



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
COLEGIADO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

Bruno Gama Nunes de Oliveira

**Módulo de controle acadêmico da
Pós-Graduação do Marvin**

Vitória, ES

2023

Bruno Gama Nunes de Oliveira

Módulo de controle acadêmico da Pós-Graduação do Marvin

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

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Colegiado do Curso de Engenharia de Computação

Orientador: Prof. Dr. Vítor E. Silva Souza

Vitória, ES

2023

Bruno Gama Nunes de Oliveira
Módulo de controle acadêmico da Pós-Graduação do Marvin/ Bruno Gama
Nunes de Oliveira. – Vitória, ES, 2023-
89 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Vítor E. Silva Souza

Monografia (PG) – Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Colegiado do Curso de Engenharia de Computação, 2023.

1. Palavra-chave1. 2. Palavra-chave2. I. Souza, Vítor Estêvão Silva. II.
Universidade Federal do Espírito Santo. IV. Módulo de controle acadêmico da
Pós-Graduação do Marvin

CDU 02:141:005.7

Bruno Gama Nunes de Oliveira

Módulo de controle acadêmico da Pós-Graduação do Marvin

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

Trabalho aprovado. Vitória, ES, (dia) de (mês) de (ano):

Prof. Dr. Vítor E. Silva Souza
Orientador

Profª Camila Zacche de Aguiar
Universidade Federal do Espírito Santo

Symone de Deus Miranda Gonçalves
Universidade Federal do Espírito Santo

Vitória, ES
2023

A minha família e amigos que sempre acreditaram em meu potencial, que me apoiaram ainda quando de maneira indireta.

Agradecimentos

Agradeço primeiramente à minha mãe Odinéia, que sempre me incentivou e investiu como pode em meus estudos, que me apoiou em minha decisão de voltar para o estado do Espírito Santo a fim de ingressar na UFES.

Minha gratidão também ao meu orientador Prof. Dr. Vítor E. Silva Souza, que me deu todo o suporte necessário para a conclusão do projeto de graduação, que sempre me deu retornos a fim de melhorar o conteúdo do trabalho em tempo extremamente rápido. Muito obrigado também por sua paciência e generosidade em me dar mais de uma chance para concluir o trabalho e que nunca desistiu de mim, embora eu mesmo tenha desistido em alguns momentos.

À minha esposa Natália, gostaria de agradecer imensamente todo o suporte, incentivo, carinho e amor que me deu durante todo o meu percurso na universidade, estando sempre comigo nos meus melhores e piores momentos.

Agradeço também a todos os amigos que fiz durante a graduação. Agradeço principalmente ao Elson, Lucas, Rafael e Vinícius, que além de amigos que conheci no curso de Engenharia da Computação agora são meus padrinhos de casamento, foram muitas madrugadas acordados estudando juntos para provas e trabalhos.

Aos meus tios Fábio e Regiane, que me acolheram e me deram abrigo em minha volta ao estado do ES. À minha tia Adalgisa que me incentivou e ajudou em meu retorno, meu sincero obrigado.

A todos que de forma direta ou indireta fizeram parte da minha formação, obrigado.

“Software é um lugar onde sonhos são plantados e pesadelos são colhidos, um pântano abstrato e místico onde demônios terríveis competem com mágicas panaceias, um mundo de lobisomens e balas de prata.”

(COX, 1990)

Resumo

O trabalho tem como objetivo apresentar a criação de um sistema de controle acadêmico para o Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Espírito Santo (UFES). O *Módulo de controle acadêmico da Pós-Graduação do Marvin* é uma ferramenta que visa melhorar o gerenciamento das demandas acadêmicas e a comunicação entre alunos, professores e coordenação do programa, permitindo o gerenciamento de períodos, de disciplinas, de oferta de disciplinas, de solicitação de matrícula em alguma disciplina, gestão de prazos para as solicitações, e permitir o gerenciamento de solicitações diversas tais como: solicitação de diploma, de aproveitamento de créditos e de solicitação de número de matrícula. O módulo será um sistema *Web* integrado ao *software* Marvin, onde o escopo do projeto não inclui a integração direta com os sistemas da UFES.

O Marvin é um sistema *Web* em desenvolvimento, que tem como propósito apoiar atividades administrativas dos cursos da instituição. Ele reúne projetos de graduação desenvolvidos por alunos de Ciência da Computação e Engenharia de Computação ao longo do tempo.

Ao final deste trabalho é apresentada a documentação completa das etapas realizadas até a concepção do *software*, que é capaz de suprir algumas das principais demandas acadêmicas do PPGI contribuindo para tornar os processos mais eficientes e fáceis de serem realizados. Todos os artefatos produzidos estão à disposição dos coordenadores do projeto Marvin para serem incorporados à sua base de código oficial.¹

Palavras-chaves: Marvin, PPGI, Sistema *Web*, Módulo.

¹ <<https://gitlab.labes.inf.ufes.br/marvin/marvin>>

Listas de ilustrações

Figura 1 – Fase de projeto (FALBO, 2018)	19
Figura 2 – Representação arquitetura MVC (WEISSMANN, 2012)	22
Figura 3 – Arquitetura baseada no padrão arquitetônico <i>Service Layer</i> (FOWLER, 2002).	24
Figura 4 – Diagrama de Casos de Uso do subsistema Gestão do PPG.	27
Figura 5 – Diagrama de Pacotes e os Subsistemas Identificados.	29
Figura 6 – Diagrama de Classes de Domínio.	29
Figura 7 – Arquitetura de Software (SOUZA, 2020).	32
Figura 8 – Pacotes do Módulo de Controle Acadêmico	33
Figura 9 – Páginas Web do Módulo de Controle Acadêmico	33
Figura 10 – Modelo de Entidades.	34
Figura 11 – Modelo de Aplicação.	35
Figura 12 – Modelo de Persistência do subsistema Gestão do PPGI.	36
Figura 13 – Modelo de Navegação para as telas de CRUD.	36
Figura 14 – Modelo de Navegação para a tela de solicitação de matrícula.	37
Figura 15 – Modelo de Navegação para a tela de solicitações diversas.	37
Figura 16 – Tela de <i>Login</i>	38
Figura 17 – Tela inicial	39
Figura 18 – Tela de Gerenciamento de Prazos (Listagem)	40
Figura 19 – Tela de Gerenciamento de Prazos (Inserção)	40
Figura 20 – Tela de Gerenciamento de Prazos (Validação de alteração de tipo)	41
Figura 21 – Tela de Gerenciamento de Prazos (Validação de deleção)	41
Figura 22 – Tela de Gerenciamento de Disciplinas (Listagem)	42
Figura 23 – Tela de Gerenciamento de Disciplinas (Inserção)	42
Figura 24 – Tela de Gerenciamento de Disciplinas (Validação de códigos iguais) . .	43
Figura 25 – Tela de Gerenciamento de Disciplinas (Validação de campos de acordo com o tipo)	43
Figura 26 – Tela de Gerenciamento de Períodos (Listagem)	44
Figura 27 – Tela de Gerenciamento de Períodos (Inserção)	44
Figura 28 – Tela de Gerenciamento de Períodos (Validação das datas)	45
Figura 29 – Tela de Gerenciamento de Períodos (Validação do nome do período) .	45
Figura 30 – Tela de Gerenciamento de Disciplina (Listagem)	46
Figura 31 – Tela de Gerenciamento de oferta das Disciplinas (Inserção)	46
Figura 32 – Tela de Gerenciamento de oferta das Disciplinas (Horário das aulas) .	47
Figura 33 – Tela de Gerenciamento de Solicitação de Matrícula visão do aluno (Listagem)	48

Figura 34 – Tela de Gerenciamento de Solicitação de Matrícula visão do aluno	49
Figura 35 – Tela de Gerenciamento de Solicitação de Matrícula visão do Administrador (Listagem)	49
Figura 36 – Tela de Gerenciamento de Solicitação de Matrícula visão do Administrador (Alteração)	50
Figura 37 – Tela de Gerenciamento de Requisição (Listagem)	51
Figura 38 – Tela de Gerenciamento de Requisição (Alerta com o prazo restante da requisição)	51
Figura 39 – Tela de Gerenciamento de Requisição (Validação de multiplas requisições com o mesmo tipo)	52

Lista de tabelas

Tabela 1 – Exemplos de estórias de usuário (Elaboração própria).	26
Tabela 2 – Descrição dos atores envolvidos nos casos de uso (Elaboração própria).	27
Tabela 3 – Subsistemas identificados e suas interdependências (Elaboração própria).	28
Tabela 4 – <i>Frameworks</i> da arquitetura do sistema separados por categoria.	34

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
AOP	Aspect Oriented Programming
API	Application Programming Interface
CDI	Contexts and Dependency Injection
DAO	Data Access Object
EE	Enterprise Edition
EJB	Enterprise Java Beans
EMF	Eclipse Modeling Framework
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JAAS	Java Authentication and Authorization Service
JPA	Java Persistence API
JSF	JavaServer Faces
JSP	JavaServer Pages
MVC	Model-View-Controller
PPGI	Programa de Pós-Graduação em Informática
PRPPG	Pró-Reitoria de Pesquisa e Pós-Graduação
SAPPG	Sistema Acadêmico de Pesquisa e Pós-Graduação
SQL	Structured Query Language
STI	Superintendência de Tecnologia da Informação
UFES	Universidade Federal do Espírito Santo
UML	Unified Modeling Language
WISs	Web-based Information Systems
XML	Extensible Markup Language

Sumário

1	INTRODUÇÃO	14
1.1	Motivação e Justificativa	14
1.2	Objetivos	15
1.3	Método de Desenvolvimento do Trabalho	15
1.4	Organização da Monografia	16
2	REFERENCIAL TEÓRICO E TECNOLOGIAS UTILIZADAS	17
2.1	Engenharia de Software	17
2.1.1	Análise e levantamento dos requisitos	18
2.1.2	Projeto	19
2.2	Desenvolvimento Web	20
2.2.1	Frameworks	21
2.2.2	Arquitetura MVC	22
2.2.3	Design Patterns e Clean Code	22
2.3	FrameWeb	23
3	ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS	25
3.1	Descrição do Escopo	25
3.2	Estórias de Usuário	25
3.3	Modelos de Caso de Uso	27
3.4	Identificação de Subsistemas	28
3.5	Diagrama de Classes	29
4	PROJETO DO SISTEMA	31
4.1	Tecnologias e Ferramentas Utilizadas	31
4.2	Arquitetura de Software	32
4.3	Modelagem FrameWeb	33
4.3.1	Camada de Negócio	34
4.3.2	Camada de Acesso a Dados	35
4.3.3	Camada de Apresentação	35
4.4	Apresentação	38
4.4.1	<i>Manage Deadlines</i>	39
4.4.2	<i>Manage School Subjects</i>	41
4.4.3	<i>Manage Periods</i>	43
4.4.4	<i>Manage Subject Offers</i>	45
4.4.5	<i>Manage Enrollment Requests</i>	48

4.4.6	<i>Requests</i>	50
5	CONCLUSÃO	53
5.1	Considerações Finais	53
5.2	Trabalhos Futuros	54
REFERÊNCIAS		56
APÊNDICES		58

1 Introdução

Tendo em vista a necessidade de formar professores e pesquisadores, a Universidade Federal do Espírito Santo (UFES) criou o Programa de Pós-Graduação em Informática (PPGI) no ano de 1994. Além do intuito da formação de professores e pesquisadores, o programa tem como objetivo oferecer formação tecnológica especializada para instituições públicas estaduais e nacionais bem como para empresas presentes no Espírito Santo, criar um corpo de pesquisadores voltados para a área de computação na UFES e ajudar no avanço científico e tecnológico no estado.¹

O PPGI oferta semestralmente vagas para o mestrado e doutorado assim como matrículas em matérias para os(as) alunos(as) já cadastrados(as) no programa. As ofertas de disciplinas são realizadas por meio de divulgação no site do PPGI e posteriormente os(as) alunos(as) são notificados(as) via e-mail. A matrícula é realizada por meio de preenchimento de formulário *online* e as demais demandas como solicitação de diploma e aproveitamento de créditos são realizadas através do e-mail.

Ao entender a dificuldade do gerenciamento das demandas citadas e a necessidade da diminuição do uso de e-mail para as solicitações, o módulo de controle acadêmico da Pós-Graduação do Marvin² apresenta-se como uma ferramenta rápida e eficiente voltada para os alunos, professores e coordenação do PPGI, visto que irá proporcionar aos usuários fácil acesso as informações e facilidade na solicitações por parte dos alunos.

1.1 Motivação e Justificativa

O Marvin é um sistema de informação que vem sendo desenvolvido ao longo dos anos a fim de apoiar atividades administrativas associadas aos cursos da UFES. Ele reúne diversos projetos de graduação desenvolvidos por alunos dos cursos de Ciência da Computação e Engenharia de Computação. Deste modo, foi identificada a necessidade de criar um novo módulo que suprisse as principais demandas do PPGI.

Analizando o sistema utilizado atualmente pelo PPGI para gerenciamento das demandas acadêmicas, verificou-se que é exigido muito trabalho manual dos funcionários da secretaria e que há dificuldade na comunicação dos alunos com o programa.

Diante desta situação, há uma sobrecarga no recebimento de e-mails, visto que este é o canal disponibilizado para solicitações das demandas e esclarecimento de dúvidas dos(as) alunos(as).

¹ <<https://informatica.ufes.br/pt-br/pos-graduacao/PPGI>>

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

Deste modo, constatou-se que o PPGI necessita de um *software* para gerenciar demandas relacionadas ao programa de pós-graduação a fim de otimizar o processo existente. Pretende-se atender esta necessidade por meio de um módulo do sistema Marvin.

No entanto, é importante delimitar que o escopo deste módulo é prover informações para que o(a) secretário(a) ou o(a) coordenador(a) do PPGI possa efetivar as ações (ex.: oferta de disciplinas, matrículas) no sistema oficial da PRPPG/UFES (SAPPG). Integrar o módulo diretamente com os sistemas da UFES sob gestão da Superintendência de Tecnologia da Informação (STI) está fora do escopo deste projeto, sendo uma sugestão de melhoria futura.

1.2 Objetivos

Para alcançar o objetivo de desenvolver um sistema *Web* que facilite o gerenciamento das demandas do PPGI, foram elencados os seguintes objetivos específicos:

1. Levantamento e documentação dos requisitos do módulo, englobando funções de oferta de disciplinas e gestão de solicitações (como aproveitamento de créditos e confecção de diplomas) e de prazos;
2. Projeto e documentação da arquitetura do módulo, integrada com a arquitetura do Marvin;
3. Implementação e testes do código-fonte do módulo, integrado à base de código do Marvin.

1.3 Método de Desenvolvimento do Trabalho

O método de desenvolvimento que será utilizado para elaborar o módulo é contido pelos seguintes tópicos:

1. Revisão bibliográfica: verificação de boas práticas de Engenharia de *Software* e Engenharia de Requisitos, programação orientada a objetos, desenvolvimento de *software Web* e projeto de banco de dados relacionais.
2. Desenvolvimento da documentação do sistema: definir os documentos do sistema, sendo eles (i) o Documento de Especificação de Requisitos, que contém uma visão geral do produto, descrição do minimundo, *stakeholders*, estórias de usuário, regras de negócio e casos de uso; (ii) Documento de Projeto de Sistemas, que contém a arquitetura do *software* que será desenvolvido e projeto detalhado dos seus componentes, com a especificação das classes, modelos, classe dos atributos e detalhamento do comportamento esperado de todos os casos de uso do módulo.

3. Uso de Tecnologias: estudo das tecnologias que serão utilizadas no desenvolvimento deste módulo, sendo elas: linguagem de programação Java; ambiente de desenvolvimento Eclipse; banco de dados mySQL; ferramenta MySQL Workbench; *framework* PrimeFaces para o desenvolvimento da interface e o *framework* JUnit para a criação de testes unitários.
4. Implementação: desenvolvimento do módulo do sistema com todas funcionalidades que foram descritas, utilizando a filosofia de desenvolvimento *Clean Code* e *Design Patterns*.
5. Redação da monografia: escrita da monografia, em L^AT_EX com o editor TeXstudio. É utilizado o *template* abnTeX, que atende as normas da Associação Brasileira das Normas Técnicas (ABNT) para a produção de documentos técnicos e científicos brasileiros.

1.4 Organização da Monografia

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

- O Capítulo 2 apresenta os aspectos relativos ao conteúdo teórico e tecnológico relevante para o trabalho;
- O Capítulo 3 apresenta o resultado do levantamento de requisitos conduzido para este projeto;
- O Capítulo 4 apresenta a arquitetura do sistema desenvolvido, bem como capturas de tela que ilustram suas diversas funcionalidades;
- O Capítulo 5 apresenta as considerações finais do trabalho.

2 Referencial Teórico e Tecnologias Utilizadas

Para dar embasamento teórico aos conceitos que serão utilizados para o desenvolvimento do módulo, foram levantados tópicos fundamentais que estão subdivididos em três seções, sendo elas: Seção 2.1, que diserta sobre a Engenharia de Software, Seção 2.2 que trata dos conceitos envolvidos no desenvolvimento de plataformas *Web*, e Seção 2.3 que aborda a metodologia FrameWeb.

2.1 Engenharia de Software

A sociedade contemporânea tem vivenciado um grande desenvolvimento tecnológico com cada vez mais inserção de soluções tecnológicas em seu meio. Com esta crescente, é necessário cada vez mais *softwares* capazes de facilitar processos que por muitas vezes antes eram exercidos predominantemente por trabalho manual em vez de digital. A Engenharia de Software surgiu com o objetivo de melhorar a qualidade dos *softwares*, e aprimorar a eficácia dos processos envolvidos na elaboração de uma aplicação (BARCELLOS, 2018).

A Engenharia de Software é recomendada para resolução de problemas grandes e/ou complexos que envolvem a construção de um *software*. Ela aborda tópicos relacionados a processos, técnicas, métodos e ferramentas utilizadas para criação de um *software* (BARCELLOS, 2018).

De modo geral, a metodologia para o desenvolvimento de um *software* é composta desde o levantamento dos primeiros dados, juntamente ao cliente, até a sua entrega ao solicitante. Esta metodologia pode ser subdividida pelos seguintes itens:

- Análise e levantamento dos requisitos;
- Projeto;
- Implementação;
- Testes;
- Entrega;
- Implantação.

A base deste módulo será abrangido pela análise e levantamento dos requisitos, projeto e implementação em si.

2.1.1 Análise e levantamento dos requisitos

Esta etapa é primordial para a criação de um *software*, pois é nela que será feita a captação de informações para compreensão dos problemas em si juntamente ao cliente, para então ser realizado o levantamento dos requisitos que devem ser satisfeitos para resolvê-los. Qualquer erro nesta fase é extremamente prejudicial às outras etapas subsequentes, visto que ela influencia diretamente nas demais fases do projeto. De acordo com [Kotonya e Sommerville \(2011\)](#), é necessário que haja uma validação dos requisitos, porque erros em um documento de requisitos podem gerar altos custos de retrabalho quando descobertos durante o desenvolvimento ou após o sistema já estar em serviço.

Os requisitos de sistema são os relatos do que o software deve fazer, as funções que ele deve ter e as restrições para o seu bom funcionamento ([KOTONYA; SOMMERVILLE, 2011](#)). Eles podem ser divididos em três tipos diferentes, sendo:

- Requisitos funcionais: são responsáveis por elencar o que o sistema deve fazer, são geralmente caracterizados de forma abstrata para que o usuário possa compreender, porém também existem requisitos mais específicos que caracterizam em detalhes o que o *software* deve fazer ([KOTONYA; SOMMERVILLE, 2011](#));
- Requisitos não funcionais: diferentemente dos requisitos funcionais, os requisitos não funcionais não estão ligados de modo direto ao que o sistema deve fazer, mas sim a propriedades do sistema, como tempo de resposta e confiabilidade, então eles podem afetar diretamente a arquitetura do programa ([KOTONYA; SOMMERVILLE, 2011](#));
- Requisitos de domínio ou regras de negócio: são as premissas de negócio que podem estar ligadas a requisitos funcionais do sistema. Temos como exemplo cálculos de tributação para um determinado produto ou a condição para aprovação de um aluno em uma determinada matéria ser possuir média maior ou igual a 7.

Para o desenvolvimento deste projeto utilizou-se técnicas provenientes da Engenharia de Software que foram estudadas ao longo da graduação, tais como: descrição do minimundo, que abrange a realidade a ser apresentada pelo sistema; levantamento de requisitos funcionais por meio de estórias de usuário, que caracteriza as funcionalidades por meio de histórias que descrevem como o sistema deve comportar-se ([PRESSMAN; MAXIM, 2016](#)); listagem de requisitos não funcionais; regras de negócio e modelagem de casos de uso, que se comportam como um contrato, que explica o comportamento do *software* sob várias condições ligadas a respostas do sistemas de acordo com solicitações do usuário ([PRESSMAN; MAXIM, 2016](#)).

2.1.2 Projeto

Após o levantamento dos requisitos junto ao usuário é possível passar para a fase de projeto, que é a etapa responsável por interligar requisitos tecnológicos com os requisitos do sistema. Ela se encontra no centro do processo de desenvolvimento conforme podemos ver na Figura 1.

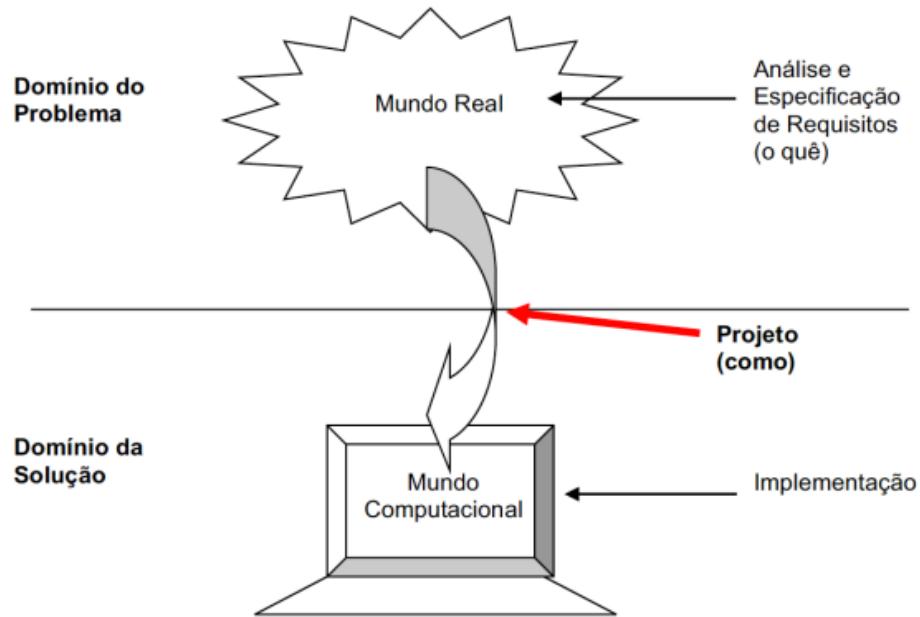


Figura 1 – Fase de projeto (FALBO, 2018).

