

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE

ANDREÍZA DE PAULA DA SILVA ELEUTERIO
ROBSON EDSON FERNANDES DA SILVA

MOOCKINHO: UM SISTEMA DE CURSOS ONLINE

Lajes/RN

2018

ANDREÍZA DE PAULA DA SILVA ELEUTERIO
ROBSON EDSON FERNANDES DA SILVA

MOOCKINHO: UM SISTEMA DE CURSOS ONLINE

Relatório de Prática Profissional apresentado ao Curso Técnico de Nível Médio em Informática, na modalidade subsequente, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, campus Avançado de Lajes, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Orientadora: Prof. Pedrina Célia Brasil.

Lajes/RN
2018

ANDREÍZA DE PAULA DA SILVA ELEUTERIO
ROBSON EDSON FERNANDES DA SILVA

MOOCKINHO: UM SISTEMA DE CURSOS ONLINE

Relatório de Prática Profissional apresentado ao Curso Técnico de Nível Médio em Informática, na modalidade subsequente, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, campus Avançado de Lajes, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Aprovado em 03 de julho de 2018.

BANCA EXAMINADORA

Pedrina Célia Brasil

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Mario Andrade Vieira de Melo Neto

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Jomar Ferreira dos Santos

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

AGRADECIMENTOS

Aos meus pais e familiares pelo amor, apoio, paciência e pela valorização dos estudos como essencial ao crescimento.

A todos aqueles que acreditaram na realização deste trabalho e deram-me forças e estímulo para dar prosseguimento a esta pesquisa e obter sucesso. Em especial, minha orientadora, Professora. Pedrina Célia Brasil, e aos meus colegas de turma.

A Deus criador dos céus e da terra, que me deu vida e forças para alcançar meus objetivos.

RESUMO

No atual contexto (de contínua evolução digital), quase todo mundo possui ou faz uso de um computador. É preciso repensar a metodologia tradicional de ensino e fazer uso dos novos recursos tecnológicos disponíveis na atualidade. Este trabalho tem como objetivo desenvolver uma ferramenta de apoio à aprendizagem, voltada para divulgação e gerenciamento de cursos online abertos e massivos. O sistema desenvolvido visa ampliar o acesso da sociedade aos cursos do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte.

Palavras-chave: Tecnologia; Metodologia; Sistema Web; MOOC.

ABSTRACT

In the current context (of continuous digital evolution), almost everybody owns or uses a computer. We need to rethink the traditional teaching methods and make use of new technological resources available today. This work aims to develop a learning support tool aimed at disseminating and managing massive open online courses. The developed system aims to broaden the access of society to the courses of the Federal Institute of Education, Science and technology of large northern river.

Keywords: Technology; Methodology; Web system; MOOC.

LISTA DE ILUSTRAÇÕES

Figura 1. Arquitetura MVC	19
Figura 2. Execução o Play Framework	19
Figura 3. Criar Play Framework	20
Figura 4. IDE Play Framework	20
Figura 5. Configuração de segurança	21
Figura 6. Execução do play deps	22
Figura 7. Importação das rotas	22
Figura 8. Exemplo do @With	23
Figura 9. Classe Security	23
Figura 10. Página de download do bootstrap	24
Figura 11. Tela inicial da Interface do Git	26
Figura 12. Exemplo do comando git add --all	26
Figura 13. Exemplo do comando git commit -m	26
Figura 14. Exemplo do comando git push	27
Figura 15. Exemplo do comando git pull	27
Figura 16. Diagrama de caso de uso	35
Figura 17. Diagrama de classe.	43
Figura 18. Funcionamento do listar cursos	44
Figura 19. Funcionamento do realizar cadastro.	45
Figura 20. Funcionamento do realizar cadastro para erro	45
Figura 21. Funcionamento do login para acesso negado	46
Figura 22. Funcionamento do login para acesso autorizado.	46
Figura 23. Funcionamento do recuperar senha.	47
Figura 24. Funcionamento do recuperar senha usuário não encontrado.	47
Figura 25. Funcionamento do detalhar curso para usuário avulso.	48
Figura 26. Funcionamento do detalhar curso para usuário não inscrito.	48
Figura 27. Funcionamento do detalhar curso para usuário inscrito.	49
Figura 28. Funcionamento do meu perfil.	50
Figura 29. Funcionamento do Alterar senha.	50
Figura 30. Funcionamento do Alterar senha para senhas que não correspondem.	51
Figura 31. Funcionamento do Solicitar promoção a administrador.	51
Figura 32. Funcionamento do Solicitar promoção a administrador código inválido.	52

Figura 33. Funcionamento de minhas matrículas.	52
Figura 34. Funcionamento do gerenciar curso.	53
Figura 35. Funcionamento do cadastro de curso.	53
Figura 36. Funcionamento do cadastro de curso erro.	54
Figura 37. Funcionamento do gerenciar videoaulas.	55
Figura 38. Funcionamento do cadastro da videoaula.	56
Figura 39. Funcionamento do cadastro de videoaula erro.	56
Figura 40. Funcionamento do assistir videoaula.	57

LISTA DE QUADROS

Quadro 1. Sprints do Projeto	15
Quadro 2. Lista de requisitos funcionais.	29
Quadro 3. Lista de requisitos não funcionais.	33
Quadro 4. Lista de requisitos não funcionais relativos às tecnologias utilizadas.	34
Quadro 5. Expansão do caso de uso Assistir Videoaula	35
Quadro 6. Expansão do caso de uso Detalhar Curso	37
Quadro 7. Expansão do caso de uso Gerenciar Curso	39
Quadro 8. Expansão do caso de uso Gerenciar Videoaula	41

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
EaD	<i>Educação a Distância</i>
MOOC	<i>Massively Open Online Courses</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hypertext Markup Language</i>
XML	<i>Xtensible Markup Language</i>
UML	<i>Unified Modeling Language</i>
IFRN	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
J2EE	<i>Java Platform Enterprise Edition</i>
IDE	<i>Integrated Development Environment</i>
MVC	<i>Model, View, Controller</i>
Play	<i>Play Framework</i>
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>
RF	<i>Requisitos Funcionais</i>
RNF	<i>Requisitos Não Funcionais</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO GERAL	14
1.2	OBJETIVOS ESPECÍFICOS	14
1.3	METODOLOGIA	14
1.4	ESTRUTURA DO TRABALHO	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	<i>MASSIVE OPEN ONLINE COURSE (MOOC)</i>	16
2.2	PLAY FRAMEWORK	17
2.3	TWITTER BOOTSTRAP	23
2.4	UNIFIED MODELING LANGUAGE (UML)	24
2.5	GIT	25
3	MODELAGEM DO PROJETO	28
3.1	VISÃO GERAL DO SISTEMA	28
3.2	ATORES DO SISTEMA	28
3.3	LEVANTAMENTO DE REQUISITO	29
3.3.1	<i>Requisitos funcionais</i>	29
3.3.2	<i>Requisitos não funcionais</i>	33
3.4	DIAGRAMAS DE CASOS DE USO	34
3.5	EXPANSÃO DOS CASOS DE USO DE MAIOR RISCO	35
3.6	DIAGRAMA DE CLASSE	43
4	RESULTADOS	44
4.1	LISTAR CURSOS	44
4.2	REALIZAR CADASTRO	44
4.3	REALIZAR <i>LOGIN</i>	45
4.4	RECUPERAR SENHA	47
4.5	DETALHAR CURSO	48
4.6	MEU PERFIL	49
4.7	MODIFICAR SENHA	50
4.8	SOLICITAR PROMOÇÃO A ADMINISTRADOR	51
4.9	MINHAS MATRÍCULAS	52

4.10	GERENCIAR CURSO	53
4.11	GERENCIAR VIDEOAULAS	54
4.12	ASSISTIR VIDEOAULA	57
5	CONCLUSÃO	58
5.1	TRABALHOS FUTUROS.....	58
	REFERÊNCIAS.....	59

1 INTRODUÇÃO

Segundo Da Matta e Figueiredo (2013), por meio da internet as pessoas compartilham o conhecimento que antes era restrito a um pequeno grupo de indivíduos. A internet abre possibilidades que precisam ser experimentadas, entre elas o potencial para conectar as pessoas, colocando-as diante de um modo de produção colaborativo (DA MATTA e FIGUEIREDO, 2013).

Em paralelo, “a aceleração do crescimento da educação, em geral, está tornando cada vez mais indistintos os limites entre disciplinas, instituições e locais geográficos” (LITTO, 2009a, p.15).

Atualmente é crescente a tendência de inserção de métodos, técnicas e tecnologias de Educação a Distância (EaD) em um sistema integrado de ensino que permita a combinação de recursos de aprendizagem que podem ser utilizados em cursos presenciais ou totalmente a distância (DA MATTA e FIGUEIREDO, 2013).

Os *Massively Open Online Courses* (MOOC) - em português *Cursos Online Abertos e Massivos* - é um modelo de curso *online*, com conteúdo distribuído de forma livre e aberta, acessível a qualquer pessoa, em qualquer lugar, por meio da internet. Eles possuem como característica principal o fato de permitir um engajamento ativo de dezenas ou centenas de milhares de estudantes que auto organizam sua participação de acordo com suas metas, conhecimentos prévios, habilidades e interesse comum. Uma característica marcante do MOOC é a formação de turmas. Nesses sistemas, os cursos têm período de realização e uma equipe de formadores que atuam sincronicamente com os aprendizes (DA MATTA e FIGUEIREDO 2013).

Visando tirar vantagem desse contexto tecnológico e ampliar o acesso da sociedade aos cursos do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN), é proposto neste projeto o desenvolvimento do sistema Moockinho, um sistema *web* com o objetivo de possibilitar a criação e divulgação de cursos online abertos e massivos. Através desse sistema de MOOC, alunos e servidores poderão criar cursos e disponibilizar videoaulas totalmente gratuitas e abertas à comunidade interna e externa do IFRN.

1.1 OBJETIVO GERAL

Este projeto tem como objetivo geral o desenvolvimento do sistema Moockinho, um sistema *web* para divulgação e gerenciamento de cursos online que incentive seus usuários a fazer treinamentos de forma aberta, gratuita e de qualidade.

1.2 OBJETIVOS ESPECÍFICOS

- a) Realizar pesquisas bibliográficas sobre o desenvolvimento de sistemas MOOC;
- b) Conceber e projetar um sistema *web* para divulgação e gerenciamento de cursos online abertos e massivos para o IFRN;
- c) Implementar o sistema projetado;
- d) Testar o sistema implementado; e
- e) Implantar em um servidor de aplicações *web* do IFRN o sistema desenvolvido.

1.3 METODOLOGIA

O processo de desenvolvimento utilizado neste projeto foi baseado na metodologia *Scrum*. Essa foi utilizada como base para acompanhamento e planejamento das atividades deste projeto. Este foi dividido em cinco etapas de desenvolvimento, também denominados de *sprints*, conforme ilustrado no quadro 1

Quadro 1. Sprints do Projeto

Sprint	Atividade	Equipe
1	Levantamento e Elaboração de Requisitos	Andreiza e Robson
2	Análise e projeto do Sistema	Andreiza e Robson
3	Implementação dos casos de uso de maior risco	Andreiza e Robson
4	Elaboração do relatório parcial do projeto.	Andreiza e Robson

5	Elaboração do relatório final do projeto.	Andreiza e Robson
---	---	-------------------

Fonte: Elaboração própria (2018)

Cada *sprint* teve duração média de 20 (vinte) a 30 (trinta) dias consecutivos de execução. E, ao final de cada *sprint*, era realizada uma reunião onde os resultados da equipe e os aprendizados daquele ciclo eram compartilhados. Em seguida a equipe selecionava as atividades a serem realizadas no próximo ciclo de execução.

Ao total 5 *sprints* foram realizadas para atender as atividades de iniciação, elaboração, construção e transição deste trabalho.

Para construção do sistema proposto foram pesquisadas tecnologias que facilitassem o desenvolvimento de sistemas *web*. Entre as tecnologias encontradas foram selecionados o *Play Framework* e o *Twitter Bootstrap* devido sua flexibilidade e facilidade de aprendizagem.

Paralelo a elas foram também utilizados: o ambiente de desenvolvimento integrado *Netbeans* para edição de código; o banco de dados *MySQL Server*, para armazenamento de dados; a linguagem de programação *J2EE*, para implementação da interface servidora deste projeto; e *HTML*, *CSS* e *Javascript* para implementação da interface cliente do sistema projetado.

1.4 ESTRUTURA DO TRABALHO

O presente documento está organizado em cinco capítulos. No primeiro capítulo é feita uma introdução ao tema deste projeto, sua motivação e seus objetivos. No segundo capítulo são apresentados os conceitos fundamentais das tecnologias e ferramentas que serviram de base à implementação deste trabalho. No terceiro capítulo é feita uma descrição do *software* proposto e sua modelagem. No quarto capítulo é feita uma apresentação das interfaces do sistema desenvolvido. No quinto capítulo é feita a conclusão e as considerações finais para este trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os aspectos teóricos deste trabalho, tais como: ferramentas de apoio ao ensino de cursos online massivos e abertos, o *Play Framework*, *Twitter Bootstrap*, *Unified Modeling Language* (UML) e o Git.

