição: Classe que define o perfil de acesso dos usuários.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
roles_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
description	Campo que descreve o nome	varchar2	50	Sim

Tabela 16 – Dicionário de Objetos – Tabela Roles - Atributos

Métodos

Nome	Descrição
create_roles	Método responsável pela criação de um perfil.
edit_roles	Método responsável pela edição dos dados de um perfil.
delete_roles	Método responsável pela exclusão dos dados de um perfil.
validate_data	Método responsável em validar os dados de um perfil.

Tabela 17 – Dicionário de Objetos – Tabela Roles – Métodos

2.8.1.3. Permissions (Permissões)

Descrição: Classe que define as permissões de acesso dos usuários.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
permissions_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
permission	Campo que descreve as permissões	varchar2	50	Sim
permission_type	Campo que descreve o tipo de permissão	varchar2	50	Sim
action	Campo que descreve a ação	varchar2	50	Sim

Tabela 18 – Dicionário de Objetos – Tabela Permissions – Atributos

2.8.1.4. Documents (Documentos)

Descrição: Classe que define os atributos de um documento.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
documents_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
number	Campo que descreve o número da solicitação	number	-	Sim
type	Campo que descreve o tipo do documento	varchar2	50	Sim
title	Campo que descreve o título	varchar2	50	Sim
body	Campo que descreve o corpo do documento	text	-	Sim
model	Campo que descreve o modelo do documento	blob	-	Sim
users_id (FK)	Campo que descreve o identificador de um aluno	number	-	Sim

Tabela 19 – Dicionário de Objetos – Tabela Documents – Atributos

Métodos

Nome	Descrição
create_documents	Método responsável pela criação de um documento.
validate_data	Método responsável em validar os dados de um documento.
find_data	Método responsável em buscar os dados de um documento.

Tabela 20 – Dicionário de Objetos – Tabela Documents – Métodos

2.8.1.5. Identities (Identidade)

Descrição: Classe que define os atributos do RG (Registro Geral).

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
identities_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
number	Campo que descreve o número do documento	number	-	Sim
issue_date	Campo que descreve a data de emissão	date	-	Sim
department	Campo que descreve o órgão expedidor	varchar2	20	Sim
state	Campo que descreve a UF	char	2	Sim

Tabela 21 – Dicionário de Objetos – Tabela Identities – Atributos

2.8.1.6. Phones (Telefones)

Descrição: Classe que define os atributos de um contato telefônico.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
phones_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
type	Campo que descreve o tipo do telefone	varchar2	20	Sim
ddd	Campo que descreve o código de discagem a distância	number	2	Sim
number	Campo que descreve o número do telefone	number	8	Sim
note	Campo que descreve uma observação	varchar2	50	Não

Tabela 22 – Dicionário de Objetos – Tabela Phones – Atributos

2.8.1.7. Locations (Localização)

Descrição: Classe que define os atributos de um endereço.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
locations_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
country	Campo que descreve um país	varchar2	20	Sim
state	Campo que descreve uma UF	char	2	Sim
city	Campo que descreve uma cidade	varchar2	20	Sim
province	Campo que descreve um bairro	varchar2	20	Sim
street	Campo que descreve uma rua	varchar2	50	Sim
complement	Campo que descreve um complemento	varchar2	50	Não
zip_code	Campo que descreve um CEP	number	-	Sim

Tabela 23 – Dicionário de Objetos – Tabela Locations – Atributos

2.8.1.8. Publications (Divulgação)

Descrição: Classe que define os atributos de uma divulgação.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
publications_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
title	Campo que descreve um título	varchar2	20	Sim
type	Campo que descreve um tipo	varchar2	20	Sim
date	Campo que descreve uma data	date	-	Não
place	Campo que descreve um local	varchar2	20	Não
note	Campo que descreve uma observação	text	-	Não

Tabela 24 – Dicionário de Objetos – Tabela Publications – Atributos

2.8.1.9. Courses (Curso)

Descrição: Classe que define os atributos dos cursos.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
courses_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome	varchar2	50	Sim
title	Campo que descreve a titulação	varchar2	50	Sim
empowerment	Campo que descreve a habilitação	varchar2	50	Sim
duration	Campo que descreve a duração	number	2	Sim
type	Campo que descreve o tipo	varchar2	50	Sim
institutions_id (FK)	Campo que descreve o identificador de uma instituição	number	-	Sim

Tabela 25 – Dicionário de Objetos – Tabela Courses – Atributos

Métodos

Nome	Descrição
create_courses	Método responsável pela criação de um curso.
edit_courses	Método responsável pela edição dos dados de um curso.
delete_courses	Método responsável pela exclusão dos dados de um curso.
validate_data	Método responsável em validar os dados de um curso.
find_data	Método responsável em buscar os dados de um curso.

Tabela 26 – Dicionário de Objetos – Tabela Courses – Métodos

2.8.1.10. Disciplines_grids (Grade_Disciplina)

Descrição: Classe que define a grade das disciplinas.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
disciplines_grids_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome	varchar2	50	Sim
description	Campo que descreve uma descrição	varchar2	50	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
courses_id (FK)	Campo que descreve o identificador de um curso	number	-	Sim

Tabela 27 – Dicionário de Objetos – Tabela Discipline_Grids – Atributos

Métodos

Nome	Descrição
create_disciplines_grids	Método responsável pela criação de uma grade de disciplinas.
edit_disciplines_grids	Método responsável pela edição dos dados de uma grade de disciplinas.
find_disciplines_grids	Método responsável em buscar os dados de uma grade de disciplinas.
validate_data	Método responsável em validar os dados de uma grade de disciplinas.

Tabela 28 – Dicionário de Objetos – Tabela Discipline_Grids – Métodos

2.8.1.11. Disciplines (Disciplina)

Descrição: Classe que define os atributos das disciplinas.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
disciplines_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome	varchar2	50	Sim
working_hours	Campo que descreve a carga horária	number	-	Sim
type	Campo que descreve o tipo da disciplina	varchar2	20	Sim
credits	Campo que descreve a quantidade de créditos	number	-	Sim
period	Campo que descreve a que período pertence à disciplina	number	-	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
disciplines_grids_id (FK)	Campo que descreve o identificador de uma grade de disciplinas	number	-	Sim

Tabela 29 – Dicionário de Objetos – Tabela Disciplines - Atributos

Métodos

Nome	Descrição
create_disciplines	Método responsável pela criação de uma disciplina.
edit_disciplines	Método responsável pela edição dos dados de uma disciplina.
delete_disciplines	Método responsável pela exclusão dos dados de uma disciplina.
find_data	Método responsável em buscar os dados de uma disciplina.
validate_data	Método responsável em validar os dados de uma disciplina.

Tabela 30 – Dicionário de Objetos – Tabela Disciplines – Métodos

2.8.1.12. Pre-requisites (Pré-requisitos)

Descrição: Classe que define os atributos das ementas.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
disciplines_id (PK / FK)	Campo que descreve o identificador de uma disciplina	number	-	Sim
disciplines_sec_id (PK / FK)	Campo que descreve o identificador de uma disciplina	number	-	Sim

Tabela 31 – Dicionário de Objetos – Tabela Prerequisites – Atributos

Métodos

Nome	Descrição
create_prerequisites	Método responsável pela criação de um pré-requisito.
edit_prerequisites	Método responsável pela edição dos dados de um pré-requisito.
delete_prerequisites	Método responsável pela exclusão dos dados de um pré-requisito.

Tabela 32 – Dicionário de Objetos – Tabela Prerequisites – Métodos

2.8.1.13. Resume_Disciplines (Ementa)

Descrição: Classe que define os atributos das ementas.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
resume_of_disciplines_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome	varchar2	50	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
disciplines_id (FK)	Campo que descreve o identificador de uma disciplina	number	-	Sim

Tabela 33 – Dicionário de Objetos – Tabela Resume of Disciplines – Atributos

Métodos

Nome	Descrição
create_resume_of_disciplines	Método responsável pela criação de uma ementa.
edit_resume_of_disciplines	Método responsável pela edição dos dados de uma ementa.
delete_resume_of_disciplines	Método responsável pela exclusão dos dados de uma ementa.
find_resume_of_disciplines	Método responsável em buscar os dados de uma ementa.
validate_data	Método responsável em validar os dados de uma ementa.

Tabela 34 – Dicionário de Objetos – Tabela Resume of Disciplines – Métodos**2.8.1.14. Descriptions (Descrição)**

Descrição: Classe que define a descrição de uma ementa.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
descriptions_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome	varchar2	50	Sim
body	Campo que descreve o texto da descrição	text	-	Sim
resume_of_disciplines_id (FK)	Campo que descreve o identificador de uma ementa	number	-	Sim

Tabela 35 – Dicionário de Objetos – Tabela Descriptions – Atributos

2.8.1.15. Contents (Conteúdo)

Descrição: Classe que define os atributos do conteúdo de uma aula.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
contents_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
date	Campo que descreve a data da aula	date	-	Sim
body	Campo que armazena o texto da descrição do conteúdo	text	-	Sim
descriptions_id (FK)	Campo que descreve o identificador de uma descrição	number	-	Sim

Tabela 36 – Dicionário de Objetos – Tabela Contents – Atributos

Métodos

Nome	Descrição
create_contents	Método responsável pela criação de um conteúdo.
edit_contents	Método responsável pela edição dos dados de um conteúdo.
validate_data	Método responsável em validar os dados de uma frequência.
find_data	Método responsável em buscar os dados de uma frequência.

Tabela 37 – Dicionário de Objetos – Tabela Contents – Métodos

2.8.1.16. Exams (Avaliação)

Descrição: Classe que define as avaliações de uma turma.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
exams_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
type	Campo que descreve tipo de avaliação	varchar2	20	Sim
weighting	Campo que descreve o peso	number	-	Não
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
groups_id (FK)	Campo que descreve o identificador de uma turma	number	-	Sim
calendars_id (FK)	Campo que descreve o identificador de um calendário	number	-	Sim

Tabela 38 – Dicionário de Objetos – Tabela Exams – Atributos

Métodos

Nome	Descrição
create_exams	Método responsável pela criação de uma avaliação.
edit_exams	Método responsável pela edição dos dados de uma avaliação.
delete_exams	Método responsável pela exclusão dos dados de uma avaliação.
find_exams	Método responsável em buscar os dados de uma avaliação.
validate_data	Método responsável em validar os dados de uma avaliação.

Tabela 39 – Dicionário de Objetos – Tabela Exams – Métodos

2.8.1.17. Users_Exams (Nota)

Descrição: Classe que define a realização de uma avaliação.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
users_exams_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
grade	Campo que descreve a nota de uma avaliação	number	-	Sim
users_id (PK / FK)	Campo que descreve o identificador de um aluno	number	-	Sim
exams_id (PK / FK)	Campo que descreve o identificador de uma avaliação	number	-	Sim

Tabela 40 – Dicionário de Objetos – Tabela User_Exams – Atributos

Métodos

Nome	Descrição
create_users_exams	Método responsável pela criação de uma nota de uma avaliação.
edit_users_exams	Método responsável pela edição dos dados de uma avaliação.
delete_users_exams	Método responsável pela exclusão dos dados de uma nota de uma avaliação.
find_users_exams	Método responsável em buscar os dados de uma nota de uma avaliação.
validate_data	Método responsável em validar os dados de uma nota de uma avaliação.

Tabela 41 – Dicionário de Objetos – Tabela User_Exams – Métodos

2.8.1.18. Groups (Turma)

Descrição: Classe que define os atributos das turmas.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
groups_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
code	Campo que descreve o código	varchar2	20	Sim
semester	Campo que descreve o semestre	number	1	Sim
year	Campo que descreve o ano	number	4	Sim
shift	Campo que descreve o turno	char	1	Sim
users_id (FK)	Campo que descreve o identificador de um professor	number	-	Sim
diciplines_id (FK)	Campo que descreve o identificador de uma disciplina	number	-	Sim
time_grids_times_id (FK)	Campo que descreve o identificador de um horário	number	-	Sim
time_grids_days_id (FK)	Campo que descreve o identificador de um dia da semana	number	-	Sim

Tabela 42 – Dicionário de Objetos – Tabela Groups – Atributos

Métodos

Nome	Descrição
create_groups	Método responsável pela criação de uma turma.
edit_groups	Método responsável pela edição dos dados de uma turma.
find_data	Método responsável em buscar os dados de uma turma.
validate_data	Método responsável em validar os dados de uma turma.

Tabela 43 – Dicionário de Objetos – Tabela Groups – Métodos

2.8.1.19. Calendars (Calendário)

Descrição: Classe que define o calendário.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
calendars_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
begin_date	Campo que descreve a data de início	date	-	Sim
end_date	Campo que descreve a data de término	date	-	Sim
label	Campo que descreve o título	varchar2	50	Não
description	Campo que descreve o texto da descrição	text	-	Não

Tabela 44 – Dicionário de Objetos – Tabela Calendars – Atributos

2.8.1.20. Allocations (Alocação)

Descrição: Classe que define os atributos de uma alocação de recurso.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
allocations_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
note	Campo que descreve uma observação	varchar2	50	Não
users_id (PK/FK)	Campo que descreve o identificador de um professor	number	-	Sim
resources_id (PK/FK)	Campo que descreve o identificador de um recurso	number	-	Sim
calendars_id (FK)	Campo que descreve o identificador de um calendário	number	-	Sim

Tabela 45 – Dicionário de Objetos – Tabela Allocations – Atributos

Métodos

Nome	Descrição
create_allocations	Método responsável pela criação de uma alocação.
edit_allocations	Método responsável pela edição dos dados de uma alocação.
find_data	Método responsável em buscar os dados de uma alocação.
validate_data	Método responsável em validar os dados de uma alocação.

Tabela 46 – Dicionário de Objetos – Tabela Allocations – Métodos**2.8.1.21. Resources (Recursos)**

Descrição: Classe que define os atributos dos recursos.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
resources_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
type	Campo que descreve o tipo do recurso	varchar2	50	Sim
quantity	Campo que descreve a quantidade	number	-	Sim

Tabela 47 – Dicionário de Objetos – Tabela Resources – Atributos**Métodos**

Nome	Descrição
create_resources	Método responsável pela criação de uma alocação.
edit_resources	Método responsável pela edição dos dados de uma alocação.
delete_resources	Método responsável pela exclusão dos dados de uma nota de uma alocação.
find_data	Método responsável em buscar os dados de uma alocação.
validate_data	Método responsável em validar os dados de uma alocação.

Tabela 48 – Dicionário de Objetos – Tabela Resources – Métodos

2.8.1.22. Registrations (Inscrição)

Descrição: Classe que define a inscrição em disciplinas.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
registration_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
date	Campo que descreve a data da inscrição	date	date	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
users_id (PK/FK)	Campo que descreve o identificador de um aluno	number	-	Sim
groups_id (PK/FK)	Campo que descreve o identificador de uma turma	number	-	Sim

Tabela 49 – Dicionário de Objetos – Tabela Registrations – Atributos

Métodos

Nome	Descrição
create_registration	Método responsável pela criação de uma inscrição.
edit_registration	Método responsável pela edição dos dados de uma inscrição.
validate_data	Método responsável em validar os dados de uma inscrição.

Tabela 50 – Dicionário de Objetos – Tabela Registrations – Métodos

2.8.1.23. Frequencies (Frequência)

Descrição: Classe que define a frequência de um aluno.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
frequencies_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
date	Campo que descreve a data da aula	date	-	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
users_id (PK/FK)	Campo que descreve o identificador de um aluno	number	-	Sim
groups_id (PK/FK)	Campo que descreve o identificador de uma turma	number	-	Sim

Tabela 51 – Dicionário de Objetos – Tabela Frequencies – Atributos

Métodos

Nome	Descrição
create_frequencies	Método responsável pela lançamento de uma frequência.
edit_frequencies	Método responsável pela edição dos dados de uma frequência..
validate_data	Método responsável em validar os dados de uma frequência.
find_data	Método responsável em buscar os dados de uma frequência.

Tabela 52 – Dicionário de Objetos – Tabela Frequencies – Métodos

2.8.1.24. Releases (Lançamento)

Descrição: Classe que define os atributos do lançamento de uma aula.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
releases_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
date	Campo que descreve a data da aula	date	-	Sim
comments	Campo que armazena o texto dos comentários do professor	text	-	Sim
contents_id (PK/FK)	Campo que descreve o identificador de um conteúdo	number	-	Sim
groups_id (PK/FK)	Campo que descreve o identificador de uma turma	number	-	Sim

Tabela 53 – Dicionário de Objetos – Tabela Releases – Atributos

Métodos

Nome	Descrição
create_releases	Método responsável pela criação de um lançamento.
edit_releases	Método responsável pela edição dos dados de um lançamento.
delete_releases	Método responsável pela exclusão dos dados de um lançamento.
validate_data	Método responsável em validar os dados de um lançamento.

