

INSTITUTO FEDERAL

Rio Grande do Norte

Campus São Gonçalo do Amarante

DIRETORIA ACADÊMICA

COORDENAÇÃO DE EXTENSÃO

CURSO TÉCNICO INTEGRADO EM INFORMÁTICA

TÁRSILA SAMILLE SANTOS DA SILVEIRA

**RELATÓRIO DE ESTÁGIO CURRICULAR – SECRETARIA DE
INFRAESTRUTURA DE SÃO GONÇALO DO AMARANTE/RN**

São Gonçalo do Amarante /RN
2018

DIREÇÃO GERAL

Dra. Luisa de Marilac de Castro Silva

DIREÇÃO ACADÊMICA

Prof. Me. Francisco Aldrin Armstrong Rufino

COORDENAÇÃO DE EXTENSÃO

Profª. Me. Viviane Ferreira de Medeiros

COORDENAÇÃO DO CURSO TÉCNICO EM INFORMÁTICA

Prof. Me. Leonardo Gomes de Paiva Amorim

TÁRSILA SAMILLE SANTOS DA SILVEIRA

**RELATÓRIO DE ESTÁGIO CURRICULAR – SECRETARIA DE
INFRAESTRUTURA DE SÃO GONÇALO DO AMARANTE/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, em cumprimento às exigências legais como requisito parcial para a obtenção do título de Técnico em Informática.

Orientador: Prof. Marcelo Damasceno de Melo

TÁRSILA SAMILLE SANTOS DA SILVEIRA

**RELATÓRIO DE ESTÁGIO CURRICULAR – SECRETARIA DE
INFRAESTRUTURA DE SÃO GONÇALO DO AMARANTE/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, em cumprimento às exigências legais como requisito parcial para a obtenção do título de Técnico em Informática.

Aprovado em: ____/____/____

Nota Final: _____

Orientador

Prof. Marcelo Damasceno de Melo
Matrícula: 1725032

Prof. Me. Leonardo Gomes de Paiva Amorim, Coordenador do Curso Técnico Integrado em Informática
Matrícula: 1918220

RESUMO

O relatório a seguir tem como objetivo apresentar as atividades realizadas no estágio na Secretaria de Infraestrutura de São Gonçalo do Amarante, as atividades de suporte técnico em Informática realizadas tanto no Setor Técnico da Secretaria de Infraestrutura, como no Grupo de Estudos Avançados em Tecnologia da Informação e Comunicação (GEATIC), e o desenvolvimento do software de gerenciamento das obras da Secretaria, abordando os primeiros estudos de caso, as tecnologias utilizadas, os aprofundamentos nas tecnologias, as visitas e auxílio dos orientadores, a evolução do sistema e os desafios e conquistas ao longo do estágio.

Palavras-Chave: Informática, Desenvolvimento de Software, Tecnologia da Informação.

SUMÁRIO

INTRODUÇÃO	7
OBJETIVOS	7
IDENTIFICAÇÃO DO ESTÁGIO	8
DESCRIÇÃO DETALHADA DAS ATIVIDADES DESENVOLVIDAS	9
ATIVIDADE 1 [DESENVOLVIMENTO DO SOFTWARE DE	
GERENCIAMENTO DE OBRAS DE SÃO GONÇALO DO AMARANTE]	9
ATIVIDADE 2 [SUPORTE TÉCNICO EM INFORMÁTICA]	25
REFERÊNCIAS	27

1 INTRODUÇÃO

O estágio foi oferecido pela Prefeitura Municipal de São Gonçalo do Amarante/RN, na unidade da SEMINFRA – Secretaria Municipal de Infraestrutura. Esta é localizada na Rua Maria do Carmo Brito, próximo ao Fórum Desembargador Ivan Meira Lima. A secretaria possui grande demanda de gerenciamento de obras de toda a cidade de São Gonçalo, e todos os seus arquivos estão em planilhas, ou, em sua maioria, em papéis. Deste ponto surge a necessidade de um software que fizesse uso de um banco de dados para que estes dados fossem gravados de forma mais organizada, segura, e com maior controle por parte dos funcionários da secretaria. O estágio na secretaria teve como objetivo desenvolver tal aplicação.

1.1 OBJETIVOS

Objetivo Geral

Realizar atividades de desenvolvimento de software e suporte técnico a funcionários da prefeitura da Prefeitura Municipal de São Gonçalo do Amarante/RN, na unidade da Secretaria de Infraestrutura.

Objetivos Específicos

- Desenvolver software de gerenciamento de obras da Prefeitura Municipal de SGA;
- Oferecer suporte técnico em Informática: foram realizados dois encontros a fim de exibir a implementação do software e a devida orientação do uso do mesmo.

2 IDENTIFICAÇÃO DO ESTÁGIO

Do Estagiário:

Nome: Társila Samille Santos da Silveira

Endereço: Rua Otaviano Monteiro, 35, Bela Vista, CEP: , 59290-000 – São Gonçalo do Amarante/RN

Contato: (84) 9 98607 7507

E-mail: tarsillasamille@gmail.com

Curso: Técnico Integrado em Informática

Matrícula: 20151164010106

Período do Estágio: 20/11/2017 a 19/11/2018

Horas semanais: 20h

Da Empresa Concedente:

Razão Social: Prefeitura Municipal de São Gonçalo do Amarante/RN

Unidade: Secretaria de Infraestrutura

Endereço: Rua Maria do Carmo Brito, SN - Próximo ao Fórum, Samburá – CEP: 59290-000

Contato: (84) 9 9921-1886

Supervisor do Estágio: Alex Sandro Pinheiro Salviano

Email do supervisor: alexsalviano@oi.com.br

3 DESCRIÇÃO DETALHADA DAS ATIVIDADES DESENVOLVIDAS

3.1 ATIVIDADE 1 [DESENVOLVIMENTO DO SOFTWARE DE GERENCIAMENTO DE OBRAS DE SÃO GONÇALO DO AMARANTE]

3.1.1 Tecnologias usadas

Para a escolha das tecnologias utilizadas no desenvolvimento da aplicação, um dos principais critérios foi a afinidade dos desenvolvedores com tecnologias específicas. No IFRN *campus* São Gonçalo do Amarante, a linguagem que é utilizada na disciplina de Programação Estruturada e Orientada à Objetos é Java. A linguagem Java é vista com mais detalhes pelos alunos e foi escolhida pela sua popularidade e por fazer parte do dia a dia da maioria dos docentes da área de programação. Devido a estas características foi acordado que Java seria a linguagem utilizada no desenvolvimento. O banco de dados MySQL também foi escolhido devido a familiaridade e a sua eficiência em projetos de larga escala.