2.1 MASSIVE OPEN ONLINE COURSE (MOOC)

MOOC pode ser determinado como um modelo que integra três elementos: a conectividade das redes sociais, o conhecimento de um especialista em determinada área e a coleção de recursos online abertos. Não só estes elementos são importantes, mas também o número de inscritos no curso. Neste sentido, sua principal característica, reside no fato deste modelo construir um engajamento ativo de dezenas ou centenas de milhares de estudantes que auto organizam sua participação de acordo com suas metas, conhecimento prévio, habilidades e interesses comuns. Embora compartilhem de uma estrutura de cursos convencionais, com um cronograma pré-definido ou tópicos semanais, os cursos do tipo MOOC normalmente não são pagos, nem requerem pré-requisitos, senão o interesse por estudar. Também não pré-definem expectativas de participação ou uma certificação formal (CREED-DIKEOGU; CLARK,2013).

MOOC é um modelo para disponibilização de conteúdos de aprendizagem online para qualquer pessoa que queira fazer um curso, que possui como principais características: ser aberto, ser gratuito, colaborativo e distribuído (PISUTOVA, 2012). O curso é gratuito, não sendo necessário pagar por ele. O curso é colaborativo, geralmente não há atribuições específicas, mas há o envolvimento dos estudantes com seus pares e com os materiais.

O MOOC permite escalabilidade, pois o desenho do curso é apropriado para atender o crescimento exponencial de estudantes, podendo chegar a centenas de milhares de estudantes participando em cada oferta do curso (MOTA; INAMORATO, 2012).

No MOOC há essencialmente quatro tipos de atividades: a agregação, a remixagem, o reaproveitamento, e a retroalimentação. A agregação permite o acesso a uma ampla variedade de recursos de leitura, vídeos e recursos web do curso. A remixagem possibilita que o conteúdo, após ser criado, possa ser utilizado em outro formato, como um blog ou fórum de discussão, por exemplo. Na atividade

de reaproveitamento, os participantes são incentivados a criar algo próprio, de forma crítica. E, por fim, na retroalimentação os participantes são incentivados a compartilhar seu trabalho não somente com outras pessoas do curso, mas também com o restante das pessoas da rede, uma vez que estes cursos, via de regra, tem a dimensão global (KOP; FOUNIER; HILL, 2011).

Tivemos com base para criação deste projeto, dois MOOC: Tim Tec e Curso em Vídeo.

Tim Tec foi a primeira plataforma MOOC desenvolvida inteiramente no Brasil. Criada em 2013 com o objetivo de desenvolver uma plataforma virtual que disponibilizasse cursos online, livres e gratuitos voltados à educação profissional e tecnológica. Essa é um software livre que pode ser instalado por qualquer instituição de ensino ou organização interessada em ter sua própria plataforma MOOC. O código está acessível para download a instalação é gratuita. Qualquer usuário pode atender as aulas e obter um certificado de horas ao final do curso. Institutos Federais, universidades, organizações do Sistema e outras instituições de ensino têm usado a plataforma para potencializar as ações de educação a distância da Rede e-Tec Brasil (INSTITUTO TIM, 2018).

Já o Curso em Vídeo é uma empresa brasileira legalmente estabelecida e presta serviços educacionais à distância e consultoria para empresas. software de plataforma livre lançado em 2013. Segundo Penatti (2014) As videoaulas estão tanto na plataforma do Curso Em Vídeo quanto no YouTube, e nem precisa ter cadastro para assisti-las. Mas, ao fazê-lo, você tem algumas vantagens, como a possibilidade de baixar pacotes de conteúdo de apoio que podem ajudar em seus estudos e exercícios para treinar, além de ganhar um certificado válido em todo o Brasil ao término de cada curso – a plataforma contabiliza as horas estudadas através dela e, quando ele chega ao fim, permite que o certificado seja baixado (PENATTI, 2014).

2.2 PLAY FRAMEWORK

Segundo Leroux (2012), O *Play* é um *framework* para aplicações *web*, ele possui uma série de bibliotecas disponíveis no repositório *maven*¹, que uniformiza todas as fases de construção de um projeto, seja ele remoto ou local. Esse framework foi criado por Guillaume Bort no ano 2007, apesar de suas

¹ É uma ferramenta de automação de compilação utilizada primariamente em projetos Java.

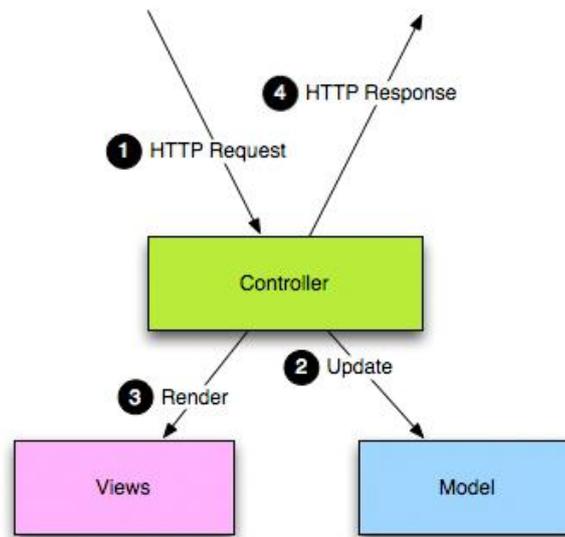
funcionalidades, apenas em outubro de 2009 foi publicada sua primeira versão, escrita em *Java* e *scala*. O *Play* tem com objetivo promover o desenvolvimento de aplicações *web* na linguagem *Java* de forma rápida e eficiente. Ele é um *framework* multi-plataforma, livre e de código aberto para desenvolvimento de aplicações *web*.

O *Play* permite que os desenvolvedores utilizem conjuntos de rotinas e padrões de programação para a criação de plataformas webs baseadas numa API *Java* ou *Scala*, sendo o mais aconselhável para iniciantes no *framework* o uso da linguagem *Java* devido a simplicidade (LEROUX et al., 2012).

Segundo Gamma (2012), esta ferramenta apresenta como arquitetura o padrão MVC, que organiza o sistema em três camadas de aplicação (GAMMA et al., 2000, p. 20):

- A camada de modelo (*Model*): representa os dados e a lógica de negócios da aplicação, ou seja, uma representação específica do domínio da informação em que a aplicação opera.
- A camada de visualização (*View*): é a apresentação na tela, em outras palavras – uma interface com o usuário. Em aplicações *web* a visão geralmente é feito utilizando *Hypertext Markup Language (HTML)* ou *Extensible Markup Language (XML)*.
- E a camada de controle (*Controller*): define a maneira como a interface do usuário reage às entradas do mesmo, ou seja, é o controlador que processa os eventos solicitado pelo usuário.

Figura 1. Arquitetura MVC



Fonte: Documentação Play (2018)

Para utilizar o *Play* basta baixar uma de suas versões na página oficial Play framework e descompactá-lo. A versão usada neste projeto foi a 1.3.4. Para utilizar o play é preciso definir nas variáveis de ambiente do sistema o caminho em que os arquivos do framework foram descompactados, conforme ilustrado na figura 2.

Figura 2. Adicionando o percurso do play no path

```
C:\Users\20162204110006>set path=%path%;C:\play
```

Fonte: Elaboração própria (2018)

Para criar uma aplicação no *play* é só digitar *play new <nome da aplicação>*, comando mostrado na figura 3.

componentes prontos para utilização, tais como o módulo *Secure*. O *Secure* é um módulo nativo do play que possibilita a configuração ou políticas de acesso e permissões de um sistema web. Através desse é possível gerenciar a autenticação e autorização dos usuários que acessam os serviços de uma aplicação web. Ele fornece um controlador simples que define um conjunto de interceptores que podem ser adicionados facilmente aos controladores específico de uma aplicação através ou anotação *@With* e *@Check* ().

Para utilizar o módulo *Secure* em uma aplicação, adicione no */conf/dependencies.yml* a dependência ao módulo *Secure*, conforme ilustrado na figura 5.

Figura 5. Configuração de segurança

```
# Application dependencies
```

```
require:
```

```
- play
```

```
- play -> secure
```

fonte: Elaboração própria (2018)

Em seguida, execute na linha de comando o comando *play deps*, para adicionar o módulo à sua aplicação, conforme ilustrado na figura 6.

Figura 8. Exemplo do @With

```
1 package controllers;
2
3 import play.mvc.With;
4 import models.*;
5 import org.apache.commons.codec.digest.Crypt;
6 import play.data.validation.Valid;
7 import play.db.jpa.JPA;
8 import play.libs.Crypto;
9
10 @With(Secure.class)
11 public class Alunos extends Usuarios {
12
13     public static void perfil() {
14         Usuario usu = Usuario.findById(Long.parseLong(session.get("userid").toString()));
15         render(usu);
16     }
17
18     public static void matriculas() {
19         Usuario usu = Usuario.findById(Long.parseLong(session.get("userid").toString()));
20         render(usu);
21     }
22 }
```

fonte: Elaboração própria (2018)

Figura 9. Classe Security

```
2 package controllers;
3
4 import java.lang.reflect.InvocationTargetException;
5 import models.Usuario;
6 import play.libs.Crypto;
7 import play.utils.Java;
8
9 public class Security extends Secure.Security {
10
11     static boolean authenticate(String username, String password) {
12         Usuario usu = Usuario.find("select u from Usuario u where lower(u.email) like ? ", username.toLowerCase()).first();
13         password = Crypto.passwordHash(password);
14         boolean autenticar = usu != null && usu.senha.equals(password);
15         if(autenticar){
16             session.put("userid", usu.id);
17             session.put("administrador", usu.administrador);
18         }
19         return autenticar;
20     }
21
22     static boolean check(String profile) {
23         Usuario usu = Usuario.find("byEmail", connected()).first();
24         return usu.administrador;
25     }
26
27 }
```

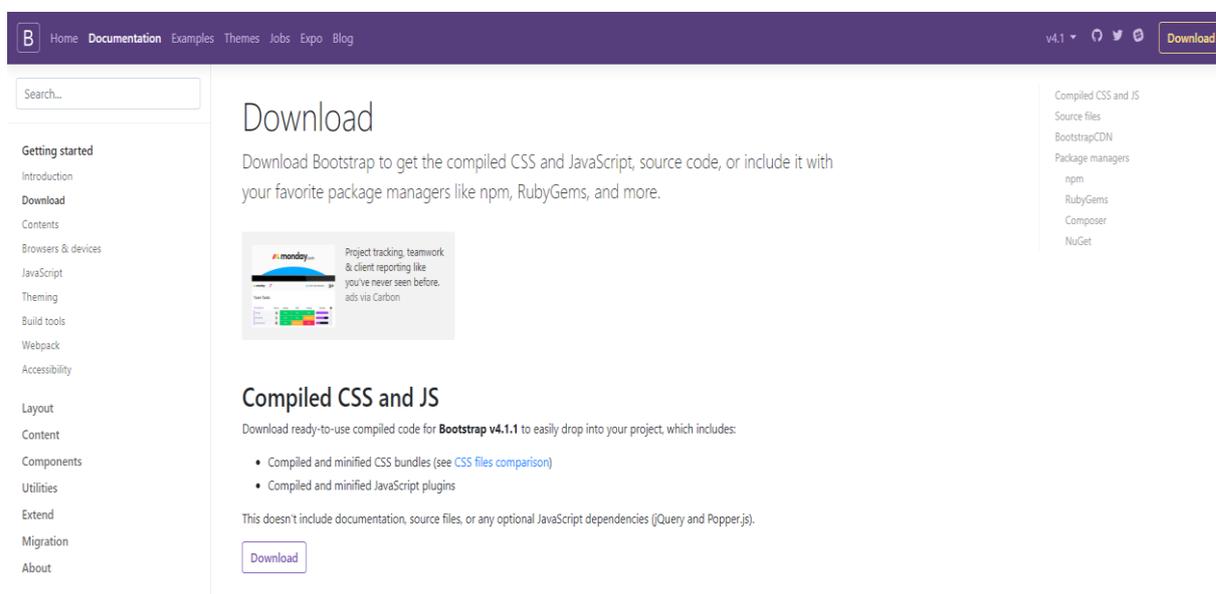
fonte: Elaboração própria (2018)

2.3 TWITTER BOOTSTRAP

O Twitter Bootstrap é um *framework front-end* para o desenvolvimento de sites e aplicações *web* responsivas. Ele torna o desenvolvimento *front-end* muito mais rápido e fácil, pois oferece componentes de interfaces prontos para o desenvolvedor. Sendo assim, essa ferramenta proporciona tanto agilidade quanto qualidade no desenvolvimento de sistemas web. (SILVA, 2015).

A versão do *Twitter Bootstrap* utilizada para a criação deste projeto foi a 4.1. Para utilizar os recursos do *bootstrap* é preciso baixar a versão disponível no site oficial da empresa, ter uma noção básica de HTML, CSS e JAVASCRIPT.

Figura 10. Página de download do bootstrap



fonte: Documentação Twitter Bootstrap (2018)

Contamos com vários recursos já prontos para facilitar o desenvolvimento das interfaces gráficas do sistema proposto neste projeto. Para o desenvolvimento deste projeto como por exemplo; formulários, botões, navegação e entre outros.

2.4 UNIFIED MODELING LANGUAGE (UML)

A UML ou linguagem de modelagem unificada, representa uma forma de uniformizar a modelagem de sistemas orientados a objeto, tornando fácil, simples e compreensivas a comunicação de uma aplicação com outras.

