**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS FATEC PROFESSOR JESSEN VIDAL**

**JOÃO MANOEL FRANCO**

**Sistema de Controle do Projeto CRIA**

São José dos Campos

2019

**1 INTRODUÇÃO**

* 1. **MOTIVAÇÃO**

Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE, 2012) no Brasil, mais de 40 milhões de pessoas convivem com alguma forma de deficiência, pois 24% da população declarou no Censo 2010 ter algum tipo de deficiência. Historicamente, as pessoas com deficiência têm em sido sua maioria atendidas através de soluções segregacionistas, tais como instituições de abrigo e escolas especiais. Atualmente, as políticas mudaram em prol das comunidades e da inclusão educacional, e as soluções focadas na medicina deram lugar a abordagens mais interativas que reconhecem que as pessoas se tornam incapacitadas devido a fatores ambientais e também por causa de seus corpos.

Dentre os equipamentos de tecnologia assistiva, a cadeira de rodas é o mais representativo dispositivo de auxílio às pessoas com deficiência. Seu principal propósito é promover a mobilidade independente (quando possível) para pessoas com locomoção dificultada e, desta forma, tem sido utilizada por pessoas em uma grande variedade de condições, cada qual a sua maneira leva a deterioração da função motora. Entretanto apesar de seu uso disseminado, a cadeira de rodas tem sido indicada por seus usuários como fator limitador da participação comunitária (CHAVES et al., 2003).

Na prática, em países em desenvolvimento como no Brasil, em que há uma enorme desigualdade social, não é o que acontece devido a limitação financeira das pessoas de baixa renda. As cadeiras de rodas disponíveis no mercado possuem valores acima do poder de compra de grande parte da população, e ainda é mais agravante quando se trata de criança, pois, a falta deste equipamento impede que uma criança engaje em uma brincadeira e tenha contato uma com as outras.

No Brasil, as cadeiras de rodas motorizadas disponibilizadas pelo Sistema Único de Saúde (SUS) geralmente não contemplam crianças abaixo de 12 anos (LOPES 2018). Segundo o mesmo autor, o desenvolvimento de uma cadeira de rodas infantil automatizada (CRIA) a partir de peças de baixo custo, destinados a crianças abaixo de 6 anos, como proposto no projeto, é altamente desejável, pois além destas não serem comtempladas pelo SUS, este dispositivo pode promover o acesso de crianças à educação formal e contribuir para sua independência e futura colocação no mercado de trabalho.

É esperado que o desenvolvimento de um protótipo de baixo custo possa diminuir a falta de acesso à cadeira de rodas infantil, que muitas vezes impossibilita a criança com deficiência motora de frequentar uma escola e sua inclusão na sociedade.

Através de uma parceria entre Universidade Federal do ABC (UFABC) e a Universidade Federal de São Paulo (UNIFESP) foi desenvolvido um protótipo da CRIA de baixo custo, baseado em um modelo de uma ONG estadunidense Open Wheelchair Foundation. A CRIA é composta por uma estrutura mecânica de tubo de PVC, um sistema de propulsão contruido com motorredutores DC 12v 25W modelo CEP 9.390.453.086, joystick resistivo, dive ponte H modelo BTS7960 40A, baterias seladas 12v 7ah e um módulo Arduino Uno® (LOPES, AMORIN, KUNKEL, 2017).

A *Open Wheelchair Foundation* é uma organização sem fins lucrativo que foi fundada em 2015, que oferece cadeiras de rodas a preços acessíveis para crianças. Esta organização é um grupo colaborativo de pessoas que trabalham para criar uma CRIA de baixo custo, leve, fácil de montagem. O intuito da organização é dar às crianças acesso a mobilidade e independência, sem o alto custo das cadeiras de rodas comerciais, e de levar esse projeto ao redor mundo, não importando seu estado físico ou situação econômica (OWC 2019).

