



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Vitor Araujo Chane

**Macs - Sistema de controle de acessos do
Marvin**

Vitória, ES

2022

Vitor Araujo Chane

Macs - Sistema de controle de acessos do Marvin

Monografia apresentada ao Curso de Ciência da Computação do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Colegiado do Curso de Ciência da Computação

Orientador: Prof. Dr. Vitor Estevão Silva Souza

Vitória, ES

2022

Vitor Araujo Chane

Macs - Sistema de controle de acessos do Marvin/ Vitor Araujo Chane. –
Vitória, ES, 2022-

137 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Vitor Estevão Silva Souza

Monografia (PG) – Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Colegiado do Curso de Ciência da Computação, 2022.

1. Palavra-chave1. 2. Palavra-chave2. I. Souza, Vítor Estêvão Silva. II. Universidade Federal do Espírito Santo. IV. Macs - Sistema de controle de acessos do Marvin

CDU 02:141:005.7

Vitor Araujo Chane

Macs - Sistema de controle de acessos do Marvin

Monografia apresentada ao Curso de Ciência da Computação do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Vitória, ES, XX de Agosto de 2022:

Prof. Dr. Vitor Estevão Silva Souza
Orientador

Prof^a. Dr^a. Patrícia Dockhorn Costa
Universidade Federal do Espírito Santo

Pedro Henrique Brunoro Hoppe
Universidade Federal do Espírito Santo

Vitória, ES
2022

Dedico este trabalho a três pessoas queridas que se foram durante estes anos em que estive cursando a graduação. Meus avós Palmira Martins Nogueira (in memoriam) e Francisco Timóteo de Araujo (in memoriam), pelos ensinamentos, preocupação e zelo com toda a família, e ao meu pai Cesar Luiz Chane (in memoriam), por toda a dedicação e esforço na criação e educação dos filhos. A presença de todos vocês foi algo essencial na minha vida.

Agradecimentos

Agradecer primeiramente a minha mãe Rosa e a meu pai Cesar, por todo o esforço investido na minha educação e por sempre estarem ao meu lado, me apoiando e não deixando eu desistir ao longo de toda a minha trajetória. Agradeço também a meu irmão Guilherme e a todos meus familiares que sempre estiveram torcendo por mim.

Aos amigos que fiz durante a graduação, que compartilharam dos inúmeros desafios que enfrentamos, sempre com o espírito colaborativo e com muitas risadas.

Ao meu orientador e xará professor Vitor E. Silva Souza, por aceitar conduzir o meu trabalho de pesquisa. Tenho certeza absoluta que se não fosse pelo seu apoio, eu não teria conseguido.

E por fim, quero agradecer à Universidade Federal do Espírito Santo e a todos os docentes que de alguma maneira fizeram parte do meu aprendizado, pela elevada qualidade do ensino oferecido.

Resumo

Um sistema onde alunos, professores e funcionários possam gerenciar tarefas de ensino e pesquisa em uma universidade se tornou algo indispensável nos dias atuais. Com isso em mente e com a crescente expansão do mercado Web, surgiu o sistema Marvin, feito e mantido por alunos e professores da Universidade Federal do Espírito Santo (UFES).

A ideia do Marvin é servir a todos os departamentos de toda a universidade, e para isso, devemos ter em mente que existirão vários usuários e nem sempre seus papéis dentro deste sistema será o mesmo. Em um sistema universitário que oferece várias funcionalidades, é muito importante estar bem definido qual acesso e qual papel cada usuário terá dentro do sistema. Até por uma questão de segurança, um aluno não deve ter os mesmos “privilégios” que um professor, já que isso poderia acarretar em alunos ditando suas próprias notas finais em um disciplina, por exemplo. Com isso em mente, uma parte muito importante do sistema final é o módulo de controle de acessos, que este trabalho propõe com o *Macs - Sistema de controle de acessos do Marvin*

O objetivo geral deste trabalho de conclusão de curso é aplicar o método FrameWeb para o desenvolvimento do módulo *Macs - Sistema de controle de acessos do Marvin* a partir de requisitos, levantados no contexto do projeto, e das tecnologias utilizadas. Por fim, espera-se que o módulo implementado sirva de auxílio nas atividades de ensino, pesquisa e tarefas administrativas nos departamentos, cursos e programas existentes na UFES.

Palavras-chaves: Marvin. Java. FrameWeb.

Lista de ilustrações

Figura 1 – O Processo da Engenharia de Requisitos (KOTONYA; SOMMERVILLE, 1998).	18
Figura 2 – Arquitetura padrão proposta pelo FrameWeb, baseado no <i>Service Layer</i> (CAMPOS; SOUZA, 2017)	22
Figura 3 – Diagrama de Casos de Uso do administrador do sistema.	29
Figura 4 – Diagrama de Casos de Uso do subsistema Departamento.	30
Figura 5 – Diagrama de Casos de Uso do subsistema Graduação.	31
Figura 6 – Diagrama de Casos de Uso do subsistema PPG.	32
Figura 7 – Diagrama de classes do subsistema Departamento.	34
Figura 8 – Diagrama de classes do subsistema Graduação.	35
Figura 9 – Diagrama de classes do subsistema PPG.	36
Figura 10 – Subsistemas do <i>Macs - Sistema de controle de acessos do Marvin</i>	39
Figura 11 – Arquitetura do módulo <i>Macs - Sistema de controle de acessos do Marvin</i>	39
Figura 12 – Pacotes do <i>Macs - Sistema de controle de acessos do Marvin</i>	40
Figura 13 – Páginas web do <i>Macs - Sistema de controle de acessos do Marvin</i>	41
Figura 14 – Modelo de Entidades do subsistema Departamento	42
Figura 15 – Modelo de Entidades do subsistema Graduação	43
Figura 16 – Modelo de Entidades do subsistema PPG	44
Figura 17 – Modelo de Aplicação genérico	45
Figura 18 – Modelo de aplicação do subsistema Departamento	46
Figura 19 – Modelo de aplicação do subsistema Graduação	47
Figura 20 – Modelo de aplicação do subsistema PPG	48
Figura 21 – Modelo de Persistência comum aos DAOs do sistema	49
Figura 22 – Modelo de Persistência do subsistema Departamento	50
Figura 23 – Modelo de Persistência do subsistema Graduação	51
Figura 24 – Modelo de Persistência do subsistema PPG	52
Figura 25 – Modelo de Navegação de um CRUD	54
Figura 26 – Modelo de Navegação de uma indicação	55
Figura 27 – Marvin - Tela de Login	56
Figura 28 – Marvin - Página Inicial	57
Figura 29 – Marvin - Página Inicial - Versão Mobile	57
Figura 30 – Marvin - Lista de Acadêmicos	58
Figura 31 – Marvin - Página de Criação de Novo Acadêmico	58
Figura 32 – Marvin - Página de Visualização de Acadêmico	59
Figura 33 – Marvin - Página de Alteração de Acadêmico	60
Figura 34 – Marvin - Página de Exclusão de Acadêmico	60

Figura 35 – Marvin - Consulta de Acadêmico	61
Figura 36 – Marvin - Listagem de Centros de Ensino	61
Figura 37 – Marvin - Criação de Novo Centro de Ensino	61
Figura 38 – Marvin - Listagem de Departamentos	62
Figura 39 – Marvin - Criação de Novo Departamento	62
Figura 40 – Marvin - Listagem de Chefias de Departamento	62
Figura 41 – Marvin - Listagem de Chefias de Departamento - Versão Mobile	63
Figura 42 – Marvin - Criação de Nova Chefia de Departamento	64
Figura 43 – Marvin - Criação de Nova Chefia de Departamento - Versão Mobile	64
Figura 44 – Marvin - Página inicial - Acesso de Chefe do Departamento	65
Figura 45 – Marvin - Listagem de Chefes do Departamento	66
Figura 46 – Marvin - Listagem de Chefes do Departamento - Versão mobile	66
Figura 47 – Marvin - Formulário de indicação de Chefes do Departamento	67
Figura 48 – Marvin - Formulário de indicação de Chefes do Departamento - Versão mobile	68
Figura 49 – Marvin - Listagem de Chefes do Departamento - Finalização de Chefia	68

Lista de tabelas

Tabela 1 – Exemplos de estórias de usuário.	26
Tabela 2 – Descrição dos atores envolvidos nos casos de uso.	27
Tabela 3 – Plataforma de Desenvolvimento e Tecnologias Utilizadas.	37
Tabela 4 – Softwares de Apoio ao Desenvolvimento do Projeto	38

Lista de abreviaturas e siglas

API	Application Program Interface
CRUD	Create, Read, Update and Retrieve
UML	Unified Modeling Language
PPG	Programa de pós-graduação
CDI	Context and Dependency Injection
DAO	Data Access Object
IDE	Integrated Development Environment
JPA	Java Persistence API
JSF	Java Server Faces
ORM	Object-Relational Mapping
SQL	Structured Query Language
XHTML	EXtensible HyperText Markup

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	13
1.2	Metodologia de Desenvolvimento do Trabalho	14
1.3	Organização da Monografia	15
2	REFERENCIAL TEÓRICO	16
2.1	Engenharia de Software	16
2.1.1	Análise e Especificação de Requisitos	17
2.1.2	Projeto e Implementação de Sistemas	19
2.2	Engenharia Web	20
2.3	Tecnologias e Ferramentas	20
2.4	O Método FrameWeb	22
3	ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS	24
3.1	Descrição do Escopo	24
3.2	Estórias de Usuário	25
3.3	Modelos de Caso de Uso	27
3.3.1	Subsistema Departamento	28
3.3.2	Subsistema Graduação	30
3.3.3	Subsistema PPG	31
3.4	Diagrama de Classes	33
3.4.1	Subsistema Departamento	33
3.4.2	Subsistema Graduação	33
3.4.3	Subsistema PPG	34
4	PROJETO DO SISTEMA	37
4.1	Tecnologias e Ferramentas Utilizadas	37
4.2	Arquitetura de Software	38
4.3	Modelagem FrameWeb	41
4.3.1	Camada de Negócio	41
4.3.2	Camada de Acesso a Dados	49
4.3.3	Camada de Apresentação	53
5	IMPLEMENTAÇÃO DO SISTEMA	56
6	CONCLUSÃO	69
6.1	Considerações Finais	69

6.2	Trabalhos Futuros	70
	REFERÊNCIAS	72
	APÊNDICES	73

1 Introdução

No contexto de sistemas de informação de apoio administrativo, é interessante que os diferentes órgãos (cursos, departamentos, programas, etc.) da Universidade Federal do Espírito Santo (UFES) não fiquem limitados apenas aos sistemas que são desenvolvidos e suportados pela Superintendência de Tecnologia da Informação (STI) e possibilitem que alunos e docentes participem na criação de sistemas complementares. Com esse pretexto, foi criado um projeto de extensão¹ com o objetivo de construir as bases de um sistema Web que ofertasse funcionalidades que os atuais sistemas da UFES ainda não oferecem.

O Marvin² é um sistema Web desenvolvido e mantido por discentes e docentes dos cursos de Ciência da Computação e Engenharia de Computação da UFES, com o intuito de auxiliar nas atividades de ensino, pesquisa e nas tarefas administrativas mais comuns no contexto da UFES que não são cobertas pelos atuais sistemas. Alunos utilizam de suas experiências adquiridas durante o curso de graduação para desenvolver módulos do Marvin como parte de seus projetos de graduação, no qual cada módulo possui uma funcionalidade única que, no fim, dentro do contexto do projeto de extensão, é integrado para formar um sistema único e com interface amigável.

Em um sistema universitário que oferece várias funcionalidades, é muito importante estar bem definido qual acesso e qual papel cada usuário terá dentro do sistema. Até por uma questão de segurança, um aluno não deve ter os mesmos “privilégios” que um professor, já que isso poderia acarretar em alunos ditando suas próprias notas finais em um disciplina, por exemplo. Com isso em mente, uma parte muito importante do sistema final é o módulo de controle de acessos, que este trabalho propõe com o módulo Macs (*Marvin Access Control System*).

1.1 Objetivos

O objetivo geral deste trabalho de conclusão de curso é aplicar o método FrameWeb para o desenvolvimento do módulo *Macs - Sistema de controle de acessos do Marvin* a partir dos requisitos levantados e das tecnologias utilizadas. Por fim, espera-se que o módulo implementado sirva de auxílio nas atividades de ensino, pesquisa e tarefas administrativas nos departamentos, cursos e programas existentes na UFES. Como explicado anteriormente, definir quais acessos que cada tipo de usuário terá em um sistema universitário é de suma importância para o funcionamento e segurança do mesmo.

¹ <<https://projetos.ufes.br/#/projetos/523/informacoes>>

² <<https://github.com/dwvs-ufes/marvin/>>

São objetivos específicos deste trabalho:

- Apresentar a documentação levantada, as tecnologias utilizadas e os critérios que foram escolhidos para implementação do sistema utilizando o método FrameWeb (SOUZA, 2020);
- Ter o próprio módulo *Macs - Sistema de controle de acessos do Marvin* implementado, devidamente testado e integrado ao Marvin,³ de modo que possa ser disponibilizado para uso geral;
- Exercitar os conceitos aprendidos ao longo do curso de Ciência da Computação, em particular nas disciplinas de Engenharia de Software, Engenharia de Requisitos, Projeto de Sistemas de Software, Programação III, Banco de Dados e Desenvolvimento Web.

1.2 Metodologia de Desenvolvimento do Trabalho

A metodologia utilizada para desenvolver este trabalho é composta pelas seguintes atividades:

- Revisão Bibliográfica: estudo do processo correto e de boas práticas da Engenharia de Software, tal como o estudo da Engenharia Web e de Requisitos, do método FrameWeb (SOUZA, 2020), da programação orientada a objetos com a linguagem Java, assim como o estudo das ferramentas e tecnologias necessárias para desenvolver o módulo;
- Elaboração do Referencial Teórico: nesta etapa, foram de suma importância os conhecimentos adquiridos em anos de curso para a análise e a elaboração dos documentos de especificação de requisitos, apresentando uma visão geral do que foi modelado no projeto, contendo estórias de usuário, requisitos não-funcionais, diagramas de caso de uso e também a apresentação do sistema modelado em UML. Por fim, também foi desenvolvido o documento de projeto do sistema, contendo a arquitetura do módulo e de seus componentes baseados na abordagem FrameWeb (SOUZA, 2020) e nas tecnologias utilizadas para o desenvolvimento;
- Implementação: codificação do módulo utilizando as tecnologias propostas;
- Testes e Ajustes: realização de testes de funcionalidade e ajustes de interface com o objetivo de melhorar a usabilidade e experiência do usuário dentro do sistema;
- Entrega: apresentação dos resultados, assim como possíveis melhorias futuras.

³ <<https://github.com/dwvs-ufes/marvin/>>

1.3 Organização da Monografia

Além desta introdução, esta monografia é composta por outros cinco capítulos:

- O Capítulo 2 apresenta os aspectos relativos ao conteúdo teórico relevante para o trabalho;
- O Capítulo 3 Apresenta os requisitos levantados e especificados no documento de requisitos que serviram de apoio para o desenvolvimento do sistema;
- O Capítulo 4 Apresenta o projeto arquitetural e as tecnologias utilizadas no desenvolvimento do sistema;
- O Capítulo 5 Apresenta as telas do sistema, mostrando as funcionalidades desenvolvidas;
- O Capítulo 6 apresenta as considerações finais do trabalho.

2 Referencial Teórico

Este capítulo apresenta a base de conceitos que fundamentaram e tornaram possíveis o desenvolvimento do módulo *Macs - Sistema de controle de acessos do Marvin*, e está organizado da seguinte forma: a Seção 2.1 apresenta os conceitos da Engenharia de Software, divididos em duas subseções; a Seção 2.2 apresenta os conceitos da Engenharia Web; a Seção 2.3 apresenta uma breve descrição das principais tecnologias e ferramentas utilizadas para a produção deste trabalho; por fim, a Seção 2.4 apresenta o método FrameWeb.

2.1 Engenharia de Software

O desenvolvimento de software é uma atividade muito importante nos dias de hoje. Todos nós, desenvolvedores, queremos construir softwares que atendam às necessidades dos usuários, operando perfeitamente durante um longo período de tempo, com um alto grau de manutenibilidade e que seja intuitivo ao uso. Porém, frequentemente, nos deparamos com softwares ruins, propensos a muitos erros, usuários insatisfeitos e uma dificuldade imensa na manutenção e utilização dos mesmos.

Para evitar a falta de qualidade dos softwares e aumentar a produtividade no processo de desenvolvimento dos mesmos, surgiu a Engenharia de Software. De acordo com Falbo (2014), a Engenharia de Software é a área da Ciência da Computação voltada à especificação, desenvolvimento e manutenção de sistemas de software, com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade no processo de software. A Engenharia de Software trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software (FALBO, 2014).

A Engenharia de Software determina que se tenha fases de processo e desenvolvimento do produto, nas quais se deve dividir o problema em partes menores, e que são determinadas por algum modelo de processo estabelecido, como por exemplos os modelos sequenciais, evolutivos, incrementais, etc. Não existe um modelo de processo ideal, sendo que cada projeto deve ter uma análise prévia para se estabelecer o modelo que melhor pode atendê-lo, mas pode-se observar que todos esses modelos possuem algumas fases básicas, sendo elas: Análise e Especificação de Requisitos, Projeto e Implementação, Testes e Entrega (SOMMERVILLE, 2011).

Esta seção sobre Engenharia de Software é uma introdução e possui as seguintes subseções referentes ao processo de desenvolvimento de software: A Seção 2.1.1 apresenta o processo de Análise e Especificação de Requisitos; e a Seção 2.1.2 apresenta o conceito

de Projeto e Implementação de Sistemas.

2.1.1 Análise e Especificação de Requisitos

Desenvolver um software, por mais simples que seja, não é uma tarefa fácil. Essa atividade requer calma e paciência para se lidar com pessoas que estão envolvidas no desenvolvimento, desde o cliente que solicita as funcionalidades que o sistema deve prover, até a equipe de desenvolvimento, que nem sempre concorda com a maneira que certa funcionalidade está sendo desenvolvida. É de suma importância estabelecer uma base de concordância entre o cliente e o fornecedor, para que as funcionalidades do software fiquem bem claras, e isso requer um processo bem definido de levantamento e análise de requisitos.

Os requisitos nada mais são do que as necessidades explicitadas pelos clientes e as restrições que o sistema deve ter para satisfazê-las. Em outras palavras, os requisitos definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar (SOMMERVILLE, 2011).

Requisitos podem ser divididos em três grupos: requisitos funcionais, que são as funcionalidades que o sistema deve prover; requisitos não-funcionais, que são restrições sobre os requisitos funcionais, que o sistema deve ter para funcionar corretamente; e os requisitos de domínio (também conhecidos como regras de negócio) que são provenientes do domínio de aplicação do sistema e refletem características e restrições desse domínio (FALBO, 2014).

O ciclo de vida do processo de Engenharia de Requisitos, de acordo com a Figura 1, inicia-se com a fase de levantamento de requisitos, que combina elementos de solução de problemas, elaboração, negociação e especificação (PRESSMAN; MAXIM, 2016). Se deve compreender o domínio da aplicação, os processos de negócio a serem apoiados pelo sistema, os problemas que se pretende resolver e as necessidades e restrições dos envolvidos (FALBO, 2014). No fim, espera-se ter o problema identificado, propostas de soluções para o problema e um conjunto preliminar de requisitos da solução documentados em um documento de requisitos.

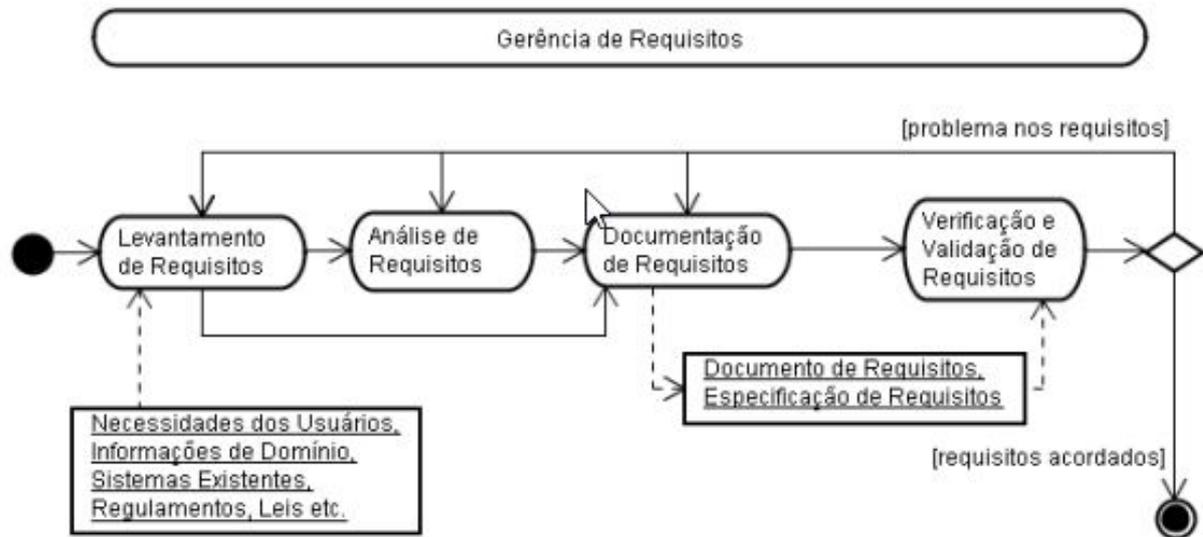


Figura 1 – O Processo da Engenharia de Requisitos (KOTONYA; SOMMERVILLE, 1998).

Após os requisitos serem levantados, é preciso garantir que o sistema tenha condições de tratá-los a partir de uma série de pontos de vista. A modelagem de sistemas possibilita o estudo, de uma forma abstrata, dos pontos fortes e fracos, fazendo com que previamente se levante a necessidade de modificações ou correções para o funcionamento correto do sistema. De acordo com Falbo (2014), a fase de análise de requisitos é, em última instância, uma fase de construção de modelos.

Para cada situação, diferentes tipos de modelos são necessários e, de acordo com Pressman e Maxim (2016), a modelagem de requisitos resulta em um ou mais dos seguintes tipos de modelos:

- Modelos baseados em cenários de requisitos, como os modelos de casos de uso, nos quais o objetivo é representar um requisito de software sob a ótica de um usuário (ator);
- Modelos orientados a classes que representam classes orientadas a objetos (atributos e operações) e a maneira como as classes colaboram para atender aos requisitos do sistema;
- Modelos comportamentais e baseados em padrões que representam como o software se comporta em consequência de “eventos” externos;
- Modelos de dados que representam o domínio de informações para o problema;
- Modelos orientados a fluxos que representam os elementos funcionais do sistema e como eles transformam os dados à medida que se movem pelo sistema.

Por fim, os modelos de requisitos e a especificação de requisitos dão ao projetista

de software informações que podem ser transformadas em projetos de arquitetura, de interface e de componentes (PRESSMAN; MAXIM, 2016).

2.1.2 Projeto e Implementação de Sistemas

Com os requisitos levantados, analisados e modelados, começamos a pensar na melhor maneira de implementar o sistema. Para isso, é necessário identificar os componentes de software e seus relacionamentos, respeitando os requisitos especificados, e utilizar as tecnologias mais adequadas. A partir disso, são gerados modelos de projeto que servirão de guia para o que será codificado e implementado.

É importante que esses guias proporcionem um quadro completo do software, e que sejam de fácil entendimento para as pessoas que irão codificá-lo, testá-lo e mantê-lo (FALBO, 2014).

De acordo com Falbo (2014), independente do ciclo de vida e dos paradigmas adotados, a partir dos modelos de requisitos, o projeto deve produzir:

- **Projeto de Arquitetura de Software:** definição da relação entre os elementos estruturais, assim como o estilo da arquitetura e os padrões que o projeto irá atender;
- **Projeto de Dados:** transforma os modelos de classes em realizações de classes de projeto e nas estruturas de dados dos requisitos necessárias para implementar o software, que servirão de base para a realização do projeto (PRESSMAN; MAXIM, 2016);
- **Projeto de Interfaces:** descreve como deverá ser a comunicação do software com outros sistemas (interfaces externas), com os usuários que irão o utilizar (interface com o usuário), e com ele mesmo (interfaces internas);
- **Projeto Detalhado:** tem por objetivo refinar e detalhar a descrição dos componentes estruturais da arquitetura do software.

Ao final da fase de projeto, espera-se que tenhamos um padrão arquitetônico bem definido, assim como os componentes, que devem estar refinados e detalhados para que possam posteriormente ser codificados e testados. Essas camadas são mostradas com mais detalhes na Seção 2.4.

Na etapa de implementação, o projeto é codificado de acordo com a descrição do projeto a partir de uma linguagem de programação. Eventualmente é gerado o primeiro executável onde o projeto pode ser compilado e testado.

2.2 Engenharia Web

Graças aos avanços das tecnologias Web, da Internet e da própria *World Wide Web*, hoje vivemos em um mundo estritamente conectado, onde coisas que há poucos anos pareciam inimagináveis, atualmente são comuns aos nossos olhos. Novas aplicações baseadas na Web são lançadas todos os dias, em diferentes setores da economia, com o intuito de criar respostas para as necessidades presentes no mercado e na sociedade contemporânea. O que se viu por muitos anos era que o desenvolvimento de sistema baseado na Web careciam de rigor, abordagens sistemáticas e garantia de qualidade (MURUGESAN et al., 2000).

À medida que a complexidade e a sofisticação das aplicações baseadas na Web crescem, cresce também uma preocupação sobre a maneira que estão sendo criadas e se torna necessário o uso de métodos da Engenharia de Software aplicados ao desenvolvimento Web, para que as aplicações mantenham um alto padrão de qualidade e integridade ao longo do tempo.

A Engenharia Web nasceu como sub-área da Engenharia de Software, sendo o estabelecimento e uso de conhecimentos científicos, de engenharia, de gestão e abordagens disciplinadas e sistemáticas para o desenvolvimento, implantação e manutenção de sistemas e aplicações de alta qualidade baseadas na Web (MURUGESAN et al., 2000). A Engenharia Web é multidisciplinar e abrange diversos princípios de diversas áreas, como interface humano-computador, gerenciamento de projetos, banco de dados, segurança da informação, comunicação social, redes, etc.

Mesmo com as diferenças de softwares convencionais e softwares de aplicações Web, se percebe uma similaridade no modo com que são divididas as fases de processo e desenvolvimento dos softwares. Murugesan et al. (2000) diz que, como nos modelos de processo da Engenharia de Software, há algumas similaridades para a Engenharia Web, a saber: necessidade de métodos, levantamento de requisitos, implementação, testes e manutenção das partes que lidam com programação e funcionalidades. O que se percebe é que o processo de desenvolvimento de um software Web é muito semelhante na parte de processos se comparado a um software convencional.

2.3 Tecnologias e Ferramentas

Para o desenvolvimento do *Macs - Sistema de controle de acessos do Marvin*, foi utilizada a IDE Eclipse, que é um ambiente de desenvolvimento *open source* para várias linguagens de programação como Java, C/C++, PHP, Python, etc. por meio de *plugins* que podem ser obtidos e integrados ao ambiente. O principal trunfo utilizando o Eclipse é a redução da complexidade de desenvolvimento, implantação e gerenciamento das aplicações

Web a partir de seus componentes de infraestrutura prontos para o uso.

Junto ao Eclipse, a linguagem de programação escolhida para escrever as classes do módulo do projeto foi [Java™ 13](#), junto da plataforma [Java Enterprise Edition \(Java EE\) 8](#). Java é uma linguagem de programação orientada a objetos pertencente à Oracle e distribuída com um vasto conjunto de bibliotecas e APIs que podem ser usadas nos projetos. Java EE 8 é uma plataforma de programação que oferece suporte a diversos *frameworks* e APIs. Desta plataforma, utilizamos para o desenvolvimento do módulo as APIs JSF, CDI, JPA e Facelets.

O [JSF \(JavaServer Faces\)](#) é uma API que permite criar aplicações Java para Web utilizando componentes visuais pré-prontos que facilitam significativamente a tarefa de escrever e manter as aplicações que são executadas em um servidor de aplicações Java e renderizar as interfaces de usuário de volta a um cliente.

O [CDI \(Context and Dependency Injection\)](#) é parte integrante do Java EE e fornece uma arquitetura que permite aos componentes da plataforma, como os servlets e EJB's (*Enterprise Java Beans*), existirem dentro do ciclo de vida de uma aplicação com escopos bem definidos, associando-os a contextos e oferecendo uma série de serviços de infraestrutura à aplicação. O CDI também oferece um mecanismo de injeção de dependências entre classes de uma aplicação Java EE, que serão associadas a contextos e possuem um ciclo de vida gerenciado por um *container*. Com isso, o CDI oferece funcionalidades como decoradores, interceptadores e qualificadores, que flexibilizam o desenvolvimento de uma aplicação Java EE.

O [JPA \(Java Persistence API\)](#) é um *framework* para persistência de dados que possui uma API de mapeamento ORM (*Object/Relational Mapping* ou Mapeamento Objeto/Relacional). Entre as vantagens, podemos citar ([NASCIMENTO, 2015](#)):

- Códigos de acesso a banco de dados com queries SQL são custosos de se desenvolver. JPA melhora a produtividade, eliminando o trabalho e deixando o desenvolvedor se concentrar na lógica de negócio;
- A manutenibilidade de sistemas desenvolvidos com ORM é excelente, pois o mecanismo faz com que menos linhas de código sejam necessárias;
- Com JPA, você pode desenvolver um sistema usando um banco de dados e colocá-lo em produção usando diversos outros banco de dados, sem precisar alterar códigos-fontes para adequar sintaxe de queries que só funcionam em SGBDs de determinados fornecedores.

Para estilização, foi utilizada uma biblioteca de interfaces pré-prontas chamada [PrimeFaces](#). O seu objetivo é otimizar o tempo gasto com desenvolvimento de componentes

e estilizações como botões, tabelas, páginas responsivas e etc. entregando uma interface robusta e amigável para o usuário final.

Para simplificar a utilização de *frameworks* como JPA, CDI e JSF, foi utilizada a ferramenta **JButler**, desenvolvida na Universidade Federal do Espírito Santo (UFES) que fornece ao usuário facilidades às funcionalidades CRUD (*create, retrieve, update, delete*), implementando as operações básicas entre a aplicação e o banco de dados.

Para nos fornecer as implementações de APIs e para hospedar as aplicações Web, usamos o **WildFly**, que é um servidor de aplicação *open source* baseado na plataforma Java EE e implementado inteiramente na linguagem Java.

Para o banco de dados foi utilizado o **MySQL Server 8**, que é um sistema de gerenciamento de banco de dados relacionais que utiliza a linguagem SQL como interface. Nele, teremos guardados todos os usuários do sistema e principalmente os papéis atribuídos a cada um. Os papéis são os pontos críticos deste módulo, pois são eles que possibilitarão a cada usuário acessar áreas restritas dentro do sistema.

2.4 O Método FrameWeb

Poupar tempo com tarefas corriqueiras é algo almejado por programadores desde o princípio da programação. Com isso em mente, foram desenvolvidos padrões arquiteturais de software e *frameworks* capazes de auxiliar no reúso de código e no aumento da produtividade.

O FrameWeb é um método de projeto de WISs (*Web-based Information Systems*) baseado em *frameworks* (SOUZA, 2020). Nele, é definida uma arquitetura baseada no padrão arquitetônico Camada de Serviço (*Service layer*) (FOWLER, 2002) utilizando esses *frameworks*. A organização das camadas desta arquitetura é mostrada na Figura 2.

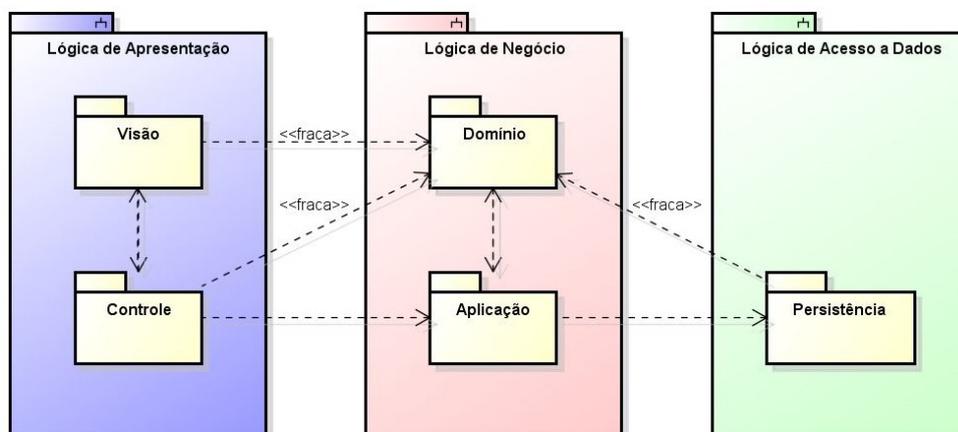


Figura 2 – Arquitetura padrão proposta pelo FrameWeb, baseado no *Service Layer* (CAMPOS; SOUZA, 2017)

De acordo com Souza (2020), a arquitetura é composta das seguintes camadas:

- **Camada de Apresentação** ou *Presentation Tier*: responsável pela interação com o usuário, é dividida em dois pacotes: **Visão** (*View*), responsável por conter os artefatos de interface com o usuário, como páginas, scripts e leiautes; e **Controle** (*Control*), contendo as classes de controle que manipulam as solicitações feitas pelo pacote de Visão, e também responsável pela comunicação com o pacote de Aplicação (*Application*), pertencente à camada de negócio;
- **Camada de Negócio** ou *Business Tier*: responsável pela lógica de negócio, assim como a camada de apresentação, é dividida em dois pacotes: pacote de **Aplicação** (*Application*), que contém as classes responsáveis pelas funcionalidades do sistema; e o pacote de **Domínio** (*Domain*), que contém as classes que representam o domínio do problema, identificadas na fase de análise de requisitos;
- **Camada de Acesso de Dados** ou *Data Access Tier*: com apenas o pacote de **Persistência** (*Persistence*), é responsável pela persistência de dados, como por exemplo, compartilhamento de arquivos e acesso a bancos de dados.

Com base nesta arquitetura, o FrameWeb define quatro tipos de diagramas baseados no Diagrama de Classes da UML, que demonstram especificamente os componentes de cada camada. São eles (CAMPOS; SOUZA, 2017):

- **Modelo de Entidades** (*Entity Model*): representa os objetos de domínio do problema e sua persistência de dados;
- **Modelo de Persistencia** (*Persistence Model*): representa a camada de acesso a dados, seguindo o padrão de projeto *Data Access Object* (DAO) (ALUR; CRUPI; MALKS, 2003), responsáveis pela persistência das classes de domínio;
- **Modelo de Navegação** (*Navigation Model*): representam o conjunto de componentes que formam a camada responsável pela interação com o usuário, como as páginas, scripts e os controladores;
- **Modelo de Aplicação** (*Application Model*): representa as classes de serviço responsáveis pela implementação das funcionalidades do WIS, e suas dependências.

3 Análise e Especificação de Requisitos

Este capítulo tem como objetivo descrever o escopo do *Macs - Sistema de controle de acessos do Marvin* e apresentar o que foi levantado na fase de Análise e Especificação de Requisitos, como por exemplo: histórias de usuário, modelos de caso de uso e diagramas de classes da UML.

3.1 Descrição do Escopo

Possibilitar o gerenciamento dos papéis que cada usuário terá dentro do Marvin, garantindo que cada um possua acesso a funcionalidades específicas de acordo com seu papel, é algo indispensável. O módulo *Macs - Sistema de controle de acessos do Marvin* tem por objetivo suprir esta necessidade.

A partir de papéis básicos já definidos no Marvin — a saber, Professor, Servidor Técnico-Administrativo, Estudante —, este novo módulo englobará vários outros papéis associados a unidades organizacionais específicas. Por exemplo: um professor poderá ter o papel de Chefe do Departamento. Um estudante poderá ter o papel de Representante Discente, mas nunca poderá ter o papel Chefe do Departamento. Analogamente, um usuário com papel Professor nunca poderá ter o papel Representante Discente, por ser um papel específico de um Estudante.

Apenas o administrador do sistema e usuários com cargos hierárquicos de topo como secretários e chefes/coordenadores podem conceder outros papéis a um usuário do sistema, de modo que seja respeitado o que cada usuário pode acessar. No final, espera-se que estes acessos sejam concedidos de maneira simples e evitando qualquer tipo de erros.

Podemos descrever o uso do módulo *Macs - Sistema de controle de acessos do Marvin* por meio dos passos a seguir.

A princípio, o administrador do sistema (definido no Marvin pelo papel *SysAdmin*) será responsável por fazer cadastros gerais de todas os objetos que compõem o sistema. Primeiramente, são cadastradas as unidades administrativas, como centros de ensino (ex.: Centro Tecnológico), departamentos (ex.: Departamento de Informática), cursos de graduação (ex.: curso de Ciência da Computação), programas de pós-graduação (PPG) (ex.: Programa de Pós-Graduação em Informática) e cursos de pós-graduação (ex.: Mestrado em Informática). Departamentos, PPGs e cursos de graduação são vinculados ao centro, já os cursos de pós-graduação são vinculados a um PPG.

Após a criação das organizações administrativas, o administrador começará a popular esses órgãos com acadêmicos. Para acessar o sistema, esses acadêmicos terão que

ser cadastrados informando: nome, e-mail, CPF, data de nascimento e senha. Também serão associados a um ou mais papéis dentre os papéis existentes.

Criado o departamento, podem ser associados a ele os seguintes papéis específicos: chefe e subchefe (exclusivo para professores(as)), secretários(as) (servidores(as) técnico-administrativos(as)), representantes discentes (estudantes) e professores(as) ou servidores(as) lotados(as) no departamento. Em todos os casos, o sistema deve registrar a data de início da relação do acadêmico com o departamento, bem como a data final para que seja mantido o histórico.

O curso de graduação deverá ter um coordenador(a) e um subcoordenador(a) (professores(as)), secretários(as) (servidores(as) técnico-administrativos(as)), representantes discentes (estudantes), um(a) coordenador(a) de estágio (professores(as)) e professores(as) ou servidores(as) que compõem o colegiado do curso. Em todos esses cadastros, o sistema deve registrar a data de início da relação do acadêmico com o curso de graduação, bem como a data final para que seja mantido o histórico. Também deve ser possível que estudantes sejam matriculados nos cursos em questão, informadas a matrícula e a data de início do estudante, sendo que também deve ser informado a data de finalização do vínculo em casos de desistência, abandono ou conclusão do curso.

Na pós-graduação ocorre um processo similar, com a diferença de que existe um programa de pós-graduação (PPG) entre o centro de ensino e o curso. Como um curso de graduação, um PPG poderá também ter coordenadores(as) e subcoordenadores(as), secretários(as), representantes discentes e professores(as) membros de seu colegiado. Além disso, há a possibilidade de professores(as) serem membros do PPG mas não do colegiado. Por fim, cursos de pós-graduação podem ter estudantes matriculados(as) como no caso de cursos de graduação.

Após os acadêmicos serem cadastrados pelo administrador do sistema, estes (dependendo de seu papel e cargo) tomarão posse da responsabilidade de conceder e revogar acesso a outros acadêmicos dentro dos subsistemas informados. Por exemplo, um chefe de departamento pode indicar que sua chefia está terminando e um(a) colega está assumindo a chefia em seu lugar a partir daquele momento.

3.2 Estórias de Usuário

Uma estória de usuário (*User Story*) é uma explicação informal e geral sobre um recurso de software escrita a partir da perspectiva do usuário final. Considerando como principais *stakeholders*, professores, estudantes, técnicos administrativos e visitantes, foram identificados vários requisitos de usuário e regras de negócio que são apresentadas no Documento de Requisitos anexo no apêndice deste documento.

Para título de exemplo, como a maioria dos casos de uso deste trabalho possuem certa semelhança por se tratarem de vários CRUD's e funcionalidades relacionadas a indicação de acadêmicos a um cargo, são demonstradas na Tabela 1 as estórias de usuário para dois exemplos: um registro de um chefe de departamento pelo administrador do sistema e uma indicação de um acadêmico a um cargo de chefia de um departamento por um outro acadêmico apto a fazer isso.

Tabela 1 – Exemplos de estórias de usuário.

Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) em cargos de chefia do departamento, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.
Crítérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema, deve ter o papel <i>Professor</i> e deve estar lotado no departamento em questão; – CA2: Para cadastrar, devem ser informados: acadêmico(a), departamento, cargo, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), departamento, cargo ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões.
Descrição:	Como chefe/sub-chefe ou secretário(a) de um departamento, eu quero registrar substituição de chefia do departamento, para que os(as) novos(as) chefes possam ter acesso a funcionalidades específicas deste papel.

Crítérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) lotados(as) no departamento em questão e com papel <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) chefes e sub-chefes atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a), cargo ou período de atuação; – CA4: Para terminar a atuação de um chefe ou sub-chefe, basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar uma nova chefia, escolhe se a indicação é de chefe ou sub-chefe, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual, o que também encerra com data atual a atuação do(a) atual acadêmico(a) que ocupa o referido cargo (se algum).
--------------------------------	---

3.3 Modelos de Caso de Uso

O modelo de casos de uso corresponde a uma tentativa de descrever a relação das funcionalidades do sistema com cada um de seus atores. Os atores identificados no contexto deste projeto estão descritos na Tabela 2.