Segundo Reis e Videira (2003, p.117) “UML (*Unified Modeling Language*) é uma linguagem para especificação, construção, visualização e documentação de artefato de um sistema de *software*”.

Ela permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados.

Essa linguagem possui vários tipos de diagramas entre os quais, estão os usados para o desenvolvimento deste projeto ou seja: Diagrama de classe, Diagrama de caso de uso.

O Diagrama de Classes é usado para fazer a representação de estruturas de classes de negócio, *interfaces* e outros sistemas e classes de controle. Além disso, o diagrama de classes é considerado o mais importante para a UML, pois serve de apoio para a maioria dos demais diagramas. Se dá pela formação de conjunto de informações sobre determinadas classes, que unidas entre si formam um sentido geral do projeto (WIKIPÉDIA, 2018).

Um diagrama de caso de uso mostra um conjunto de casos de uso e atores (um tipo especial de classe) e seus relacionamentos. Aplique esses diagramas para ilustrar a visão estática do caso de uso de um sistema. Os diagramas de caso de uso são importantes principalmente para organização e modelagem dos comportamentos de um sistema (WIKIPÉDIA, 2018).

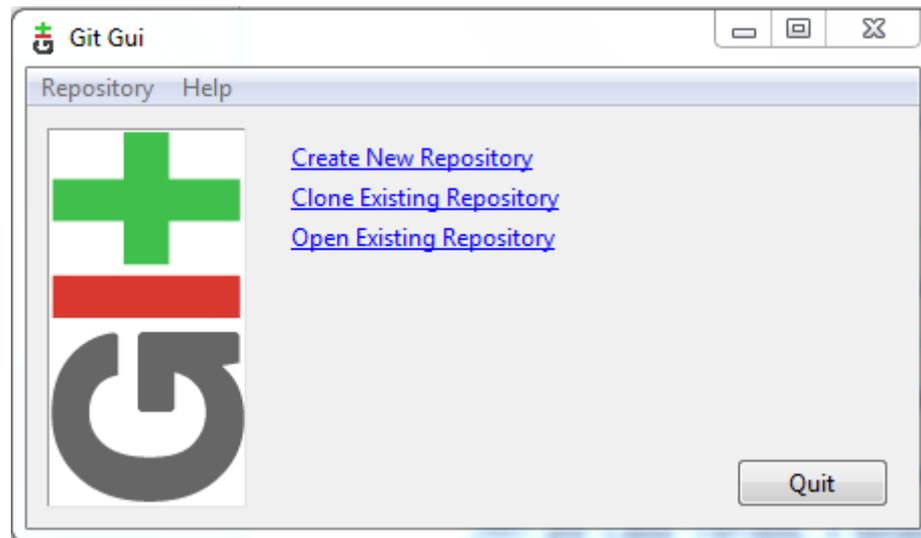
2.5 GIT

Git é um sistema de controle de versão distribuído (SCV). Ele possui uma ênfase em velocidade, integridade ou dados e suporte para *workflows* não lineares e distribuídos. Hoje ele pode ser considerado um SCV mais utilizado por desenvolvedores de *software*, atingindo uma grande aderência desde sua criação em 2005 (S. CHACON e B. STRAUB, 2009).

“Foi criado em 2005 por Linus Torvalds, o mesmo criador do *Linux*, que estava descontente com o *BitKeeper*, o sistema de controle de versão utilizado no desenvolvimento do *kernel* do *Linux*” (AQUILES, 2014, p.3).

Existem duas formas de trabalhar com o *git*, são elas: linha de comando e interface gráfica. Por interface gráfica é apresentado ao usuário 3 (três) opções. A primeira é para criar um novo repositório, a segunda é para clonar um que já exista e a terceira é para abrir um repositório que exista.

Figura 11. Tela inicial da Interface do Git

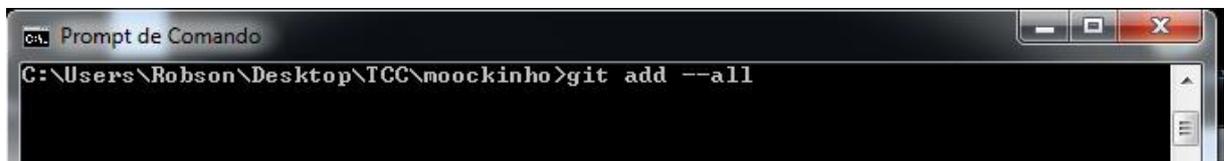


Fonte: Elaboração própria (2018)

Já por linha de comando existem 4 (quatro) comandos que são básicos para trabalhar com o *git*.

1. O *git add --all* que serve para adicionar as mudanças feitas no pacote.

Figura 12. Exemplo do comando git add --all



Fonte: Elaboração própria (2018)

2. O *git commit -m "mensagem de alteração"* que serve para criar o pacote de mudança.

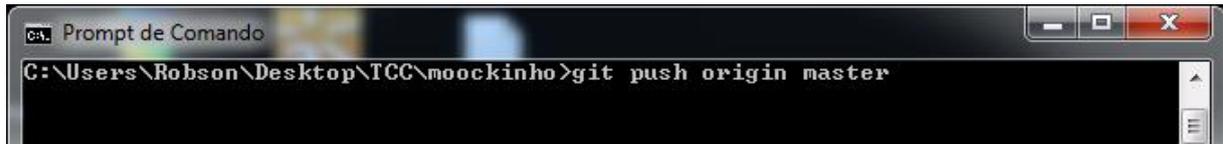
Figura 13. Exemplo do comando git commit -m



Fonte: Elaboração própria (2018)

3. O *git push origin master* que serve para enviar o pacote de mudança para o servidor online.

Figura 14. Exemplo do comando git push

A screenshot of a Windows Command Prompt window. The title bar reads "Prompt de Comando". The command prompt shows the current directory as "C:\Users\Robson\Desktop\TCC\moockinho" and the command "git push origin master" has been entered. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Fonte: Elaboração própria (2018)

4. O *git push origin master* que serve para pegar os pacotes de mudanças salvos no servidor online.

Figura 15. Exemplo do comando git pull

A screenshot of a Windows Command Prompt window. The title bar reads "Prompt de Comando". The command prompt shows the current directory as "C:\Users\Robson\Desktop\TCC\moockinho" and the command "git pull origin master" has been entered. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Fonte: Elaboração própria (2018)

3 MODELAGEM DO PROJETO

Este projeto prevê a criação de um sistema *web* para a criação e divulgação de cursos *online*, chamado Moockinho.

Esta seção detalha os resultados da modelagem do sistema proposto, oferecendo informações cruciais ao projeto, implementação, testes e manutenção do sistema.

3.1 VISÃO GERAL DO SISTEMA

O Moockinho é um *software* educacional com objetivo de disponibilizar cursos *online*, livres e gratuitos. Através desse, os usuários poderão se inscrever nos cursos e ter acesso às videoaulas cadastradas no sistema. Este é dividido em 3 módulos, conforme descrito a seguir:

- **Módulo Público:** é utilizado pelos usuários que não estão matriculados no sistema, chamados aqui de usuários avulsos, esses poderão visualizar cursos, cadastrar-se no sistema, detalhar cursos, efetuar login e recuperar senha.
- **Módulo Aluno:** é utilizado pelos usuários com registro no sistema e acesso de aluno. Eles estendem o usuário avulso e, além disso, poderão editar perfil, realizar matrículas, desmatricular-se de cursos, gerenciar suas matrículas, detalhar cursos inscritos, assistir videoaulas, avaliar cursos, realizar comentários as videoaulas e solicitar promoção a administrador.
- **Módulo Administrativo:** é utilizado pelos usuários com registro no sistema e acesso de administrador. Eles estendem os usuários alunos e, além disso, podem gerenciar cursos e gerenciar as videoaulas do sistema.

3.2 ATORES DO SISTEMA

Ao total o sistema possui 4 (quatro) atores, conforme descrito a seguir:

- **Avulso:** diz respeito aos usuários sem registro no sistema. Podem visualizar os cursos oferecidos e se cadastrar no sistema.
- **Aluno:** diz respeito aos usuários com registro no sistema e acesso de aluno. Eles podem acessar o sistema, se inscrever nos cursos, assistir

as videoaulas dos cursos que estão inscritos, acompanhar seu progresso acadêmico e solicitar acesso de administrador.

- **Administrador:** diz respeito aos usuários com registro no sistema e acesso de professor. Podem manter os cursos do sistema.
- **Youtube:** diz respeito ao sistema externo responsável por carregar as videoaulas do Moockinho.

3.3 LEVANTAMENTO DE REQUISITO

Foram utilizadas duas técnicas na elicitación de requisitos. Estudo de documentação/análise de conteúdo de terceiros e entrevista com o orientador.

No primeiro caso, foram analisados as documentações dos sistemas online Curso em Vídeo² e o TimTec³, sistemas similares ao aqui proposto.

Além dessa estratégia, também foram feitas entrevistas com a orientadora deste projeto com objetivo de especificar os requisitos do sistema proposto.

Ambas as estratégias foram aplicadas pelos autores deste trabalho e facilitaram a coleta de boa parte das informações necessárias para o desenvolvimento dos requisitos do sistema proposto.

3.3.1 Requisitos funcionais

Os requisitos funcionais representam as principais funcionalidades que o sistema deve efetuar. Eles são representados pelo quadro 2.

Quadro 2. Lista de requisitos funcionais do Moockinho.

Código	Nome	Descrição	Categoria
RF001	Listar cursos cadastrados	O sistema deve permitir aos usuários visualizar os cursos cadastrado do sistema ao clicar na opção “cursos” no menu principal. Ao clicar nesta opção é exibido ao usuário capa, título do curso, data de início, nome do professor (a) e a nota avaliativa do curso.	<i>Média</i>
RF002	Detalhar curso	O sistema deve permitir aos usuários detalhar as informações de um curso do sistema. Ao clicar na opção “Detalhar” é	<i>Alta</i>

² <https://www.cursoemvideo.com>

³ <http://timtec.com.br/pt/>

		exibido ao usuário título do curso, data de início, nome do professor (a), nota avaliativa do curso, capa, descrição e estrutura de aulas.	
RF003	Realizar cadastro no sistema	O sistema deve permitir aos usuários realizar cadastro no sistema informando seu nome, sobrenome, email e senha de registro.	<i>Média</i>
RF004	Realizar login	O sistema deve permitir aos usuários realizar login para acesso a área restrita do sistema. O usuário deverá informar seu e-mail e senha de registro para realizar seu login.	<i>Média</i>
RF005	Recuperar Senha	O sistema deve permitir aos usuários que eles recuperem sua senha informando o e-mail de cadastro.	<i>Baixa</i>
RF006	Realizar inscrição em curso	O sistema deve permitir aos usuários cadastrados realizar inscrição em cursos do sistema. Ao clicar na opção “Detalhar” é exibido ao usuário título do curso, data de início, nome do professor (a), nota avaliativa do curso, capa, descrição, estrutura de aulas e uma opção de realizar matrícula.	<i>Média</i>
RF007	Gerenciar minhas matrículas	O sistema deve permitir aos usuários cadastrados desfazer inscrição em cursos que eles já estiverem com matrícula ativa. Ao clicar em “Minhas Matrículas” é mostrado ao usuário os cursos em que eles estão matriculados, o progresso do aluno, uma opção de detalhar curso e outra opção de desmatricular-se.	<i>Média</i>
RF008	Desmatricular-se	O sistema deve permitir aos usuários que estes possam desmatricular-se dos cursos em que estão inscritos. Ao clicar em “Cancelar Matrícula” o sistema modificará o estado da matrícula para inativa.	<i>Baixa</i>
RF009	Assistir videoaula	O sistema deve permitir aos usuários cadastrados assistir videoaulas dos cursos	<i>Alta</i>

		que eles possuem matrícula ativa. Ao clicar em “Assistir” é exibido ao usuário título do curso, número da aula, descrição da aula, comentários sobre a aula, lista das videoaulas, um campo para fazer comentários, uma opção para comentar e outra para cancelar comentário.	
RF010	Visualizar progresso de aprendizagem	O sistema deve permitir aos usuários cadastrados acompanhar seu progresso de aprendizagem baseando-se na relação do total de aulas do curso e quantitativo de aulas já assistidas.	<i>Baixa</i>
RF011	Gerenciar meu perfil de usuário	O sistema deve permitir aos usuários cadastrados modificar as informações de seu perfil de cadastro. Ao clicar em “Meu Perfil” é mostrado ao usuário nome, sobrenome, e-mail, um botão solicitar promoção, um botão de modificar senha e um botão de salvar alterações.	<i>Baixa</i>
RF012	Solicitar Promoção a administrador	O sistema deve permitir aos usuários alunos do sistema solicitar promoção a administrador. Ao clicar no botão “Solicitar Promoção” é solicitado ao usuário um código de validação, o sistema realizará a promoção do usuário a administrador caso o código informado for validado. Ao final da validação, o sistema modificará o perfil do usuário para administrador.	<i>Média</i>
RF013	Fazer comentários às aulas	O sistema deve permitir aos usuários cadastrados realizar comentários as videoaulas dos cursos que eles estão inscritos. Ao clicar em “Assistir” é exibido ao usuário nome do curso, número da aula, descrição da aula, comentários sobre a aula, lista das videoaulas, um campo para fazer comentários, uma opção para comentar e outra para cancelar comentário.	<i>Baixa</i>
RF014	Avaliar curso	O sistema deve permitir aos usuários cadastrados avaliar os cursos que eles estão	<i>Baixa</i>

		inscritos a partir de uma quantidade numérica.	
RF015	Cadastrar cursos	O sistema deve permitir aos administradores cadastrar novos cursos no sistema. O usuário administrador poderá cadastrar cursos informando o título do curso, estado, data de início (publicação), descrição do curso e imagem de capa.	<i>Alta</i>
RF016	Alterar cursos	O sistema deve permitir aos administradores alterar seus cursos cadastrados. O usuário administrador poderá alterar título do curso, estado, data de início (publicação), descrição do curso e imagem de capa.	<i>Média</i>
RF017	Excluir cursos	O sistema deve permitir aos administradores excluir seus cursos. O usuário administrador poderá excluir os seus cursos já cadastrados.	<i>Alta</i>
RF018	Cadastrar videoaula	O sistema deve permitir aos administradores adicionar videoaulas aos seus cursos já cadastrados. O usuário administrador poderá cadastrar videoaulas informando título da vídeoaula, ordem de apresentação, descrição e url da videoaula.	<i>Alta</i>
RF019	Alterar videoaula	O sistema deve permitir aos administradores alterar suas videoaulas cadastradas. O usuário administrador poderá alterar as suas videoaulas modificando título da vídeoaula, ordem de apresentação, descrição e url da videoaula.	<i>Média</i>
RF020	Remover videoaula	O sistema deve permitir aos administradores remover suas videoaulas cadastradas. O usuário administrador poderá remover suas videoaulas já cadastradas.	<i>Alta</i>

Fonte: Elaboração própria (2018).