Um dos pontos fortes do Java é sua comunidade muito forte e atuante nos projetos criados para a plataforma. Há várias ferramentas que podem servir de exemplos, como o Eclipse e o Apache Tomcat (LUCKOW, MELO, 2015). Algumas tecnologias utilizadas foram escolhidas pela afinidade que tem com java, por exemplo o Apache TomCat. Outras tecnologias foram escolhidas por já estar bem estabelecida no mercado, pela eficiência, por facilitar o desenvolvimento ou a utilização da aplicação. Com exceção do MySQL e de Java e Bootstrap, não havia conhecimento prévio das tecnologias utilizadas, que foram Hibernate, RESTful Web Services, JavaServer Faces 2.2, PrimeFaces, Apache Tomcat®, Bitbucket e Gradle.

. As tecnologias foram aprendidas por meio de livros e cursos on-line, com a intenção ter o domínio básico sobre as tecnologias para possibilitar o desenvolvimento da aplicação. A descrição da tecnologia e os motivos para a escolha estão listados abaixo.

1 - Java EE

Java Platform, Enterprise Edition (ou Java EE, ou EE, ou em português Plataforma Java, Edição Empresarial) é uma plataforma de programação para servidores na linguagem de programação Java. A plataforma fornece uma API e um ambiente de tempo de execução para o desenvolvimento e execução de

softwares corporativos, incluindo serviços de rede e web, e outras aplicações de rede de larga escala, multicamadas, escaláveis, confiáveis e seguras.

Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90, diferente das linguagens de programação convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual, por isso pode ser utilizada em diferentes sistemas operacionais sem ter que ter o código reescrito.

2 - XHTML e CSS

O XHTML, ou eXtensible Hypertext Markup Language, é uma reformulação da linguagem de marcação HTML, baseada em XML. Combina as tags de marcação HTML com regras da XML. O HTML (o Hypertext Markup Language) e CSS (Cascading Style Sheets). O HTML fornece a estrutura da página, CSS o layout (visual) para uma variedade de dispositivos. O HTML e o CSS são a base da construção de páginas e aplicativos Web, porém foram utilizados juntamente com o bootstrap e o primefaces frameworks que facilitam o desenvolvimento de páginas web com uma interface rica.

3 - Java Persistence API

Java Persistence API (ou simplesmente JPA) nada mais é do que um conjunto de regras, normas e interfaces para definir um comportamento também conhecido como especificação. No mundo Java temos diversas outras como JSF, EJB, JAAS e muito mais. Toda especificação precisa de uma implementação para que possa ser usada num projeto. No caso da JPA, a implementação mais famosa e utilizada no mercado é o Hibernate (COELHO, 2013), o qual foi utilizado no projeto.

4 – Hibernate

O Hibernate é uma implementação da especificação Java Persistence API (BAUER, 2005). Como tal, ele pode ser facilmente usado em qualquer ambiente que suporte JPA, incluindo aplicativos Java SE, servidores de aplicativos Java EE, contêineres OSGi corporativos, etc.

5 - MySQL

MySQL, que é um SGBD (Sistema Gerenciador de Banco de Dados) relacional, que utiliza a linguagem SQL (Structured Query Language, ou traduzindo, Linguagem de Consulta Estruturada) (SUEHRING, 2002). MySQL também é multiusuário e multi tarefas.

6 - MySQL Workbench 5.1

MySQL Workbench é uma ferramenta de design de banco de dados visual que integra desenvolvimento SQL, administração, design, criação e manutenção de banco de dados em um único ambiente de desenvolvimento integrado para o sistema de banco de dados MySQL (MYSQL WORKBENCH, 2015).

8 - SQL

Structured Query Language(Damas, 1999), ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional), foi utilizada por ser o padrão.