Tabela 2 – Descrição dos atores envolvidos nos casos de uso.

Ator	Descrição
Administrador do Sistema	Pessoa responsável pelo cadastro inicial dos usuários dentro do sistema.
Chefe de Departamento	Professor responsável por coordenar e supervisionar todas as atividades da competência do departamento. Deve conceder acesso aos outros usuários do departamento.
Subchefe de Departamento	Professor responsável por coordenar e supervisionar todas as atividades da competência do departamento, na ausência do Chefe de Departamento. Deve conceder acesso aos outros usuários do departamento.
Secretário(a) do Departamento	Encarregado da execução de todos os serviços administrativos do departamento, incluindo concessão de acesso aos usuários.

Ator	Descrição
Coordenador(a) de Curso de Graduação	Professor responsável por representar oficialmente o colegiado do curso de graduação. Deve poder conceder acesso aos usuários relacionados ao curso de graduação.
Subcoordenador(a) de Curso de Graduação	Professor responsável por representar oficialmente o colegiado do curso de graduação na ausência do coordenador. Deve poder conceder acesso aos usuários relacionados ao curso de graduação.
Secretario(a) do Colegiado da Graduação	Encarregado da execução de todos os serviços administrativos do curso de graduação, incluindo concessão de acesso aos usuários.
Coordenador(a) PPG	Professor responsável por representar oficialmente o colegiado de um programa de pós-graduação (PPG). Deve poder conceder acesso aos acadêmicos participantes de um PPG.
Subcoordenador(a) PPG	Professor responsável por representar oficialmente o colegiado PPG na ausência do coordenador. Deve poder conceder acesso aos acadêmicos participantes de um PPG.
Secretario(a) PPG	Encarregado da execução de todos os serviços administrativos do PPG, incluindo concessão de acesso aos acadêmicos.

Todos os atores descritos na Tabela 2 são acadêmicos que devem, de alguma forma, lidar com os cadastros mencionados na Seção 3.1. O Administrador do Sistema é um tipo de superusuário cujos privilégios são maiores do que os outros atores apresentados, podendo ser capaz de cadastrar todos os elementos mencionados na Seção 3.1, a saber: cadastro básico de acadêmicos, cadastro das unidades administrativas e cadastro dos papéis específicos de cada unidade administrativa. Os casos de uso referentes ao Administrador do Sistema são apresentados em seu diagrama de caso de uso na Figura 3.

Além do administrador, os próprios acadêmicos com papéis específicos podem também indicar outros acadêmicos para ocupar papéis dentro do contexto de um departamento, curso de graduação ou programa de pós-graduação. O módulo foi, então, dividido em três subsistemas, descritos a seguir.

3.3.1 Subsistema Departamento

Após o administrador fazer os cadastros iniciais, teremos um departamento cadastrado com seus elementos como descrito na Seção 3.1. Após esse cadastro inicial, os próximos cadastros relativos ao departamento ficam a cargos dos chefes do departamento e dos secretários(as) do departamento, como pode ser visto na Figura 4 e detalhados a seguir.

Nos casos de uso **Indicar chefe ou subchefe de departamento**, **Indicar**

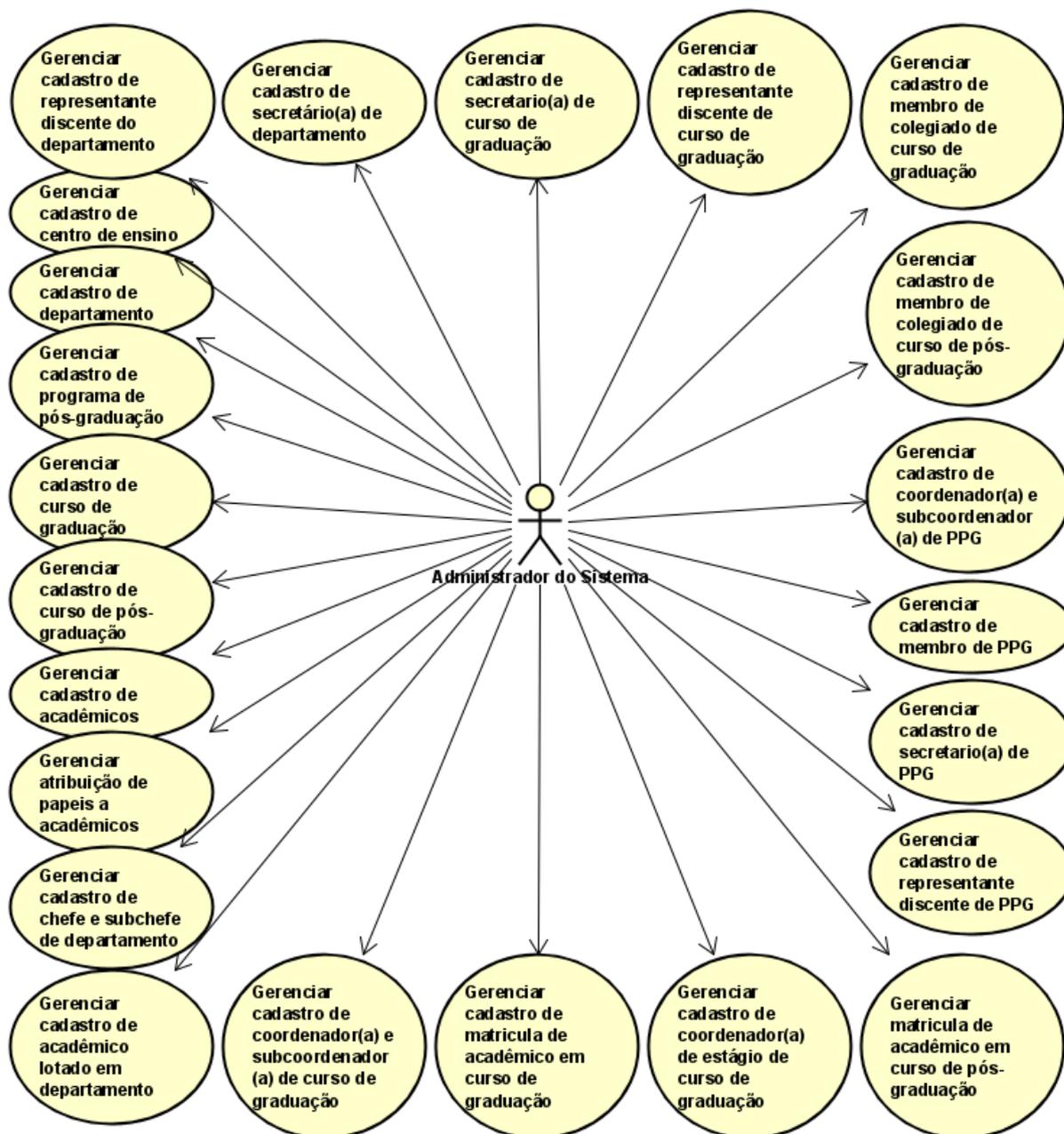


Figura 3 – Diagrama de Casos de Uso do administrador do sistema.

secretário(a) do departamento e **Indicar acadêmico lotado em departamento**, os atores (secretário(a), chefe e subchefe do departamento) poderão indicar uma nova relação entre um acadêmico e o departamento (ex.: novo chefe ou subchefe), ou uma conclusão desta relação quando o acadêmico finaliza seu período de atuação ou se aposenta. Para o caso dos chefes de departamento, é definido pelo regimento interno que o período de atuação é de apenas 2 anos, podendo ser renovado por igual período, como é citado no Documento de Requisitos deste módulo.

Para o caso de uso **Indicar representante discente em departamento** um estudante pode ser indicado em uma nova relação entre acadêmico e departamento pelos

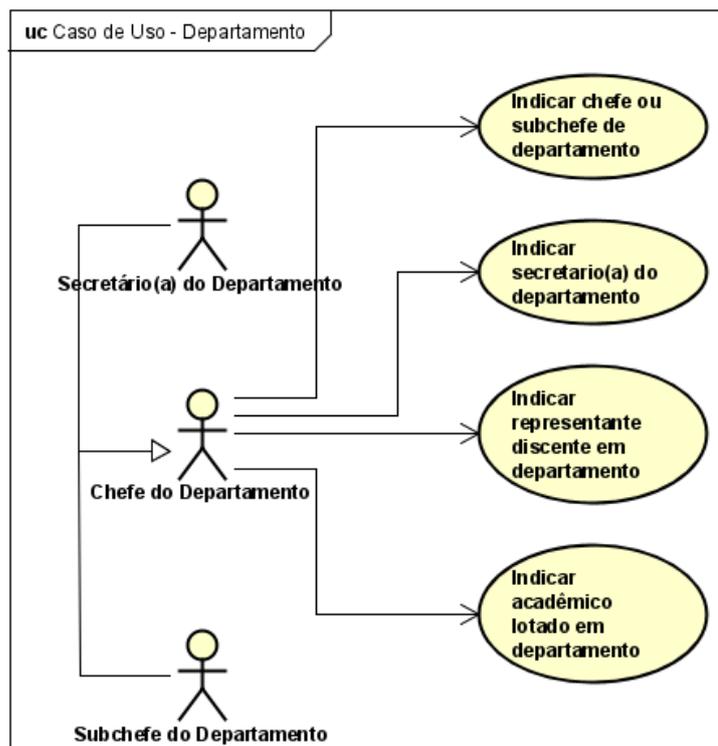


Figura 4 – Diagrama de Casos de Uso do subsistema Departamento.

atores, para exercer o cargo de representante discente, mantendo-se esta relação até a conclusão da mesma, seja por tempo ou em caso de desistência/conclusão do curso. Ao final, todas as datas de indicações e conclusões devem ser armazenadas no histórico.

3.3.2 Subsistema Graduação

Com o cadastro dos cursos de graduação e seus elementos vinculados pelo administrador do sistema, como descrito na Seção 3.1, os próximos cadastros relativos a um curso de graduação ficarão a cargo dos coordenadores(as) e dos secretários(as), como é mostrado na Figura 5 e detalhado a seguir.

Nos casos de uso **Indicar coordenador(a) ou subcoordenador(a) de curso de graduação**, **Indicar secretário(a) de curso de graduação**, **Indicar coordenador(a) de estágio de curso de graduação** e **Indicar membro(a) de colegiado de curso de graduação**, os atores (secretário(a), coordenadores(as) e subcoordenadores(as)) poderão indicar uma nova relação entre um acadêmico e o curso de graduação (ex.: novo secretário(a)), ou uma conclusão desta relação quando o acadêmico finaliza seu período de atuação ou se aposenta. Igualmente ao caso dos chefes de departamento (Subseção 3.3.1), é definido pelo regimento interno que o período de atuação dos coordenadores(as) do curso de graduação e de estágio é de apenas 2 anos, podendo ser renovado por igual período.

Para os casos de uso **Indicar representante discente de curso de graduação** e **Matricular estudante em curso de graduação** um estudante pode ser indicado em

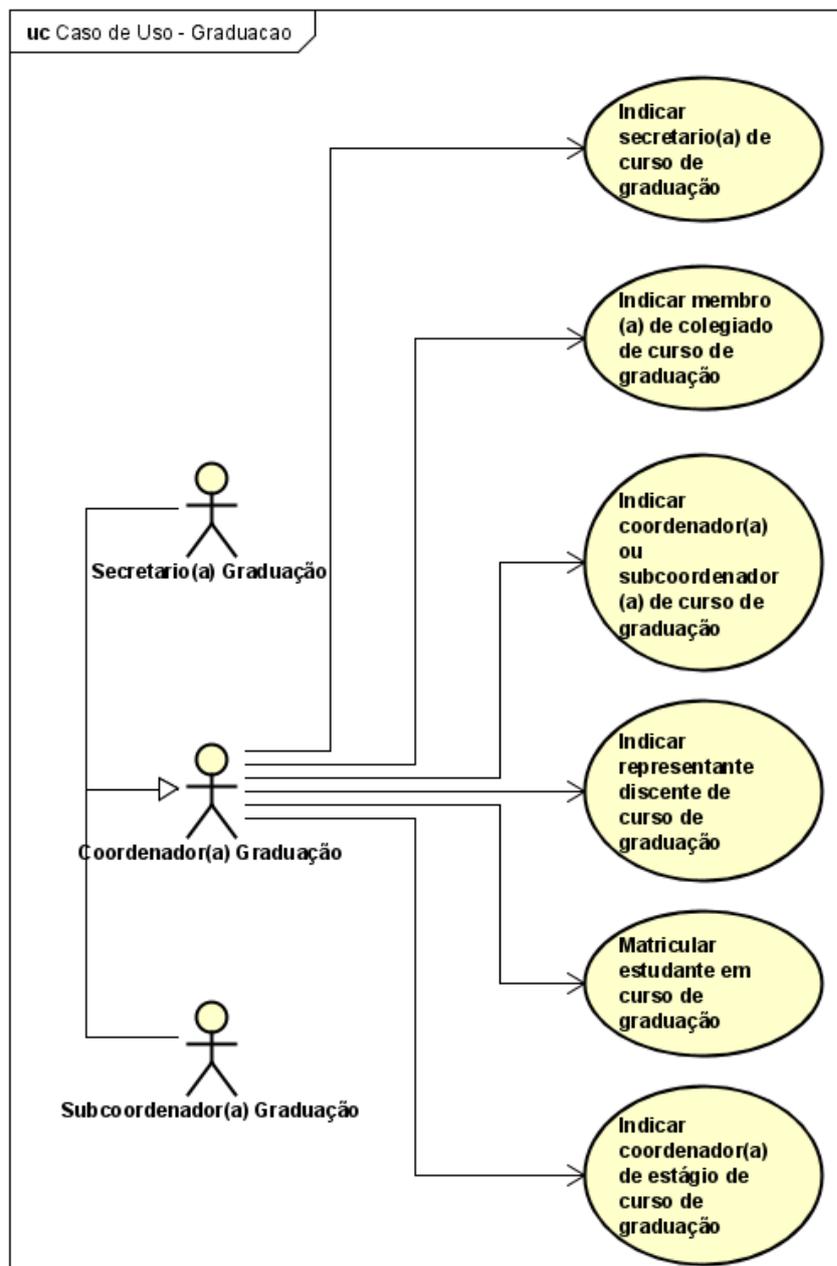


Figura 5 – Diagrama de Casos de Uso do subsistema Graduação.

uma nova relação entre acadêmico e um curso de graduação pelos atores, para exercer o cargo de representante discente ou para ser matriculado em um dos cursos de graduação disponíveis. Esta relação é mantida até a conclusão da mesma, seja por tempo ou em caso de desistência/conclusão do curso. Ao final, todas as datas de indicações e conclusões devem ser armazenadas no histórico.

3.3.3 Subsistema PPG

Por fim, analogamente como acontece na Subseção 3.3.2, os próximos cadastros relativos a um programa de pós-graduação e dos cursos de pós-graduação ficarão a cargo dos coordenadores(as) e dos secretários(as), como é mostrado na Figura 6 e detalhado a

seguir.

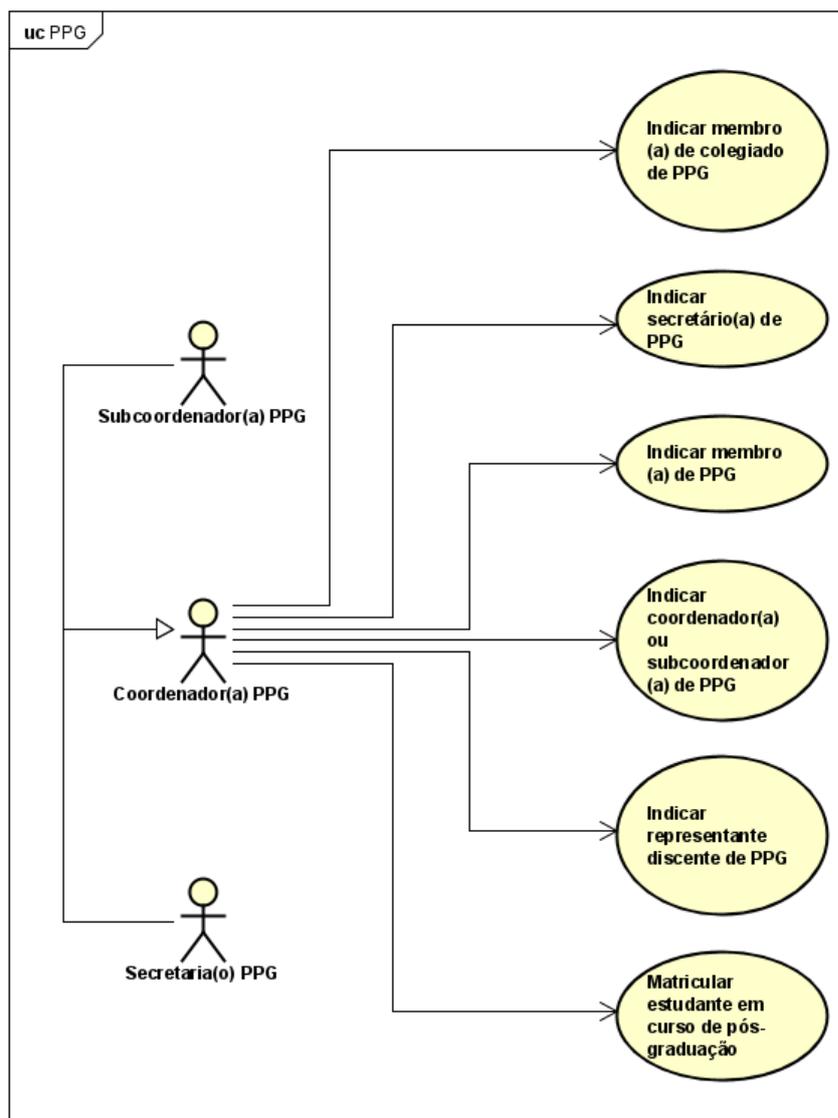


Figura 6 – Diagrama de Casos de Uso do subsistema PPG.

Nos casos de uso **Indicar coordenador(a) ou subcoordenador(a) de PPG**, **Indicar secretário(a) de PPG**, **Indicar membro(a) de PPG** e **Indicar membro(a) de colegiado de PPG**, os atores (secretário(a), coordenadores(as) e subcoordenadores(as)) poderão indicar uma nova relação ou uma conclusão de relação entre um acadêmico e um PPG, ou um acadêmico e um curso de pós-graduação (ex.: novo membro(a) de um PPG). Os coordenadores(as) de um PPG também seguem a regra de que seus cargos devem ter um período de duração de apenas 2 anos, podendo ser renovado por igual período.

Para os casos de uso **Indicar representante discente de PPG** e **Matricular estudante em curso de pós-graduação** um estudante pode ser indicado em uma nova relação entre acadêmico e um curso de pós-graduação, ou entre um acadêmico e um PPG. Os responsáveis por essa indicação serão os atores, que definirão se o cargo a ser exercido é o de representante discente ou se é para ser uma matrícula de um estudante em um dos

cursos de pós-graduação disponíveis. Esta relação é mantida até a conclusão da mesma, seja por tempo ou em caso de desistência/conclusão do curso. Ao final, todas as datas de indicações e conclusões devem ser armazenadas no histórico.

3.4 Diagrama de Classes

As figuras 7, 8 e 9 apresentam os diagramas de classes dos subsistemas Departamento, Graduação e PPG respectivamente.

Nota-se que as classes mais importantes são as classes das organizações administrativas (**CentroEnsino**, **Departamento**, **PPG** e **CursoGrad**) e a classe **Acadêmico**, pois são as classes que possuem a maior quantidade de relações com as outras classes do sistema, inclusive entre elas mesmas.

3.4.1 Subsistema Departamento

Dentro do diagrama de classe do subsistema Departamento, representado na Figura 7, temos a classe **Departamento** como classe central. A partir dela, temos a associação com a classe **CentroEnsino**, que acaba servindo como ligação do subsistema departamento com os demais subsistemas (Graduação e PPG), tendo em vista que **CentroEnsino** tem um papel em todos eles. As demais associações da classe **Departamento** são relações com os papéis específicos que um dado acadêmico pode adquirir, sendo esses papéis também relacionados com a classe **Academic** (ex.: um acadêmico pode participar de um cargo de chefia do departamento tendo associação com a classe **ChefiaDepartamento**). Podemos dizer então que as classes intermediárias (de papéis de um acadêmico específicos a uma unidade organizacional) são a ponte entre os acadêmicos e o departamento dentro do sistema. O nome de cada uma delas indica de maneira clara de qual papel específico se trata.

3.4.2 Subsistema Graduação

Analogamente ao diagrama de classe do subsistema Departamento (Figura 7), no diagrama de classes do subsistema Graduação, representado pela Figura 8, temos as classes **CursoGrad** e **Academic** sendo as classes centrais do subsistema. A classe **CentroEnsino** também está presente e serve como ponte entre os subsistemas. As classes intermediárias (entre **CursoGrad** e **Academic**) são relações entre o curso de graduação, o papel que um acadêmico terá neste curso, e o próprio acadêmico (ex.: um acadêmico pode participar de um cargo de coordenação sendo associado com a classe **CoordenacaoCursoGrad** que é associada a classe **CursoGrad**).

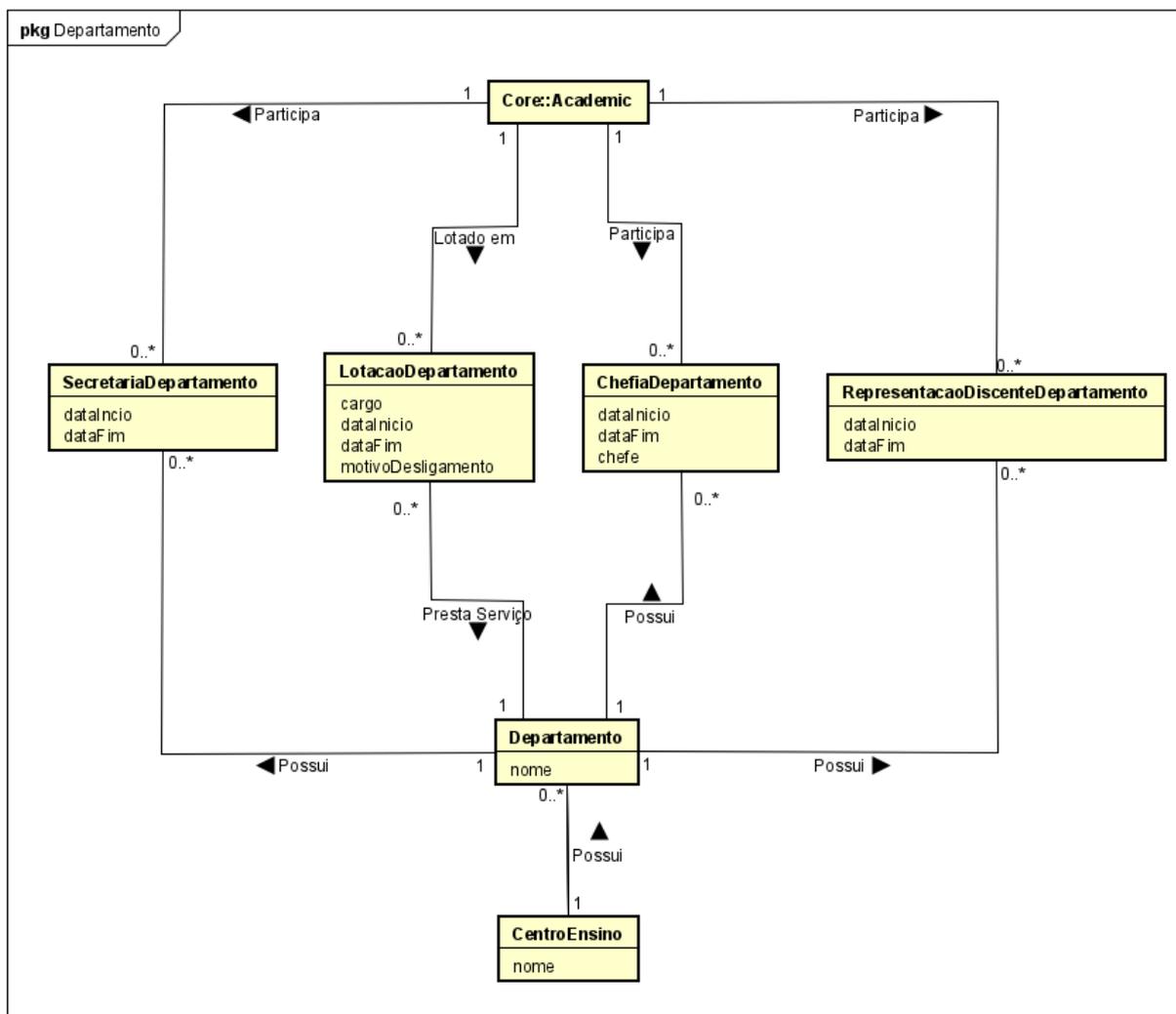


Figura 7 – Diagrama de classes do subsistema Departamento.

3.4.3 Subsistema PPG

Por fim, dentro do subsistema PPG temos a mesma ideia. As classes **PPG** e **Academic** sendo as classes centrais do subsistema, com a classe **CentroEnsino** presente e servindo como ponte entre os subsistemas. Classes intermediárias sendo relações entre o PPG, o acadêmico e o papel que um acadêmico terá no PPG. A exceção é a classe **CursoPosGrad** que representa cursos de pós-graduação que, diferentemente dos de graduação, não são ligados diretamente ao CentroEnsino, mas sim ao PPG. A relação de matrícula de estudantes em cursos, portanto, acontece entre Academic e CursoPosGrad. O diagrama de classes do subsistema PPG é apresentado na Figura 9.

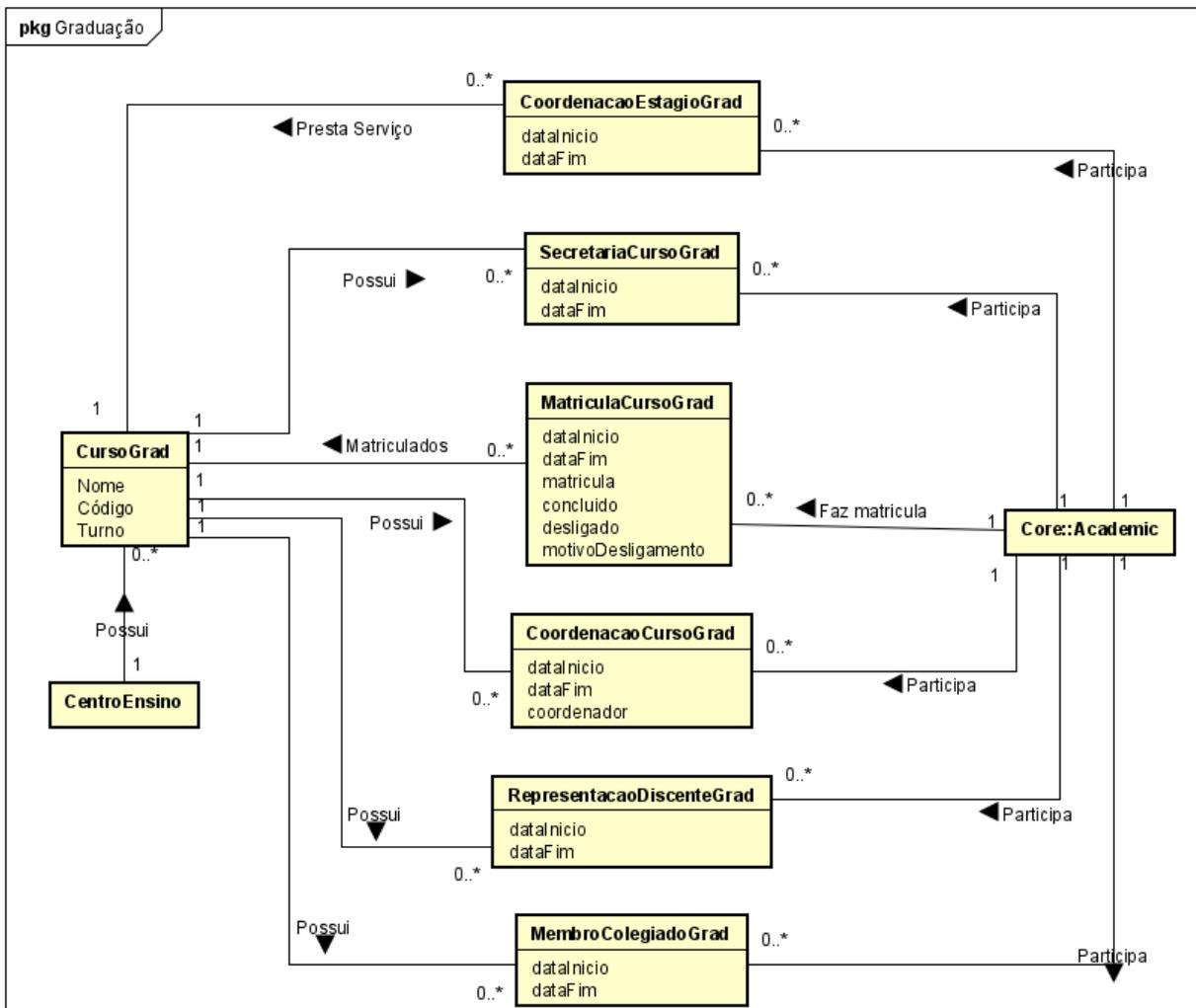


Figura 8 – Diagrama de classes do subsistema Graduação.

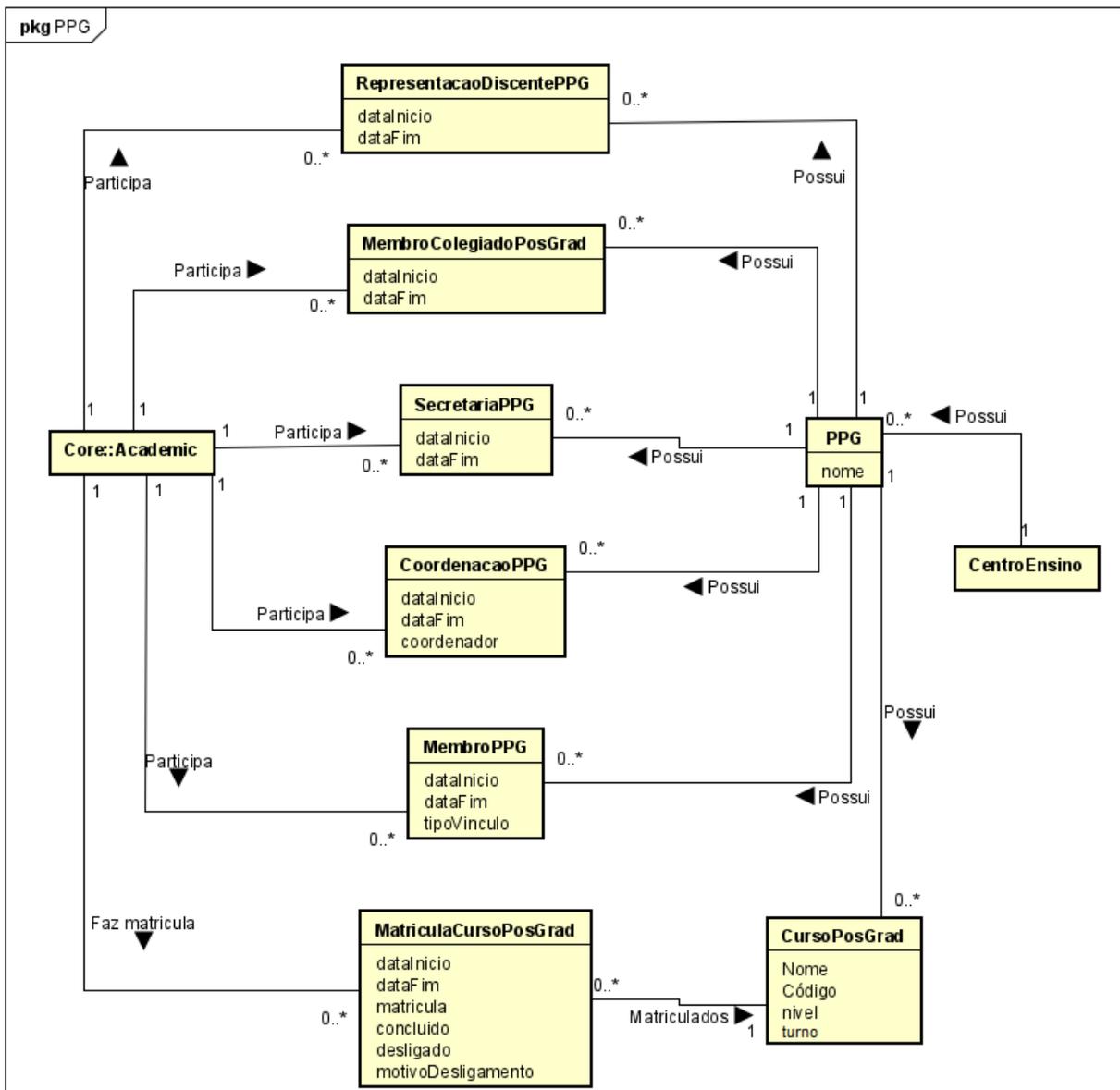


Figura 9 – Diagrama de classes do subsistema PPG.

4 Projeto do Sistema

Este capítulo tem como objetivo descrever o projeto arquitetural do *Macs - Sistema de controle de acessos do Marvin* e apresentar o que foi levantado na fase de Projeto. Na Seção 4.1 são apresentadas as tecnologias e ferramentas utilizadas no desenvolvimento do módulo; a Seção 4.2 descreve a arquitetura do módulo; e por fim, na Seção 4.3 temos a apresentação dos modelos FrameWeb.

4.1 Tecnologias e Ferramentas Utilizadas

Na Tabela 3 são listadas as tecnologias utilizadas no desenvolvimento da ferramenta, bem como o propósito de sua utilização.

Tabela 3 – Plataforma de Desenvolvimento e Tecnologias Utilizadas.

Tecnologia	Versão	Descrição	Propósito
Java EE	8.0	Conjunto de especificação de APIs e tecnologias, que são implementadas por programas servidores de aplicação.	Redução da complexidade do desenvolvimento, implantação e gerenciamento de aplicações Web a partir de seus componentes de infra-estrutura prontos para o uso.
Java	13	Linguagem de programação orientada a objetos e independente de plataforma.	Escrita do código-fonte das classes que compõem o sistema.
EJB	4.0	API para construção de componentes transacionais gerenciados por container.	Implementação das regras de negócio em componentes distribuídos, transacionais, seguros e portáteis.
JSF	2.3	API para a construção de interfaces de usuários baseada em componentes para aplicações Web	Criação das páginas Web e sua comunicação com as classes Java.
JPA	2.2	API para persistência de dados por meio de mapeamento objeto/-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
CDI	2.0	API para injeção de dependências.	Integração das diferentes camadas da arquitetura.
Facelets	2.0	API para definição de decoradores (<i>templates</i>) integrada ao JSF.	Reutilização da estrutura visual comum às páginas, facilitando a manutenção do padrão visual do sistema.
PrimeFaces	8.0	Conjunto de componentes visuais JSF <i>open source</i> .	Reutilização de componentes visuais Web de alto nível.
AdminFaces	1.3	<i>Template</i> visual completo integrado ao Facelets/JSF e ao PrimeFaces.	Fornecimento do padrão visual do sistema.

Tecnologia	Versão	Descrição	Propósito
JButler	2.0	<i>Framework</i> para facilitar o desenvolvimento de aplicações Java EE.	Fornecimento de superclasses prontas para entidades (classes de domínio) persistentes, DAOs e controladores/serviços básicos de cadastro (CRUD).
MySQL Server	8.0	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pelo sistema.
WildFly	19	Servidor de Aplicações para Java EE.	Fornecimento de implementação das APIs citadas acima e hospedagem da aplicação Web, dando acesso aos usuários via HTTP.

Na Tabela 4 vemos os softwares que apoiaram o desenvolvimento de documentos e também do código fonte.

Tabela 4 – Softwares de Apoio ao Desenvolvimento do Projeto

Tecnologia	Versão	Descrição	Propósito
Eclipse Java EE IDE for Web Developers	4.22.0	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento Java EE.	Implementação, implantação e testes da aplicação Web Java EE.
Apache Maven	3.8.0	Ferramenta de gerência/construção de projetos de software.	Obtenção e integração das dependências do projeto.
Git	2.35.1	Sistema de controle de versões distribuído	Registrar o histórico de edições de qualquer tipo de arquivo.
Bitbucket	2022	Plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git	Permitir que os membros do projeto contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.
Slack	4.23.0	Plataforma de colaboração entre membros de uma equipe	Colaboração entre os membros do projeto Marvin.
Trello	2022	Aplicativo web de gerenciamento de projeto que utiliza do paradigma kanban	Gerenciamento de fluxo de trabalho, maximizando a eficiência.
Astah UML	8.4.1	Ferramenta de modelagem UML	Criação de diagramas e modelos na fase de modelagem do sistema.
Overleaf	2022	Editor de LaTeX baseado em Cloud.	Escrita e edição da documentação do sistema.

4.2 Arquitetura de Software

No projeto arquitetural, o *Macs - Sistema de controle de acessos do Marvin* foi dividido em três módulos implementados como pacotes Java, seguindo a divisão de subsistemas feita na análise de requisitos e apresentada no Capítulo 3. A Figura 10 mostra os subsistemas que forma a arquitetura do Macs.

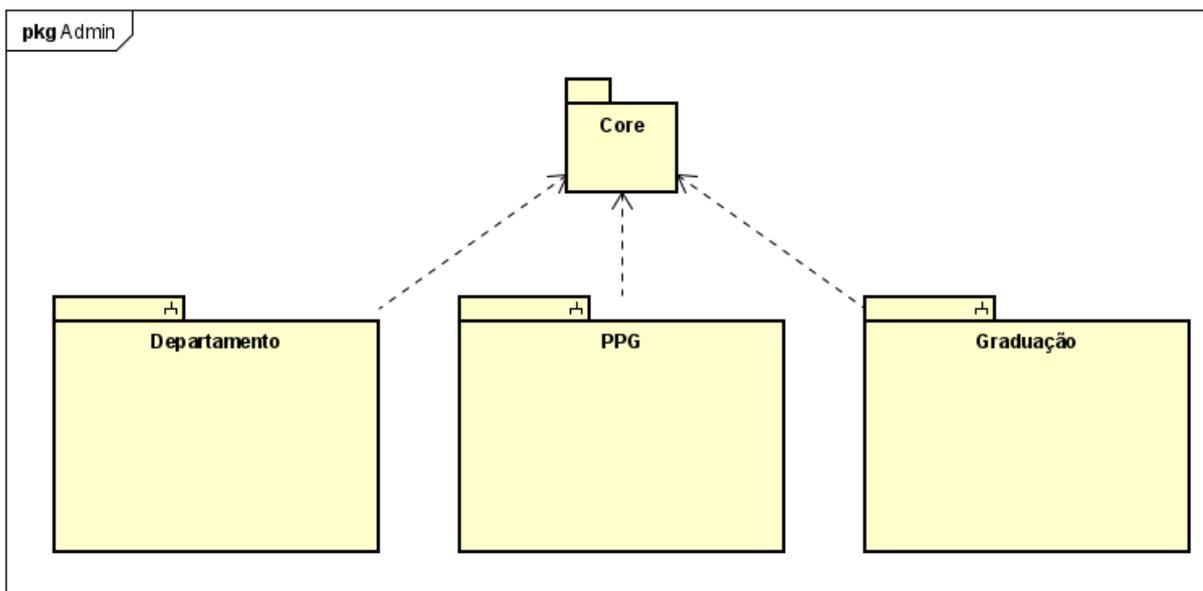


Figura 10 – Subsistemas do *Macs - Sistema de controle de acessos do Marvin*

Para o desenvolvimento do módulo *Macs - Sistema de controle de acessos do Marvin* foi utilizada uma arquitetura baseada no padrão arquitetônico Camada de Serviço (*Service layer*) (FOWLER, 2002) utilizando *frameworks*. A Figura 11 mostra a arquitetura do sistema e indica os componentes que estão presentes em cada pacote, bem como a atuação das tecnologias Java EE envolvididas.