3.3.2 Requisitos não funcionais

Os requisitos não funcionais estão relacionados às restrições do sistema quanto a sua disponibilidade, desempenho e segurança. Eles são representados pelo quadro 3, mostrada a seguir.

Quadro 3. Lista de requisitos não funcionais.

Código	Nome	Descrição	Categoria
RNF001	Paginação	O sistema deve limitar a quantidade de cursos visualizados por paginação de até 20 cursos por página.	Baixa
RNF002	Limitação de caracteres	O sistema deve limitar os comentários a 255 caracteres.	Baixa
RNF003	Acesso Restrito	O sistema não deverá permitir o acesso de usuários não cadastrados a área restrita do sistema. As funcionalidades do sistema serão dispostas conforme os seguintes níveis de permissão: público, alunos sem matrícula, alunos com matrícula, administrador.	Alta
RNF004	Criptografia de senha	O sistema deverá criptografar a senha de usuário através do padrão MD5.	Média
RNF005	Duplicidade de inscrição	O usuário não poderá se inscrever em cursos que já estiver com inscrição ativa.	Alta
RNF006	Duplicidade de cursos	O sistema não deverá permitir cadastro de curso com mesmo título de curso ativo.	Alta
RNF007	Integridade de dados	O sistema não deverá permitir a exclusão de curso/videoaula que tenham dados de usuários usuários ativos.	Alta
RNF008	YouTube	Todas as vídeos aulas do sistema deverão estar no YouTube e serão carregadas a partir da URL do vídeo nesse sistema.	Alta
RNF009	Duplicidade de videoaula	O sistema não deverá permitir cadastro de video aula com mesmo url em um mesmo curso ativo.	Alta

RNF010	Avaliação de curso	Somente usuários com a matrícula ativa podem avaliar o curso.	Média
--------	--------------------	---	-------

Fonte: Elaboração própria (2018).

Os requisitos relativos às tecnologias utilizadas no desenvolvimento do sistema são representados pelo quadro 4.

Quadro 4. Lista de requisitos não funcionais relativos às tecnologias utilizadas.

Código	Nome	Descrição	Categoria
RNF08	Arquitetura MVC	A implementação do sistema deve empregar uma arquitetura em 3 (três) camadas: apresentação, negócio e dados.	Alta
RNF09	Linguagem de programação	A implementação do sistema deve utilizar a linguagem Java, adotando padrão J2EE.	Média
RNF10	Framework de desenvolvimento	Deve ser utilizado Play Framework 1.3.4	Baixa
RNF011	Disponibilidade	O portal de conteúdo do sistema deverá estar disponível aos usuários 24 horas por dia e 7 dias por semana.	Alta
RNF012	Banco de dados	A implementação do sistema deve empregar o MySQL Server como servidor de banco de dados.	Alta

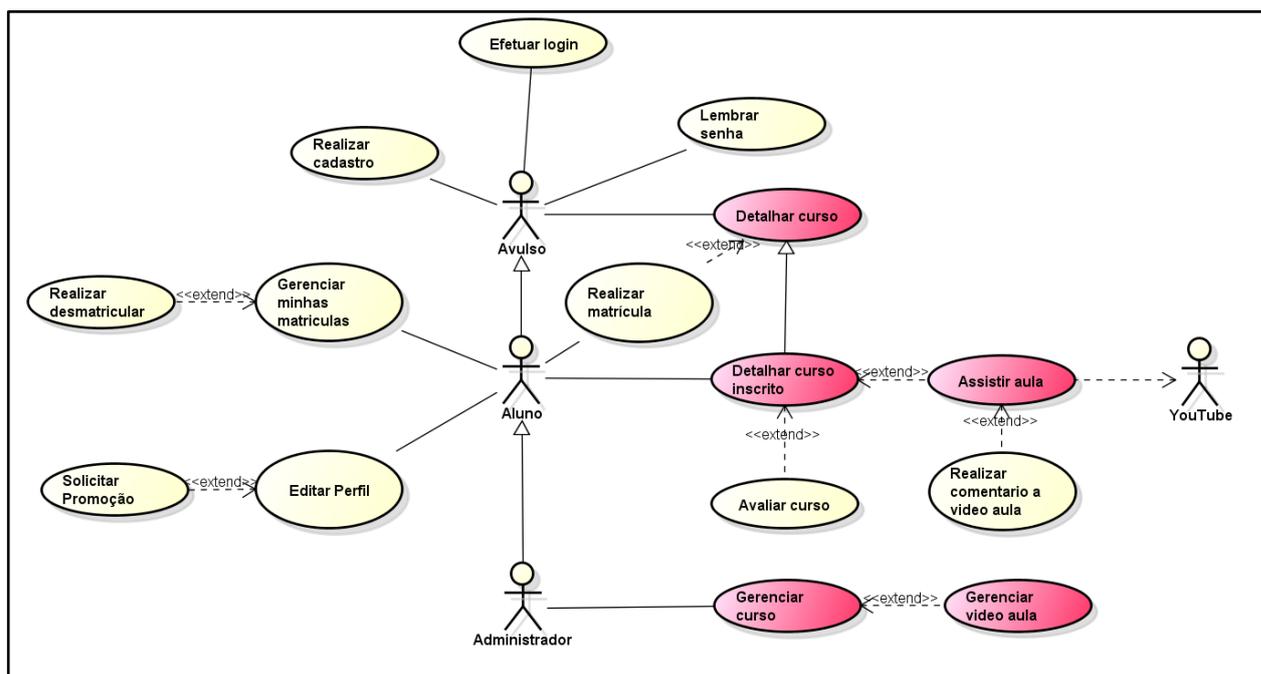
Fonte: Elaboração própria (2018).

3.4 DIAGRAMAS DE CASOS DE USO

O diagrama de casos de uso descreve o escopo do sistema projetado, além de especificar textualmente este escopo, a técnica de casos de uso é uma excelente ferramenta para abstrair os requisitos funcionais e atores do sistema, mostrando quem eles são e como interagem entre si.

Dessa forma, levando em consideração os atores e os requisitos funcionais deste trabalho, foram definidos 15 (quinze) casos de uso, conforme ilustrado na Figura 16.

Figura 16. Diagrama de caso de uso



Fonte: Elaboração própria (2018)

Ao total 5 (cinco) casos de uso foram considerados como de maior risco. São eles: detalhar curso, detalhar curso inscrito, assistir aula, gerenciar curso e gerenciar videoaula.

3.5 EXPANSÃO DOS CASOS DE USO DE MAIOR RISCO

Nessa seção são apresentadas a expansão de 4 (quatro) dos 5 (cinco) casos de uso de maior risco do sistema proposto.

Quadro 5. Expansão do caso de uso Assistir Videoaula

Assistir videoaula:

Este caso de uso especifica as ações que um usuário executa no sistema, com o objetivo de assistir uma videoaula. Apenas usuários cadastrados que possuem matrícula ativa no curso podem assistir videoaulas e realizar comentários a elas.

Atores:

Aluno, Administrador

Pré-condição:

O usuário deve possuir matrícula ativa no curso e estar visualizando o curso que deseja assistir a videoaula.

A aula deve estar disponível na URL ou origem.

Pós-condição:

O sistema registra que o usuário assistiu a vídeo aula.

Requisitos Funcionais:

RF009 - Assistir videoaula

RF010 - Visualizar progresso de aprendizagem

RF013 - Fazer comentários às aulas

Requisitos Não Funcionais:

RNF002 - Limitação dos caracteres

RNF08 - YouTube

RNF003 - Acesso Restrito

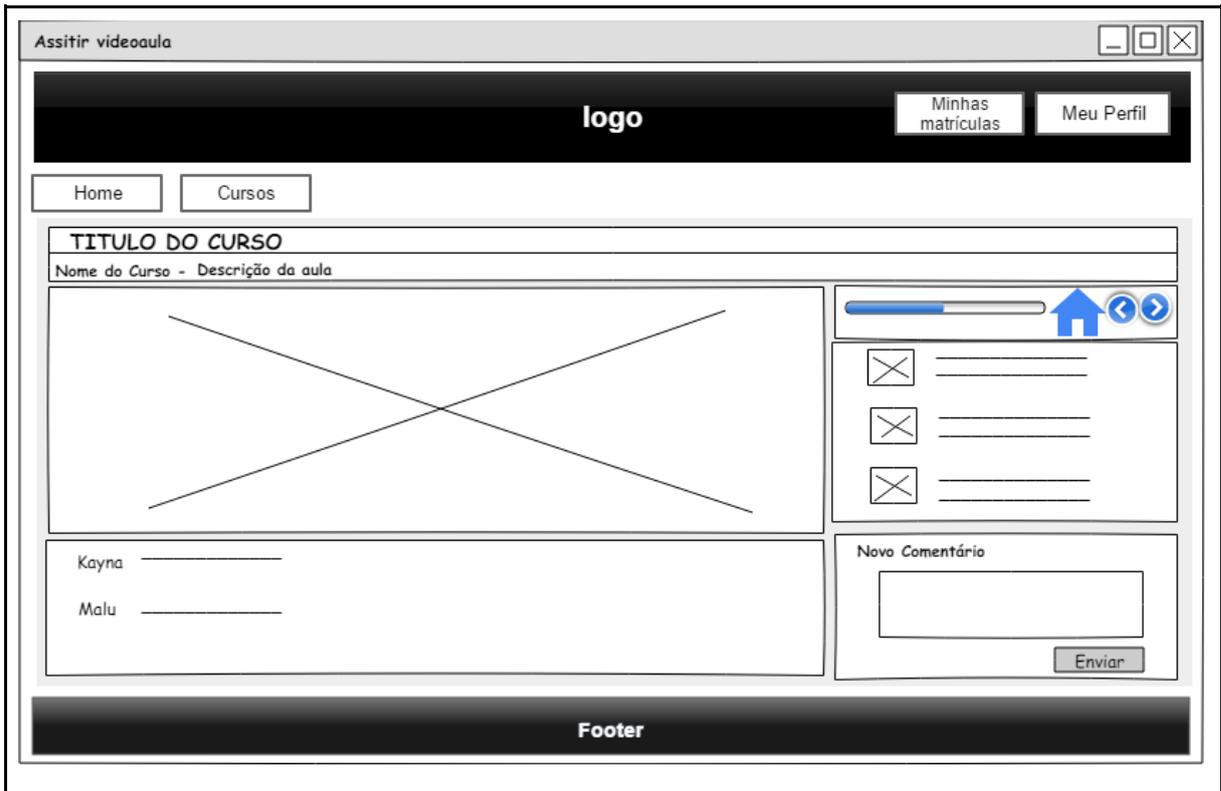
Fluxo Básico:

1. Na tela de detalhes do curso, o usuário clica em “Continuar”.
2. O sistema redireciona o usuário para a tela de assistir videoaula.
3. O sistema verifica a última videoaula assistida pelo aluno e carrega a próxima.
4. O sistema carrega nome do curso, descrição da aula, comentários sobre a aula, lista ordenada das videoaulas do curso e um formulário para fazer comentários.
5. O sistema requisita videoaula a API YouTube.
6. A API do YouTube carrega a videoaula.
7. O sistema inicia a videoaula carregada.
8. O sistema registra progresso acadêmico do aluno.

Fluxos Alternativos:

- 6a. Youtube fora do ar - Passado 30 segundos não havendo resposta
1. O sistema informa o erro ao ator.
- 3a. Usuário já assistiu todas as videoaulas do curso
1. O sistema informa que o usuário já concluiu o curso.
 2. O sistema executa o passo 4.

Protótipo de Tela:



Quadro 6. Expansão do caso de uso Detalhar Curso

Detalhar curso

Este caso de uso especifica as ações que um usuário executa no sistema com o objetivo de detalhar um curso. Qualquer usuário pode detalhar as informações de um curso do sistema.

