

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE
CAMPUS AVANÇADO LAJES
CURSO TÉCNICO INTEGRADO EM INFORMÁTICA

ANA SARA DA SILVA CLEMENTINO

**ITARETIF - UM PORTAL PARA DIVULGAÇÃO DE INFORMAÇÕES NO
CAMPUS LAJES**

LAJES/RN
2021

ANA SARA DA SILVA CLEMENTINO

**ITARETIF - UM PORTAL PARA DIVULGAÇÃO DE INFORMAÇÕES NO
CAMPUS LAJES**

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

Orientador (a): Prof. Me. Fernando Helton Linhares Soares

AGRADECIMENTOS

Agradeço aos meus familiares e amigos por todo o apoio.

Agradeço ao IFRN em geral por me proporcionar uma experiência inesquecível.

Por fim, agradeço aos nossos orientadores e aos demais integrantes do grupo que contribuíram para a construção deste projeto.

RESUMO

Após a aprovação no processo seletivo do Instituto Federal de Educação, Ciência e Tecnologia, os estudantes ingressam na instituição sem conhecer seu funcionamento e suas instalações, quais são as normas de conduta exigidas para eles e, tanto aos novos ingressantes quanto ao público externo, existe um déficit de comunicação no que toca novos projetos e informes da instituição. A maioria dessas informações podem ser encontradas no portal oficial da instituição, porém, a linguagem utilizada nos documentos e a quantidade de dados contidos nele, dificultam o entendimento e a pesquisa das normas, fazendo com que a página da instituição não seja atrativa para o público. Neste sentido, este projeto propõe a criação de um portal capaz de abrigar as informações mais solicitadas pelos estudantes do Campus Avançado Lajes, com o objetivo de proporcionar aos alunos, assim como à população externa, informações sobre a instituição utilizando-se de uma linguagem mais empregada no cotidiano das pessoas, tornando o portal atrativo e de fácil entendimento.

Palavras-chave: informática, informação, campus, Lajes, instituição, portal.

ABSTRACT

After approval in the selection process of the Instituto Federal de Educação, Ciência e Tecnologia, students enter the institution without knowing its operation and facilities, what are the standards of conduct required for them and, both for new entrants and the external public, there is a communication deficit regarding new projects and reports from the institution. Most of this information can be found on the institution's official website, however, the language used in the documents and the amount of data contained on that website make it difficult to understand and search the rules, making the institution's website unattractive to the public. In this sense, this project proposes the creation of a portal capable of housing the information most requested by the students entering the Campus Avançado Lajes, with the objective of providing students, as well as the external population, with information about the institution using a language that is more used in people's daily lives, making the portal attractive and easy to understand.

Keyword: Computing, information, Campus, Lajes, institution, Portal.

LISTA DE ABREVIATURAS E SIGLAS

HTML	<i>Hypertext Markup Language</i> (Linguagem de marcação de hipertexto)
CSS	<i>Cascading Style Sheets</i> (Folhas de Estilo em Cascata)
SI	Sistema de Informação
UML	Unified Modeling Language (Linguagem Unificada de Modelagem)
RN	Rio Grande do Norte

SUMÁRIO

1 INTRODUÇÃO	8
1.1 OBJETIVO	9
1.2 JUSTIFICATIVA	9
2 DADOS GERAIS DA PESQUISA/EXTENSÃO/ESTÁGIO	10
3 FUNDAMENTAÇÃO TEÓRICA	12
3.1 HTML e CSS	12
3.2 Bootstrap 7	13
3.3 Eclipse	13
3.4 Spring	14
3.5 Github	14
3.6 MySQL	15
3.7 UML	15
4 METODOLOGIA	16
5 ANÁLISE E PROJETO DO SISTEMA PROPOSTO	19
5.1 APLICAÇÃO DO QUESTIONÁRIO	19
5.2 REQUISITOS E CASOS DE USO	22
5.3 DESENVOLVENDO DIAGRAMA DE CLASSES	24
6 PORTAL ITARETIF	26
6.1 FUNCIONALIDADES DESENVOLVIDAS	27
6.2 IMPLEMENTAÇÃO	38
7 ANÁLISE E DISCUSSÃO DOS RESULTADOS	46
8 CONSIDERAÇÕES FINAIS	47
REFERÊNCIAS	48
ANEXO A – FORMULÁRIO DE IDENTIFICAÇÃO	52

1 INTRODUÇÃO

O Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, assim como os demais institutos do país, conta com um portal que tem por finalidade publicar informações, como notícias sobre determinados assuntos das instituições, comunicados mais pertinentes sobre os campi, processos seletivos, concursos públicos, projetos e a difusão de determinados temas e eventos.

Dentro desse portal, cada Campus tem a sua página específica, para que as publicações sejam organizadas de forma mais particular em relação às postagens gerais. Porém, tais publicações acabam não recebendo o devido destaque, ou apresentando certas dificuldades de acesso para quem está visitando o *site*, o que limita bastante a transmissão de informações básicas - como, por exemplo, normas, estrutura física, localidade e setores do Campus - para o público externo e para os calouros que irão ingressar nas suas respectivas instituições.

Além disso, trata-se de uma página que segue fielmente todas as normas gramaticais da língua portuguesa, tornando a formalidade uma característica visível na estrutura do Portal, seja na escrita dos textos, seja na disponibilidade dos conteúdos, o que resulta em pouca atratividade e, apesar de não ser um grande obstáculo, torna o processo de interação entre o visitante e o portal demasiadamente cansativo.

Pensando nessa problemática, surgiu a ideia inicial do ItaretIF, que tem como objetivo disponibilizar um portal exclusivo para o público externo e interno do Campus Avançado Lajes, principalmente os calouros e visitantes interessados em conhecer o lugar. O portal, além de todas as informações básicas do Campus em questão, contém detalhes sobre o desenvolvimento e o funcionamento da instituição de ensino, que veio a ser umas das maiores conquistas para a cidade de Lajes e região. Como já citado neste tópico, as informações publicadas no portal serão exclusivamente sobre o Campus, apresentando uma linguagem mais utilizada no cotidiano da população lajense e, portanto, menos formal, oferecendo conforto e facilidade de navegação aos usuários.

Esta prática tem como finalidade o crescimento do Campus como instituição,

a disponibilidade de recursos mais acessíveis para conhecimento sobre o que acontece na comunidade escolar e, acima de todos os pontos, o foco principal: disponibilizar aos alunos ingressantes no Campus Lajes todas as informações necessárias sobre o polo.

Ao decorrer deste relatório serão apresentadas a caracterização da prática profissional realizada, sua fundamentação teórica, a metodologia utilizada no trabalho, o detalhamento das atividades desenvolvidas, a análise dos resultados alcançados e, por fim, as considerações finais.

1.1 OBJETIVO

Disponibilizar e facilitar o acesso às informações sobre o Campus Avançado Lajes com uma linguagem familiar aos usuários, os ingressantes do Campus em questão e o público externo. Para isso, o Portal ItaretIF trará uma linguagem que considera um contexto mais jovem e regional, de forma clara e objetiva, além de apresentar leveza em sua configuração visual. Esse portal irá disponibilizar, também, as normas e regulamentos, informações sobre os setores e servidores, sobre o município e sobre a instituição em geral.

1.2 JUSTIFICATIVA

Ao ingressar no Instituto Federal - Campus Lajes, os estudantes não sabem como funciona a instituição e sentem dificuldade em encontrar essas informações, como, por exemplo, quais são as regras do IFRN, quais as normas de fardamento, quais são os setores do prédio principal e quem são os profissionais que atuam em cada um deles, e qual caminho fazer para chegar até um determinado setor quando necessitar de algum tipo de atendimento. O desconhecimento dessas informações pode fazer com que o aluno cometa faltas disciplinares, e, também, perca prazos de requerimento, por exemplo, o prazo para justificativa de ausências.

2 DADOS GERAIS DA PESQUISA/EXTENSÃO/ESTÁGIO

Título do projeto: ItaretIF - Um portal para divulgação de informações do campus Lajes

Período de realização: de 02/12/2019 a 30/01/2021

Total de horas: mínimo de 550 horas.

Orientador: Fernando Helton Linhares Soares

Nome do co-orientador: Dannylo Johnathan Bernardino Egidio

Função: Aluno Voluntário

Formação profissional: Técnico em Informática

Abaixo seguem as atividades desenvolvidas ao longo do período de realização do projeto destacando a carga horária empenhada e os resultados alcançados em cada uma das atividades.

Quadro 1 – Síntese das Atividades do Aluno no Projeto.

CARGA HORÁRIA	ATIVIDADES DESENVOLVIDAS	RESULTADOS ALCANÇADOS
30	Levantamento de Requisitos	Todos os requisitos funcionais do portal foram levantados e o diagrama de caso de uso foi desenvolvido.
60	Projeto de arquitetura do sistema	Desenvolvimento do diagrama de classe e do diagrama de sequência;

400	Implementação do sistema	Finalização da implementação do portal proposto com todos os requisitos funcionais levantados anteriormente ;
30	Escrita do relatório de atividades	Relatório final concluído
30	Reuniões do acompanhamento do projeto	As reuniões ocorriam nas salas da própria instituição, mas, devido a pandemia, passaram a ser realizadas por vídeo chamadas através do <i>Google meets</i> .

Fonte: autoria própria (ano).

3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentadas as tecnologias utilizadas para o desenvolvimento do portal.

3.1 HTML e CSS

O HTML é um conjunto de tags responsáveis pela definição da estrutura do conteúdo de uma página no navegador. As tags são marcações que definem os elementos que compõem as páginas.

[...] HTML é usada para criar páginas web. Veremos que você começa escrevendo as palavras que quer que apareçam na página. Então adicione tags ou elementos às palavras de modo que o navegador saiba o que é um título, onde um parágrafo começa e termina etc. (DUCKETT, 2011, p. 6).

O *Cascading Style Sheets* (CSS) é a linguagem de estilos, utilizada para a implementação de páginas, em que desempenha alteração de fonte, cor, layout e outros.

Cascading Style Sheets (ou simplesmente CSS) é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento. Ao invés de colocar a formatação dentro do documento, o desenvolvedor cria um link (ligação) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta portanto modificar apenas um arquivo. Cabem a CSS todas as funções de apresentação de um documento e ao HTML todas as funções de marcação e estruturação de conteúdos. Uma página web é composta então de CSS + HTML. (MEIRA, 2013, p. 4).

3.2 BOOTSTRAP 7

Segundo Alvim (2010, p. 12), “O *framework* é um conjunto de classes que colaboram entre si, proporcionando melhores práticas de desenvolvimento e diminuição da repetição de tarefas.” O Bootstrap é um *framework* que ajuda no desenvolvimento rápido e seguro de aplicações que utilizam HTML, CSS e JavaScript. Essa ferramenta é gratuita e possui um conjunto de componentes pré-construídos que podem ser reutilizados na aplicação. Com ele é possível criar páginas responsivas de maneira simples e padronizada. Dessa forma, Alvim (2010, p.12) discorre que o *framework* evita variações de soluções diferentes para um mesmo tipo de problema. O que facilita a reutilização e customização dos códigos.

3.3 ECLIPSE

O Eclipse é um Ambiente de Desenvolvimento Integrado (IDE) que promove um ambiente de desenvolvimento Java, capaz de integrar diversas tecnologias e facilitar o desenvolvimento de aplicações. Existem diversos *plugins* que podem estender as funcionalidades da ferramenta adaptando esta para as necessidades específicas de cada projeto.

