



Mateus Oliveira Almeida

# **SysMap - Sistema de Mapeamento Sistemático**

Vitória, ES

2018

Mateus Oliveira Almeida

## **SysMap - Sistema de Mapeamento Sistemático**

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

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Departamento de Informática

Orientador: Prof. Dr. Vítor E. Silva Souza

Vitória, ES

2018

Mateus Oliveira Almeida

## **SysMap - Sistema de Mapeamento Sistemático**

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

---

**Prof. Dr. Vítor E. Silva Souza**  
Orientador

---

**Monalessa Perini Barcellos**  
Universidade Federal do Espírito Santo

---

**Cássio Capucho Peçanha**  
Universidade Federal do Espírito Santo

Vitória, ES  
2018

# Agradecimentos

Não é fácil agradecer todas as pessoas que de algum modo, passaram e marcaram minha vida. Por isso primeiro agradeço a todos.

Dediquei este trabalho “in memoriam” a minha mãe (Janete), que não pode me ver ingressar na faculdade, mas foi a que mais me ajudou a conseguir. Dando exemplo de dedicação e comprometimento em tudo que fazia.

Agradeço aos meus tios que abriram as portas de sua casa e me acolheram no início da faculdade. E também a todos os outros que me ajudaram e compartilham da minha felicidade.

Agradeço aos amigos que fiz durante a faculdade, que me ajudaram a enfrentar esse período, Higor, Jordana, Miguel, Thiago e vários outros.

Por último agradeço as professores que conheci e tive o prazer de estudar na UFES, vários dispostos a ajudar e tornar a graduação um espaço de aprendizagem. E claro agradeço o Prof. Dr. Vítor Souza pelo ótima orientação que pode me dar, sempre disposto a ajudar e muito compreensível esse tempo todo.

*“Se há tantas cabeças quantas são as maneiras de pensar, há de haver tantos tipos de amor quantos são os corações.”*

*(Anna Karenina, Liev Tolstói)*

# Resumo

Grupos de pesquisa ligados ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo (PPGI/UFES) poderiam se beneficiar de um sistema de informação para ajudar no mapeamento sistemático da literatura, um método para selecionar publicações que podem ser do interesse de um pesquisador dado um tema.

Foi utilizado o método FrameWeb (SOUZA, 2007) para produzir o projeto. O método sugere a utilização de vários *frameworks* e modelos de projeto para uma representação direta de componentes, com o objetivo de obter uma construção mais rápida e eficiente do sistema.

Nesse sistema utilizou-se vários conceitos e experiências adquiridos durante cursos ministrados na graduação, tais como: Engenharia de Requisitos, Engenharia de Software, Programação Orientada a Objetos, Projeto de Sistema de Software e Desenvolvimento Web.

**Palavras-chaves:** WebApps, Java, JSF, FrameWeb.

# Lista de ilustrações

Figura 1 – Arquitetura padrão para Sistema de Informação <i>Web</i> baseada no padrão arquitetônico <i>Service Layer</i> (FOWLER, 2002). . . . .	17
Figura 2 – Diagrama de Casos de Uso do sistema SysMap . . . . .	22
Figura 3 – Diagrama de Classes do Sistema SysMap. . . . .	24
Figura 4 – Marvin - Arquitetura - Sistema (LIMA, 2015). . . . .	28
Figura 5 – SysMap - Implementação - Pacotes. . . . .	29
Figura 6 – Marvin - Implementação - Páginas Web . . . . .	29
Figura 7 – SysMap - Modelo de Entidades. . . . .	31
Figura 8 – Modelo de Navegação de um CRUD JButler, usado como base para funcionalidades dos cadastros do sistema SysMap. (COSTA, 2015) . . . . .	32
Figura 9 – Modelo de Navegação Importar Pesquisa da Fonte, Verificar Duplicatas e Exportar Dados. . . . .	33
Figura 10 – Modelo de Navegação Convidar Acadêmico. . . . .	34
Figura 11 – Modelo de Navegação Upload PDF. . . . .	34
Figura 12 – Modelo de Aplicação genérica da ferramenta JButler (LIMA, 2015). . . . .	35
Figura 13 – Parte do modelo de Aplicação do Marvin para o módulo <i>SysMap</i> . . . . .	36
Figura 14 – Parte do modelo de Aplicação do Marvin para o módulo <i>SysMap</i> . . . . .	37
Figura 15 – Modelo de Persistência genérico da ferramenta JButler (LIMA, 2015). . . . .	37
Figura 16 – Parte do Modelo de Persistência do Marvin para o módulo <i>SysMap</i> . . . . .	38
Figura 17 – Parte do Modelo de Persistência do Marvin para o módulo <i>SysMap</i> . . . . .	38
Figura 18 – Marvin - Tela Home. . . . .	39
Figura 19 – Marvin - Tela Login. . . . .	39
Figura 20 – Marvin - Tela Listagem SysMaps. . . . .	40
Figura 21 – Marvin - Tela de Cadastro. . . . .	40
Figura 22 – Marvin - Opção de Excluir. . . . .	41
Figura 23 – Marvin - Tela de um SysMap. . . . .	41
Figura 24 – Marvin - CSV gerado . . . . .	42
Figura 25 – Marvin - Convidar Acadêmico . . . . .	42
Figura 26 – Marvin - Convidar Acadêmico . . . . .	43

# Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
CDI	Contexts and Dependency Injection
DAO	Data Access Object
FrameWeb	Framework-based Design Method for Web Engineering
HTML	Htpertext Markup Language
Java EE	Java Platform, Enterprise Edition
JPA	Java Persistence API
JSF	JavaServer Faces
MVC	Model-View-Controller
ORM	Object-relational Mapping
ORM	Object-relational Mapping
SQL	Structured Query Language
UML	Unified Modeling Language



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	Objetivos	10
1.2	Metodologia	11
1.3	Organização da Monografia	12
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>13</b>
2.1	Engenharia de Software	13
2.2	Especificação de Requisitos	14
2.3	Projeto de Sistemas e Implantação	15
2.4	O método FrameWeb	16
<b>3</b>	<b>ESPECIFICAÇÃO DE REQUISITOS</b>	<b>19</b>
3.1	Descrição do Escopo	19
3.2	Diagrama de Casos de Uso	21
3.3	Diagrama de Classes	23
<b>4</b>	<b>PROJETO ARQUITETURAL E IMPLEMENTAÇÃO</b>	<b>26</b>
4.1	Tecnologias Utilizadas	26
4.2	Arquitetura do Sistema	28
4.2.1	Camada de Apresentação	28
4.2.2	Camada de Negócio	29
4.2.3	Camada de Acesso a Dados	30
4.3	Modelos FrameWeb	30
4.3.1	Modelo de Entidades	30
4.3.2	Modelo de Navegação	31
4.3.3	Modelo de Aplicação	34
4.3.4	Modelo de Persistência	35
4.4	Apresentação do Sistema	36
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>44</b>
5.1	Conclusões	44
5.2	Limitações e Trabalhos Futuros	45
	<b>REFERÊNCIAS</b>	<b>46</b>



# 1 Introdução

Grupos de pesquisa ligados ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo (PPGI/UFES) poderiam se beneficiar de um sistema de informação para ajudar no mapeamento sistemático da literatura. Um mapeamento sistemático é uma forma de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma questão de pesquisa, tópico, ou fenômeno de interesse (KITCHENHAM, 2004).

Temos como razões para se fazer um mapeamento: resumir a evidência existente sobre um ou tecnologia; identificar lacunas na pesquisa atual, a fim de sugerir áreas para uma investigação mais profunda e também prover um conhecimento para posicionar novas atividades de pesquisa (KITCHENHAM, 2004).

Como o mapeamento sistemático usa uma metodologia bem definida para identificar, analisar e interpretar todos as evidências disponíveis relacionado a uma questão de pesquisa, ele acaba sendo imparcial e até repetitivo (KITCHENHAM; CHARTERS, 2007). Sendo assim, é possível desenvolver um sistema que possa ajudar na sua elaboração e condução. Tal é o tema central deste trabalho.

Esse sistema foi desenvolvido como um módulo do Marvin,<sup>1</sup> um sistema de informação baseado na Web que agrega ferramentas úteis para o gerenciamento de tarefas de ensino e pesquisa em uma universidade. Tal projeto visa integrar ferramentas desenvolvidas por estudantes de uma forma que pode ser realmente usado por pessoas.

A ideia é que, tendo como base o Marvin, vários subsistemas ou módulos possam ser integrados. Cada um servindo ao seu objetivo, e assim, centralizar processos que possam ser automatizados dentro da universidade.

## 1.1 Objetivos

Esse trabalho tem como objetivo desenvolver um sistema **Web** para auxiliar pesquisadores no processo de mapeamento sistemático, utilizando a experiência e conhecimentos adquiridos no curso de Ciência da Computação. São objetivos específicos deste projeto:

- Levantar e analisar os requisitos do sistema;
- Definir as estruturas e arquiteturas do projeto do sistema;
- Desenvolver um protótipo do sistema de acordo com a estrutura definida no projeto

<sup>1</sup> <<http://github.com/dwws-ufes/Marvin>>; <<http://dev.nemo.inf.ufes.br:8180/Marvin>>.

arquitetural;

- Utilizar a abordagem FrameWeb (SOUZA, 2007), de modo a exercitar suas propostas e ferramentas, bem como sugerir melhorias futuras para as mesmas;
- Exercitar os conceitos aprendidos ao longo do curso de Ciência da Computação, em particular nas disciplinas de Engenharia de Software, Engenharia de Requisitos, Projeto de Sistemas de Software, Programação III, Banco de Dados e Desenvolvimento Web.

## 1.2 Metodologia

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

1. *Revisão Bibliográfica:* Consultar boas práticas de Engenharia de Software e de Requisitos, uso de Banco de Dados Relacional, Programação Orientada a Objetos, Padrões de Projeto de Sistemas aplicados à linguagem de programação Java, entre outros, além do estudo da abordagem FrameWeb (SOUZA, 2007);
2. *Elaboração da Documentação do Sistema:* Nesta etapa, foram definidos os documentos do sistema. Em primeiro lugar, foi elaborado o Documento de Levantamento de Requisitos, apresentando uma descrição geral do minimundo do sistema e definição dos requisitos funcionais e não funcionais, além das regras de negócio. Em seguida, foi elaborado o Documento de Especificação de Requisitos, apresentando os subsistemas, casos de uso, modelo estrutural, modelo dinâmico e glossário do projeto. Por fim, foi elaborado o Documento do Projeto, contendo a arquitetura do software e projeto detalhado de cada um dos seus componentes;
3. *Estudo das Tecnologias:* Nesta etapa, foi necessário o estudo de tecnologias utilizadas para o desenvolvimento do sistema, tais como: linguagem de programação Java; ambiente de desenvolvimento Eclipse Java EE; banco de dados MySQL; JSF, CDI, JPA; PrimeFaces (utilizado para implementação da interface com o usuário); entre outras;
4. *Implementação:* Nesta etapa, o sistema foi implementado. Ele foi codificado a partir da descrição levantada na fase de projeto.
5. *Redação da Monografia:* Nesta etapa, foi realizada a escrita desta monografia. Vale ressaltar que a mesma foi escrita em *LaTeX*<sup>2</sup> utilizando o editor *TexStudio*<sup>3</sup> e o

<sup>2</sup> <<http://www.latex-project.org/>>.

<sup>3</sup> <<http://www.texstudio.org/>>.

template *abnTeX*<sup>4</sup> que atende os requisitos das normas da ABNT (Associação Brasileira de Normas Técnicas) para elaboração de documentos técnicos e científicos brasileiros;

6. *Apresentação do Projeto*: Apresentação final do projeto. O projeto e a ferramenta foram entregues e demonstrados.

## 1.3 Organização da Monografia

Além desta introdução, esta monografia possui outros cinco capítulos.

Capítulo 2: realiza-se um levantamento dos principais temas abordados ao longo deste trabalho que são: Engenharia de Software, Especificação de Requisitos, Projeto de Sistemas e Implementação e o método FrameWeb;

Capítulo 3: apresenta a especificação de requisitos do sistema, descrevendo o minimundo e exibindo os seus diagramas de classes e casos de uso;

Capítulo 4: apresenta a arquitetura do sistema, assim como as partes principais de sua implementação, ilustradas por capturas de telas do sistema;

Capítulo 5: apresenta as conclusões do trabalho, dificuldades encontradas, limitações e propostas de trabalhos futuros.

---

<sup>4</sup> <<http://www.abntex.net.br>>.

## 2 Referencial Teórico

Este capítulo apresenta os principais conceitos teóricos que fundamentaram o desenvolvimento do sistema SysMap e está organizado em 4 seções, sendo as mesmas: Engenharia de Software (2.1), Especificação de Requisitos (2.2), Projeto de Sistemas e Implantação (2.3) e o método FrameWeb (2.4).

### 2.1 Engenharia de Software

A engenharia é aplicação de princípios matemáticos e científicos, experiência, julgamento e bom senso para trazer coisas que beneficiam as pessoas. A Engenharia de Software segue este mesmo raciocínio, tendo como objetivo definir e exercitar processos, métodos, ferramentas e ambientes para construção de software que satisfaça necessidades de cliente e usuário dentro de prazos e custos previsíveis (ASEE, 2000).

O termo *Engenharia* foi usado justamente para associar o “conceito” de *Engenharia* ao desenvolvimento de software, em outras palavras, ter uma abordagem sistemática, disciplinada e quantificada ao desenvolvimento, operação e manutenção de software (IEEE, 1990).

A Engenharia de Software propõe estratégias de desenvolvimento, chamadas modelos de ciclo de vida de desenvolvimento de software ou modelos de processo. Estes modelos de ciclo de vida, como o nome diz, ajudam o desenvolvimento do início ao fim do projeto. Ela se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação (SOMMERVILLE et al., 2003).

Ela é tratada como uma “tecnologia em camadas” (PRESSMAN, 1995). Toda iniciativa de Engenharia de Software deve ser apoiada por um compromisso com a qualidade. Acima da camada da qualidade encontram-se os processos, logo acima, os métodos e, acima destes, as ferramentas. Ao longo da história da Engenharia de Software foram sendo construídas ferramentas computadorizadas para apoiar o desenvolvimento. Essas iniciativas avançaram bastante, mas ainda assim necessitam da intervenção humana. Foram concebidos vários modelos de processos de software e nenhum pode ser considerado o ideal, devido às suas divergências. Entretanto, todos compartilham de atividades fundamentais como especificação, projeto e implementação, validação e evolução (SOMMERVILLE et al., 2003).

## 2.2 Especificação de Requisitos

Resumidamente os requisitos são as funcionalidades do sistema, eles são extraídos após o processo de análise das necessidades do clientes pela coleta de dados.

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de Engenharia de Requisitos (SOMMERVILLE et al., 2003).

As atividades de levantamento de requisitos (também conhecido como elicitación de requisitos) correspondem à etapa de compreensão do problema aplicada ao desenvolvimento de software. O principal objetivo do levantamento de requisitos é que usuário e desenvolvedores tenham a mesma visão do problema a ser resolvido. Nessa etapa, os desenvolvedores, juntamente com os clientes, tentam levantar e definir as necessidades dos futuros usuários do sistema a ser desenvolvido. Essas necessidades são geralmente denominadas requisitos (BEZERRA, 2007).

Temos como um tipo de requisito os requisitos funcionais, que descrevem as funções que um software deverá desempenhar. Esses requisitos dependem do tipo de software que está sendo desenvolvido, dos usuários a que o software se destina e da abordagem geral considerada pela organização ao redigir os requisitos. Eles descrevem as funções do sistema detalhadamente (SOMMERVILLE et al., 2003).

Já os requisitos não funcionais descrevem as qualidades mais globais. A avaliação dos mesmos começa na fase de desenvolvimento e vai até a fase de testes finais do projeto. Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema. Eles podem estar relacionados as propriedades emergentes do sistema, como confiabilidade, tempo de resposta e espaço de armazenamento (SOMMERVILLE et al., 2003).

Nessa fase é necessário um modelo que descreva o sistema em termos de suas tarefas, seu ambiente e como sistema e ambiente estão relacionados. Tal modelo deve ser passível de compreensão tanto por desenvolvedores — analistas, projetistas, programadores e testadores — como pela comunidade usuária — clientes e usuários. Modelos de casos de uso (*use cases*) são uma forma de estruturar essa visão (FALBO, 2017a).

O modelo de casos de uso é um modelo comportamental, mostrando as funções do sistema de maneira estática. Ele é composto de dois artefatos: os diagramas de casos de uso e as descrições de casos de uso. O diagrama de casos de uso descreve graficamente o sistema, seu ambiente e como sistema e ambiente se relacionam. Assim, ele descreve o sistema segundo uma perspectiva externa. As descrições dos casos de uso descrevem o passo

a passo para a realização dos casos de uso e são essencialmente textuais. Elas tratam de como os casos de uso são realizados internamente, complementando os diagramas (FALBO, 2017a).

Também um produto dessa fase do desenvolvimento de um software temos a modelagem conceitual estrutural, que visa modelar de forma computacional os requisitos e os casos de uso identificados anteriormente. Por meio deste modelo, é possível identificar de forma clara os tipos de entidade e os tipos de relacionamentos presentes no sistema. O propósito da modelagem conceitual estrutural, segundo o paradigma orientado a objetos, é definir as classes, atributos e associações que são relevantes para tratar o problema a ser resolvido. Para tal, as seguintes tarefas devem ser realizadas (FALBO, 2017a):

- Identificação de Classes;
- Identificação de Atributos e Associações;
- Especificação de Hierarquias de Generalização/Especialização.

## 2.3 Projeto de Sistemas e Implantação

Depois de definir o que o sistema deverá oferecer, suas funcionalidades e necessidades tem-se a fase de Projeto, que se dá durante a especificação de um software e tem por objetivo definir e especificar uma solução a ser implementada (FALBO, 2017b). Nesta fase se busca definir “como” as funcionalidades devem ser implementadas e todas as tecnologias que deverão ser usadas.

Inicialmente, o projeto é representado em um nível alto de abstração, enfocando a estrutura geral do sistema. Definida a arquitetura, o projeto passa a tratar do detalhamento de seus elementos. Esses refinamentos conduzem a representações de menores níveis de abstração, até se chegar ao projeto de algoritmos e estruturas de dados. Assim, independentemente do paradigma adotado, o processo de projeto envolve as seguintes atividades (FALBO, 2017b):

- Projeto da Arquitetura do Software: Define os elementos estruturais do software e seus relacionamentos;
- Projeto dos Elementos da Arquitetura: Projeta em um maior nível de detalhes cada um dos elementos estruturais definidos na arquitetura, o que envolve a decomposição de módulos em outros módulos menores;
- Projeto Detalhado: Cuida de refinar e detalhar os elementos mais básicos da arquitetura do software, i.e., as interfaces, os procedimentos e as estruturas de dados. Deve-se descrever como se dará a comunicação entre os elementos da arquitetura



(interfaces internas), a comunicação do sistema em desenvolvimento com outros sistemas (interfaces externas) e com as pessoas que vão utilizá-lo (interface com o usuário), bem como se devem projetar detalhes de algoritmos e estruturas de dados.

Ao final da fase de Projeto de *Software*, tem-se a arquitetura na qual o sistema será montado juntamente com o projeto dos componentes do sistema. Esses componentes, como proposto por FOWLER (2002), podem ser classificados da seguinte forma:

- Interface com o Usuário: componente responsável por gerenciar todo o contato do usuário com o sistema, seja para receber dados quanto para apresentar dados;
- Lógica de Negócio: encarregado de aplicar todas as regras de negócios e requisitos do sistema. É uma camada intermediária de um sistema, que recebe os dados vindo da interface com o usuário e após aplicada a lógica de negócio repassa os dados para a camada seguinte;
- Persistência de Dados: guarda os dados de forma a ser possível recuperá-los quando necessário em um sistema de banco de dados integrando com o sistema.

## 2.4 O método FrameWeb

O FrameWeb é um método para projetar *WebApps*, onde se assume o uso de certos tipos de *frameworks* durante o processo de software. Ele define uma arquitetura básica para *WebApps* e propõe modelos mais próximos sua implementação usando esses *frameworks* (SOUZA; FALBO, 2007). Essa arquitetura é baseada no padrão Camada de Serviço (FOWLER, 2002). Tal arquitetura é organizada em camadas como na Figura 1 e descritas a seguir (MARTINS, 2016):

- Camada de Apresentação (*Presentation Tier*): que diz respeito às interfaces gráficas do usuário. Dividida em Visão (*View*), que contém páginas da Web, imagens, modelos de layout e outros arquivos relacionados exclusivamente com a exposição de informações para o usuário; e Controle (*Control*), que compreende classes de ação e outros arquivos relacionados ao *framework* MVC;
- Camada de Negócio (*Business Tier*): onde a lógica de negócio é implementada. Dividida em Lógica de Domínio (*Domain*), que inclui os conceitos de domínio de negócios identificados e modelados por diagramas de classes durante a Análise e refinados durante o Projeto; e Lógica de Aplicação (*Application*), que implementa os casos de uso definidos na especificação de requisitos;
- Camada de Acesso a Dados (*Data Access Tier*): que é responsável por armazenar os

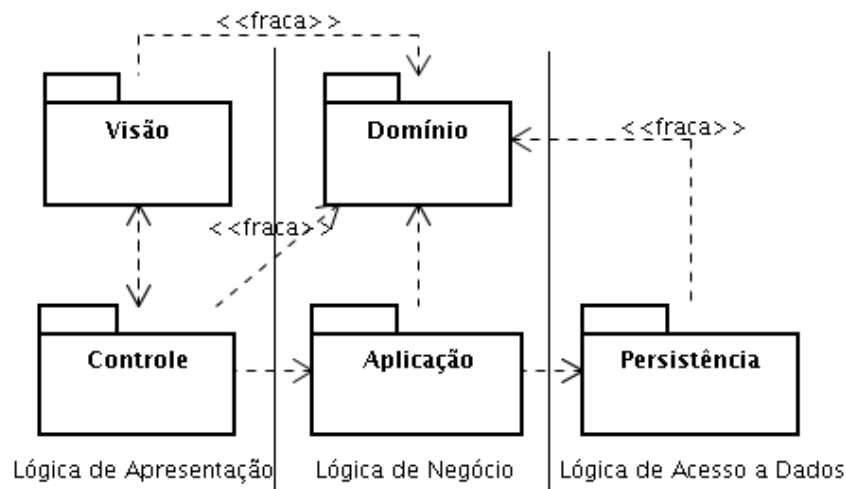


Figura 1 – Arquitetura padrão para Sistema de Informação Web baseada no padrão arquitetônico *Service Layer* (FOWLER, 2002).

objetos persistentes em mídia de longa duração, como por exemplo com o uso de bancos de dados ou arquivos.

Sendo um método para a fase de Projeto, ele não prescreve um processo de software completo. Além da definição de uma arquitetura padrão apresentada acima, o método propõe um conjunto de modelos de projeto que trazem conceitos utilizados pelos *frameworks* para esta fase do processo, deixando assim os diagramas mais próximos da implementação (SOUZA, 2007).

O método FrameWeb está focado na fase de projeto e originalmente definia uma linguagem de modelagem baseada em extensões leves (*lightweight extensions*) ao metamodelo da UML, com a finalidade de criar um perfil UML para ser utilizado pelo método e que serve para a construção de diagramas.

Entretanto, a evolução do FrameWeb proposta por Martins (2016) define uma linguagem Específica de Domínio (*Domain Specific Language* ou DSL) formal para o FrameWeb, por meio de metamodelos que estendem uma parte do metamodelo da UML. Tais metamodelos especificam a sintaxe abstrata da linguagem, descrevendo os conceitos e as regras de modelagem específicas do método e, conseqüentemente, permitem a criação de ferramentas e mecanismos para que esta linguagem possa ser utilizada de forma correta e direcionada (MATOS, 2017).