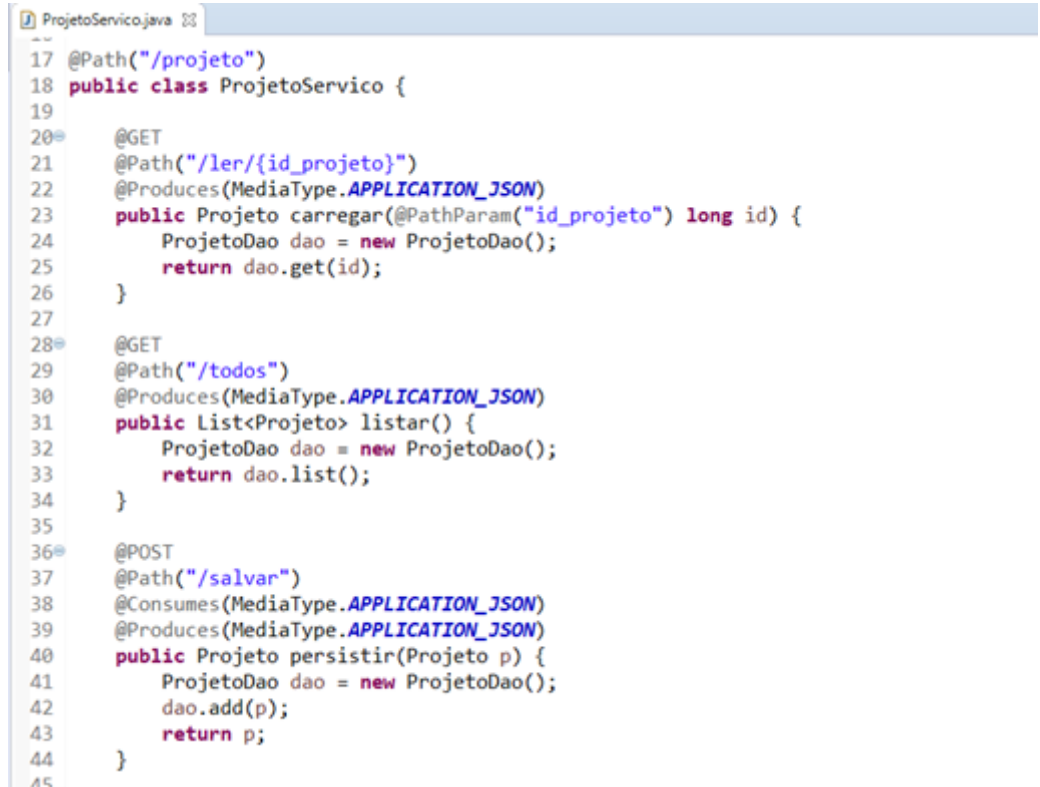
9 - RESTful Web Services

Representational State Transfer (REST) é um estilo de arquitetura que especifica restrições, como a interface uniforme, que se aplicadas a um serviço da Web, induzem propriedades desejáveis, como desempenho, escalabilidade e modificabilidade, que permitem que os serviços funcionem melhor na Web. No estilo arquitetural REST, os dados e a funcionalidade são considerados recursos e são acessados usando URIs (Uniform Resource Identifiers), geralmente links na Web. Os recursos são acionados usando um conjunto de operações simples e bem definidas. O estilo arquitetural REST restringe uma arquitetura a uma arquitetura cliente / servidor e é projetado para usar um protocolo de comunicação sem estado, geralmente HTTP. No estilo de arquitetura REST, clientes e servidores trocam representações de recursos usando uma interface e um protocolo padronizado.

Os princípios a seguir incentivam os aplicativos RESTful a serem simples, leves e rápidos:

- Identificação de recursos por meio de URI
- Interface uniforme

- Mensagens auto-descritivas
- Interações com estado por meio de hiperlinks



```

17 @Path("/projeto")
18 public class ProjetoServico {
19
20     @GET
21     @Path("/ler/{id_projeto}")
22     @Produces(MediaType.APPLICATION_JSON)
23     public Projeto carregar(@PathParam("id_projeto") long id) {
24         ProjetoDao dao = new ProjetoDao();
25         return dao.get(id);
26     }
27
28     @GET
29     @Path("/todos")
30     @Produces(MediaType.APPLICATION_JSON)
31     public List<Projeto> listar() {
32         ProjetoDao dao = new ProjetoDao();
33         return dao.list();
34     }
35
36     @POST
37     @Path("/salvar")
38     @Consumes(MediaType.APPLICATION_JSON)
39     @Produces(MediaType.APPLICATION_JSON)
40     public Projeto persistir(Projeto p) {
41         ProjetoDao dao = new ProjetoDao();
42         dao.add(p);
43         return p;
44     }
45

```

Figura 1. Exemplo prático de uma classe provedora de serviços, utilizando comandos HTTP. Ex: GET, POST.

10 - JavaServer Faces 2.2

JavaServer Faces, também conhecido como JSF, é uma tecnologia para desenvolvimento web que utiliza um modelo de interfaces gráficas baseado em eventos (FARIA, 2013).

11 - PrimeFaces

PrimeFaces é uma bibliotecas de componentes ricos em JavaServer Faces. A suíte de componentes inclui diversos campos de entrada, botões, tabelas de dados, árvores, gráficos, diálogos, etc.(FARIA, 2013). Foi uma tecnologia adotada pois conta com rico conjunto de componentes, contém Ajax embutido baseado em APIs JSF Ajax padrão, é leve, tem configuração zero e sem dependências necessárias, conta com uma extensa documentação e com uma comunidade de usuários ativos.

12 - Bootstrap 4.0

O Bootstrap é uma coleção grátis e open-source de ferramentas para criar Websites e aplicações (Mark Otto, 2011). É possível criar protótipos rapidamente ou aplicações completas, conta com sistemas de grid responsivo, componentes pré-construídos e plugins com jQuery. Foi adotado pela equipe pela facilidade de uso e pela estética, já que o componente do Bootstrap já vem com design prévio.

13 - Eclipse IDE

A própria plataforma do Eclipse é estruturada como subsistemas que são implementados em um ou mais plug-ins (ECLIPSE HELP, 2005), como mostrado na figura 2 simplificada.

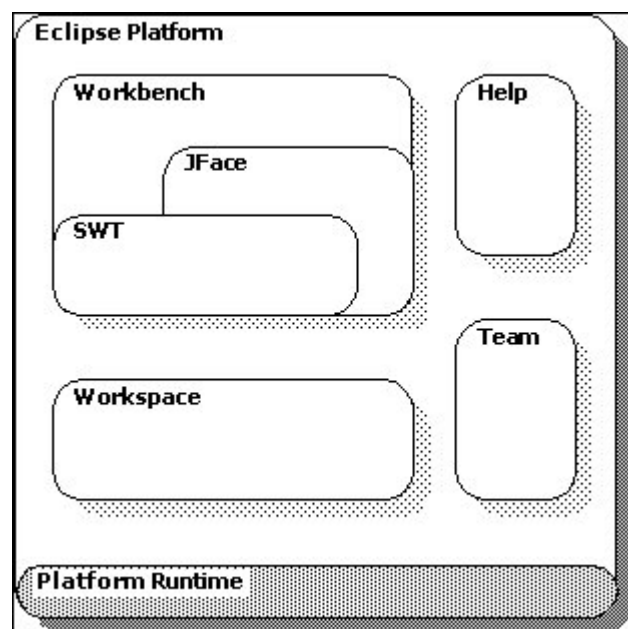


Figura 2. Modelo Eclipse

O Eclipse IDE for Java EE Developers é uma versão do eclipse que contém ferramentas para desenvolvedores Java que criam aplicativos Java EE e Web, incluindo um IDE Java, ferramentas para Java EE, JPA, JSF, Mylyn, EGit e outros.