A plataforma do Eclipse fornece vários pacotes de desenvolvimento, tais como Eclipse JDT, que é a base para qualquer plug-in na linguagem java, o Eclipse SDK, que é o pacote de distribuição da IDE java, o Eclipse WTP (*Web Tools Platform*), que é usado para desenvolvimento de linguagem para web, e o compilador do JDT, que é seu próprio compilador java, que é mais rápido e de código aberto. (FARIA; LIMA; DIAS; SILVA; COSTA; BITTAR, 2010, p. 4).

3.4 SPRING

O Spring é um *framework* que abarca vários Projetos Springs que facilitam a criação e configuração de aplicações java. Alguns projetos springs utilizados para o desenvolvimento deste projeto foram o Spring Boot, que auxilia na configuração da aplicação, o Spring MVC, responsável pelas classes de controle e injeção de dependências, o Spring *Security*, encarregado pelos processos de autenticação e autorização, e o Spring *Data*, que fornece o acesso aos dados existentes em um banco de dados. Segundo Moreira e Afonso (2017, p. 12), “O Spring não é um *framework* apenas, mas um conjunto de projetos que resolvem várias situações do cotidiano de um programador, ajudando a criar aplicações Java com simplicidade e flexibilidade”.

3.5 GITHUB

O GitHub é uma plataforma de hospedagem de código e colaboração que utiliza o Git, um sistema de controle de versão gratuito e de código aberto. Essa plataforma permite que seus usuários construam um projeto e acompanhe as mudanças feitas pelos demais sem a necessidade de estar no mesmo ambiente. O GitHub fornece um caminho direto para a criação de *sites* para seus projetos, o que o torna uma opção natural para publicar e manter a documentação. No GitHub é possível criar vários repositórios públicos, desde que estes projetos tenham licença *open-source*. Dentro desse repositório podem estar: pastas e arquivos, imagens, vídeos, planilhas e conjuntos de dados, tudo o que for necessário para o desenvolvimento do projeto. Em suma, o Github é um serviço de *web* que porta várias funcionalidades, em que “o Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos.” (SCHMITZ, 2015).

3.6 MySQL

O MySQL é um gerenciador de banco de dados de código aberto, sendo este facilmente configurável. Essa aplicação utiliza a Linguagem de Consulta Estruturada (SQL), através da qual o conteúdo que está incluso no armazenamento do banco de dados é administrado. Para utilizar o MySQL, é preciso que seja efetuada a instalação de um servidor e de uma aplicação cliente. O servidor será o responsável pelo armazenamento de dados, como também, por responder às requisições. Já a comunicação do cliente com o servidor irá acontecer por meio da SQL.

3.7 UML

A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de *software* (BOOCH; RUMBAUGH; JACOBSON, 2006). A Linguagem Unificada de Modelagem (UML) é uma linguagem de notação que permite aos programadores modelar e documentar de forma detalhada a estrutura e o comportamento de seus projetos de sistema. Essa modelagem é feita por meio de diagramas e cada diagrama possui seus próprios elementos gráficos que, por sua vez, possuem relação entre si. O objetivo dessa padronização é ajudar na comunicação entre os profissionais envolvidos na produção de um mesmo *software*.

4 METODOLOGIA

O principal método que utilizamos neste projeto foi o da pesquisa-ação que, de acordo com André Fontenelle (2020), fundamenta-se em identificar um problema prático de uma comunidade e, em seguida, desenvolver um projeto com ações para solucionar esse problema; tendo como principal característica, a proatividade e o envolvimento ativo do pesquisador para a mudança dos fenômenos.

Ademais, para comprovar a existência da problemática e auxiliar no desenvolvimento do projeto - definir quais informações seriam realmente pertinentes fazer parte do portal ItaretIF, bem como o que seria necessário à implementação de tais informações - fizemos uma pesquisa estruturada com o público alvo, os alunos do Campus Avançado Lajes. A pesquisa consistiu em um breve questionário, criado através do *google forms*, contendo perguntas de múltipla escolha obrigatórias e contabilizou 131 respostas (devido à pandemia do COVID-19, as aulas dos IFRNs estão sendo remotas, portanto, o questionário foi enviado aos grupos das respectivas salas por meio de *links* no *Whatsapp*). Esse método de pesquisa é categorizado como pesquisa quantitativa que, segundo Naína Tumelero (2019), consiste em traduzir os números e opiniões coletadas em informações que serão utilizadas para a sua classificação e posterior análise.

“A pesquisa quantitativa, ou quanti, é uma modalidade de pesquisa baseada na aplicação de questionários estruturados a um grupo de pessoas. [...] Quando falamos que a aplicação é realizada com auxílio de questionários estruturados, quer dizer que as mesmas perguntas são feitas a todos os participantes do levantamento” (DIAS, 2018).

Utilizamos os computadores dos laboratórios de informática, da biblioteca e nossos próprios *notebooks* para fazer as pesquisas, implementar o código e

escrever o relatório.

Após o levantamento dos requisitos, dividimos as atividades entre os integrantes do grupo. Ficamos responsáveis por elaborar os diagramas, de caso de uso e de classe, e cada um ficou responsável pelo seu próprio diagrama de sequência - utilizamos a UML (Linguagem Unificada de Modelagem) para modelar e documentar os diagramas. Além disso, cada um ficou encarregado de implementar, no mínimo, 2 partes e/ou funcionalidades do portal.

Para a implementação do projeto, a linguagem de programação utilizada foi Java, tanto por ser compatível com projetos *web*, rodar em qualquer sistema operacional, quanto por fazer parte da ementa acadêmica e ter sido estudada durante o curso Técnico Integrado em informática que o Campus oferece. O banco de dados ao qual optamos foi o MySQL, visto que é um gerenciador de banco de dados de código aberto, seguro, bastante fácil de configurar, manusear, amplamente utilizado por desenvolvedores, além de também ter sido estudado durante o curso. Para a definição da estrutura do conteúdo da página, utilizamos a linguagem de marcação HTML aliado ao CSS e bootstrap, para um visual mais agradável e elegante. Para a criação e configuração da nossa aplicação, utilizamos o *framework* Spring, que abarca vários projetos Springs, como o spring boot, que auxilia na configuração da aplicação, o spring *web*, que já vem com o servidor Apache TOMCAT embutido, o Spring MVC, responsável pelas classes de controle e injeção de dependências, o Spring *Security*, encarregado pelos processos de autenticação e autorização, e o Spring *Data*, que fornece o acesso aos dados existentes em um banco de dados. A IDE que utilizamos para o desenvolvimento do *software* foi o Eclipse, plataforma para desenvolvimento java e amplamente difundida.

Para que as mudanças no projeto pudessem ser feitas simultaneamente pelos integrantes do grupo, sem a necessidade de estarmos no mesmo local, criamos um repositório público no GitHub, plataforma de hospedagem de código e colaboração que utiliza o Git, um sistema de controle de versão gratuito e de código aberto.

Na efetivação do projeto, utilizamos o modelo de implementação separado

por camadas, o spring MVC (*Model, View, Controller*), no qual separamos o código nas camadas modelo (*Model*), classe java que retrata as entidades do sistema e é responsável pelo acesso e manipulação dos dados; visão (*View*), responsável pela interface que mostra as informações presentes no modelo e fica entre o usuário e o sistema; e controle (*Controller*), classe java responsável por implementar as funcionalidades e que faz a ligação entre o modelo e a visão.

5 ANÁLISE E PROJETO DO SISTEMA PROPOSTO

É sabido que informações importantes sobre o Campus Avançado Lajes podem ser encontradas no portal oficial do IFRN, como, por exemplo, notícias, eventos, editais, comunicação social e atos administrativos. Porém, percebemos que, além das informações sobre os Campi estarem em segundo plano e oferecer certas dificuldades de acesso para quem está visitando o *site* pela primeira vez, todo o portal possui uma estrutura estática e textos formais, sendo que alguns deste estão em PDF, tendo que, por vezes, ser feito *download* para poder visualizá-los. Todos esses fatores colaboram para a pouca atratividade, tornando o processo de interação entre o visitante e o portal bastante cansativo. Tal pressuposto foi comprovado com a aplicação do questionário, pois 88,5% dos entrevistados acharam necessário um portal exclusivo do Campus Avançado Lajes com características próprias, componentes mais intuitivos, com um conteúdo mais limpo e fácil de encontrar.

Sendo assim, o portal proposto - ItaretIF - vem com uma abordagem diferente, disponibilizando não somente as normas, regulamentos e informações sobre os setores e servidores, como, também, sobre a instituição em geral e projetos que serão desenvolvidos pelos professores e/ou mestres do Campus. Tudo isso com dinamismo, maior leveza visual e uma linguagem mais próxima do contexto regional em que o Campus está inserido.

5.1 APLICAÇÃO DO QUESTIONÁRIO

Como forma de validar e agregar à ideia do portal, foi aplicado um questionário estruturado às turmas do IFRN - Campus Avançado Lajes. Abaixo, seguem as imagens com as perguntas do questionário, os gráficos e as respostas obtidas:

Você conhece o portal IFRN?

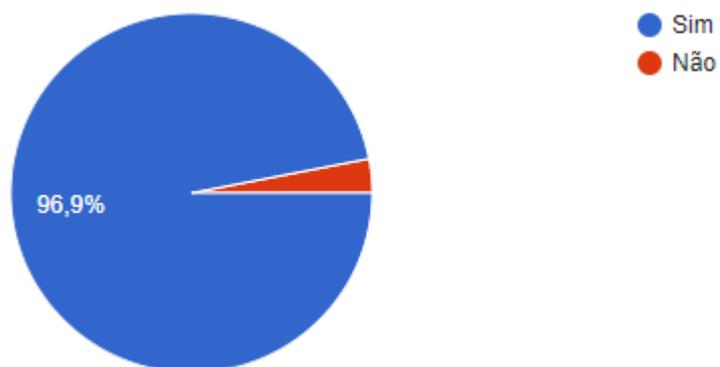


Figura 1 - Gráfico I

Com que frequência você acessa o portal IFRN?

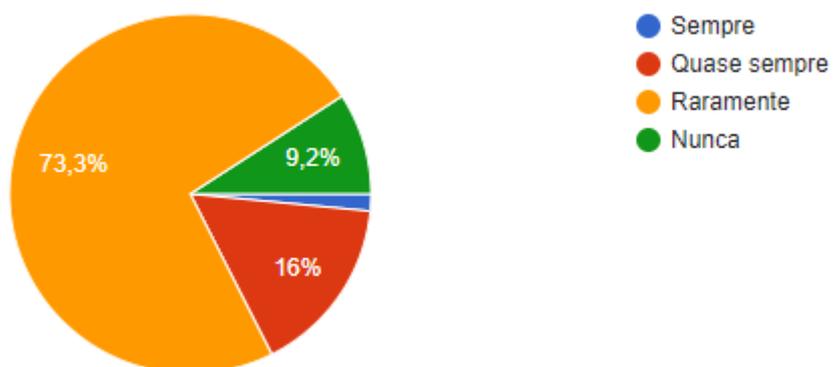


Figura 2 - Gráfico II

Consegue localizar as informações que procura com facilidade?