Nesta fase, serão identificados os componentes de software e os relacionamentos com base nos requisitos do cliente (KOTONYA; SOMMERVILLE, 2011). Então, é fundamental entender as tecnologias disponíveis para o desenvolvimento do projeto e os ambientes de hardware e software que a aplicação será construída (FALBO, 2018).

No projeto está presente a modelagem de como o *software* será construído, são considerados os requisitos de sistema que foram levantados juntamente com os requisitos não funcionais, que são de viés tecnológico. De acordo com Pressman e Maxim (2016) esta é a primeira atividade que leva em consideração os aspectos tecnológicos no processo de criação do sistema.

Este estágio pode ser dividido em duas grandes etapas: projeto de arquitetura, que tem como função definir a arquitetura do sistema que representará a estrutura da aplicação com os seus principais componentes; e o projeto detalhado, que irá detalhar todos os elementos que foram identificados na etapa de projeto de arquitetura (BARCELLOS, 2018).

2.2 Desenvolvimento Web

Com a constante evolução da tecnologia e a necessidade de *softwares* integráveis com outros, diversos sistemas *desktop* ficaram obsoletos (legado). Isto ocorre devido a alguns fatores, tais como: tecnologia, acoplamento de código, ou arquitetura inapropriada, visto que a tecnologia presente na época ficou defasada com tempo (CHAVES, 2017). Outros pontos relevantes a serem considerados é que os sistemas legados são antigos, com mais de uma década de uso, e utilizam de linguagens de programação legadas, banco de dados legados, e operam de forma independente, com pouca ou nenhuma interface com outros sistemas (CHAVES, 2017).

Com a disseminação da Internet para grande parte da população mundial e com a criação de diversos dispositivos capazes de acessá-la, houve um grande aumento no número de sistemas *Web*. Estes *softwares*, por serem *Web*, permitem ao usuário facilidade no acesso, pois pode ser conectado através de qualquer dispositivo que tenha acesso à Internet. Para além disso, é necessário destacar outra vantagem deste tipo de solução em comparação a sistemas legados, tendo ela a possibilidade de *designs* mais intuitivos e visualmente mais bonitos e a maior facilidade de manutenção sem a necessidade do programador estar fisicamente presente onde está o *software*.

As aplicações *Web* utilizam o protocolo *Hypertext Transfer Protocol* (HTTP) que é um protocolo utilizado para transferência de dados, com isto é possibilitada a comunicação entre diferentes dispositivos conectados à Internet.

Para o desenvolvimento deste módulo será utilizada a tecnologia Jakarta *Enterprise Edition* (Jakarta EE), que anteriormente era chamada de Java EE, na qual foi cedido pela Oracle à Eclipse Foundation e renomeado para Jakarta EE.¹ O Jakarta EE é uma plataforma voltada para a criação de aplicações Java ou para o desenvolvimento de *softwares Web*, ela inclui funcionalidades e bibliotecas para implementar aplicações distribuídas (FARIA, 2015).

Esta plataforma é considerada como uma plataforma guarda-chuva, ela possui algumas especificações com diversos objetivos, estão entre as mais conhecidas:

- *Servlets*: eles são executados no servidor, utilizados para gerar conteúdo dinâmico para a *Web*, ex: HTML e XML.
- *Java Persistence API* (JPA): JPA é um *framework* para persistir objetos Java no banco de dados, com ele é possível desenvolver *softwares* sem precisar escrever comandos em SQL.
- *Enterprise Java Beans* (EJB): pode-se dizer que é um dos principais componentes

¹ <<https://jakarta.ee/>>.

do Jakarta EE, ele disponibiliza diversos componentes prontos, com isso torna o desenvolvimento rápido e mais eficiente.

- *JavaServer Faces* (JSF): é um *framework* de interface com o usuário, com ele é possível simplificar a construção de telas e utilizar componentes reutilizáveis.
- *Contexts and Dependency Injection* (CDI): fornece uma arquitetura que permite aos componentes existirem dentro do ciclo de vida de um sistema com escopos bem definidos.
- *Java Authentication and Authorization Service* (JAAS): é uma *Application Programming Interface* (API) de autenticação e autorização do Java.

2.2.1 Frameworks

Framework é um tema citado diversas vezes em diferentes tópicos no decorrer deste trabalho. Eles permitem reutilizar uma arquitetura inteira como um componente isolado, com isso é possível aumentar a produtividade, qualidade e manutenibilidade devido a possibilidade do reuso de algo já existente.

Este termo foi criado por volta do ano de 1980 com a popularização da linguagem Smalltalk, paralelamente às bibliotecas de classes, dando início à construção de *frameworks* de aplicação ([MALDONADO et al., 2002](#)).

Inúmeros *frameworks* foram desenvolvidos nas últimas décadas, eles tem como objetivo o reuso de código, sendo um conjunto de objetos que se comunicam entre si para a execução de um grupo de responsabilidades. Ele é definido por [Coad \(1992\)](#) como um esqueleto de classes, objetos e relacionamentos agrupados para construir aplicações específicas. De maneira similar é definido por [Johnson \(1988\)](#) como um conjunto de classes abstratas e concretas que provê uma infra-estrutura genérica de soluções para um conjunto de problemas.

De acordo com [Souza \(2007\)](#) é possível estruturar os *frameworks* em categorias distintas, sendo algumas das mais populares:

- *Frameworks* MVC (Controladores Frontais);
- *Frameworks* Decoradores;
- *Frameworks* de Mapeamento Objeto/Relacional;
- *Frameworks* de Injeção de Dependência (Inversão de Controle);
- *Frameworks* para Programação Orientada a Aspectos (AOP);
- *Frameworks* para Autenticação e Autorização.

2.2.2 Arquitetura MVC

Juntamente com o aumento do número de aplicações *Web*, surgiram diversas arquiteturas e *frameworks* com o intuito de facilitar o processo de desenvolvimento de um *software*. Uma das principais destas arquiteturas é a MVC, que hoje em dia é um dos padrões mais utilizados pelos *frameworks Web* (WEISSMANN, 2012). Esta arquitetura divide o desenvolvimento em três camadas: *Model*, *View* e *Control*.

O *Model* também conhecido como modelo é a parte que fica toda a lógica de negócio, também está presente nesta parte tudo que é referente a persistência de dados, controle de cache, integração com outras aplicações etc. (WEISSMANN, 2012).

A *View* (visualização) é a camada visível ao usuário, é onde os dados presentes no banco de dados são representados visualmente como por exemplo tabelas, diagramas e telas (WEISSMANN, 2012).

A camada de controle ou controlador é a que faz interligação entre a *View* e ao *Model*. Ela pode ser acionada no momento que o usuário clica em um botão de pesquisa por exemplo, esta ação irá interagir com a *View* chamando o controlador referente a aquela ação, após isso será chamado o *Model* referente a aquela rota que está presente no controlador. Nesta camada também é realizado a conversão dos dados em um formato compatível com a interface disponível na camada de negócio (WEISSMANN, 2012).

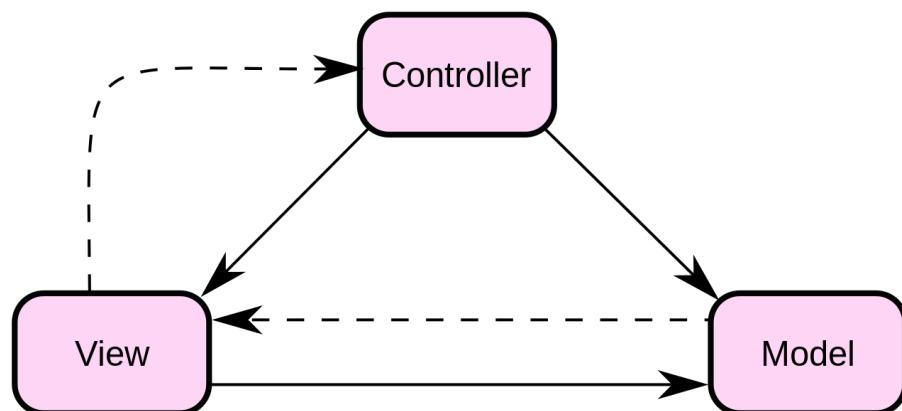


Figura 2 – Representação arquitetura MVC (WEISSMANN, 2012)

Este padrão soluciona problemas de acoplamento, visto que ele divide a aplicação em camadas, fazendo que seja mais fácil a substituição de uma camada inteira ou parte dela, pois as camadas são independentes entre si. (WEISSMANN, 2012).

2.2.3 Design Patterns e Clean Code

Embora esses dois termos não sejam obrigatórios no desenvolvimento de um *software* os mesmos são de grande utilidade, pois facilitam na legibilidade e manutenção do código, possibilita escalabilidade do sistema e permitem aumentar a velocidade de desenvolvimento.

A fim de criar um *software* que possa ser alterado e escalado facilmente, o módulo desenvolvido neste trabalho foi codificado seguindo padrões de projeto (*Design Patterns*) e a filosofia de código limpo (*Clean Code*).

Os *Design Patterns* são padrões que são utilizados como solução para um determinado contexto. Estes padrões não descrevem qualquer tipo de solução existente na programação mas sim padrões que foram empregados com sucesso na solução de problemas em diferentes contextos (GUERRA, 2014).

Estes padrões também não são simplesmente trechos de códigos para que desenvolvedores possam utilizar como exemplos para soluções cotidianas, mas sim conhecimentos que devem ser adquiridos por todo desenvolvedor (GUERRA, 2014). Com eles em mente é possível resolver problemas complexos facilmente utilizando-os separadamente ou em conjunto uns com outros.

Códigos confusos e mal feitos causam dificuldade na legibilidade e manutenção de software, fazendo com que programadores percam tempo desnecessário no desenvolvimento ou manutenção de um código. Os mesmos também podem causar falhas no sistema devido a alterações futuras, trazendo transtorno e insatisfação para o usuário. O *Clean Code* é um conceito que visa minimizar tais fatos, ele defende que um código deve ser implementado de forma limpa a fim de evitar confusões e erros (MARTIN, 2009). O seu principal objetivo é aplicar técnicas simples que possibilitem facilitar a leitura de um código, com isso o tornando de fácil entendimento e mostrando a sua real intenção.

2.3 FrameWeb

O método *FrameWeb* (*Framework-based Design Method for Web Engineering*) é um abordagem para o projeto de WISs (*Web-based Information Systems*) baseados em *frameworks*. Atualmente, o método está concentrado em auxiliar na fase de projeto arquitetural do *software* (SOUZA, 2020).

Este método também é baseado em orientação a objetos, utilizando a modelagem de UML (*Unified Modeling Language*) durante o processo. O *FrameWeb* é direcionado para aplicações baseadas em *frameworks*. O *FrameWeb* prescreve quatro modelos, todos baseados no diagrama de classes da UML: o modelo de entidade, de persistência, de aplicação e o modelo de navegação (SOUZA, 2020).

- Modelo de entidade: define os relacionamentos dos objetos do domínio do problema por meio de diagrama UML.
- Modelo de persistência: utilizado para demonstrar as classes de acesso aos dados (DAO) existentes no projeto, elas são responsáveis pela modelagem de persistência das informações no banco de dados.

- Modelo de aplicação: utilizado para simbolizar as classes de serviço, que são encarregadas pela implementação dos casos de uso e suas dependências.
- Modelo de navegação: usado para demonstrar os diferentes componentes que são integrados ao *framework* MVC, é nesta camada que é formada a interface com o usuário.

No *FrameWeb* a arquitetura é baseada no padrão arquitetônico *Service Layer* (Camada de Serviço) ([FOWLER, 2002](#)), segundo a qual o *software* é dividido em três principais camadas conforme pode ser visto na Figura 3: camada de negócios, camada de apresentação e camada de acesso aos dados ([SALVATORE, 2016](#)).

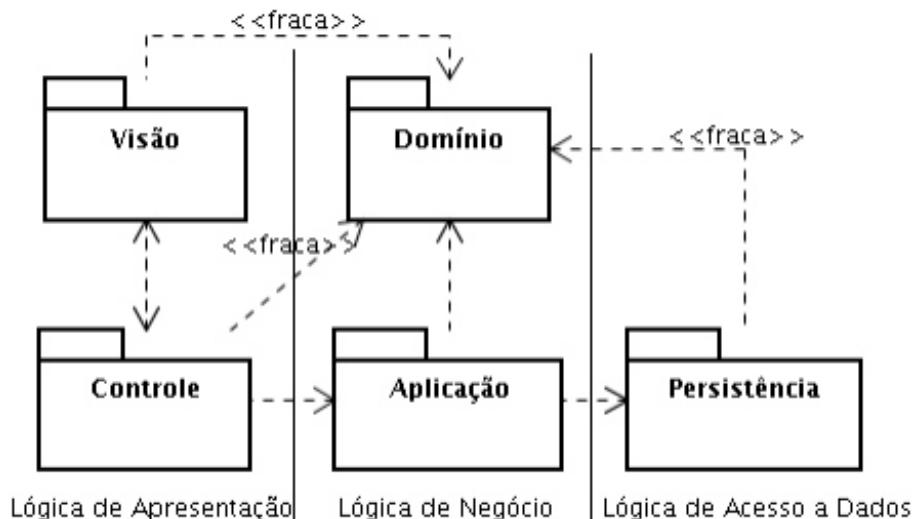


Figura 3 – Arquitetura baseada no padrão arquitetônico *Service Layer* ([FOWLER, 2002](#)).

- Camada de negócios (*Business Tier*): ela é subdividida em duas camadas: lógica de aplicação (*Application*) e lógica de Domínio (*Domain*). Neste ponto que são providas as funcionalidades de acordo com as regras de negócio do sistema;
- Camada de apresentação (*Presentation Tier*): análogo ao seu nome, esta faixa é referente a criação da interface onde o usuário irá interagir com o sistema, nela estão presentes duas camadas provenientes da arquitetura MVC, sendo elas a visão (*View*) e controle (*Control*);
- Camada de acesso a dados (*Data Access Tier*): aqui são realizados os tratamentos interligados à persistência de informações no banco de dados.

O Módulo de controle acadêmico da Pós-Graduação do Marvin foi desenvolvido com base nos conceitos presentes no método *FrameWeb*, abordando em si todas as três camadas que foram citadas.

3 Análise e Especificação de Requisitos

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

3.1 Descrição do Escopo

O objetivo do *Módulo de controle acadêmico da Pós-Graduação do Marvin* é otimizar o processo de gerenciamento de demandas acadêmicas pelo Programa de Pós-Graduação em Informática (PPGI) que atualmente é realizado manualmente pela secretaria do programa e apresenta dificuldades na comunicação com os alunos. O canal atual de solicitações de demandas e esclarecimentos de dúvidas é por e-mail, o que acarreta uma sobrecarga no recebimento desses e-mails.

O sistema a ser desenvolvido é um módulo do sistema Marvin, que já é utilizado para o gerenciamento de atividades acadêmicas. O módulo em questão será implementado com o objetivo de suprir as demandas do PPGI e amenizar a sobrecarga de trabalhos manuais e o uso excessivo de e-mails.

Em relação à oferta de disciplinas, atualmente a divulgação é feita no site do PPGI e a matrícula é realizada via formulário enviado por e-mail. O sistema deve possibilitar o cadastro de disciplinas e períodos, permitir que os alunos possam ver as ofertas de maneira simples, possibilitar a visualização do estado atual da solicitação e dar ao responsável pelo gerenciamento uma visão clara e objetiva dos requerimentos.

Além disso, a solicitação de aproveitamento de créditos de disciplinas, solicitação de número de matrícula e a solicitação de diploma devem ser feitas via sistema, permitindo um fácil controle e visualização da situação dos requerimentos tanto para o aluno quanto para a secretaria.

No geral, o sistema deve possibilitar a automatização do processo de gerenciamento de demandas acadêmicas, tornando-o mais eficiente e simples tanto para os alunos quanto para a secretaria do PPGI.

3.2 Estórias de Usuário

Uma estória de usuário (*User Story*) é uma maneira simples de expressar desejos de um usuário acerca das funcionalidades de um *software*, estes desejos são denominados

como critérios de aceitação, que são escritos do ponto de vista de um utilizador em relação a algum *software*, descrevendo como o sistema pode ser útil para ele. Esta abordagem auxilia no levantamento de funcionalidades do sistema, provendo informações que ajudam na concepção de uma aplicação.

Para este projeto foram considerados como *stakeholders* Aluno, Professor, e Secretaria/Coordenação PPGI, e para cada um deste tipo de usuário foram levantadas diversas *user stories* que estão presentes no apêndice deste trabalho. Exemplos de estórias de usuário pode ser vistos na Tabela 1. Estas estórias foram criadas com base pelo que foi dito brevemente na Seção 3.1 sobre cadastro de disciplinas e períodos.

Tabela 1 – Exemplos de estórias de usuário (Elaboração própria).

Descrição:	Como secretaria/coordenação, quero cadastrar disciplinas, para que os professores possam ofertá-las no período vigente.
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, deve ser informado: nome, tipo,¹ código, créditos, carga horária, ementa e bibliografia; – CA2: Não é permitido cadastrar disciplinas com o mesmo código; – CA3: Consultas podem ser feitas por: nome, código e tipo; – CA4: Disciplinas podem ser excluídas apenas se não estiverem associadas a outros objetos.
Descrição:	Como secretaria/coordenação, quero cadastrar períodos letivos, para que o professor possa ofertar disciplinas.
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, deve ser informado: nome, período (data inicial e final) de oferta, período de matrícula, datas de início e fim do período letivo (aulas); – CA2: Não é permitido cadastrar um período letivo que tenha períodos de oferta, de matrícula ou de aula conflitantes; – CA3: Não é permitido cadastrar períodos letivos com o mesmo nome; – CA4: Consultas podem ser feitas por: nome; – CA5: Períodos letivos podem ser excluídos apenas se não estiverem associados a outros objetos.

¹ De acordo com o regimento atual do PPGI, um item da organização curricular dos cursos de pós-graduação pode ser de um dos seguintes tipos: Disciplina Regular; Disciplina Eventual; Tópico Especial; Estágio de Docência; Estudo Dirigido; Seminário Temático; Dissertação de Mestrado; Exame de Qualificação de Doutorado; Exame de Proposta de Tese de Doutorado; Tese de Doutorado

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 (Elaboração própria).

Ator	Descrição
Ator 1	Aluno.
Ator 2	Professor.
Ator 3	Secretaria/Coordenação PPGI.

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

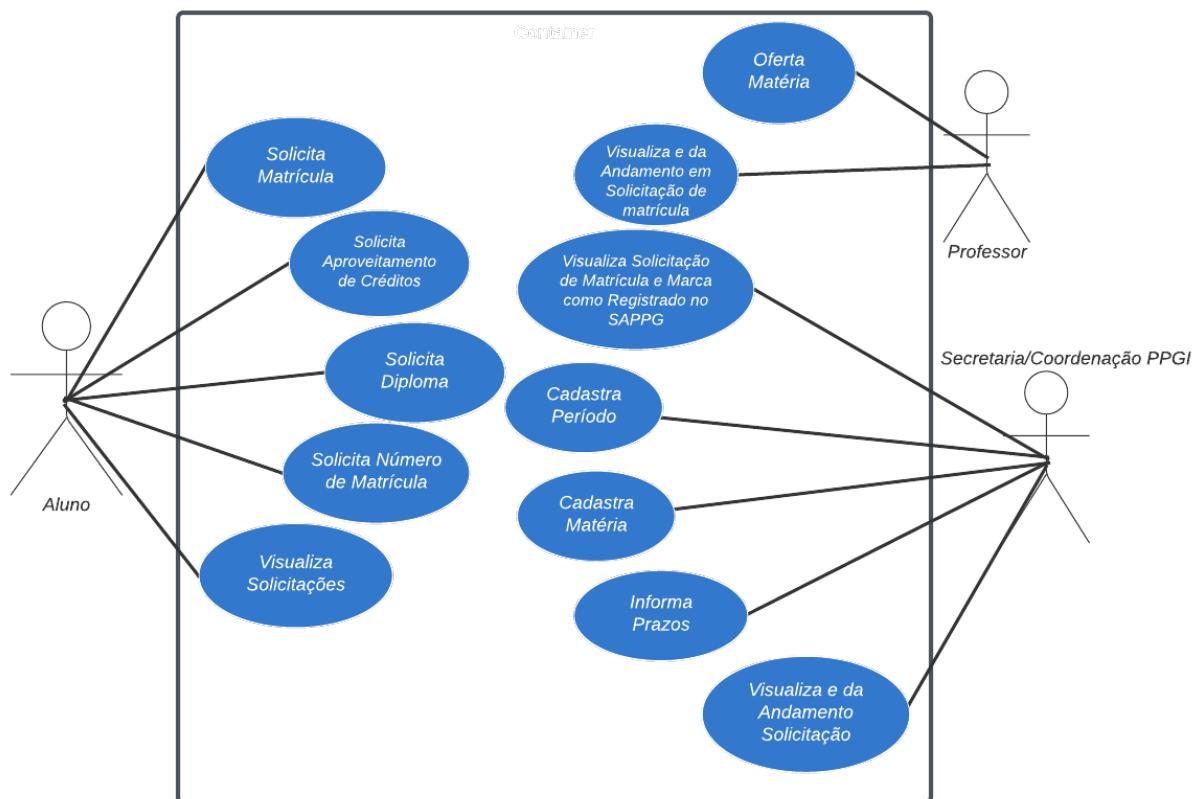


Figura 4 – Diagrama de Casos de Uso do subsistema Gestão do PPG.

O caso de uso **Solicita Matrícula** é a solicitação de matrícula em disciplinas, que posteriormente será aprovada ou reprovada por um professor/orientador.

O caso de uso **Solicita Aproveitamento de Créditos** é a solicitação de aproveitamento de créditos de alguma(s) disciplina(s) específica(s).

O caso de uso **Solicita Diploma** é a solicitação de diploma de conclusão de curso.

O caso de uso **Solicita Número de Matrícula** é referente ao aluno realizar a solicitação do número de matrícula, pode acontecer casos em que o aluno ainda não o tenha.

