

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

LORENN A BEATRIZ DE SOUZA SILVA

**IMPLEMENTAÇÃO DO SISTEMA WEB PARA DIVULGAÇÃO DA FEIRA DA
AGRICULTURA FAMILIAR DE LAJES/RN**

**LAJES/RN
2019**

LORENNA BEATRIZ DE SOUZA SILVA

**IMPLEMENTAÇÃO DO SISTEMA WEB PARA DIVULGAÇÃO DA FEIRA DA
AGRICULTURA FAMILIAR DE LAJES/RN**

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. Fernando H. L. Soares

RESUMO

A Incubadora Tecnológica para o Fortalecimento dos Empreendimentos Econômicos Solidários do IFRN (IFSOL) – Núcleo Lajes/RN tem desenvolvido suas ações em âmbito interno e externo. E em parceria com a Secretaria Municipal de Agricultura e Meio Ambiente (SEMAGMA) - Lajes/RN, um dos resultados positivos do IFSOL é a Feira da Agricultura Familiar, que desempenha papel fundamental na parceria entre os agricultores, consumidores e o poder público. Nesse contexto, este projeto tem, portanto, como objetivo a criação do sistema web, que por sua vez é uma solução a ser desenvolvidas para tornar processos manuais mais simples, rápidos e eficazes. Esse ajudará a expandir ainda mais os empreendimentos econômicos, fazendo a divulgação da Feira da Agricultura Familiar de Lajes/RN, controle de produtores e dos clientes. Logo, os agricultores divulgam os produtos, esses são encomendados e comprados. Assim os produtores trazem a quantidade certa para a cidade, o produtor não terá prejuízo ou não trará uma baixa quantidade. O sistema web será desenvolvido com o auxílio de quatro ferramentas: msyql, que ajudará no sistema de gerenciamento de banco de dados utilizado por diferentes aplicações; materialize, que ajudará no design do sistema; spring boot, que ajudará a facilitar o processo de configuração e publicação de nossas aplicações, por último o eclipse uma IDE de programação. Deste modo, no geral o projeto ocorreu tudo como planejado e com suas funcionalidades devidas, de acordo com o que foi exposto na proposta, levantamento de requisitos e arquitetura do projeto. Assim, obteve – se como resultado uma página web totalmente funcional e acessível.

Palavras-chave: IFSol. Feira. Sistema Web.

SUMÁRIO

1 INTRODUÇÃO	6
2 DADOS GERAIS DA PESQUISA/EXTENSÃO/ESTÁGIO	8
3 FUNDAMENTAÇÃO TEÓRICA	10
3.1 ECONOMIA SOLIDÁRIA	10
3.2 COMÉRCIO JUSTO	11
3.3 AUTOGESTÃO	13
3.4 DESENVOLVIMENTO WEB	14
3.4.1 MSYQL	15
3.4.2 MATERIALIZE	15
3.4.3 SPRING BOOT	17
3.4.4 ECLIPSE	17
4 METODOLOGIA	19
5 CARACTERIZAÇÃO DAS ATIVIDADES DESENVOLVIDAS	20
5.1 LEVANTAMENTO DE REQUISITOS FUNCIONAIS	20
5.1.1 LEVANTAMENTO DOS REQUISITOS NÃO FUNCIONAIS	20
5.2 PROJETO DE ARQUITETURA DO SISTEMA	21
5.2.2 DESCRIÇÃO DAS FUNCIONALIDADES DO SISTEMA	23
5.3 IMPLEMENTAÇÃO DO SISTEMA	25
5.4 APRESENTAÇÃO DO PROTÓTIPO	34
6 ANÁLISE E DISCUSSÃO DOS RESULTADOS	41
7 CONSIDERAÇÕES FINAIS	43
REFERÊNCIAS	44
ANEXO A – FORMULÁRIO DE IDENTIFICAÇÃO	46

1 INTRODUÇÃO

O projeto da IFSOL passou a ser desenvolvido no ano de 2018 no Campus Natal - Zona Norte/RN. Originalmente, foram desenvolvidas exclusivamente ações técnicas voltadas para a elaboração do planejamento e desenvolvimento das marcas desses Empreendimentos Econômicos Solidários – EES – e na concepção de novas estratégias de comercialização de produtos condizentes com o perfil do público alvo.

Com o objetivo de ampliar o número de serviços realizados e EES atendidos, bem como fortalecer institucionalmente o projeto com a criação de novos Núcleos, a IFSOL promoveu em junho de 2018 formação com servidores de diversos Campus. A proposta foi que mais interessados participassem da Incubadora e possam realizar pesquisas e prestar formação e assessoria junto aos estudantes, em empreendimentos de economia solidária e agricultura familiar, de forma conjunta e em seus respectivos campi (IFSOL, 2018).

Nesse processo de construção da identidade, após levantamento das características/expertises dos campi e servidores dos núcleos que compõem o projeto no ano de 2018, Lajes/RN ficou responsável por desenvolver, prioritariamente, ações voltadas na área de Tecnologia da Informação, por envolver os cursos de Administração e Informática.

Em abril de 2019, foi aberto processo seletivo para inclusão de membros (educandos) no projeto. Diante disso, estão envolvidos nas ações do Núcleo atualmente cinco alunas voluntárias, uma bolsista e três servidores professores.

A Incubadora Tecnológica para o Fortalecimento dos Empreendimentos Econômicos Solidários do IFRN (IFSOL) – Núcleo Lajes/RN tem desenvolvido suas ações em âmbito interno e externo.

Em parceria com a Secretaria Municipal de Agricultura e Meio Ambiente (SEMAGMA) - Lajes/RN, um dos resultados positivos do IFSOL é a Feira da Agricultura Familiar, que desempenha papel fundamental na parceria entre os agricultores, consumidores e o poder público, intensificando a valorização do trabalhador e da produção agrícola.

Tal cooperação comprova a importância do trabalho como forma de identidade, fortificação e resistência comercial, uma vez que agricultores recebem orientação e apoio para realizar e expandir as suas vendas.

Nesse sentido, o fortalecimento das feiras com apoio do IFSol, além de oferecer um espaço para a comercialização dos produtos agrícolas e valorização ao trabalhador local, traz consigo a valorização das tradicionais feiras populares, que sempre tiveram destaque em cenário nacional. O espaço reservado garante processos de socialização e encontros permitindo, assim, o fortalecimento da economia popular e solidária.

Este projeto tem, portanto, como objetivo a criação do sistema web, que por sua vez é uma solução a ser desenvolvidas para tornar processos manuais mais simples, rápidos e eficazes. Esse, ajudará a expandir ainda mais os empreendimentos econômicos, fazendo a divulgação da Feira da Agricultura Familiar de Lajes/RN, controle de produtores e dos clientes.

Para isso, o Sistema Web contará com três perfis: administrador, produtor e cliente e se utilizará dos seguintes requisitos: autenticação, quando o usuário consegue fazer login no sistema e autorização, quando, de acordo com o perfil do usuário, há restrição de acesso a determinadas páginas do sistema.

Os agricultores divulgam os produtos, esses são encomendados e comprados. Assim os produtores trazem a quantidade certa para a cidade, o produtor não terá prejuízo ou não trará uma baixa quantidade.

O sistema web será desenvolvido com o auxílio de quatro ferramentas: msql, que ajudará no sistema de gerenciamento de banco de dados utilizado por diferentes aplicações; materialize, que ajudará no design do sistema; spring boot, que ajudará a facilitar o processo de configuração e publicação de nossas aplicações, por último o eclipse uma IDE de programação.

2 DADOS GERAIS DA PESQUISA EXTENSÃO

Título do projeto: Incubadora Tecnológica para o Fortalecimento dos Empreendimentos Econômicos Solidários do IFRN (IFSol): Fortalecendo os Empreendimentos Econômicos Solidários (EES) do Rio Grande do Norte.

Período de realização: 01/04/2019 à 01/12/2019

Total de horas: 340hrs

Orientador: Prof. Fernando Helton Linhares Soares

Função: Professor

Formação profissional: Mestre

Quadro 1 – Síntese das Atividades do Aluno no Projeto.

CARGA HORÁRIA	ATIVIDADES DESENVOLVIDAS	RESULTADOS ALCANÇADOS
15h	Promover oficina de formação sobre economia solidária com participação dos servidores do campus: oficina programada para ser realizada durante a II SEMADEC do Campus Lajes.	ALCANÇADO
15h	Realizar visita a empreendimento econômico solidário.	ALCANÇADO
15h	Criar identidade visual da feira no campus: cartaz elaborado e adaptado conforme realização de feiras.	ALCANÇADO
15h	Promover oficina com produtores participantes.	ALCANÇADO
15h	Atender 1 EEs de forma efetiva (diagnóstico e/ou assessoria).	ALCANÇADO
15h 15h	Levantamento de requisitos do sistema.	ALCANÇADO
15h	Projeto de arquitetura do sistema.	ALCANÇADO

15h	Implementação do sistema.	ALCANÇADO
15h	Testes e validação do sistema.	NÃO ALCANÇADO

Fonte: autoria própria (2019).