No projeto brasileiro da CRIA, foram realizadas algumas adaptações em relação ao projeto realizado pela OWF (Fig.1), isto se deve a diferença de preço e facilidade de aquisição de peças encontradas no Brasil.

|  |  |
| --- | --- |
| D:\Desktop\cadeira-pvc.jpg(A) | https://openwheelchair.org/Gallery/EvanD4.jpg(B) |

Figura 1- Diferenças entre os modelos da CRIA brasileira (à esq.) e o modelo de CRIA estadunidense

O projeto será desenvolvido no Laboratório de Informática UNIFESP, incluindo a produção das unidades em forma de kits a serem disponibilizados para doação. Os recursos materiais (hardware, software e suprimentos) e recursos humanos serão provenientes de parceria com empresas, estudantes e ONGs.

Estão sendo implementados os meios de captação de recursos para a produção das cadeiras de rodas, sendo necessária a implantação de um sistema com aplicação web para cadastramento de entidades interessadas em receber o produto para doar a crianças apoiadas, mais um controle dos pedidos e demais procedimentos até a entrega do produto.

A motivação para o desenvolvimento de um sistema computacional surgiu das dificuldades nos processos de gerenciamento logístico de produção, distribuição e suporte da cadeira de rodas infantil fabricada com tubos de PVC, que será distribuída para crianças em diferentes partes do Brasil. Para facilitar o gerenciamento logístico de pedido, produção e transporte do projeto CRIA, é de fundamental importância criar uma ferramenta em forma de sistema com aplicação web que seja acessível a todos os envolvidos no projeto, além da possibilidade de apoiar o modelo de negócio social que está sendo implementado para a geração de receitas objetivando a compra de materiais, montagem, distribuição e suporte da CRIA.

* 1. **OBJETIVO**

Através de reuniões com as pessoas envolvidas no projeto, foi apontada a necessidade do desenvolvimento de um sistema web para atender a necessidade de um cadastro das entidades ou pessoas físicas interessadas em receber o produto e o cadastro dos colaboradores que irão ter acesso aos pedidos. É necessário ter um controle dos pedidos realizados que permita aos colaboradores controlar e acompanhar a produção, assim como estimar os prazos para entrega de cada item, atualizando no sistema de acordo com o andamento dos processos concluídos para cada pedido. Na conclusão da produção do pedido e no envio do produto para entrega, esta informação precisa estar disponível a quem fez a solicitação, e esta pessoa poderá confirmar o recebimento do produto através do sistema para fechar o ciclo de cada pedido.

Existe uma página web da UNIFESP que faz alusão ao projeto CRIA e que contém informações de interesse público. A proposta inicial é adicionar um formulário a ser implementado a partir desta página para o cadastro de entidades e colaboradores, com a possibilidade de acesso ao sistema por meio de uma página de login, onde os usuários autorizados receberão uma senha por e-mail após a confirmação do cadastro, e assim poderão ter acesso ao sistema para realização de pedidos e demais informações sobre o projeto.

Há necessidade de um controle dos pedidos aceitos e demais procedimentos até a entrega do produto, e deve haver uma base de dados de fácil acesso aos colaboradores do projeto, oriundos de diferentes campus e setores.

É necessário um sistema para controlar os acessos, armazenar as informações, gerar relatórios personalizados, sendo acessível a todos os envolvidos, mas com as devidas restrições sobre a que parte do sistema cada tipo de usuário tem acesso e quais dados pode alterar ou atualizar.

São necessárias página web adicionais nas quais serão disponibilizadas informações importantes sobre a CRIA e a forma de envio do kit. Além das informações técnicas, nestas páginas estarão disponíveis vídeos tutoriais para montagem da CRIA, úteis para as pessoas que receberão o kit para montar.

Para o funcionamento do sistema será necessário implementar um banco de dados em SQL, objetivando agrupar os dados do projeto CRIA para auxílio da tomada de decisão dos gestores. Através do banco de dados criado para o sistema, poderão ser disponibilizados relatórios dos pedidos realizados e relatórios referentes à produção e entrega dos produtos, assim como relatório dos usuários cadastrados e relatório de acessos ao sistema.

**2 CONTEXTUALIZAÇÃO TECNOLÓGICA**

**2.1 Tecnologias Utilizadas**

Para o desenvolvimento do projeto foi utilizado um banco de dados relacional MariaDB, com linguagem semelhante ao SQL. Para o mapeamento entre as tabelas do banco de dados foi utilizado o framework Hibernate, conectando o banco a páginas dinâmicas tipo JSP que utilizam bibliotecas do Java EE, usando um servidor Apache Tomcat para o protótipo inicial rodando no localhost.