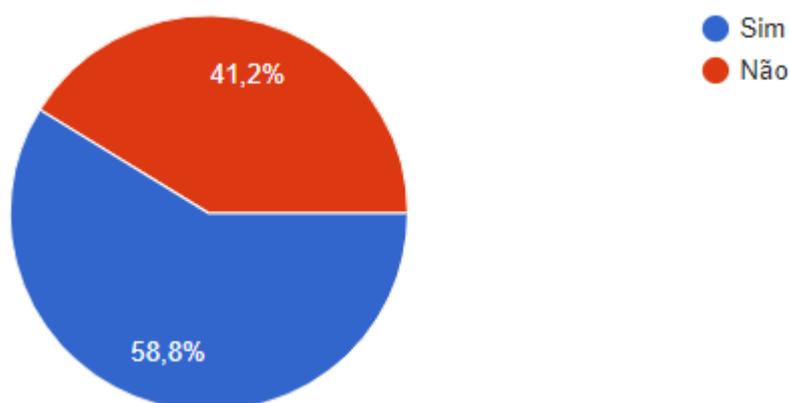


Figura 3 - Gráfico III

O que você acha de um portal que traga, em seu conteúdo, informações específicas do Campus Lajes? (informações sobre normas de conduta de alunos e servidores, sobre o Campus em geral, etc.)

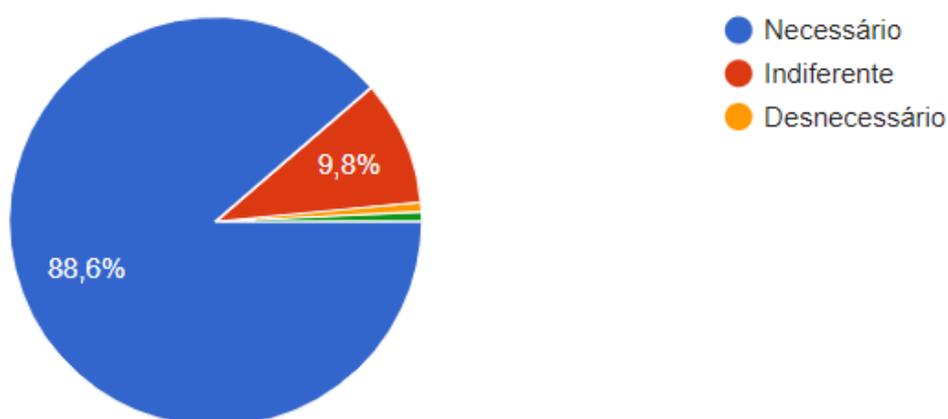


Figura 4 - Gráfico IV

Diante dos gráficos, concluímos que, apesar de 96,9% dos alunos terem conhecimento do portal oficial do IFRN (figura 1 - Gráfico I) e pouco mais da metade (58,8%) conseguir encontrar as informações que procuram com

facilidade (figura 3 - Gráfico III), 88,6% dos alunos entrevistados acham necessário a existência de um portal exclusivo para o Campus Avançado Lajes (figura 4 - Gráfico IV) que traga informações a respeito da instituição de ensino. Portanto, conclui-se que a maioria das respostas obtidas acerca desta temática considera válido um Portal para o Campus Avançado Lajes.

Além disso, analisando as respostas sobre o tópico que abria espaço para que os alunos fornecessem ideias a respeito das informações a serem divulgadas no portal, foram obtidas respostas como: “Acho que um *site* voltado a informações do Campus Lajes seria ótimo, principalmente se tivesse uma interface fácil de acessar”; “As informações precisam ser mais visíveis”; “Um *designer* legal”; “Um portal específico para o campus de Lajes é essencial, tanto para facilitar a encontrar informações, quanto para ficarmos atualizados das notícias. Um portal mais objetivo”; “Acredito que um portal com fácil acesso seja de grande ajuda, pois por mais que o sistema que utilizamos seja prático, ainda assim tem algumas peculiaridades que podem melhorar em relação ao acesso, então acho que a iniciativa de um projeto que visa essa melhoria é bastante efetiva.”; “Deixar o portal mais chamativo, para que os telespectadores que vão visitar o *site* sintam-se chamados e interessados a ler o conteúdo da página”; “Uma interface mais intuitiva!”.

5.2 REQUISITOS E CASOS DE USO

Inicialmente, fizemos o levantamento dos requisitos que o *software* deveria atender ao final do seu desenvolvimento e, então, após análise, começamos a decidir as estruturas que formariam o *software*, quais as funcionalidades de cada elemento, o acesso a dados, sincronização, desempenho, entre outros. Segue abaixo a lista com a definição dos requisitos do portal:

- Cadastrar usuários: Administradores, professores(as) e alunos(as);
- Permitir que usuários já cadastrados façam *login* no portal (autenticação);
 - Permitir que diferentes usuários tenham acesso a diferentes páginas (autorização);
 - Cadastrar servidor(a);
 - Cadastrar setor;
 - Cadastrar comentário;
 - Emitir relatório dos comentários;
 - Cadastro de projetos (apenas por professores já cadastrados no portal).

Após o levantamento dos requisitos, criamos o diagrama de caso de uso, de classe e os diagramas de sequência.

O diagrama de caso de uso é uma forma gráfica de demonstrar as funcionalidades do projeto, quem são os usuários que podem - e devem - participar da execução dessas funcionalidades e quais delas podem ou não ser realizadas por cada um deles. Na figura 5 podemos ver quais tarefas podem ser realizadas e quem são os usuários responsáveis por realizar cada uma delas.

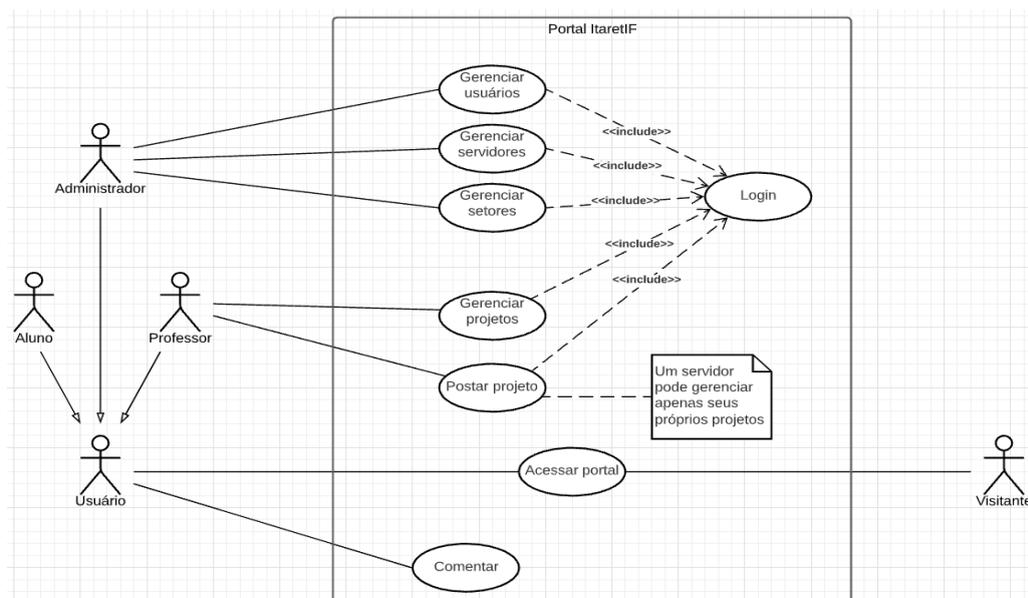


Figura 5 - Diagrama de Casos de uso

O visitante - que também pode ser um(a) aluno(a), servidor(a) ou administrador(a) - pode acessar o portal e navegar por ele sem a necessidade de fazer *login*. O aluno pode comentar; o(a) servidor(a) pode comentar, postar e gerenciar projetos; e o(a) administrador(a) pode comentar, cadastrar um novo servidor(a), setor ou administrador(a), editar o conteúdo do portal e acessar todas as listas de usuários, setores e servidores já cadastrados. No entanto, como representado no diagrama acima, determinadas ações só podem ser realizadas se o usuário fizer *login* (utilizando matrícula e senha).

Um visitante que não se encaixe em nenhum desses 3 usuários, não pode fazer *login* e poderá apenas navegar pelo portal.

5.3 DIAGRAMA DE CLASSES

Na figura abaixo (figura 6) vemos o diagrama de classe. Este descreve a estrutura estática de um *software* por meio de métodos, atributos e relações entre as classes, definindo assim a sua estrutura lógica. Em resumo, o diagrama de classes descreve o que estará presente no sistema modelado.

Para darmos início ao nosso diagrama foi necessário estudarmos um pouco sobre a definição de cada componente, sendo assim temos: A classe, que é uma forma de definir um tipo de dado. Para definir os dados utilizamos atributos, que apresentam o conjunto de propriedades da classe. E por último vem o método que basicamente irá ditar o comportamento e as funcionalidades de uma classe. Na figura abaixo mostramos especificamente a atuação de cada um desses componentes:

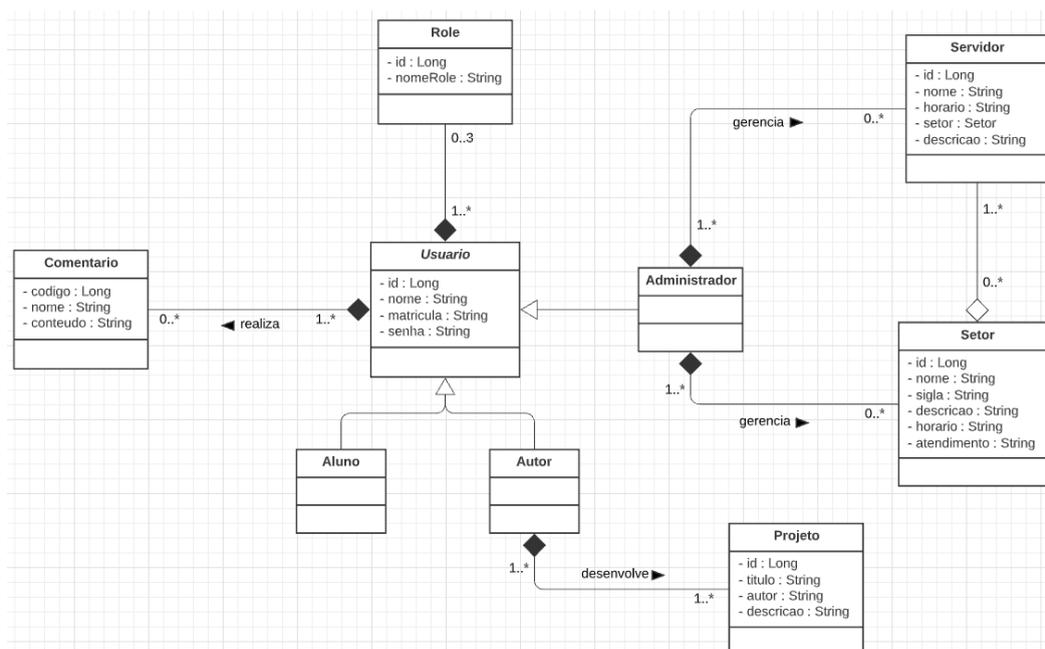


Figura 6 - Diagrama de classes

A classe *Role* é a classe responsável por definir os papéis dos usuários e depende da classe primária *Usuario* para existir. A classe *Usuario* possui as classes *Aluno*, *Autor* e *Administrador* como suas classes secundárias, logo, essas subclasses herdam todos os atributos da superclasse *Usuario*. Isso é necessário, pois, além de evitar repetição de código, o usuário pode ter diferentes papéis no nosso sistema. A classe *Comentario* é uma classe que depende da classe *Usuario*, pois sem a existência dessa classe, também não existiria a classe *Comentario*. Da mesma forma, a classe *Projeto* depende da classe *Autor*, pois a classe *projeto* não pode existir por conta própria, ela precisa da classe *Autor*. As classes *Setor* e *Servidor* também possuem uma relação de composição com a classe *Administrador* já que esta é a classe responsável pelos cadastros e gerenciamento dos servidores e setores. Por fim, as classes *Servidor* e *Setor* possuem uma relação de agregação, visto que, um servidor não necessariamente precisa de um setor. Sendo assim, a classe *Servidor* não depende da classe *Setor*.