14 - Apache Tomcat ®

O Tomcat é um servidor que implementa as tecnologias web – Java Servlet e JavaServer Pages – do Java EE, o que possibilita que o Java funcione para um ambiente Web (LUCKOW, MELO, 2015)..

15 - Bitbucket

Para melhorar o fluxo do trabalho em equipe, utilizamos o Bitbucket, uma solução Git para versionamento de códigos (ATLASSIAN, 2017). Para, que várias pessoas trabalhem no mesmo projeto sem haver conflito de código, facilitando o trabalho na hora junção do código.

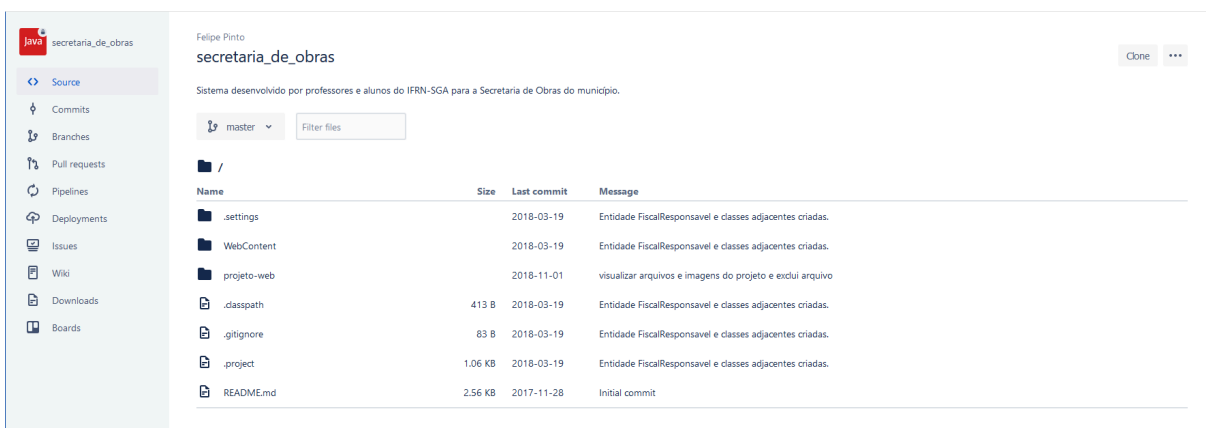


Figura 3 - Repositório do projeto no Bitbucket. Foto: Autoria própria.



Figura 4 - Ramificações, trabalhos paralelos, e junções. Foto: Autoria própria

16 - Gradle

Gradle é um sistema de automação de compilação de código aberto (B. MUCHO, 2014), além de gerenciar dependências da aplicação, é uma ferramenta de construção que substitui scripts de construção baseados em XML por uma DSL interna baseada na linguagem de programação Groovy.

3.1.2 Funcionalidades

O software foi idealizado para auxiliar na gestão das obras da prefeitura de São Gonçalo do Amarante e na divulgação das mesmas. O módulo de divulgação das obras será de acesso público que disponibiliza o acesso à população aos dados

da obra que são disponibilizados pela secretaria, e uma parte interna da aplicação que permitirá o gerenciamento das obras e a persistência de dados da obra informados pelos funcionários.

Visualmente, cada obra será retratada como um marcador no mapa, indicando a localização. Ao selecionar um desses pontos, ou fazer uma busca específica por um deles, o sistema permitirá que você acesse os dados da obra e os seus arquivos armazenados no sistema.

1. Especificações do Sistema

Atores

1. Administrador: Administrador do sistema.
2. Pessoa: As pessoas que acessam o sistema externamente.

Casos de Uso

1. Login
2. Cadastro de Usuário
3. Cadastro de Obra
4. Excluir obra
5. Adicionar arquivos à obra
6. Alteração de dados
7. Atribuição da Função de Administrador
8. Pesquisa
9. Ver dados da obra

2. Exemplo de casos de uso: Login e Cadastro de Obra

I. Nome: Login

1. Breve Descrição

Este caso de uso descreve como um administrador efetua login no Sistema de Gerenciamento de Obras e permite que as operações críticas do sistema sejam efetuadas por administrador cadastrados

Autor: Administrador.

2. Fluxo de Eventos

O caso de uso começa quando o agente digita seu nome e senha no formulário de login.

2.1 Fluxo Básico – Login

1. O sistema encaminha o administrador para a página de login.
2. O administrador preenche seus dados na página de login.
3. O administrador clica no botão "Entrar" para enviar seus dados.
4. O sistema verifica que a senha é correta para o nome de Administrador especificado.
5. O administrador é direcionado para a página principal do Sistema.

2.2 Fluxos Alternativos

1. Nome / Senha Inválidos

Se no fluxo básico o sistema não puder localizar o nome ou a senha for inválida, uma mensagem de erro será exibida. O administrador pode digitar um novo nome ou senha, ou ainda optar por cancelar a operação, ponto no qual o caso de uso é finalizado.

II. Nome: Cadastro de Obra

1. Breve Descrição

Este caso de uso descreve como um administrador efetua o cadastro no Sistema de Gerenciamento de Obras.

Autor: Administrador e Pessoa.

2. Fluxo de Eventos

O caso de uso começa quando o administrador é direcionado para a página principal do Sistema de Gerenciamento de Obras e clica em Cadastro.

