



**GABRIEL DE MATOS OLIVEIRA FERNANDES**

**RELATÓRIO DE ESTÁGIO:  
DESENVOLVIMENTO DE SOFTWARE NA  
EMPRESA GT4W - GEOTECHNOLOGY FOR WEB**

**LAVRAS – MG**

**2020**



**GABRIEL DE MATOS OLIVEIRA FERNANDES**

**RELATÓRIO DE ESTÁGIO:**

**DESENVOLVIMENTO DE SOFTWARE NA EMPRESA GT4W -  
GEOTECHNOLOGY FOR WEB**

Relatório de estágio supervisionado apresentado à  
Universidade Federal de Lavras, como parte das  
exigências do Curso de Ciência da Computação,  
para a obtenção do título de Bacharel

Prof. DSc. André Vital Saúde

Orientador

**LAVRAS – MG**

**2020**

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos  
da Biblioteca Universitária da UFLA**

Fernandes, Gabriel de Matos Oliveira

Relatório de Estágio : Desenvolvimento de Software na empresa  
GT4W - Geotechnology For Web / Gabriel de Matos Oliveira  
Fernandes. – Lavras : UFLA, 2020.

40 p. :

Relatorio de Estágio(graduação)–Universidade Federal de Lavras, 2020.

Orientador: Prof. DSc. André Vital Saúde.

Bibliografia.

1. TCC. 2. Monografia. 3. Dissertação. 4. Tese. 5. Trabalho Científico – Normas. I. Universidade Federal de Lavras. II. Título.

CDD-808.066

**GABRIEL DE MATOS OLIVEIRA FERNANDES**

**RELATÓRIO DE ESTÁGIO: DESENVOLVIMENTO DE SOFTWARE NA  
EMPRESA GT4W - GEOTECHNOLOGY FOR WEB**

Relatório de estágio supervisionado apresentado à  
Universidade Federal de Lavras, como parte das  
exigências do Curso de Ciência da Computação,  
para a obtenção do título de Bacharel

APROVADA em 08 de Setembro de 2020.

Prof. DSc. André Vital Saúde  
Orientador

**LAVRAS – MG  
2020**



*Dedico aos meus familiares, pelo companheirismo, pela cumplicidade e pelo apoio em todos os momentos delicados da minha vida.*



## **AGRADECIMENTOS**

Agradeço ao apoio dos meus amigos e familiares, sobretudo da minha mãe e do meu irmão. Sem eles, eu jamais chegaria onde cheguei.



## RESUMO

O presente trabalho tem como objetivo descrever as atividades realizadas durante o período de estágio na empresa GT4W - Geotechnology For Web. Atualmente com sede em Lavras - MG e operando em diversas regiões do Brasil, a GT4W tem como principal atividade o desenvolvimento de soluções em geoprocessamento para seus clientes. Durante o período de estágio, foi desenvolvido um módulo para um sistema de gerenciamento e controle de concessões florestais usando tecnologias JavaScript, HTML e CSS com AngularJS para front-end, Java e PostgreSQL com Play para back-end. Utilizou-se o padrão MVC (Model, View, Controller) e as atividades foram desenvolvidas seguindo a metodologia ágil Scrum.

**Palavras-chave:** Estágio, Geoprocessamento, MVC, AngularJS, Java, JavaScript.



## **ABSTRACT**

This work aims to describe the activities carried out during the internship period at the company GT4W - Geotechnology For Web. Currently based in Lavras - MG, operating in several regions of Brazil, the main activity of GT4W is the development of solutions in geoprocessing for its customers. During the internship period, a module was developed for a forest concession management and control system using JavaScript, HTML and CSS technologies with AngularJS as frontend, Java and PostgreSQL with Play as backend. The MVC (Model, View, Controller) standard was used, and the activities were developed using the agile Scrum methodology.

**Keywords:** Internship, Geoprocessing, MVC, AngularJS, Java, JavaScript.



## LISTA DE FIGURAS

Figura 2.1 – Padrão MVC do Play Framework. . . . .	25
Figura 4.1 – Exemplo de relação em banco de dados PostgreSQL. . . . .	32
Figura 4.2 – Exemplo de modelagem de classe Java usando JPA. . . . .	33
Figura 4.3 – Exemplo de serviço Java. . . . .	33
Figura 4.4 – Exemplo de página Web desenvolvida em AngularJS. . . . .	34
Figura 4.5 – Exemplo de gráfico de Burndown. . . . .	35