Tem-se seguir um resumo dos modelos propostos no FrameWeb, todos eles baseados no diagrama de classes da UML:

- **Modelo de Entidades (*Entity Model*)**, referente ao pacote de Domínio da aplicação, para representar o domínio do problema da aplicação Web e o mapeamento

desses objetos para a persistência;

- **Modelo de Persistência (*Persistence Model*)**, para representar os objetos persistentes utilizando o padrão de projeto DAO (ALUR; MALKS; CRUPI, 2013), referente aos elementos contidos no pacote de Persistência;
- **Modelo de Navegação (*Navigation Model*)**, para representar os diversos componentes contidos no pacote de Visão e Controle, que formam a interface com o usuário, a partir da qual é possível interagir com a lógica de negócio;
- **Modelo de Aplicação (*Application Model*)**, para representar as estruturas de serviço pertencentes ao pacote de Aplicação, que são responsáveis pela implementação dos casos de uso, bem como as dependências de/para componentes de outras camadas (Controle e Persistência).

## 3 Especificação de Requisitos

Neste capítulo, serão abordados alguns resultados da Engenharia de Requisitos para a construção do módulo SysMap para o Marvin. Na Seção 3.1 é apresentado o escopo do projeto; na Seção 3.2 são apresentados diagramas de casos de uso e na Seção 3.3 são apresentados os diagramas de classes. Os requisitos funcionais, requisitos não funcionais e regras de negócio podem ser encontrados no **Documento de Especificação de Requisitos** que está disponível no Apêndice ao final desta monografia.

### 3.1 Descrição do Escopo

Grupos de pesquisa ligados ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo (PPGI/UFES) poderiam se beneficiar de um sistema de informação para ajudar no mapeamento sistemático da literatura (abreviado ao longo deste documento para *SysMap*), um método para selecionar publicações que podem ser do interesse de um pesquisador dado um tema.

Tal sistema, daqui em diante denominado “Sistema SysMap” será desenvolvido como um módulo do Marvin,<sup>1</sup> um sistema de informação baseado na Web que agrega ferramentas úteis para o gerenciamento de tarefas de ensino e pesquisa em uma universidade. Tal projeto visa integrar ferramentas desenvolvidas por estudantes de uma forma que pode ser realmente usado por pessoas.

De maneira geral, podemos descrever o uso do Sistema SysMap por meio dos passos que compõem o protocolo de mapeamento sistemático.

**1º passo:** Para poder acessar o sistema, os acadêmicos terão um pré-cadastro realizado por um administrador do sistema, possuindo: nome, email, login e senha. Acadêmicos também poderão cadastrar outros acadêmicos.

Na primeira etapa um acadêmico cria um novo SysMap com informações gerais do mesmo e se torna o Dono do SysMap, podendo editar e excluir o seu SysMap. Um SysMap possui: nome, objetivo, perguntas de pesquisa; *string* de busca, período, área, tipos de publicação, critérios de inclusão, critérios de exclusão e artigos de controle.

Por sua vez, perguntas de pesquisa possuem: identificador, descrição, justificativa e possíveis respostas. Já critérios possuem: identificador e descrição, enquanto artigos de controle possuem: identificador, título e fonte.

O Dono do SysMap convida outros acadêmicos para colaborar em um determinado

---

<sup>1</sup> <<http://github.com/dwws-ufes/Marvin>>; <<http://dev.nemo.inf.ufes.br:8180/Marvin>>.

SysMap que ele possui, tornando-os Pesquisadores do SysMap para aquele SysMap. Dono do SysMap também podem remover Pesquisadores do SysMap de seu SysMap.

**2º passo:** No segundo passo, um Pesquisador do SysMap pode fazer operações para editar um resultado bruto do SysMap. Um resultado bruto indica uma publicação em sua primeira etapa de processamento, na qual posteriormente serão agrupados e analisados segundos critérios pré selecionados, para então saber se ele é relevante para a área de interesse. Um resultado bruto contém: número, fonte, ano, título, autores, lugar, DOI, resumo e palavras-chave.

Dado um SysMap, o sistema executa um processamento de dados, que foram exportados em formatos específicos (ex.: CSV), esse processamento cria automaticamente vários resultados brutos. Esses dados são obtidos em sites de pesquisas de artigos usando uma *string* de busca. Uma *string* de busca possui termos relacionados ao tópico de pesquisa que sejam comumente usados, concatenados pelo conectivo OU (OR em inglês) e os vários termos são concatenados pelo conectivo E (AND em inglês), formando assim uma conjunção de disjunção (FALBO, s.d.).

**3ª passo:** No terceiro passo, quando indicado pelo Dono do SysMap, o sistema agrupa automaticamente os resultado brutos por título e ano, criando entradas, que possuem, cada uma, um identificador e agrupam um ou mais resultados brutos.

Resultado bruto e entrada não correspondem às publicações em si, a diferença é que o segundo é gerado e corresponde a agrupamento do primeiro e as informações da entrada são a mesma dos resultado(s) bruto(s) que a gerou.

Um passo importante é a validação dos resultados encontrados ate agora, ou seja as entradas. Em um primeiro passo o sistema verifica se todos os artigos de controle foram encontrados, e um segundo passo verifica se todas as entradas pertencem ao mapeamento.

O segundo passo é feito checando se o título, resumo e palavras-chave das publicações (entradas) contém os termos da *string* de busca. Entradas que não tem ao menos um termo para cada disjunção, não pertencem ao mapeamento (provavelmente o resultado de um mecanismo de busca falho na fonte). Sendo uma *string* de busca conjunções de disjunções temos que para pertencer ao SysMap pelo menos um termo de cada disjunção tem que ser achado.

Para uma revisão manual das entradas encontradas o Dono do SysMap divide as mesmas entre os Pesquisadores do SysMaps, para que eles possam trabalhar nelas.

**4º passo:** No quarto passo, Um Pesquisador do SysMap filtra uma entrada que tenha sido atribuída a ele. Uma entrada é aceita se ela preenche todos os critérios de inclusão e não satisfaz nenhum critério de exclusão. Nesta etapa, o Pesquisador considera apenas título, resumo e palavras-chave.

Depois de uma entrada ser aceita, o próximo passo exigirá a leitura completa da publicação. O Pesquisador do SysMap faz o upload do PDF da publicação para o próximo passo ficar mais fácil.

No final do quarto passo há uma segunda revisão manual das entradas, o Dono do SysMap novamente divide as mesmas entre os Pesquisadores do SysMap, para que eles possam trabalhar nelas. É feita uma segunda filtragem, agora considerando todo o artigo lido.

**5ª passo:** No quinto passo um Pesquisador do SysMap responde às perguntas de pesquisa para uma entrada que foi atribuída a ele, depois de ler toda a publicação. Neste ponto a primeira avaliação feita no passo anterior pode ser sobrescrita com o conhecimento adquirido após ler toda a publicação.

**Outras funcionalidades:** O sistema também deve permitir o Pesquisador do SysMap visualizar todas as informações produzidas até uma data etapa. Que se dará do seguinte modo:

- Até 1ª etapa - exibir informações gerais do SysMap;
- Até 2ª etapa - visualizar resultado brutos;
- Até 3ª etapa - visualizar entradas e sua validação;
- Até 4ª etapa - visualizar o resultado da primeira filtragem das entradas;
- Até 5ª etapa - visualizar o resultado da segunda filtragem das entradas, comentários e perguntas de pesquisa.

O sistema deve também permitir exportar todas as informações em um formato CSV para ser analisado posteriormente. A exportação de dados segue o mesmo modelo de visualizar as informações produzidas até o momento, sendo a diferença que os dados serão apresentados em CSV.

## 3.2 Diagrama de Casos de Uso

O modelo de casos de uso visa capturar e descrever as funcionalidades que um sistema deve prover para os atores que interagem com o mesmo. Os atores identificados no contexto deste projeto estão descritos na Tabela 1.

Todos os usuários do sistemas são Acadêmicos, que no contexto do sistema são professores pesquisadores. *Acadêmico* é um ator já presente no Marvin, portanto os casos de uso de cadastro (CRUD) do mesmo não serão apresentados.

Tabela 1 – Atores

Ator	Descrição
Acadêmico	Representa um acadêmico que tem como objetivo participar ou visualizar dados de um mapeamento sistemático (SysMap).
Dono do SysMap	Acadêmico que é o responsável por um SysMap, ele irá cadastrar as informações referentes ao mesmo.
Pesquisador do SysMap	Acadêmico convidado pelo Dono do SysMap para que possa contribuir com o mesmo.

Já um Dono do SysMap será acadêmico que criou o SysMap, ele passa a ser o responsável pelo mesmo se tornando seu administrador. Assim, ele pode gerenciar todas as informações do SysMap e convidar Acadêmicos para se tornarem Pesquisadores do SysMap.

Um Pesquisador do SysMap é um Acadêmico que pode visualizar todas as informações do SysMap que ele é colaborador e também realizar a filtragem de entradas para ajudar no mapeamento sistemático.

A Figura 2 apresenta o diagrama de casos de uso do sistema SysMap, que serão descritos abaixo.

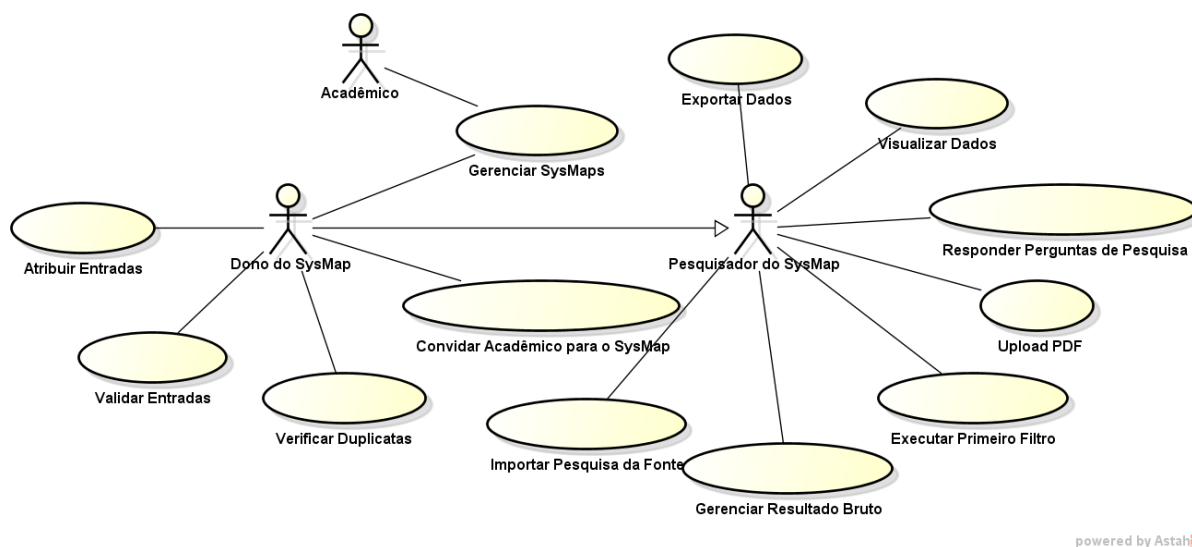


Figura 2 – Diagrama de Casos de Uso do sistema SysMap

Quando um Acadêmico entra no sistema ele tem a opção de criar um novo SysMap, e assim que o fizer ele se torna o Dono do SysMap criado. Essa atribuição está presente no caso de uso **Gerenciar SysMaps**, um caso de uso cadastral.

Pelo caso de uso **Convidar Acadêmico para o SysMap**, o Dono do SysMap pode convidar um Acadêmico para colaboração no mesmo, tornando-se o convidado um Pesquisador do SysMap. Também é atribuição desse caso de uso a remoção de Acadêmicos

convidados.

O caso de uso **Importar Pesquisa da Fonte** é responsável por importar resultados da pesquisa associados a uma fonte. Depois de importados, o caso de uso **Verificar Duplicatas** agrupa os resultados encontrados anteriormente para criar outro grupo de dados sem duplicadas.

Posteriormente, a validação dos dados é feita pelo caso de uso **Validar Entradas**, que verifica se todos os artigos de controle foram achados e todas as entradas pertencem ao mapeamento. Assim o Dono do SysMap indica quais entradas um Pesquisador do SysMap pode trabalhar, sendo essa funcionalidade representada pelo caso de uso **Atribuir Entradas**.

O caso de uso **Gerenciar Resultado Bruto** é um caso de uso cadastral para resultados brutos. Os casos de uso **Executar Primeiro Filtro** e **Responder Perguntas de Pesquisa** são casos de uso que editam as informações de uma entrada atribuída a um Pesquisador. O primeiro tem por objetivo fazer uma primeira filtragem das entradas e o segundo tem por objetivo responder as perguntas de pesquisa para uma entrada e também informar o resultado da segunda filtragem.

O caso de uso **Upload PDF** serve ao propósito de fazer o upload da publicação representada por uma entrada e o caso de uso **Exportar Dados** tem por objetivo exportar todas as informações geradas no mapeamento sistemático em formato CSV. O caso de uso **Visualizar Dados** é contemplado sendo possível visualizar todas as informações geradas até uma etapa do mapeamento sistemático.

Todos os casos de uso de um Pesquisador do SysMaps também podem ser desempenhados pelo Dono do SysMap, mas o contrário não é válido.

### 3.3 Diagrama de Classes

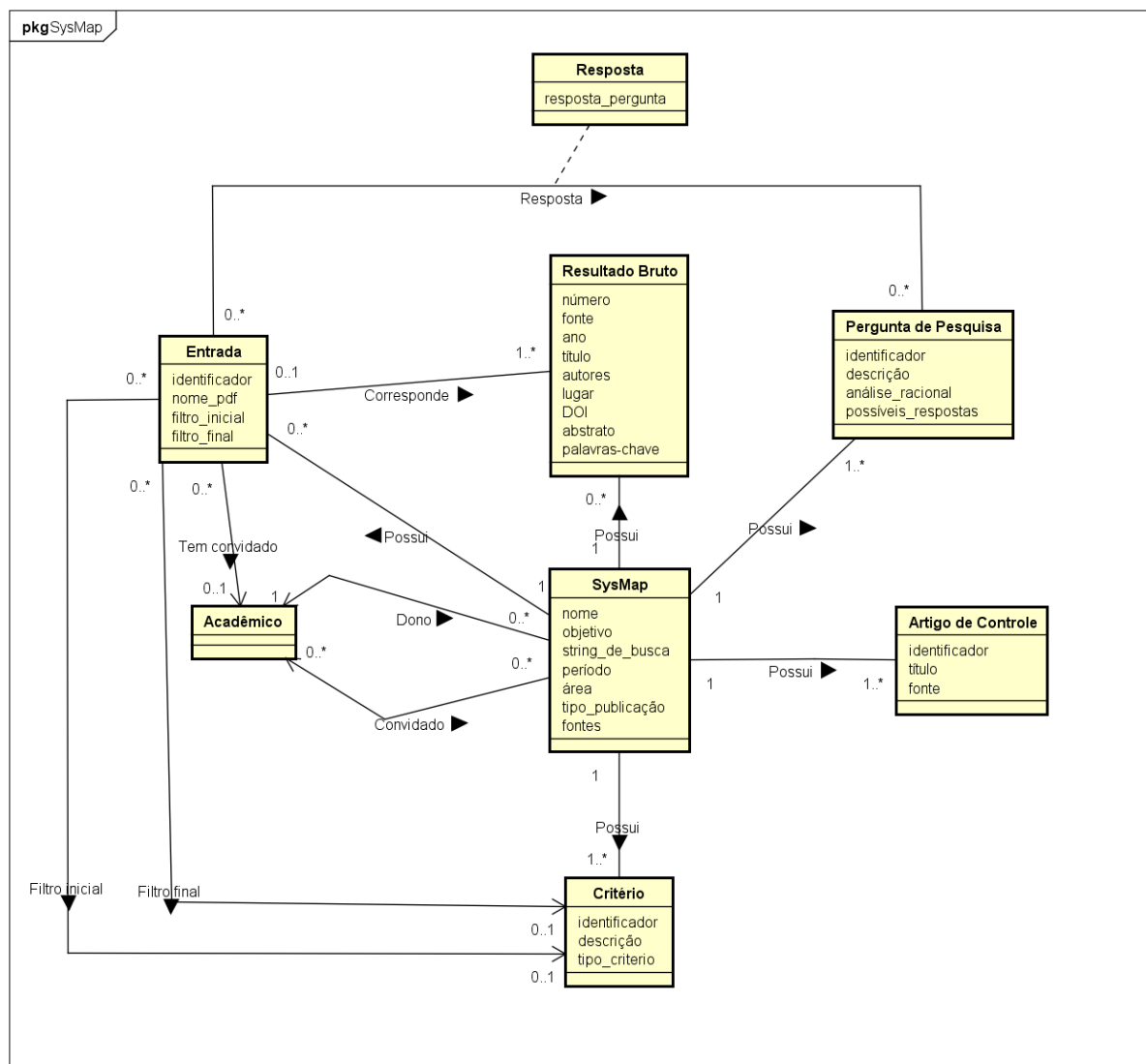
A Figura 3 apresenta o diagrama de classes do SysMap. As classes e suas associações são descritas a seguir.

Podemos notar que a classe mais importante é a classe **SysMap**, pois ela possui a maior quantidade de ligações com as demais classes. Essa classe pode possuir vários elementos das classes: **Entrada**, **Resultado Bruto**, **Pergunta de Pesquisa**, **Artigo de Controle** e **Critério**.

**SysMap** e **Acadêmico** têm duas associações. Uma referente ao Acadêmico que cria o SysMap, se tornando o dono do mesmo e a outra referente aos Acadêmicos que o dono do SysMap pode convidar para conceder acesso ao SysMap e também para edição de Entradas.

A classe **Entrada** é a segunda mais importante. Ela está associada a classe **Aca-**





powered by Astah

Figura 3 – Diagrama de Classes do Sistema SysMap.

**dêmico**, com o intuito de saber qual Acadêmico convidado pode editar essa entrada. Também existe duas associações com a classe **Critério**, pois uma Entrada possui um Critério que corresponde ao resultado da primeira filtragem e um que corresponde ao resultado da segunda filtragem.

Há também a associação da classe **Entrada** com a classe **Pergunta de Pesquisa** que é feita através da classe **Resposta**, que registra a resposta propriamente dita da Entrada referente à Pergunta de Pesquisa. E por fim a associação com a classe **Resultado Bruto** se dá porque uma Entrada corresponde a um ou mais agrupamento de Resultado Bruto.

Além dos elementos visíveis no diagrama de classes, restrições de integridade têm um papel muito importante, pois elas complementam a informações de um modelo. Tais regras são capturadas a partir de modelos conceituais estruturais e são obtidas através das

restrições relativas aos relacionamentos entre elementos de um modelo que não podem ser representadas graficamente. Tais regras devem ser documentadas junto ao modelo conceitual estrutural do sistema (FALBO, 2017a).

Foram encontradas as seguintes restrições de integridades:

1. A associação de um acadêmico a uma entrada ocorre se tanto um quanto outro pertencem ao mesmo SysMap.
2. A associação de um resultado bruto a uma entrada ocorre se tanto um quanto outro pertencem ao mesmo SysMap.
3. A associação de uma entrada e uma pergunta de pesquisa ocorre se tanto uma quanto a outra pertencem ao mesmo SysMap.
4. A associação de uma entrada e um critério, como resultado de uma filtragem, ocorre se tanto uma quanto o outro pertencem ao mesmo SysMap.

## 4 Projeto Arquitetural e Implementação

Logo seguinte às fases de especificação e análise de requisitos, realiza-se a fase de projeto do sistema. O projeto de arquitetura está preocupado com a compreensão de como um sistema deve ser organizado e com a estrutura geral desse sistema. Ele identifica os principais componentes estruturais de um sistema e os relacionamentos entre eles.

O projeto e implementação de software é um estágio do processo no qual um sistema de software executável é desenvolvido. O projeto de software é uma atividade criativa em que se identificam os componentes de software e seus relacionamentos com base nos requisitos do cliente (SOMMERVILLE et al., 2003).

Este capítulo descreve sucintamente o Projeto de Arquitetural e a Implementação do SysMap. Na Seção 4.1 estão descritas as tecnologias utilizadas para desenvolvimento do sistema, na Seção 4.2, a arquitetura do sistema é descrita,

### 4.1 Tecnologias Utilizadas

A linguagem de programação utilizada para implementar o projeto foi Java<sup>TM</sup> EE 7 (Java Enterprise Edition 7), devido à independência da plataforma e por oferecer, de forma nativa, suporte a diversos *frameworks* e APIs. Desta plataforma, foram utilizados os padrões JSF, CDI e JPA, descritos a seguir.

O JSF ou *JavaServer Faces* é uma tecnologia que faz parte do Java<sup>TM</sup> EE e permite criar aplicações Java para a Web utilizando componentes visuais pré-prontos, de forma a não se preocupar no desenvolvimento com Javascript ou HTML. Os componentes adicionados são renderizados em HTML permitindo a abstração do mesmo e a separação entre as camadas de apresentação e de aplicação.

O CDI ou *Contexts and Dependency Injection* para a plataforma Java EE é um dos vários recursos desta plataforma que ajudam a unir a camada da Web e a camada transacional (serviços). É usado para gerenciamento de dependências entre componentes (chamados de *beans*) de uma aplicação (com ou sem o uso de EJBs, os *Enterprise Java Beans*), associando-os a contextos e oferecendo uma série de serviços de infraestrutura à aplicação.

O CDI provê um mecanismo de injeção de dependências entre classes de uma aplicação Java EE, permitindo também o acesso por páginas Web a partir da linguagem de expressões unificada (EL, *Expression Language*). Dependências são associadas a contextos e têm seu ciclo de vida gerenciado automaticamente pelo container. Ao redor deste mecanismo, o CDI oferece uma série de funcionalidades como qualificadores, alternativas, decoradores,

interceptadores e eventos que permitem uma grande flexibilidade no desenvolvimento da aplicação (SOUZA, 2010).

A JPA ou *Java Persistence API* é um framework para persistência em Java, que oferece uma API de mapeamento objeto-relacional (ORM, *Object/Relational Mapping*) e soluções para integrar persistência com sistemas corporativos escaláveis (FARIA, 2013). Entre as vantagens podemos citar (NASCIMENTO, 2015):

- Códigos de acesso a banco de dados com *queries SQL* são custosos de se desenvolver. JPA elimina muito do trabalho e deixa você se concentrar na lógica de negócio, melhorando a produtividade;
- A manutenibilidade de sistemas desenvolvidos com ORM é excelente, pois o mecanismo faz com que menos linhas de código sejam necessárias. Além de facilitar o entendimento, menos linhas de código deixam o sistema mais fácil de ser alterado;
- ORM abstrai sua aplicação do banco de dados e do dialeto SQL. Com JPA, você pode desenvolver um sistema usando um banco de dados e colocá-lo em produção usando diversos outros banco de dados, sem precisar alterar códigos-fontes para adequar sintaxe de queries que só funcionam em SGBDs de determinados fornecedores.

Uma biblioteca de componentes de interface com o usuário, chamada *Primefaces*,<sup>1</sup> também foi utilizada. Ela tem como principal função otimizar o tempo gasto no desenvolvimento de componentes Web responsivos, como, por exemplo, tabelas, campos de formulários, botões, etc. Como resultado, a interface se torna mais robusta e agradável para o usuário. Em conjunto foi usado com *Bootstrap*,<sup>2</sup> um *framework* decorador que também ajuda no desenvolvimento de componentes.