3 FUNDAMENTAÇÃO TEÓRICA

3.1 ECONOMIA SOLIDÁRIA

A Incubadora para o Fortalecimento dos Empreendimentos Econômicos Solidários do IFRN (IFSol) atua desde 2018 no IFRN e tem como função fortalecer os Empreendimentos Econômicos Solidários (EES) do Rio Grande do Norte. O núcleo da IFSol do Campus Lajes foi criado também em 2018 e atua principalmente no fortalecimento da Feira da Agricultura Familiar de Lajes/RN, que funciona no IFRN Campus Lajes semanalmente.

A economia solidária é um eixo de pesquisa e desenvolvimento que faz parte deste projeto, principalmente baseada nas ideias de Vergara e Peci (2013), MTE (2007) e Cunha (2003), que exemplificam concepções sobre o conceito de economia solidária e suas aplicações no ramo do empreendedorismo econômico.

As pesquisas científicas se inserem em um sistema de regras e estratégias voltadas para a inovação, de modo que é observado e descoberto em seu objeto, ou seja, surge como resultado de interação com técnicas. Neste caso, as pesquisas científicas influenciam diretamente na economia solidária, quando a construção dos pressupostos paradigmas para a análise de uma determinada lógica ou estrutura de pesquisa parte de múltiplas dimensões estratégicas que quando concluídas são voltadas para a sociedade em que vivemos, ajudando em controles de monitoramento sociais e econômicos. Como observado por Vergara e Peci (2003, p. 15),

a racionalidade de uma estratégia particular de pesquisa é baseada em uma rede de pressupostos implícitos ou explícitos, relativos à ontologia e à natureza humana que definem o ponto de vista do pesquisador sobre o mundo social. Tais pressupostos fornecem os fundamentos da prática de pesquisa, inclinando o pesquisador a ver e a interpretar o mundo com base em uma perspectiva, ao invés de outra. (VERGARA; PECI, 2003, p. 15).

A economia solidária cresceu no país a partir da articulação da “esquerda” brasileira, com o objetivo diferente de produzir, vender, comprar e trocar o que é preciso para viver de forma confortável, sem explorar os outros querendo levar vantagem e sem destruir o ambiente. Entre as entidades no país que apoiam as ações da economia solidária, estão a Central de Cooperativas e Empreendimentos

Solidários do Brasil e a Central Única dos Trabalhadores (CUT), organizações apoiadas pelas Incubadoras Universitárias e escolas Técnicas Federais e Estaduais, e por setores da Igreja que envolvem o comércio, consumo e clubes de troca. Contudo, podemos observar que um “Movimento por uma Economia Solidária” possui algumas características comuns. No Brasil, as iniciativas denominadas economia solidária compreendem:

Os Empreendimentos Econômicos Solidários e suas formas de organização: cooperativas, associações, empresas auto gestionárias, redes e cadeias produtivas, complexos cooperativos, centrais de comercialização e outras formas de organização do trabalho; Redes, Fóruns e Coletivos diversos de Economia Solidária, a relação entre empreendimentos, entidades de apoio e poder público; Finanças Solidárias: cooperativas de crédito, microcrédito, fundos rotativos, bancos comunitários, aval solidário etc.; Comercialização: comércio ético, justo e solidário, feiras, clubes de troca, consumo consciente; as Entidades e organizações da sociedade civil, entidade de fomento e apoio e outras organizações e movimentos sociais. (MTE, 2007).

Podemos observar que o território brasileiro apresenta uma grande diversidade social, econômica e ambiental, com isso os gestores públicos, que atuam na economia solidária procura sempre preservar as relações (econômicas, sociais, políticas, culturais) presentes em cada região, pelo conhecimento que os gestores têm em suas regiões, com questões de diversidades e desigualdades, vão modelando suas ações solidárias e forma de trabalhos que valorizam a solidariedade e a identidade territorial. É de suma importância as ações locais na economia solidária. Como observado por Cunha (2003, p. 69):

Iniciativas locais isoladas correm o risco de não gerar a economia solidária, ainda que bem-sucedidas individualmente, por isso é fundamental sistematizar experiências, divulgar ideias e transformá-las em propostas, o que também é movimento essencialmente político. E é só se integrando a debates políticos mais amplos, e ampliando a politização entre suas unidades, que a economia solidária pode garantir sua dimensão política e se converter em proposta concreta de desenvolvimento alternativo e transformação social. (CUNHA, 2002, p. 69).

3.2 COMÉRCIO JUSTO

O comércio justo, por sua vez, é um movimento social que busca promover padrões produtivos e comerciais responsáveis e sustentáveis assim como oportunidades de desenvolvimento para os pequenos agricultores(as), camponeses(as) e artesãos em desvantagem econômica e social, com relação aos fatores dominantes no mercado.

Neste projeto, o comércio justo foi abordado principalmente baseando-se nas ideias de FINE (2005) e WFTO (2013), que exemplificam concepções sobre o conceito de economia solidária e suas atribuições para facilitar o comércio entre os pequenos agricultores e consumidores.

A construção teórica de comércio justo surgiu para tentar sanar as injustiças e o desequilíbrio social existente nas décadas de 60 e 70. Esses problemas surgiram a partir do comércio internacional. Um dos momentos marcantes na história foi as negociações entre Menninite Central Committee com as comunidades pobres do sul. Vale ressaltar que esse movimento tem crescido significativamente desde aquela época, mas ainda continua a ser relativamente pequeno em termos de negócios no mercado. Já entre a comunidade acadêmica há uma atenção maior, destacando as áreas de marketing, economia, design e agricultura, que trabalham bastante esse movimento.

Vale ressaltar também que o conceito de Comércio Justo não possui uma única definição, mas diferentes definições que se coadunam de forma a tentar definir melhor o seu conceito em acordo com as diferentes entidades internacionais.

Definição segundo a FINE (2005):

O Comércio Justo é uma relação de troca, baseada no diálogo, na transparência e no respeito, que busca maior igualdade no comércio internacional. Contribui ao desenvolvimento sustentável oferecendo melhores condições comerciais; assegurando o direito dos pequenos produtores e trabalhadores marginalizados, especialmente do Sul. Organizações de Comércio Justo estão envolvidas ativamente em apoiar produtores, divulgar o comércio justo e em campanhas para modificar as regras e as práticas do mercado convencional internacional. (FINE, 2005).

Definição da WFTO (2013):

O Comércio Justo é uma parceria comercial baseada no diálogo, transparência e respeito, que busca maior equidade no comércio internacional. Contribui para o desenvolvimento sustentável, oferecendo melhores condições comerciais, e garantir os direitos dos produtores e trabalhadores marginalizados - especialmente no sul. Organizações de Comércio Justo, apoiadas pelos consumidores, estão envolvidos ativamente no apoio aos produtores, na sensibilização e nas campanhas para mudanças nas regras e práticas do comércio internacional convencional. (WFTO, 2013).

É indiscutível que os pequenos produtores são os que mais sofrem com o mercado. As regras não são justas para quem produz em pequena quantidade. O movimento do comércio justo surge justamente para garantir uma remuneração justa para os trabalhadores.

3.3 AUTOGESTÃO

A autogestão é um conjunto de práticas organizacionais que buscam distribuir a autoridade, dando clareza de responsabilidades e o máximo de autonomia a cada integrante da organização.

Assim, o conceito de autogestão foi fundamental para nossa pesquisa, para que pudesse mostrar que há organizações que ajudam o empoderamento dos pequenos agricultores. Para a construção de ideias sobre esses conceitos, foi utilizado as concepções sobre Singer, SENAES e Storch, que falam sobre a contribuição dada aos agricultores.

Segundo Singer (2000), a autogestão pode ser compreendida como uma forma de promoção da democracia em instituições sociais em que trabalhadores são os sócios e os sócios são os trabalhadores.

A autogestão, antes de tudo, é uma prática social e política, pautada na mobilização de trabalhadores das mais diferentes áreas de atuação, que retoma ainda, a ideia de Rosa Luxemburgo sobre o fato de que, é agindo de forma coletiva que as massas aprendem a se autogerir, logo, uma sociedade autogestionária, é aquela que se institui, experimenta e constrói por si mesma. A economia solidária traz consigo a valorização da vida, do meio ambiente, da igualdade, da justiça social, da autogestão e da democracia, ultrapassando as questões econômicas, retomando práticas coletivas de organização do trabalho e da produção e reprodução do trabalho, que visam alçar, como nas experiências agroecológicas, soberania e segurança alimentar, outro modelo de desenvolvimento com centralidade na sustentabilidade humana (SENAES, 2004; 2007).

A autogestão segundo Storch (1987) envolve a participação direta no controle e gestão do empreendimento e não somente a propriedade do mesmo. Num ambiente cooperativo, característico das comunidades rurais, observa-se o princípio de

autogestão é condição *sinequa non* à participação de todos os cooperados. No contexto empresarial, autogestão caracteriza-se pela autonomia dos empregados nas tomadas de decisões que seguem pelo processo de planejamento interno, trabalho e renda. Em outras palavras, é a gestão democrática.

3.4 DESENVOLVIMENTO WEB

A área de desenvolvimento web vem crescendo cada vez mais em nossa sociedade. Como forma de deixar mais prática e rápida as atividades, além de controle e divulgação de informações. Neste caso, o desenvolvimento web é utilizado para construção de um site que controla o produtor, o usuário, os clientes e os produtos a serem vendidos.