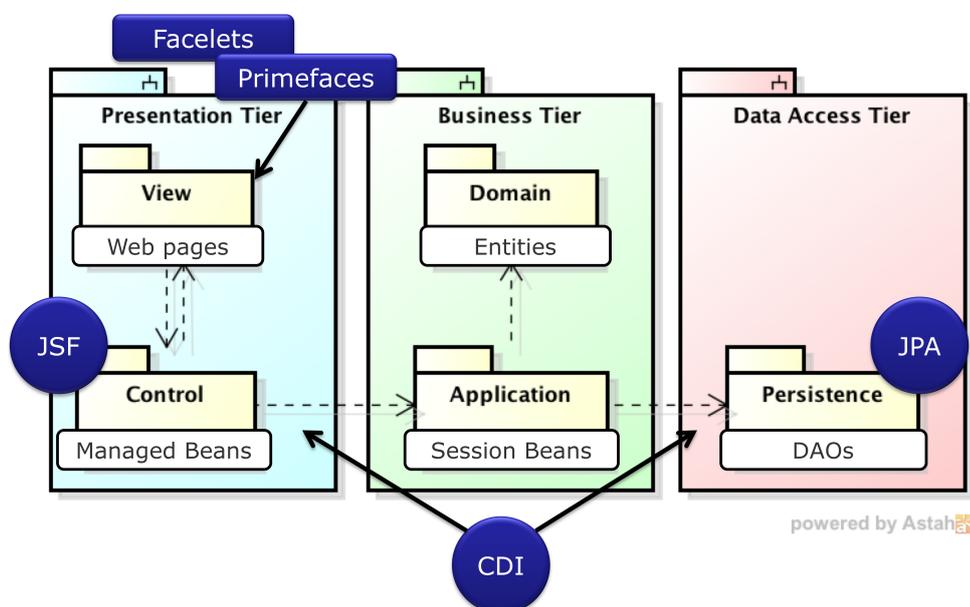


Figura 11 – Arquitetura do módulo *Macs - Sistema de controle de acessos do Marvin*.

O Marvin tem como base o pacote `br.ufes.informatica.marvin` sob o qual se encontram os outros subsistemas já implementados e integrados. O núcleo e os subsistemas são as partições da arquitetura, sendo que cada subsistema terá:

- **Camada de Apresentação** ou *Presentation Tier*: responsável pela interação com

o usuário, é dividida em dois pacotes: **Visão** (*View*), responsável por conter os artefatos de interface com o usuário, como páginas, scripts e leiautes; e **Controle** (*Control*), contendo as classes de controle que manipulam as solicitações feitas pelo pacote de Visão, e também responsável pela comunicação com o pacote de Aplicação (*Application*), pertencente à camada de negócio;

- **Camada de Negócio** ou *Business Tier*: responsável pela lógica de negócio, assim como a camada de apresentação, é dividida em dois pacotes: pacote de **Aplicação** (*Application*), que contém as classes responsáveis pelas funcionalidades do sistema; e o pacote de **Domínio** (*Domain*), que contém as classes que representam o domínio do problema, identificadas na fase de análise de requisitos;
- **Camada de Acesso de Dados** ou *Data Access Tier*: com apenas o pacote de **Persistência** (*Persistence*), é responsável pela persistência de dados, como por exemplo, compartilhamento de arquivos e acesso a bancos de dados.

Para ligar o pacote de **Aplicação** (*Application*) ao pacote de **Persistência** (*Persistence*) e o pacote **Controle** (*Control*) ao pacote de **Aplicação** é utilizado o CDI, que realiza a injeção de dependências automaticamente a partir de anotações inseridas sobre as declarações de variáveis.

Para o Macs, os pacotes foram agrupados a partir do pacote base do Marvin e pelas camadas da arquitetura, como é mostrado na Figura 12. A pasta visão (**view**) mostrada na camada de apresentação foi substituída por pastas que possuem o nome do subsistema, seguido da entidade relacionada, e em seu interior, páginas web definidas pelos arquivos XHTML, como pode ser visto na Figura 13.

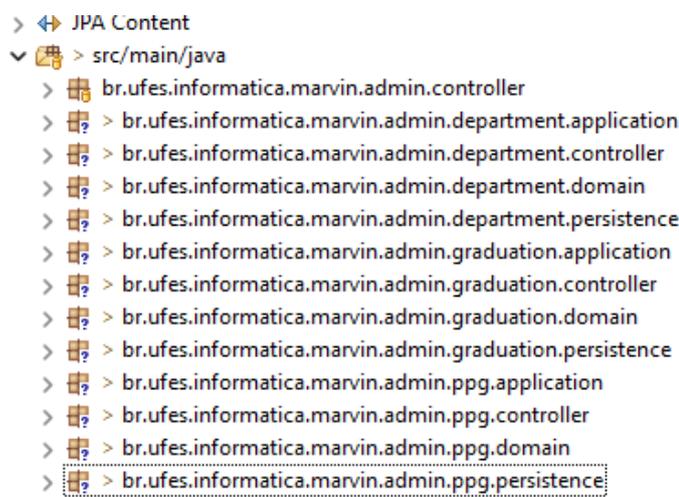


Figura 12 – Pacotes do *Macs - Sistema de controle de acessos do Marvin*

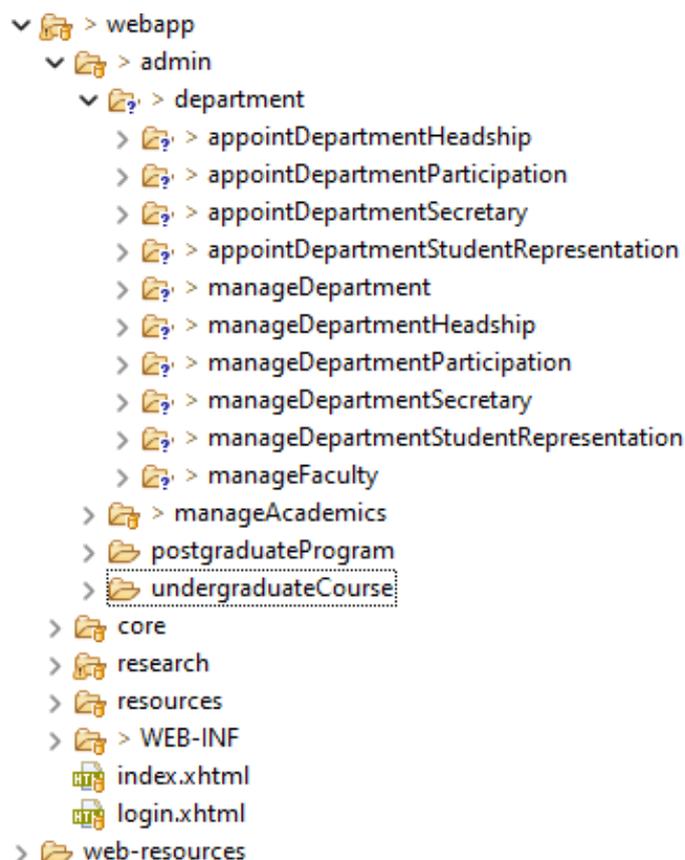


Figura 13 – Páginas web do *Macs - Sistema de controle de acessos do Marvin*

4.3 Modelagem FrameWeb

Após a definição da arquitetura (ilustrada na Figura 11), que inclui a utilização de *frameworks* (listados na Tabela 3), o projeto do Macs foi construído seguindo-se o método FrameWeb (descrito na Seção 2.4). Nesta seção, apresentamos os modelos FrameWeb do módulo Macs para cada camada da arquitetura, a saber: Negócio (modelos de Entidade e de Aplicação), Acesso a Dados (modelos de Persistência) e Apresentação (modelos de Navegação).

4.3.1 Camada de Negócio

A camada de negócio (ou *Business Tier*) é responsável pela lógica de negócio e é dividida nos pacotes de aplicação e domínio. Em aplicação, temos as classes que implementam as funcionalidades do sistema, e no pacote domínio, temos as classes de domínio. Esta seção será dividida em três sub-seções, indicando os modelos de entidade e de aplicação dos 3 subsistemas do módulo *Macs - Sistema de controle de acessos do Marvin*.

Para o pacote domínio, vale dizer que todas as classes de domínio estendem *PersistentObjectSupport* do JButler, mas essa herança não é mostrada nos diagramas pois

as várias associações iriam prejudicar a visualização do modelo. Os modelo de entidade dos subsistemas Departamento, Graduação e PPG são mostrados nas figuras 14, 15 e 16 respectivamente.

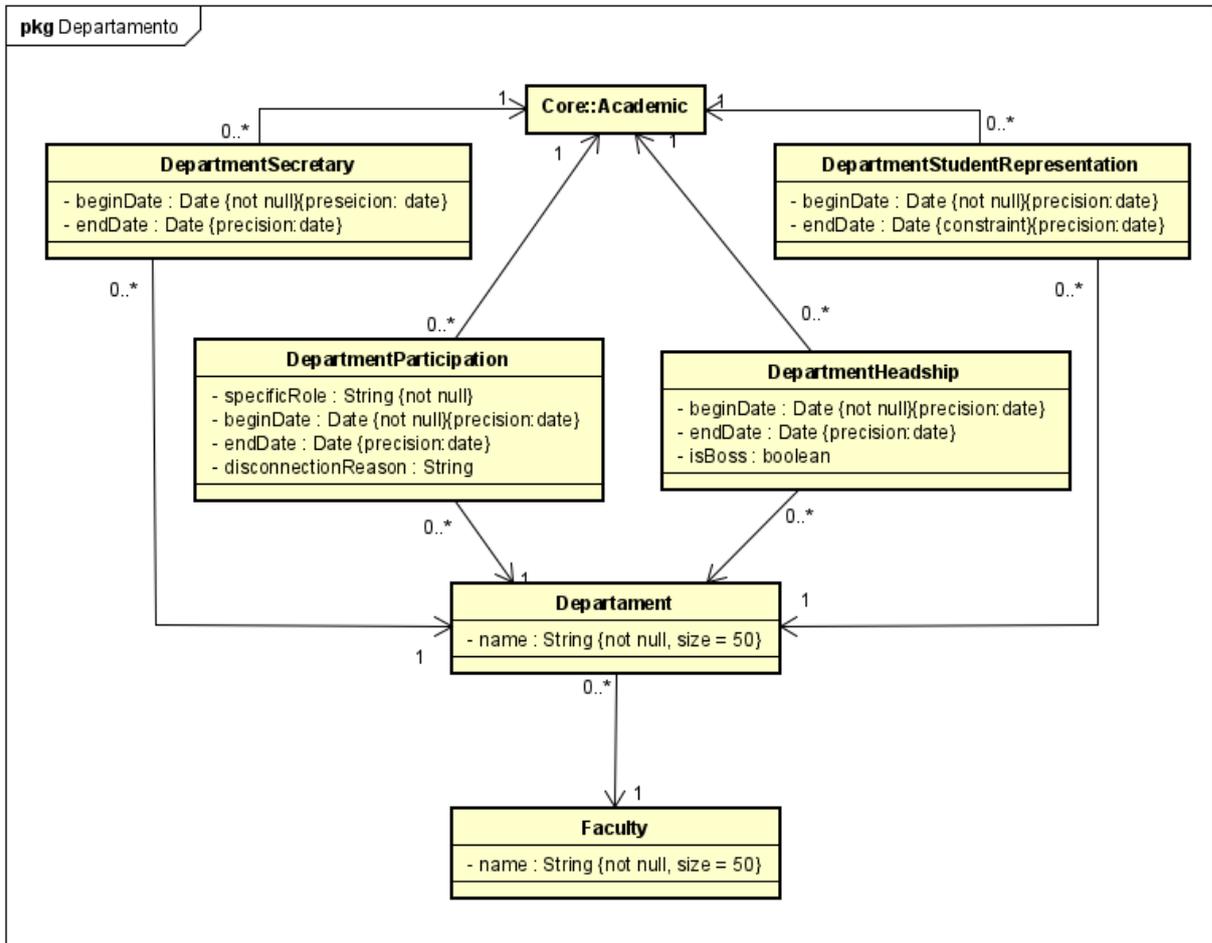


Figura 14 – Modelo de Entidades do subsistema Departamento

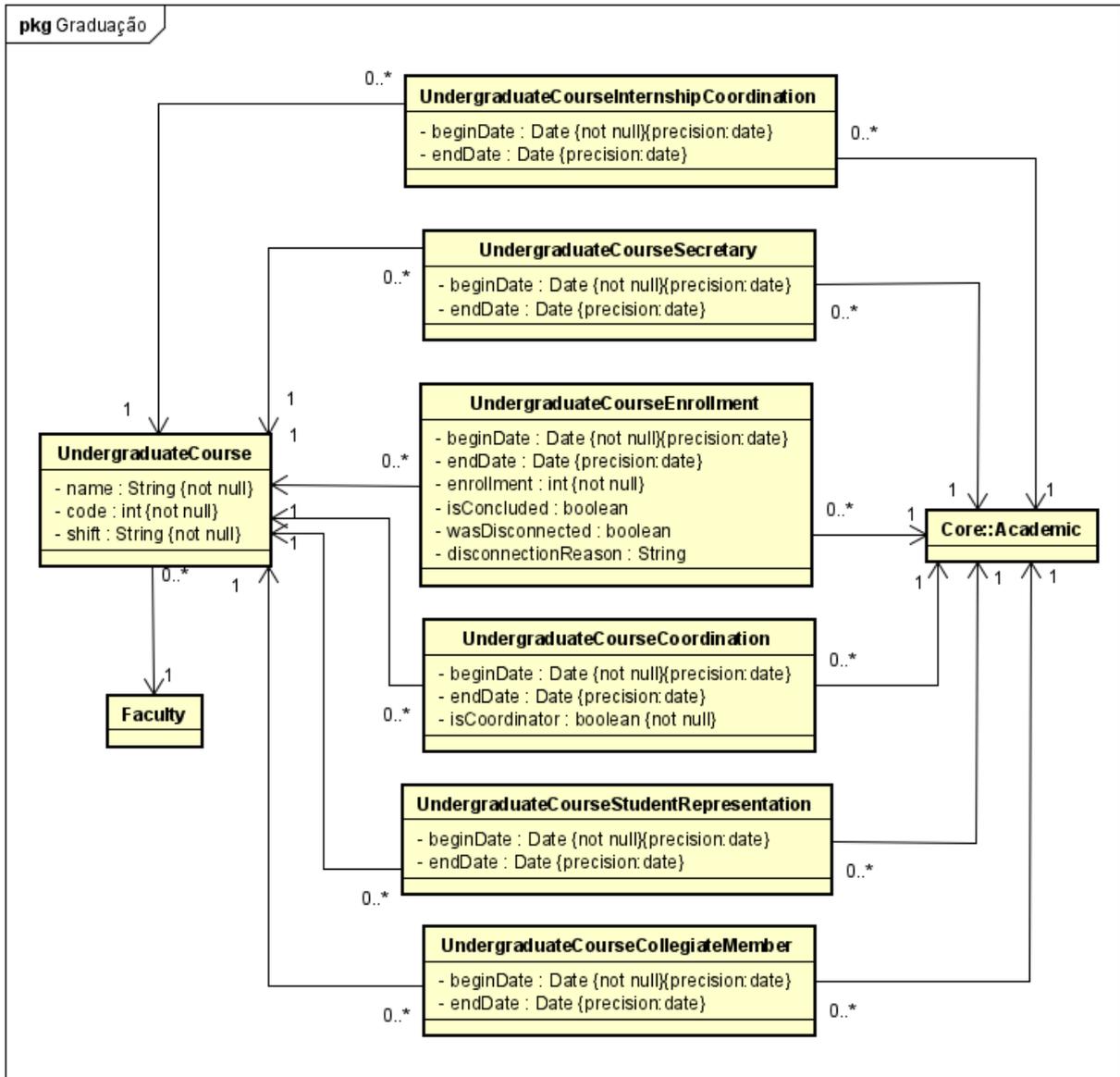


Figura 15 – Modelo de Entidades do subsistema Graduação

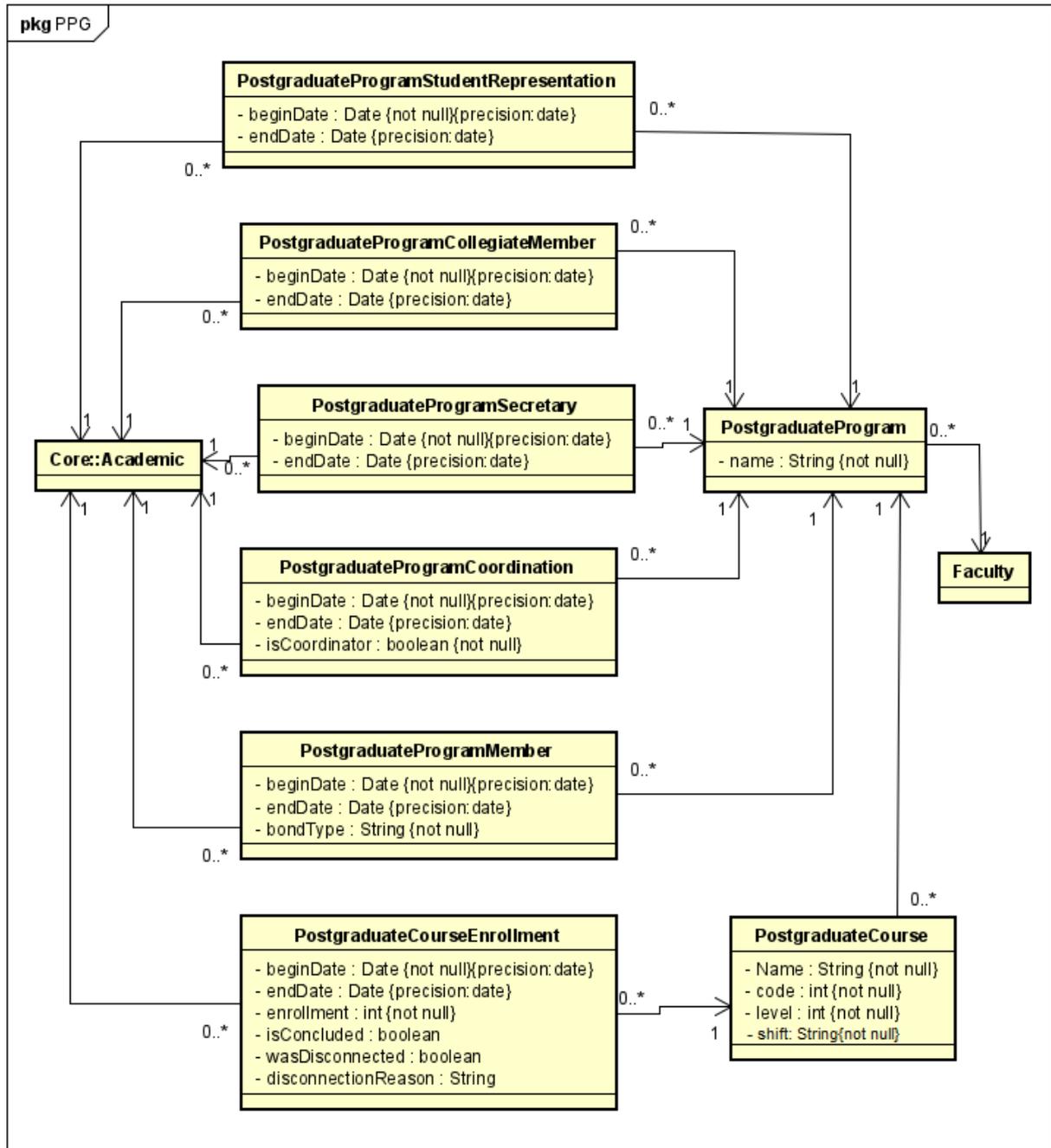


Figura 16 – Modelo de Entidades do subsistema PPG

Já os **Modelos de Aplicação**, que são diagramas de classes da UML que representam as classes de serviço, responsáveis pela codificação dos casos de uso e suas dependências, são utilizados para guiar a implementação das classes do pacote Aplicação e a configuração das dependências entre os pacotes Controle, Aplicação e Persistência (SALVATORRE, 2016).

No pacote aplicação, todas as classes relacionadas a cadastros estendem de *Crud-ServiceBean*, que é mostrado na Figura 17 de forma genérica. Assim como o pacote de domínio, a herança das outras classes de serviço foram ocultadas dos diagramas por conta da poluição visual que traria. Os modelo de aplicação dos subsistemas Departamento,

Graduação e PPG são mostrados nas figuras 18, 19 e 20 respectivamente.

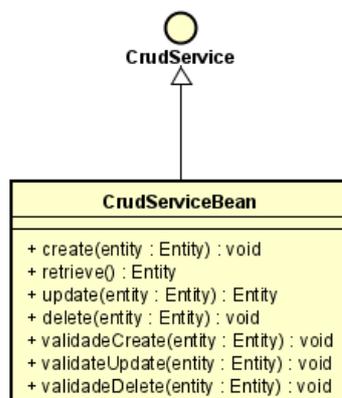


Figura 17 – Modelo de Aplicação genérico

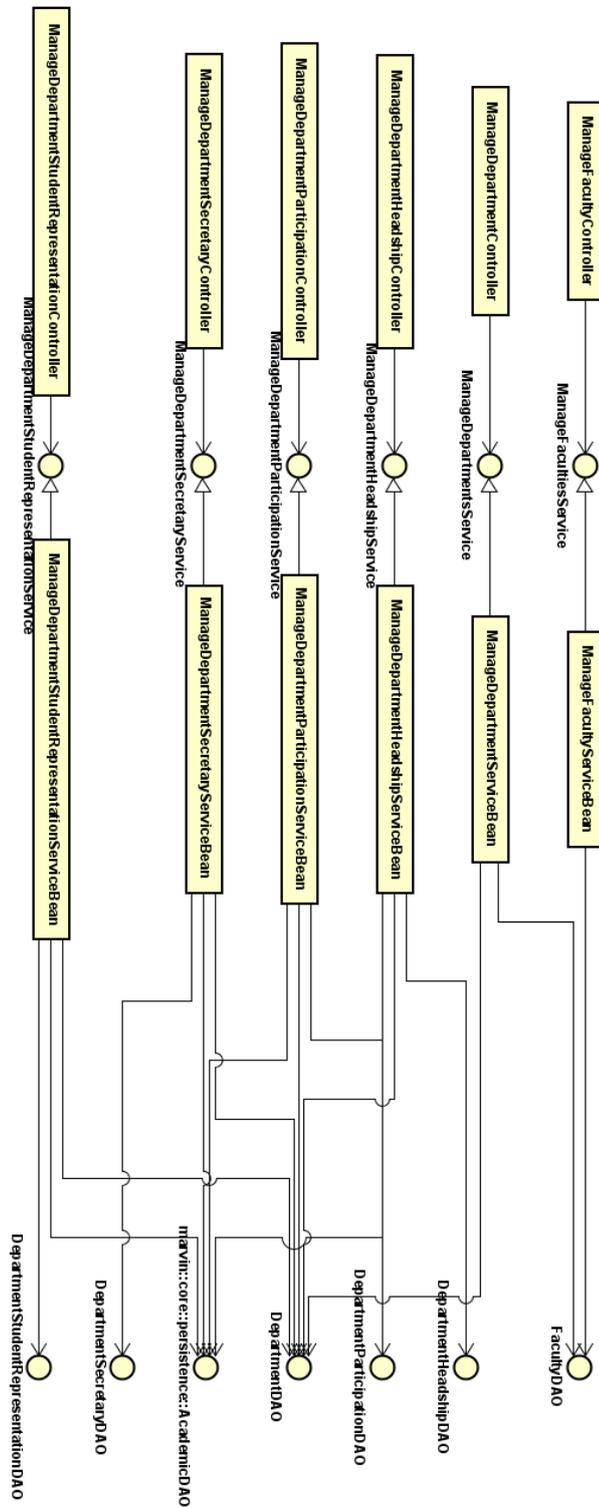


Figura 18 – Modelo de aplicação do subsistema Departamento

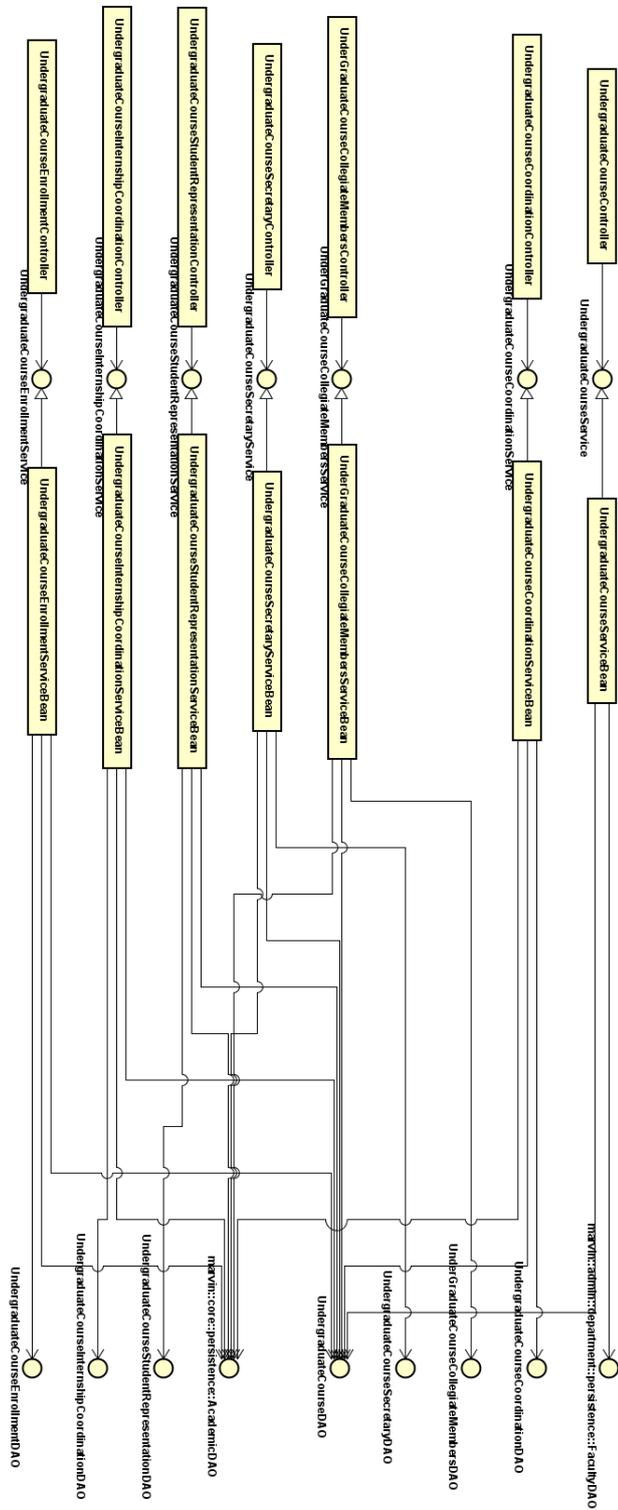


Figura 19 – Modelo de aplicação do subsistema Graduação

4.3.2 Camada de Acesso a Dados

Como mostrado na Seção 4.2, a camada de acesso a dados (ou *Data Access Tier*) do *Macs - Sistema de controle de acessos do Marvin* é baseada em uma arquitetura de camadas de serviços, possuindo um pacote de **Persistência** (*Persistence*) e é responsável pela persistência de dados.

O modelo de Persistência representa as interfaces DAO (*Data Access Object*) existentes. Uma interface DAO é responsável pela persistência das classes de domínio do sistema, sendo responsável por definir os métodos de persistência. Para cada classe de domínio que precisará acessar o banco de dados, deveremos ter uma interface <Nome da Entidade>DAO e uma classe <Nome da Entidade>JPADAO que implementará a interface DAO.

Todas as interfaces DAO existentes no módulo herdarão as propriedades da classe **BaseJPADAO**, do JButler, como é mostrado na Figura 21. Isto é feito com o propósito de se evitar a repetição de funcionalidades comuns a outros DAO's que existirão no sistema (ex.: criar, excluir, recuperar objetos, etc.).

Os modelo de persistência dos subsistemas Departamento, Graduação e PPG são mostrados nas figuras 22, 23 e 24 respectivamente. É importante notar que esses modelos possuem operações específicas (que foram implementadas neste trabalho) com o intuito de auxiliar no desenvolvimento das funcionalidades propostas nos modelos de caso de uso.

A título de exemplo, a operação *academicIsActiveDepartmentSecretary* de **DepartmentSecretaryJPADAO** retorna verdadeiro ou falso caso o acadêmico passado como argumento seja secretário(a) ativo(a) do departamento. Este método precisou ser implementado pois o JButler não possui esta operação.

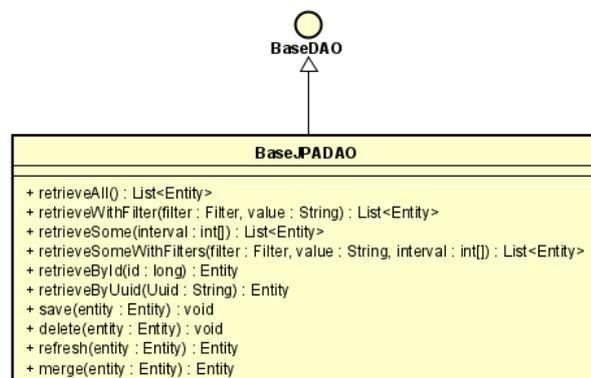


Figura 21 – Modelo de Persistência comum aos DAOs do sistema

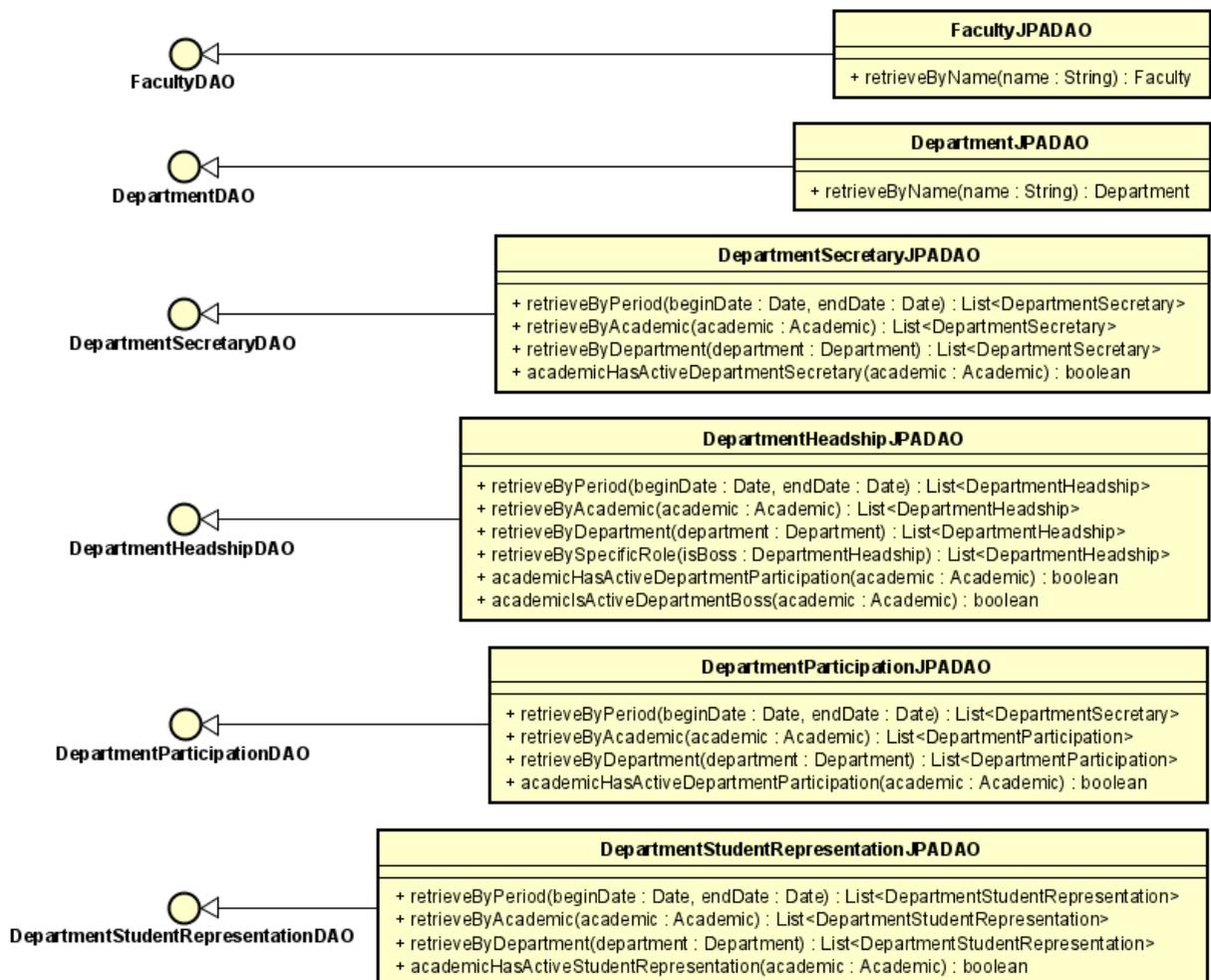


Figura 22 – Modelo de Persistência do subsistema Departamento

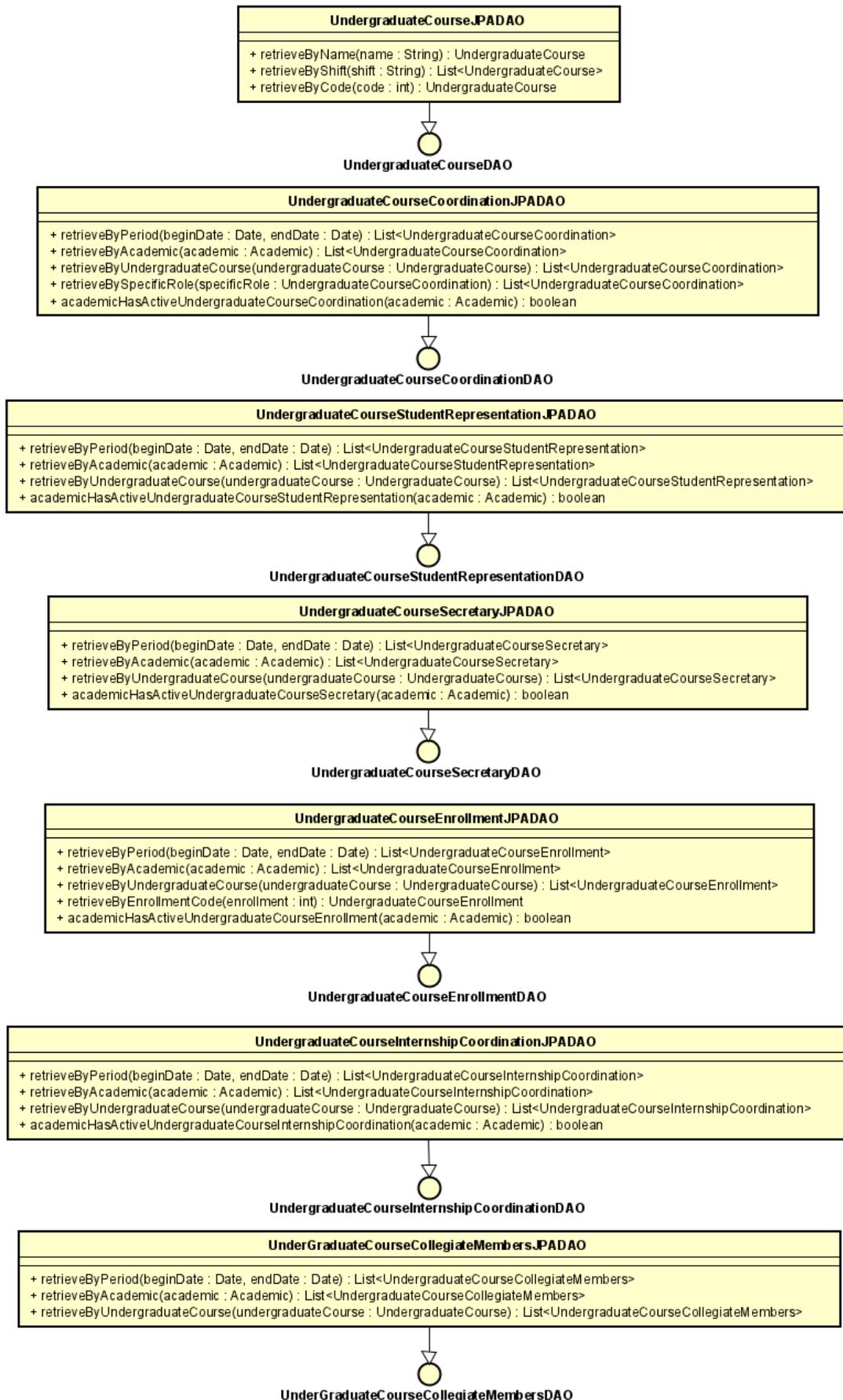


Figura 23 – Modelo de Persistência do subsistema Graduação

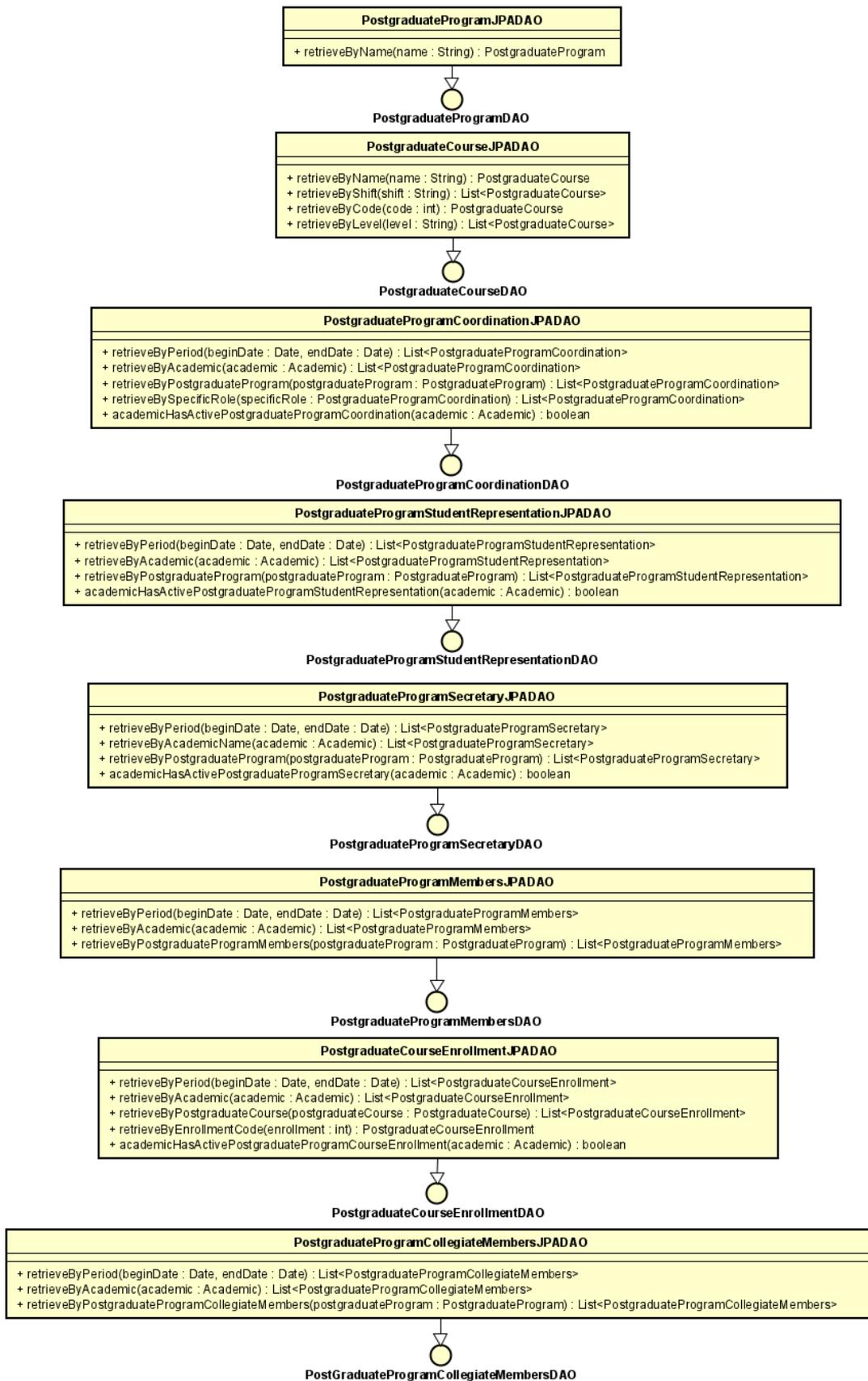


Figura 24 – Modelo de Persistência do subsistema PPG

4.3.3 Camada de Apresentação

De acordo com a arquitetura apresentada na Seção 4.2, a **Camada de Apresentação** é formada por diferentes componentes representados pelo **Modelo de Navegação** que tem um papel de guiar os desenvolvedores na hora de implementar as classes e funcionalidades dos pacotes **Visão** (*View*) e **Controle** (*Control*). Para cada entidade dentro do sistema, dentro da pasta Visão, teremos os formulários e as páginas, representadas por `form.xhtml` e `index.xhtml`. Já na pasta do controle, teremos uma classe `Manage<nome da entidade>Controller`, que consumirá os serviços da camada de negócio. Os campos do formulário `form.xhtml` não são exibidos na figura que representa este Modelo de Navegação pois dependem da entidade específica que está sendo cadastrada.

As funcionalidades de criação, recuperação, atualização e exclusão (abreviação de CRUD, do inglês *create, retrieve, update, delete*) são comuns de todas as classes do módulo *Macs - Sistema de controle de acessos do Marvin*. Essas funcionalidades são similares para todos os casos de uso cadastrais relativos ao administrador do sistema (como detalhado no Documento de Requisitos deste trabalho) pois utilizam a ferramenta JButler (listada na Seção 4.1). O modelo de navegação referente a esses CRUDs pode ser visto na Figura 25.