O caso de uso **Visualiza Solicitação** é a possibilidade do aluno conseguir ver as suas solicitações de tipos diferentes em uma mesma tela.

O caso de uso **Oferta Matéria** é a realização de oferta de matéria por um determinado Professor.

O caso de uso **Visualiza e da Andamento em Solicitação de Matrícula** é o acompanhamento do Professor/Orientador das solicitações de matrículas dos alunos.

O caso de uso **Visualiza Solicitação de Matrícula e Marca como Registrado no SAPPG** é o acompanhamento do Secretaria/Coordenação PPGI as solicitações de matrículas dos alunos, e o registro no SAPPG (sistema oficial da PRPPG/UFES) destas matrículas.

O caso de uso **Cadastra Período** é o gerenciamento de períodos no sistema que posteriormente serão utilizados para ofertas de disciplinas.

O caso de uso **Informa Prazos** é o gerenciamento de prazos no sistema que posteriormente serão utilizados como parametrização para solicitações.

O caso de uso **Visualiza e da Andamento em Solicitação** é referente ao acompanhamento e resposta das solicitações dos alunos pela Secretaria/Coordenação.

3.4 Identificação de Subsistemas

Para atender aos requisitos elencados de acordo com o escopo do projeto não houve necessidade de divisão em subsistemas, porém o *Módulo de controle acadêmico da Pós-Graduação do Marvin* é um subsistema do Marvin fazendo o uso de dois outros conforme pode ser visto na Figura 5, que ilustra os subsistemas e suas interdependências, enquanto a Tabela 3 apresenta uma breve descrição dos subsistemas já existentes e do subsistema criado.

Tabela 3 – Subsistemas identificados e suas interdependências (Elaboração própria).

Subsistema	Descrição
Núcleo	Módulo que gerencia o cadastro e autenticação de acadêmicos.
Administração	Módulo que gera o cadastro das unidades organizacionais da UFES (centros, departamentos, programas de pós-graduações, cursos).
Gestão do PPGI	Módulo de controle acadêmico da Pós-Graduação do Marvin.

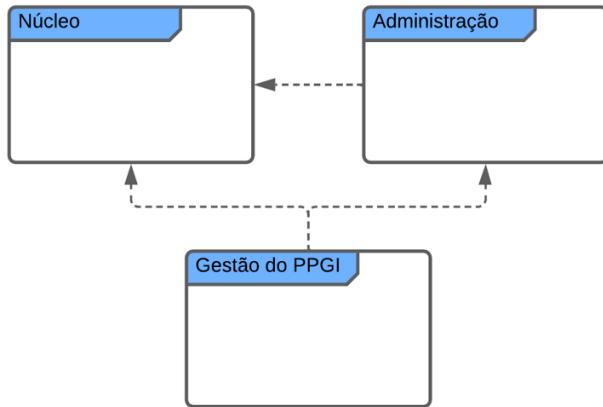


Figura 5 – Diagrama de Pacotes e os Subsistemas Identificados.

3.5 Diagrama de Classes

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 no apêndice.

A Figura 6 apresenta o diagrama de classes do subsistema Gestão do PPGI. O apêndice contém o dicionário de projeto, que apresenta as descrições das classes e atributos presentes neste diagrama.

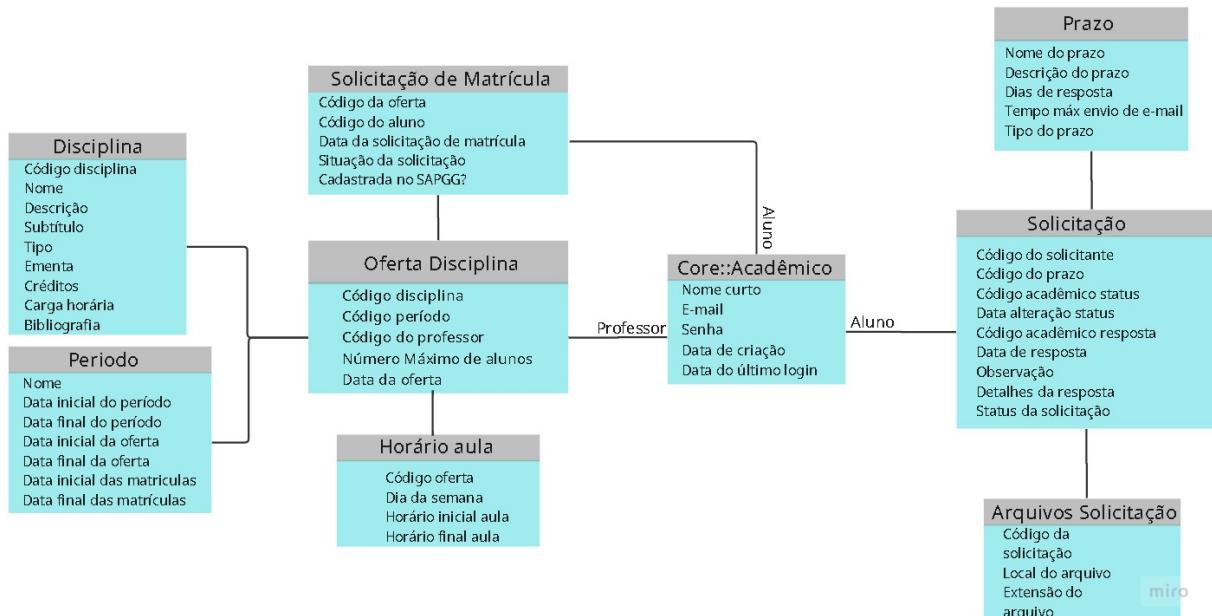


Figura 6 – Diagrama de Classes de Domínio.

A classe **Período** representa o período letivo de aulas, onde cada novo período deverá ser cadastrado no sistema, como por exemplo 2022/1.

A classe **Disciplina** representa as matérias que serão cadastradas no sistema, posteriormente elas poderão ser ofertadas para que os alunos solicitem a matrícula.

A classe **Oferta Disciplina** possui os dados referentes à oferta de disciplinas para os alunos. Ela tem ligações com a classe Disciplina e Acadêmico, onde será informada a matéria que está sendo oferecida e o Professor que está à oferecendo.

A classe **Horário Aula** simboliza a agenda da aula, bem como o seu horário de fim e início. Esta classe possui referência com a classe de oferta de disciplina.

A classe **Solicitação** tem por finalidade representar diferentes solicitações, sendo elas: solicitação de matrícula em uma determinada matéria ofertada, solicitação de diploma e de aproveitamento de créditos. Ela tem ligações com a classe Oferta Disciplina, Acadêmico e Fluxo Solicitação, onde será informada a oferta de matéria que o aluno está requisitando, qual o aluno que a está solicitando e o fluxo da solicitação para as solicitações de diploma e de aproveitamento de créditos.

A classe **Acadêmico** possui dados referentes aos usuários do sistema, tal classe já existe atualmente no Marvin, portanto não é representada em detalhes.

A classe **Pessoa** é contida de informações sobre uma pessoa, ela é utilizada como uma subclasse da classe Acadêmico.

A classe **Prazo** representa os prazos que serão cadastrados no sistema, tais prazos serão utilizados em determinadas demandas, onde cada uma quantidade máxima de dias até a sua conclusão.

4 Projeto do Sistema

Este capítulo tem como objetivo descrever o projeto arquitetural do *Módulo de controle acadêmico da Pós-Graduação 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; na Seção 4.3 temos a apresentação dos modelos FrameWeb; e por fim, na Seção 4.4 contém a demonstração do módulo após implementação.

4.1 Tecnologias e Ferramentas Utilizadas

As principais ferramentas e tecnologias utilizadas para o desenvolvimento deste trabalho foram:

- Java 13: linguagem de programação orientada a objetos e independente de plataforma, foi utilizada para desenvolver todo o *back-end* da aplicação;
- JSF 2.3: API para a construção de interfaces de usuários baseada em componentes para aplicações Web, que foi utilizada na criação das páginas Web e comunicação com as classes Java;
- JPA 2.2: API para persistência de dados por meio de mapeamento objeto/relacional, seu uso se deu na camada de persistência dos dados com o banco de dados;
- PrimeFaces 8.0: conjunto de componentes visuais JSF *open source*. Recurso utilizado para criar todo o *front-end* da aplicação;
- MySQL Server 8.0: sistema gerenciador de banco de dados relacional gratuito;
- WildFly 19: Servidor de Aplicações para Java EE, que foi utilizado para hospedagem da aplicação Web, dando acesso aos usuários via HTTP;
- Eclipse Java EE IDE for Web Developers: Ambiente de desenvolvimento (IDE) com suporte ao Java EE, que foi a IDE utilizada para criar o código da aplicação e realizar os devidos testes.

Também é possível citar diversas outras ferramentas usadas ao longo deste projeto, estas tecnologias podem ser vistas no apêndice contido neste trabalho.

4.2 Arquitetura de Software

O desenvolvimento deste módulo é baseado no padrão arquitetônico *Service Layer* (Camada de Serviço) (FOWLER, 2002), onde o software é dividido em três principais camadas conforme pode ser visto na Figura 7.

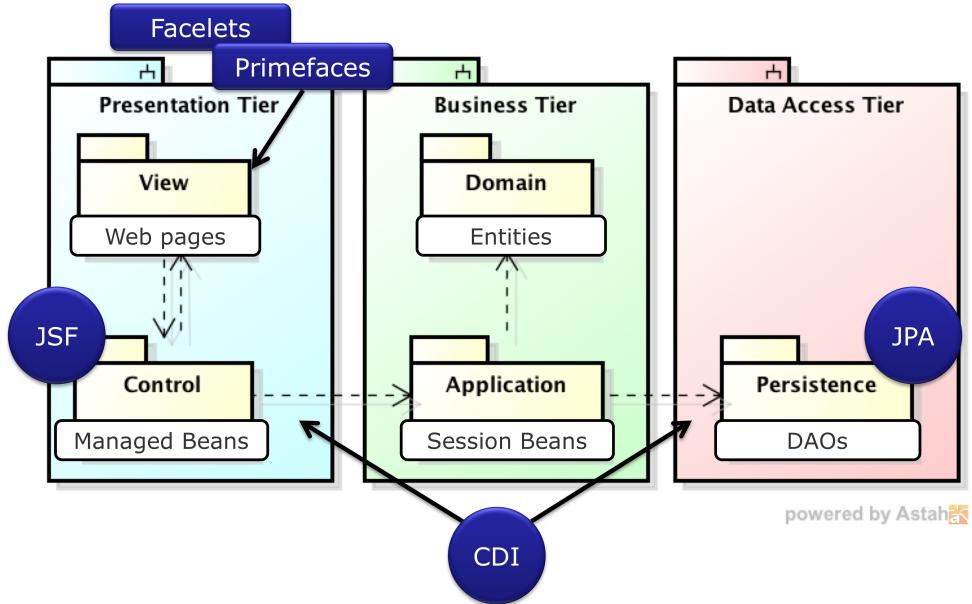


Figura 7 – Arquitetura de Software (SOUZA, 2020).

- Camada de negócio (*Business Tier*): ela é subdividida em duas camadas: lógica de aplicação (*Application*) e lógica de Domínio (*Domain*). Neste ponto que são providas as funcionalidades de acordo com as regras de negócio do sistema;
- Camada de apresentação (*Presentation Tier*): análogo ao seu nome, esta faixa é referente a criação da interface onde o usuário irá interagir com o sistema, nela estão presentes duas camadas provenientes da arquitetura MVC, sendo elas a visão (*View*) e controle (*Control*);
- Camada de acesso a dados (*Data Access Tier*): aqui são realizados os tratamentos interligados à persistência de informações no banco de dados.

No *Módulo de controle acadêmico da Pós-Graduação do Marvin*, os pacotes foram agrupados a partir do pacote base do Marvin juntamente com as camadas da arquitetura, como pode-se observar na Figura 8. A pasta visão (*view*) que é exibida na camada de apresentação foi alterada por pastas que contêm o nome do subsistema criado, dentro destas pastas contém páginas Web com a extensão XHTML, conforme presente na Figura 9.

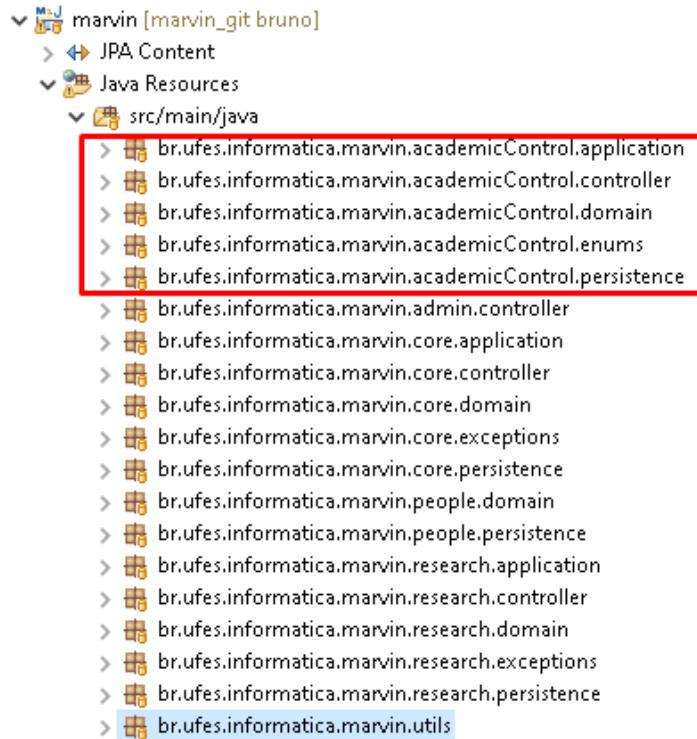


Figura 8 – Pacotes do Módulo de Controle Acadêmico

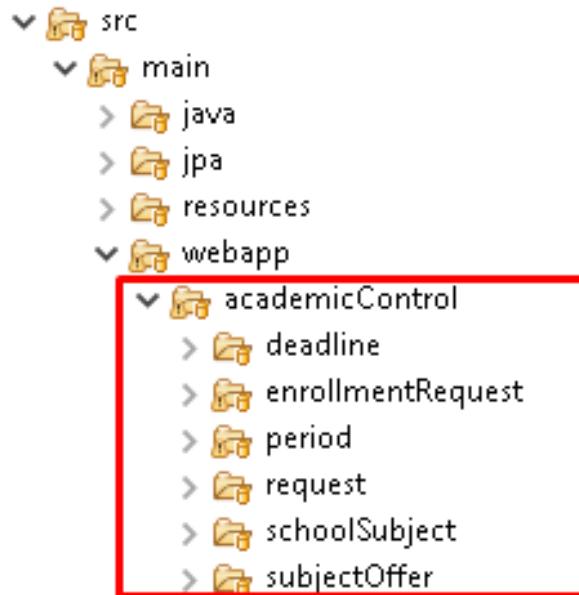


Figura 9 – Páginas Web do Módulo de Controle Acadêmico

4.3 Modelagem FrameWeb

Módulo de controle acadêmico da Pós-Graduação 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

4.3.1 Camada de Negócio

A camada de negócio é subdividida em duas camadas: Lógica de aplicação e lógica de domínio. A Figura 10 demonstra as entidades que compõe o domínio deste projeto.

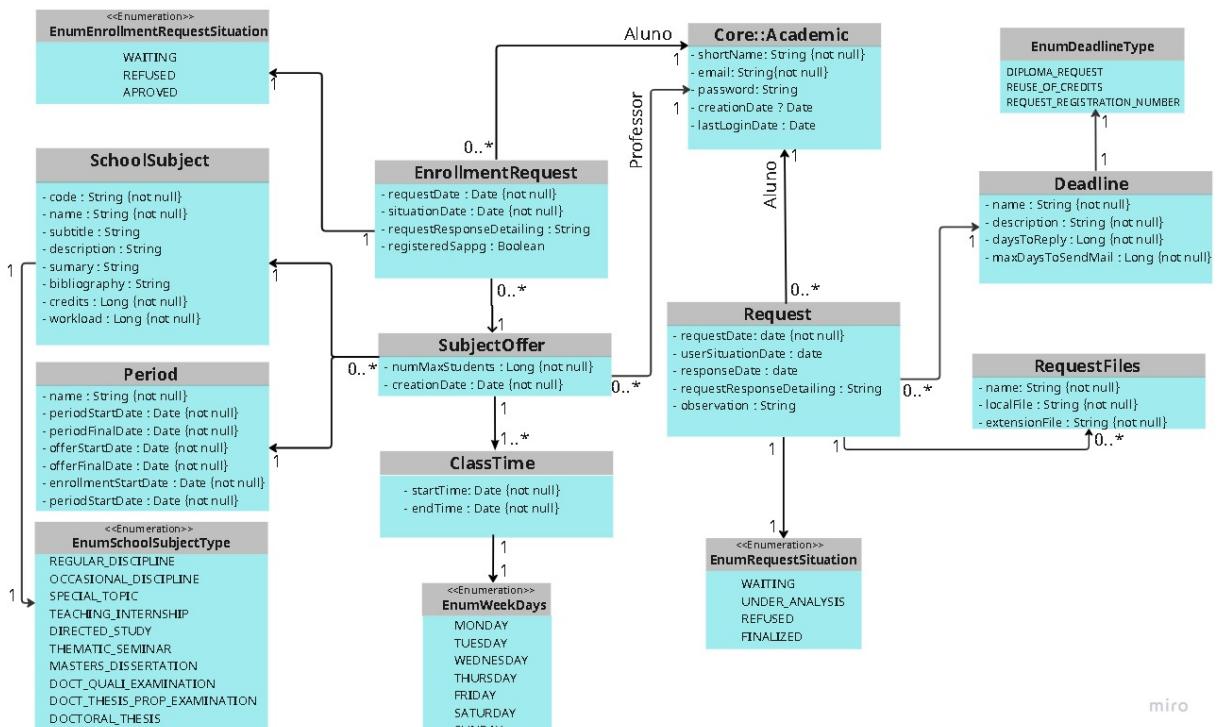


Figura 10 – Modelo de Entidades.

O pacote domínio que compõe a camada de negócio contém todas as classes de domínio do projeto, todas essas classes estendem a classe *PersistentObjectSupport*, do utilitário JButler,¹ já utilizado anteriormente pelo Marvin.

Já no pacote referente à aplicação, todas as classes estendem a classe *CrudServiceBean* que contém os métodos de CRUD já implementados. O modelo de aplicação do *Módulo de controle acadêmico da Pós-Graduação do Marvin* pode ser visto na Figura 11.

¹ <<https://gitlab.labes.inf.ufes.br/labes/jbutler>>

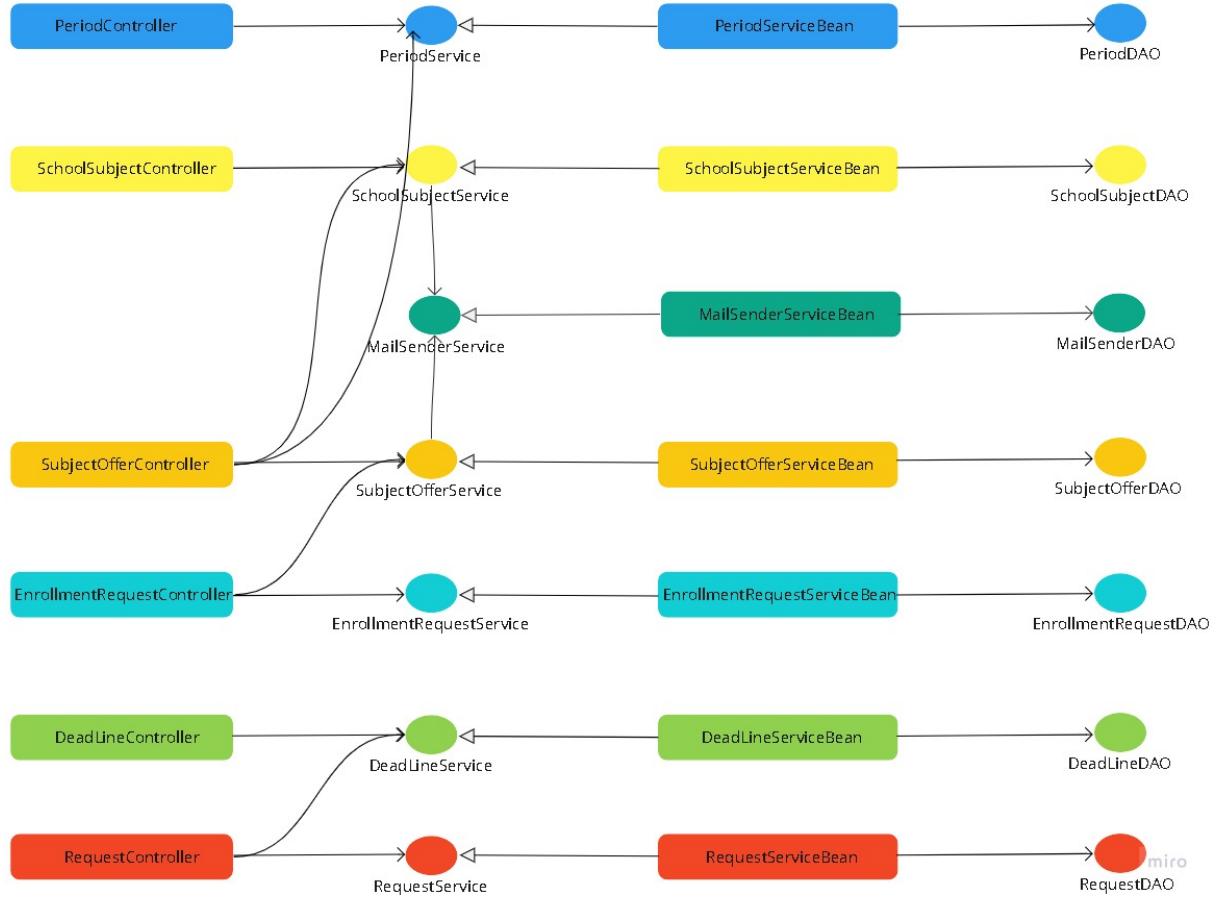


Figura 11 – Modelo de Aplicação.

4.3.2 Camada de Acesso a Dados

Na camada de acesso a dados são realizados os tratamentos ligados à persistência de informações no banco de dados. Todas as classes presentes nesta camada estendem a classe *BaseDAO* do JButler, que contém todas as funcionalidades básicas de acesso e persistência de dados. O modelo de persistência do subsistema Gestão do PPGI é representado pela Figura 12. Neste modelo estão contidas as interfaces DAO que foram criadas no trabalho, elas são responsáveis pelos métodos de persistência da camada de domínio do sistema. Cada classe existente no nível de domínio necessita de um DAO para que os dados sejam persistidos ou consultados no banco de dados.