Como abordado no site Kingly (2017): “Os sistemas criados por Desenvolvimento Web são uma alternativa aos softwares, por serem utilizados diretamente pelo navegador de internet, sem a necessidade de fazer download dos programas e instalar no computador, bastando ter o endereço (URL) do site desejado e utilizá-lo no navegador de qualquer dispositivo.”

Logo, no intuito de propor dinamismo as páginas Web, que antes eram estáticas e com a necessidade de uma maior rapidez em seus desenvolvimentos, várias tecnologias surgiram para tornar a web um ambiente mais agradável, interativo e acessível.

Sobre o uso da web, em que sistemas começaram a ser implementados de forma dinâmica, com isso o intuito de sanar todos os problemas enfrentados e auxiliando assim seus usuários declara:

O uso da Web explodiu no meio da década de 90 depois do primeiro navegador gráfico ter aparecido. A necessidade de computação associada com documentos HTML, que, por si só, são completamente estáticos, rapidamente, tornou-se crítica. A computação do lado do servidor foi possível com o Common Gateway Interface (CGI), que permitiu a documentos HTML requisitar a execução de programas no lado do servidor, com o resultado da computação retornando para o navegador em forma de documentos HTML. Computação do lado do navegador tornou-se disponível com o advento dos applets Java. Ambas destas abordagens estavam sendo lentamente substituídas por novas tecnologias em grande parte, por linguagens de script.

Portanto, o sistema web será desenvolvido com o auxílio de quatro ferramentas:

3.4.1 MSYQL

O MySQL é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

O sistema foi desenvolvido pela empresa sueca MySQL AB e publicado, originalmente, em maio de 1995. Logo depois, a empresa foi comprada pela Sun Microsystems e, em janeiro de 2010, integrou a transação bilionária da compra da Sun pela Oracle Corporation.

Para utilizar o MySQL, é necessário instalar um servidor e uma aplicação cliente. O servidor é o responsável por armazenar os dados, responder às requisições, controlar a consistência dos dados, bem como a execução de transações concomitantes entre outras. O cliente se comunica com o servidor através da SQL. A versão gratuita do MySQL é chamada de Edição da Comunidade e possui o servidor e uma interface gráfica cliente.

O servidor deve ser instalado e configurado para receber conexões dos clientes. No MySQL, o principal cliente é a interface gráfica cliente fornecida pela Oracle, denominada MySQL Workbench. Através do MySQL Workbench, pode-se executar consultas SQL, administrar o sistema e modelar, criar e manter a base de dados através de um ambiente integrado. O MySQL Workbench está disponível para Windows, Linux e Mac OS.

Na interface de modelagem de base de dados do MySQL Workbench, pode-se definir as entidades da base de dados, seus atributos e relacionamentos. Em banco de dados, deve-se definir configurações importantes para os bancos de dados, como as chaves primárias e estrangeiras e os atributos que devem ser indexados.

3.4.2 MATERIALIZE

O Materialize é um dos frameworks mais famosos entre os desenvolvedores Front-End, e também um dos mais solicitados. Ele é um framework Front-End que resolve os mesmos problemas, mas, claro, com suas próprias características.

“Desde o lançamento, sua metodologia é a que mais se destaca no mercado de design para web.” (LATYPOV, 2017)

Vale salientar que o Materialize, por sua vez, está dando os primeiros passos e seguindo por um caminho um pouco diferente, para também conquistar seu espaço.

O grande foco do Materialize é o visual. Para isso, ele utiliza as filosofias de design do Material Design do Google, tecnologia que foca na simplicidade para a indicação de como criar interfaces de usuário modernas. A preocupação está em como os elementos interagem um com o outro e com o usuário. Se analisarmos mais a fundo, veremos que a especificação desse framework é muito mais completa do que isso, envolvendo muitas outras coisas, como comportamento físico (movimentos em duas e três dimensões), luz, espaço, etc., para que se obtenha uma uniformidade de reações dos elementos entre as plataformas, ou seja, um botão, por exemplo, tenha o mesmo comportamento em diferentes navegadores e mesmo em diferentes sistemas operacionais, como Windows e Google Android.

A primeira versão do material foi apresentada em 2014 no sistema operacional Android e sua implementação é fortemente baseada em objetos do mundo real. Seus elementos de design são sólidos, impenetráveis, mutáveis, dobráveis e se movimentam nos 3 eixos cartesianos. (RALLO, 2017).

Também é de suma importância apresentar o que é um Framework. Ele consiste em um conjunto de ferramentas prontas que são utilizadas por programadores para agilizar o desenvolvimento de sites e programas, tendo em vista que ele não gastará a maior parte do tempo com estilização e design, focando seu tempo somente na funcionalidade das ferramentas. Como muitos pensam, o framework não é uma palavra restrita da programação Web. Apesar de que é muito utilizado neste ambiente, é empregado geralmente em qualquer ramo de desenvolvimento de software.

3.4.3 SPRING BOOT

O Spring Boot busca solucionar a complexidade da inicialização e gerenciamento de dependências de um projeto com Spring, além de tratar de maneira coesa e eficiente a questão da configuração.

Essencialmente, o Spring Boot pode ser considerado um plugin para a ferramenta de building, seja ela o Maven ou o Gradle. Seus principais objetivos são gerenciar dependências de maneira opinativa e automática, e simplificar a execução do projeto em tempo de desenvolvimento e depuração.

O principal benefício do Boot, entretanto, é a configuração de recursos baseada no que se encontra no classpath. Se o POM de seu Maven inclui a dependência do JPA e o driver do PostgreSQL, ele irá criar uma unidade de persistência baseada no PostgreSQL.

Se adicionarmos alguma dependência web, iremos perceber que o Spring MVC assumirá configurações default e dependências com relação a diversos aspectos, como a tecnologia de apresentação (o default é o Thymeleaf), o mapeamento de recursos e marshalling de JSON (o default é o Jackson) e/ou XML (o default é o JAXB 2) para o tratamento de dados de requisição e resposta, que necessitamos em uma aplicação REST, por exemplo.

Em suma, o que o Spring Boot faz principalmente, de maneira muito elegante, é nos livrar de qualquer preocupação desnecessária com dependências e configurações que são as mesmas em 99% dos casos. E se o caso está no 1% restante, ele acatou nossa opinião com a mesma elegância, como veremos em nosso projeto.

3.4.4 ECLIPSE

O Eclipse é uma IDE, ferramenta de desenvolvimento de aplicações Java, mas também suporta outras linguagens como C/C++, PHP, Python, Scala e plataforma Android. Possui um gigantesco número de plug-ins gratuitos disponíveis,

para adicionar e ampliar mais recursos ao IDE também possui refactoring, geração de códigos, atalhos e ícones intuitivos.

Existem várias razões para utilizar o Eclipse, dentre elas destacam-se a robustez e desempenho dentro dos programas Webs criados, além da facilidade em sua instalação, configuração e uso. A diversidade de opções de plug-ins para escolha, proporciona total liberdade para aprimorar cada vez mais esse IDE deixando mais fácil e atrativo a programação sua plataforma.

O Eclipse Possibilita a geração automática de entidades a partir de um schema de dados. A versão 3.5 é totalmente compatível com a versão anterior, permitindo assim uma fácil migração. "O Eclipse é um dos IDE's mais conceituados para a programação Java, por consequência um dos mais utilizados. (MOREIRA, 2009).

4 METODOLOGIA

Este projeto consiste num trabalho original explicativo de natureza mista, conforme definido em Wazlawick (2014). A abordagem Design Thinking para Educadores (IDEO, 2010) foi utilizada como inspiração para definição desta metodologia. Neste serão seguidos os seguintes procedimentos metodológicos.

Etapa 1: Descoberta e Interpretação	Realizar survey com especialistas do assunto, como os professores e coordenadores do IFSol.
	Realizar revisão bibliográfica sobre estudos primários relacionados aos conceitos empregados no IFSol.
	Definir modelo de domínio do sistema a ser construído.
Etapa 2: Ideação e Experimentação	Levantamento de Requisitos do sistema.
	Projeto de Arquitetura do sistema.
	Implementação do sistema.
Etapa 3: Publicação e Entrega de Relatório.	Teste e Validação do sistema.
	Sistematizar e analisar resultados desta pesquisa.
	Promover resultados desta pesquisa junto à comunidade do IFSol.
	Entregar relatório final deste trabalho.

5 CARACTERIZAÇÃO DAS ATIVIDADES DESENVOLVIDAS

5.1 LEVANTAMENTO DOS REQUISITOS FUNCIONAIS

- **RF001 - Cadastro e Controle de Usuário:** O sistema deve permitir a inclusão de usuários. Usuário terá em seu cadastro (Nome, Senha, E-mail, CPF, Perfil.). O perfil dentro do sistema pode assumir 3 tipos: Produtor, Comprador e Administrador. O produtor tem a permissão de cadastrar produtos. O Comprador tem a permissão de encomendar produtos e cancelar pedido caso desista da compra. O administrador tem a permissão de acesso ao relatório final com controle de dados estatísticos de quantidades de produtos vendidos, além disso ele pode gerenciar as informações publicadas pelo produtor, para corrigir eventuais raro.
- **RF002 - Cadastro e Controle de Produto:** O sistema deve permitir a inclusão de produtos. O produto terá o nome, quantidade, a descrição, preço, e o nome do produtor associado. Além disso, o sistema deve permitir a alteração, listagem e exclusão.
- **RF003 - Geração de Relatório:** O sistema deve permitir a geração de relatórios, como o relatório das pessoas que encomendaram os produtos, para que o produtor possa saber os nomes e informações e o relatório ao final de cada produtor após o término da feira para que o administrador possa ter controle de vendas de cada produtor.
- **RF004 - Geração de Encomendas:** O sistema deve permitir a encomenda de produtos por parte dos clientes. Dentro do sistema este usuário possui o perfil de comprador. Numa encomenda deve constar o produto e quantidade.