Para o front-end, foram implementadas páginas web escritas para o HTML-5 e utilizando estilo CSS. Para atender às necessidades de adaptação ao modelo de página web utilizado pela UNIFESP, também foi utilizado o framework Bootstrap, atendendo à necessidade de responsividade e padronização de estilo.

* + 1. **MariaDB**

MariaDB é um sistema de gerenciamento de banco de dados relacional que utiliza a linguagem de consulta estruturada SQL como interface de acesso e extração de informações do banco de dados em uso. Advém do gerenciador MySQL, apresentando-se como seu substituto, pois foi construído por alguns dos autores originais do MySQL, com a ajuda da grande comunidade de desenvolvedores de software livre e de código aberto. Além das funcionalidades básicas do MySQL, o MariaDB oferece um rico conjunto de aprimoramentos, incluindo mecanismos de armazenamento alternativo, otimizações de servidores e patches. [1]

* + 1. **Hibernate**

Segundo Bauer e King (2005, p.11) [2] os bancos de dados relacionais estão fortemente presentes no núcleo da empresa moderna, e por causa dos mesmos serem largamente utilizados em projetos de softwares orientados a objetos, faz-se necessário um mapeamento entre as tabelas do banco de dados e os objetos da aplicação. A fim de tornar compatível o paradigma da orientação a objetos ao paradigma de entidade e relacionamento, foram desenvolvidos frameworks de mapeamento objeto/relacional (Object Relational Mapping - ORM), conforme estrutura mostrada na Figura 1.

O Hibernate é um framework open-source de mapeamento objeto/relacional desenvolvido exclusivamente para linguagem Java, e suas principais vantagens são permitir que a aplicação permaneça totalmente orientada a objetos, e também fazer com que possíveis mudanças na base de dados, impliquem em um menor impacto sobre a aplicação, tendo em vista que apenas os objetos envolvidos com essa base de dados precisem ser modificados, ficando evidenciadas as vantagens da adoção do Hibernate na análise e no desenvolvimento de sistemas orientados a objetos [2].

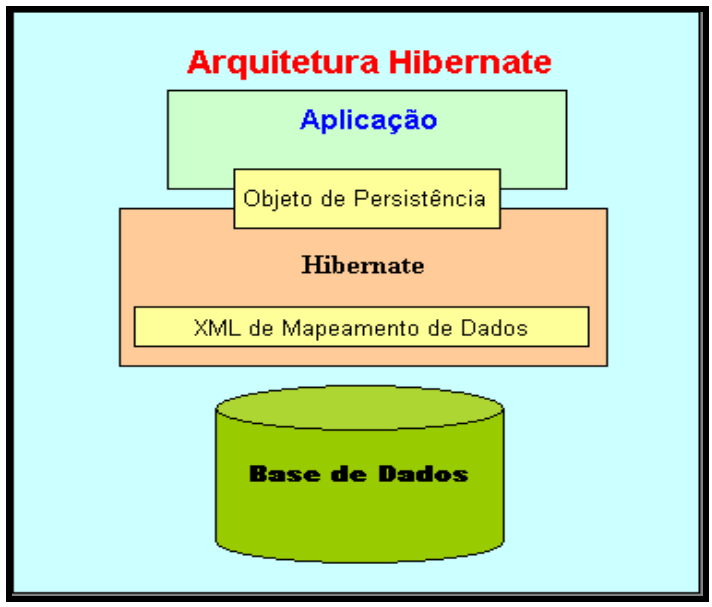


Figura 1: Arquitetura do Hibernate

* + 1. **JSP**

JavaServer Pages (JSP) é uma tecnologia que ajuda os desenvolvedores de software a criarem páginas web geradas dinamicamente baseadas em HTML, XML ou outros tipos de documentos. Lançada em 1999 pela Sun Microsystems, JSP é similar ao PHP, mas usa a linguagem de programação Java.

* + 1. **Java JEE**

Java Platform Enterprise Edition (JEE) é uma plataforma para desenvolvimento Java de aplicações voltadas para internet, redes ou intranet, pois fornece bibliotecas para  
desenvolvimento de software distribuído e multicamadas que executam em um servidor de aplicações.

* + 1. **Tomcat**

O Apache Tomcat é um container Web baseado em Java desenvolvido pela Apache  
Software Foundation. Ele é gratuito e possui recursos completos de um container Web  
para processar Java Servlets e JSP´s.

* + 1. **HTML**

HTML é o acrônimo de Hiper Text Markup Language, que significa linguagem de marcação de hipertexto. É uma linguagem utilizada para estruturar os elementos dentro de uma página de maneira que o navegador web possa interpretar o que é cada elemento através de etiquetas chamadas de tags. Tags são as marcações no corpo do conteúdo. É uma linguagem interpretada do lado do cliente.

* + 1. **CSS**

CSS é o acrônimo de Cascading Style Sheet que significa folhas de estilo em cascata. É uma linguagem interpretada do lado do cliente onde sua funcionalidade se baseia em acrescentar estilo visual aos elementos HTML, para torná-lo visualmente mais agradáveis aos usuários. Estilos como posição, cor, animação, etc. Essa linguagem formata o HTML através do uso de classe e id, onde esse id é acessado na folha de estilo através do símbolo “#” no início do nome do id.

* + 1. **Bootstrap**

Bootstrap é um framework front-end que veio para facilitar e agilizar o trabalho, oferecendo padrões para HTML, JavaScript e CSS. O Bootstrap permite que menos código possa ser escrito, já que ele entrega uma série de formatações visuais prontas, e faz reuso do código em outras páginas através da atribuição de classe; além disso, sua padronização permite que os sites tenham um melhor aspecto, sendo uma forma de criar páginas esteticamente agradáveis e com responsividade.

* 1. **Soluções Existentes**

O desenvolvimento web pode variar desde simples páginas estáticas a aplicações ricas, comércios eletrônicos ou redes sociais. Existem diversos tipos de plataformas para o desenvolvimento de aplicações web. A parte comum a todas é codificação no cliente (front-end), que usa HTML, CSS e Javascript, que é o resultado final vista pelo cliente.

O que varia bastante é codificação no servidor (back-end) e banco de dados. Existem soluções em diferentes linguagens, onde a escolha depende do ambiente em que o sistema será usado e o contexto do cliente. Em comum entre as diversas linguagens é a utilização do CRUD (acrônimo do inglês Create, Read, Update and Delete) são as quatro operações básicas (criação, consulta, atualização e destruição de dados) utilizadas em bases de dados relacionais (RDBMS) fornecidas aos utilizadores do sistema, conforme mostrado na Tabela 1.

**Tabela 1: Matriz CRUD**



* + 1. **CRUD com DJANGO**

Existem vários framewoks que facilitar a criação de CRUDs web. Entre eles encontramos o Django, que é um framework de alto nível, escrito em Python que encoraja o desenvolvimento limpo de aplicações web. No Django, uma View é uma forma de processar os dados de uma URL específica, pois ela descreve qual informação é apresentada, a View normalmente delega a informação para um template, que descreve como a informação é apresentada.

O Django tem uma diversidade enorme de Views, uma para cada finalidade, por exemplo:

*CreateView*: Para criar de objetos (É o Create do CRUD)

*DetailView*: Traz os detalhes de um objeto (É o Retrieve do CRUD)

*UpdateView*: Para atualização de um objeto (É o Update do CRUD)

*DeleteView*: Para deletar objetos (É o Delete do CRUD).

No caso do Django, é o próprio framework que faz o trabalho pesado de processar e rotear uma requisição para a View apropriada de acordo com a configuração de URL descrita pelo desenvolvedor.

* 1. **Levantamento de Requisitos**

Esta seção apresenta o levantamento de requisitos da solução proposta.

* + 1. **Definição dos Stakeholders**

A proposta inicial partiu de uma professora coordenadora do curso de Mestrado Profissional Interdisciplinar em Inovação Tecnológica da UNIFESP São José dos Campos (SP), tomando como base a necessidade que surgiu depois do projeto piloto que desenvolveu o primeiro protótipo do projeto CRIA, como tese de mestrado. Atualmente há um aluno deste curso de mestrado envolvido diretamente no projeto e foram consultados alguns alunos do curso de graduação de Bacharelato em Ciências da UNIFESP interessados em aderir ao projeto. Também colaboraram com algumas definições do sistema alguns professores da FATEC SJC que fazem parte do Centro de Apoio ao Desenvolvimento e Inovação (CADI).

* + 1. **Metodologia Utilizada**

Foram feitas entrevistas através de reuniões com os stakeholders, onde foram levantados os principais requisitos do sistema, além de sugestões para os modelos de implantação a serem utilizados na arquitetura do sistema.

* + 1. **Escopo**

O sistema usará um formulário a ser implementado numa página da UNIFESP, página que também servirá como meio de informação e contato com as entidades cadastradas. Haverá um controle dos pedidos aceitos e demais procedimentos até a entrega do produto, e deve ter uma base de dados de fácil acesso a todos os envolvidos no projeto, oriundos de diferentes campus e setores. O sistema deve controlar os acessos, armazenar as informações, gerar relatórios personalizados, e ser acessível a todos os envolvidos, mas com as devidas restrições sobre a que parte do sistema cada tipo de usuário tem acesso e quais dados pode alterar. A categoria de cada necessidade é apresentada na Tabela 2.

**Tabela 2: Escopo do Sistema**

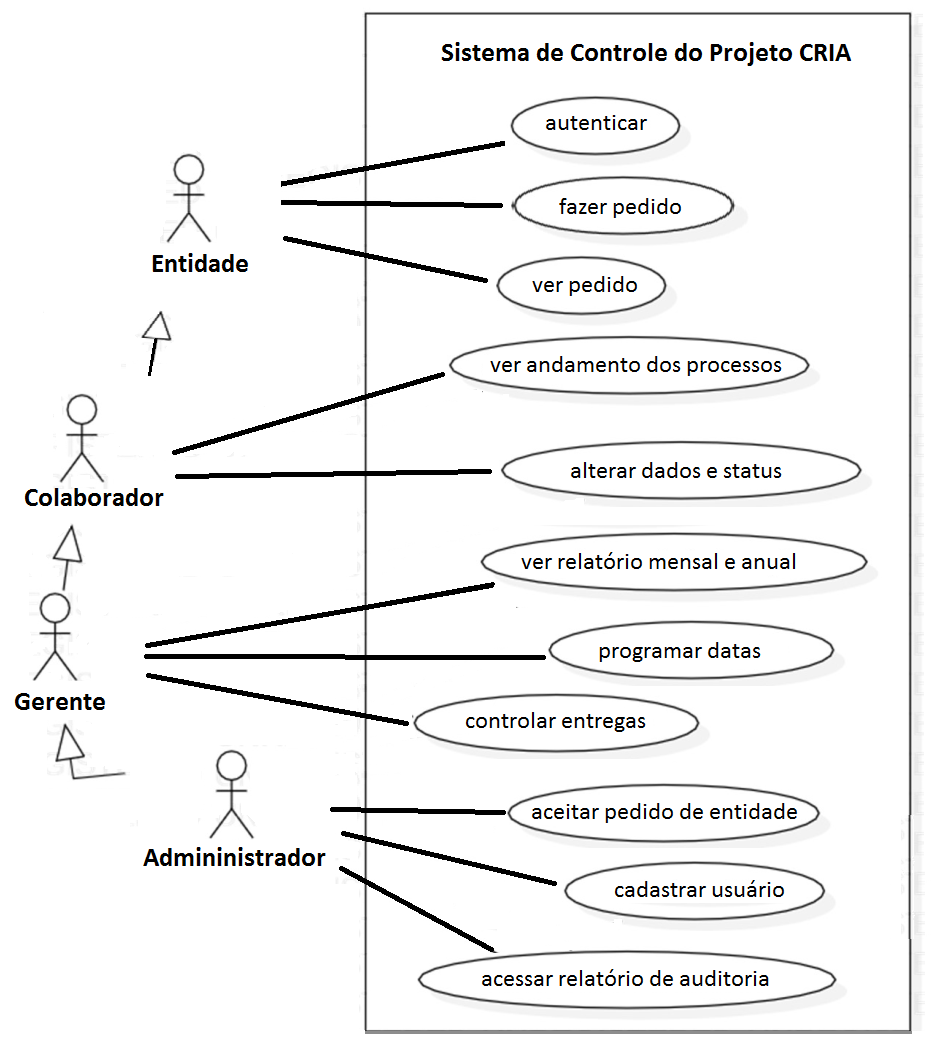
|  |  |
| --- | --- |
| Necessidades | Categoria |
| 1. Cadastro de entidades com informações de contato | Essencial |
| 1. Cadastro de usuários com diferentes níveis de acesso | Essencial |
| 1. Registro provisório de acesso | Essencial |
| 1. Páginas de serviços do sistema | Essencial |
| 1. Cadastro de pedido individual com medidas da criança | Essencial |
| 1. Controle de pedidos do produto | Essencial |
| 1. Controle de entregas do produto | Essencial |
| 1. Planejamento de produção do produto | Importante |
| 1. Banco de dados online | Essencial |
| 1. Relatório mensal, semestral e anual | Importante |
| 1. Relatório de usuários do projeto | Essencial |
| 1. Relatório de auditoria de acessos | Essencial |
| 1. Relatório por período especificado | Importante |
| 1. Backup do banco de dados | Importante |
| 1. Manual do usuário | Desejável |
| 1. Ajuda On-Line | Desejável |

* + 1. **Requisitos Funcionais**

Os Requisitos Funcionais foram definidos de acordo com o Diagrama de Casos de Uso ilustrado na Figura 1.

* + - 1. **Requisito 1 - Realizar cadastro:** Todos os atores orgânicos: alunos, mestrandos, professores, entidades e profissionais envolvidos no projeto CRIA deverão realizar um cadastro para ter acesso ao sistema.
      2. **Requisito 2 - Opção de acesso:** O sistema deve permitir o tipo de acesso de acordo com o envolvimento do usuário, a partir de diferentes páginas, que abrem um formulário específico e que será validado por um administrador para autorizar o acesso às telas permitidas a cada tipo de usuário.
      3. **Requisito 3 - Registro provisório:** O sistema deve permitir um registro provisório automaticamente, a partir do preenchimento de um formulário, que vai direcionar o usuário a uma página de informação temporária, até que o registro do usuário seja validado por um administrador. Após a validação o usuário terá acesso de acordo com o tipo de registro confirmado.
      4. **Requisito 4 - Controle de Usuários:** O sistema deve permitir ao usuário administrador visualizar e fazer alterações no Controle de Usuários, assim como validar os usuários com registro provisório.
      5. **Requisito 5 - Formulário de pedidos:** O sistema deve permitir a solicitação de pedidos através do formulário específico, que ficará pendente de aceitação por um usuário administrador
      6. **Requisito 6 - Aceitação e cadastro do pedido:** Para seguir adiante, um pedido deverá ser analisado por um usuário administrador, que verificará a possibilidade de atendimento, autorizando e cadastrando um número de pedido
      7. **Requisito 7 - Confirmação de pedido:** O sistema deve enviar aviso automático vinculado a cada entidade cadastrada, quando for autorizado o pedido, informando que a solicitação foi aceita, com o respectivo número de pedido.
      8. **Requisito 8 - Padronização e controle dos processos:** Para cada número de pedido será aberto um campo de padronização dos processos, colocando em ordem temporal para execução correta de cada processo envolvido e o status de cada um (concluído ou não concluído).
      9. **Requisito 9 - Cronograma de Pedidos:** Paralelo ao [RF8] será necessário ter um controle dos pedidos em andamento, com a possibilidade de alterar o status do processo, adicionando a informação de data prevista e data de conclusão de cada processo.
      10. **Requisito 10 - Programar entrega:** O sistema deve permitir a programação geral de data para a entrega de cada produto em produção, com registro da data que foi finalizado e disponibilizado para entrega.
      11. **Requisito 11 - Controle de entrega:** O sistema deve possuir um campo onde o usuário responsável pela entrega dos produtos marcará as datas previstas e receberá a confirmação das entregas dos pedidos.
      12. **Requisito 12 - Visualização de entrega:** O sistema passará a informação do [RF11] para um campo correspondente onde a entidade cadastrada para receber veja essa informação.
      13. **Requisito 13 - Confirmação de entrega:** O sistema permitirá que a entidade cadastrada confirme a data que recebeu o produto, fechando o ciclo de entrega.
      14. **Requisito 14 - Relatório de auditoria:** O sistema deve gerar um relatório que mostra linhas contendo um registro de login de usuários com horário e eventos realizados.
      15. **Requisito 15 - Relatório mensal:** O sistema deve gerar relatório filtrado mensalmente, para melhor assertividade de todos os processos estarem sendo cumpridos de acordo com o cronograma inicial.
      16. **Requisito 16 - Relatório anual:** O sistema deve gerar relatório anual, com comparação dos dados analisados das atividades principais ocorridas durante o ano selecionado
      17. **Requisito 17 - Relatório de usuários:** O sistema deve gerar um relatório de usuários, com as informações do cadastro, tipo de usuário e data de cadastro.
      18. **Requisito 18 - Backup de dados:** O sistema deve possibilitar um backup automático, com cópia regular do banco de dados para um servidor alternativo.

**Figura 2. Diagrama de Casos de Uso**



* + 1. **Requisitos Não Funcionais**
       1. **Requisito 1 - Regras de pedido:** O sistema deve gerar pedidos de acordo com as regras estabelecidas para as solicitações das entidades cadastradas.
       2. **Requisito 2 - Visualização de pedidos:** O sistema deve gerar uma visualização de pedidos através de gráficos coloridos e relatórios por tempo e quantidade.
       3. **Requisito 3 - Relatório por período:** O sistema deve permitir a opção de gerar relatório por um período de tempo selecionado por data, além dos relatórios mensais e anuais.
       4. **Requisito 4 - Servidor de backup de dados:** O servidor alternativo para o backup automático pode ser em nuvem, caso não exista um servidor físico extra disponível, e deve ter proteção de acesso seguro.
       5. **Requisito 5 -** **Design responsivo:** O sistema web deverá possui um design responsivo. A interface do sistema deverá se comportar adequadamente independente do dispositivo que será utilizado para acesso – browser, smartphone ou tablet.
       6. **Requisito 6 - Avisos do sistema:** O sistema deve informar ao usuário quando ele tentar fazer uma operação não permitida ou quando ele está prestes a realizar uma operação que pode alterar dados.
       7. **Requisito 7 - Salvamento automático:** O sistema deve possuir mecanismos que garantam que o usuário não perca informações. O sistema precisa oferecer recursos que possibilitem que o usuário recupere o conteúdo lançado no sistema caso ocorra, por exemplo, erro de execução do aplicativo, queda de energia, etc.
       8. **Requisito 8 - Disponibilidade do sistema:** O sistema deve permanecer online 24 horas por dia, exceto quando for programada alguma manutenção no sistema.
       9. **Requisito 9 – Compartilhamento de dados:** O sistema deve compartilhar dos mesmos protocolos de conexão, bem como os mesmos bancos de dados para armazenamento de informação nas diferentes plataformas de utilização, web ou aplicativo móvel (se houver). Também poderá ter caminho conector com outros softwares e protocolos externos ao sistema.
       10. **Requisito 10 - Segurança do sistema:** A segurança do sistema será dada através do protocolo SSL e SH512 para senhas, bem como algoritmos de autenticações criadas pela equipe, com criptografia segura.
       11. **Requisito 11 - Linguagens de programação:** Interface web deve ser baseada nas linguagens HTML, Java e Javascript, gerada a partir de scripts, para utilização da ferramenta em navegadores Web.
       12. **Requisito 12 - Hardware:** Na interface de hardware será necessário uma placa de rede /10/100 mpbs, com possibilidade de cem conexões simultâneas, bem como placa mãe 1.8GHz, 2 GB de RAM, espaço em HD de 80GB(incluindo banco de dados).
       13. **Requisito 13 - Sistema Operacional:** Os aplicativos do back-end do sistema devem rodar em Sistema Operacional Windows 7, 8 ou 10.
       14. **Requisito 14 - Interfaces de comunicação:** Será dado primeiramente através do teclado em formato ABNT2 e mouse, também a tela de impressão digital. No caso de acessibilidade serão utilizadas as saídas de áudios disponíveis.
       15. **Requisito 15 - Manual de Usuário:** O manual de usuário precisar ser uma referência completa do sistema com o qual o usuário possa aprender todas as funcionalidades, bem como os elementos de interface.Ele deve ser o mais completo possível, e com seções especiais com perguntas mais frequentes.
       16. **Requisito 16 - Ajuda on-line:** O sistema deve oferecer uma ajuda on-line contendo informações expressas de forma simples que efetivamente auxiliem o usuário a efetuar uma tarefa.
       17. **Requisito 17 - Ajuda por contexto:** O sistema deve oferecer um sistema de ajuda que ofereça informações de acordo com o contexto.
       18. **Requisito 18 - Tópicos de Ajuda:** O sistema deve oferecer um sistema de ajuda que tenha uma pequena quantidade de informações, e que só exiba informações detalhadas quando requisitado pelo usuário.

**3 DESENVOLVIMENTO**

* 1. **Protótipo do Sistema**
     1. **Cadastro de Entidade e Colaboradores**

As páginas estáticas do sistema web devem seguir o padrão definido pela UNIFESP. A hospedagem no servidor também segue as regras definidas pelo setor de informática da UNIFESP. O acesso ao sistema parte de uma página web já existente, por isso a continuação segue um modelo semelhante ao já utilizado, para que o usuário não perceba uma diferença muito grande ao fazer o acesso. Na página inicial e de login há um link para um formulário de cadastro. Ao se cadastrar no sistema o usuário é avisado sobre as regras de uso e garantia de privacidade, com link de acesso a estas informações, de acordo com os padrões da UNIFESP.

Apesar de haver vários projetos em andamento, em todos os casos do setor que solicitou o apoio para o desenvolvimento da aplicação web existem apenas páginas estáticas que trazem informações sobre os projetos existente. Esta é a primeira iniciativa de criar um cadastro de entidades interessadas em serem apoiadas por um dos projetos, assim como a criação de um cadastro de colaboradores envolvidos neste projeto em questão.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 2: Tela de Cadastro

* + 1. **Relatórios de pedidos**

De maneira semelhante ao exposto anteriormente, existem páginas estáticas que trazem informações sobre o andamento dos projetos existente, entretanto a atualização dessas informações não é automatizada, dependem de um funcionário autorizado que faz cada atualização de acordo com uma solicitação prévia do setor. Ao ser implantado o sistema, ele permitirá a atualização automática dos relatórios dos pedidos, tanto para as entidades como para os colaboradores, à medida que os próprios usuários forem atualizando os dados disponíveis.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 3: Relatório de Pedidos

* 1. **Telas do Sistema**
     1. **Tela de Login**

A tela de login segue o padrão para este tipo de tela, e após a confirmação da senha pelo Servlet, o usuário é redirecionado para uma tela Home, com links personalizados de acordo com o tipo de usuário.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 4: Tela de Login

**Uma imagem contendo captura de tela

Descrição gerada automaticamente**

Figura 5: Tela Home Administrador

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 6: Tela Home Gerente

* + 1. **Telas de Menu**

Existe uma tela de menu para cada tipo de usuário.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 7: Menu Entidade

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 8: Tela de Pedido

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 9: Menu Gerente

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Figura 10: Menu Administrador

**4 CONCLUSÃO**

Foi utilizado um backend em Java com formulários JSP e um Servlet para conexão com um banco de dados MariaDB para o protótipo do sistema web que foi apresentado, embora para a implementação final sejam buscadas algumas alternativas para melhorar a apresentação e funcionalidade do sistema. Foi apesentada uma alternativa viável que atende à necessidade de criar um cadastro de entidades e cooperadores do Projeto CRIA, além de facilitar a inserção e visualização dos pedidos, com atualização feita pelos cooperadores pelo acesso ao próprio sistema, atendendo ao propósito do sistema e fornecendo uma ferramenta importante para o desenvolvimento do Projeto CRIA.**REFERÊNCIAS BIBLIOGRÁFICAS**

[2] BAUER, Christian; KING, Gavin. Hibernate in action. Greenwitch: Manning Publications, 2005.

[1] Manzano, Jose A. N. G. MariaDB 5.5 - Interativo - Guia Essencial de Orientação e Desenvolvimento - Para Windows. Editora Érica, 2015.