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	17
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	19
<b>2.1</b>	<b>Arquitetura MVC</b>	19
<b>2.1.1</b>	<b>Model</b>	19
<b>2.1.2</b>	<b>View</b>	20
<b>2.1.3</b>	<b>Controller</b>	20
<b>2.2</b>	<b>Metodologia Scrum</b>	21
<b>2.3</b>	<b>PostgreSQL</b>	22
<b>2.4</b>	<b>Java</b>	23
<b>2.4.1</b>	<b>JPA - Java Persistence API</b>	24
<b>2.4.2</b>	<b>Play Framework</b>	24
<b>2.5</b>	<b>AngularJS</b>	25
<b>3</b>	<b>METODOLOGIA</b>	27
<b>3.1</b>	<b>Procedimentos Metodológicos</b>	27
<b>3.2</b>	<b>Desenvolvimento do software</b>	27
<b>3.3</b>	<b>Processos</b>	28
<b>4</b>	<b>RESULTADOS</b>	31
<b>4.1</b>	<b>Mapeamento de Entidades</b>	31
<b>4.2</b>	<b>Criação de Serviços</b>	31
<b>4.3</b>	<b>Criação de Páginas Web</b>	31
<b>4.4</b>	<b>Experiências</b>	32
<b>4.5</b>	<b>Desempenho</b>	34
<b>5</b>	<b>CONCLUSÃO</b>	37
	<b>REFERÊNCIAS</b>	39



## 1 INTRODUÇÃO

A GT4W Consultoria e Serviços Em Geoprocessamento LTDA, fundada em 2011 e atualmente sediada na cidade Lavras, Minas Gerais, é uma empresa especializada em Geotecnologia e desenvolvimento de soluções para os setores de gestão pública, ambiental, de ativos e territorial, com foco no desenvolvimento sustentável. Tem experiência em diversos projetos de relevância nacional contando com um time de aproximadamente 100 colaboradores que encontra-se em constante expansão através de novos projetos.

No período de estágio, foi desenvolvido um módulo do sistema de gestão de concessões de florestas públicas para exploração no estado do Pará (SGC). Para o desenvolvimento das atividades do projeto, foram feitos treinamentos durante um mês, que possibilitou aprender e se familiarizar com as ferramentas e linguagens de programação utilizados.

O sistema foi desenvolvido utilizando os frameworks Play e AngularJS, as linguagens Java, JavaScript, HTML e CSS, e também o banco de dados PostgreSQL. Foi usado o MVC (Model, View, Controller) como padrão para organização do código e metodologia Scrum para gerenciamento ágil. Este sistema oferece suporte para os navegadores Google Chrome e Mozilla Firefox em suas edições mais recentes, em sua versão para computadores com sistema operacional Linux e Windows.



## 2 REFERENCIAL TEÓRICO

Neste capítulo, serão discutidos conceitos importantes para o desenvolvimento do projeto, o primeiro conceito a ser abordado é a arquitetura MVC, que tem grande relevância por ser aplicado em todo o projeto. Em seguida será abordada a metodologia de gerenciamento ágil Scrum que foi utilizada no projeto e as tecnologias empregadas, iniciando pelo PostgreSQL, seguido do Java e, por último, o AngularJS.

### 2.1 Arquitetura MVC

Conforme (DEACON, 2009), é razoável propor que qualquer aplicativo possa alterar sua interface com o passar do tempo ou mesmo ter várias interfaces a qualquer momento. No entanto, o aplicativo subjacente pode muito bem ser bastante constante.

No paradigma MVC, a entrada do usuário, a modelagem do mundo externo e o feedback visual para o usuário são explicitamente separados e manipulados por três tipos de objetos, cada um especializado em sua tarefa (BURBECK, 1992).

Essa estratégia foi escolhida para atender a dois objetivos: (1) criar o conjunto especial de componentes do sistema necessário para suportar um processo de desenvolvimento de software altamente interativo e (2) fornecer um conjunto geral de componentes do sistema que possibilite a programadores criarem aplicações gráficas interativas portáteis facilmente, (KRASNER; POPE et al., 1988).

#### 2.1.1 Model

O Model gerencia o comportamento e os dados do domínio do aplicativo, responde a solicitações de informações sobre seu estado, geralmente na visualização (BURBECK, 1992).

De acordo com (DEACON, 2009), em termos orientados a objetos, o model consistirá no conjunto de classes que modelam e apoiam o problema e que, portanto, tenderão a ser estáveis e durar tanto quanto o próprio problema.

Segundo (KRASNER; POPE et al., 1988), o modelo de um aplicativo é a simulação de software específica do domínio ou a implementação da estrutura central do aplicativo. Pode ser tão simples quanto um número inteiro (como o modelo de um contador) ou uma string (como o modelo de um editor de texto), ou pode ser um objeto complexo que é uma instância de uma subclasse de alguma classe composta.