6 PORTAL ITARETIF

O portal ItaretIF tem como proposta disponibilizar e facilitar o acesso às informações mais solicitadas pelos alunos do Campus Avançado Lajes, bem como o conhecimento básico sobre o município no qual o polo está localizado. Além disso, ainda no portal, é possível cadastrar e gerenciar projetos, cadastrar setores, servidores, alunos, administradores, listar cada um desses cadastros, acessar os dados de cada usuário e editar ou excluir cadastros.

A estrutura do portal conta com páginas de fácil manuseio, bastante intuitivas e limpas, contendo as funções que podem ser executadas pelos seus respectivos usuários e uma linguagem mais próxima dos jovens e do vocabulário local. Ademais, todas as páginas contam com uma barra de navegação (ou menu) para acesso rápido a outras páginas e funções. A seguir, apresentaremos as funcionalidades desenvolvidas.

A página inicial do portal (figura 7) possui uma breve apresentação, para que quem esteja visitando o portal pela primeira vez possa se situar. No menu, temos o *link* para as páginas: *O Campus*, *Regras*, *Lajes/RN*, *Mapeamento*, *Projetos*, *Sobre nós*, *Login* e *Cadastre-se!*. Com exceção da página de *login* e de cadastro, todas as outras páginas base do portal contam com esse mesmo menu e uma imagem do Campus como plano de fundo.

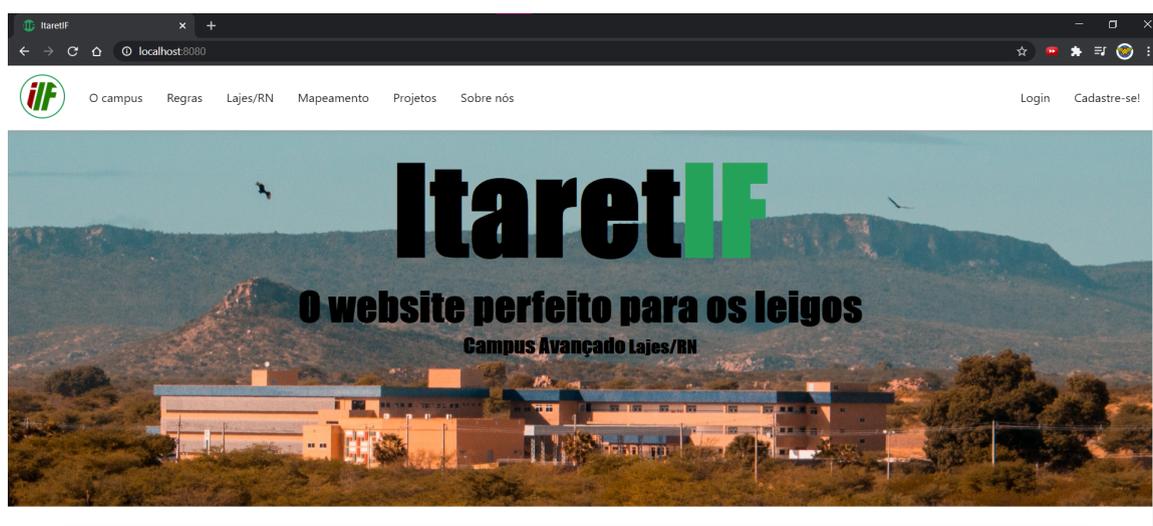


Figura 7 - página inicial do portal

6.1 FUNCIONALIDADES DESENVOLVIDAS

Nessa seção, vamos apresentar todas as funcionalidades que foram desenvolvidas de acordo com os requisitos levantados e mencionados anteriormente.

A página *O Campus*, além de conter informações sobre a história do Campus, sua estrutura e localização, conta com uma lista de servidores - cadastrados anteriormente - que trabalham no Campus, e a lista de setores, como podemos ver nas figuras 8 e 9. Além disso, a página conta com um pequeno menu do lado direito da tela com um *link* para a lista de servidores e outro para a de setores, levando o usuário rapidamente para os tópicos em questão sem a necessidade de rolar a tela.

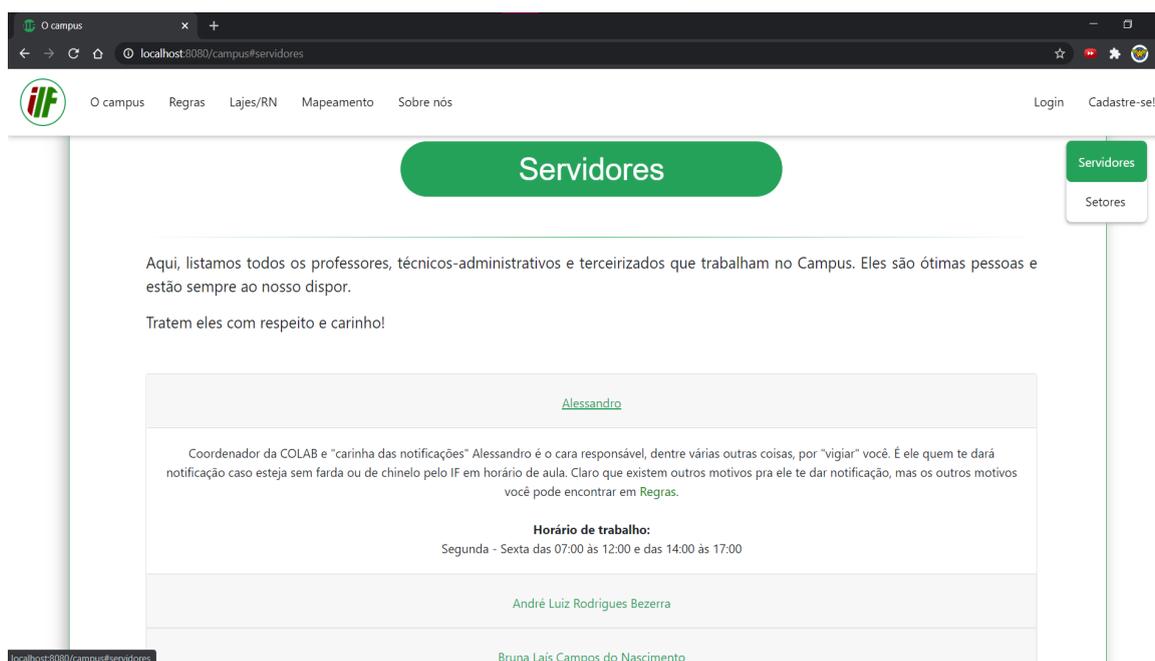


Figura 8 - Página *O Campus*, lista de servidores

Setor	Sigla	Descrição	Horário de funcionamento	Atendimento
Direção Acadêmica	DIAC	Setor voltado aos assuntos acadêmicos	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	Osailton
Direção Administrativa	DIAD	-	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	A amável Chris (Christiane Maria).
Direção Geral	DG	-	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	Helaine
Secretaria Acadêmica	SEAC	-	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	-

Figura 9 - Tabela com todos os setores do Campus

Na página *Sobre nós* encontra-se, dentre informações sobre os desenvolvedores, o espaço para comentários e sugestões (figura 10).

Comentários e sugestões

Para que o nosso site tenha sua cara, precisamos da sua ajuda! Para isso, basta apenas nos dar o seu feedback sobre o portal e/ou trazer sugestões sobre o que achar necessário.

Informe seu nome.

Deixa um comentário ou uma sugestão pra gente :)

Opinar

Figura 10 - página *Sobre nós*, campo para comentários

A página de *login* (figura 11), que pode ser acessada pelo aluno, professor, mestre e administrador, possui dois campos para preenchimento: matrícula e senha. Apenas ao fazer o *login* é possível ter acesso às páginas ou a funcionalidades determinadas ao respectivo uso desses usuários.



Faça login

Matrícula:

Senha:

Entrar

Novo por aqui?! [Crie logo sua conta!](#)

Figura 11 - página de *login*

Caso o usuário não esteja cadastrado no sistema, ele pode se cadastrar indo no *link Cadastre-se!* (que encontra-se no *menu* de todas as páginas irrestritas), ou clicando no *link* logo abaixo do formulário de *login*: *Crie logo sua conta!*. Ao fazer isso, inicialmente, ele terá que informar se é aluno ou servidor - como mostra a figura 12 - para só então ter acesso a página de cadastro (figura 13).



Figura 12 - página de pré-cadastro

O cadastro só será efetivado se o usuário preencher todos os campos, e uma mensagem de sucesso aparecerá para o usuário. Caso contrário, aparecerá uma mensagem de erro na tela.

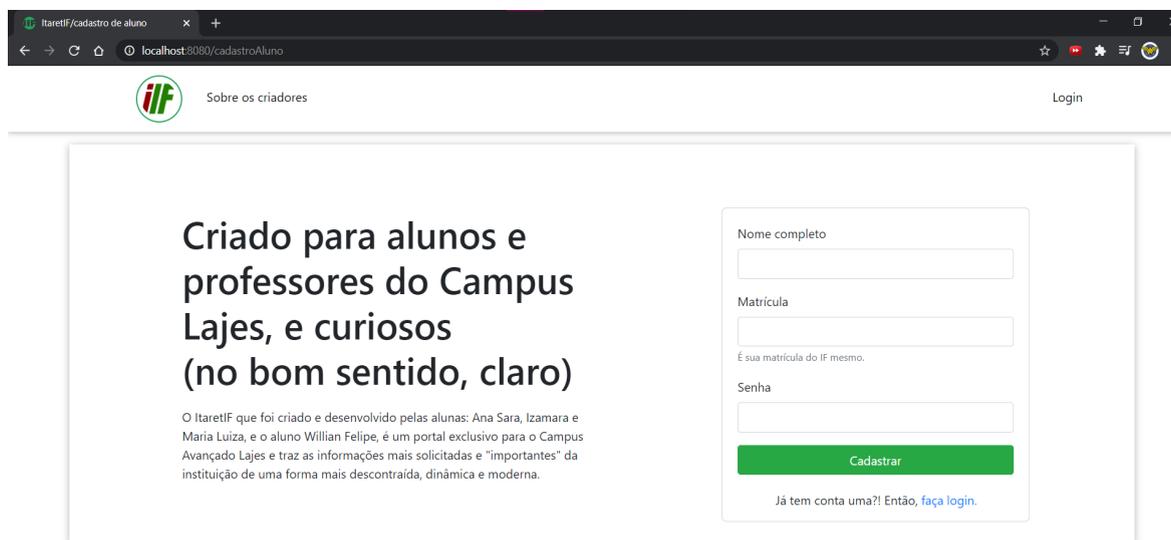


Figura 13 - página com o formulário para cadastro

Na página inicial do(a) administrador(a) há uma barra de navegação com dois *links*: Um para acessar um *menu* com *links* para as listas de cadastros e

outro para fazer *logout* (figura 14). No corpo da página temos três formulários: Um para o cadastro de administradores(as), outro para o cadastro de servidores(as) e o último para o cadastro de setores. Ao finalizar algum desses cadastros, aparecerá uma mensagem de sucesso indicando que o usuário, servidor ou setor foi cadastrado corretamente.