4.3.3 Camada de Apresentação

Este nível é referente à criação das interfaces por meio das quais o usuário irá interagir com o sistema, nela estão presentes duas camadas provenientes da arquitetura MVC, sendo elas a visão (*View*) e controle (*Control*) conforme já apresentado na Figura 7.

As telas referentes ao cadastro de matérias, cadastro de prazos, cadastro de períodos e oferta de disciplinas são telas simples que contêm apenas funções de CRUD, podendo ser representadas pela Figura 13.

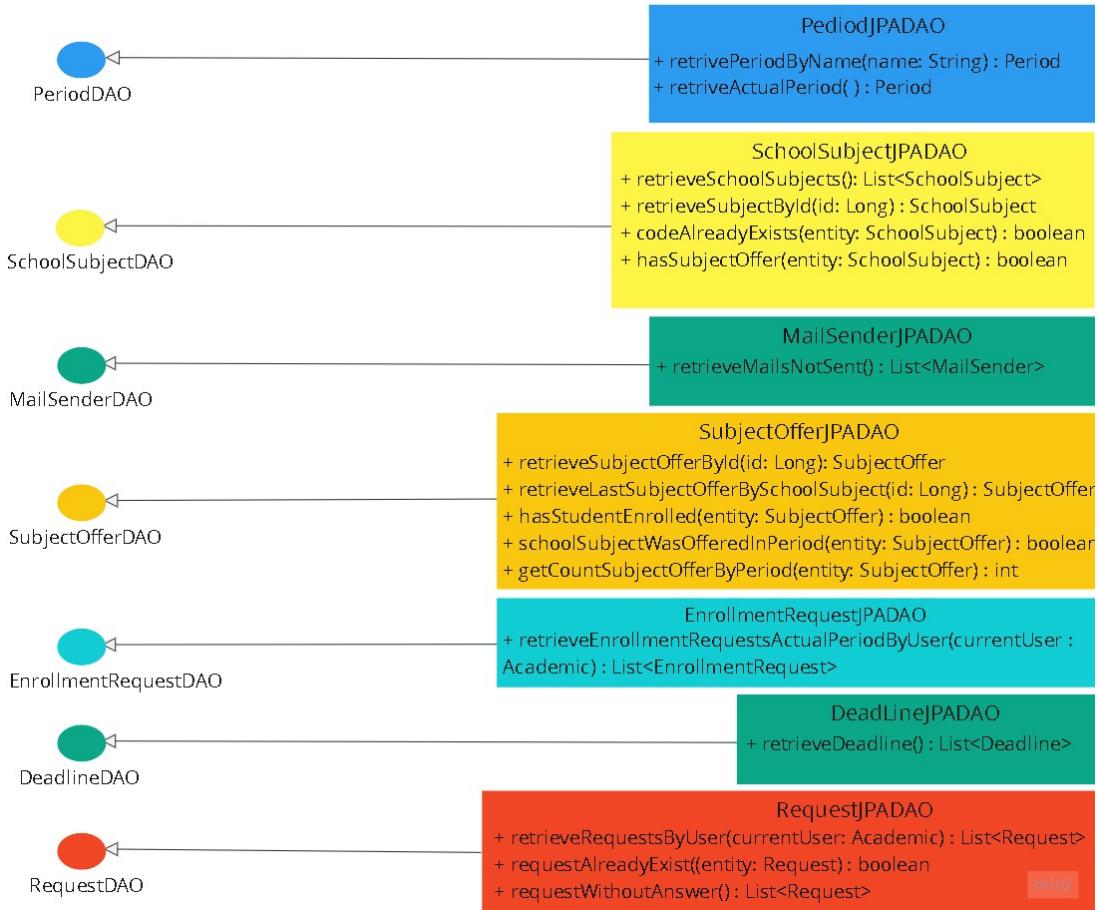


Figura 12 – Modelo de Persistência do subsistema Gestão do PPGI.

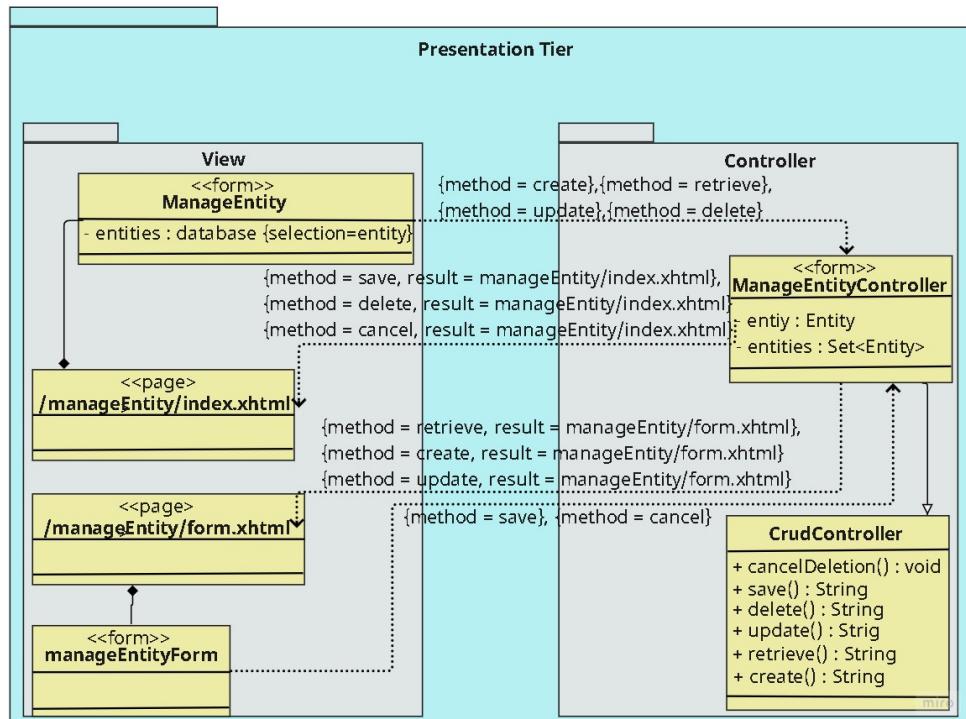


Figura 13 – Modelo de Navegação para as telas de CRUD.

Para as duas telas de solicitação (solicitações diversas e solicitação de matrícula) além das funções padrões de CRUD existem outras funcionalidades. A representação dessas outras funções podem ser vistas nas figuras 14 e 15.

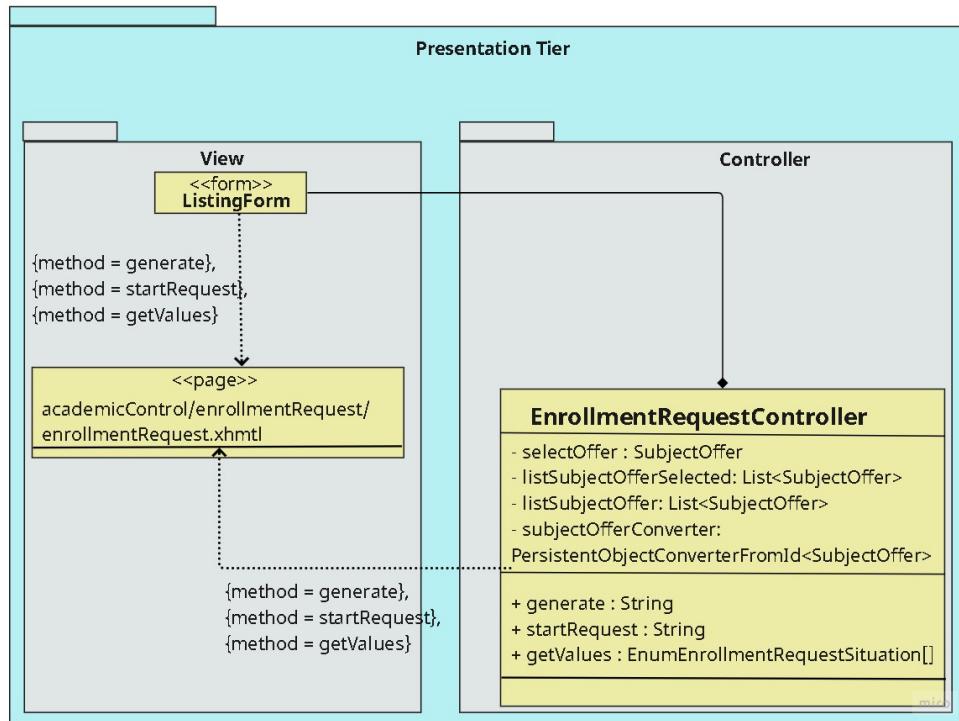


Figura 14 – Modelo de Navegação para a tela de solicitação de matrícula.

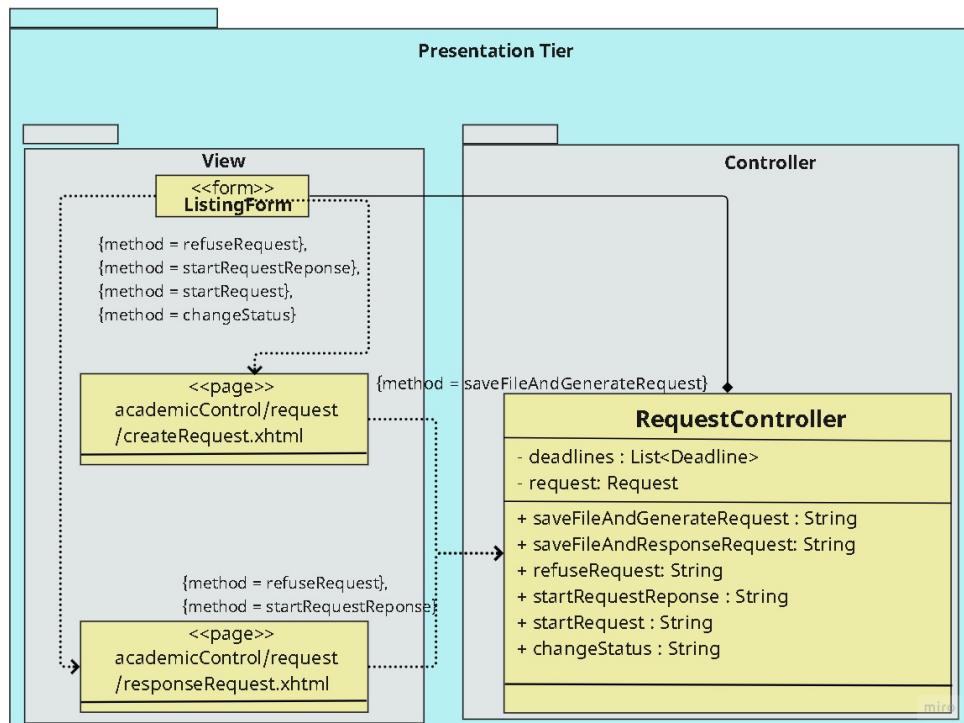


Figura 15 – Modelo de Navegação para a tela de solicitações diversas.

4.4 Apresentação

Esta seção apresenta as funcionalidades do *Módulo de controle acadêmico da Pós-Graduação do Marvin* do Marvin por meio de capturas de tela demonstrando o fluxo a ser seguido pelos usuários.

Na Figura 16 é possível ver a tela de *Login* do sistema, esta funcionalidade já estava presente no projeto e não houve necessidade de ser implementada neste trabalho. Para entrar no sistema é necessário ter um cadastro, caso não o tenha existe a possibilidade de criar uma conta clicando na função *I want to create an account*. Com o cadastro criado basta informar o e-mail e a senha para entrar no sistema.

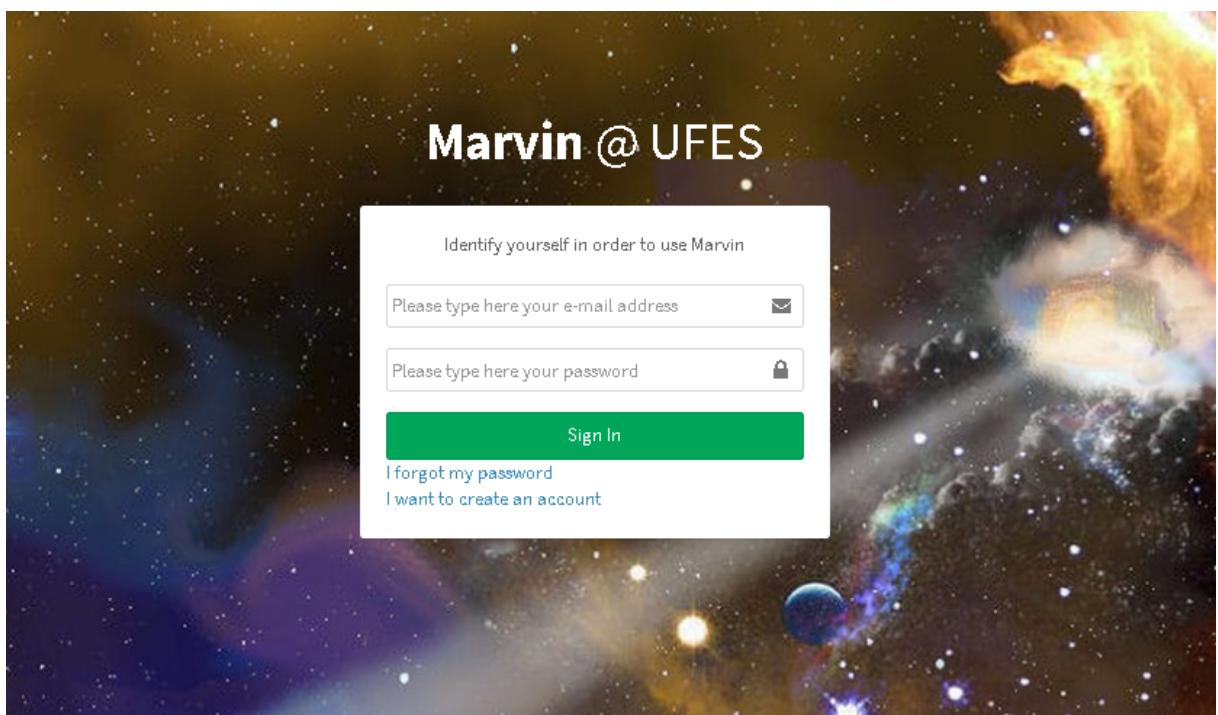


Figura 16 – Tela de *Login*

Após entrar no sistema é exibida a tela inicial. A visualização das funcionalidades são distintas de acordo com o papel do usuário, a Figura 17 é referente ao que é exibido para um usuário com permissão de Administrador do sistema. Nesta tela são visíveis as funcionalidades do *software*, sendo elas:

- Gerenciar Acadêmicos (*Manage Academics*);
- Importar Currículos Lattes (*Upload Lattes CV*);
- Gerar Bibliografia (*Generate Bibliography*);
- Gerenciar Prazos (*Manage Deadlines*);
- Gerenciar Disciplinas (*Manage School Subjects*);

- Gerenciar Períodos (*Manage Periods*);
- Gerenciar oferta de Disciplinas (*Manage Subject Offers*);
- Gerenciar solicitações de matrículas (*Manage Enrollment Requests*);
- Gerenciar requisições (*Manage Requests*).



Figura 17 – Tela inicial

As funcionalidades de Gerência de acadêmicos, Importar Currículos Lattes e Gerar Bibliografia já estavam presentes no Marvin e não foram partes do escopo de desenvolvimento deste trabalho. Neste módulo foi desenvolvido as demais funcionalidades que serão descritas nos próximas seções.

4.4.1 *Manage Deadlines*

A Figura 18 mostra a tela de listagem dos prazos cadastrados no sistema. Esta funcionalidade está visível apenas para o Administrador do sistema ou para a Secretaria/- Coordenação do PPGI. Nela são parametrizados os prazos de resposta das requisições dos alunos. Ao clicar no botão *New* pode-se criar um novo prazo conforme a Figura 19.

Name	Max Days to Reply	Max Days to Send email
Solicitação de Diploma	365	30
Solicitação de aproveitamento de créditos	15	5
Solicitação de numero de Matricula	7	3

1 2 3 4 5 6 7 8 9 10

New View Modify Delete

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

Figura 18 – Tela de Gerenciamento de Prazos (Listagem)

Deadline registration form

Name*:	Name of Deadline
Type of Deadline*:	DIPLOMA_REQUEST
Max Days to Reply*:	Inform the maximum turnaround time of requests
Max Days to Send email*:	Inform the maximum days to send email
Description*:	Describe what the deadline is for

Cancel Save

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

Figura 19 – Tela de Gerenciamento de Prazos (Inserção)

A Figura 20 demonstra uma das validações presentes nesta tela, que é uma validação de alteração que não permite a mudança no tipo do prazo. A Figura 21 demonstra como é tratada uma validação de integridade, para não permitir a deleção do registro caso ele esteja vinculado a alguma requisição.

The screenshot shows a 'Deadline registration form' with the following fields:

- Name*: Solicitação de Diploma
- Type of Deadline*: REQUEST_REGISTRATION_NUMBER
- Max Days to Reply*: 365
- Max Days to Send email*: 30
- Description*: Solicitação de Diploma

A red error message at the top states: "It is not possible to change the type of deadline." The bottom right of the form has 'Cancel' and 'Save' buttons.

Figura 20 – Tela de Gerenciamento de Prazos (Validação de alteração de tipo)

The screenshot shows a table of deadlines:

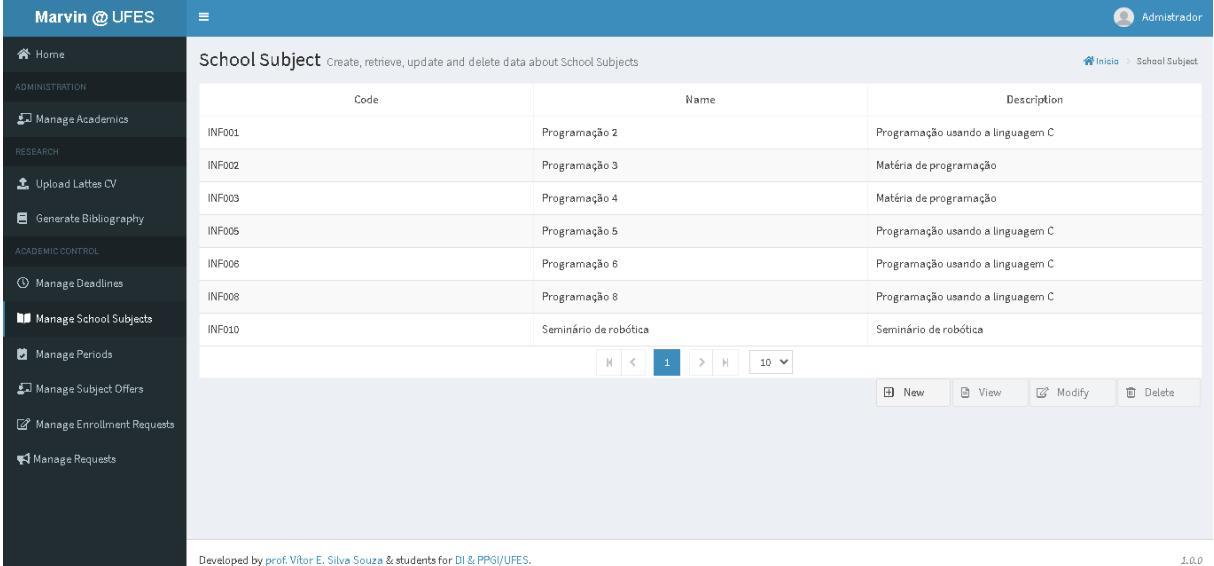
Name	Max Days to Reply	Max Days to Send email
Solicitação de Diploma	365	30
Solicitação de aproveitamento de créditos	15	5
Solicitação de numero de Matricula	7	3

An error message at the top states: "It is not possible delete record because the deadline is linked in some Request." The bottom right of the table has 'New', 'View', 'Modify', and 'Delete' buttons.

Figura 21 – Tela de Gerenciamento de Prazos (Validação de deleção)

4.4.2 Manage School Subjects

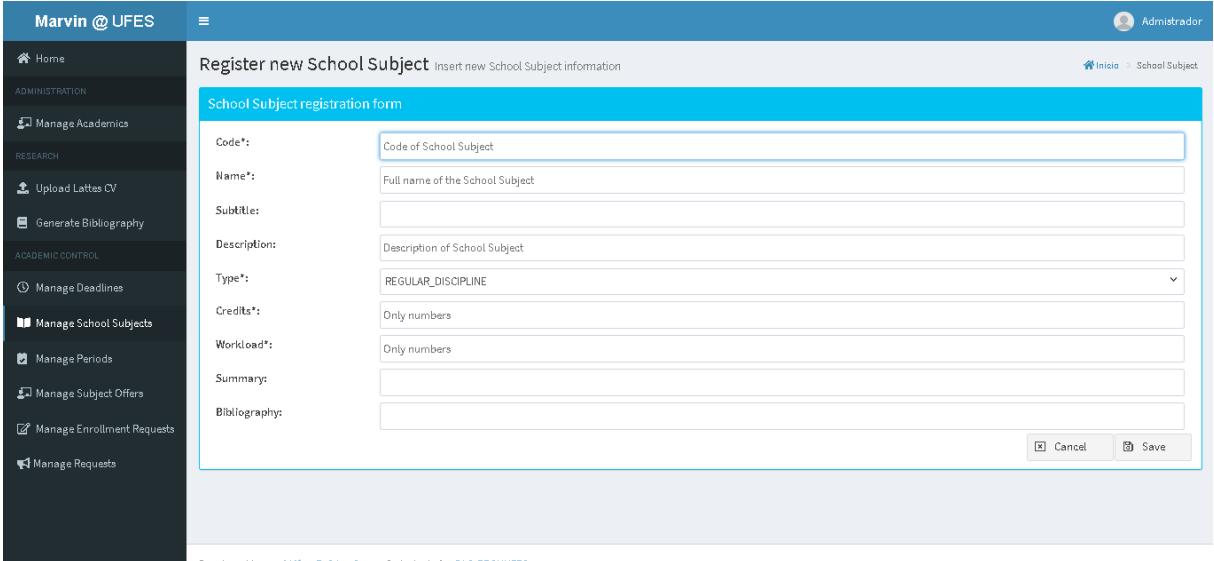
A Figura 22 mostra a tela de listagem das disciplinas cadastradas no sistema. Esta funcionalidade está visível para o Administrador do sistema, Secretaria/Coordenação PPGI e Professores. Nela é realizado o gerenciamento das disciplinas que serão ofertadas. Ao clicar no botão *New* é aberta a tela de inserção para cadastrar uma nova disciplina conforme a Figura 23.