### **2.1.2 View**

Quanto a view, (KRASNER; POPE et al., 1988) explica que, as views lidam com tudo gráfico; elas solicitam dados de seu modelo e exibem os dados. Elas contêm não apenas os componentes necessários para exibição, mas também podem conter subviews e estar contidos nas supervisões. A superview fornece a capacidade de executar transformações gráficas, janelas e cortes, entre os níveis dessa hierarquia de subviews / superview. As mensagens de exibição geralmente são transmitidas da visualização de nível superior (a visualização padrão do sistema da janela do aplicativo) até as subviews (os objetos da view usados nas subviews).

### **2.1.3 Controller**

Segundo (KRASNER; POPE et al., 1988), os controllers contêm a interface entre a model e a view associados. De acordo com (DEACON, 2009), um controller é um objeto que permite manipular uma exibição. Simplificando um pouco demais, o controller lida com a entrada enquanto a view lida com a saída. Os controllers têm o maior conhecimento de plataformas e sistemas operacionais.

## 2.2 Metodologia Scrum

A ideia das metodologias ágeis é o enfoque nas pessoas e não em processos ou algoritmos. Além disso, existe a preocupação de gastar menos tempo com documentação e mais com a implementação. Uma característica das metodologias ágeis é que elas são adaptativas ao invés de serem preditivas (SOARES, 2004).

De acordo com (SINGH, 2008), metodologias Ágeis é um grupo de métodos de desenvolvimento de software baseados em desenvolvimento iterativo e incremental. As quatro principais características que são fundamentais para todas as metodologias ágeis são: planejamento adaptativo, desenvolvimento iterativo e evolutivo, resposta rápida e flexível para mudar e promover a comunicação.

A ideia principal da Scrum é que o desenvolvimento de softwares envolve muitas variáveis técnicas e do ambiente, com requisitos, recursos e tecnologia, que podem mudar durante o processo. Isto torna o processo de desenvolvimento imprevisível e complexo, requerendo flexibilidade para acompanhar as mudanças. O resultado do processo deve ser um software que é realmente útil para o cliente, segundo (SOARES, 2004).

Explicam (SILVA; HOENTSCHE; SILVA, 2009) que o Scrum prevê alguns artefatos, sendo os principais: Product Backlog e Sprint Backlog. O Product Backlog é o documento mutável definido pelo cliente no início do projeto, onde estão contidas suas características esperadas em forma de lista ordenada por prioridade. O Sprint Backlog é um documento gerado na reunião de planejamento (sprint planning) com a divisão do Product Backlog em pedaços de código para serem implementados.

Ainda de acordo com (SILVA; HOENTSCHE; SILVA, 2009), os processos do Scrum se dividem em várias iterações chamadas de sprints. O processo iterativo tem o objetivo de minimizar riscos, oferecendo aos usuários uma rápida avaliação do que está sendo construído. É estabelecido um tempo fixo para cada sprint, geralmente de 15 a trinta dias, onde o time deve trabalhar para atingir o objetivo

especificado para a iteração. Ao término do período, o time apresenta o que foi desenvolvido.

No primeiro dia do sprint é feita a reunião de planejamento do sprint (sprint planning). No primeiro planejamento o líder do processo, chamado de Scrum Master, se reúne com o cliente para definir o Product Backlog, ou lista de funcionalidades. Nessa reunião, a equipe deve definir quanto trabalho será necessário para cada tarefa baseado no tamanho do grupo e então é decidido que trabalho será realizado no sprint. Durante todos os dias do sprint são realizadas reuniões diárias (daily scrum) de quinze minutos para se avaliar o que foi feito no dia anterior, no dia atual e quais foram os impedimentos encontrados (SILVA; HOENTSCH; SILVA, 2009).

No último dia do sprint, é feita uma reunião com o cliente que poderá dar novo direcionamento ao projeto. Essa reunião é chamada de reunião de revisão (sprint review). Posteriormente à reunião de revisão é feita uma reunião de retrospectiva (sprint retrospective), onde será avaliado o que foi aprendido e os ajustes necessários ao projeto, visando a melhoria contínua. Iniciado a iteração seguinte, começa-se um novo planejamento, e assim sucessivamente. Para o Scrum, uma tarefa só é considerada pronta se tiver o potencial de entrar em produção assim que o cliente decidir. Ao término de cada iteração, a equipe deve apresentar ao cliente algo realmente funcional e não somente documentos (SILVA; HOENTSCH; SILVA, 2009).

### **2.3 PostgreSQL**

Segundo (MAGALHÃES et al., 2015), o PostgreSQL é um SGBD (sistema de gerenciamento de banco de dados) relacional e orientado a objetos, possui recursos em comum a sistemas gerenciadores de bancos de dados de grande porte, é versátil, seguro, gratuito e tem código aberto, se tornando uma ótima opção para

a comunidade acadêmica e empresas que querem alta confiabilidade e baixos custos de licenciamentos.