Como mostra a figura, todas as entidades deverão possuir a página `index.xhtml` que terá a listagem de todas os objetos recuperados do banco de dados, relativos à entidade em questão. Esta página possui também uma relação com um formulário `manageEntity` que contém as entidades mostradas em uma tabela.

Ao se selecionar um objeto dentro da tabela de de objetos renderizada, ele é atribuído ao atributo `entity`, existente em `ManageEntityController`, ficando disponíveis algumas funcionalidades para que seja executada ações como, por exemplo: atualizar o objeto selecionado (`update`), recuperar o objeto selecionado (`retrieve`), e excluir o objeto selecionado (`delete`). Para excluir o objeto selecionado após escolher a ação de exclusão (`delete`), é necessária uma confirmação, sendo também possível cancelar a exclusão. Além destas funcionalidades, o usuário pode escolher por criar um novo objeto escolhendo a ação `create`, que irá redirecionar para a página de formulário `form.xhtml`, na qual os atributos relativos à entidade em questão estarão disponíveis para preenchimento, e então salvar este novo objeto através da ação `save`, retornando após isso para a página de listagem dos objetos relativos à entidade em questão. Todas essas funcionalidades, ao serem executadas, irão chamar os respectivos métodos existentes em `ManageEntityController`.

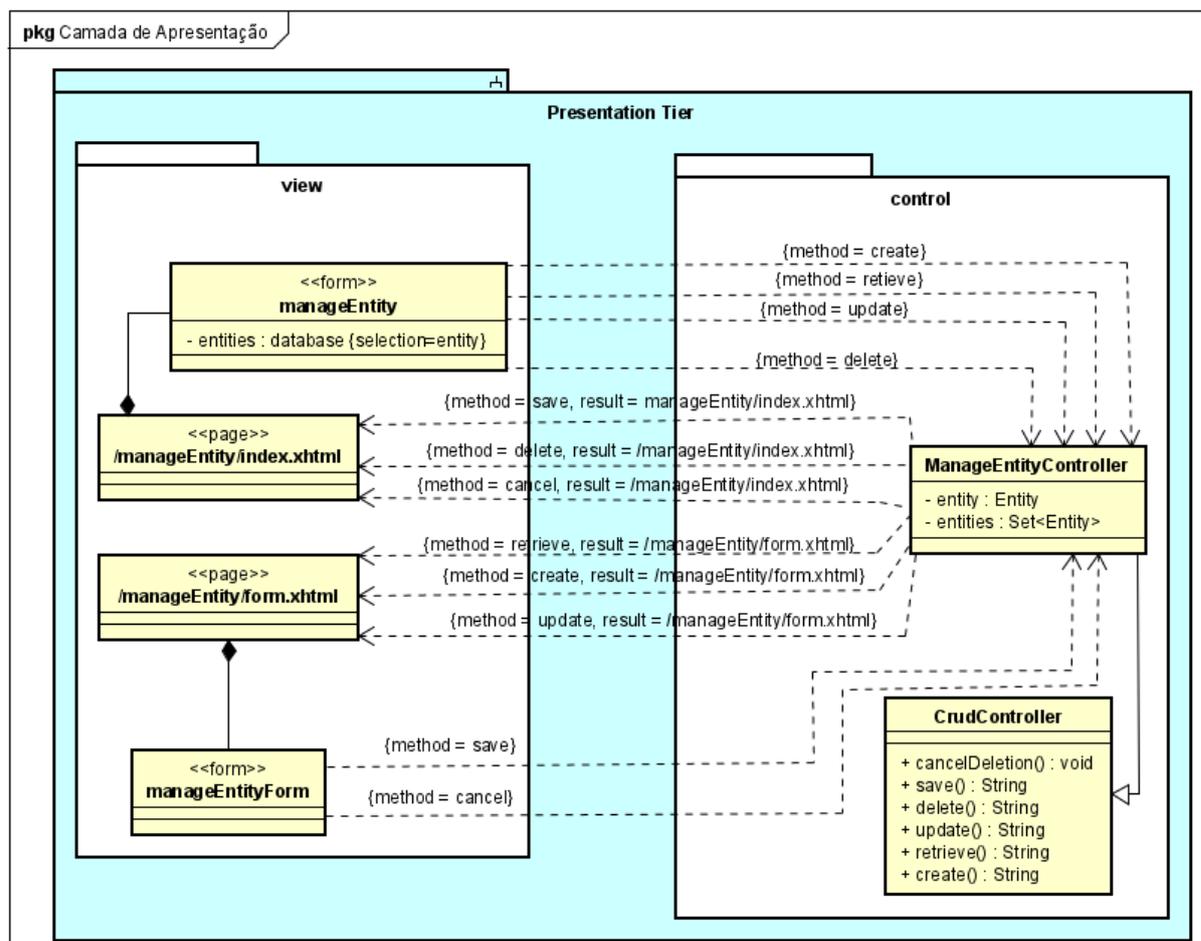


Figura 25 – Modelo de Navegação de um CRUD

Para os casos de uso onde as funções são diferentes das funções básicas oferecidas pelo JButler, temos o modelo de navegação mostrado na Figura 26. Apesar de possuir um fluxo parecido com o fluxo do modelo de navegação de um CRUD, várias condições de controle de permissão foram incluídas nas funcionalidades implementadas. Para um modelo de navegação de uma indicação, todas as entidades deverão possuir a página `index.xhtml` que terá a listagem de todos os objetos recuperados do banco de dados relativos a condições de acesso como, por exemplo: se o(a) usuário(a) logado(a) no momento tiver um papel de Chefe de Departamento, apenas objetos relacionados ao seu departamento serão mostrados para que ele(a) possa executar as ações. A página `index.xhtml` possui relação com o formulário `appointEntity` que armazena as entidades que serão mostradas.

Partindo do mesmo exemplo citado acima, um(a) Chefe de Departamento poderá selecionar um objeto da lista mostrada e executar a ação de finalizar participação (`finishParticipation`), fazendo com que o método de mesmo nome seja executado por `ManageEntityController`. Neste caso, para finalizar uma participação, será necessária uma confirmação. Caso o usuário opte por confirmar a finalização, o método `setEndDate` será executado, finalizando assim a participação e registrando a data atual como data final de participação, mantendo o registro do objeto na tabela. Caso o usuário opte por cancelar a

finalização da participação (`cancelFinishParticipation`), o método com o mesmo nome será executado e o objeto selecionado se manterá da mesma maneira que estava anteriormente de se escolher a ação de finalizar participação.

Também será possível indicar uma nova participação através da ação Apontar Novo (`Appoint New`) que irá executar o método de mesmo nome e fará com que ocorra um redirecionamento para o formulário (`form.xhtml`) onde os campos relativos aos atributos do novo objeto que será indicado estarão disponíveis para preenchimento. Em seguida, o usuário poderá salvar através do ação `save` que irá disparar o método `saveAppointment` ou cancelar através da ação `cancel` que irá disparar o método `cancelAppointment`. Ao final, independente da ação escolhida, será novamente redirecionado para a página `appointEntity (index.xhtml)`.

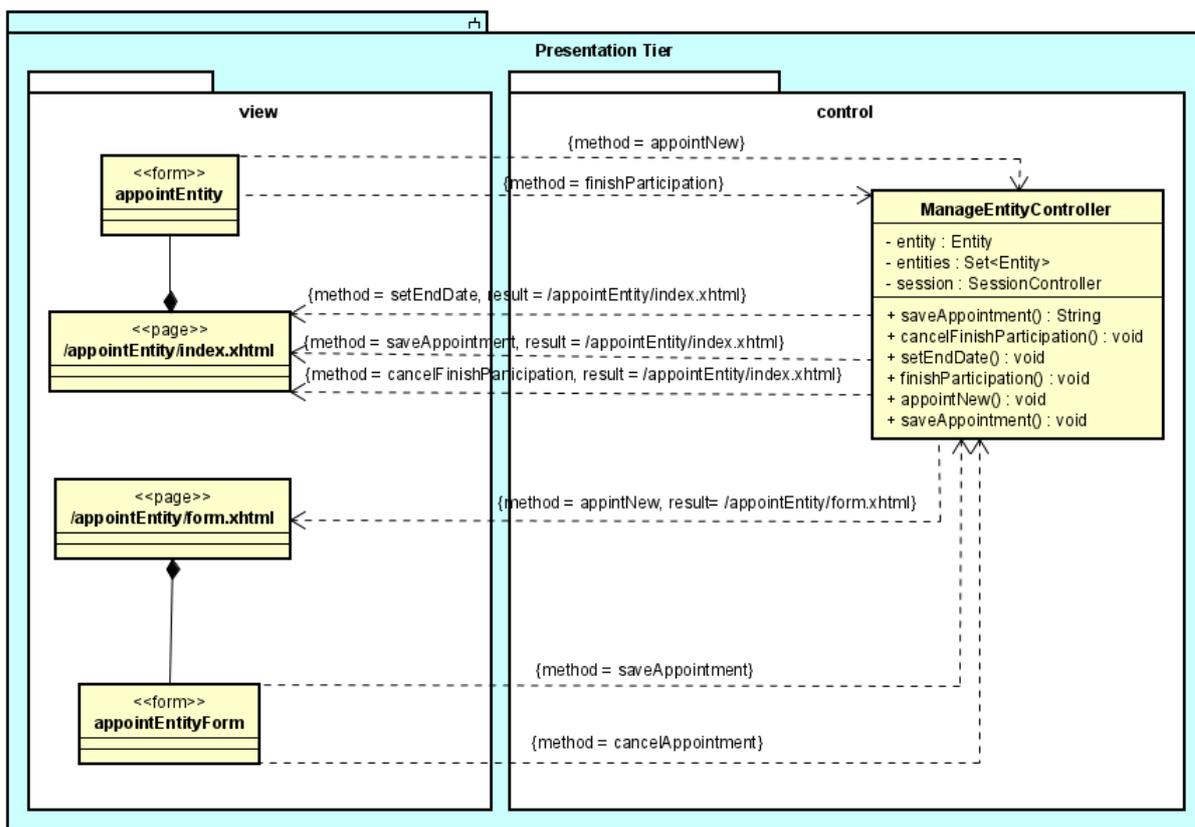


Figura 26 – Modelo de Navegação de uma indicação

5 Implementação do Sistema

Nesta seção o sistema desenvolvido é apresentado por meio de uma série de capturas de tela. A Figura 27 mostra a tela inicial de login no sistema. A tela de Login não foi implementada neste trabalho, sendo que já estava pronta quando foi iniciado a implementação do módulo *Mac's - Sistema de controle de acessos do Marvin*, mas será mostrada para que o fluxo de apresentação não fique incompleto.

O login utiliza e-mail e senha. O campo do e-mail possui validação para verificar se o mesmo é válido.

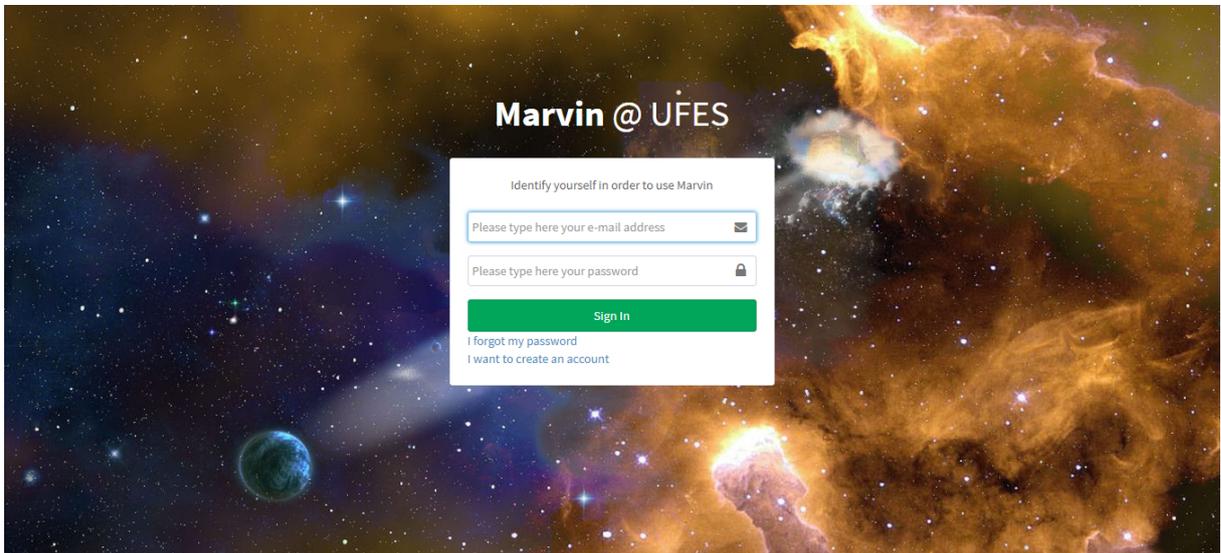


Figura 27 – Marvin - Tela de Login

Após o usuário efetuar o login, dependendo de sua hierarquia, ele poderá visualizar todas as funcionalidades implementadas e que já existem no sistema Marvin¹ referentes ao seu tipo de acesso. No caso da Figura 28, é exemplificado o tipo de acesso de Administrador do Sistema, podendo visualizar no menu esquerdo todas as funcionalidades que esse tipo de acesso possui. Importante lembrar também que o sistema é responsivo, ou seja, independente do tamanho da tela as funcionalidades estarão disponíveis para uso de uma forma que a interface continue amigável. Um exemplo é mostrado na Figura 29 onde temos a tela inicial se for visualizada pela versão *mobile*.

¹ <<https://github.com/dwss-ufes/marvin/>>

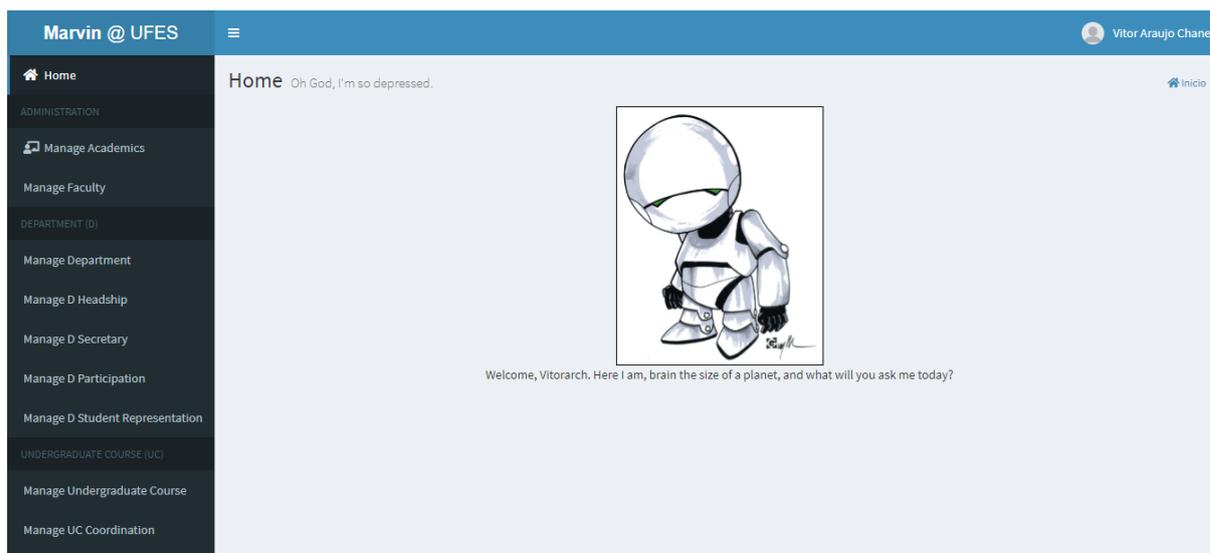


Figura 28 – Marvin - Página Inicial

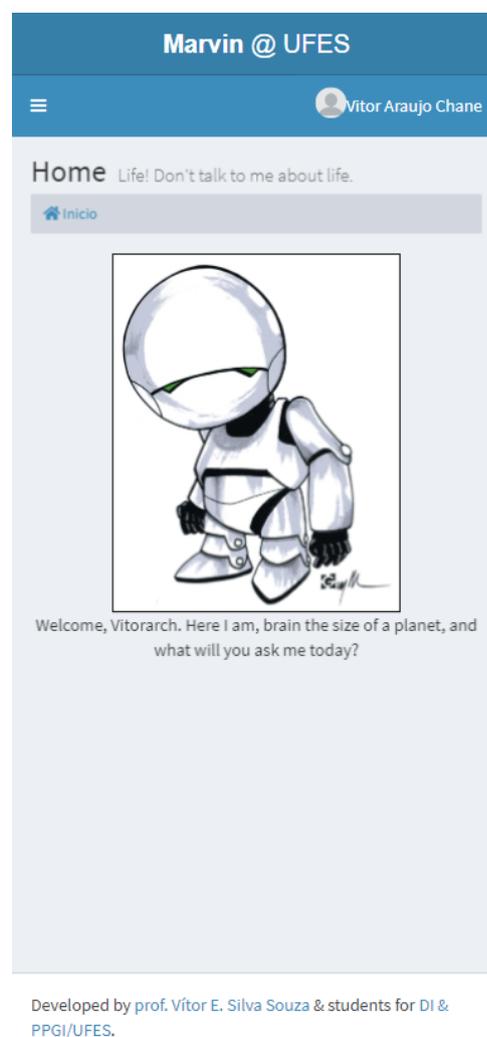


Figura 29 – Marvin - Página Inicial - Versão Mobile

Como foram implementadas várias funcionalidades de criação e concessão de acessos que possuem telas que seguem o mesmo padrão, se faz desnecessário mostrar todas elas

neste documento. Para evitar repetição, criaremos nesta demonstração uma criação de acesso para um chefe de um departamento, mostrando as funcionalidades de listagem, cadastro, recuperação, atualização, exclusão e por fim, funcionalidades de indicação por parte do próprio chefe do departamento.

Para inicializar, deveremos possuir acadêmicos cadastrados no sistema. A Figura 30 mostra a lista de acadêmicos cadastrados no sistema em um dado momento.

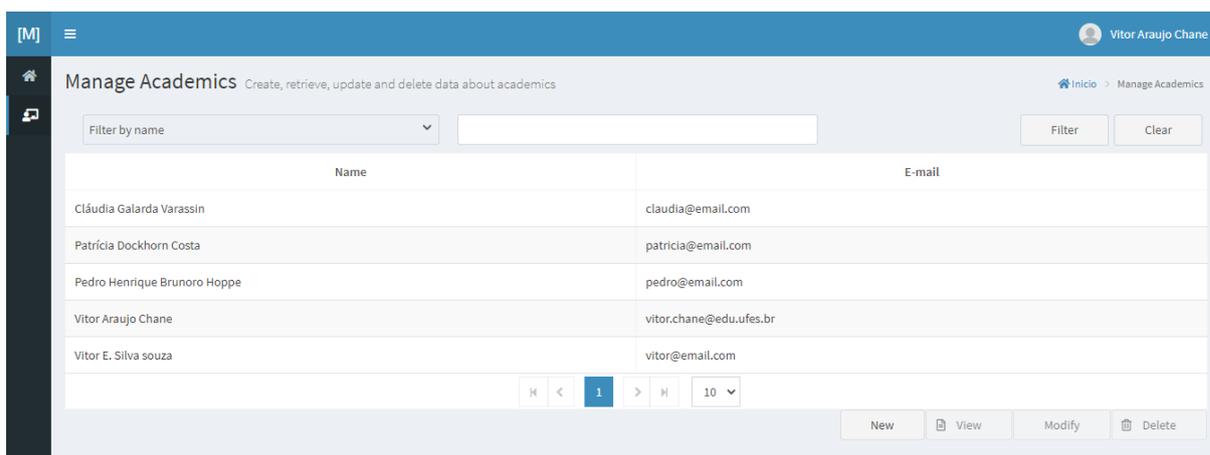


Figura 30 – Marvin - Lista de Acadêmicos

Caso se queira criar outro acadêmico dentro do sistema, o botão *New* (novo) ao ser clicado irá redirecionar o usuário para a página de criação de um novo acadêmico conforme mostra a Figura 31. Vale ressaltar que ao clicar em *New*, o formulário será exibido em branco, possibilitando a criação de um novo acadêmico ao inserir as informações obrigatórias como: nome do acadêmico, e-mail e CPF.

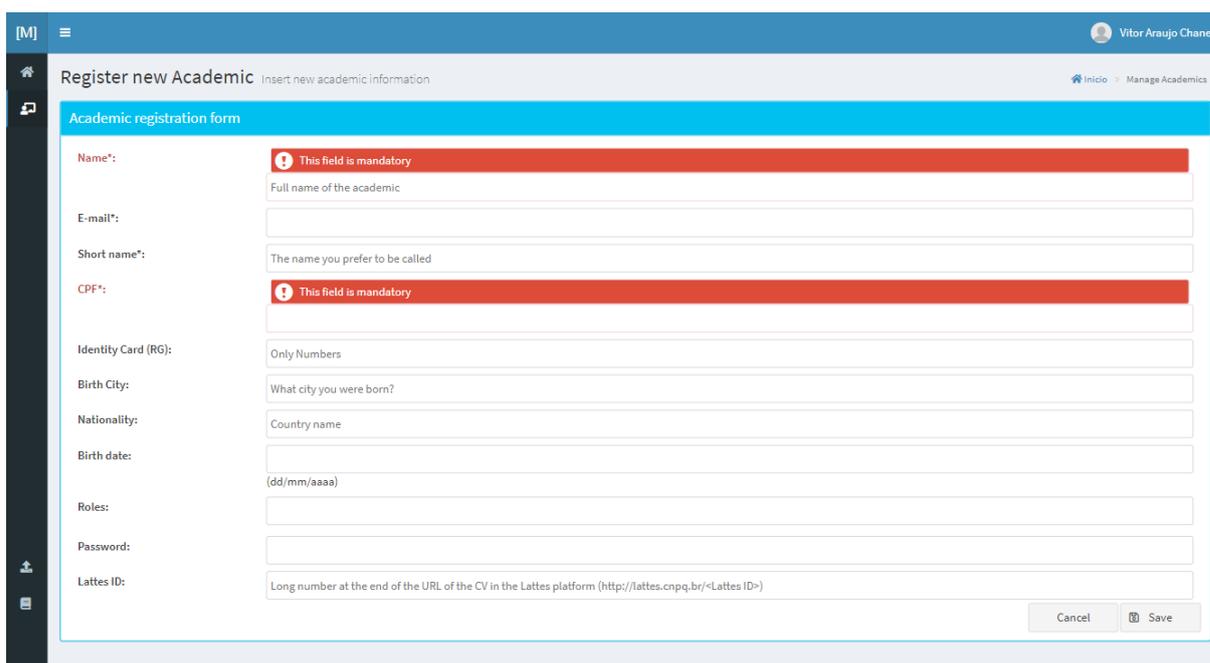
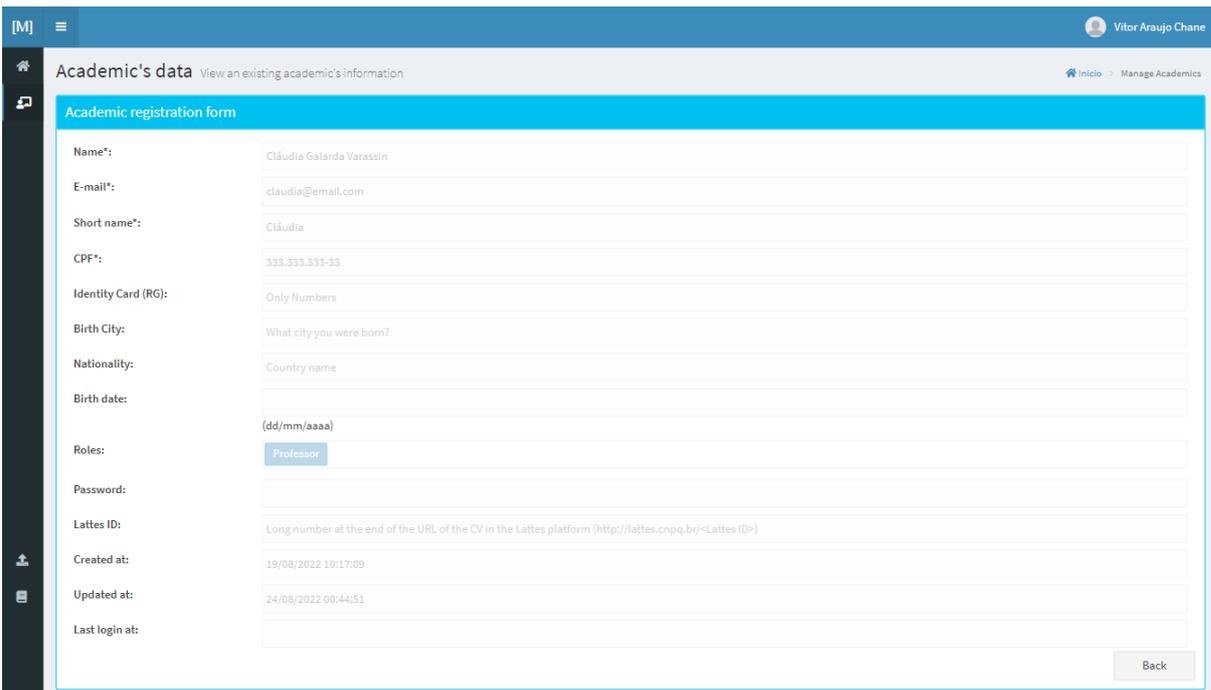


Figura 31 – Marvin - Página de Criação de Novo Acadêmico

Como o módulo *Macs - Sistema de controle de acessos do Marvin* se trata de um sistema de controle de acessos, é importante que neste sistema não possam ser criadas entidades duplicadas, evitando assim uma série de erros. Logo, é importante destacar que ao se tentar criar um acadêmico com e-mail ou CPF que já estejam sendo usados, o sistema irá impedi-los lançando uma mensagem de erro.

Ao selecionar um Acadêmico presente na listagem, os botões *View* (visualizar), *Modify* (alterar) e *Delete* (excluir) ficarão disponíveis. Ao clicar em *View* os campos do formulário serão somente leitura, como mostrado na Figura 32. Ao se clicar em *Modify* os campos do formulário já virão preenchidos com as informações do acadêmico selecionado, sendo possível editar o mesmo conforme mostrado na Figura 33. E por último, ao se clicar em *Delete*, um novo painel será exibido, para confirmação da exclusão como é mostrado na Figura 34. O usuário poderá, então, confirmar a exclusão, excluindo o acadêmico, ou cancelá-la. Ao final de qualquer uma das funcionalidades demonstradas, é necessário clicar no botão *Save* (salvar) para salvar as alterações.



The screenshot displays a web interface for viewing an academic's data. The page title is "Academic's data" with a subtitle "View an existing academic's information". The user is identified as "Vitor Araujo Chane". The form is titled "Academic registration form" and contains the following fields:

Name*	Cláudia Galarda Yarassin
E-mail*	claudia@email.com
Short name*	Cláudia
CPF*	333.333.333-33
Identity Card (RG):	Only Numbers
Birth City:	What city you were born?
Nationality:	Country name
Birth dates:	(dd/mm/aaaa)
Roles:	Professor
Password:	
Lattes ID:	Long number at the end of the URL of the CV in the Lattes platform (http://lattes.cnpq.br/<Lattes ID>)
Created at:	19/08/2022 10:17:09
Updated at:	24/08/2022 00:44:51
Last login at:	

A "Back" button is located at the bottom right of the form.

Figura 32 – Marvin - Página de Visualização de Acadêmico

[M] [Menu Icon] Vitor Araujo Chane

Update Academic Modify an existing academic's information Início > Manage Academics

Academic registration form

Name*: Cláudia Galarda Varassin

E-mail*: claudia@email.com

Short name*: Cláudia

CPF*: 333.333.333-33

Identity Card (RG): Only Numbers

Birth City: What city you were born?

Nationality: Country name

Birth date: (dd/mm/aaaa)

Roles: Professor

Password:

Lattes ID: Long number at the end of the URL of the CV in the Lattes platform (http://lattes.cnpq.br/<Lattes ID>)

Cancel Save

Figura 33 – Marvin - Página de Alteração de Acadêmico

[M] [Menu Icon] Vitor Araujo Chane

Manage Academics Create, retrieve, update and delete data about academics Início > Manage Academics

Filter by name [Dropdown] [Text Input] Filter Clear

Name	E-mail
Cláudia Galarda Varassin	claudia@email.com
Patrícia Dockhorn Costa	patricia@email.com
Pedro Henrique Brunoro Hoppe	pedro@email.com
Vitor Araujo Chane	vitor.chane@edu.ufes.br
Vitor E. Silva souza	vitor@email.com

New View Modify Delete

Items to delete:

1. Cláudia Galarda Varassin (claudia@email.com)

Cancel Confirm deletion

Figura 34 – Marvin - Página de Exclusão de Acadêmico

Por fim, na página de listagem de Acadêmicos também foi implementado um campo de consulta por filtro. Neste campo poderá ser selecionado o tipo de consulta dentre as existentes, informado a palavra chave a ser pesquisada e ao se clicar em *Filter* (filtrar) serão mostrados os resultados da busca. Um exemplo de consulta é mostrado na Figura 35

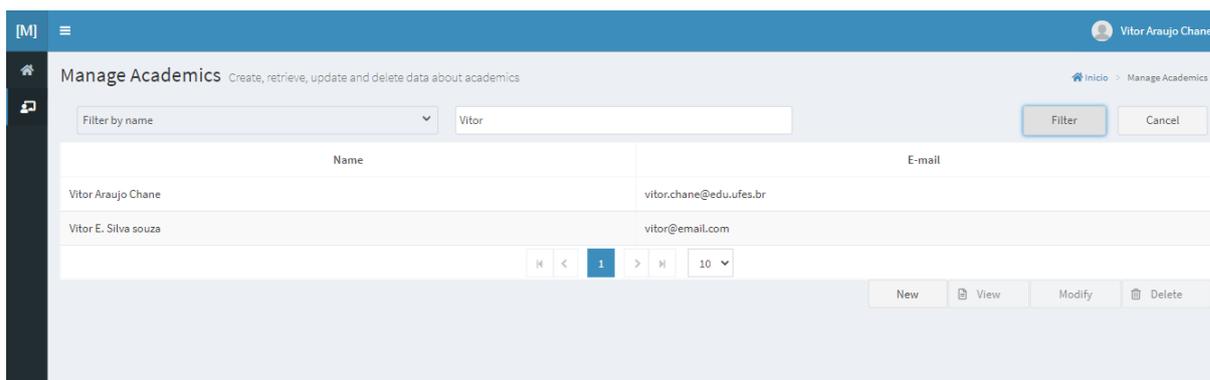


Figura 35 – Marvin - Consulta de Acadêmico

Para seguirmos, o próximo passo é criar um Centro de Ensino. Seguindo a mesma proposta da criação de um Acadêmico, temos a página de listagem de centros de ensino mostrada na Figura 36. Para a criação de um novo Centro de Ensino devemos clicar em *New* e o formulário de criação será renderizado em uma página como é mostrado na Figura 37. Centros de Ensino com nomes duplicados serão impedidos de serem criados.

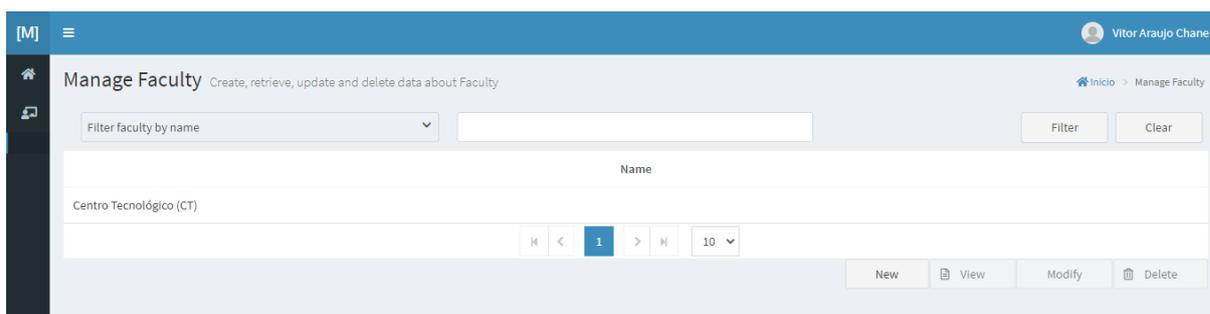


Figura 36 – Marvin - Listagem de Centros de Ensino

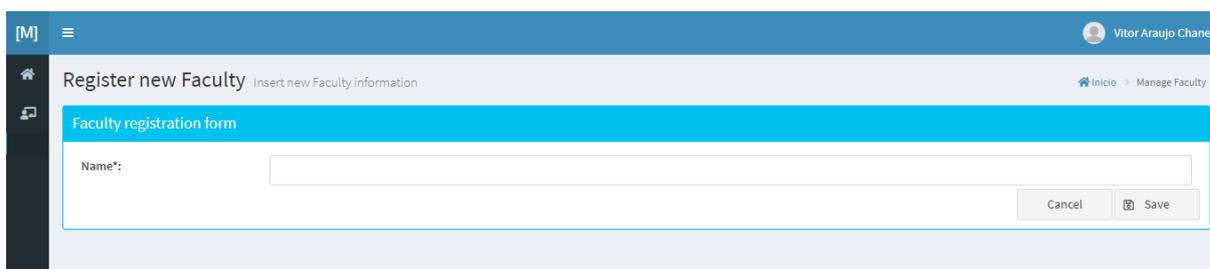


Figura 37 – Marvin - Criação de Novo Centro de Ensino

Após criarmos os Centros de Ensino, poderemos criar Departamentos que poderão se relacionar com estes Centros de Ensino criados. Seguindo o mesmo fluxo das demais criações já mostradas, teremos a página de listagem dos Departamentos como mostra a Figura 38. Para a criação de um novo Departamento basta clicar em *New* e o formulário de criação será mostrado. Diferente das outras criações mostradas até aqui, a criação de um novo Departamento requer a seleção de um Centro de Ensino previamente criado. Após inserirmos o início do nome do Centro de Ensino desejado, uma lista contendo os

Centros de Ensino disponíveis será mostrada possibilitando a escolha como é mostrada na Figura 39. Departamentos com nomes duplicados serão impedidos de serem criados.

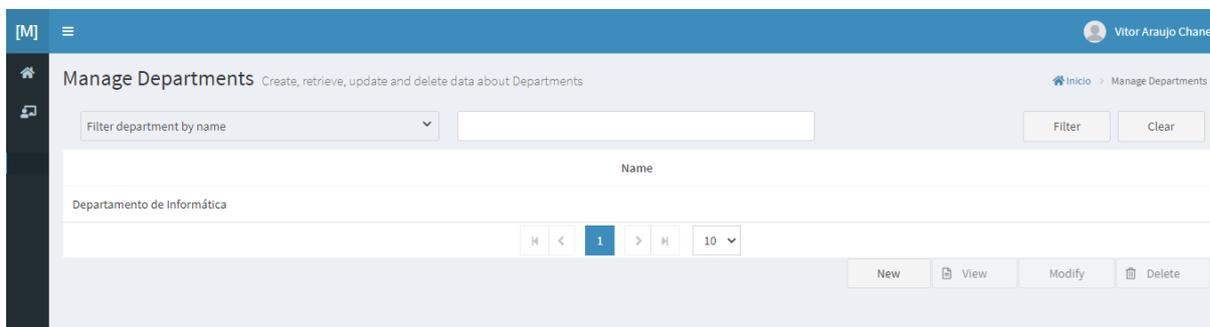


Figura 38 – Marvin - Listagem de Departamentos

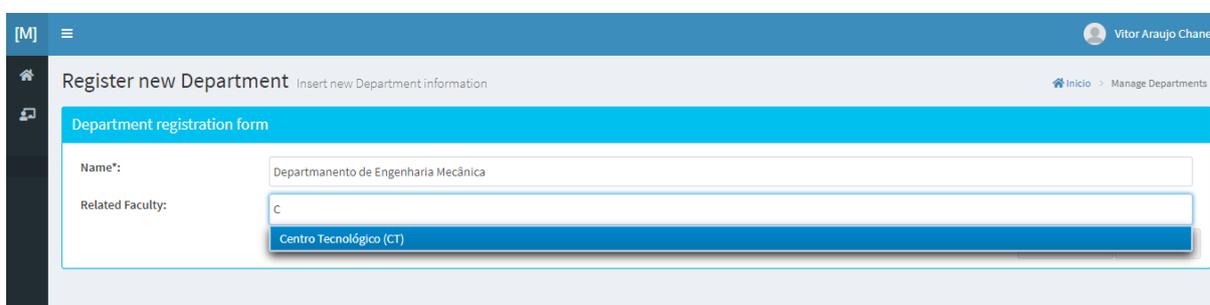


Figura 39 – Marvin - Criação de Novo Departamento

Por fim, após termos criado Acadêmicos, Centros de Ensino e Departamentos relacionados a estes Centros de Ensino, podemos finalmente criar algo que relacione tudo isso. Como foi definido no início deste capítulo, foi escolhida a criação de uma Chefia de um Departamento. Seguindo o mesmo fluxo das demais criações já mostradas, teremos a página de listagem das Chefias de Departamentos ativas e inativas como mostra a Figura 40 e sua versão *mobile* na Figura 41.

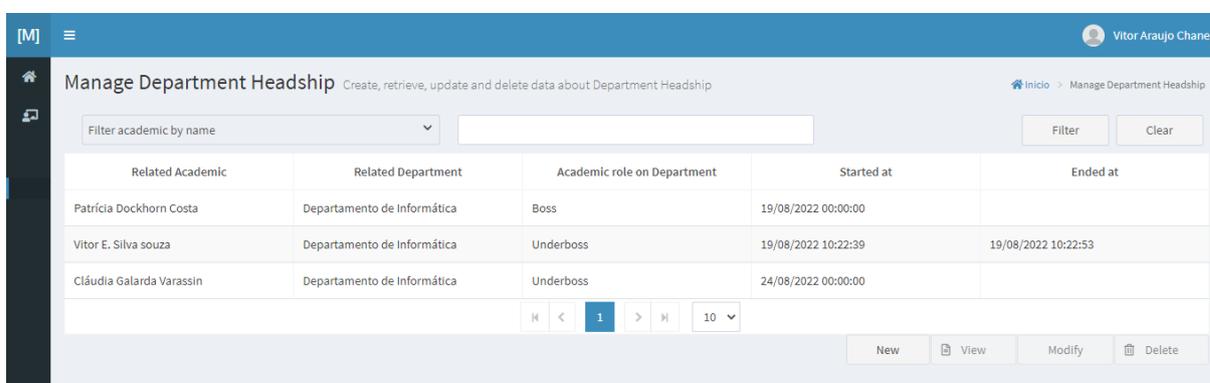


Figura 40 – Marvin - Listagem de Chefias de Departamento

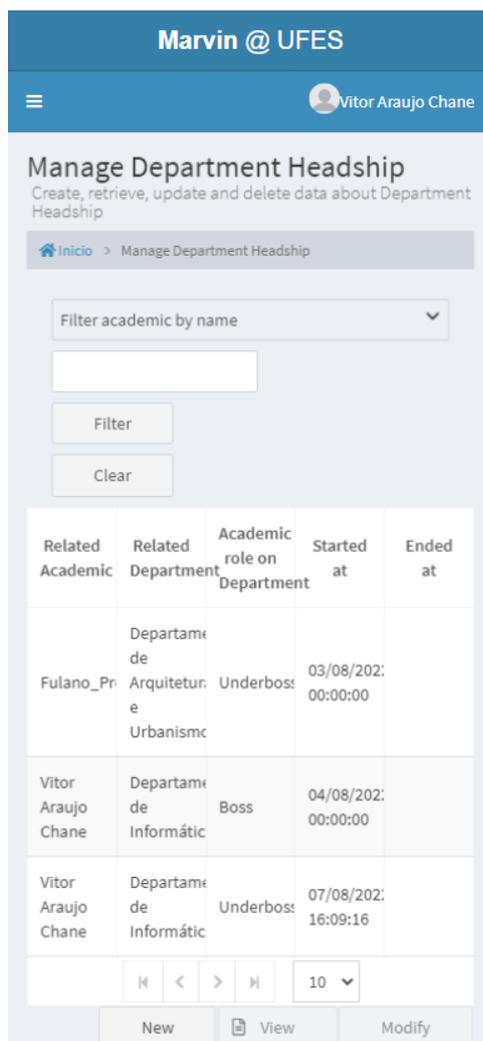
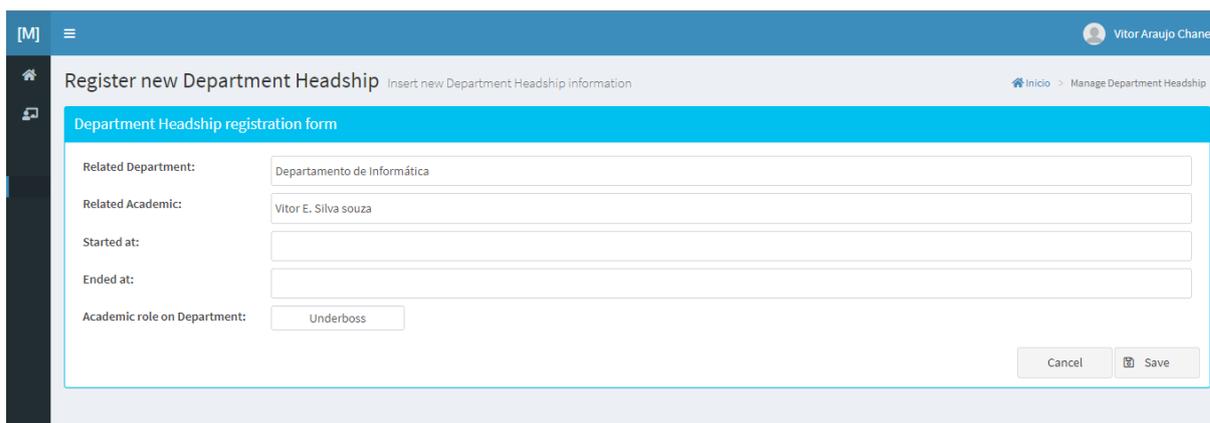


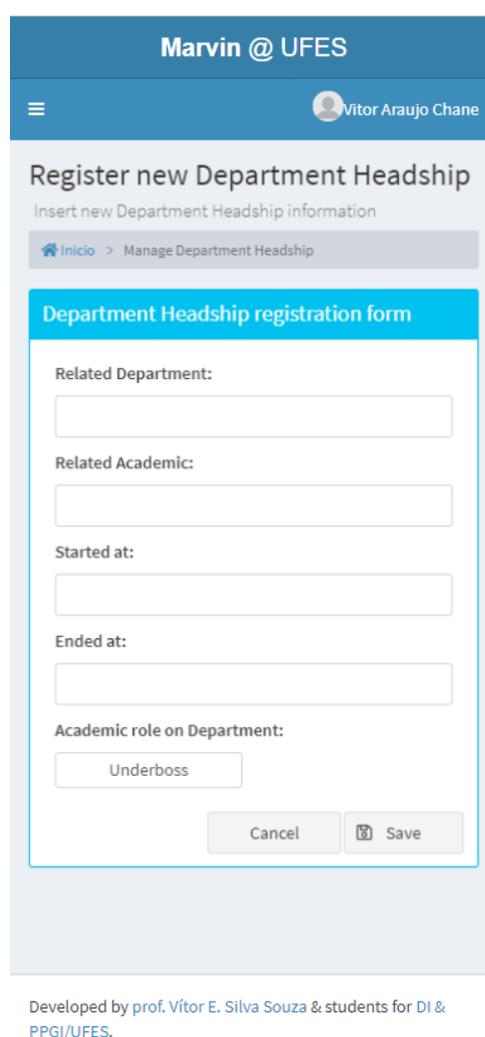
Figura 41 – Marvin - Listagem de Chefias de Departamento - Versão Mobile

Para a criação de uma nova Chefia de Departamento clicamos em *New* e o formulário de criação será mostrado. A criação de uma nova Chefia de Departamento requer a seleção de um Departamento e um Acadêmico com o papel Professor previamente criados, além de obrigatoriamente ser necessário o preenchimento dos campos de data de início e se o Acadêmico em questão é Chefe do Departamento ou não (subchefe neste último caso). Após inserirmos o início dos nomes do Departamento e do Acadêmico desejados, uma lista contendo seus respectivos resultados será mostrada possibilitando a escolha, como na Figura 42 e sua versão *mobile* na Figura 43.