5.1.1 LEVANTAMENTO DOS REQUISITOS NÃO FUNCIONAIS

- **RNF001** - O sistema deve ter conexão à internet;
- **RNF002** - O sistema é desenvolvido com as tecnologias WEB;
- **RNF003** - O sistema deve utilizar Banco de Dados (Mysql).

5.2 PROJETO DE ARQUITETURA DO SISTEMA

O IFSol – Web é um sistema web que permite o gerenciamento do fluxo de encomendas e vendas entre produtores rurais e potenciais clientes compradores. Por meio desse sistema, é possível o produtor rural dimensionar o quantitativo de produtos que deverá levar à feira para que sejam vendidos. As funcionalidades do IFSol – são exibidas no **Quadro 1 - Funcionalidades do IFSol - Web.**

Quadro 1 - Funcionalidades do IFSol - Web

Módulo	Funcionalidades
Produto	Cadastro dos produtos
	Consulta dos produtos
	Alteração dos produtos
	Exclusão dos produtos
Usuário	Cadastro dos usuários
	Consulta dos usuários
	Exclusão dos produtos
Encomenda	Encomendar produtos
Venda	Registro de vendas
Relatórios	Consulta às encomendas



Fonte: autoria própria (2019)

As funcionalidades de um sistema podem ser descritas formalmente através de casos de uso.

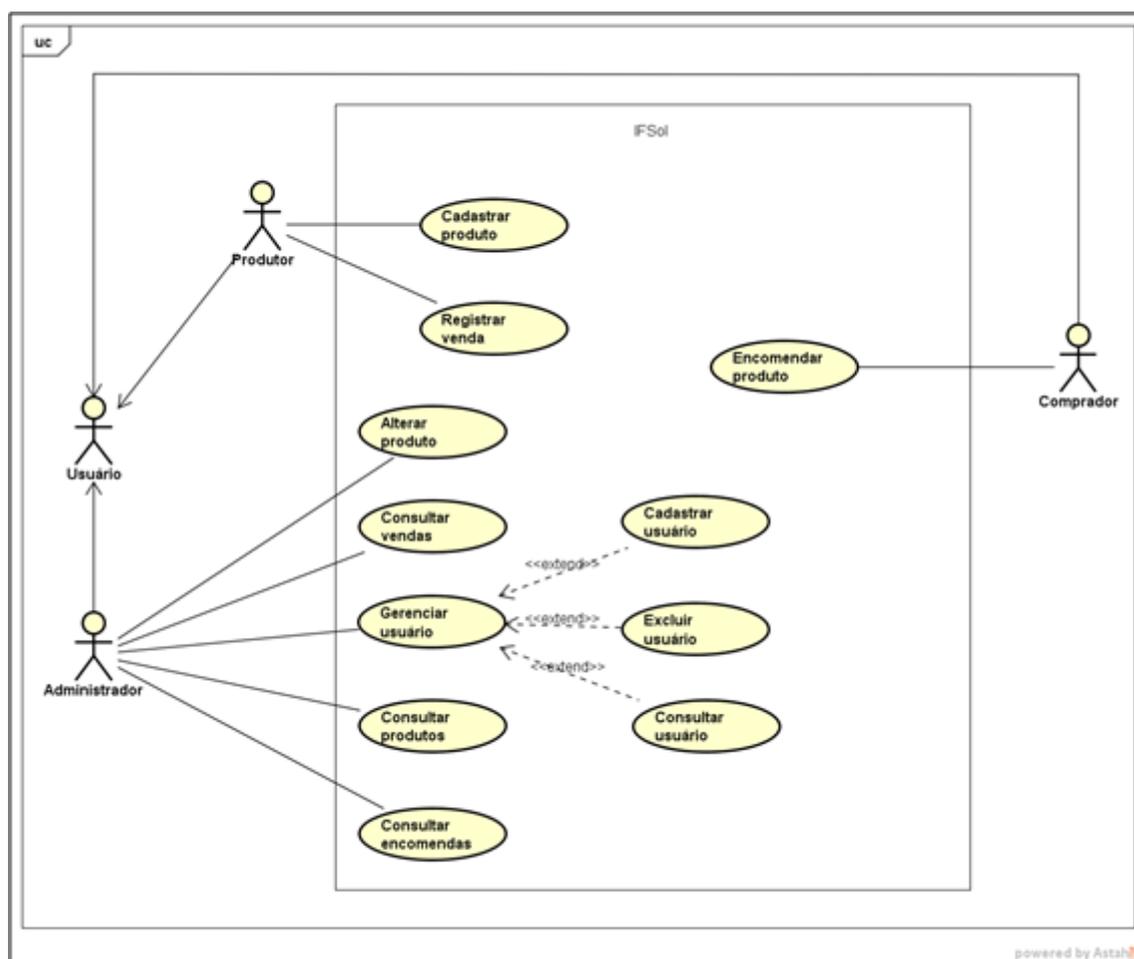
Modelos de caso de uso (use cases) são modelos passíveis de compreensão tanto por desenvolvedores – analistas, projetistas, programadores e testadores – como pela comunidade usuária – clientes e usuários. Como o próprio nome sugere, um caso de uso é uma maneira de usar o sistema. Usuários interagem com o sistema, interagindo com seus casos de uso. (FALBO; BARCELLOS, p. 25).

Por se tratar de uma descrição de alto nível, os casos de uso podem ser utilizados tanto para a apresentação das funcionalidades do sistema, quanto para a “negociação” dos requisitos do sistema junto ao cliente.

Ainda segundo Falbo e Barcellos (p. 25), os casos de uso são descrições textuais das funcionalidades de um sistema, ainda que essa descrição possa ser construída e apresentada recorrendo-se a linguagens de modelagem como a UML (*Unified Modeling Language*). O diagrama de caso de uso é um dos nove diagramas da UML (DEV MEDIA, 2012).

Na figura abaixo é apresentado o diagrama de casos de uso para o sistema IFSol – Web. Dessa forma podemos visualizar as funcionalidades do sistema e sua relação com os atores – usuários do sistema.

Figura 1 - Diagrama de casos de uso do IFSol – Web



Fonte: autoria própria (2019)

De acordo com o exposto no diagrama, é possível perceber que o sistema possui níveis de acesso diferente. O IFSol – Web possui três tipos de usuário, a saber: administrador, comprador e produtos. Ambos desempenham funções distintas e com base em seu nível de acesso. O usuário com acesso de administrador é possível cadastrar outros usuários, consulta-los e excluí-los. Além dessa permissão, o usuário administrador tem acesso aos relatórios de encomendas e vendas, bem como a alteração nas informações de um produto cadastrado no sistema. O usuário com acesso de produtor pode cadastrar um novo produto e registrar vendas. Já o acesso de comprador permite encomendar um produto.

5.2.2 DESCRIÇÃO DAS FUNCIONALIDADES DO SISTEMA

O IFSol – Web possui as seguintes URLs que por meio delas são disponibilizados os recursos do sistema.

URL	Descrição da funcionalidade
/cadastrarUsuario	Insere um novo usuário no sistema.
/usuários	Permite visualizar os usuários cadastrados no sistema.
/cadastrarProduto	Insere produtos no sistema
/produtos	Permite visualizar os produtos cadastrados no sistema e a partir disso, alterar informações dos produtos ou excluir.
/registrarEncomenda	Permite a encomenda de produtos.
/encomendas	Relatório contendo as encomendas.
/registrarVenda	Permite registrar vendas.
/vendas	Relatório contendo as vendas registradas.

5.3 IMPLEMENTAÇÃO DO SISTEMA

Figura 3 - Implementação do ControllerEncomenda

```
@@ -0,0 +1,92 @@  
+ package com.example.ifsol.controllers;  
+  
+ import java.util.ArrayList;  
+  
+ import org.springframework.beans.factory.annotation.Autowired;  
+ import org.springframework.security.core.Authentication;  
+ import org.springframework.security.core.context.SecurityContextHolder;  
+ import org.springframework.stereotype.Controller;  
+ import org.springframework.web.bind.annotation.RequestMapping;  
+ import org.springframework.web.bind.annotation.RequestMethod;  
+ import org.springframework.web.bind.annotation.RequestParam;  
+ import org.springframework.web.servlet.ModelAndView;  
+  
+ import com.example.ifsol.models.Encomenda;  
+ import com.example.ifsol.models.ItemEncomendaProduto;  
+ import com.example.ifsol.models.Produto;  
+ import com.example.ifsol.models.Usuario;  
+ import com.example.ifsol.repository.EncomendaRepository;  
+ import com.example.ifsol.repository.ItemEncomendaProdutoRepository;  
+ import com.example.ifsol.repository.ProdutoRepository;  
+ import com.example.ifsol.repository.UsuarioRepository;  
+  
+ @Controller  
+ public class EncomendaController {  
+  
+     ArrayList<ItemEncomendaProduto> produtos = new ArrayList<ItemEncomendaProduto>();  
+  
+     @Autowired  
+     private ProdutoRepository pr;  
+  
+ }
```

A partir da figura 3, pode-se perceber que no contexto do IFSol – Web uma encomenda (Figura 4) é realizada por um usuário que possui o perfil de comprador. Uma encomenda é tratada como uma entidade, logo há uma classe Java que representa essa entidade. Essa classe possui um atributo do tipo Usuário, que também é outra classe do sistema, e uma lista de itens da encomenda, representada pelo o atributo **itens_encomendas**.