Segundo (DRAKE; WORSLEY, 2002), o PostgreSQL aborda os dados com um modelo objeto-relacional e é capaz de lidar com rotinas e regras complexas. Exemplos de seu funcionamento avançado são consultas SQL declarativas, controle de simultaneidade em várias versões, suporte a múltiplos usuários, transações, otimização de consulta, herança e arrays.

A tecnologia que o PostgreSQL usa para evitar bloqueios desnecessários, é o MVCC, ou Multi-Version Concurrency Control. O MVCC é considerado melhor que o bloqueio no nível da linha, porque uma leitura nunca é bloqueada por uma gravação. Em vez disso, o PostgreSQL mantém o controle de todas as transações realizadas pelos usuários do banco de dados. O PostgreSQL pode gerenciar registros sem que as pessoas esperem que os registros estejam disponíveis, (DRAKE; WORSLEY, 2002).

Por utilizar licença de uso BSD, o PostgreSQL pode livremente ser incorporado por aplicações pessoais e/ou comerciais, sem nenhum custo para o desenvolvedor ou fornecedor do software em questão, afirma (MILANI, 2008).

## 2.4 Java

Java é uma das linguagens de programação mais populares e possui alguns recursos notáveis, como a orientação a objetos e independente de plataforma. O compilador Java converte um programa Java em bytecode em vez de código binário dependente da máquina. O bytecode é executado pela Java Virtual Machine (JVM) e em qualquer máquina com qualquer sistema operacional. Portanto, o bytecode é independente da plataforma. Essa é a ideia por trás de "escreva uma vez, execute em qualquer lugar"(SHAHJAMALI, 2019).

O java foi usado com o apoio de outras duas tecnologias, o JPA (java persistence API) e o Play Framework.

### 2.4.1 JPA - Java Persistence API

De acordo com (SAMPAIO, 2011), a API de persistência do Java (JPA) serve para desacoplar as aplicações dos meios de persistência, como banco de dados relacionais, por exemplo, provendo uma camada uniforme que utiliza o padrão Domain Model, (CRUPI; MALKS; ALUR, 2001).

Afirma ainda (SAMPAIO, 2011) que todo mapeamento objeto-relacional é feito de maneira simples e transparente para o desenvolvedor, ficando este apenas com o trabalho de lidar com as entidades envolvidas.

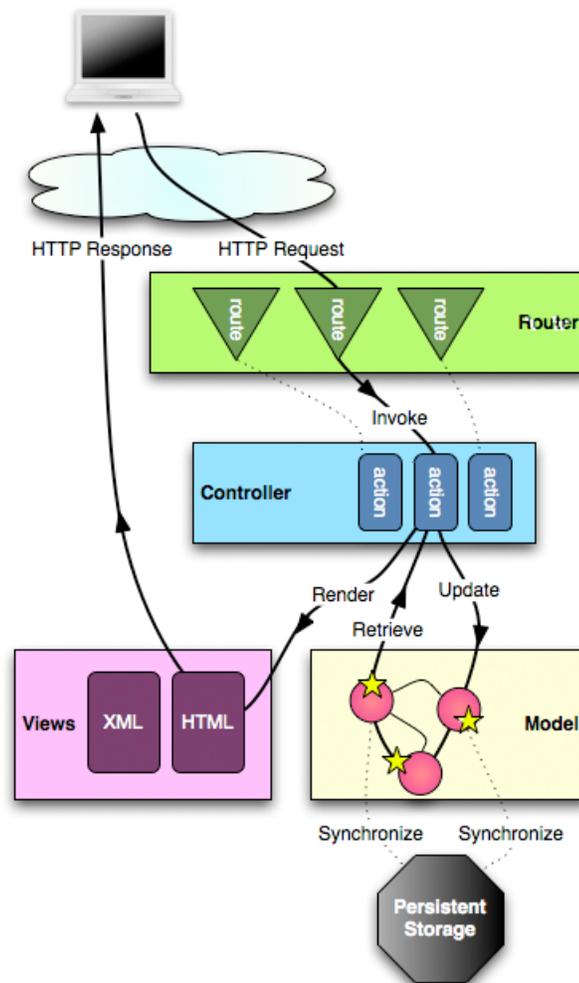
### 2.4.2 Play Framework

Play Framework é um dos muitos frameworks disponíveis para o desenvolvimento de aplicações para Internet que utilizam a linguagem e a plataforma Java. O projeto teve início no ano de 2007 pela empresa Zenexity (hoje Zengularity) e no ano de 2009 tornou-se um projeto de código fonte aberto, atualmente é mantido pelas empresas Zengularity e Typesafe, possui uma grande e ativa comunidade de usuários e é utilizada por empresas como LinkedIn e The Guardian, (SANTOS, 2015).