The screenshot shows a desktop web interface for the Marvin system. At the top, there is a navigation bar with a menu icon, the user name 'Vitor Araujo Chane', and a breadcrumb trail: 'Inicio > Manage Department Headship'. The main heading is 'Register new Department Headship' with a sub-heading 'Insert new Department Headship information'. Below this is a 'Department Headship registration form' with the following fields: 'Related Department:' (text input with 'Departamento de Informática'), 'Related Academic:' (text input with 'Vitor E. Silva souza'), 'Started at:' (empty text input), 'Ended at:' (empty text input), and 'Academic role on Department:' (dropdown menu with 'Underboss' selected). At the bottom right of the form are 'Cancel' and 'Save' buttons.

Figura 42 – Marvin - Criação de Nova Chefia de Departamento



The screenshot shows the mobile version of the Marvin system. The header includes 'Marvin @ UFES' and the user name 'Vitor Araujo Chane'. The main heading is 'Register new Department Headship' with a sub-heading 'Insert new Department Headship information'. Below this is a breadcrumb trail: 'Inicio > Manage Department Headship'. The main heading for the form is 'Department Headship registration form'. The form fields are: 'Related Department:' (empty text input), 'Related Academic:' (empty text input), 'Started at:' (empty text input), 'Ended at:' (empty text input), and 'Academic role on Department:' (dropdown menu with 'Underboss' selected). At the bottom right of the form are 'Cancel' and 'Save' buttons. At the bottom of the page, there is a footer: 'Developed by prof. Vitor E. Silva Souza & students for DI & PPGI/UFES.'

Figura 43 – Marvin - Criação de Nova Chefia de Departamento - Versão Mobile

Após a criação da Chefia do Departamento, ela passará a ser listada na página de listagem de chefias de departamento (Figura 40). No caso de haver uma tentativa de criação de uma nova Chefia de Departamento onde o Acadêmico em questão não possui o papel Professor ou não esteja com uma participação ativa em algum departamento,

o sistema mostrará um erro alegando que é necessário que o Acadêmico cumpra essas regras antes de poder ser vinculado como Chefe ou Subchefe de um Departamento. Essas regras variam de acordo com qual tipo de acesso esteja querendo se dar a um determinado acadêmico. A criação de uma Secretaria de Departamento por exemplo, só é possível se o Acadêmico em questão possuir o papel de Staff (Servidor Técnico Administrativo), e assim sucessivamente para as outras classes do sistema.

Para os Casos de Uso onde as funcionalidades não são CRUDs (*create, retrieve, update, delete*) foram criadas funcionalidades em que fosse respeitado o que foi modelado no Documento de Especificação de Requisitos por meio das histórias de usuário criadas. O Documento de Especificação de Requisitos referente a este trabalho se encontra anexado ao final do documento, na parte de apêndices.

A seguir são apresentadas as funcionalidades de indicação e finalização de participação por parte de um Chefe de Departamento. Como é mostrado na Figura 44, após efetuar o login, poderá ser visto no menu esquerdo todas as funcionalidades implementadas e que já existem no sistema Marvin² referentes ao papel de Chefe do Departamento.

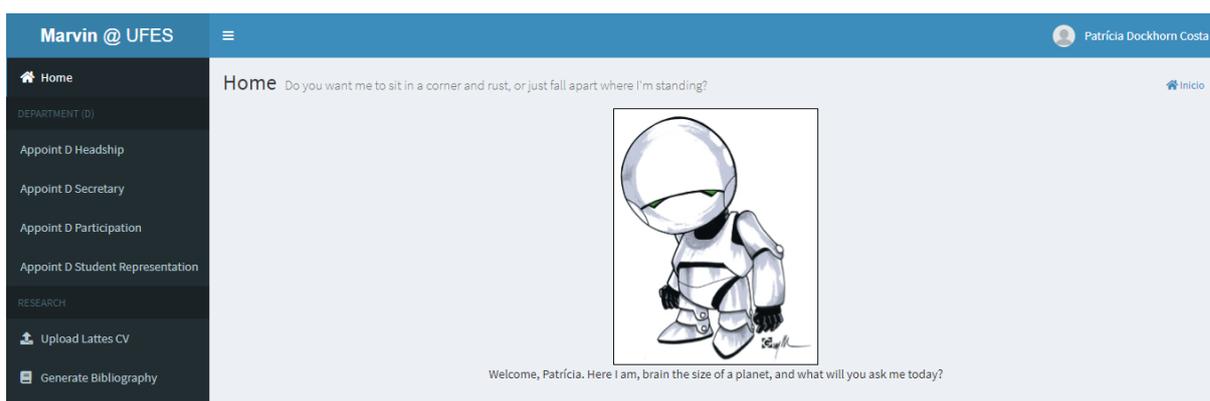


Figura 44 – Marvin - Página inicial - Acesso de Chefe do Departamento

A partir daqui, já teremos acesso a funcionalidades de indicação de participação para outras entidades como chefias do departamento, secretarias do departamento, participações no departamento e representação discente no departamento. Clicando no item referente a indicar nova chefia de departamento (*Appoint D. Headship*) somos redirecionados a uma página bem similar com as outras páginas que já vimos, porém com ações diferentes. Esta página pode ser visualizada na Figura 45 e em sua versão *mobile* na Figura 46.

² <<https://github.com/dwws-ufes/marvin/>>

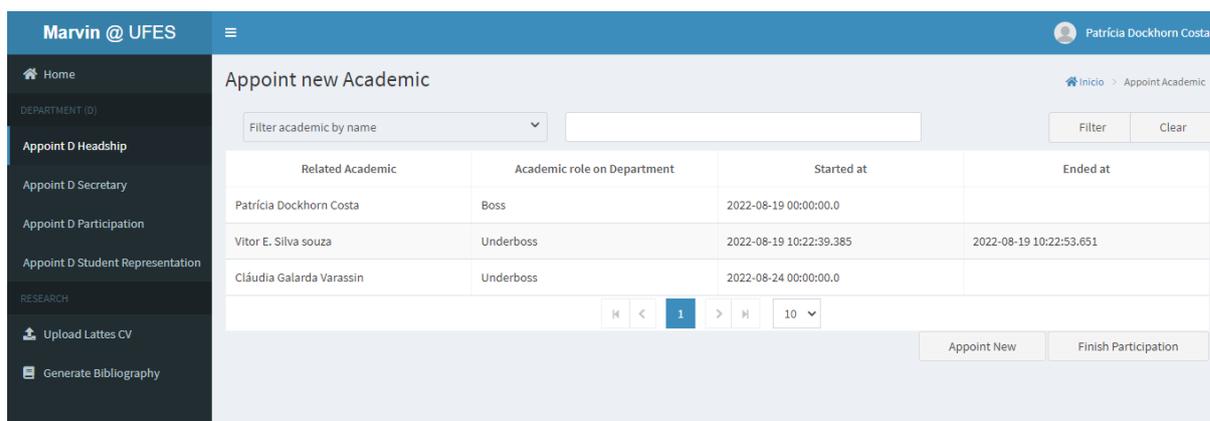


Figura 45 – Marvin - Listagem de Chefes do Departamento

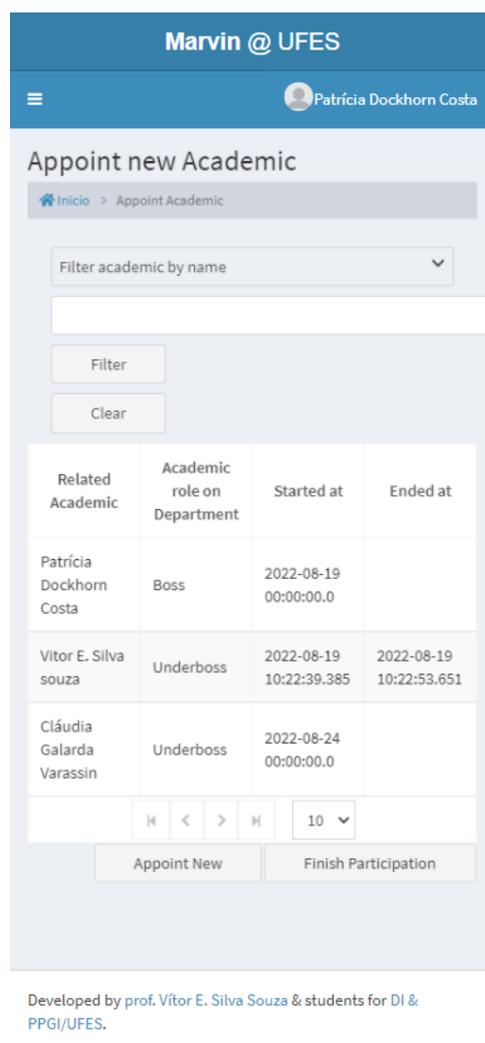


Figura 46 – Marvin - Listagem de Chefes do Departamento - Versão mobile

Uma lista com todos os chefes e subchefes ativos e inativos do departamento que estejam vinculados ao mesmo departamento do usuário logado será exibida, junto com algumas informações como: nome, papel, data de início e data de finalização. Um objeto que possua data de início mas não possua data de finalização é um chefe ou sub-chefe ativo do departamento, a depender do seu papel.

Para a indicação de uma nova Chefia de Departamento clicamos em *Appoint New* e o formulário de criação será mostrado. A criação de uma nova Chefia de Departamento requer a seleção de um Acadêmico com o papel Professor previamente criado, podendo-se pesquisar por nome, CPF ou e-mail do acadêmico, e se o Acadêmico em questão é Chefe do Departamento ou não (subchefe neste último caso). O departamento vinculado a este novo chefe/sub-chefe será o mesmo do usuário logado, e a data de início será a mesma do momento de criação. Esta página de formulário é apresentada na Figura 47 e em sua versão mobile na Figura 48.

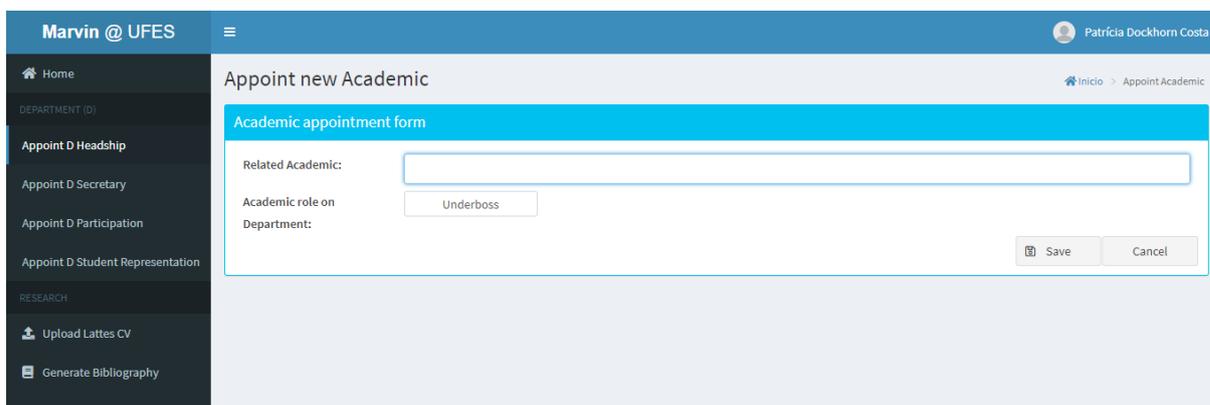
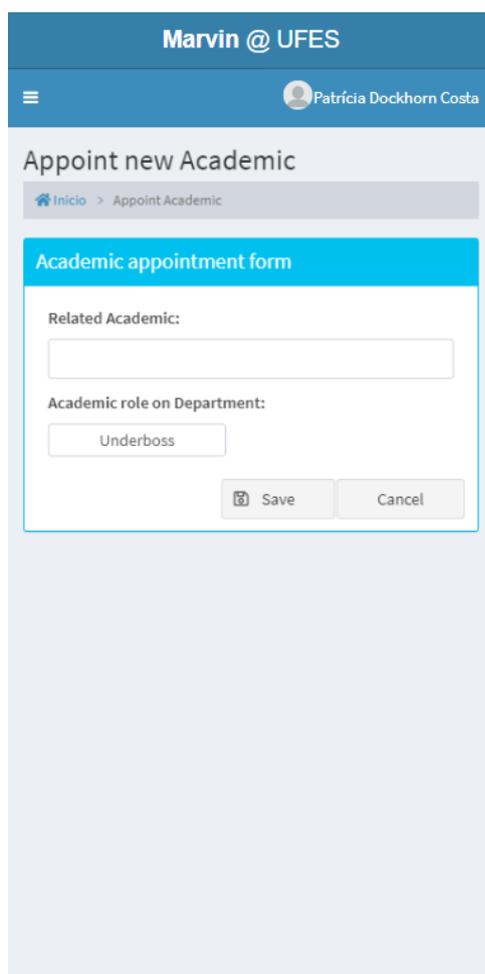


Figura 47 – Marvin - Formulário de indicação de Chefes do Departamento

Após o preenchimento do campo, poderá se clicar em *Save* caso se queira indicar a nova chefia do departamento ou *Cancel* caso queira abortar a indicação. Nos dois casos o sistema irá redirecionar para a página de listagem de chefias, mas caso *Save* for clicado, o novo chefe/sub-chefe será adicionado à lista de chefias. Em caso de antes de clicar em *Save* existir algum chefe ou subchefe ativo, essa atribuição será finalizada com data e hora atual a depender da escolha da nova indicação de chefia.

Por fim, também existe a possibilidade de selecionar diretamente o acadêmico da lista e finalizar diretamente (sem ter que criar outro acadêmico para isso) sua chefia. Após selecionar o acadêmico na lista e clicar em *Finish Participation*, o acadêmico atual é movido para uma listagem de finalização de participação, onde o usuário pode decidir se quer ou não finalizar a chefia clicando em *Confirm* para confirmar a finalização ou *Cancel* para cancelar a finalização. Esta ação pode ser vista na Figura 49. Caso opte por confirmar, a data de finalização do objeto em questão é salva com a data e hora atual.



Developed by prof. Vitor E. Silva Souza & students for DI & PPGI/UFES.

Figura 48 – Marvin - Formulário de indicação de Chefes do Departamento - Versão mobile

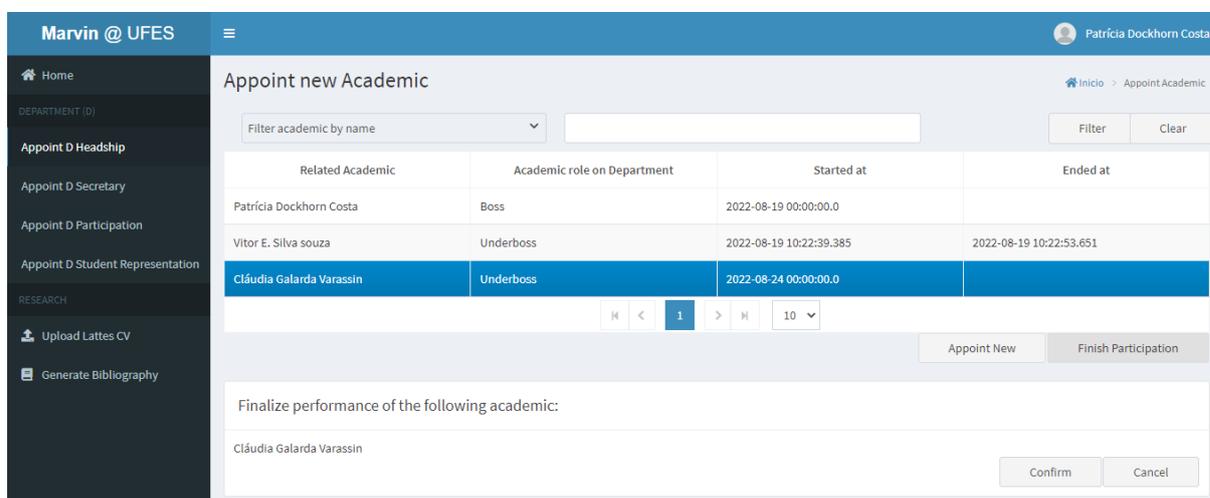


Figura 49 – Marvin - Listagem de Chefes do Departamento - Finalização de Chefia

6 Conclusão

Neste capítulo serão apresentadas as considerações finais sobre o trabalho desenvolvido, sendo apresentadas suas principais contribuições, limitações, lições aprendidas durante o desenvolvimento, dificuldades enfrentadas e perspectivas de trabalhos futuros. A Seção 6.1 apresenta as contribuições do trabalho desenvolvido, assim como as dificuldades encontradas. A Seção 6.2 mostra uma reflexão pessoal sobre a ferramenta, fazendo um paralelo entre o que foi desenvolvido e o que era almejado de início. Por fim dou opiniões de o que poderá ser melhorado no futuro dado a experiência adquirida.

6.1 Considerações Finais

Dada a importância de um sistema acadêmico onde alunos, professores e servidores coexistam e tenham acesso a funcionalidades específicas de cada papel, é visto que dentro do sistema Marvin¹ isso era inexistente. Surgiu então a necessidade da criação de um módulo para este sistema principal onde essa necessidade fosse atendida. Importante ressaltar que o sistema Marvin, assim como seus módulos, são comumente desenvolvidos por alunos de graduação do DI/UFES como parte de seus projetos finais de graduação, e no final espera-se ter um sistema integrado por esses módulos desenvolvidos onde toda a universidade possa usufruir.

Esta documentação foi produzida de acordo com os padrões de Engenharia de Software. Foram levantados todos os requisitos junto dos *stakeholders*, sendo posteriormente feita a análise e documentação de tais requisitos em um Documento de Especificação de Requisitos. Neles foram inseridos informações sobre requisitos funcionais em forma de estórias de usuário, requisitos não funcionais, regras de negócio, definição dos atores, criação dos casos de uso e modelagem por diagrama de classes através da ferramenta Astah UML. Após isso, foi criado o Documento de Projeto, contendo informações a respeito das tecnologias utilizadas no sistema, assim como a arquitetura de software e os modelos propostos na Seção 2.4, seguindo a abordagem FrameWeb.

Certamente a maior dificuldade encontrada foi a implementação de um sistema utilizando Java EE e suas tecnologias (JSF, CDI, JPA), tendo em vista que minha experiência até então tinha sido com desenvolvimento web utilizando *frameworks* como Angular e Ionic que usam JavaScript/TypeScript como linguagem de programação e CSS e bibliotecas relacionadas para estilização. Várias horas foram gastas assistindo tutoriais, lendo documentações e programando para fixação do conteúdo. Além disso, o *FrameWeb*

¹ <<https://github.com/dwvs-ufes/marvin/>>

não é um conceito fácil principalmente no que se diz na confecção dos modelos. A confecção dos Documentos de Especificação de Requisitos e Documento de Projeto também foram de grande valia para o aprendizado do ciclo de desenvolvimento de um software totalmente do zero.

Por fim, apesar de considerar que existem algumas disciplinas dentro do curso que não agregaram muito na minha formação e que poderiam ser substituídas por outras que foquem em demandas do mercado global, é importante citar como algumas disciplinas como Engenharia de Software, Projeto de Sistemas, Modelagem Organizacional e Gerência de Projetos, Programações, Lógicas e Banco de Dados foram extremamente importantes para minha carreira profissional e como ajudaram no desenvolvimento deste trabalho. Após a experiência adquirida nesta monografia, também é interessante citar o quão é importante a confecção de um trabalho que proporcione juntar o conhecimento de vários anos de curso.

6.2 Trabalhos Futuros

Como qualquer software, é sempre possível melhorar alguns detalhes e adicionar novas funcionalidades. O ciclo de desenvolvimento nunca acaba e é importante que o sistema esteja sempre em manutenção e atualização, visando a melhoria contínua, a identificação de problemas que possam vir, atualizações de tecnologias visando a melhoria do desempenho, etc.

A partir dos resultados alcançados, foram encontradas algumas limitações, o que dá margem para a realização de trabalhos futuros, a saber:

- Consumir API's externas da universidade, visando não dar controle de criação de entidades para os atores que podem ter acessos de criação. Essas entidades somente poderiam ser criados se existissem dentro dessa API, sendo suas informações importadas da mesma. Assim como integração para API's que retornem professores, estudantes e servidores com suas informações corretas, pois desta maneira, a possibilidade de erros e de criação de entidades inexistentes dentro do contexto da universidade seria drasticamente reduzida;
- Integração do módulo *Macs - Sistema de controle de acessos do Marvin* aos outros módulos que estão sendo implementados, para assim possibilitar o acesso de funcionalidades baseadas em papéis para estes outros módulos. Um exemplo seria os módulos AlocaWeb e BibLattes, que foram módulos produzidos por [Salvatore \(2016\)](#). Neste caso, professores, servidores técnico administrativos e cientistas (alunos e professores) deveriam possuir acesso ao Marvin para auxiliá-los em atividades de alocação de salas de aula e de controle de publicações. Para isso, o módulo Macs

serviria como base para criação de regras de acesso para acessar as funcionalidades relativas a esses módulos, como por exemplo um servidor técnico administrativo ou um professor relacionados a um curso de graduação podendo alocar uma sala de aula;

- Como os requisitos do sistema foram baseados em um escopo do Departamento de Informática da UFES durante a pandemia da COVID-19, não foi possível visitar outros departamentos e ver qual era a organização deles, assim como as organizações nos colegiados, cursos, programas de pós-graduação e etc. Apesar de ter sido feito uma análise em outros centros de ensino pelo site da UFES e dentro desta impossibilidade de visitaç o, pode ser que nesses outros departamentos existam outros tipos de organizaç es que no futuro possam vir a serem modeladas dentro do sistema Marvin, como uma extens o deste m dulo;
- Incluir funcionalidade de finalizaç o de participaç o autom tica a partir de uma data limite;
- Reunir membros da comunidade acad mica com experi ncia em seguranç a para a realizaç o de testes de esforço e verificar se conseguem burlar o acesso ao sistema. Observar os usu rios finais utilizando o sistema e verificar se o acesso por login e senha n o apresenta erros e se eles possuem acesso as funcionalidades relativas ao perfil;
- Novos papeis, como por exemplo Ex-aluno e Ex-professor, que assim poder o ter acesso a funcionalidades como “Recuperar hist rico” ou “Recuperar notas”. Ou ent o novas funcionalidades onde isso seja poss vel atrav s da verificaç o pelo hist rico (caso a data de finalizaç o esteja preenchida);
- Criaç o de um tutorial cujo foco ser  a apresentaç o da implementaç o do sistema, mostrando algumas linhas de c digo e explicando qual deveria ser o comportamento ao final. Explicar tamb m como os *frameworks* se comunicam a n vel de c digo.

Referências

- ALUR, D.; CRUPI, J.; MALKS, D. *Core J2EE Patterns: Best Practices and Design Strategies*. 2nd. ed. [S.l.]: Prentice Hall / Sun Microsystems Press, 2003. Citado na página 23.
- CAMPOS, S. L.; SOUZA, V. E. S. FrameWeb Editor: Uma Ferramenta CASE para suporte ao Método FrameWeb. In: *Anais do 16^o Workshop de Ferramentas e Aplicações, 23^o Simpósio Brasileiro de Sistemas Multimedia e Web (WFA/WebMedia 2017)*. Gramado, RS, Brazil: SBC, 2017. p. 199–203. Citado 3 vezes nas páginas 7, 22 e 23.
- FALBO, R. d. A. *Engenharia de Software - Notas de Aula*. [S.l.: s.n.], 2014. 4 p. Citado 4 vezes nas páginas 16, 17, 18 e 19.
- FOWLER, M. *Patterns of Enterprise Application Architecture*. 1. ed. [S.l.]: Addison-Wesley, 2002. ISBN 9780321127426. Citado 2 vezes nas páginas 22 e 39.
- KOTONYA, G.; SOMMERVILLE, I. *Requirements Engineering: Processes and Techniques*. 1st. ed. [S.l.]: Wiley Publishing, 1998. Citado 2 vezes nas páginas 7 e 18.
- MURUGESAN, S. et al. Web engineering: A new discipline for development of web-based systems. Springer, p. 3–13, 2000. Citado na página 20.
- NASCIMENTO, B. *SAE - Sistema de Acompanhamento de Egressos*. 2015. 27 p. Citado na página 21.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software - Uma Abordagem Profissional*. 8th. ed. [S.l.]: AMGH EDITORA LTDA, 2016. Citado 3 vezes nas páginas 17, 18 e 19.
- SALVATORRE, T. *AlocaWeb e BibLattes – Módulos do sistema Marvin*. 2016. Citado 2 vezes nas páginas 44 e 70.
- SOMMERVILLE, I. e. a. *Engenharia de Software 9^o edição*. 9th. ed. [S.l.]: Pearson, 2011. Citado 2 vezes nas páginas 16 e 17.
- SOUZA, V. E. S. The FrameWeb Approach to Web Engineering: Past, Present and Future. In: ALMEIDA, J. P. A.; GUIZZARDI, G. (Ed.). *Engineering Ontologies and Ontologies for Engineering*. 1. ed. Vitória, ES, Brazil: NEMO, 2020. cap. 8, p. 100–124. ISBN 9781393963035. Disponível em: <<http://purl.org/nemo/celebratingfalbo>>. Citado 2 vezes nas páginas 14 e 22.

Apêndices



Documento de Requisitos de Sistema

Macs - Sistema de controle de acessos do Marvin

Vitória, ES

2022

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Vitor Araujo Chane	06/02/2022	Versão inicial do documento incluindo seções 1-7.
1.0	Vitor Estevão Silva Souza	18/02/2022	Revisa seções 1-4, comentários seções 5-7.
1.1	Vitor Araujo Chane	27/02/2022	Ajustes e finalizando seção 8.
1.2	Vitor Araujo Chane	03/03/2022	Ajustes de modelos e informações.
1.2	Vitor Estevão Silva Souza	04/03/2022	Revisa seções 3-7.
1.3	Vitor Araujo Chane	07/03/2022	Ajustes de modelos e informações.
1.3	Vitor Estevão Silva Souza	10/03/2022	Revisão na seção 8.
1.4	Vitor Araujo Chane	13/03/2022	Ajustes de imagens.
1.5	Vitor Araujo Chane	14/03/2022	Ajustes de imagens e pequenas alterações.
1.5	Vitor Estevão Silva Souza	14/03/2022	Revisão do documento de requisitos.
1.6	Vitor Araujo Chane	21/07/2022	Ajustes em detalhes que passaram despercebidos. Adicionando registro de alterações de acordo com histórico do Bitbucket.

1 Introdução

Este documento apresenta os requisitos de usuário e a análise dos requisitos do módulo Macs, responsável pelo controle de acessos do sistema Marvin. Para a análise de requisitos deste módulo, foram utilizadas as técnicas de modelagem de classes em UML, bem como modelagem de casos de uso e levantamento de requisitos com estórias de usuário.

Este documento está organizado da seguinte forma: A Seção 2 contém uma descrição do propósito do sistema. A Seção 3 contém uma descrição do minimundo, apresentando o problema. A Seção 4 descreve os requisitos levantados junto aos *stakeholders*. A Seção 5 explica a divisão em subsistemas, descrevendo brevemente cada um deles. A Seção 6 apresenta o modelo de casos de uso, incluindo descrições de atores, os diagramas de casos de uso e suas respectivas descrições. A Seção 7 traz os modelos conceituais estruturais do sistema na forma de diagramas de classes. Por fim, a Seção 8 detalha o dicionário do projeto, contendo as definições das classes identificadas.

2 Descrição do Propósito do Sistema

Um sistema onde alunos, professores e funcionários possam gerenciar tarefas de ensino e pesquisa em uma universidade se tornou algo indispensável nos dias atuais. Com isso em mente e com a crescente expansão do mercado Web, surgiu o sistema Marvin, feito e mantido por alunos e professores da Universidade Federal do Espírito Santo (UFES).

A ideia do Marvin é servir a todos os departamentos de toda a universidade, e para isso, devemos ter em mente que existirão vários usuários e nem sempre seus papéis dentro deste sistema será o mesmo. É estranho pensar em um Discente que tenha os mesmos privilégios de acessos que um Docente, por exemplo. Fica claro que um módulo muito importante desse sistema é o módulo de controle de acessos dos usuários que utilizarão o sistema.

O módulo Macs (*Marvin Access Control System*) é uma iniciativa que pretende facilitar o acesso de um dado usuário, com um dado “papel” dentro desse sistema, a funcionalidades específicas daquele papel. É de extrema importância que estes acessos sejam concedidos de maneira simples e contra qualquer tipo de erros.

3 Descrição do Minimundo

Dentro do Marvin, que é o sistema principal com o propósito de servir a universidade, composto de ferramentas capazes de auxiliar em tarefas administrativas comuns no contexto da UFES e não cobertas pelos atuais sistemas do NTI, se faz necessário um módulo onde tenhamos o gerenciamento dos papéis específicos de cada usuário.

O objetivo principal do Macs (*Marvin Access Control System*) ou Sistema de Controle de Acessos do Marvin, é justamente possibilitar o gerenciamento dos papéis de cada usuário dentro do sistema principal, garantindo que cada usuário possua acesso a funcionalidades específicas de acordo com seu papel dentro da universidade.

A universidade é composta por centros de ensino, que possuem departamentos, cursos de graduação e programas de pós-graduação. Os departamentos e colegiados de cursos e programas devem possuir uma quantidade específica de pessoas com papéis de discentes, docentes, funcionários ou visitantes. Dentro de cada papel citado, temos outros papéis, que são descritos a seguir:

Um(a) docente pode ser...

- Lotado(a) em um departamento: todo(a) professor(a) com vínculo à UFES deve ser lotado(a) em um departamento;
- Chefe de Departamento: eleito(a) por votação com mandato de 2 anos, o(a) chefe do departamento é o(a) professor(a) responsável por coordenar e supervisionar todas as atividades da competência do departamento;
- Subchefe de Departamento: eleito(a) por votação com mandato de 2 anos, o(a) subchefe do departamento é o(a) professor(a) que substituirá o(a) chefe do departamento nas suas faltas e impedimentos;
- Coordenador(a) de curso de graduação: eleito(a) por votação com mandato de 2 anos, o(a) coordenador(a) de curso de graduação é o(a) professor(a) responsável por representar oficialmente o colegiado do curso de graduação;
- Subcoordenador(a) de curso de graduação: eleito(a) por votação com mandato de 2 anos, o(a) subcoordenador(a) de curso de graduação é o(a) professor(a) responsável por representar oficialmente o colegiado do curso de graduação na ausência do(a) coordenador(a);
- Coordenador(a) de programa de pós-graduação: eleito(a) por votação com mandato de 2 anos, o(a) coordenador(a) de programa de pós-graduação é o(a) professor(a) responsável por representar oficialmente o colegiado do programa de pós-graduação;
- Subcoordenador(a) de programa de pós-graduação: eleito(a) por votação com mandato de 2 anos, o(a) subcoordenador(a) de programa de pós-graduação é o(a) professor(a) responsável por representar oficialmente o colegiado do programa de pós-graduação na ausência do(a) coordenador(a);
- Coordenador(a) de estágio de curso de graduação: eleito(a) por votação com mandato de 2 anos, é o(a) professor(a) responsável pelo acompanhamento dos estudantes, em termos de seu rendimento escolar na UFES, durante o período de estágio.

Um discente pode ser...

- Matriculado(a) em um curso de graduação: todo(a) discente de graduação em situação regular estará matriculado(a) em um curso de graduação;
- Matriculado(a) em um curso de pós-graduação: todo(a) discente de pós-graduação em situação regular estará matriculado(a) em um curso de pós-graduação;
- Representante discente em departamento: eleito(a) por votação com mandato de 2 anos, é o(a) discente com função de representar, dentro do departamento, os interesses do conjunto dos estudantes;
- Representante discente em curso de graduação: eleito(a) por votação com mandato de 2 anos, é o(a) discente com função de representar, dentro do curso de graduação, os interesses do conjunto dos estudantes;
- Representante discente em programa de pós-graduação: eleito(a) por votação com mandato de 2 anos, é o(a) discente com função de representar, dentro do programa de pós-graduação, os interesses do conjunto dos estudantes.

Um funcionário pode ser...

- Lotado(a) no departamento: funcionário(a) encarregado da execução de determinados serviços no departamento (ex.: secretário(a), suporte em TI, etc.);
- Secretário(a) do departamento: funcionário(a) encarregado da execução dos serviços administrativos do departamento;
- Secretário(a) de curso de graduação: funcionário(a) encarregado da execução dos serviços administrativos do curso de graduação;
- Secretário(a) de programa de pós-graduação: funcionário(a) encarregado da execução dos serviços administrativos do programa de pós-graduação.

Apenas o administrador do sistema e usuários com cargos hierárquicos de topo como secretários e chefes/coordenadores podem conceder outros papéis a um usuário do sistema, de modo que seja respeitado o que cada usuário pode acessar.

Com os acessos concedidos, espera-se que cada usuário consiga utilizar as ferramentas específicas a seu papel dentro do sistema.

4 Definição de Requisitos

Tomando por base o contexto do sistema descrito na Seção 3 e considerando como principais *stakeholders*, professores, estudantes, técnicos administrativos e visitantes, foram identificados os seguintes requisitos de usuário e regras de negócio.

As histórias de usuário são apresentadas na Tabela 1 e os requisitos não-funcionais globais (ou seja, aqueles que não são caracterizados como critérios de aceitação de histórias de usuário específicas) na Tabela 2.

Tabela 1 – Estórias de Usuário.

ID:	US-1	Depende:		Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar centros de ensino no sistema, para que departamentos, cursos de graduação e programas de pós graduação possam ser associados a eles.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, deve ser informado: nome; – CA2: A consulta pode ser feita por: nome; – CA3: Não é permitido ter dois centros de ensino com mesmo nome; – CA4: Centros de Ensino podem ser excluídos do sistema apenas se não estiverem associados a outros objetos; – CA5: Deve ser requerida confirmação para exclusões. 				
ID:	US-2	Depende:	US-1	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar departamentos no sistema, para que acadêmicos(as) possam ser lotados nesses departamentos, bem como se associem de alguma maneira a eles (ex.: chefe, representante discente, etc.).				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, devem ser informados: centro de ensino e nome; – CA2: A consulta pode ser feita por: nome; – CA3: Não é permitido ter dois departamentos com mesmo nome; – CA4: Departamentos podem ser excluídos do sistema apenas se não estiverem associados a outros objetos além do centro; – CA5: Deve ser requerida confirmação para exclusões. 				
ID:	US-3	Depende:	US-1	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar programas de pós-graduação associados a um centro de ensino, para que cursos de pós-graduação possam ser associados a eles, bem como acadêmicos(as) se associem de alguma maneira a eles (ex.: coordenador, representante discente, etc.).				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, devem ser informados: centro de ensino, nome; – CA2: A consulta pode ser feita por: nome; – CA3: Não é permitido ter dois programas de pós-graduação com mesmo nome; – CA4: Programas de pós-graduação podem ser excluídos do sistema apenas se não estiverem associados a outros objetos além do centro de ensino; – CA5: Deve ser requerida confirmação para exclusões.
--------------------------------	--

ID:	US-4	Depende:	US-1	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar cursos de graduação associados a um centro de ensino, para que os acadêmicos(as) se associem de alguma maneira a eles (ex.: coordenador, estudante matriculado, etc.).				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, devem ser informados: centro de ensino, nome, código e turno; – CA2: A consulta pode ser feita por: código, nome e turno; – CA3: Não é permitido ter dois cursos de graduação com mesmo nome e turno; – CA4: Cursos de graduação podem ser excluídos do sistema apenas se não estiverem associados a outros objetos além do centro; – CA5: Deve ser requerida confirmação para exclusões. 				

ID:	US-5	Depende:	US-3	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar cursos de pós-graduação associados a um programa de pós-graduação para que os acadêmicos(as) se associem de alguma maneira a eles (ex.: coordenador, estudante matriculado, etc.).				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, devem ser informados: programa de pós-graduação, nome, código e nível; – CA2: A consulta pode ser feita por: código, nome, turno e nível; – CA3: Não é permitido ter dois cursos de pós-graduação com mesmo nome e turno; – CA4: Cursos de pós-graduação podem ser excluídos do sistema apenas se não estiverem associados a outros objetos além do programa de pós-graduação; – CA5: Deve ser requerida confirmação para exclusões. 				

ID:	US-6	Depende:		Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar acadêmicos(as) no sistema, para que eles possam usar as funcionalidades do mesmo.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, devem ser informados: nome, e-mail, cpf, data de nascimento (opcional), id Lattes (opcional) e senha (opcional); – CA2: Não é permitido ter dois(uas) acadêmicos(as) com mesmo e-mail ou mesmo CPF; – CA3: A consulta pode ser feita por: nome ou e-mail; – CA4: Acadêmicos(as) podem ser excluídos do sistema apenas se não estiverem associados a outros objetos; – CA5: Deve ser requerida confirmação para exclusões.
--------------------------------	--

ID:	US-7	Depende:	US-6	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero atribuir papéis aos acadêmicos(as) cadastrados no sistema, para que eles possam ter acesso a funcionalidades específicas a seus papéis.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O acadêmico(a) deve estar válido no sistema; – CA2: Para atribuir um papel, devem ser informados: o acadêmico e um ou mais papéis dentre os possíveis: <i>Professor</i>, <i>Staff</i>, <i>Student</i> ou <i>Visitor</i>. 				