Figura 4

```

@Entity
public class Encomenda implements Serializable{

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int codigoEncomenda;

    @OneToOne
    private Usuario usuario;

    @OneToMany
    List<ItemEncomendaProduto> itens_encomenda = new ArrayList<ItemEncomendaProduto>();
}

```

Fonte: autoria própria (2019)

O fluxo dessa funcionalidade começa quando o usuário acessa a página de registro de encomenda (Figura 5).

Figura 5: ModelEncomenda

```

@RequestMapping(value="/registrarEncomenda", method=RequestMethod.GET)
public ModelAndView formRegistrarEncomenda() {
    ModelAndView mv = new ModelAndView("encomenda/formularioEncomendarProdutos");
    Iterable<Produto> produtos = pr.findAll();
    mv.addObject("produtos", produtos);
    return mv;
}

```

Fonte: autoria própria (2019)

Quando o usuário acessa o menu principal através do link referente ao registro de encomendas, a requisição **/registrarEncomenda** com o tipo **GET** é lançada ao **controller** da classe encomenda. Esse **controller** recebe essa requisição através do método **formRegistrarEncomenda** e retorna a página (Figura 6) com o formulário de encomenda de produtos. Nesse retorno é adicionada uma lista com todos os produtos cadastrados no banco de dados, para que o usuário possa selecionar quais produtos deseja encomendar.

Figura 6: Tela de registro de encomendas

IF-Sol - Encomendar produtos

Carrinho

Abacaxi

1.5

 ENCOMENDAR

Manga

5.0

 ENCOMENDAR

Fonte: autoria própria (2019)

O usuário clica no botão de encomendar e escolhe a quantidade do produto que deseja e adiciona esse produto à encomenda. Essa ação lança uma requisição ao servidor pedindo para que o produto seja adicionado.

Figura 7: Código do ControllerEncomenda

```
@RequestMapping(value="/addItemEncomenda", method=RequestMethod.POST)
public String addItemEncomenda(@RequestParam("quantidade") String quantidade, @RequestParam("codigo") String codigo) {
    Produto produto = pr.findByCodigo(Integer.parseInt(codigo));
    ItemEncomendaProduto item = new ItemEncomendaProduto();
    item.setProduto(produto);
    item.setQuantidade(Integer.parseInt(quantidade));
    produtos.add(item);

    return "redirect:/registrarEncomenda";
}
```

Fonte: autoria própria (2019)

No **controller** da classe encomenda (Figura 7), o método **addItemEncomenda** recebe a requisição do tipo POST. Esse método recebe a quantidade e o código do produto escolhido pelo o usuário.

O método **addItemEncomenda** pesquisa o produto pelo código passado pela requisição. Essa pesquisa é viabilizada por meio da interface de repositório referente a classe produto. O produto é pesquisado, cria-se uma instância da classe `ItemEncomendaProduto`, setamos o produto e quantidade. Além disso esse item é armazenado em tempo de execução na memória do computador por meio de um `ArrayList`.

Figura 8: RepositoryEncomenda

```
import org.springframework.data.repository.CrudRepository;

public interface ProdutoRepository extends CrudRepository<Produto, String>{

    Produto findByCodigo(int codigo);

}
```

Fonte: autoria própria (2019)

Quando utilizamos o Spring Boot, podemos criar uma interface que estende a classe `CrudRepository`, logo teremos como utilizar os métodos de acesso a banco de dados que essa classe fornece e podemos também reescrever alguns deles, como por exemplo, o método de pesquisa (Find).

O usuário pode escolher diversos produtos e adicionar ao “carrinho” (a listagem de produtos em memória). Ao escolher todos os produtos que deseja, o usuário acessa o carrinho através do link fornecido na interface.

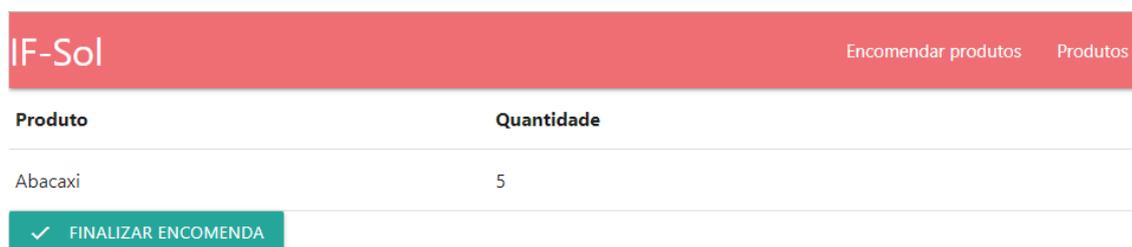
Figura 9: ModelEncomenda

```
@RequestMapping(value="/carrinho", method=RequestMethod.GET)
public ModelAndView carrinho() {
    ModelAndView mv = new ModelAndView("encomenda/carrinho");
    mv.addObject("produtos", produtos);
    return mv;
}
```

Fonte: autoria própria (2019)

O **controller**, então, recebe a requisição e retorna todos produtos da listagem que foi preenchida pelo o usuário, como apresentado na figura acima.

Figura 10: Lista de Encomendas



Produto	Quantidade
Abacaxi	5

✓ FINALIZAR ENCOMENDA

Fonte: autoria própria (2019)

Uma vez que o usuário tem todos os produtos que deseja encomendar, no carrinho, pode então finalizar a encomenda clicando no botão. A figura acima exemplifica a tela referente ao carrinho e o botão de finalizar a encomenda.

Quando o botão é clicado, outra requisição é feita ao servidor e tratada

pele **controller** da classe encomenda. O método que trata essa requisição é apresentado na figura a abaixo.

Figura 11: ControllerEncomenda

```
@RequestMapping(value="/finalizarEncomenda", method=RequestMethod.POST)
public String finalizarEncomenda() {

    Authentication authentication = (Authentication) SecurityContextHolder.getContext().getAuthentication();
    System.out.println(authentication.getName());

    Encomenda encomenda = new Encomenda();
    Usuario usuario = ur.findByEmail(authentication.getName());
    encomenda.setUsuario(usuario);

    for(int i=0; i<produtos.size(); i++) {
        ItemEncomendaProduto item = produtos.get(i);
        encomenda.setItens_produtos(item);
        iepr.save(produtos.get(i));
        produtos.get(i).setEncomenda(encomenda);
        er.save(encomenda);
    }
    produtos.clear();
    return "redirect:/registrarEncomenda";
}
```

O método finalizar encomenda executa as seguintes ações:

- Pega o contexto da autenticação;
- Cria um objeto da classe Encomenda;
- Recupera o usuário do banco de dados através do nome de usuário logado;
- Seta o atributo usuário da classe encomenda;
- Salva todos os itens da encomenda;
- Salva a encomenda no banco de dados.

Logo, todos os outros *controllers* e *cruds* seguem a mesma lógica.

Figura 12 - Implementação do ControllerProduto

```
+ package com.example.ifsol.controllers;
+
+ import org.springframework.beans.factory.annotation.Autowired;
+ import org.springframework.stereotype.Controller;
+ import org.springframework.web.bind.annotation.PathVariable;
+ import org.springframework.web.bind.annotation.PostMapping;
+ import org.springframework.web.bind.annotation.RequestMapping;
+ import org.springframework.web.bind.annotation.RequestMethod;
+ import org.springframework.web.servlet.ModelAndView;
+
+ import com.example.ifsol.models.Produto;
+ import com.example.ifsol.repository.ProdutoRepository;
+
+ @Controller
+ public class ProdutoController {
+
+     @Autowired
+     private ProdutoRepository pr;
+
+     @RequestMapping(value="/cadastrarProduto", method=RequestMethod.GET)
+     public String formulario() {
+         return "produto/formularioCadastroProdutos";
+     }
+
+     @RequestMapping(value="/cadastrarProduto", method=RequestMethod.POST)
+     public String formulario(Produto produto) {
+         pr.save(produto);
+         return "redirect:/cadastrarProduto";
+     }
+
+     @RequestMapping(value = "/produtos")
```

Fonte: autoria própria (2019)

Figura 13 - Implementação do ControllerProduto

```

+
+ @RequestMapping(value = "/produtos")
+ public ModelAndView listaProdutos() {
+     ModelAndView mv = new ModelAndView("produto/produtosCadastrados");
+     Iterable<Produto> produtos = pr.findAll();
+     mv.addObject("produtos", produtos);
+     return mv;
+ }
+
+ @RequestMapping("/excluirProduto")
+ public String excluirProduto(int codigo) {
+     Produto produto = pr.findByCodigo(codigo);
+     pr.delete(produto);
+     return "redirect:/produtos";
+ }
+
+ @RequestMapping(value =("/{codigo}", method = RequestMethod.GET)
+ public ModelAndView editarProduto(@PathVariable("codigo") int codigo) {
+     Produto produto = pr.findByCodigo(codigo);
+     ModelAndView mv = new ModelAndView("produto/formularioEditarProduto");
+     mv.addObject("produto", produto);
+     return mv;
+ }
+
+ @PostMapping("atualizar/{codigo}")
+ public String atualizarProduto(@PathVariable("codigo") int codigo, Produto produto) {
+     pr.save(produto);
+     return "redirect:/produtos";
+ }
+ }