Ainda de acordo com (SANTOS, 2015) o Play foi projetado não para ser apenas um framework, mas sim uma solução completa para o desenvolvimento web, que envolve servidor HTTP (HyperText Transfer Protocol), cache, persistência de dados, sistema de build e vários outros componentes. A grande premissa do projeto é tornar simples e altamente produtivo o desenvolvimento de aplicações para Internet utilizando a plataforma Java.

O Play apresenta como arquitetura o padrão Model, View, Controller (MVC), Figura 2.1. Essa arquitetura é organizada em três camadas: modelo (Model), visualização (View) e o controlador (Controller), (SOUSA et al., 2015).

Figura 2.1 – Padrão MVC do Play Framework.



Fonte: (SOUSA et al., 2015)

## 2.5 AngularJS

Conforme (SESHADRI; GREEN, 2014), o AngularJS é um metaframework para SPA (Single Page Application, ou aplicação de página única). Com templating do lado do cliente e uso intenso de JavaScript, criar e manter uma aplicação pode se tornar maçante e trabalhoso. O AngularJS evita códigos redundantes e

faz o trabalho pesado, de modo que podemos focar exclusivamente no núcleo da aplicação.

O AngularJS ajuda a criar aplicativos da Web baseados em HTML, CSS, JavaScript. O AngularJS traz o recurso MVC para o aplicativo da Web, tornando-o mais modular e fácil de desenvolver, manter e testar. O AngularJS introduz tags adicionais, chamadas de diretivas. Essas diretivas têm o prefixo "ng-" e o objetivo da diretiva é vincular os dados à visualização / modelos por meio do controlador. Os controladores AngularJS são escritos em JavaScript que adicionam a lógica de negócios a visualizações que nada mais são que páginas HTML, (JADHAV; SAWANT; DESHMUKH, 2015).

### **3 METODOLOGIA**

Nesta seção são descritos os procedimentos metodológicos utilizados durante o período de estágio no desenvolvimento de software.

#### **3.1 Procedimentos Metodológicos**

Este trabalho foi realizado no período de abril a agosto de 2020, com o objetivo de documentar a experiência do discente no desenvolvimento de software, na empresa GT4W Consultoria e Serviços em Geoprocessamento. O discente atuou como desenvolvedor no desenvolvimento de um módulo do sistema de gestão de concessões (SGC), onde foi possível aprimorar e aplicar conhecimentos vistos durante toda a graduação, bem como adquirir conhecimento sobre processos e ferramentas que agilizam o desenvolvimento de software.

#### **3.2 Desenvolvimento do software**

O sistema foi dividido em módulos e funcionalidades, que foram desenvolvidas ao longo do andamento do projeto. A prioridade para se desenvolver uma atividade é definida pelo Product Owner e separadas em cada sprint de acordo com seu escopo. Dentro do desenvolvimento do Software, também foram feitas correções de erros encontrados pelo cliente ou apontados pelo Product Owner.

Foi desenvolvido um módulo para um sistema legado da empresa. Esse sistema conta, ao todo, com sete módulos: Gestão de Florestas Públicas, Gestão de Licitação, Gestão de Concessionários, Gestão de Contratos, Central de Comunicação, Central do Administrador e Gestão de Indicador A2. Os módulos podem ser acessados por dois perfis (concessionário e cliente). Apenas três módulos são acessíveis para ambos os perfis: Gestão de Indicador A2, Gestão de Contratos e Central de Comunicação.

O módulo Gestão de Indicador A2 é o módulo responsável por gerenciar investimentos em infraestrutura e serviços para a comunidade local. Indicadores são cadastrados pelo perfil de concessionário e avaliados pelo perfil do cliente. O sistema deve ser capaz de processar os dados dos indicadores para gerar planilhas e gráficos.

No processo de desenvolvimento de software, os desenvolvedores atuam em duas áreas do desenvolvimento: o front-end e o back-end, o front-end é a parte relacionada ao usuário da aplicação, já o back-end é a parte relacionada ao servidor da aplicação, sendo responsável pelo processamento de dados. No sistema de gestão de concessões, as tecnologias utilizadas foram HTML, CSS e JavaScript (AngularJS) no front-end. O back-end foi feito em Java (Play Framework) e PostgreSQL. O estagiário participou tanto do desenvolvimento front-end quanto do back-end, executando atividades como mapeamento de entidades, criação de serviços e criação de páginas web.