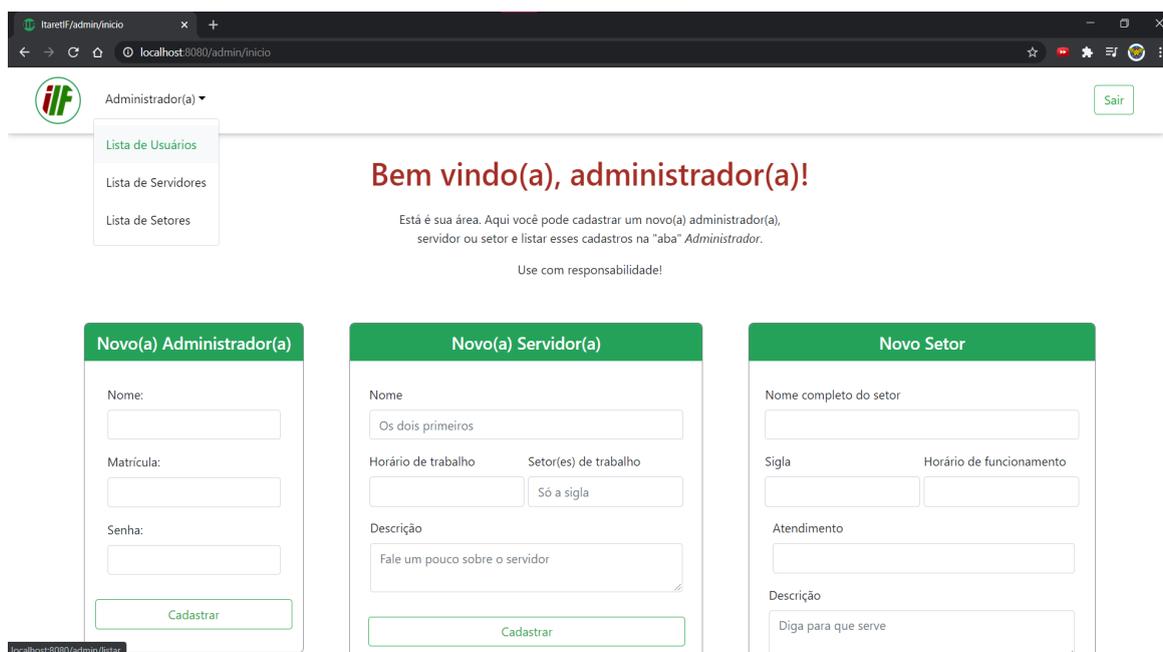
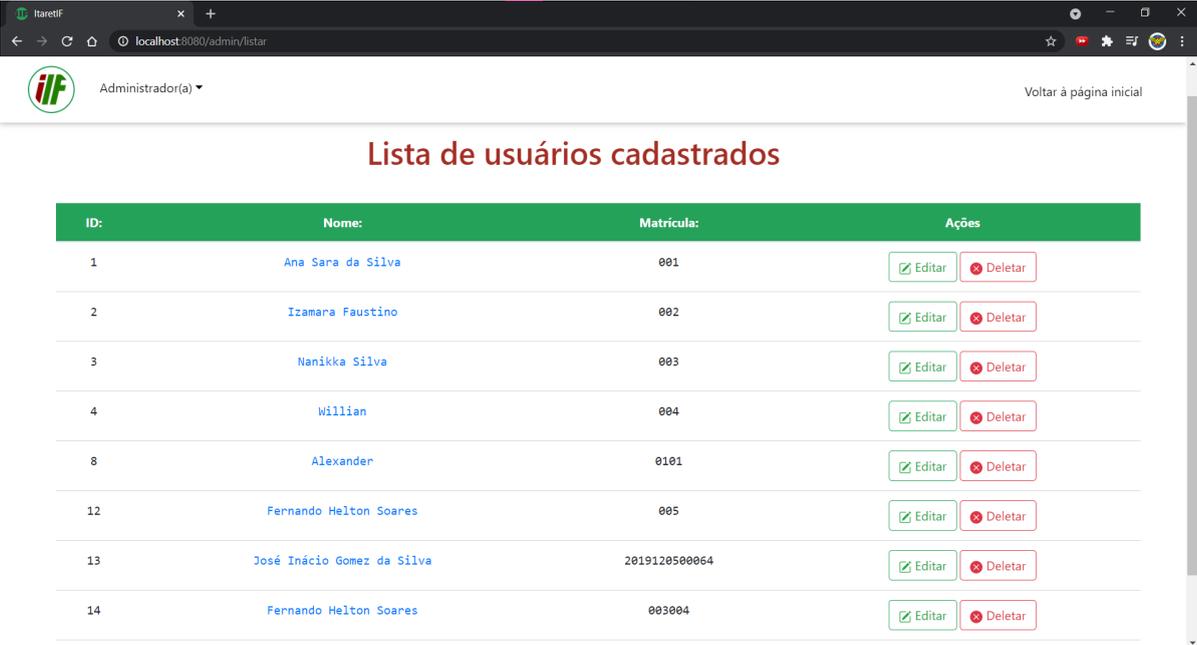


Figura 14 - página inicial do(a) administrador(a)

Ao clicar no *link Lista de Usuários*, o usuário é redirecionado à página com todos os usuários cadastrados que possuem um ou mais papéis (Aluno, Servidor ou Administrador), como demonstra a figura 15. Ao clicar nos nomes dos usuários, somos redirecionados para uma página com mais detalhes sobre o usuário em questão, como podemos ver na figura 16. Também é possível editar (figura 17) ou excluir os usuários.

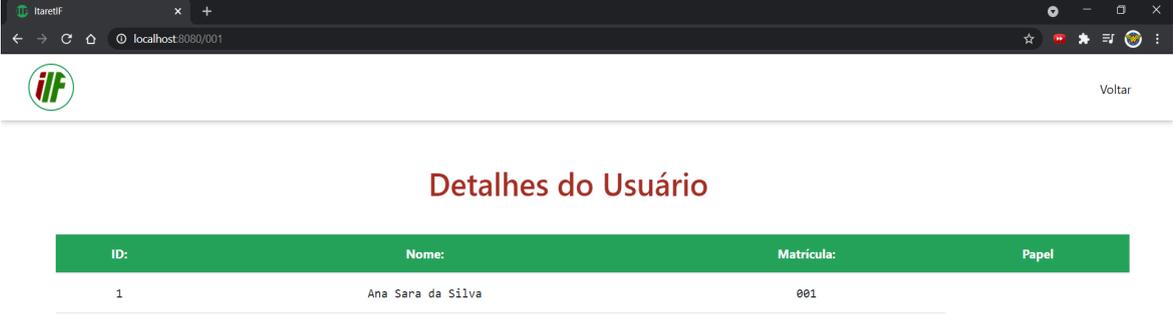


Administrador(a) Voltar à página inicial

Lista de usuários cadastrados

ID:	Nome:	Matrícula:	Ações
1	Ana Sara da Silva	001	Editar Deletar
2	Izamara Faustino	002	Editar Deletar
3	Nanikka Silva	003	Editar Deletar
4	Willian	004	Editar Deletar
8	Alexander	0101	Editar Deletar
12	Fernando Helton Soares	005	Editar Deletar
13	José Inácio Gomez da Silva	2019120500064	Editar Deletar
14	Fernando Helton Soares	003004	Editar Deletar

Figura 15 - Lista de usuários cadastrados



Voltar

Detalhes do Usuário

ID:	Nome:	Matrícula:	Papel
1	Ana Sara da Silva	001	

Figura 16 - Página com os detalhes de um usuário

ItaretiF/admin/Edição de Usuário x +

localhost:8080/editarUsuario/001

Cancelar

Edição do usuário

Nome

Matrícula

Senha

Figura 17 - formulário de edição dos dados do usuário

Ao clicar no *link Lista de Servidores*, o usuário é direcionado à página que contém todos os servidores que fazem parte do Campus Lajes, contendo informações como: nome, horário de trabalho, nome do setor em que o servidor(a) fica e a descrição deste (figura 18). Essa lista também aparece na página *O Campus*, página já apresentada anteriormente na figura 8. Também é possível editar (figura 19) ou excluir servidores.

ItaretiF/Lista de Servidores x +

localhost:8080/admin/ListaServidores

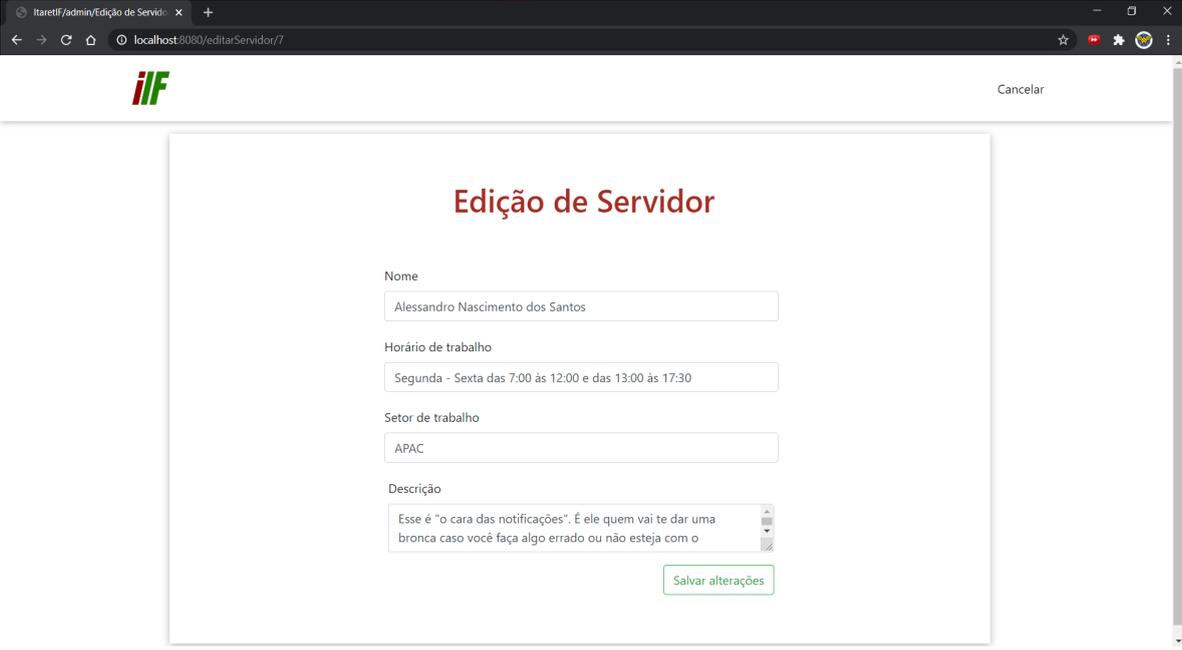
Administrador(a) ▾

Voltar à página inicial

Lista de Servidores Cadastrados

Nome	Horário de Trabalho	Setor	Descrição	Ações
Alessandro Nascimento dos Santos	Segunda - Sexta das 7:00 às 12:00 e das 13:00 às 17:30	APAC	Esse é "o cara das notificações". É ele quem vai te dar uma bronca caso você faça algo errado ou não esteja com o fardamento completo. Mas ele é super legal, boe! Serião	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>
André Luiz Rodrigues Bezerra		DIAC	Deus magnifico, senhor de tudo, mestre de Arte. Esse é o cara ao qual o Google consulta antes de publicar uma pesquisa. skfdkkk Brincadeiras a parte, o mestre é uma ótima pessoa... aproveitem.	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>
Bruna Laís Campos do Nascimento		ASBIB	Ela trabalha na biblioteca, muito simpática e educada (amo). Mas não irrite ela ou ela vai parecer gentilmente assustadora.	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>
Cassia de Fatima Matos dos Santos		DIAC	A melhor professora de literatura e português que já tive na vida. Sério, boe! E ela é tão animadinha e apaixonada pela disciplina... chega a pessoa se empolga junto com ela. Linda perfeita	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>