Por fim, o *nemo-utils*<sup>3</sup> é uma ferramenta que foi desenvolvida pelo Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO) da Universidade Federal do Espírito Santo (UFES). Ele auxilia no desenvolvimento de aplicações *Web* que usam os *frameworks* JSF, JPA e CDI. A partir de Abril de 2016, no entanto, o projeto mudou de nome (e de repositório) para *JButler*.<sup>4</sup>

O *nemo-utils/JButler* fornece ao usuário uma série de facilidades no que diz respeito a funcionalidades do tipo CRUD (create, read, update, delete). Este framework implementa, genericamente, as operações básicas existentes entre a aplicação e o banco de dados, fazendo com que o desenvolvedor se preocupe apenas em adaptar e indicar quais as entidades de domínio que serão utilizadas e/ou mapeadas, diminuindo, assim, drasticamente o tempo

<sup>1</sup> <<https://www.primefaces.org>>

<sup>2</sup> <<https://getbootstrap.com>>

<sup>3</sup> <<https://github.com/nemo-ufes/nemo-utils>>

<sup>4</sup> <<https://github.com/dwvs-ufes/jbutler/>>

gasto para implementar tais funcionalidades (SALVATORE, 2016).

## 4.2 Arquitetura do Sistema

O sistema Marvin foi concebido inicialmente apenas com o subsistema `core`, que continha as principais funcionalidades. Criado o módulo SysMap, foi preciso criar seu subsistema, seguindo a divisão de subsistemas feita na análise dos requisitos e apresentada no Capítulo 3. No módulo `br.ufes.inf.nemo.marvin.sysmap` estão contidas as funcionalidades do subsistema SysMap.

O módulo SysMap é subdividido em camadas segundo a arquitetura, que pode ser verificada na Figura 4. O sistema é dividido em três camadas, sendo elas: apresentação (Presentation Tier), negócio (Business Tier) e acesso a dados (Data Access Tier).

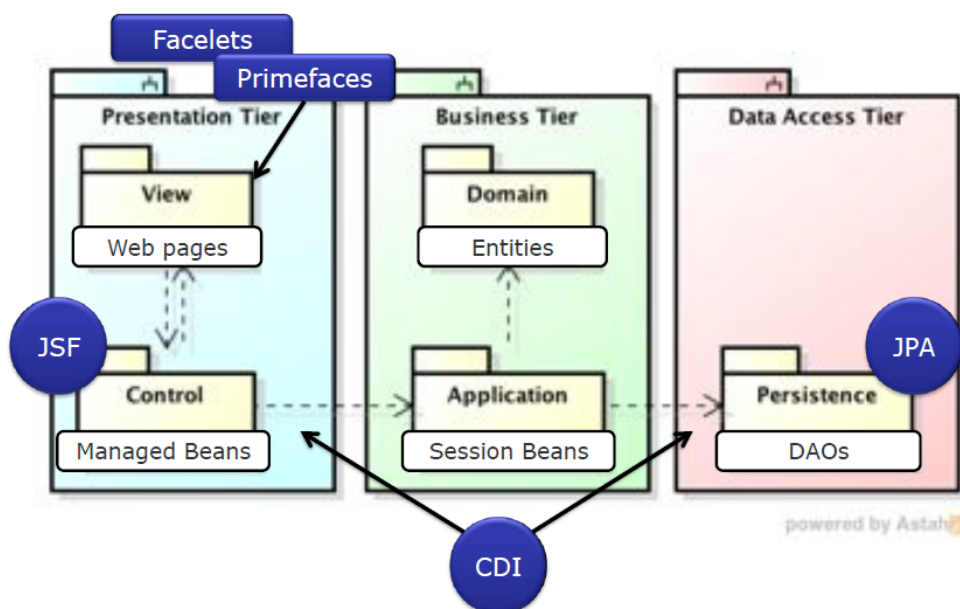


Figura 4 – Marvin - Arquitetura - Sistema (LIMA, 2015).

A Figura 5 exibe os pacotes do sistema SysMap, agrupados pelos módulos principais e pelas camadas da arquitetura, como apresentado na Figura 4. Nas seções seguintes, serão detalhados um pouco mais cada um deles.

### 4.2.1 Camada de Apresentação

A **camada de apresentação** foi subdividida em visão (*View*) e controle (*Control*). A parte da visão é formada pelas páginas Web. A parte de controle contém os controladores que realizam a comunicação entre a interface e a aplicação.

A estrutura Web do sistema Marvin, cujas páginas Web fazem parte da visão da camada de apresentação está organizada conforme a Figura 6. Existe uma pasta raiz

```

> br.ufes.inf.nemo.marvin.core.application
> br.ufes.inf.nemo.marvin.core.controller
> br.ufes.inf.nemo.marvin.core.domain
> br.ufes.inf.nemo.marvin.core.exceptions
> br.ufes.inf.nemo.marvin.core.persistence
> br.ufes.inf.nemo.marvin.core.view
> br.ufes.inf.nemo.marvin.people.domain
> br.ufes.inf.nemo.marvin.people.persistence
> br.ufes.inf.nemo.marvin.sysmap
> br.ufes.inf.nemo.marvin.sysmap.application
> br.ufes.inf.nemo.marvin.sysmap.controller
> br.ufes.inf.nemo.marvin.sysmap.domain
> br.ufes.inf.nemo.marvin.sysmap.persistence

```

Figura 5 – SysMap - Implementação - Pacotes.

chamada *WebContent* e dentro dela, existem as pastas *core* e *sysmap*. Na pasta *sysmap* foram criadas subpastas para salvar as páginas de acordo com os casos de uso do módulo SysMap.

As páginas *index.html* e *index.xhtml*, presentes na pasta *WebContent*, são as páginas iniciais do sistema. A pasta *resource/default* possui um decorador que é usado para definir o *layout* da página e o menu que está sendo utilizado.

```

v WebContent
  v core
    > installSystem
    > login
    > manageAcademic
  > META-INF
  > resources
  v sysmap
    > manageAcademicGuest
    > manageControlArticle
    > manageCriteria
    > manageEntry
    > manageRawResult
    > manageResearchQuestion
    > manageSysMap
  > WEB-INF
  index.html
  index.xhtml
  sysmap.xml

```

Figura 6 – Marvin - Implementação - Páginas Web

#### 4.2.2 Camada de Negócio

A **camada de negócio** foi subdividida em domínio (*Domain*) e aplicação (*Application*). A parte do domínio é formada pelas entidades de negócio (classes de domínio),

enquanto a aplicação contém as validações dos dados e implementação dos casos de uso (funcionalidades do sistema).

O pacote `sysmap.domain` contém a definição das entidades do sistema SysMap. Cada uma dessas entidades está definida em um arquivo `*.java` e, através de anotações, identificam como o mapeamento objeto-relacional irá ocorrer. Através desse mapeamento, o JPA irá gerenciar os objetos no banco de dados automaticamente, sem precisar de nenhuma intervenção do desenvolvedor. É nesse momento que é realizado também o relacionamento entre as classes do sistema utilizando as anotações `@OneToMany`, `@ManyToOne` ou `@ManyToMany` de JPA. Por fim, nesse pacote existe um arquivo para cada classe com o mesmo nome e um “\_” no final (chamada de *static meta-model* ou metamodelo estático), que declara os atributos que poderão ser utilizados para realizar as consultas no banco de dados utilizando os conceitos de Criteria API. Essas consultas serão implementadas na camada de persistência.

Os pacotes `core.application` e `sysmap.application` contém os componentes que fazem a comunicação entre a apresentação (controladores) e a persistência (DAOs), implementando as funcionalidades do sistema descritas em seus casos de uso (cf. Capítulo 3). Faz também as validações das informações antes de chamar os métodos de acesso a dados. Essas validações serão feitas, por exemplo, ao se tentar criar, modificar ou excluir uma informação.

### 4.2.3 Camada de Acesso a Dados

A **camada de acesso a dados** é responsável pela persistência dos dados (*Persistence*). Sua implementação está no pacote `sysmap.persistence`, contendo os DAOs, objetos responsáveis pela comunicação com banco de dados para armazenamento e recuperação de dados das entidades de domínio.

## 4.3 Modelos FrameWeb

Esta seção contém os modelos FrameWeb como descrito na Seção 2.4. Esses modelos também estão divididos nas camadas da arquitetura do sistema, conforme citado na Seção 4.2

### 4.3.1 Modelo de Entidades

A Figura 7 mostra o Modelo de Entidades para o módulo `sysmap`. A classe `AcademicGuest` não tem atributos, mas ela foi criada, ao invés de se mapear uma associação direta entre `SysMap` e `Academic`, porque assim pode-se ter um controle maior de todos os Acadêmicos convidados de um dado SysMap.

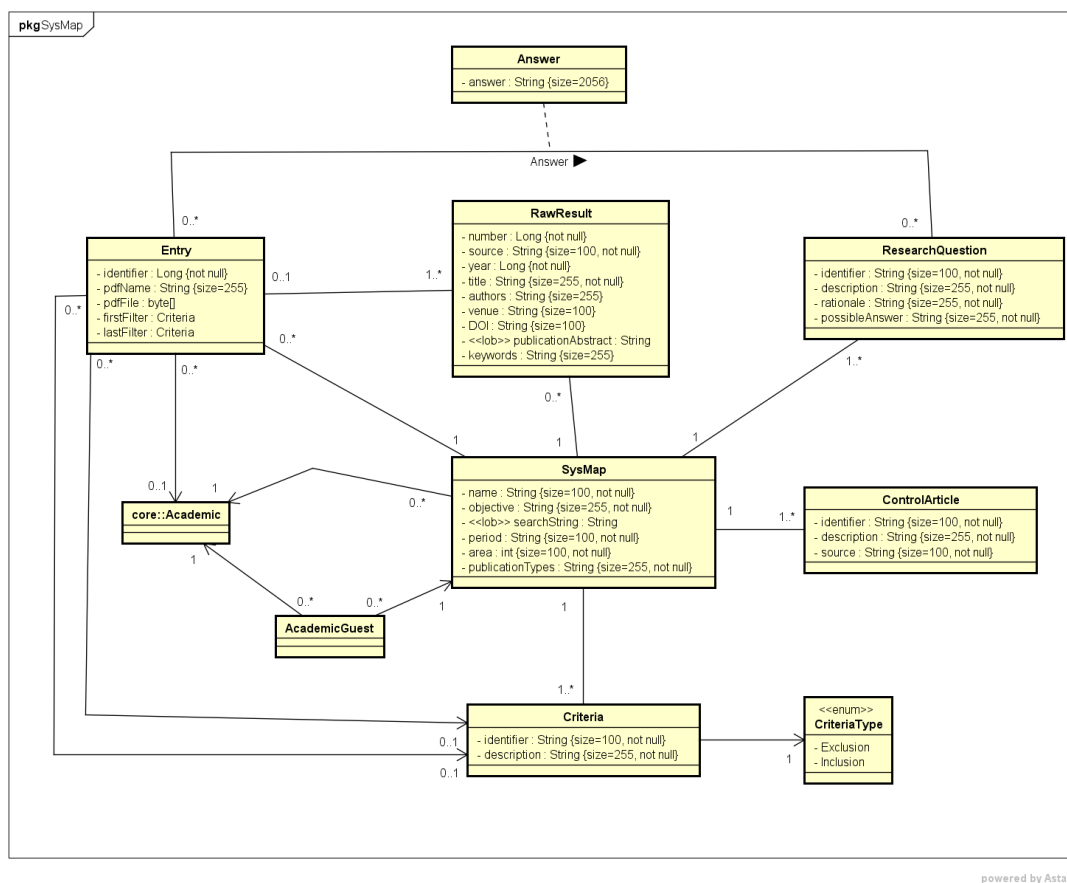


Figura 7 – SysMap - Modelo de Entidades.

Todas as classes de domínio estendem de `PersistentObjectSupport` do *framework* JButler, sendo que essa herança não é mostrada no diagrama com o intuito de não poluí-lo com várias relações de herança.

### 4.3.2 Modelo de Navegação

As funcionalidades criar, visualizar, editar e excluir (abreviadas de CRUD, do inglês *create, read, update and delete*), seguem um mesmo fluxo de execução e de interação com o usuário. Tais funcionalidades são similares para todos os casos de uso cadastrais devido a utilização da ferramenta JButler. Esse fluxo de execução similar é representado na Figura 8 por meio de um modelo de apresentação genérico.

Em relação ao modelo de CRUD da Figura 8, deve existir, para cada entidade, uma página chamada *list.xhtml*, que irá conter a listagem de todos os objetos deste tipo existentes no banco. A exibição destas entidades é feita em seu formulário *manageEntityList* por meio de uma tabela, onde ao selecionar uma entidade da tabela, esta entidade é alocada no atributo *entity* de *ManageEntityControl*, possibilitando então executar ações como atualizar o objeto selecionado (*update*), ler o objeto selecionado (*retrieve*) ou excluir o objeto selecionado (*delete*). Independente de haver uma entidade selecionada na tabela,



pode-se também criar um novo objeto (*create*). Essas ações, ao serem executadas, chamam os seus respectivos métodos do controlador. As relações de dependência representam as formas de retorno para cada ação realizada. Por exemplo, caso se queira atualizar dados de alguma entidade, executa-se o método `update` e o usuário é redirecionado para a página `form.xhtml`, onde poderá fazer as alterações desejadas e, assim, executar o método `save` para salvar tais alterações. Após isso, usuário é redirecionado para a página inicial.

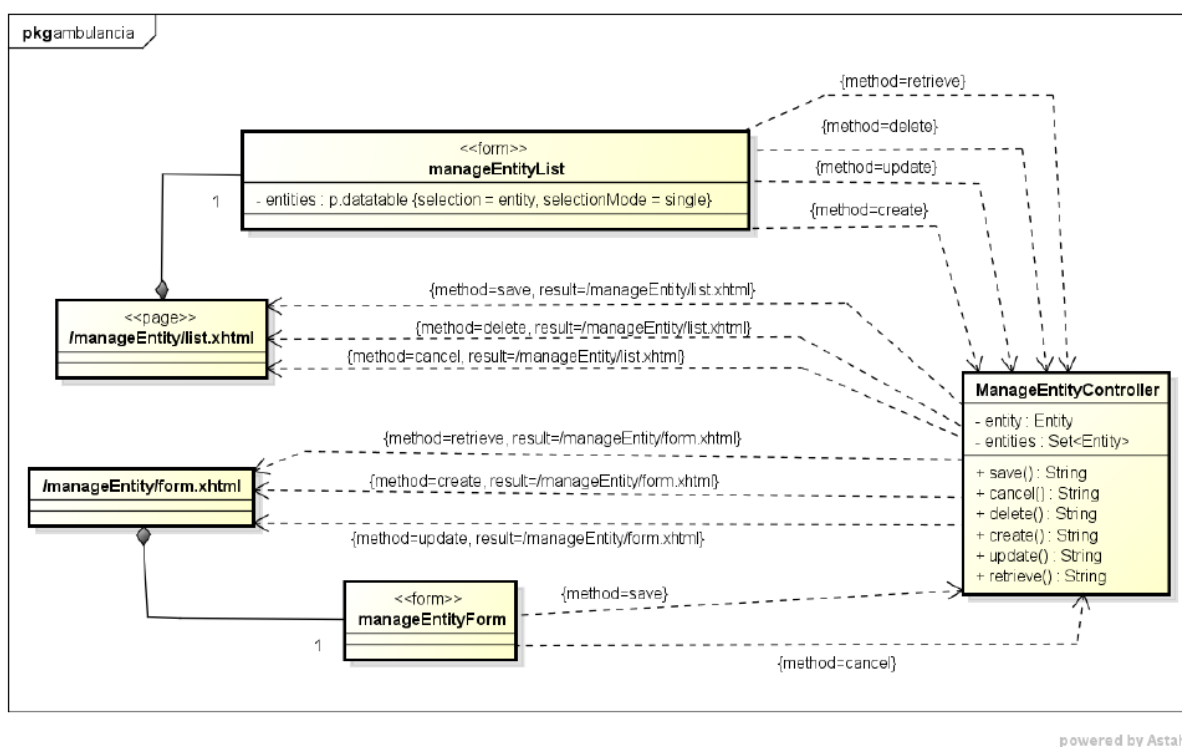


Figura 8 – Modelo de Navegação de um CRUD JButler, usado como base para funcionalidades dos cadastros do sistema SysMap. (COSTA, 2015)

Para os casos de uso que apresentam funções diferentes de apenas as básicas de cadastro, o modelo de navegação da Figura 8 não pode ser aplicado. A Figura 9 apresenta o modelo de navegação para os casos de uso *Importar Pesquisa da Fonte*, *Verificar Duplicatas* e *Exportar Dados*, que foram agrupados pois são executados no mesmo lugar, a partir da página de um elemento SysMap, e em todos eles o fluxo segue para a mesma página. Como podemos ver pelo modelo temos uma página `list.xhtml` que exibe uma listagem de SysMaps, quando um é selecionado o método `retrieve` é disparado, e o fluxo segue para a página `form.xhtml`.

O método `importData()` é o responsável por fazer o *upload* do arquivo que será processado pelo sistema a fim de criar os resultados brutos. Esse *upload* se dá por AJAX<sup>5</sup> e o arquivo não é armazenado depois de ser lido. No caso do PrimeFaces o *upload* de

<sup>5</sup> *Asynchronous Javascript and XML*, tecnologia que permite a comunicação com o servidor e a atualização de partes da página Web aberta sem a necessidade de uma requisição e nova renderização de página completas.

qualquer arquivo é feito por meio de evento e não é necessário fazer o *binding* (ligação entre o valor de um componente visual Web e um atributo na classe controladora). O método `duplicateChecking()` é o responsável por agrupar os resultados brutos e cria as entradas, ele não tem nenhum parâmetro. Por fim, nos métodos de *download* será gerado um arquivo em CSV contendo uma dada informação do SysMap.

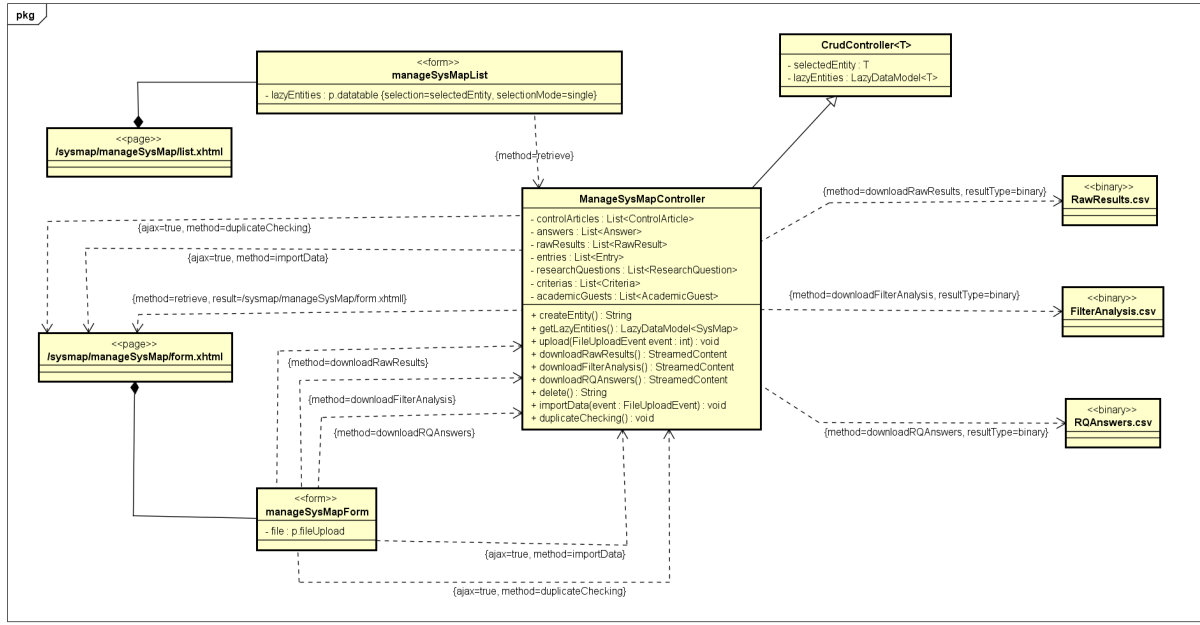


Figura 9 – Modelo de Navegação Importar Pesquisa da Fonte, Verificar Duplicatas e Exportar Dados.

O modelo *Convidar Acadêmico para o SysMap* apresentado na Figura 10 é mais simples. Só se tem a página `list.xhtml` com a listagem dos objetos por linha, cada uma contendo o nome do acadêmico e um botão. Esse botão executa o `addGuest()` quando o acadêmico não está associado ao SysMap ou `removeGuest()` caso ele esteja.

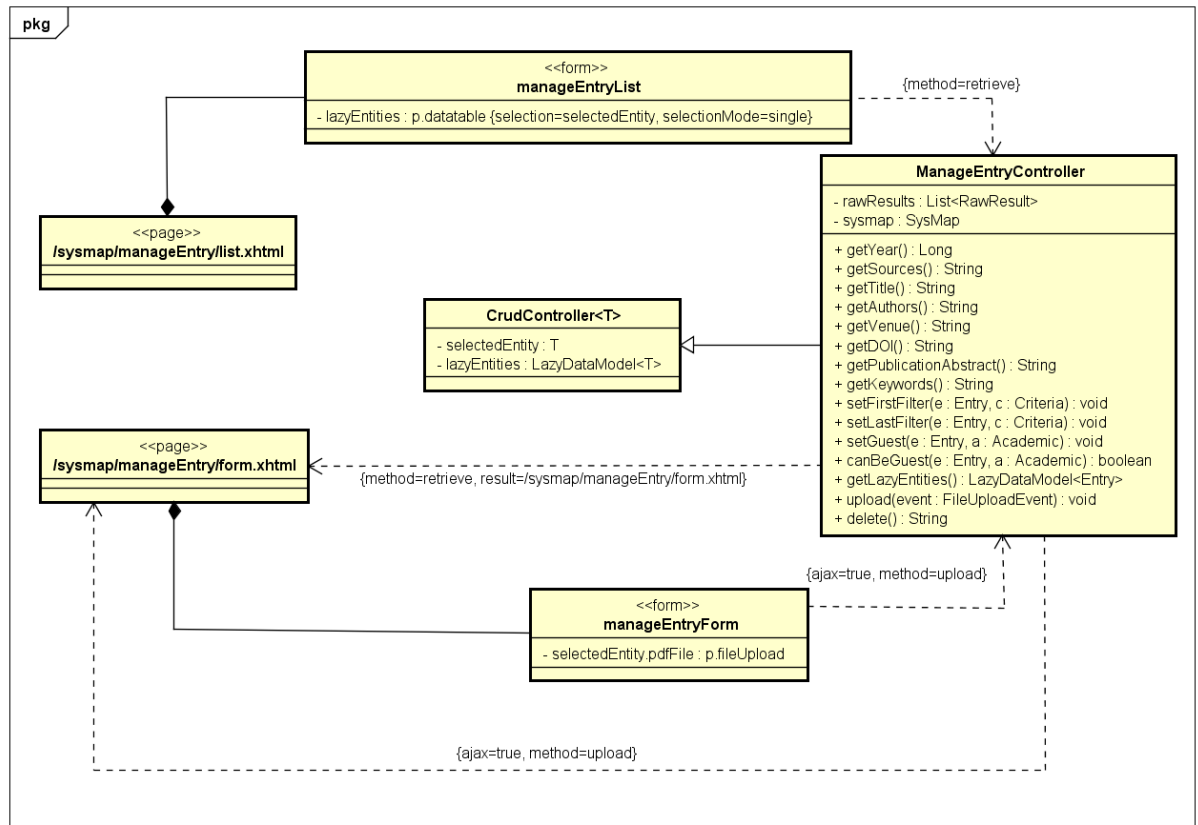
A Figura 11 exibe o modelo de navegação para o fluxo *Upload PDF*. Ele se inicia na página de listagem de uma entrada, e após selecionar uma e ir para a página `form.xhtml` de uma entrada, tem-se a opção de fazer um upload do arquivo PDF referente ao conteúdo que a entrada representa. Essa ação é executada pelo método `upload()`, esse método tem a execução similar ao `importData()` descrito anteriormente, com a diferença que o arquivo PDF depois de ser lido é armazenado.