```

Fonte: autoria própria (2019)

Figura 14 - Implementação do ControllerUsuário

```

@@ -0,0 +1,47 @@
+ package com.example.ifsol.controllers;
+
+ import java.util.Iterator;
+
+ import org.springframework.beans.factory.annotation.Autowired;
+ import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
+ import org.springframework.stereotype.Controller;
+ import org.springframework.web.bind.annotation.RequestMapping;
+ import org.springframework.web.bind.annotation.RequestMethod;
+ import org.springframework.web.servlet.ModelAndView;
+
+ import com.example.ifsol.models.Usuario;
+ import com.example.ifsol.repository.UsuarioRepository;
+
+ @Controller
+ public class UsuarioController {
+
+     @Autowired
+     private UsuarioRepository ur;
+
+     @RequestMapping(value = "/cadastrarUsuario", method = RequestMethod.GET)
+     public String formCadastrarNovoUsuario() {
+         return "usuario/formCadastrarNovoUsuario";
+     }
+
+     @RequestMapping(value = "/cadastrarUsuario", method = RequestMethod.POST)
+     public String formCadastrarNovoUsuario(Usuario usuario) {
+         String senhaEncriptada = new BCryptPasswordEncoder().encode(usuario.getSenha());

```

Fonte: autoria própria (2019)

Figura 15 - Implementação do ControllerVendas

```
@@ -0,0 +1,97 @@  
+ package com.example.ifsol.controllers;  
+  
+ import java.util.ArrayList;  
+  
+ import org.springframework.beans.factory.annotation.Autowired;  
+ import org.springframework.security.core.Authentication;  
+ import org.springframework.security.core.context.SecurityContextHolder;  
+ import org.springframework.stereotype.Controller;  
+ import org.springframework.web.bind.annotation.PathVariable;  
+ import org.springframework.web.bind.annotation.RequestMapping;  
+ import org.springframework.web.bind.annotation.RequestMethod;  
+ import org.springframework.web.bind.annotation.RequestParam;  
+ import org.springframework.web.servlet.ModelAndView;  
+  
+ import com.example.ifsol.models.ItemVenda;  
+ import com.example.ifsol.models.Produto;  
+ import com.example.ifsol.models.Usuario;  
+ import com.example.ifsol.models.Venda;  
+ import com.example.ifsol.repository.ItemVendaRepository;  
+ import com.example.ifsol.repository.ProdutoRepository;  
+ import com.example.ifsol.repository.UsuarioRepository;  
+ import com.example.ifsol.repository.VendaRepository;  
+  
+ @Controller  
+ public class VendaController {  
+  
+     ArrayList<ItemVenda> produtos = new ArrayList<ItemVenda>();
```

Fonte: autoria própria (2019)

Figura 16 - Implementação do ControllerVendas

```

public class VendaController {
    +
    +   ArrayList<ItemVenda> produtos = new ArrayList<ItemVenda>();
    +
    +   @Autowired
    +   private UsuarioRepository ur;
    +
    +   @Autowired
    +   private ItemVendaRepository ivr;
    +
    +   @Autowired
    +   private VendaRepository vr;
    +
    +   @Autowired
    +   private ProdutoRepository pr;
    +
    +   @RequestMapping(value = "/registrarVenda", method = RequestMethod.GET)
    +   public ModelAndView formRegistrarVenda() {
    +       ModelAndView mv = new ModelAndView("venda/formularioRegistrarVendas");
    +       Iterable<Produto> produtos = pr.findAll();
    +       mv.addObject("produtos", produtos);
    +       return mv;
    +   }
    +
    +   @RequestMapping(value = "/detalhesProdutoVenda/{codigo}")
    +   public ModelAndView detalhesProduto(@PathVariable("codigo") int codigo) {
    +       ModelAndView mv = new ModelAndView("venda/detalhesProdutoVenda");
    +       Produto prod = pr.findById(codigo);
    +       mv.addObject("produto", prod);
    +       return mv;
    +   }
    +
    +   @RequestMapping(value = "/addItemVenda", method = RequestMethod.POST)
    +   public String addItemVenda(@RequestParam("quantidade") String quantidade, @RequestParam("codigo") String codigo) {
    +       Produto produto = pr.findById(Integer.parseInt(codigo));
    +
    +       ItemVenda item = new ItemVenda();
    +       item.setProduto(produto);
    +       item.setQuantidade(Integer.parseInt(quantidade));
    +       produtos.add(item);
    +
    +       return "redirect:/registrarVenda";
    +   }
    +
    +   @RequestMapping(value = "/carrinhoVenda", method = RequestMethod.GET)
    +   public ModelAndView carrinhoVenda() {
    +       ModelAndView mv = new ModelAndView("venda/carrinhoVenda");
    +       mv.addObject("produtos", produtos);
    +       return mv;
    +   }
    +
    +   @RequestMapping(value = "/finalizarVenda", method = RequestMethod.POST)
    +   public String finalizarVenda() {
    +
    +       Authentication authentication = (Authentication) SecurityContextHolder.getContext().getAuthentication();
    +       System.out.println(authentication.getName());
    +
    +   }
}

```

Fonte: autoria própria (2019)

Figura 17 - Implementação do ControllerVendas

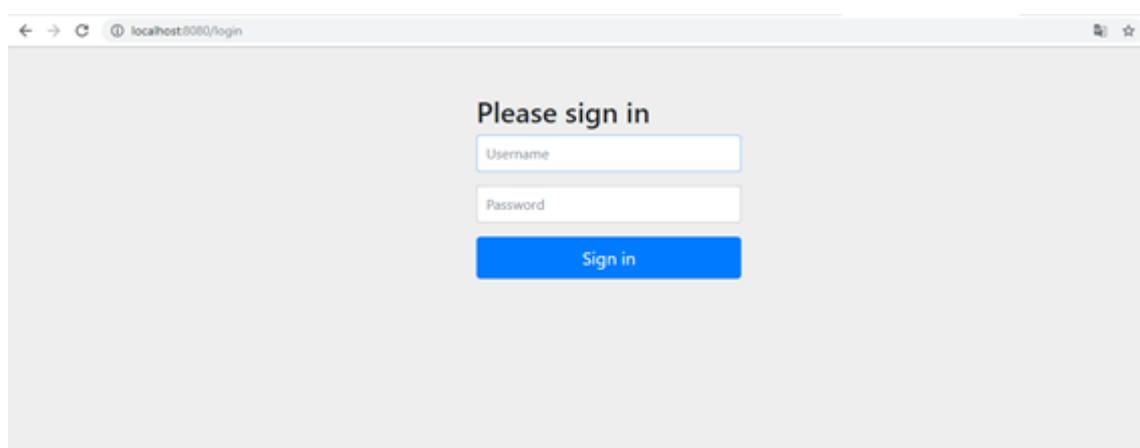
```
+
+ @RequestMapping(value="/carrinhoVenda", method=RequestMethod.GET)
+ public ModelAndView carrinhoVenda() {
+     ModelAndView mv = new ModelAndView("venda/carrinhoVenda");
+     mv.addObject("produtos", produtos);
+     return mv;
+ }
+
+ @RequestMapping(value="/finalizarVenda", method=RequestMethod.POST)
+ public String finalizarVenda() {
+
+     Authentication authentication = (Authentication) SecurityContextHolder.getContext().getAuthentication();
+     System.out.println(authentication.getName());
+
+     Venda venda = new Venda();
+     Usuario usuario = ur.findByEmail(authentication.getName());
+     venda.setUsuario(usuario);
+
+     for(int i=0; i<produtos.size(); i++) {
+         ItemVenda item = produtos.get(i);
+         venda.setItens(item);
+         ivr.save(produtos.get(i));
+         produtos.get(i).setVenda(venda);
+         vr.save(venda);
+     }
+     produtos.clear();
+     return "redirect:/registrarVenda";
+ }
```

Fonte: autoria própria (2019)

5.4 APRESENTAÇÃO DO PROTÓTIPO

A partir da página de login abaixo, é possível, acessar as funcionalidades do sistema de acordo com o perfil do usuário.

Figura 18 - Página de login



Fonte: autoria própria (2019)

Na figura abaixo é possível visualizar a tela de cadastro de um novo usuário do sistema.

Figura 19 - Tela de cadastro de usuários

A imagem mostra a interface de usuário para o cadastro de novos usuários no sistema IF-Sol. No topo, há uma barra de navegação vermelha com o logo 'IF-Sol' à esquerda e os links 'Cadastrar usuários', 'Usuários' e 'Produtos' à direita. Abaixo, o título principal 'IF-Sol - Cadastro de usuários' é exibido em uma fonte grande e preta. O formulário de cadastro contém quatro campos de entrada: 'Nome' e 'CPF' na primeira linha, e 'Email' e 'Password' na segunda linha. Abaixo dos campos, há três opções de perfil de usuário, cada uma com um botão de opção desativado: 'Administrador', 'Comprador' e 'Produtor'. No final do formulário, há um botão verde com o ícone de um documento e o texto 'CADASTRAR'.