Atores:

Avulso, Aluno, Administrador

Pré-condição:

O usuário deve estar visualizando a lista de cursos cadastrados no sistema.

Pós-condição:

O sistema exibe as informações do curso selecionado.

Requisitos Funcionais:

RF001 - Listar cursos cadastrados

RF002 - Detalhar curso

Requisitos Não Funcionais:

RNF001 - Paginação

Fluxo Básico:

1. O usuário clica em “Visualizar curso”.
2. O sistema redireciona o usuário para a tela de detalhes do curso selecionado.
3. O sistema carrega título, resumo, porquê fazer, imagem de capa, data de publicação do curso e estrutura de aulas.

Fluxo Alternativo:

2a. usuário com matrícula ativa no curso

1. O sistema carrega o progresso acadêmico do usuário no curso e o botão de continuar.
2. O sistema carrega o formulário de avaliação do curso com a avaliação do usuário (nota e comentário).
3. Continua no passo 3.

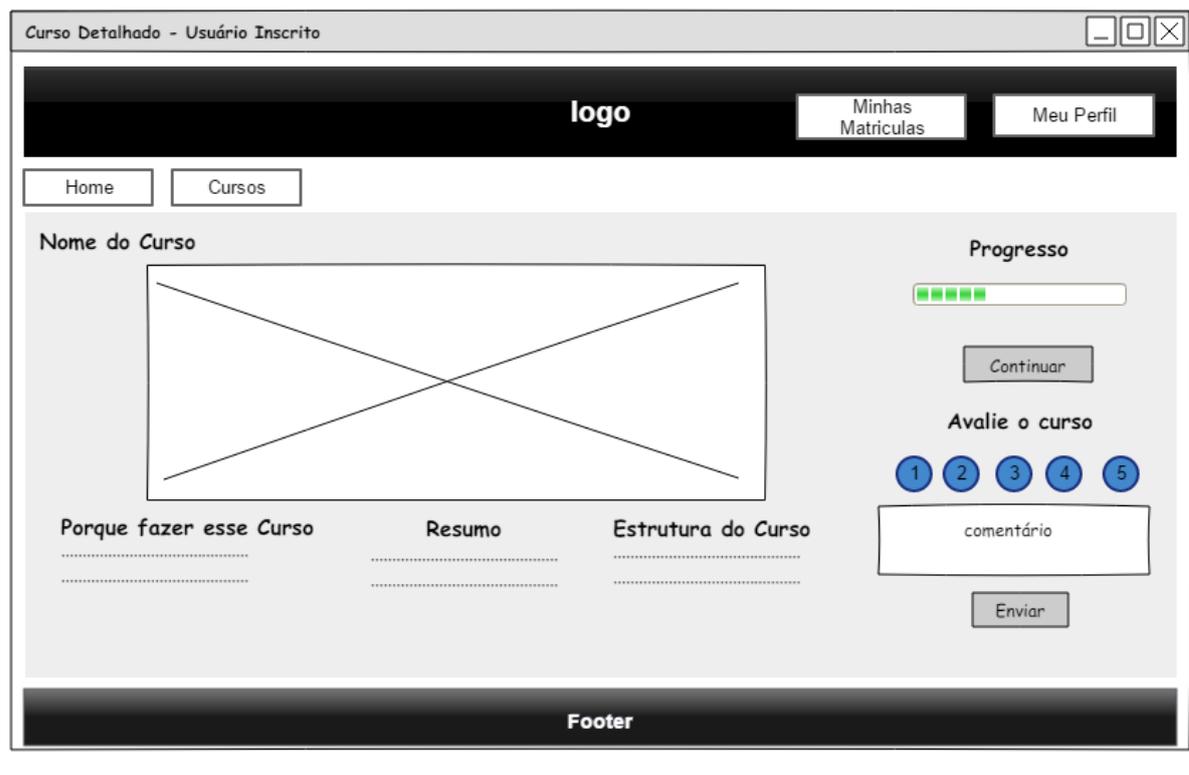
2b. usuário sem matrícula ativa no curso

1. O sistema carrega um botão para o usuário realizar matrícula no curso.
2. Continua no passo 3.

2c. usuário é administrador do curso

1. O sistema carrega um botão para gerenciar videoaulas do curso.
2. Continua no passo 3.

Protótipo:



Quadro 7. Expansão do caso de uso Gerenciar Curso

Gerenciar Cursos

Este caso de uso especifica as ações que um usuário executa no sistema com o objetivo de cadastrar e manter os cursos do sistema. Somente usuário administrador poderá cadastrar cursos.

Ator:

Administrador

Pré-condição:

O usuário deve ter feito login e estar em “Meus Cursos”.

Pós-condição:

O sistema exibe as informações recém cadastradas do curso.

Requisitos Funcionais:

RF015 - Cadastrar cursos

RF016 - Alterar cursos

RF017 - Excluir cursos

RF002 - Detalhar curso

Requisitos Não Funcionais:

RNF003 - Acesso Restrito

RNF006 - Duplicidade de cursos

RNF007 - Integridade de dados

Fluxo de Básico:

1. Na tela de meus cursos, o usuário clica em “Cadastrar curso”.
2. O sistema redireciona o usuário para a tela de cadastro de curso.
3. O sistema apresenta o formulário de cadastro de curso.
4. O usuário insere os dados como título, resumo, o porquê fazer o curso, estado, data de início (publicação) e imagem de capa nos campos correspondentes.
5. O usuário clica no botão “Cadastrar”.
6. O sistema salva os dados no banco de dados.
7. O sistema informa que o cadastro foi realizado com sucesso.
8. O sistema exibe o curso e suas informações.

Fluxo de Alternativo:

1a. Usuário quer alterar dados do curso

1. O usuário seleciona curso e clica na opção “Alterar”.
2. O sistema carrega as informações do curso.
3. O usuário informa os dados a serem atualizados.
4. O usuário confirma as alterações.
5. O sistema salva as alterações.
6. Continua no passo 8.

1b. Usuário quer remover curso

1. O usuário seleciona curso e clica na opção “Excluir”.
2. O sistema exclui todos os dados do banco referente ao curso.

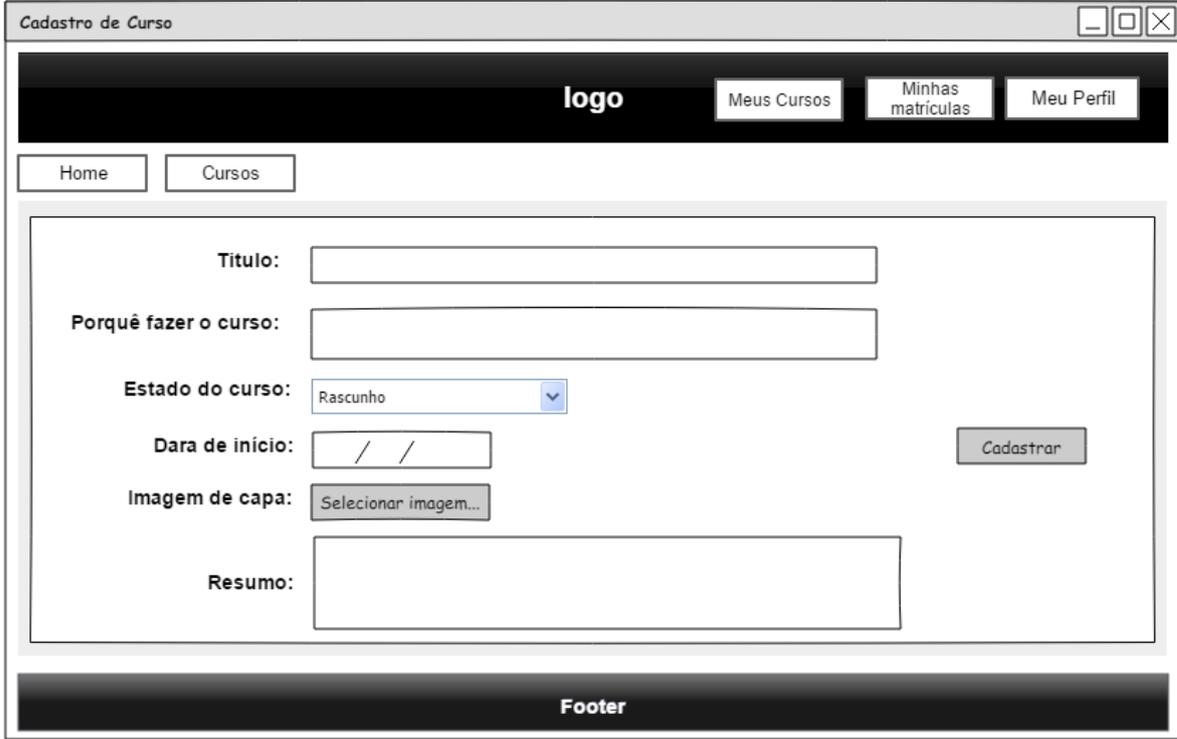
4a. Campo obrigatório não foi preenchido

1. O sistema informa que os campos obrigatórios devem ser preenchidos.

4b. Duplicidade de curso

1. O sistema informa que já existe um curso cadastrado de mesmo título.

Protótipo:



O protótipo de tela, intitulado "Cadastro de Curso", apresenta uma interface com uma barra superior contendo o "logo" e botões para "Meus Cursos", "Minhas matrículas" e "Meu Perfil". Abaixo, há uma barra de navegação com "Home" e "Cursos". O formulário principal contém os seguintes campos: "Titulo:" (campo de texto), "Porquê fazer o curso:" (campo de texto), "Estado do curso:" (menu suspenso com "Rascunho" selecionado), "Data de início:" (campo de data com delimitadores "/ /"), "Imagem de capa:" (botão "Selecionar imagem...") e "Resumo:" (campo de texto). Um botão "Cadastrar" está posicionado à direita dos campos de data e imagem. A barra inferior da tela é rotulada como "Footer".

Quadro 8. Expansão do caso de uso Gerenciar Videoaula

Gerenciar Videoaulas

Este caso de uso especifica as ações que um usuário executa no sistema com o objetivo de cadastrar e manter videoaulas no sistema. Somente usuário administrador poderá cadastrar cursos.

Ator:

Administrador

Pré-condição:

O usuário deve ter feito login e estar detalhando um curso que é administrador.

Pós-condição:

O sistema exibe as informações da videoaula cadastrada.

Requisitos Funcionais:

RF018 - Cadastrar videoaula

RF019 - Alterar videoaula

RF020 - Remover videoaula

RF009 - Assistir videoaula

RF002 - Carregar videoaula

Requisitos Não Funcionais:

RNF003 - Acesso Restrito

RNF009 - Duplicidade de videoaula

RNF007 - Integridade de dados

Fluxo de Básico:

1. Na tela do curso detalhado, o usuário clica em “Cadastrar videoaula”.
2. O sistema redireciona o usuário para a tela de cadastro de videoaula.
3. O sistema apresenta o formulário de cadastro de videoaula.
4. O usuário insere os dados como título da videoaula, curso da videoaula, descrição, e URL da videoaula e imagem de capa nos campos correspondentes.
5. O usuário clica no botão “Cadastrar”.
6. O sistema salva os dados no banco de dados.
7. O sistema informa que o cadastro foi realizado com sucesso.
8. O sistema exibe a tela do curso que a videoaula foi cadastrada.

Fluxo de Alternativo:

1a. Usuário quer alterar dados da videoaula

1. O usuário seleciona videoaula e clica na opção “Alterar”.
2. O Sistema carrega as informações da videoaula.
3. O usuário informa os dados a serem atualizados.
4. O usuário confirma as alterações.
5. O sistema salva as alterações.
6. Continua no passo 8.

1b. Usuário quer remover videoaula

1. O usuário seleciona a videoaula e clica na opção “Excluir”.
2. O sistema exclui todos os dados do banco referente ao curso.
3. Continua no passo 8.