Os casos de uso *Atribuir Entradas*, *Executar Primeiro Filtro* e *Responder Perguntas de Pesquisa* não estão representados pois o fluxo de execução seria o mesmo da funcionalidade editar que está representada na Figura 8. O caso de uso *Validar Entradas* não foi implementado.



powered by Astah

Figura 10 – Modelo de Navegação Convidar Acadêmico.



powered by Astah

Figura 11 – Modelo de Navegação Upload PDF.

### 4.3.3 Modelo de Aplicação

O **Modelo de Aplicação** é um diagrama de classes da UML que representa as classes de serviço, responsáveis pela codificação dos casos de uso, e suas dependências. Esse diagrama é utilizado para guiar a implementação das classes do pacote *Aplicação* e a configuração das dependências entre os pacotes *Controle*, *Aplicação* e *Persistência*, ou seja, quais classes de ação dependem de quais classes de serviço e quais DAOs são necessários

para que as classes de serviço alcancem seus objetivos (SOUZA, 2007).

As classes de aplicação estendem de `CrudServiceBean` do JButler. Essa classe está representada na Figura 12 de forma genérica. Como nos diagramas anteriores, para não poluí-lo com várias associações, essa herança não é mostrada no diagrama.

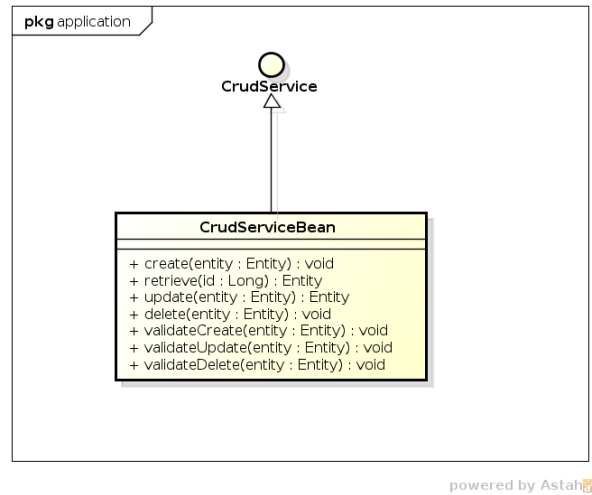


Figura 12 – Modelo de Aplicação genérica da ferramenta JButler (LIMA, 2015).

As figuras 13 e 14 mostram o modelo de aplicação para o módulo *SysMap*, e é possível identificar todas as interfaces e classes de aplicação do mesmo e de qual interface de persistência elas dependem. As ligações entre as classes do diagrama indicam a interdependência entre as três camadas que compõe o sistema através dos prefixos *controller::* e *persistence::* presentes no diagrama que apresentam a conexão entre a camada de aplicação e interface com o usuário e entre a de aplicação e de acesso a dados respectivamente, e essas dependências são satisfeitas pelo CDI automaticamente.

Por exemplo, no caso de uso *Responder Perguntas de Pesquisa*, a transição dos dados ocorre da seguinte forma: o controlador da página de entrada (*ManageEntryController*) envia uma resposta para o sistema. Essa classe possui uma dependência com a *ManageEntryServiceBean* através da interface *ManageEntryService* que contém o método necessário para a execução do caso de uso. Essa associação é satisfeita através da injeção de dependência provida pelo CDI permitindo assim, ao controlador, solicitar a camada de aplicação a execução do caso de uso. Nesta etapa a camada de aplicação, também através da injeção de dependência, repassa os dados para a camada de acesso a dados para que esta possa realizar a persistência da resposta por meio do *EntryDao*.

#### 4.3.4 Modelo de Persistência

O FrameWeb indica a utilização do padrão de projeto DAO para a construção da camada de acesso a dados. Para que não seja necessário repetir em cada interface DAO operações que são comuns a todas elas (ex.: `save()`, `delete()`, `retrieveById()`, etc.),

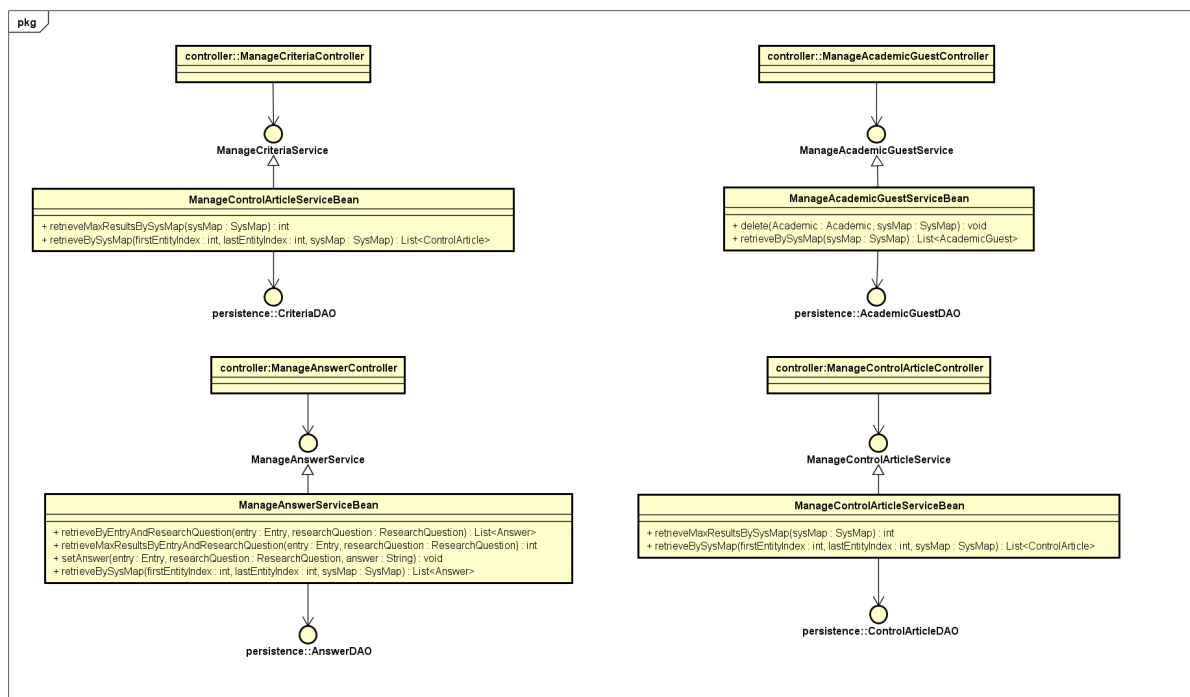


Figura 13 – Parte do modelo de Aplicação do Marvin para o módulo *SysMap*.

podemos apresentar DAOs base que declaram esses métodos — novamente, divididos entre interface implementação. Automaticamente, todas as interfaces DAO de todos os diagramas herdam as definições da interface base, ocorrendo o mesmo com as implementações concretas de cada tecnologia de persistência, sem que isso precise estar explícito no diagrama.

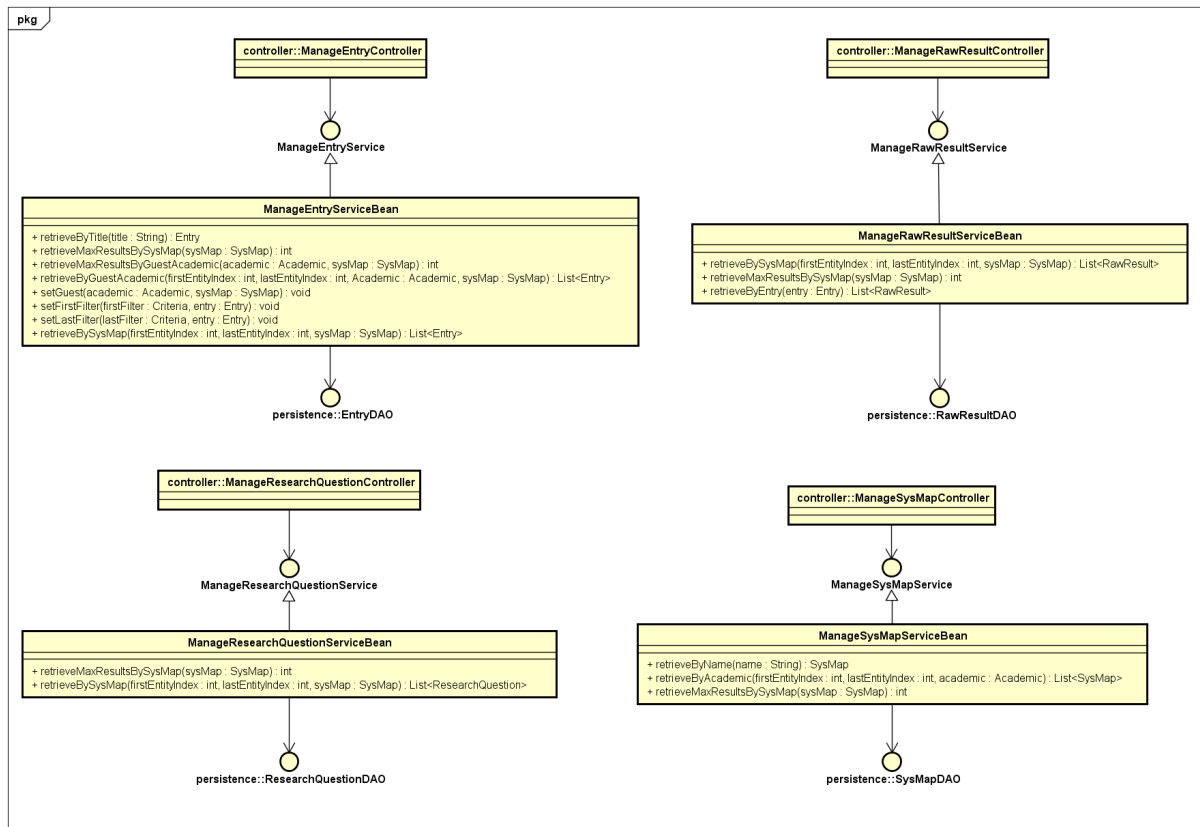
Observa-se que com o nome das classes já é indicado qual tecnologia de persistência foi utilizada, esse sistema de nomenclatura é mais uma sugestão do FrameWeb para simplificar o processo de software. Vale notar também que na Figura 15 está representado o diagrama de persistência genérico provido pelo JButler.

Note-se que a relação de herança entre os DAOs específicos e o DAO base não é representada explicitamente nos diagramas para evitar poluição visual. Esta também é uma recomendação do FrameWeb, ficando, portanto, o desenvolvedor incumbido de derivar essa relação implicitamente ao analisar o modelo.

As figuras 16 e 17 mostram o modelo de persistência para o módulo *SysMap*.

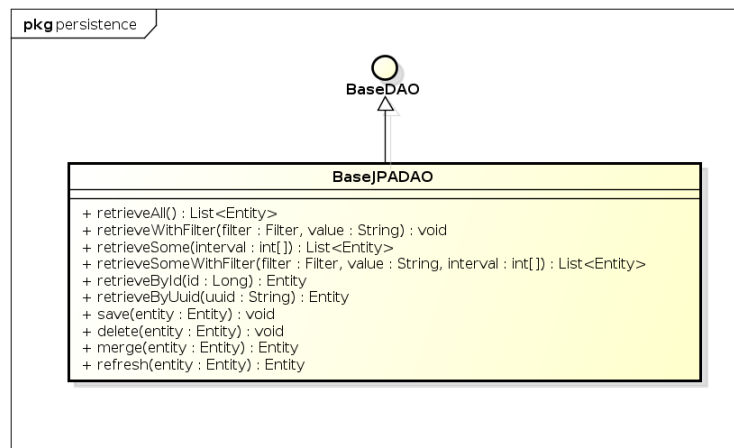
## 4.4 Apresentação do Sistema

Nessa seção será mostrado o resultado final do projeto, indicando suas principais funcionalidades, por meio de capturas de tela. A Figura 18 mostra a tela inicial do sistema. As telas de *Login* e *ManageAcademics* já estavam presentes no sistema Marvin anteriormente. Do lado esquerdo da tela é possível ver o menu contendo as funcionalidades de Gerenciar Acadêmicos e de Login.



powered by Astah

Figura 14 – Parte do modelo de Aplicação do Marvin para o módulo *SysMap*.

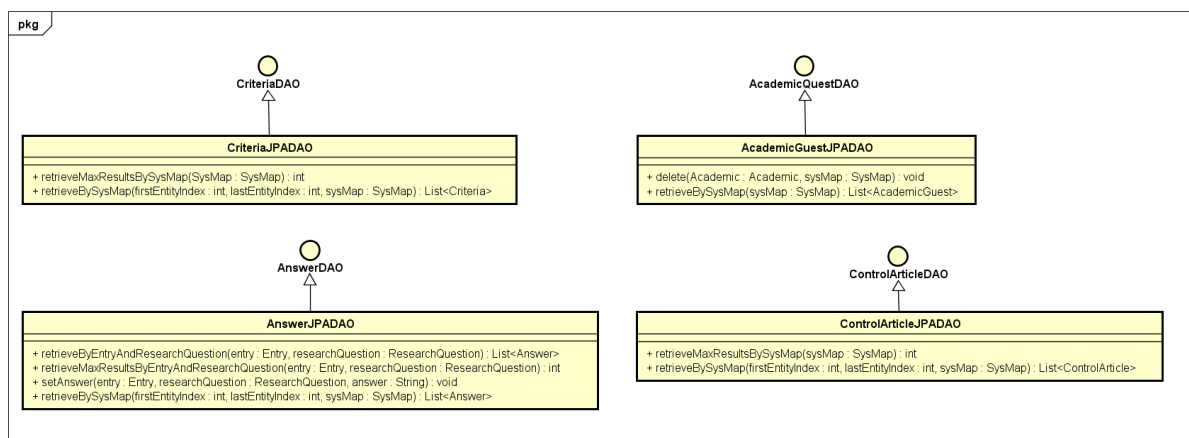


powered by Astah

Figura 15 – Modelo de Persistência genérico da ferramenta JButler (LIMA, 2015).

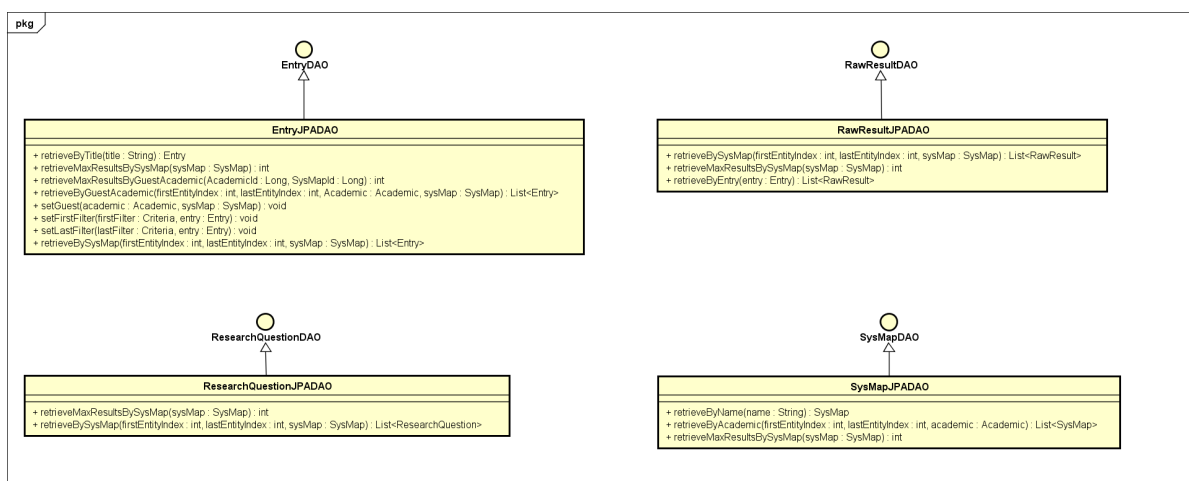
Ao clicar na opção de login, é mostrada a tela de login do sistema, como mostra a Figura 19, com os campos de login e senha. Para efetuar login é necessário email e senha. É feita uma verificação do email para confirmar se é um email válido que está sendo inserido. O campo da senha aceita qualquer caractere alfanumérico.

Após feito o login, a opção Manage SysMap é exibida no menu à esquerda, e após a mesma ser selecionada o sistema exibe a tela exibida na Figura 20. Essa tela mostra



powered by Astah

Figura 16 – Parte do Modelo de Persistência do Marvin para o módulo *SysMap*.



powered by Astah

Figura 17 – Parte do Modelo de Persistência do Marvin para o módulo *SysMap*.

a listagem de SysMaps que um acadêmico tem acesso, sendo ele dono ou convidado. Ao selecionar um item da lista as opções **New** (novo), **View** (visualizar), **Modify** (alterar) e **Delete** (deletar) são exibidas.

Ao se clicar nos botões **New**, **View** ou **Modify**, o usuário será redirecionado para a página de cadastro de um item conforme a Figura 21. Em View, os campos do formulário serão somente leitura. Em Modify os campos do formulário já virão preenchidos com as informações do item selecionado, sendo possível editar as mesmas. E por último, em New, o formulário será exibido em branco, possibilitando a criação de um novo item.

Para se excluir de um item é necessário clicar no botão **Delete**, então será exibido um novo painel para confirmação da exclusão como podemos ver na Figura 22, na parte inferior da tela informando o item que será excluído e os botões para confirmar ou cancelar a exclusão.

Para acessar as funcionalidades de um dado SysMap, tem que se clicar no botão

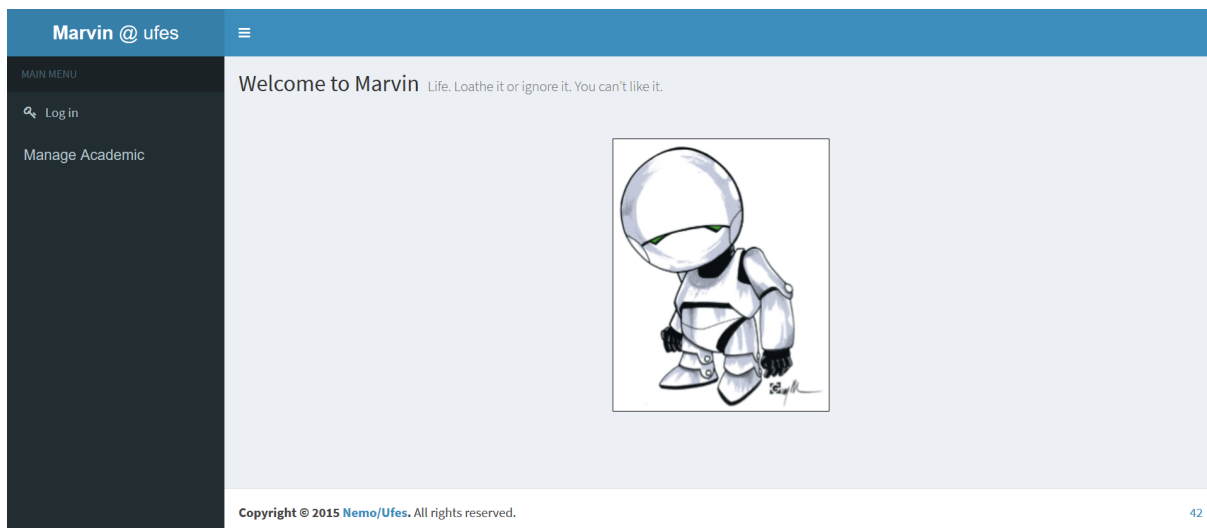


Figura 18 – Marvin - Tela Home.

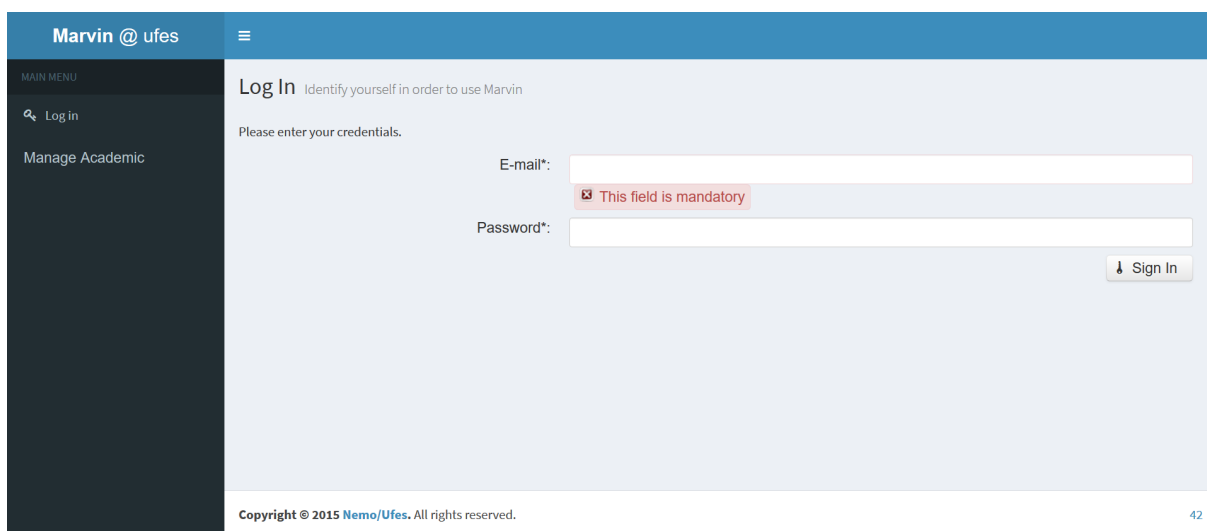


Figura 19 – Marvin - Tela Login.

**View** na tela de listagem de SysMaps, exibido na Figura 20. Para então entrar na página do mesmo com todas as suas funcionalidades, como mostra a Figura 23. O botão **Import Data** tem a funcionalidade de ler um arquivo CSV contendo os resultados da pesquisa feita em uma fonte, para então o sistema processar esses dados criando resultados brutos.

Os botões **Download Raw Results**, **Download Filter Analysis** e **Download RQ Answers** têm a funcionalidade de exportar as informações geradas do mapeamento sistemático no formato CSV. O primeiro exibe todos os resultados brutos, o segundo a análise da filtragem para cada entrada e o último a resposta de pesquisa para as entradas. Um exemplo de dado exportado, gerado pelo **Download Filter Analysis** é mostrado na Figura 24.