Figura 18 - lista de servidores do Campus



The screenshot shows a web browser window with the URL `localhost:8080/editarServidor/7`. The page title is "Edição de Servidor" and features the IIF logo. The form contains the following fields:

- Nome:** Alessandro Nascimento dos Santos
- Horário de trabalho:** Segunda - Sexta das 7:00 às 12:00 e das 13:00 às 17:30
- Setor de trabalho:** APAC
- Descrição:** Esse é "o cara das notificações". É ele quem vai te dar uma bronca caso você faça algo errado ou não esteja com o

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

Figura 19 - formulário de edição dos dados do servidor

Por fim, ao clicar no *link Lista de Setores*, o usuário será redirecionado à página contendo uma lista com todos os setores da instituição, contendo informações como: o nomes dos setores, a sigla, uma breve descrição sobre o setor, o horário de funcionamento deste e o nome dos servidores que trabalham em cada um deles (figura 20). Também é possível editar (figura 21) ou excluir os setores.



Nome do Setor	Sigla	Descrição	Horário	Atendimento	Ações
Direção Acadêmica	DIAC	Setor voltado aos assuntos acadêmicos	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	Osailton	Editar Deletar
Direção Administrativa	DIAD	-	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	A amável Chris (Christiane Maria).	Editar Deletar
Direção Geral	DG	-	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	Helaine	Editar Deletar
Secretaria Acadêmica	SEAC	-	Segunda - Sexta das 8:00 às 12:00 e das 13:00 às 18:00	-	Editar Deletar

Figura 20 - lista de setores cadastrados



Edição de Setor

Nome do setor

Sigla

Horário de Funcionamento

Atendimento

Descrição

Figura 21 - formulário de edição dos dados do setor

Ao estudarmos o Portal IFRN, percebemos que os editais e projetos desenvolvidos pelo Campus não apresentam grande visibilidade, dificultando a

divulgação de tais informações necessárias tanto para a comunidade interna quanto para a comunidade externa do instituto. Pensando nesta problemática, resolvemos implementar outra funcionalidade no nosso portal, uma página específica para comportar tais informações (figura 22).

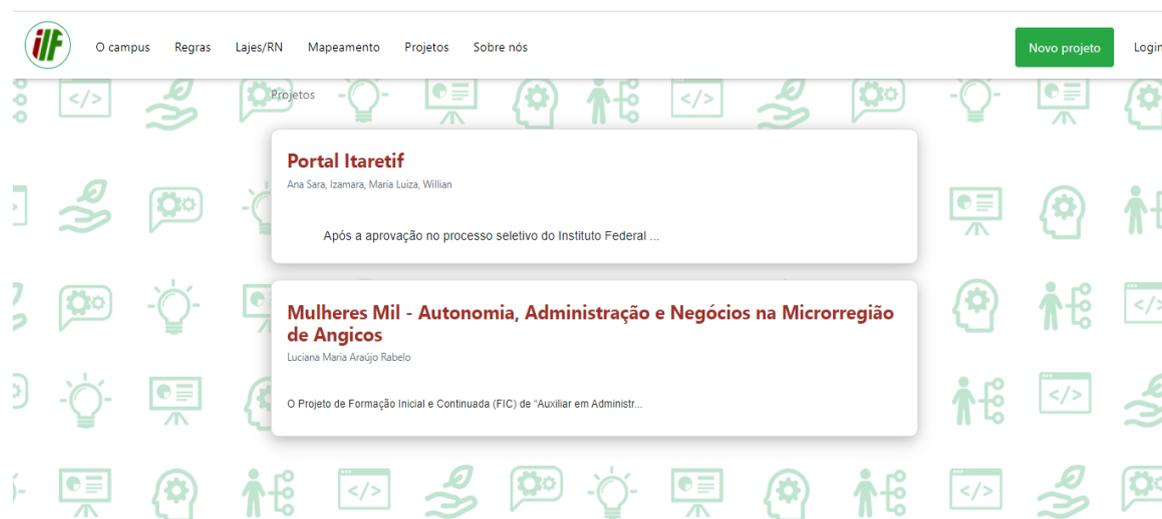


Figura 22 - Página de projetos

Como vemos na figura 22, ao clicar no botão “Novo projeto”, aparecerá os campos “Título”, “Autor”, e “Descrição” (figura 23). Obrigatoriamente o coordenador do projeto irá preencher todos para que o projeto seja devidamente cadastrado. Caso um campo não receba o devido preenchimento, aparecerá uma mensagem de alerta exigindo o complemento.

Projetos / Novo Projeto

Título *

Autor *

Segoe UI

Descrição *

Publicar Cancelar

Campos obrigatórios

Figura 23 - Formulário para adicionar um novo projeto

Ao preencher a descrição, ele também poderá adicionar um estilo, podendo escolher a fonte, a adição de tabelas, imagens e vídeos e entre outras opções de edição. Ao cadastrar, o projeto se juntará à lista de projetos, como podemos ver na figura 22, onde poderemos detalhar cada um e ler por completo. Na página onde ficará o detalhamento, colocamos um botão para deletar o projeto quando ele chegar ao fim e outro para editar, caso precise alterar alguma informação (figura 24). Mas essas opções só podem ser executadas pelo servidor ou administrador.



Figura 24 - Detalhamento do projeto

Essas funcionalidades não podem ser executadas e administradas por qualquer pessoa, logo, durante o processo de idealização do nosso sistema, percebemos a necessidade de que determinadas páginas e funcionalidades tivessem acesso restrito, podendo ser acessadas somente após a autenticação e autorização do usuário por meio de uma matrícula e senha. À vista disso, criamos a funcionalidade e a página para o *login* do usuário.

6.2 IMPLEMENTAÇÃO

O foco deste trabalho está na funcionalidade de postar e gerenciar projetos, então para iniciar, desenvolvi um diagrama de sequência para detalhar a função já descrita no diagrama de casos de uso.

Como o próprio nome já sugere, o diagrama mostra a sequência dos eventos que deverão ocorrer, as interações entre cada objeto, qual a ordem e o seu tempo de vida. Vejamos na figura abaixo (figura 25):

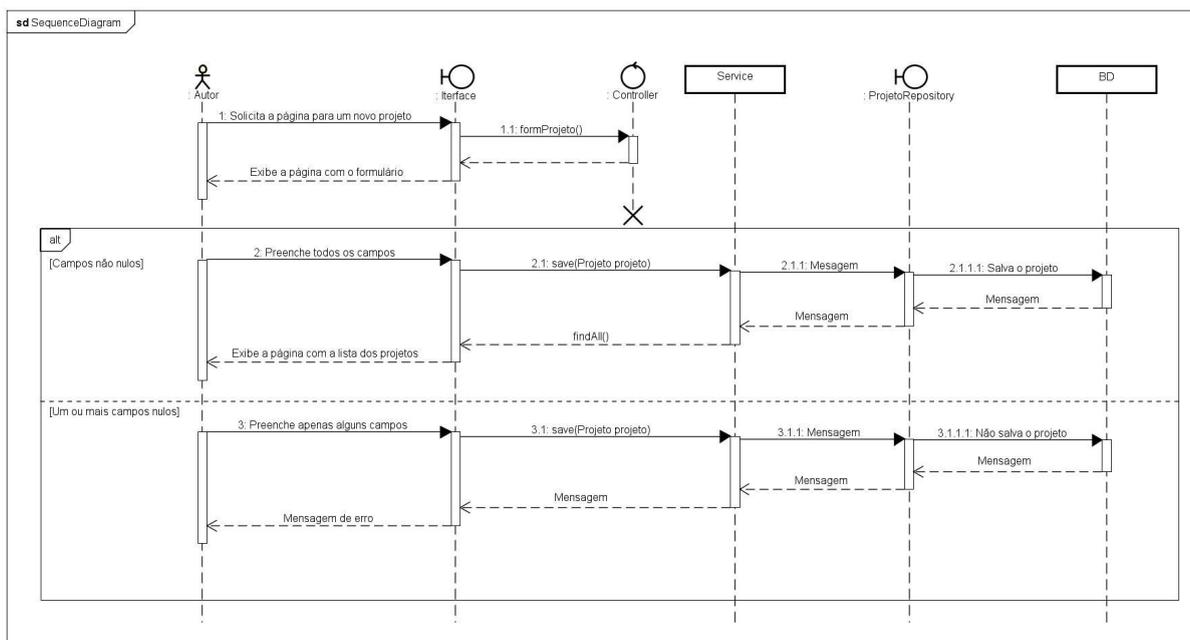


Figura 25 - Diagrama de sequência da função postar projeto

No primeiro cenário o ator será o autor, isto é, aquele que irá solicitar o formulário para cadastrar um novo projeto. A interface irá mandar uma mensagem para o *controller* que retornará o formulário e exibirá na tela a página de cadastro para que o autor preencha os campos devidamente. No segundo cenário, em que o autor preenche todos os campos, a interface manda as informações para o *controller* que faz a validação e envia a mensagem para o *Service*, que, por sua vez, invoca o método *save(Projeto projeto)* para salvar o projeto no banco de dados. Depois que o projeto é salvo, o *service* manda uma mensagem ao *repository*, que retorna todos os atributos já salvos no banco de dados. Então, o *service* invoca o método *findAll()*, responsável por retornar uma lista com todos os projetos já cadastrados para o *controller*. Este, por sua vez, retornará a página com os projetos cadastrados para a interface. Por fim, o autor verá o seu projeto juntamente com todos os demais. Já no terceiro cenário, o autor não preencheu um ou mais campos, logo, a interface envia as informações para o *controller*, mas ele não as valida e retorna uma mensagem de erro, solicitando o preenchimento de todos os campos, que é exibida ao autor na

própria página que contém o formulário.