### **3.3 Processos**

A empresa adota a metodologia Scrum da seguinte forma. Os projetos são divididos em sprints de 15 dias. O sprint representa um conjunto de funcionalidades que deve ser executado. As funcionalidades são divididas em atividades menores que podem ser desenvolvidas separadamente por cada membro da equipe de forma rápida. No início de cada sprint a equipe se reúne para decidir quais as funcionalidades serão desenvolvidas em um período de 15 dias e entregues ao seu final. Funcionalidades, das quais atividades não foram finalizadas, são incrementadas em sprints futuras.

Durante o desenvolvimento, a equipe faz duas reuniões diárias com duração de no máximo, dez minutos para disseminar conhecimento sobre o andamento do projeto e identificar possíveis impedimentos. Em cada reunião diária, os membros devem citar o que foi desenvolvido desde a última reunião, quais atividades

desenvolver até a próxima reunião e reportar caso haja algum impedimento em suas atividades. Toda a equipe tem acesso ao estado atual de todas as atividades, que são mantidas em um quadro digital. No término de cada sprint, é feita uma reunião para apresentar as funcionalidades desenvolvidas ao cliente, e também discutir seus pontos negativos e positivos.

Todo código deve seguir o modelo MVC (Model View Controller), bem como padrões pré-estabelecidos pela equipe. Atividades em que a etapa de programação foi finalizada devem passar por uma revisão de código, feita por todos os outros desenvolvedores que não participaram do desenvolvimento da mesma, para depois serem testadas e por fim entregues ao cliente.



## **4 RESULTADOS**

Os resultados obtidos pelo discente foram de grande importância para o módulo desenvolvido, o que é possível visualizar por meio das atividades desenvolvidas durante o estágio. O discente pôde atuar no mapeamento de entidades, criação de serviços e criação de páginas web.

### **4.1 Mapeamento de Entidades**

O mapeamento é feito de acordo com os requisitos fornecidos pelo cliente, a partir do mapeamento o discente foi capaz de criar relações no banco de dados com PostgreSQL, responsáveis pelo armazenamento dos dados do sistema, Figura 4.1. Todas as relações foram validadas pelo Administrador de Banco de Dados da empresa antes de serem executadas no sistema. Além disso, o discente também participou na modelagem por meio de classes Java usando a JPA, Figura 4.2. Todo o código gerado neste processo era revisado pelos outros desenvolvedores para garantir que o código seguisse os padrões pré-definidos pela equipe.

### **4.2 Criação de Serviços**

A partir do mapeamento no banco de dados e da modelagem de classes, o discente atuou na criação de serviços Java para executar operações e funcionalidades do sistema, Figura 4.3. Esses serviços seguem um conjunto de regras pré-definidas, e também devem validar os dados para garantir a integridade de todo o sistema. Tal como no mapeamento, todo o código resultante dessa atividade passa por uma revisão feita pelos outros desenvolvedores da equipe.

### **4.3 Criação de Páginas Web**

O desenvolvimento contava com protótipos de alta fidelidade, que auxiliaram o discente na criação de páginas web, utilizando as tecnologias HTML, CSS

Figura 4.1 – Exemplo de relação em banco de dados PostgreSQL.



Fonte: Próprio autor.

e JavaScript (com o framework AngularJS), Figura 4.4. As páginas do sistema normalmente consomem dados do back-end. Os dados são validados e retornam mensagens de erro ou confirmação de acordo com as regras do sistema, permitindo que o usuário interaja com a aplicação por meio de um navegador compatível.

#### 4.4 Experiências

As atividades desenvolvidas sobre diversas ferramentas, foram integradas em um único sistema e a integração dessas ferramentas gerou grande conhecimento, pois, durante a graduação as ferramentas eram abordadas na maioria das vezes de forma separada. Como por exemplo, a integração entre as tabelas do PostgreSQL com as classes Java por meio da JPA (Java Persistence API).

O conhecimento das frameworks utilizadas para agilizar o desenvolvimento, foram de grande relevância, pois tais frameworks são amplamente usadas

Figura 4.2 – Exemplo de modelagem de classe Java usando JPA.

```

@Entity
@Table(schema=Config.SCHEMA_GESTAO_CONCESSOES, name="historico_investimento")
public class HistoricoInvestimento extends GenericModel {

    @Id
    @Column(name="id")
    @GeneratedValue(
        strategy=GenerationType.SEQUENCE,
        generator="historico_investimento_id_seq"
    )
    @SequenceGenerator(
        name="historico_investimento_id_seq",
        sequenceName= Config.SCHEMA_GESTAO_CONCESSOES + ".historico_investimento_id_seq",
        allocationSize=1
    )
    public Integer id;

    @Column(name="data_cadastro")
    public Date dataCadastro;

    public String detalhamento;

    @OneToOne
    @JoinColumn(
        name = "id_tipo_acao_historico_investimento",
        referencedColumnName = "id"
    )
    public TipoAcaoHistoricoInvestimento tipoAcao;

    @ManyToOne
    @JoinColumn(
        name = "id_investimento",
        referencedColumnName = "id"
    )
    public Investimento investimento;

    @OneToOne
    @JoinColumn(name = "id_conversa", referencedColumnName = "id")
    public Conversa conversa;
}

```

Fonte: Próprio autor.