O botão **Verify Duplicates** tem a funcionalidade de agrupar os resultados brutos do SysMap por título e ano, gerando assim as entradas. O botão **Manage Academic**



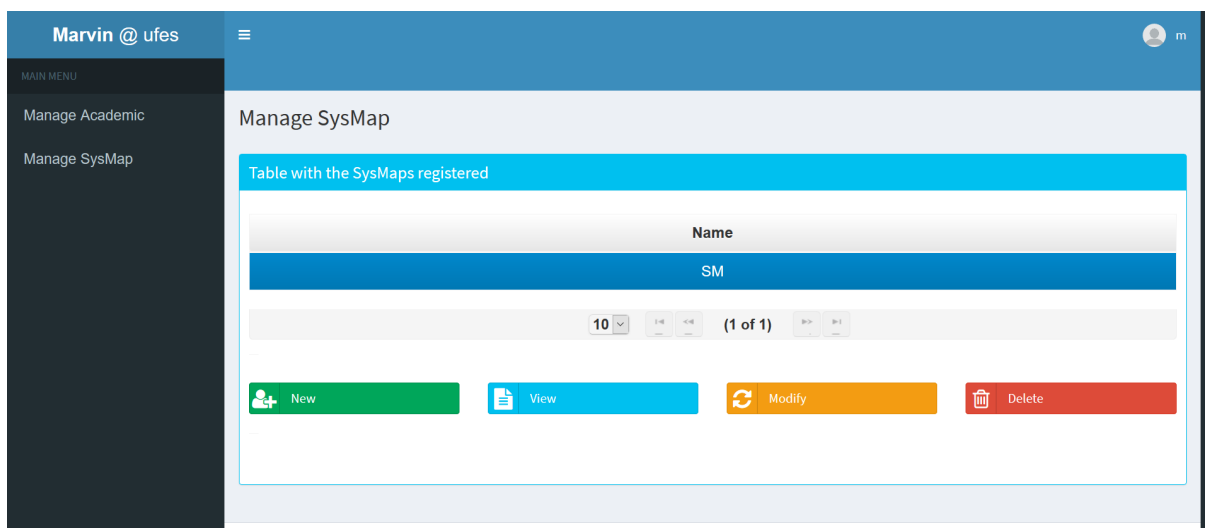


Figura 20 – Marvin - Tela Listagem SysMaps.

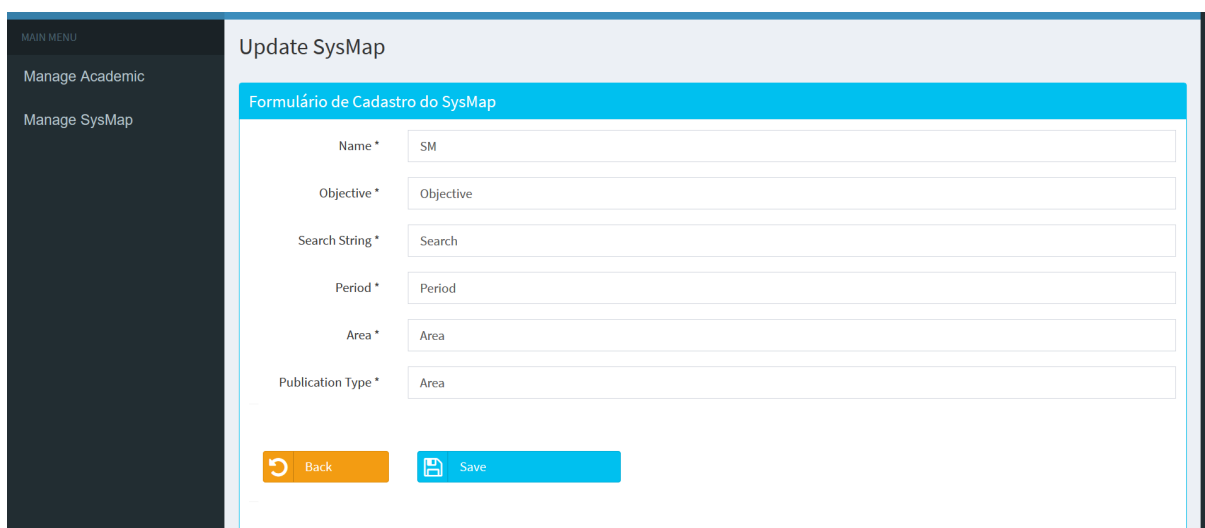


Figura 21 – Marvin - Tela de Cadastro.

**Guest** leva o usuário para a tela onde se adiciona ou remove um acadêmico como convidado de um dada SysMap, como mostrado na Figura 25.

Os botões **Manage Control Article**, **Manage Reserach Question**, **Manage Criteria** e **Manage Raw Result** têm o funcionamento de CRUD como o apresentado pelas figuras 20 e 21.

O botão **Manage Entry** leva para a tela de listagem de entradas que, diferentemente de outras telas de cadastro, mostra apenas as entradas de um único SysMap. Esta tela verifica ainda se o usuário é dono do SysMap, mostrando-lhe todas as entradas, ou se é um acadêmico convidado, mostrando-lhe só as entradas que lhe foram atribuídas. Não existe a opção para criar uma nova entrada porque ela é somente criada pelo agrupamento de resultados brutos. Na tela de edição os campos que vieram dos resultados brutos não são editáveis, sendo editável apenas a escolha dos filtros, a escolha do acadêmico que pode

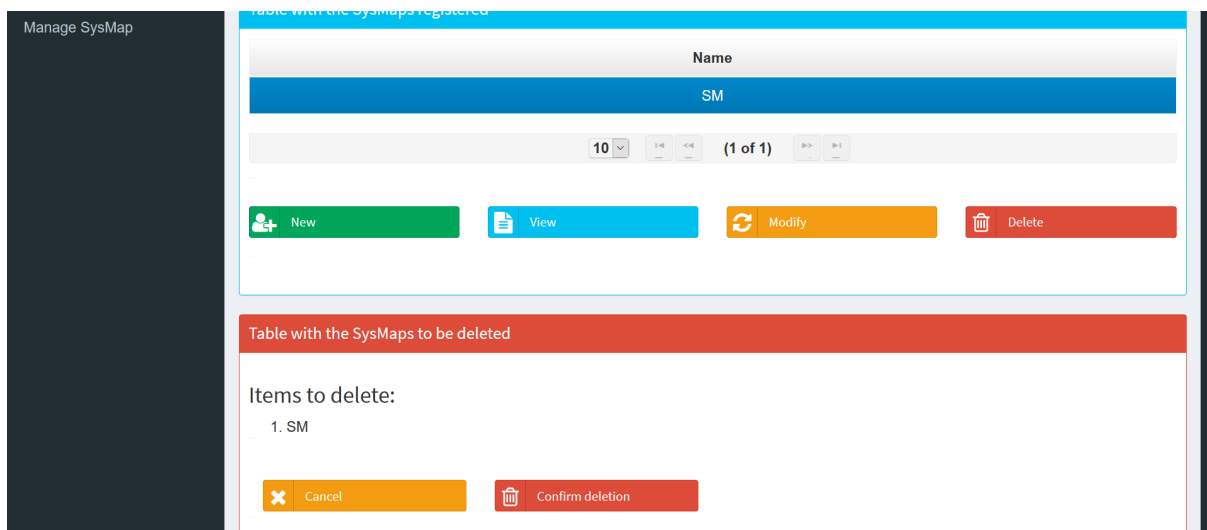


Figura 22 – Marvin - Opção de Excluir.

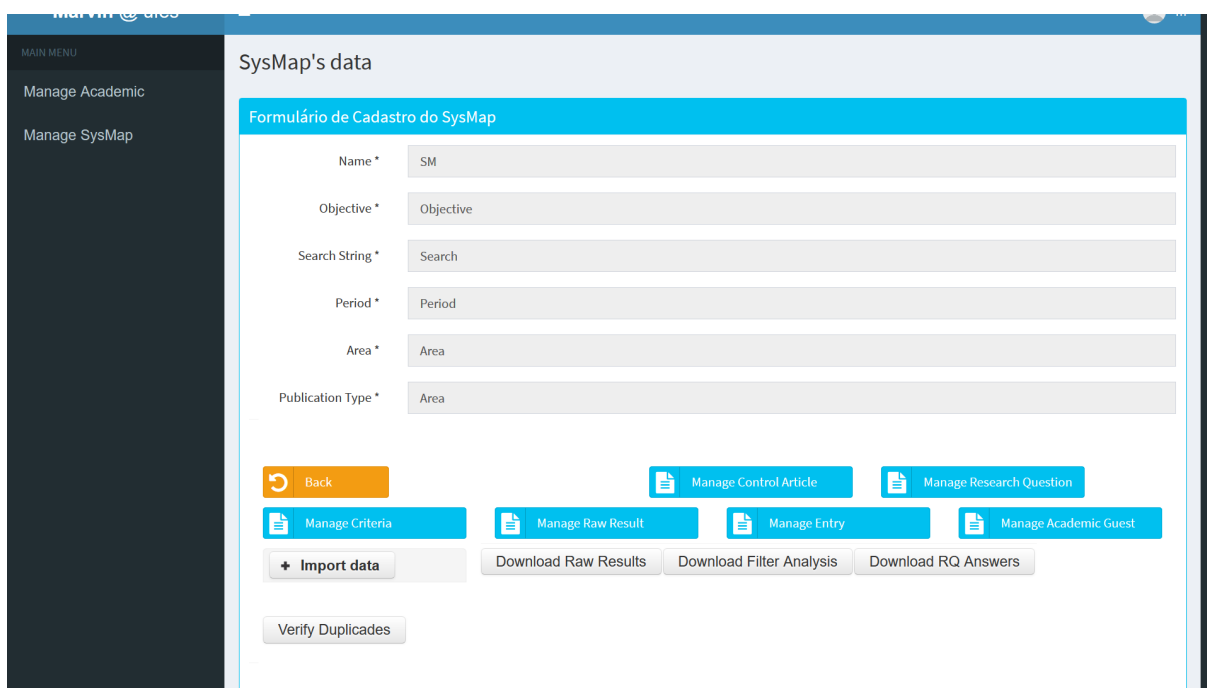


Figura 23 – Marvin - Tela de um SysMap.

editar a entrada e a resposta da pergunta de pesquisa. O botão **Upload Article** é usado para ser feito o *upload* do arquivo relativo a entrada e o botão **Download Article** faz o *download* do mesmo, quando presente. A Figura 26 mostra a tela de edição de uma entrada.

	1	2	3	4	5	6
1	#	Year	Sources	Title	1st Filter Analysis	2nd Filter Analysis
2	1	2011	IEEE	Reasoning about adaptive requirements for self-adaptive systems at runtime	EC1	
3	2	2010	IEEE	Continuous adaptive requirements engineering: An architecture for self-adaptive service-based applications		~IC1
4	3	2013	IEEE	Runtime goal models: Keynote		
5	4	2010	IEEE	Run-time monitoring of system performance: A goal-oriented and system architecture simulation approach		
6	5	2011	IEEE	Dealing with softgoals at runtime: A fuzzy logic approach		
7	6	2012	IEEE	Stateful requirements monitoring for self-repairing socio-technical systems		
8	7	2010	IEEE	Requirements reflection: requirements as runtime entities		
9	8	1998	IEEE	Reconciling system requirements and runtime behavior		
10	9	2014	IEEE	Taking goal models downstream: A systematic roadmap		
11	10	2005	IEEE	Monitoring with behavior view diagrams for debugging		

Figura 24 – Marvin - CSV gerado

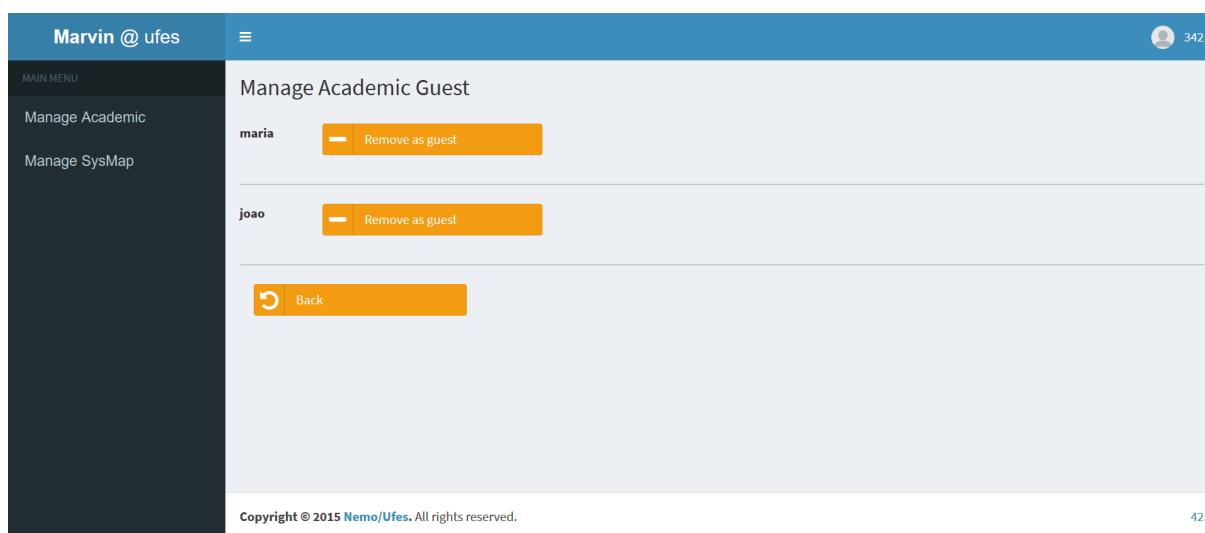


Figura 25 – Marvin - Convidar Acadêmico

**Marvin @ ufes** m

MAIN MENU

- Manage Academic
- Manage SysMap

### Update Entry

#### Formulário de Cadastro de Entry

Source \*

Year \*

Title \*

Authors

Venue

DOI

Abstract

Keywords

First Filter

Last Filter

---

Select Quest Academic

---

Research Questions: RQ1 Description

Research Question's Possible Answer: resposta

---

---

Figura 26 – Marvin - Convidar Acadêmico

## 5 Considerações Finais

Este capítulo apresenta as conclusões do trabalho realizado, mostrando suas contribuições. Por fim, são apresentadas suas limitações e perspectivas de trabalhos futuros.

### 5.1 Conclusões

Devido ao mapeamento sistemático ser um método para construir esquemas de classificação e estruturar uma determinada área de interesse, ele é bastante importante principalmente quando se está desenvolvendo uma tese de mestrado ou doutorado. Pois ele tem como objetivo mostrar a frequência/quantidade de publicações por categoria dentro de um determinado esquema. Desta forma, a cobertura de uma determinada área de pesquisa ou tópico pode ser determinada.

Com a motivação de criar um sistema que pudesse auxiliar os estudantes e professores do PPGI/UFES, apareceu a oportunidade de desenvolvê-lo como parte do projeto de graduação usando o método *FrameWeb*, a fim de facilitar o desenvolvimento.

A maioria dos objetivos identificados no Capítulo 1 foram completamente alcançados, com exceção de dois deles, satisfeitos apenas parcialmente: (1) algumas partes do sistema não foram implementadas, pois o escopo se mostrou bem extenso para um projeto de graduação; e (2) não foi utilizada ferramenta *FrameWeb Editor* (CAMPOS; SOUZA, 2017), ferramenta CASE de apoio ao método *FrameWeb*, pois poderia tomar tempo da implementação para dominar seu uso.

Uma dificuldade inicial foi a curva de aprendizagem de todas as ferramentas utilizadas: JavaScript, Java EE e suas tecnologias (JSF, CDI, JPA) aplicadas com o *FrameWeb*. Isso é resolvido lendo sobre tais tecnologias, vendo exemplos, desenvolvendo o sistema e também após vários erros e acertos. Em especial, *FrameWeb* não é um método fácil de ser consolidado em tão pouco tempo, principalmente no que diz respeito à construção dos modelos, em particular o Modelo de Navegação, que por vezes se mostra difícil de ser compreendido, por querer representar várias informações em um único diagrama.

No entanto, depois do aprendizado o *FrameWeb* se mostrou bastante útil, dado que sua modelagem é bem próxima de como o sistema irá se comportar na Web a partir da integração dos *frameworks* utilizados. Isso acelera o desenvolvimento já que os componentes necessários já estão especificados nos modelos.

Uma outra dificuldade é que, sendo mais tecnológico, o curso de Ciência da Computação não prepara suficientemente o aluno para ter que escrever. Isso acaba sendo uma dificuldade comum quando os alunos chegam no final do curso e se deparam com a

necessidade de redigir um documento bem estruturado como uma monografia.

Ao final do projeto é clara a importância de documentar um sistema. Ela ajuda a manter um registro de todas as fases do projeto, de como cada item foi realizado, quais decisões foram tomadas, como e porquê. Uma documentação completa auxilia o desenvolvedor a ter uma visão mais completa do sistema e com isso tomar decisões mais acertadas, facilitando também no caso de manutenção ou expansão do sistema.

## 5.2 Limitações e Trabalhos Futuros

Quando um software chega no final de seu ciclo de vida, limitações são encontradas, o que dá a margem para mudanças futuras, como adição de novas funcionalidades e correção de erros.

Algumas limitações foram encontradas, assim como melhorias, que podem ser aplicadas ao sistema a fim de tornar este um software mais completo. Dentre as possíveis, vale citar:

- Implementação do caso de uso Validar Entradas;
- Integrar o SysMap com todos os outros módulos do Marvin desenvolvidos até o momento;
- Informar quando um arquivo CSV, que vai ser lido, não tem a formatação adequada;
- Integrar todos os relatórios em um só, mais completo;
- Implementar uma forma de distribuição de entradas de maneira automática entre os acadêmicos convidados;
- Melhorias na interface com o usuário.

# Referências

- ALUR, D.; MALKS, D.; CRUPI, J. *Core J2EE Patterns (paperback): Best Practices and Design Strategies*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2013. 462 p. ISBN 9780133807462. Citado na página 18.
- ASEE. *American Society for Engineering Education*. 2000. Citado na página 13.
- BEZERRA, E. *Princípios de Análise e Projetos de Sistemas com UML*. [S.l.]: Campos Rio de Janeiro, 2007. Citado na página 14.
- CAMPOS, S. L.; SOUZA, V. E. S. FrameWeb Editor: Uma Ferramenta CASE para suporte ao Método FrameWeb. In: *Anais do 16º Workshop de Ferramentas e Aplicações, 23º Simpósio Brasileiro de Sistemas Multimedia e Web*. Gramado, RS, Brazil: SBC, 2017. p. 199–203. Citado na página 44.
- COSTA, T. M. *Resgate - Despacho de ambulância automatizado*. Monografia (Projeto de Graduação) — Universidade Federal do Espírito Santo, 2015. Citado 2 vezes nas páginas 6 e 32.
- FALBO, R. A. *Engenharia de Requisitos*. [s.n.], 2017. 178 p. Disponível em: <[https://inf.ufes.br/~falbo/files/ER/Notas\\_Aula\\_Engenharia\\_Requisitos.pdf](https://inf.ufes.br/~falbo/files/ER/Notas_Aula_Engenharia_Requisitos.pdf)>. Citado 3 vezes nas páginas 14, 15 e 25.
- FALBO, R. A. *Projeto de Sistemas*. [s.n.], 2017. 135 p. Disponível em: <[https://inf.ufes.br/~falbo/files/PSS/Notas\\_Aula\\_Projeto\\_Sistemas\\_2017.pdf](https://inf.ufes.br/~falbo/files/PSS/Notas_Aula_Projeto_Sistemas_2017.pdf)>. Citado na página 15.
- FALBO, R. de A. *Mapeamento Sistemático*. s.d. Disponível em: <[https://inf.ufes.br/~falbo/files/MP/TP/Sobre\\_MS.pdf](https://inf.ufes.br/~falbo/files/MP/TP/Sobre_MS.pdf)>. Citado na página 20.
- FARIA, T. *Java EE 7 com JSF, PrimeFaces e CDI*. [S.l.: s.n.], 2013. Citado na página 27.
- FOWLER, M. *Patterns of Enterprise Application Architecture*. 2. ed.. ed. [S.l.]: Addison-Wesley, 2002. Citado 3 vezes nas páginas 6, 16 e 17.
- IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. 1990. Citado na página 13.
- KITCHENHAM, B. *Procedures for Performing Systematic Reviews*. [S.l.], 2004. Citado na página 10.
- KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. [S.l.], 2007. Citado na página 10.
- LIMA, L. V. F. *SAP - Sistema de Apoio ao Professor*. [S.l.], 2015. Citado 4 vezes nas páginas 6, 28, 35 e 37.
- MARTINS, B. F. S. *Evolução do Método FrameWeb para o Projeto de Sistemas de Informação Web Utilizando uma Abordagem Dirigida a Modelos*. [S.l.], 2016. Citado 2 vezes nas páginas 16 e 17.

- MATOS, R. P. de. *Aplicação do método FrameWeb no desenvolvimento de um sistema de informação usando os frameworks Spring MVC e Vaadin*. Monografia (Projeto de Graduação) — Universidade Federal do Espírito Santo, 2017. Citado na página 17.
- NASCIMENTO, B. M. do. *SAE - Sistema de Acompanhamento de Egressos*. [S.l.], 2015. Citado na página 27.
- PRESSMAN, R. S. *Engenharia de Software*. [S.l.]: Makron Books São Paulo, 1995. Citado na página 13.
- SALVATORE, T. R. *AlocaWeb e BibLattes - Módulos do sistema Marvin*. [S.l.], 2016. Citado na página 28.
- SOMMERVILLE, I. et al. *Engenharia de software*. [S.l.]: Addison Wesley São Paulo, 2003. v. 6. Citado 3 vezes nas páginas 13, 14 e 26.
- SOUZA, V. E. S. *FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web*. 2007. Disponível em: <[https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/frameweb\\_\\_um\\_metodo\\_baseado\\_em\\_frameworks\\_para\\_o\\_projeto\\_de\\_sistemas\\_de\\_informacao\\_web\\_2007.pdf](https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/frameweb__um_metodo_baseado_em_frameworks_para_o_projeto_de_sistemas_de_informacao_web_2007.pdf)>. Citado 4 vezes nas páginas 5, 11, 17 e 35.
- SOUZA, V. e. S. *Java EE 6 e o CDI – Contexts and Dependency Injection*. 2010. Disponível em: <<https://www.devmedia.com.br/java-ee-6-e-o-cdi-contexts-and-dependency-injection/18231>>. Citado na página 27.
- SOUZA, V. S.; FALBO, R. *FrameWeb: A framework-based design method for Web Engineering*. 2007. 9 p. Citado na página 16.



# Apêndices



Documento de Especificação de Requisitos

## SysMap - Sistema de Mapeamento Sistemático

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Mateus Oliveira Almeida	25/04/2016	<i>versão inicial</i>
1.1	Vítor E. Silva Souza	20/05/2016	<i>primeira revisão</i>
2.0	Mateus Oliveira Almeida	19/06/2016	<i>correção e casos de uso</i>
2.1	Vítor E. Silva Souza	22/06/2016	<i>segunda revisão</i>
3.0	Mateus Oliveira Almeida	10/12/2016	<i>correção e adição dos capítulos 5 e 6</i>
3.1	Vítor E. Silva Souza	14/12/2016	<i>terceira revisão</i>
4.0	Mateus Oliveira Almeida	03/11/2017	<i>correções gerais</i>
4.1	Vítor E. Silva Souza	17/11/2017	<i>quarta revisão</i>
5.0	Mateus Oliveira Almeida	09/01/2018	<i>correções gerais</i>
5.1	Vítor E. Silva Souza	05/04/2018	<i>revisão final</i>

Vitória, ES

2018

# 1 Introdução

O Departamento de Informática da Universidade Federal do Espírito Santo (DI/Ufes) necessita de um sistema de informação para ajudar no mapeamento sistemático da literatura, um método para selecionar publicações que podem ser do interesse de um pesquisador dado um tema.

Este documento apresenta os requisitos de usuário do sistema SysMap - Sistema de Mapeamento Sistemático e está organizado da seguinte forma: a Seção 2 apresenta uma descrição do minimundo apresentando o problema. A Seção 3 apresenta as listas de requisitos de usuário levantados junto ao cliente. A seção 4 apresenta o modelo de casos de uso, os diagramas de casos de uso e descrições de casos de uso.