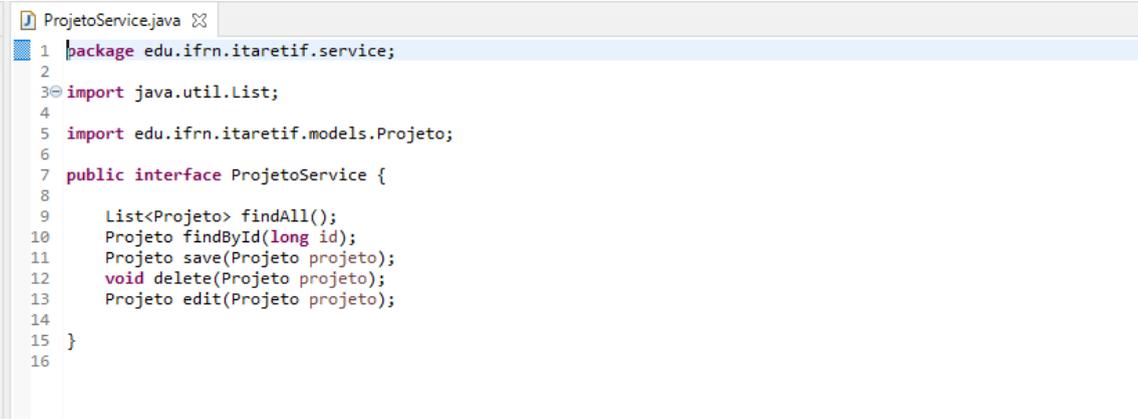
Para a implementação desta funcionalidade, a qual fiquei responsável, foi utilizado alguns recursos já mencionados na nossa metodologia, como: *Spring MVC*, *Spring Security* (uma vez que a funcionalidade de postar e gerenciar projetos foi desenvolvida apenas para servidores e administradores), *Bootstrap* para construir as páginas HTML e o *Thymeleaf* para renderizá-las.

Ao iniciar, foi gerado o projeto *spring* e todas dependências foram baixadas. Após isso, o projeto foi importado para dentro da IDE (usamos o Eclipse, já mencionado acima), foi feita a conexão com banco de dados por meio do *MySQL*, e lá, foi criado o banco por nome *itaretif*, responsável por armazenar os dados da nossa funcionalidade.

Tendo já configurado o nosso banco, no nosso diretório *model* (responsável por receber tabelas no banco de dados), criamos a classe java por nome *Projeto.java*. Nesta classe foi adicionada a anotação *@Entity*, e através dela juntamente com o comando "*spring.jpa.hibernate.ddl-auto=update*", uma tabela será gerada automaticamente. Após isso, foram criados os atributos que essa tabela irá receber, sendo eles: o *Id*, anotado com *@Id*, responsável por identificar a chave primária, e *@GeneratedValue (strategy = GenerationType.IDENTITY)* que define a estratégia adotada para gerar a chave primária. Adicionamos também os atributos: *título*, *autor* e *descrição*, anotados com a validação *@NotBlank*, ou seja, não poderão ser vazios ou nulos. Na descrição especificamente, anotamos com *@Lob*, pois esse atributo pode receber uma quantidade relativamente maior de caracteres. E por fim, foram gerados os *Getters and Setters* desses atributos adicionados.

Seguindo o mesmo modelo acima, no repositório principal foi criada a interface por nome *ProjetoRepository.java*, que ficará responsável pelas transações com o banco, sejam elas atualizações, consultas e entre outras. Dentro dele fizemos a ponte com o *model Projeto.java* e o *id*, identificado como *long*. Tendo estendido o JPA para ter acesso aos métodos do *SpringData*,

facilitando a comunicação com o banco de dados, foi criado então um diretório chamado *Service*, responsável pela parte lógica da nossa funcionalidade. Dentro dele foi criado uma interface chamada *ProjetoService.java* para comportar os métodos do nosso código. Esses métodos serão ações a serem implementadas, que são: Listar, para retornar uma lista de projetos, encontrar pelo id, para que possamos fazer buscas mais específicas, salvar, quando um novo projeto for adicionado, deletar, para quando um projeto for apagado do nosso banco, e editar, para fazer alterações no projeto já postado. Na Figura 26, podemos ver nossos métodos a serem implementados.



```
1 package edu.ifrn.itaretif.service;
2
3 import java.util.List;
4
5 import edu.ifrn.itaretif.models.Projeto;
6
7 public interface ProjetoService {
8
9     List<Projeto> findAll();
10    Projeto findById(long id);
11    Projeto save(Projeto projeto);
12    void delete(Projeto projeto);
13    Projeto edit(Projeto projeto);
14
15 }
16
```

Figura 26 - *ProjetoService.java*

Após a criação dos nossos métodos, foi criado mais um diretório por nome *ServiceImple* para implementá-los. Dentro dele foi criado uma classe chamada *ProjetoServiceImple.java*, e nela foi importada a interface *ProjetoService.java*. Implementamos também os nossos métodos anotados com `@Override` para que o sistema entenda que estamos sobrescrevendo um método, e não criando um novo. Foi utilizado também a anotação `@Autowired` criando um ponto de injeção para utilizar as instâncias do nosso *repository*, e por fim, todos os métodos da nossa interface foram implementados. Vale ressaltar que também utilizamos a anotação `@Service`, uma vez que será o *spring* que irá gerenciar a classe

ProjetoServiceImple.java.

Foi criada também a classe *controller* responsável por gerenciar as requisições. Nela foi adicionada a anotação `@Controller`, para marcar essa classe como um *controller MVC* que irá receber as requisições dos clientes, e `@Autowired` para criar um ponto de injeção da classe *ProjetoService.java*, onde estão localizados os métodos que serão utilizados. Tendo já implementado a estrutura necessária, irei descrever apenas a implementação da funcionalidade de postar projeto.

Inicialmente, na nossa classe *controller*, implementamos os seguintes métodos:

```

47
48Ⓜ @RequestMapping(value="/serv/novoprojeto", method=RequestMethod.GET)
49     public String getProjetoForm(){
50         return "projetoForm" ;
51     }
52
53Ⓜ @RequestMapping(value="/serv/novoprojeto", method=RequestMethod.POST)
54     public String saveProjeto(@Valid Projeto projeto, BindingResult result, RedirectAttributes attributes){
55         if(result.hasErrors()){
56             attributes.addFlashAttribute("mensagem", "Verifique se os campos obrigatórios foram preenchidos!");
57             return "redirect:/serv/novoprojeto";
58         }
59
60         projetoService.save(projeto);
61         attributes.addFlashAttribute("mensagem", "Projeto cadastrado com sucesso!");
62         return "redirect:/projetos";
63     }
64

```

Figura 27 - ProjetoController (Métodos utilizados para salvar um projeto)

Em resumo, o método `getProjetoForm()` irá retornar o formulário a ser preenchido. Adicionamos a anotação `@RequestMapping`, definimos também o valor da URL e o método que essa requisição irá receber (linha 48 a 51) .

Abaixo dele temos o método `saveProjeto(@Valid Projeto projeto, BindingResult result, RedirectAttributes attributes)`, ou seja, após preencher o formulário corretamente, o projeto será salvo no banco, e o usuário será redirecionado a página HTML contendo todos os projetos. Foi utilizada também a anotação `@RequestMapping`, e em seguida foi definido o valor da URL e o método que a requisição recebe (linha 53 - 58). Caso um ou mais campos do formulário não seja preenchido, ou seja, se a validação vier com erro, o formulário será retornado para ser preenchido novamente com uma mensagem

de erro, por isso adicionamos a anotação `@valid` presente nos parâmetros (linha 54).

Tendo criado os métodos no *controller*, criamos uma página HTML chamada *Projetos.html* (figura 28).

```

20 </a>
21 <a class="" href="/campus" style="margin-top: 20px;">O campus</a>
22 <a class="" href="/regras" style="margin-top: 20px;">Regras</a>
23 <a class="" href="/lajes" style="margin-top: 20px;">Lajes/RNK</a>
24 <a class="" href="/mapeamento" style="margin-top: 20px;">Mapeamento</a>
25 <a class="" href="/projetos" style="margin-top: 20px;">Projetos</a>
26 <a class="" href="/sobre" style="margin-top: 20px;">Sobre nós</a>
27 <a class="" href="/login" style="margin-top: 20px; float: right;">Login</a>
28 <a href="/serv/novoprojeto" class="btn btn-success az" role="button">Novo projeto</a>
29 </div>
30
31 </header>
32
33 <div class="section">
34
35 <section>
36 <div class="container" style="width: 60%; padding-bottom: 30px; padding-top: 50px;">
37 <nav aria-label="breadcrumb">
38 <ol class="breadcrumb" style="background-color: transparent; padding-left: 0px;">
39 <li class="breadcrumb-item active" aria-current="page">Projetos</li>
40 </ol>
41 </nav>
42 <div class="card shadow-sm bg-white rounded" style="margin-top: 20px; border-radius: 10px !important; box-shadow: -1px 5px 20px #b3b3b3 !important;" th:each="projeto : ${
43 <div class="card-body">
44 <a th:href="${(#mvc.url('CP#getProjetoDetails').arg(0, projeto.id)).build()}" style="color: #A32E25;">ch4 class="card-title" style="font-weight: bold; color: #A
45 <p class="card-subtitle mb-2 text-muted" style="font-size: 0.8rem;">
46 <span th:text="${projeto.autor}"></span>
47 </p>
48 <p class="card-text"><span th:text="${#strings.abbreviate(projeto.descricao,450)}"></span></p>
49 <!-- <a th:href="${(#mvc.url('CP#getProjetoDetails').arg(0, post.id)).build()}" class="card-link">Acessar</a> -->
50 </div>
51 </div>
52 </div>
53 </section>
54 </div>
55
56 </body>
57 </html>
58

```

Figura 28 - *projetos.html*

Para a construção desta página, foi utilizado alguns recursos já prontos do Bootstrap, assim como as demais, seguindo um *layout* padrão. Nela irá conter uma lista com todos os projetos já cadastrados, sendo cada um organizado dentro de um *card*, na seguinte ordem: Título, autor e os primeiros 450 caracteres da descrição. Vale ressaltar que adicionamos um *link* ao título, que nos leva a outra página HTML chamada *projetosDetails.html*, onde será apresentado o projeto de forma completa e mais detalhada, contendo também o botão para adicionar um novo projeto (no topo da página), e os botões das demais funcionalidades (figura 29).

Para a construção do formulário, foi seguido o padrão de construção das outras páginas com alguns recursos do *bootstrap*, mantendo o *layout* semelhante às demais páginas. Especificando a construção do formulário, foi adicionado mais uma *div*, e dentro dela foi adicionada a *tag form* junto com o método *post*. Após isso, foi implementado os elementos do nosso formulário, sendo eles: Título, autor e a descrição. Na descrição especificamente, foi inserido um formatador utilizando *bootstrap*, e alguns *links* e *scripts* do *jQuery* e *Summernote*. Podemos ver da linha 65 até a 71 (figura 31), que foi adicionado um pequeno *script* para definir o dimensionamento do formatador. Após isso, foram adicionados os botões “publicar” e “cancelar”. Tendo já inserido o formatador e os botões essenciais para a funcionalidade de postar projeto, foi criado mais uma página HTML para implementar a mensagem de que erro que irá aparecer para o usuário quando o mesmo deixar de preencher algum campo do formulário, chamamos de *mensagemValidacao.html* (figura 31).

```
mensagemValidacao.html
1 <!DOCTYPE html>
2 <html lang="pt"
3   xmlns="http://www.w3.org/1999/xhtml"
4   xmlns:th="http://thymeleaf.org">
5 <head>
6
7 </head>
8 <body>
9 <div class="alert alert-danger alert-dismissable" role="alert" th:if="{not #strings.isEmpty(mensagem)}">
10
11   <span th:text="{mensagem}"></span>
12 </div>
13 </body>
14 </html>
```

Figura 31 - *mensagemValidacao.html*

Nesta página foram utilizadas algumas classes de alerta do *bootstrap*, e uma condição utilizando *thymeleaf* (linha 9), ou seja, caso seja encontrado erro, retornará uma mensagem determinada no *controller*. Após isso, a mensagem de validação foi inserida no *form*, representado na figura 30 (linha 40).