Figura 4.3 – Exemplo de serviço Java.

```

public static void findById(Integer idInvestimento) {
    Secure.isAuthenticated(secur.Permissao.VISUALIZAR_INVESTIMENTO);

    if (idInvestimento == null || idInvestimento < 1) {
        throw new ValidationException().userMessage(Messages.get("parametros.nao.encontrado"));
    }

    Investimento investimento = Investimento.findById(idInvestimento);

    if (investimento == null) {
        throw new GestaoConcessaoException(
            Messages
                .get("aplicacao.nao.cadastrada"))
            .userMessage(Messages.get("aplicacao.nao.cadastrada"));
    }

    investimento.historicos = HistoricoInvestimento
        .find("investimento.id = :idInvestimento ORDER BY dataCadastro asc")
        .setParameter("idInvestimento", investimento.id).fetch();

    InformacaoInvestimento informacaoInvestimento = InformacaoInvestimento.getDados(investimento);

    renderJSON(informacaoInvestimento, InvestimentoSerializer.visualizar);
}

```

Fonte: Próprio autor.

no mercado de trabalho e raramente vistas no curso de graduação. Por ser um projeto legado eram usadas frameworks em versões mais antigas ou até mesmo descontinuadas, o que gerou dificuldade para o discente se familiarizar com as mesmas no começo do desenvolvimento. A empresa poderia manter suas fra-

Figura 4.4 – Exemplo de página Web desenvolvida em AngularJS.



The screenshot shows a web page with a header titled "Histórico" and a table with three columns: "Motivo", "Data da Ação", and "Detalhamento". The table contains seven rows of data, each representing a different action taken on an application.

Motivo	Data da Ação	Detalhamento
Cadastro da aplicação	16/03/2014	Cadastro da aplicação.
Solicitação de correção	30/03/2014	<Justificativa informada pelo Ideflor>
Envio de mensagem	16/04/2014	Mensagem enviada para concessionário. <a href="#">Link</a> para informações da mensagem.
Edição da aplicação	16/04/2014	Edição da aplicação.
Aprovação	16/04/2014	Aprovação da aplicação.
Reprovação	16/04/2014	<Justificativa informada pelo Ideflor>

Fonte: Próprio autor.

meworks sempre atualizadas para que novos membros em projetos legado não tenham dificuldade em desenvolver.

A utilização diária do Scrum pôde contribuir com o conhecimento prático da ferramenta, visto que, ferramentas de gerenciamento ágil como o Scrum são abordadas na graduação apenas de forma teórica. Foi possível encontrar algumas falhas no processo utilizado, como correções solicitadas pelo cliente no decorrer do desenvolvimento que atrasavam a entrega das funcionalidades planejadas. Uma forma de mitigar tal falha seria manter um contato mais próximo do cliente.

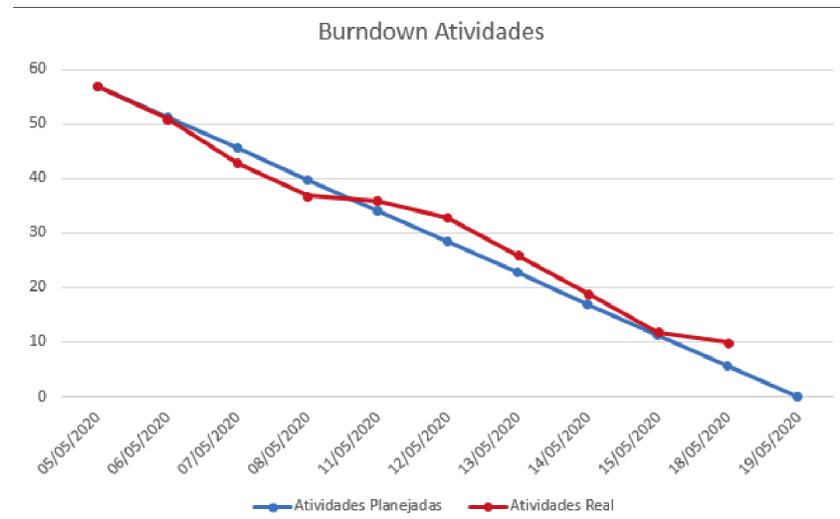
As revisões de código feitas ao final do desenvolvimento das atividades permitiram ao estagiário assimilar conhecimentos em reutilização de código, boas práticas de programação e uso correto dos padrões de projeto.