## 2 Descrição do Minimundo

O Departamento de Informática da Universidade Federal do Espírito Santo (DI/U-FES) necessita de um sistema de informação para ajudar no mapeamento sistemático da literatura (abreviado ao longo deste documento para *SysMap*), um método para selecionar publicações que podem ser do interesse de um pesquisador dado um tema.

Tal sistema, daqui em diante denominado “Sistema SysMap” será desenvolvido como um módulo do Marvin,<sup>1</sup> um sistema de informação baseado na Web que agrega ferramentas úteis para o gerenciamento de tarefas de ensino e pesquisa em uma universidade. Ele tenta integrar ferramentas desenvolvidas por estudantes de uma forma que pode ser realmente usado por pessoas.

**1º passo:** Para poder acessar o sistema, os acadêmicos terão um pré-cadastro realizado por um administrador do sistema, possuindo: nome, email, login e senha. Acadêmicos também poderão cadastrar outros acadêmicos.

Na primeira etapa um acadêmico cria um novo SysMap com informações gerais do mesmo e se torna o Dono do SysMap, podendo editar e excluir o seu SysMap. Um SysMap possui: nome, objetivo, perguntas de pesquisa; string de busca, período, área, tipos de publicação, fontes, critérios de inclusão, critérios de exclusão e artigos de controle.

Por sua vez, perguntas de pesquisa possuem: identificador, descrição, justificativa e possíveis respostas. Já critérios possuem: identificador e descrição, enquanto artigos de controle possuem: identificador, título e fonte.

O Dono do SysMap convida outros acadêmicos para colaborar em um determinado SysMap que ele possui, tornando-os Pesquisadores do SysMap para aquele SysMap. Dono do SysMap também podem remover Pesquisadores do SysMap de seu SysMap.

**2º passo:** No segundo passo, um Pesquisador do SysMap pode fazer operações para editar um resultado bruto do SysMap. Um resultado bruto contém: número, fonte, ano, título, autores, lugar, DOI, resumo e palavras-chave.

Dado um SysMap, o sistema executa um processamento de dados, que foram exportados em formatos específicos (ex.: CSV). Dados obtidos em sites de pesquisas de artigos. A partir dessas informações os resultados brutos são criados automaticamente.

**3ª passo:** No terceiro passo, quando indicado pelo Dono do SysMap, o sistema agrupa automaticamente os resultado brutos por título e ano, criando entradas, que possuem, cada uma, um identificador e agrupam um ou mais resultados brutos.

---

<sup>1</sup> <<http://dev.nemo.inf.ufes.br:8180/Marvin/>>.

Resultado bruto e entrada não correspondem às publicações em si, a diferença é que o segundo é gerado e corresponde a agrupamento do primeiro e as informações da entrada são a mesma dos resultado(s) bruto(s) que a gerou.

Um passo importante é a validação dos resultados encontrados ate agora, ou seja as entradas. Em um primeiro passo o sistema verifica se todos os artigos de controle foram encontrados, e um segundo passo verifica se todas as entradas pertencem ao mapeamento.

O segundo passo é feito checando se o título, resumo e palavras-chave das publicações (entradas) contém os termos da string de busca. Entradas que não tem ao menos um termo para cada disjunção, não pertencem ao mapeamento (provavelmente o resultado de um mecanismo de busca falho na fonte). String de busca são conjunções de disjunções e para pertencer ao SysMap pelo menos um termo de cada disjunção tem que ser achado.

Para uma revisão manual das entradas encontradas o Dono do SysMap divide as mesmas entre os Pesquisadores do SysMaps, para que eles possam trabalhar nelas.

**4º passo:** No quarto passo, Um Pesquisador do SysMap filtra uma entrada que tenha sido atribuída a ele. Uma entrada é aceita se ela preenche todos os critérios de inclusão e não satisfaz nenhum critério de exclusão. Nesta etapa, o Pesquisador considera apenas título, resumo e palavras-chave.

Depois de uma entrada ser aceita, o próximo passo exigirá a leitura completa da publicação. O Pesquisador do SysMap faz o upload do PDF da publicação para o próximo passo ficar mais fácil.

No final do quarto passo há uma segunda revisão manual das entradas, o Dono do SysMap novamente divide as mesmas entre os Pesquisadores do SysMap, para que eles possam trabalhar nelas. É feita uma segunda filtragem, agora considerando todo o artigo lido.

**5ª passo:** No quinto passo um Pesquisador do SysMap responde às perguntas de pesquisa para uma entrada que foi atribuída a ele, depois de ler toda a publicação. Neste ponto a primeira avaliação feita no passo anterior pode ser sobrescrita com o conhecimento adquirido após ler toda a publicação.

**Outras funcionalidades:** O sistema também deve permitir o Pesquisador do SysMap visualizar todas as informações produzidas até uma data etapa. Que se dará do seguinte modo:

- Até 1ª etapa - exibir informações gerais do SysMap.
- Até 2ª etapa - visualizar resultado brutos.
- Até 3ª etapa - visualizar entradas e sua validação.
- Até 4ª etapa - visualizar o resultado da primeira filtragem das entradas.

- Até 5ª etapa - visualizar o resultado da segunda filtragem das entradas, comentários e perguntas de pesquisa.

O sistema deve também permitir exportar todas as informações em um formato CSV para ser analisado posteriormente. A exportação de dados segue o mesmo modelo de visualizar as informações produzidas até o momento, sendo a diferença que os dados serão apresentados em CSV.

### 3 Requisitos de Usuário

Tomando por base o contexto do sistema, foram identificados os seguintes requisitos de usuário e regras de negócio:

Tabela 1 – Requisitos Funcionais

Identificador	Descrição	Prioridade	Depende
RF-1	O sistema deve permitir o cadastro de SysMap pelo Dono do SysMap, com as seguintes informações: nome, objetivo, perguntas de pesquisa; string de busca, período, área, tipo de publicação, fontes, critérios de inclusão, critérios de exclusão; e artigos de controle	Alta	<a href="#">RN-1</a>
RF-2	O sistema deve permitir o cadastro das perguntas de pesquisa pelo Dono do SysMap, com as seguintes informações: identificador, descrição, justificativa e possíveis respostas	Alta	<a href="#">RN-1</a> , <a href="#">RF-1</a>
RF-3	O sistema deve permitir o cadastro de critérios pelo Dono do SysMap, com as seguinte informações: identificador, descrição e tipo de critério	Alta	<a href="#">RN-1</a> , <a href="#">RF-1</a>
RF-4	O sistema deve permitir o cadastro dos artigos de controle pelo Dono do SysMap, com as seguintes informações: identificador, título e fonte	Alta	<a href="#">RN-1</a> , <a href="#">RF-1</a>
RF-5	O sistema deve permitir o cadastro e remoção de acadêmico em um SysMap pelo Dono do SysMap	Alta	<a href="#">RN-1</a> , <a href="#">RF-1</a>
RF-6	O sistema deve permitir que o Pesquisador do SysMap gerencie um resultado bruto do SysMap, o resultado bruto possui: número, fonte, ano, título, autores, lugar, DOI, resumo e palavras-chave.	Alta	<a href="#">RF-1</a> , <a href="#">RF-5</a>
RF-7	O sistema deve permitir que o Pesquisador do SysMap execute uma busca automática usando a string de busca e colete todos os resultado brutos encontrados	Alta	<a href="#">RF-1</a> , <a href="#">RF-5</a>

RF-8	O sistema deve permitir que o Dono do SysMap execute um agrupamento automático dos resultado brutos por título e ano, gerando assim entradas, que possuem identificador, nome pdf, arquivo pdf, filtro inicial e filtro final.	Alta	RF-1, RN-1, RN-1
RF-9	O sistema deve permitir que o Dono do SysMap execute uma ação de validação das entradas encontradas até o momento	Alta	RF-1, RN-1
RF-10	O sistema deve permitir que o Dono do SysMap divida as entradas entre os Pesquisadores do SysMap para que esses possam ficar responsáveis pelas mesmas	Alta	RF-1, RN-1, RF-8
RF-11	O sistema deve permitir que o Pesquisador do SysMap faça upload de uma publicação	Alta	RF-1, RF-8
RF-12	O sistema deve permitir que o Pesquisador do SysMap exporte as informações do mapeamento sistemático produzidas até o passo atual	Média	RF-1
RF-13	O sistema deve permitir que o SysMap Research visualize as informações do mapeamento sistemático produzidas até o passo atual	Média	RF-1
RF-14	O sistema deve permitir que o Pesquisador do SysMap filtre uma entrada que tenha sido atribuída a ele, considerando apenas título, palavras-chave e keywords	Média	RF-1, RF-5, RF-9, RF-10
RF-15	O sistema deve permitir que o Pesquisador do SysMap responda as perguntas de pesquisa para uma entrada que foi atribuída a ele	Média	RF-1, RF-5, RF-9, RF-10, RF-14

Tabela 2 – Regras de Negócio

Identificador	Descrição	Prioridade
RN-1	Quando um acadêmico cria um SysMap automaticamente ele vira o Dono do SysMap	Alta



Tabela 3 – Requisitos Não Funcionais

Identificador	Descrição	Categoria	Escopo	Prioridade
RNF-1	A ferramenta deve estar disponível como uma aplicação Web, acessível a partir dos principais navegadores disponíveis no mercado.	Portabilidade	Sistema	Alta
RNF-2	A ferramenta deve ser de aprendizado fácil, não sendo necessário nenhum treinamento especial para seu uso.	Facilidade de Aprendizado	Sistema	Média
RNF-3	A ferramenta deve ser de fácil operação, não sendo necessário uso contínuo para uma boa operação do sistema.	Facilidade de Operação	Sistema	Alta
RNF-4	O sistema deve controlar o acesso as funcionalidades.	Segurança de acesso	Sistema	Alta
RNF-5	O desenvolvimento do sistema deve explorar o potencial de reutilização de componentes, tanto no que se refere ao desenvolvimento com reúso quanto ao desenvolvimento para reúso.	Reusabilidade	Sistema	Média

## 4 Modelo de Casos de Uso

O modelo de casos de uso visa capturar e descrever as funcionalidades que um sistema deve prover para os atores que interagem com o mesmo. Os atores identificados no contexto deste projeto estão descritos na Tabela 4.

Tabela 4 – Atores

Ator	Descrição
Acadêmico	Representa um acadêmico que tem como objetivo participar ou visualizar dados de um mapeamento sistemático (SysMap)
Dono do SysMap	Acadêmico que é o responsável por um SysMap, ele irá cadastrar as informações referentes ao mesmo
Pesquisador do SysMap	Acadêmico convidado pelo Dono do SysMap para que possa contribuir com o mesmo

A seguir, são apresentados os diagramas de casos de uso e descrições associadas, do sistema SysMap.

### 4.1 Sistema SysMap

A Figura 1 apresenta o diagrama de casos de uso do sistema SysMap.

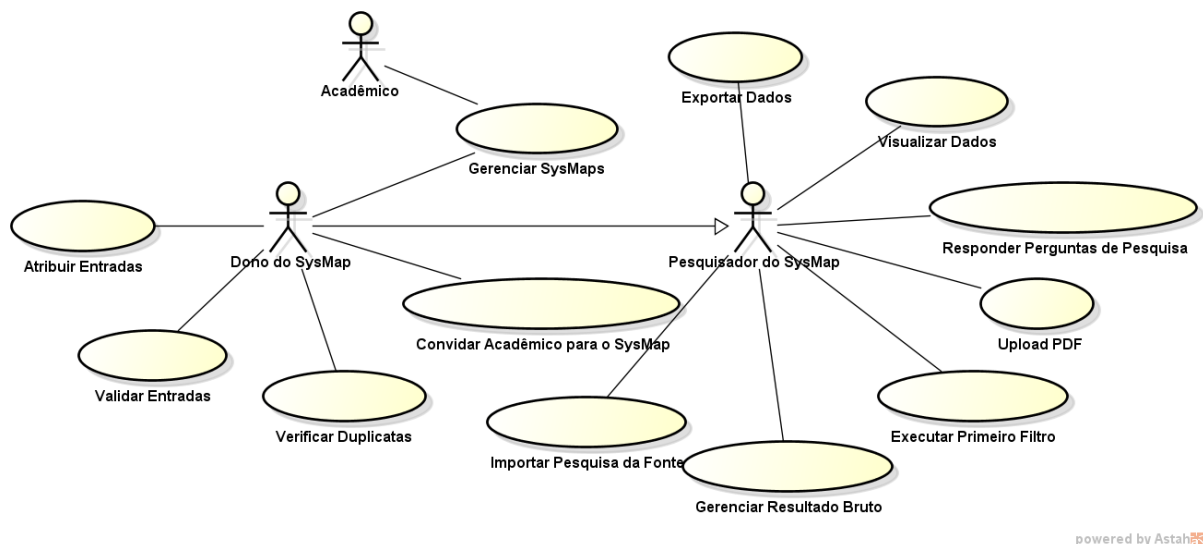


Figura 1 – Diagrama de Casos de Uso do sistema SysMap

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

Tabela 5 – Casos de Uso Cadastrais

Id	Nome	Ações	Observações	Requisitos	Classes
UC-1	Gerenciar SysMaps	I	Informar: objetivo, perguntas de pesquisa e seus atributos (identificador, descrição, análise racional, possíveis respostas), <i>string</i> de busca, período, área, tipos de publicação, fontes, critérios de inclusão e/ou critérios de exclusão com seus atributos (identificador, descrição e tipo de critério), e artigos de controle (possuindo identificador, título, fonte). Assim que o Acadêmico fizer o cadastro, ele se torna o dono do sysmap.	RF-1, RF-2, RF-3, RF-4	SysMap, Critério, Artigo de Controle, Pergunta de Pesquisa, Entrada
		A			
		C			
		E	Não é permitido a exclusão de um sysmap por quem não é o seu dono.		
UC-2	Gerenciar Resultado Bruto	I	Informar: número, fonte, ano, título, autores, lugar, DOI, resumo, palavras-chave.	RF-6	Resultado Bruto
		A			
		C			
		E			

## Descrição de Caso de Uso

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-3

**Caso de Uso:** Convidar Acadêmico para o SysMap

**Descrição Sucinta:** Este caso de uso é responsável por convidar outro Acadêmico para colaborar em um SysMap de um dono do SysMap, assim o Acadêmico convidado vira um pesquisador do SysMap daquele SysMap. Também o dono pode remover pesquisadores dos seus SysMaps.

Tabela 6 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Convidar Acadêmico		<ol style="list-style-type: none"><li>1. O dono do SysMap seleciona um SysMap</li><li>2. O dono do SysMap seleciona um Acadêmico</li><li>3. O sistema coloca o Acadêmico selecionado como um Pesquisador do SysMap daquele SysMap</li></ol>
Remover Acadêmico		<ol style="list-style-type: none"><li>1. O dono do SysMap seleciona um SysMap</li><li>2. O dono do SysMap seleciona um dos Pesquisadores do SysMap</li><li>3. O sistema remove o Pesquisador do SysMap daquele SysMap</li></ol>

**Requisitos Relacionados:** [RF-5](#)

**Classes Relacionadas:** Acadêmico, SysMap.

## Descrição de Caso de Uso

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-4

**Caso de Uso:** Importar Pesquisa da Fonte

**Descrição Sucinta:** Este caso de uso é responsável por importar resultados da pesquisa associados a uma fonte.

Tabela 7 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Executar importação		<ol style="list-style-type: none"><li>1. O pesquisador informa o arquivo csv contendo os resultados da pesquisa feita em uma fonte.</li><li>2. O sistema importa esses dados como resultados brutos associados a essa fonte.</li></ol>

**Requisitos Relacionados:** [RF-7](#)

**Classes Relacionadas:** SysMap, Resultado Bruto.

## Descrição de Caso de Uso

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-5

**Caso de Uso:** Verificar Duplicatas

**Descrição Sucinta:** Quando indicado pelo Dono do SysMap, o sistema automaticamente agrupa resultado brutos por título e ano, criando uma entrada

Tabela 8 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Executar agrupamento		<ol style="list-style-type: none"><li>1. O Dono do SysMap seleciona um SysMap.</li><li>2. O sistema agrupa os resultados brutos do SysMap por título e ano, gerando assim as entradas. Cada entrada tem um número identificador, começando de 1 e sendo incrementado a cada nova entrada gerada.</li></ol>

**Requisitos Relacionados:** [RF-8](#)

**Classes Relacionadas:** Entrada, Resultado Bruto.

## Descrição de Caso de Uso

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-6

**Caso de Uso:** Validar Entradas

**Descrição Sucinta:** Quando indicado pelo Dono do SysMap, o sistema automaticamente verifica se todos os artigos de controle foram achados e todas as entradas pertencem ao mapeamento.<sup>1</sup>

Tabela 9 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Validar		<ol style="list-style-type: none"> <li>1. Dono do SysMap seleciona um SysMap.</li> <li>2. O sistema verifica se todos os artigos de controle foram achados e retorna uma resposta positiva ou negativa.</li> <li>3. O sistema verifica cada entrada, se pertence ou não ao mapeamento, apresenta ao Dono do SysMap aquelas que não pertencem e pergunta se ele gostaria de excluí-las.</li> <li>4. O Dono do SysMap indica sua preferência.</li> <li>5. Caso a resposta seja positiva, o sistema exclui as entrada que não pertencem ao mapeamento.</li> </ol>

**Requisitos Relacionados:** RF-9

**Classes Relacionadas:** Entrada.

<sup>1</sup> O passo de verificar se todas as entradas pertencem ao mapeamento é feito checando: título, palavras-chave e keywords das publicações por termos na string de busca. Entries que não tem pelo menos um termo para cada disjunção não pertencem ao mapeamento (são provavelmente o resultado de um mecanismo de busca falho). As search strings são conjunções de disjunções: (“termo 1.1” OR “termo 2.2” OR ... OR “termo 1.n”) AND (“termo 2.1” OR ... OR “termo 2.m”) AND ... AND (“termo k.1” OR ... OR “termo k.p”). Para pertencer ao SysMap, pelo menos um termo de cada grupo de disjunção 1, 2, ..., k precisa ser encontrado.

**Descrição de Caso de Uso**

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-7

**Caso de Uso:** Atribuir Entradas

**Descrição Sucinta:** O Dono do SysMap, indica quais entradas um Pesquisador do SysMap pode trabalhar.

Tabela 10 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Distribuir entradas	1. A associação de um acadêmico a uma entrada ocorre se tanto um quanto outro pertencem ao mesmo SysMap.	1. O Dono do SysMap seleciona as entradas para um Pesquisador do SysMap. 2. O sistema coloca o Pesquisador do SysMap como responsável pelas entradas.

**Requisitos Relacionados:** [RF-10](#)

**Classes Relacionadas:** Entrada, Acadêmico



## Descrição de Caso de Uso

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-8

**Caso de Uso:** Executar Primeiro Filtro

**Descrição Sucinta:** Um Pesquisador do SysMap filtra as entradas que foram designadas a ele, considerando o título, resumo e palavra-chave. Uma entrada passa pelo filtro se ela satisfaz todos os critérios de inclusão e não satisfaz nenhum critério de exclusão.

Tabela 11 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Executar primeiro filtro	1. Apenas o pesquisador dono do SysMap ou convidado pelo dono podem informar dados sobre os critérios de uma entrada.	1. O pesquisador do SysMap informa uma entrada. 2. O pesquisador informa quais critérios de inclusão/exclusão ela atende. 3. O sistema registra o que o Pesquisador designou para a entrada.

**Requisitos Relacionados:** [RF-14](#)

**Classes Relacionadas:** Entrada, Acadêmico

**Descrição de Caso de Uso**

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-9

**Caso de Uso:** Upload PDF

**Descrição Sucinta:** Depois de uma entrada ser filtrada e aceita, o Pesquisador do SysMap faz o upload do PDF da publicação.

Tabela 12 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Inserir PDF		<ol style="list-style-type: none"><li>1. O Pesquisador do SysMap seleciona uma entrada</li><li>2. O Pesquisador do SysMap seleciona um pdf</li><li>3. O sistema salva o pdf</li><li>4. O sistema relaciona o PDF da publicação a entrada</li></ol>

**Requisitos Relacionados:** [RF-11](#)

**Classes Relacionadas:** Entrada, Acadêmico

## Descrição de Caso de Uso

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-10

**Caso de Uso:** Responder Perguntas de Pesquisa

**Descrição Sucinta:** O Pesquisador do SysMap responde as perguntas de pesquisa para uma entrada que foi atribuída a ele, depois de ler toda a publicação, e também informa o resultado da segunda filtragem.

Tabela 13 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Responder perguntas de pesquisa	1. Apenas o pesquisador dono do SysMap ou convidado pelo dono podem informar as respostas para as perguntas de pesquisa.	1. O pesquisador informa a entrada e as respostas para cada pergunta de pesquisa, após a leitura do artigo. 2. O sistema registra as respostas. 3. O pesquisador informa quais critérios de inclusão/exclusão a entrada atende. 4. O sistema registra o resultado da segunda filtragem.

**Requisitos Relacionados:** [RF-15](#)

**Classes Relacionadas:** Entrada, Acadêmico

**Descrição de Caso de Uso**

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-11

**Caso de Uso:** Visualizar Dados

**Descrição Sucinta:** Visualizar as informações geradas até uma etapa do mapeamento sistemático.

Tabela 14 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Visualizar informações		<ol style="list-style-type: none"><li>1. O dono indica o SysMap que quer visualizar as informações.</li><li>2. O sistema apresenta todas as informações atuais do SysMap, conforme descrição na Seção 2.</li></ol>

**Requisitos Relacionados:** [RF-13](#)

**Classes Relacionadas:** SysMap, Entrada, Acadêmico, Resultado Bruto, Artigo de Controle, Critério, Pergunta de Pesquisa

## Descrição de Caso de Uso

**Projeto:** SysMap - Sistema de Mapeamento Sistemático

**Identificador do Caso de Uso:** UC-12

**Caso de Uso:** Exportar Dados

**Descrição Sucinta:** O Pesquisador do SysMap exporta as informações geradas do mapeamento sistemático no formato CSV.

Tabela 15 – Fluxos de Eventos Normais

Nome do Fluxo	Precondição	Descrição
Exportar informações		1. O dono indica o SysMap que possui as informações para exportar. 2. O sistema exporta todas as informações presentes do SysMap.

**Requisitos Relacionados:** [RF-12](#)

**Classes Relacionadas:** SysMap, Entrada, Acadêmico, Resultado Bruto, Artigo de Controle, Critério, Pergunta de Pesquisa

## 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 na seção anterior. A seguir, é apresentado o diagrama de classes do sistema. Na Seção 6 – Glossário de Projeto – são apresentadas as descrições das classes, atributos e operações presentes no diagrama apresentado nesta seção.

### 5.1 Diagrama de Classes

A Figura 2 apresenta o diagrama de classes.

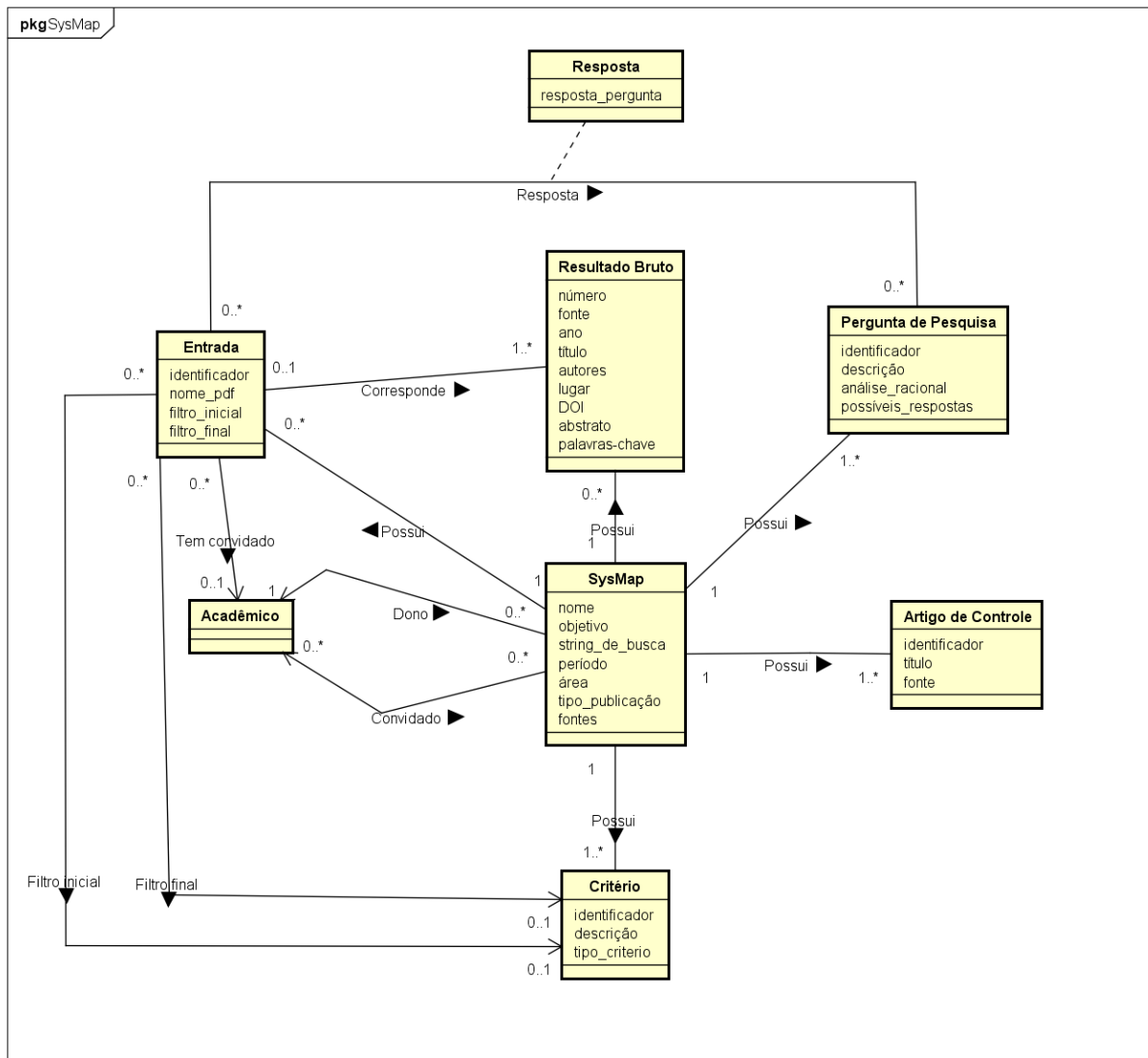


Figura 2 – Diagrama de Classes do Sistema SysMap.

## 5.2 Restrição de Integridade

1. A associação de um acadêmico a uma entrada ocorre se tanto um quanto outro pertencem ao mesmo SysMap.
2. Apenas o pesquisador dono do SysMap ou convidado pelo dono podem alterar dados de uma entrada.
3. Apenas o pesquisador dono do SysMap ou convidado pelo dono podem informar as respostas para as perguntas de pesquisa.
4. A associação de uma entrada e uma pergunta de pesquisa ocorre se tanto uma quanto a outra pertencem ao mesmo SysMap.
5. A associação de uma entrada e um critério, como resultado de uma filtragem, ocorre se tanto uma quanto o outro pertencem ao mesmo SysMap.

# 6 Glossário de Projeto

## 6.1 Dicionário

Esta seção apresenta as definições dos principais conceitos envolvidos no projeto.

- **SysMap:** Registro de um mapeamento sistemático
  - nome: nome do SysMap
  - objetivo: objetivo do SysMap
  - string de busca: string de busca usada para procurar artigos que possam ter informações úteis para o mapeamento sistemático
  - período: período de artigos aceitos na busca
  - área: área de conhecimento a que o mapeamento se refere
  - tipo de publicação: tipo de publicações aceitas para o mapeamento
  - fontes: fontes usadas para a procura de artigos
  
- **Critério:** Critérios usados para decidir se um artigo específico pertence ao mapeamento sistemático ou não
  - identificador: um identificador para o critério
  - descrição: a descrição detalhada do Critério
  - tipo de critério: tipo do critério, que pode ser: inclusão ou exclusão
  
- **Artigo de controle:** Artigo usado para validar as entradas geradas
  - identificador: identificador do artigo de controle
  - título: título do artigo de controle
  - fonte: fonte de origem do artigo de controle
  
- **Pergunta de Pesquisa:** Perguntas que serão respondidas após ler artigos selecionados
  - identificador: identificador da pergunta
  - descrição: conteúdo da pergunta em si
  - análise racional: indica qual o objetivo para o qual a pergunta é feita
  - possíveis respostas: categorias a serem utilizadas na resposta, de modo a facilitar a criação de gráficos analíticos
  
- **Resultado Bruto:** Indica um artigo em sua primeira etapa de processamento



- número : numero para identificação do resultado bruto
  - fonte: lugar de onde foi retirado o artigo
  - ano: ano em que o artigo foi publicado
  - título: título do artigo
  - autores: autores do artigo
  - lugar: lugar de publicação do artigo
  - DOI: Digital Object Identifier, um identificar digital do artigo
  - resumo: resumo do artigo
  - palavras-chave: palavras-chave do artigo
- **Entrada:** Indica um artigo em uma avançada etapa de processamento, em relação ao resultado bruto, uma entrada é gerada a partir do agrupamento de resultados brutos por ano e título.
    - identificador : identificador da entrada
    - nome pdf : nome do arquivo em pdf referente ao artigo
    - arquivo pdf : arquivo do pdf
    - filtro inicial : análise inicial da entrada, será aceita caso preencha todos os critérios de inclusão e não satisfaz nenhum critério de exclusão, nessa etapa é considerado apenas o título, resumo, e palavras-chave
    - filtro final : análise final da entrada, são considerados os mesmos critérios do filtro inicial, após a leitura completa do artigo
  - **Resposta:** Indica a resposta de uma entrada para uma pergunta de pesquisa específica
    - resposta pergunta : a resposta para a pergunta sendo feita
  - **Convidado:** Indica um Acadêmico que foi convidado a participar um dado sysmap

**Vítor:** Falta descrever as associações entre as classes.

**Mateus:** Adicionada a seção abaixo

## 6.2 Associações

Esta seção apresenta as associações entre as classes do projeto.

- **SysMap - Artigo de Controle:**
  - Possui : Um SysMap possui Artigo(s) de Controle.

- **SysMap - Critério:**
  - Possui : Um SysMap possui Critério(s).
  
- **SysMap - Pergunta de Pesquisa:**
  - Possui : Um SysMap possui Pergunta(s) de Pesquisa.
  
- **SysMap - Resultado Bruto:**
  - Possui : Um SysMap possui Resultado(s) Bruto.
  
- **SysMap - Entrada:**
  - Possui : Um SysMap possui Entrada(s).
  
- **Entrada - Pergunta de Pesquisa:**
  - Resposta : Uma Entrada está associada a Pergunta de Pesquisa por uma Resposta, que possui a resposta propriamente dita da entrada referente a pergunta.
  
- **Entrada - Resultado Bruto:**
  - Corresponde : Uma Entrada corresponde a um ou mais agrupamento de Resultado Bruto.
  
- **Entrada - Critério:**
  - Filtro inicial : Uma Entrada possui um Critério que corresponde ao resultado da primeira filtragem da mesma.
  - Filtro final : Uma Entrada possui um Critério que corresponde ao resultado da segunda filtragem da mesma.
  
- **Entrada - Acadêmico:**
  - Tem convidado : Uma Entrada possui um Acadêmico convidado, que assim como o dono do SysMap, pode editar informações da mesma.
  
- **Acadêmico - SysMap:**
  - Dono : O Acadêmico que cria o SysMap se torna o dono do mesmo
  - Convidado : O dono do SysMap pode convidar outros acadêmicos para conceder acesso ao SysMap e também para edição de Entradas.



Documento de Projeto de Sistema

## SysMap - Sistema de Mapeamento Sistemático

Registro de Alterações:

Versão	Responsável	Data	Alterações
1.0	Mateus Oliveira Almeida	15/03/2018	Versão inicial
1.1	Vítor E. Silva Souza	05/04/2018	Primeira revisão
2.0	Mateus Oliveira Almeida	16/04/2018	Correções gerais
2.1	Vítor E. Silva Souza	27/05/2018	Segunda revisão
3.0	Mateus Oliveira Almeida	27/05/2018	Correções gerais

Vitória, ES

2018

# 1 Introdução

Este documento apresenta o documento de projeto (*design*) do sistema SysMap - Sistema de Mapeamento Sistemático. Este documento está organizado da seguinte forma: a Seção 2 apresenta a plataforma de software utilizada na implementação da ferramenta; a Seção 3 trata de táticas utilizadas para tratar requisitos não funcionais (atributos de qualidade); por fim, a Seção 4 apresenta o projeto da arquitetura de software e suas subseções explicam cada uma de suas camadas.

## 2 Plataforma de Desenvolvimento

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

Tecnologia	Versão	Descrição	Propósito
JavaEE	7	Conjunto de especificação de APIs e tecnologias, que são implementadas por programas servidores de aplicação.	Reduzir a complexidade do desenvolvimento, implantação e gerenciamento de aplicações, de modo que o desenvolvedor não se preocupe demasiadamente com segurança, escalabilidade e desempenho.
Java	8	Linguagem de programação orientada a objetos e independente de plataforma.	Desenvolvimento de aplicativos em linguagem de programação orientada a objetos e independente de plataforma.
JSF	2.2.12	API para a construção de interfaces de usuários baseada em componentes para aplicações web	Criação das páginas Web utilizando componentes visuais pré-prontos e sua integração com o restante da aplicação.
EJB	4.0.9	Componente da plataforma JEE que roda em um container de um servidor de aplicação.	Fornecer um desenvolvimento rápido e simplificado de aplicações Java, com base em componentes distribuídos, transacionais, seguros e portáteis.
JPA	2.1	API para persistência de dados por meio de mapeamento objeto-relacional.	Eliminar muito do trabalho com consultas SQL e facilitar a manutenção visto que menos linhas de código são necessárias.
CDI	1.1	<i>Frameworks</i> para injeção de dependências.	Integração das diferentes camadas da arquitetura e serviços de transação.
AdminLTE	2.3.0	Template Bootstrap 3 responsivo	Utilizar um template open source que seja facilmente personalizado e englobe scripts JS e folhas de estilos CSS para prover um layout responsivo.
Facelets	2.0	Sistema de template Web de código aberto.	Reusar estrutura comum às paginas e facilitar futura manutenção do padrão visual do sistema.
PrimeFaces	6.0	Conjunto de componentes JSF open source com várias extensões.	Reutilizar componentes não sendo necessário a criação dos mesmos.
MySQL Server	5.6	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Persistência dos dados manipulados pela ferramenta.
WildFly	10.0.0	Servidor de Aplicações para Java EE.	Prover acesso a aplicações web por meio do protocolo HTTP (HyperText Transfer Protocol).

Tabela 1 – Plataforma de Desenvolvimento e Tecnologias Utilizadas

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

Tecnologia	Versão	Descrição	Propósito
Eclipse Java EE IDE for Web Developers	4.5.2	Ambiente de desenvolvimento (IDE) para a linguagem Java.	Facilitar a atividade de implementação de software.
Astah Community	7.0.0	Ferramenta para modelagem em UML	Modelar os diagramas de classes, casos de uso, etc.
Apache Maven	3.3.3	Ferramenta de gerência de projeto baseada em project object model (POM).	Simplificar o download das dependências do projeto.
TeXstudio	5.5.1	Editor de LaTeX.	Escrever a documentação do sistema, sendo usado o <i>template abnTeX</i> < <a href="http://www.abntex.net.br">http://www.abntex.net.br</a> >.
TeX Live	2015	Implementação de TeX	Distribuição TeX usada para gerar a documentação do sistema escrita em arquivos TeX.

Tabela 2 – Softwares de Apoio ao Desenvolvimento do Projeto

### 3 Atributos de Qualidade e Táticas

Na Tabela 3 são listados os atributos de qualidade considerados neste projeto, com uma indicação se os mesmos são condutores da arquitetura ou não e as táticas a serem utilizadas para tratá-los.

Categoria	Requisitos Não Funcionais	Condutor da Arquitetura	Tática
Portabilidade	RNF-1	Sim	Desenvolver o sistema de modo a estar disponível como uma aplicação Web, acessível a partir dos principais navegadores disponíveis no mercado.
Facilidade de Aprendizado, Facilidade de Operação	RNF-2, RNF-3	Sim	Utilizar quando possível nomes auto-descritivos para os campos e botões do sistema, deixando assim o sistema mais amigável. Quando possível, utilizar a entrada de dados por meio de seleção ao invés de digitação.
Segurança de Acesso	RNF-4	Sim	Verificar o usuário logado, para que só efetue alguma operação ou visualize uma informação, o qual tem permissão.
Reusabilidade	RNF-5	Sim	Reutilizar componentes e <i>frameworks</i> existentes. Quando não houver componentes disponíveis e houver potencial para reuso, devem-se desenvolver novos componentes para reuso.

Tabela 3 – Atributos de Qualidade e Táticas Utilizadas

## 4 Arquitetura de Software

Primeiro deve-se informar que o SysMap será implementado como um módulo do Marvin, que é um Sistema de Informação baseado na Web e que agrega ferramentas úteis para o gerenciamento de tarefas de ensino e pesquisa em uma universidade. Marvin é uma tentativa de unificar ferramentas desenvolvidas por estudantes, com o objetivo da real utilização por pessoas.

A arquitetura de software do subsistemas SysMap, baseia-se na combinação de camadas e módulo. O módulo está organizado em três camadas seguindo o proposto pelo FrameWeb, a saber: Camada de Apresentação (*Presentation Tier*), Camada de Negócios (*Business Tier*) e Camada de Acesso a Dados (*Data Access Tier*). De forma a dar suporte para a construção da aplicação, a ferramenta de apoio JButler<sup>1</sup> será utilizada. Tal ferramenta provê classes que auxiliam na implementação dos casos de uso cadastrais que seguem o modelo de arquitetura a ser utilizado.

A primeira camada contém os pacotes de Visão (*View*) e Controle (*Control*), a segunda contém o de Domínio (*Domain*) e o de Aplicação (*Application*) e a terceira somente o pacote de Persistência (*Persistence*). Cada pacote será explicado melhor nas próximas seções onde será descrito o módulo SysMap. A Figura 1 apresenta a visão geral das camadas e seus pacotes juntamente com o relacionamento que existe entre eles e as tecnologias Java EE utilizadas em cada pacote.

A Figura 2 apresenta a subdivisão do módulo nas camadas descritas acima, a saber a Camada de Apresentação (`control`), Camada de Negócios (`domain` e `application`) e Camada de Acesso a Dados (`persistence`). A camada de visão (`view`) corresponde às páginas Web, que encontram-se em uma outra pasta, separada das classes Java, como mostrado na Figura 3.

Os diagramas das subseções a seguir seguem o padrão proposto pelo FrameWeb (SOUZA, 2007).

### 4.1 Camada de Apresentação

As funcionalidades criar, visualizar, editar e excluir (abreviadas de CRUD, do inglês *create, read, update and delete*), seguem um mesmo fluxo de execução e de interação com o usuário. Tais funcionalidades são similares para todos os casos de uso cadastrais devido a utilização da ferramenta JButler. Esse fluxo de execução similar é representado na Figura 4 através de um modelo de apresentação genérico.

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



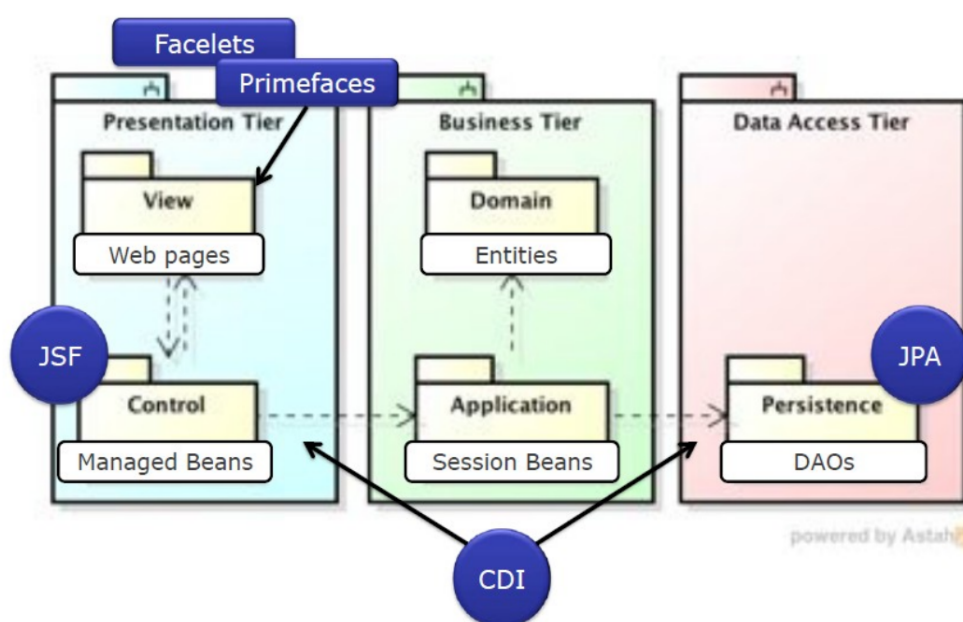


Figura 1 – Arquitetura de Software do Sistema (LIMA, 2015)

- > > br.ufes.inf.nemo.marvin.sysmap
- > > br.ufes.inf.nemo.marvin.sysmap.application
- > > br.ufes.inf.nemo.marvin.sysmap.controller
- > > br.ufes.inf.nemo.marvin.sysmap.domain
- > > br.ufes.inf.nemo.marvin.sysmap.persistence

Figura 2 – Subdivisão do módulo SysMap.

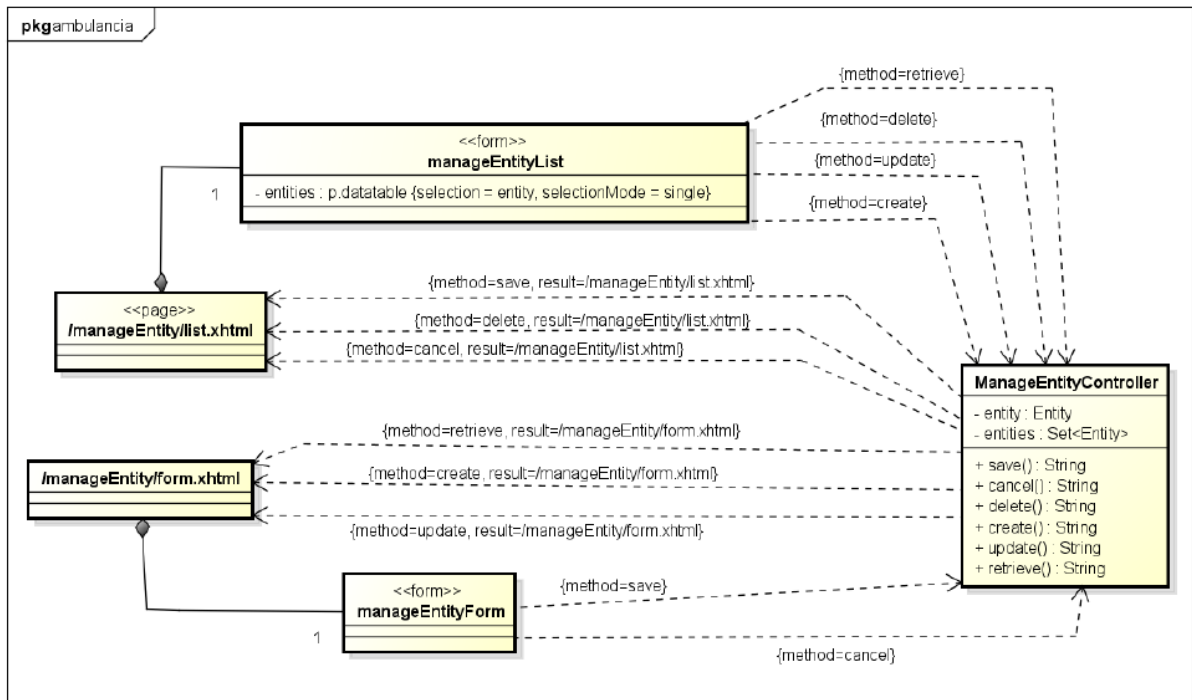
- > > manageAcademicGuest
- > > manageControlArticle
- > > manageCriteria
- > > manageEntry
- > > manageRawResult
- > > manageResearchQuestion
- > > manageSysMap

Figura 3 – Camada de visão.

Para os casos de uso que apresentam funções diferentes de apenas as básicas de cadastro, o modelo de navegação mostrado anteriormente não pode ser aplicado. A seguir seguem os modelos de navegação. Os casos de uso: *Importar Pesquisa da Fonte*, *Verificar Duplicatas* e *Exportar Dados* estão agrupados na Figura 7, pois são executados no mesmo lugar, na página de um elemento SysMap.

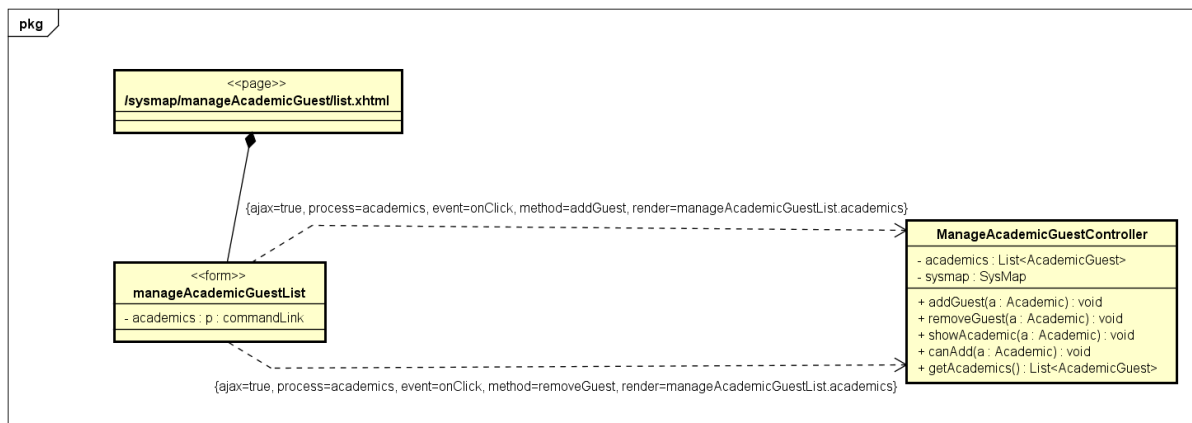
Tem-se também os modelos para *Convidar Acadêmico para o SysMap* (Figura 5) e *Upload PDF* (Figura 6). Os casos de uso *Atribuir Entradas*, *Executar Primeiro Filtro* e *Responder Perguntas de Pesquisa* não estão representados pois o fluxo de execução seria o mesmo da funcionalidade editar que está representada na Figura 4. O caso de uso Validar

Entradas não foi implementado.



powered by Astah

Figura 4 – Modelo de Navegação de um CRUD nemo-utils (predecessor do JButler), usado como base para funcionalidades dos cadastros do SysMap. (COSTA, 2015)



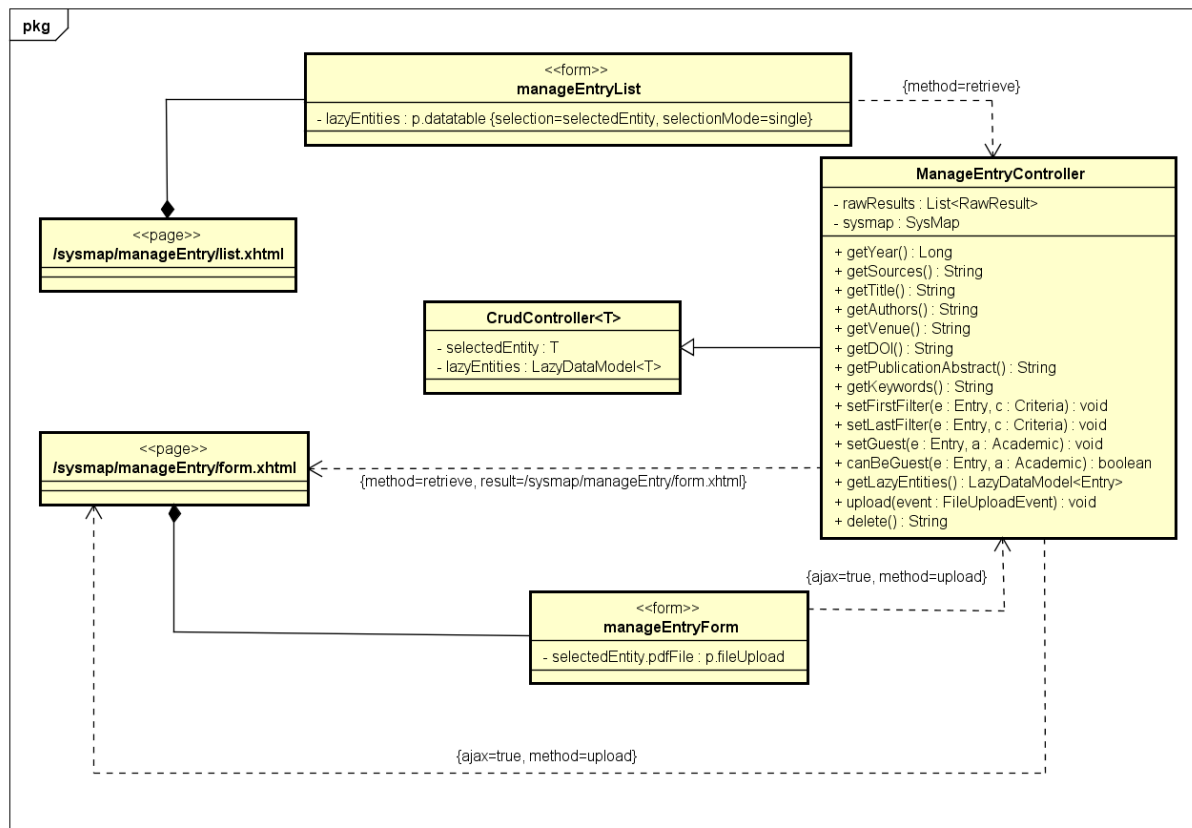
powered by Astah

Figura 5 – Modelo de Navegação Convidar Acadêmico.

## 4.2 Camada de Negócios

### 4.2.1 Dominio

Todas as classes de domínio estendem `PersistentObjectSupport` do JButler, mas essa herança não é mostrada nos diagramas com o intuito de não poluição do mesmo com várias associações. A Figura 8 mostra o Modelo de Entidades para o módulo *SysMap*.



powered by Astah

Figura 6 – Modelo de Navegação Upload PDF.

## 4.2.2 Aplicação

As classes de aplicação estendem de `CrudServiceBean` do JButler. Essa classe está representada na Figura 9 de forma genérica. Como nos diagramas anteriores, para não poluí-lo com várias associações, essa herança não é mostrada no diagrama.

As Figuras 10 e 11 mostram o modelo de aplicação para o módulo *SysMap*.

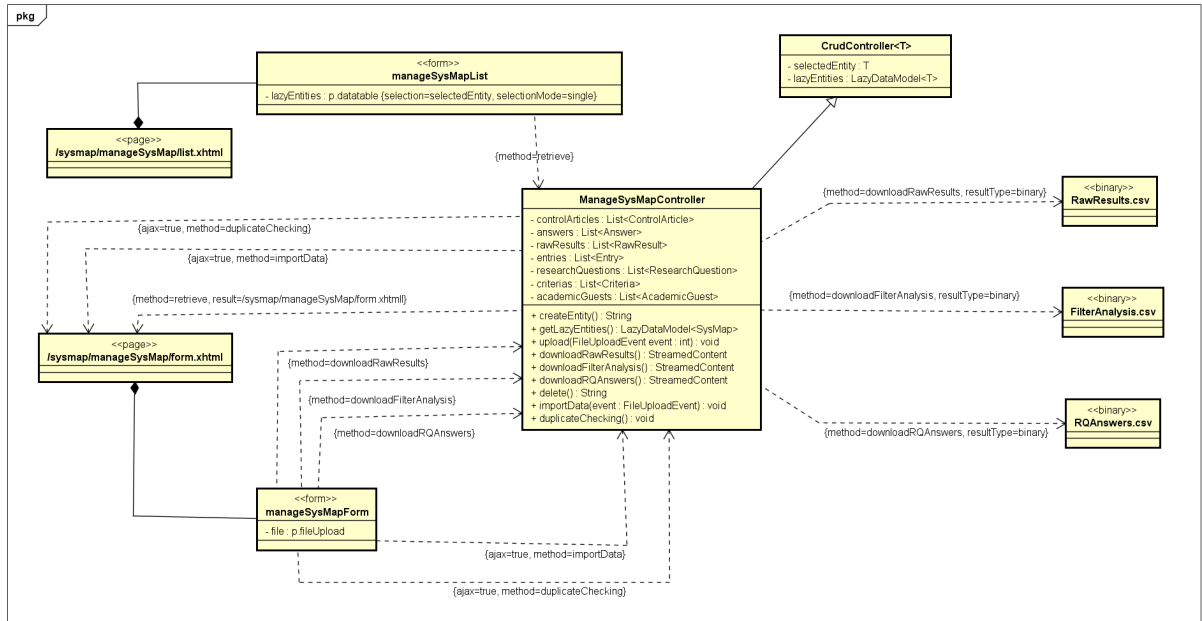


Figura 7 – Modelo de Navegação Importar Pesquisa da Fonte, Verificar Duplicatas e Exportar Dados.

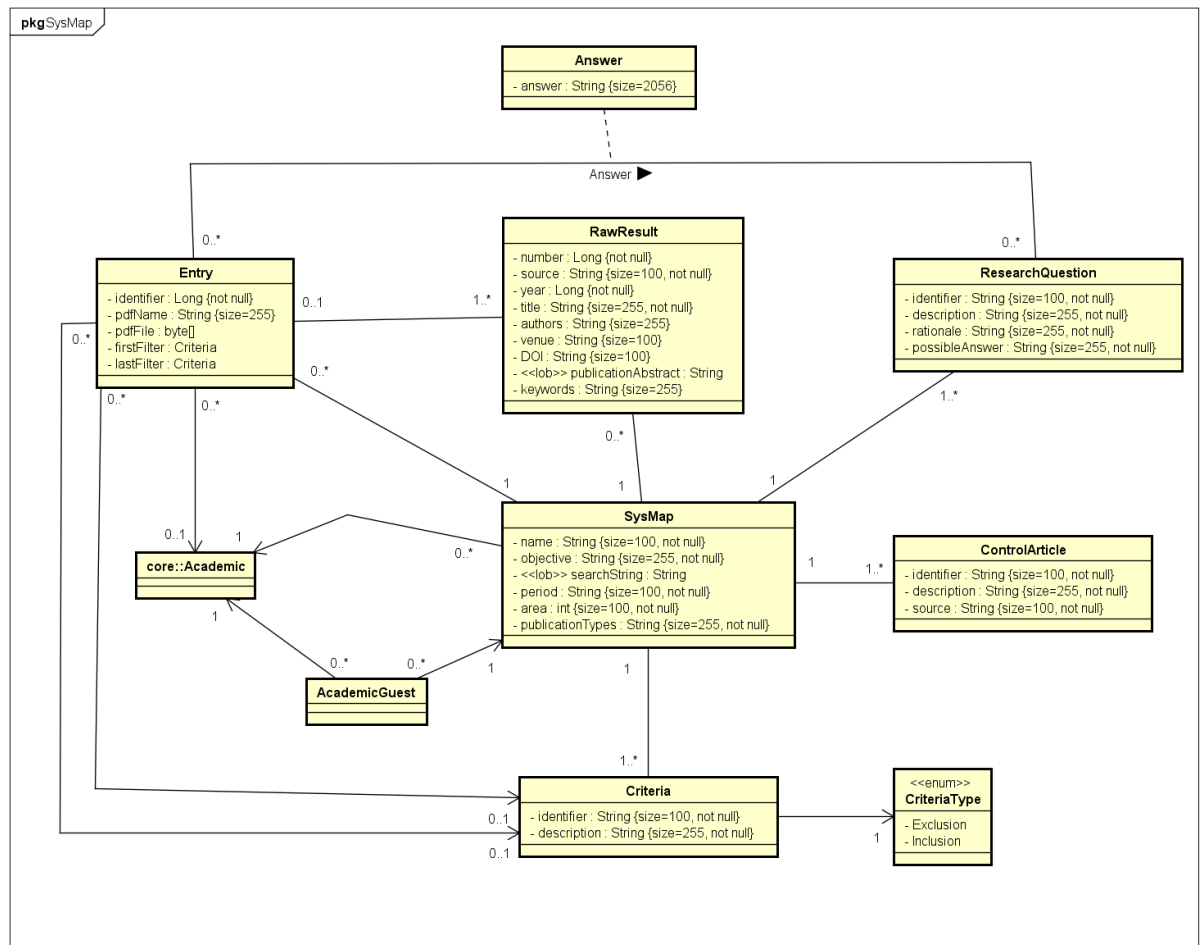
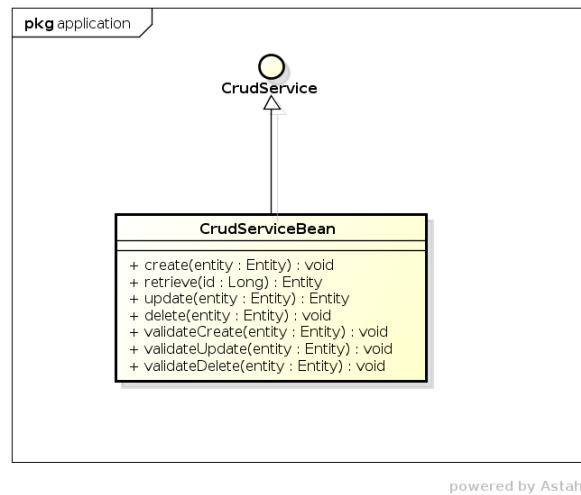
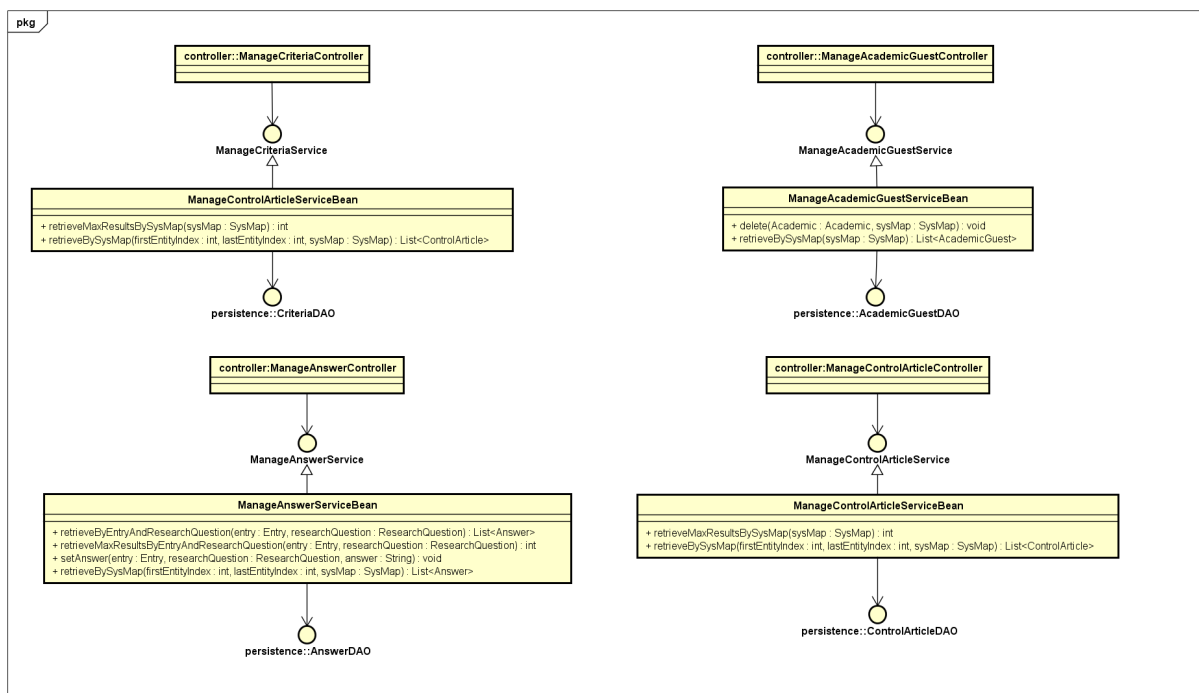


Figura 8 – Modelo de Entidades do Marvin para o módulo SysMap.



powered by Astah

Figura 9 – Modelo de Aplicação genérica da ferramenta JButler (LIMA, 2015).



powered by Astah

Figura 10 – Parte do modelo de Aplicação do Marvin para o módulo SysMap.

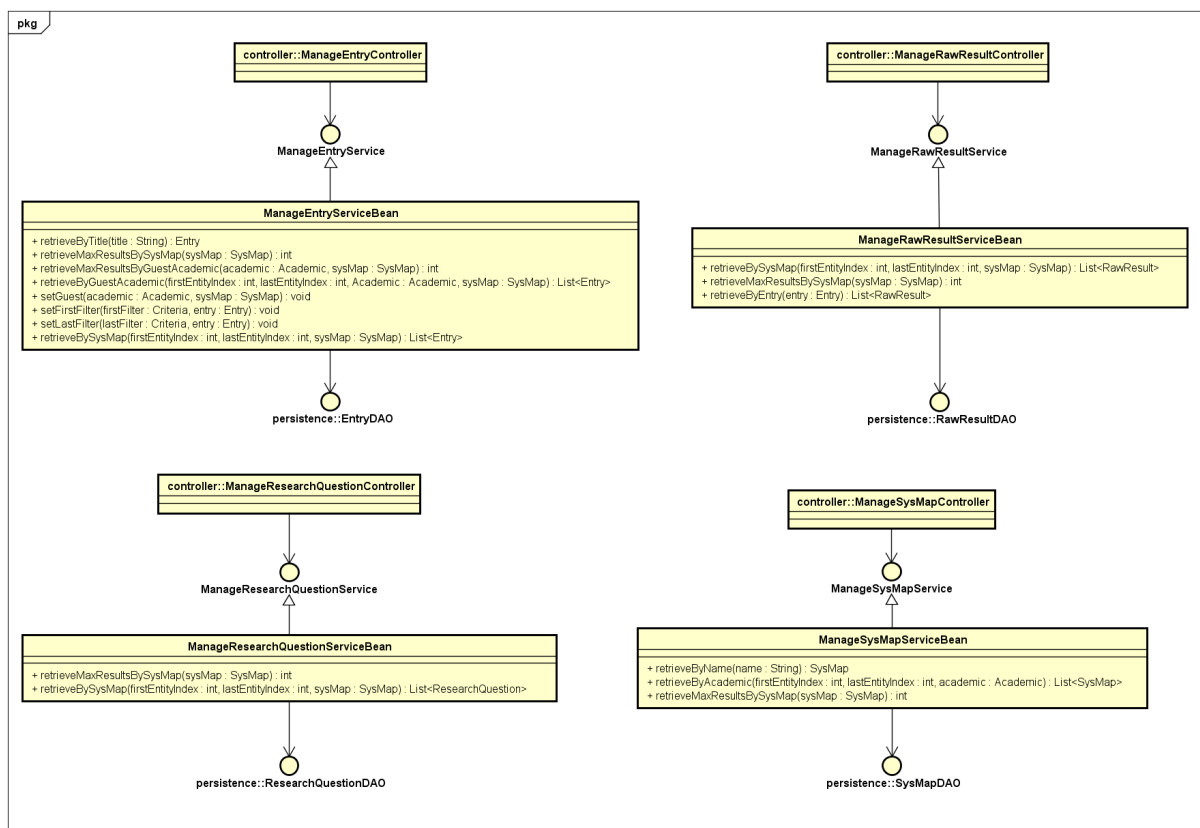


Figura 11 – Parte do modelo de Aplicação do Marvin para o módulo *SysMap*.

### 4.3 Camada de Acesso a Dados

Serão apresentados os Modelos de Persistência desenvolvidos para o módulo SysMap que foram usados na implementação do pacote de persistência.

Observa-se que com o nome das classes já é indicado qual tecnologia de persistência foi utilizada, esse sistema de nomenclatura é mais uma sugestão do FrameWeb para simplificar o processo de software. Vale notar também que na Figura 12 está representado o diagrama de persistência genérico provido pela ferramenta JButler.

Note-se que a relação de herança entre os DAOs específicos e o DAO base não é representada explicitamente nos diagramas para evitar poluição visual. Esta também é uma recomendação do FrameWeb, ficando, portanto, o desenvolvedor incumbido de derivar essa relação implicitamente ao analisar o modelo.

As Figuras 13 e 14 mostram o modelo de persistência para o módulo *SysMap*.

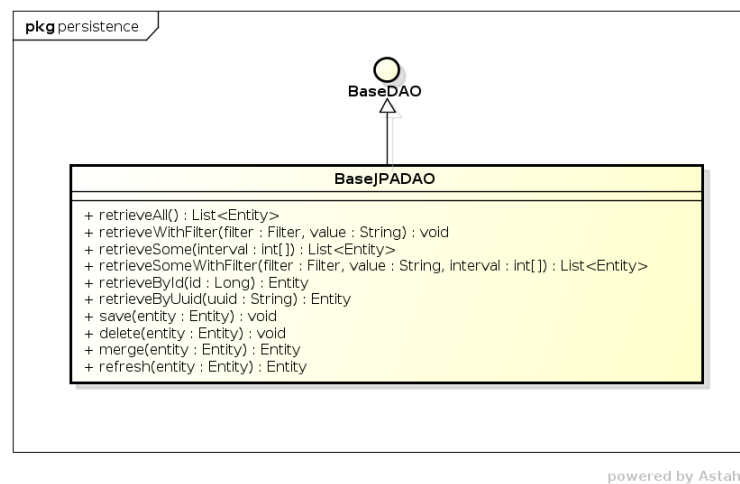


Figura 12 – Modelo de Persistência genérico da ferramenta JButler (LIMA, 2015).

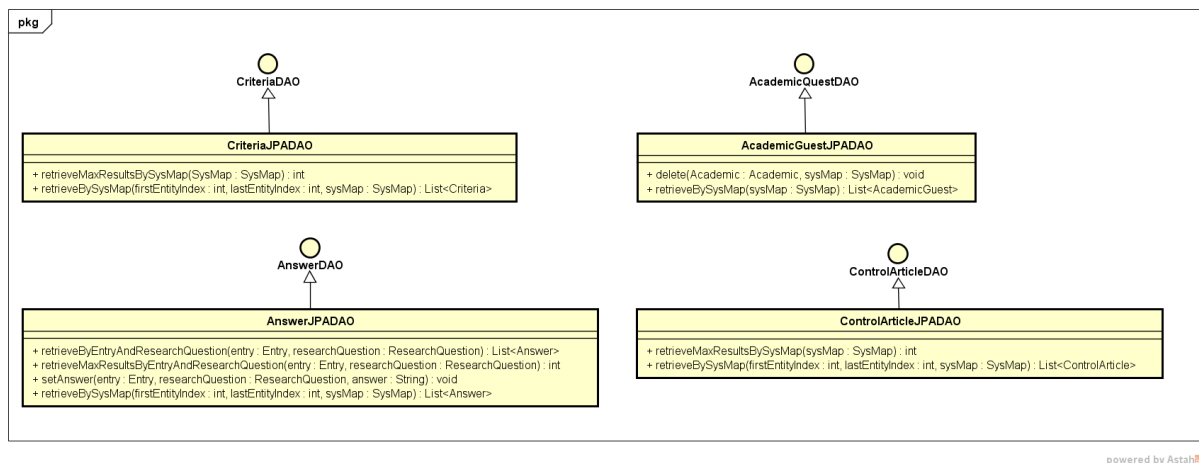


Figura 13 – Parte do Modelo de Persistência do Marvin para o módulo *SysMap*.

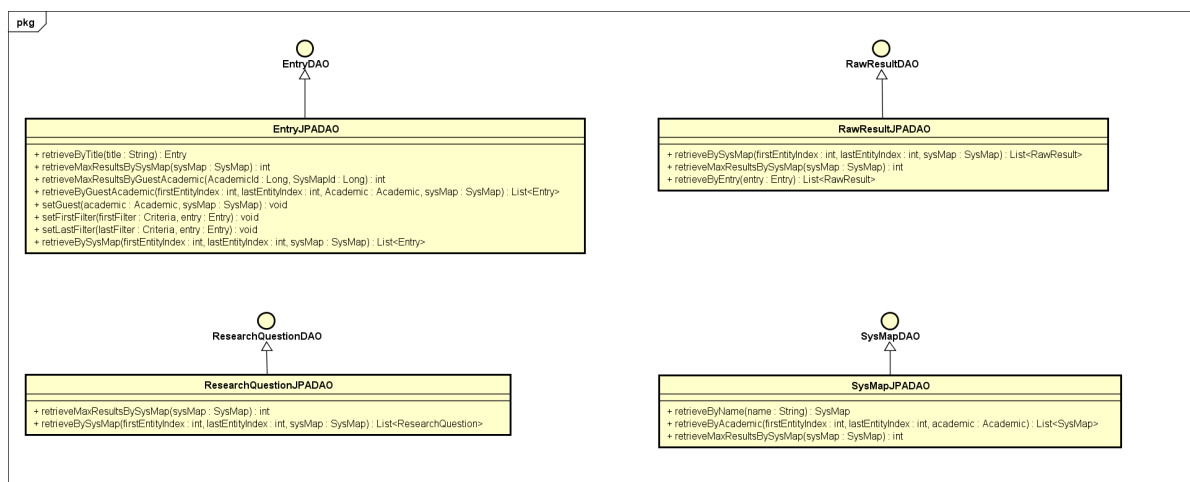


Figura 14 – Parte do Modelo de Persistência do Marvin para o módulo *SysMap*.



# Referências

COSTA, T. M. *Resgate - Despacho de ambulância automatizado*. Monografia (Projeto de Graduação) — Universidade Federal do Espírito Santo, 2015. Citado na página 7.

LIMA, L. V. F. *SAP - Sistema de Apoio ao Professor*. [S.l.], 2015. Citado 3 vezes nas páginas 6, 10 e 12.

*FrameWeb - A Framework-based Design Method for Web Engineering*. Citado na página 5.