7 ANÁLISE E DISCUSSÃO DOS RESULTADOS

De acordo com os resultados obtidos ao final do projeto, podemos ver que alcançamos o objetivo proposto. O portal facilita o acesso às informações tanto a comunidade interna quanto externa do Campus Avançado Lajes, apresenta uma linguagem familiar, além de proporcionar uma interface criativa e confortável ao usuário.

Como foi proposto, o portal disponibiliza também informações essenciais para os ingressantes, tais como: normas de conduta, quais os setores existentes e seus respectivos servidores, as plantas do campus para orientar alunos e visitantes, informações sobre o município em que está situado e entre outras.

Partindo para as funcionalidades descritas no diagrama de caso de uso, com reforço e aprofundamento nas disciplinas de programação, conseguimos desenvolvê-las com sucesso. É possível fazer *login*, cadastrar aluno(a), servidor(a) e setor, adicionar comentários, cadastrar e gerenciar projetos com êxito.

O projeto foi desenvolvido em cima de uma problemática importante, visível não só na nossa experiência como discentes, mas também como comunidade externa, uma vez que recorreremos ao portal IFRN para buscarmos informações acerca dos processos seletivos e eventos ofertados pelo campus. Isso se tornou um fator determinante para estabelecermos a temática do nosso projeto. Não deixando de citar os dados levantados a partir do questionário, fortalecendo ainda mais a necessidade de um portal específico para comportar tais informações.

Como sabemos, sempre há melhorias a serem feitas no âmbito da programação, levando em consideração o avanço dos estudos e dos novos métodos adotados para se desenvolver uma aplicação. Contudo, nosso projeto faz jus ao conhecimento que nos foi passado pelos nossos professores ao longo do curso.

8 CONSIDERAÇÕES FINAIS

Diante dos dados levantados ao longo do projeto por meio de pesquisas, videoaulas, e entre outros recursos, é possível detectar que a programação é uma área ampla, e se nos aprofundarmos ainda mais teremos acesso a uma infinidade de recursos.

Vale ressaltar que, além de requerer a prática de diversos conteúdos estudados em classe, este projeto exigiu dedicação e aprofundamento em outras ferramentas, como o *github*. Não esquecendo de citar também o estudo da UML (*Unified Modeling Language*), que enriqueceu ainda mais a composição deste relatório.

Como sugestão para melhor eficiência e melhoramento do portal, pode-se ampliar a utilização de *JavaScript*, permitindo a adição de outras funcionalidades na dinâmica do portal. Ao se tratar de uma ferramenta que visa a integração da comunidade interna e externa do campus, fica a sugestão de uma galeria contendo fotos do campus, eventos escolares e exposições. Não esquecendo da identidade visual como atrativo para os usuários, fica sugestão a atualização de alguns recursos, a fim de manter a página chamativa para todos os públicos.

Levando em conta os resultados obtidos, fica a nossa contribuição ao campus como ideia a ser trabalhada posteriormente, e a experiência adquirida que irá nos auxiliar na nossa jornada no mercado de trabalho.

REFERÊNCIAS

DUCKETT, Jon. HTML & CSS projete e construa websites, Rio de Janeiro, 2016. Disponível em: <<https://www.passeidireto.com/arquivo/75371077/html-and-css-projete-e-construa-websites>> Acesso em: 20 de Fevereiro de 2020.

MEIRA, Regilan. Apostila de programação para web: HTML e CSS, Bahia, 2013. Disponível em: <http://www.regilan.com.br/wp-content/uploads/2013/10/Apostila-Programacao-WEB_HTML_CSS.pdf> Acesso em: 08 Nov. 2019.

Desenvolvimento Web com HTML, CSS e JavaScript. Caelum. Disponível em: <<https://www.caelum.com.br/apostila-html-css-javascript/>> Acesso em: 08 Nov. 2019.

ALVIM, Paulo. Tirando o Máximo do Java EE 6 Open Source com jCompany© Developer Suite. 3. Ed. Belo Horizonte: Powerlogic Publishing, 2010.

FARIA, Fernanda. *et al.* Evolução e Principais Características do IDE Eclipse, Goiás, São Paulo, 2010. Disponível em <https://www.enacomp.com.br/2010/anais/artigos/completos/enacomp2010_23.pdf> Acesso em: 20 de Fevereiro de 2020.

MOREIRA JR, Normandes José; AFONSO, Alexandre. Produtividade no Desenvolvimento de Aplicações Web com Spring Boot. 2. ed. Uberlândia: AlgaWorks Softwares, Treinamentos e Serviços Ltda, 2017.

SCHMITZ, Daniel. Tudo que você queria saber sobre Git e GitHub, mas tinha vergonha de perguntar. TABLELESS, 2015. Disponível em: <<https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>>. Acesso em: 17 de Março de 2020.

DEVMEDIA, Bruno dos Santos Nascimento. MySQL. Disponível em: <<https://www.devmedia.com.br/mysql-tutorial/33309>> Acesso em: 20 jul. 2019.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: Guia do Usuário. Rio de Janeiro: Elsevier Brasil, 2006.

Pesquisa-Ação – André Fontenelle | Como fazer um TCC. Andrefontenelle.com.br. Disponível em: <<https://novo.andrefontenelle.com.br/tag/pesquisa-acao/#site-header>>. Acesso em: 25 Outubro 2020.

NAÍNA TUMELERO. Metodologia de pesquisa: guia rápido de possibilidades. Disponível em: <<https://blog.mettzer.com/metodologia-de-pesquisa/>>. Acesso em: 20 Outubro de 2020.

MATHEUS DIAS. Pesquisa Quantitativa | Metodologia, Questionário e Coleta de Dados. Instituto OPUS | Pesquisa Eleitoral, de Mercado e Opinião Pública. Disponível em: <<https://www.opuspesquisa.com/blog/tecnicas/pesquisa-quantitativa/>>. Acesso em: 20 Outubro de 2020.

MySQL, Introdução ao MySQL. Disponível em: <<https://dev.mysql.com/doc/mysql-getting-started/en/#mysql-getting-started-basic-ops>>. Acesso em: 12 novembro de 2019.

HTML & CSS projete e construa websistes. Disponível em: <https://www.academia.edu/36660331/HTML_and_CSS_projete_e_construa_websites?auto=download>. Acesso em: 11 de Março de 2020.

Getting Started, O que é um Framework? Entendendo o conceito. Disponível em: <<https://tableless.github.io/iniciantes/manual/js/o-que-framework.html>> Acesso em: 08 de novembro de 2019.

IBM, Chris Aniszczyk e David Gallardo. Introdução à Plataforma Eclipse. Disponível em: <<https://www.ibm.com/developerworks/br/library/os-eclipse-platform/index.html>> Acesso em: 02 ago. 2019.

PRODUTIVIDADE NO DESENVOLVIMENTO DE APLICAÇÕES WEB COM SPRING BOOT. Disponível em: <<http://s3.amazonaws.com/algaworks-assets/ebooks/algaworks-livro-spring-boot-v2.0>> Acesso em: 17 de Março de 2020.

DEVMEDIA, Eduardo. Desenvolvendo uma Aplicação Web com Spring Boot e Spring MVC. Disponível em: <<https://www.devmedia.com.br/desenvolvendo-uma-aplicacao-web-com-spring-boot-e-spring-mvc/34122>> Acesso em: 08 ago. 2019.

DEVLOG, Hélio Márcio Filho. Web Service RESTful Spring Boot 2.0 com múltiplos banco de dados. Disponível em: <<https://medium.com/dev-log/web-service-restful-spring-boot-2-0-com-m%C3%BAltiplos-bancos-de-dados-ff1676b412cf>> Acesso em: 12 ago. 2019

Spring By Pivotal, Apresentando o Spring. Disponível em: <<https://docs.spring.io/spring-boot/docs/2.2.1.RELEASE/reference/html/getting-started.html#getting-started>> Acesso em: 08 de novembro de 2019.

Introdução à Plataforma Eclipse. IBM, 2012. Disponível em: <<https://www.ibm.com/developerworks/br/library/os-eclipse-platform/index.html>> Acesso em: 10 set.2019.

Evolução e Principais Características do IDE Eclipse. Disponível em:

<https://www.enacomp.com.br/2010/anais/artigos/completos/enacomp2010_23.pdf>. Acesso em: 02 de Março de 2020.

Eclipse documentation - Current Release, Help - Eclipse Platform. Disponível em: <<https://help.eclipse.org/2019-09/index.jsp>> Acesso em: 08 de novembro de 2019.

GitHubGuides, O que é o GitHub?. Disponível em: <<https://guides.github.com/activities/hello-world/>> Acesso em: 14 de novembro de 2019.

GitHubGuides, Documenting your projects on GitHub. Disponível em: <<https://guides.github.com/features/wikis/>> Acesso em: 14 novembro de 2019.

Desenvolvimento de Sistemas de Informação: Da Construção de Sistemas Informáticos à Reengenharia Organizacional. Disponível em: <<http://piano.dsi.uminho.pt/~jac/SI/zdocumentos/DSI.pdf>> Acesso em: 09 de Março de 2020.

MARYLENE GUEDES. O que é MVC? - Blog da TreinaWeb. Blog da TreinaWeb. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-mvc/>>. Acesso em: 14 Novembro de 2020.

O que é Spring Security? Blog da AlgaWorks. Disponível em: <<https://blog.algaworks.com/spring-security/>>. Acesso em: 14 Novembro de 2020.

GILBERTO CALEGARI BRANDÃO. Top 10 Spring Framework Annotations para desenvolvedores Java. Medium. Disponível em: <<https://medium.com/@gcbrandao/top-10-spring-framework-annotations-para-de-senvolvedores-java-ecc1c3b11dae>>. Acesso em: 14 Novembro de 2020.

ANEXO A – FORMULÁRIO DE IDENTIFICAÇÃO

Dados do Relatório Científico	
Título e subtítulo: ItaretIF - Um portal para divulgação de informações no campus Lajes.	
Tipo de relatório: Técnico-Científico.	Data:
Título do projeto/ programa/ plano: ItaretIF.	
Autor(es): Ana Sara da Silva Clementino; Maria Izamara Faustino; Maria Luiza da Silva Matias; Willian Felipe da S. dos Santos.	
Instituição e endereço completo: Instituto Federal de Educação, Ciência e Tecnologia - Campus Lajes. BR-304, Km 120, s/n - Centro, Lajes - RN, 59535-000.	
Resumo: Após a aprovação no processo seletivo do IFRN – Lajes os estudantes ingressam na instituição sem saber como ela funciona, quais são as normas de conduta exigidas para eles e quais são os setores existentes e seus responsáveis. A maioria dessas informações podem ser encontradas no portal oficial da instituição, porém, a linguagem utilizada nos documentos e a quantidade de informações contidas neste portal, dificulta o entendimento e a pesquisa dessas normas, fazendo com que esse portal não seja atrativo para os alunos. Além disso, há também uma dificuldade em encontrar onde estão localizadas as salas, laboratórios e setores. Neste sentido, este projeto propõe a criação de um portal capaz de abrigar as informações mais solicitadas pelos estudantes ingressantes no Campus Avançado	

Lajes. Com o objetivo de proporcionar aos alunos, e também a população externa, informações sobre a instituição com a linguagem utilizada no cotidiano desses alunos, tornando o portal atrativo e de fácil entendimento.

Palavras-chave/descriptores: informática, informação, campus, Lajes, instituição, portal.

Nº de páginas:

Jornada de trabalho:

Horas semanais:

Total de horas:

Observações/notas: