

**UNIVERSIDADE DE RIO VERDE (UniRV)**  
**ENGENHARIA DE SOFTWARE**

**DOUGLAS PIMENTEL BARBOSA JÚNIOR**

**DESENVOLVIMENTO DE UM PORTAL DE CONTEÚDOS E ACERVOS  
ACADÊMICOS PARA DISCENTES E DOCENTES DA FACULDADE DE  
ENGENHARIA DE SOFTWARE DA UNIVERSIDADE DE RIO VERDE**

**RIO VERDE - GO**  
**2016**

**DOUGLAS PIMENTEL BARBOSA JÚNIOR**

**DESENVOLVIMENTO DE UM PORTAL DE CONTEÚDOS E ACERVOS  
ACADÊMICOS PARA DISCENTES E DOCENTES DA FACULDADE DE  
ENGENHARIA DE SOFTWARE DA UNIVERSIDADE DE RIO VERDE**

Monografia apresentada à Banca Examinadora do Curso de Engenharia de *Software* da Universidade de Rio Verde (UniRV) como exigência parcial para obtenção do título de Engenheiro de *Software*.

Orientador: Prof. Esp. Edgar Pereira dos Santos

**RIO VERDE - GO**

**2016**

**DOUGLAS PIMENTEL BARBOSA JÚNIOR**

**DESENVOLVIMENTO DE UM PORTAL DE CONTEÚDOS E ACERVOS  
ACADÊMICOS PARA DISCENTES E DOCENTES DA FACULDADE DE  
ENGENHARIA DE SOFTWARE DA UNIVERSIDADE DE RIO VERDE**

Monografia apresentada à Banca Examinadora do Curso de Engenharia de *Software* da Universidade de Rio Verde (UniRV) como exigência parcial para a obtenção do título de Bacharel em Engenheiro de *Software*.

Rio Verde, GO, \_\_\_\_ de \_\_\_\_\_ de 2016.

**BANCA EXAMINADORA**

.....  
Prof. Esp. Edgar Pereira dos Santos  
Presidente

.....  
Prof. Me. Fabian Correa Cardoso  
Examinador

.....  
Prof. Me. Márcio Rubens Sousa Santos  
Examinador

## **DEDICATÓRIA**

Desejo a toda Faculdade de Engenharia de *Software* (FASOFT) da Universidade de Rio Verde que encare este projeto como uma melhoria para seus estudos assim como dedico a todos os discentes e docentes que fazem parte desta família e que façam bom uso do mesmo, enriquecendo constantemente o conhecimento e fortalecendo nossa área e mercado.

Dedico, também, a todos interessados pela área que ainda não fazem parte da Engenharia de *Software* mas que possam adotar este futuro em suas vidas, que saibam da existência deste projeto, desta ideia e que a cada dia se fortaleça e nos fortaleça, produzindo uma base de conhecimento sem limites de expansão.

## **AGRADECIMENTOS**

Em primeiro lugar, venho agradecer a Deus pela oportunidade e por me permitir investir e dar continuidade nesta área independente de qualquer dificuldade que foi encontrada do início até a trajetória final.

Com grande consideração e agradecimento venho aqui gratificar ao meu orientador Prof. Esp. Edgar Pereira dos Santos e também ao meu grande conselheiro Prof. João Graciano Silva Filho, pela atenção, disponibilidade, dedicação e comprometimento como educadores e orientadores perante o projeto e a mim.

Nosso querido Prof. Me. João Dionísio Paraíba por dispor de parte de seu tempo auxiliando, ouvindo e apresentando dicas ao projeto, dando ênfase a avaliação e no resultado final esperado.

## **RESUMO**

No atual cenário, é possível deparar-se com a necessidade de buscar conhecimento extra para auxiliar na elaboração de atividades e entendimento de conteúdo, onde, encontra-se algumas vezes, carência de informações voltadas para Engenharia de *Software*, além de conteúdos desatualizados. Outro fator relacionado que se sabe, é a existência de conteúdos fisicamente degradados. Tais problemas foram resolvidos através do desenvolvimento de uma aplicação *web* que fornece ao usuário docente uma maneira simples de publicar arquivos em um formato padrão, no caso PDF, onde os discentes acessam esta mesma aplicação e conseguem buscar conteúdo nela dispostos, voltados de preferência para temas e disciplinas do curso em específico. A dificuldade de acesso sem uma ferramenta digital é um dos principais motivos para a criação do projeto, pois quando a informação só pode ser adquirida através de material físico, sabe-se que o interessado deverá se deslocar até o local onde se encontra este material, como uma biblioteca por exemplo. O *site* proporciona a publicação de arquivos, leitura e também obtenção do mesmo. Além disso, possui filtros de busca intuitivos e facilitados para o usuário. Para desenvolver a aplicação *web* foram utilizadas tecnologias e linguagens de programação como HTML, CSS e *JavaScript* que operam no lado cliente e nelas foram implantadas as telas e suas interações, no lado servidor, encontra-se o PHP sendo responsável por processar as requisições e realizar operações no banco de dados MySQL. Com o projeto finalizado, aplicou-se testes de *software* para verificação e validação através de três ferramentas, sendo elas, RIPS para análise estática automatizada voltada para parte de vulnerabilidades, PHPUnit para teste unitário em trecho de código e *Badboy* com teste funcional automatizado que atua no tempo de resposta das páginas *web*, onde foi garantido resultados satisfatórios em ambos os testes.

Palavras-chave: Conhecimento. Conteúdos. Acesso. Informação.

## **ABSTRACT**

In the current scenario, it is possible to be faced with the need to seek extra knowledge to assist in the preparation of activities and understanding of content, where, is sometimes lack of information facing software engineering, in addition to outdated content. Another factor that is known, is the existence of content physically degraded. Such issues were resolved through the development of a web application that provides the user with professor a simple way to publish files in a standard format, in case pdf, where the learners accessing this same application and they can fetch content therein prepared, face of preference for themes and course subjects in specific. The difficulty of access without a digital tool is one of the main reasons for the creation of the project, because when the information can only be acquired through physical media, it is known that the person concerned should move to where this material, as a library for example. The site provides the publication of files, reading and also obtain the same. In addition, features intuitive search filters and facilitated for the user. To develop the web application were used technologies and programming languages such as HTML, CSS and JavaScript that operate on the client side and in them were implanted the screens and their interactions, on the server side, is the PHP being responsible for processing the requisitions and perform operations on the MySQL database. With the project finalized, it was applied software testing to verification and validation through three tools, them being, RIPS for automated static analysis facing part of vulnerabilities, PHPUnit for unit test in code snippet and Badboy with functional test automated acting on the response time of web pages, where was guaranteed satisfactory results in both tests.

Keywords: Knowledge. Content. Access. Information.

## LISTA DE FIGURAS

|  |    |
|--|----|
| FIGURA 1 - Estrutura básica de uma página em HTML.....   | 27 |
| FIGURA 2 - Exemplo de imagens criadas somente com CSS.....   | 32 |
| FIGURA 3 - Botões modelados pelo <i>framework bootstrap</i> .....  | 35 |
| FIGURA 4 - Primeira questão do levantamento de dados .....   | 40 |
| FIGURA 5 - Segunda questão do levantamento de dados .....  | 41 |
| FIGURA 6 - Terceira questão do levantamento de dados.....  | 41 |
| FIGURA 7 - Quarta questão do levantamento de dados .....   | 42 |
| FIGURA 8 - Diagrama de Caso de Uso eu contempla todos os requisitos funcionais do sistema .....          | 43 |
| FIGURA 9 - Diagrama de Classe do Projeto .....   | 44 |
| FIGURA 10 - Diagrama de Sequência para emissão de relatório de editoras .....                            | 45 |
| FIGURA 11 - <i>Workbench</i> , ferramenta utilizada na criação da modelagem do banco de dados .....      | 46 |
| FIGURA 12 - <i>Astah Community</i> , ferramenta utilizada para elaboração dos diagramas no projeto ..... | 48 |
| FIGURA 13 - <i>GanttProject</i> , ferramenta utilizada na elaboração do cronograma do projeto ..         | 49 |
| FIGURA 14 - <i>NetBeans</i> 8.1, IDE de Desenvolvimento .....  | 53 |
| FIGURA 15 - <i>NetBeans</i> 8.1, algumas opções de arquivos.....   | 54 |
| FIGURA 16 - Editor de Código Brackets .....  | 54 |
| FIGURA 17 - Tela para realizar <i>upload</i> de arquivo, ou seja, publicar o arquivo no sistema ..       | 59 |
| FIGURA 18 - Janela para selecionar arquivo para realizar <i>upload</i> no sistema.....                   | 61 |
| FIGURA 19 - Visualização do arquivo selecionado e pronto para ser inserido no sistema....                | 61 |
| FIGURA 20 - Janela modal para realização de cadastro de novo tema .....                                  | 64 |
| FIGURA 21 - Janela de alerta do <i>framework lobibox</i> quando há uma validação em ativa.....           | 65 |

|   |    |
|---|----|
| FIGURA 22 - Tela de pesquisar arquivo disponibilizada para discentes, sem necessidade de <i>login</i> ..... | 72 |
| FIGURA 23 - Tela de visualização do arquivo após efetuada a busca .....                                     | 74 |
| FIGURA 24 - Tela de visualização do arquivo após efetuada a busca .....                                     | 75 |
| FIGURA 24 - Tela do primeiro requisito funcional, autenticação de <i>login</i> .....                        | 76 |
| FIGURA 25 - Tela do segundo requisito funcional, manter usuário com o perfil de administrador .....         | 77 |
| FIGURA 26 - Tela do segundo requisito funcional, manter usuário com o perfil de docente.                    | 77 |
| FIGURA 27 - Tela do terceiro requisito funcional, manter tema.....  | 78 |
| FIGURA 28 - Tela do quarto requisito funcional, manter autor .....  | 78 |
| FIGURA 29 - Tela do quinto requisito funcional, manter editora .....  | 79 |
| FIGURA 30 - Tela do sexto requisito funcional, manter disciplina.....                                       | 80 |
| FIGURA 31 - Tela de manter disciplina, operação de cadastro de nova disciplina.....                         | 80 |
| FIGURA 32 - Tela de manter disciplina, operação de editar nome da disciplina existente.....                 | 81 |
| FIGURA 33 - Tela de manter disciplina, operações de excluir ou inativar disciplina.....                     | 82 |
| FIGURA 34 - Tratativa de erro para evitar disciplina de ser removida quando há arquivos ligados a ela.....  | 82 |
| FIGURA 35 - Tratativa de erro para evitar disciplina de ser inserida com nome já existente.                 | 83 |
| FIGURA 36 - Tela do sétimo requisito funcional, manter palavra chave .....                                  | 84 |
| FIGURA 37 - Tela do oitavo requisito funcional, manter tipo de arquivo .....                                | 84 |
| FIGURA 38 - Tela do nono requisito funcional, manter arquivo .....  | 85 |
| FIGURA 39 - Tratativa que valida se todos os parâmetros foram inseridos para liberar anexo de arquivo ..... | 86 |
| FIGURA 40 - Tela de edição e exclusão de arquivo, requisito manter arquivo.....                             | 87 |
| FIGURA 41 - Tela do décimo requisito funcional, pesquisar arquivo.....                                      | 88 |
| FIGURA 42 - Tela do décimo primeiro requisito funcional, visualizar arquivo .....                           | 89 |

|  |     |
|--|-----|
| FIGURA 43 - Tela do décimo segundo requisito funcional, obter arquivo, utilizando o navegador <i>Google Chrome</i> ..... | 90  |
| FIGURA 44 - Tela do décimo segundo requisito funcional, obter arquivo, utilizando o navegador Mozilla Firefox.....       | 91  |
| FIGURA 45 - Tela do décimo terceiro requisito funcional, emitir relatório de usuários .....                              | 92  |
| FIGURA 46 - Tela do décimo quarto requisito funcional, emitir relatório de disciplinas.....                              | 93  |
| FIGURA 47 - Tela do décimo quinto requisito funcional, emitir relatório de autores .....                                 | 93  |
| FIGURA 48 - Tela do décimo sexto requisito funcional, emitir relatório de temas .....                                    | 94  |
| FIGURA 49 - Tela do décimo sétimo requisito funcional, emitir relatório de editoras.....                                 | 94  |
| FIGURA 50 - Tela do décimo oitavo requisito funcional, emitir relatório de palavra-chave..                               | 95  |
| FIGURA 51 - Tela do décimo nono requisito funcional, emitir relatório de arquivos .....                                  | 96  |
| FIGURA 52 - Tela do vigésimo requisito funcional, emitir relatório de tipo de arquivo .....                              | 96  |
| FIGURA 53 - Tela do vigésimo primeiro requisito funcional, emitir relatório por data de publicação .....                 | 97  |
| FIGURA 54 - Resultado do teste de análise estática realizado na ferramenta <i>RIPS</i> .....                             | 98  |
| FIGURA 55 - Janela de resultados da execução do teste de <i>software</i> com <i>PHPUnit</i> .....                        | 101 |
| FIGURA 56 - Janela de saída da execução do teste de <i>software</i> com <i>PHPUnit</i> .....                             | 102 |
| FIGURA 57 - Relatório gerado pelo teste funcional automatizado com <i>Badboy</i> v.2.2.5 .....                           | 102 |
| FIGURA 58 - Histórico do script com ações realizadas e parâmetros acessados via <i>Badboy</i> .....                      | 103 |

## **LISTA DE ABREVIATURAS E SIGLAS**

AOL – América Online.

CRB1 – Conselho Regional de Biblioteconomia - Região 01.

CRUD – *Create, Read, Update, Delete.*

CSS – *Cascading Style Sheets.*

DERS – Documento de Especificação de Requisitos de *Software*.

HTML – *HyperText Markup Language.*

IE – Internet Explorer.

MVC – *Model View Controller.*

PDF – *Portable Document Format.*

PDO – *PHP Data Object*

PHP – *Personal Home Page.*

SGBD – Sistema Gerenciador de Banco de Dados.

TV – Televisão.

URL – *Uniform Resource Language.*

# SUMÁRIO

|  |    |
|--|----|
| <b>1 INTRODUÇÃO .....</b>                                | 13 |
| <b>2 REVISÃO BIBLIOGRÁFICA .....</b>                     | 15 |
| 2.1 ACERVO.....  | 15 |
| 2.1.1 Acervo digital .....                               | 16 |
| 2.1.2 Direito do autor.....                              | 17 |
| 2.1.3 Situação atual (bibliotecário).....                | 18 |
| 2.1.4 Controle do acervo .....                           | 19 |
| 2.2 PORTAIS.....   | 19 |
| 2.2.1 Conteúdos de portais .....                         | 20 |
| 2.2.2 Tipos de portal .....                              | 22 |
| 2.2.3 Disponibilidade.....                               | 22 |
| 2.3 BIBLIOTECONOMIA E ARQUIVOLOGIA.....                  | 23 |
| 2.4 CIÊNCIA DA INFORMAÇÃO VOLTADA A BIBLIOTECONOMIA..... | 23 |
| 2.4.1 Mudanças.....                                      | 24 |
| 2.4.2 Conceitos e definições pós mudanças .....          | 24 |
| 2.4.3 Reconhecimento da área no Brasil .....             | 25 |
| 2.5 HTML .....   | 25 |
| 2.5.1 Estrutura básica de um documento HTML.....         | 26 |
| 2.6 URL E LINKS .....                                    | 27 |
| 2.7 INTERNET .....                                       | 28 |
| 2.7.1 Benefícios da internet para o conhecimento.....    | 29 |
| 2.8 PHP .....  | 30 |
| 2.9 CSS .....  | 31 |
| 2.10 BANCO DE DADOS .....                                | 32 |
| 2.10.1 Sistema gerenciador de banco de dados (SGBD)..... | 33 |
| 2.11 JAVASCRIPT .....                                    | 33 |
| 2.12 FRAMEWORKS .....                                    | 34 |
| 2.13 NAVEGADORES .....                                   | 35 |
| 2.14 TESTE DE SOFTWARE .....                             | 36 |
| 2.14.1 Teste de verificação e validação.....             | 36 |

|   |     |
|---|-----|
| 2.14.2 Garantia da qualidade .....                  | 38  |
| <b>3 MATERIAIS E MÉTODOS</b> .....                  | 40  |
| 3.1 LEVANTAMENTO DE DADOS .....                     | 40  |
| 3.1.1 Gráfico das amostras .....                    | 40  |
| 3.2 Diagramas .....                                 | 42  |
| 3.2.1 Diagrama de caso de uso .....                 | 42  |
| 3.2.2 Diagrama de classe .....                      | 43  |
| 3.2.3 Diagrama de interação (ou de sequência) ..... | 44  |
| 3.3 WORKBENCH.....                                  | 45  |
| 3.3.1 Dicionário de dados .....                     | 46  |
| 3.4 ASTAH COMMUNITY .....                           | 47  |
| 3.5 GANTTPROJECT .....                              | 48  |
| 3.6 REQUISITOS FUNCIONAIS .....                     | 49  |
| 3.7 AMBIENTE DE DESENVOLVIMENTO .....               | 50  |
| <b>4 RESULTADOS E DISCUSSÕES</b> .....              | 76  |
| <b>5 CONCLUSÃO</b> .....                            | 105 |
| <b>REFERÊNCIAS</b> .....                            | 107 |
| <b>ANEXOS</b> .....                                 | 109 |

## 1 INTRODUÇÃO

Encara-se atualmente com problemas em que se interessa obter conhecimento em algum tema ou disciplina relacionada a Engenharia de *Software*, como, dificuldade na localização de alguns conteúdos, pouca opção em pesquisa de informações provenientes de algum embasamento científico ou relevância como possui em livros ou artigos publicados, conteúdos demasiadamente obsoletos, o que pode acarretar um grande problema devido a constante atualização e crescimento da área envolvida e da tecnologia que a cerca.

A tecnologia hoje é um grande avanço e este avanço é responsável por algumas mudanças na sociedade, um dos marcos tecnológicos foi a internet que causou grande impacto e foi capaz, até mesmo, de mudar o comportamento cotidiano das pessoas. Com isso surgiram novas possibilidades de comunicação, relacionamento, interatividade e o fator principal que é levar a informação para agir o mundo todo, desde que conectados à rede. A internet provê a informação através de portais de acesso, eles podem ser educativos, institucionais, acadêmicos, empresariais, pessoais, organizacionais, governamentais, dentre outros e seja lá qual for o tipo de portal, sua finalidade será disponibilizar conteúdo ao público conectado à rede, durante o dia todo, todos os dias da semana, é simplesmente o meio de comunicação mais fantástico que já se conhece.

A sociedade apresentou mudanças por meio de tamanho avanço tecnológico, passaram-se a ser mais dependente dela em atividades particulares, no trabalho e até mesmo no lazer, e fazem com que fiquem em maior contato com o computador, desta forma, é de suma importância tratar este assunto e a proposta de uma ferramenta para disponibilizar acesso à informação voltada a um público específico pode ser uma boa ideia de se tratar o tema.

O objetivo geral é desenvolver uma aplicação *web* que, de maneira simples e viável, atenda às necessidades de um público que busca e compartilha informação em boa parte de seu tempo, fazendo com que, um portal de conteúdos e acervos acadêmicos passe a ser um local que disponha de opções e funcionalidades ideais para disseminar o conhecimento entre discentes e docentes que buscam o mesmo caminho para aprendizado e ensino.

Pretende-se adquirir conhecimento para o desenvolvimento de um portal de conteúdo, tanto o conhecimento técnico dentro da área tecnológica, como também o conhecimento geral com base na situação atual que se encontra com maior frequência sem ser de forma digital, o

qual possui algumas limitações de acesso que não podem ser tratadas devido a ser caracteristicamente impossível de lidar, por exemplo, não é possível locomover uma biblioteca ou um acervo literário ou qualquer outro material tangível para o público alvo, é o público que precisa ter recursos para ir até onde está a informação de seu interesse, o que já pode ser tratado isto de maneira a que este conteúdo seja armazenado de forma digital.

Outro pretexto para a elaboração deste *site* é o fato de que algumas pessoas, sentem-se desconfortáveis em realizar pesquisas através de livros ou artigos, onde estes materiais, não são dispostos com facilidades para encontrar o assunto desejado na maioria das vezes, ocupando-se maior tempo para encontrar o que precisa e que facilmente podem ser encontrados com partes danificadas, além de nem todos estarem disponíveis para leitura fora do ambiente apropriado.

Ao se pensar por outro ponto de vista, é também uma maneira de atingir as pessoas que são mais familiarizadas com a era digital, os jovens deste tempo moderno na maioria das vezes passam maior parte de seu tempo utilizando o computador para suas diversas tarefas do dia a dia, para ocupar o tempo com lazer e inclusive para realizar trabalhos escolares, o que se dirige diretamente à necessidade de pesquisa e busca da informação. Ao trabalhar este ponto de vista, torna-se mais simples direcionar a informação a este tipo de público, pois lida-se com pessoas que são hábeis a utilizar deste tipo de informação digital, isto é, a interatividade, relacionamento e principalmente interesse serão melhores trabalhados se for possível levar a estas pessoas, conteúdos e informações de modo em que elas estão acostumadas a interagir, além de economizar tempo que seria gasto em recursos para ir em busca de informação, alguns recursos financeiros também, isto poderá ser investido de melhor forma para ter um conteúdo digital.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 ACERVO

A definição de um acervo pode ser dada como uma coleção de elementos de origem humana ou ambiental, organizados e mantidos informados do assunto relacionado com que se tratam. Assim como documentos, obras de arte ou acontecimentos (MASON; WEEKS, 2004).

Sabe-se, então, que o acervo pode ser público ou privado, que possui diversas características, ou melhor dizendo, diversas áreas, sendo assim, pode-se caracterizar acervo como a aglomeração de obras artísticas, bibliográficas, científicas, documentais, históricas, dentre outros campos.

No gerenciamento de um museu, espera-se que padrões sejam alcançados, como segurança, infraestrutura predial para a finalidade do tipo de museu, as necessidades tanto do público quanto dos funcionários e até mesmo de seu acervo (MASON; WEEKS, 2004).

Um acervo merece e espera-se que tenha um capricho no cuidado sobre ele, no caso de museu, o acervo é incluso na parte de gerenciamento do patrimônio e de instalações, dependendo do tipo do museu, ou seja, da função que ele exerce, seu acervo pode ser bem variado, o que pode encontrar desde obras de arte, peças de arqueologia, diversos elementos de patrimônio histórico e também é claro as obras literárias que podem e geralmente estão inclusive categorizadas não só como uma obra literária mas também como patrimônio histórico e outros se for o caso. O valor que o acervo tem para um museu é grandioso, algumas vezes incalculável pois muitos tratam de materiais não comuns à sociedade atual e que possuem relatos e histórias de antepassados sobre a humanidade, grandes nomes de estudiosos e suas descobertas, em contexto geral, do mundo como um todo.

Não se espera que um acervo bibliotecário seja tão diferente do acervo de um museu independente de sua categoria, claro que cada um tem suas particularidades, porém as importâncias são de certa forma interligadas. As necessidades organizacionais e as necessidades do funcionário e de seu público são pontos importantes para qualquer instituição, pois estas tratam de cuidados materiais, culturais e informativos, lidam-se diretamente com a informação e as oferecem para seu público, esta é a ideia que também se aplica a um portal de conteúdos e acervos digitais.

### 2.1.1 Acervo digital

Franciscon (2013) aponta que, tornar um elemento para a forma digital, faz com que sejam armazenadas em banco de dados qualquer informação referente a ele são armazenadas em bancos de dados e ao constituir a coleção destes elementos com o tempo, tornam-se um acervo considerável.

Seguindo o mesmo conceito de acervo convencional, o acervo digital aponta para todo o conjunto público ou privado, independente do que se trata, porém com sua disponibilidade pode ser realizada através da internet. A grande diferença entre acervo convencional e acervo digital é que o conteúdo do acervo digital é de certa forma, intangível, isto é, não se tem contato direto ao material e/ou conteúdo seja qual for sua natureza. Ao contrário, o acervo não digital é tangível e se pode ter o contato físico direto com o material a qual pertence seu conteúdo. Cada tipo de acervo é proveniente de suas características e qualidades específicas.

Mello (2012), citado por Serra (2014) especifica que atualmente se vivencia uma comunidade feita por jovens da era digital que geralmente fica a maior parte do dia atrás do computador ou celular conectadas à internet, expostas a receber diversas informações a todo instante.

Sabe-se que não são apenas os jovens, hoje se conta também com uma grande parte da sociedade em todas as faixas etárias que estão sempre conectados à internet de alguma forma, tanto em casa como no trabalho ou no lazer. Aproveitando esta mudança de comportamento comparado há décadas atrás e compreendendo que estes tempos modernos é uma completa realidade, pode-se usufruir do meio digital para agregar valores de conhecimentos, compartilhar informações, criar oportunidades e divulgar para as pessoas estes dados que são de interesse para ambos. A forma de se comunicar, manter-se relacionado ou interligado às informações e às demais pessoas é algo bastante fácil, simples e comum com o uso da internet e para os que sabem utilizar desta ferramenta a seu favor, podem obter resultados satisfatórios em seus produtos ou serviços, como é o caso dos portais de conteúdo, por exemplo, e outras ferramentas que estão disponíveis de forma digital.

## 2.1.2 Direito do autor

Na visão de Menezes (2007), para se tornar alvo de proteção jurídica da obra ou do bem, é preciso identificá-lo, isto é, registrá-lo a sua devida autoridade para que o mesmo tenha os direitos legais.

Também conhecido como Direito Autoral, o direito do autor refere-se a obras propriamente elaboradas, e podem elas serem de uma entre as diversas naturezas, ou seja, uma obra artística, obra literária ou intelectual por exemplo. Dependendo da criação, para cada propriedade existe seus direitos autorais e cabe ao autor tornar-se apto a ser protegido pelas leis de autoria existentes.

Menezes (2007) esclarece que o Direito de Autor tem como finalidade guardar uma obra pessoal sob seus direitos e penas da lei, seja ela literária ou artística mantendo-a protegida e referenciada a seu devido autor.

Com isso, o direito autoral no Brasil por ser uma área que além de ter sido vastamente enriquecida, proveniente do crescimento de certos aspectos culturas e principalmente tecnológicos, surgiram necessidades de revisar certos conceitos nas leis, pois o impacto da era digital não era algo tão previsto de acontecimentos e a quantidade de informações e oportunidades que ela abriu, despertou o interesse de muitos.

De acordo com a lei nº 9.610, o autor de uma obra pessoal independente de sua característica possui o direito próprio para autorizar sua reprodução ou exploração de tal criação e pode ele ser concedido ou não. É o que se diz proteção ao direito do autor (ROCHA, 2001).

Sendo assim, sabe-se que existem leis e direitos legais de proteção ao autor em diversos casos que envolvem sua matéria de criação intelectual, assim como também artesanal ou de qualquer natureza. Este é um assunto jurídico do qual o criador de alguma obra sem sombra de dúvidas precisa se atentar em ter o conhecimento e se aprofundar nas leis para concretizar que seu direito seja prevalecido em seu nome e que fique por dentro para não deixar passar nenhuma possibilidade de intrusão a sua obra por terceiros.

### 2.1.3 Situação atual (bibliotecário)

Em uma reunião com a bibliotecária Izaura Ferreira Neta – CRB1:2771, presente na biblioteca da Universidade de Rio Verde em cinco de outubro de dois mil e quinze, realizou-se o levantamento de alguns itens da situação atual, ou seja, os processos que fazem parte da estrutura do acervo bibliográfico.

Izaura ainda explica que, para manter o acervo bibliográfico não é somente receber os materiais e colocá-los na estante, tem uma rotina específica que trata de diversos aspectos antes do material tomar qualquer destino direto ao público, trata-se do processamento técnico dos livros.

Com a chegada de material na biblioteca, é realizado uma leitura dinâmica para classificar e catalogar o material. É criado o número de classificação que é o endereço que vai para a estante e depois é inserido o material no sistema. Este número de classificação não é aleatório e só pode ser criado por um bibliotecário, este traduz o assunto central do livro.

É preciso identificar na catalogação pelo menos três palavras chave, ou seja, assuntos principais do livro, para então inseri-lo no sistema e quando vier a procurar, conseguir fazer a recuperação do material para indicar ao aluno. No sistema atual, os exemplares são catalogados por títulos, onde por exemplo dez exemplares idênticos são atribuídos ao mesmo título e/ou assunto, assim é inserido no sistema as dez unidades de uma só vez. Existem também as etiquetas, que são impressas e atribuídas ao material, no caso delas, são impressas uma para cada exemplar e no caso de dez, serão dez trabalhos manuais por vez, um para cada etiqueta em cada material.

Izaura também comentou a questão de disponibilização do material, pois ressaltou que como a biblioteca é de acesso público e não pode ser impedido que qualquer pessoa tenha acesso a seu material, existe uma certa análise do conteúdo que ali é disponibilizado, isto é, nem todo material pode ir para as prateleiras e ficar em aberto ao público. Essa regra funciona para o seguinte propósito, caso a biblioteca receba livro de caráter científico ou específico em uma determinada área, citou-se exemplo de livros específicos de psicologia com ensinamentos de experimentos ou testes a serem realizados em pacientes, onde para ter acesso a este tipo de conteúdo, é necessário ser psicólogo, não sendo um público qualquer com certo interesse de pesquisa em ir lá e adquirir tal conhecimento sem antes entender do assunto. Mas quem decide se este conteúdo é impróprio ou não, não cabe a bibliotecária, então é realizado um contato com o diretor do curso para se obter informação dos materiais, ele é quem informa

se o material deve ser restrito ao público específico ou não, isto fica no processo de catalogação.

Pode-se observar que há uma série de fatores envolvidos para disponibilizar um material ao público aberto, aproximadamente este processo leva cerca de trinta minutos por título e caso tenha oitenta exemplares do mesmo título por exemplo, seriam gastos os trinta minutos além de aproximadamente cinco minutos individuais para o carimbo e colagem de etiquetas para cada exemplar.

#### 2.1.4 Controle do acervo

De acordo com a reunião realizada com a bibliotecária Izaura Ferreira Neta, por questões de organização, é interessante ter certos controles sobre um acervo, no caso de materiais físicos, pode-se observar que este controle é realizado em sua maior parte de forma manual através de funcionários ou voluntários responsáveis por manter determinado acervo.

No caso do acervo digital, há mecanismos que podem auxiliar a manter o controle de determinadas formas diferentes, mas o que auxilia a manter tudo em ordem mesmo é o banco de dados e claro também precisa-se de trabalho manual para sua configuração e criação, porém as possibilidades de automação de tarefas que ele permite a quem gerencia o conteúdo, é bem mais segura, confiável e melhor gerenciável do que através de materiais tangíveis.

### 2.2 PORTAIS

No entendimento de Gebran (2009), a definição de um portal pode ser dada como uma página de internet, que aglomera e distribui serviços e/ou produtos para usuários ou outros consumidores destas informações, desde que organizados e classificados por área de interesse.

Conclui-se que cada página disponível na rede, é um portal, agora basta caracterizar o tipo deste portal, porém, independente de qual seja o tipo ou para que cada porta está direcionado, todos possuem basicamente o mesmo conceito que é a junção e distribuição de serviços ou produtos para o usuário que é o consumidor final das informações contidas nestes portais.

Ainda na concepção de Gebran (2009), os portais tiveram início na década de 90, através da AOL (América *Online*) e também da Compuserve<sup>1</sup>, que proviam diversos serviços através de portais restritamente a seus clientes.

Portais, voltado para o ambiente tecnológico e digital, nada mais é que um local onde se agrupam funcionalidades para determinado assunto. Portal pode ser também uma forma de acessar dados, melhor dizendo, um mecanismo para conciliar o acesso de um indivíduo a um determinado conteúdo, local ou material.

### 2.2.1 Conteúdos de portais

A internet é um universo de informações sem limites de expansão, com seu crescimento constante, surge a necessidade de ter maior controle sobre seu conteúdo, tais como identificação e categorização e os portais são apenas uma das formas de se fazer isto (GEBRAN, 2009).

A *web* é um portal de comunicação que permite incontáveis possibilidades de realizar tarefas diversas, por se tratar de comunicação, o seu maior ponto forte é a forma de conexão das informações com o público em geral. A internet atualmente afeta grandes quantidades de pessoa no mundo todo e o poder da divulgação somado com a facilidade de acesso faz uma união soberana para circular informações em qualquer lugar desde que seja possível ter acesso a esta imensa rede de dados.

Com a criação dos portais para atender algumas necessidades, foi possível unificar a informação e manipular suas atividades e interesses, não importando qual a sua categoria, seja ela educacional, corporativa ou outras (GEBRAN, 2009).

Neste conceito, entende-se que os portais são ricos de informações e conteúdo de diversos tipos de interesse, com isso, foi preciso investir em segurança, em mecanismos e formas de tratar tanto a informação para garantir que aquela seja de fato verídica, como também de evitar conflitos e infringir leis seja accidentalmente com informação divergente ou por terceiros que agem com intenção prejudicial, ou seja, alguns *hackers*. A informação devidamente gerenciada, armazenada e controlada é de fato segura, mas nunca se pode dizer completamente segura, pode-se concluir sobre isso devido às empresas de grandes nomes e

---

<sup>1</sup> Provedor de Serviço, assim como a AOL.

grandes organizações de sistema de informação que já sofreram invasões ou perdas em ataques cibernéticos.

Em seus apontamentos Gebran (2009) expõe que alguns portais são como caminhos que surgiram para atender certas expectativas e atribuir algo para a comunidade, como os portais educativos, e torna cabível ao usuário obter aprendizado às informações que lhes são de interesse.

Entende-se que no caso do portal de conteúdo digital, alguma finalidade ele possui e geralmente o foco é o usuário levar esta finalidade até ele, isto pode ocorrer de várias formas, dependendo do portal que o usuário está acessando, por exemplo, pode-se levar ao usuário a venda de um item de propriedade dele, ou a compra de um item de terceiros, pode-se levar a ele informações em tempo real, informações das quais ele deseja, algumas que são baseadas em suas buscas pela internet, independente do que seja, os portais hoje em dia sempre estão voltados a oferecer algo ao usuário, desde pequenas ofertas e oportunidades como também o conhecimento geral e específico.

As informações disponibilizadas pelos portais da internet auxiliam em um método de aprendizagem colaborativa, pois a quantidade de indivíduos dispostos a obter e oferecer conhecimento coletivo é um dos fatores que torna o conteúdo *web* tão vasto (GEBRAN, 2009).

O conceito de Gebran em 2009, é de total relevância, a quantidade de acessos que são realizados na internet é incomparável com qualquer outro meio de comunicação ou acessibilidade de informação, no século XXI, a internet está cada vez mais presente na vida das pessoas, tanto em casa quanto no trabalho ou universidades, mesmo com a dificuldade de alguns países subdesenvolvidos em investir na infraestrutura para trazer a população o serviço de internet com qualidade e preço acessível, é fato que hoje em dias as pessoas são alvos de dependências desta grande rede e esta comunhão de pessoas interligadas diretamente e indiretamente fez com que novas tendências fossem surgindo, principalmente de mercado, onde pessoas oferecem seus conhecimentos e os divulga para que outras pessoas adquiram também este conhecimento e na maioria das vezes isto gera reconhecimento e também muito dinheiro.

## 2.2.2 Tipos de portal

Eastman et al. (2008) identifica que há dois tipos de portais que vêm surgindo no mercado, são eles os portais públicos e os portais privados.

E sobre estes tipos, existem suas diferenças entre o público e o privado, a própria palavra já diz e isto são características que fazem certo diferencial nos portais, essa diferença é notada devido ao conteúdo a ser aplicado, isto é, um determinado portal voltado a um público específico, sendo necessário algum cadastro e aprovação, algum convite, possuir algum pré-requisito como o número de matrícula por exemplo, esta é a característica do portal privado, e que pode ser utilizado geralmente por empresas, instituições de ensino e até mesmo pelo exército.

Portais públicos oferecem informações para a comunidade através de recursos públicos, onde todos podem criar como também ter acesso aos conteúdos, fóruns e *blogs* são exemplos destes recursos. Neste tipo de portal, também encontram-se suporte a algumas ferramentas de auxílio ao usuário, é o caso das ferramentas de busca e pesquisa, *download* e *upload* de arquivos (EASTMAN et al., 2008).

Conhecendo, desta forma, que o portal público pode ter algumas características do privado ou vice-versa, também pode ser utilizado por instituições de ensino e talvez até por algumas empresas, depende do seu modo de uso, de como ele difere os acessos e também do que se trata, se pode ser ou não de acesso ao público geral.

## 2.2.3 Disponibilidade

No caso dos portais privados, eles permitem a comunicação e compartilhamento de seus conteúdos com quem possui as devidas permissões sendo gerenciados e controlados por um servidor. É mais comum em ambientes corporativos, como uma intranet por exemplo, sendo possível controlar o acesso para um público específico (EASTMAN et al., 2008).

Sabe-se, então, que portais de acesso, isto é, portais de conteúdo, são hospedados em servidores dedicados dos quais possuem autonomia para estarem dispostos vinte e quatro horas por dia, sete dias na semana, e alcançar margens de *uptime*<sup>2</sup> equivalentes a cem por

---

<sup>2</sup> “Tempo em Atividade”, traduzido do inglês, é o período que um computador fica operando atividades contínuas sem nenhuma forma de ininterruptão.

cento ou extremamente próximo a esta quantia, graças à infraestrutura que estes ambientes disponibilizam.

## 2.3 BIBLIOTECONOMIA E ARQUIVOLOGIA

Oliveira (2005) apontam que os consultores compreendem a biblioteconomia como disciplina que busca, organiza e propaga a informação contida em qualquer tipo de material. Também faz parte deste conceito a arquivologia.

Entende-se que ambas as disciplinas ligadas a busca e propagação de informação como objetivo aplicado está presente não apenas entre elas mas também em tudo que se faz, a informação é o primeiro passo para tudo, sem adquirir o conhecimento necessário sobre determinado assunto, sem ter a informação essencial sobre determinada coisa, tudo fica mais difícil, tudo se complica e na maioria das vezes quando se nota isso, cedo ou tarde, a única saída é: adquirir informação.

É notável que Ciência da Informação é conhecida como uma área extremamente vasta e que possui subáreas ligadas a ela e a Biblioteconomia é um exemplo disso, geralmente mais efetiva com sua parte de conceitos técnicos e assim se observa o quão são áreas flexíveis (OLIVEIRA, 2005).

Percebe-se que flexibilidade é a palavra chave para conjunto e união e são estes os fatores que enriquecem cada área, mesmo que independentemente falando de cada uma delas é quase impossível não destacar uma ligada a outra, pois são tão interligadas que o conceito entre elas são algumas vezes necessários para o entendimento.

## 2.4 CIÊNCIA DA INFORMAÇÃO VOLTADA A BIBLIOTECONOMIA

Oliveira (2005) entende que, ter o conhecimento de determinado assunto e este se tornar claro e entendível, é preciso saber mais sobre o contexto do assunto em si, desde a origem até outros fatores importantes que o marcaram.

Com o desenvolvimento econômico, cultural e social, a biblioteconomia ganhou espaço e se tornou conceituada na organização dos registros de documentação que são realizados com base em histórias, acontecimentos, pesquisas ou fatos que geram trabalho de autoria escrita e informativa através de livros ou outro tipo de documentos formais.

Atualmente, encontra-se em uma fase de transição de conhecimentos gerais de documentação com origem material para o mundo digital, isto influencia o público a ter uma nova visão sobre a biblioteca, a forma de adquirir conhecimento e buscar os conteúdos (OLIVEIRA, 2005).

São novos costumes, novas formas de acesso, novas oportunidades, comodidade, melhorias e demais conceitos com tal migração para a era digital, uma inovação que proporciona ganhos para muita gente e também dispõe de novos desafios para alguns, tudo para conseguir manter a ordem e organização acima de tudo nesta mudança que na verdade é uma união, pois os registros não deixaram de ser materialmente para serem virtuais e sim somaram além do material já conhecido, para uma nova forma que possui exatamente o mesmo conteúdo, porém de forma e características digitais.

#### 2.4.1 Mudanças

Oliveira (2005) ressalta que o desenvolvimento tecnológico que a informática trouxe após 1960, ocasionou grande impacto relacionado à informação. O computador e as telecomunicações criaram a necessidade de repensar sobre várias questões já existentes como também estudos que deveriam ser feitos para a área com tamanha inovação e mudança que a Ciência da Informação apresentou ao mundo.

Aparentemente a palavra informação tornou-se algo completamente diferente, onde Ciência da Informação passa a ser parte de demais áreas já conhecidas que de certa forma se interligavam devido aos avanços ocorridos; a Tecnologia da Informação também era algo a ser discutido com visões futuras.

#### 2.4.2 Conceitos e definições pós mudanças

Nos conceitos de Borko (1968), citado por Oliveira (2005), a Ciência da Informação categorizada como disciplina seria responsável por cuidar do comportamento e propriedades das informações, assim como a obtenção de seu acesso e uso, dentre outras relações.

Supõe-se que a Ciência da Informação na época em que aconteceram os grandes avanços da informática, foi o campo em que surgiu os maiores desafios para pesquisadores como também para todas as pessoas que de alguma forma eram direta ou indiretamente

ligadas a esta área e que com certeza notaram a diferença que ela fez e tem feito no decorrer do tempo.

Borko (1968), citado por Oliveira (2005) elucida que esta disciplina não foi somente ponto positivo, alguns dos desafios encontrados podem ser relatos de problemas, como o caso da organização e disponibilidade da informação voltado à produção cultural.

Entende-se que não havia propriamente uma definição certeira sobre o leque de opções que abrangia os avanços ocorridos, o que poderia infringir leis, direitos e assim ser de certa forma uma ameaça para algumas pessoas. Às vezes até mesmo algumas das opções que a informática proporcionou às pessoas nem havia ainda uma regulamentação e esta foi uma das necessidades que surgiu com as mudanças.

#### 2.4.3 Reconhecimento da área no Brasil

O Órgão Federal de Financiamento À Pesquisa No Brasil, Conselho Nacional de Desenvolvimento Científicos e Tecnológico (CNPq), reconhece conceituando-se a Ciência da Informação e Biblioteconomia em 1983 com finalidade em controlar e disponibilizar demandas financeiras de intuito à pesquisa. Realizado através de consultores de ambas as áreas e também da Arquivologia, a UNESCO auxiliou para que houvesse uma infraestrutura nacional de sistemas de informação (OLIVEIRA, 2005).

Desta forma, sabe-se que houve propósitos e interesses sobre a área e com isso o comitê que em conjunto ao órgão federal de financiamento à pesquisa no país, dentre outras parcerias, passa a não só reconhecer a disciplina como também a investir e dispor de recursos para que a mesma seja explorada nacionalmente com recursos próprios.

### 2.5 HTML

Pode-se caracterizar uma página HTML (*HyperText Markup Language*), qualquer arquivo de texto de extensão .HTM ou .HTML, mas para ser um conteúdo comprehensível pelos navegadores, dentro deste arquivo é preciso conter códigos especiais denominados *tags*, que têm diversas funções próprias (RAMALHO, 2005).

Desta forma, sabe-se que um *site* é composto por páginas HTML e as páginas HTML são compostas por códigos que em conjunto realizam determinadas funções que são

interpretados e assim visual e funcionalmente disponíveis para interação do usuário. Além de *site*, as páginas HTML podem ser apenas documentos de visualização ou manipulação através de navegadores, não vinculadas a um endereço da internet mas seguidos do mesmo conceito de sua estrutura de funções.

Ramalho (2005) reconhece que quando o navegador interpreta o arquivo HTML, toda característica do conteúdo e apresentação do mesmo é realizada e proveniente através do que estiver estabelecido nas *tags* e elas também podem direcionar para outras páginas HTML.

Entende-se, assim, que as *tags* são o esqueleto do documento HTML, definindo suas características e propriedades, ou seja, com as *tags* pode-se manipular o documento HTML varia de acordo com as necessidades de uso e caso o conteúdo apresente-se por suposta página possua ou precise de outro elemento, continuação ou extensão, pode-se aplicar o endereço de outra página e assim seguir o conteúdo acessado.

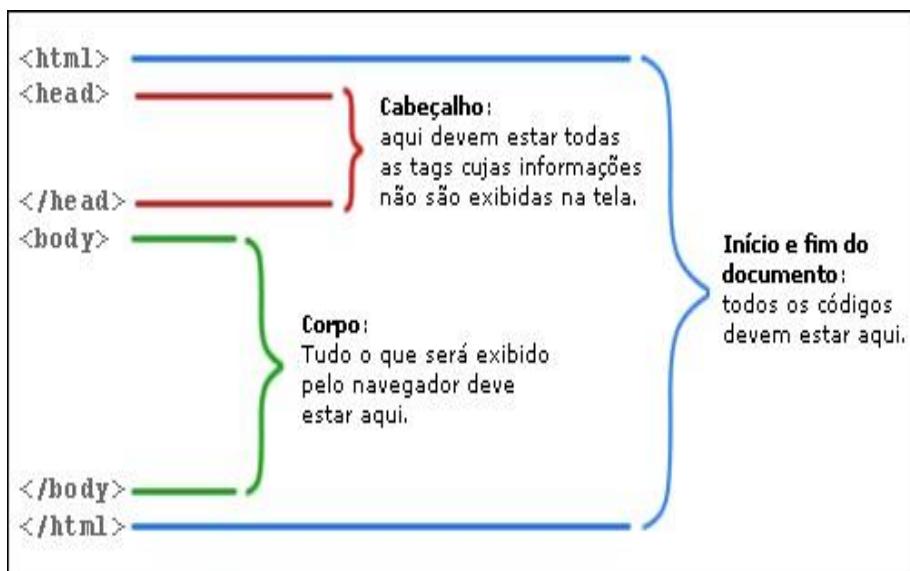
### 2.5.1 Estrutura básica de um documento HTML

Algumas características são essenciais na identificação e interpretação de conteúdo *web*, são elas as estruturas do documento, alguns comandos próprios definem esta estrutura e organizam seus componentes para serem apresentados (RAMALHO, 2005).

Desta forma, sabe-se que as *tags* classificam as propriedades dessa estrutura, e possuem os comandos necessários para manipulação dos componentes, alinhando assim todo o conteúdo da página *web*. Existem também *frameworks* que disponibilizam funções pré-definidas e outros auxílios para o desenvolvedor conseguir manipular todos os componentes e estruturas de um documento HTML geralmente de forma mais ágil, simples e robusta.

As *tags* possuem semânticas diferentes entre si, por exemplo, *<body></body>* são *tags* que representam o corpo da página e *<head></head>* são representações para o cabeçalho. Na Figura 1 é mostrada sua estrutura básica para formar um documento HTML.

**FIGURA 1** - Estrutura básica de uma página em HTML



Fonte: Site <http://www.educacaopublica.rj.gov.br/oficinas/informatica/html/estrutura.htm>, acessado em 24/05/2016 às 13:40.

Ramalho (2005) disserta que toda página em HTML é formada por três componentes básicos, sua estrutura principal, cabeçalho e o corpo do conteúdo, todos eles controlados pelos comandos essenciais da linguagem para seu devido funcionamento.

Ramalho (2005) diz que se pode observar estas estruturas presentes ao visualizar o código fonte de páginas HTML, que é possível ser realizado através dos navegadores inspecionando o elemento da página, algumas, inclusive já é notável na própria parte visual ao usuário, por ganharem algum destaque nestes componentes. Os componentes de uma página *web* ganharam avanços com o tempo e atualmente é de costume encontrar o resultado destes avanços presente em vários *sites*, devido ao crescimento de ferramentas como os *frameworks* e de técnicas avançadas no desenvolvimento para *web* em geral.

## 2.6 URL E LINKS

A URL ou *Uniform Resource Locator* é o caminho ou endereço de uma página *web*, mas também pode ser de um documento local. Este caminho é formado basicamente por três partes, são elas: protocolo do endereço ou documento, o caminho do servidor ou do documento local e a localização do arquivo desejado. Desta forma, com a junção destes itens, a sintaxe de uma URL é finalmente composta (RAMALHO, 2005).

Compreende-se então que a maneira de acesso ao documento, *site* ou qualquer conteúdo que seja de extensão .html, após possuir o acesso à internet, basta inserir no navegador o endereço, no caso, a URL seguindo a sua devida composição.

Quando se trata de informações extensas onde não ficam bem organizadas em um só documento, implementa-se o uso de hipertexto em *link* para separar os tópicos e mantê-los organizados por alguma categoria e ao clicar sobre o *link*, é realizado acesso ao conteúdo (RAMALHO, 2005).

Desta forma, pode-se também utilizar o *link* para outras finalidades, como a chamada de um *site* ou alguma página HTML qualquer, é característico do hipertexto realizar métodos de chamada de conteúdo, funcionalidades ou alguma ação. Além de organizar, é bastante utilizado para referenciar pois ele permite que escreva um nome desejado que será destacado na página em cor predominante azul e sublinhado e ao passar o mouse sobre ele, é possível observar no próprio navegador para onde o *link* se destina.

## 2.7 INTERNET

Ramalho (2005) ensina que um dos maiores atrativos que a internet trouxe foi a possibilidade de trazer uma informação de qualquer lugar do mundo para seu próprio computador da maneira mais simples possível, ou seja, com alguns cliques do *mouse*. Essa interação com o mundo proporcionou uma série de possibilidades que independentemente de onde esteja o conteúdo e o usuário, desde que conectados à internet, havia uma forma de ter acesso a ele.

Compreende-se que a internet é o meio de comunicação que por si só agrupa uma imensidão de recursos capazes de proporcionar ao usuário formas e mais formas de realizar tarefas, executar ações, acessar informações, entre outras atividades. A internet foi capaz de unir as pessoas com uma interatividade incrível e a sua gama de recursos, inovação e desenvolvimento foram pontos primordiais para se tornar hoje esta enorme rede de comunicação.

Reding (2005), citado por Serra (2014) aduz que a internet apresentou um desenvolvimento espetacular após demonstrar que era possível interligar todo conteúdo cultural e espalhar as informações em nível mundial.

Sabe-se que a expansão que a internet trouxe para as pessoas foi incrível e realmente ter um meio de comunicação global através do mesmo endereço de acesso é algo fantástico. Graças a esta tecnologia, foi possível levar a informação e cultura de lugares para lugares, e fazer com que mais quantidade de pessoas ficassem bem informadas de assuntos que sem a internet haveria uma dificuldade muito grande de adquirir tais informações, às vezes a dificuldade encontrada é o idioma, mas praticamente sempre existirá conteúdo para atender a busca de informação solicitada pelas pessoas e material em quase todos os casos não faltará, lembrando que a cada momento que se passa, cada vez mais informações são inseridas na internet.

### 2.7.1 Benefícios da internet para o conhecimento

Pinho (2002) acrescenta que a Internet passou a ser de uso comercial em nosso país em 1995, mais precisamente em maio, passando a não ser somente de acesso acadêmico e sim de todos os públicos, isto aconteceu devido ao ingresso em massa de vários setores na rede.

Sabe-se que naquela época a internet já era um grande passo e para os que aprofundaram o saber nesta tecnologia, já estavam cientes do quanto ela seria de impacto na sociedade e não teve como ser diferente no Brasil ou em qualquer lugar do mundo, assim que sua expansão deixou de ser apenas em universidades e de instituições de ensinos e pesquisa, todas as áreas sejam elas comerciais, empresariais, industriais, governamentais e outras passaram a ter influência nesta rede de comunicação e disponibilizaram serviços e informações vinte e quatro horas por dia, sete dias por semana para todos que tinham acesso à rede.

Inovações tecnológicas causam grandes impactos na sociedade e fazem com que há mudanças em seu dia a dia. A Internet enquanto era emergente e de certa forma simples de ser utilizada pelas pessoas foi uma ferramenta diferenciada na comunicação entre as pessoas, seus diferenciais destacavam-nas dos demais tipos tradicionais de comunicação (PINHO, 2002).

Conclui-se que a sociedade foi expandindo seu interesse sobre esta tecnologia e foi comum que o comportamento das pessoas fossem sofrendo mudanças, para a maioria no trabalho, onde na época era mais oportuno de ter contato com a internet e também de conhecer, em seguida tornou-se necessário, hoje em dia é praticamente essencial para muitas pessoas, empresas e organizações como um todo, diferente das demais mídias de

comunicação como a TV<sup>3</sup>, rádio e jornais, a Internet tinha tudo isso e algo a mais para oferecer, mesmo as pessoas não a conhecendo o suficiente e sentindo a necessidade de aprender mais sobre ela, já existia um interesse despertado comparado com qualquer outro meio.

## 2.8 PHP

A linguagem PHP (*Personal Home Page*) apareceu como um sistema muito simples que apenas interpretava alguns comandos e interagia com alguns elementos das *homepages*, como contador de visitas por exemplo, e surgiu no outono de 1994 por Rasmus Lerdorf (MUTO, 2006).

Sabe-se, que a realidade é outra e a linguagem PHP não aparece nas páginas *web* como sinônimo de um sistema simples, mas sim como reconhecimento de qualidade, boas práticas, desempenho, profissionalismo, facilidade, usabilidade e diversas outras qualidades tanto para o usuário quanto para o programador.

Para o avanço do PHP, ele passou a ter uma equipe organizada para desenvolvimento de seu interpretador, para onde este foi a versão 3 do PHP, para isso ele deixou de ser um projeto de Rasmus e teve seu interpretador reescrito por Zeev Suraski e Andi Gutmans (MUTO, 2006).

E foi assim que, a “pequena equipe” elaborou o grande sucesso, a versão 3 do PHP foi o deslanche para grandes avanços do desenvolvimento voltado para *web*, com melhor organização dedicada ao projeto, passou a ser cada vez mais conhecido, testado, utilizado e na maioria das vezes aprovado. A linguagem PHP é uma das mais usadas no mundo e a mais utilizada em *backends*<sup>4</sup> de *sites*.

O grande avanço do PHP fez com que ele seja reconhecido como uma das mais poderosas ferramentas para a programação *web* sendo capaz de produzir mecanismos para as páginas da internet em menor tempo e com bastante eficiência (BRUNO et al., 2010).

Entende-se o sucesso que o PHP tornou-se, muitas pessoas passaram a olhar esta linguagem de formas diferentes, com o resultado de alguns trabalhos feitos em PHP, as pessoas foram tornando-se mais confiantes e hoje em dia os mecanismos são tantos que até

<sup>3</sup> Abreviação de Televisão.

<sup>4</sup> É popularmente conhecido como “a parte de trás” do que é visível em uma página *web*, é a parte lógica e intangível responsável pelo funcionamento daquilo que se vê.

mesmo pessoas sem conhecimento da linguagem são capazes de criar páginas da internet com o auxílio de alguns *sites* que elaboram pré-configurações, *layouts*, componentes e outros recursos, todos utilizando o PHP.

A linguagem PHP além de seu reconhecimento em potencialidade também é uma das que possui aprendizado extremamente fácil para aqueles que já tiveram contato em linguagens como C, Java e Perl, pois são linguagens com características semelhantes (BRUNO et al., 2010).

Desta forma, ficou relativamente fácil um programador de outra área despertar ainda mais sua curiosidade em aprender PHP, utilizar e investir em componentes para a *web*, a semelhança de algumas propriedades entre as linguagens despertaram de certa forma o interesse das pessoas do mundo da tecnologia, em especial, os programadores ou os já familiares de outras linguagens de programação.

## 2.9 CSS

CSS ou *Cascading Style Sheets* é responsável por dar estilo à página *web* assim como também formatar seu conteúdo e melhorar o visual da página, textos, posicionamento de conteúdos e imagens. É a folha de estilo que manipula boa parte do *design* (QUIERELLI, 2012).

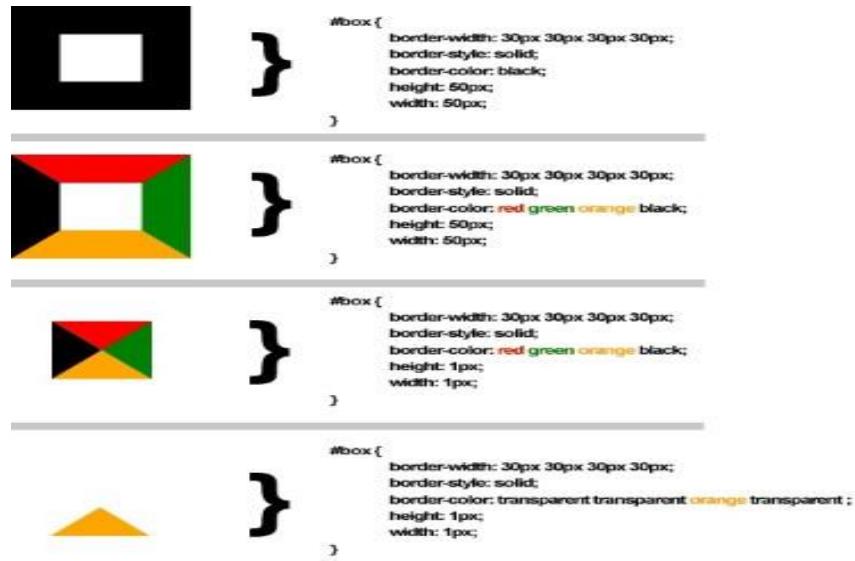
Compreende-se que o CSS por se tratar de parte visual ao usuário, mais bem trabalhado no *frontend*<sup>5</sup>, são boas vindas do seu *site* ou página, com a utilização dele pode-se obter uma conquista maior do usuário, porque a parte visual pode proporcionar maior harmonia dos componentes, interatividade, melhor usabilidade e organização, resumidamente, é tudo que o usuário quer.

De acordo com Figura 2, é possível observar como o CSS pode se destacar visualmente na manipulação de alguns elementos, onde é possível aplicar vários estilos na página, tornando-a visualmente mais atrativa e até mesmo organizada.

---

<sup>5</sup> É “a parte da frente” de uma página *web*, é o lado onde o usuário vê.

**FIGURA 2** - Exemplo de imagens criadas somente com CSS



Fonte: Site <http://johnylab.net/?id=106>, acessado em 24/05/2016 às 14:03.

Quierelli (2012) esclarece que além de formatar o conteúdo da página, o CSS também é aplicado para obter maior flexibilidade de *layout*, permitindo o desenvolvedor ser mais criativo nos elementos que a compõe como os *containers* e tabelas.

Certamente, sabe-se que na parte visual, criatividade é tudo e concorda com Quierelli, o CSS proporciona com que o programador ofereça ao usuário final, um conteúdo diferenciado, que pode ser utilizado efeitos e técnicas específicas no *layout*, formas personalizadas de componentes, dentre milhares de outras opções visualmente agradáveis.

## 2.10 BANCO DE DADOS

Setzer (2005) explica que após definir um dado e caracterizar a informação, não se pode guardá-la no computador, pois este é incapaz de interpretá-las e depende de uma pessoa para isso, por isso se diz banco de dados e não banco de informações. Desta forma, o computador é caracterizado como equipamento de processar dados, dos quais são incompreensíveis em algumas formas.

Sabe-se que, um conjunto de dados forma uma informação comprehensível e que a mesma não pode ser associada diretamente a um computador, por exemplo, quando o conceito de banco de dados entra em ação, imagine uma tabela com dados de um usuário como: nome,

idade e estado civil. Estes podem ser os dados armazenados e então a informação pode ser numerosa, como identificar se o usuário é adulto ou criança, interpretar se ele é casado ou não, dentre outras informações que com a inserção de dados elas podem ser geradas.

### 2.10.1 Sistema gerenciador de banco de dados (SGBD)

Tanto Muto (2006) como Lobo (2008) apontam que o banco de dados MySQL é uma proposta forte para aplicações de pequeno a médio porte sendo um banco com suporte a funções SQL<sup>6</sup> através do ANSI SQL, entre suas vantagens pode-se destacar a rapidez e flexibilidade que oferece tornando-se conhecido no mundo todo, além de ser um SGBD compatível em diversas plataformas, conquistou uma vasta comunidade de usuários.

Compreende-se que o MySQL tem um público alvo, este público caracterizado de pequeno a médio porte com certeza será muito bem atendido, considerado um dos mais utilizados, principalmente em aplicações *web*, ele atende o que promete e possui uma comunidade bastante forte que auxilia em dúvidas ou desafios que possam ser encontrados com seu uso.

## 2.11 JAVASCRIPT

Também presente em uma página HTML, o *JavaScript* é um linguagem de utilização em harmonia com outra linguagem, geralmente o próprio HTML ou da aplicação, escrita na camada de comportamento é bastante utilizada para interagir os componentes ou elementos da página, como as imagens, caixas e formulários que pode ser designadas funções para alguma finalidade específica e até mesmo animações que manipulem a interface (REMOALDO, 2008).

Compreende-se que o *JavaScript* é, desta forma, presente no *backend* da página *web*, que depende da estrutura e se pode observar o *JavaScript* presente na página ao visualizar os componentes da mesma através da inspeção de elementos do navegador, alguns são presentes na mesma página, outros possuem *link* de importação, outros são presentes em repositórios e

---

<sup>6</sup> “Structure Query Language”, em português, Linguagem de Consulta Estruturada, é a linguagem padrão que comunica com os mais conhecidos bancos de dados de modelo relacional.

às vezes é possível obter acesso a eles do lado cliente, alguns são mais bem incorporados e não há tanta facilidade para encontrá-los.

A camada de comportamento como o nome já diz, está relacionada ao comportamento de elementos da página, para manipular e ter controle sobre eles, utiliza-se o *JavaScript* e as ações que ele permite realizar interagem com as ações do usuário e quando aplicado juntamente a outras técnicas como o *CSS*, pode-se obter resultados impressionantes.

## 2.12 FRAMEWORKS

Em seus conceitos, Horstmann (2007) apresenta que *framework* pode ser definido como a disposição de classes de funcionalidades que em conjunto constrói mecanismos de soluções para problemas específicos, geralmente os *frameworks* utilizam padrões múltiplos.

Sendo assim, sabe-se que as classes que consistem o *framework* são pré-definidas, por exemplo de calendários, que já possuem sua estrutura basicamente estabelecida mas nem sempre as classes do *framework* sozinhas atendem as necessidades do programador e aí que surgem o que eles chamam de subclasses.

Ainda nos estudos de Horstmann (2007), é função do *framework* identificar o momento em que determinada função deve ser usada, assim que for chamada. O programador já tem o papel de alimentar os métodos responsáveis pelas funcionalidades do projeto.

Entende-se que existe uma harmonia entre o *framework* e demais códigos construídos pelo próprio programador, na maioria dos casos utiliza-se classes de *framework* e subclasses propriamente escritas para atender as necessidades de cada aplicação, um conjunto de funcionalidades é alimentado pelos métodos do *framework* e também pelo desenvolvedor.

Na Figura 3, pode-se observar exemplos de como os botões ficam com a utilização do *framework bootstrap*, são modelos já prontos para uso, com características como cores e tamanhos variados e sua aplicação pode tornar a página mais atrativa.

**FIGURA 3** - Botões modelados pelo framework *bootstrap*

| Button  | Large Button | Small Button | Disabled Button | Button with Icon | Split Button |
|---------|--------------|--------------|-----------------|------------------|--------------|
| Default | Default      | Default      | Default         | Heart Default    | Default ▾    |
| Primary | Primary      | Primary      | Primary         | Heart Primary    | Primary ▾    |
| Info    | Info         | Info         | Info            | Heart Info       | Info ▾       |
| Success | Success      | Success      | Success         | Heart Success    | Success ▾    |
| Warning | Warning      | Warning      | Warning         | Heart Warning    | Warning ▾    |
| Danger  | Danger       | Danger       | Danger          | Heart Danger     | Danger ▾     |

Fonte: Site <http://pikock.github.io/bootstrap-magic/app/#!/editor>, acessado em 24/05/2016 às 14:16.

## 2.13 NAVEGADORES

Nos apontamentos de Péricas (2012), a origem do navegador gráfico foi, em torno de 1993, criado por Marc Andreessen, na Universidade de Illinois, teve o nome de *Mosaic* e sua evolução foi o *Netscape Navigator*. O Internet Explorer (IE<sup>7</sup>) da Microsoft apareceu dois anos depois.

Conclui-se que a interatividade veio à tona a partir deste momento, o conteúdo gráfico para a época era algo deslumbrante, uma imagem e um botão nem se falava, total avanço e realmente foi, o *Mosaic* permitiu que isto fosse possível e também foi a base para que outros navegadores surgissem através da disponibilidade do código fonte do *Mosaic*. O Mozilla *Browser* surgiu logo em seguida alguns anos depois do Internet Explorer e era composto por uma organização de comunidade própria que também é responsável pela sua evolução para Mozilla Firefox, que é conhecido até os dias de hoje. Outros navegadores também foram surgindo neste meio tempo, como o Safari para a Apple.

De acordo com o entendimento de HTML e páginas web, o navegador que é responsável por interpretá-las realiza esta tarefa através de duas formas, são elas o *client side*, ou seja, o lado cliente, no caso o usuário, mais propriamente dito, o navegador e o *server side*,

---

<sup>7</sup> Internet Explorer – Navegador lançado pela Microsoft em 1998 com base no código fonte do Netscape Navigator

que é o lado do servidor, ou seja, quem oferece os serviços através de protocolos interpretados pelo navegador, basicamente esta é a forma com que ele se comunica para repassar a informação.

## 2.14 TESTE DE SOFTWARE

Martins (2007) identifica que todo *software* deve ter sua fase de teste, pois é ai que se encontram erros em algum lugar do projeto ou de sua construção. Teste de *software* é realizado a partir de uma estratégia, existindo o caminho a ser feito e seu roteiro planejado em cada projeto.

Conclui-se que, no início um *software* basicamente vai ter falhas, se fosse possível evitar que isso acontecesse com total eficácia, não seria necessário absolutamente todo *software* possuir uma etapa de testes e elas existem justamente para reduzir a quantidade e o impacto de falhas para o cliente consumidor, além de ser uma forma de demonstrar e garantir a qualidade do *software*, fazendo com que seja um diferencial no momento de divulgação, apresentação e *marketing* para as pessoas sentirem-se confiantes do produto que estão adquirindo, e fizerem com que seja uma opção de escolha mais forte comparado a outro *software* caso não tenha realizado o teste de *software*.

Para ter um planejamento de teste de *software* adequadamente, é preciso reunir dados de métricas no momento do teste e utilizar também da confiabilidade de *software*, que possui modelos existentes para conseguir estabelecer diretrizes, e saber onde deve parar o teste (MARTINS, 2007).

Entende-se que durante todo o esquema de passo a passo que forma o planejamento do teste de *software*, existem outros processos que trazem dados importantes para a pessoa responsável pela realização dos testes, estes também são meios de avaliar o comportamento do *software* e até mesmo do sistema e levar em conta que a confiabilidade de *software* é direcionada pelos resultados que se aplicam.

### 2.14.1 Teste de verificação e validação

Martins (2007) reconhece que existem níveis de composição do teste de verificação, que dão início pelo teste de unidade, responsável por cada unidade presente no *software*

baseado em sua composição, em seguida é realiza-se o teste de integração, focado em validar a arquitetura do *software* na sequência como os componentes vão sendo integrados no sistema.

Conclui-se que o teste de verificação é geralmente onde se inicia a fase de testes, caracterizado pelo teste de unidade, processo que representa a composição do *software*, ou seja, as classes e também os componentes são também caracterizados pelo teste de integração que é voltado para a parte de arquitetura onde se aplicam os componentes que irão ser integrados no sistema.

Após finalizado o teste de verificação, entra em fase o teste de validação, que analisa se o *software* está de acordo com os requisitos que foram especificados. É aqui que se garante que o *software* atende aos requisitos essenciais do projeto (MARTINS, 2007).

Entende-se que o teste de validação é responsável pela análise do *software* depois dele já se encontrar de certa forma montado pois ele vem após toda a fase de teste de verificação, sendo assim, ele checa e valida se o *software* está de acordo com os requisitos funcionais e se tudo está nos conformes com o especificado, se o *software* mantém o comportamento esperado e o desempenho necessário, estes são alguns dos pontos essenciais de um *software* que são validados, caso algum item não atenda o que foi definido como essencial, é preciso revisar o projeto pois algo saiu errado.

Martins (2007) conceitua que o teste de sistema é uma ferramenta importante onde o *software* e também as composições do sistema como banco de dados, *hardware* e até mesmo pessoas por exemplo, passam por testes em conjunto com intuito de analisar se trabalham como o esperado.

Compreende-se que o teste de sistema é considerado uma ferramenta importante juntamente ao teste de validação, ambos adotam basicamente os mesmos objetivos de testes e é executado com base no usuário, no aspecto de ponto de vista dele, utilizando-se de todas as funcionalidades que o sistema oferece para tentar detectar erros, falhas ou inconformidades das quais não se esperava durante a fase de projeto. Neste processo do teste já existe certa interatividade com os conteúdos e componentes do *software*, muitas vezes o ambiente é muito similar ao ambiente do resultado final do projeto, pois este já possui os requisitos funcionais e devem estar prontos para uso e testes.

## 2.14.2 Garantia da qualidade

Presente em fase de execução, a garantia da qualidade é o foco de qualidade do projeto e para isso existe todo um gerenciamento da qualidade na qual é encontrado onde se segue a orientação do planejado e garante que o projeto atenderá aos padrões de qualidade relevantes (MARTINS, 2007).

Entende-se que o gerenciamento da qualidade é encontrado e só pode ser realizado dentro de algum objetivo proposto, geralmente isto é o planejamento do projeto e é com base nele que o processo da qualidade irá exercer suas atividades de controle da qualidade, checando se tudo está respeitando o que foi definido, para isso uma das formas de garantir que tudo está correndo conforme planejado é realizando auditorias e não basta apenas seguir o plano, é preciso que ele também esteja correto para não correr o risco das implementações saírem de forma inesperada. O processo de garantir, ou seja, a garantia só é obtida quando a parte de controle da qualidade está rigorosamente de acordo com o resultado de testes. Este controle também é utilizado para tomar algumas medidas, caso o projeto tenha bastante erros ou falhas, se forem identificados problemas demasiados de uma certa parte do projeto, esta poderá ser revisada junto à equipe responsável com a finalidade de compreender o que está saindo fora dos padrões ou o que falta para se adequar aos padrões e boas práticas que são consequentemente resultados de boa qualidade.

O papel da garantia da qualidade de *software* vai muito além de apenas testes e verificações, o grupo da qualidade estabelece procedimentos que se aplicam como padrões no processo de *software*, do qual deve estar de acordo, também há monitoramento como forma de garantir que os padrões serão seguidos, resumidamente, o papel de garantir a qualidade vai desde o *software* e suas etapas a serem seguidas até o produto final com qualidade (SCHACH, 2010).

Conclui-se que, garantir a qualidade não é algo tão simples para quem pensa que basta fazer seu papel bem feito, para se ter qualidade garantida não basta apenas um ou uma parte do projeto sair com tudo em ordem, é necessário que todo o processo de *software* seja de certa forma monitorado para que possa ter e levar a prática da garantia de qualidade ao produto, no caso, ao *software*.

Testes planejados são fundamentais para comprovar que o produto apresentou os resultados conforme o que foi definido ou não e em todas as etapas está presente o controle da

qualidade que irá atuar tanto nos resultados com sucesso quanto nos resultados com falha, ou seja, é um processo incorporado ao *software* desde seu início até sua reta final e durante todo este período existe uma série de responsabilidades que juntas formam o que hoje se chama de engenharia de *software*.

## 3 MATERIAIS E MÉTODOS

### 3.1 LEVANTAMENTO DE DADOS

Foi realizado um levantamento de dados com uma amostra de alunos dentre todos os períodos totalizando aproximadamente 45,2% de um total de 135 alunos matriculados, de acordo com a coordenação do curso de Engenharia de *Software* da Universidade de Rio Verde presente no Anexo C.

#### 3.1.1 Gráfico das amostras

Segue abaixo a Figura 4 relacionada à primeira questão do levantamento de dados que está presente no anexo b.

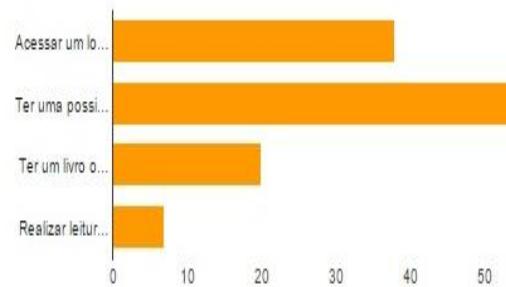
**FIGURA 4** - Primeira questão do levantamento de dados



Segue abaixo a Figura 5 relacionada à segunda questão do levantamento de dados que está presente no anexo b.

**FIGURA 5** - Segunda questão do levantamento de dados

**2. Para você, o que é mais interessante na hora das pesquisas da faculdade?**



Acessar um local com conteúdos disciplinares, não importando onde você esteja. 38 62.3%

Ter uma possibilidade de busca avançada de conteúdo no material disponível em um portal a qualquer hora. 54 88.5%

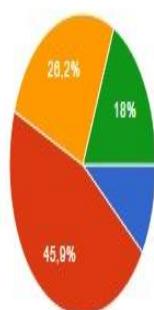
Ter um livro ou qualquer outro material físico em mãos para ler. 20 32.8%

Realizar leituras e pesquisas em livros ou outro material físico até encontrar o que precisa. 7 11.5%

Segue abaixo a Figura 6 relacionada à terceira questão do levantamento de dados que está presente no anexo b.

**FIGURA 6** - Terceira questão do levantamento de dados

**3. Você está satisfeito com o conteúdo DISCIPLINAR ACADÊMICO que encontra na internet?  
É completo e de fontes confiáveis?**



Sim, sempre encontro conteúdos que fazem total sentido e de fontes renomadas. 6 9.8%

Sim, me atende bem, mas não tenho total certeza se está correto. 28 45.9%

Sim, até o momento sempre me atendeu, mas não faço a menor idéia de onde são as fontes e se está totalmente completo. 16 26.2%

Não, ultimamente ando me deparando com muita propaganda, títulos falsos ou coisas sem sentido. 11 18%

Segue abaixo a Figura 7 relacionada à quarta questão do levantamento de dados que está presente no anexo b.

**FIGURA 7** - Quarta questão do levantamento de dados



### 3.2 Diagramas

Para cada parte de sistema, é adequado o desenho de diagramas e suas finalidades, porém alguns deles serão encontrados em qualquer parte de projeto, são eles os diagramas de classe, de caso de uso e também de interação (LIMA, 2014).

Conclui-se que os diagramas são de certa forma variantes para quem os aplica, é preciso entender bem da estrutura do projeto para não ser prejudicado na etapa de planejar os diagramas, e certificar que não ficará nenhum faltando ou também que não ficará redundante demais as informações presentes em vários diagramas sem necessidade. Sabendo que todo projeto irá possuir pelo menos o diagrama de classe, diagrama de caso de uso e diagrama de interação ou seja, diagrama de sequência.

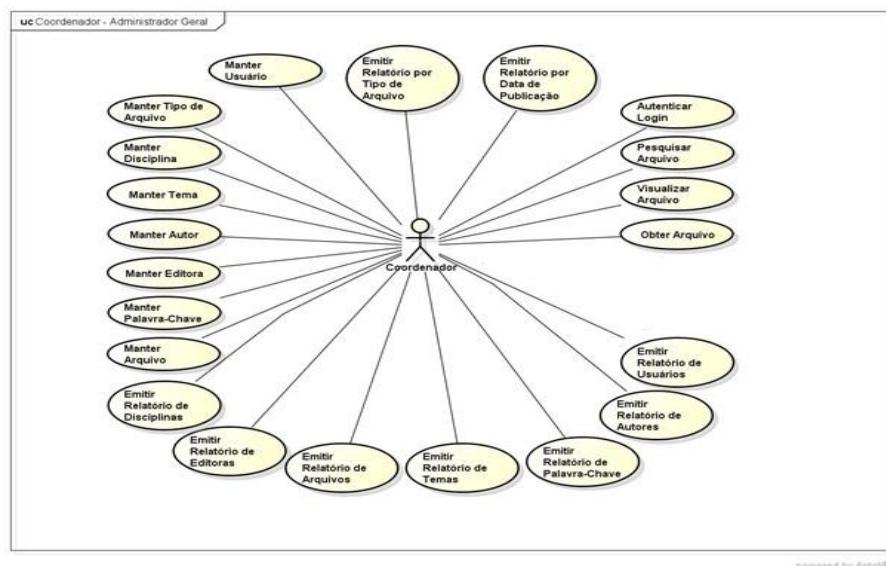
#### 3.2.1 Diagrama de caso de uso

Um diagrama de caso de uso é composto por eclipses que formam um conjunto dos casos de uso do sistema e são ligadas até um ator através de linhas, que formam assim a associação entre estes objetos, esta associação é a representação da interação dos atores com determinado caso de uso do sistema. Os relacionamentos entre estes objetos também podem ser de extensão, que é quando um caso de uso pode estar sujeito a instanciar em outro, que vai

depender do comportamento definido entre eles e também existe o relacionamento de inclusão, que indica que a instância de um caso de uso também haverá este mesmo comportamento presente em outro caso (LIMA, 2014).

Compreende-se que o diagrama de caso de uso mantém o relacionamento de determinado ator com determinadas ações do sistema e também é forma de representação de quais atores, ou seja, quais usuários poderão acessar quais serão os recursos que o sistema oferece, através deste diagrama simples que apresenta com clareza quem são os usuários e o que estes usuários irão ter de acesso no sistema (Figura 8).

**FIGURA 8** - Diagrama de Caso de Uso que contempla todos os requisitos funcionais do sistema



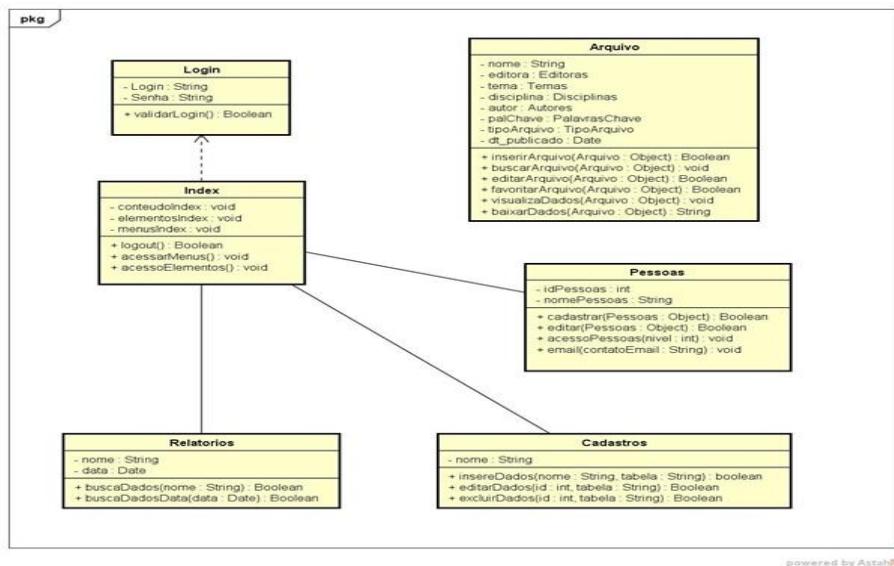
### 3.2.2 Diagrama de classe

Sampaio (2007) ensina que o diagrama de classe possui uma forma de visão sobre um modelo de objetos, esta visão é do tipo estática e a principal função é descrever as classes do sistema e seus relacionamentos entre si. Na representação de classe para este diagrama, existem três itens que são eles o nome da classe, logo abaixo são os atributos e abaixo novamente são os métodos.

Entende-se que o diagrama de classe está voltado na composição das classes do projeto, o que forma o objeto, dividido entre tudo o que é método e tudo o que é atributo, simples assim, também é possível representar através do diagrama de classes todas as

associações entre as classes, tudo de maneira estática. Segue abaixo na Figura 9 um exemplo deste diagrama.

**FIGURA 9** - Diagrama de Classe do Projeto

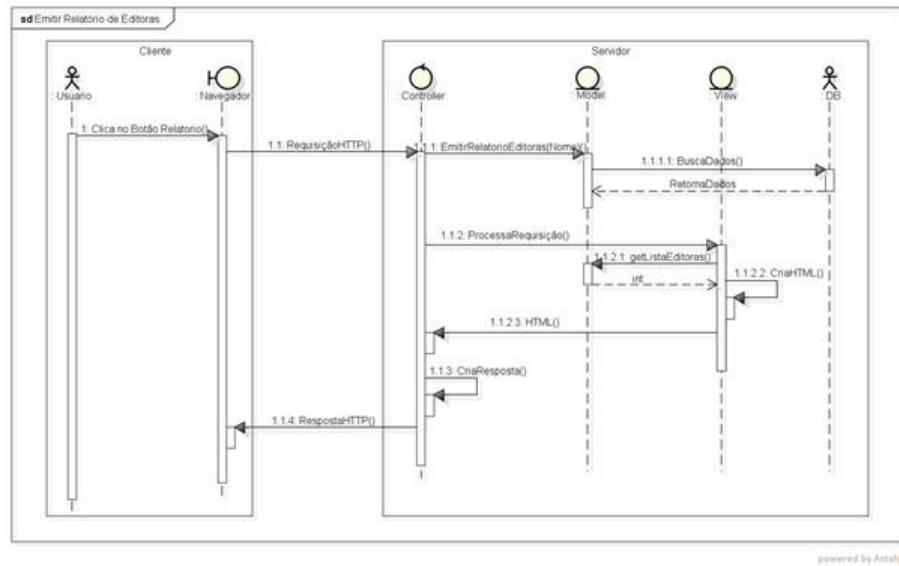


### 3.2.3 Diagrama de interação (ou de sequência)

Lima (2014) apresenta que o diagrama de interação é responsável por mostrar como percorre os eventos do projeto separados por cada caso de uso, dentre esse processo, é possível notar os passos de início de ação do ator, os objetos ao qual estão envolvidos na ação, as informações que são trocados entre eles e as etapas em que tudo acontece, ou seja, a ordem dos eventos, resumidamente, é o diagrama que apresenta como deve ser implementado as ações do sistema.

Sabe-se, então, que o diagrama de interação, também conhecido como diagrama de sequência, apresenta a parte em que envolve os fluxos entre o usuário e a ação que o *software* disponibiliza para ser tomada, neste processo, é possível observar através do diagrama por quais etapas todo o processo passa, e facilitar assim o entendimento das ações e principalmente como elas se relacionam e se comunicam, por quais camadas estão passando e desta forma subentende-se que tenha maior organização e controle de fluxo das operações e dos eventos que o *software* realiza, segue abaixo na Figura 10 um modelo de sua estrutura para sistemas *web* com o lado cliente e o lado servidor.

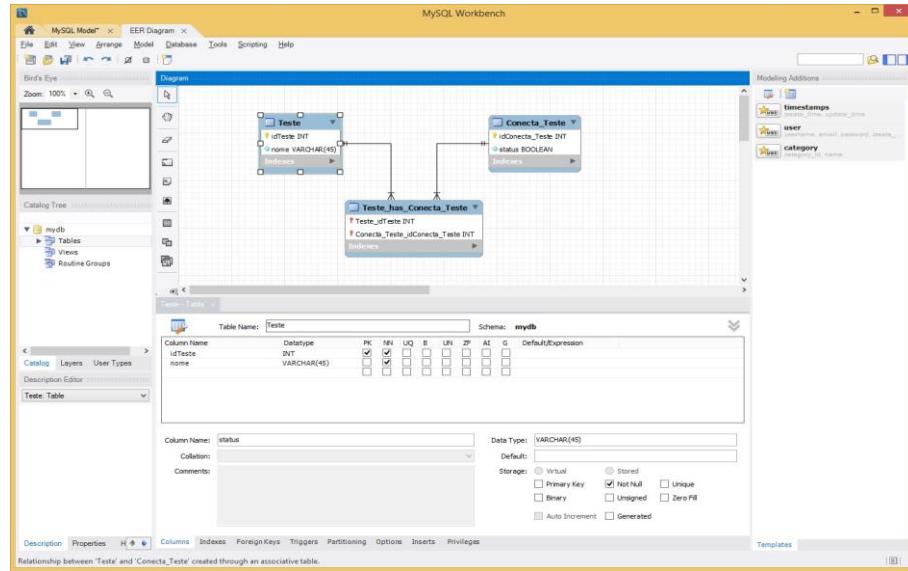
**FIGURA 10** - Diagrama de Sequência para emissão de relatório de editoras



### 3.3 WORKBENCH

*Software* livre já utilizado no processo de elaboração da modelagem de entidades e relacionamentos do banco de dados do projeto, sabe-se que ela proporciona uma praticidade incrível e possui um arsenal de opções que são completamente aliados ao desenvolvedor, possui interatividade de clicar e arrastar objetos da paleta para realizar a modelagem de banco de dados e seus relacionamentos dentre os componentes disponíveis como as tabelas por exemplo, ferramenta bastante completa que possui recursos para projetos simples como também para projetos mais avançados, e permite inclusão de plug-ins para realizar algumas tarefas mais específicas. Segue abaixo, na Figura 11, uma tela do *software* com algumas opções de interação.

**FIGURA 11 - Workbench**, ferramenta utilizada na criação da modelagem do banco de dados



### 3.3.1 Dicionário de dados

Processo já elaborado no atual projeto, pode-se dizer que o dicionário de dados é a representação dos elementos de dados de um banco em forma de tabela, é nada mais nada menos é uma base de dados que serve para consulta de informações de alguns detalhes da estrutura do banco, ou seja, informações das entidades nele presente, alguns campos, elementos em si que compõem o banco, variam também de acordo com o nível de detalhamento do dicionário de dados. Geralmente utilizado para que toda a equipe, tanto os que estão diretamente ligados ao acesso no banco, quanto os que não o possuem, saibam quais são as informações principais de todos os objetos que compõem determinado banco. Não há um padrão específico, é normal encontrar variados tipos de dicionário de dados. Segue na Tabela 1, um exemplo que foi elaborado para este projeto.

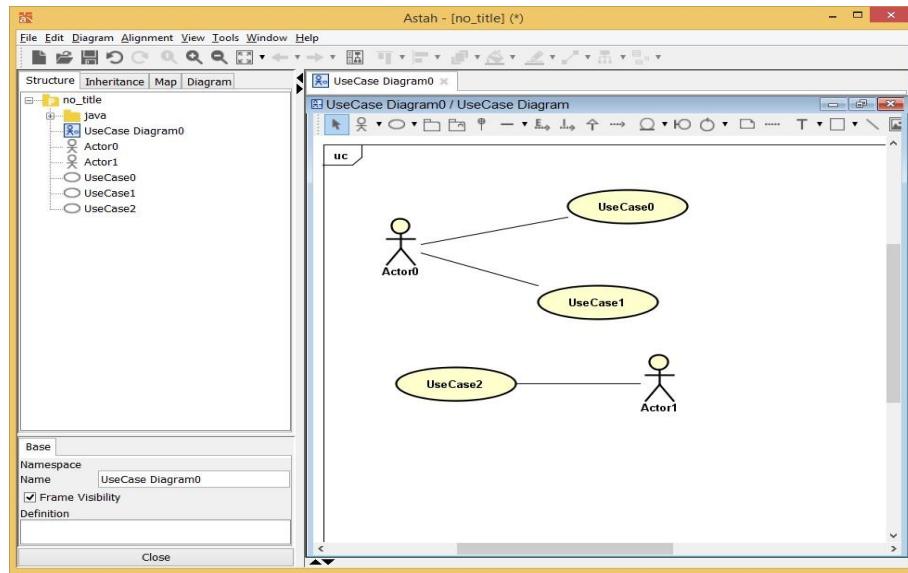
**TABELA 1** - Dicionário de dados da entidade denominada arquivos

| Atributo                  | Classe       | Domínio  | Tamanho | Descrição          |
|---------------------------|--------------|----------|---------|--------------------|
| ARQ_ID                    | Determinante | Numérico |         | Chave da Entidade  |
| ARQ_NOME                  | Simples      | Texto    | 45      | Nome do Item       |
| ARQ_DT_PUB                | Simples      | Data     |         | Formato dd/mm/aaaa |
| ARQ_CAM_ARQ               | Simples      | Texto    | 255     | Caminho do Arquivo |
| Editoras_idEditoras       | Determinante | Numérico |         | FK                 |
| TipoArquivo_idTipoArquivo | Determinante | Numérico |         | FK                 |

### 3.4 ASTAH COMMUNITY

Por ser uma ferramenta de código livre e já utilizada na etapa de modelagem UML do projeto, através das representações em forma de diagramas, pode-se dizer sobre ela que é de uso bastante simples no sentido de aprendizado e manipulação de seus elementos para composição dos diagramas, a mesma possui uma interface bastante amigável para o usuário e não exige dele muito conhecimento na ferramenta, sendo desejável maior conhecimento no próprio trabalho a ser desenvolvido, no caso, na modelagem dos diagramas para determinado projeto. Abaixo na Figura 12, pode-se observar uma representação do *layout* do sistema.

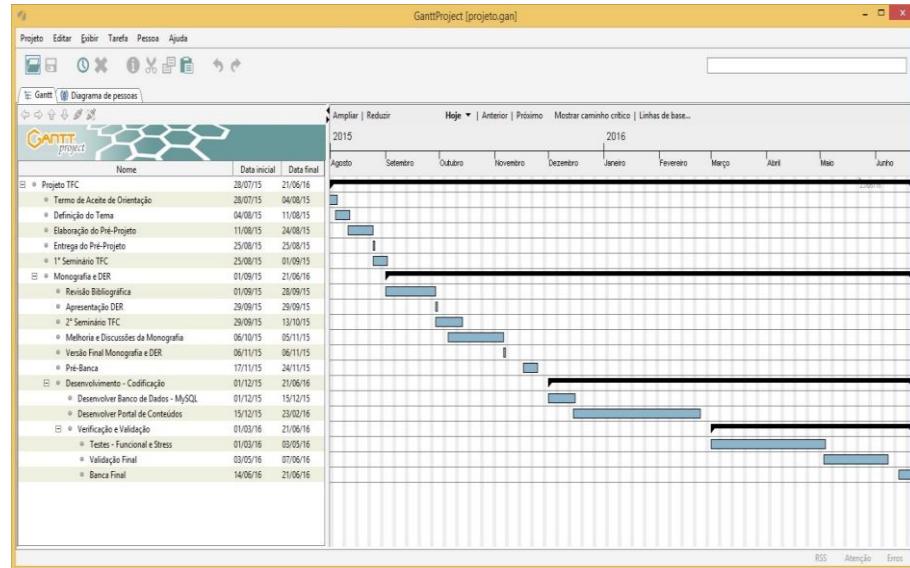
**FIGURA 12 - Astah Community, ferramenta utilizada para elaboração dos diagramas no projeto**



### 3.5 GANTTPROJECT

Também se trata de um *software* livre, isto é, de uso aberto ao público geral, o *GanttProject* já foi utilizado no projeto e ocupa o espaço da primeira ferramenta a ser utilizada no processo de elaboração do *software*, ela é responsável por auxiliar na elaboração do calendário de etapas e tarefas do projeto, estipula quando será dado início, no que será após dado o início, quanto tempo irá levar e quando se pretende finalizar as tarefas a serem realizadas no decorrer do processo de elaboração do *software*, e formar de certa forma uma linha do tempo bastante intuitiva, é de simples utilização não exige experiência do usuário com a ferramenta. Observa-se a Figura 13, logo abaixo, com a tela do *software* e um modelo de cronograma sendo realizado nela.

**FIGURA 13** - GanttProject, ferramenta utilizada na elaboração do cronograma do projeto



### 3.6 REQUISITOS FUNCIONAIS

Ao todo foram definidos vinte e três requisitos funcionais que se pretende conter no *software* assim que estiver em fase final de desenvolvimento, e lembra que todos eles são essenciais na etapa de conclusão do sistema. Segue abaixo uma lista que compõe todos eles:

- 1 – Autenticar Login
- 2 – Manter Usuário
- 3 – Manter Tema
- 4 – Manter Autor
- 5 – Manter Editora
- 6 – Manter Disciplina
- 7 – Manter Palavra-Chave
- 8 – Manter Tipo de Arquivo
- 9 – Manter Arquivo
- 10 – Pesquisar Arquivo
- 11 – Visualizar Arquivo

- 12 – Obter Arquivo
- 13 – Emitir Relatório de Usuários
- 14 – Emitir Relatório de Disciplinas
- 15 – Emitir Relatório de Autores
- 16 – Emitir Relatório de Temas
- 17 – Emitir Relatório de Editoras
- 18 – Emitir Relatório de Palavra-Chave
- 19 – Emitir Relatório de Arquivos
- 20 – Emitir Relatório por Tipo de Arquivo
- 21 – Emitir Relatório por Data de Publicação

### 3.7 AMBIENTE DE DESENVOLVIMENTO

Posteriormente, foi dado início ao desenvolvimento da aplicação *web* baseada nas instruções da documentação do Anteprojeto e DERS, tendo estas orientações planejadas e descritas foi possível ter um embasamento de como deveria ser o *site*, então ofereceram dados para os primeiros passos que foi instalar e configurar o ambiente de desenvolvimento e demais ferramentas necessárias para dar início ao projeto. Dentro deste ambiente de desenvolvimento conhecido como XAMPP, foi configurado um repositório denominado portal\_academico, onde é o local para acomodar as páginas *web* e tecnologias de auxílio no desenvolvimento, que são elas os *frameworks*, também foi configurado toda a parte de banco de dados, suas conexões e tratativas de falhas para identificar, prevenir e corrigir possíveis erros, e permitir assim que fosse possível realizar a criação do banco de dados de acordo com o MER(Modelo de Entidade Relacional) presente em anexo ao DERS. Neste mesmo ambiente de desenvolvimento também se encontra o servidor *web*, conhecido como Apache, ele é o responsável por processar as requisições HTTP (*Hyper Text Transfer Protocol*), o protocolo padrão da web.

Com a base de dados Acervo criada e suas conexões relacionadas ao XAMPP estabelecidas, o próximo passo foi elaborar um arquivo denominado conexao.php, que é o arquivo responsável por fazer a manipulação, transição ou troca de informações entre a

aplicação e o SGBD. Neste arquivo, encontra-se uma técnica chamada PDO(*PHP Data Objects*) que trabalha com certos procedimentos relacionados a banco de dados e seu ponto forte é a abstração do banco de dados. O exemplo de implementação do PDO para conexão ao banco segue abaixo:

```
<?php
define( 'MYSQL_HOST', 'localhost' );
define( 'MYSQL_USER', 'root' );
define( 'MYSQL_PASSWORD', "" );
define( 'MYSQL_DB_NAME', 'acervo' );

try {
    $conn = new PDO( 'mysql:host=' . MYSQL_HOST . ';dbname=' .
    MYSQL_DB_NAME, MYSQL_USER, MYSQL_PASSWORD );
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $conn->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
} catch ( PDOException $e ) {
    echo 'Erro ao conectar com o MySQL: ' . $e->getMessage();
}
?>
```

A forma de fazer uma solicitação do banco de dados com a utilização do PDO ficaria da seguinte forma, por exemplo:

```
<?php
$sql = $conn->query("select * from temas");
while ($linha = $sql->fetch(PDO::FETCH_ASSOC)) {
    echo $linha["TEM_NOME"];
}
?>
```

Dentre estes exemplos também existe a forma específica para inserir, atualizar e excluir dados do banco de dados, que são estas três formas semelhantes ao exemplo abaixo:

```
if ($conn) {

    $nova_disciplina = $_POST['txtnovadisciplina'];
    $id_disciplina = $_POST['dis-id'];
    $sql = "UPDATE disciplinas SET DIS_NOME='".$nova_disciplina."' where
DIS_ID='".$id_disciplina."'";
    $stmt = $conn->prepare($sql);
    $stmt->bindParam( ':nova_disciplina', $nova_disciplina );
    $stmt->bindParam( ':id_disciplina', $id_disciplina );

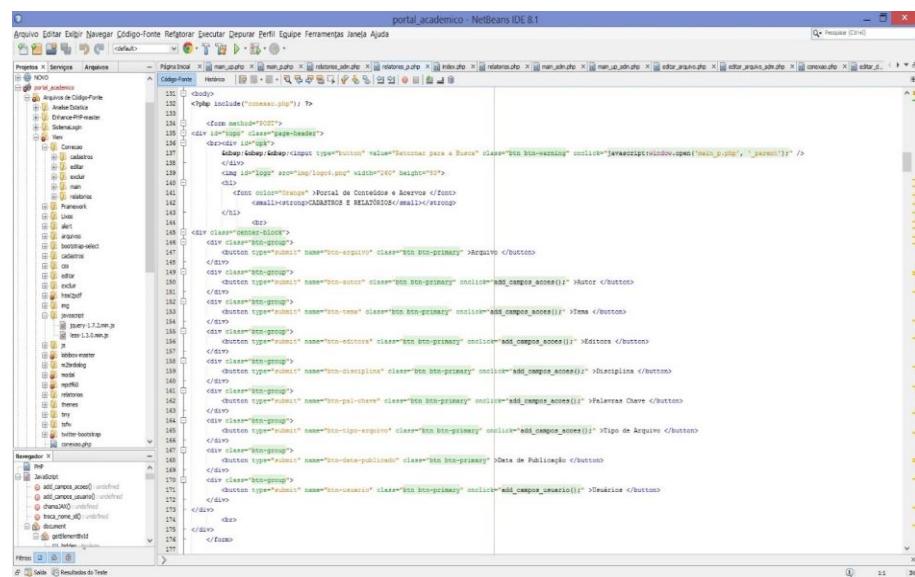
    $result = $stmt->execute();
} else {
    echo "<b>erro de conexão</b>";
}
```

Ciente de como realizar manipulações em informações dentro do banco de dados através da aplicação, pode-se partir do princípio de elaboração das funcionalidades que foram definidas e propostas, e conseguir assim dar início a interação da aplicação com a base de dados, mas antes disso foi preciso entender outros assuntos que estão interligados ao projeto, que é o conhecimento com a linguagem PHP para fazer a parte de codificação que é interpretada pelo lado servidor, ou seja, pelo lado onde se encontra a infraestrutura de implantação do projeto. O PHP tem seu papel básico de interpretar alguns comandos e fazer a parte de interação com demais elementos contidos na página *web*, e para que esta página fique satisfatória e rica em recursos, funcionalidades e com visual agradável é preciso ter uma harmonia pelo menos entre o PHP, HTML, CSS e JavaScript.

O princípio do desenvolvimento PHP foi realizado através da ferramenta NetBeans 8.1, uma IDE de desenvolvimento para várias linguagens, uma delas é o PHP. Nesta ferramenta também há possibilidades de desenvolvimento de arquivos CSS, HTML e JavaScript, onde é possível obter um auxílio tanto na organização dos arquivos como também no manuseio, interação e demais facilidades como autocompletamento de palavras reservadas

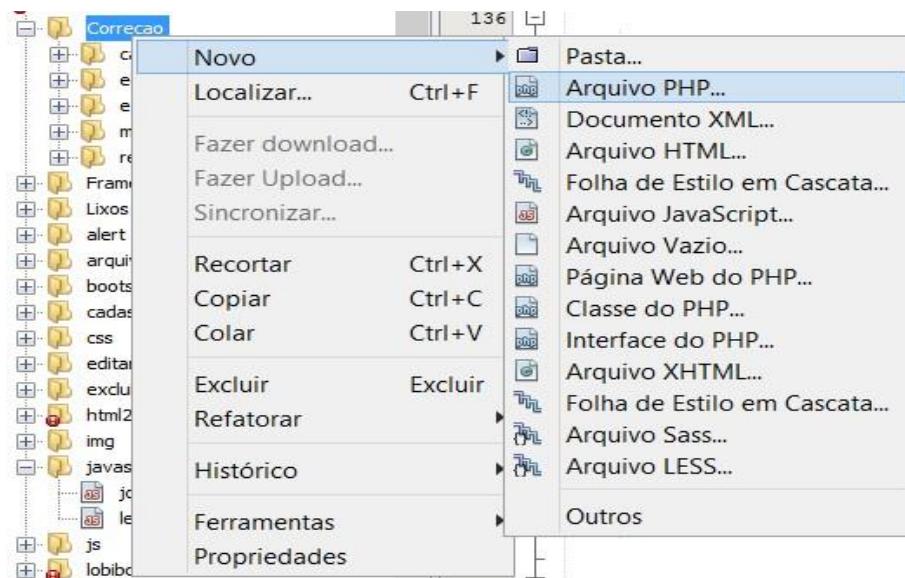
específicas para cada tipo de arquivo a ser desenvolvido, acompanhamento de erros e alertas, possíveis correções de falhas, inclusão e relacionamento entre arquivos dentro do mesmo diretório ou de diretórios diferentes com possibilidade de visualizar os nomes sem necessidade de entrar manualmente em cada diretório, esta é apenas algumas de outras diversas funcionalidades e apoio que a ferramenta proporciona ao desenvolvedor. Segue na Figura 14, o ambiente de desenvolvimento do *NetBeans* para PHP.

**FIGURA 14 - NetBeans 8.1, IDE de Desenvolvimento**



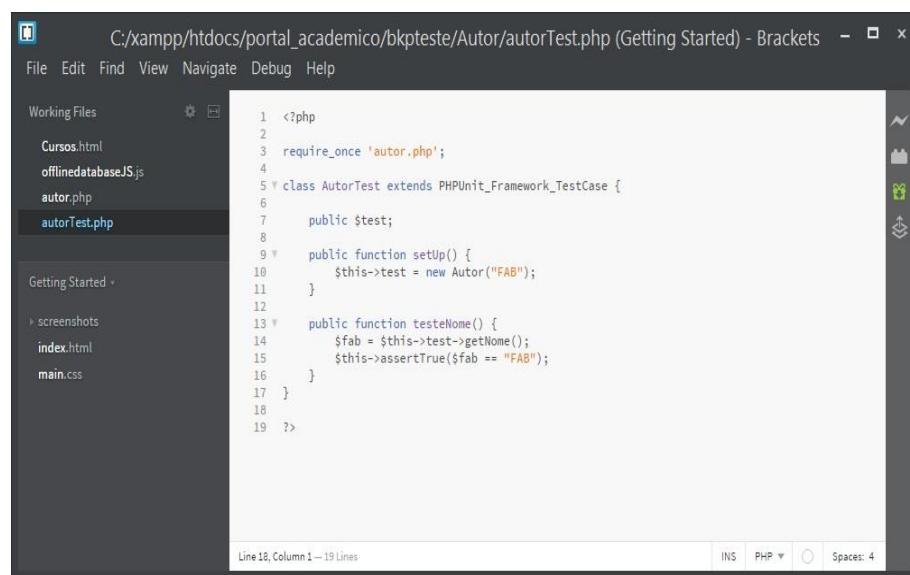
Dentro do próprio *NetBeans*, como dito anteriormente no parágrafo acima, há opções variadas de arquivos a serem desenvolvidos, como os vistos na Figura 15.

**FIGURA 15** - NetBeans 8.1, algumas opções de arquivos



Não se limitando em apenas uma ferramenta para codificação, o *brackets*, editor de código para desenvolvedores *web* é bastante interessante em algumas edições, a ferramenta proporciona diversos *plug-ins* que podem inclusive ser editados por qualquer um que os utilize, desta forma consegue-se adequar para a necessidade do desenvolvedor. Segue abaixo a Figura 16, a tela do editor.

**FIGURA 16** - Editor de Código *Brackets*



Conhecidas as ferramentas que foram utilizadas no projeto, obtém-se então o ambiente de desenvolvimento por completo e a partir delas pode-se prosseguir com o desenvolvimento do sistema.

No DERS, foram elaborados vinte e três requisitos funcionais, dos quais apenas vinte e um foram implantados, sendo removidos os requisitos de número 13 - Favoritar Arquivos e número 14 - Apresentar Favoritos. Esta remoção de dois requisitos funcionais foi necessária devido à remoção da necessidade de realizar *login* para o usuário comum discente, pensando em evitar a burocratização para acessar o sistema e visto uma interação direta do usuário final para a tela principal sem ter que passar por nenhum tipo de cadastro. Desta forma, foi necessária a remoção destes dois requisitos, pois com a ausência de um *login* de usuário para os discentes, não há como registrar o arquivo favorito para determinado indivíduo, pois não há como realizar a ligação dele com o arquivo sem um cadastro de usuário.

Sobre as funcionalidades do site, dos vinte e um requisitos, encontram-se oito relacionados a manter, que são as operações de cadastrar, editar e excluir ou inativar. As páginas PHP responsáveis para conduzir as funções de manter na aplicação estão divididas entre os seguintes diretórios: cadastros, editar, excluir, inativar e ativar. Em poder de decisão para exclusão do arquivo pelo administrador, é válido afirmar que o sistema trata das ligações daquele arquivo e as mesmas que estão vinculadas àquele único arquivo são exclusas. As informações pertinentes que possam estar ligadas a outros arquivos não são removidas do sistema, para isto, se for o caso, pode-se realizar a inativação de algum autor, editora, disciplina, tipo de arquivo, palavra-chave ou tema. No caso de usuários, existem os níveis de acesso docente e administrador, o discente acessa sem necessidade de usuário e caso algum docente precise de ter mais acesso ao sistema, no caso, torna-se um administrador, o mesmo pode ser alterado e assim terá acesso as páginas relacionadas a seu nível de acesso. Também é possível bloquear o usuário para que o mesmo não consiga acessar além do mesmo que um discente.

Os níveis de acesso do sistema foram elaborados através do uso de sessões em PHP, para identificar qual usuário está logado foi simples ao utilizar o código abaixo como exemplo:

```
$nivel_necessario = 2;

if (!isset($_SESSION['usuarioID']) OR ($_SESSION['usuarioNivel'] < $nivel_necessario))
{
    session_destroy();
    header("Location: index-login.php"); exit;
}
```

O código acima indica uma variável declarada como `$nivel_necessario`, e recebe valor inteiro 2, na condicional “`if`”, verifica-se não há a variável de sessão que identifica o usuário ou se o nível dele é inferior ao necessário para acessar a página. Esta tratativa indica que, não há usuário logado ou o usuário que efetuou *login* não possui permissões necessárias para acesso, nestes dois casos, a sessão é destruída por segurança através do comando `session_destroy()`, em seguida, o usuário é direcionado para a página de *login* do sistema, onde ele pode efetuar um novo *login* ou ir para a página principal sem necessidade de usuário. Caso o usuário tenha o nível de acesso necessário para visualização da página, a sessão dele não é destruída e ele segue o acesso normalmente.

Ao efetuar *login* no sistema, três arquivos são consultados no total, o primeiro é a página `index-login.php`, que é responsável por ser a parte visual para o usuário, onde o mesmo insere *login* e senha de acesso. Ao clicar no botão entrar, a página `index-login.php` passa os dados via método `$_POST` através de seu formulário para um arquivo chamado `login.php`, aqui é realizado a checagem se o usuário e senha estão corretos e se foram devidamente inseridos, para isso é realizado neste arquivo uma consulta ao banco de dados na tabela usuários, aproximadamente como pode ser visto no trecho de código abaixo:

```

$login = mysql_real_escape_string($_POST['login']);
$senha = mysql_real_escape_string($_POST['senha']); $entrar = $_POST['entrar'];

if (isset($entrar)) {
    $sql = "SELECT * FROM usuarios WHERE USU_LOGIN = :login AND
USU_SENHA = :senha;";
    $verifica = $conn->prepare($sql);
    $verifica->bindParam( ':login', $login );
    $verifica->bindParam( ':senha', $senha );
    $verifica->execute();
    $rows = $verifica->fetchColumn();
    if ($rows <= 0){
        echo "<script language='javascript' type='text/javascript'>alert('Login e/ou
senha incorretos');window.location.href='index-login.php';</script>";
    }else{
        $queryacesso = "SELECT USU_ID,USU_LOGIN,USU_ACESSO FROM
usuarios WHERE(USU_LOGIN = :login) AND (USU_SENHA = :senha) AND
(USU_ACESSO != 0) LIMIT 1";
        $sqlacesso = $conn->prepare($queryacesso);
        $sqlacesso->bindParam( ':login', $login );
        $sqlacesso->bindParam( ':senha', $senha );
        $sqlacesso->execute();
        $rows_acesso = $sqlacesso->fetch(PDO::FETCH_NUM);
    }
}

```

Em seguida, com o retorno das informações buscadas no banco de dados, são armazenadas em sessões ainda no arquivo login.php, as informações de ID, login de usuário e seu respectivo nível de acesso. Essas informações após serem mantidas em sessões, é direcionado a um outro arquivo denominado restrito.php, como pode ser visto como exemplo no trecho de código abaixo.

```
if (!isset($_SESSION))
{
    session_start();
    $_SESSION['usuarioID'] = $rows_acesso[0];
    $_SESSION['usuarioNome'] = $rows_acesso[1];
    $_SESSION['usuarioNivel'] = $rows_acesso[2];

    header("Location: restrito.php");
    exit;
}
```

No arquivo restrito.php é feito a comparação de qual nível de acesso o usuário pertence. E dependendo do seu nível, ele possui seu direcionamento correto para determinadas informações, por exemplo o trecho de código abaixo recebe o valor de seu nível de acesso na variável \$nivel\_necessario e o direciona para a página index.php para que ele use as informações nele contidas, caso o usuário tente acessar qualquer página que não lhes cabe o devido acesso, este será barrado e direcionado para tela de *login*, como visto anteriormente.

```
$nivel_necessario = $_SESSION['usuarioNivel'];

if ($nivel_necessario == 1) {
    header("Location: index.php");
    exit;
}
```

Ao passar para a etapa de cadastro de arquivo no sistema, será visto na Figura 17, tela que pode ser acessada ao clicar no botão acessar *upload*, localizado no cabeçalho da página de busca no perfil docente ou administrador, sem grande complexidade, o usuário deverá apenas preencher e selecionar os campos existentes na página e em seguida clicar em *Browse* para conseguir buscar o arquivo a partir de seu computador, como será visto na Figura 18, adiante.

**FIGURA 17** - Tela para realizar *upload* de arquivo, ou seja, publicar o arquivo no sistema

REALIZAR LOGOUT

Portal de Conteúdos e Acervos Acadêmicos de Engenharia de Software

ACESSAR BUSCA CADASTROS E RELATÓRIOS

Selezione Todos os Campos e Inclua o Arquivo para Realizar Upload no Sistema.

| Nome da Obra | Autor           | Tema            | Editora         | Disciplina      | Palavras-Chave  | Tipo de Arquivo |
|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|              | Não Selecionado |

Upload de Arquivo:

Browse ...

O elemento para *upload* de arquivo foi desenvolvido através de códigos HTML, que já possui uma estrutura relativamente pronta para uso, e basta apenas implementá-la no sistema através dos seguintes comandos:

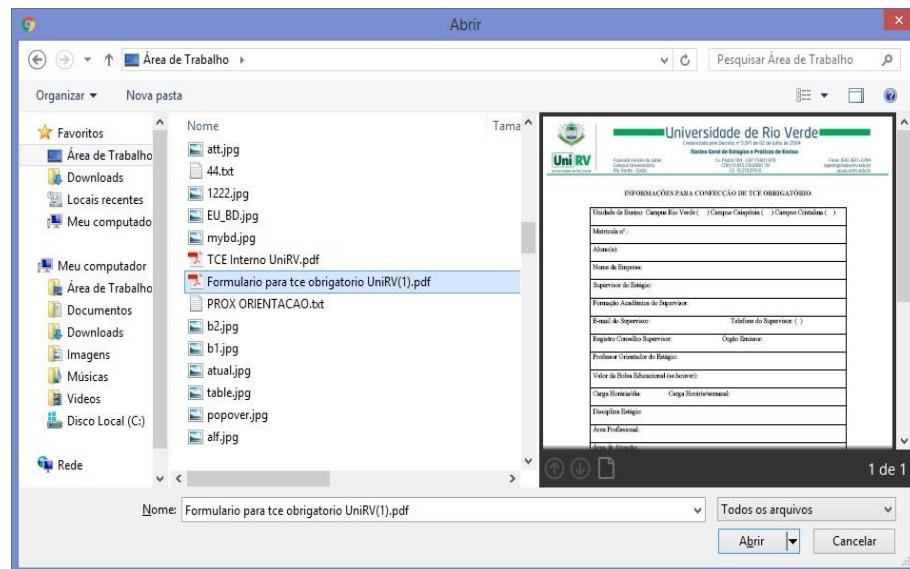
```
<div id="upload-form">
    <label class="control-label" >Upload de Arquivo: <i class="icon-upload"
    ></i></label>

    <input id="input-2" name="arquivo" type="file" class="file" data-show-
    upload="true" data-show-caption="true" onmouseover="valida_forms();">
</div>
```

Para validar a inserção de dados nos campos que envolvem informações ao arquivo que será inserido no sistema, foi elaborado uma função de validação de preenchimento campo a campo através da linguagem *JavaScript* e adicionado um evento de sobreposição do *mouse* no botão *browse* que ativa a função. O sistema não permite a seleção de arquivos para tentativa de *upload* sem que todos os campos estejam devidamente preenchidos e caso algum campo não esteja válido, um alerta personalizado será exibido para o usuário, que foi possível ser criado através do *framework* responsivo de notificações chamado *lobibox*. Um trecho de código responsável por esta validação está presente a seguir:

```
function valida_forms() {  
  
    if(document.getElementById("txtarquivo").value == ""){  
        Lobibox.alert("warning", {  
            msg: "Por favor, preencha o campo Nome da Obra !"  
        });  
        document.getElementById("txtarquivo").focus(); exit;  
    }  
  
    var comboNome = document.getElementById("selautor").selectedIndex;  
    if(comboNome < 0){  
        Lobibox.alert("warning", {  
            msg: "Por favor, preencha o campo Autor !"  
        });  
        exit;  
    }  
  
    var comboNome = document.getElementById("seltema").selectedIndex;  
    if(comboNome < 0){  
        Lobibox.alert("warning", {  
            msg: "Por favor, preencha o campo Tema !"  
        });  
        exit;  
    }  
  
    var comboNome = document.getElementById("seleditora").selectedIndex;  
    if(comboNome < 0){  
        Lobibox.alert("warning", {  
            msg: "Por favor, preencha o campo Editora !"  
        });  
        exit;  
    }  
}
```

**FIGURA 18** - Janela para selecionar arquivo para realizar *upload* no sistema



Após selecionar o arquivo através da janela nativa do sistema operacional que aparecerá depois de clicar em *Browse*, basta o usuário confirmar o arquivo buscado e em seguida ele irá visualizá-lo em uma pequena janela presente na tela de *upload* de arquivo. Esta visualização está apresentada na Figura 19 a seguir, finalizado os passos anteriores, basta clicar em *Upload*.

**FIGURA 19** - Visualização do arquivo selecionado e pronto para ser inserido no sistema.



A visualização do arquivo selecionado é uma facilidade disponível através do *bootstrap framework*, o posicionamento do elemento na página é dado graças ao uso de CSS e junto ao HTML, resulta em um elemento robusto e bem elaborado para esta finalidade. Com um simples CSS também é realizado a tratativa de sobreposição do elemento, para que o mesmo não se sobreponha nos itens que são dispostos a serem selecionados por exemplo, isto foi definido com o atributo z-index do CSS, como pode ser visto no trecho a seguir:

```
#upload-form {  
    position: relative;  
    width: 50%;  
    z-index: 2;  
}
```

Posteriormente, desenvolveram-se os relatórios e a ele foram integrados alguns cadastros na disposição de suas páginas. Com isso, os relatórios visualizados no sistema são basicamente retornados de uma listagem e no mesmo local é possível realizar alterações sem precisar navegar entre outras páginas, várias ações podem ser realizadas na página de cadastros e relatórios, entre elas estão por exemplo, cadastrar um novo autor, tema, editora, disciplina, tipo de arquivo, palavra-chave ou usuário e também estão presentes operações de editar, excluir, inativar e ativar, além de claro, os relatórios visualizados em tela e também podendo exportá-los para .pdf se preferir. Para conseguir desenvolver diversas funcionalidades na mesma página, é preciso harmonizar os componentes, caso contrário poderá facilmente dificultar a utilização da página para o usuário do sistema. Ao pensar em tornar as funcionalidades mais acessíveis, organizadas, amigáveis e tudo no mesmo *layout*, algumas medidas precisam ser tomadas, por exemplo, a utilização de componentes modal, isto é, componentes que sobrepõem a tela atual com um formulário diferente para ser utilizado, desta forma, pode-se manipular diferentes recursos rapidamente. O trecho de código que é responsável pela criação da janela modal de cadastrar tema por exemplo, está disposto a seguir.

```
<!-- CADASTRAR TEMA -->
<div id="cad-modal-tema" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">CADASTRO DE TEMA</h4>
            </div>
            <div class="modal-body">
                <form action="cadastros/cadastrar_tema.php" method="POST">

                    <label>Digite o nome do Tema: </label> <br>
                    <input type="text" name="txtnometema" /> <br><br>

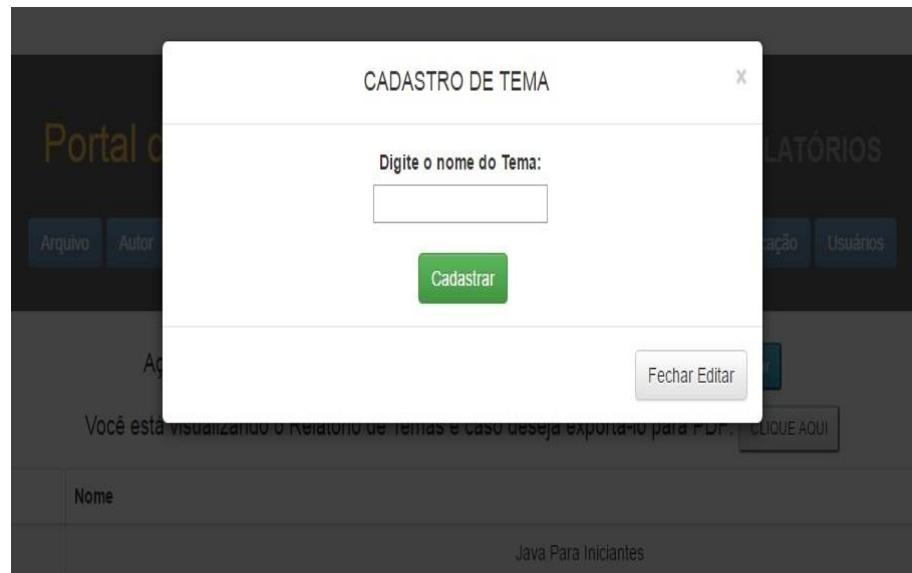
                    <input type="submit" class="btn btn-success" value="Cadastrar" />
                </form>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-dismiss="modal">Fechar
                    Editar</button>
                </div>
            </div>
        </div>
    </div>
</div>
```

Para apresentar a janela modal, basta inserir alguns atributos no botão HTML da página, que são interpretados através do uso do *framework bootstrap*. Segue abaixo um exemplo de código para um botão de chamada do modal.

```
<button type="button" class="btn-sm btn-success" id="modal-tema" name="cadastrar-tema" data-toggle="modal" data-target="#cad-modal-tema" >Cadastrar</button>
```

O resultado pode ser visto na Figura 20 logo abaixo, onde é possível observar a sobreposição do modal com uma leve opacidade da página de fundo, e direcionar a atenção para a janela sobreposta.

**FIGURA 20** - Janela modal para realização de cadastro de novo tema



Anteriormente foi comentado sobre o cadastro de arquivo, realizado na área de *upload*, onde foi demonstrado o trecho do código em *JavaScript* para validação se o campo foi devidamente preenchido ou não, na parte de relatórios também há o uso de validações com o auxílio do *framework lobibox* para harmonizar a visualização dos retornos de validações. Um exemplo de como fica visualmente para o usuário a utilização deste *framework* pode ser observada na Figura 21 a seguir.

**FIGURA 21** - Janela de alerta do framework *lobibox* quando há uma validação em ativa



Outros recursos interessantes na página de cadastros e relatórios é a exportação do relatório para um arquivo pdf que é visualizado inicialmente em uma nova guia do navegador, e pode ser impresso ou baixado para o computador pessoal. Recurso que foi elaborado através da biblioteca mPDF versão 6.0 e foi preciso elaborar código em HTML para que a biblioteca interpretasse o mesmo e disponibilizasse em formato pdf, o resultado como se estivesse sendo exportado de uma própria página *web*, por exemplo, o sistema exporta o resultado para uma tabela em arquivo pdf, esta tabela foi gerada através deste código HTML dentro de uma única variável que foi preciso ser concatenada algumas vezes para executar o PHP que também foi utilizado no momento de busca das informações no banco de dados. O trecho responsável por este recurso pode ser visto abaixo.

```

function criaPDF() {
    include("../conexao.php");
    include("../mpdf60/mpdf.php");

    $sql = $conn->query('select * from autores');

    $implao = '<style type="text/css">
        .tbl {border-collapse:collapse; border-spacing:0; border-color:#000000;}
        .tbl .item{background-color:#c0c0c0;color:#000000;vertical-align:top}
        .tbl .cel{background-color:#efefef;color:#000000;vertical-align:top}
    </style>

<h1> Lista de Autores </h1>
<table class="tbl">
    <tr>
        <td class="item"><strong>ID</strong><br></td>
        <td class="item"><strong>Autores</strong></td>
        <td class="item"><strong>Bloqueados</strong></td>
    </tr>';
    while ($linha = $sql->fetch(PDO::FETCH_ASSOC)) {
        $implao .= '<tr><td class="cel">' . $linha['AUT_ID'] . '</td>
                    <td class="cel">' . $linha['AUT_NOME'] . '</td>
                    <td class="cel">' . $linha['AUT_ACESSO'] . '</td></tr>';
    }
    $implao .= '</table>';

    $mpdf=new mPDF();
    $mpdf->WriteHTML($implao);
    $mpdf->Output();
}

criaPDF();

```

O botão que faz a chamada do arquivo onde executa a função de gerar o relatório em formato pdf pode ser visualizado no trecho de código abaixo.

```
<button type="button" name="cria" id="cria" class="btn-sm btn-default"
value=<?=$linha['AUT_ID'];?>
onclick="javascript:window.open('./relatorios/export_autores.php', '_blank');"> CLIQUE
AQUI </button>
```

Alcançada as operações de CRUD (*Create, Read, Update, Delete*) e seus devidos relatórios, também visto a autenticação de *login*, tela para *upload* de arquivos, tratativas e validações entre várias operações existentes no sistema, restaram-se a pesquisa dos arquivos, visualização do arquivo e obtenção do mesmo.

Para cumprir as etapas restantes, começando pela pesquisa dos arquivos, encontra-se a utilização de duas tabelas, onde uma é responsável por apresentar os campos disponíveis para realizar a busca e ao clicar em buscar encontra-se outra tabela que se dispõe logo abaixo com os resultados da busca realizada. Para manipular os elementos na mesma página, sem a necessidade de enviar as informações do formulário para outra página que seria o resultado da busca, mantém-se os elementos dentro do mesmo formulário e as ações ficam armazenadas em funções *JavaScript*, desta forma, quando desejar executar alguma funcionalidade, basta chamar a função no componente da página específico daquela determinada função, por exemplo, para visualizar o arquivo, uma função simples em *JavaScript* é chamada, que faz o papel de inserir uma ação no formulário da página e então o envia, este processo é conhecido como *submit form*, ou em português, envio de formulário. Pode ser visto como foi criado, através do código abaixo.

```
function add_action() {
    document.getElementById("form-busca").action = "visualiza_p.php";
    document.getElementById("form-busca").submit();
}
```

Para utilização da função *add\_action()*, por exemplo, utiliza-se um evento *onclick*, que é o evento de clicar no botão, onde o mesmo irá fazer uma chamada da função, como no código abaixo.

```
<button id="cry" type="submit" name="endereco" value=<?=$linha['ARQ_ID'] ?>"  
onclick="add_action();" ></button>
```

No código acima, pode-se observar que o botão não possui uma apresentação visual, ou seja, um nome ou disposição na tela a não ser sua estrutura e formato de botão, isto se deve pelo motivo deste exemplo ser um botão desenhado em CSS. Segue abaixo o código do CSS que faz a apresentação do mesmo na página através do seu id.

```
#cry {  
    background-image: url('img/pdf4.png');  
    width:48px;  
    height:48px;  
    border:0;  
    cursor: pointer;  
}
```

Foi preciso tomar esta medida de elaborar o desenho do botão através do CSS pelo motivo de compatibilidade, a ideia seria fazer com que clicasse na imagem para abertura do arquivo, mas o elemento *img* não conseguia levar o parâmetro de id do arquivo para a outra página, a solução seria um *input* do tipo *image*, onde foi identificado que no navegador Mozilla Firefox, esta possível solução não era compatível e a função *JavaScript* não conseguia levar o valor de identificação do arquivo para a página de visualização do mesmo. Para sanar qualquer tipo de incompatibilidade, foi mantido a estrutura de um botão comum, ao qual consegue trafegar informações via *\$\_POST* normalmente e o uso do CSS foi inteiramente para organizar a posição e *design* do botão na tela, além de ajustar seu tamanho exato com o da imagem desejada a ser utilizada.

Retornando-se à página de pesquisa do arquivo, é utilizada uma consulta no banco de dados bem interessante, onde o trecho de código para busca da informação pode ser visto abaixo.

```

$codigo = "SELECT DISTINCT ARQ_ID,ARQ_NOME,conc_busca_autores(ARQ_ID)
as AUT_NOME, conc_busca_temas(ARQ_ID) as TEM_NOME, EDI_NOME,
conc_busca_disciplina(ARQ_ID) as DIS_NOME, conc_busca_palchave(ARQ_ID) as
PAL_CHA_NOME, TIP_ARQ_NOME, ARQ_CAM_ARQ FROM `arquivos` a

LEFT JOIN arquivos_has_disciplinas b ON a.ARQ_ID = b.Arquivos_ARQ_ID
LEFT JOIN disciplinas dis ON b.Disciplinas_DIS_ID = dis.DIS_ID

LEFT JOIN arquivos_has_temas c ON a.ARQ_ID = c.Arquivos_ARQ_ID
LEFT JOIN temas tem ON c.Temas_TEM_ID = tem.TEM_ID

LEFT JOIN autores_has_arquivos d ON a.ARQ_ID = d.Arquivos_ARQ_ID
LEFT JOIN autores aut ON d.Autores_AUT_ID = aut.AUT_ID

LEFT JOIN palavraschave_has_arquivos e ON a.ARQ_ID = e.Arquivos_ARQ_ID
LEFT JOIN palavraschave pal ON e.PalavrasChave_PAL_CHA_ID = pal.PAL_CHA_ID

LEFT JOIN editoras edi ON edi.EDI_ID = a.Editoras_idEditoras
LEFT JOIN tipoarquivo tip ON tip.TIP_ARQ_ID = a.TipoArquivo_idTipoArquivo";
$codigo = $codigo." ";

if($txtnomearquivo != "" or $id_editora != 0 or $id_ta != 0 or $id_disciplina != 0 or
$id_tema != 0 or $id_autor != 0 or $id_palchave != 0){
    $codigo = $codigo."where ( a.ARQ_NOME LIKE '%$txtnomearquivo%' or
'$txtnomearquivo' LIKE "
        and ( a.editoras_ideditoras = $id_editora or $id_editora = " )
        and ( a.TipoArquivo_idTipoArquivo = $id_ta or $id_ta = " )
        and ( b.Disciplinas_DIS_ID in ($disciplina) or '$disciplina' = 0 )
        and ( c.Temas_TEM_ID in($tema) or '$tema' = 0 )
        and ( d.Autores_AUT_ID in ($autor) or '$autor' = 0 )
}

```

//continua...

```
//continuação...

        and ( e.PalavrasChave_PAL_CHA_ID in ($chave) or '$chave' = 0 );
}

$codigo = $codigo." LIMIT $primeiro_registro, $num_por_pagina";

$sqlbusca = $conn->query($codigo);
```

Observa-se que há funções de banco de dados incorporadas na variável \$sqlbusca, são elas responsáveis por concatenar a identificação dos dados para busca de mais de um autor, tema, disciplina ou palavra-chave, o que é um diferencial no sistema de busca do projeto. Para exemplificar como são estas funções, segue um código para exemplo logo abaixo. Lembrar-se que são funções semelhantes, alterar apenas o nome da tabela entre elas e que a visualização do código da função na base de dados pode ser levemente diferente do que será visto aqui, pois o gerenciador do banco de dados pode apresentar parâmetros nativos que não foram definidos e são atribuídos por padrão, sendo assim, segue-se o código de criação da função.

```

DELIMITER $$

create function conc_busca_autores(idarquivo integer)

RETURNS varchar(4000)

BEGIN

DECLARE done INT DEFAULT FALSE;
DECLARE aux varchar(100);
DECLARE result varchar(400);

DECLARE dados CURSOR FOR SELECT AUT_NOME FROM autores_has_arquivos
INNER JOIN autores ON Autores_AUT_ID = AUT_ID WHERE Arquivos_ARQ_ID =
idarquivo;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

SET result = "";

OPEN dados;
read_loop: LOOP
    FETCH dados INTO aux;

    IF done THEN
        LEAVE read_loop;
    END IF;

    IF result = "" THEN
        SET result = aux;
    ELSE
        SET result = concat(result, ", ", aux);
    END IF;
END LOOP;

CLOSE dados;
return result;
END $$

DELIMITER ;

```

De maneira visual, fala-se que a página de pesquisar arquivo dispõe-se aproximadamente como na Figura 22, e pode variar levemente de acordo com o perfil do usuário.

**FIGURA 22** - Tela de pesquisar arquivo disponibilizada para discentes, sem necessidade de *login*

| Nome da Obra         | Autor           | Tema            | Editora         | Disciplina      | Palavras-Chave  | Tipo de Arquivo |
|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| <input type="text"/> | Não Selecionado |

| Nome da Obra      | Autor                            | Tema                          | Editora | Disciplina | Palavras-Chave               | Tipo de Arquivo |
|-------------------|----------------------------------|-------------------------------|---------|------------|------------------------------|-----------------|
| GRU               | Francisco Ferreira, Jose Tavares | Java Para Iniciantes, Java Me | Abril   | Osm        | Desenvolvimento Mobile, Java | Artigo          |
| Java para Novatos | Jose Tavares                     | Java Para Iniciantes, Java Me | Abril   | Estagio    | Java                         | Monografia      |

A visualização de arquivo pode ser realizada após efetuar uma busca e clicar na imagem de arquivo em formato pdf, localizada na primeira coluna da tabela que se dispõe após efetuar a busca. Em seguida, o usuário será encaminhado para uma página de visualização do arquivo e esta página recebe a identificação do arquivo para que assim seja possível disponibilizá-lo através da biblioteca *pdfobject*. Veja no trecho de código abaixo, o objeto que recebe esta informação.

```
<?php $buscaobra = "arquivos/".$linha["ARQ_CAM_ARQ"]; ?>

<div id="pdf">

    <object data="<?php echo htmlentities($buscaobra);?>" type="application/pdf"
width="100%" height="100%"></object>

</div>
```

De acordo com o trecho de código anterior, o elemento possui propriedades de CSS aplicadas através da identificação da *div*, onde estas propriedades são diferentes para

compatibilidade em cada navegador do qual o sistema foi projeto para garantir funcionalidade completa. Ficando da seguinte forma.

```
#pdf {  
    width: 97%;  
    height: min-content;  
    margin: 2em auto;  
    border: 1px solid #6699FF;  
}  
  
.pdf p {  
    padding: 1em;  
}  
  
.pdf object {  
    display: block;  
    border: solid 1px #666;  
}
```

Caso o navegador seja *mozilla firefox*, o seguinte código CSS entrará em vigor para garantir a compatibilidade correta:

```
@-moz-document url-prefix() {  
    .pdf {  
        height: 1400px;  
        border: 1px solid #666;  
    }  
}
```

A Figura 23 a seguir demonstra como ficaria o *layout* da página.

**FIGURA 23** - Tela de visualização do arquivo após efetuada a busca

The screenshot shows a web-based document viewer. At the top, there's a header with a 'Realizar Nova Busca' button and the portal's name. Below the header, a message says 'Você está visualizando o arquivo: GRU' and 'e na tabela abaixo segue suas informações relacionadas'. A table follows, with columns for Autor, TEMA, EDITORA, DISCIPLINAS, PALAVRAS-CHAVE, and TIPO DE ARQUIVO. The data in the table is:

| AUTOR                            | TEMA                          | EDITORAS | DISCIPLINAS | PALAVRAS-CHAVE               | TIPO DE ARQUIVO |
|----------------------------------|-------------------------------|----------|-------------|------------------------------|-----------------|
| Francisco Ferreira, Jose Tavares | Java Para Iniciantes, Java Me | Abril    | Osm         | Desenvolvimento Mobile, Java | Artigo          |

Below the table, the document content is displayed in a large box. It shows the date '19/02/2016' and the title 'GRU - Cobrança'. There is a dashed line below the title, followed by the text 'Corte na linha pontilhada'.

Importante mencionar que há um pequeno *JavaScript* responsável por carregar o objeto em forma de pdf e para isso, precisa-se da biblioteca *pdfobject*, onde ela dispõe também de algumas funcionalidades que podem ser definidas, como opção de *download* por exemplo e este é o principal motivo para adotar esta biblioteca para uso no projeto, visto que não foram encontradas estas possibilidades nas funções nativas com *iframe*. Veja o trecho de código para esta funcionalidade.

```
window.onload = function (){
    var myPDF = new PDFObject({
        url: "<?php echo htmlentities($buscaobra); ?>",
        pdfOpenParams: {
            navpanes: 0,
            toolbar: 1,
            statusbar: 0,
            view: "FitV"
        }
    }).embed("pdf");
};
```

Para obter o arquivo é bastante simples, com a funcionalidade ativada através do código anterior, como pode ser visto o parâmetro *toolbar* está setado em 1, isto significa que ele está ativo, então será disponibilizada uma barra ao passar o mouse por cima do arquivo pdf em aberto, esta barra contém a opção para baixar o arquivo em seu próprio computador, segue na Figura 24 o local para realizar esta operação.

**FIGURA 24** - Tela de visualização do arquivo após efetuada a busca



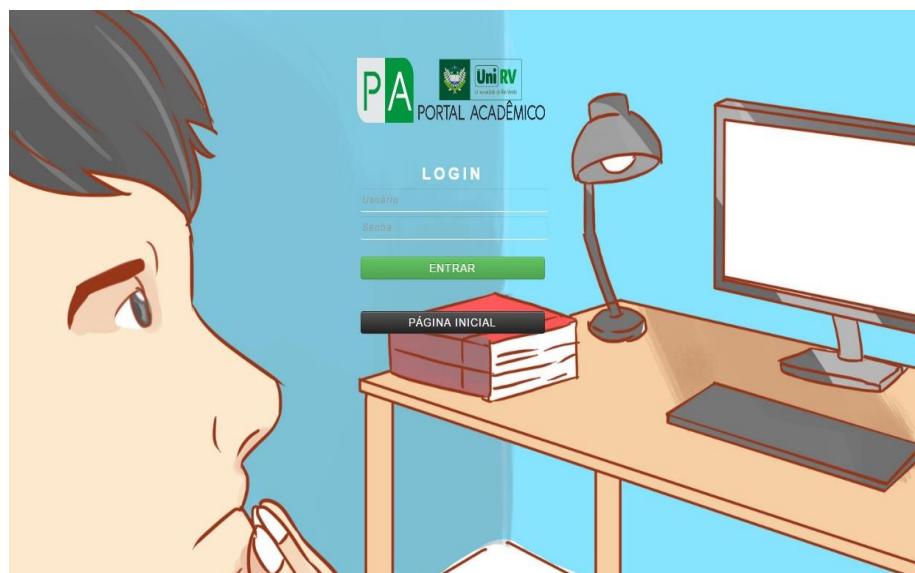
## 4 RESULTADOS E DISCUSSÕES

Ao iniciar o projeto, foram definidas funcionalidades que são elas os requisitos funcionais do *software*, parte essencial a ser implementada para que o resultado final seja conforme o esperado.

O primeiro requisito elaborado corresponde a autenticação de *login*, que no caso é utilizado para que um usuário docente ou administrador consiga acessar as páginas com opções relacionadas a seu perfil.

A tela onde se efetua o *login* está representada na Figura 25 a seguir.

**FIGURA 25** - Tela do primeiro requisito funcional, autenticação de *login*



O próximo requisito é o de Manter Usuário, este é um requisito que pertence as operações CRUD, que juntas totalizam oito requisitos funcionais do *software*, onde todos mantém basicamente a mesma semelhança de *layout* entre si, mas há pequenas diferenças em alguns quando o perfil de usuário logado é docente ou administrador. Para notar as diferenças, estas telas estarão representadas na Figura 26.

**FIGURA 26** - Tela do segundo requisito funcional, manter usuário com o perfil de administrador

| ID | Nome    | Acesso | E-mail                |
|----|---------|--------|-----------------------|
| 24 | admin   | 3      | admin@admin.com       |
| 25 | Douglas | 2      | affedouglas@gmail.com |

Seguindo o mesmo conceito da Figura 27, onde representa o requisito de manter usuário com perfil Administrador, a Figura 28 permite comparar a diferença entre os perfis de usuário, pois o Administrador consegue criar e manipular informações de usuário, já o docente consegue apenas alterar sua própria senha pessoal, visto que não cabe a ele criar ou modificar usuários com permissões para acesso ao sistema e permite ao Administrador um maior controle sobre estas operações.

**FIGURA 27** - Tela do segundo requisito funcional, manter usuário com o perfil de docente

| ID | Nome    | Acesso | E-mail                |
|----|---------|--------|-----------------------|
| 24 | admin   | 3      | admin@admin.com       |
| 25 | Douglas | 2      | affedouglas@gmail.com |

Assim como o de manter usuário, seguem os demais requisitos de operações CRUD bastante semelhantes em questão visualmente e funcional. A Figura 28 mostra o terceiro requisito funcional manter tema, que segue o mesmo *layout* para docente ou administrador.

**FIGURA 28** - Tela do terceiro requisito funcional, manter tema

| ID | Nome                 |
|----|----------------------|
| 12 | Java Para Iniciantes |
| 13 | Java Me              |

O mesmo é válido para o quarto requisito funcional, manter autor, onde a tela segue a mesma para ambos os perfis de usuário. Confira na Figura 29 abaixo, a tela para manter autor.

**FIGURA 29** - Tela do quarto requisito funcional, manter autor

| ID  | Nome               |
|-----|--------------------|
| 115 | Francisco Ferreira |
| 116 | Jose Tavares       |
| 117 | Claudio Souza      |
| 119 | Douglas            |

Seguindo o mesmo conceito dos requisitos de operações de CRUD, o quinto requisito funcional denominado manter editora também segue o mesmo esquema de *layout*, não havendo diferenças entre os perfis de acesso, sendo totalmente semelhantes ao terceiro e quarto requisito visto anteriormente. Na Figura 30 abaixo, pode-se conferir a tela para manter editora.

**FIGURA 30** - Tela do quinto requisito funcional, manter editora

| ID | Nome    |
|----|---------|
| 5  | Amburgo |
| 6  | Abril   |

Não menos importante, o próximo passo é o sexto requisito funcional denominado manter disciplina, este requisito segue basicamente a mesma lógica e *design* dos demais vistos anteriormente devido a também fazer parte das operações de CRUD.

É importante e imprescindível não só visualizar a tela de cada requisito, mas sim as funcionalidades a qual ele se propõe. Por esta razão, será demonstrada as operações de cadastrar, editar, inativar e ativar relacionadas a manter disciplina, porém o mesmo é válido para todos os demais requisitos funcionais de CRUD, levemente diferenciado a parte de usuário, que será vista também como ficou esta diferença existente. Segue na Figura 31 abaixo, o sexto requisito funcional, manter disciplina.

**FIGURA 31** - Tela do sexto requisito funcional, manter disciplina

The screenshot shows a dark-themed web application interface. At the top, there's a navigation bar with a 'Voltar' button (with a back arrow icon) and several tabs: Arquivo, Autor, Tema, Editora, Disciplina, Palavras Chave, Tipo de Arquivo, Data de Publicação, and Usuários. Below the navigation, a message says 'Ações de Cadastro e Manipulação de Disciplinas:' followed by four buttons: Cadastrar (green), Editar (orange), Inativar (red), and Ativar (blue). Another message below says 'Você está visualizando o Relatório de Disciplinas e caso deseja exportá-lo para PDF: CLIQUE AQUI' with a 'CLIQUE AQUI' button. The main content area contains a table with columns 'ID' and 'Nome'. The data rows are:

| ID | Nome         |
|----|--------------|
| 12 | Osm          |
| 13 | Algoritmos I |
| 14 | Estagio      |
| 15 | Redes        |

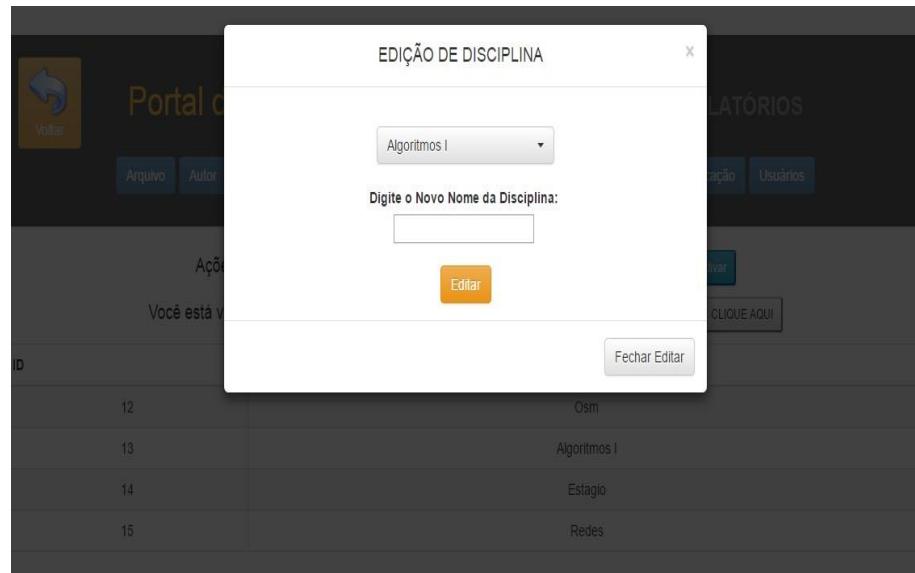
Como informado anteriormente no parágrafo acima, ao clicar no botão Cadastrar, a tela para cadastro de uma nova disciplina, por exemplo, seria como na Figura 32.

**FIGURA 32** - Tela de manter disciplina, operação de cadastro de nova disciplina

This screenshot shows a modal dialog box titled 'CADASTRO DE DISCIPLINA' overlaid on the main application interface. The dialog contains a text input field labeled 'Digite o nome da Disciplina:' and a green 'Cadastrar' button. In the bottom right corner of the dialog, there are 'Fechar' and 'Editar' buttons. The background of the application shows the same table of disciplines as in Figure 31. The table data is identical to Figure 31.

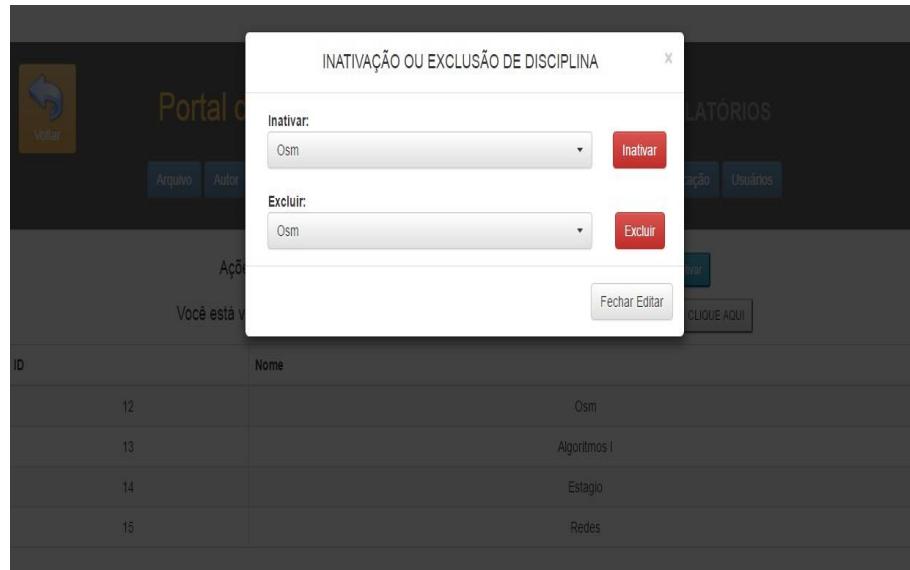
A seguir, na Figura 33 logo abaixo, pode-se observar a próxima operação, no caso, de Editar e também relacionada a disciplina.

**FIGURA 33** - Tela de manter disciplina, operação de editar nome da disciplina existente



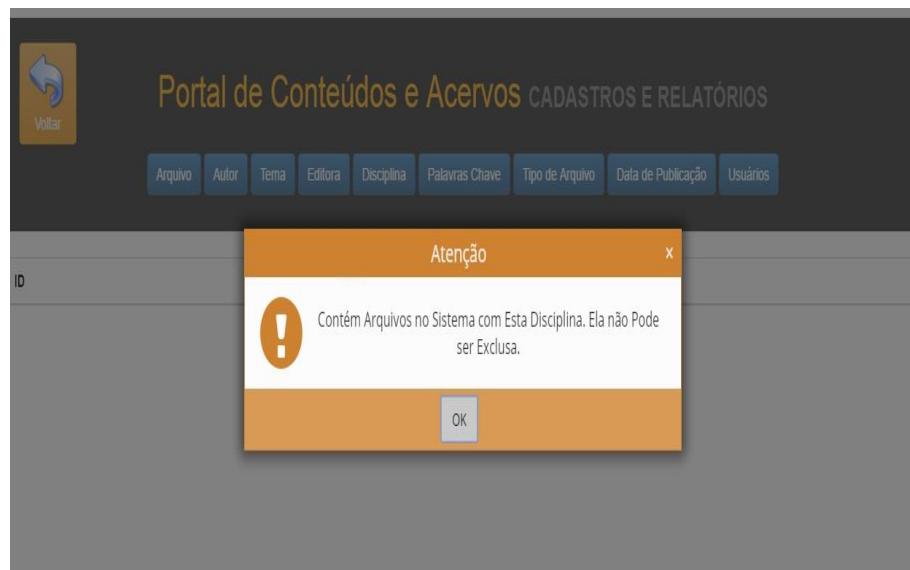
No momento de excluir ou inativar a disciplina, por exemplo, é interessante estar atento, mas o sistema também não deixa que o usuário faça uma operação que possa prejudicar qualquer tipo de funcionamento relacionado àquele item, isto é, caso tenha alguma disciplina já vinculada em algum arquivo, esta com certeza não poderá ser removida do sistema e poderá apenas ser inativada caso não possa mais ser inserido novos arquivos no sistema com aquela determinada disciplina e também os arquivos relacionados a ela devem se manter cadastrados, pois ainda há usuários que possam ter o interesse de pesquisá-los. Caso cadastre uma disciplina e tenha notado que cadastrou errado, obviamente será possível removê-la do sistema, visto que ainda não foi cadastrado nenhum tipo de arquivo que pertence aquela disciplina. Essa verificação existe em todas as operações de CRUD. Veja na Figura 34 abaixo a tela de excluir ou inativar disciplina.

**FIGURA 34** - Tela de manter disciplina, operações de excluir ou inativar disciplina



Como informado, caso tente realizar a exclusão de uma disciplina que contém algum arquivo relacionado, a tratativa, para que esta operação não prossiga, entra em vigor, de acordo com Figura 35.

**FIGURA 35** - Tratativa de erro para evitar disciplina de ser removida quando há arquivos ligados a ela



Outra tratativa interessante é que o sistema não permite em hipótese alguma que o cadastro seja realizado de forma duplicada, isto vale para novos cadastros e também para edição dos cadastros, como segue na Figura 36, o exemplo com a tentativa de cadastrar uma

disciplina que já existe no sistema. Ele também trata de questões de maiúsculas, minúsculas, não permite de forma alguma que o cadastro seja duplicado e as informações são gravadas no banco de dados seguidas de um padrão, onde a primeira letra de cada palavra é maiúscula e as demais minúsculas, para isso, foi necessário criar uma função apropriada no banco de dados.

**FIGURA 36** - Tratativa de erro para evitar disciplina de ser inserida com nome já existente



Seguindo ao sétimo requisito funcional, ainda está dentro de operações com CRUD e não há novidades no *layout*, basicamente é o que já foi visto nos requisitos anteriores que pertencem também a este tipo de operação, segue-se o mesmo padrão. Pode ser visto na Figura 37 logo abaixo o requisito de Manter Palavra Chave.

**FIGURA 37** - Tela do sétimo requisito funcional, manter palavra chave



| ID | Nome                   |
|----|------------------------|
| 8  | Desenvolvimento Mobile |
| 9  | Arvore Binaria         |
| 10 | Java                   |
| 11 | Php                    |
| 12 | C++                    |

Em continuidade às funcionalidades do sistema, segue-se para o oitavo requisito funcional conhecido como Manter Tipo de Arquivo, no mesmo padrão dos demais como pode ser visto na Figura 38 abaixo.

**FIGURA 38** - Tela do oitavo requisito funcional, manter tipo de arquivo



| ID | Nome       |
|----|------------|
| 5  | Monografia |
| 6  | E-book     |
| 7  | Artigo     |

Para finalizar as operações de CRUD, tem-se o nono requisito funcional, denominado Manter Arquivo. Este requisito tem características diferentes dos demais vistos anteriormente,

pois ele não segue o mesmo padrão de *layout* e possui uma estrutura totalmente diferente e voltada para ele, é o que será visto na Figura 39.

**FIGURA 39** - Tela do nono requisito funcional, manter arquivo

| Nome da Obra | Autor                  | Tema                 | Editora | Disciplina          | Palavras-Chave                | Tipo de Arquivo |
|--------------|------------------------|----------------------|---------|---------------------|-------------------------------|-----------------|
| Java XLZ     | Claudio Souza, Douglas | Java Para Iniciantes | Abril   | Algoritmos I, Redes | Desenvolvimento Mobile , Java | Artigo          |

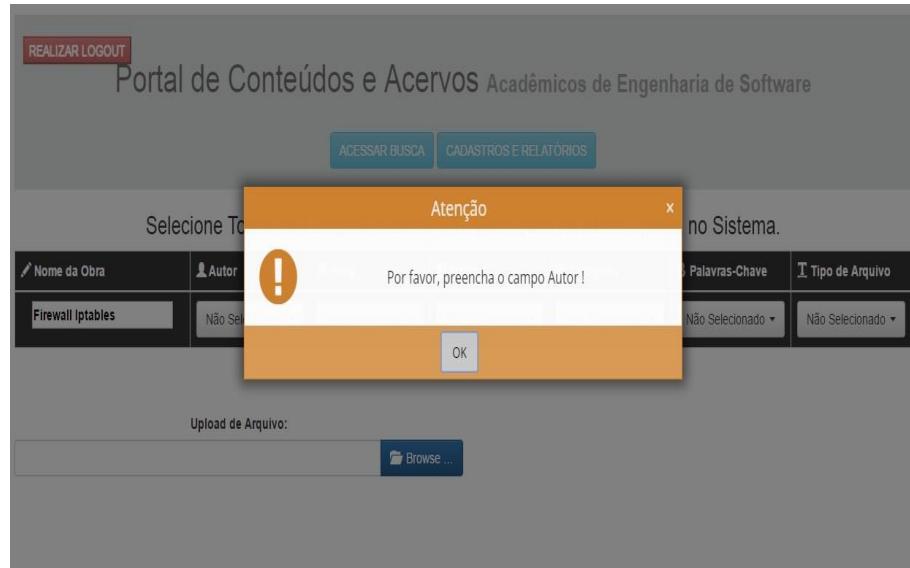
Upload de Arquivo:

012430\_COMPLETE0.pdf

Remove Upload Browse...

O Manter Arquivo possui uma tela onde o cadastro ou publicação do arquivo é feito com base em parâmetros que se seguem para serem selecionados em uma tabela e inserido o nome do arquivo, é obrigatório o preenchimento de todos os parâmetros, o sistema não aceita inserção de arquivos com algum parâmetro vazio, e acusa assim um alerta de erro antes mesmo de permitir que o usuário faça anexo do arquivo para realizar *upload*. Esta tratativa para não permitir que falte parâmetros na inserção do arquivo pode ser vista na Figura 40 abaixo.

**FIGURA 40** - Tratativa que valida se todos os parâmetros foram inseridos para liberar anexo de arquivo



Lembrar que para publicar arquivo no sistema, é preciso estar logado com usuário de perfil docente ou administrador e para realizar a edição do arquivo, que faz parte do mesmo requisito também é preciso que esteja com estes níveis de acesso. A edição fica em página diferente do cadastro do arquivo, para ter acesso a página de edição do arquivo, o docente ou administrador precisa realizar a busca do arquivo pelo nome ou qualquer outro filtro de busca e clicar no botão Atualizar na coluna de Ações localizada na tabela de resultados da busca, segue Figura 41 de como acessar a página de edição e também exclusão do arquivo. Somente o administrador pode excluir o arquivo, mesmo que o docente consiga adicionar um arquivo, ele precisa solicitar ao administrador para que o mesmo seja removido, caso necessário.

**FIGURA 41** - Tela de edição e exclusão de arquivo, requisito manter arquivo

The screenshot shows a web application interface for managing files. At the top, there is a red button labeled 'REALIZAR LOGOUT'. Below it, the title 'Portal de Conteúdos e Acervos' and the subtitle 'Acadêmicos de Engenharia de Software'. A horizontal menu bar with three buttons: 'ACESSAR BUSCA' (highlighted in blue), 'ACESSAR UPLOAD', and 'ACESSAR RELATORIOS'. The main content area has a heading 'Este é o Arquivo a ser Editado'. Below this is a table with columns: 'Nome da Obra', 'Autor', 'Tema', 'Editora', 'Disciplina', 'Palavras-Chave', and 'Tipo de Arquivo'. The table contains one row of data: 'GRU', 'Francisco Ferreira, Jose Tavares', 'Java Para Iniciantes, Java Me', 'Abril', 'Osm', 'Desenvolvimento Mobile, Java', and 'Artigo'. Each column has an 'Editar' button. Below the table, a message 'Deseja excluir o arquivo?' is displayed, with a 'Clique Aqui' button underneath.

Após verificar toda a parte de CRUD do sistema e seus níveis de acesso, há outros requisitos essenciais para o funcionamento do sistema e são os mais utilizados pelo usuário final. Segue na Figura 42, o décimo requisito funcional, denominado Pesquisar Arquivo, neste requisito, o usuário possui um único campo de texto para digitar o nome do arquivo ao que se deseja fazer a busca e outros seis campos que são selecionáveis, importante mencionar que cada campo que pode ser selecionado, há um recurso de busca dentro dele, graças a utilização do *bootstrap-select*, foi possível adicionar esta funcionalidade que assim permite que o usuário encontre o que procura com mais facilidade caso tenha uma lista muito grande de informações cadastradas que possam ser selecionadas, ele não precisa utilizar o *scroll* de rolagem e ler um a um, e pensar na agilidade desta busca, basta digitar o nome dentro do campo ainda não selecionado que ele irá retornar a cada caractere digitado, as informações correspondentes, este é sem dúvidas, um diferencial do sistema, pois os campos permitem que seja realizado uma pesquisa de um ou mais autores, temas, disciplinas e palavras-chave limitando-se em até cinco itens em cada um destes campos, para os campos de editora e tipo de arquivo segue apenas a possibilidade de pesquisar um por vez, visto que um arquivo só pode ter uma editora e ser de um único tipo.

Após realizar a busca, o usuário irá visualizar os resultados em forma de tabela logo abaixo do botão de Realizar Pesquisa, nesta tabela ele encontra as informações relacionadas ao arquivo buscado e ao clicar na imagem com o nome do arquivo, ele realiza a abertura do mesmo para visualização, que é o nosso próximo requisito.

**FIGURA 42** - Tela do décimo requisito funcional, pesquisar arquivo

REALIZAR LOGIN

Portal de Conteúdos e Acervos Acadêmicos de Engenharia de Software

Selecione Um ou Mais Campos Abaixo para Buscas.

| Nome da Obra         | Autor  | Tema   | Editora  | Disciplina                                     | Palavras-Chave                                 | Tipo de Arquivo                                |
|----------------------|--|--|--|--|--|--|
| <input type="text"/> | <input type="button" value="Não Selecionado"/> |

Anterior [ 1 ] Próxima

| Nome da Obra      | Autor                            | Tema                          | Editora | Disciplina | Palavras-Chave               | Tipo de Arquivo |
|-------------------|----------------------------------|-------------------------------|---------|------------|------------------------------|-----------------|
| GRU Cobrança      | Francisco Ferreira, Jose Tavares | Java Para Iniciantes, Java Me | Abril   | Osm        | Desenvolvimento Mobile, Java | Artigo          |
| Java para Novatos | Jose Tavares                     | Java Para Iniciantes, Java Me | Abril   | Estagio    | Java                         | Monografia      |
| teste             | Jose Tavares                     | Java Me                       | Abril   | Estagio    | Java                         | Artigo          |

Sabendo-se como funciona o método de filtros para pesquisa de arquivos e que ao clicar na imagem do arquivo o mesmo será aberto em uma página para visualização, este é o próximo passo a ser discutido, o requisito visualizar arquivo é de fato o que irá retornar de conteúdo para o usuário final, é sem dúvidas o requisito chave do projeto que foi informado na documentação DERS a garantia funcional em dois navegadores, sendo eles o *Google Chrome* e *Mozilla Firefox*, desta forma, unindo as tecnologias de ambos navegadores juntamente com a biblioteca PDFObject, foi possível desenvolver uma solução em que o cliente não precisa ter um leitor de PDF instalado no computador, dependendo assim apenas do acesso à internet e do navegador para conseguir acessar todas as funcionalidades do sistema, inclusive a de visualização de PDF. Segue na Figura 43, a tela de visualização do arquivo.

**FIGURA 43** - Tela do décimo primeiro requisito funcional, visualizar arquivo

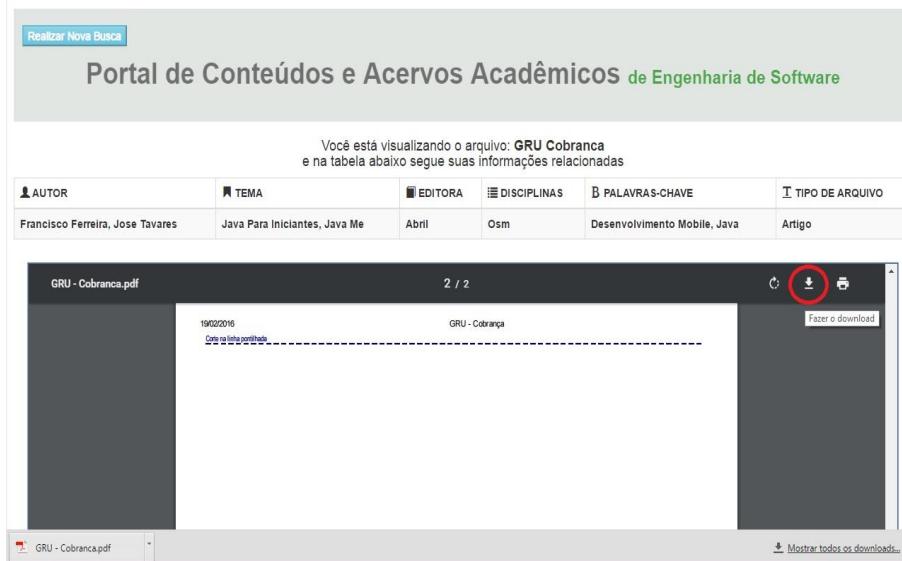
The screenshot shows a web page titled "Portal de Conteúdos e Acervos Acadêmicos de Engenharia de Software". A button at the top left says "Realizar Nova Busca". Below the title, a message states: "Você está visualizando o arquivo: GRU Cobrança e na tabela abaixo segue suas informações relacionadas". A table follows, with columns: AUTOR, TEMA, EDITORA, DISCIPLINAS, PALAVRAS-CHAVE, and TIPO DE ARQUIVO. The data in the table is:

| AUTOR                            | TEMA                          | EDITORAS | DISCIPLINAS | PALAVRAS-CHAVE               | TIPO DE ARQUIVO |
|----------------------------------|-------------------------------|----------|-------------|------------------------------|-----------------|
| Francisco Ferreira, Jose Tavares | Java Para Iniciantes, Java Me | Abril    | Osm         | Desenvolvimento Mobile, Java | Artigo          |

Below the table is a large grayed-out area representing the document content. At the top left of this area, there is some small, illegible text.

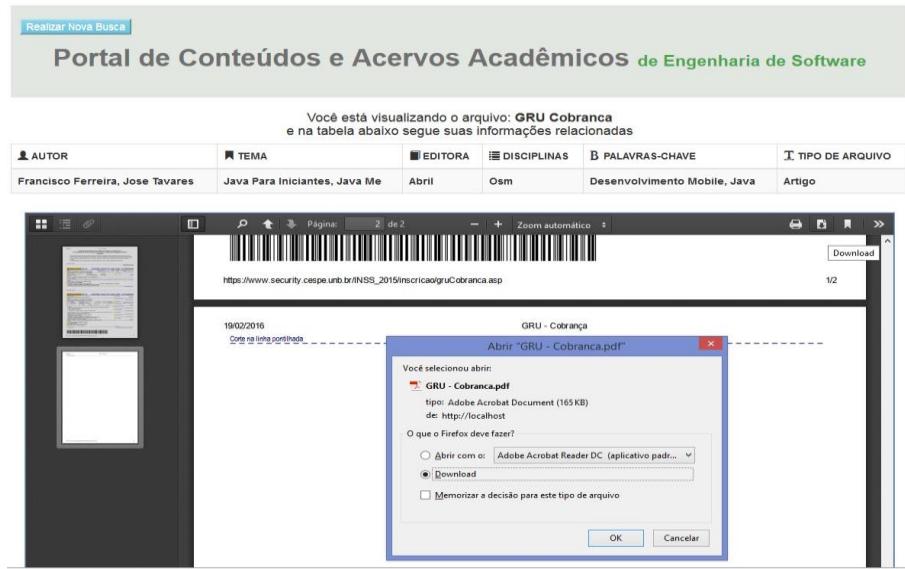
Com o arquivo sendo visualizado, caso tenha a necessidade de obter o arquivo para sua própria leitura fora do Portal Acadêmico, basta passar o cursor acima do arquivo e este mostrará uma barra de navegação com algumas informações como nome do arquivo que foi realizado *upload* no portal, quantidade de páginas, rotação da página, opção de *download* e impressão, maximização do arquivo PDF ao redor do local de leitura do arquivo e também opções para ampliar ou reduzir a visualização do arquivo no local de leitura. Segue na Figura 44, a visualização destas opções no navegador Google Chrome.

**FIGURA 44** - Tela do décimo segundo requisito funcional, obter arquivo, utilizando o navegador Google Chrome.



É importante mencionar que este *layout* da visualização do arquivo varia de acordo com o navegador que está sendo utilizado. Na Figura 45 abaixo segue o modelo de exemplo como fica a visualização com o navegador Mozilla Firefox, com algumas outras funcionalidades diferentes, cada uma é nativa de cada navegador.

**FIGURA 45** - Tela do décimo segundo requisito funcional, obter arquivo, utilizando o navegador Mozilla Firefox



Para emitir o relatório em formato pdf, é preciso acessar a área de cadastros e relatórios e ao clicar no botão relacionado a modalidade que deseja visualizar o relatório, ele será listado na tela, onde também é informado um botão para clicar que atribui os dados listados na tabela do relatório visualizado na tela diretamente para um arquivo pdf também em formato tabela, esta funcionalidade só foi possível ser desenvolvida graças ao uso da biblioteca mPDF que possibilita gerar arquivos pdf a partir do HTML.

Os relatórios são páginas acessíveis por qualquer usuário ou perfil, porém há algumas restrições, o relatório de usuário é disponibilizado apenas para os docentes e administrador e entre eles também há pequenas diferenças, onde o administrador consegue criar usuários docentes e também alterar o nível de acesso deles, assim como o nome que é o *login* do usuário e também a senha, já o docente consegue apenas editar a sua própria senha pessoal. O usuário comum, ou seja, discente, consegue visualizar os demais relatórios porém ele não consegue exportá-los para pdf, funcionalidade que já é liberada para os docentes e administrador. Todas as exportações de relatórios para arquivo pdf são semelhantes entre si, porém cada uma com suas informações específicas como ID, Nome e Status. Se está bloqueado irá aparecer um “B” para as informações que são relacionadas aos arquivos, por exemplo autor, tema, disciplina ou “0” caso seja um usuário inativo no sistema, na parte de exportação da Lista de Usuários. Na Figura 46, segue a tela do décimo terceiro requisito Emitir Relatório de Usuários.

**FIGURA 46** - Tela do décimo terceiro requisito funcional, emitir relatório de usuários

| ID | Nome    | Acesso | E-mail                |
|----|---------|--------|-----------------------|
| 24 | admin   | 3      | admin@admin.com       |
| 25 | Douglas | 2      | affedouglas@gmail.com |

No caso de usuário, há também informação de e-mail, como visto na Figura 46, o *layout* de exportação do relatório de usuários, nele é possível observar a questão de acesso, onde se for igual a “3”, o usuário é administrador, “2” é o acesso de docente, “1” é um perfil de usuário normal que pode ser utilizado para remover acesso de um usuário sem bloqueá-lo, tem os mesmos acessos que alguém sem usuário, no caso, o discente e “0” é a inativação ou bloqueio do usuário.

Prosseguindo as funcionalidades do sistema, onde está por agora na parte de Emitir Relatórios, que é responsável por nove requisitos funcionais, todos eles são resultados dos nove requisitos de CRUD vistos anteriormente, desta forma, cada CRUD possui seu relatório. O próximo requisito será o décimo quarto, denominado Emitir Relatório de Disciplinas, como visto na Figura 47.

**FIGURA 47** - Tela do décimo quarto requisito funcional, emitir relatório de disciplinas

| ID | Disciplinas  | Bloqueados |
|----|--------------|------------|
| 12 | Osm          |            |
| 13 | Algoritmos I |            |
| 14 | Estagio      |            |
| 15 | Redes        |            |

Seguindo basicamente o mesmo esquema de *layout* dos relatórios anteriores, pode-se citar o décimo quinto requisito funcional denominado Emitir Relatório de Autores, visto na Figura 48.

**FIGURA 48** - Tela do décimo quinto requisito funcional, emitir relatório de autores

| ID  | Autores            | Bloqueados |
|-----|--------------------|------------|
| 115 | Francisco Ferreira |            |
| 116 | Jose Tavares       |            |
| 117 | Claudio Souza      |            |
| 119 | Douglas            |            |

O décimo sexto requisito funcional do sistema é o Emitir Relatório de Temas, onde pode-se observar na Figura 49 em seguida, que possui um “N” na coluna Bloqueados, isto significa que este tema já foi ativado alguma vez e permanece ativo normalmente. Se o campo

bloqueados estiver vazio, significa que o acesso também está normal e que este item nunca sofreu nenhuma alteração se quer.

**FIGURA 49** - Tela do décimo sexto requisito funcional, emitir relatório de temas



The screenshot shows a web page titled "Lista de Temas" (List of Topics) from a file named "export\_temas.php". The page displays a table with two rows. The columns are labeled "ID", "Temas", and "Bloqueados". The first row contains ID 12 and the topic "Java Para Iniciantes". The second row contains ID 13 and the topic "Java Me", with the "Bloqueados" column showing the value "N". The page has a dark header bar with icons for refresh, download, and print. On the right side, there are three circular buttons with symbols: a double arrow, a plus sign, and a minus sign.

| ID | Temas                | Bloqueados |
|----|----------------------|------------|
| 12 | Java Para Iniciantes |            |
| 13 | Java Me              | N          |

O próximo da lista de requisitos é o Emitir Relatório de Editoras, e que ocupa o décimo sétimo lugar e mantém basicamente a mesma sequência dos anteriores como pode ser visto na Figura 50 logo abaixo.

**FIGURA 50** - Tela do décimo sétimo requisito funcional, emitir relatório de editoras



The screenshot shows a web page titled "Lista de Editoras" (List of Publishers) from a file named "export\_editoras.php". The page displays a table with two rows. The columns are labeled "ID", "Editoras", and "Bloqueados". The first row contains ID 5 and the publisher "Amburgo". The second row contains ID 6 and the publisher "Abril". The page has a dark header bar with icons for refresh, download, and print. On the right side, there are three circular buttons with symbols: a double arrow, a plus sign, and a minus sign.

| ID | Editoras | Bloqueados |
|----|----------|------------|
| 5  | Amburgo  |            |
| 6  | Abril    |            |

Seguindo para o décimo oitavo requisito funcional Emitir Relatório de Palavra Chave, observa-se na Figura 51 que há uma palavra chave bloqueada e que não será acessada na aplicação. Quando se possui um valor bloqueado, como por exemplo uma palavra chave, ela pode conter em algum arquivo cadastrado com esta palavra, porém não se pode cadastrar novos arquivos com ela, e fica assim somente os que foram inseridos até o momento de seu bloqueio.

**FIGURA 51** - Tela do décimo oitavo requisito funcional, emitir relatório de palavra-chave

| ID | Palavras Chave         | Bloqueados |
|----|------------------------|------------|
| 8  | Desenvolvimento Mobile |            |
| 9  | Arvore Binaria         | B          |
| 10 | Java                   |            |
| 11 | Php                    |            |
| 12 | C++                    |            |

O décimo nono requisito funcional, Emitir Relatório de Arquivos é bastante simples, seguido do modelo dos anteriores já vistos, é válido informar que estes relatórios são independentes entre si e com objetivo de retornar para quem os emite a ciência do que possui cadastrado no sistema, com base nisso ele não vincula informações entre um relatório e outro, por exemplo, retornar a quais autores pertence um determinado arquivo, os relatórios foram especificados independentes, sendo cada um deles um único requisito, sendo assim foram desenvolvidos de formas independentes sem relacionamentos. Segue na Figura 52 o requisito de Emitir Relatório de Arquivos.

**FIGURA 52** - Tela do décimo nono requisito funcional, emitir relatório de arquivos

A screenshot of a web browser window titled "export\_arquivos.php". The main content area is titled "Lista de Arquivos". Below it is a table with two columns: "ID" and "Nome". The table contains three rows of data:

| ID  | Nome              |
|-----|-------------------|
| 124 | GRU Cobranca      |
| 125 | Java para Novatos |
| 126 | teste             |

On the right side of the browser window, there are three circular buttons with icons: a magnifying glass, a plus sign, and a minus sign.

O vigésimo requisito denominado Emitir Relatório por Tipo de Arquivo também está presente e seguindo a ordem dos demais relacionados a sua característica e modelo, respeitando a forma como foi especificado no DERS, como pode ser visto na Figura 53 abaixo.

**FIGURA 53** - Tela do vigésimo requisito funcional, emitir relatório de tipo de arquivo

A screenshot of a web browser window titled "export\_tipo\_arquivo.php". The main content area is titled "Lista de Tipo de Arquivos". Below it is a table with three columns: "ID", "Tipo de Arquivo", and "Bloqueados". The table contains three rows of data:

| ID | Tipo de Arquivo | Bloqueados |
|----|-----------------|------------|
| 5  | Monografia      |            |
| 6  | E-book          |            |
| 7  | Artigo          |            |

On the right side of the browser window, there are three circular buttons with icons: a magnifying glass, a plus sign, and a minus sign.

Para encerrar a questão de funcionalidade do sistema, segue o vigésimo primeiro requisito, Emitir Relatório por Data de Publicação que informa no formato data/dia, quando o arquivo foi publicado no sistema e qual o arquivo. Veja abaixo na Figura 54.

**FIGURA 54** - Tela do vigésimo primeiro requisito funcional, emitir relatório por data de publicação

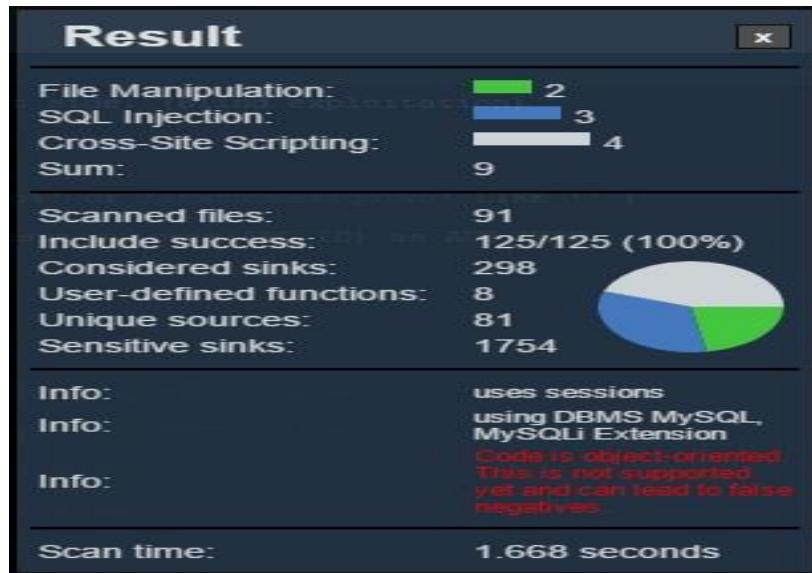
The screenshot shows a web browser window with the title 'export\_dt\_publicacao.php'. The main content is a table titled 'Lista de Datas de Publicações' with three rows. The columns are 'DATA PUBLICAÇÃO' and 'ARQUIVO'. The data is as follows:

| DATA PUBLICAÇÃO | ARQUIVO           |
|-----------------|-------------------|
| 2016-05-20      | teste             |
| 2016-05-19      | GRU Cobrança      |
| 2016-05-19      | Java para Novatos |

Finalizado o último requisito, entende-se que foi possível elaborar e cumprir todas as especificações do sistema que foram definidas na lista de casos de uso, e permite passar para a próxima etapa que são os testes de *software*, responsáveis para verificação e validação da aplicação *web* e garantia da qualidade.

Foram realizados três testes de *software*, o primeiro sendo o teste de análise estática automatizada através da ferramenta RIPS versão 0.55, mostrando-se relevante por localizar vulnerabilidades existentes na aplicação *web*, observa-se os tipos de falhas que foram encontradas de acordo com a Figura 55.

**FIGURA 55** - Resultado do teste de análise estática realizado na ferramenta RIPS



Do teste obtido, encontra-se os seguintes resultados:

- *File Manipulation*: o invasor pode realizar gravação de arquivo arbitrário ou escrever códigos arbitrários em um arquivo que possua essa vulnerabilidade;
- *SQL Injection*: o invasor pode manipular consultas ao banco de dados através da aplicação, que são enviadas para o banco diretamente;
- *Cross-Site Scripting*: o invasor pode realizar uma injeção de *JavaScript* ou *HTML* arbitrário em uma página que possua essa vulnerabilidade e ser exibido para outro usuário qualquer.

No resultado obtido, constata-se 9 vulnerabilidades que não afetam a funcionalidade do sistema e não impedem a utilização do mesmo, dentre estas vulnerabilidades encontradas, encontra-se quatro *Cross-Site Scripting* que relata insegurança ao utilizar o comando *echo* no PHP para impressão de caracteres, porém os servidores *web* atualmente, possuem um recurso chamado *content-type-security police* que não permite a execução de *scripts* através de cabeçalho de página, caso semelhante para o *File Manipulation*.

As três vulnerabilidades de *SQL Injection* estão presentes nas páginas de busca, onde não foi possível a inserção do método *mysqli\_real\_escape\_string()* que ressolveria a falha, pois as entradas de dados não se organizavam dentro da estrutura das funções existentes no banco de dados. Porém esta falha não se seguiria adiante, visto que a aplicação com PDO está apta a aceitar apenas um parâmetro por *query*, isto é, irá realizar a primeira consulta que é executada

pela aplicação, caso fosse utilizado alguma função *mysqli\_multi\_query()*, então esta vulnerabilidade teria maior potencial, pois aceitaria instruções além da existente pela aplicação.

O segundo teste de *software* realizado foi o teste unitário através da ferramenta PHPUnit inserida no NetBeans, que por sua característica, é responsável por analisar o trecho de um código e por não ser um teste automatizado, é preciso criar a rotina para execução do teste manualmente, isto implica que o desenvolvedor saberá exatamente o que irá ser testado e qual será a possível saída, pois ele precisa fazer tudo manualmente, mostrar qual é a entrada de dados e qual deverá ser a saída, se for diferente disso, o teste falhou, se for igual ao que ele esperava, o teste será aprovado.

Para realizar o teste, foi desenvolvido o seguinte trecho de código:

```
class TesteUnit extends PHPUnit_Framework_TestCase {

    public function testUnitAndFF(){
        $var = array();
        $this->assertEquals(0, count($var));
        array_push($var, 'foo');
        $this->assertEquals('foo', $var[count($var)-1]);
        $this->assertEquals(1, count($var));
        $this->assertEquals('foo', array_pop($var));
        $this->assertEquals(0, count($var));
    }
}
```

Este código declara uma variável \$var do tipo array(), em seguida utiliza-se o *assertEquals* que é um comando da extensão *PHPUnit Framework TestCase* responsável por fazer comparações, desta forma o código acima valida se esta variável é 0 ou vazia. Neste caso, o teste passou pois apenas foi declarado a variável do tipo array() mas não houve nenhuma inserção de valor nela, ou seja, o teste foi realizado com sucesso. Um trecho de código similar a este é utilizado no sistema, da seguinte forma:

```

$id_disciplina = Array();
$id_tema = Array();
$id_palchave = Array();
$id_autor = Array();

if (empty($_POST['id-disciplina'])){
    $id_disciplina = 0;
} else {
    $id_disciplina = $_POST['id-disciplina'];
}

if (empty($_POST['id-tema'])){
    $id_tema = 0;
} else {
    $id_tema = $_POST['id-tema'];
}

if (empty($_POST['id-autor'])){
    $id_autor = 0;
} else {
    $id_autor = $_POST['id-autor'];
}

if (empty($_POST['id-pal-chave'])){
    $id_palchave = 0;
} else {
    $id_palchave = $_POST['id-pal-chave'];
}

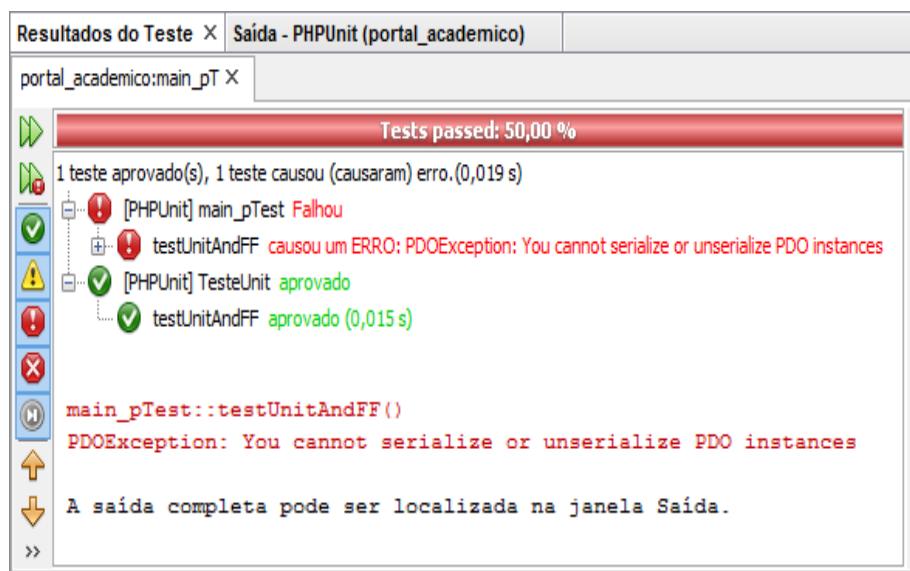
$tema = join(',',$id_tema);
$autor = join(',',$id_autor);
$chave = join(',',$id_palchave);
$disciplina = join(',',$id_disciplina);

```

Este código é utilizado no sistema para receber em variáveis do tipo array() os id's de temas, autores, palavras chave e disciplinas, pois estes campos podem conter nenhuma, uma ou até cinco informações, em seguida é realizado um *join* dos resultados separados por vírgula dentro de outra variável para algumas tratativas e consultas no banco de dados.

Segue a janela de saída com os resultados do teste na Figura 56.

**FIGURA 56** - Janela de resultados da execução do teste de *software* com PHPUnit



Lembrando que, o *PHPUnit* não realiza serialização do PDO, ou melhor dizendo, as instâncias PDO não podem ser serializadas ou desenserializadas, portanto ele acusa o erro encontrado, pois são utilizadas conexões com o banco de dados no arquivo main\_pTest que utiliza as instâncias PDO. Contudo há uma forma de ignorar o PDO, e consegue assim ser realizado o teste, através do código abaixo acrescentado acima da classe que foi realizado a extensão do *PHPUnit Framework TestCase*, que fica da seguinte forma por exemplo:

```
/**
 * @backupGlobals disabled
 * @backupStaticAttributes disabled
 */
class TesteUnit extends PHPUnit_Framework_TestCase {
    //códigos de teste inseridos aqui...
}
```

Na Figura 57, observa-se a continuidade do teste de *software* realizado, na visualização da janela de saída da IDE, com as informações de que foram realizados 2 testes, com 5 verificações e um deles retornou erro, no caso, um erro falso positivo por não conseguir serializar o PDO para validação e verificação.

**FIGURA 57** - Janela de saída da execução do teste de *software* com PHPUnit

| Resultados do Teste | Saída - PHPUnit (portal_academico) X  |
|---------------------|---|
|                     | <pre>Time: 4 seconds, Memory: 2.75Mb  There was 1 error:  1) main_pTest::testUnitAndFF PDOException: You cannot serialize or unserialize PDO instances  FAILURES! Tests: 2, Assertions: 5, Errors: 1.  Concluido.</pre> |

Para finalizar os testes de *software*, foi realizado o teste funcional automatizado através da ferramenta *Badboy* versão 2.2.5, onde se obtém o retorno de resultados através dos procedimentos realizados de busca, visualização do arquivo, visualização dos relatórios, *login*, acesso à edição de arquivo e realizado alteração do tema do arquivo, *upload* de um novo arquivo e *logout* do sistema. A ferramenta gera alguns relatórios para acompanhamento do resultado obtido, como pode ser visto na Figura 58 e também na Figura 59.

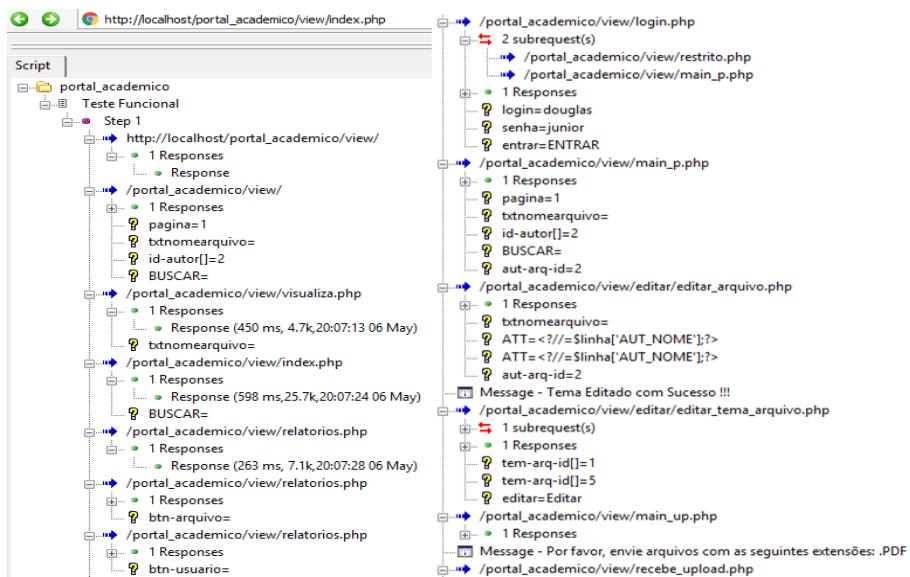
**FIGURA 58** - Relatório gerado pelo teste funcional automatizado com *Badboy* v.2.2.5

| Badboy Test Results<br>Report created 20:12 06 May 2016   |   |        |                            |  |              |          |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
|---|---|--------|----------------------------|--|--------------|----------|--|-------|------|------|-----|------|-----|-----|----|----|---|---|---|-----|------|----|----|---|---|---|-----|------|----|----|---|---|---|-----|------|
| Overview  |   |        |                            |  |              |          |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| Total Played  | Succeeded   | Failed | Warnings                   | Assertions   | Average Time | Max Time |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 16  | 16  | 0      | 0                          | 0  | 606          | 2256     |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| Success   |   |        |                            |  |              |          |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| <a href="#">Expand All</a><br>portal_academico .....,<br>Teste Funcional .....,<br>Step 1 ..... |   |        |                            | <table border="1"> <thead> <tr> <th>Total</th> <th>Succ</th> <th>Fail</th> <th>Wrm</th> <th>Asst</th> <th>Avg</th> <th>Max</th> </tr> </thead> <tbody> <tr> <td>16</td><td>16</td><td>0</td><td>0</td><td>0</td><td>606</td><td>2256</td></tr> <tr> <td>16</td><td>16</td><td>0</td><td>0</td><td>0</td><td>606</td><td>2256</td></tr> <tr> <td>16</td><td>16</td><td>0</td><td>0</td><td>0</td><td>606</td><td>2256</td></tr> </tbody> </table> |              |          |  | Total | Succ | Fail | Wrm | Asst | Avg | Max | 16 | 16 | 0 | 0 | 0 | 606 | 2256 | 16 | 16 | 0 | 0 | 0 | 606 | 2256 | 16 | 16 | 0 | 0 | 0 | 606 | 2256 |
| Total   | Succ  | Fail   | Wrm                        | Asst   | Avg          | Max      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 16  | 16  | 0      | 0                          | 0  | 606          | 2256     |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 16  | 16  | 0      | 0                          | 0  | 606          | 2256     |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 16  | 16  | 0      | 0                          | 0  | 606          | 2256     |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| All Responses   |   |        |                            |  |              |          |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| ID/Label  | URL / Reference   | Count  | Status (Success or Failed) | Avg Time   |              | Max Time |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 5   | http://localhost/portal_academico/view/                               | 1      | Success                    | .1   |              | 0        |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 47  | http://localhost/portal_academico/view/                               | 1      | Success                    | 441  |              | 441      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 67  | http://localhost/portal_academico/view/visualiza.php                  | 1      | Success                    | 450  |              | 450      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 73  | http://localhost/portal_academico/view/index.php                      | 1      | Success                    | 598  |              | 598      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 92  | http://localhost/portal_academico/view/relatorios.php                 | 1      | Success                    | 263  |              | 263      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 97  | http://localhost/portal_academico/view/relatorios.php                 | 1      | Success                    | 236  |              | 236      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 103   | http://localhost/portal_academico/view/relatorios.php                 | 1      | Success                    | 262  |              | 262      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 109   | http://localhost/portal_academico/view/index.php                      | 1      | Success                    | 342  |              | 342      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 116   | http://localhost/portal_academico/view/index+login.php                | 1      | Success                    | 764  |              | 764      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 127   | http://localhost/portal_academico/view/login.php                      | 1      | Success                    | 396  |              | 396      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 177   | http://localhost/portal_academico/view/main.php                       | 1      | Success                    | 466  |              | 466      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 199   | http://localhost/portal_academico/view/editar/editar_arquivo.php      | 1      | Success                    | 422  |              | 422      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 248   | http://localhost/portal_academico/view/editar/editar_tema_arquivo.php | 1      | Success                    | 1887   |              | 1887     |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 258   | http://localhost/portal_academico/view/main_up.php                    | 1      | Success                    | 494  |              | 494      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 529   | http://localhost/portal_academico/view/recebe_upload.php              | 1      | Success                    | 2256   |              | 2256     |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |
| 544   | http://localhost/portal_academico/view/logout.php                     | 1      | Success                    | 406  |              | 406      |  |       |      |      |     |      |     |     |    |    |   |   |   |     |      |    |    |   |   |   |     |      |    |    |   |   |   |     |      |

Na Figura 58 é possível observar que não houve nenhum tipo de falha, avisos ou alertas e nem asserções, desta forma o relatório gerado pela ferramenta com seu teste automatizado é capaz de informar que o fluxo demonstrado é satisfatório. A média de tempo de resposta das requisições realizadas foram de 606 milissegundos, o que é um tempo considerável satisfatório de espera para receber o resultado.

Na Figura 59 é possível observar detalhadamente o comportamento de algumas páginas em relação aos parâmetros e requisições que elas acessam, por exemplo, a página acessada, o botão pressionado, as requisições solicitadas e retornadas, os id's que são passados com as ações realizadas, mensagem de tratativa de erro, no caso foi realizado a tentativa de *upload* de arquivo que não seja pdf, onde se obteve retorno do aviso que só é permitido extensões pdf. Informações de usuário que realizou *login* no sistema e também *subrequests*, que são acessos a outras páginas .php para checagem ou validação para então poder encaminhar um redirecionamento ou aviso ao usuário da ação que ele está tentando executar.

**FIGURA 59** - Histórico do *script* com ações realizadas e parâmetros acessados via *Badboy*



Algo a se discutir e que está presente na documentação DERS, projetada em supostamente desenvolver o *software* e utilizar o padrão de arquitetura MVC (*Model, View, Controller*), da qual não se encontra aplicada neste projeto, onde, sua ausência foi mantida

devido à demanda de tempo para sua aplicação ser um fator extremamente relevante, principalmente para aplicações de pequeno porte, como é este projeto.

Na visão de Jobstraibizer (2009), uma das desvantagens do MVC é que o mesmo apesar de manter a organização separando os aspectos de controle, visão e regra de negócio, o número de classes presente na aplicação aumentaria de forma significativa.

Para cumprir o prazo proposto de entrega do projeto e garantir suas funcionalidades e obedecer as demais exigências de documentação, foi optado pela utilização da arquitetura procedural para desenvolvimento.

Jobstraibizer (2009) também deixa claro que um fator a ser levado em conta é o maior gasto de tempo para análise e modelagem do sistema, assim como também na organização e este tempo seria maior do que o próprio desenvolvimento.

## 5 CONCLUSÃO

No princípio, identificaram-se alguns problemas voltados a pouca quantidade de conteúdos em ambiente bibliotecário da universidade relacionados a variados temas ligados ao curso de Engenharia de *Software*, precisando assim sempre recorrer a fontes de pesquisa que às vezes podem não ser confiáveis ou não possuir base científica sobre o mesmo assunto, assim como também há limitações quando encontrado algum conteúdo de estudo em ambiente bibliotecário. Por se tratar de conteúdos físicos, como livros, artigos e monografias por exemplo, são materiais que se degradam com o tempo, com o uso indevido, podem ser facilmente danificados sendo por acidentes ou não, em alguns ambientes é até difícil de manter a organização principalmente se forem em grande volume, pois geralmente são ambientes públicos que recebem pessoas a qualquer momento e que podem retirar os itens das devidas localidades com frequência. Outra questão interessante de ser observada foi a acessibilidade ao conteúdo, onde no meio físico, o interessado é preciso ir até o local para buscar o conteúdo de seu interesse, podendo este local estar a uma distância relevante. Com a intenção de sanar essas dificuldades encontradas, surgiu a hipótese de desenvolver um portal de conteúdos e acervos acadêmicos voltado especificamente para o curso de engenharia de *software*, este portal seria uma aplicação *web* que teria como objetivo principal armazenar conteúdos de interesse do curso através dos docentes do mesmo.

Contudo, tem-se uma disseminação de informação bastante flexível e alcançável para grande parte do público que abrange o curso, pois dependeria apenas de conexão com a internet em um computador para utilização da ferramenta, que são instrumentos hoje em dia essenciais para atividades de trabalho, lazer, estudo e outros fins.

O docente se sentirá confortável ao saber que as informações contidas ali no portal acadêmico seriam facilmente encontradas e acessadas, e que permite a ele que disponibilizasse conteúdo em formato .pdf para que os discentes accessem e possam realizar estudo e obter conhecimento para as disciplinas ou temas de seu interesse. A facilidade é encontrada devido aos filtros de busca que o sistema disponibiliza, onde é possível buscar em sete campos diferentes de informações, sendo elas: nome do arquivo, editora, tema, disciplina, tipo de arquivo, palavra chave ou autor.

O sistema pode ser acessado sem necessidade de cadastro de usuário, onde esta funcionalidade existe e só é necessária em caso de publicação de conteúdo, ou seja, para o

docente ou administrador. Para o usuário final que deseja apenas buscar conteúdo para aprendizado, não há necessidade de cadastro, descomplicando ainda mais para o usuário e permitindo ele a um acesso direto ao sistema.

Os *layouts* das páginas iniciais do sistema seguem um certo padrão com tabelas para busca e também para o resultado da mesma, as informações estão dentro de cada item indicado através das colunas das tabelas, a usabilidade das telas dependem apenas de uma básica leitura dos elementos dispostos na página para ser indicado onde realizar determinadas ações do sistema, não exigindo assim nenhum conhecimento específico para manipulação do sistema.

Para publicar conteúdo no sistema, o docente ou administrador que efetuar *login* no sistema não terá dificuldades, há uma página específica responsável por realizar *upload* dos arquivos no sistema com todas tratativas de possíveis erros ou ausência de informação obrigatória para que o arquivo seja publicado, desta forma, pode-se dizer que o sistema possui uma usabilidade bastante simples e este é o intuito do projeto desde o início.

No período de desenvolvimento do sistema, houve grandes dificuldades que foram enfrentadas e superadas, porém, algumas modificações foram necessárias como a remoção de dois requisitos funcionais, que para serem utilizados seria preciso adicionar ao usuário discente um cadastro e necessidade de *login*, o que fez por tomada de decisão a remoção destes requisitos funcionais, pois seria mais intuitivo o sistema com um acesso direto. Os filtros de busca sem dúvida é um diferencial interessante do projeto e possui recursos que foram inseridos graças ao auxílio de alguns *frameworks* disponíveis para o ambiente *web*.

Futuras possíveis implementações ou perspectiva de trabalhos futuros podem ser analisadas como melhoria no sistema, o vínculo entre filtros de busca, onde determinado autor selecionado, os demais filtros apresentados são apenas ao daquele autor, ou editora e assim por diante, como também manter um controle maior na publicação dos arquivos, mantendo quem seria o usuário que o publicou, por exemplo, o que poderia auxiliar como um novo campo de busca para os arquivos.

Com a visão destas perspectivas para trabalhos futuros tornar o projeto mais robusto e melhor em alguns aspectos, a implementação concluída cumpre o objetivo que foi especificado inicialmente nas documentações. O esforço presente na elaboração deste projeto foi necessário substituir momentos de lazer e entretenimento em horas de estudos e pesquisas para formar resultados capazes de cumprir o projeto proposto no prazo especificado.

## REFERÊNCIAS

- BRUNO, O. M. *Programando para a internet com PHP*. Rio de Janeiro: Brasport, 2010.
- EASTMAN, C. et al. *BIM: um guia de modelagem da informação da construção para arquitetos, engenheiros, gerentes, construtores e incorporadores*. Porto Alegre: John Wiley & Sons, 2008.
- FRANCISCON, M. J. *A importância do registro digital das ocorrências policiais-militares: as funções do registro de ocorrências*. São Paulo: Edição do autor, 2013.
- GEBRAN, M. P. *Tecnologias educacionais*. Curitiba: IESDE, 2009.
- HORSTMANN, C. *Padrões e projeto orientados a objetos*. 2.ed. Porto Alegre: John Wiley & Sons, 2006.
- JOBSTRAIBIZER, F. *Guia profissional PHP*. São Paulo. Digerati Books, 2009.
- LIMA, A. da S. *UML 2.5 Do requisito à solução*. São Paulo: Érica, 2014.
- LOBO, E. J. R. *Curso prático de MySQL*. São Paulo: Digerati Books, 2008.
- MARTINS, J. C. C. *Técnicas para gerenciamento de projetos de Software*. Rio de Janeiro: Brasport, 2007.
- MASON, T.; WEEKS, J. *Museologia: relatórios técnicos - Planos para a Certificação de Museus na Grã-Bretanha: Padrões. Da Austrália a Zanzibar: Planos de Certificação de Museus em Diversos Países*. Tradução de Maurício O. Santos e Patrícia Souza. São Paulo. Edusp, 2004.
- MENEZES, E. D. *Curso de direito autoral*. Belo Horizonte. Del Rey, 2007.
- MUTO, C. A. *PHP & MySQL: guia introdutório*. 3.ed. Rio de Janeiro. Brasport, 2006.
- OLIVEIRA, M. (Coord.). *Ciência da informação e biblioteconomia: novos conteúdos e espaços de atuação*. Belo Horizonte: UFMG, 2005.
- PÉRICAS, F. A. *Redes de computadores: conceitos e a arquitetura internet*. 3.ed. Blumenau: Edição do autor, 2012.
- PINHO, J. B. *Relações Públicas na internet: técnicas e estratégias para informar e influenciar públicos de interesse*. São Paulo: Summus Editorial, 2002.
- QUIERELLI, D. A. *Criando sites com HTML-CSS-PHP: construindo um projeto – iniciante*. Leme. Edição do Autor, 2012.
- RAMALHO, J. A. A. *Curso completo para desenvolvedores Web*. Rio de Janeiro: Elsevier, 2005.

REMOALDO, P. *O guia prático do Dreamweaver CS3: com PHP, JAVASCRIPT E AJAX.* Lisboa – Portugal: Centro Atlântico, 2008.

ROCHA, D. da S. *Direito de autor.* São Paulo: Irmãos Vitale, 2001.

SAMPAIO, C. *Guia do Java Enterprise Edition 5: desenvolvendo aplicações corporativas.* 5.ed. Rio de Janeiro: Brasport, 2007.

SCHACH, S. R. *Engenharia de software: os paradigmas clássico & orientado a objetos.* 7.ed. Porto Alegre: AMGH, 2010.

SERRA, L. G. *Livro digital e bibliotecas.* Rio de Janeiro: FGV Editora, 2014.

SETZER, V. W. *Bancos de dados: aprenda o que são; melhore seu conhecimento; construa os seus.* São Paulo: Edgar Blücher, 2005.

**ANEXOS**

## ANEXO A - Documento de Especificação de Requisitos de Software (DERS)

### 1 - O PROPÓSITO DO PROJETO

#### 1a. O Negócio do Usuário ou o Conhecimento do Esforço do Projeto

Ter acesso a bons conteúdos sobre assuntos relacionados ao curso de interesse, com a intenção de fortalecer o conhecimento sobre diversos temas de nosso amplo mercado de trabalho, conhecer e manter-se informado. Resumidamente, é estar por dentro do assunto abordado, seja ele qual for, ciente de que está no caminho certo. A dificuldade de acesso é o principal motivo para investir neste projeto.

#### 1b. Metas do Projeto

Agilizar a forma de acessar conteúdos de interesse acadêmico, no caso, do próprio curso e também melhorar a viabilidade do mesmo, publicar também resultados de pesquisas e estudos próprios, criando uma base de conhecimento cada vez mais forte.

Hoje em dia, tempo está sendo algo primordial e muito valioso na vida de todos e na maioria das vezes não permite que se disponha parte dele para algumas práticas importantes, só de pensar em todo processo que deverá ser feito para encontrar, pesquisar, estudar e entender do assunto, desmotiva-se de exercer esta prática de leitura para enriquecer de informação, este sim, é um sério problema que as pessoas enfrentam no dia a dia e este projeto irá auxiliar na trajetória de um caminho mais amplo para que possa incentivar esta prática atualmente.

### 2 – OS INTERESSADOS

#### 2a. O Cliente

O projeto é inicialmente específico e exclusivo para o curso de Engenharia de Software da Universidade de Rio Verde, podendo se expandir dependendo de seu resultado final. Mas a intenção do projeto é atingir todos os envolvidos deste curso, criando um acervo acadêmico rico com o passar dos dias e ajudando a desenvolver profissionais e talentos nesta incrível área crescendo em valorização em nossa região.

#### 2b. O Consumidor

Os consumidores do projeto, ou seja, os usuários, são todos os discentes e docentes do

curso Engenharia de Software da UniRV. Diretamente ou indiretamente, todos estarão relacionados ao projeto, são eles então, os *stakeholders*<sup>8</sup>, através de um questionário de pesquisa, obtemos algumas informações e através da necessidade própria como aluno cliente e visão de melhoria dos estudos, surge o interesse pelo projeto.

## 2c. Outros Interessados

Editoras interessadas em divulgar seus conteúdos relacionados ao curso.

## 2d. Mãos à Obra no Produto

- Coordenador de Engenharia de Software da UniRV
- Discentes de Engenharia de Software da UniRV
- Docentes de Engenharia de Software da UniRV

Habilidades relacionadas à tecnologia

## 2e. Prioridades Atribuídas aos Usuários

- **Coordenador(es):** Maior nível de controle do sistema, podendo realizar download, visualizar, pesquisar, incluir e excluir conteúdo dentro do portal. Responsável por todo conteúdo do portal e cabe a ele decisão de manter ou remover.
- **Docentes:** Segundo maior nível de controle do sistema, podendo realizar download, visualizar, pesquisar e incluir conteúdo dentro do portal. Responsável por aquilo que ele próprio publica, ciente e de acordo com os termos, após publicado, só poderá ser alterado solicitando ao coordenador.
- **Discentes:** Nível de acesso ao sistema, podendo realizar download, visualizar e pesquisar os conteúdos contidos no portal, não arcando com nenhuma responsabilidade relacionada diretamente com o sistema, usuário direto dos serviços que o portal lhe oferece.

## 2f. Participação do Usuários

É exigido pelos *stakeholders* em geral, uma usabilidade agradável, facilidade e interface amigável e com os *feedbacks*<sup>9</sup> que ambos os interessados no projeto vão reunindo e repassando, tornará possível elaborar em cima da documentação uma melhoria para o resultado final, além da participação com o conhecimento das regras de negócio.

---

<sup>8</sup> São todos os interessados diretamente ou indiretamente no projeto.

<sup>9</sup> É o retorno obtido de informação por algum meio, geralmente utilizado para avaliar algum resultado.

## **2g. Usuários de Manutenção e Técnicos de Serviço**

Não haverá uma pessoa jurídica responsável por trás do software, fornecendo atualizações, correções e outros serviços. O Software ficará sobre guarda da Coordenação do curso de Engenharia de Software da Universidade de Rio Verde e seus docentes, cabendo a eles a sustentabilidade do sistema após ser entregue com suas funcionalidades essenciais atendendo as necessidades.

## **3 - RESTRIÇÕES OBRIGATÓRIAS**

### **3a. Restrições de Soluções**

Descrição: O produto deve operar no navegador Chrome e Firefox.

Razão: Navegadores homologados para utilização do serviço.

Descrição: O produto deve conter legalidade totalmente open-source

Razão: A sustentabilidade do software não deverá depender de licenças para qualquer finalidade.

Descrição: O produto deve estar no padrão de projeto MVC (*Model-View-Control*).

Razão: Melhores práticas e facilidade em manutenção ou implementações futuras.

### **3b. Ambiente de Implementação do Sistema Atual**

- Plataforma – Web;
- Gerenciador de Banco de Dados MySQL;
- Gerenciador de Projeto – Gantt Project;
- Ferramenta de Teste – RIPS;
- Ferramenta de Teste 2 – PHPUnit;
- Ferramenta de Teste 3 – BadBoy;
- Diagramas (UML) – Astah;
- Editor de texto com suporte HTML / CSS/ JavaScript – Brackets;
- Servidor Web – Apache;

- IDE de Desenvolvimento – NetBeans;

Sujeito a alterações mediante necessidade do sistema ou conhecimento de novos ambientes utilitários.

## **4 - NOMEANDO CONVENÇÕES E DEFINIÇÕES**

### **4a. Definições de Todos os Termos, Incluindo Acrônimos, Usados no Projeto**

MVC: Padrão de projeto de desenvolvimento de software com 3 camadas. (Modelo, Visão, Controle).

Open-Source: Arquitetura com código de programação livre para todos os usuários.

## **5 - FATOS E SUPOSIÇÕES RELEVANTES**

### **5a. Fatos Relevantes**

Dependendo da extensão do conteúdo e do tempo gasto em cima dele, é preciso estar ciente dos riscos de saúde caso o mesmo seja utilizado sem uma pausa, pois pode ser prejudicial a visão através do monitor. Neste caso, nada que uma impressão não resolva.

Precisa-se estar ciente, ser objetivo e focado, pois o portal deve ser acessado pela internet, ela também pode ser um desvio do tempo do usuário para outras distrações, esquecendo que seu principal objetivo são os conteúdos de estudo.

### **5b. Regras do Negócio**

- O autor da publicação do conteúdo é o seu próprio responsável.
- Somente docentes podem realizar publicação de conteúdo no portal.
- Somente o coordenador pode excluir publicações, mediante solicitação do autor.
- Publicações favoritadas, para serem removidas sofrem uma semana em aguardo.
- Discentes não possuem nenhum tipo de controle sobre as publicações.
- O sistema permite visualização e download dos arquivos para qualquer usuário.
- O sistema é exclusivo de uso dos discentes e docentes da faculdade de Engenharia de Software da Universidade de Rio Verde.
- O sistema deverá estar disponível 24 horas por dia, 7 dias por semana.

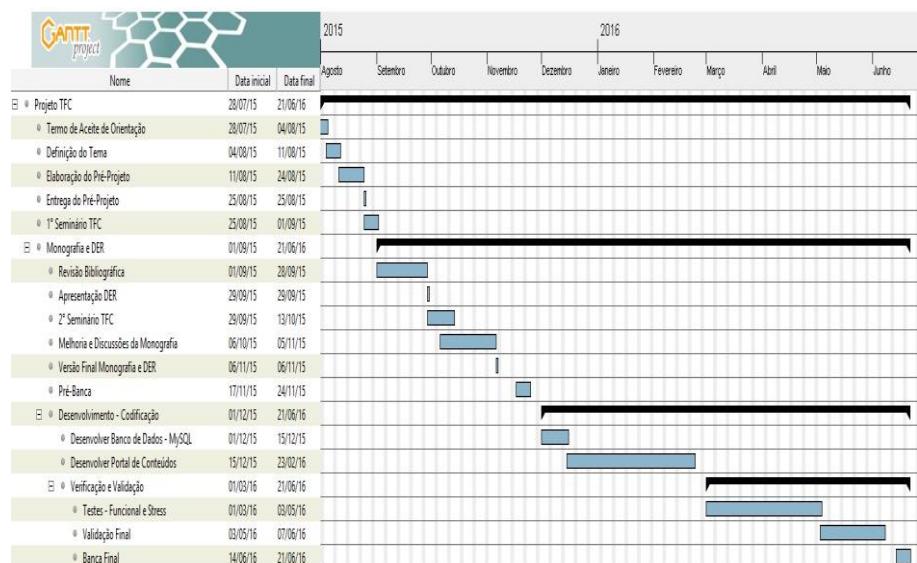
- A Faculdade de Engenharia de Software é a responsável pelo armazenamento dos arquivos publicados, estes, em seus servidores.

## 6 - ESTIMATIVA DE CUSTOS E PRAZOS PARA A ELABORAÇÃO

### 6a. Estimativa de prazo para elaboração

Segue abaixo na Figura 1, o cronograma de atividades a serem realizadas no projeto.

**FIGURA 1** - Cronograma do calendário de tarefas, elaborado com GanttProject



## 7 - ESCOPO DO TRABALHO

### 7a. A Situação Atual

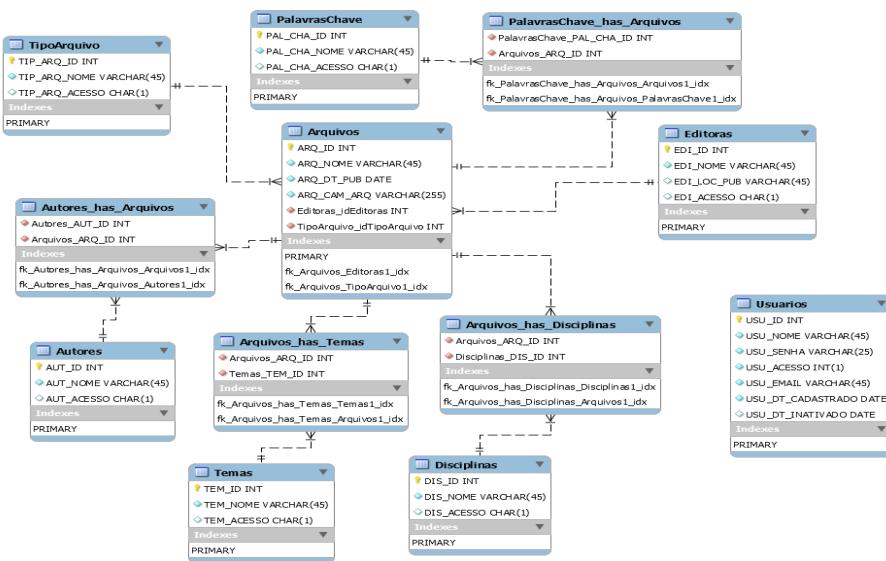
Atualmente, os discentes e docentes não contam com nenhum tipo de portal de conteúdo específico e muito menos serviço de acervo digital ou algo semelhante, com disponibilidade a qualquer dia, em qualquer hora e qualquer necessidade de busca de arquivos, é através de pesquisas na internet ou nas bibliotecas mais próximas que irá se deparar com a situação atual e também a pesquisa de campo assim que houver prototipação do projeto para melhor análise e argumentos.

## **8 - ESPECIFICAÇÃO TÉCNICA**

## 8a. Modelo dos Dados

Segue na Figura 2 abaixo, o modelo das tabelas do banco de dados com os campos e seus respectivos relacionamentos.

**FIGURA 2** - Modelo de entidade e relacionamento do banco de dados



## 8b. Dicionário de Dados

**TABELA 1** - Entidade do banco de dados denominada arquivos

| Atributo                  | Classe       | Domínio  | Tamanho | Descrição          |
|---------------------------|--------------|----------|---------|--------------------|
| ARQ_ID                    | Determinante | Numérico |         | Chave da Entidade  |
| ARQ_NOME                  | Simples      | Texto    | 45      | Nome do Item       |
| ARQ_DT_PUB                | Simples      | Data     |         | Formato dd/mm/aaaa |
| ARQ_CAM_ARQ               | Simples      | Texto    | 255     | Caminho do Arquivo |
| Editoras_idEditoras       | Determinante | Numérico |         | FK                 |
| TipoArquivo_idTipoArquivo | Determinante | Numérico |         | FK                 |

**TABELA 2** - Entidade do banco de dados denominada arquivos\_has\_temas

| Atributo        | Classe       | Domínio  | Tamanho | Descrição              |
|-----------------|--------------|----------|---------|------------------------|
| Arquivos_ARQ_ID | Determinante | Numérico |         | Chave da Entidade (fk) |
| Temas_TEM_ID    | Determinante | Numérico |         | Chave da Entidade (fk) |

**TABELA 3** - Entidade do banco de dados denominada autores

| Atributo   | Classe       | Domínio  | Tamanho | Descrição         |
|------------|--------------|----------|---------|-------------------|
| AUT_ID     | Determinante | Numérico |         | Chave da Entidade |
| AUT_NOME   | Simples      | Texto    | 45      | Nome do Item      |
| AUT_ACESSO | Simples      | Char     | 1       | Status do Autor   |

**TABELA 4** - Entidade do banco de dados denominada autores\_autores\_has\_arquivos

| Atributo        | Classe       | Domínio  | Tamanho | Descrição              |
|-----------------|--------------|----------|---------|------------------------|
| Autores_AUT_ID  | Determinante | Numérico |         | Chave da Entidade (fk) |
| Arquivos_ARQ_ID | Determinante | Numérico |         | Chave da Entidade (fk) |

**TABELA 5** - Entidade do banco de dados denominada arquivos\_has\_disciplinas

| Atributo           | Classe       | Domínio  | Tamanho | Descrição              |
|--------------------|--------------|----------|---------|------------------------|
| Arquivos_ARQ_ID    | Determinante | Numérico |         | Chave da Entidade (fk) |
| Disciplinas_DIS_ID | Determinante | Numérico |         | Chave da Entidade (fk) |

**TABELA 6** - Entidade do banco de dados denominada disciplinas

| Atributo   | Classe       | Domínio  | Tamanho | Descrição            |
|------------|--------------|----------|---------|----------------------|
| DIS_ID     | Determinante | Numérico |         | Chave da Entidade    |
| DIS_NOME   | Simples      | Texto    | 45      | Nome do Item         |
| DIS_ACESSO | Simples      | Char     | 1       | Status da Disciplina |

**TABELA 2** - Entidade do banco de dados denominada editoras

| Atributo    | Classe       | Domínio  | Tamanho | Descrição                   |
|-------------|--------------|----------|---------|-----------------------------|
| EDI_ID      | Determinante | Numérico |         | Chave da Entidade           |
| EDI_NOME    | Simples      | Texto    | 45      | Nome do Item                |
| EDI_LOC_PUB | Simples      | Texto    | 45      | Local da Publicação do Item |
| EDI_ACESSO  | Simples      | Char     | 1       | Status da Editora           |

**TABELA 3** - Entidade do banco de dados denominada palavras-chave.

| Atributo       | Classe       | Domínio  | Tamanho | Descrição               |
|----------------|--------------|----------|---------|-------------------------|
| PAL_CHA_ID     | Determinante | Numérico |         | Chave da Entidade       |
| PAL_CHA_NOME   | Simples      | Texto    | 45      | Nome do Item            |
| PAL_CHA_ACESSO | Simples      | Char     | 1       | Status da Palavra Chave |

**TABELA 4** - Entidade do banco de dados denominada palavraschave\_has\_arquivos

| Atributo                 | Classe       | Domínio  | Tamanho | Descrição                 |
|--------------------------|--------------|----------|---------|---------------------------|
| Arquivos_ARQ_ID          | Determinante | Numérico |         | Chave da Entidade<br>(fk) |
| PalavrasChave_PAL_CHA_ID | Determinante | Numérico |         | Chave da Entidade<br>(fk) |

**TABELA 5** - Entidade do banco de dados denominada temas

| Atributo   | Classe       | Domínio  | Tamanho | Descrição         |
|------------|--------------|----------|---------|-------------------|
| TEM_ID     | Determinante | Numérico |         | Chave da Entidade |
| TEM_NOME   | Simples      | Texto    | 45      | Nome do Item      |
| TEM_ACESSO | Simples      | Char     | 1       | Status do Tema    |

**TABELA 6** - Entidade do banco de dados denominada tipo arquivo

| Atributo       | Classe       | Domínio  | Tamanho | Descrição                 |
|----------------|--------------|----------|---------|---------------------------|
| TIP_ARQ_ID     | Determinante | Numérico |         | Chave da Entidade         |
| TIP_ARQ_NOME   | Simples      | Texto    | 45      | Nome do Item              |
| TIP_ARQ_ACESSO | Simples      | Char     | 1       | Status do Tipo de Arquivo |

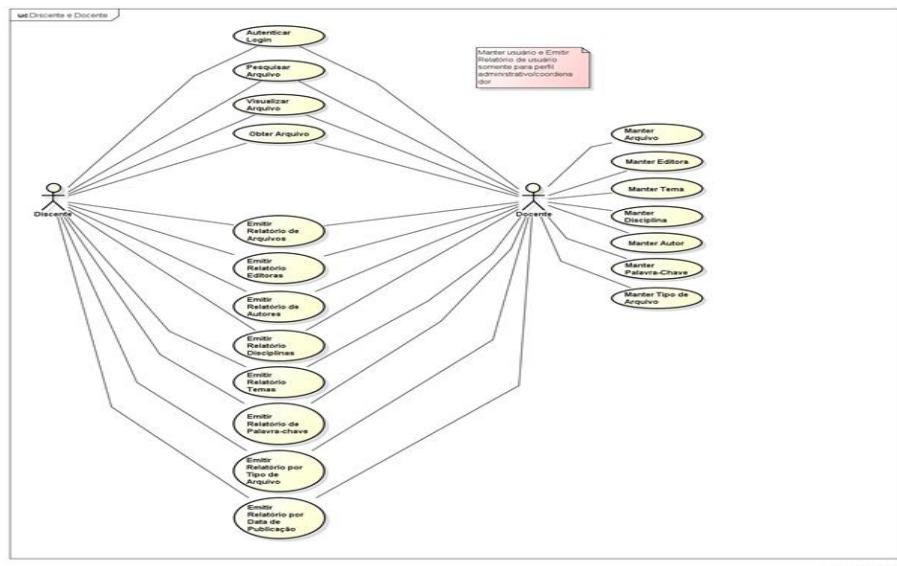
**TABELA 7** - Entidade do banco de dados denominada usuários.

| Atributo          | Classe       | Domínio  | Tamanho | Descrição          |
|-------------------|--------------|----------|---------|--------------------|
| USU_ID            | Determinante | Numérico |         | Chave da Entidade  |
| USU_LOGIN         | Simples      | Texto    | 45      | Nome do Item       |
| USU_SENHA         | Simples      | Texto    | 25      | Senha do Usuario   |
| USU_ACESSO        | Simples      | Inteiro  | 1       | Nível de Acesso    |
| USU_EMAIL         | Simples      | Texto    | 45      | Email de Contato   |
| USU_DT_CADASTRADO | Simples      | Data     |         | Formato dd/mm/aaaa |

### 8c. Fronteiras do Produto

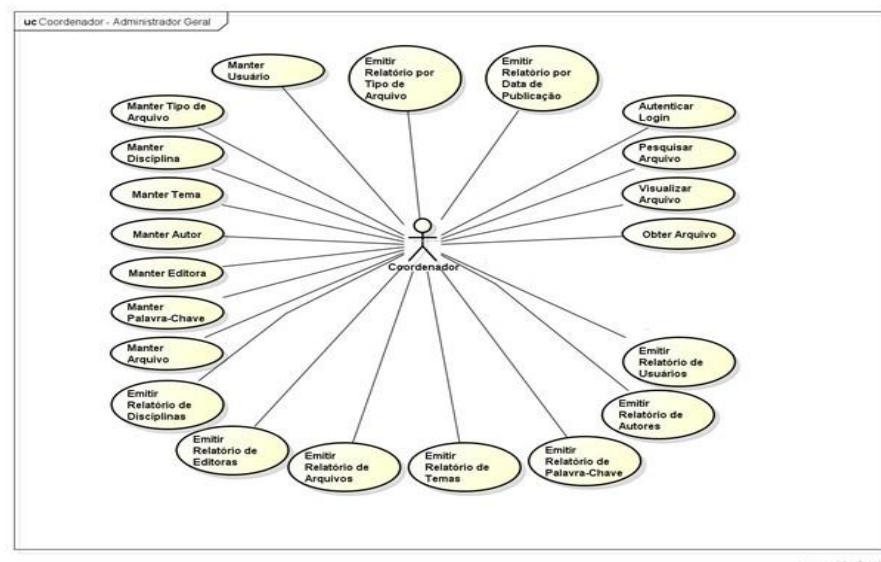
Segue abaixo na Figura 3 o diagrama de caso de uso voltado para os atores discentes e docentes.

**FIGURA 3 - Diagrama de caso de uso para os discentes e docentes**



Segue abaixo na Figura 4 o diagrama de caso de uso voltado para o usuário administrador geral ou coordenador.

**FIGURA 4 - Diagrama de caso de uso para o coordenador ou administrador geral**



#### 8d. Listas de Casos de Uso

1 – Autenticar Login

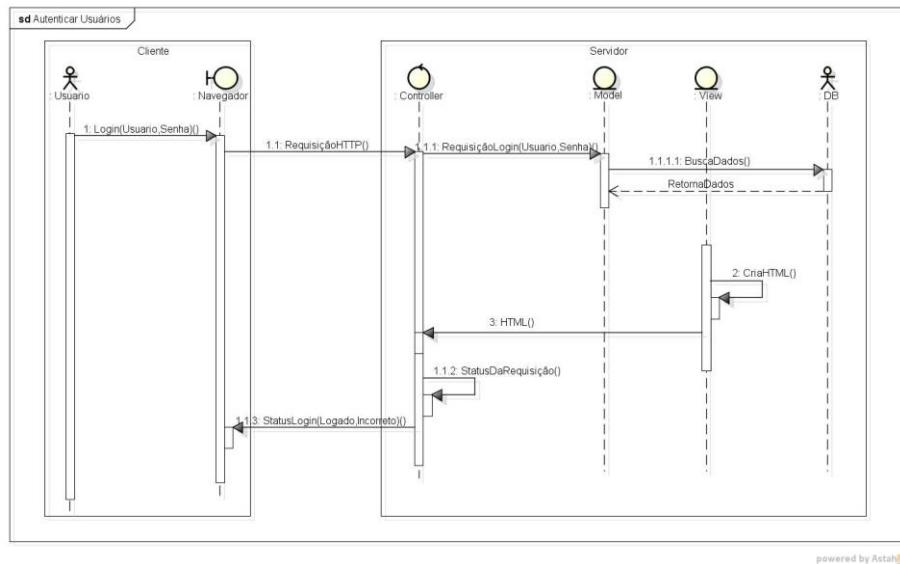
2 – Manter Usuário

- 3 – Manter Tema
- 4 – Manter Autor
- 5 – Manter Editora
- 6 – Manter Disciplina
- 7 – Manter Palavra-Chave
- 8 – Manter Tipo de Arquivo
- 9 – Manter Arquivo
- 10 – Pesquisar Arquivo
- 11 – Visualizar Arquivo
- 12 – Obter Arquivo
- 13 – Emitir Relatório de Usuários
- 14 – Emitir Relatório de Disciplinas
- 15 – Emitir Relatório de Autores
- 16 – Emitir Relatório de Temas
- 17 – Emitir Relatório de Editoras
- 18 – Emitir Relatório de Palavra-Chave
- 19 – Emitir Relatório de Arquivos
- 20 – Emitir Relatório por Tipo de Arquivo
- 21 – Emitir Relatório por Data de Publicação

#### **8e. Sequências do Produto**

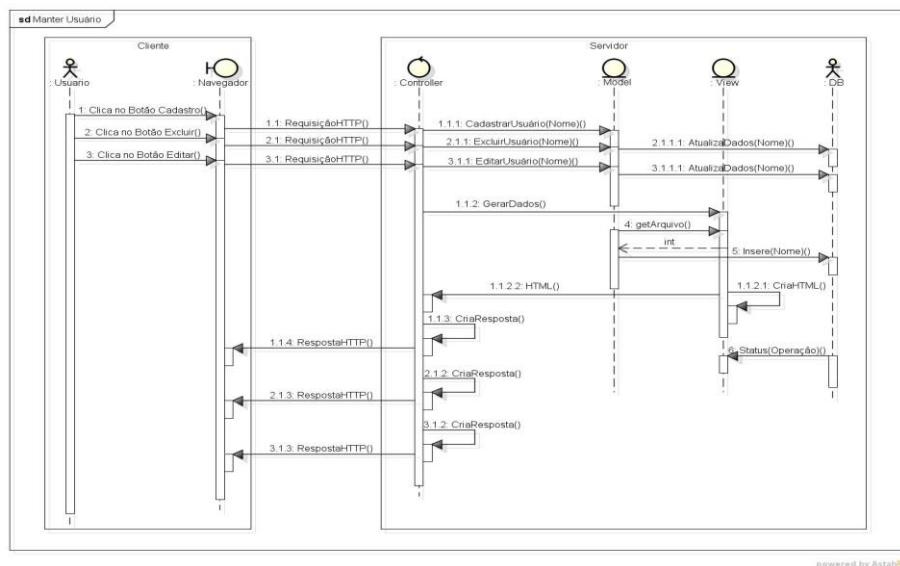
Segue abaixo na Figura 5 o diagrama de sequência voltado para autenticação de usuários.

**FIGURA 5 - Diagrama de sequência para autenticar usuários**



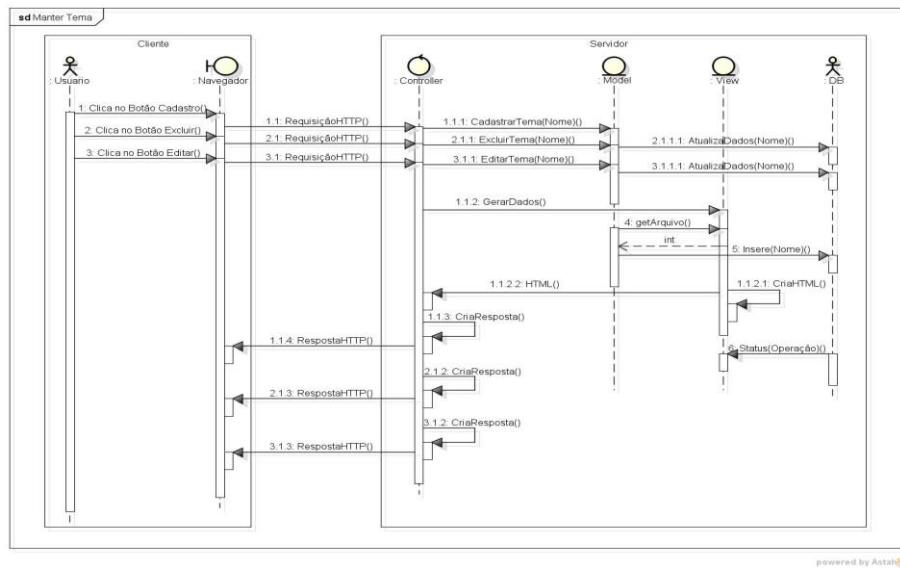
Segue abaixo na Figura 6 o diagrama de sequência voltado para manter usuário, ou seja, cadastro, edição e exclusão de usuários.

**FIGURA 6 - Diagrama de sequência para manter usuário**



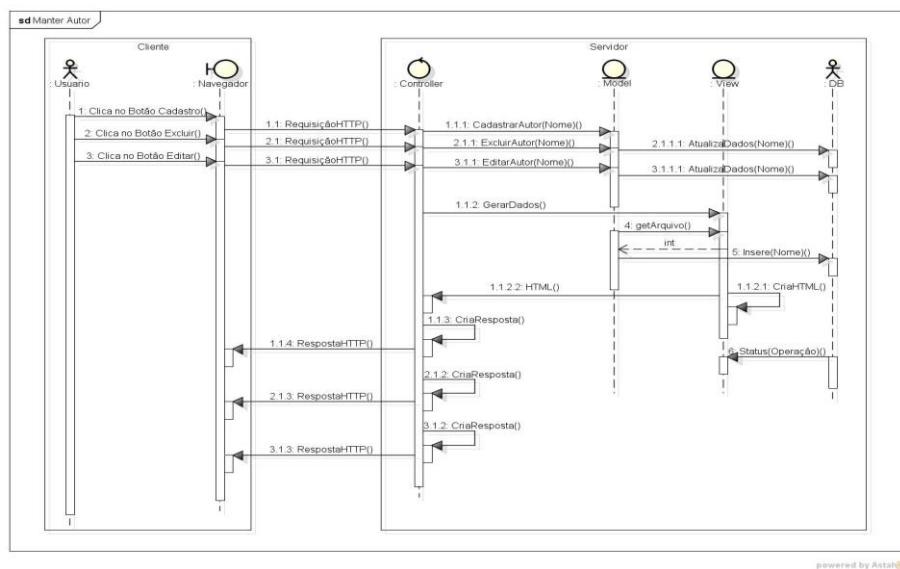
Segue abaixo na Figura 7 o diagrama de sequência para manter tema, ou seja, cadastro, edição e exclusão de temas.

**FIGURA 7 - Diagrama de sequência para manter tema**



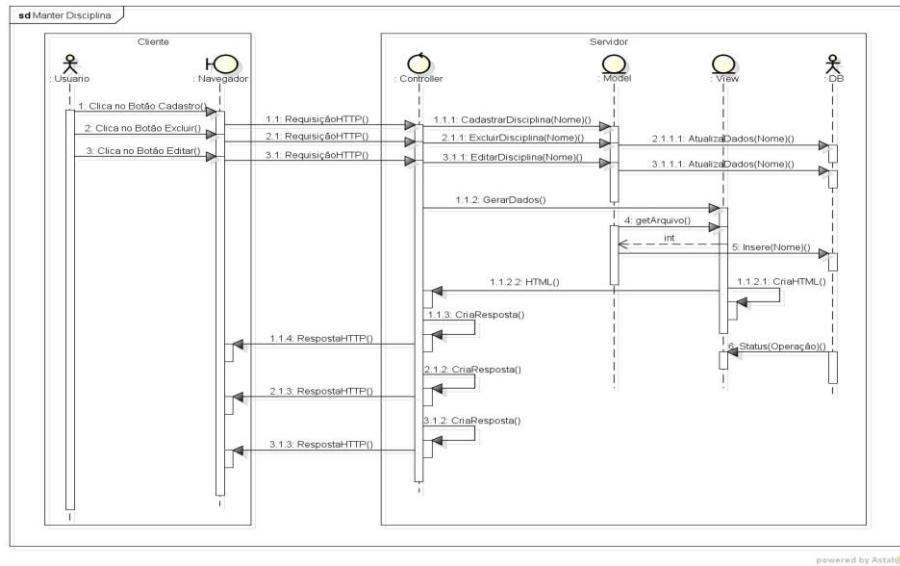
Segue abaixo na Figura 8 o diagrama de sequência voltado para manter autor, ou seja, cadastro, edição e exclusão de autores.

**FIGURA 8 - Diagrama de sequência para manter autor**



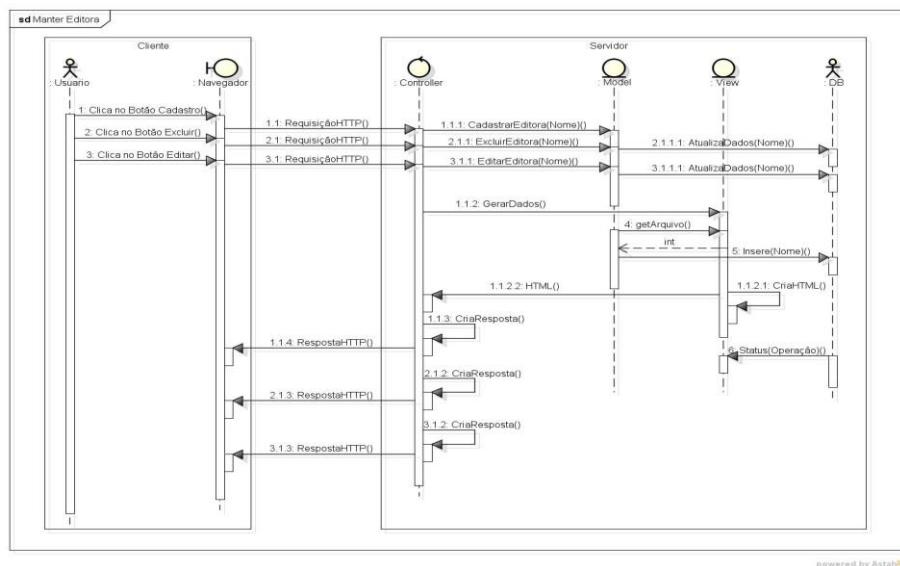
Segue abaixo na Figura 9 o diagrama de sequência voltado para manter disciplina, ou seja, cadastro, edição e exclusão de disciplinas.

**FIGURA 9** - Diagrama de sequência para manter a disciplina



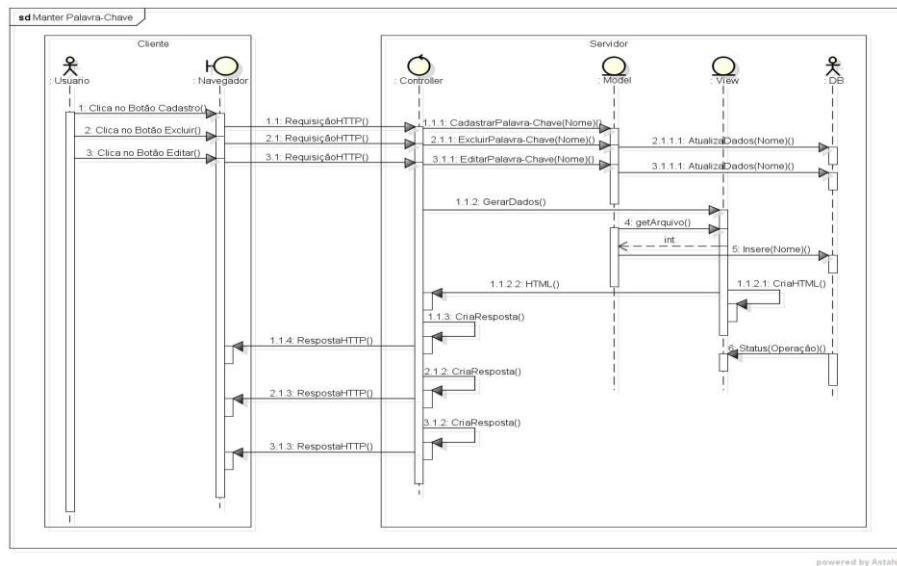
Segue abaixo na Figura 10 o diagrama de sequência voltado para manter editora, ou seja, cadastro, edição e exclusão de editoras.

**FIGURA 10** - Diagrama de sequência para manter editora



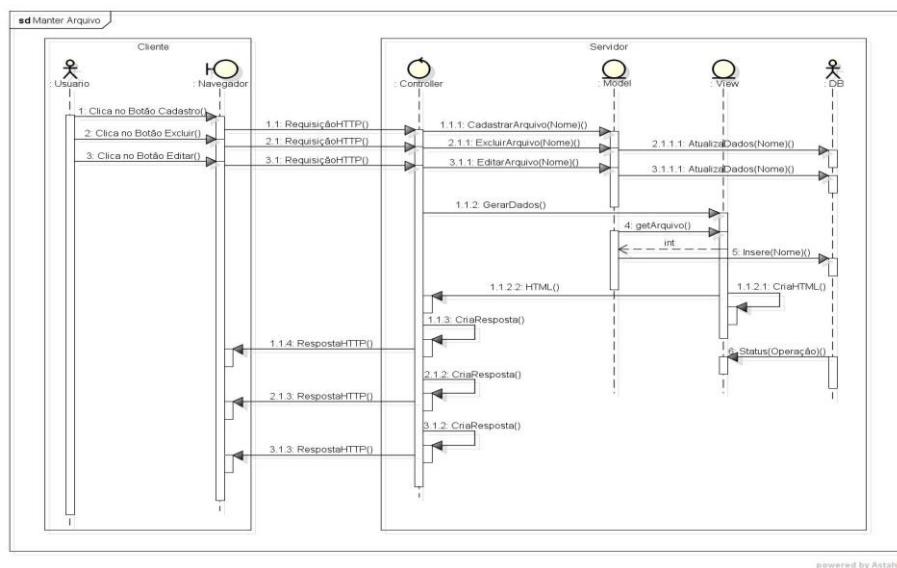
Segue abaixo na Figura 11 o diagrama de sequência voltado para manter palavra-chave, ou seja, cadastro, edição e exclusão de palavras-chave.

**FIGURA 11 - Diagrama de sequência para manter palavra-chave**



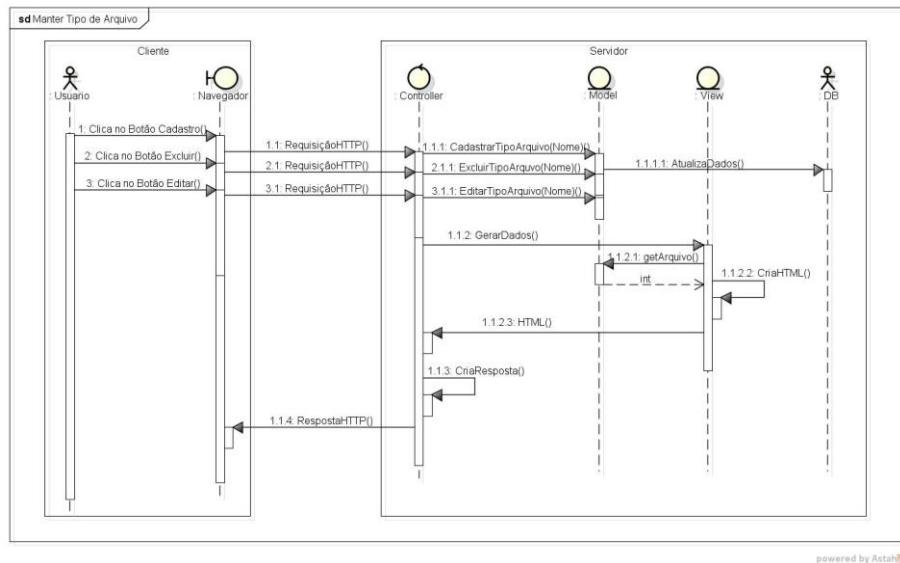
Segue abaixo na Figura 12 o diagrama de sequência voltado para manter arquivo, ou seja, cadastro, edição e exclusão de arquivos.

**FIGURA 12 - Diagrama de sequência para manter arquivo**



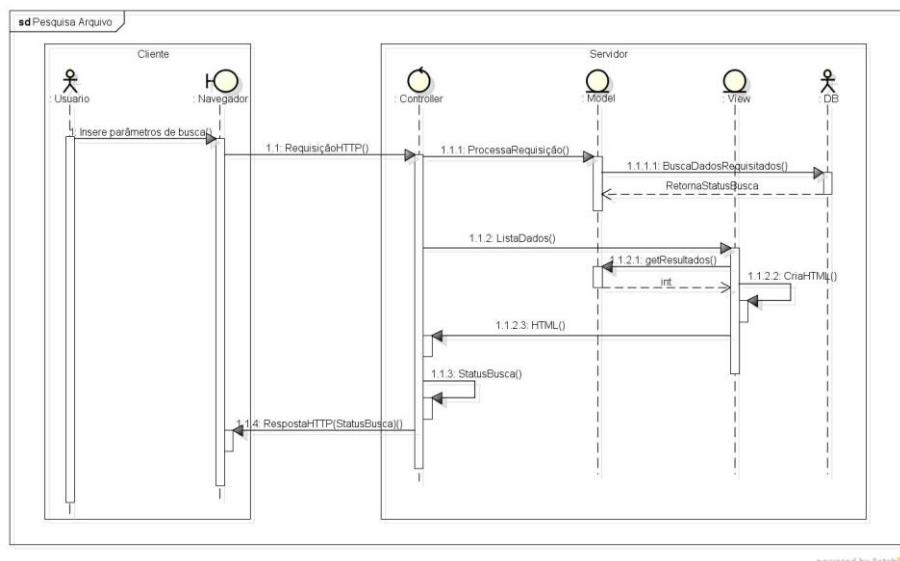
Segue abaixo na Figura 13 o diagrama de sequência voltado para manter tipo de arquivo, ou seja, cadastro, edição e exclusão de tipos de arquivos.

**FIGURA 13 - Diagrama de sequência para manter tipo de arquivo**



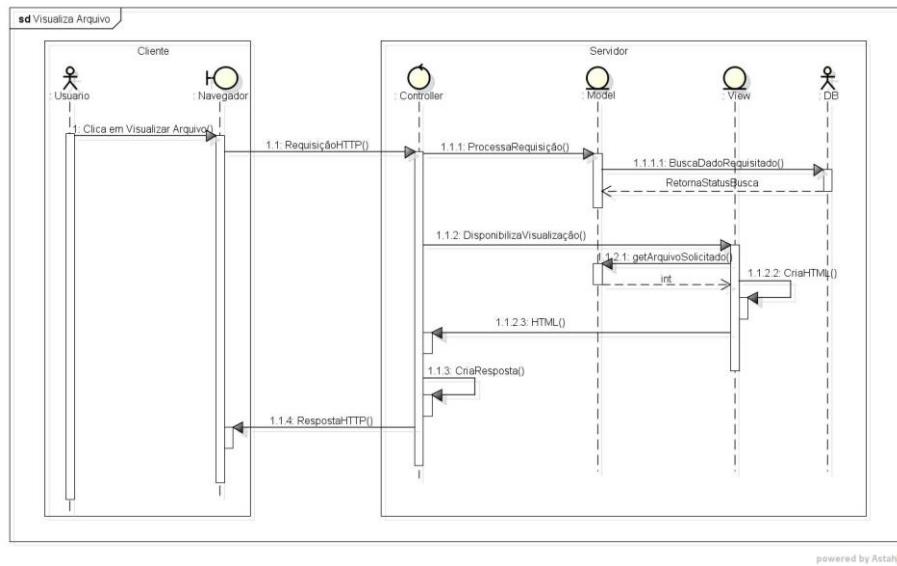
Segue abaixo na Figura 14 o diagrama de sequência voltado para pesquisar arquivo.

**FIGURA 14 - Diagrama de sequência para pesquisar arquivo**



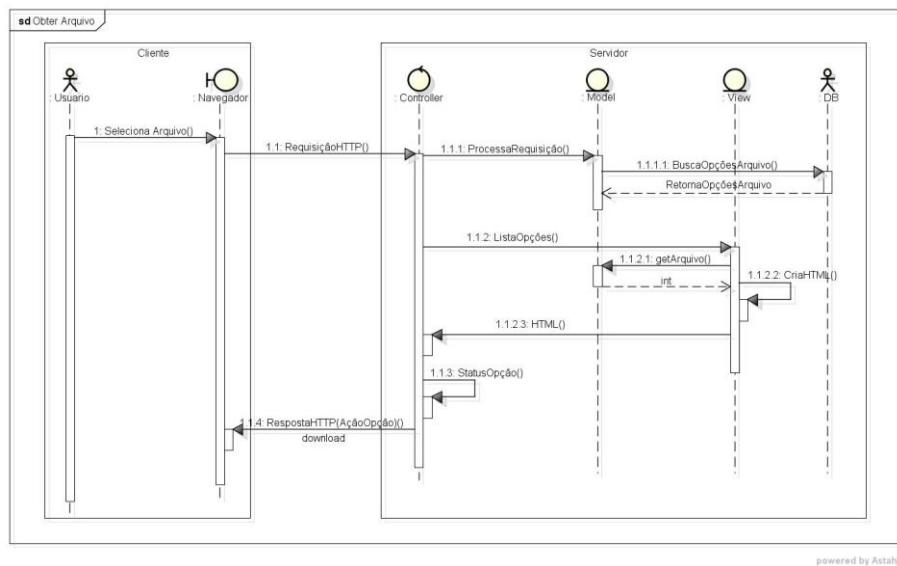
Segue abaixo na Figura 15 o diagrama de sequência voltado para visualizar arquivo.

**FIGURA 15 - Diagrama de sequência para visualizar arquivo**



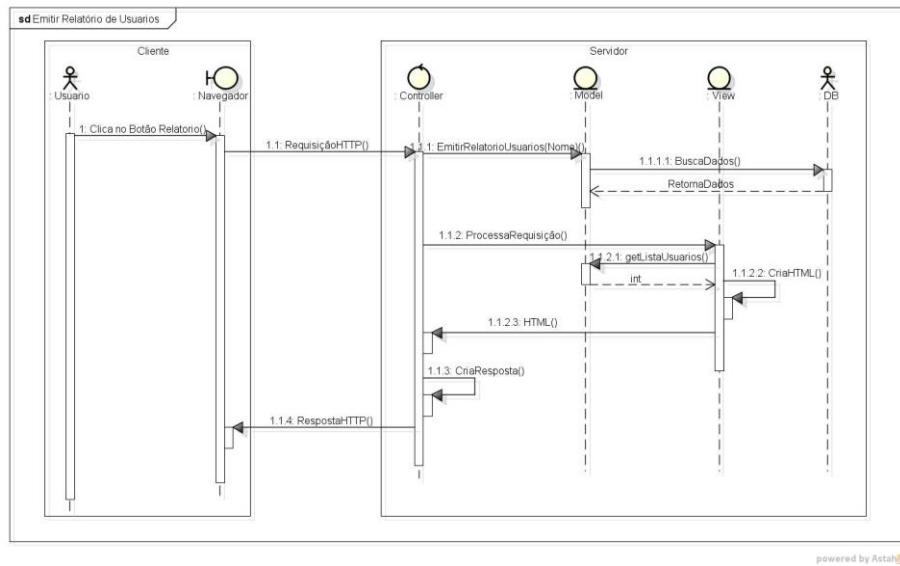
Segue abaixo na Figura 16 o diagrama de sequência voltado para obter arquivo.

**FIGURA 16 - Diagrama de sequência para obter arquivo**



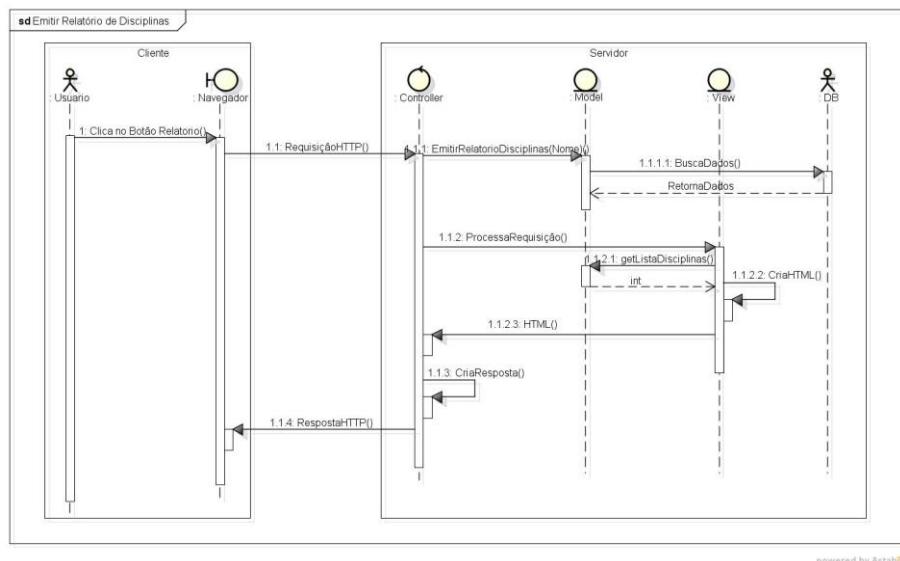
Segue abaixo na Figura 17 o diagrama de sequência voltado para emitir relatório de usuários.

**FIGURA 17 - Diagrama de sequência para emitir relatório de usuários**



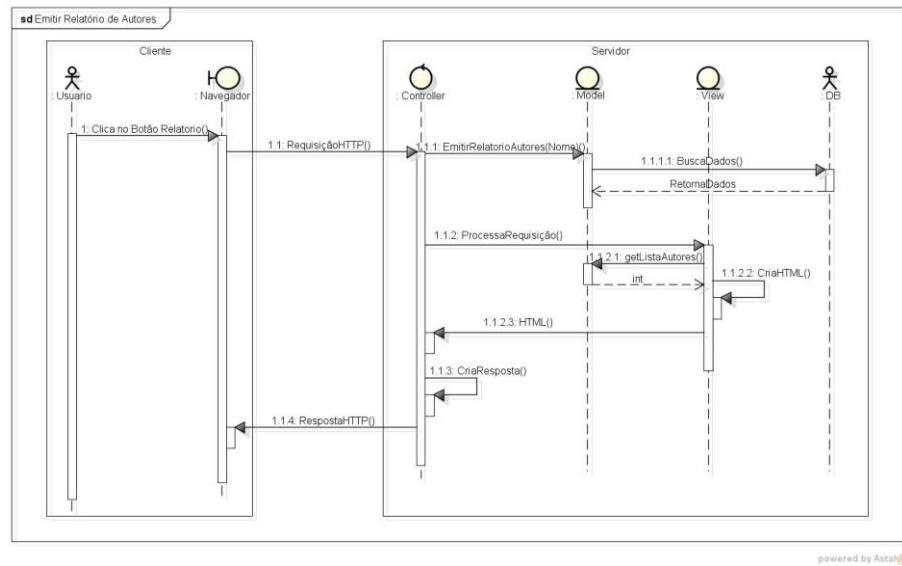
Segue abaixo na Figura 18 o diagrama de sequência voltado para emitir relatório de disciplinas.

**FIGURA 18 - Diagrama de sequência para emitir relatório de disciplinas**



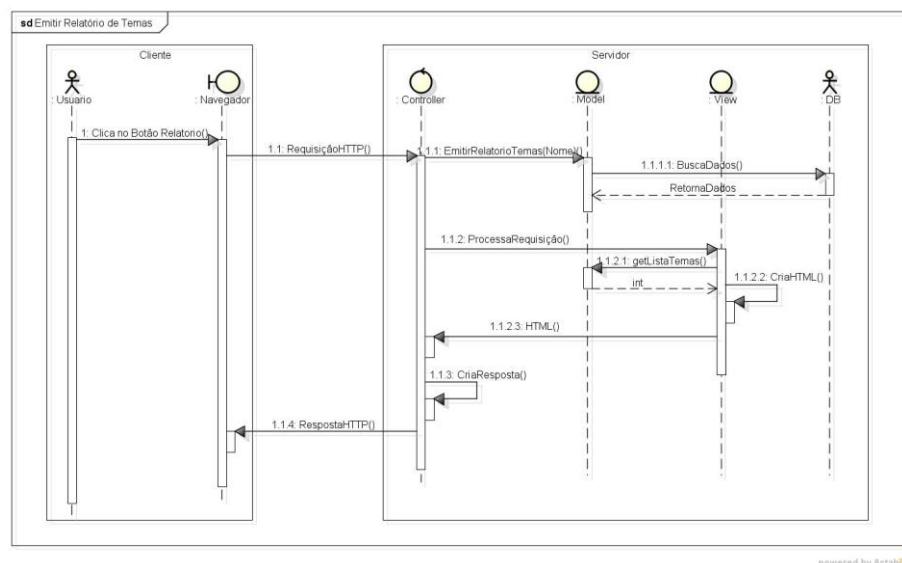
Segue abaixo na Figura 19 o diagrama de sequência voltado para emitir relatório de autores.

**FIGURA 19** - Diagrama de sequência para emitir relatório de autores



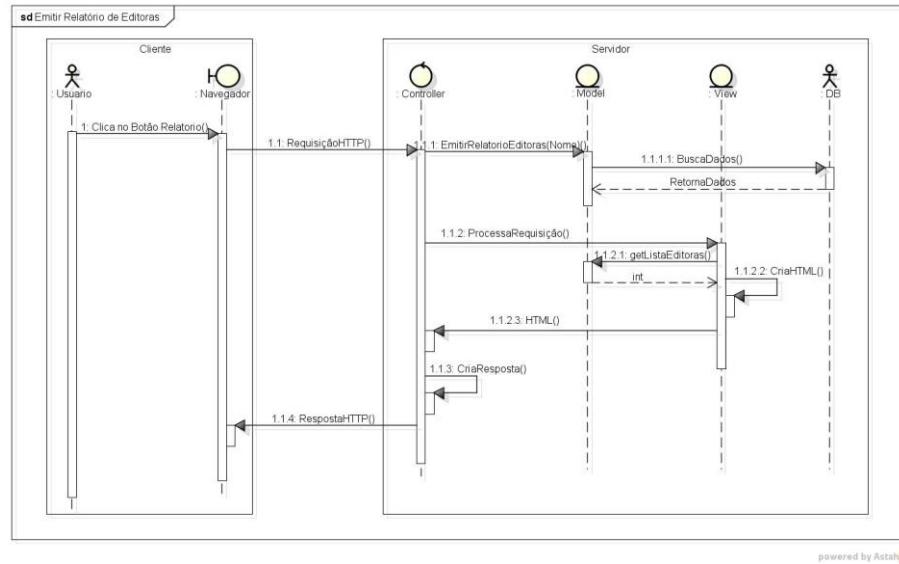
Segue abaixo na Figura 20 o diagrama de sequência voltado para emitir relatório de temas.

**FIGURA 20** - Diagrama de sequência para emitir relatório de autores.



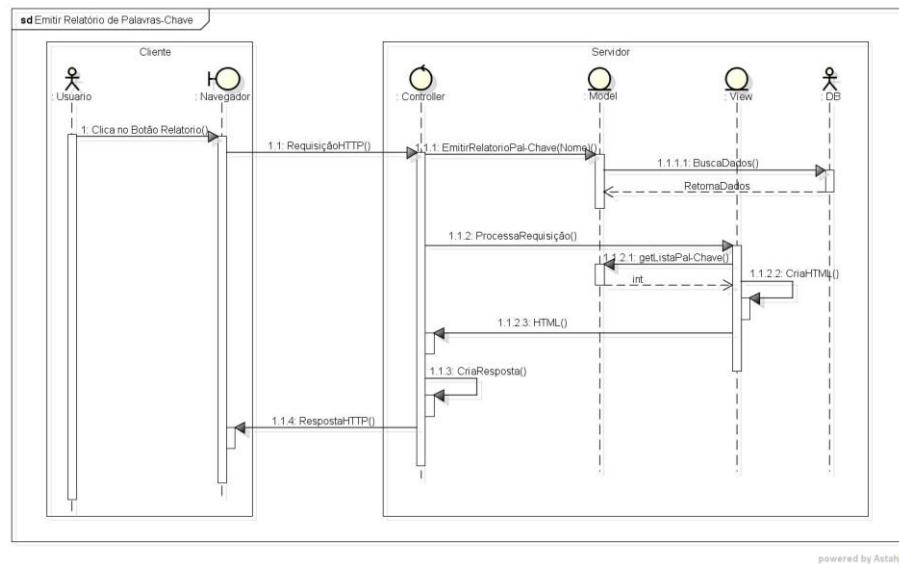
Segue abaixo na Figura 21 o diagrama de sequência voltado para emitir relatório de editoras.

**FIGURA 21** - Diagrama de sequência para emitir relatório de editoras.



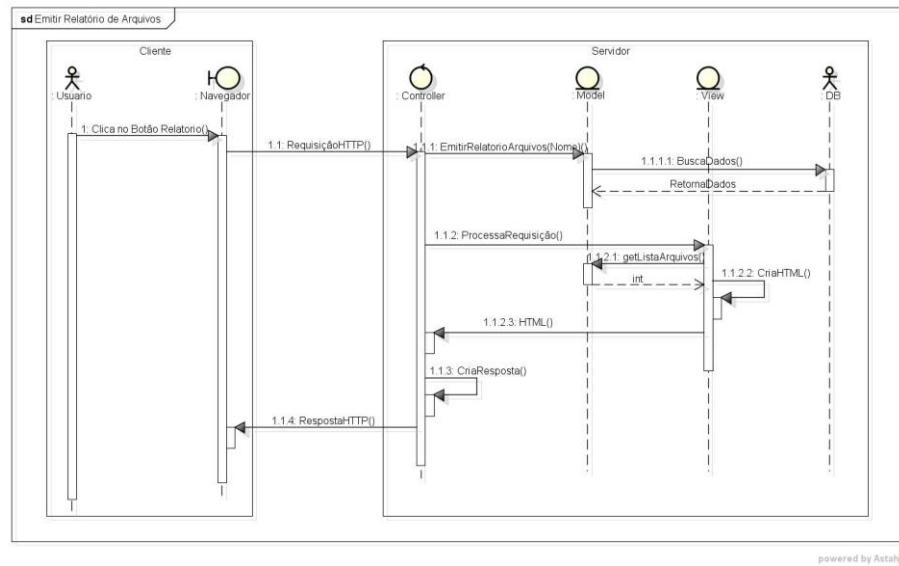
Segue abaixo na Figura 22 o diagrama de sequência voltado para emitir relatório de palavras-chave.

**FIGURA 22** - Diagrama de sequência para emitir relatório de palavras-chave



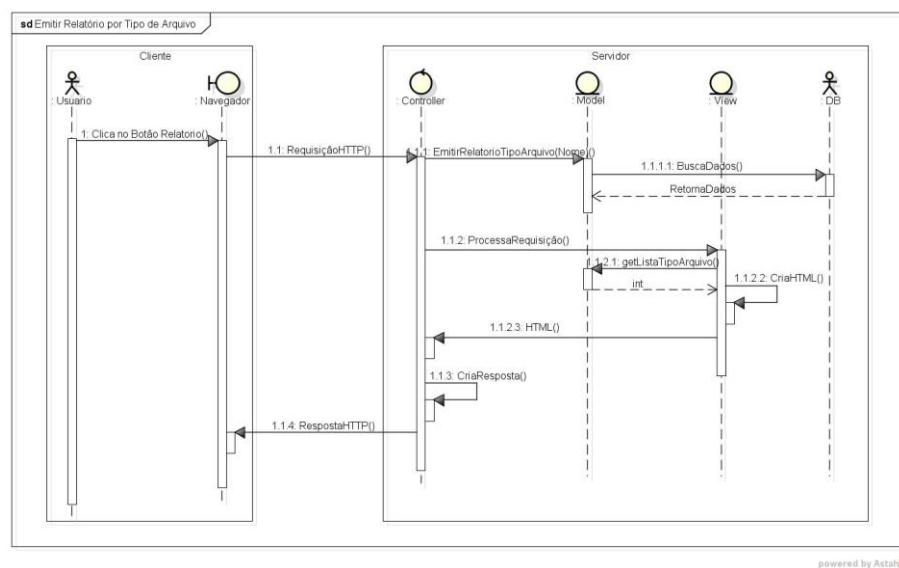
Segue na Figura 23 o diagrama de sequência voltado para emitir relatório de arquivos.

**FIGURA 23** - Diagrama de sequência para emitir relatório de arquivos



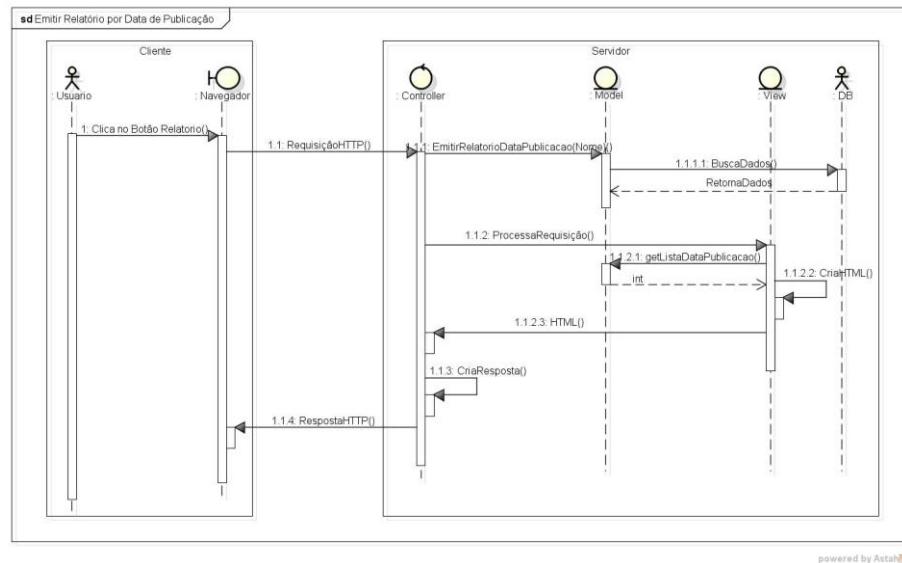
Segue abaixo na Figura 24 o diagrama de sequência voltado para emitir relatório por tipo de arquivo.

**FIGURA 24** - Diagrama de sequência para emitir relatório por tipo de arquivo



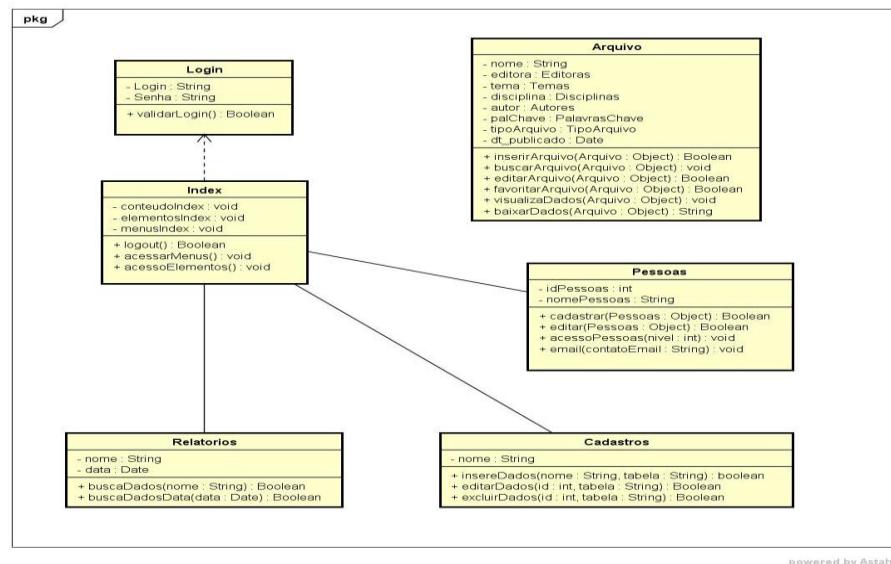
Segue abaixo na Figura 25 o diagrama de sequência voltado para emitir relatório por data de publicação.

**FIGURA 25** - Diagrama de sequência para emitir relatório por data de publicação



### 8f. Diagrama de Classe

**FIGURA 26** - Diagrama de classes



### 8g. Prioridade dos Requisitos

Essencial: Requisito onde se houver ausência, o software não entra em produção.

Importante: Requisito onde se houver ausência, o software funcionará, porém não será da melhor forma possível, podendo deixar a desejar na satisfação.

Desejável: Requisito onde se houver ausência, o software funcionará perfeitamente sem comprometer a funcionalidade satisfatória.

## 9 - REQUISITOS FUNCIONAIS

### 9a. Descrição dos Requisitos Funcionais

|                    |  |                     |    |            |
|--------------------|--|---------------------|----|------------|
| Requisito n°       | 01   | Autenticar Usuários |    |            |
| Descrição          | Permitir ou não o acesso do usuário ao software. |                     |    |            |
| Prioridade:        | Essencial  | Dependências:       | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos                        |                     |    |            |

|                    |   |                 |    |            |
|--------------------|---|-----------------|----|------------|
| Requisito n°       | 02  | Manter Usuários |    |            |
| Descrição          | Cadastrar, excluir ou alterar usuários para acesso ao sistema.<br><br>Dados obrigatórios: Nome, Sexo, E-mail, Senha |                 |    |            |
| Prioridade:        | Essencial   | Dependências:   | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos   |                 |    |            |

|                    |  |               |    |            |
|--------------------|--|---------------|----|------------|
| Requisito n°       | 03   | Manter Temas  |    |            |
| Descrição          | Cadastrar, excluir ou alterar temas que serão propostos nos arquivos publicados.<br><br>Dado Obrigatório: Nome |               |    |            |
| Prioridade:        | Essencial  | Dependências: | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |               |    |            |

|                    |  |               |    |            |
|--------------------|--|---------------|----|------------|
| Requisito n°       | 04   | Manter Autor  |    |            |
| Descrição          | Cadastrar, excluir ou alterar autores que serão relacionados aos arquivos.<br>Dado Obrigatório: Nome |               |    |            |
| Prioridade:        | Essencial  | Dependências: | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |               |    |            |

|                    |   |                |    |            |
|--------------------|---|----------------|----|------------|
| Requisito n°       | 05  | Manter Editora |    |            |
| Descrição          | Cadastrar, excluir ou alterar editoras que serão propostos nos arquivos publicados.<br>Dado Obrigatório: Nome |                |    |            |
| Prioridade:        | Essencial   | Dependências:  | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos   |                |    |            |

|                    |   |                   |    |            |
|--------------------|---|-------------------|----|------------|
| Requisito n°       | 06  | Manter Disciplina |    |            |
| Descrição          | Cadastrar, excluir ou alterar as disciplinas mantidas no sistema.<br>Dado Obrigatório: Nome |                   |    |            |
| Prioridade:        | Essencial   | Dependências:     | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos   |                   |    |            |

|                    |  |                      |    |            |
|--------------------|--|----------------------|----|------------|
| Requisito n°       | 07   | Manter Palavra-Chave |    |            |
| Descrição          | Cadastrar, excluir ou alterar Palavras-Chave que auxiliam na busca do arquivo.<br>Dado Obrigatório: Nome |                      |    |            |
| Prioridade:        | Essencial  | Dependências:        | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |                      |    |            |

|                    |   |                        |    |            |
|--------------------|---|------------------------|----|------------|
| Requisito nº       | 08  | Manter Tipo de Arquivo |    |            |
| Descrição          | Cadastrar, editar ou excluir Tipo de Arquivo que será atribuído ao conteúdo pelo qual ele pertence.<br><br>Dado Obrigatório: Nome |                        |    |            |
| Prioridade:        | Essencial   | Dependências:          | 01 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos   |                        |    |            |

|                    |   |                |                   |            |
|--------------------|---|----------------|-------------------|------------|
| Requisito nº       | 09  | Manter Arquivo |                   |            |
| Descrição          | Cadastrar, editar ou excluir arquivos que serão acessados pelos usuários.<br><br>Dados Obrigatórios: Nome, Disciplina, Autor, Tema, Editora, Palavra-Chave, Tipo de Arquivo |                |                   |            |
| Prioridade:        | Essencial   | Dependências:  | 03 04 05 06 07 08 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos   |                |                   |            |

|                    |  |                   |    |            |
|--------------------|--|-------------------|----|------------|
| Requisito nº       | 10   | Pesquisar Arquivo |    |            |
| Descrição          | Realizar a pesquisa do arquivo por tema, nome, editora, disciplina, autor ou palavra-chave.<br><br>Dados Obrigatórios: Pelo menos um entre os citados acima. |                   |    |            |
| Prioridade:        | Essencial  | Dependências:     | 09 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |                   |    |            |

|                    |  |                     |    |            |
|--------------------|--|---------------------|----|------------|
| Requisito nº       | 11   | Visualizar Arquivos |    |            |
| Descrição          | Permitir a visualização do arquivo antes de obtê-lo. |                     |    |            |
| Prioridade:        | Essencial  | Dependências:       | 10 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos                            |                     |    |            |

|                    |   |                |    |            |
|--------------------|---|----------------|----|------------|
| Requisito n°       | 12  | Obter Arquivos |    |            |
| Descrição          | Permitir o download de arquivos publicados. |                |    |            |
| Prioridade:        | Essencial                                   | Dependências:  | 11 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos                   |                |    |            |

|                    |   |                              |    |            |
|--------------------|---|------------------------------|----|------------|
| Requisito n°       | 13  | Emitir Relatório de Usuários |    |            |
| Descrição          | Emitir o relatório dos usuários cadastrados no sistema.<br>Dados Obrigatórios: Nome |                              |    |            |
| Prioridade:        | Essencial   | Dependências:                | 02 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos   |                              |    |            |

|                    |  |                                 |    |            |
|--------------------|--|---------------------------------|----|------------|
| Requisito n°       | 14   | Emitir Relatório de Disciplinas |    |            |
| Descrição          | Emitir o relatório das disciplinas cadastradas no sistema.<br>Dados Obrigatórios: Nome |                                 |    |            |
| Prioridade:        | Essencial  | Dependências:                   | 06 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |                                 |    |            |

|                    |  |                             |    |            |
|--------------------|--|-----------------------------|----|------------|
| Requisito n°       | 15   | Emitir Relatório de Autores |    |            |
| Descrição          | Emitir o relatório dos autores cadastrados no sistema.<br>Dados Obrigatórios: Nome |                             |    |            |
| Prioridade:        | Essencial  | Dependências:               | 04 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |                             |    |            |

|                    |  |                           |    |            |
|--------------------|--|---------------------------|----|------------|
| Requisito n°       | 16   | Emitir Relatório de Temas |    |            |
| Descrição          | Emitir o relatório dos temas cadastrados no sistema.<br>Dados Obrigatórios: Nome |                           |    |            |
| Prioridade:        | Essencial  | Dependências:             | 03 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |                           |    |            |

|                    |   |                              |    |            |
|--------------------|---|------------------------------|----|------------|
| Requisito n°       | 17  | Emitir Relatório de Editoras |    |            |
| Descrição          | Emitir o relatório das editoras cadastradas no sistema.<br>Dados Obrigatórios: Nome |                              |    |            |
| Prioridade:        | Essencial   | Dependências:                | 05 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos   |                              |    |            |

|                    |   |                                   |             |           |
|--------------------|---|-----------------------------------|-------------|-----------|
| Requisito n°       | 18  | Emitir Relatório de Palavra-Chave |             |           |
| Descrição          | Emitir o relatório das palavras-chave cadastradas no sistema.<br>Dados Obrigatórios: Nome |                                   |             |           |
| Prioridade:        | Essencial   | Dependências:                     | Prioridade: | Essencial |
| Material de Apoio: | Diagrama de Casos de Usos   |                                   |             |           |

|                    |  |                              |    |            |
|--------------------|--|------------------------------|----|------------|
| Requisito n°       | 19   | Emitir Relatório de Arquivos |    |            |
| Descrição          | Emitir o relatório dos arquivos presente no sistema.<br>Dados Obrigatórios: Nome |                              |    |            |
| Prioridade:        | Essencial  | Dependências:                | 09 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |                              |    |            |

|                    |  |                                      |    |            |
|--------------------|--|--------------------------------------|----|------------|
| Requisito nº       | 20   | Emitir Relatório por Tipo de Arquivo |    |            |
| Descrição          | Emitir o relatório pelos tipos de arquivos presentes no sistema.<br>Dados Obrigatórios: Nome |                                      |    |            |
| Prioridade:        | Essencial  | Dependências:                        | 09 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |                                      |    |            |

|                    |  |   |    |            |
|--------------------|--|---|----|------------|
| Requisito nº       | 21   | Emitir Relatório por Data de Publicação |    |            |
| Descrição          | Emitir o relatório dos arquivos de acordo com sua data de publicação que estão no sistema.<br>Dados Obrigatórios: Nome |   |    |            |
| Prioridade:        | Essencial  | Dependências:                           | 09 | Conflitos: |
| Material de Apoio: | Diagrama de Casos de Usos  |   |    |            |

## 10 - REQUISITOS NÃO FUNCIONAIS

### 10a. Requisitos de Aparência

Deverá ser intuitiva, atrativa e interativa, proporcionando um ambiente agradável e de fácil navegação a qualquer usuário, seja seu primeiro contato com o sistema ou não.

### 10b. Requisitos de Usabilidade e Humanidade

O Design moderno e intuitivo, especialmente modelado para o sistema, contando com uma atenção voltada para a usabilidade, em conjunto com boas práticas de programação, é uma dupla imbatível para o termo qualidade, sendo pensado especialmente neste ponto, se tornando um diferencial no projeto.

### 10c. Requisitos de Desempenho

Maior desenvoltura de acordo com a velocidade de conexão e latência, por ser um portal acessado através da internet, a maior dependência de um bom desempenho está relacionado com a conexão, depois seguidos pelo hardware do cliente e o software(navegador) utilizado. Recomenda-se conexões banda larga ou serviços de conexões móveis sem limite de tráfego de banda para usufruir do maior desempenho.

**10d. Requisitos de Segurança**

O acesso deve ser gerenciado por discentes no qual sua matrícula pertence a faculdade de Engenharia de Software, o mesmo segue para os docentes e ambos pertencente a um grupo de acesso específico com suas devidas funções.

## ANEXO B – Levantamento de dados (Questionário)

Segue Figura 1 abaixo demonstrando a estrutura do questionário realizado.

**FIGURA 1** - Questionário do levantamento de dados para os discentes

### Portal de Conteúdos e Acervos Acadêmicos

Coleta de feedback de estudantes do curso de Engenharia de Software

\*Obrigatório

1.

**1. Selecione o estado de conservação dos livros, artigos ou monografias encontrados em bibliotecas que você já presenciou. \***

*Marque todas que se aplicam.*

- Livros ou outros, totalmente novos
- Livros ou outros, descuidados e rabiscados
- Livros ou outros, parcialmente descuidados
- Livros ou outros, faltando páginas ou destruídos

2.

**2. Para você, o que é mais interessante na hora das pesquisas da faculdade? \***

*Marque todas que se aplicam.*

- Acessar um local com conteúdos disciplinares, não importando onde você esteja.
- Ter uma possibilidade de busca avançada de conteúdo no material disponível em um portal a qualquer hora.
- Ter um livro ou qualquer outro material físico em mãos para ler.
- Realizar leituras e pesquisas em livros ou outro material físico até encontrar o que precisa.

3.

**3. Você está satisfeito com o conteúdo DISCIPLINAR ACADÊMICO que encontra na internet? É completo e de fontes confiáveis? \***

*Marcar apenas uma oval.*

- Sim, sempre encontro conteúdos que fazem total sentido e de fontes renomadas.
- Sim, me atende bem, mas não tenho total certeza se está correto.
- Sim, até o momento sempre me atendeu, mas não faço a menor idéia de onde são as fontes e se está totalmente completo.
- Não, ultimamente ando me deparando com muita propaganda, títulos falsos ou coisas sem sentido.

4.

**4. Você está satisfeito com o conteúdo DISCIPLINAR ACADÊMICO que encontra em bibliotecas? \***

*Marcar apenas uma oval.*

- Sim, acho tudo que eu quero, rapidamente e fácil de encontrar.
- Sim, mas tenho dificuldades em achar o conteúdo certo.
- Não, sempre há dificuldades de encontrar o tema que preciso.
- Não, geralmente falta o conteúdo que procuro e não tem muito o que fazer para mudar isso.

## ANEXO C - COMUNICAÇÃO DO DIRETOR DO CURSO PARA O LEVANTAMENTO DE DADOS.

Segue Figura 1 abaixo demonstrando contato por e-mail para obtenção da totalidade de alunos efetivos.

**FIGURA 1** - Contato por e-mail para obtenção da totalidade de alunos efetivos

31/10/2015 Gmail - Documentação de alunos efetivos no curso de Engenharia de Software atualmente

**Gmail** by Google Douglas Júnior Aracnídeo <affedouglas@gmail.com>

---

**Documentação de alunos efetivos no curso de Engenharia de Software atualmente**

Fabian Corrêa Cardoso <fabian@unirv.edu.br> 22 de outubro de 2015 16:25  
Para: Douglas Júnior Aracnídeo <affedouglas@gmail.com>

Em 22 de outubro de 2015 12:29, Douglas Júnior Aracnídeo <affedouglas@gmail.com> escreveu:  
| Boa tarde Fabian,

Boa tarde, Douglas,

Assim como conversamos, favor me encaminhar de forma documentada a quantidade de alunos efetivos atualmente no curso de Engenharia de Software.

O novo sistema ainda não está me disponibilizando estes relatórios.

A finalidade deste documento é que será realizado um levantamento de dados através de questionário de pesquisa a ser realizado nesta semana com intuito de obter algumas respostas para análises de contexto a ser aplicado em monografia.

Pode considerar 135 o número de alunos matriculados neste semestre no curso de Engenharia de Software. Este era o número do início do semestre de forma que não aumentou, mas podem ter diminuído alguns alunos.

Preciso que seja de forma documentada, por exemplo: Relatório, Relação de Alunos ou qualquer documento em nome da diretoria do curso com identificação da faculdade para que seja deferido válido e anexado junto ao item de materiais e métodos para monografia.

Pode imprimir e considerar este e-mail como forma de documentação da informação.

Fabian C. Cardoso.