ID:	US-8	Depende:	US-6, US-7, US-2	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) em cargos de chefia do departamento, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema, deve ter o papel <i>Professor</i> e deve estar lotado no departamento em questão; – CA2: Para cadastrar, devem ser informados: acadêmico(a), departamento, cargo, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), departamento, cargo ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-9	Depende:	US-6, US-7, US-2	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar secretários(as) do departamento, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Staff</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), departamento, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), departamento ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-10	Depende:	US-6, US-7, US-2	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar estudantes em representação discente no departamento, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Student</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), departamento, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), departamento ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-11	Depende:	US-6, US-7, US-2	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) ou servidores(as) técnico-administrativos(as) em um departamento, para que esses possam exercer no sistema funções relacionadas a este departamento.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Staff</i> ou <i>Professor</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), departamento, cargo, data de início e data de fim da lotação (esta última, quando nula, indica que a lotação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), cargo ou período de lotação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-12	Depende:	US-6, US-7, US-4	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) em cargos de coordenação de curso de graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Professor</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), curso de graduação, cargo, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), curso de graduação, cargo ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-13	Depende:	US-6, US-7, US-4	Prioridade:	Alta
------------	-------	-----------------	------------------	--------------------	------

Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar secretários(as) de curso de graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Staff</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), curso de graduação, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), curso de graduação ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões.

ID:	US-14	Depende:	US-6, US-7, US-4, US-16	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar estudantes em representação discente em curso de graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema, deve ter o papel <i>Student</i> e estar matriculado no curso de graduação em questão; – CA2: Para cadastrar, devem ser informados: acadêmico(a), curso de graduação, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), curso de graduação e período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-15	Depende:	US-6, US-7, US-4	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) no cargo de coordenadores(as) de estágio em curso de graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Professor</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), curso de graduação, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), curso de graduação e período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-16	Depende:	US-6, US-7, US-4	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar a matrícula de estudantes em cursos de graduação, para que eles(as) possam ter acesso às funcionalidades correspondentes aos estudantes no sistema.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Student</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), curso de graduação, matrícula, data de início e data de fim da matrícula (esta última, quando nula, indica que o(a) estudante está atualmente matriculado(a)); – CA3: A consulta pode ser feita por: nome do acadêmico(a), matrícula, curso de graduação e período de matrícula; – CA4: Deve ser requerida confirmação para exclusões.
--------------------------------	---

ID:	US-17	Depende:	US-6, US-7, US-4	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) ou servidores(as) técnico-administrativos(as) em um colegiado de curso de graduação, para que esses possam exercer no sistema funções relacionadas a este colegiado.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Staff</i> ou <i>Professor</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), curso de graduação, data de início e data de fim da lotação (esta última, quando nula, indica que a lotação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a) ou período de lotação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-18	Depende:	US-6, US-7, US-3	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores em cargos de coordenação de programa de pós-graduação, para que esses possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema, deve ter o papel <i>Professor</i> e deve ser membro do programa de pós-graduação; – CA2: Para cadastrar, devem ser informados: acadêmico(a), programa de pós-graduação, cargo, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), programa de pós-graduação, cargo ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-19	Depende:	US-6, US-7, US-3	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar secretários(as) de programa de pós-graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Staff</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), programa de pós-graduação, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), programa de pós-graduação ou período de atuação; – CA4: Deve ser requerida confirmação para exclusões.
--------------------------------	--

ID:	US-20	Depende:	US-6, US-7, US-3, US-21	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar estudantes em representação discente em programa de pós-graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema, deve ter o papel <i>Student</i> e estar matriculado em algum curso de pós-graduação do programa em questão; – CA2: Para cadastrar, devem ser informados: acadêmico(a), programa de pós-graduação, data de início e data de fim da atuação (esta última, quando nula, indica que a atuação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a), programa de pós-graduação e período de atuação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-21	Depende:	US-6, US-7, US-3, US-5	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar a matrícula de estudantes em cursos de pós-graduação, para que eles(as) possam ter acesso às funcionalidades correspondentes aos estudantes no sistema.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Student</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), curso, matrícula, data de início e data de fim da matrícula (esta última, quando nula, indica que o(a) estudante está atualmente matriculado(a)); – CA3: A consulta pode ser feita por: nome do acadêmico(a), matrícula, curso e período de matrícula; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-22	Depende:	US-6, US-7, US-3, US-5	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) ou servidores(as) técnico-administrativos(as) em um colegiado de programa de pós-graduação, para que esses possam exercer no sistema funções relacionadas a este colegiado.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Staff</i> ou <i>Professor</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), programa de pós-graduação, data de início e data de fim da participação (esta última, quando nula, indica que a lotação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a) ou período de lotação; – CA4: Deve ser requerida confirmação para exclusões.
--------------------------------	--

ID:	US-23	Depende:	US-6, US-7, US-3	Prioridade:	Alta
Descrição:	Cadastro/CRUD: Como administrador(a) do sistema, eu quero cadastrar professores(as) em um programa de pós-graduação, para que esses possam exercer no sistema funções relacionadas a este programa.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: O(A) acadêmico(a) deve estar válido no sistema e deve ter o papel <i>Professor</i>; – CA2: Para cadastrar, devem ser informados: acadêmico(a), programa de pós-graduação, data de início e data de fim da participação (esta última, quando nula, indica que a lotação está ativa); – CA3: A consulta pode ser feita por: nome do(a) acadêmico(a) ou período de lotação; – CA4: Deve ser requerida confirmação para exclusões. 				

ID:	US-24	Depende:	US-6, US-7, US-2, US-8, US-9	Prioridade:	Média
Descrição:	Como chefe/sub-chefe ou secretário(a) de um departamento, eu quero registrar substituição de chefia do departamento, para que os(as) novos(as) chefes possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) lotados(as) no departamento em questão e com papel <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) chefes e sub-chefes atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a), cargo ou período de atuação; – CA4: Para terminar a atuação de um chefe ou sub-chefe, basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar uma nova chefia, escolhe se a indicação é de chefe ou sub-chefe, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual, o que também encerra com data atual a atuação do(a) atual acadêmico(a) que ocupa o referido cargo (se algum). 				

ID:	US-25	Depende:	US-6, US-7, US-2, US-8, US-9	Prioridade:	Média
------------	-------	-----------------	------------------------------	--------------------	-------

Descrição:	Como chefe/sub-chefe ou secretário(a) de um departamento, eu quero indicar os(as) secretários(as) do departamento, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) lotados(as) no departamento em questão e com papel <i>Staff</i>; – CA2: Inicialmente o sistema deve listar os(as) secretários(as) atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período de atuação; – CA4: Para terminar a atuação de um(a) secretário(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar um(a) novo(a) secretário(a), pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual.

ID:	US-26	Depende:	US-6, US-7, US-2, US-8, US-9	Prioridade:	Média
Descrição:	Como chefe/sub-chefe ou secretário(a) de um departamento, eu quero indicar os(as) representantes discentes do meu departamento, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Student</i>; – CA2: Inicialmente o sistema deve listar os(as) representantes atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período de atuação; – CA4: Para terminar a atuação de um(a) representante discente, basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar um(a) novo(a) representante discente, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual. 				

ID:	US-27	Depende:	US-6, US-7, US-2, US-8, US-9	Prioridade:	Média
Descrição:	Como chefe/sub-chefe ou secretário(a) de um departamento, eu quero registrar os professores(as) ou servidores(as) técnico-administrativos(as) lotados(as) no departamento, para que esses possam exercer no sistema funções relacionadas a este departamento.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Staff</i> ou <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) acadêmicos(as) atualmente lotados e aqueles(as) que já estiveram lotados no passado, com respectivas datas de lotação, cargo e, para os antigos, o motivo de desligamento; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a), cargo ou período de lotação; – CA4: Para terminar a lotação de um(a) acadêmico(a), basta escolhê-lo(a) na lista, indicar o motivo do desligamento (ex.: aposentadoria, redistribuição, etc.) e confirmar o encerramento, que é efetivado com data atual; – CA5: Para registrar a lotação de um(a) novo(a) acadêmico(a), pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a), indica o cargo (opcional para docentes) e confirma o registro, que é efetivado com a data atual.
--------------------------------	---

ID:	US-28	Depende:	US-6, US-7, US-4, US-12, US-13	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um curso de graduação, eu quero registrar substituição de coordenação de um curso de graduação, para que os(as) novos(as) coordenadores possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) membros(as) no colegiado do curso em questão e com papel <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) coordenadores e sub-coordenadores atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a), cargo ou período de atuação; – CA4: Para terminar a atuação de um(a) coordenador(a) ou e sub-coordenador(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar uma nova coordenação, escolhe se a indicação é de coordenador ou sub-coordenador, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual, o que também encerra com data atual a atuação do(a) atual acadêmico(a) que ocupa o referido cargo (se algum). 				

ID:	US-29	Depende:	US-6, US-7, US-4, US-12, US-13	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um curso de graduação, eu quero indicar os(as) secretários(as) de um curso de graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Staff</i>; – CA2: Inicialmente o sistema deve listar os(as) secretários(as) atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período de atuação; – CA4: Para terminar a atuação de um(a) secretário(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar um(a) novo(a) secretário(a), pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual.
--------------------------------	---

ID:	US-30	Depende:	US-6, US-7, US-4, US-12, US-13	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um curso de graduação, eu quero indicar os(as) representantes discentes de um curso de graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Student</i> e matriculados(a) no curso de graduação em questão; – CA2: Inicialmente o sistema deve listar os(as) representantes atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período de atuação; – CA4: Para terminar a atuação de um(a) representante discente, basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar um(a) novo(a) representante discente, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual. 				

ID:	US-31	Depende:	US-6, US-7, US-4, US-12, US-13	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um curso de graduação, eu quero registrar substituição de coordenação de estágio de um curso de graduação, para que os(as) novos(as) coordenadores possam ter acesso a funcionalidades específicas deste papel.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) coordenadores atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a), cargo ou período de atuação; – CA4: Para terminar a atuação de um(a) coordenador(a) ou e sub-coordenador(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar uma nova coordenação de estágio, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual, o que também encerra com data atual a atuação do(a) atual acadêmico(a) que ocupa o referido cargo (se algum).
--------------------------------	---

ID:	US-32	Depende:	US-6, US-7, US-4, US-12, US-13	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um curso de graduação, eu quero matricular, desligar e registrar conclusões de estudantes em um curso de graduação, para que os(as) matriculados(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser matriculados(as) apenas acadêmicos(as) com papel <i>Student</i>; – CA2: Inicialmente o sistema deve listar os(as) estudantes matriculados atualmente, com respectivas datas de início; – CA3: O sistema também deve listar os(as) estudantes que foram desligados ou que concluíram o curso, com respectivas datas de início e fim. – CA4: Pode-se pesquisar nestas listas por nome do(a) acadêmico(a) ou período de matrícula; – CA5: Para encerrar a matrícula de um estudante, basta escolhê-lo(a) na lista, indicar se concluiu ou foi desligado, no segundo caso o motivo do desligamento (ex.: abandono, jubramento, etc.) e confirmar o encerramento, que é efetivado com data atual; – CA6: Para matricular um(a) novo(a) estudante, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirmar a matrícula, que é efetivada com a data atual. 				

ID:	US-33	Depende:	US-6, US-7, US-4, US-12, US-13	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um curso de graduação, eu quero registrar professores(as) ou servidores(as) técnico-administrativos(as) para serem membros(as) de um colegiado de um curso de graduação, para que esses possam exercer no sistema funções relacionadas a este colegiado.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Staff</i> ou <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) acadêmicos(as) atualmente membros(as) e aqueles(as) que já foram membros(as) no passado, com respectivas datas de participação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período que foi membro(a); – CA4: Para terminar a participação de um(a) acadêmico(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para registrar a participação de um(a) novo(a) acadêmico(a), pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma o registro, que é efetivado com a data atual.
--------------------------------	--

ID:	US-34	Depende:	US-6, US-7, US-3, US-5, US-18, US-19	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um programa de pós-graduação, eu quero registrar substituição de coordenação de um programa de pós-graduação, para que os(as) novos(as) coordenadores possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) membros(as) do programa em questão e com papel <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) coordenadores e sub-coordenadores atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a), cargo ou período de atuação; – CA4: Para terminar a atuação de um(a) coordenador(a) ou e sub-coordenador(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar uma nova coordenação, escolhe se a indicação é de coordenador ou sub-coordenador, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual, o que também encerra com data atual a atuação do(a) atual acadêmico(a) que ocupa o referido cargo (se algum). 				

ID:	US-35	Depende:	US-6, US-7, US-3, US-5, US-18, US-19	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um programa de pós-graduação, eu quero indicar os(as) secretários(as) de um programa de pós-graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Staff</i>; – CA2: Inicialmente o sistema deve listar os(as) secretários(as) atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período de atuação; – CA4: Para terminar a atuação de um(a) secretário(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar um(a) novo(a) secretário(a), pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual.
--------------------------------	---

ID:	US-36	Depende:	US-6, US-7, US-3, US-5, US-18, US-19	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um programa de pós-graduação, eu quero indicar os(as) representantes discentes de um programa de pós-graduação, para que eles(as) possam ter acesso a funcionalidades específicas deste papel.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Student</i> e matriculados(a) em um curso do programa de pós-graduação em questão; – CA2: Inicialmente o sistema deve listar os(as) representantes atuais e aqueles(as) que já foram no passado, com respectivas datas de atuação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período de atuação; – CA4: Para terminar a atuação de um(a) representante discente, basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para indicar um(a) novo(a) representante discente, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma a indicação, que é efetivada com a data atual. 				

ID:	US-37	Depende:	US-6, US-7, US-3, US-5, US-18, US-19	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um programa de pós-graduação, eu quero matricular, desligar e registrar conclusões de estudantes em um curso de pós-graduação, para que os(as) matriculados(as) possam ter acesso a funcionalidades específicas deste papel.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser matriculados(as) apenas acadêmicos(as) com papel <i>Student</i>; – CA2: Inicialmente o sistema deve listar os(as) estudantes matriculados atualmente, com respectivas datas de início; – CA3: O sistema também deve listar os(as) estudantes que foram desligados ou que concluíram o curso, com respectivas datas de início e fim. – CA4: Pode-se pesquisar nestas listas por nome do(a) acadêmico(a) ou período de matrícula; – CA5: Para encerrar a matrícula de um estudante, basta escolhê-lo(a) na lista, indicar se concluiu ou foi desligado, no segundo caso o motivo do desligamento (ex.: abandono, jubramento, etc.) e confirmar o encerramento, que é efetivado com data atual; – CA6: Para matricular um(a) novo(a) estudante, pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirmar a matrícula, que é efetivada com a data atual.
--------------------------------	--

ID:	US-38	Depende:	US-6, US-7, US-3, US-5, US-18, US-19	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um programa de pós-graduação, eu quero registrar professores(as) ou servidores(as) técnico-administrativos(as) para serem membros(as) de um colegiado de programa de pós-graduação, para que esses possam exercer no sistema funções relacionadas a este colegiado.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Staff</i> ou <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) acadêmicos(as) atualmente membros(as) e aqueles(as) que já foram membros(as) no passado, com respectivas datas de participação; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a) ou período que foi membro(a); – CA4: Para terminar a participação de um(a) acadêmico(a), basta escolhê-lo(a) na lista e confirmar o encerramento, que é efetivado com data atual; – CA5: Para registrar a participação de um(a) novo(a) acadêmico(a), pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a) e confirma o registro, que é efetivado com a data atual. 				

ID:	US-39	Depende:	US-6, US-7, US-3, US-5, US-18, US-19	Prioridade:	Média
Descrição:	Como coordenador/sub-coordenador ou secretário(a) de um programa de pós-graduação, eu quero registrar professores(as) ou servidores(as) técnico-administrativos(as) para serem membros(as) de um programa de pós-graduação, para que esses possam exercer no sistema funções relacionadas a este programa.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Podem ser indicados(as) apenas acadêmicos(as) com papel <i>Staff</i> ou <i>Professor</i>; – CA2: Inicialmente o sistema deve listar os(as) acadêmicos(as) atualmente membros(as) e aqueles(as) que já foram no passado, com respectivas datas de lotação, tipo de vínculo (professor permanente, professor colaborador, secretário, etc.) e, para os antigos, o motivo de desligamento; – CA3: Pode-se pesquisar nesta lista por nome do(a) acadêmico(a), tipo de vínculo ou período de lotação; – CA4: Para terminar a lotação de um(a) acadêmico(a), basta escolhê-lo(a) na lista, indicar o motivo do desligamento (ex.: solicitado pelo(a) membro(a), não atendeu aos critérios de recredenciamento, etc.) e confirmar o encerramento, que é efetivado com data atual; – CA5: Para registrar a lotação de um(a) novo(a) acadêmico(a), pode-se pesquisar por nome, CPF ou e-mail do(a) acadêmico(a), indica o tipo de vínculo e confirma o registro, que é efetivado com a data atual.
--------------------------------	--

Tabela 2 – Requisitos Não Funcionais.

ID	Descrição	Categoria	Escopo	Prioridade
RNF-1	A plataforma deve estar disponível como uma aplicação Web, acessível a partir dos principais navegadores disponíveis no mercado.	Portabilidade	Sistema	Alta
RNF-2	O sistema deve controlar o acesso às suas funcionalidades por meio de autenticação e autorização, utilizando o papel de cada usuário para definir a que funcionalidades ele terá acesso.	Segurança	Sistema	Alta
RNF-3	É de extrema importância que informações sigilosas como senhas e afins sejam criptografadas.	Confidenciabilidade	Sistema	Alta
RNF-4	O sistema deve ser de fácil aprendizado, não sendo necessário nenhum treinamento especial para seu uso.	Facilidade de Aprendizado	Sistema	Média
RNF-5	O sistema deve garantir a consistência de dados por meio da obrigatoriedade no momento da inserção de certos dados.	Confiabilidade	Sistema	Alta
RNF-6	O desenvolvimento do sistema deve facilitar manutenções futuras, utilizando padrões de design de software.	Redigibilidade	Sistema	Média

5 Identificação de Subsistemas

Para atender aos requisitos elencados na Seção 4 e a fim de facilitar o gerenciamento do projeto, o sistema foi dividido em 3 subsistemas: Departamento, Graduação e PPG. O pacote core é um módulo do Marvin que possui a classe Academic, necessária para o módulo de controle de acessos. Neste caso é mostrado a dependência dos outros subsistemas pelo core. A classe CentroEnsino está diretamente ligada aos subsistemas, porem não está ligada ao core.

A Figura 1 ilustra os subsistemas e suas interdependências, enquanto a Tabela 3 apresenta uma breve descrição de cada subsistema identificado.

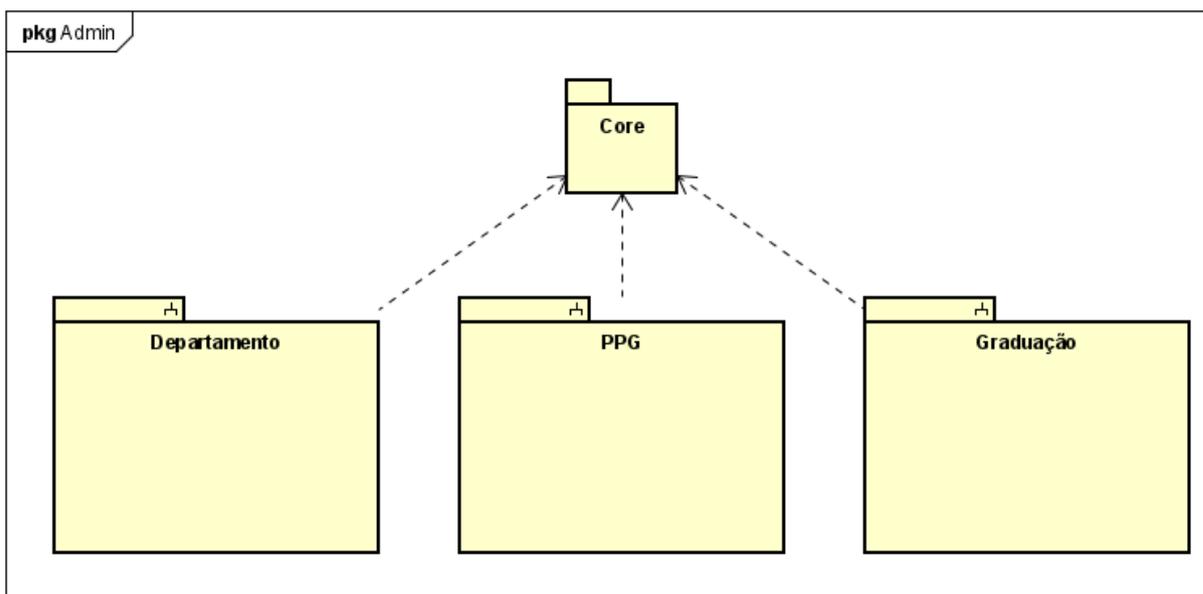


Figura 1 – Diagrama de Pacotes e os Subsistemas Identificados.

Tabela 3 – Subsistemas identificados e suas interdependências.

Subsistema	Descrição
Departamento	Subsistema responsável pelas funcionalidades relativas aos departamentos.
Graduação	Subsistema responsável pelas funcionalidades relativas aos cursos de graduação.
PPG	Subsistema responsável pelas funcionalidades relativas aos programas de pós-graduação.

6 Modelo de Casos de Uso

O modelo de casos de uso corresponde a uma tentativa de descrever a relação das funcionalidades do sistema com cada um de seus atores. Os atores identificados no contexto deste projeto estão descritos na Tabela 4.

Tabela 4 – Descrição dos atores envolvidos nos casos de uso.

Ator	Descrição
Administrador do Sistema	Pessoa responsável pelo cadastro inicial dos usuários dentro do sistema.
Chefe de Departamento	Professor responsável por coordenar e supervisionar todas as atividades da competência do departamento. Deve conceder acesso aos outros usuários do departamento.
Subchefe de Departamento	Professor responsável por coordenar e supervisionar todas as atividades da competência do departamento, na ausência do Chefe de Departamento. Deve conceder acesso aos outros usuários do departamento.
Secretário(a) do Departamento	Encarregado da execução de todos os serviços administrativos do departamento, incluindo concessão de acesso aos usuários.
Coordenador(a) de Curso de Graduação	Professor responsável por representar oficialmente o colegiado do curso de graduação. Deve poder conceder acesso aos usuários relacionados ao curso de graduação.
Subcoordenador(a) de Curso de Graduação	Professor responsável por representar oficialmente o colegiado do curso de graduação na ausência do coordenador. Deve poder conceder acesso aos usuários relacionados ao curso de graduação.
Secretário(a) do Colegiado da Graduação	Encarregado da execução de todos os serviços administrativos do curso de graduação, incluindo concessão de acesso aos usuários.
Coordenador(a) PPG	Professor responsável por representar oficialmente o colegiado de um programa de pós-graduação (PPG). Deve poder conceder acesso aos acadêmicos participantes de um PPG.
Subcoordenador(a) PPG	Professor responsável por representar oficialmente o colegiado PPG na ausência do coordenador. Deve poder conceder acesso aos acadêmicos participantes de um PPG.
Secretário(a) PPG	Encarregado da execução de todos os serviços administrativos do PPG, incluindo concessão de acesso aos acadêmicos.

A seguir, são apresentados os diagramas de casos de uso e descrições associadas, organizados por subsistema. Foi decidido que todos os casos de uso, incluindo as seções 6.1, 6.2 e 6.3 servirão apenas para dar uma visão geral, em formato diagramático, do módulo, julgando-se que o que foi documentado nas histórias de usuário é o suficiente.

A Figura 2 apresenta o diagrama de casos do administrador do sistema.

A seguir, são apresentadas as descrições de cada um dos casos de uso identificados. Os casos de uso cadastrais de baixa complexidade, envolvendo inclusão, consulta, alteração e exclusão (CRUDs), são descritos na Tabela 5.

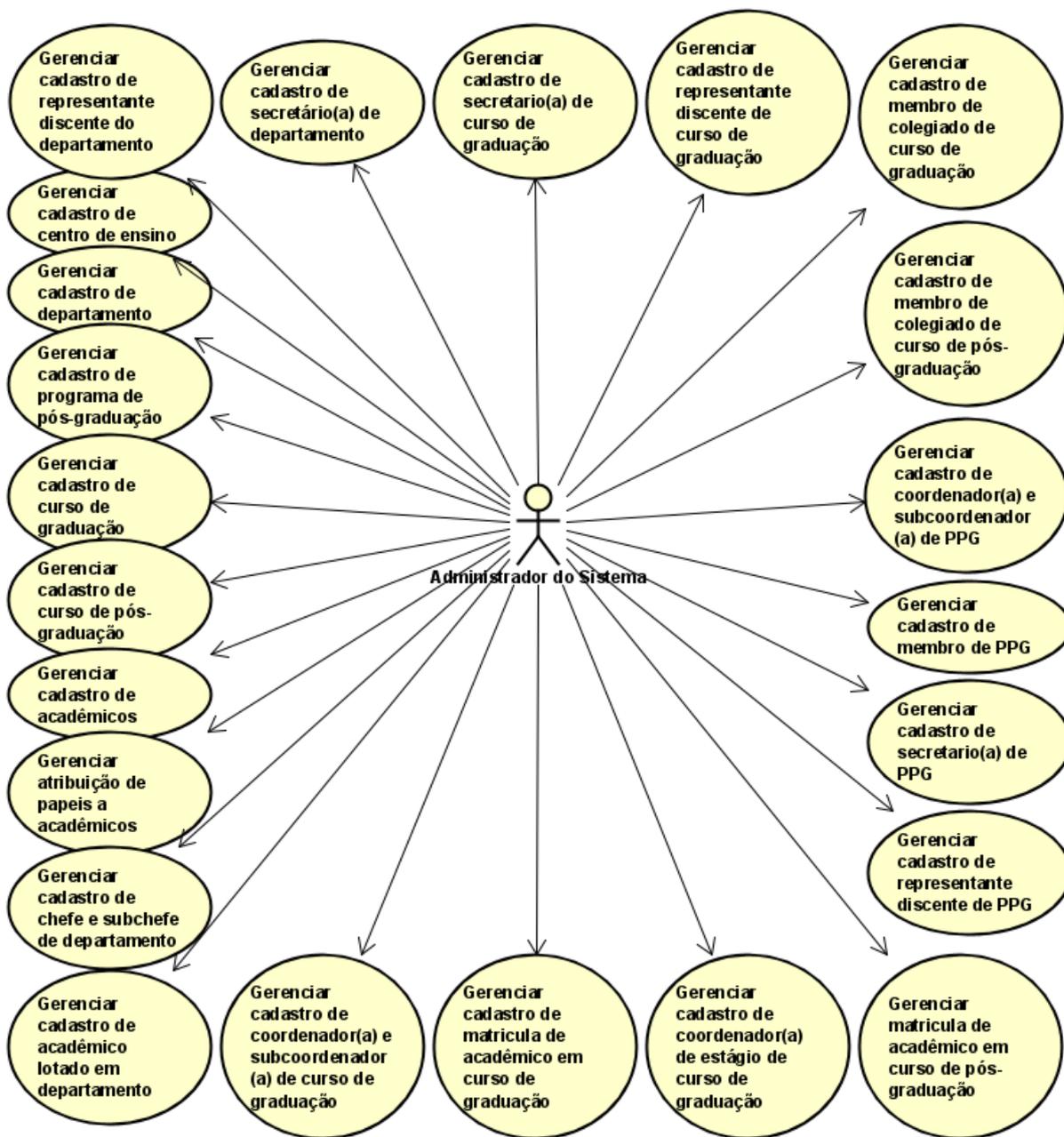


Figura 2 – Diagrama de Casos de Uso do administrador do sistema.

Tabela 5 – Caso de uso cadastrais do administrador do sistema.

Id	Nome	Requisitos
UC-1	Gerenciar cadastro de centro de ensino	US-1
UC-2	Gerenciar cadastro de departamento	US-2
UC-3	Gerenciar cadastro de programa de pós-graduação	US-3
UC-4	Gerenciar cadastro de curso de graduação	US-4
UC-5	Gerenciar cadastro de curso de pós-graduação	US-5
UC-6	Gerenciar cadastro de acadêmicos	US-6
UC-7	Gerenciar atribuição de papéis a acadêmicos	US-7
UC-8	Gerenciar cadastro de chefe ou subchefe do departamento	US-8
UC-9	Gerenciar cadastro de Secretário(a) do Departamento	US-9

Id	Nome	Requisitos
UC-10	Gerenciar cadastro de representante discente do departamento	US-10
UC-11	Gerenciar cadastro de acadêmico lotado em departamento	US-11
UC-12	Gerenciar cadastro de coordenador(a) ou subcoordenador(a) de curso de graduação	US-12
UC-13	Gerenciar cadastro de secretário(a) de curso de graduação	US-13
UC-14	Gerenciar cadastro de representante discente de curso de graduação	US-14
UC-15	Gerenciar cadastro de coordenador(a) de estágio de curso de graduação	US-15
UC-16	Gerenciar matrícula de acadêmico em curso de graduação	US-16
UC-17	Gerenciar cadastro de membro de colegiado de curso de graduação	US-17
UC-18	Gerenciar cadastro de coordenador(a) e subcoordenador(a) de PPG	US-18
UC-19	Gerenciar cadastro de secretário(a) de PPG	US-19
UC-20	Gerenciar cadastro de representante discente de PPG	US-20
UC-21	Gerenciar matrícula de acadêmico em curso de pós-graduação	US-21
UC-22	Gerenciar cadastro de membro de colegiado de curso de pós-graduação	US-22
UC-23	Gerenciar cadastro de membro de PPG	US-23

6.1 Subsistema Departamento

A Figura 3 apresenta o diagrama de casos de uso do subsistema Departamento.

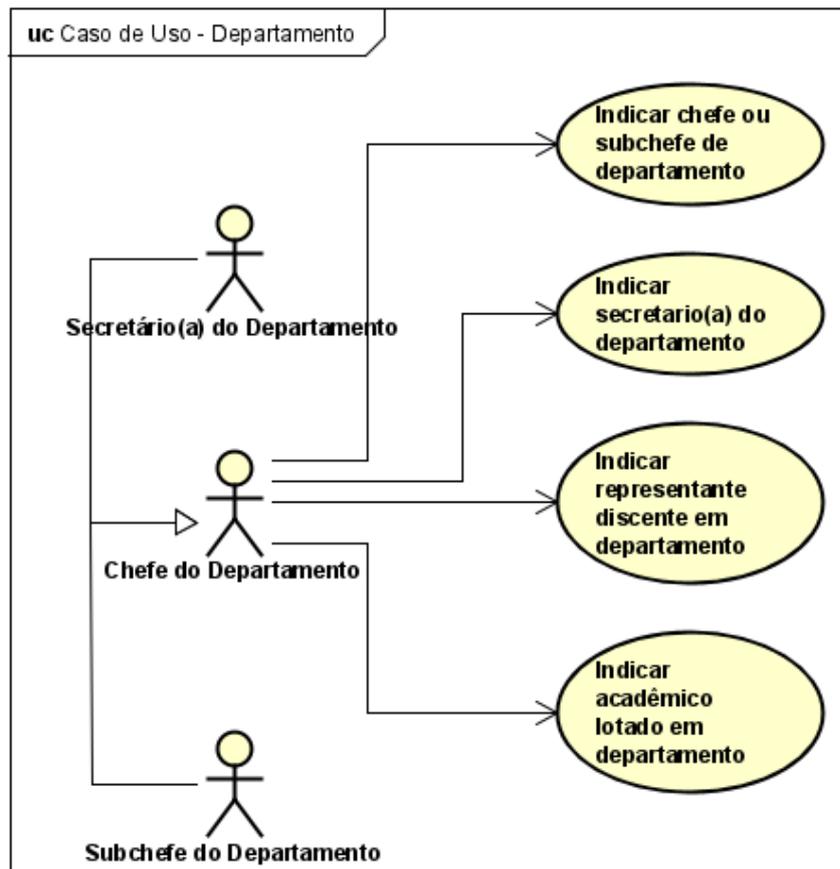


Figura 3 – Diagrama de Casos de Uso do subsistema Departamento.

A seguir, são apresentadas as descrições dos casos de uso identificados, na Tabela 6.

Tabela 6 – Caso de uso do subsistema Departamento.

Id	Nome	Requisitos
UC-24	Indicar Chefe ou Subchefe do Departamento	US-24
UC-25	Indicar Secretário do Departamento	US-25
UC-26	Indicar Representante Discente em Departamento	US-26
UC-27	Indicar Acadêmico Lotado em Departamento	US-27

6.2 Subsistema Graduação

A Figura 4 apresenta o diagrama de casos de uso do subsistema Graduação.

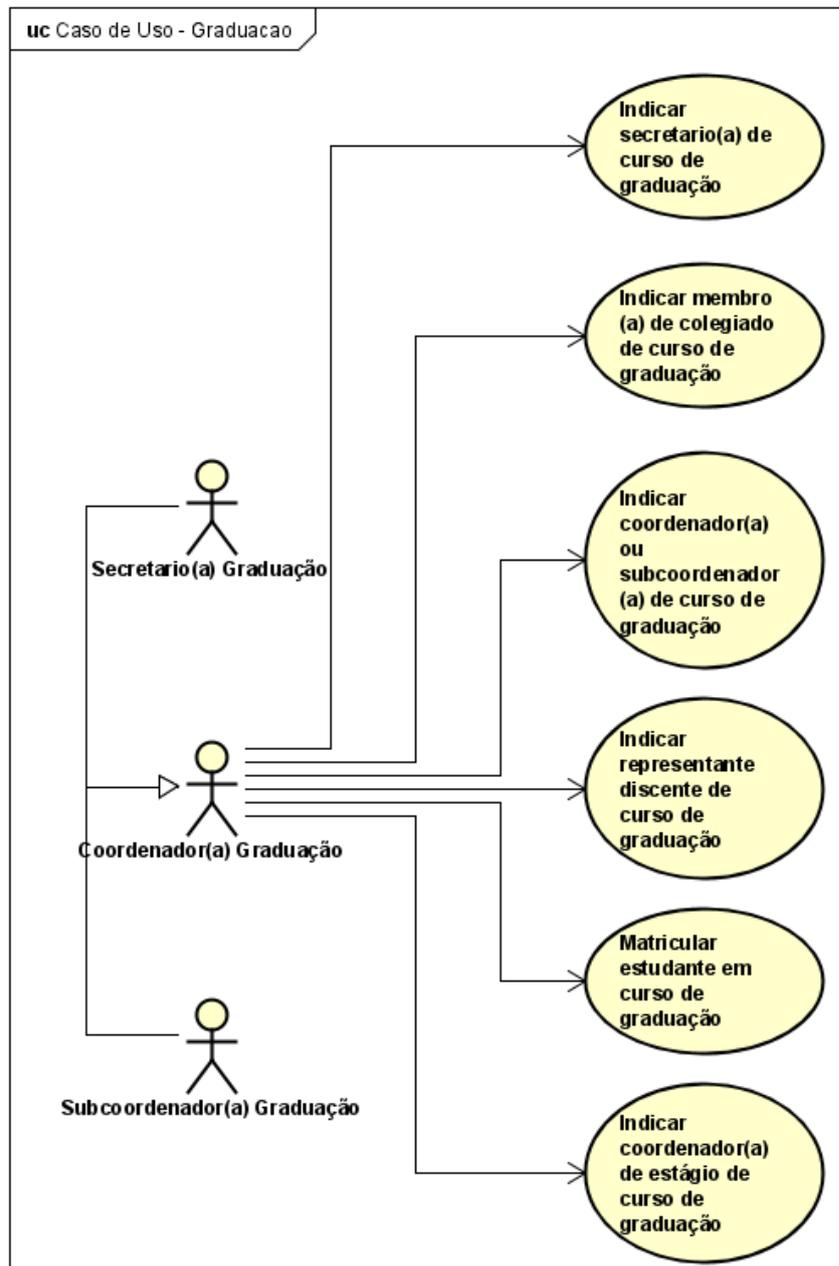


Figura 4 – Diagrama de Casos de Uso do subsistema Graduação.

A seguir, são apresentadas as descrições dos casos de uso identificados, na Tabela 7. Foi decidido que os casos de uso da Seção 6.2 servirão apenas para dar uma visão geral, em formato diagramático, do módulo, julgando-se que o que foi documentado nas estórias de usuário é o suficiente.

Tabela 7 – Caso de uso do subsistema Graduação.

Id	Nome	Requisitos
UC-28	Indicar coordenador(a) ou subcoordenador(a) de curso de graduação	US-28
UC-29	Indicar secretario(a) de curso de graduação	US-29
UC-30	Indicar representante discente de curso de graduação	US-30
UC-31	Indicar coordenador(a) de estágio de curso de graduação	US-31
UC-32	Matricular estudante em curso de graduação	US-32
UC-33	Indicar membro(a) de colegiado de curso de graduação	US-33

6.3 Subsistema PPG

A Figura 5 apresenta o diagrama de casos de uso do subsistema PPG.

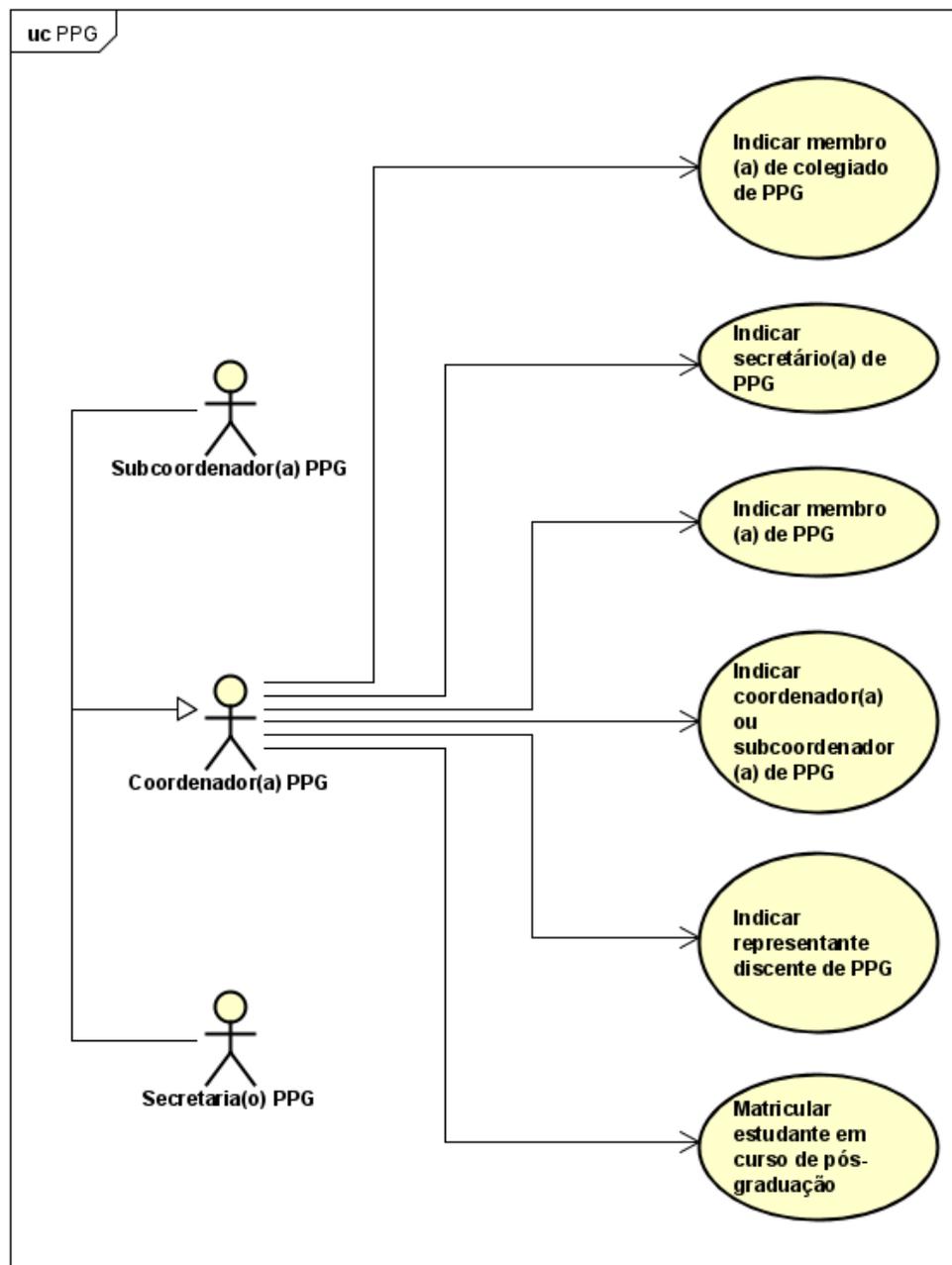


Figura 5 – Diagrama de Casos de Uso do subsistema PPG.

A seguir, são apresentadas as descrições dos casos de uso identificados, na Tabela 8. Foi decidido que os casos de uso da Seção 6.3 servirão apenas para dar uma visão geral, em formato diagramático, do módulo, julgando-se que o que foi documentado nas estórias de usuário é o suficiente.

Tabela 8 – Caso de uso cadastrais do subsistema PPG.

Id	Nome	Requisitos
UC-34	Indicar coordenador(a) ou subcoordenador(a) de PPG	US-34
UC-35	Indicar secretário(a) de PPG	US-35
UC-36	Indicar representante discente em PPG	US-36
UC-37	Indicar membro(a) de colegiado de PPG	US-37
UC-38	Matricular estudante em curso de pós-graduação	US-38
UC-39	Indicar membro(a) de PPG	US-39

7 Modelo Estrutural

O modelo conceitual estrutural visa capturar e descrever as informações (classes, associações e atributos) que o sistema deve representar para prover as funcionalidades descritas nos casos de uso especificados na Seção 6.

A seguir, é apresentado o diagrama de classe dos subsistemas identificados no contexto deste projeto. Na Seção 8 — Dicionário de Projeto — são apresentadas as descrições das classes, atributos e operações presentes no diagrama apresentado nesta seção.

7.1 Subsistema Departamento

A Figura 6 apresenta o diagrama de classes do subsistema Departamento.

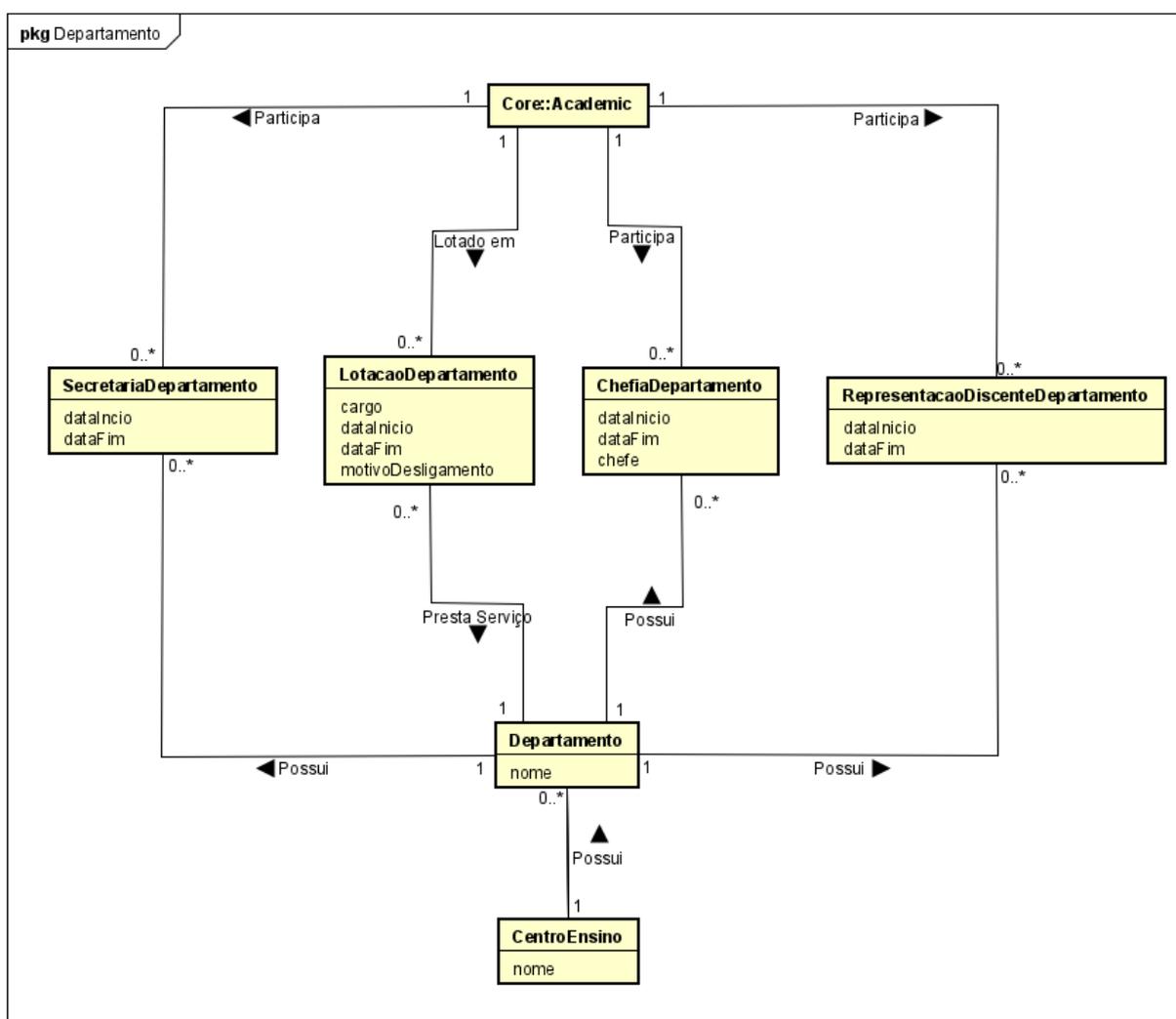


Figura 6 – Diagrama de classes do subsistema Departamento.

7.2 Subsistema Graduação

A Figura 7 apresenta o diagrama de classes do subsistema Graduação.

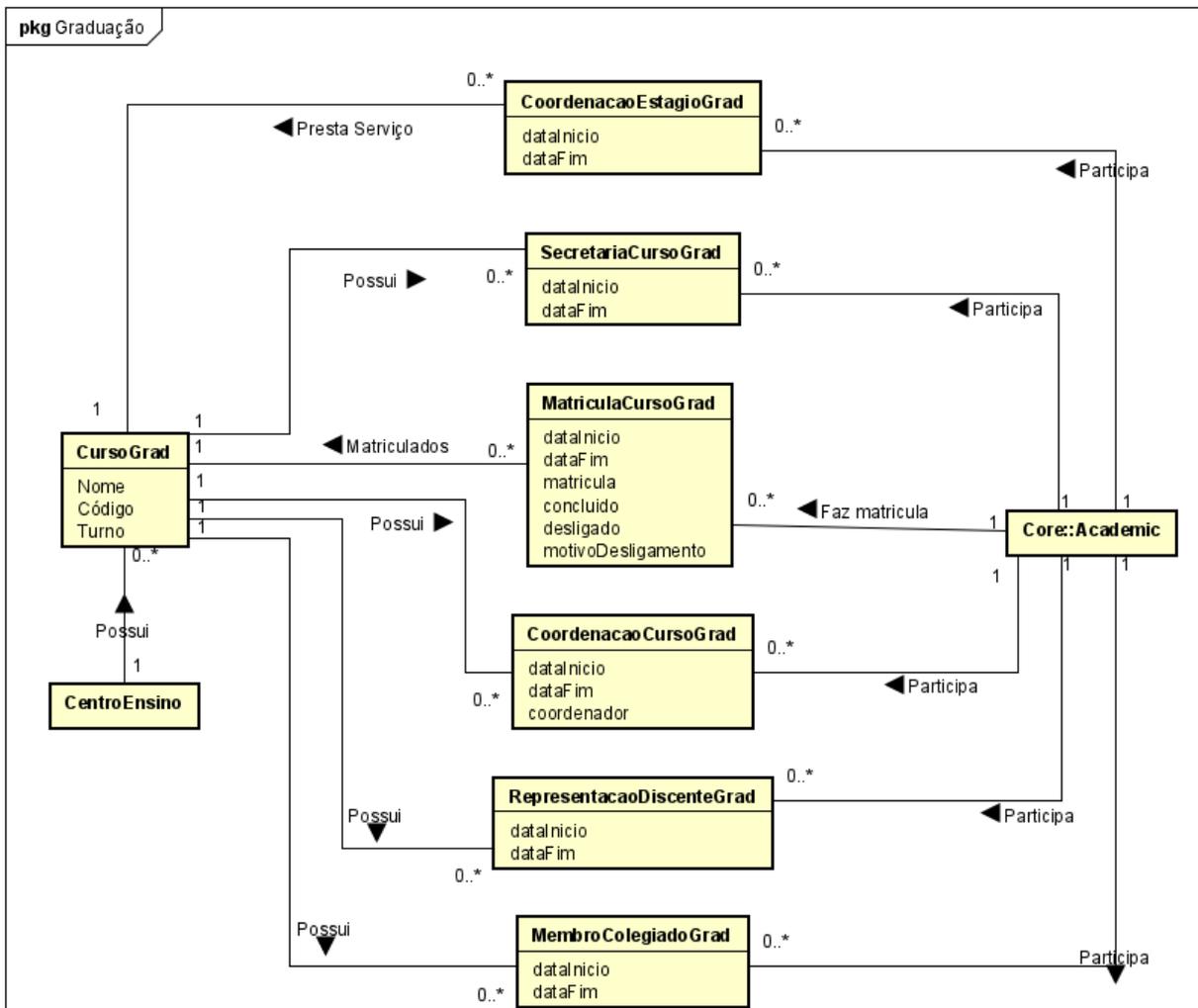


Figura 7 – Diagrama de classes do subsistema Graduação.

7.3 Subsistema PPG

A Figura 8 apresenta o diagrama de classes do subsistema PPG.

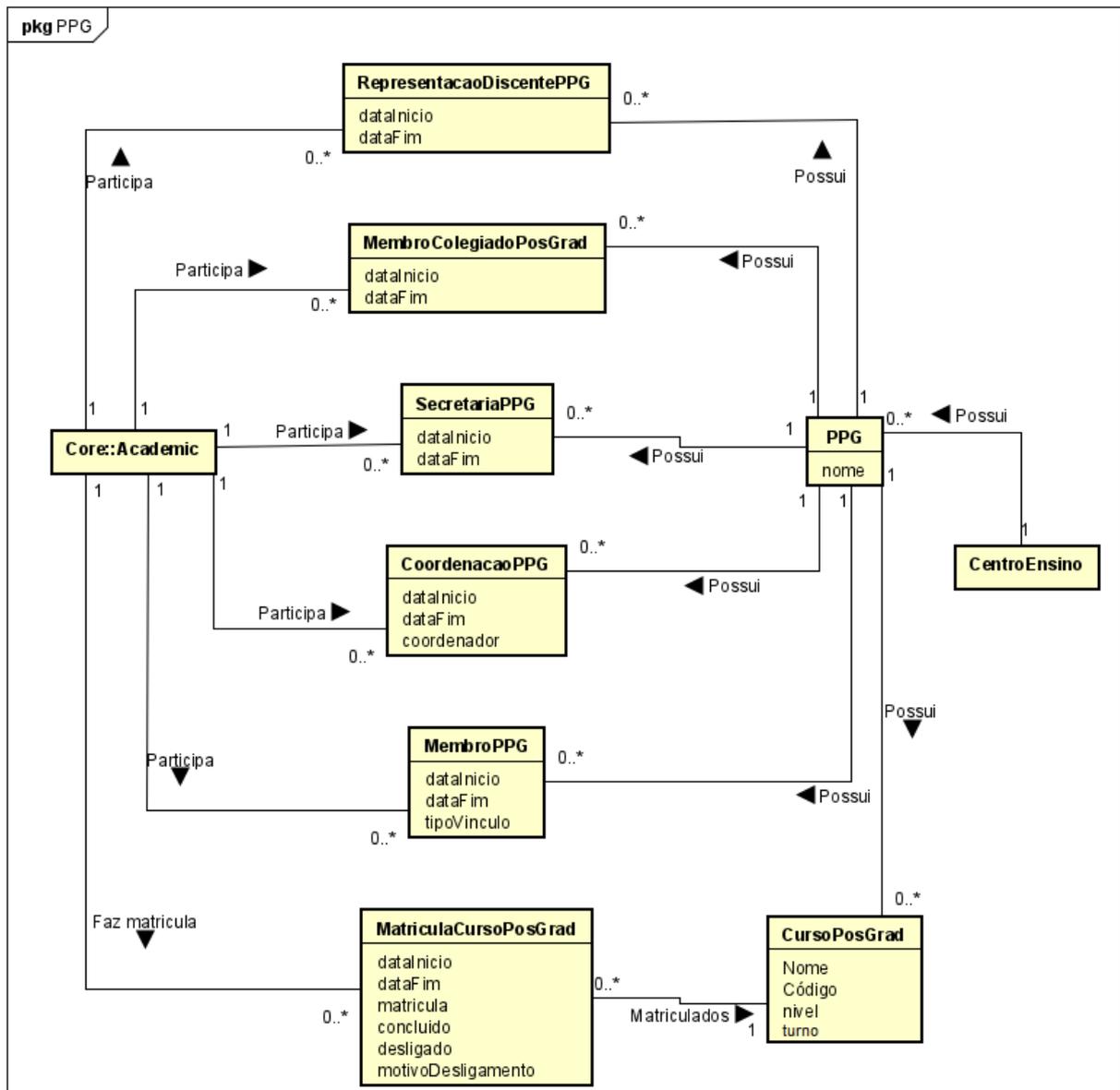


Figura 8 – Diagrama de classes do subsistema PPG.

8 Dicionário de Projeto

Esta seção apresenta as definições detalhadas das classes, descrevendo seus atributos e associações e servindo como um glossário do projeto. As definições são organizadas por subsistema, cada classe sendo apresentada em uma tabela separada. A coluna “Obr.?” indica com um “x” se o atributo é obrigatório (deve possuir um valor para se criar um objeto da classe).

Vale destacar que eventuais operações que estas classes vierem a ter não são listadas e descritas nesta fase do projeto. Além disso, na Seção 7, algumas classes podem ser incluídas nos diagramas de outros subsistemas para ilustrar a relação entre eles. No dicionário de projeto, no entanto, classes são descritas apenas em seus subsistemas de origem.

Tabela 9 – Detalhamento da classe *Academic*.

Propriedade	Tipo	Obr.?	Descrição
nome	String	x	Nome do acadêmico.
email	String	x	Email do acadêmico.
cpf	int	x	CPF do acadêmico.
dataNascimento	Date	x	Data de nascimento do acadêmico.
idLattes	int	x	Id Lattes do acadêmico.
password	String	x	Senha que o acadêmico irá usar para acessar o sistema.

Tabela 10 – Detalhamento da classe *Role*.

Propriedade	Tipo	Obr.?	Descrição
papel	String	x	Papel que o acadêmico terá no sistema.

Tabela 11 – Detalhamento da classe *CentroEnsino*.

Propriedade	Tipo	Obr.?	Descrição
nome	String	x	Nome do Centro de Ensino.

Tabela 12 – Detalhamento da classe *Departamento*.

Propriedade	Tipo	Obr.?	Descrição
nome	String	x	Nome do Departamento.

Tabela 13 – Detalhamento da classe *ChefiaDepartamento*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
chefe	Boolean	x	Se o acadêmico é chefe do departamento ou não

Tabela 14 – Detalhamento da classe *SecretariaDepartamento*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.

Tabela 15 – Detalhamento da classe *RepresentacaoDiscenteDepartamento*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.

Tabela 16 – Detalhamento da classe *LotacaoDepartamento*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
cargo	String	x	Cargo do acadêmico dentro do departamento.
motivoDesligamento	String	x	Motivo do desligamento do acadêmico do departamento.

Tabela 17 – Detalhamento da classe *CursoGrad*.

Propriedade	Tipo	Obr.?	Descrição
nome	String	x	Nome do curso de graduação.
codigo	int	x	Código do curso de graduação.
turno	String	x	Turno do curso de graduação.

Tabela 18 – Detalhamento da classe *CoordenacaoCursoGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
coordenador	Boolean	x	Se o acadêmico é coordenador dentro do curso de graduação.

Tabela 19 – Detalhamento da classe *CoordenacaoEstagioGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.

Tabela 20 – Detalhamento da classe *SecretariaCursoGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.

Tabela 21 – Detalhamento da classe *RepresentacaoDiscenteGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.

Tabela 22 – Detalhamento da classe *MembroColegiadoGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
matricula	Date	x	Matricula do acadêmico.

Tabela 23 – Detalhamento da classe *MatriculaCursoGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
matricula	Date	x	Matricula do acadêmico.
concluido	Boolean	x	Se o acadêmico concluiu o curso ou não.
desligado	Boolean	x	Se o acadêmico foi desligado do curso ou não.
motivoDesligamento	String		Motivo do desligamento do acadêmico, caso ele tenha sido desligado do curso

Tabela 24 – Detalhamento da classe *PPG*.

Propriedade	Tipo	Obr.?	Descrição
nome	String	x	Nome do PPG.

Tabela 25 – Detalhamento da classe *CursoPosGrad*.

Propriedade	Tipo	Obr.?	Descrição
nome	String	x	Nome do curso de pós-graduação.
codigo	int	x	Código do curso de pós-graduação.
nível	int	x	Nível do curso de pós-graduação (pós, mestrado, doutorado).
turno	String	x	Turno do curso de pós-graduação.

Tabela 26 – Detalhamento da classe *CoordenacaoPPG*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
coordenador	Boolean	x	Se o acadêmico é coordenador dentro do PPG.

Tabela 27 – Detalhamento da classe *SecretariaPPG*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.

Tabela 28 – Detalhamento da classe *RepresentacaoDiscentePPG*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.

Tabela 29 – Detalhamento da classe *MatriculaCursoPosGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
matricula	Date	x	Matricula do acadêmico.
concluido	Boolean	x	Se o acadêmico concluiu o curso ou não.
desligado	Boolean	x	Se o acadêmico foi desligado do curso ou não.
motivoDesligamento	String		Motivo do desligamento do acadêmico, caso ele tenha sido desligado do curso

Tabela 30 – Detalhamento da classe *MembroPPG*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.
tipoVinculo	String	x	Vinculo do acadêmico dentro do PPG .

Tabela 31 – Detalhamento da classe *MembroColegiadoPosGrad*.

Propriedade	Tipo	Obr.?	Descrição
dataInicio	Date	x	Data de início de exercício de função.
dataFim	Date	x	Data final de exercício de função.



Documento de Projeto de Sistema

Macs - Sistema de controle de acessos do Marvin

Vitória, ES

2022

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Vitor Araujo Chane	02/03/2022	Versão inicial adicionando capítulos 1, 2 e 3.
1.1	Vitor Araujo Chane	03/03/2022	Ajuste de modelos.
1.2	Vitor Araujo Chane	07/03/2022	Adicionando capítulo 4.
1.2	Vitor Estevão Silva Souza	10/03/2022	Revisão dos capítulos 1-4.
1.3	Vitor Araujo Chane	14/03/2022	Ajuste nos capítulos 1-4.
1.3	Vitor Estevão Silva Souza	14/03/2022	Revisão dos capítulos 1-4.
1.4	Vitor Araujo Chane	16/03/2022	Adicionando modelo de navegação.
1.4	Vitor Estevão Silva Souza	14/03/2022	Revisão do modelo de navegação
1.5	Vitor Araujo Chane	16/03/2022	Adicionando modelos de entidade e persistencia no capítulo 5.
1.5	Vitor Estevão Silva Souza	17/03/2022	Revisão da seção 5.
1.6	Vitor Araujo Chane	18/03/2022	Adicionando novos modelos no capítulo 5.
1.6	Vitor Estevão Silva Souza	18/03/2022	Revisão das seções 4 e 5.
1.7	Vitor Araujo Chane	19/03/2022	Ajuste dos modelos frameweb para o capítulo 5.
1.8	Vitor Araujo Chane	25/07/2022	Ajuste dos modelos frameweb para o capítulo 5.
1.9	Vitor Araujo Chane	07/08/2022	Revisão do Documento de Projeto, ajuste dos modelos de navegação e persistência relativos ao capítulo 5 e finalização do Documento de Projeto.

1 Introdução

Este documento apresenta o projeto (*design*) do módulo *Macs - Sistema de controle de acessos do Marvin*. O módulo Macs tem como objetivo permitir que usuários com diferentes papéis dentro do sistema Marvin, tenham acesso a funcionalidades específicas relativas ao seu papel. Este projeto foi construído com base nos requisitos deste módulo, especificados em seu Documento de Requisitos de Sistema.

Além desta introdução, este documento está organizado da seguinte forma: A Seção 2 apresenta a plataforma de software utilizada na implementação do sistema. A Seção 3 apresenta a especificação dos requisitos não funcionais (atributos de qualidade), definindo as táticas e o tratamento a serem dados aos atributos de qualidade considerados condutores da arquitetura. A Seção 4 apresenta a arquitetura de software. Por fim, a Seção 5 apresenta os modelos FrameWeb que descrevem os componentes da arquitetura.

2 Plataforma de Desenvolvimento

Na Tabela 1 são listadas as tecnologias utilizadas no desenvolvimento da ferramenta, bem como o propósito de sua utilização.

Tabela 1 – Plataforma de Desenvolvimento e Tecnologias Utilizadas.

Tecnologia	Versão	Descrição	Propósito
Java EE	8.0	Conjunto de especificação de APIs e tecnologias, que são implementadas por programas servidores de aplicação.	Redução da complexidade do desenvolvimento, implantação e gerenciamento de aplicações Web a partir de seus componentes de infra-estrutura prontos para o uso.
Java	13	Linguagem de programação orientada a objetos e independente de plataforma.	Escrita do código-fonte das classes que compõem o sistema.
EJB	4.0	API para construção de componentes transacionais gerenciados por container.	Implementação das regras de negócio em componentes distribuídos, transacionais, seguros e portáteis.
JSF	2.3	API para a construção de interfaces de usuários baseada em componentes para aplicações Web	Criação das páginas Web e sua comunicação com as classes Java.
JPA	2.2	API para persistência de dados por meio de mapeamento objeto/-relacional.	Persistência dos objetos de domínio sem necessidade de escrita dos comandos SQL.
CDI	2.0	API para injeção de dependências.	Integração das diferentes camadas da arquitetura.
Facelets	2.0	API para definição de decoradores (<i>templates</i>) integrada ao JSF.	Reutilização da estrutura visual comum às páginas, facilitando a manutenção do padrão visual do sistema.
PrimeFaces	8.0	Conjunto de componentes visuais JSF <i>open source</i> .	Reutilização de componentes visuais Web de alto nível.
AdminFaces	1.3	<i>Template</i> visual completo integrado ao Facelets/JSF e ao PrimeFaces.	Fornecimento do padrão visual do sistema.
JButler	2.0	<i>Framework</i> para facilitar o desenvolvimento de aplicações Java EE.	Fornecimento de superclasses prontas para entidades (classes de domínio) persistentes, DAOs e controladores/serviços básicos de cadastro (CRUD).
MySQL Server	8.0	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pelo sistema.
WildFly	19	Servidor de Aplicações para Java EE.	Fornecimento de implementação das APIs citadas acima e hospedagem da aplicação Web, dando acesso aos usuários via HTTP.

Na Tabela 2 vemos os softwares que apoiaram o desenvolvimento de documentos e também do código fonte.

Tabela 2 – Softwares de Apoio ao Desenvolvimento do Projeto

Tecnologia	Versão	Descrição	Propósito
Eclipse Java EE IDE for Web Developers	4.22.0	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento Java EE.	Implementação, implantação e testes da aplicação Web Java EE.
Apache Maven	3.8.0	Ferramenta de gerência/construção de projetos de software.	Obtenção e integração das dependências do projeto.
Git	2.35.1	Sistema de controle de versões distribuído	Registrar o histórico de edições de qualquer tipo de arquivo.
Bitbucket	2022	Plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git	Permitir que os membros do projeto contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.
Slack	4.23.0	Plataforma de colaboração entre membros de uma equipe	Colaboração entre os membros do projeto Marvin.
Trello	2022	Aplicativo web de gerenciamento de projeto que utiliza do paradigma kanban	Gerenciamento de fluxo de trabalho, maximizando a eficiência.
Astah UML	8.4.1	Ferramenta de modelagem UML	Criação de diagramas e modelos na fase de modelagem do sistema.
Overleaf	2022	Editor de LaTeX baseado em Cloud.	Escrita e edição da documentação do sistema.

3 Requisitos Não Funcionais

A Tabela 3 apresenta a especificação dos requisitos não funcionais identificados no Documento de Especificação de Requisitos.

Tabela 3 – Especificação de Requisitos Não Funcionais.

RNF-1 – A plataforma deve estar disponível como uma aplicação Web, acessível a partir dos principais navegadores disponíveis no mercado.	
Categoria:	Portabilidade.
Tática / Tratamento:	Utilizar estilizadores que são suportados por tais navegadores; utilizar módulos e bibliotecas capazes de prover as devidas funcionalidades, independente da forma de renderização do navegador.
Medida:	Ser possível o acesso nos navegadores mais conhecidos do mercado (Google Chrome, Mozilla Firefox e Microsoft Edge) sendo que não haja diferenças de leiaute e funcionalidades que impossibilitem o funcionamento do sistema.
Critério de Aceitação:	Nenhuma diferença significativa no leiaute ou nas funcionalidades do sistema que impeça o uso por parte dos usuários.

RNF-2 – O sistema deve controlar o acesso às suas funcionalidades por meio de autenticação e autorização, utilizando o papel de cada usuário para definir a que funcionalidades ele terá acesso.	
Categoria:	Segurança
Tática / Tratamento:	Identificar usuários por meio de nome e senha criptografada; identificar usuários logados por papéis.
Medida:	Testes a fim de verificar se o acesso está correto. Caso possível, ao final do desenvolvimento do módulo, reunir membros da comunidade acadêmica com experiência em segurança para a realização de testes de esforço e verificar se conseguem burlar o acesso ao sistema. Observar os usuários finais utilizando o sistema e verificar se o acesso por login e senha não apresenta erros e se eles possuem acesso as funcionalidades relativas ao perfil.
Critério de Aceitação:	Não ser possível logar sem que o login e senha informada esteja compatível com o que se encontra no banco de dados; usuário logado conseguir acessar funcionalidades respectivas a seu papel.

RNF-3 – É de extrema importância que informações sigilosas como senhas e afins sejam criptografadas.	
Categoria:	Confiabilidade
Tática / Tratamento:	Geração de senha criptografada no banco de dados; geração de outras informações sigilosas criptografadas.
Medida:	Validação e verificação no banco de dados. Caso possível, ao final do desenvolvimento do módulo, reunir acadêmicos com experiência na área de segurança digital, para execução de testes de esforço com o objetivo de burlar o acesso ao sistema.
Critério de Aceitação:	Nenhuma informação sigilosa ou acesso obtidos indevidamente nos testes.

RNF-4 – O sistema deve ser de fácil aprendizado, não sendo necessário nenhum treinamento especial para seu uso.	
Categoria:	Usabilidade.
Tática / Tratamento:	Desenvolvimento do software se atentando às boas práticas de usabilidade e experiência do usuário.
Medida:	Verificar com usuário final se o ambiente está agradável e ele consegue utilizar sem auxílio de tutorial ou ajuda externa; Observar os usuários finais utilizando o sistema e verificar quanto tempo os mesmos demoram para realizar cada tarefa e a quantidade de erros cometidos ao realizá-las.
Critério de Aceitação:	Usuários conseguirem utilizar o sistema sem nenhum tipo de treinamento prévio, de forma com que o tempo para executar cada atividade seja próximo ao tempo de quem já sabe utilizar o sistema.

RNF-5 – O sistema deve garantir a consistência de dados por meio da obrigatoriedade no momento da inserção de certos dados.	
Categoria:	Confiabilidade.
Tática / Tratamento:	Validação no momento do preenchimento dos formulários; garantia de preenchimento em campos obrigatórios.
Medida:	Caso possível, ao final do desenvolvimento do módulo, reunir acadêmicos com experiência na área de segurança digital, para execução de testes de esforço com o objetivo de burlar os campos que deveriam ser preenchidos obrigatoriamente. Verificar se os usuários finais conseguem preencher campos obrigatórios sem erros. Checagem de validação de dados durante a vida do sistema e, caso seja encontrada alguma falha, deve-se realizar um levantamento com o propósito de diagnosticar a falha e corrigi-la.
Critério de Aceitação:	Usuários conseguirem preencherem os campos obrigatórios e submeter o formulário. Mensagem de erro sinalizando que um dado campo é requisitado aparecerá sempre que algum campo obrigatório não for preenchido. Caso algum campo obrigatório não for preenchido, a submissão do formulário estará bloqueada até que o campo obrigatório seja preenchido com parâmetros válidos.

RNF-6 – O desenvolvimento do sistema deve facilitar manutenções futuras, utilizando padrões de design de software e ferramentas de controle de versão.	
Categoria:	Manutenibilidade.
Tática / Tratamento:	Desenvolvimento em arquitetura limpa e escalável; boas práticas de desenvolvimento utilizando programação orientada a objetos; utilização de ferramentas que auxiliem no processo de desenvolvimento de software; desenvolvimento utilizando a ferramenta Git.
Medida:	Uso de ferramentas que monitoram a qualidade do software, indicando bugs antes de cada implantação da aplicação, durante todo o ciclo de vida do projeto. Utilização de ferramenta de controle de versão para medição de quantidade de <i>commits</i> , assim como acompanhamento dos desenvolvedores que estão no projeto.
Critério de Aceitação:	Código hospedado em diretório que utilize Git. Durante todo o ciclo de vida de desenvolvimento do sistema Marvin o código deve estar organizado e seguindo os padrões arquiteturais (modularização em pastas específicas) e os padrões de <i>clean code</i> (nomes de classes, nomes de arquivos e identificação do código seguindo os padrões do projeto) estando disponível para os professores e estudantes que continuarem a desenvolver o sistema. Sistema compilando normalmente a partir do ‘clone’ no diretório remoto, sem apresentação de bugs que impossibilitem a utilização na máquina do desenvolvedor.

4 Arquitetura de Software

Para o desenvolvimento do módulo *Macs - Sistema de controle de acessos do Marvin* foi utilizada uma arquitetura baseada no padrão arquitetônico Camada de Serviço (*Service layer*) (FOWLER, 2002) utilizando *frameworks*. A Figura 1 mostra a arquitetura do sistema e indica os componentes que estão presentes em cada pacote, bem como a atuação das tecnologias Jakarta EE envolvidas:

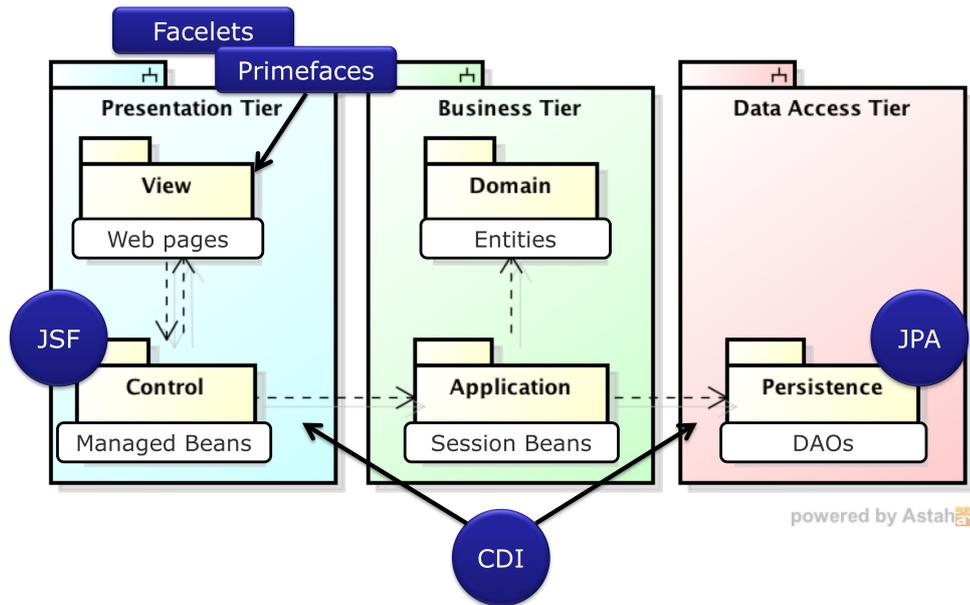


Figura 1 – Arquitetura do módulo *Macs - Sistema de controle de acessos do Marvin*.

O Marvin tem como base o pacote `br.ufes.informatica.marvin` sob o qual se encontram os outros subsistemas já implementados e integrados. O núcleo e os subsistemas são as partições da arquitetura, sendo que cada subsistema terá:

- **Camada de Apresentação** ou *Presentation Tier*: responsável pela interação com o usuário, é dividida em dois pacotes: **Visão** (*View*), responsável por conter os artefatos de interface com o usuário, como páginas, scripts e layouts; e **Controle** (*Control*), contendo as classes de controle que manipulam as solicitações feitas pelo pacote de Visão, e também responsável pela comunicação com o pacote de Aplicação (*Application*), pertencente à camada de negócio;
- **Camada de Negócio** ou *Business Tier*: responsável pela lógica de negócio, assim como a camada de apresentação, é dividida em dois pacotes: pacote de **Aplicação** (*Application*), que contém as classes responsáveis pelas funcionalidades do sistema; e o pacote de **Domínio** (*Domain*), que contém as classes que representam o domínio do problema, identificadas na fase de análise de requisitos;

- **Camada de Acesso de Dados** ou *Data Access Tier*: com apenas o pacote de **Persistência** (*Persistence*), é responsável pela persistência de dados, como por exemplo, compartilhamento de arquivos e acesso a bancos de dados.

Para o sistema (subsistema do Marvin) *Macs - Sistema de controle de acessos do Marvin* as classes e interfaces foram modularizadas em 3 subsistemas (Departamento, Graduação e PPG) como pode ser visto na Figura 2, e os pacotes foram agrupados a partir do pacote base do Marvin e pelas camadas da arquitetura, como é mostrado na Figura 3. A pasta visão (view) mostrada na camada de apresentação foi substituída por pastas que possuem o nome do subsistema, e dentro dela, uma pasta com o nome da entidade relacionada, seguido em seu interior de páginas Web definidas pelos arquivos `index.xhtml` e `form.xhtml`, como pode ser visto na Figura 4.

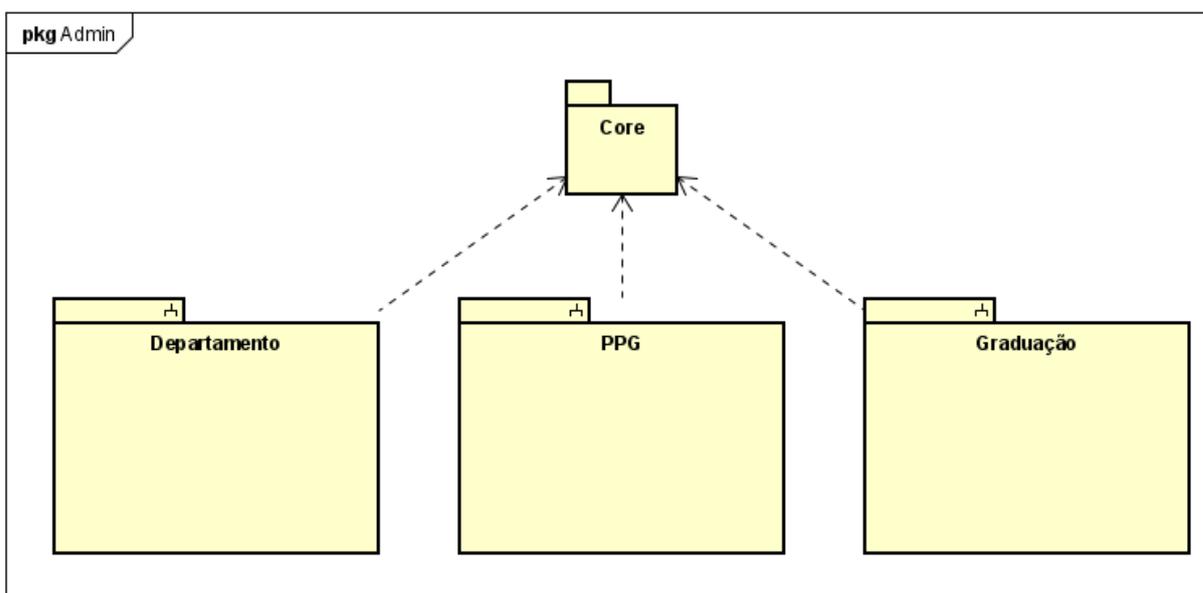
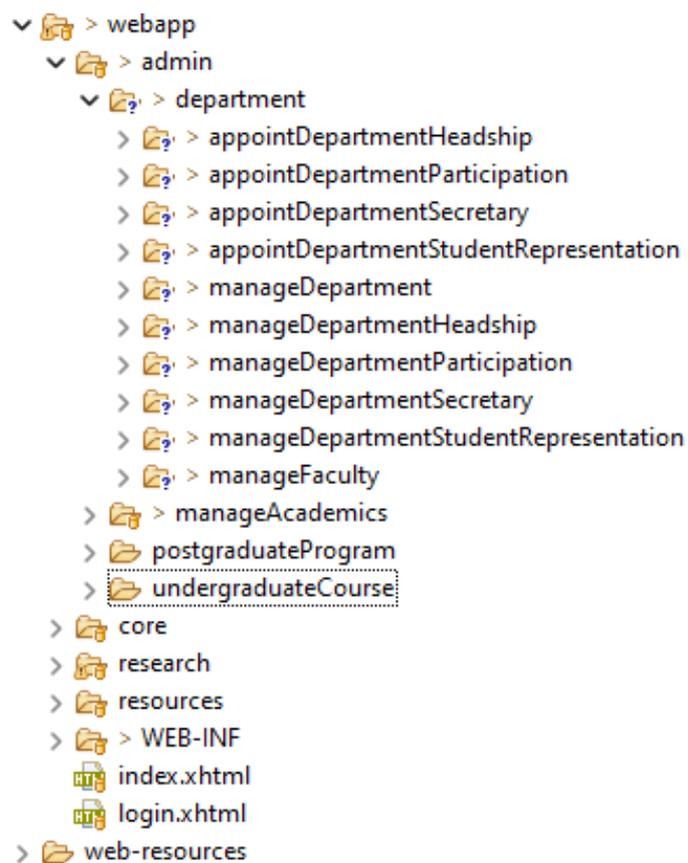


Figura 2 – Subsistemas do *Macs - Sistema de controle de acessos do Marvin*.

```
> <> JPA Content
  > <> > src/main/java
    > <> br.ufes.informatica.marvin.admin.controller
    > <> > br.ufes.informatica.marvin.admin.department.application
    > <> > br.ufes.informatica.marvin.admin.department.controller
    > <> > br.ufes.informatica.marvin.admin.department.domain
    > <> > br.ufes.informatica.marvin.admin.department.persistence
    > <> > br.ufes.informatica.marvin.admin.graduation.application
    > <> > br.ufes.informatica.marvin.admin.graduation.controller
    > <> > br.ufes.informatica.marvin.admin.graduation.domain
    > <> > br.ufes.informatica.marvin.admin.graduation.persistence
    > <> > br.ufes.informatica.marvin.admin.ppg.application
    > <> > br.ufes.informatica.marvin.admin.ppg.controller
    > <> > br.ufes.informatica.marvin.admin.ppg.domain
    > <> > br.ufes.informatica.marvin.admin.ppg.persistence
```

Figura 3 – Pacotes do *Macs - Sistema de controle de acessos do Marvin*

Figura 4 – Páginas web do *Macs - Sistema de controle de acessos do Marvin*

5 Modelagem FrameWeb

Macs - Sistema de controle de acessos do Marvin é um sistema Web cuja arquitetura utiliza *frameworks* comuns no desenvolvimento para esta plataforma. Desta forma, o sistema pode ser modelado utilizando a abordagem FrameWeb (SOUZA, 2020).

A Tabela 4 indica os *frameworks* presentes na arquitetura do sistema que se encaixam em cada uma das categorias de *frameworks* que FrameWeb dá suporte. Em seguida, os modelos FrameWeb são apresentados para cada camada da arquitetura.

Tabela 4 – *Frameworks* da arquitetura do sistema separados por categoria.

Categoria de <i>Framework</i>	<i>Framework</i> Utilizado
Controlador Frontal	JSF
Injeção de Dependências	CDI
Mapeamento Objeto/Relacional	JPA
Interface de Usuário	Primefaces

5.1 Camada de Negócio

Como mostrado na Seção 4, a camada de negócio (ou *Business Tier*) é responsável pela lógica de negócio e é dividada nos pacotes de aplicação e domínio. Em aplicação, temos as classes que implementam as funcionalidades do sistema, e no pacote domínio, temos as classes de domínio. Esta seção será dividida em três sub-seções, indicando os modelos de entidade e de aplicação dos 3 subsistemas do módulo *Macs - Sistema de controle de acessos do Marvin*.

Para o pacote domínio, vale dizer que todas as classes de domínio estendem *PersistentObjectSupport* do JButler, mas essa herança não é mostrada nos diagramas pois as várias associações iriam prejudicar a visualização do modelo. Os modelo de entidade dos subsistemas Departamento, Graduação e PPG são mostrados nas figuras 6, 8 e 10 respectivamente.

No pacote aplicação, todas as classes de aplicação relacionadas a cadastros estendem de *CrudServiceBean*, que é mostrado na Figura 5 de forma genérica. Assim como o pacote de domínio, a herança das outras classes de serviço foram ocultadas dos diagramas por conta da poluição visual que traria. Os modelo de aplicação dos subsistemas Departamento, Graduação e PPG são mostrados nas figuras 7, 9 e 11 respectivamente.

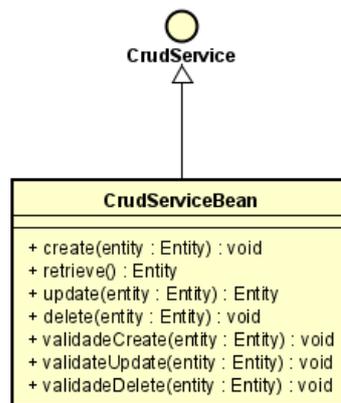


Figura 5 – Modelo de Aplicação genérico

5.1.1 Subsistema Departamento

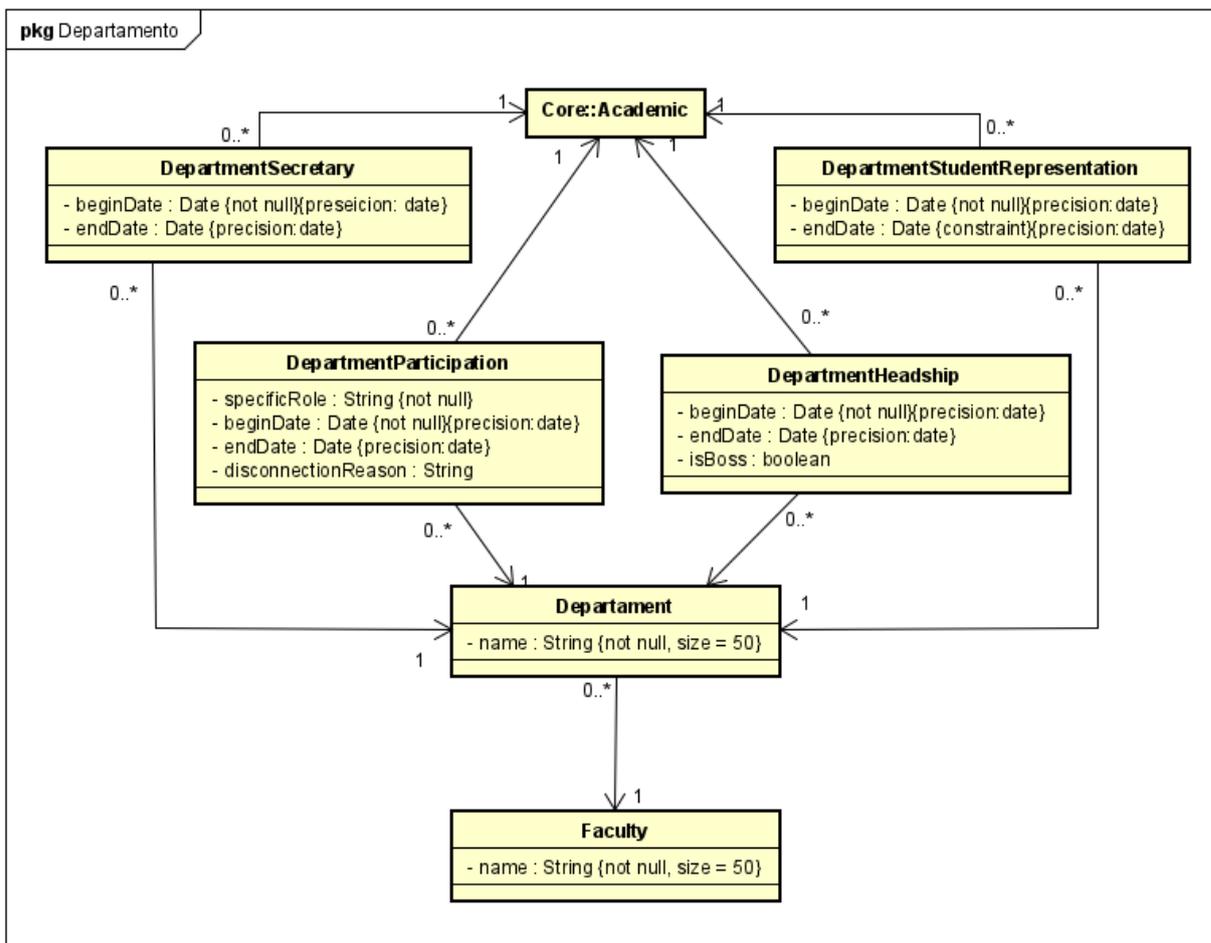


Figura 6 – Modelo de Entidades do subsistema Departamento

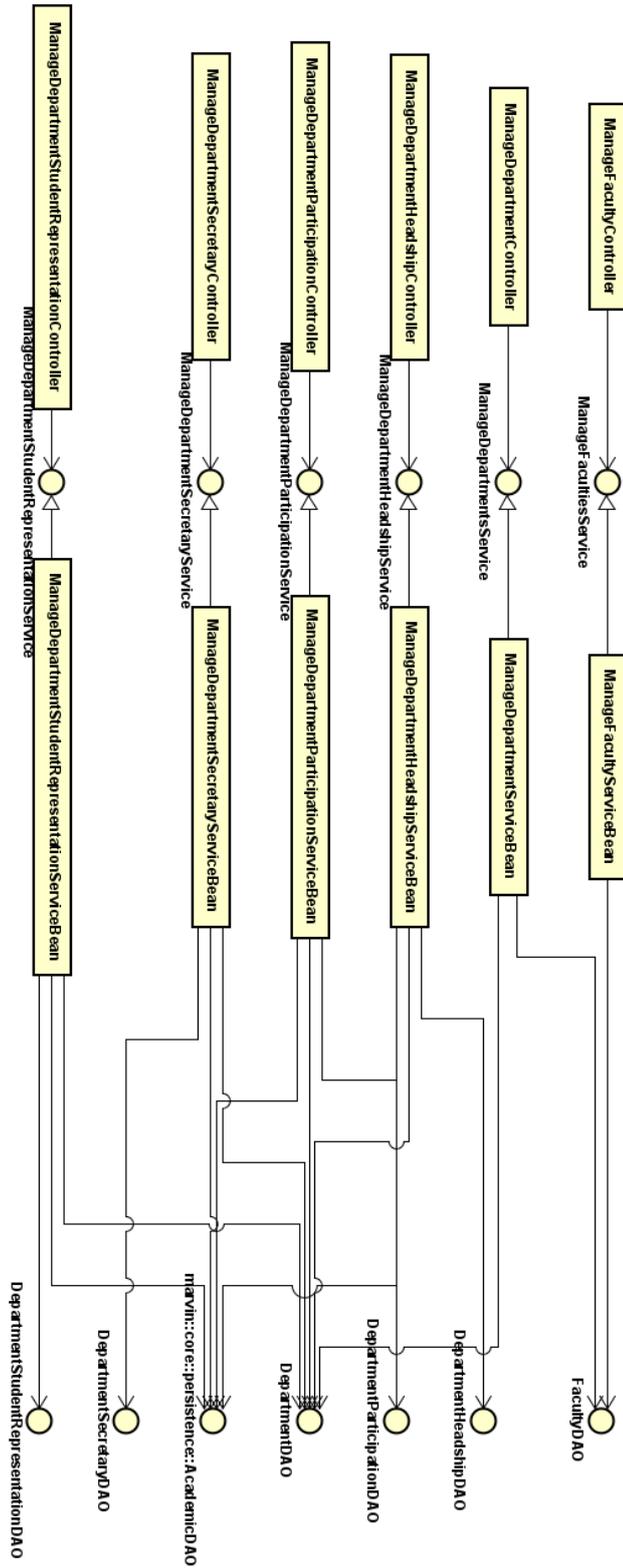


Figura 7 – Modelo de aplicação do subsistema Departamento

5.1.2 Subsistema Graduação

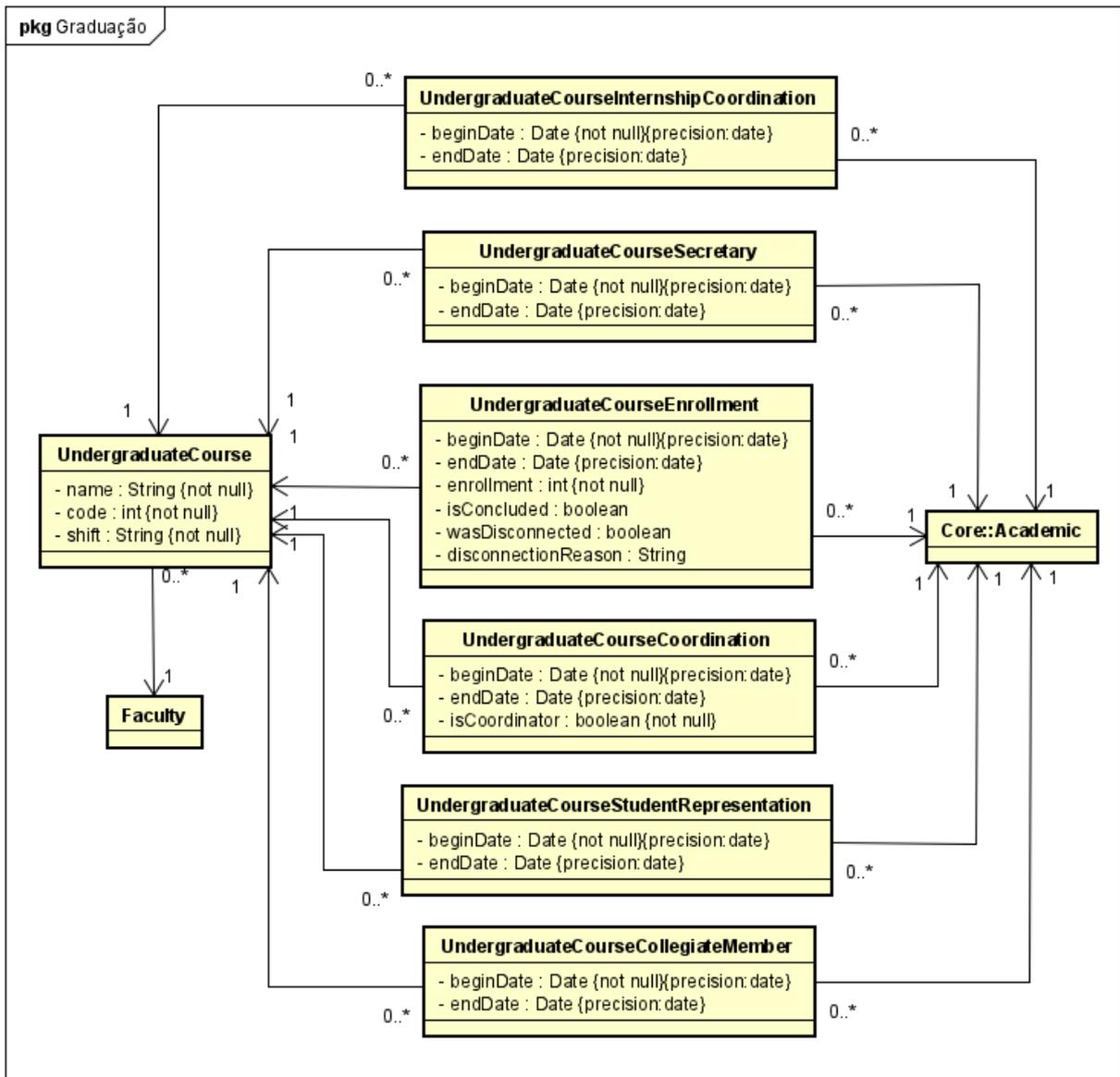


Figura 8 – Modelo de Entidades do subsistema Graduação

5.1.3 Subsistema PPG

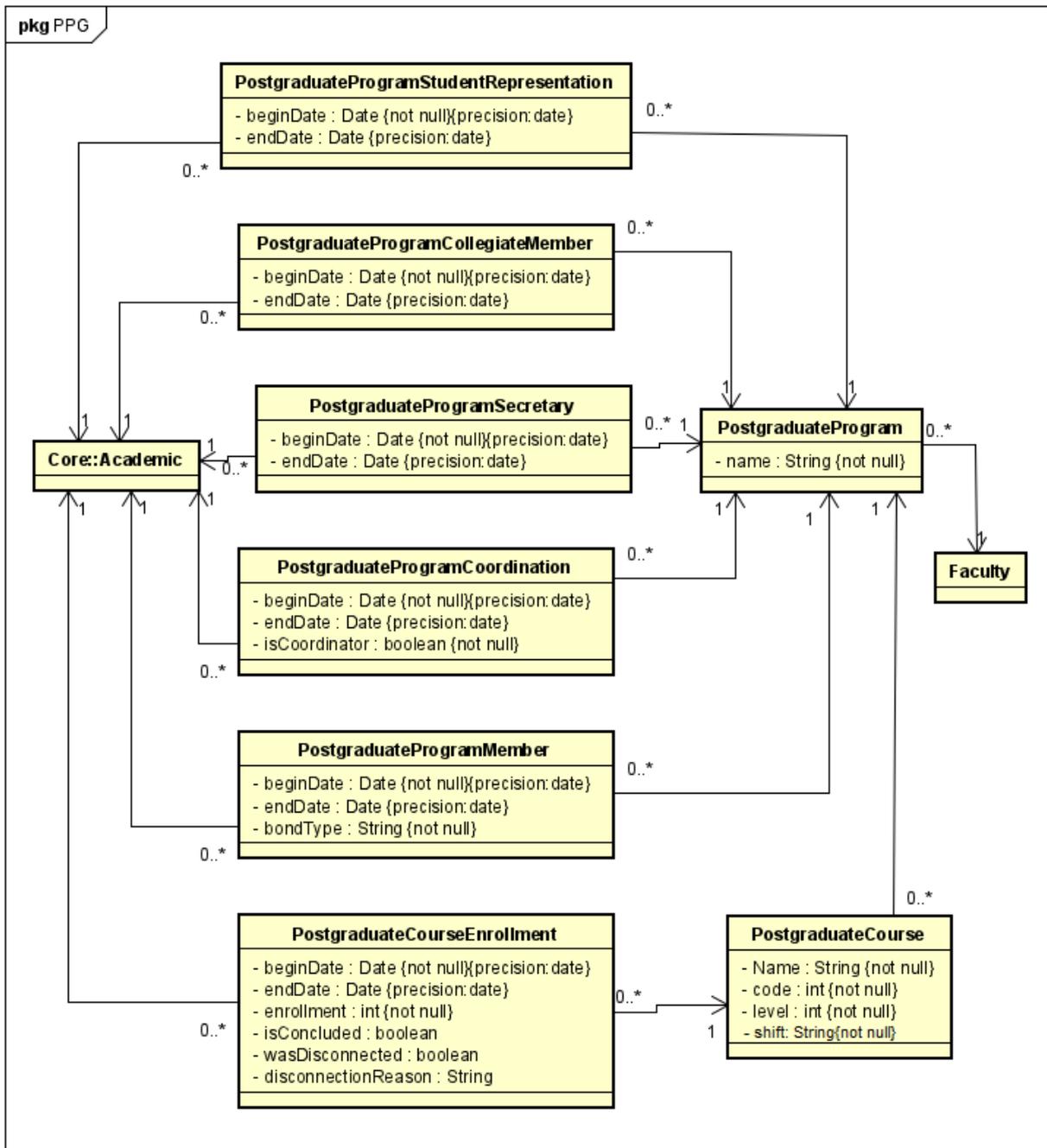


Figura 10 – Modelo de Entidades do subsistema PPG

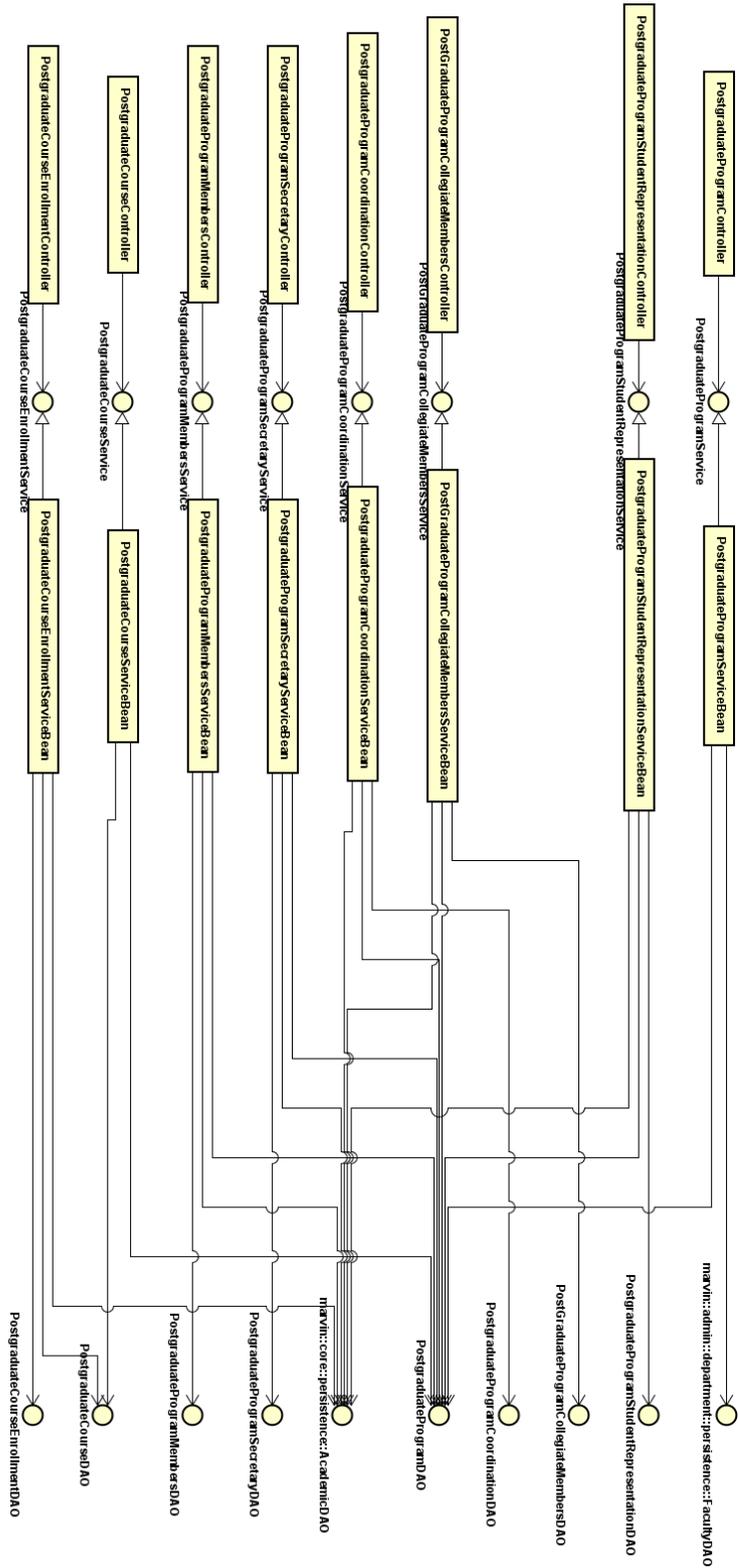


Figura 11 – Modelo de aplicação do subsistema PPG

5.2 Camada de Acesso a Dados

Como mostrado na Seção 4, a camada de acesso a dados (ou *Data Access Tier*) do *Macs - Sistema de controle de acessos do Marvin* é baseada em uma arquitetura de camadas de serviços, possuindo um pacote de **Persistência** (*Persistence*) e é responsável pela persistência de dados.

O modelo de Persistência representa as interfaces DAO (*Data Access Object*) existentes. Uma interface DAO é responsável pela persistência das classes de domínio do sistema, sendo responsável por definir os métodos de persistencia. Para cada classe de domínio que precisará acessar o banco de dados, deveremos ter uma interface <Nome da Entidade>DAO e uma classe <Nome da Entidade>JPADAO que implementará a interface DAO.

Todas as interfaces DAO existentes no módulo herdarão as propriedades da classe **BaseJPADAO**, do JButler, como é mostrado na Figura 12. Isto é feito com o propósito de se evitar a repetição de funcionalidades comuns a outros DAO's que existirão no sistema (ex.: criar objetos, deletar objetos, recuperar objetos..etc.). Além desses métodos já definidos pelo **BaseJPADAO**, também foram implementados e declarados nos diagramas dos subsistemas outros métodos que se referem a consultas específicas que precisaram ser implementadas para o funcionamento de algumas funcionalidades determinadas nos casos de uso.

Os modelo de persistência dos subsistemas Departamento, Graduação e PPG são mostrados nas figuras 13, 14 e 15 respectivamente.

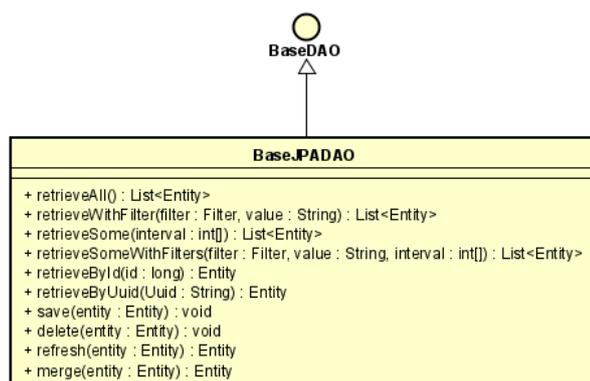


Figura 12 – Modelo de Persistência comum às DAO's do sistema

5.2.1 Subsistema Departamento

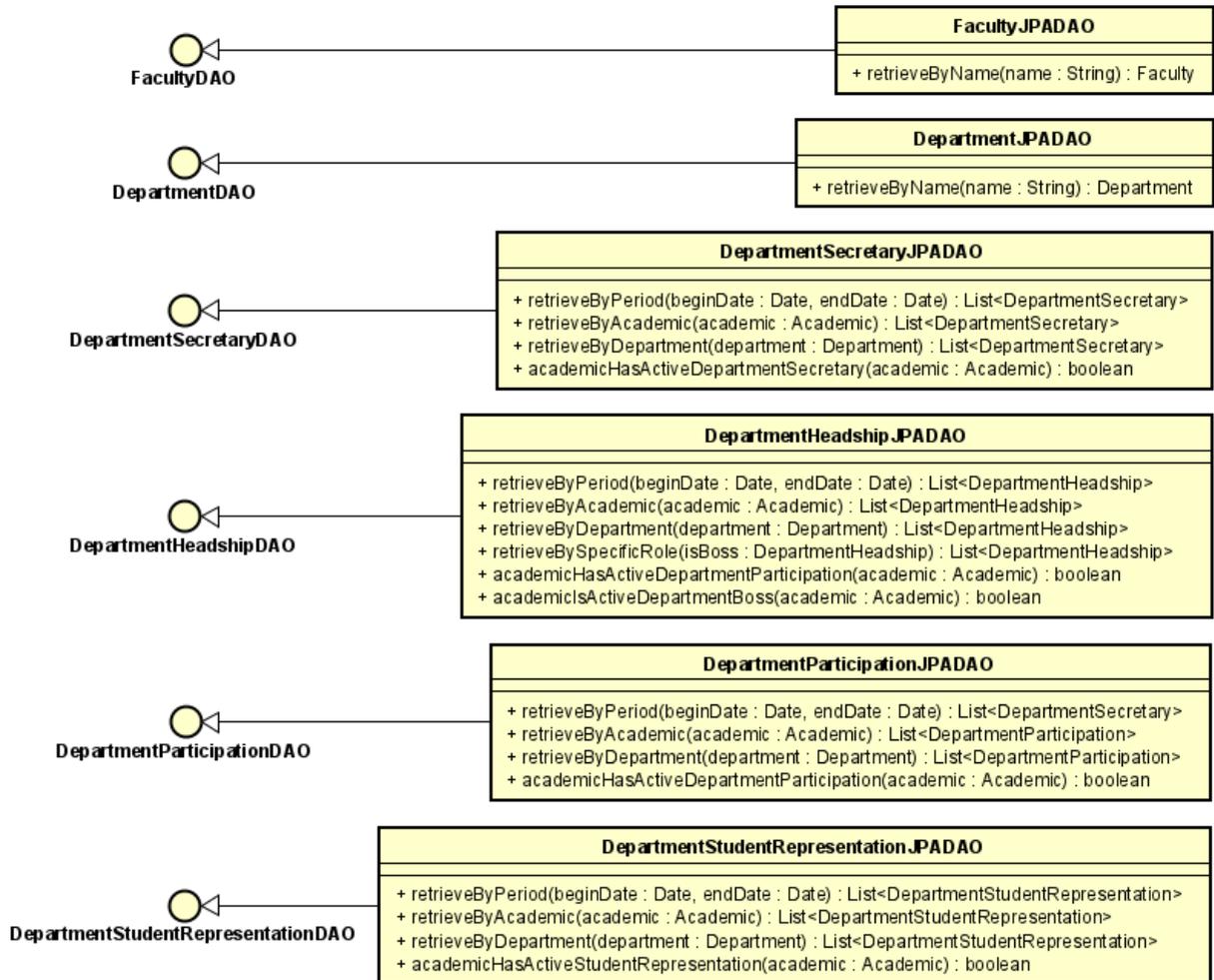


Figura 13 – Modelo de Persistência do subsistema Departamento

5.2.2 Subsistema Graduação

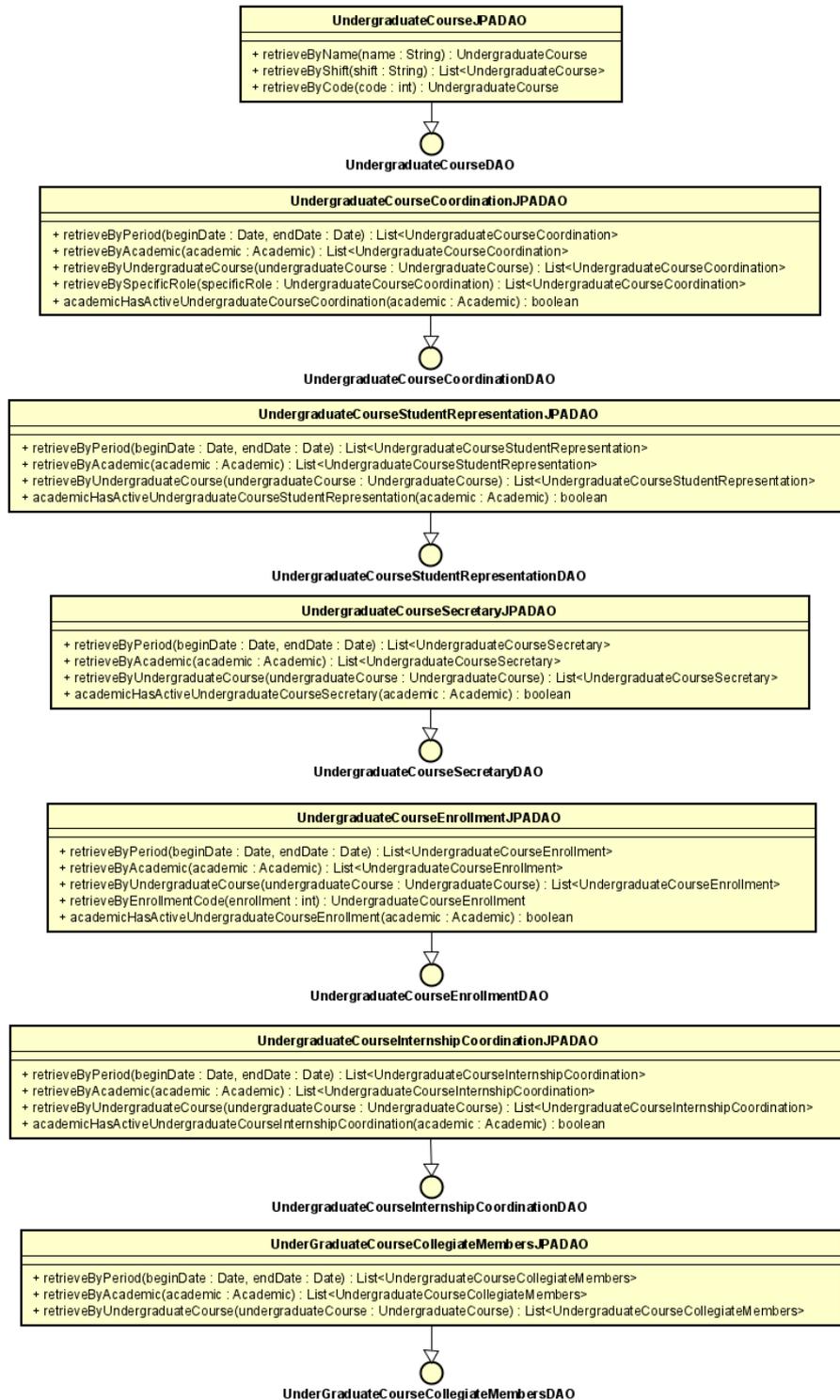


Figura 14 – Modelo de Persistência do subsistema Graduação

5.2.3 Subsistema PPG

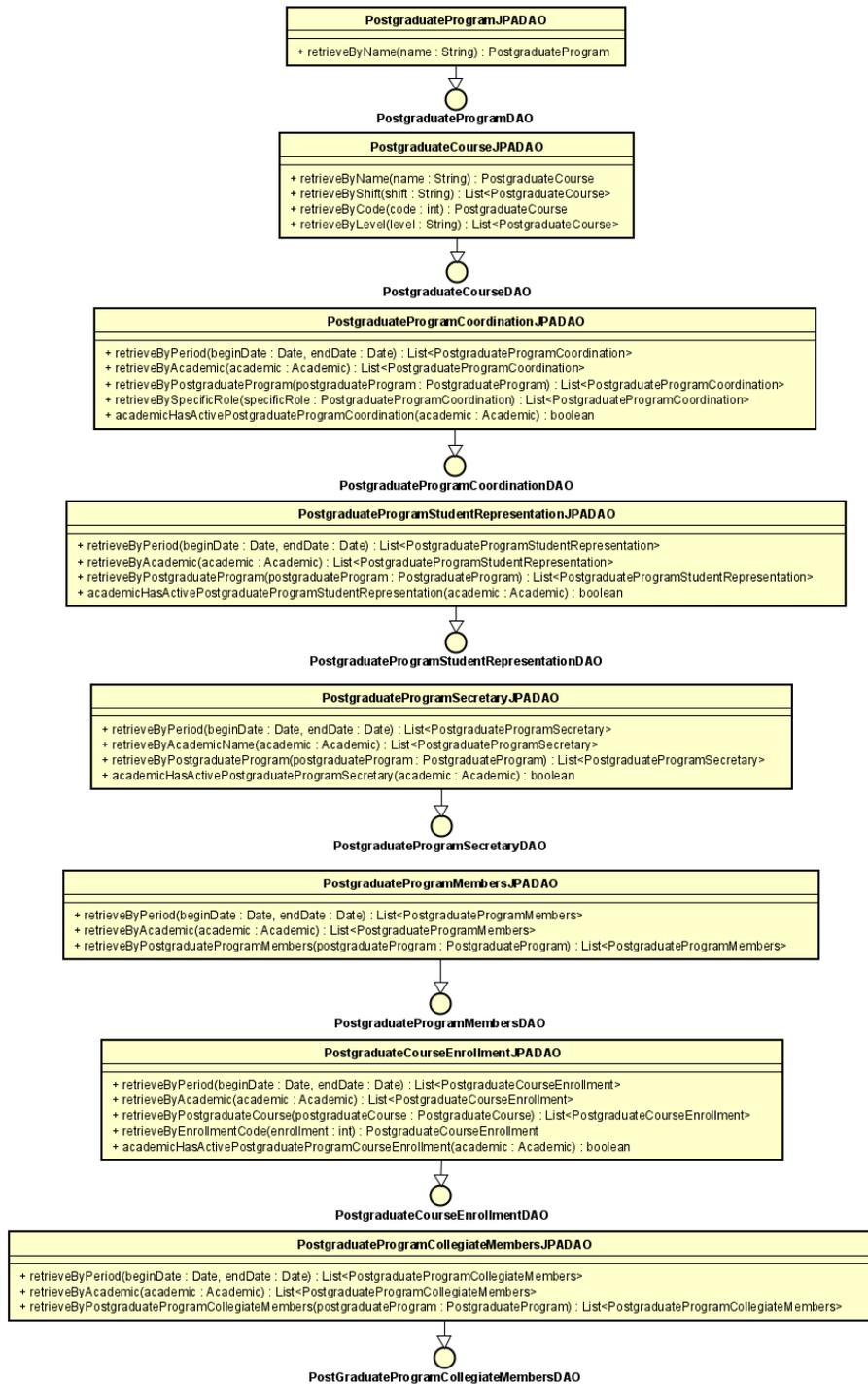


Figura 15 – Modelo de Persistência do subsistema PPG

5.3 Camada de Apresentação

As funcionalidades de criação, recuperação, atualização e exclusão (abreviação de CRUD, do inglês *create, retrieve, update, delete*) são comuns de todas as classes do módulo *Macs - Sistema de controle de acessos do Marvin*. Essas funcionalidades são similares para todos os casos de uso cadastrais relativos ao administrador do sistema (como detalhado no Documento de Requisitos deste trabalho) pois utilizam a ferramenta JButler (listada na Seção 2). O modelo de navegação referente a esses CRUD's pode ser visto na Figura 16.

De acordo com a arquitetura apresentada na Seção 4, a camada de apresentação possui as pastas de Visão e Controlador. Para cada entidade dentro do sistema, dentro da pasta Visão, teremos os formulários e as páginas, representadas por `form.xhtml` e `index.xhtml`. Já na pasta controlador, teremos uma classe `Manage<nome da entidade>Controller`, que consumirá os serviços da camada de negócio. Os campos do formulário `form.xhtml` não são exibidos na figura pois dependem da entidade específica que está sendo cadastrada.

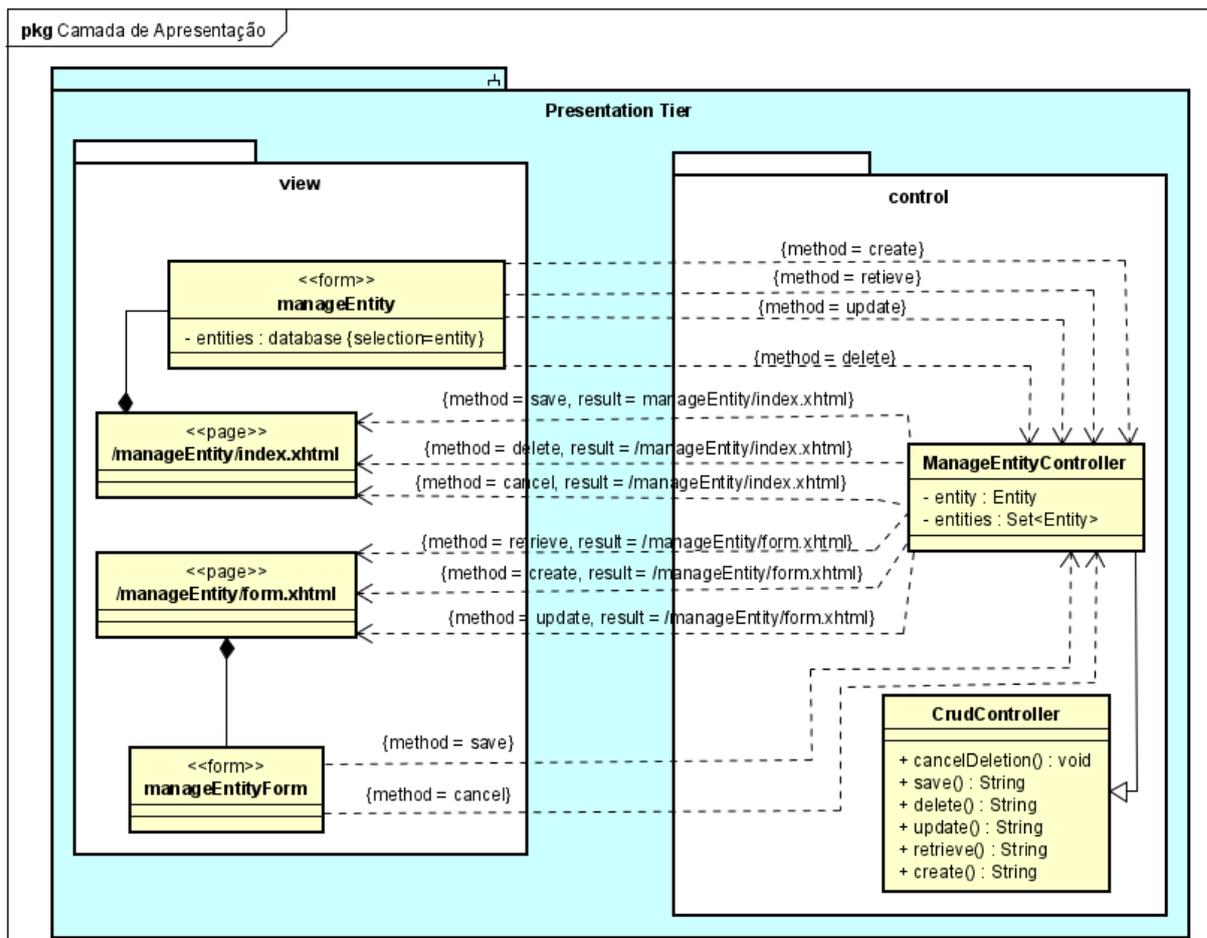


Figura 16 – Modelo de Navegação de um CRUD

Para os casos de uso onde as funções são diferentes das funções básicas oferecidas pelo JButler (listada na Seção 2), temos o modelo de navegação mostrado na Figura 17. Apesar de possuir um fluxo parecido com o fluxo do modelo de navegação de um CRUD, várias condições de controle de permissão foram incluídas nas funcionalidades implementadas. Assim como no modelo de navegação de um CRUD, para um modelo de navegação de uma indicação, teremos dentro da pasta **Visão** (*View*) formulários e páginas, representadas por form.xhtml e index.xhtml.

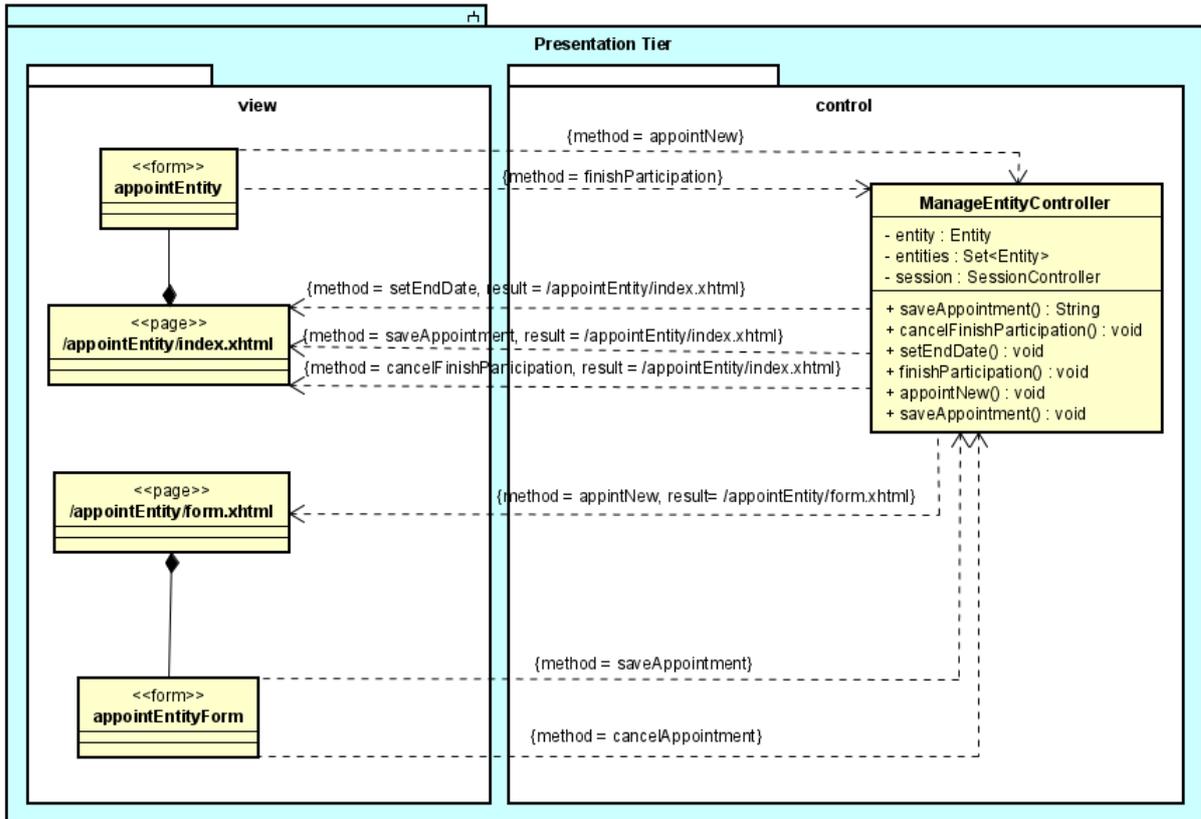


Figura 17 – Modelo de Navegação de indicação

Referências

FOWLER, M. *Patterns of Enterprise Application Architecture*. 1. ed. [S.l.]: Addison-Wesley, 2002. ISBN 9780321127426. Citado na página 8.

SOUZA, V. E. S. The FrameWeb Approach to Web Engineering: Past, Present and Future. In: ALMEIDA, J. P. A.; GUIZZARDI, G. (Ed.). *Engineering Ontologies and Ontologies for Engineering*. 1. ed. Vitória, ES, Brazil: NEMO, 2020. cap. 8, p. 100–124. ISBN 9781393963035. Disponível em: <<http://purl.org/nemo/celebratingfalbo>>. Citado na página 11.