Fonte: autoria própria (2019)

Nesta página o usuário com perfil de administrador pode cadastrar novos usuários. Para isso, é necessário preencher informações básicas como nome, cpf, email e uma senha para acesso ao sistema e em seguida escolher qual o perfil que o usuário terá no sistema.

O usuário com perfil de administrador, no módulo de usuário, tem acesso também aos usuários cadastrados.

Figura 20 - Tela contendo usuários cadastrados

IF-Sol			Cadastrar usuários	Usuários	Produtos
E-mail	Nome	CPF			
hiago@gmail.com	Hiago	12345678900			
clebson@gmail.com	Clébson	70279320400			
lore@gmail.com	Lorrena	12345678900			
admin@gmail.com	admin	12345678900			

Fonte: autoria própria (2019)

O perfil de administrador tem acesso também aos produtos cadastrados por um produtor. Dessa forma, o administrador pode alterar possíveis erros que pode ocorrer na hora do cadastro.

Figura 21 - Tela de produtos cadastrados

IF-Sol						Cadastrar usuários	Usuários	Produtos	Registrar Encomenda
#Código	Produto	Quantidade	Preço	Descrição					
3	Abacaxi	50	1.5	Unidade					
2	Manga	100	5.0	10 mangas por 5,00					
4	Melância	20	3.0	Venda pro Kg					

Fonte: autoria própria (2019)

No contexto do IFSol – Web, o usuário encarregado de cadastrar novos produtos no sistema é o que possui o perfil de **produtor**. Na figura abaixo é possível visualizar a tela referente ao cadastro de produto.

Figura 22 - Tela de cadastro de produtos

Nome do produto
Produto

Descrição

Quantidade
Informe a quantidade

Preço
Informe o preço

Produtor
Nome do produtor

CADASTRAR

Fonte: autoria própria (2019)

O produtor pode ainda verificar as encomendas à produtos, feitas pelos clientes.

Figura 23 - Tela de relatório de encomendas

Relatório de encomendas

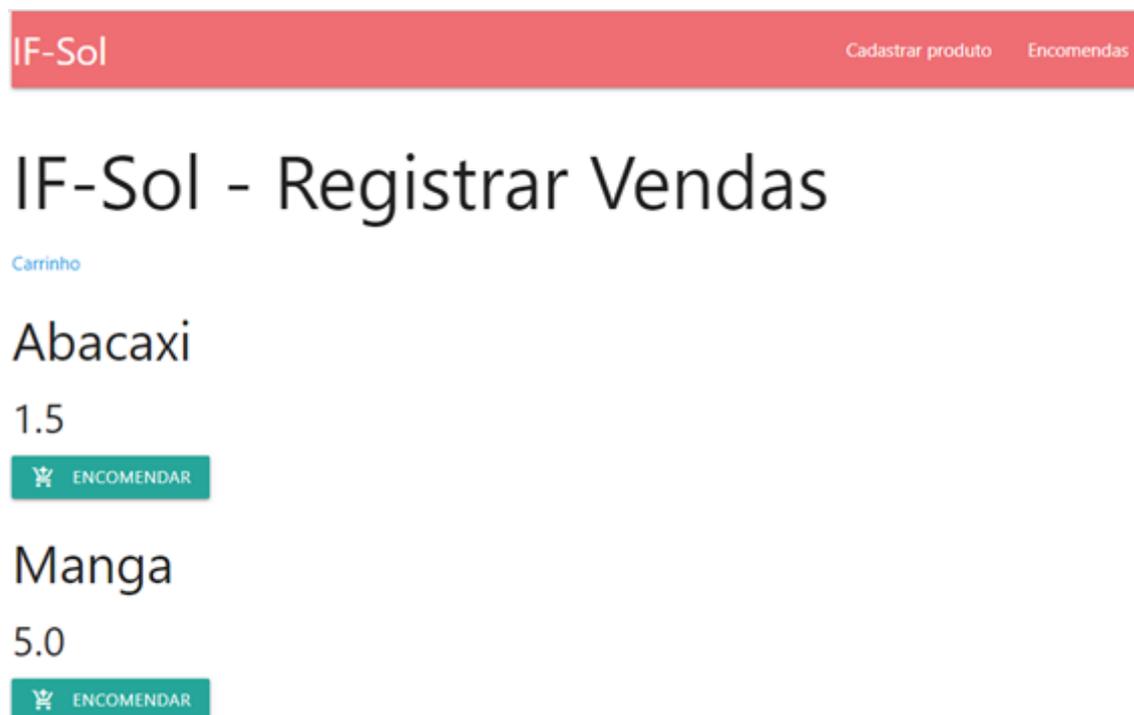
#Código da encomenda	Comprador	Produtos
6	Clébson	Produto Abacaxi Quantidade 2
		Manga 10
9	Clébson	Produto Melância Quantidade 3
		Manga 10

Fonte: autoria própria (2019)

Neste relatório, o produtor pode visualizar a quantidade de produtos encomendados e por quem foi encomendado. Isso ajuda no controle do quantitativo de produtos que deverá ser levado à feira.

Uma vez que os registros das encomendas foram feitos, o produtor deverá registrar suas vendas. Para isso, deverá acessar a tela de vendas.

Figura 24 - Tela para registrar vendas



Fonte: autoria própria (2019)

Nesta tela, o produtor adiciona os produtos da venda e em seguida ao acessar o carrinho, finalizar a venda.

O perfil de comprador possui autorização para consultar os produtos no sistema e realizar uma encomenda. Essa visualização dos produtos pode ser feita acessando a tela de listagem dos produtos ou na hora do registro da encomenda, como mostrado na figura a seguir.

Figura 25 - Tela de registro de encomendas

IF-Sol - Encomendar produtos

Carrinho

Abacaxi

1.5

ENCOMENDAR

Fonte: autoria própria (2019)

No registro da encomenda, o cliente clica no botão referente ao produto. Dessa forma, será aberta uma outra tela para que seja possível visualizar informações do produto e definir a quantidade que deseja encomendar.

Figura 26 - Tela para a escolha do produto a ser encomendado

Abacaxi

Descrição:

Unidade

Produtor:

Lorena

R\$1.5

Quantidade

Informe a quantidade

ADICIONAR

Fonte: autoria própria (2019)

Por fim, além do desenvolvimento do sistema web, foram realizadas atividades relacionadas ao projeto de extensão Incubadora Tecnológica para o Fortalecimento dos Empreendimentos Econômicos Solidários do IFRN (IFSol): Fortalecendo os Empreendimentos Econômicos Solidários (EES) do Rio Grande do Norte, entre elas: promover oficina de formação sobre economia solidária com participação dos servidores do campus: oficina programada para ser realizada durante a II SEMADEC do Campus Lajes, realizar visita a empreendimento econômico solidário, criar identidade visual da feira no campus: cartaz elaborado e adaptado conforme realização de feiras, promover oficina com produtores participantes e atender 1 EEs de forma efetiva (diagnóstico e/ou assessoria).

6 ANÁLISE E DISCUSSÃO DOS RESULTADOS

A partir da demanda exposta pelo Núcleo IFSol do Campus Lajes em relação à carência dos produtores rurais, que comercializam seus produtos na Feira da Agricultura de Lajes, por ferramentas que divulguem seu trabalho, produtos e necessidade de rendas extras, foi proposto a implementação de um sistema web que pudesse divulgar a Feira da Agricultura Familiar de Lajes/RN.

A partir disso, foi desenvolvido o IFSol – Web, que por sua vez, é um sistema que permite o gerenciamento do fluxo de encomendas e vendas entre produtores rurais e potenciais clientes compradores. Por meio desse sistema, é possível o produtor rural dimensionar o quantitativo de produtos que deverá levar à feira para que sejam vendidos, para que não ocorra desperdício de produtos.

Deste modo, após o desenvolvimento do sistema web pode – se analisar que as funcionalidades referentes ao gerenciamento do produto no sistema, se comportaram como previsto diante da possibilidade do usuário poder cadastrar, alterar e excluir.

Na codificação da funcionalidade de encomendas e vendas, há uma relação de um para muitos com os produtos. E se seguirmos o padrão de anotação sugerido e a cardinalidade advinda da modelagem desses dados do banco, perceberemos que o produto terá uma encomenda ou uma venda atrelada. Logo, para contorna esse problema, foi adotada a criação de uma nova classe que é responsável por integrar um produto, a quantidade desse produto e a encomenda ou a venda, dependendo de qual funcionalidade se trata.

Com relação a parte da interface com o usuário, o sistema apresenta boa usabilidade diante da exibição de menus com base no perfil do usuário. Com isso, o usuário só terá acesso as funcionalidades configuradas para o seu perfil, tendo uma boa visualização e compreensão na simplicidade da página web.

Além disso, o sistema se comportou bem diante da utilização do spring security, que implementou a funcionalidade de acesso e autorização. Com relação ao acesso os usuários conseguem realizar o login e utilizar o sistema de acordo com o seu usuário e senha previamente cadastrados. Logo, a autorização, no contexto do

IFSol - Web possibilitou o controle a funcionalidade do sistema de acordo com o perfil do usuário específico.

7 CONSIDERAÇÕES FINAIS

No geral, as metas e as atividades planejadas foram desenvolvidas, de acordo com a proposta, cumprindo assim o objetivo geral de criar uma página web totalmente funcional e acessível.

Essa ferramenta irá facilitar o fluxo de encomendas e vendas entre os agricultores e possíveis clientes, o controle do qualitativo de produtos e produtores que participam da feira. Além disso, o projeto apresenta alta relevância no processo de divulgação da Feira da Agricultura Familiar de Lajes/RN.

No processo de aprendizagem técnica e profissional houve um aprendizado maior, que levou a pesquisas sobre todas as dificuldades encontradas, com ajudas de vídeos aulas e orientação no desenvolvimento do sistema web. Além disso, pode contribuir para resolução de futuros problemas no desenvolvimento de outros projetos na vida profissional técnica.

Por fim, recomenda-se para futuros trabalhos dos bolsistas/voluntários do Núcleo IFSol do Campus Lajes o aperfeiçoamento no design, na interatividade e na criação app mobile para facilitar a utilização dos usuários.

REFERÊNCIAS

CIA WEBSITES. **MATERIALIZE: O QUE É E PARA QUE SERVE ESTE FRAMEWORK?**. DISPONÍVEL EM:

<HTTPS://WWW.CIAWEBSITES.COM.BR/SITES/O-QUE-E-MATERIALIZE/>.

ACESSO EM: 30 NOV. 2019.

DEVMEDIAS. **PRIMEIROS PASSOS COM O SPRING BOOT**. DISPONÍVEL EM:

<HTTPS://WWW.DEVMEDIA.COM.BR/PRIMEIROS-PASSOS-COM-O-SPRING-BOO>

T/33654. ACESSO EM: 30 NOV. 2019.

KINGLY. **MAS, O QUE É DESENVOLVIMENTO WEB MESMO?** . DISPONÍVEL EM:

<HTTP://WWW.KINGLYSTUDIO.COM.BR/DESCOMPLICA/MAS-O-QUE-E-DESENV>

OLVIMENTO-WEB-MESMO/. ACESSO EM: 30 NOV. 2019.

LUCENA, F. A. O. **SISTEMA DE INFORMATIZAÇÃO DO PROCESSO DE AQUISIÇÃO DOS DADOS DOS AGENTES DE ENDEMIAS: SIADE: MÓDULO APLICAÇÃO WEB**. INSTITUTO FEDERAL DO RIO GRANDE DO NORTE, PAU

DOS FERROS, V. 1, N. 1, P. 23-24, OUT./2015. DISPONÍVEL EM:

<HTTPS://MEMORIA.IFRN.EDU.BR/BITSTREAM/HANDLE/1044/801/LUCENA.PDF?>

<SEQUENCE=1&ISALLOWED=Y>. ACESSO EM: 30 NOV. 2019.

TECHTUDO. **O QUE É E COMO USAR O MYSQL?** . DISPONÍVEL EM:

<HTTPS://WWW.TECHTUDO.COM.BR/ARTIGOS/NOTICIA/2012/04/O-QUE-E-E-CO>

MO-USAR-O-MYSQL.HTML. ACESSO EM: 30 NOV. 2019.

VERGARA, S. C.; PECL, **AS ESCOLHAS METODOLÓGICAS EM ESTUDOS ORGANIZACIONAIS . ORGANIZAÇÕES E SOCIEDADE**, V. 10, N. 27, 2003.

MINISTÉRIO DO TRABALHO E EMPREGO. SECRETARIA NACIONAL DE ECONOMIA SOLIDÁRIA – **FÓRUM BRASILEIRO DE ECONOMIA SOLIDÁRIA –**

GRUPO DE TRABALHO DE FORMAÇÃO EM ECONOMIA SOLIDÁRIA.

RELATÓRIO FINAL DA II OFICINA NACIONAL DE FORMAÇÃO/EDUCAÇÃO EM ECONOMIA SOLIDÁRIA. BRASÍLIA: 2007. ACESSO EM: 30 NOV. 2019.

CUNHA, G. C. (2003). **DIMENSÕES DA LUTA POLÍTICA NAS PRÁTICAS DE ECONOMIA SOLIDÁRIA**. UMA OUTRA ECONOMIA É POSSÍVEL: PAUL SINGER

E A ECONOMIA SOLIDÁRIA. A.R. SOUZA, G. C. CUNHA AND R. Y. DAKUZAKU

(ORGS.). SÃO PAULO: CONTEXTO. 45-72. ACESSO EM: 30 NOV. 2019.
BLASCHEK. REPOSITÓRIO ROCA UTFPR. **DESENVOLVIMENTO DE UM SOFTWARE WEB PARA GERENCIAMENTO DE REQUISITOS DE SOFTWARE**, 2014. ACESSO EM: 30 NOV. 2019.

CARVALHO, KEILA LÚCIO. IPEA. **ECONOMIA SOLIDÁRIA COMO ESTRATÉGIA DE DESENVOLVIMENTO**, 2011. ACESSO EM: 30 NOV. 2019.

SILVA, SANDRO PEREIRA. REPOSITÓRIO IPEA. **O CAMPO DE PESQUISA DA ECONOMIA SOLIDÁRIA NO BRASIL: ABORDAGENS METODOLÓGICAS E DIMENSÕES ANALÍTICAS**, 2018. ACESSO EM: 30 NOV. 2019.

LINHARES, J. A. SOUSA. MONOGRAFIAS UFRN. **JSOIL: UMA FERRAMENTA WEB PARA ESTIMATIVA DE CAPACIDADE DE CARGA DE ESTACAS**, 2016.
ACESSO EM: 30 NOV. 2019.

LUCENA, F. A OLIVEIRA. MEMORIA IFRN. **SISTEMA DE INFORMATIZAÇÃO DO PROCESSO DE AQUISIÇÃO DOS DADOS DOS AGENTES DE ENDEMIAS-SIADE: MÓDULO APLICAÇÃO WEB**, 2015. ACESSO EM: 30 NOV. 2019.

ANEXO A – Formulário de identificação

Dados do Relatório Científico	
Título e subtítulo: IMPLEMENTAÇÃO DO SISTEMA WEB PARA DIVULGAÇÃO DA FEIRA DA AGRICULTURA FAMILIAR DE LAJES/RN	
Tipo de relatório: Projeto de Extensão	Data: 18/12/2019
<p>Título do projeto/plano:</p> <p>Título:</p> <p>Incubadora Tecnológica para o Fortalecimento dos Empreendimentos Econômicos Solidários do IFRN (IFSol): Fortalecendo os Empreendimentos Econômicos Solidários (EES) do Rio Grande do Norte.</p> <p>Plano:</p> <ul style="list-style-type: none"> ○ Promover oficina de formação sobre economia solidária com participação dos servidores do campus; ○ Realizar visita a empreendimento econômico solidário; ○ Criar identidade visual da feira no campus: cartaz elaborado e adaptado conforme realização de feiras; ○ Promover oficina com produtores participantes; ○ Atender 1 EEs de forma efetiva (diagnóstico e/ou assessoria); ○ Levantamento de requisitos do sistema; ○ Projeto de arquitetura do sistema; ○ Implementação do sistema; ○ Testes e validação do sistema. 	
Autor(es): Lorena Beatriz de Souza Silva	
Instituição e endereço completo: Instituto Federal do Rio Grande do Norte, BR-304, Km 120, s/n - Centro, Lajes - RN, 59535-000	
<p>Resumo: A Incubadora Tecnológica para o Fortalecimento dos Empreendimentos Econômicos Solidários do IFRN (IFSol) – Núcleo Lajes/RN tem desenvolvido suas ações em âmbito interno e externo. E em parceria com a Secretaria Municipal de Agricultura e Meio Ambiente (SEMAGMA) - Lajes/RN, um dos resultados positivos do IFSOL é a Feira da Agricultura Familiar, que desempenha papel fundamental na parceria entre os agricultores, consumidores e o poder público. Nesse contexto, este projeto tem, portanto, como objetivo a criação do sistema web, que por sua vez é uma solução a ser desenvolvidas para tornar processos manuais mais simples, rápidos e eficazes. Esse ajudará a expandir ainda mais os empreendimentos econômicos, fazendo a divulgação da Feira da Agricultura Familiar de Lajes/RN, controle de produtores e dos clientes. Logo, os agricultores divulgam os produtos, esses são encomendados e comprados. Assim os produtores trazem a quantidade certa para a cidade, o produtor não terá prejuízo ou não trará uma baixa quantidade. O sistema web será desenvolvido com o auxílio de quatro ferramentas: mysql, que ajudará no sistema de gerenciamento de banco de dados utilizado por diferentes aplicações; materialize, que ajudará no design do sistema; spring boot,</p>	

que ajudará a facilitar o processo de configuração e publicação de nossas aplicações, por último o eclipse uma IDE de programação. Deste modo, no geral o projeto ocorreu tudo como planejado e com suas funcionalidades devidas, de acordo com o que foi exposto na proposta, levantamento de requisitos e arquitetura do projeto. Assim, obteve – se como resultado uma página web totalmente funcional e acessível.	
Palavras-chave/descriptores: IFSol. Feira. Sistema Web.	
Nº de páginas: 46	
Jornada de trabalho: 3 dias por semana	Horas semanais: 15h
Total de horas: 360h	
Observações/notas:	