4a. Campo obrigatório não foi preenchido

1. O sistema informa que os campos obrigatórios devem ser preenchidos.

4b. Duplicidade de curso

1. O sistema informa que já existe um curso cadastrado de mesmo título.

Protótipo:

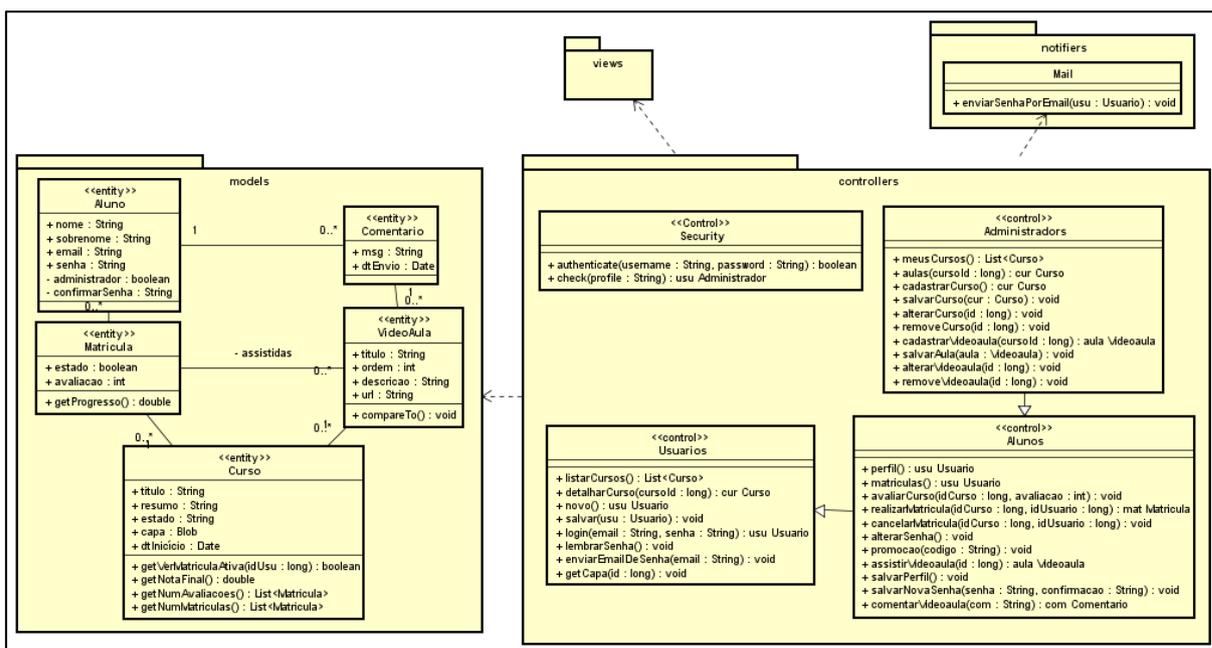


O protótipo de tela, intitulado "Cadastrar videoaula", apresenta uma interface com uma barra superior escura contendo o "logo" e três botões de navegação: "Meus Cursos", "Minhas matrículas" e "Meu Perfil". Abaixo, há uma barra de navegação com "Home" e "Cursos". O formulário principal contém os seguintes campos: "Título da videoaula:" (campo de texto), "Curso da videoaula:" (campo de texto), "Descrição:" (campo de texto maior), e "URL:" (campo de texto). À esquerda do formulário, há uma caixa de imagem com um "X" vermelho e um menu suspenso com o valor "1". Um botão "Cadastrar" está posicionado na base do formulário. Uma barra escura no rodapé contém o texto "Footer".

3.6 DIAGRAMA DE CLASSE

Seguindo o padrão MVC proposto pelo Play Framework, a arquitetura do sistema foi estabelecida em 4 camadas: um pacote Models, que contém 5 (cinco) classes com os componentes de negócio do sistema, neles são definidas as entidades e regras de negócio da aplicação; um pacote Controllers que contém 4 (quatro) classes de controle que possui todos os componentes que tratam e manipulam os dados enviados pela visão, funciona, basicamente, como intermediador entre visão e negócio do sistema; um pacote views com todos os componentes da interface utilizados pelos usuários, esses são, basicamente, arquivos HTML compostos com Tecnologia *Web* (HTML, CSS, e Javascript); por fim, um pacote notifiers com uma classe responsável por enviar as mensagens de e-mail da aplicação.

Figura 17. Diagrama de classe.



Fonte: Elaboração própria (2018)

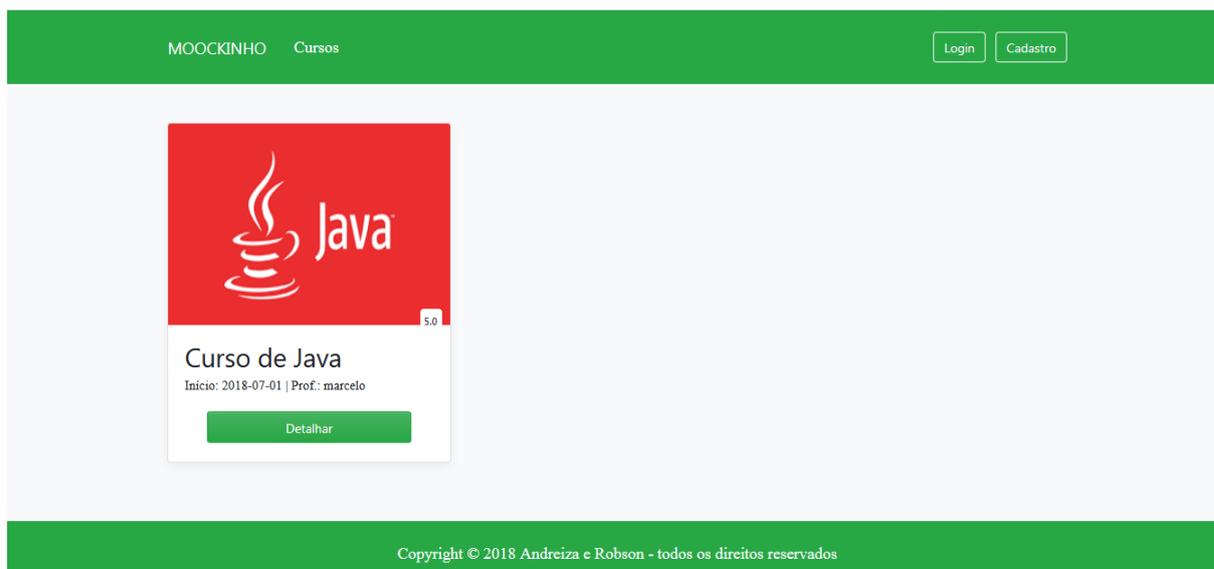
4 RESULTADOS

Nesta seção, é feita a apresentação das funcionalidades implementadas para o sistema proposto neste trabalho.

4.1 LISTAR CURSOS

Na tela de cursos é onde os usuários poderão ver todos os cursos cadastrados no sistema, como também detalhá-los.

Figura 18. Funcionamento do listar cursos.



Fonte: Elaboração própria (2018).

4.2 REALIZAR CADASTRO

Na tela de cadastro é onde o usuário preencherá os dados de registro informando seu nome, sobrenome, *email* e senha de registro para que possa se cadastrar no sistema e ter acesso à área de aluno. Uma vez preenchido os dados, o usuário clica no botão salvar, onde o cadastro será confirmado e o usuário será redirecionado para a tela de *login*.

Figura 19. Funcionamento do realizar cadastro.

The screenshot shows the registration page of the MOOCKINHO system. The header is green with the text 'MOOCKINHO Cursos' on the left and 'Login' and 'Cadastro' buttons on the right. The main content area is white and contains a registration form with the following fields: 'Nome:' with the value 'Marcelo', 'Sobrenome:' with 'Bittercount', 'E-mail:' with 'relfds@outlook.com', 'Senha:' with six dots, and 'Confirmar senha:' with six dots. A green 'Salvar' button is positioned below the form. At the bottom of the page, a green footer contains the text 'Copyright © 2018 Andreiza e Robson - todos os direitos reservados'.

Fonte: Elaboração própria (2018).

Caso o *email* informado pelo usuário já tenha cadastro, o sistema exibirá uma mensagem de erro.

Figura 20. Funcionamento do realizar cadastro para erro.

The screenshot shows the registration page of the MOOCKINHO system with an error message. The header is green with the text 'MOOCKINHO Cursos' on the left and 'Login' and 'Cadastro' buttons on the right. The main content area is white and contains a registration form. At the top of the form area, a pink error message box reads: 'Não foi possível realizar a operação, email já cadastrado.' The form fields are: 'Nome:' with 'Robson', 'Sobrenome:' with 'Edson', 'E-mail:' with 'relfds@outlook.com', 'Senha:' (empty), and 'Confirmar senha:' (empty). A green 'Salvar' button is positioned below the form. At the bottom of the page, a green footer contains the text 'Copyright © 2018 Andreiza e Robson - todos os direitos reservados'.

Fonte: Elaboração própria (2018).

4.3 REALIZAR LOGIN

Para acessar o sistema o usuário deve informar seu *email* de *login* e senha. Caso um dos itens citados for informado errado, o sistema apresentará uma mensagem de erro e negará o acesso do usuário ao sistema.

Figura 21. Funcionamento do login para acesso negado.

The screenshot shows the Mookinho website's login interface. At the top, there is a green navigation bar with the text "MOOCKINHO Cursos" on the left and "Login" and "Cadastro" buttons on the right. The main content area is white and features a red error message box at the top that reads "Oops, senha ou usuário inválido(s)". Below this, there are two input fields: "E-mail:" with the value "relfds@outlook.com" and "Senha:". A green "Entrar" button is positioned below the password field, and a grey "Esqueci minha senha" button is below it. At the bottom of the page, a green footer contains the text "Copyright © 2018 Andreiza e Robson - todos os direitos reservados".

Fonte: Elaboração própria (2018).

Caso *email* de login e senha sejam informados corretamente, o sistema encaminhará o usuário para a tela inicial e exibirá menu do tipo do usuário logado.

Figura 22. Funcionamento do login para acesso autorizado.

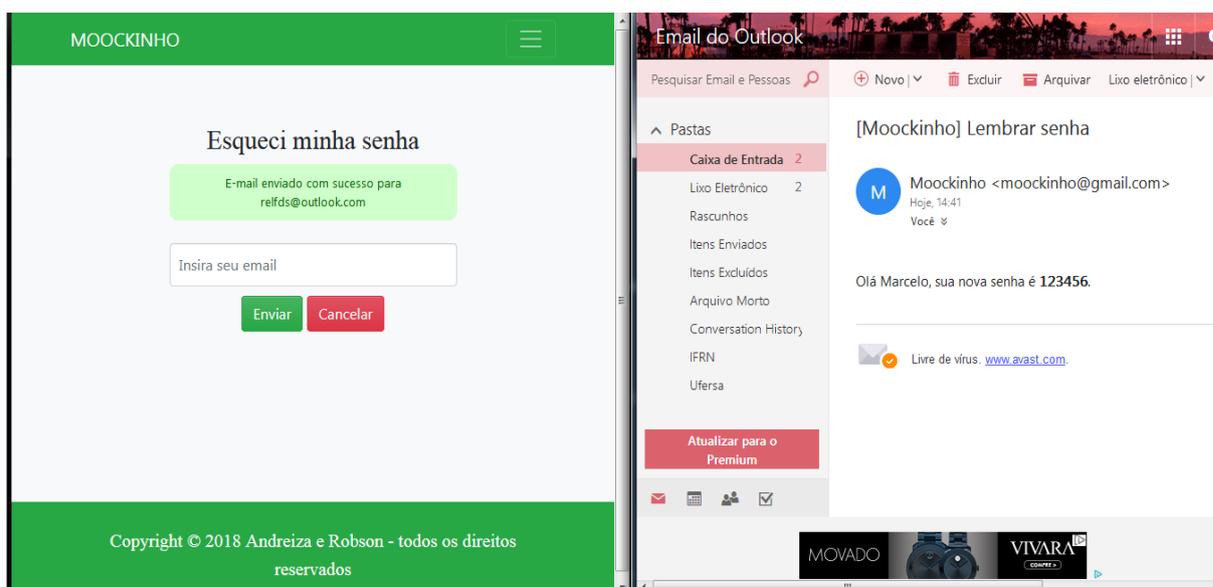
The screenshot displays the Mookinho home page for an authorized user. The green navigation bar at the top includes "MOOCKINHO Cursos" and buttons for "Meu Perfil", "Minhas Matriculas", and "Sair". The main content area is white and features the heading "Mookinho" followed by a paragraph: "O Mookinho é um software educacional com objetivo de disponibilizar cursos online, livres e gratuitos. Através deste sistema os usuários poderão se inscrever nos cursos, ter acesso as videoaulas e acompanhar seu progresso de aprendizagem." Below this text are three columns, each with an illustration and a text box. The first column, titled "Software", shows a mobile app and a crane, with text: "O sistema mookinho é dividido em 3 módulos: público, aluno e administrativo. Cada módulo disponibiliza certas funcionalidades do sistema para o usuário." The second column, titled "Cursos", shows silhouettes of people with lightbulbs, with text: "Os cursos oferecidos pelo sistema são online, livres e gratuitos. Os usuários do software poderão se inscrever nos cursos e ter acesso as videoaulas no sistema." The third column, titled "MOOC", shows a laptop with a world map and the text "MOOC", with text: "MOOC é a sigla para Massive Open Online Courses – cursos ministrados a distância, pela internet, e que suportam um grande número de alunos." At the bottom, a green footer contains the text "Copyright © 2018 Andreiza e Robson - todos os direitos reservados".

Fonte: Elaboração própria (2018).

4.4 RECUPERAR SENHA

Caso o usuário possua registro no sistema, porém não recorde sua senha de acesso, ele pode solicitar o envio do lembrete de senha. Para isso basta informar o *email* utilizado no cadastro.

Figura 23. Funcionamento do recuperar senha.



Fonte: Elaboração própria (2018).

Caso o *email* inserido não tiver cadastro, o sistema apresentará uma mensagem de erro.

Figura 24. Funcionamento do recuperar senha usuário não encontrado.



Fonte: Elaboração própria (2018).

4.5 DETALHAR CURSO

A tela de detalhar curso permitirá aos usuários verem as informações dos cursos cadastrados no sistema.

Figura 25. Funcionamento do detalhar curso para usuário avulso.



Fonte: Elaboração própria (2018).

Caso o usuário esteja logado no sistema e não tenha matrícula ativa no curso, ele poderá realizar matrícula.

Figura 26. Funcionamento do detalhar curso para usuário não inscrito.



Fonte: Elaboração própria (2018).

Caso o usuário tenha matrícula ativa no curso, ele poderá ver o seu progresso.

Figura 27. Funcionamento do detalhar curso para usuário inscrito.



Fonte: Elaboração própria (2018).

4.6 MEU PERFIL

A tela de meu perfil é onde os usuários com cadastro no sistema poderão fazer alterações nos seus dados de registro, como por exemplo, nome, sobrenome e *email*. Além disso, os usuários poderão ter acesso às telas de modificar sua senha de registro e de solicitar promoção a administrador, caso o usuário seja um usuário aluno.

Figura 28. Funcionamento do meu perfil.

MOOCKINHO Cursos

Meu Perfil Minhas Matriculas Sair

Perfil

Nome:

Sobrenome:

E-mail:

[Salvar alterações](#) [Modificar senha](#)

[Solicitar promoção a administrador](#)

Copyright © 2018 Andreiza e Robson - todos os direitos reservados

Fonte: Elaboração própria (2018).

4.7 MODIFICAR SENHA

Na tela de alterar senha o usuário cadastrado poderá modificar sua senha de registro, informando uma nova senha e a confirmando.

Figura 29. Funcionamento do Alterar senha.

MOOCKINHO Cursos

Meu Perfil Minhas Matriculas Sair

Alterar Senha

[Alterar](#) [Cancelar](#)

Copyright © 2018 Andreiza e Robson - todos os direitos reservados

Fonte: Elaboração própria (2018).

Caso um dos itens citados for informado errado, o sistema apresentará uma mensagem de erro e negará a mudança de senha.

Figura 30. Funcionamento do Alterar senha para senhas que não correspondem.

MOOCKINHO Cursos Meu Perfil Minhas Matrículas Sair

Alterar Senha

As senhas não correspondem.

Nova senha

Confirmar nova senha

Alterar Cancelar

Copyright © 2018 Andreiza e Robson - todos os direitos reservados

Fonte: Elaboração própria (2018).

4.8 SOLICITAR PROMOÇÃO A ADMINISTRADOR

Sendo o usuário um usuário aluno, ele poderá solicitar promoção a um usuário administrador informando um código válido no sistema.

Figura 31. Funcionamento do Solicitar promoção a administrador.

MOOCKINHO Cursos Meu Perfil Minhas Matrículas Sair

Solicitar Promoção

Código:

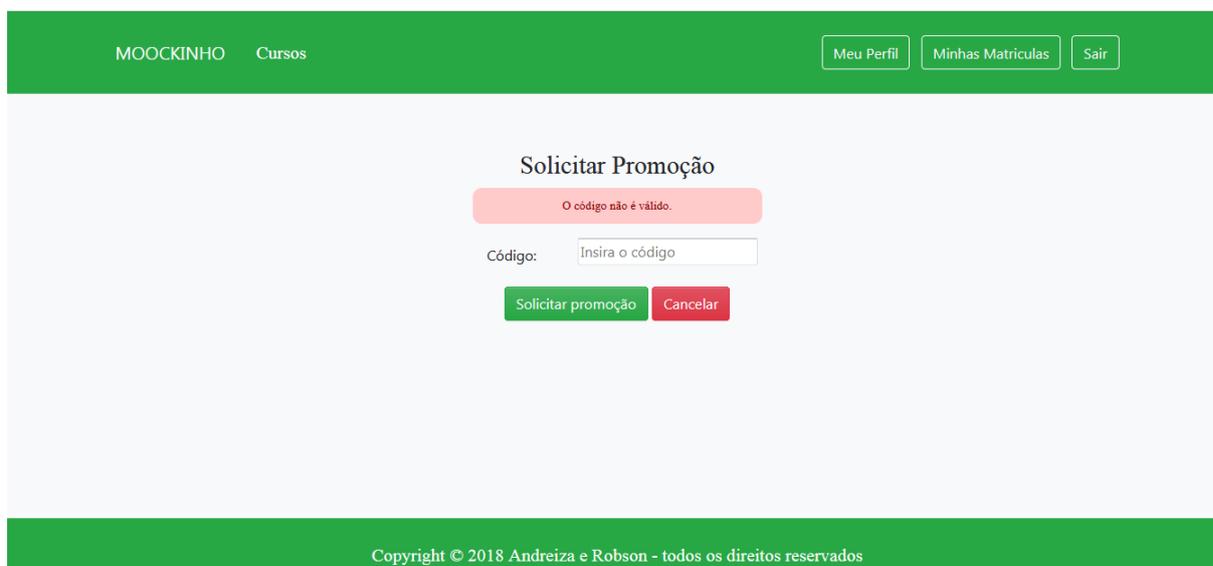
Solicitar promoção Cancelar

Copyright © 2018 Andreiza e Robson - todos os direitos reservados

Fonte: Elaboração própria (2018).

Caso o usuário aluno informe um código inválido, o sistema apresentará uma mensagem de erro e negará a promoção.

Figura 32. Funcionamento do Solicitar promoção a administrador código inválido.

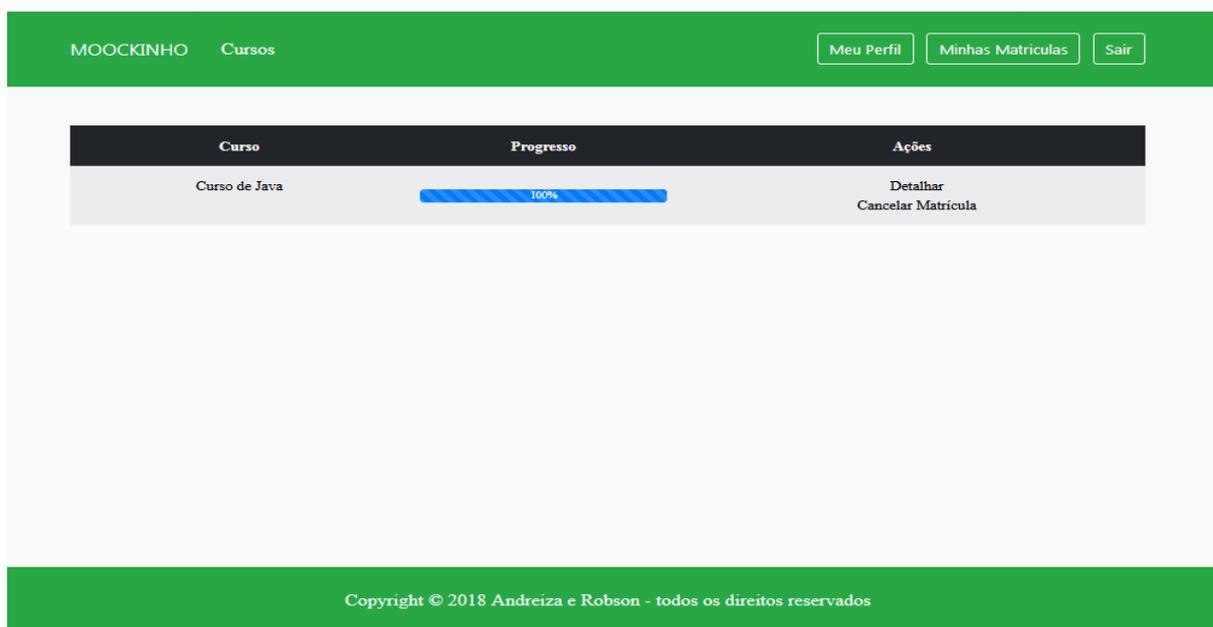


Fonte: Elaboração própria (2018).

4.9 MINHAS MATRÍCULAS

Na tela de minhas matrículas o usuário cadastrado poderá acompanhar o progresso dos cursos que tenha matrícula ativa, assim como detalhá-los e cancelar matrícula.

Figura 33. Funcionamento de minhas matrículas.

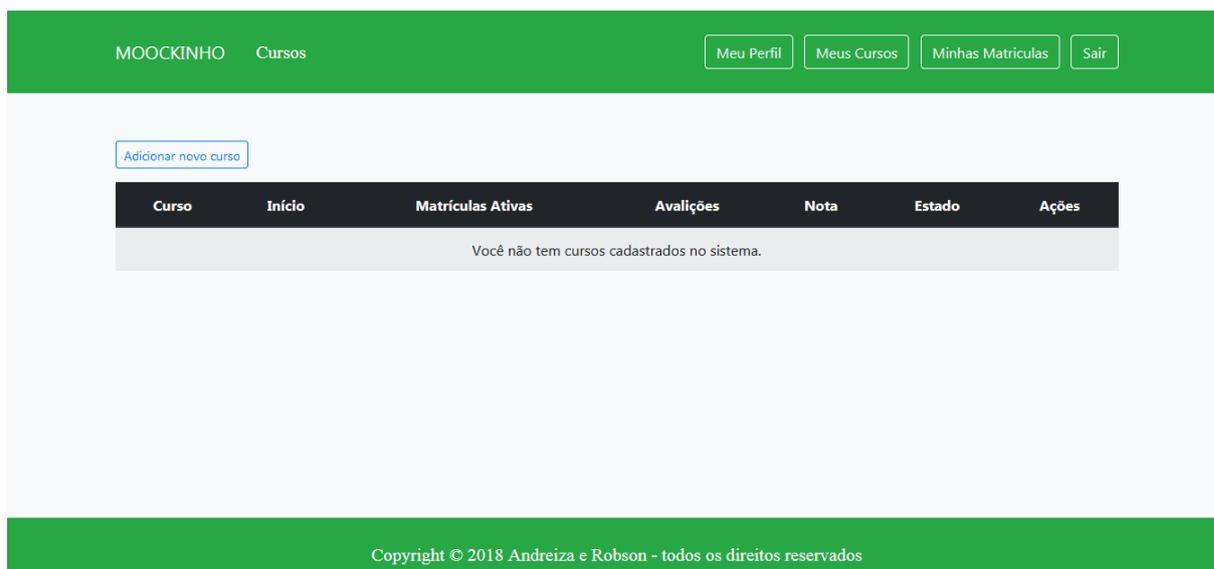


Fonte: Elaboração própria (2018).

4.10 GERENCIAR CURSO

Sendo o usuário um usuário administrador, ele terá acesso a tela de meus cursos. Nessa tela o usuário pode cadastrar, alterar e excluir cursos de sua autoria no sistema.

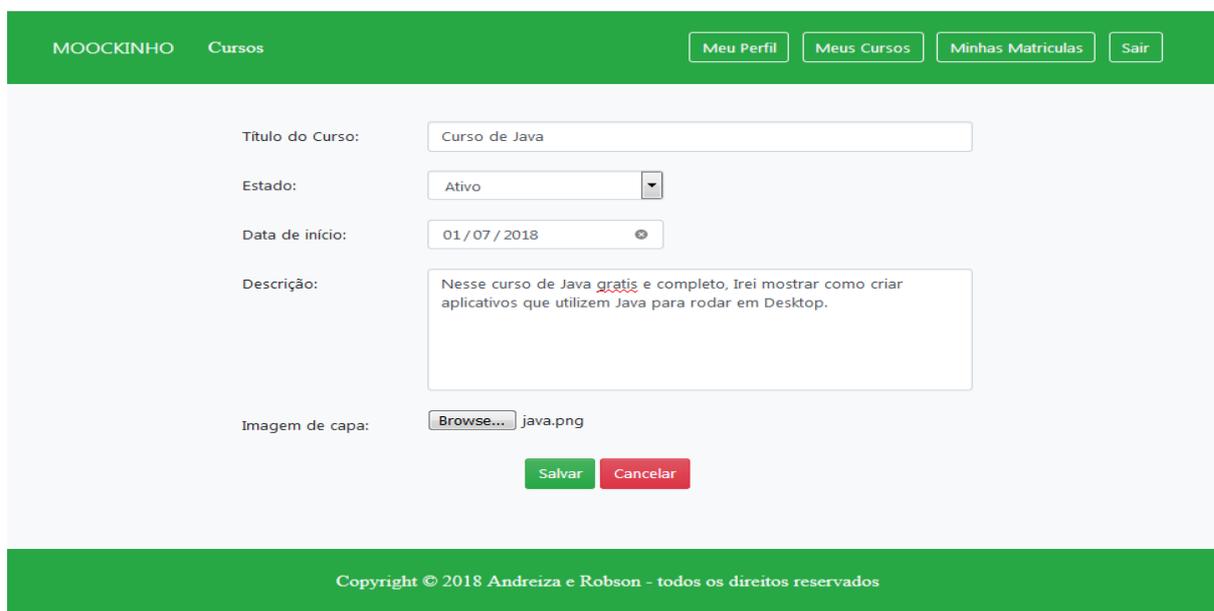
Figura 34. Funcionamento do gerenciar curso.



Fonte: Elaboração própria (2018).

É nessa tela que o usuário administrador poderá cadastrar os seus cursos, clicando no botão de adicionar novo curso ele será redirecionado a tela de cadastramento e terá que preencher o formulário com as informações do curso.

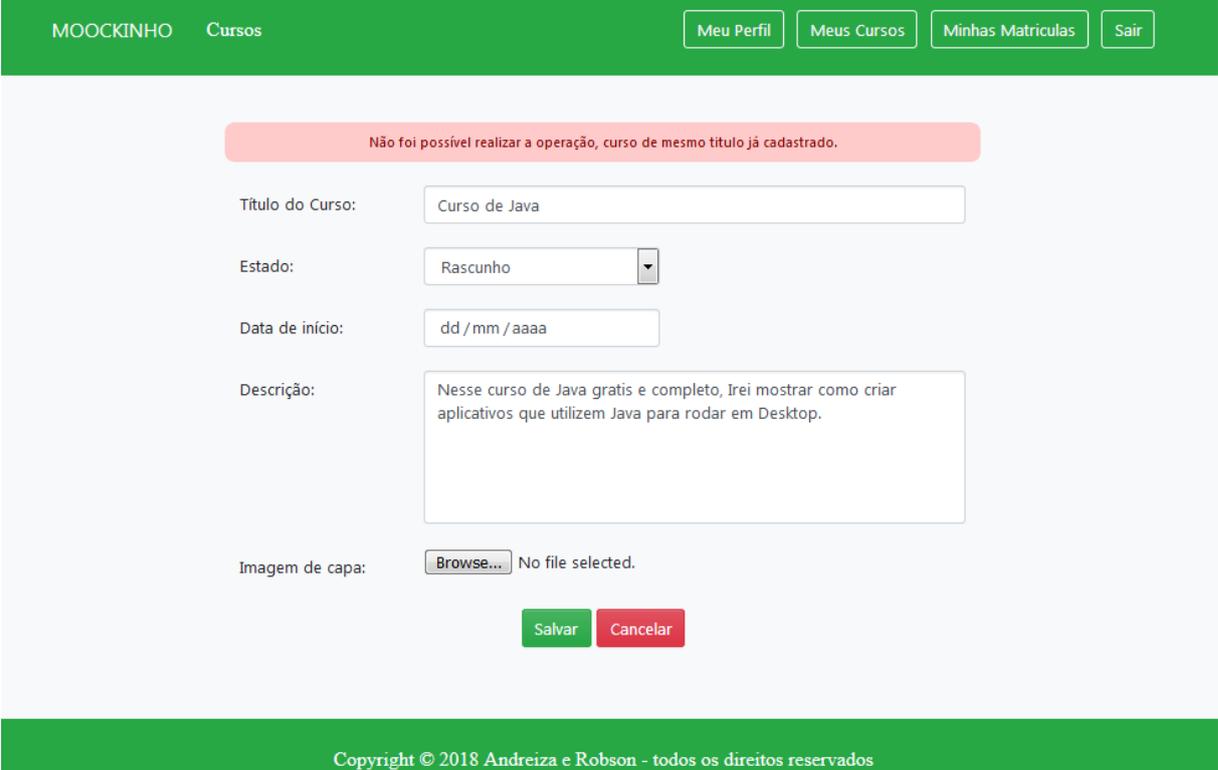
Figura 35. Funcionamento do cadastro de curso.



Fonte: Elaboração própria (2018).

Caso o título do curso informado pelo usuário administrador já tenha cadastro, o sistema exibirá uma mensagem de erro.

Figura 36. Funcionamento do cadastro de curso erro.



The screenshot shows a web interface for course registration. At the top, there is a green navigation bar with the text "MOOCKINHO Cursos" on the left and four buttons: "Meu Perfil", "Meus Cursos", "Minhas Matriculas", and "Sair". Below the navigation bar, a red error message is displayed in a rounded rectangle: "Não foi possível realizar a operação, curso de mesmo titulo já cadastrado." The form fields are as follows: "Título do Curso:" with a text input containing "Curso de Java"; "Estado:" with a dropdown menu showing "Rascunho"; "Data de início:" with a date input field showing "dd/mm/aaaa"; "Descrição:" with a text area containing the text "Nesse curso de Java gratis e completo, irei mostrar como criar aplicativos que utilizem Java para rodar em Desktop."; and "Imagem de capa:" with a "Browse..." button and the text "No file selected." At the bottom of the form, there are two buttons: "Salvar" (green) and "Cancelar" (red). The footer of the page is a green bar with the text "Copyright © 2018 Andreiza e Robson - todos os direitos reservados".

Fonte: Elaboração própria (2018).

4.11 GERENCIAR VIDEOAULAS

Sendo o usuário um usuário administrador, ele terá acesso a tela de aulas. Nessa tela o usuário pode cadastrar, alterar e excluir videoaulas do curso, como também pode assisti-las.

Figura 37. Funcionamento do gerenciar videoaulas.

The screenshot shows a web interface for managing video lessons. At the top, a green navigation bar contains the text 'MOCKINHO Cursos' on the left and four buttons: 'Meu Perfil', 'Meus Cursos', 'Minhas Matrículas', and 'Sair'. Below this, the main content area is titled 'Curso de Java' with a subtitle 'Data de Início: 2018-07-01 | Avaliação: 0.0'. On the left, there is a red square with the Java logo (a white coffee cup with steam) and the word 'Java' in white. Below the logo, a white box contains the text: 'Nesse curso de Java gratis e completo, Irei mostrar como criar aplicativos que utilizem Java para rodar em Desktop.' To the right of the logo, the text 'Aulas' is displayed, followed by 'O curso não tem aulas cadastradas'. A blue button labeled 'Adicionar Videoaula' is positioned in the top right corner of the main content area. At the bottom, a green footer bar contains the text 'Copyright © 2018 Andreiza e Robson - todos os direitos reservados'.

Fonte: Elaboração própria (2018).

É nessa tela que o usuário administrador poderá cadastrar as suas aulas no curso, clicando no botão de adicionar nova videoaulas, ele será redirecionado a tela de cadastramento e terá que preencher o formulário com as informações da videoaula.

Figura 38. Funcionamento do cadastro da videoaula.

The screenshot shows a web interface for a course named "CURSO DE JAVA". At the top, there is a navigation bar with "MOOCKINHO" and "Cursos" on the left, and buttons for "Meu Perfil", "Meus Cursos", "Minhas Matriculas", and "Sair" on the right. The main content area contains a form with the following fields: "Curso: CURSO DE JAVA", "Título da Vídeoaula:" with the value "História do Java", "Ordem:" with the value "1", "Descrição:" with the value "Aprenderemos sobre a história do Java.", and "URL:" with the value "https://youtu.be/sTX0UEpIF54". Below the form are two buttons: "Salvar" (green) and "Cancelar" (red). At the bottom of the page, there is a green footer with the text "Copyright © 2018 Andreiza e Robson - todos os direitos reservados".

Fonte: Elaboração própria (2018).

Caso o URL da videoaula informado pelo usuário administrador já esteja cadastrado no curso, o sistema exibirá uma mensagem de erro.

Figura 39. Funcionamento do cadastro de videoaula erro.

The screenshot shows the same web interface as Figure 38, but with an error message displayed at the top in a pink box: "Não foi possível realizar a operação. Vídeoaula de mesmo URL já cadastrada." The form fields are: "Curso: CURSO DE JAVA", "Título da Vídeoaula:" with the value "Introdução", "Ordem:" with the value "2", "Descrição:" with the value "Aprenderemos o conceito de java e muito mais.", and "URL:" with the value "https://youtu.be/sTX0UEpIF54". The "Salvar" and "Cancelar" buttons are still present at the bottom. The footer at the bottom of the page is the same as in Figure 38: "Copyright © 2018 Andreiza e Robson - todos os direitos reservados".

Fonte: Elaboração própria (2018).

4.12 ASSISTIR VIDEOAULA

A tela de assistir videoaula é onde os usuários com cadastro no sistema e matrícula ativa no curso poderão assistir às aulas referentes a aquele curso. Além disso, o usuário poderá fazer comentários sobre a aula.

Figura 40. Funcionamento do assistir videoaula.

The screenshot shows the user interface for watching a video lesson. At the top, a green navigation bar contains the text 'MOOCKINHO Cursos' and three buttons: 'Meu Perfil', 'Minhas Matrículas', and 'Sair'. Below this, the page title is 'Curso de Java' and the lesson title is 'Aula 1 - História do Java'. The main content area features a video player with a colorful thumbnail. The thumbnail includes the text 'Curso de Java #01 - História do Java - Gustavo Guanabara', 'História do Java', '01', 'PARA INICIANTES', 'Java', and a gold award badge that says 'O SEU MELHOR CURSO DE JAVA YouTube'. To the right of the video player is a list item with a blue circle containing the number '1' and the text 'História do Java - Aprenderemos sobre a história do Java.' Below the video player is a text input field and two buttons: 'Comentar' (green) and 'Cancelar' (red). At the bottom of the page, a green footer contains the text 'Copyright © 2018 Andreiza e Robson - todos os direitos reservados'.

Fonte: Elaboração própria (2018).

5 CONCLUSÃO

Ao final deste trabalho foi desenvolvido um sistema *web* que propõem alcançar os objetivos aqui estabelecidos. Foi desenvolvida uma ferramenta de apoio à aprendizagem, voltada para divulgação e gerenciamento de cursos online abertos e massivos. O Moockinho – sistema desenvolvido - foi devidamente projetado para disponibilizar cursos online, livre e gratuitos aos seus usuários. O sistema foi elaborado, projetado, implementado e testado. Todos os casos de uso, com exceção do de realizar comentários, foram testados e estão totalmente funcionais.

5.1 TRABALHOS FUTUROS

Para futuras versões do sistema são sugeridas as seguintes otimizações:

- a) A aplicação de um sistema mobile de notificações a partir do qual o aluno e o professor possam receber em tempo real informativos de seu interesse, por exemplo, alertas de adição de cursos ou videoaulas.
- b) Implementar funcionalidade de interação inter-usuários (fóruns, chat e mensagem).
- c) Divulgar o sistema na comunidade interna do IFRN.

REFERÊNCIAS

AQUILES, Alexandre e Rodrigo Ferreira. **Controlando Versões com Git e GitHub**. 1ª Edição. São Paulo. Casa do Código. 2014.

BAUER, Christian et al. **Java Persistence com Hibernate**, Rio de Janeiro.

CHACON, S.; STRAUB, B. **Pro git**, Apress, 2009.

CREED-DIKEOGU, G. E CLARK, C. **Are you MOOC-ing yet?** A review for academic.

DOEDERLEIN, Osvaldo Pinali. **Dados e Mapeamento explorando técnicas e tecnologias para persistência de dados**. JAVA magazine, Ed.42, ano.V, p. 22-30, 2006.

DA MATTA, Cláudia Eliane; FIGUEIREDO, Ana Paula Silva. **Mooc**: transformação das práticas de aprendizagem. In: X Congresso Brasileiro de Ensino Superior a Distância. 2013.

EMERGING ELEARNING TECHNOLOGIES AND APLICATIONS, 10., 2012, Stará.

GUERRA, Rafael Laurino; ZAINA, Luciana Aparecida Martinez. Mapeamento objeto relacional: um estudo de caso utilizando hibernate. **Reverte-Revista de Estudos e Reflexões Tecnológicas da Faculdade de Indaiatuba**, n. 6, 2008.

INSTITUTO TIM; **Portal TIM Tec - MOOC**. [2017] Disponível em: <<http://timtec.com.br/pt/>>. Acesso em: 25 jun. 2018.

KOP, R., FOURNIER, H. E MARK, J. **A pedagogy of abundance or pedagogy to support human beings?** Participant support on massive open online courses. The international review of research in Open & Distance learning, v. 12, n.7, p. 74-93, 2011. Disponível em: <<http://www.irrodl.org/index.php/irrodl/article/view/1041/2025>>.

Acesso em: 25 junho. 2018.

Lesná Anais. Stará Lesná, IEEE, 2012.p. 297-300.

Libraries. **CULS Proceedings**, vol 3, 2013.

LITTO, F. M. **O atual cenário internacional da EAD**. In: LITTO, F. M.; FORMIGA, M. M. M. (orgs.). Educação a distância: o estado da arte. São Paulo: Pearson Education do Brasil, 2009a, cap. 3, p.14-20.

MOTA, R.; INMORATO, A. **MOOC, uma revolução em curso**. *Jornal da ciência*, nov. [2012]. Disponível em: <<http://www.jornaldaciencia.org.br/Detailhe.jsp?id=85111>>. Acesso em: 25 jun. 2018.

PENATTI Giovana. **Plataforma brasileira usa vídeos para ensinar a programar gratuitamente**. [2014]. Disponível em: <<https://tecnoblog.net/161403/plataforma-brasileira-usa-videos-ensinar-a-programar-gratuitamente/>>. Acesso em: 25 jun. 2018.

PISUTOVA, K. **Open education**. In: IEEE INTERNATIONAL CONFERENCE ON.

REIS, Rodrigo Quites. **Usando UML para Especificação de sistema Orientados a Objetos**. 2003.

Silva, M. S. (2015). **Bootstrap 3.3.5**. Novatec Editora, São Paulo, SP.

SOUSA, Gilbert Firmino Carvalho de et al. **WebDiet: um estudo de caso de uso do uso de Templates em Play Framework**. 2015.

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. **UML**. [2018]. Disponível em: <https://pt.wikipedia.org/wiki/UML#Hist%C3%B3ria_da_UML>. Acesso em: 25 jun. 2018.

ZENEXTY; TYPESAFE; **Play 1.3.x Documentation**. [200?] Disponível em: <<https://www.playframework.com/documentation/1.3.x/home>>. Acesso em: 25 jun. 2018.

_____ ; **Secure - 1.2.5**. [200?] Disponível em: <<https://www.playframework.com/documentation/1.2.5/secure>>. Acesso em: 25 jun. 2018.

ZIMMERMANN, T.; ZELLER, A.; WEISSGERBER, P.; DIEHL, S. **Mining version histories to guide software changes**. In: IEEE Transactions on Software Engineering, 31(no.6)(2005): 429-445.