Tabela 54 – Dicionário de Objetos – Tabela Releases – Métodos

2.8.1.25. Files (Arquivo)

Descrição: Classe que define os atributos de um arquivo.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
files_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome do arquivo	varchar2	50	Sim
description	Campo que armazena a descrição	varchar2	50	Não
type	Campo que descreve o tipo do arquivo	char	4	Sim
date	Campo que descreve a data	date	-	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
groups_id (FK)	Campo que descreve o identificador de uma turma	number	-	Sim

Tabela 55 – Dicionário de Objetos – Tabela Files – Atributos

Métodos

Nome	Descrição
create_files	Método responsável pela criação de um arquivo.
edit_files	Método responsável pela edição dos dados de um arquivo.
delete_files	Método responsável pela exclusão dos dados de um arquivo.
find_data	Método responsável em buscar os dados de um arquivo.
validate_data	Método responsável em validar os dados de um arquivo.

Tabela 56 – Dicionário de Objetos – Tabela Files – Métodos

2.8.1.26. Institutions (Instituição)

Descrição: Classe que define os atributos de uma instituição de ensino.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
companies_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome	varchar2	50	Sim
cnpj	Campo que descreve o CNPJ	number	15	Sim
website	Campo que descreve a homepage da instituição	varchar2	50	Não
email	Campo que descreve o e-mail	varchar2	50	Não
contact	Campo que descreve o contato da instituição	varchar2	50	Não
locations_id (FK)	Campo que descreve o identificador de uma localização	number	-	Sim
phones_id (FK)	Campo que descreve o identificador de um telefone	number	-	Sim

Tabela 57 – Dicionário de Objetos – Tabela Institutions – Atributos

Métodos

Nome	Descrição
create_institutions	Método responsável pela criação de uma instituição.
edit_institutions	Método responsável pela edição dos dados de uma instituição.
delete_institutions	Método responsável pela exclusão dos dados de uma instituição.
validate_data	Método responsável em validar os dados de uma instituição.

Tabela 58 – Dicionário de Objetos – Tabela Institutions – Métodos

2.8.1.27. Time_Grids (Grade_Horário)

Descrição: Classe que define os atributos do horário de uma turma.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
times_id (PK/FK)	Campo que descreve o identificador de um horário	number	-	Sim
days_id (PK/FK)	Campo que descreve o identificador de um dia	number	-	Sim
rooms_id (FK)	Campo que descreve o identificador de uma sala	number	-	Sim

Tabela 59 – Dicionário de Objetos – Tabela Time_Grids – Atributos

2.8.1.28. Times (Horário)

Descrição: Classe que define os atributos de um horário.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
times_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
time	Campo que descreve o número	varchar2	20	Sim

Tabela 60 – Dicionário de Objetos – Tabela Times – Atributos

2.8.1.29. Days (Dia)

Descrição: Classe que define os atributos de um dia.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
days_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
day	Campo que descreve o dia	varchar2	20	Sim

Tabela 61 – Dicionário de Objetos – Tabela Days – Atributos

2.8.1.30. Rooms (Sala)

Descrição: Classe que define os atributos de uma sala de aula.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
rooms_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
number	Campo que descreve o número	varchar2	20	Sim
size	Campo que descreve a capacidade	number	-	Sim
places_id (FK)	Campo que descreve o identificador de um local	number	-	Sim

Tabela 62 – Dicionário de Objetos – Tabela Rooms – Atributos

Métodos

Nome	Descrição
create_rooms	Método responsável pela criação de uma sala.
edit_rooms	Método responsável pela edição dos dados de uma sala.
validate_data	Método responsável em validar os dados de uma sala.
find_data	Método responsável em buscar os dados de uma sala.

Tabela 63 – Dicionário de Objetos – Tabela Rooms – Métodos

2.8.1.31. Places (Lugar)

Descrição: Classe que define os atributos de um local.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
places_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
type	Campo que descreve o tipo	varchar2	50	Sim
description	Campo que representa a descrição	varchar2	50	Não
hierarchy	Campo que descreve a hierarquia dos locais	number	-	Sim

Tabela 64 – Dicionário de Objetos – Tabela Places – Atributos

Métodos

Nome	Descrição
create_places	Método responsável pela criação de um local.
edit_places	Método responsável pela edição dos dados de um local.
validate_data	Método responsável em validar os dados de um local.
find_data	Método responsável em buscar os dados de um local.

Tabela 65 – Dicionário de Objetos – Tabela Places – Métodos

2.8.2. Mural de Estágios

2.8.2.1. Places_Institutions (Empresa)

Descrição: Classe que define os atributos de uma empresa.

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
institutions_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
name	Campo que descreve o nome	varchar2	50	Sim
cnpj	Campo que descreve o CNPJ	number	15	Sim
website	Campo que descreve a homepage da empresa	varchar2	50	Não
email	Campo que descreve o e-mail	varchar2	50	Não
contact	Campo que descreve o contato	varchar2	50	Não
locations_id (FK)	Campo que descreve o identificador da localização	number	-	Sim
phones_id (FK)	Campo que descreve o identificador do telefone	number	-	Sim

Tabela 66 – Dicionário de Objetos – Tabela Places_Institutions – Atributos

Métodos

Nome	Descrição
create_institutions	Método responsável pela criação de uma empresa.
edit_institutions	Método responsável pela edição dos dados de uma empresa.
delete_institutions	Método responsável pela exclusão dos dados de uma empresa.
find_data	Método responsável em buscar os dados de uma empresa.
validate_data	Método responsável em validar os dados de uma empresa.

Tabela 67 – Dicionário de Objetos – Tabela Places_Institutions – Métodos

2.8.2.2) Trainees (Vaga)

Descrição: Classe que define os atributos de uma vaga de estágio

Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
trainees_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
date	Campo que descreve a data	date	-	Sim
title	Campo que descreve o título	varchar2	50	Sim
description	Campo que armazena a descrição	text	-	Não
quantity	Campo que armazena a quantidade de vagas	number	-	Sim
states_id (FK)	Campo que descreve o identificador de um estado	number	-	Sim
companies_id (FK)	Campo que descreve o identificador da empresa	number	-	Sim

Tabela 68 – Dicionário de Objetos – Tabela Trainees – Atributos

Métodos

Nome	Descrição
create_trainees	Método responsável pela criação de uma vaga de estágio.
edit_trainees	Método responsável pela edição dos dados de uma vaga de estágio.
delete_trainees	Método responsável pela exclusão dos dados de uma vaga de estágio.
find_data	Método responsável em buscar os dados de uma vaga de estágio.
validate_data	Método responsável em validar os dados de uma vaga de estágio.

Tabela 69 – Dicionário de Objetos – Tabela Trainees – Métodos

2.8.3. Plugins

A aplicação utiliza alguns *plugins*, os quais geram algumas tabelas que não estão previstas no Diagrama Físico do sistema.

Abaixo segue uma descrição sobre cada um desses *plugins* e as tabelas geradas pelos mesmos.

2.8.3.1) Trainees_Act_as_Permissible

Para realizar as permissões do sistema foi utilizado o *plugin acts_as_permisible*¹, que gera as tabelas *roles* e *roles_memberships*. Este *plugin* permite realizar qualquer tipo de permissão para qualquer tabela do sistema, entretanto, para este sistema utilizou-se somente para as permissões dos usuários.

Tabela: Roles_Memberships - Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
role_memberships_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
roleable_id	Campo que descreve o identificador de um campo polimórfico	number	-	Sim
roleable_type	Campo que descreve um campo polimórfico (nome da tabela)	varchar2	50	Sim
roles_id (FK)	Campo que descreve o identificador de um perfil	number	-	Sim

Tabela 70 – Dicionário de Objetos – Tabela Roles_Memberships – Atributos

A tabela *roles* já foi descrita no início deste documento.

¹Disponível em: < http://github.com/NoamB/acts_as_permisible/tree/master >

2.8.3.2. Trainees_Act_as_Audited

Para realizar a auditoria do sistema foi utilizado o *plugin acts_as_audited*², que gera a tabela *audits*. Para que o *plugin* realize a auditoria, as tabelas desejadas devem ser indicadas.

Tabela: Audits - Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
audits_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
auditable_id	Campo que descreve o identificador de um campo polimórfico	number	-	Sim
auditable_type	Campo que descreve um campo polimórfico (nome da tabela)	varchar2	50	Sim
user_id	Campo que descreve o identificador de um campo polimórfico	number	-	Sim
user_type	Campo que descreve um campo polimórfico (nome da tabela)	varchar2	50	Sim
username	Campo que descreve o login do usuário	varchar2	50	Não
action	Campo que descreve a ação realizada	varchar2	50	Sim
changes	Campo que descreve as mudanças realizadas	text	-	Sim
version	Campo que descreve a versão	number	-	Não
created_at	Campo que descreve a data da modificação	datetime	-	Sim

Tabela 71 – Dicionário de Objetos – Tabela Audits – Atributos

² Disponível em: < http://github.com/collectiveidea/acts_as_audited/tree/master >

2.8.3.3. Railstat

Para realizar algumas estatísticas da aplicação foi utilizado o *plugin railstat*³, que gera as tabelas *rail_stats*, *search_terms* e *iptocs*. Este *plugin* realiza um conjunto de estatísticas de acesso a aplicação, tais como: páginas mais navegadas, *browsers* e sistemas operacionais que mais acessam o sistema, dentre outras estatísticas.

Tabela: Railstat - Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
railstat_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
remote_ip	Campo que descreve o IP	varchar2	50	Não
country	Campo que descreve o país	varchar2	50	Não
language	Campo que descreve a língua	varchar2	50	Não
domain	Campo que descreve o domínio	varchar2	50	Não
subdomain	Campo que descreve o sub-domínio	varchar2	50	Não
referer	Campo que descreve a referência	varchar2	50	Não
resource	Campo que descreve o recurso	varchar2	50	Não
user_agent	Campo que descreve o usuário agente	varchar2	50	Não
platform	Campo que descreve a plataforma	varchar2	50	Não
browser	Campo que descreve o <i>browser</i>	varchar2	50	Não
version	Campo que descreve a versão	varchar2	50	Não
created_at	Campo que descreve a data e hora da criação	datetime	-	Não
created_on	Campo que descreve a data da criação	date	-	Não
screen_size	Campo que descreve a resolução da tela	varchar2	50	Não
colors	Campo que descreve as cores	varchar2	50	Não
java	Campo que descreve a utilização de Java	varchar2	50	Não
java_enabled	Campo que descreve a ativação de Java	varchar2	50	Não
flash	Campo que descreve a utilização do Flash	varchar2	50	Não

Tabela 72 – Dicionário de Objetos – Tabela Railstat – Atributos

³ Disponível em: <<http://www.railstat.com>>

Tabela: Search_Terms - Atributos

Nome	Descrição	Tipo	Tamanho	Obrigatório
search_terms_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
subdomain	Campo que descreve o sub-domínio	varchar2	50	Não
searchterms	Campo que descreve a procura do termo	varchar2	50	Não
count	Campo que descreve um contador	number	-	Não
domain	Campo que descreve o domínio	varchar2	50	Não

Tabela 73 – Dicionário de Objetos – Tabela Search_Terms – Atributos**Tabela: Ip_Tocs - Atributos**

Nome	Descrição	Tipo	Tamanho	Obrigatório
ip_tocs_id (PK)	Campo que descreve o atributo de identificação único	number	-	Sim
ip_from	Campo que descreve o ip de origem	number	-	Não
ip_to	Campo que descreve o ip de destino	number	-	Não
country_code2	Campo que descreve o código do país de origem	varchar2	50	Não
country_code3	Campo que descreve o código do país de destino	varchar2	50	Não
country_name	Campo que descreve o nome do país	varchar2	50	Não

Tabela 74 – Dicionário de Objetos – Tabela Ip_Tocs – Atributos

3.1.2. Mural de Estágio

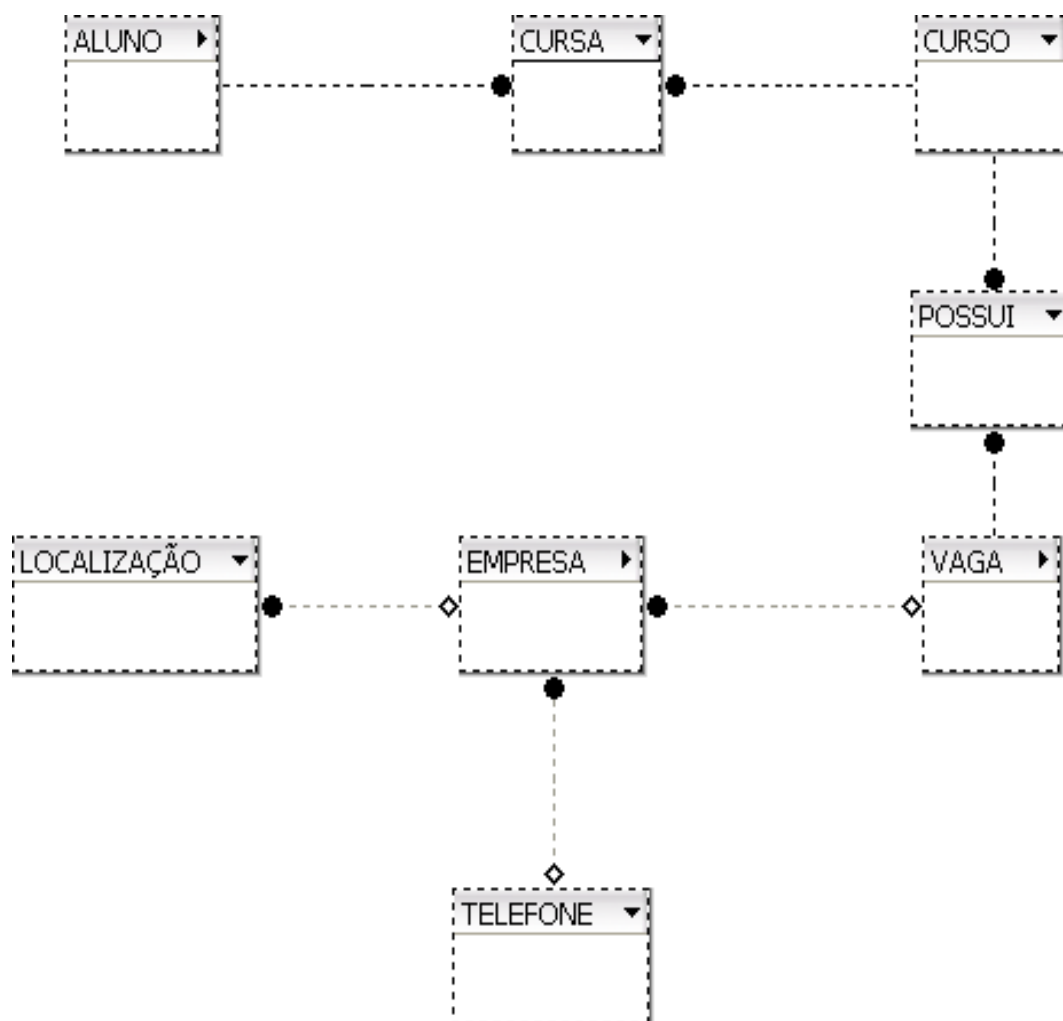


Figura 66 – Diagrama de Estrutura de Dados – Mural de Estágios

3.1.3. Consultas Parametrizadas – Avaliações

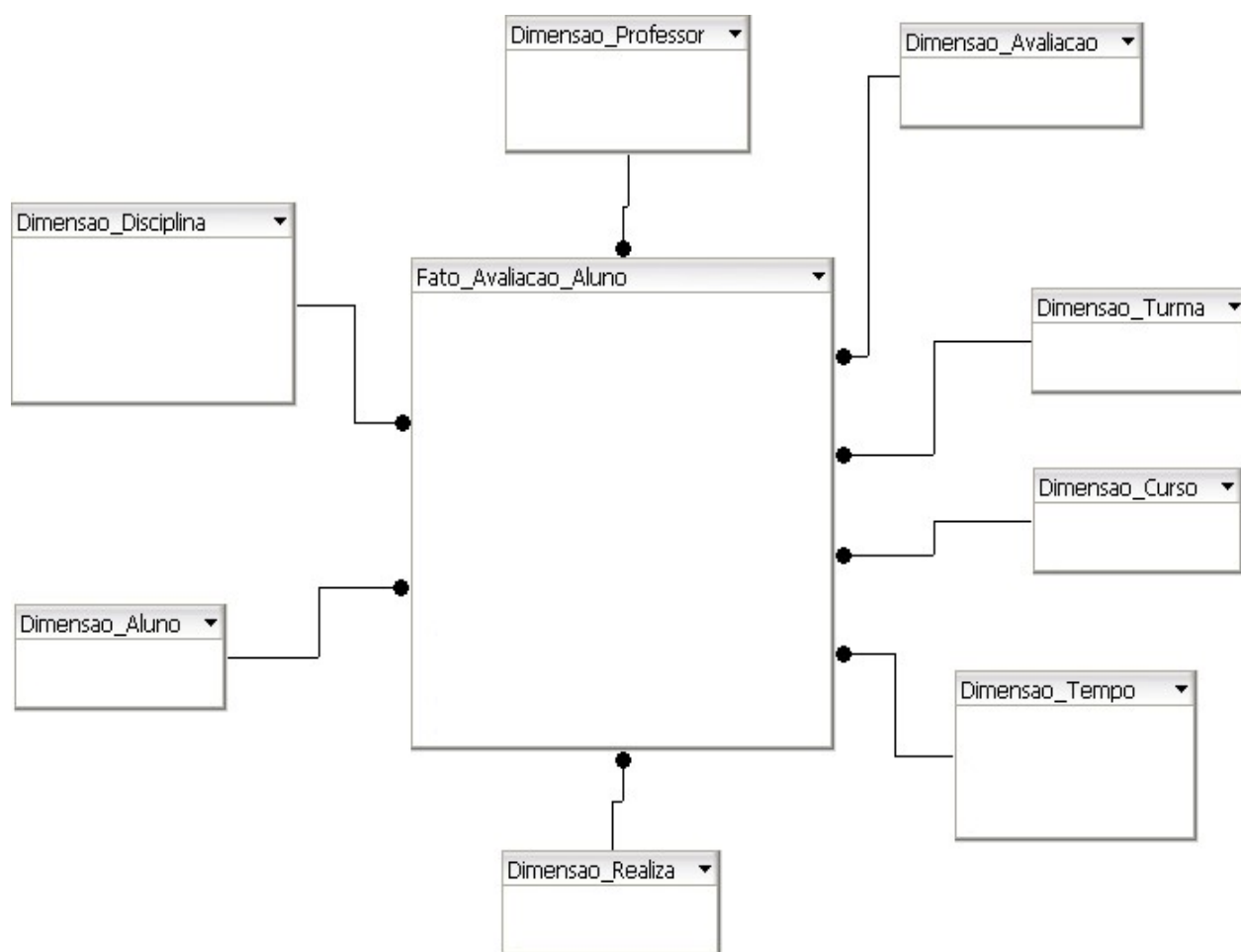


Figura 67 – Diagrama de Estrutura de Dados – Consultas Parametrizadas das Avaliações

3.1.4. Consultas Parametrizadas – Frequências

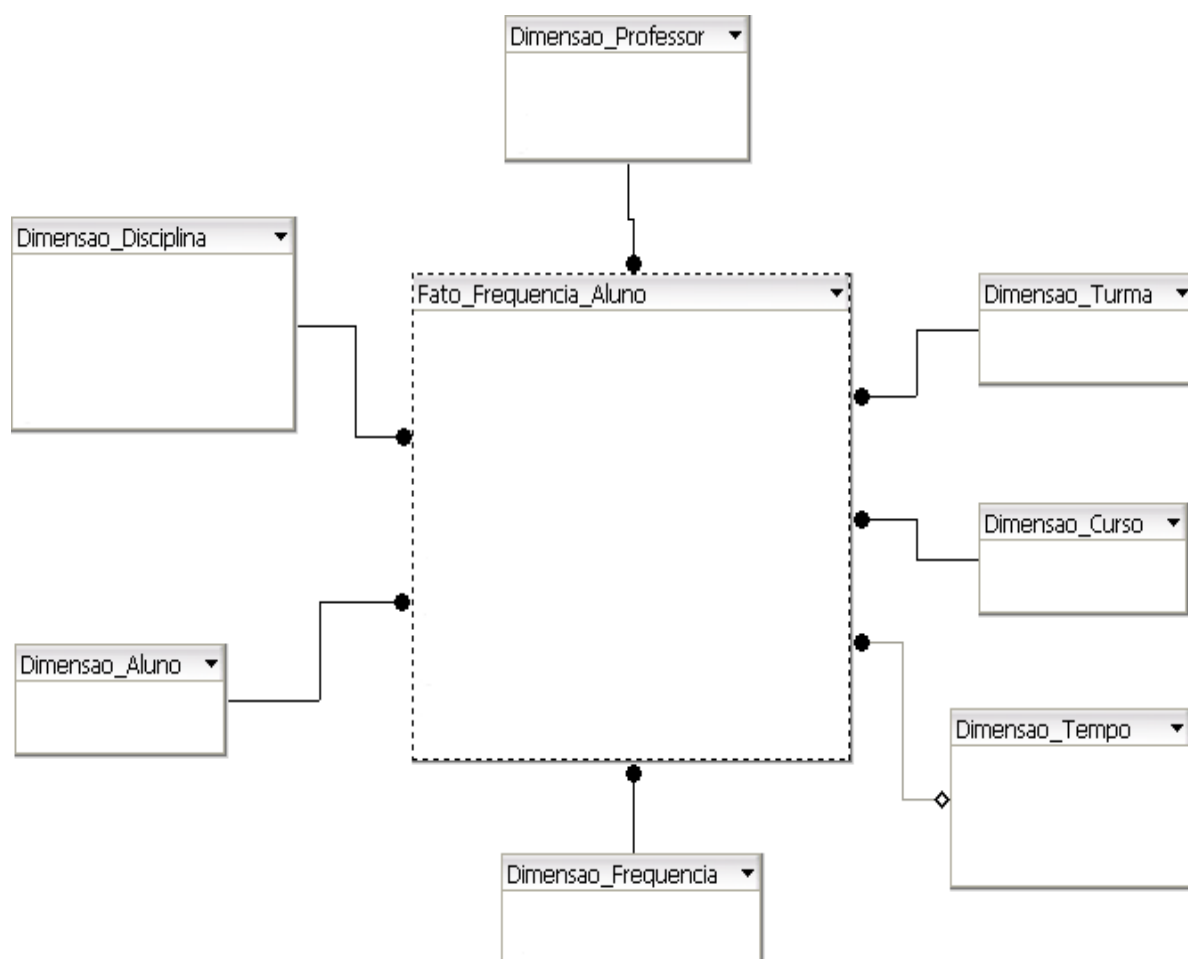


Figura 68 – Diagrama de Estrutura de Dados – Consultas Parametrizadas das Frequências

3.2. Diagrama Físico

3.2.1. Gerenciamento de Informações Acadêmicas

Figura 69 – Diagrama Físico – Gerenciamento de Informações Acadêmicas

3.2.2. Mural de Estágios

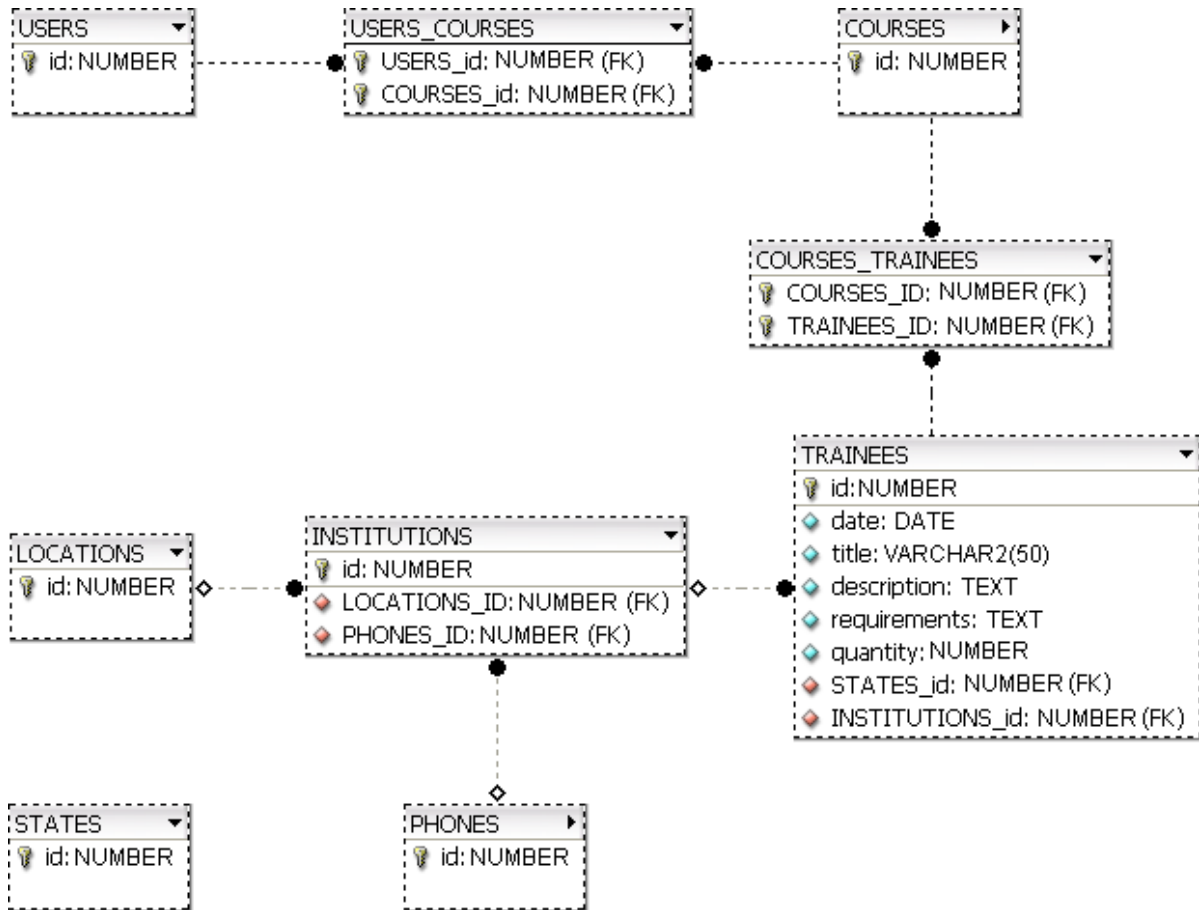


Figura 70 – Diagrama Físico – Mural de Estágios

3.2.3. Consultas Parametrizadas – Avaliações

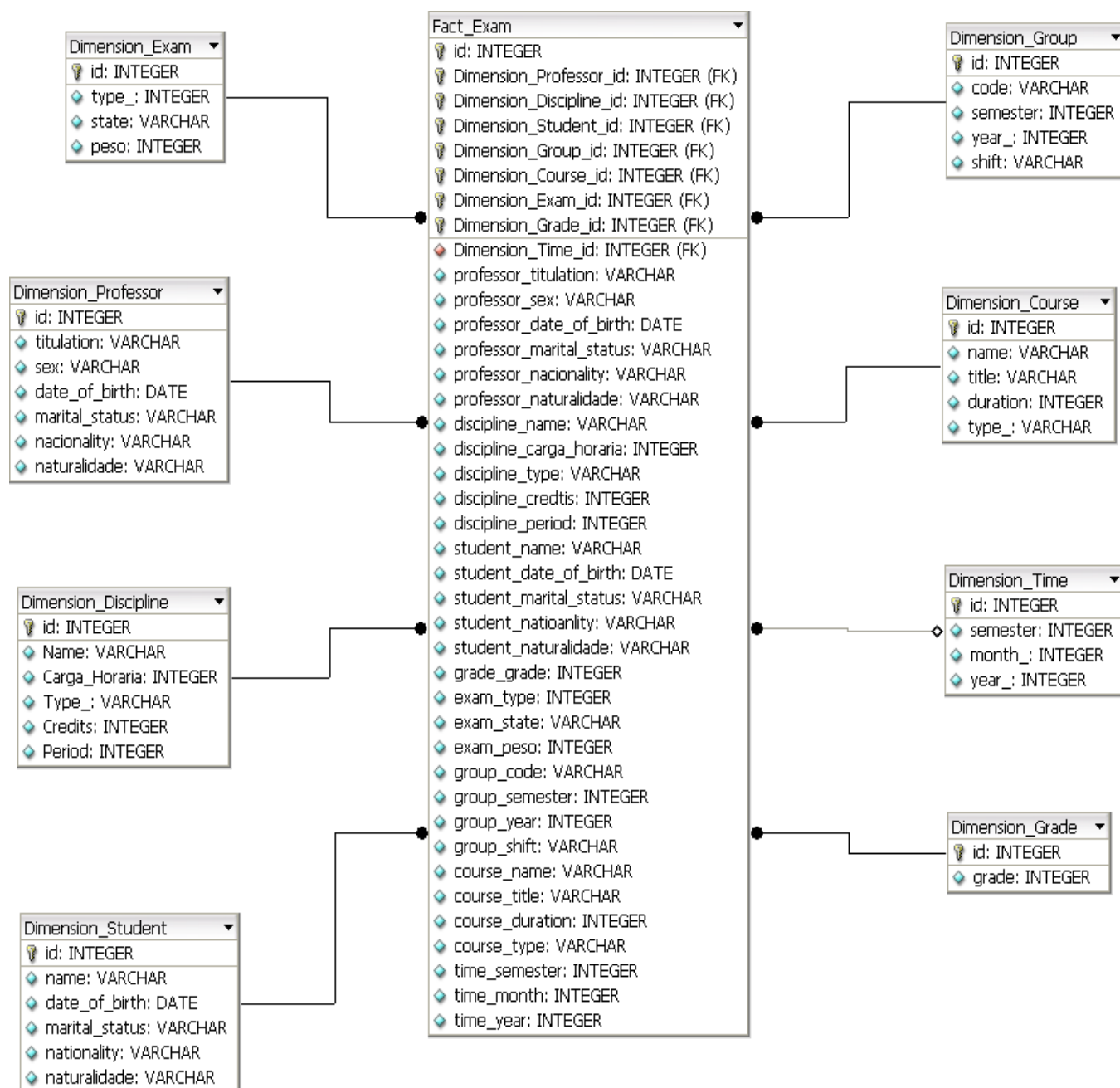


Figura 71 – Diagrama Físico – Consultas Parametrizadas das Avaliações

3.2.4. Consultas Parametrizadas – Frequências

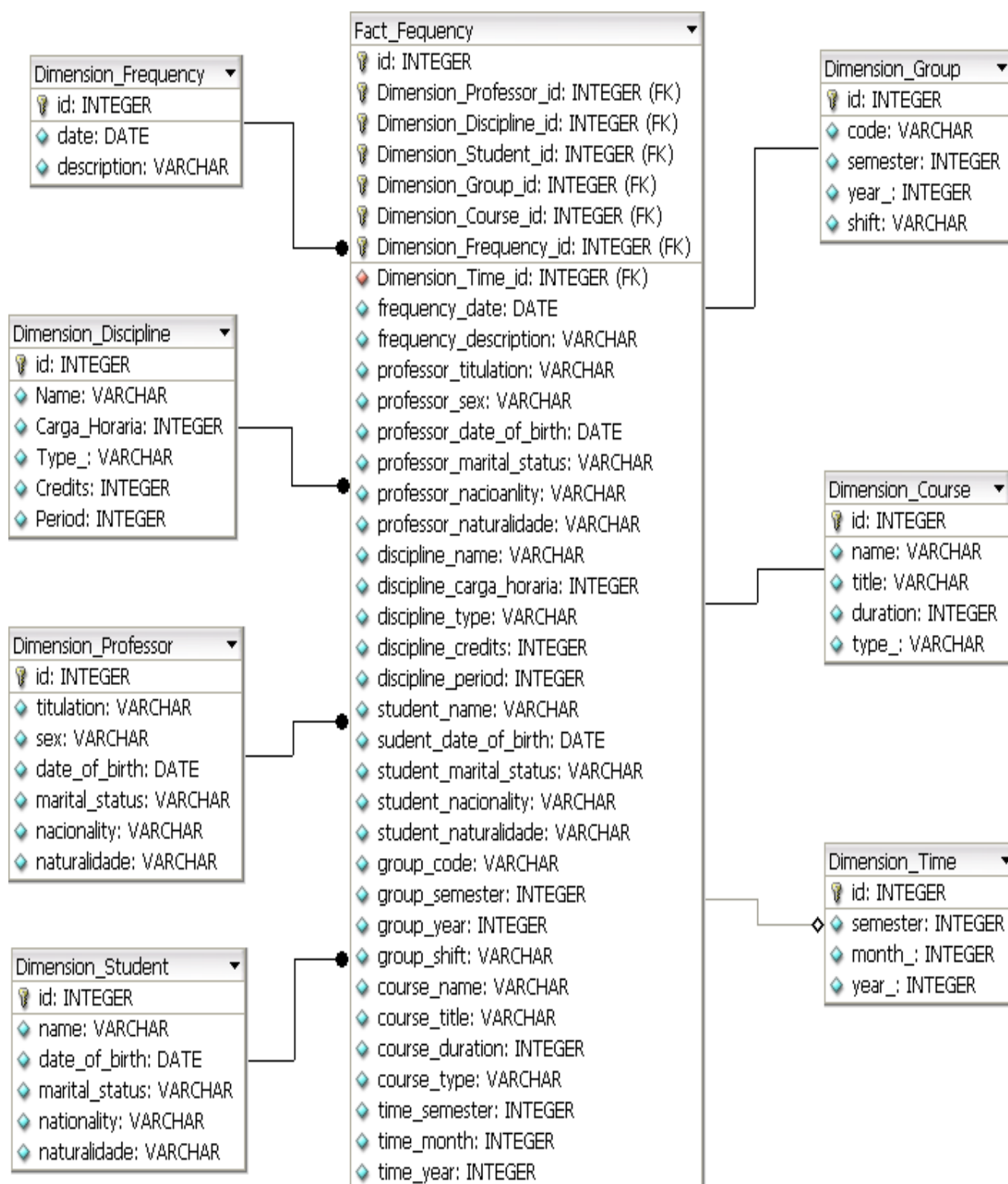


Figura 72 – Diagrama Físico – Consultas Parametrizadas das Frequências

3.3. Padrão de Nomenclatura

Para o desenvolvimento do portal acadêmico, as regras abaixo deverão ser seguidas:

3.3.1. Regras Gerais

- Toda a codificação deve estar em inglês, incluindo os comentários;
- Os nomes devem ser semânticos e simples;
- Não devem ser utilizados acentos, cedilhas e caracteres especiais;

3.3.2. Banco de Dados

3.3.2.1. Tabelas:

- São nomeadas com todas as letras minúsculas e no plural, se for um nome composto deverá ser separado por *underscore*;

3.3.2.2. Campos:

- Os campos são nomeados com todas as letras minúsculas e as palavras separadas por *underscore*;

3.3.2.3. Chaves Primárias:

- Sempre vai se chamar *id*;

3.3.2.4. Chave Estrangeira:

- A chave estrangeira sempre será nomeada com o nome da tabela de origem no singular seguido de *_id*;

3.3.2.5. Tabelas com relacionamento N x N:

- O nome da tabela deve conter o nome das tabelas utilizadas em ordem alfabética e separadas por *underscore*;

3.3.3. Aplicação

3.3.3.1. Identação:

- Tamanho máximo da linha será de 80 caracteres;
- As linhas deverão ser quebradas apenas após uma vírgula ou um operador;
- Deverão existir linhas em branco entre classes, módulos e métodos, depois de blocos e comentários de apenas uma linha e entre a parte lógica dentro de

um método para aumentar a legibilidade;

- O número de espaços em branco deverá ser balanceado, por exemplo: (a+b) e (a + b).

3.3.3.2. Classes e Módulos

- Classes e Módulos são nomeados com CamelCase;

3.3.3.3) Model

- O modelo recebe o nome no singular da tabela referente utilizando CamelCase;

3.3.3.4. Controller

- São nomeados com CamelCase, recebem nomes no plural seguido da palavra 'Controller';

3.3.3.5. Arquivos

- Os arquivos são nomeados com letras minúsculas e caso seja um nome composto deve ser separado por underscore;

3.3.3.6. Métodos e Campos

- Devem ser nomeados com letras minúsculas e separados por underscore caso seja um nome composto;

3.4. Script das tabelas do banco de dados do sistema

Abaixo seguem os scripts relacionados ao banco de dados do sistema. Os mesmos estão divididos em três partes: criação das tabelas, restrições de integridade e ações de integridade das tabelas.

3.4.1. Criação das Tabelas

3.4.1.1. Tabela Archives (Arquivos)

```
CREATE TABLE archives (  
    id integer NOT NULL,  
    name character varying(255),  
    description character varying(255),  
    kind character varying(255),  
    state_id integer,  
    group_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.2. Tabela Audits (Auditoria)

```
CREATE TABLE audits (  
    id integer NOT NULL,  
    auditable_id integer,  
    auditable_type character varying(255),  
    user_id integer,  
    user_type character varying(255),  
    username character varying(255),  
    action character varying(255),  
    changes text,  
    version integer DEFAULT 0,  
    created_at timestamp without time zone  
);
```

3.4.1.3. Tabela Bdrb_Job_Queues

```
CREATE TABLE bdrb_job_queues (
    id integer NOT NULL,
    args bytea,
    worker_name character varying(255),
    worker_method character varying(255),
    job_key character varying(255),
    taken integer,
    finished integer,
    timeout integer,
    priority integer,
    submitted_at timestamp without time zone,
    started_at timestamp without time zone,
    finished_at timestamp without time zone,
    archived_at timestamp without time zone,
    tag character varying(255),
    submitter_info character varying(255),
    runner_info character varying(255),
    worker_key character varying(255),
    scheduled_at timestamp without time zone
);
```

3.4.1.4. Tabela Contents (Conteúdos)

```
CREATE TABLE contents (
    id integer NOT NULL,
    title character varying(255),
    body text,
    description_id integer,
    created_at timestamp without time zone,
    updated_at timestamp without time zone
);
```

3.4.1.5. Tabela Courses (Cursos)

```
CREATE TABLE courses (  
    id integer NOT NULL,  
    name character varying(255),  
    title character varying(255),  
    empowerment character varying(255),  
    duration integer,  
    kind character varying(255),  
    institution_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.6. Tabela Courses_Users (Cursos_Usuários)

```
CREATE TABLE courses_users (  
    id integer NOT NULL,  
    course_id integer,  
    user_id integer  
);
```

3.4.1.7. Tabela Day_Timetables (Grade_Horário)

```
CREATE TABLE day_timetables (  
    id integer NOT NULL,  
    day_id integer,  
    timetable_id integer,  
    group_id integer  
);
```

3.4.1.8. Tabela Days (Dias)

```
CREATE TABLE days (  
    id integer NOT NULL,  
    day character varying(255),  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone);
```

3.4.1.9. Tabela Descriptions (Descrição)

```
CREATE TABLE descriptions (  
    id integer NOT NULL,  
    body text,  
    resume_discipline_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.10. Tabela Discipline_Grids (Grade_Disciplina)

```
CREATE TABLE discipline_grids (  
    id integer NOT NULL,  
    name character varying(255),  
    description character varying(255),  
    course_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.11. Tabela Disciplines (Disciplina)

```
CREATE TABLE disciplines (  
    id integer NOT NULL,  
    name character varying(255),  
    working_hours integer,  
    kind character varying(255),  
    credits integer,  
    period integer,  
    state_id integer,  
    discipline_grid_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.12. Tabela Documents (Documentos)

```
CREATE TABLE documents (  
    id integer NOT NULL,  
    number integer,  
    kind character varying(255),  
    title character varying(255),  
    body text,  
    model character varying(255),  
    users_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.13. Tabela Exams (Avaliação)

```
CREATE TABLE exams (  
    id integer NOT NULL,  
    kind character varying(255),  
    weighting integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.14. Tabela Groups (Turma)

```
CREATE TABLE groups (  
    id integer NOT NULL,  
    code character varying(255),  
    semester character varying(255),  
    year integer,  
    shift character varying(255),  
    discipline_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.15. Tabela Institutions (Instituição)

```
CREATE TABLE institutions (  
    id integer NOT NULL,  
    name character varying(255),  
    cnpj character varying(255),  
    website character varying(255),  
    email character varying(255),  
    contact character varying(255),  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.16. Tabela Iptocs

```
CREATE TABLE iptocs (  
    id integer NOT NULL,  
    ip_from integer NOT NULL,  
    ip_to integer NOT NULL,  
    country_code2 character varying(255) NOT NULL,  
    country_code3 character varying(255) NOT NULL,  
    country_name character varying(255) NOT NULL  
);
```


3.4.1.17. Tabela Locations (Localização)

```
CREATE TABLE locations (  
    id integer NOT NULL,  
    locationable_id integer,  
    locationable_type character varying(255),  
    country character varying(255),  
    state character varying(255),  
    city character varying(255),  
    province character varying(255),  
    street character varying(255),  
    complement character varying(255),  
    zip_code integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.18. Tabela Permissions (Permissão)

```
CREATE TABLE permissions (  
    id integer NOT NULL,  
    permissible_id integer,  
    permissible_type character varying(255),  
    action character varying(255),  
    granted boolean,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.19. Tabela Phones (Telefone)

```
CREATE TABLE phones (  
    id integer NOT NULL,  
    phoneable_id integer,  
    phoneable_type character varying(255),  
    kind character varying(255),  
    ddd integer,  
    number integer,  
    note character varying(255),  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.20. Tabela Places (Locais)

```
CREATE TABLE places (  
    id integer NOT NULL,  
    kind character varying(255),  
    description character varying(255),  
    hierarchy integer,  
    places_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.21. Tabela Publications (Divulgação)

```
CREATE TABLE publications (  
    id integer NOT NULL,  
    title character varying(255),  
    kind character varying(255),  
    date timestamp without time zone,  
    place character varying(255),  
    note character varying(255),  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone);
```

3.4.1.22. Tabela Rail_Stats

```
CREATE TABLE rail_stats (  
    id integer NOT NULL,  
    remote_ip character varying(255),  
    country character varying(255),  
    language character varying(255),  
    domain character varying(255),  
    subdomain character varying(255),  
    referer character varying(255),  
    resource character varying(255),  
    user_agent character varying(255),  
    platform character varying(255),  
    browser character varying(255),  
    version character varying(255),  
    created_at timestamp without time zone,  
    created_on date,  
    screen_size character varying(255),  
    colors character varying(255),  
    java character varying(255),  
    java_enabled character varying(255),  
    flash character varying(255)  
);
```

3.4.1.23. Tabela Registrations (Inscrição)

```
CREATE TABLE registrations (  
    id integer NOT NULL,  
    date timestamp without time zone,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.24. Tabela Releases (Lançamento)

```
CREATE TABLE releases (  
    id integer NOT NULL,  
    comment text,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.25. Tabela Resources (Recursos)

```
CREATE TABLE resources (  
    id integer NOT NULL,  
    type character varying(255),  
    quantity integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.26. Tabela Resume_Disciplines (Ementa)

```
CREATE TABLE resume_disciplines (  
    id integer NOT NULL,  
    name character varying(255),  
    discipline_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.27. Tabela Role_Memberships

```
CREATE TABLE role_memberships (  
    id integer NOT NULL,  
    roleable_id integer,  
    roleable_type character varying(255),  
    role_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone);
```

3.4.1.28. Tabela Roles (Perfil)

```
CREATE TABLE roles (  
    id integer NOT NULL,  
    name character varying(255),  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.29. Tabela Rooms (Sala)

```
CREATE TABLE rooms (  
    id integer NOT NULL,  
    number character varying(255),  
    size integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.30. Tabela Schema_Migration

```
CREATE TABLE schema_migrations (  
    version character varying(255) NOT NULL  
);
```

3.4.1.31. Tabela Search_Terms

```
CREATE TABLE search_terms (  
    id integer NOT NULL,  
    subdomain character varying(255) DEFAULT ''::character varying,  
    searchterms character varying(255) DEFAULT ''::character varying,  
    count integer DEFAULT 0 NOT NULL,  
    domain character varying(255)  
);
```

3.4.1.32. Tabela States (Estado)

```
CREATE TABLE states (  
    id integer NOT NULL,  
    name character varying(255),  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.33. Tabela Timetables (Horário)

```
CREATE TABLE timetables (  
    id integer NOT NULL,  
    begin_time time without time zone,  
    end_time time without time zone,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.34. Tabela Trainees (Estágio)

```
CREATE TABLE trainees (  
    id integer NOT NULL,  
    title character varying(255),  
    description text,  
    requirements text,  
    status_id character varying(255),  
    institution_id integer,  
    course_id integer,  
    created_at timestamp without time zone,  
    updated_at timestamp without time zone  
);
```

3.4.1.35. Tabela Users (Usuários)

```
CREATE TABLE users (
    id integer NOT NULL,
    login character varying(50),
    crypted_password character varying(40),
    salt character varying(40),
    name character varying(100),
    registration character varying(40),
    cpf integer,
    email character varying(80),
    sex character varying(1),
    date_of_birth timestamp without time zone,
    father_name character varying(100),
    mother_name character varying(100),
    marital_status character varying(40),
    nationality character varying(40),
    naturalidade character varying(40),
    remember_token character varying(40),
    remember_token_expires_at timestamp without time zone,
    last_login character varying(255),
    states_id integer,
    identities_id integer,
    type character varying(40),
    title character varying(50),
    newsletter character varying(1),
    job_status character varying(20),
    workplace character varying(50),
    sector character varying(40),
    created_at timestamp without time zone,
    updated_at timestamp without time zone,
    activation_code character varying(40),
    activated_at timestamp without time zone
);
```

3.4.2. Restrições de Integridade das Tabelas

3.4.2.1. Restrição de Integridade da Tabela Archives

```
ALTER TABLE archives ALTER COLUMN id SET DEFAULT  
nextval('archives_id_seq'::regclass);
```

3.4.2.2. Restrição de Integridade da Tabela Audits

```
ALTER TABLE audits ALTER COLUMN id SET DEFAULT nextval('audits_id_seq'::regclass);
```

3.4.2.3. Restrição de Integridade da Tabela Bdrb_Job_Queues

```
ALTER TABLE bdrb_job_queues ALTER COLUMN id SET DEFAULT  
nextval('bdrb_job_queues_id_seq'::regclass);
```

3.4.2.4. Restrição de Integridade da Tabela Contents

```
ALTER TABLE contents ALTER COLUMN id SET DEFAULT  
nextval('contents_id_seq'::regclass);
```

3.4.2.5. Restrição de Integridade da Tabela Courses

```
ALTER TABLE courses ALTER COLUMN id SET DEFAULT nextval('courses_id_seq'::regclass);
```

3.4.2.6. Restrição de Integridade da Tabela Courses_Users

```
ALTER TABLE courses_users ALTER COLUMN id SET DEFAULT  
nextval('courses_users_id_seq'::regclass);
```

3.4.2.7. Restrição de Integridade da Tabela Day_Timetables

```
ALTER TABLE day_timetables ALTER COLUMN id SET DEFAULT  
nextval('day_timetables_id_seq'::regclass);
```

3.4.2.8. Restrição de Integridade da Tabela Days

```
ALTER TABLE days ALTER COLUMN id SET DEFAULT nextval('days_id_seq'::regclass);
```


3.4.2.9. Restrição de Integridade da Tabela Descriptions

```
ALTER TABLE descriptions ALTER COLUMN id SET DEFAULT  
nextval('descriptions_id_seq'::regclass);
```

3.4.2.10. Restrição de Integridade da Tabela Discipline_Grids

```
ALTER TABLE discipline_grids ALTER COLUMN id SET DEFAULT  
nextval('discipline_grids_id_seq'::regclass);
```

3.4.2.11. Restrição de Integridade da Tabela Disciplines

```
ALTER TABLE disciplines ALTER COLUMN id SET DEFAULT  
nextval('disciplines_id_seq'::regclass);
```

3.4.2.12. Restrição de Integridade da Tabela Documents

```
ALTER TABLE documents ALTER COLUMN id SET DEFAULT  
nextval('documents_id_seq'::regclass);
```

3.4.2.13. Restrição de Integridade da Tabela Exams

```
ALTER TABLE exams ALTER COLUMN id SET DEFAULT nextval('exams_id_seq'::regclass);
```

3.4.2.14. Restrição de Integridade da Tabela Groups

```
ALTER TABLE groups ALTER COLUMN id SET DEFAULT nextval('groups_id_seq'::regclass);
```

3.4.2.15. Restrição de Integridade da Tabela Institutions

```
ALTER TABLE institutions ALTER COLUMN id SET DEFAULT  
nextval('institutions_id_seq'::regclass);
```

3.4.2.16. Restrição de Integridade da Tabela Iptocs

```
ALTER TABLE iptocs ALTER COLUMN id SET DEFAULT nextval('iptocs_id_seq'::regclass);
```

3.4.2.17. Restrição de Integridade da Tabela Locations

```
ALTER TABLE locations ALTER COLUMN id SET DEFAULT  
nextval('locations_id_seq'::regclass);
```

3.4.2.18. Restrição de Integridade da Tabela Permissions

```
ALTER TABLE permissions ALTER COLUMN id SET DEFAULT  
nextval('permissions_id_seq'::regclass);
```

3.4.2.19. Restrição de Integridade da Tabela Phones

```
ALTER TABLE phones ALTER COLUMN id SET DEFAULT nextval('phones_id_seq'::regclass);
```

3.4.2.20. Restrição de Integridade da Tabela Places

```
ALTER TABLE places ALTER COLUMN id SET DEFAULT nextval('places_id_seq'::regclass);
```

3.4.2.21) Restrição de Integridade da Tabela Publications

```
ALTER TABLE publications ALTER COLUMN id SET DEFAULT  
nextval('publications_id_seq'::regclass);
```

3.4.2.22. Restrição de Integridade da Tabela Rail_Stats

```
ALTER TABLE rail_stats ALTER COLUMN id SET DEFAULT  
nextval('rail_stats_id_seq'::regclass);
```

3.4.2.23. Restrição de Integridade da Tabela Registrations

```
ALTER TABLE registrations ALTER COLUMN id SET DEFAULT  
nextval('registrations_id_seq'::regclass);
```

3.4.2.24. Restrição de Integridade da Tabela Releases

```
ALTER TABLE releases ALTER COLUMN id SET DEFAULT nextval('releases_id_seq'::regclass);
```

3.4.2.25. Restrição de Integridade da Tabela Resources

```
ALTER TABLE resources ALTER COLUMN id SET DEFAULT  
nextval('resources_id_seq'::regclass);
```

3.4.2.26. Restrição de Integridade da Tabela Resume_Disciplines

```
ALTER TABLE resume_disciplines ALTER COLUMN id SET DEFAULT  
nextval('resume_disciplines_id_seq'::regclass);
```

3.4.2.27. Restrição de Integridade da Tabela Role_Memberships

```
ALTER TABLE role_memberships ALTER COLUMN id SET DEFAULT  
nextval('role_memberships_id_seq'::regclass);
```

3.4.2.28. Restrição de Integridade da Tabela Roles

```
ALTER TABLE roles ALTER COLUMN id SET DEFAULT nextval('roles_id_seq'::regclass);
```

3.4.2.29. Restrição de Integridade da Tabela Rooms

```
ALTER TABLE rooms ALTER COLUMN id SET DEFAULT nextval('rooms_id_seq'::regclass);
```

3.4.2.30. Restrição de Integridade da Tabela Search_Terms

```
ALTER TABLE search_terms ALTER COLUMN id SET DEFAULT  
nextval('search_terms_id_seq'::regclass);
```

3.4.2.31. Restrição de Integridade da Tabela States

```
ALTER TABLE states ALTER COLUMN id SET DEFAULT nextval('states_id_seq'::regclass);
```

3.4.2.32. Restrição de Integridade da Tabela Timetables

```
ALTER TABLE timetables ALTER COLUMN id SET DEFAULT  
nextval('timetables_id_seq'::regclass);
```

3.4.2.33. Restrição de Integridade da Tabela Trainees

```
ALTER TABLE trainees ALTER COLUMN id SET DEFAULT nextval('trainees_id_seq'::regclass);
```

3.4.2.34. Restrição de Integridade da Tabela Users

```
ALTER TABLE users ALTER COLUMN id SET DEFAULT nextval('users_id_seq'::regclass);
```

3.4.3. Ações de Integridade – Chaves Primárias das Tabelas

3.4.3.1. Primary Key da Tabela Archives

ALTER TABLE ONLY archives

ADD CONSTRAINT archives_pkey PRIMARY KEY (id);

3.4.3.2. Primary Key da Tabela Audits

ALTER TABLE ONLY audits

ADD CONSTRAINT audits_pkey PRIMARY KEY (id);

3.4.3.3. Primary Key da Tabela Bdrb_Job_Queues

ALTER TABLE ONLY bdrb_job_queues

ADD CONSTRAINT bdrb_job_queues_pkey PRIMARY KEY (id);

3.4.3.4. Primary Key da Tabela Contents

ALTER TABLE ONLY contents

ADD CONSTRAINT contents_pkey PRIMARY KEY (id);

3.4.3.5. Primary Key da Tabela Courses

ALTER TABLE ONLY courses

ADD CONSTRAINT courses_pkey PRIMARY KEY (id);

3.4.3.6. Primary Key da Tabela Courses_Users

ALTER TABLE ONLY courses_users

ADD CONSTRAINT courses_users_pkey PRIMARY KEY (id);

3.4.3.7. Primary Key da Tabela Day_Timetables

ALTER TABLE ONLY day_timetables

ADD CONSTRAINT day_timetables_pkey PRIMARY KEY (id);

3.4.3.8. Primary Key da Tabela Days

ALTER TABLE ONLY days

ADD CONSTRAINT days_pkey PRIMARY KEY (id);

3.4.3.9. Primary Key da Tabela Descriptions

ALTER TABLE ONLY descriptions

ADD CONSTRAINT descriptions_pkey PRIMARY KEY (id);

3.4.3.10. Primary Key da Tabela Discipline_Grids

ALTER TABLE ONLY discipline_grids

ADD CONSTRAINT discipline_grids_pkey PRIMARY KEY (id);

3.4.3.11. Primary Key da Tabela Disciplines

ALTER TABLE ONLY disciplines

ADD CONSTRAINT disciplines_pkey PRIMARY KEY (id);

3.4.3.12. Primary Key da Tabela Documents

ALTER TABLE ONLY documents

ADD CONSTRAINT documents_pkey PRIMARY KEY (id);

3.4.3.13. Primary Key da Tabela Exams

ALTER TABLE ONLY exams

ADD CONSTRAINT exams_pkey PRIMARY KEY (id);

3.4.3.14. Primary Key da Tabela Groups

ALTER TABLE ONLY groups

ADD CONSTRAINT groups_pkey PRIMARY KEY (id);

3.4.3.15. Primary Key da Tabela Institutions

ALTER TABLE ONLY institutions

ADD CONSTRAINT institutions_pkey PRIMARY KEY (id);

3.4.3.16. Primary Key da Tabela Iptocs

ALTER TABLE ONLY iptocs

ADD CONSTRAINT iptocs_pkey PRIMARY KEY (id);

3.4.3.17. Primary Key da Tabela Locations

ALTER TABLE ONLY locations

ADD CONSTRAINT locations_pkey PRIMARY KEY (id);

3.4.3.18. Primary Key da Tabela Permissions

ALTER TABLE ONLY permissions

ADD CONSTRAINT permissions_pkey PRIMARY KEY (id);

3.4.3.19. Primary Key da Tabela Phones

ALTER TABLE ONLY phones

ADD CONSTRAINT phones_pkey PRIMARY KEY (id);

3.4.3.20. Primary Key da Tabela Places

ALTER TABLE ONLY places

ADD CONSTRAINT places_pkey PRIMARY KEY (id);

3.4.3.21. Primary Key da Tabela Publications

ALTER TABLE ONLY publications

ADD CONSTRAINT publications_pkey PRIMARY KEY (id);

3.4.3.22. Primary Key da Tabela Rail_Stats

ALTER TABLE ONLY rail_stats

ADD CONSTRAINT rail_stats_pkey PRIMARY KEY (id);

3.4.3.23. Primary Key da Tabela Registration

ALTER TABLE ONLY registrations

ADD CONSTRAINT registrations_pkey PRIMARY KEY (id);

3.4.3.24. Primary Key da Tabela Releases

ALTER TABLE ONLY releases

ADD CONSTRAINT releases_pkey PRIMARY KEY (id);

3.4.3.25. Primary Key da Tabela Resources

ALTER TABLE ONLY resources

ADD CONSTRAINT resources_pkey PRIMARY KEY (id);

3.4.3.26. Primary Key da Tabela Resume_Disciplines

ALTER TABLE ONLY resume_disciplines

ADD CONSTRAINT resume_disciplines_pkey PRIMARY KEY (id);

3.4.3.27. Primary Key da Tabela Role_Memberships

ALTER TABLE ONLY role_memberships

ADD CONSTRAINT role_memberships_pkey PRIMARY KEY (id);

3.4.3.28. Primary Key da Tabela Roles

ALTER TABLE ONLY roles

ADD CONSTRAINT roles_pkey PRIMARY KEY (id);

3.4.3.29. Primary Key da Tabela Rooms

ALTER TABLE ONLY rooms

ADD CONSTRAINT rooms_pkey PRIMARY KEY (id);

3.4.3.30. Primary Key da Tabela Search Terms

ALTER TABLE ONLY search_terms

ADD CONSTRAINT search_terms_pkey PRIMARY KEY (id);

3.4.3.31. Primary Key da Tabela States

ALTER TABLE ONLY states

ADD CONSTRAINT states_pkey PRIMARY KEY (id);

3.4.3.32. Primary Key da Tabela Timetables

ALTER TABLE ONLY timetables

ADD CONSTRAINT timetables_pkey PRIMARY KEY (id);

3.4.3.33. Primary Key da Tabela Trainees

ALTER TABLE ONLY trainees

ADD CONSTRAINT trainees_pkey PRIMARY KEY (id);

3.4.3.34. Primary Key da Tabela Users

ALTER TABLE ONLY users

ADD CONSTRAINT users_pkey PRIMARY KEY (id);

3.4.4. Ações de Integridade – Chaves Estrangeiras das Tabelas

3.4.4.1. Foreign Key da Tabela Archives

ALTER TABLE archives

```
ADD CONSTRAINT groups_fkey FOREIGN KEY (group_id)
REFERENCES groups;
```

ALTER TABLE archives

```
ADD CONSTRAINT states_fkey FOREIGN KEY (state_id)
REFERENCES states;
```

3.4.4.2. Foreign Key da Tabela Contents

ALTER TABLE contents

```
ADD CONSTRAINT descriptions_fkey FOREIGN KEY (description_id)
REFERENCES descriptions;
```

3.4.4.3. Foreign Key da Tabela Courses_Users

ALTER TABLE courses _users

```
ADD CONSTRAINT courses_fkey FOREIGN KEY (course_id)
REFERENCES courses;
```

ALTER TABLE courses _users

```
ADD CONSTRAINT users_fkey FOREIGN KEY (user_id)
REFERENCES users;
```

3.4.4.4. Foreign Key da Tabela Day_Timetables

```
ALTER TABLE day_timetables  
    ADD CONSTRAINT days_fkey FOREIGN KEY (day_id)  
    REFERENCES days;
```

```
ALTER TABLE day_timetables  
    ADD CONSTRAINT timetables_fkey FOREIGN KEY (timetable_id)  
    REFERENCES timetables;
```

```
ALTER TABLE day_timetables  
    ADD CONSTRAINT groups_fkey FOREIGN KEY (group_id)  
    REFERENCES groups;
```

3.4.4.5. Foreign Key da Tabela Descriptions

```
ALTER TABLE descriptions  
    ADD CONSTRAINT resume_disciplines_fkey FOREIGN KEY  
    (resume_discipline_id)  
    REFERENCES resume_disciplines;
```

3.4.4.6. Foreign Key da Tabela Discipline_Grids

```
ALTER TABLE discipline_grids  
    ADD CONSTRAINT courses_fkey FOREIGN KEY (course_id)  
    REFERENCES courses;
```

3.4.4.7. Foreign Key da Tabela Disciplines

```
ALTER TABLE disciplines  
    ADD CONSTRAINT discipline_grids_fkey FOREIGN KEY  
    (discipline_grid_id)  
    REFERENCES discipline_grids;
```

```
ALTER TABLE disciplines  
    ADD CONSTRAINT states_fkey FOREIGN KEY (state_id)  
    REFERENCES states;
```

3.4.4.8. Foreign Key da Tabela Documents

ALTER TABLE documents

```
ADD CONSTRAINT users_fkey FOREIGN KEY (user_id)
REFERENCES users;
```

3.4.4.9. Foreign Key da Tabela Exams

ALTER TABLE exams

```
ADD CONSTRAINT calendars_fkey FOREIGN KEY (calendar_id)
REFERENCES calendars;
```

ALTER TABLE exams

```
ADD CONSTRAINT groups_fkey FOREIGN KEY (group_id)
REFERENCES groups;
```

3.4.4.10. Foreign Key da Tabela Groups

ALTER TABLE groups

```
ADD CONSTRAINT disciplines_fkey FOREIGN KEY (discipline_id)
REFERENCES disciplines;
```

ALTER TABLE groups

```
ADD CONSTRAINT users_fkey FOREIGN KEY (user_id)
REFERENCES users;
```

3.4.4.11. Foreign Key da Tabela Registrations

ALTER TABLE registrations

```
ADD CONSTRAINT groups_fkey FOREIGN KEY (group_id)
REFERENCES groups;
```

ALTER TABLE registrations

```
ADD CONSTRAINT users_fkey FOREIGN KEY (user_id)
REFERENCES users;
```

3.4.4.12. Foreign Key da Tabela Releases

ALTER TABLE releases

ADD CONSTRAINT groups_fkey FOREIGN KEY (group_id)
REFERENCES groups;

3.4.4.13. Foreign Key da Tabela Resume_Disciplines

ALTER TABLE resume_disciplines

ADD CONSTRAINT disciplines_fkey FOREIGN KEY (discipline_id)
REFERENCES disciplines;

3.4.4.14. Foreign Key da Tabela Role_Memberships

ALTER TABLE role_memberships

ADD CONSTRAINT roles_fkey FOREIGN KEY (role_id)
REFERENCES roles;

3.4.4.15. Foreign Key da Tabela Trainees

ALTER TABLE trainees

ADD CONSTRAINT courses_fkey FOREIGN KEY (course_id)
REFERENCES courses;

ALTER TABLE trainees

ADD CONSTRAINT institutions_fkey FOREIGN KEY (institution_id)
REFERENCES institutions;

3.4.4.16. Foreign Key da Tabela Users

ALTER TABLE users

ADD CONSTRAINT identities_fkey FOREIGN KEY (identity_id)
REFERENCES identities;

4. Tecnologias e Metodologias utilizadas

4.1. Tecnologias

O sistema está baseado em padrões de software livre. Em 2005 foi publicado o Guia Livre, que é a referência de migração para *software* livre criado pelo governo federal. Diversas instituições estatais como o Banco do Brasil, a Marinha do Brasil, a EMBRAPA, entre outras, já aderiram ao *software* livre e o implementaram, a fim de seguir os padrões de interoperabilidade estipulados pelo governo federal. Há diversas razões para a utilização de *software* livre, entre elas estão:

- alto nível de segurança proporcionado;
- independência tecnológica – alta portabilidade;
- possibilidade da realização de auditoria nos sistemas;
- independência de um fornecedor único;
- eliminação de mudanças compulsórias propostas pelos modelos proprietários, em face da descontinuidade de suporte a versões ou soluções;

“Governos de todo o mundo estão investindo fortemente no desenvolvimento de políticas, processos e estabelecimento de padrões em TIC, montando estruturas dedicadas para obter a interoperabilidade buscando o provimento de serviços de melhor qualidade a custos reduzidos.” (<http://www.governoeletronico.gov.br>)

Para preparar a infra-estrutura do sistema, seguindo a vertente do governo federal, optou-se por utilizar plataformas altamente portáteis e não proprietárias, ou seja, passíveis de implementação em praticamente qualquer arquitetura de computador/servidor e sistema operacional. Colaborando com a alta portabilidade, também pensou-se na relação custo/benefício no intuito de aumentar a viabilidade da realização deste trabalho.

A figura 72 que segue abaixo, representa um demonstrativo da relação custo/benefício mencionada anteriormente. Esses números referem-se ao resultado da adoção do *software* livre por parte do Banco do Brasil.

Economias Geradas	R\$
Adoção do OpenOffice.org	14.401.795,00
Adoção do GNU/Linux	36.864.232,00
Uso de outras aplicações/produtos livres	8.982.950,00
TOTAL:	60.248.977,00
Economia estimada até o ano de 2010 é de R\$ 89.260.000,00	

Figura 72 – Economias Geradas com a Implementação do Software Livre – Banco do Brasil

Para tornar isto possível, foi utilizada uma série de ferramentas que compõem a estrutura do sistema.

4.1.1. Infra-estrutura

Com o objetivo de realizar o desenvolvimento do sistema, fez-se necessário um servidor que suportasse o sistema gerenciador de banco de dados e o servidor da aplicação, para fins de criação de um ambiente de desenvolvimento e testes.

A estrutura do servidor foi composta dos seguintes componentes: sistema operacional livre – Ubuntu Server 8.04, linguagem de programação Ruby utilizando o *framework* Ruby on Rails para desenvolvimento *web*, Git e GitHub para realizar o controle de versão de todo o código-fonte do sistema. O SGBD utilizado durante a fase de testes/desenvolvimento do sistema foi o SQLite 3.0. No ambiente de produção foi utilizado o PostgreSQL 8.3.

Todos os componentes listados acima corroboram para o movimento do *software* livre, pois são ferramentas gratuitas, disponíveis na internet para *download*. No Brasil, há uma organização não governamental que reúne instituições públicas e privadas, a PSL-Brasil – Projeto de Software Livre – que fomenta este movimento. Seu principal objetivo é “...a promoção do uso e do desenvolvimento do software livre como uma alternativa de liberdade de expressão, econômica e tecnológica”. (<http://www.softwarelivre.org/theproject.php>)

Segundo Ricardo Rivaldo (2004), livre significa “...poder fazer escolhas. Sendo livre, defender uma opinião, significa respeitar a opinião dos outros, mesmo que esta esteja diametralmente oposta à que defendemos na nossa consciência”.

4.1.2. Gerenciamento do Projeto – Redmine

De acordo com o PMI (*Project Management Institute*), “o gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas para projetar atividades que visem atingir os requisitos do projeto” (2000).

Segundo Koontz e O’Donnel,

“...gerenciar consiste em executar atividades e tarefas que têm como propósito planejar e controlar atividades de outras pessoas para atingir objetivos que não podem ser alcançados caso as pessoas atuem por conta própria, sem o esforço sincronizado dos subordinados” (1980).

Sendo assim, com o intuito de facilitar o acompanhamento tanto por parte dos desenvolvedores, como do orientador deste trabalho, foi utilizado o Redmine como ferramenta de gerenciamento de projetos. Trata-se de um utilitário *web* que aplica muitos dos conceitos do *Project Management Body of Knowledge* (PMBOK), que é a referência internacional no assunto de gestão/gerência de projetos.

O Redmine é um aplicativo *opensource* e encontra-se disponível na Internet para ser utilizado por qualquer pessoa, desde que esta siga as regras impostas pelo licenciamento *GNU General Public License* (GPL) v2 da *Free Software Foundation* (FSF). Assim, como este trabalho, o utilitário também foi desenvolvido utilizando-se a linguagem de programação Ruby com *framework* Ruby on Rails. O Redmine também é multi-plataforma e suporta diversos sistemas gerenciadores de banco de dados.

A ferramenta possui diversas funcionalidades e dentre elas estão: *time tracking*, gráfico de Gantt, calendário de atividades, Wiki, fórum de discussão, entre outras. Nele é possível criar diversos projetos, especificar os usuários que terão acesso aos mesmos e o que cada um poderá fazer. Também oferece um repositório *online* de arquivos, no qual é possível armazenar os dados inerentes aos diversos projetos.

A figura 73 exibe a tela inicial da ferramenta, onde é possível identificar os diferentes projetos existentes e um quadro de notícias.

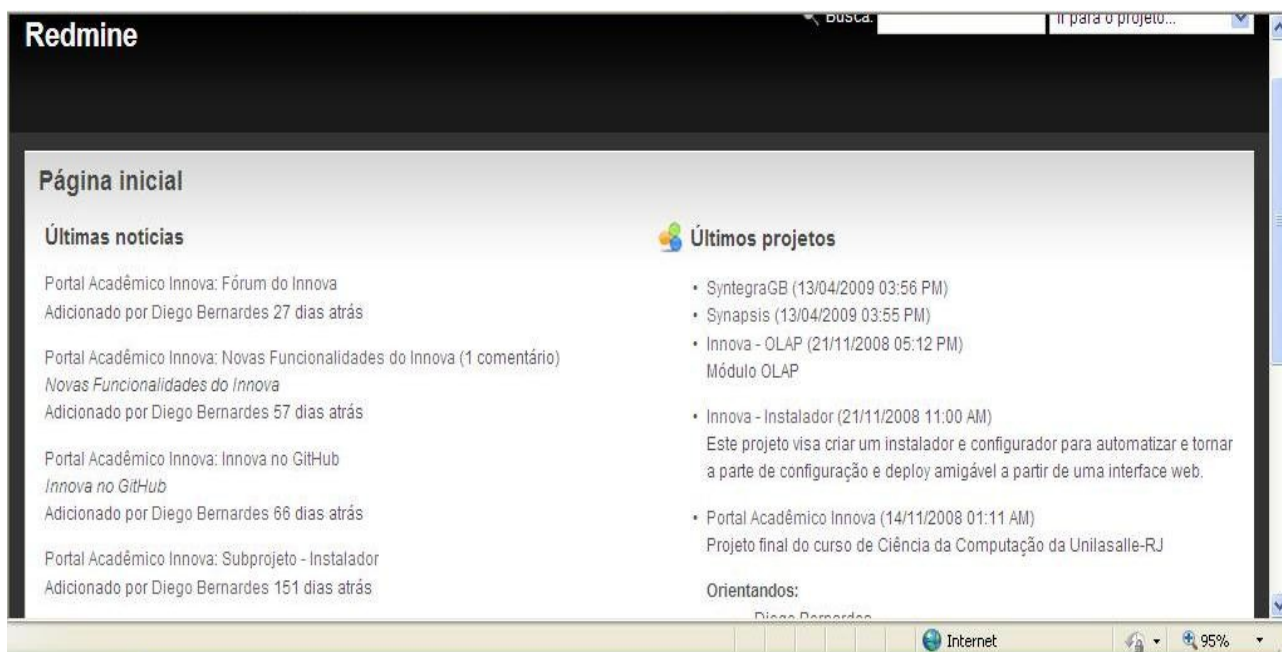


Figura 73 – Redmine – Tela Inicial

Já a figura 74 abaixo, mostra um exemplo do gráfico de Gantt, que demonstra as atividades realizadas por um integrante do projeto em um determinado período, bem como quanto tempo foi gasto para a realização das mesmas.

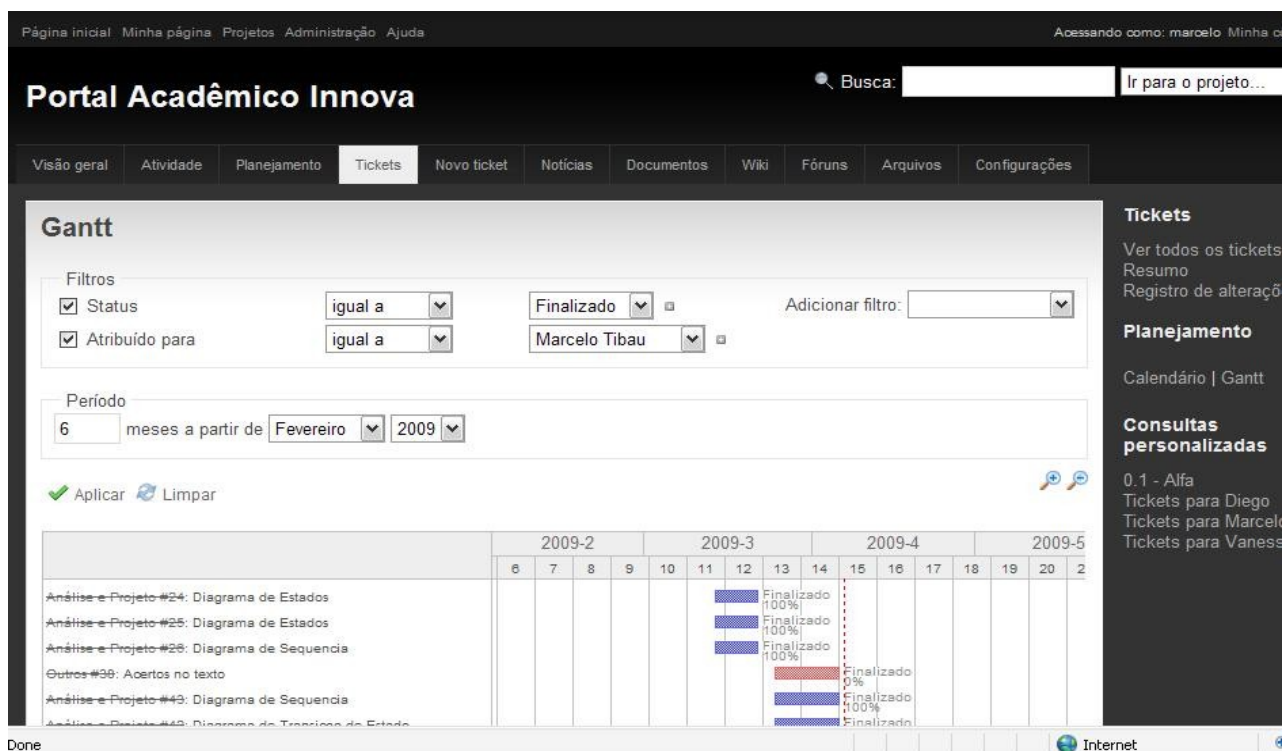
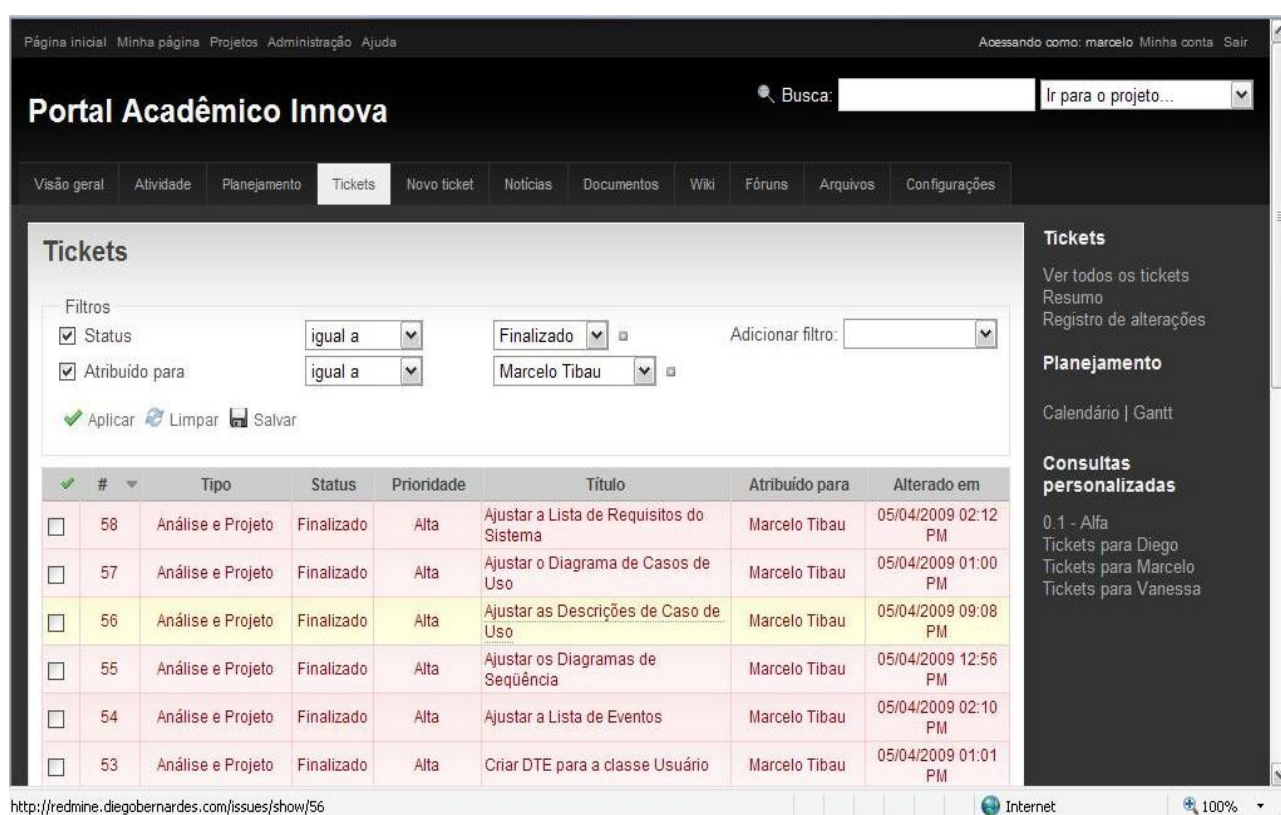


Figura 74 – Redmine – Gráfico de Gantt

Procurou-se, ao desenvolver este projeto, uma proximidade com a realidade vivida em muitas corporações, ou seja, fazer uso de uma ferramenta ou sistema para controlar/gerenciar o andamento do projeto. Entretanto, essa ferramenta deveria ser capaz de possibilitar o acesso às informações inerentes ao projeto por todos seus integrantes onde quer que estivessem.

Esse acompanhamento, tanto do orientador como dos demais participantes do projeto, pode ser feito através da análise do gráfico de Gantt, mostrado na figura acima, ou então através dos *tickets*. Estes nada mais são do que atribuições de tarefas a um ou mais integrantes do projeto, incluindo o orientador, que também é um participante. Quando a tarefa é atribuída, seu realizador recebe um e-mail contendo informações da mesma e de quem a atribuiu.



Portal Acadêmico Innova

Busca: Ir para o projeto...

Visão geral | Atividade | Planejamento | **Tickets** | Novo ticket | Notícias | Documentos | Wiki | Fóruns | Arquivos | Configurações

Tickets

Filtros

☒ Status Adicionar filtro:

☒ Atribuído para

☒ Aplicar

#	Tipo	Status	Prioridade	Título	Atribuído para	Alterado em
58	Análise e Projeto	Finalizado	Alta	Ajustar a Lista de Requisitos do Sistema	Marcelo Tibau	05/04/2009 02:12 PM
57	Análise e Projeto	Finalizado	Alta	Ajustar o Diagrama de Casos de Uso	Marcelo Tibau	05/04/2009 01:00 PM
56	Análise e Projeto	Finalizado	Alta	Ajustar as Descrições de Caso de Uso	Marcelo Tibau	05/04/2009 09:08 PM
55	Análise e Projeto	Finalizado	Alta	Ajustar os Diagramas de Sequência	Marcelo Tibau	05/04/2009 12:56 PM
54	Análise e Projeto	Finalizado	Alta	Ajustar a Lista de Eventos	Marcelo Tibau	05/04/2009 02:10 PM
53	Análise e Projeto	Finalizado	Alta	Criar DTE para a classe Usuário	Marcelo Tibau	05/04/2009 01:01 PM

Tickets

Ver todos os tickets
Resumo
Registro de alterações

Planejamento

Calendário | Gantt

Consultas personalizadas

0.1 - Alfa
Tickets para Diego
Tickets para Marcelo
Tickets para Vanessa

http://redmine.diegobernardes.com/issues/show/56

Figura 75 – Redmine – Lista de Tickets

A seguir, a figura 75 ilustra o funcionamento dos *tickets*. Pode-se alterar o modo de exibição mudando seus filtros de busca, no intuito de identificar mais rápido os realizadores, quais tarefas estão pendentes ou foram concluídas e quem é ou foi o responsável pelas mesmas.

Acima podemos identificar dados das diversas tarefas, como: *status*, prioridade, para quem foi atribuído, entre outros.

O realizador de uma determinada tarefa, ao concluí-la ou dar andamento à mesma, pode atualizar os dados do *ticket*. Cada vez que uma informação do *ticket* é atualizada, os demais integrantes do projeto recebem um e-mail informando sobre o andamento da tarefa em questão. Esta correspondência contém as informações que foram alteradas acerca da atividade como demonstrado na figura 76.

[Portal Acadêmico Innova - Análise e Projeto #56] (Finalizado) Ajustar as Descrições de Caso de Uso Innova | X

★ contato@diegobernardes.com [mostrar detalhes](#) 21:08 (12 horas atrás) [Responder](#)

Ticket #56 alterado (por Marcelo Tibau).

- Status alterado(a) de Novo para Finalizado
- % Terminado alterado(a) de 0 para 100
- Tempo estimado alterado(a) para 2.00

Todos os ajustes foram realizados e novas descrições foram adicionadas em decorrência dos novos eventos identificados.

Análise e Projeto #56: Ajustar as Descrições de Caso de Uso

- Autor: Marcelo Tibau
- Status: Finalizado
- Prioridade: Alta
- Atribuído para: Marcelo Tibau
- Categoria: 2ª Etapa - Elaboração
- Versão: 0.1 - Alfa

Realizar os ajustes nas descrições de caso de uso de acordo com as orientações do George e criar as novas descrições dos novos eventos identificados.

Links patrocinados

[Kali Software - Cursos](#)
Projeto OO com UML e Padrões - 24hs
Sábados, no Rio. Início: 09/05/09.
www.kalisoftware.com

[SYNCHRO Solução Fiscal](#)
12 mil usuários, 17 anos de mercado
SPED Fiscal/Contábil/WFe/CTe/MCe
www.synchro.com.br

[UML -Enterprise Architect](#)
Cursos, Consultoria e Ferramenta
de modelagem de software
www.oatsolutions.com.br

[Cursos de OO e UML 2.0](#)
Aulas 100% on-line, via internet.
Peça sua proposta sem compromisso.
www.buzzcenter.com.br

[Sobre estes links](#)

Contatos

Bate-papo

Procure, inclua, convide

- Marcelo Tibau [Acessar bate-papo](#)
- Cassiano diniz
- Cynthia - LGC
- Delcimário Castro
- Diego Bernardes
- Diogo Mululo
- Domicio
- edgardoliver
- elizabeth.delgado1...
- Fábio - LGC
- Fábio Vieira Ribeiro
- Feline Amorim - M

[Opções](#) [Adicionar contato](#)

Marcadores

- Innova
- Unilasalle (1)

Done Internet 100%

Figura 76 – Redmine – Informativo de um Ticket

4.1.3. Desenvolvimento

Como mencionado em parágrafos anteriores, para o desenvolvimento do sistema foi utilizada a linguagem de programação Ruby. Ela foi criada em meados de 1993, tendo sua primeira versão divulgada em 1995, mesma época do Java. Entretanto, passou a ganhar mercado nos últimos anos.

Inicialmente essa linguagem foi projetada para trabalhar com Linux, mas hoje ela é multi-plataforma, podendo ser utilizada em diversos sistemas operacionais, como: Microsoft Windows, MAC-OS, Solaris, o próprio Linux, entre outros mais.

Ruby reúne as melhores idéias de outras linguagens como Perl, Python e SmallTalk. Assim como Java, também é interpretada e orientada a objetos. Foi projetada para atender tanto à programação em larga escala, como à codificação rápida. Ainda possui tipagem dinâmica, mas forte, ou seja, isso significa que todas as variáveis devem ter um tipo, isto é, fazer parte de uma classe, mas a classe pode ser alterada dinamicamente. Essa linguagem apresenta uma forma de emular a herança múltipla, que são os *mixins*, todavia sem passar pelos problemas conhecidos por se utilizar a multiplicidade de herança.

O *framework* Ruby on Rails utiliza a metodologia de boas práticas no seu desenvolvimento, isto é, sempre que algo é criado, como um *model* ou um *controller*, automaticamente é gerado uma série de testes que depois podem ser melhorados. Possui também vários plugins para desenvolvimento, tais como: *Test Driven Development* (TDD) e *Behavior Driven Development* (BDD) que se integram perfeitamente ao *framework*.

Um dos conceitos modernos que o Ruby on Rails contempla é o *Convention Over Configuration* (COC). No dia-a-dia da programação, utiliza-se convenções, em geral, para facilitar o entendimento e a manutenção por parte de outros desenvolvedores. Levando isto em consideração e sabendo que o tempo gasto para criar arquivos de configuração em alguns frameworks de outras linguagens é extremamente alto, decidiu-se adotar esse conceito de desenvolvimento.

Outro conceito implementado pelo Rails é o *Don't Repeat Yourself* (DRY). Este consiste em uma técnica para definir nomes, propriedades e códigos em um lugar somente e reaproveitar essas informações em outros lugares. Assim, como na orientação a objetos, este conceito prega a reutilização de código de forma bastante prática e eficiente. Desta forma, permite um desenvolvimento rápido, reduz o tempo de codificação e diminui consideravelmente a quantidade de erros no código.

Colaborando para o exercício das boas práticas no desenvolvimento de um software, foi realizado o controle de versão do código-fonte do sistema. Para fazer realizar esta tarefa, foi feito uso de duas ferramentas: o Git e o GitHub. A primeira, é um software *opensource* que foi criado pelo mesmo criador do Linux, Linus Torvalds, que o utilizou para controlar as versões do *kernel*.

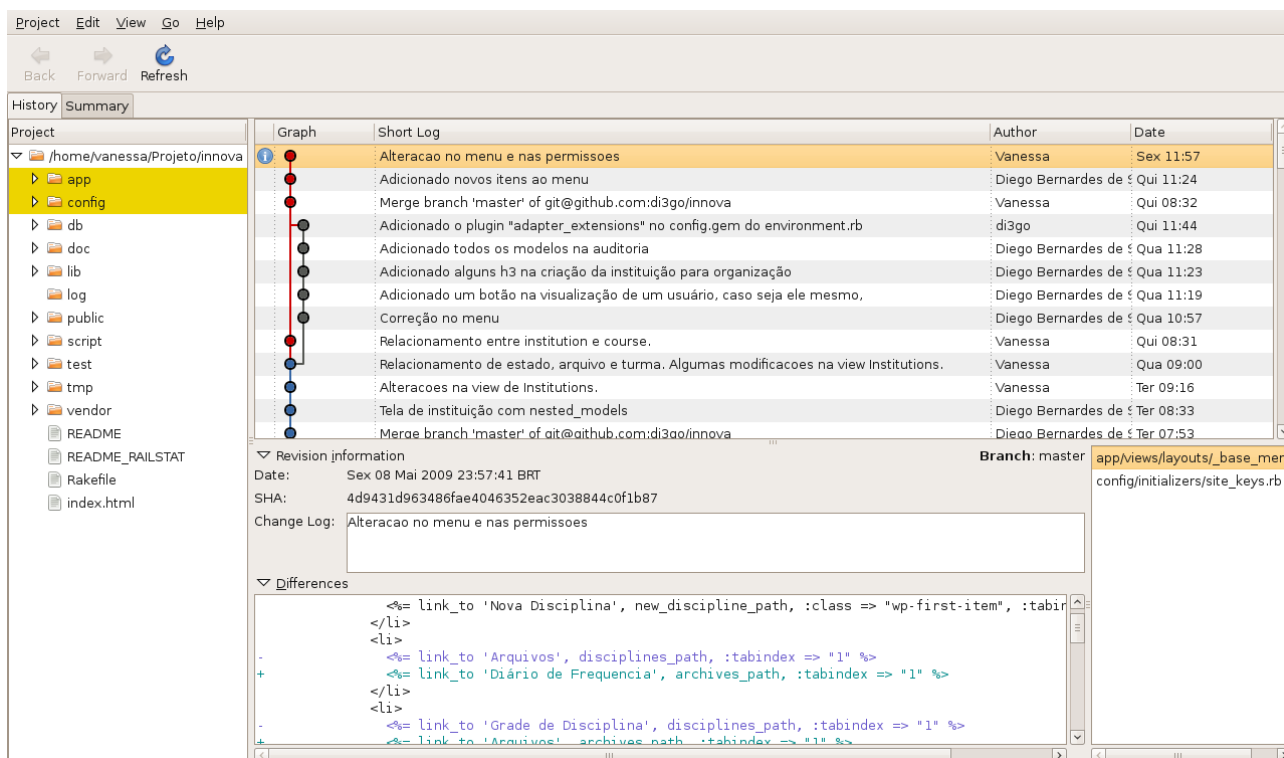


Figura 77 – Git – Árvore de Commits

Na figura 77, pode-se identificar a árvore de *commits* realizados. Cada vez que esta operação é realizada, o código fonte é atualizado e uma nova linha e/ou “folha” são adicionados. Também é possível verificar as diferenças entre as versões e o que foi alterado em cada uma, como demonstrado.

A segunda ferramenta, é um repositório *online* também desenvolvido em Ruby on Rails, para o armazenamento das versões das aplicações que estão sendo desenvolvidas e que utilizam o Git como controlador de versão. Este repositório é aberto, isto é, qualquer pessoa pode acessar os diversos projetos existentes, baixar seus códigos-fonte e utilizá-los.

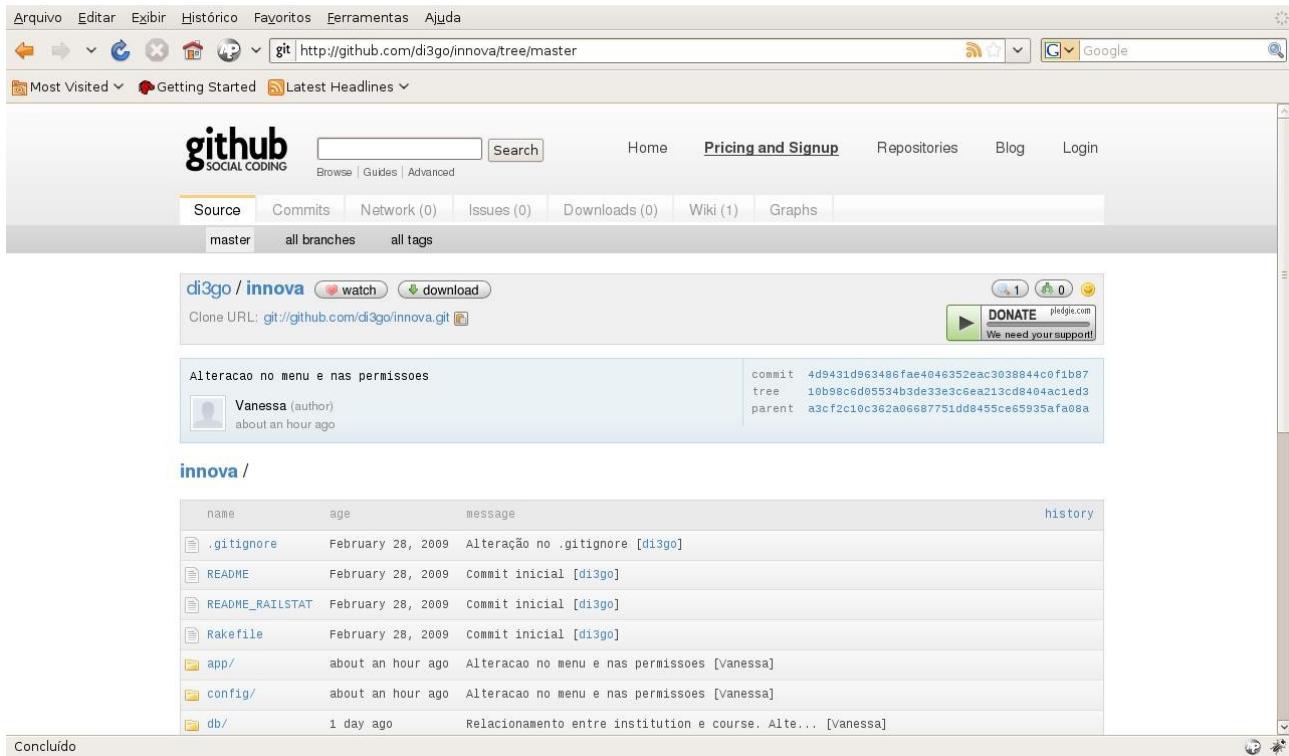


Figura 78 – GitHub – Source do Projeto Innova

A figura 78, é a representação do projeto do Portal Acadêmico Innova que se encontra disponível no GitHub.

4.1.4. Consultas Parametrizadas - OLAP

O módulo de consultas parametrizadas é um módulo que se mostra muito mais ágil quando sua implementação se dá através de um ambiente OLAP, ou seja, utilizando tabelas desnormalizadas que são representadas através de um esquema estrela.

O termo OLAP (Online Analytical Processing), pode ser definido como “*o processo interativo de criar, gerenciar, analisar e gerar relatórios sobre dados*” (DATE, 2000). Ele foi citado pela primeira vez pelo Dr. Edgar F. Codd em meados da década de 1960 quando falou-se sobre o desenvolvimento das aplicações multidimensionais e ao que elas deveriam atender.

A visão conceitual multidimensional foi uma das regras das aplicações multidimensionais citadas por Codd, a qual se tornou a característica fundamental no desenvolvimento deste tipo de aplicação. A multidimensionalidade da visão consiste em consultas que fornecem dados a respeito de medidas de desempenho, decompostas por uma ou mais dimensões dessas mesmas medidas. Elas também podem ser filtradas pela dimensão e/ou pelo valor da medida. Essas visões multidimensionais fornecem as técnicas básicas para cálculo e análise requeridos pelas aplicações de BI (*Business Intelligence*).

“*O objetivo final de uma ferramenta OLAP é transformar dados em informações capazes de dar suporte a decisões gerenciais de forma amigável e flexível ao usuário e em tempo hábil*” (THOMSEM, 2002). Nas ferramentas de navegação OLAP é possível navegar entre diferentes níveis de detalhamento dos dados.

Enfim, as instituições acadêmicas de ensino são excelentes fontes de dados que proporcionam uma grande gama de informação, pois possuem diversos cursos, com diversos professores, alunos, avaliações, recursos, etc. Para auxiliar os professores, coordenadores e demais funcionários da instituição no processo de tomada de decisão, nada melhor do que utilizar uma ferramenta OLAP.

Através dela, é possível fazer o cruzamento de informações que relatórios estáticos não conseguiriam fazer.

4.1.5. Documentação

Para que um projeto, seja ele de engenharia ou seja ele de software, esteja bem estruturado, o mesmo deve estar bem documentado. Sistemas com nível de complexidade alto demandam boa modelagem, pois:

“A modelagem é uma parte central de todas as atividades que levam à implantação de um bom software. Construímos modelos para comunicar a estrutura e o comportamento desejados do sistema. Construímos modelos para compreender melhor o sistema que estamos elaborando, muitas vezes expondo oportunidades de simplificação e reaproveitamento. Construímos modelos para gerenciar riscos”. (BOOCH, RUMBAUGH, JACOBSON, 2005, p.3)

Com a modelagem, pode-se visualizar e controlar o desenvolvimento de um sistema de maneira eficaz, identificando e gerenciando riscos, estipulando e cumprindo prazos, dentro das estimativas de custo.

Sendo assim, no intuito de corroborar para a utilização das melhores práticas no desenvolvimento de um software, procurou-se utilizar a UML (*Unified Modeling Language*) para realizar toda a documentação do sistema.

A UML é apenas uma linguagem, portanto, ela não é uma metodologia, mas sim parte de um método para desenvolvimento de software, como por exemplo, o processo unificado.

Seguindo a filosofia de utilização de ferramentas gratuitas, colaborando para a divulgação do pensamento livre, foram utilizadas as seguintes aplicações para realizar a documentação do sistema: OpenOffice na versão 3.0.1, para fazer toda a parte de textos, JUDE (*Java UML Development Environment*) Community na versão 5.4, para realizar a toda a parte de diagramas da UML e DB Design versão 4.4 que foi utilizado para fazer a representação do esquema físico do projeto.

4.1.6. Ensino à Distância – EAD – Moodle

O Brasil é um país continental com enorme extensão territorial e possui diversas comunidades remotas, ou seja, distantes dos grandes centros, onde existe maior concentração de universidades e centros de ensino. Assim, o EAD, colabora para encurtar a distância entre os grandes centros urbanos e estas comunidade, deselitizando o ensino.

O ensino à distância promovido através da internet, também conhecido como “*e-learn*”, é uma categoria de EAD recente, porém vem sendo cada vez mais procurada e adotada pelas instituições de ensino, para diversos fins.

O *e-learn* possibilita a realização de cursos de graduação, pós-graduação, mestrados, ou simplesmente, o cumprimento de uma disciplina pertencente à uma grade curricular, fazer cursos de férias, de extensão, entre outros mais.

Existem diversas ferramentas no mercado para disponibilizar um ambiente de EAD para as instituições de ensino. Existem aplicações comerciais, leia-se pagas, como por exemplo, o Blackboard e o WebCT e também uma série de softwares gratuitos que têm o mesmo propósito, como o Moodle, Dokeos, dotLRN, OpenUSS, entre outros. Algumas universidades do Rio de Janeiro como a UNIRIO, a UFF, a Uni La Salle, entre outras, já fazem uso de ferramentas de EAD e, neste caso, a escolhida pelas instituições citadas foi o Moodle.

Uma pesquisa¹, de autoria de Sabine Graf e Beate List, da escola de pós-graduação de tecnologia da Universidade de Tecnologia de Vienna , foi realizada no intuito de identificar qual ferramenta seria a melhor no quesito adaptação. O Moodle foi o único a atingir nota máxima e abaixo seguem outras razões que fazem dele, um dos mais procurados e mais utilizados:

- Fácil “customização”.
- É gratuito e *opensource*.
- Também funciona sob a licença GPL (*General Public License*).
- Pode ser instalado em um servidor *web*, no intuito de gerenciar o ambiente EAD.

De acordo com a filosofia seguida nesse trabalho, que foi a utilização de *softwares* livres e face ao exposto acima, o Moodle foi escolhido como ferramenta para realizar a integração do Portal Acadêmico Innova com o módulo de EAD.

Neste trabalho, o intuito não foi desenvolver uma ferramenta ou criar um ambiente de ensino à distância, mas sim integrar ambientes existentes com a o portal. Dessa forma, o controle de acesso ao ambiente EAD é feito através do Portal Acadêmico Innova, isto é, não é necessário criar usuários nos dois sistemas, no Moodle e no portal.

¹ Pesquisa disponível em: www.campussource.de/aktuelles/docs/icalt2005.pdf

No ambiente de testes, utilizado para o desenvolvimento do Portal Acadêmico Innova, o Moodle foi instalado e configurado de modo que a integração entre as duas ferramentas fosse testada.

4.2. Metodologias

4.2.1. Levantamento de Requisitos do Sistema

A análise e o levantamento de requisitos são fundamentais para o desenvolvimento de sistemas, pois tratam justamente de descobrir o que os clientes esperam do sistema e de dar as diretrizes para o andamento do projeto.

O desenvolvimento de um sistema se dá através da análise de informações levantadas. Estes dados podem ser provenientes de entrevistas com clientes, questionários, pesquisas, observações, entre outras maneiras mais.

De acordo com Menga Lüdke e Marli André (2004), *“Definindo-se claramente o foco da investigação e sua configuração espaço-temporal, ficam mais ou menos evidentes quais aspectos do problema serão cobertos pela observação e qual a melhor forma de captá-los.”*

O Portal Acadêmico Innova foi idealizado a partir de pesquisas realizadas em ferramentas disponíveis no mercado, que possuem propósito semelhante à que foi desenvolvida neste trabalho. Durante a pesquisa, foi feita uma análise crítica de cada ferramenta a fim de extrair o que se julgou ser positivo em cada uma.

Também foram criados questionários, no intuito de fazer uma pesquisa de opinião junto aos professores da Uni La Salle. Essa pesquisa, teve como principal objetivo, realizar o alinhamento das idéias que foram levantadas para o desenvolvimento do trabalho, quais as operações que o sistema devia realizar e quais as restrições que existiriam sobre estas operações.

4.2.2. Documentação – Processo Unificado (PU)

Para realizar a documentação do sistema, utilizou-se a metodologia do Processo Unificado (PU). O PU tem por objetivo² “*garantir a produção de software de alta qualidade que está de acordo com as necessidades de seus usuários finais com um cronograma e custo previstos*”.

Através dessa metodologia, é possível elaborar as fases do projeto, que são: concepção, elaboração, construção e transição. Esta organização facilita a definição dos processos inerentes ao desenvolvimento de software. O processo unificado é um conjunto de atividades bem definidas, com modelo de ciclo de vida e, além de ser iterativo e incremental, serve como “guia” para a utilização da UML.

O PU foi elaborado de acordo com as seis melhores práticas (que são as disciplinas do PU) no que concerne ao desenvolvimento de um projeto de software. A figura 80 mostra quais são as práticas e o que deve ser feito em cada uma.

Análise	Limitar o escopo
	Identificar necessidades
	Determinar prazo
Projeto	Definir Infra-estrutura
	Regras de negócio
	Protótipo
Construção	Desenvolvimento
	Controle de versão
	Testes
	Versão para homologação
Homologação	Testes funcionais e estruturais
	Análise da documentação
	Validação dos requisitos
Implantação	Manual do usuário
	Treinamento
	Migrar dados legados
	Instalar a solução no ambiente de produção
Manutenção	Corretiva e/ou evolutiva
	Identificar novas necessidades
	Implementar as mudanças
	Suporte ao cliente

Figura 80 – Disciplinas do PU

² Disponível em: www.fabsoft.cesupa.br/site/images/introducao_arquitetura_software.pdf

5. Protótipos de Tela e Cronograma do Trabalho

5.1. Telas do Sistema

As telas do sistema são apresentadas no manual do usuário.

5.2. Cronograma

5.2.1. Segundo semestre de 2008

SEGUNDO SEMESTRE DE 2008					
ETAPAS	AGOSTO	SETEMBRO	OUTUBRO	NOVEMBRO	DEZEMBRO
Objetivo e Justificativa do Projeto	X				
Descrição/Desenvolvimento do Minimundo	X	X	X		
Requisitos do Sistema	X	X	X		
Diagrama de Classes Conceitual		X	X	X	
Contextualização			X		
Lista de Eventos			X		
Diagrama de Casos de Uso			X		
Diagrama de Seqüência			X	X	X
Diagrama de Estado			X	X	
Diagrama de Classe Lógico			X		
Dicionário de Objetos			X	X	
Diagrama de Estrutura de Dados			X	X	
Esquema Relacional			X	X	
Script das Tabelas				X	

5.2.2. Primeiro semestre de 2009

PRIMEIRO SEMESTRE DE 2009						
ETAPAS	JANEIRO	FEVEREIRO	MARÇO	ABRIL	MAIO	JUNHO
Revisão da Documentação	X	X	X	X	X	X
Protótipo de Telas			X	X	X	
Implementação	X	X	X	X	X	X
Manual do Usuário		X	X	X		
Ambiente OLAP		X	X	X	X	
Revisão Bibliográfica	X	X	X	X	X	X
Apresentação do Projeto Final						X

Considerações Finais

O principal objetivo, durante o desenvolvimento deste trabalho, foi “não reinventar a roda” e sim reaproveitá-la, agregando mais funcionalidades a ela. Foram verificadas várias soluções existentes no mercado, que têm o mesmo propósito do Portal Acadêmico Innova, bem como, foi feita uma análise crítica para extrair o que se julgou ser o melhor que cada uma delas oferecia. A partir desta avaliação, foi possível levantar informações para a elaboração do plano de ação do trabalho, ou seja, o que deveria ser feito, como e quando fazê-lo.

A dificuldade estava em reunir as melhores idéias de diferentes ferramentas que têm o mesmo objetivo do Innova, como também, as opiniões de possíveis clientes e colocá-las em prática. De acordo com Grady Booch, James Rumbaugh e Ivar Jacobson, “*A escolha dos modelos a serem criados tem profunda influência sobre a maneira como um determinado problema é atacado ou uma solução é definida*” (2005).

Sendo assim, procurou-se utilizar as melhores práticas no que concerne ao desenvolvimento de um *software*, como por exemplo, a metodologia do processo unificado (PU) em conjunto com a linguagem UML (*Unified Modeling Language*), para realizar toda a estruturação das fases do desenvolvimento do trabalho e de sua documentação. Desse modo, o intuito foi experimentar, de certa forma, a realidade vivida em diversas empresas que fabricam *software* e também transformar um trabalho final de curso de graduação em um produto de alta qualidade.

Ao desenvolver este trabalho, o foco foi construir um produto que pudesse ser viável no mercado, além disso, utilizar a solução Innova como ponte para o processo de formação continuada, ou seja, o ingresso em cursos de especialização e mestrado.

O Innova tem uma infra-estrutura altamente portátil, isto é, passível de implementação em diferentes ambientes, plataformas. Esta portabilidade se dá em detrimento à utilização somente de ferramentas gratuitas, seguindo os padrões de interoperabilidade existentes no Brasil. Dessa forma, o Innova pode ser implementado a custos reduzidos, podendo ser disponibilizado para as diversas instituições acadêmicas. A solução ainda segue uma tendência do mercado de ferramentas *web*, que é oferecer um ambiente *clean* e objetivo, de fácil utilização e navegação.

Com o intuito de permitir uma facilidade de integração e customização, o sistema foi dividido em uma estrutura modular, a qual possibilita ao cliente a escolha pelos módulos que melhor se adequam ao seu negócio.

O módulo de Gerenciamento de Informações Acadêmicas vem para facilitar os principais processos que dizem respeito à comunicação entre alunos, professores, instituições de ensino e seus funcionários. Ainda colabora para auxiliar os níveis mais estratégicos no processo de tomada de

decisão.

O módulo de Mural de Estágios tem como principal meta agregar valor aos portais das instituições de ensino, buscando auxiliar seus usuários no processo de inserção no mercado de trabalho; serve como uma importante ferramenta na divulgação de vagas, na qual professores e profissionais das instituições de ensino podem disponibilizar informações sobre oportunidades de estágio, não apenas ao público próximo, mas também aos demais alunos; colaborando para o processo de integração entre as instituições de ensino e as empresas, estreitando sua relação.

Enquanto isso o módulo de Ensino à Distância, tem como objetivo integrar os portais das instituições de ensino com as ferramentas de EAD, visando agregar mais valor aos seus *websites*. Neste trabalho o Moodle foi o escolhido por uma série de razões previamente apresentadas.

Tendo em vista os diversos requisitos identificados ao longo do trabalho, uns foram priorizados em detrimento de outros, no intuito de cumprir o prazo determinado e com as principais funcionalidades do sistema implementadas. A seguir, são listadas algumas sugestões que são passíveis de implementação, com a finalidade de agregar mais valor ao produto no futuro.

- Integração com o *software* Redmine;
- Integração com a Plataforma *Lattes*;
- Expansão das consultas parametrizadas para abranger mais áreas, fornecendo mais dados estatísticos;
- Proteção contra múltiplas sessões de um único usuário;
- Permitir diversos tipos de parametrizações do sistema, para torná-lo mais flexível;
- Versão do sistema para dispositivos móveis (celulares, *smartphones*, etc);
- Permitir que os professores possam informar sua disponibilidade de horários através do sistema;

O Portal Acadêmico Innova, portanto, ainda pode ser aprofundado em uma pesquisa de pós-graduação, elevando seu nível de complexidade técnica, incrementando sua fundamentação teórica, que é o que uma pesquisa desse nível demanda.

Referências Bibliográficas

- AKITA, Fabio; Repensando a Web com Rails, Rio de Janeiro: Brasport, 2006.
- ANZANELLO, Cynthia; OLAP Conceitos e Utilização, Rio Grande do Sul: UFRGS, 2002.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2ª ed. Rio de Janeiro: Elsevier, 2005.
- DATE, C. J. Introdução a Sistemas de Banco de Dados. Rio de Janeiro: Campus, 2004.
- FERNANDEZ, Obie. Programando Rails – A Bíblia. Rio de Janeiro: Alta Books, 2008.
- FURTADO Jr., L. M., FIGUEIREDO, F. de A. Sistema de Transferência de Alunos – S.T.A. Orientador: George Hamilton Andrade Costa. Niterói: UNILASALLE RJ, 2007. Trabalho de Conclusão de Curso (Licenciatura Plena em Computação).
- GRÜNBLATT, Alice S., OLIVEIRA, Josane S. Sistema OLAP de Consultas Parametrizadas – SOLAPAD. Orientador: George Hamilton Andrade Costa. Niterói: UNILASALLE RJ, 2005. Trabalho de Conclusão de Curso (Licenciatura Plena em Computação).
- KOONTZ, H. e O'DONNEL, C. Os Princípios de Administração: Uma Análise das Funções Administrativas. São Paulo: Pioneira, 1980.
- LARMAN, Craig. Utilizando UML e padrões. 3ª ed. Porto Alegre: Bookman, 2004.
- LÜDKE, M. e ANDRE, M. E. D. A, Pesquisa em Educação: Abordagens Qualitativas. São Paulo: EPU, 2004.
- OLIVEIRA JUNIOR, Eustaquio Rangel. Ruby Conhecendo a Linguagem. Rio de Janeiro: Brasport, 2006.
- PINTO, A., NEY, T. Sistema de Gestão Acadêmica - Módulo de Auxílio ao Professor. Niterói: UNILASALLE-RJ, 2005. Monografia (Licenciatura Plena em Computação).
- TACHIZAWA, T. e ANDRADE, R. O. B. Gestão de Instituições de Ensino. Rio de Janeiro: Fundação Getúlio Vargas, 1999.
- TATE, Bruce A.; HIBBS, Curt. Ruby on Rails: Executando. Rio de Janeiro: Alta Books, 2008.
- THOMSEM, Erik: OLAP – Construindo sistemas de informações dimensionais, Rio de Janeiro: Campus, 2002.

Links

An Evaluation of Open Source E-Learning Platforms Stressing Adaption Issues – Women’s Postgraduate College of Internet Technologies – Vienna University of Technologies – Disponível em: www.campussource.de/aktuelles/docs/icalt2005.pdf – Acessado em 19/05/2009 às 22:32

GNU – GPL – Disponível em:

http://pt.wikipedia.org/wiki/GNU_General_Public_License – Acessado: 07/03/2009 às 12:40

História do Moodle – Disponível em:

http://docs.moodle.org/pt/Hist%C3%B3ria_do_Moodle – Acessado: 07/03/2009 às 14:43

Introdução a Arquitetura de Software – Fábrica de Software do CESUPA (Centro Universitário do Pará) – Disponível em:

www.fabsoft.cesupa.br/site/images/introducao_arquitetura_software.pdf – Acessado em 01/06/09 às 15:45

Livres de Bill Gates – Disponível em:

<http://www.softwarelivre.org/articles/44> – Acessado: 28/03/2009 às 15:21.

Moodle da Universidade Federal Fluminense – UFF - Disponível em:

<http://moodle.neami.uff.br/> - Acessado em: 07/03/2009 às 15:27

Moodle do Instituto Superior de Educação La Salle – Disponível em:

<http://www.unilasalle.info/moodle/> - Acessado em: 07/03/2009 às 15:51

O que é educação à distância – Disponível em:

http://umbu.ied.dcc.ufmg.br/moodle/file.php/11/Nivel_0/Conteudo/O_que_educacao_a_distancia.pdf - Acessado: 08/12/2008 às 20:00.

PMI – PROJECT MANAGEMENT INSTITUTE. A guide to the project management body of knowledge. Syba: PMI Publishing Division, 2000. Disponível em:

<http://www.pmi.org> – Acessado: 25/04/2009.

PMI – PROJECT MANAGEMENT INSTITUTE. Site oficial do PMI – Disponível em:

<http://www.pmi.org> – Acessado em 25/04/2009.

Portal do Git – Controle de versão de código-fonte – Disponível em:

<http://git-scm.com> – Acessado em: 13/02/2009 às 19:43

Portal do Governo Eletrônico – Disponível em:

<http://www.governoeletronico.gov.br> – Acessado: 25/04/2009 às 11:53

Portal de Software Livre no Brasil – Disponível em:

<http://www.softwarelivre.org>

Portal OLAP Report – Disponível em:

<http://www.olapreport.com>

Portal do Projeto GNU – Disponível em:

<http://www.gnu.org/> - Acessado: 20/04/2009 às 15:42

Portal do Redmine – Disponível em:

<http://www.redmine.org/wiki/redmine> – Acessado: 07/03/09 – 12:30

Uma comunidade – Disponível em:

<http://www.softwarelivre.org/articles/57> – Acessado: 28/03/2009 às 17:22.

Portal do Ubuntu – Disponível em:

<http://www.ubuntu.com>