Code	Name	Description
INF001	Programação 2	Programação usando a linguagem C
INF002	Programação 3	Materia de programação
INF003	Programação 4	Materia de programação
INF005	Programação 5	Programação usando a linguagem C
INF006	Programação 6	Programação usando a linguagem C
INF008	Programação 8	Programação usando a linguagem C
INF010	Seminário de robótica	Seminário de robótica

Developed by prof. Vitor E. Silva Souza & students for DI & PPG/Ufes. 1.0.0

Figura 22 – Tela de Gerenciamento de Disciplinas (Listagem)



Register new School Subject Insert new School Subject information

School Subject registration form

Code*:	Code of School Subject
Name*:	Full name of the School Subject
Subtitle:	
Description:	Description of School Subject
Type*:	REGULAR_DISCIPLINE
Credits*:	Only numbers
Workload*:	Only numbers
Summary:	
Bibliography:	

Developed by prof. Vitor E. Silva Souza & students for DI & PPG/Ufes. 1.0.0

Figura 23 – Tela de Gerenciamento de Disciplinas (Inserção)

Além das validações de integridade também existem duas outras validações, sendo que elas podem ser acionadas na inserção ou alteração do registro, onde não é possível ter mais de uma matéria com o mesmo código como demonstrado na Figura 24 ou caso a matéria for do tipo Tópico Especial ou Seminário Temático deve ser informados os campos de subtítulo, ementa e bibliografia, como é apresentado na Figura 25.

Figura 24 – Tela de Gerenciamento de Disciplinas (Validação de códigos iguais)

Figura 25 – Tela de Gerenciamento de Disciplinas (Validação de campos de acordo com o tipo)

4.4.3 Manage Periods

A Figura 26 exibe a tela de listagem dos períodos cadastrados no Marvin. Este recurso está visível para o Administrador do sistema e Secretaria/Coordenação PPGI, nela é possível gerenciar os períodos letivos. Ao clicar no botão *New* é redirecionado para a tela de cadastro de uma nova disciplina conforme a Figura 27.

Name	Offer Start Date	Offer Final Date	Enrollment Start Date	Enrollment Final Date	Period Start Date	Period Final Date
2023/1	2023-02-05	2023-02-10	2023-02-10	2023-02-15	2023-03-01	2023-07-31

Figura 26 – Tela de Gerenciamento de Períodos (Listagem)

Period registration form

Name*:	Ex: 2021/1 or 2021-1
Offer Start Date*:	(dd/mm/aaaa)
Offer Final Date*:	(dd/mm/aaaa)
Enrollment Start Date*:	(dd/mm/aaaa)
Enrollment Final Date*:	(dd/mm/aaaa)
Period Start Date*:	(dd/mm/aaaa)
Period Final Date*:	(dd/mm/aaaa)

Figura 27 – Tela de Gerenciamento de Períodos (Inserção)

Existem duas principais validações nesta tela além da validação de integridade (não permitir excluir caso esteja vinculado a algum outro registro) que é a validação da ordem das datas (ex: não ter a data de solicitação de matrícula anterior a data de oferta de disciplina), conforme visto na Figura 28, e a verificação da existência de período com o mesmo nome do registro a ser persistido, como pode ser visto na Figura 29.

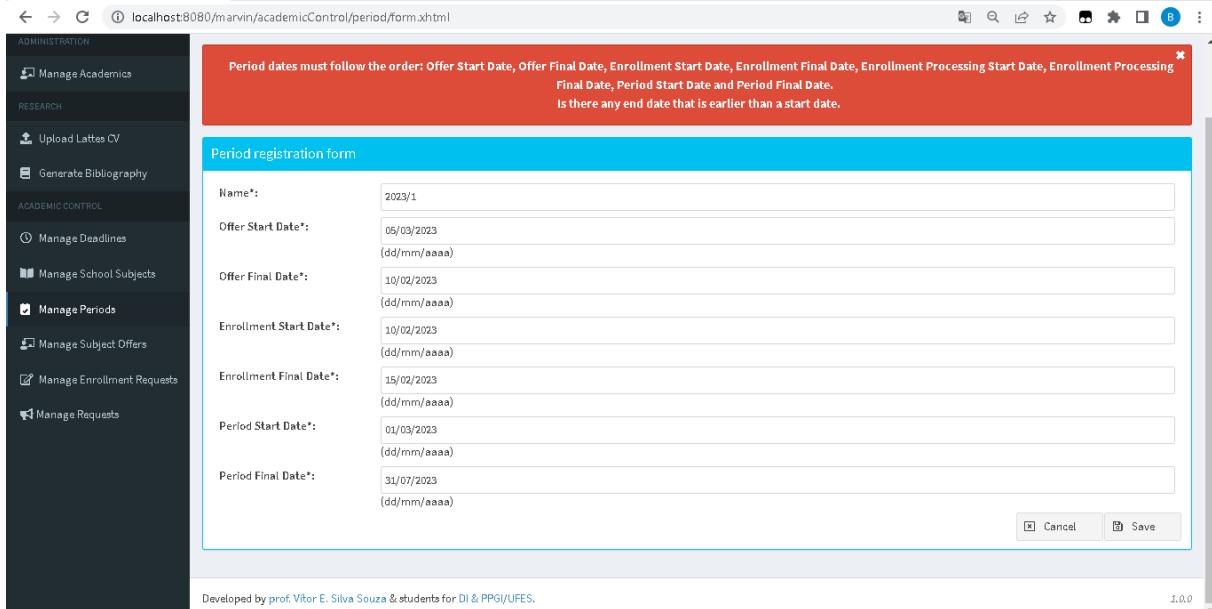


Figura 28 – Tela de Gerenciamento de Períodos (Validação das datas)

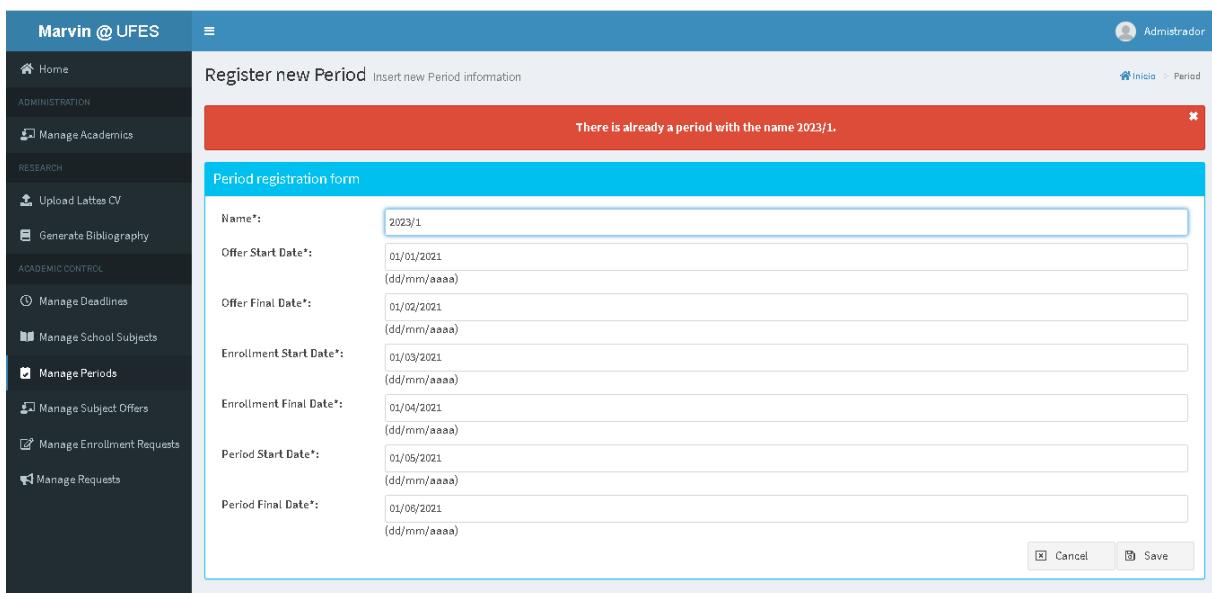


Figura 29 – Tela de Gerenciamento de Períodos (Validação do nome do período)

4.4.4 Manage Subject Offers

A Figura 30 é referente a tela de listagem das disciplinas que foram ofertadas no Marvin, na qual é realizado o gerenciamento de ofertas das matérias. Esta funcionalidade está visível para o Administrador, Secretaria/Coordenação PPGI ou Professor. Ao clicar no botão *New*, o usuário é redirecionado para a tela de inserção de uma oferta conforme as figuras 31 e 32. Esta funcionalidade do sistema tem um facilitador para quando o usuário for ofertar uma disciplina, caso ela tenha sido disponibilizada para matrícula em períodos passados a inserção criará o horário das aulas da matéria automaticamente com base em registros de oferta anteriores.

The screenshot shows a web application interface for managing academic subjects. On the left is a dark sidebar with navigation links for Home, Administration (Manage Academics), Research (Upload Lattes CV, Generate Bibliography), Academic Control (Manage Deadlines, Manage School Subjects, Manage Periods, Manage Subject Offers, Manage Enrollment Requests, Manage Requests). The main content area has a blue header "Subject Offer Create, retrieve, update and delete data about Subject Offer". Below it is a table with columns: Period, School Subject, Professor Name, Max Number of Students, Offer Start Date, and Offer Final Date. The table contains several rows of data, with the first row highlighted in blue. At the bottom of the table are navigation buttons (first, previous, next, last) and links for New, View, Modify, and Delete.

Period	School Subject	Professor Name	Max Number of Students	Offer Start Date	Offer Final Date
2023/1	Programação 2	Vitor Souza	40	2023-02-05	2023-02-10
2023/1	Programação 3	Vitor Souza	40	2023-02-05	2023-02-10
2023/1	Programação 4	Vitor Souza	40	2023-02-05	2023-02-10
2023/1	Programação 5	Vitor Souza	40	2023-02-05	2023-02-10
2023/1	Programação 6	Vitor Souza	40	2023-02-05	2023-02-10
2023/1	Programação 8	Vitor Souza	40	2023-02-05	2023-02-10

Developed by prof. Vitor E. Silva Souza & students for DI & PPGI/UFES. 1.0.0

Figura 30 – Tela de Gerenciamento de Disciplina (Listagem)

The screenshot shows a registration form for a new subject offer. The sidebar and header are identical to Figure 30. The main content area has a blue header "Register new Subject Offer Insert new Subject Offer information". Below it is a form titled "Subject Offer registration form" with fields: Period* (2023/1), School Subject* (Programação 2), Professor Name* (Vitor Souza), and Max Number of Students* (40). At the bottom right are buttons for Cancel, Save, and Schedules.

Developed by prof. Vitor E. Silva Souza & students for DI & PPGI/UFES. 1.0.0

Figura 31 – Tela de Gerenciamento de oferta das Disciplinas (Inserção)

The screenshot shows the Marvin @ UFES web application interface. The left sidebar contains a navigation menu with categories like Home, Administration, Research, Academic Control, and Requests. The main content area is titled 'Class schedule' and 'Class schedule creation'. It features a form to 'Generate class schedule' for 'Programação 2' on 'THURSDAY' from '09:00' to '11:00'. Below this, a table lists scheduled classes for Monday and Tuesday. At the bottom, there are links for 'Back', 'Clean', and 'Save'.

Class day	Start time	End time
Monday	09:00	11:00
Tuesday	09:00	11:00

Developed by prof. Vitor E. Silva Souza & students for DI & PPG/UFES. 1.0.0

Figura 32 – Tela de Gerenciamento de oferta das Disciplinas (Horário das aulas)

Nesta tela existem diversas validações de inserção, alteração e deleção, sendo elas: Não possibilidade de ofertar a mesma disciplina mais de uma vez em um mesmo período, não existir a possibilidade de deleção caso o período de solicitação de matrícula tenha inicializado, não possibilidade de alteração da oferta de disciplina caso a data de solicitação de matrícula seja menor que a data atual.

4.4.5 Manage Enrollment Requests

Esta função do sistema pode ser utilizada por um Administrador, Professor, Secretaria/Coordenação PPGI, ou por um aluno, porém a visualização da tela diverge de acordo com papel do usuário, exibindo apenas informações pertinentes para o usuário em questão. Um aluno consegue ter acesso apenas às suas próprias solicitações de matrícula conforme visto na Figura 33 e não consegue visualizar funções não necessárias à ele.

Ao clicar no botão *New* o sistema redireciona para a tela de solicitação de matrícula como visto na Figura 34 onde são informadas as disciplinas que serão solicitadas a matrícula.

Period	School Subject	Professor	Enrollment Request Situation
2023/1	Programação 2	Vitor Souza	Waiting
2023/1	Programação 3	Vitor Souza	Waiting
2023/1	Programação 4	Vitor Souza	Waiting

Figura 33 – Tela de Gerenciamento de Solicitação de Matrícula visão do aluno (Listagem)

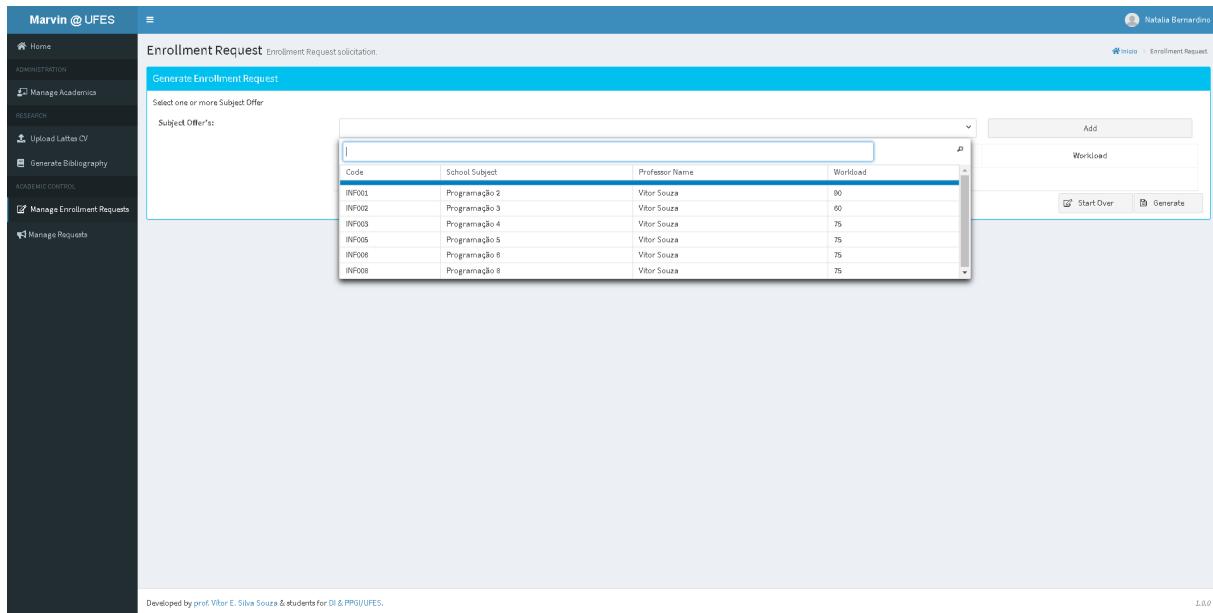


Figura 34 – Tela de Gerenciamento de Solicitação de Matrícula visão do aluno

Na visão do Administrador, Professor ou Secretaria/Coordenação PPGI são exibidas todas as solicitações de matrícula cadastradas no sistema, conforme mostra a Figura 35. Também é possível editar o registro para aprovar a matrícula do aluno ou rejeitá-la. Existe a possibilidade de marcar a matrícula como cadastrada no SAPPG clicando no botão SAPPG ou por meio de edição do registro como visto na Figura 36.

Enrollment Request					
Enrollment Request					
Student	Period	School Subject	Professor	Enrollment Request Situation	Registered on SAPPG?
Rafael	2023/1	Programação 2	Vitor Souza	Approved	<input checked="" type="checkbox"/> SAPPG?
Rafael	2023/1	Programação 3	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?
Rafael	2023/1	Programação 4	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?
Rafael	2023/1	Programação 5	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?
Rafael	2023/1	Programação 6	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?
Lucas	2023/1	Programação 2	Vitor Souza	Approved	<input checked="" type="checkbox"/> SAPPG?
Lucas	2023/1	Programação 3	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?
Lucas	2023/1	Programação 4	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?
José	2023/1	Programação 6	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?
José	2023/1	Programação 5	Vitor Souza	Waiting	<input type="checkbox"/> SAPPG?

Figura 35 – Tela de Gerenciamento de Solicitação de Matrícula visão do Administrador (Listagem)

The screenshot shows the Marvin @ UFES administrative interface. On the left is a dark sidebar with various management options: Home, ADMINISTRATION (Manage Academics, Upload Lattes CV, Generate Bibliography), RESEARCH (Upload Lattes CV, Generate Bibliography), ACADEMIC CONTROL (Manage Deadlines, Manage School Subjects, Manage Periods, Manage Subject Offers, Manage Enrollment Requests, Manage Requests). The main area is titled 'Update Enrollment Request' with a sub-instruction 'Modify an existing Enrollment Request's information'. It contains a form with the following fields: 'Period*' (2023/1), 'School Subject' (INF002), 'Student' (Lucas), 'Professor' (Vitor Souza), 'Enrollment Request Situation' (set to 'WAITING'), and 'Registered on SAPPG?' (unchecked). At the bottom are 'Cancel' and 'Save' buttons. The footer indicates the application was developed by prof. Vitor E. Silva Souza & students for DI & PPGI/UFES.

Figura 36 – Tela de Gerenciamento de Solicitação de Matrícula visão do Administrador (Alteração)

4.4.6 Requests

A tela de solicitações que pode ser vista na Figura 37 é utilizada para fazer as solicitações de diploma, aproveitamento de créditos e solicitação do número de matrícula. Caso o usuário do sistema seja um aluno as funções de alteração de situação, revogar situação e responder solicitação não serão exibidas, e o aluno verá apenas as suas próprias solicitações em tela.

Quando a Secretaria/Coordenação PPGI alterar a situação de alguma solicitação será enviado um e-mail para o aluno que solicitou o requerimento, informando da atualização do estado da solicitação.

Os registros presentes nesta tela também são utilizados para realizar envios de e-mails quando uma solicitação estiver com o prazo perto de finalizar, onde a Secretaria/- Coordenação PPGI será alertada por e-mail, vide Figura 38, com o tempo restante para responder a solicitações dos alunos sem extrapolar o prazo máximo. O tempo considerado como gatilho de envio de e-mail e o prazo máximo são de acordo com a parametrização dos prazos das solicitações.

Nesta tela existem alguns comportamentos dinâmicos nos botões, a função de deleção fica desativada caso a situação do registro for diferente de aguardando e o botão de revogar fica desativado caso a solicitação esteja com a situação aguardando. Também existe uma validação impedindo de um aluno ter mais de uma solicitação do mesmo tipo com a situação igual aguardando ou em análise conforme pode ser visto na Figura 39.

The screenshot shows the Marvin @ UFES Request management interface. On the left, there is a sidebar with various menu items under categories like Home, Administration, Research, and Academic Control. The main area is titled "Request" and contains a table with columns: Requester, Type of Deadline, Situation, Request Date, and Response Date. There are 10 entries in the table, each representing a request from different users (e.g., Bruno Gama, Rafael, Lucas, José) with various statuses (Finalized, Under Analysis, Waiting) and dates ranging from 2023-02-03 to 2023-02-06.

Requester	Type of Deadline	Situation	Request Date	Response Date
Bruno Gama	Solicitação de Diploma	Finalized	2023-02-03 12:18:22.748	2023-02-03 15:10:47.473
Natalia Bernardino	Solicitação de Diploma	Finalized	2023-02-03 15:03:43.04	2023-02-03 21:29:48.696
Natalia Bernardino	Solicitação de aproveitamento de créditos	Waiting	2023-02-03 21:30:18.601	
Rafael	Solicitação de Diploma	Under Analysis	2023-02-06 10:38:04.563	
Rafael	Solicitação de aproveitamento de créditos	Waiting	2023-02-06 10:38:14.632	
Rafael	Solicitação de número de Matrícula	Waiting	2023-02-06 10:38:33.007	
Lucas	Solicitação de Diploma	Under Analysis	2023-02-06 10:41:51.168	
Lucas	Solicitação de aproveitamento de créditos	Waiting	2023-02-06 10:41:57.632	
José	Solicitação de Diploma	Under Analysis	2023-02-06 10:43:02.009	
José	Solicitação de aproveitamento de créditos	Waiting	2023-02-06 10:43:18.199	

Figura 37 – Tela de Gerenciamento de Requisição (Listagem)

The screenshot shows a search interface with a search bar containing "in:sent". Below the search bar is a toolbar with icons for back, forward, search, and other functions. The main content area is titled "Marvin - Solicitações com prazo perto de expirar" and displays a table of pending requests. The table has columns: Nome do Aluno, Tipo de solicitação, Situação da solicitação, and Tempo restante. The data includes multiple entries for Aluno 1 through Aluno 5, with various types of requests (Solicitação de número de matrícula, Aproveitamento de créditos, Solicitação de Diploma) and their respective statuses (Aguardando, Em Análise) and time remaining (0, 2, 5, 7, 9, 353, 356, 358, 360, 365).

Nome do Aluno	Tipo de solicitação	Situação da solicitação	Tempo restante
Aluno 1	Solicitação de número de matrícula	Aguardando	0
Aluno 2	Solicitação de número de matrícula	Em Análise	0
Aluno 3	Solicitação de número de matrícula	Em Análise	0
Aluno 3	Aproveitamento de créditos	Em Análise	2
Aluno 2	Aproveitamento de créditos	Em Análise	5
Aluno 1	Aproveitamento de créditos	Em Análise	7
Aluno 4	Aproveitamento de créditos	Aguardando	9
Aluno 3	Solicitação de Diploma	Aguardando	353
Aluno 2	Solicitação de Diploma	Aguardando	356
Aluno 1	Solicitação de Diploma	Aguardando	358
Aluno 4	Solicitação de Diploma	Aguardando	360
Aluno 5	Solicitação de Diploma	Em Análise	365

Figura 38 – Tela de Gerenciamento de Requisição (Alerta com o prazo restante da requisição)

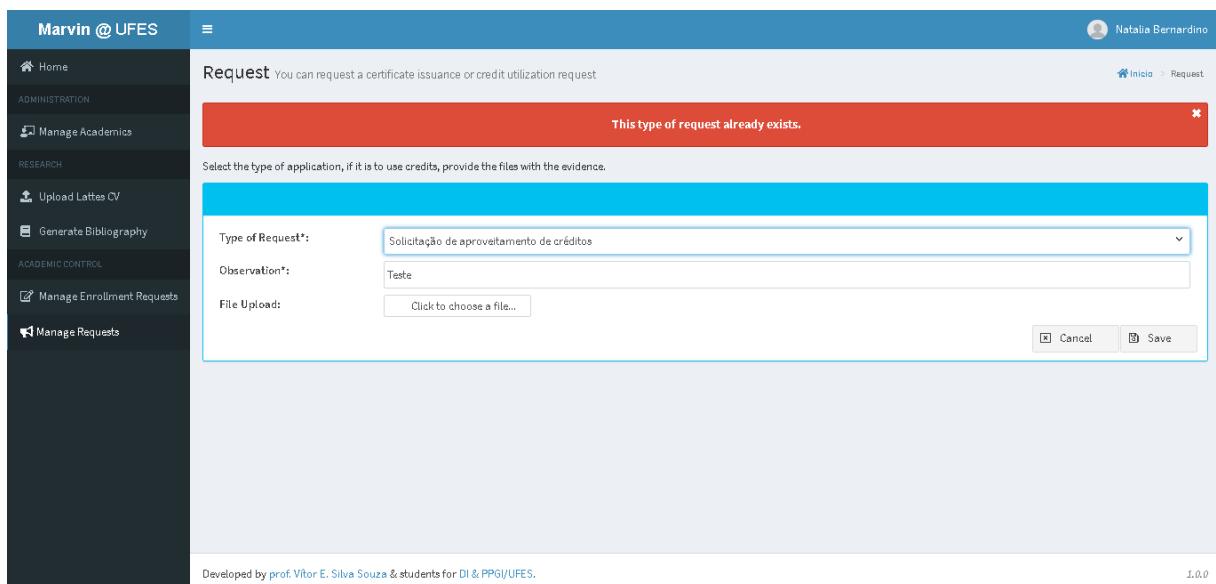


Figura 39 – Tela de Gerenciamento de Requisição (Validação de multiplas requisições com o mesmo tipo)

5 Conclusão

Este capítulo apresenta as principais contribuições, limitações, lições aprendidas durante o desenvolvimento do trabalho, dificuldades enfrentadas e perspectivas de trabalhos futuros.

5.1 Considerações Finais

A criação do *Módulo de controle acadêmico da Pós-Graduação do Marvin* é uma solução importante para a otimização do processo de gerenciamento das demandas acadêmicas do Programa de Pós-Graduação em Informática da UFES. O objetivo é reduzir o trabalho manual dos funcionários da secretaria, melhorar a comunicação com os alunos e diminuir a sobrecarga de e-mails. Embora o escopo deste projeto não inclua a integração direta com os sistemas da UFES, ele representa um importante passo em direção a uma gestão mais eficiente e efetiva no PPGI.

Considerando os objetivos apresentados no Capítulo 1, é possível afirmar que este trabalho conseguiu alcançar seu objetivo geral de desenvolver um sistema *Web* que centralize as necessidades do PPGI. Ao cumprir os objetivos específicos, como o levantamento e documentação dos requisitos do módulo, projeto, documentação da arquitetura integrada, a implementação e testes do código-fonte, é possível afirmar que o sistema é coerente e atende às necessidades do programa. É importante destacar que este trabalho não apenas otimiza os processos, mas também contribui para a modernização e agilidade das operações, tornando-as mais eficientes e fáceis de serem realizadas.

Apesar de ter seguido os passos presentes no Capítulo 1, surgiram algumas dificuldades para a conclusão deste projeto, sendo uma das principais o levantamento e documentação de requisitos, devido a falta de conhecimento dos processos, valores possíveis em alguns campos e validações necessárias. Nesta etapa, era importante atender as reais necessidades e expectativas dos usuários, onde qualquer erro nesta fase poderia gerar um imenso retrabalho em processos futuros do desenvolvimento.

Outro problema encontrado durante o desenvolvimento deste trabalho foi a falta de familiaridade com o *PrimeFaces*. Embora a ferramenta facilite na estilização das páginas, foram encontrados empecilhos ao tentar construir alguma tela ou funcionalidade que fugisse do *layout* padrão.

No decorrer deste trabalho foram encontrados inúmeros desafios, contudo estes foram superados com sucesso devido ao conhecimento adquirido ao longo do curso de Engenharia da Computação, bem como os conteúdos aprendidos nas disciplinas de Engenharia

de *Software*, Programação Orientada a Objetos e Banco de Dados, foram primordiais para a conclusão deste projeto.

5.2 Trabalhos Futuros

Em termos de trabalhos futuros para melhorar o projeto, é importante considerar as possibilidades de expansão e melhoria do *Módulo de controle acadêmico da Pós-Graduação do Marvin*. Algumas ações que poderiam ser consideradas incluem:

- Integração com outras ferramentas acadêmicas, como o Sistema Acadêmico de Pesquisa e Pós-Graduação (SAPPG), sistemas de gerenciamento de notas e calendários acadêmicos, para fornecer uma visão abrangente e atualizada de todas as informações relacionadas à pós-graduação e reduzir a duplicação de informações;
- Ampliação da funcionalidade do módulo, para atender ainda mais às necessidades dos usuários, por exemplo: a possibilidade de agendamento de reuniões, de geração de relatórios personalizados e alertas de vencimento de prazos em tela para melhorar a eficiência e a eficácia do sistema;
- Melhoria na funcionalidade de importação e exportação de arquivos, permitindo ao usuário inserir e excluir um ou vários anexos;
- Criação de filtros na tela e possibilitar ao usuário que ordene as colunas exibidas da forma que preferir, melhorando a usualidade do sistema permitindo que o usuário veja em tela somente a informação que ele realmente deseja;
- Desenvolvimento de uma nova funcionalidade para a oferta de disciplina, podendo ser realizada uma oferta em massa de várias matérias distintas de uma só vez. Esta função facilitaria a usualidade para caso um coordenador de curso deseje ofertar várias matérias para Professores distintos;
- Criar novas rotinas de envio de e-mail automático, por exemplo: De estatísticas com a quantidade de solicitações que ainda estão sem resposta e quantos e quais alunos estão sem número de matrícula cadastrado, auxiliando para que não haja necessidade do aluno fazer a requisição do seu número de matrícula;
- Adição de funcionalidades para a geração automática de histórico escolar, a partir das informações registradas no módulo de controle acadêmico;
- Criação de uma interface de relatórios para o coordenador do curso, permitindo acompanhar o desempenho do programa, com gráficos e estatísticas.

Em geral, é importante continuar avaliando as necessidades do PPGI e dos usuários, e trabalhar para melhorar e expandir o módulo de controle acadêmico do Marvin, de forma a atender às suas necessidades.

Referências

- BARCELLOS, M. P. Engenharia de software, notas de aula. Vitória: UFES, 2018. Disponível em: <<https://nemo.inf.ufes.br/wp-content/uploads/Monalessa/EngSoftware/NotasDeAula-EngSw-EngComp-v2018.pdf>>. Citado 2 vezes nas páginas 17 e 19.
- CHAVES, B. d. F. Modernização de sistemas legados para disponibilização em dispositivos móveis com arquitetura baseada em microservices. UFPE - Pernambuco, 2017. Citado na página 20.
- COAD, P. Object-oriented patterns. Communications of the ACM, V, 1992. Citado na página 21.
- COX, B. J. Planning the softwareindustrial revolution. 1990. Citado na página 6.
- FALBO, R. d. A. Projeto de sistema de sofware. 2018. Notas de Aula. Citado 2 vezes nas páginas 8 e 19.
- FARIA, T. Java ee 7 com jsf, primefaces e cdi. 2015. Citado na página 20.
- FOWLER, M. Patterns of enterprise application architecture. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 2002. Citado 3 vezes nas páginas 8, 24 e 32.
- GUERRA, E. Design patterns com java. Casa do Código, 2014. Citado na página 23.
- JOHNSON, R. E. Designing reusable classes. Journal of Object Oriented Programming – JOOP, 1988. Citado na página 21.
- KOTONYA, G.; SOMMERVILLE, I. Engenharia de sofware. Pearson Education, 2011. Citado 2 vezes nas páginas 18 e 19.
- MALDONADO, J. C. et al. Padrões e frameworks de software. *Notas Didáticas, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, ICMC/USP, São Paulo, SP, Brasil*, 2002. Citado na página 21.
- MARTIN, R. C. Código limpo, habilidades práticas do agile software. Alta Books, 2009. Citado na página 23.
- PRESSMAN, R. S.; MAXIM, B. R. Engenharia de sofware, uma abordagem profissional. AMGH, 2016. Citado 2 vezes nas páginas 18 e 19.
- SALVATORE, T. R. Alocaweb e biblattes - módulos do sistema marvin. monografia (projeto de graduação). Universidade Federal do Espírito Santo, 2016. Citado na página 24.
- SOUZA, V. E. S. Frameweb – a framework-based design method for web engineering. dissertação (mestrado). Universidade Federal do Espírito Santo, 2007. Citado na página 21.
- 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 4 vezes nas páginas 8, 23, 32 e 33.

WEISSMANN, H. L. Vire o jogo com spring framework. Casa do Código, 2012. Citado 2 vezes nas páginas 8 e 22.

Apêndices



Documento de Requisitos de Sistema

Módulo de controle acadêmico da Pós-Graduação do Marvin

Vitória, ES

2023

1 Introdução

Este documento apresenta a especificação dos requisitos do sistema *Módulo de controle acadêmico da Pós-Graduação do Marvin*.

Esta especificação foi construída aplicando-se técnicas de levantamento de requisitos, estórias de usuário, modelagem de caso de uso e de classes utilizando a linguagem UML.

A Seção 2 descreve os requisitos levantados junto aos *stakeholders*. A Seção 3 explica a divisão em subsistemas, descrevendo brevemente cada um deles. A Seção 4 apresenta o modelo de casos de uso, incluindo descrições de atores, os diagramas de casos de uso e suas respectivas descrições. A Seção 5 traz os modelos conceituais estruturais do sistema na forma de diagramas de classes. Por fim, a Seção 6 detalha o dicionário do projeto, contendo as definições das classes identificadas.

2 Definição de Requisitos

Esta seção descreve o resultado da atividade de levantamento de requisitos. A Subseção 2.1 resume de forma geral o propósito do sistema e a motivação para seu desenvolvimento. A Subseção 2.2 descreve o minimundo do sistema, apresentando superficialmente suas principais características. A Subseção 2.3 lista os requisitos de usuário do sistema, na forma de estórias de usuário e requisitos não-funcionais.

2.1 Descrição do Propósito do Sistema

Analizando o sistema utilizado atualmente pelo Programa de Pós-Graduação em Informática (PPGI) para gerenciamento das demandas acadêmicas, verificou-se que é exigido muito trabalho manual dos funcionários da secretaria do PPGI e que há dificuldade na comunicação dos alunos com o programa.

Diante desta situação, há uma sobrecarga no recebimento de e-mails, visto que, este é o canal disponibilizado para solicitações das demandas e esclarecimento de dúvidas dos(as) alunos(as).

Deste modo, constatou-se que o PPGI necessita de um sistema de *software* para gerenciar demandas relacionadas ao programa de pós-graduação a fim de otimizar o processo existente. Pretende-se atender essa necessidade por meio de um módulo do sistema Marvin.¹

2.2 Descrição do Minimundo

O Marvin é um sistema *Web* voltado para o gerenciamento de atividades do meio acadêmico, seu código vem sendo implementado desde 2019 e ele integra módulos desenvolvidos ao longo dos anos por estudantes do curso de graduação da UFES, sob a orientação do prof. Vítor E. Silva Souza.

O propósito do módulo implementado no presente estudo é suprir algumas demandas do PPGI a fim de amenizar a sobrecarga de trabalhos manuais e o uso exacerbado de e-mail. Para tal, foram identificadas as necessidades descritas a seguir.

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

2.2.1 Oferta de disciplinas

Atualmente, toda oferta de disciplinas se dá por meio de divulgação das disciplinas disponíveis no site do PPGI, sendo avisado aos(as) estudantes via e-mail e, posteriormente, a matrícula feita via formulário. Deste modo, foram identificados os seguintes problemas:

- Dificuldade do(a) estudante em saber se seu formulário realmente foi recebido pelo responsável em efetivar a matrícula;
- Alto número de e-mails relacionados à solicitação de matrícula e dúvidas sobre o assunto, sobrecarregando a secretaria;
- Dificuldade do(a) estudante em saber o andamento do processamento de matrícula;
- Dificuldade por parte do responsável pelo gerenciamento, em visualizar de forma simples o número de alunos interessados em cada disciplina ofertada, no momento de efetivar as matrículas no sistema da PRPPG.

Tendo em vista esses problemas, é necessário possibilitar o cadastro de disciplinas no sistema a fim de permitir a oferta das mesmas aos alunos. Além disso, deseja-se permitir ao(à) estudante ver as ofertas das disciplinas de uma maneira simples no momento de solicitar a matrícula, assim como a visualização do estado atual da solicitação. É preciso que o sistema possibilite ao responsável pelo gerenciamento das matrículas uma visão clara e objetiva por meio de listagem dos requerimentos por parte dos(as) estudantes.

2.2.2 Aproveitamento de créditos e solicitação de diploma

Neste momento, a solicitação de aproveitamento de créditos das disciplinas e a solicitação de diplomas são realizadas via e-mail com anexo de documentos, sendo gerado um grande número de e-mails aos responsáveis pela resolução destas demandas. Visto isto, deve ser possível que o(a) estudante possa fazer tais solicitações via sistema, possibilitando um fácil controle e visualização da situação dos requerimentos por parte do(a) estudante e da secretaria/coordenação do PPGI.

2.2.3 Gerenciamento de prazos

Atualmente, os diversos prazos a serem cumpridos por alunos do PPGI são publicados no site do programa e enviados por e-mail pela secretaria quando os estudantes entram no curso. Isso gera uma obrigação para a secretaria periodicamente verificar se há estudantes descumprindo ou próximos de descumprir prazos. Deseja-se um sistema que permita que a secretaria e a coordenação do PPGI possam visualizar esta informação e eventualmente notificar os estudantes com maior facilidade.

Também é interessante que os(as) estudantes tenham em vista os prazos de retorno após alguma solicitação feita à secretaria, como as descritas nas seções 2.2.1 e 2.2.2, para que não haja procura de forma desnecessária por meio eletrônico antes do prazo estabelecido.

2.3 Requisitos de Usuário

Tomando por base o contexto do sistema descrito na Seção 2.2 e considerando como principais *stakeholders* alunos, professores, secretaria e coordenação do PPGI, foram identificadas estórias de usuário e requisitos não-funcionais.

As estórias 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 estórias de usuário específicas, na Tabela 2.

É importante ressaltar que este módulo do Marvin parte do pressuposto que há outro módulo responsável pelos cadastros básicos das unidades administrativas da universidade (ex.: centros de ensino, programas de pós-graduação, cursos de pós-graduação) bem como de acadêmicos e suas relações com tais unidades administrativas (ex.: coordenador(a)/secretário(a) de programa de pós-graduação, estudante de pós-graduação). A lista abaixo indica estórias de usuário que pressupomos já tenham sido implementadas neste outro módulo:

- US-1: como administrador(a) do sistema, eu quero cadastrar centros de ensino, para que programas de pós graduação possam ser associados a eles;
- US-2: 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(a), etc.);
- US-3: 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.: estudante matriculado, etc.).
- US-4: como administrador(a) do sistema, eu quero cadastrar acadêmicos(as) no sistema, para que eles possam usar as funcionalidades do mesmo;
- US-5: 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;
- US-6: 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;

US-7: 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;

US-8: 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.

Tabela 1 – Estórias de Usuário.

ID:	US-9	Depende:	US-5 , US-6	Prioridade:	Alta
Descrição:	Como secretaria/coordenação, quero cadastrar períodos letivos, para que o professor possa ofertar disciplinas.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, deve ser informado: nome, período (data inicial e final) de oferta, período de matrícula, datas de início e fim do período letivo (aulas); – CA2: Não é permitido cadastrar um período letivo que tenha períodos de oferta, de matrícula ou de aula conflitantes; – CA3: Não é permitido cadastrar períodos letivos com o mesmo nome; – CA4: Consultas podem ser feitas por: nome (Ex: 2021/2); – CA5: Períodos letivos podem ser excluídos apenas se não estiverem associados a outros objetos. 				

ID:	US-10	Depende:	US-3 , US-5 , US-6	Prioridade:	Alta
Descrição:	Como secretaria/coordenação, quero cadastrar disciplinas, para que os professores possam ofertá-las no período vigente.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Para cadastrar, deve ser informado: nome, tipo,² código, créditos, carga horária, ementa e bibliografia; – CA2: Não é permitido cadastrar disciplinas com o mesmo código; – CA3: Consultas podem ser feitas por: nome, código e tipo; – CA4: Disciplinas podem ser excluídas apenas se não estiverem associadas a outros objetos. 				

² De acordo com o regimento atual do PPGI, um item da organização curricular dos cursos de pós-graduação pode ser de um dos seguintes tipos: Disciplina Regular; Disciplina Eventual; Tópico Especial; Estágio de Docência; Estudo Dirigido; Seminário Temático; Dissertação de Mestrado; Exame de Qualificação de Doutorado; Exame de Proposta de Tese de Doutorado; Tese de Doutorado

ID:	US-11	Depende:	US-8, US-9, US-10	Prioridade:	Alta
Descrição:	Como professor, quero ofertar disciplinas a cada período letivo, para que os alunos possam se matricular.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Deve haver um período letivo cadastrado no sistema e dentro do prazo de oferta; – CA2: Para ofertar, deve ser informado: disciplina, o número máximo de estudantes permitido, os dias e horários de aula; – CA3: No caso de disciplina do tipo Tópico Especial ou Seminário Temático, deve ser informado ainda: subtítulo, ementa e bibliografia; – CA4: A disponibilização da disciplina será para o período letivo que se encontra dentro do prazo de oferta; – CA5: O sistema deve mostrar as disciplinas que já foram ofertadas no período letivo atual; – CA6: A remoção da oferta só pode ser realizada dentro do prazo de oferta do período letivo atual. 				

ID:	US-12	Depende:	US-7, US-11	Prioridade:	Alta
Descrição:	Como aluno, quero solicitar matrícula em disciplinas, para cursá-las no período letivo vigente.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Deve haver um período letivo cadastrado no sistema e dentro do prazo de matrícula; – CA2: Devem ser exibidas as disciplinas que estão na oferta do período letivo atual, com todas as informações pertinentes (professor, horários de aula, dados curriculares da disciplina); – CA3: Para solicitar matrícula, deve ser informado: o(a) professor orientador(a) e ao menos uma disciplina ou, caso o(a) estudante já tenha cumprido todos os créditos, indicar a opção “Dissertação de Mestrado” ou “Tese de Doutorado” (dependendo do seu curso). Deve ser registrada a data da solicitação; – CA4: O sistema deve mostrar o pedido de matrícula feito para o período letivo vigente, caso já tenha sido efetuado; – CA5: Deve ser possível fazer novo pedido de matrícula para o período letivo vigente, caso o prazo ainda esteja aberto, substituindo o anterior; – CA6: Deve ser possível cancelar o pedido de matrícula do período letivo vigente, caso o prazo ainda esteja aberto.
--------------------------------	---

ID:	US-13	Depende:	US-12	Prioridade:	Alta
Descrição:	Como professor ou coordenador, quero aprovar uma solicitação de matrícula de alunos (no caso de professores, apenas daqueles(as) sob minha orientação), para que a secretaria possa efetivá-las.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Deve haver um período letivo cadastrado no sistema e dentro do prazo de matrícula ou de aula; – CA2: No caso de professor, devem ser exibidas todas as solicitações de estudantes sob sua orientação para o período letivo corrente, exibindo nome do(a) estudante, data da solicitação e situação (aprovação pendente, não aprovada, atendida); – CA3: No caso do(a) coordenador(a), poderão ser exibidas, se solicitado, todas as solicitações de todos os(as) estudantes para o período letivo corrente, exibindo as mesmas informações e ainda o nome do(a) orientador(a); – CA4: Ao selecionar uma solicitação, devem ser exibidos também os detalhes da mesma (quais disciplinas foram solicitadas e a situação de cada uma delas); – CA5: Na exibição da listagem de solicitações, o professor/coordenador(a) deve indicar as solicitações que deseja aprovar e confirmar; – CA6: Como alternativa, na exibição de detalhes de uma solicitação, o professor pode indicar que deseja aprová-la e confirmar; – CA7: Aprovações não podem ser desfeitas e deve-se registrar a data da aprovação.
--------------------------------	---

ID:	US-14	Depende:	US-12 , US-13	Prioridade:	Alta
Descrição:	Como secretaria/coordenação, quero visualizar a situação das matrículas de forma consolidada, para efetuar os cadastros no sistema oficial da PRPPG/UFES (SAPPG).				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Deve ser possível exibir as matrículas por disciplina, reunindo numa mesma tela os dados necessários para realizar o cadastro no SAPPG; – CA2: Deve ser possível exibir as matrículas também por aluno; – CA3: Em qualquer um dos casos acima, deve ser possível ver se a solicitação de matrícula foi aprovada pelo(a) orientador(a); – CA4: Deve ser possível marcar e desmarcar as solicitações de matrícula que já tenham sido cadastradas no SAPPG, permitindo (1) que a tarefa seja interrompida e posteriormente retomada com facilidade; e (2) que os(as) estudantes vejam o estado atual de suas solicitações de matrícula; – CA5: Deve ser possível gerar um relatório em PDF com as mesmas informações exibidas em tela.
--------------------------------	--

ID: US-15 Depende: US-12 Prioridade: Alta	
Descrição:	Como aluno, quero ver a situação das minhas solicitações de matrícula, para acompanhar se já foram atendidas.
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Devem ser exibidas todas as solicitações já feitas (em todos os períodos letivos), exibindo período letivo, data da solicitação e a situação (aprovação pendente, não aprovada, não atendida, parcialmente atendida, totalmente atendida); – CA2: Ao selecionar uma solicitação, devem ser exibidos também os detalhes da mesma (quais disciplinas foram solicitadas e a situação de cada uma delas).

ID: US-16 Depende: US-12 Prioridade: Alta	
Descrição:	Como coordenação, quero parametrizar as demandas.
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: No prazo deve ser informado o seu nome, e a quantidade máxima de dias para sua conclusão; – CA2: Deve ser possível parametrizar solicitação de diploma, aproveitamento de créditos e solicitação do número de matrícula; – CA3: Para as solicitações deve ter um campo para armazenar o prazo máximo para envios de e-mails automáticos de alerta.

ID: US-17 Depende: US-16 Prioridade: Media	
---	--

Descrição:	Como aluno, quero solicitar aproveitamento de créditos.
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Apenas alunos com número de matrícula ativo podem fazer solicitações; – CA2: Devem ser enviados como anexos os documentos necessários para esta solicitação, conforme regulamentação atual do programa de pós-graduação; – CA3: Não deve ser possível deletar a solicitação caso a situação seja diferente de Aguardando; – CA4: Não deve ser possível ter mais de uma solicitação de aproveitamento de créditos, com a situação igual a aguardando ou em análise.

ID:	US-18	Depende:	US-16	Prioridade:	Media
Descrição:	Como aluno, quero solicitar meu número de matrícula.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Deve existir possibilidade de solicitar o número de matrícula, caso não o tenha; – CA2: Só deve ser possível solicitar o número de matrícula caso o aluno não o tenha; – CA3: Não deve ser possível deletar a solicitação caso a situação seja diferente de Aguardando. – CA4: Não deve ser possível ter mais de uma solicitação de número de matrícula, com a situação igual a aguardando ou em análise.
-------------------------	---

ID:	US-19	Depende:	US-16	Prioridade:	Media
Descrição:	Como aluno, quero solicitar diploma de conclusão.				

Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Apenas alunos com número de matrícula ativo podem fazer solicitações; – CA2: Devem ser enviados como anexos os documentos necessários para esta solicitação, conforme regulamentação atual do programa de pós-graduação; – CA3: Não deve ser possível deletar a solicitação caso a situação seja diferente de Aguardando; – CA4: Não deve ser possível ter mais de uma solicitação de diploma de conclusão, com a situação igual a aguardando ou em análise.
-------------------------	--

ID:	US-20	Depende:	US-17 US-18 US-19	Prioridade:	Alta
Descrição:	Como aluno, quero acompanhar e ser notificado sobre o andamento das minhas demandas.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Deve ser possível visualizar todas as demandas juntamente com sua situação; – CA2: Ao ter alguma alteração no status da minha demanda devo ser notificado via e-mail; 				

ID:	US-21	Depende:	US-17 US-18 US-19	Prioridade:	Média
Descrição:	Como secretaria/coordenação do PPGI, quero visualizar a situação dos requerimentos, para poder dar retorno aos(as) alunos(as), cobrar dos professores, etc.				
Critérios de Aceitação:	<ul style="list-style-type: none"> – CA1: Deve ser possível visualizar todas as demandas dos alunos; – CA2: Os requerimentos necessitam serem ordenados por prazo em ordem ascendente; – CA3: Deve haver a possibilidade responder as requisições aos(as) alunos(as) respondendo-a com um texto descritivo para caso haja necessidade de rejeição de alguma solicitação, e para caso de solicitação de matrícula deve ser possível informar o número de matrícula. Também é necessário ter a possibilidade de alterar a situação do requerimento; – CA4: É necessário que haja a possibilidade de dar como concluída a demanda do aluno; – CA5: É necessário ter a possibilidade de revogar a conclusão do requerimento do aluno retornando-a para a situação em análise; – CA6: Os alunos devem ser avisados após a conclusão das etapas de uma demanda por meio do e-mail cadastrado no usuário do aluno; – CA7: Ao concluir a solicitação de diploma deve ser possível inativar o número de matrícula do aluno. 				

Tabela 2 – Requisitos Não Funcionais.

ID	Descrição	Categoria	Prioridade
RNF-1	O sistema deve estar disponível para os principais navegadores <i>Web</i> .	Portabilidade	Alta
RNF-2	O <i>software</i> deve preservar a consistência dos dados, onde toda a informação a ser inserida tenha todos os campos obrigatórios preenchidos.	Confiabilidade	Alta
RNF-3	O sistema deve ser de fácil manutenção, utilizando controle de versões e padrões de <i>design</i> .	Redigibilidade	Alta
RNF-4	As páginas do módulo devem ser de rápida execução com tempo inferior a 3 segundos de carregamento de dados.	Eficiência de tempo	Média
RNF-5	O sistema deve ser de fácil aprendizado, não sendo necessário a realização de nenhum tipo de tutorial ou treinamento para o seu uso.	Facilidade de Aprendizado	Média
RNF-6	A ferramenta deve ser de fácil utilização, não sendo preciso acessar muitas páginas para realizar as funções.	Facilidade de Operação	Alta
RNF-7	O sistema deve ser seguro contra os ataques mais comuns presentes na internet.	Segurança	Média
RNF-8	O sistema deve ser desenvolvido utilizando a metodologia <i>Frame Web</i> na linguagem Java e na plataforma Jakarta EE.	Tecnologia	Alta

3 Identificação de Subsistemas

Para atender aos requisitos elencados na Seção 2 não houve necessidade de divisão em subsistemas, porém o *Módulo de controle acadêmico da Pós-Graduação do Marvin* é um subsistema do Marvin fazendo o uso de dois outros conforme pode ser visto na Figura 1, que ilustra os subsistemas e suas interdependências, enquanto a Tabela 3 apresenta uma breve descrição dos subsistemas já existentes e do subsistema criado.

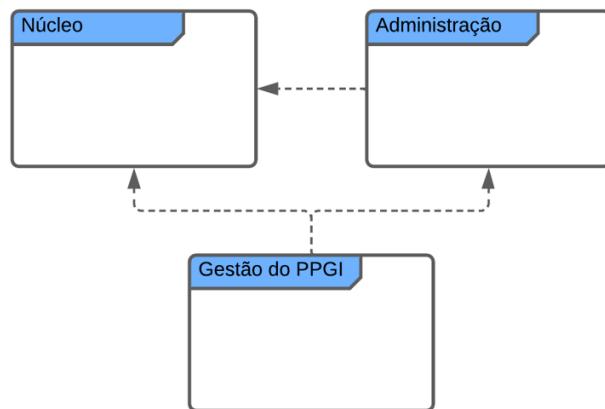


Figura 1 – Diagrama de Pacotes e os Subsistemas Identificados.

Tabela 3 – Subsistemas identificados e suas interdependências (Elaboração própria).

Subsistema	Descrição
Núcleo	Módulo que gerencia o cadastro e autenticação de acadêmicos.
Administração	Módulo que gera o cadastro das unidades organizacionais da UFES (centros, departamentos, programas de pós-graduações, cursos).
Gestão do PPGI	Módulo de controle acadêmico da Pós-Graduação do Marvin.

4 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 (Elaboração própria).

Autor	Descrição
Autor 1	Aluno.
Autor 2	Professor.
Autor 3	Secretaria/Coordenação PPGI.

A Figura 2 apresenta o diagrama de casos de uso do subsistema Gestão do PPGI.

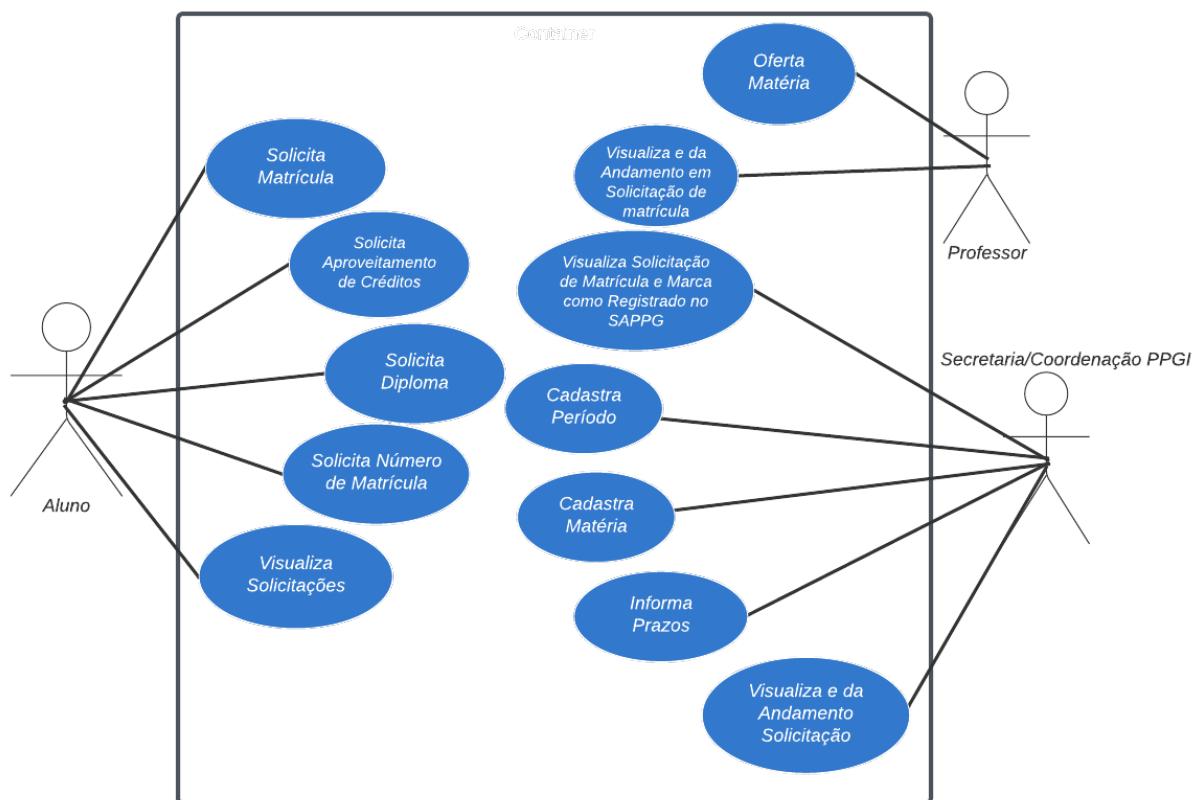


Figura 2 – Diagrama de Casos de Uso do subsistema Gestão do PPGI.

A Tabela 5 apresenta a rastreabilidade dos casos de uso em relação às estórias de usuário apresentadas na Seção 2.3, que já se encontram suficientemente detalhadas (considerando seus critérios de aceitação), dispensando, portanto, subsequentes descrições.

Tabela 5 – Rastreabilidade dos casos de uso do subsistema Gestão do PPG.

Id	Nome	Requisitos
UC-1	Cadastra Período	US-9
UC-2	Cadastra Matéria	US-10
UC-3	Oferta Matéria	US-11
UC-4	Solicita Matrícula	US-12 US-15
UC-5	Solicita Número de Matrícula	US-18
UC-6	Visualiza e da Andamento em Solicitação de Matrícula	US-13
UC-7	Visualiza Solicitação de Matrícula e Marca como Registrado no SAPPG	US-14
UC-8	Informa Prazos	US-16
UC-9	Solicita Aproveitamento de Créditos	US-17
UC-10	Solicita Diploma	US-19
UC-11	Visualiza Solicitações	US-20
UC-12	Visualiza e da Andamento em Solicitação	US-21

5 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 4.

A Figura 3 apresenta o diagrama de classes do subsistema Gestão do PPGI. Na Seção 6 — Dicionário de Projeto — são apresentadas as descrições das classes, atributos e operações presentes neste diagrama.

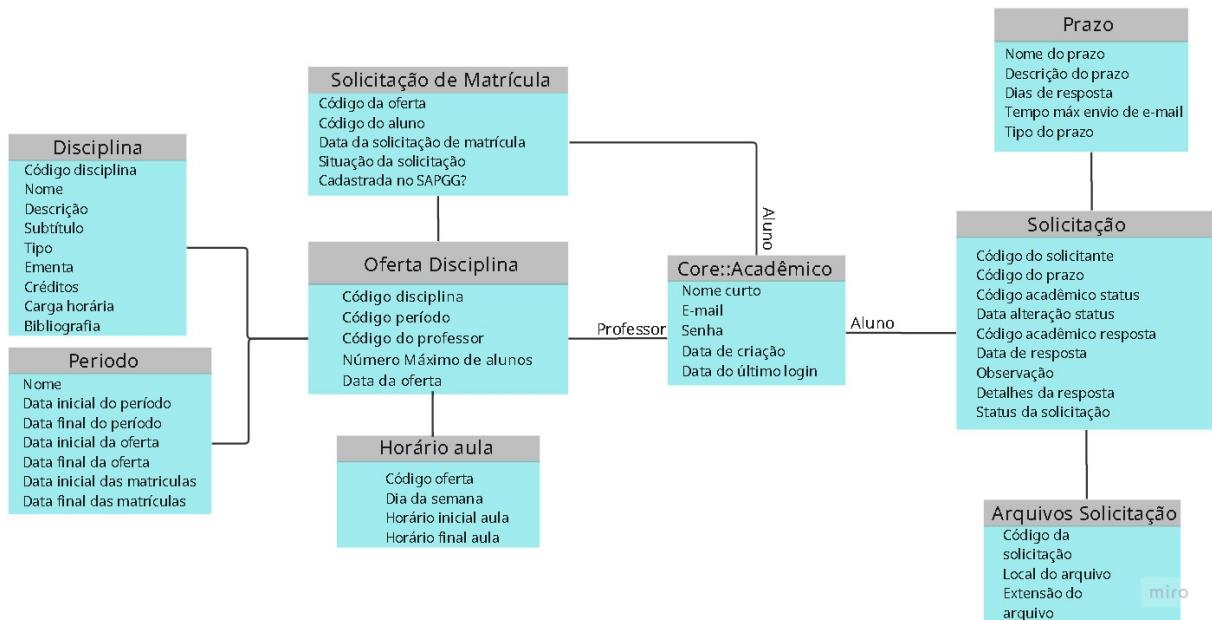


Figura 3 – Diagrama de classes do subsistema Gestão do PPGI.

6 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 5, 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 6 – Detalhamento da classe *Período*.

Propriedade	Tipo	Obr.?	Descrição
Código	Inteiro		Código único do período (Chave primária).
Nome do período	Texto	x	Nome do período letivo, exemplo 2021/1.
Data inicial do período	Data	x	Data de início do período letivo.
Data final do período	Data	x	Data de fim do período letivo.
Data inicial de matrícula	Data	x	Data de início do período de solicitação de matrícula.
Data final de matrícula	Data	x	Data de fim do período de solicitação de matrícula.
Data inicial de oferta	Data	x	Data de início de oferta das disciplinas.
Data final de oferta	Data	x	Data de fim de oferta das disciplinas.

Tabela 7 – Detalhamento da classe *Disciplina*.

Propriedade	Tipo	Obr.?	Descrição
Código	Inteiro		Código único da disciplina (Chave primária).
Nome	Texto	x	Nome da disciplina.
Subtítulo	Texto		Subtítulo da disciplina.
Ementa	Texto		Ementa da matéria a ser ofertada.
Descrição	Texto		Texto de descrição da disciplina.

Propriedade	Tipo	Obr.?	Descrição
Bibliografia	Texto		Bibliografia da matéria.
Carga horária	Inteiro	x	Valor em horas da carga horária da matéria.
Créditos	Inteiro	x	Créditos da disciplina.

Tabela 8 – Detalhamento da classe *Oferta Disciplina*.

Propriedade	Tipo	Obr.?	Descrição
Código	Inteiro		Código único da oferta matéria (Chave primária).
Código acadêmico	Inteiro	x	Código referencial com a classe Acadêmico.
Código do período	Inteiro	x	Código referencial com a classe Período.
Código disciplina	Inteiro	x	Código referencial com a classe Disciplina.
Número máximo de alunos	Inteiro	x	Quantidade máxima de alunos permitida na turma ofertada.
Data oferta	Data	x	Data da oferta da matéria.

Tabela 9 – Detalhamento da classe *Horário Aula*.

Propriedade	Tipo	Obr.?	Descrição
Código	Inteiro		Código único da classe horário aula (Chave primária).
Código oferta matéria	Inteiro	x	Código referencial com a classe Oferta matéria.
Dia da semana	Enumerador	x	Dia da aula da matéria ofertada.
Horário inicial aula	Hora	x	Horário inicial da aula.
Horário final aula	Hora	x	Horário final da aula.

Tabela 10 – Detalhamento da classe *Solicitação*.

Propriedade	Tipo	Obr.?	Descrição
Código	Inteiro		Código único da classe solicitação (Chave primária).
Código do Prazo	Inteiro		Código referencial com a classe Prazo.
Código aluno	Inteiro	x	Código referencial com a classe Acadêmico.
Situação solicitação	Enumerador	x	Enumerador contendo a situação da solicitação.
Data solicitação	Data	x	Data da solicitação.
Código acadêmico status	Inteiro		Código referencial com a classe Acadêmico.

Propriedade	Tipo	Obr.?	Descrição
Data alteração status	Data		Data de alteração do status.
Código acadêmico resposta	Inteiro		Código referencial com a classe Acadêmico.
Data da resposta da solicitação	Data		Data da conclusão de solicitações do tipo Diploma, aproveitamento de créditos e solicitação de número de matrícula.

Tabela 11 – Detalhamento da classe *Solicitação de matrícula*.

Propriedade	Tipo	Obr.?	Descrição
Código	Inteiro		Código único da classe solicitação matrícula (Chave primária).
Código da Oferta	Inteiro	x	Código referencial com a classe Oferta Disciplina.
Código aluno	Inteiro	x	Código referencial com a classe Acadêmico.
Data solicitação de matrícula	Data	x	Data que foi realizada a solicitação de matrícula.
Situação solicitação	Enumerador	x	Situação da solicitação.
Cadastrada no SAPPG?	Booleano	x	A matrícula foi efetivada no SAPPG?

Tabela 12 – Detalhamento da classe *Prazo*.

Propriedade	Tipo	Obr.?	Descrição
Código	Inteiro		Código único da classe Prazo (Chave primária).
Nome do prazo	Texto	x	Nome do prazo.
Descrição do prazo	Texto		Descrição do prazo.
Dias de resposta	Inteiro	x	Tempo máximo de retorno.
Tempo máx envio de e-mail	Inteiro	x	Tempo máximo em relação ao fim do prazo que o sistema começará a enviar os e-mails.
Tipo do prazo	Enumerador	x	Tipos de prazos de solicitação (Solicitação de diploma, aproveitamento de créditos e solicitação do número de matrícula).



Documento de Projeto de Sistema

**Módulo de controle acadêmico da
Pós-Graduação do Marvin**

Vitória, ES

2023

1 Introdução

Este documento apresenta o projeto (*design*) do sistema *Módulo de controle acadêmico da Pós-Graduação do Marvin*. O módulo é voltado para o gerenciamento das demandas acadêmicas relacionadas ao programa de pós-graduação em informática (PPGI), a fim de otimizar o processo existente e auxiliar os secretários e coordenação do PPGI.

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
Jakarta EE	8	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.
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.
EJB	4.0	API para construção de componentes transacionais gerenciados por <i>container</i> .	Implementação das regras de negócio em componentes distribuídos, transacionais, seguros e portáveis.
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.

Tecnologia	Versão	Descrição	Propósito
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.
MySQL Server	8.0	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Armazenamento dos dados manipulados pela ferramenta.
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
Kdiff3	0.9	Ferramenta de comparação de arquivos.	Auxiliar em ajustes após correções e em resoluções de conflitos.
Miro	Versão web 2022	Plataforma de lousa interativa digital.	Criação de diagramas.
Git	2.26	Sistema de controle de versões distribuído.	Registro de histórico de alterações no repositório.
Git Extensions	3.5	Sistema de controle de versões distribuído.	Registro de histórico de alterações no repositório.
Sourcetree	3.4	Sistema de controle de versões distribuído.	Registro de histórico de alterações no repositório.
TeXstudio	4.2	Editor de LaTeX.	Escrita da documentação do sistema, sendo usado o <i>template abnTeX</i> . ¹
Eclipse Java EE IDE for Web Developers	4.24	Ambiente de desenvolvimento (IDE) com suporte ao desenvolvimento Java EE.	Implementação, implantação e testes da aplicação Web Java EE.
Apache Maven	3.8	Ferramenta de gerência/construção de projetos de software.	Obtenção e integração das dependências do projeto.

3 Requisitos Não Funcionais

A Tabela 3 apresenta a especificação dos requisitos não funcionais identificados no Documento de Especificação de Requisitos, os quais foram considerados condutores da arquitetura.

¹ <<http://www.abntex.net.br>>.

Tabela 3 – Especificação de Requisitos Não Funcionais.

RNF-1 – O sistema deve ser de fácil aprendizado, não sendo necessário a realização de nenhum tipo de tutorial ou treinamento para o seu uso..	
Categoria:	Usabilidade
Tática / Tratamento:	Interface seguindo os <i>designs</i> padrões voltados para melhor usabilidade e experiência do usuário.
Medida:	Verificar com o usuário final se o sistema ficou intuitivo e de fácil uso e se o mesmo conseguiu utilizá-lo sem nenhum tipo de auxílio.
Critério de Aceitação:	Usuários conseguirem utilizar todas as funções do sistema sem um treinamento prévio.

RNF-2 – O *software* deve preservar a consistência dos dados, onde toda a informação a ser inserida tenha todos os campos obrigatórios preenchidos.

Categoria:	Confiabilidade
Tática / Tratamento:	Validações em todas as ações do sistema que possam gerar inconsistências nos registros, tais como falta de campos obrigatórios e restrições de integridade.
Medida:	Criações de validações no <i>software</i> .
Critério de Aceitação:	Usuário conseguir utilizar o sistema sem que haja nenhum tipo de erro inesperado devido a dados inválidos. Usuário ser informado sobre as restrições com mensagens de validação e visibilidade de campos obrigatórios.

RNF-3 – O sistema deve ser de fácil manutenção, utilizando controle de versões e padrões de design.

Categoria:	Redigibilidade
Tática / Tratamento:	Seguir padrões baseados em <i>Design Patterns</i> , e utilização de ferramentas de controle de versão.
Medida:	Desenvolvimento do sistema seguindo padrões e utilizar ferramentas de controle de versão tais como o git.
Critério de Aceitação:	Código de fácil leitura e entendimento, facilidade de manutenção por qualquer outra pessoa com prévio entendimento no assunto.

RNF-4 – As páginas do módulo devem ser de rápida execução com tempo inferior a 3 segundos de carregamento de dados.

Categoria:	Eficiencia
Tática / Tratamento:	Criação de códigos simples, normalização em tabelas e utilização de Design Patterns.
Medida:	Criar códigos otimizados que degradem o mínimo o tempo de execução. Criar tabelas na terceira forma normal.
Critério de Aceitação:	Realização de testes de estresse no sistema.

RNF-5 – O sistema deve estar disponível para os principais navegadores *Web*.

Categoria:	Portabilidade
Tática / Tratamento:	Criação de códigos responsivos e utilização de bibliotecas disponíveis nos principais navegadores.
Medida:	Ser possível acessar o sistema nos principais navegadores (Google Chrome, Mozilla Firefox e Microsoft Edge) sem que haja erros no layout.

Critério de Aceitação:	Testar a aplicação nos principais navegadores sem que haja alterações no <i>layout</i> ou comportamento.
------------------------	--

4 Arquitetura de Software

O desenvolvimento deste módulo é baseado no padrão arquitetônico *Service Layer* (Camada de Serviço) (FOWLER, 2002), onde o software é dividido em três principais camadas conforme pode ser visto na Figura 1.

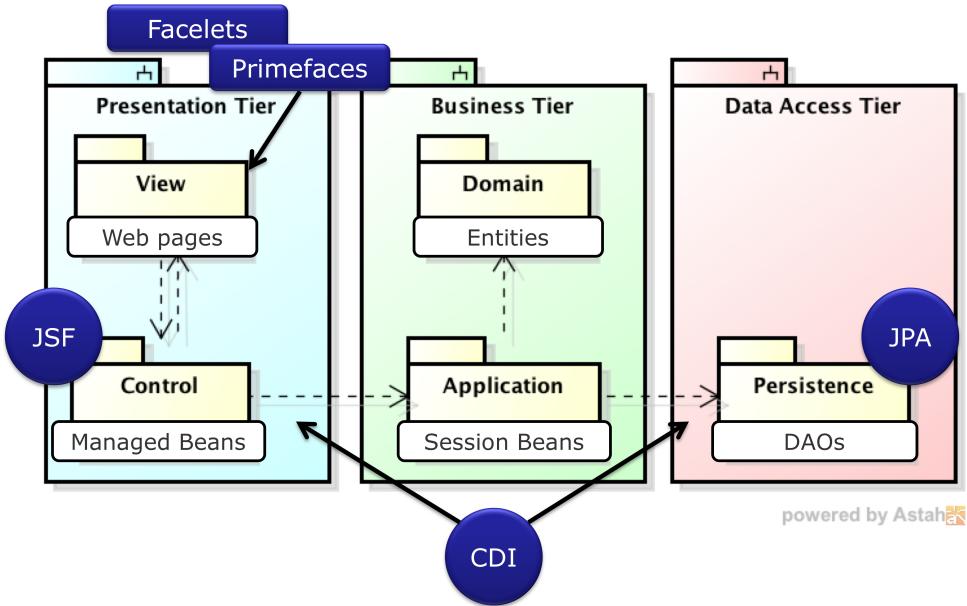


Figura 1 – Arquitetura de Software FrameWeb (SOUZA, 2020).

- Camada de negócio (*Business Tier*): ela é subdividida em duas camadas: lógica de aplicação (*Application*) e lógica de Domínio (*Domain*). Neste ponto que são providas as funcionalidades de acordo com as regras de negócio do sistema;
- Camada de apresentação (*Presentation Tier*): análogo ao seu nome, esta faixa é referente a criação da interface onde o usuário irá interagir com o sistema, nela estão presentes duas camadas provenientes da arquitetura MVC, sendo elas a visão (*View*) e controle (*Control*);
- Camada de acesso a dados (*Data Access Tier*): aqui são realizados os tratamentos interligados à persistência de informações no banco de dados.

5 Modelagem FrameWeb

Módulo de controle acadêmico da Pós-Graduação 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

A camada de negócio é subdividida em duas camadas: Lógica de aplicação e lógica de domínio. A Figura 2 demonstra as entidades que compõe o domínio deste projeto.

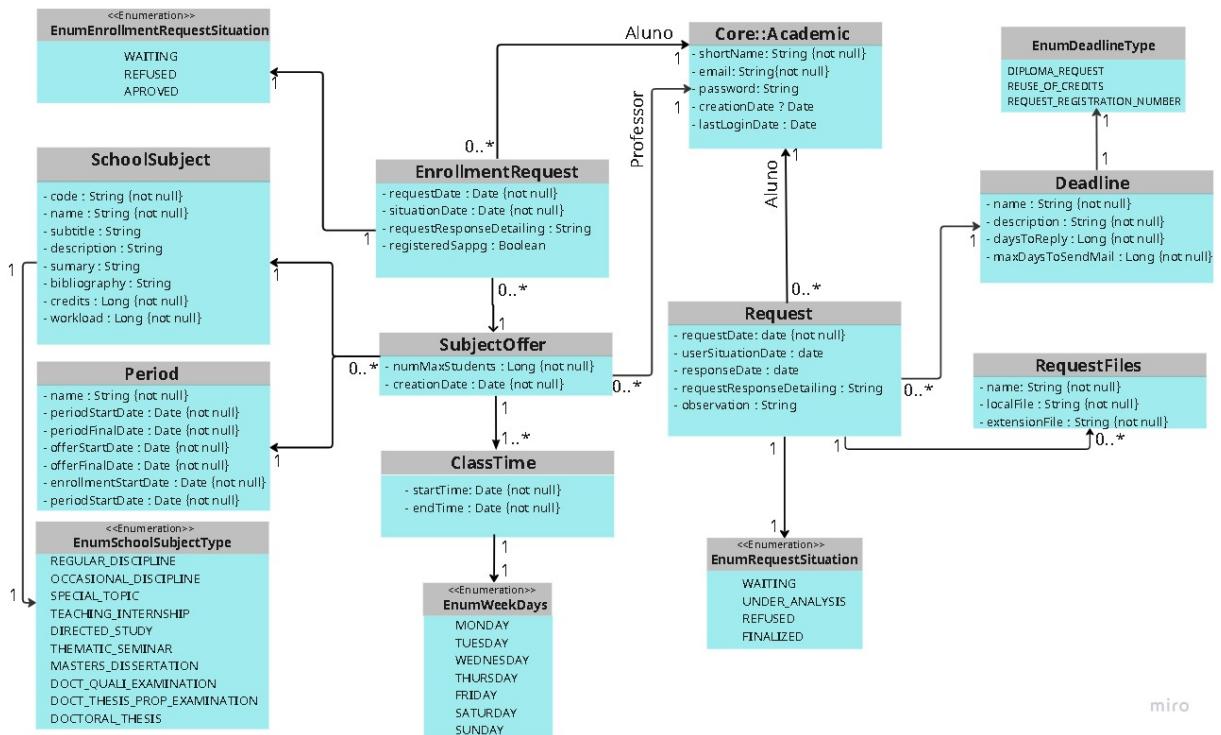


Figura 2 – Modelo de entidades.

O pacote domínio que compõe a camada de negócio contém todas as classes de domínio do projeto, todas essas classes estendem a classe *PersistentObjectSupport*.

Já no pacote referente a aplicação, todas as classes estendem a classe *CrudServiceBean* que contém os métodos de crud já implementados. O modelo de aplicação do *Módulo de controle acadêmico da Pós-Graduação do Marvin* pode ser visto na Figura 3.

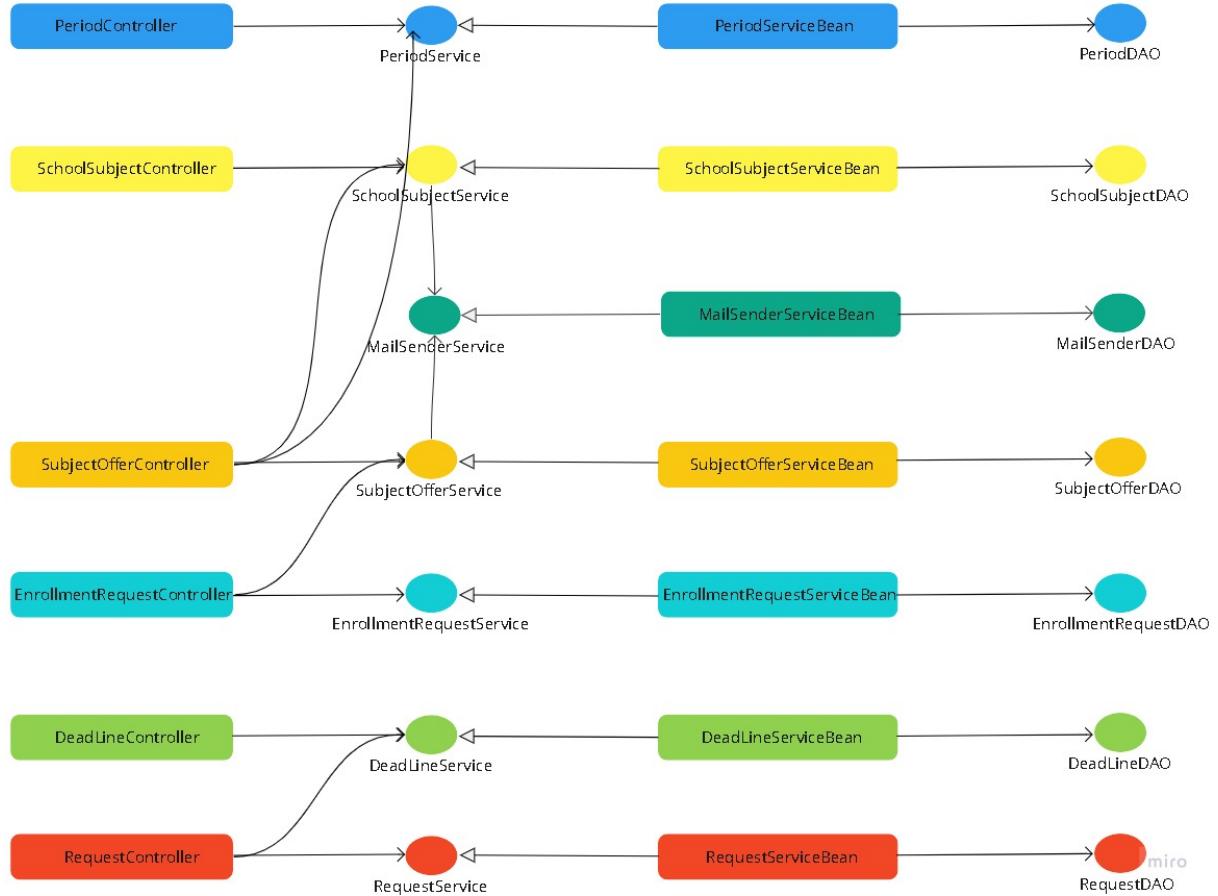


Figura 3 – Arquitetura de Software.

5.2 Camada de Acesso a Dados

Na camada de acesso a dados são realizados os tratamentos ligados à persistência de informações no banco de dados. Todas as classes presentes nesta camada estendem a classe *BaseDAO* que contém todas as funcionalidades básicas de acesso e persistência de dados. O modelo de persistência do subsistema Gestão do PPGI é representado pela Figura 4. Neste modelo estão contidas as interfaces DAO que foram criadas no trabalho, elas são responsáveis pelos métodos de persistência da camada de domínio do sistema. Cada classe existente no nível de domínio necessita de um DAO para que os dados sejam persistidos ou consultados no banco de dados.

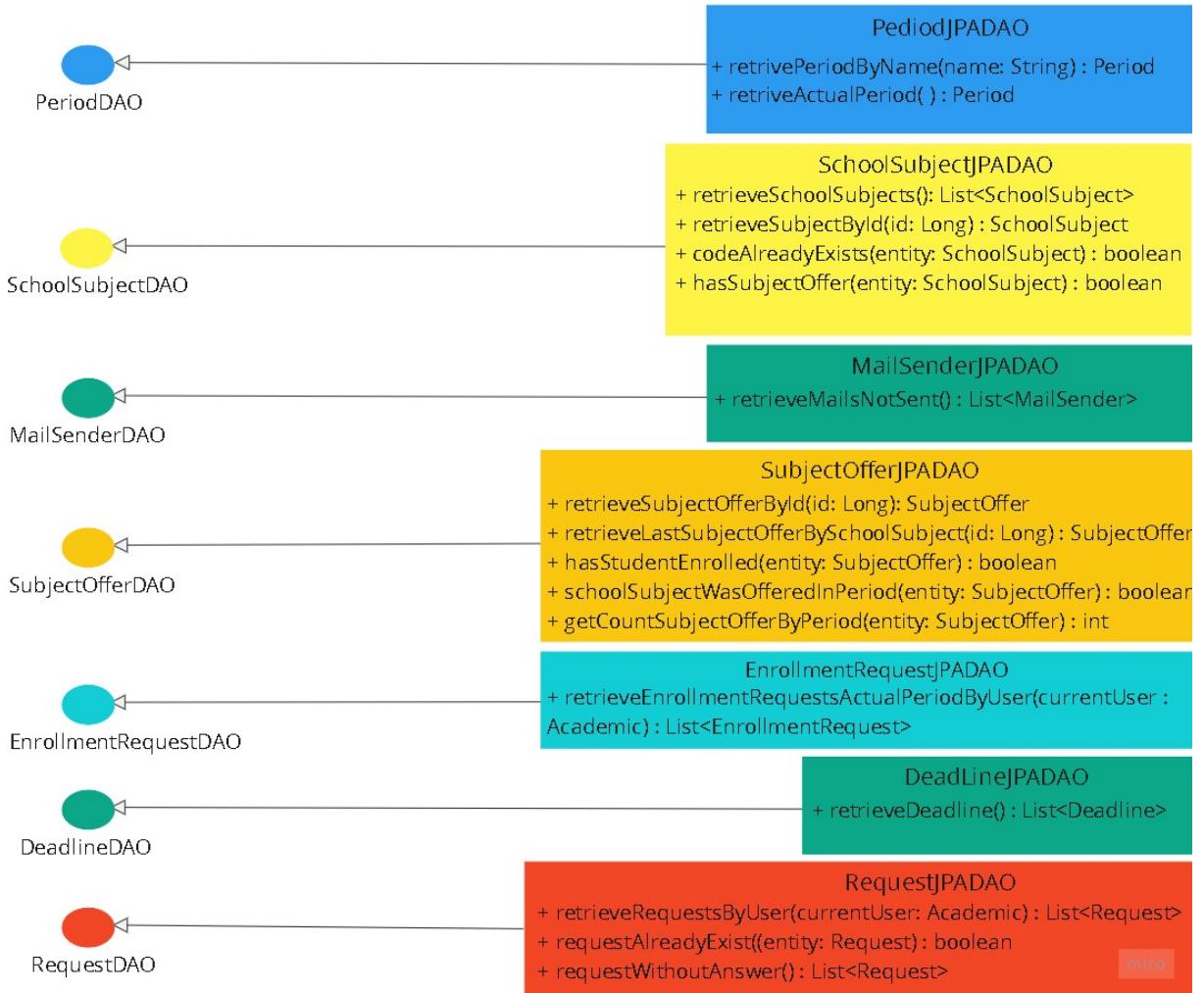


Figura 4 – Modelo de Persistência do subsistema Gestão do PPGI.

5.3 Camada de Apresentação

Este nível é referente à criação das interfaces por meio das quais o usuário irá interagir com o sistema, nela estão presentes duas camadas provenientes da arquitetura MVC, sendo elas a visão (*View*) e controle (*Control*) conforme já apresentado na Figura 1.

As telas referentes ao cadastro de matérias, cadastro de prazos, cadastro de períodos e oferta de disciplinas são telas simples que contém apenas funções de CRUD, elas podem ser representadas pela a Figura 5.

Para as duas telas de solicitação (solicitações diversas e solicitação de matrícula) além das funções padrões de crud existem outras funcionalidades. A representação dessas outras funções podem ser vistas nas figuras 6 e 7.

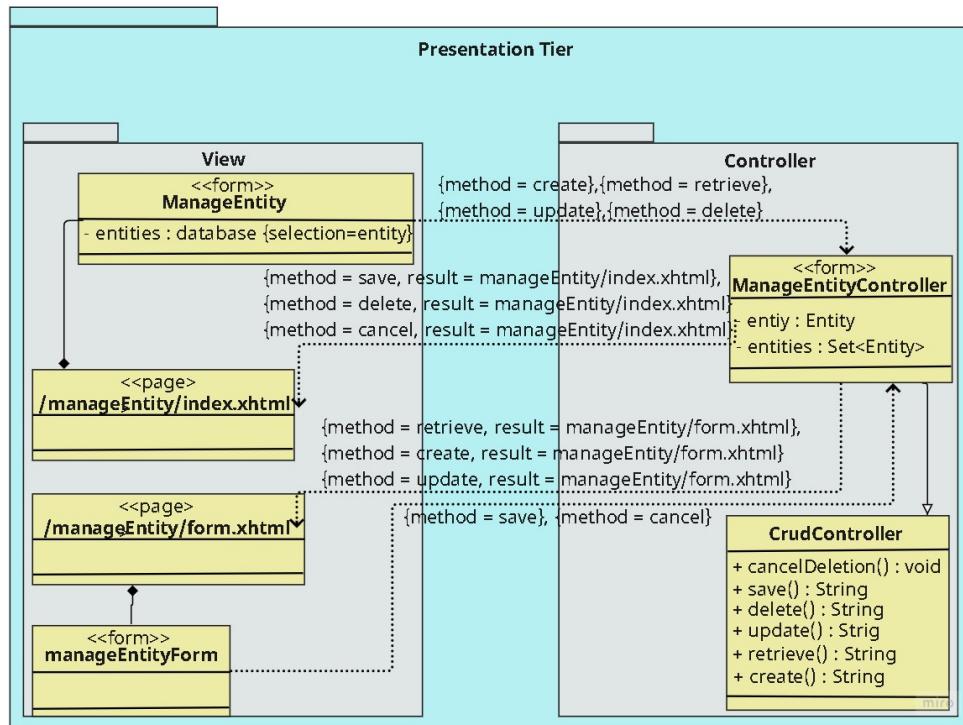


Figura 5 – Modelo de Navegação para as telas de CRUD.

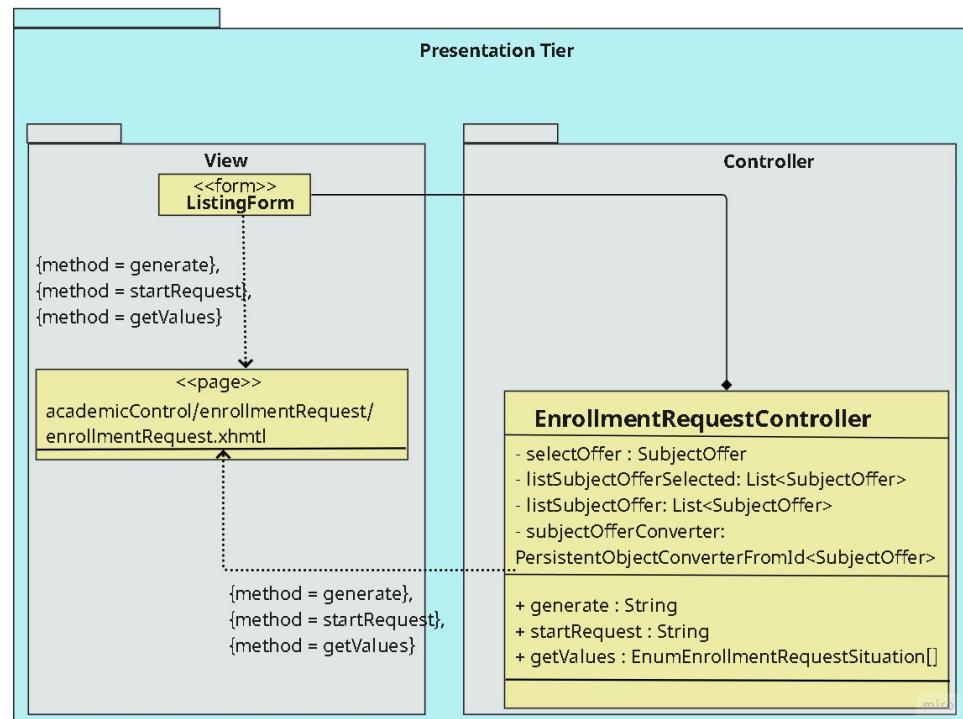


Figura 6 – Modelo de Navegação para a tela de solicitação de matrícula.

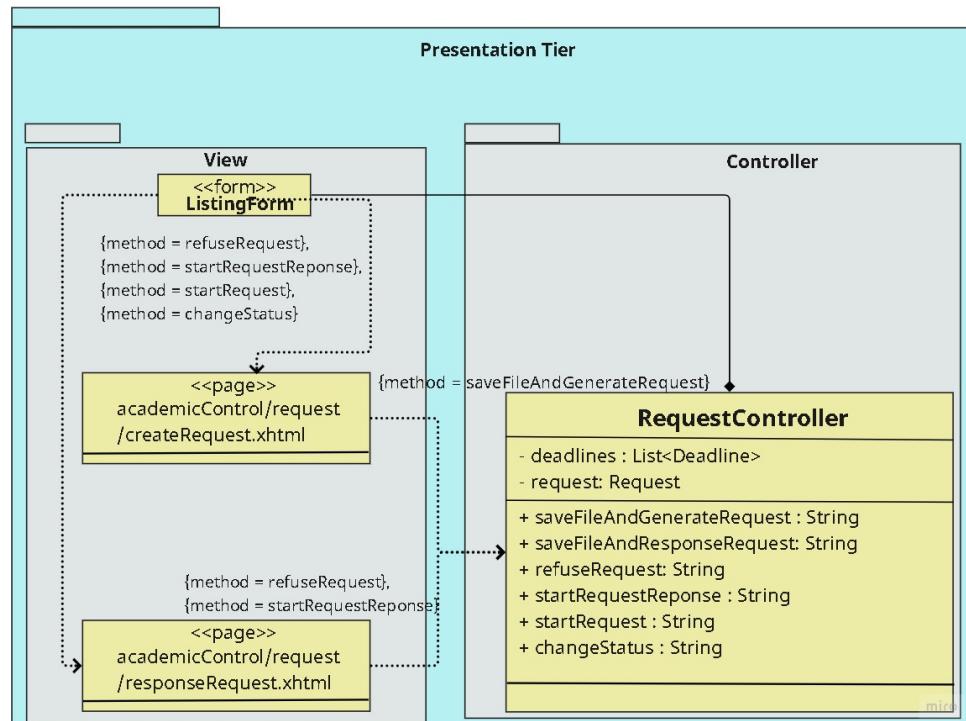


Figura 7 – Modelo de Navegação para a tela de solicitações diversas.

Referências

- FOWLER, M. Patterns of enterprise application architecture. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 2002. Citado na página [4](#).
- 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 [4](#) e [5](#).