2.1 Fluxo Básico – Cadastro de Obra

1. O administrador é direcionado para a página principal do Sistema de Gerenciamento de Obras.

2. A página principal do Sistema de Gerenciamento de Obras contém seguintes as opções
 - Lista
 - Cadastro
 - Mapa
 - Gerenciar Usuários
3. O administrador clica em Cadastro.
4. O administrador preenche o formulário com os dados da obra. O formulário de cadastro contém os seguintes campos a serem preenchidos, na primeira fase são as informações gerais onde todos os campos são obrigatórios:
 - Descrição do Projeto(campo editável)
 - Endereço da Obra (campo editável)
 - Número do Contrato (campo editável)
 - Previsão de conclusão (campo editável)
 - Fiscais Responsáveis (campos editáveis)
 - Principal
 - Suplente
 - Situação (campo editável)
 - Empresa (campo editável)
 - Tipo de Obra (campo editável)
 - Valor Licitado (campo editável)
5. O administrador clica no botão ">>>" para ir para fase do Local no Mapa.
6. O administrador clica no botão "Adicionar Localidade".
7. Seleciona a localização no mapa.
8. O administrador clica no botão ">>>" para ir para fase de Confirmação.
9. O administrador Confirma os dados e decide se entre ir para novo cadastro ou ir para o mapa selecionando uma das alternativas da caixa de diálogo aberta.

2.2 Fluxos Alternativos

1. Os dados especificados são inválidos

O usuário pode redigitar os dados(caso tenha errado na digitação), ou ainda optar por cancelar a operação, ponto no qual o caso de uso é finalizado.

3. Interfaces de usuário

1. **Tela de Login:** Interface on-line para a entrada de administradores no sistema.
2. **Tela de Usuários:** Interface on-line para inclusão, consulta, alteração e exclusão de usuários.
3. **Tela Principal:** Interface on-line com atalho ir para outras telas e um mapa da obra tal como será mostrado ao público.
4. **Tela de Cadastro de Obra:** Interface on-line para preenchimento, confirmação e persistência de dados sobre a obra.
5. **Tela de Modificação de dados:** Interface on-line com a lista das obras realizadas pela secretaria e uma interface gráfica que permite a pesquisa, alteração/atualização de dados e uploads arquivos.

4. Imagens do Sistema

As figuras 5 a 11 que estão abaixo mostram imagens do sistema.

Figura 5 - Primeira parte do cadastro de uma obra. Foto: Autoria própria

Id	Descrição	Endereço	Tipo de Projeto	Previsão De Conclusão	Situação	Excluir	Alterar	Arquivos
1	Ponte de Utinga	Bela Vista	Ponte	20/09/2018	Em andamento	Excluir	Alterar	Arquivos
2	Creche Mariazinha	Centro	Escola	26/09/2018	Em andamento	Excluir	Alterar	Arquivos

Figura 6 - Listagem de Projetos, conta com opções de: Excluir, Alterar e Adicionar arquivos.

Foto: Autoria própria

Figura 7 - Alterar obra. Foto: Autoria própria

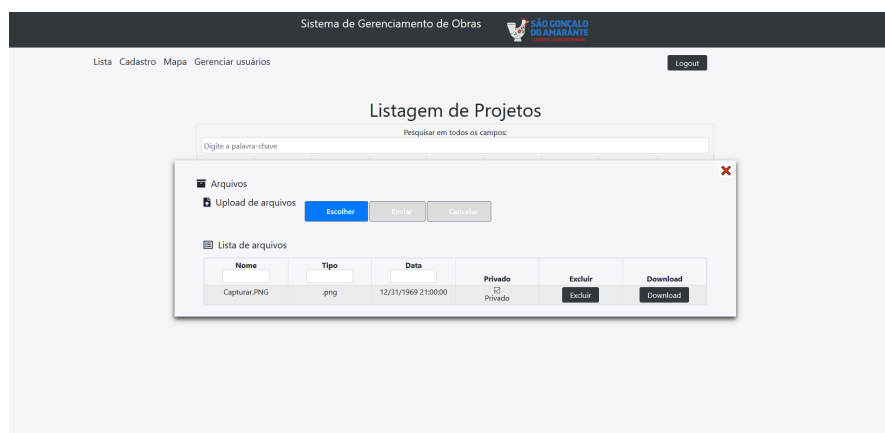


Figura 8 - Arquivos da obra, com opções de upload, exclusão e download. Foto: Autoria própria