#### 4.5 Desempenho

O Desempenho de toda a equipe é medido diariamente de acordo com a quantidade de atividades desenvolvidas durante o dia, que pode ser visto por meio de um gráfico (burndown) gerado pelo scrum master ao final de cada dia. O discente bem como toda a equipe obtiveram um bom desempenho durante todo

o desenvolvimento do projeto, como pode ser visto na Figura 4.5, que mostra o gráfico de dias da sprint x atividades a serem desenvolvidas.

Figura 4.5 – Exemplo de gráfico de Burndown.



Fonte: Próprio autor.



## 5 CONCLUSÃO

Para discentes que pretendem ingressar no mercado de trabalho o estágio é de suma importância. Os conhecimentos obtidos durante o curso de graduação foram fundamentais para o desenvolvimento das atividades realizadas no estágio. As principais disciplinas foram as ligadas à programação e tecnologias utilizadas, Algoritmo e Estruturas de Dados (I, II e III), Programação Orientada a Objetos, Redes de Computadores, Sistemas Distribuídos e Banco de Dados (I e II). Outras disciplinas que auxiliaram por meio do estudo de conceitos ligados ao desenvolvimento de software, como Interação Humano-computador, Engenharia de Software (I e II) e Gerência de Projetos de Software.

O objetivo deste trabalho foi mostrar o dia-a-dia do desenvolvimento de software como desenvolvedor. Utilizando processos e ferramentas que agilizam o desenvolvimento.

Por fim, conclui-se que este trabalho foi de grande relevância para o estagiário pois as atividades desenvolvidas durante o estágio agregaram vários conhecimentos, e também para a empresa, o que pode ser visto a partir dos comentários dos colaboradores da empresa que participaram do desenvolvimento.



## REFERÊNCIAS

- BURBECK, S. Applications programming in smalltalk-80 (tm): How to use model-view-controller (mvc). **Smalltalk-80 v2**, v. 5, p. 1–11, 1992.
- CRUPI, J.; MALKS, D.; ALUR, D. **Core J2EE Patterns**. [S.l.]: Gulf Professional Publishing, 2001.
- DEACON, J. Model-view-controller (mvc) architecture. **Online**[[Citado em: 10 de março de 2006.] <http://www.jdl.co.uk/briefings/MVC.pdf>, 2009.
- DRAKE, J. D.; WORSLEY, J. C. **Practical PostgreSQL**. [S.l.]: "O'Reilly Media, Inc.", 2002.
- JADHAV, M. A.; SAWANT, B. R.; DESHMUKH, A. Single page application using angularjs. **International Journal of Computer Science and Information Technologies**, Citeseer, v. 6, n. 3, p. 2876–2879, 2015.
- KRASNER, G. E.; POPE, S. T. et al. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. **Journal of object oriented programming**, v. 1, n. 3, p. 26–49, 1988.
- MAGALHÃES, L. B. et al. Análise comparativa dos algoritmos de otimização de consultas do postgresql. In: **Colloquium Exactarum. ISSN: 2178-8332**. [S.l.: s.n.], 2015. v. 7, n. 1, p. 01–21.
- MILANI, A. **PostgreSQL-Guia do Programador**. [S.l.]: Novatec Editora, 2008.
- SAMPAIO, C. **Java Enterprise Edition 6-Desenvolvendo Aplicações Corporativas**. [S.l.]: Brasport, 2011.
- SANTOS, R. D. d. S. Desenvolvimento de um sistema web para monitoramento de sensores. Universidade Tecnológica Federal do Paraná, 2015.
- SESHADRI, S.; GREEN, B. **Desenvolvendo com AngularJS: aumento de produtividade com aplicações web estruturadas**. [S.l.]: Novatec Editora, 2014.
- SHAHJAMALI, F. **An Empirical Analysis of Java Language Use in Open Source Applications**. Tese (Doutorado), 2019.
- SILVA, F.; HOENTSCHE, S. C.; SILVA, L. Uma análise das metodologias ágeis fdd e scrum sob a perspectiva do modelo de qualidade mps. br. **Scientia Plena**, v. 5, n. 12, 2009.
- SINGH, M. U-scrum: An agile methodology for promoting usability. In: IEEE. **Agile 2008 Conference**. [S.l.], 2008. p. 555–560.

SOARES, M. dos S. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação**, v. 3, n. 1, 2004.

SOUSA, G. F. C. d. et al. Webdiet: um estudo de caso de uso do uso de templates em play framework. Insitituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, 2015.