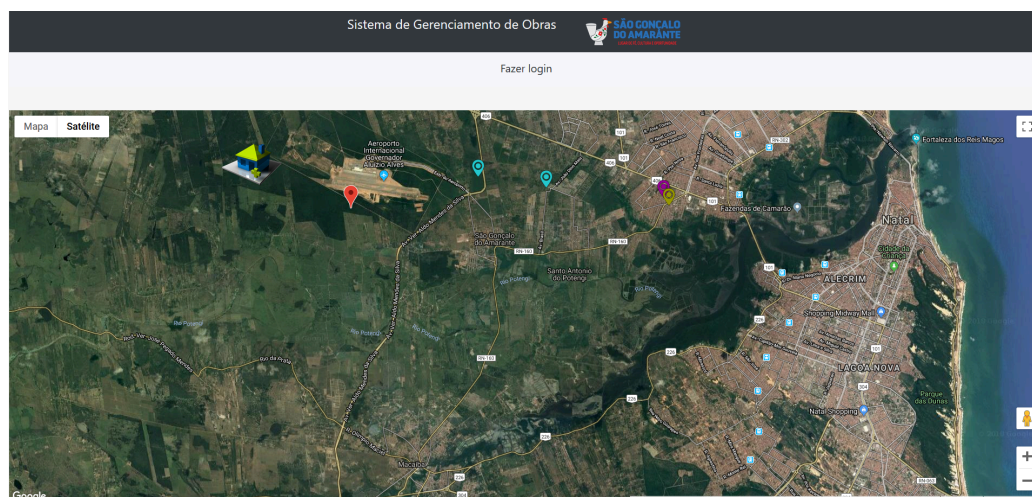


Figura 9 - Visualização dos marcadores das obras no mapa. Foto: Autoria própria

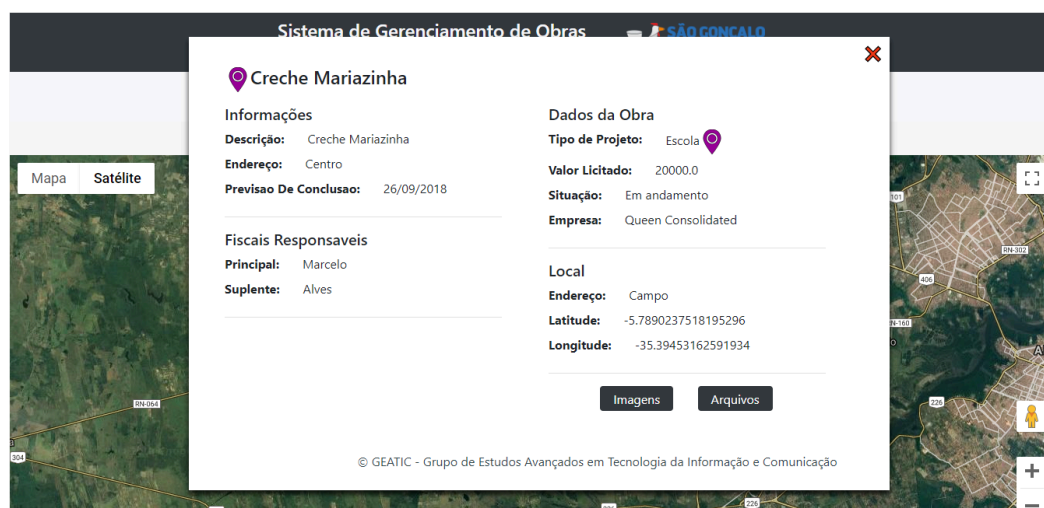


Figura 10 - Visualização de informações da obra clicando no marcador. Foto: Autoria própria



Figura 11 - Tela de Login. Foto: Autoria própria.

3.1.3 Ambiente de Desenvolvimento

Com a finalidade de que as tecnologias apresentadas funcionem bem em conjunto, foi criado um ambiente de desenvolvimento nos computadores presentes no laboratório do GEATIC (Grupo de Estudos Avançados em Tecnologia da Informação e Comunicação) no IFRN que se dedica à Estudos Avançados em Tecnologia da Informação.

As especificações das máquinas utilizadas ao longo do desenvolvimento da aplicação estão listadas abaixo, respectivamente o primeiro e o segundo semestre.

1.

- Sistema Operacional: Windows 10 Education de 64 bits
- Memória instalada (RAM): 8GB
- Processador: AMD A9 PRO-7800B R7, 12 Compute Cores 4C + 8G 3.50 GHz

2.

- Sistema Operacional: Windows 10 Pro de 64bits
- Memória instalada (RAM): 16GB
- Processador: Intel(R) Core(TM) i7-7700T CPU @ 2.90GHz 2.90 GHz

O ambiente de desenvolvimento contou com as tecnologias listadas

anteriormente:

- IDE Eclipse
- MySQL Server e Workbench
- Java EE
- BitBucket
- Gradle

3.1.4 Desenvolvimento

O primeiro estágio do desenvolvimento foi entender a necessidade da secretaria e listar as funcionalidades da aplicação, a partir disso começamos a modelagem do sistema com a ajuda dos orientadores e segundo o padrão de arquitetura de software MVC (Model - View - Controller).

O padrão MVC determina que um sistema seja separado em três camadas, chamadas de Model, View e Controller, mas o principal objetivo do MVC não é definir como separar em camadas, mas, sim, definir com as camadas devem interagir. Em outras palavras, a separação de responsabilidades é um conceito benéfico na implementação do MVC (LUCKOW, MELO, 2015, p.185).

Primeiro criamos os POJO's, o primeiro objeto a ser criado foi projeto como mostrado na figura 12, começamos com apenas um simples objeto com alguns atributos mas conforme o número de informações demandadas foi aumentando percebemos que alguns atributos deveriam ter a sua própria classe, e se tornar objetos separados, mantendo relacionamento com a classe principal como demonstrado na figura 13, a classe Projeto.

```
Projeto.java
30 @Id
31 @GeneratedValue(strategy = GenerationType.IDENTITY)
32 private long id;
33
34 private String descricao;
35 private String endereco;
36 private String imagem;
37
38 private String numContrato;
39 private Date previsaoDeConclusao;
40 private Double recursoDaObra;
41
42 @Transient
43 List<Localidade> localidades;
44
45 @Transient
46 Localidade local; //para mostrar no mapa
47
48 @OneToOne(cascade = CascadeType.ALL)
49 @JoinColumn(name = "valor_id")
50 private Valor valorDaObra;
51
52 private long valorDaObraId;
53
54 @ManyToOne
55 @JoinColumn(name = "tipoDeProjeto_id")
56 private TipoDeProjeto tipoDeProjeto;
57 private long tipoId;
58
```

Figura 12 - Atributos da Classe Projeto. Foto: Autoria própria

```
Projeto.java
47
48 @OneToOne(cascade = CascadeType.ALL)
49 @JoinColumn(name = "valor_id")
50 private Valor valorDaObra;
51
52 private long valorDaObraId;
53
54 @ManyToOne
55 @JoinColumn(name = "tipoDeProjeto_id")
56 private TipoDeProjeto tipoDeProjeto;
57 private long tipoid;
58
59 @ManyToOne
60 private FiscalResponsavel fiscalResponsavel1;
61 private long fiscalResponsavelId1;
62
63 @ManyToOne
64 private FiscalResponsavel fiscalResponsavel2;
65 private long fiscalResponsavelId2;
66
67 @ManyToOne
68 private Empresa empresa;
69 private long empresaId;
70
71 @ManyToOne
72 private Situacao situacao;
73 private long situacaoId;
74
```

Figura 13 - Alguns atributos que representam classes e mantém relacionamentos com a classe Projeto, sendo usadas anotações do JPA/Hibernate. Ex: @ManyToOne, @OneToOne. Foto: Autoria própria

Para persistir os dados que seriam cadastrados criamos os DAO's - Objeto de acesso a dados (acrônimo de Data Access Object), é um padrão para persistência de dados que permite separar regras de negócio das regras de acesso a banco de dados, também foi criado uma classe DAO genérica de persistência como na Figura 14, com métodos que serviriam para todas as classes.

```
GenericDao.java
25 public void add(T entity) {
26     EntityManager em = JPAUtil.getEntityManager();
27
28     em.getTransaction().begin();
29     em.persist(entity);
30     em.getTransaction().commit();
31
32     em.close();
33
34 }
35
36 public void alterar(T entity) {
37     EntityManager em = JPAUtil.getEntityManager();
38
39     em.getTransaction().begin();
40     em.merge(entity);
41     em.getTransaction().commit();
42
43     em.close();
44
45 }
46
47 public void remover(T entity) {
48     EntityManager em = JPAUtil.getEntityManager();
49
50     em.getTransaction().begin();
51     entity = em.merge(entity);
52     em.remove(entity);
53     em.getTransaction().commit();

```

Figura 14 - Classe genérica com exemplo de 3 métodos: adicionar, alterar e remover. Foto: Autoria própria

O que liga as classes de persistência e as classes dos objetos ao que for digitado pelo usuário durante o cadastro dos dados é o nosso Managed Bean, como na figura 15.

"Os beans gerenciados do JSF podem receber dados digitados pelo usuário através de alguma página, processar métodos através de ações dos usuários e fornecer dados para apresentação da página" (FARIAS, 2013, p.48).

Foi preciso criar uma solução para que apenas administradores da secretaria pudessem manipular os dados de cadastro e listagem das obras, e o público externo pudesse visualizar apenas o mapa com os marcadores. Para isso foi criada uma classe objeto Usuário como na figura 16, que receberá alguns atributos, dentre eles sua senha, que será modificada na hora do cadastro para md5 hash, um tipo de criptografia. Então, nem o banco de dados terá acesso a senha original, apenas à senha criptografada, os métodos estão mostrados na figura 17. Na hora do login, transcrevemos a senha digitada para o md5 hash e comparamos com a senha criptografada no banco, se as duas forem iguais, o login é efetuado com sucesso e o usuário poderá ter acesso a todo o software, se não, ele é redirecionado de volta para a tela de login.

```
ProjetoMBean.java
248
249 public String buscar() {
250     Client c = ClientBuilder.newClient();
251
252     Builder b = c.target(URL_BASE).path("/ler/" + projeto.getId()).request();
253     projeto = b.get(Projeto.class);
254
255     c.close();
256
257     return "";
258 }
259
260 public String buscarAlterar() {
261     Client c = ClientBuilder.newClient();
262
263     Builder b = c.target(URL_BASE).path("/ler/" + projeto.getId()).request();
264     projeto = b.get(Projeto.class);
265
266     c.close();
267
268     return "";
269 }
270
271 public void salvar() {
272
273     Client c = ClientBuilder.newClient();
274     Builder b = c.target(URL_BASE).path("/salvar").request();
275
276     projeto = b.post(Entity.entity(projeto, MediaType.APPLICATION_JSON), Projeto.class);
```

Figura 15 - Classe Managed Bean de Projeto. Foto: Autoria própria

```

8 @Entity
9 public class Usuario {
10     @Id
11     @Column(nullable = false)
12     private String email;
13
14     @Transient
15     private String senhaSemCriptografia;
16
17     @Column(nullable = false)
18     private String senha;
19
20     private boolean ativo;
21     private String tipo;
22
23     public String getTipo() {
24         return tipo;
25     }
26
27     public void setTipo(String tipo) {
28         this.tipo = tipo;
29     }
30
31     public boolean isAtivo() {
32         return ativo;
33     }
34
35     public void setAtivo(boolean ativo) {
36         this.ativo = ativo;

```

Figura 16 - Classe Usuário. Foto: Autoria própria.

```

57 public void autenticar() {
58     try {
59         UsuarioDAO dao = new UsuarioDAO();
60         usuarioLogado = dao.autenticar(usuario.getEmail(), usuario.getSenha());
61
62         if (usuarioLogado == null) {
63             FacesContext.getCurrentInstance().addMessage(null,
64                 new FacesMessage(FacesMessage.SEVERITY_ERROR, "Usuário não encontrado!", "Erro no Login!"));
65             Faces.redirect("./pages/login.xhtml");
66         }
67         else if (usuarioLogado.getTipo().equals("ADMIN")) {
68             ehAdm = true;
69             Faces.redirect("./pages/principal.xhtml");
70         }
71     } catch (IOException e) {
72         Messages.addGlobalError(e.getMessage());
73     }
74 }
75
76
77 public void logout() {
78     try {
79         FacesContext context = FacesContext.getCurrentInstance();
80         context.getExternalContext().getSessionMap().remove("#{autenticacaoBean}");
81         HttpSession session = (HttpSession) FacesContext.getCurrentInstance().getExternalContext().getSession(true);
82         session.invalidate();
83
84         Faces.redirect("./pages/principal.xhtml");
85     } catch (IOException e) {

```

Figura 17 - Métodos de autenticação e logout. Foto: Autoria própria.

3.2 ATIVIDADE 2 [SUPORTE TÉCNICO EM INFORMÁTICA]

No suporte técnico em informática, as atividades realizadas foram de formatação e instalação de Windows 10 e 7 em algumas máquinas que precisavam ser utilizadas para implementação da aplicação, e ajudar os funcionários da secretária quanto a dúvidas a cerca de tecnologia da informação em geral, o que proporcionou oportunidade de pôr em prática os conhecimentos adquiridos nos quatro anos de curso técnico, respondendo estas dúvidas da forma mais clara e simples possível. Porém, como o foco do estágio era o desenvolvimento e não a

manutenção, não foi uma atividade tão realizada como a programação da aplicação de gerenciamento de obras.

Também foram realizados dois encontros a fim de exibir a implementação do software e a devida orientação do uso do mesmo, mostrando as funcionalidades e a forma de acesso ao Login, Cadastro de Usuário, Cadastro e Exclusão de obra, Adicionar arquivos à obra, Alteração de dados, Atribuição da Função de Administrador, e como dados cadastrados no Mapa ou em forma de Lista. Na reunião também foram tiradas dúvidas e dado as instruções para funcionalidades que ainda estavam sendo desenvolvidas.

4 CONSIDERAÇÕES FINAIS

Devido a importância dada atualmente ao acesso do cidadão às informações de interesse geral, o software solicitado pela secretaria veio com a proposta principal de tornar informações públicas de fácil acesso, democratizando-as por meio de um mapa acessível que constasse com dados da obra, como suas localizações, seus valores e fotos atuais. No contexto de nossa sociedade em que se torna cada vez mais imprescindível a fiscalização do cidadão para com os serviços e as despesas públicas, me sinto feliz e satisfeita de ter sido parte de um projeto que visa aumentar a transparência do governo além de facilitar e auxiliar o cidadão a ter acesso a informações.

Minha experiência de estágio muito enriquecedora e importante profissionalmente e academicamente, trabalhei em uma área com a qual me identifiquei e com tecnologias que em sua maioria eram novas para mim, pude aprender muita coisa que será importante caso continue no campo de desenvolvimento de software, participei do desenvolvimento do sistema, desde sua fase inicial, contando sempre que necessário com o auxílio dos orientadores, que deram todo o suporte possível, me alegrei a cada fase concluída e cada layout finalizado, pude aplicar bastante do que já havia aprendido em sala de aula e também conhecer novas possibilidades na área. Acredito que durante esse período eu pude obter um amadurecimento não só profissional mas também pessoal, o que

será extremamente importante.

Infelizmente, por questões de infraestrutura, a maior parte do estágio foi realizada no IFRN, e não na secretaria, o que nos deu mais liberdade quanto ao trabalho, porém, não era como estar num ambiente real de trabalho, com pessoas e relacionamentos sociais, o que pode ter tirado parte da experiência profissional.

Por fim, levarei comigo os ensinamentos que o IFRN me passou através destes quatro anos, que foram transmitidos por professores e funcionários, que me proporcionaram além de uma formação técnica, uma formação humana.

REFERÊNCIAS

ATLASSIAN. **Software Development And Collaboration Tools. Bitbucket. 2017.** Disponível em:<<https://bitbucket.org/>>. Acesso em: 7 dez. 2018.

BOOTSTRAP. **Bootstrap em Português · O mais popular framework front-end responsivo e focado para dispositivos móveis do mundo.** Disponível em: <<https://getbootstrap.com.br/>>. Acesso em: 7 dez. 2018.

ÇIVICI, Çağatay. **Primefaces User's Guide 3.3.**Turquia, 2012. 475 p.

CORDEIRO, Gilliard (2012). **Aplicações Java para a web com JSF e JPA 1 ed.** São Paulo: Casa do Código. 270 páginas.

ECLIPSE HELP. **Help - Eclipse Platform.** Disponível em: <<http://help.eclipse.org/2018-09/index.jsp>>. Acesso em: 7 dez. 2018.

ECLIPSE. **Eclipse IDE for Java Developers | Eclipse Packages** Disponível em: <<https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers>>. Acesso em: 7 dez. 2018.

HIBERNATE. **Your relational data. Objectively. - Hibernate ORM** Disponível em: <<http://hibernate.org/orm/>>. Acesso em: 7 dez. 2018.

MYSQL WORKBENCH. **Mysql Workbench. Productos.** Disponível em: <<https://www.mysql.com/products/workbench/>> Acesso em: 7 dez. 2018.

MYSQL. **MySQL** :: Download MySQL Community Server Disponível em: <<https://dev.mysql.com/downloads/mysql/>>. Acesso em: 7 dez. 2018.

ORACLE. **Differences between Java EE and Java SE - Your First Cup: An Introduction to the Java EE Platform.** Disponível em: <<https://docs.oracle.com/javaee/6/firstcup/doc/gkhoy.html>>. Acesso em: 7 dez. 2018.

ORACLE. **Java™ EE at a Glance.** Disponível em: <<https://www.oracle.com/technetwork/java/javaee/overview/index.html>>. Acesso em: 7 dez. 2018.

ORACLE. **What Are RESTful Web Services?** - The Java EE 6 Tutorial Disponível em: <<https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>>. Acesso em: 7 dez. 2018.

OTTO, Mark. **Bootstrap from Twitter.** Disponível em: <<https://blog.twitter.com/2011/bootstrap-twitter>> Acesso em: 7 dez. 2018.

W3. **Cascading Style Sheets** Disponível em: <<https://www.w3.org/Style/CSS/>>. Acesso em: 7 dez. 2018.

W3. **W3C HTML** Disponível em: <<https://www.w3.org/html/>>. Acesso em: 7 dez. 2018.

B. Muscho. **Gradle in Action**, 1 st ed Greenwich, CT, US: Manning Publications Co., 2014.

BAUER, C,; KING, G. **Hibernate in Action.** ed. MANNING 2005

COELHO, Hébert. **JPA eficaz: as melhores práticas de persistência de dados em Java.** Cada do Código, 2013. 167 p

Damas, L. 1999. **SQL - Structured Query Language.** FCA.

FARIAS, Thiago. **Java EE 7 com JSF, Primefaces e CDI.** 1 ed. São Paulo: Algaworks Softwares, 2013.

IFRN. **Projeto pedagógico do curso Técnico de Nivel Médio em Informática na forma integrada, na modalidade presencial.** Natal-RN, 2009.

Leonard, R. and Ruby, S. (2007). **RESTful web services.**

LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair De. **Programação Java**

para a Web: Aprenda a desenvolver uma aplicação financeira pessoal com as ferramentas mais modernas da plataforma Java. 2 ed. São Paulo: Novatec, 2015.

SUEHRING, Steve. **MySQL, a Bíblia.** 1. ed. Rio de Janeiro: Campus, 2002.